



Guía del usuario

# AWS Modernización de mainframe



# AWS Modernización de mainframe: Guía del usuario

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

---

# Table of Contents

¿Qué es la modernización del AWS mainframe? .....	1
Características de la modernización del AWS mainframe .....	2
Patrones .....	3
¿Cómo empezar con la modernización del mainframe AWS .....	3
Servicios relacionados .....	4
Acceder a la modernización del AWS mainframe .....	5
¿Es la primera vez que utiliza Mainframe Modernization? AWS .....	5
Precios .....	5
Configuración de la modernización del AWS mainframe .....	6
Inscríbese en una Cuenta de AWS .....	6
Creación de un usuario con acceso administrativo .....	6
Introducción .....	9
Tutorial: Tiempo de ejecución gestionado para AWS Blu Age .....	9
Requisitos previos .....	10
Paso 1: Cargar la aplicación de demostración .....	10
Paso 2: Crear la definición de aplicación .....	10
Paso 3: Crear un entorno de tiempo de ejecución .....	11
Paso 4: Crear una aplicación .....	16
Paso 5: Implementar una aplicación .....	18
Paso 6: Iniciar una aplicación .....	21
Paso 7: Acceder a la aplicación .....	21
Paso 8: Probar la aplicación .....	22
Eliminar recursos .....	24
Tutorial: Tiempo de ejecución administrado para Micro Focus .....	24
Requisitos previos .....	25
Paso 1: Crear y cargar un bucket de Amazon S3 .....	25
Paso 2: Crear y configurar una base de datos .....	26
Paso 3: Crear y configurar un AWS KMS key .....	29
Paso 4: Crear y configurar un secreto de AWS Secrets Manager base de datos .....	30
Paso 5: Crear un entorno de ejecución .....	31
Paso 6: Crear una aplicación .....	37
Paso 7: implementar una aplicación .....	43
Paso 8: Importar conjuntos de datos .....	45
Paso 9: Iniciar una aplicación .....	51

Paso 10: Conectarse a la CardDemo aplicación CICS .....	52
Eliminar recursos .....	59
Sigüientes pasos .....	60
Enfoque de modernización .....	61
Fase de evaluación .....	61
Fase de movilización .....	62
Fase de migración y modernización .....	62
Fase de operación y optimización .....	63
Conceptos .....	64
Aplicación .....	64
Definición de aplicación .....	64
Trabajo por lotes .....	65
Configuración .....	66
Conjunto de datos .....	66
Entorno .....	66
Modernización de la unidad central .....	66
Viaje migratorio .....	66
Punto de montaje .....	66
Refactorización automatizada .....	67
Recreación de plataformas .....	67
Recurso .....	67
Motor de tiempo de ejecución .....	67
AWS Refactorización de Blu Age .....	68
AWS Notas de lanzamiento de Blu Age .....	69
Notas de la versión 4.1.0 .....	70
Runtime, versión 4.1.0 .....	71
Herramientas de modernización, versión 4.1.0 .....	76
Notas de la versión 4.0.0 .....	78
Runtime, versión 4.0.0 .....	79
Herramientas de modernización, versión 4.0.0 .....	84
Notas de la versión 3.10.0 .....	87
Runtime, versión 3.10.0 .....	88
Herramientas de modernización, versión 3.10.0 .....	91
Notas de la versión 3.9.0 .....	92
Runtime, versión 3.9.0 .....	93
Herramientas de modernización, versión 3.9.0 .....	98

Notas de la versión 3.8.0 .....	101
Runtime, versión 3.8.0 .....	102
Herramientas de modernización, versión 3.8.0 .....	105
Notas de la versión 3.7.0 .....	107
Runtime, versión 3.7.0 .....	108
Herramientas de modernización, versión 3.7.0 .....	110
Notas de la versión 3.6.0 .....	113
Runtime, versión 3.6.0 .....	113
Herramientas de modernización, versión 3.6.0 .....	116
Notas de la versión 3.5.0 .....	119
Runtime, versión 3.5.0 .....	120
Herramientas de modernización, versión 3.5.0 .....	123
AWS Instrucciones de actualización de Blu Age .....	126
Migración de la versión 3.10.0 a la 4.0.0 .....	126
AWS Conceptos de Blu Age Runtime .....	128
Arquitectura de alto nivel .....	128
Estructura de una aplicación modernizada .....	133
Data Simplifier .....	170
AWS Configuración de Blu Age Runtime .....	178
Principios básicos de la configuración de aplicaciones .....	179
Prioridad de la aplicación .....	181
JNDI para bases de datos .....	181
Uso de secretos AWS .....	181
Otros archivos (groovy, sql, etc.) .....	187
Aplicación web adicional .....	188
Habilitación de propiedades .....	189
Configuración de la seguridad para las aplicaciones de Gapwalk .....	243
AWS API de ejecución de Blu Age .....	259
Creación de direcciones URL .....	259
Aplicación Gapwalk .....	260
Puntos de conexión REST de la consola de aplicaciones de Blusam .....	280
Consola de la aplicación JICS .....	302
Estructuras de base de datos .....	323
AWS Configuración de Blu Age Runtime (no gestionado) .....	332
AWS Requisitos previos de tiempo de ejecución de Blu Age .....	333
AWS Incorporación de Blu Age Runtime .....	333

Requisitos de configuración de infraestructura .....	339
Implementación en Amazon ECS gestionada por AWS Fargate .....	345
Implementación en Amazon EC2 .....	354
Pruebe la aplicación PlanetsDemo .....	368
Modificación del código fuente con el IDE para desarrolladores de Blu Age .....	372
Tutorial: Configurar AppStream 2.0 para el IDE de desarrolladores de AWS Blu Age .....	372
Tutorial: Utilice AWS Blu Age Developer en AppStream 2.0 .....	378
Redefinición de plataformas con Micro Focus .....	395
Configuración de tiempo de ejecución de Micro Focus (en Amazon EC2) .....	395
Requisitos previos .....	396
Cree puntos de conexión de Amazon VPC para Amazon S3 .....	396
Solicitud de actualización de la lista de permitidos de la cuenta .....	399
Crear el rol AWS Identity and Access Management .....	400
Concesión de permisos necesarios con License Manager .....	407
Suscríbase a Amazon Machine Images .....	408
Lance una instancia de Micro Focus para la modernización de AWS mainframe .....	411
Subred o VPC sin acceso a Internet .....	417
Solución de problemas de licencia .....	424
Tutorial: Configuración de la AppStream versión 2.0 para Enterprise Analyzer y Enterprise Developer .....	426
Requisitos previos .....	427
Paso 1: Obtenga las imágenes AppStream 2.0 .....	428
Paso 2: Cree la pila con la plantilla AWS CloudFormation .....	428
Paso 3: Crea un usuario en la AppStream versión 2.0 .....	431
Paso 4: Inicie sesión en AppStream 2.0 .....	432
Paso 5: Verificar los buckets de Amazon S3 (opcional) .....	434
Sigüentes pasos .....	435
Eliminar recursos .....	435
Configuración de Enterprise Analyzer .....	435
Contenido de la imagen .....	436
Requisitos previos .....	437
Paso 1: Configuración .....	438
Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows .....	438
Paso 3: Crear un origen de ODBC para la instancia de Amazon RDS .....	439
Sesiones posteriores .....	441
Solución de problemas de conexión de espacio de trabajo .....	442

Eliminar recursos .....	446
Configuración de Enterprise Developer .....	446
Contenido de la imagen .....	447
Requisitos previos .....	448
Paso 1: Configuración por parte de usuarios individuales de Enterprise Developer .....	448
Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows (opcional) .....	449
Paso 3: Clonar el repositorio .....	450
Sesiones posteriores .....	451
Eliminar recursos .....	451
Configure la automatización AppStream 2.0 .....	452
Configuración de la automatización al inicio de la sesión .....	452
Configuración de la automatización al final de la sesión .....	453
Ver los conjuntos de datos como tablas .....	453
Requisitos previos .....	454
Paso 1: Configurar la conexión ODBC al almacén de datos de Micro Focus (base de datos de Amazon RDS) .....	454
Paso 2: Crear el archivo MFDBFH.cfg .....	457
Paso 3: Crear un archivo de estructura (STR) para el diseño del cuaderno .....	457
Paso 4: Crear una vista de base de datos con el archivo de estructura (STR) .....	460
Paso 5: Ver los conjuntos de datos de Micro Focus como tablas y columnas .....	460
Uso de plantillas con Enterprise Developer .....	461
Caso de uso 1: Uso de la plantilla de proyecto de COBOL que contiene los componentes de origen .....	462
Caso de uso 2: Uso de la plantilla de proyecto de COBOL sin los componentes de origen ..	464
Caso de uso 3: Uso del proyecto de COBOL predefinido que enlaza con las carpetas de origen .....	466
Uso de la plantilla JSON de definición de región .....	469
Tutorial: Configurar la compilación de la BankDemo muestra .....	472
Requisitos previos .....	473
Paso 1: Crear un bucket de Amazon S3 .....	474
Paso 2: Crear el archivo de especificaciones de compilación .....	474
Paso 3: Cargar los archivos de origen .....	476
Paso 4: Crear políticas de IAM .....	476
Paso 5: Crear un rol de IAM .....	478
Paso 6: Asociar las políticas de IAM al rol de IAM. ....	479
Paso 7: Crea el CodeBuild proyecto .....	480

Paso 8: Iniciar la compilación .....	481
Paso 9: Descargar los artefactos de salida .....	481
Eliminar recursos .....	482
Tutorial: Configuración de una canalización de CI/CD para su uso con Micro Focus Enterprise Developer .....	482
Requisitos previos .....	483
Creación de una infraestructura básica de canalización de CI/CD .....	485
Cree un AWS CodeCommit repositorio y una canalización de CI/CD .....	489
Creación de Enterprise Developer 2.0 AppStream .....	494
Configuración y prueba de Enterprise Developer .....	494
Ejercicio 1: Mejorar el cálculo de los préstamos en la aplicación de BANKDEMO .....	500
Ejercicio 2: Extraiga el cálculo del préstamo en BankDemo la solicitud .....	504
Limpieza de recursos .....	508
Utilidades por lotes .....	508
Ubicación binaria .....	509
Utilidad por lotes M2SFTP .....	509
Utilidad por lotes M2WAIT .....	516
Utilidad TXT2PDF por lotes .....	518
Utilidad por lotes M2DFUTIL .....	524
Utilidad por lotes M2RUNCMD .....	531
Replicación de datos con Precisely .....	536
Requisitos previos .....	536
Suscripción a imágenes de máquina de Amazon .....	536
Lance la replicación de datos AWS de Mainframe Modernization con Precisely .....	537
Creación de una política de IAM .....	538
Creación de un rol de IAM .....	539
Asociación del rol de IAM a la instancia de Amazon EC2 .....	539
Integración de Charon .....	540
Introducción a Charon-SSP .....	540
Sistemas operativos huéspedes compatibles .....	542
Requisitos previos de la instancia en la nube de Charon-SSP .....	543
Requisitos previos para las instancias .....	544
Creación y configuración de una instancia en la nube para Charon (nueva AWS GUI) .....	546
Requisitos previos generales .....	546
AWS Management Console Utilizándolo para lanzar una nueva instancia .....	547
Redefinición de plataforma con NTT DATA .....	553



Requisitos previos .....	553
Suscripción a imágenes de máquina de Amazon .....	553
Lance la replataforma AWS de modernización del mainframe con una instancia de NTT DATA .....	554
Introducción al uso de NTT Data .....	555
Aplicaciones .....	557
Crear una aplicación .....	558
Crear una aplicación .....	558
Implementar una aplicación .....	559
Implementar una aplicación .....	559
Actualización de una aplicación .....	560
Actualización de una aplicación .....	560
Eliminar una aplicación de un entorno. ....	561
Eliminar una aplicación de un entorno. ....	561
Eliminar una aplicación .....	562
Eliminar una aplicación .....	562
Envío de trabajos por lotes para las aplicaciones .....	562
Envío de un trabajo por lotes. ....	563
Reinicie un trabajo por lotes .....	564
Importe conjuntos de datos para aplicaciones .....	565
Importar un conjunto de datos .....	565
Gestión de las transacciones de las aplicaciones .....	566
Gestión de las transacciones de las aplicaciones .....	566
Cree AWS recursos para una aplicación migrada .....	567
Permisos necesarios .....	568
Bucket de Amazon S3 .....	568
Database .....	569
AWS Key Management Service clave .....	569
AWS Secrets Manager secreto .....	570
Configuración de la aplicación administrada. ....	571
Estructura de las aplicaciones gestionadas de AWS Blu Age .....	571
Configuración del acceso a las utilidades para las aplicaciones administradas .....	573
Configuración de propiedades adicionales .....	582
Referencia de definición de aplicaciones .....	602
Sección de cabecera general .....	603
Descripción general de la sección de definiciones .....	604

AWS Ejemplo de definición de aplicación de Blu Age .....	605
AWS Detalles de la definición de Blu Age .....	606
Definición de la aplicación Micro Focus .....	611
Detalles de definición de Micro Focus .....	613
Referencia de definición de conjuntos de datos .....	621
Propiedades comunes .....	621
Ejemplo de formato de solicitud de conjunto de datos para VSAM .....	623
Ejemplo de formato de solicitud de conjunto de datos para base GDG .....	626
Ejemplo de formato de solicitud de conjunto de datos para las generaciones PS o GDG .....	627
Ejemplo de formato de solicitud de conjunto de datos para PO .....	628
Entornos de tiempo de ejecución administrados .....	630
Creación de un entorno de tiempo de ejecución .....	631
Creación de un entorno de tiempo de ejecución .....	631
Actualizar un entorno de tiempo de ejecución .....	633
Actualizar un entorno de tiempo de ejecución .....	634
Periodo de mantenimiento .....	634
Detener un entorno de tiempo de ejecución .....	636
Detener un entorno de tiempo de ejecución .....	636
Reiniciar un entorno de tiempo de ejecución .....	637
Reiniciar un entorno de tiempo de ejecución .....	637
Eliminar un entorno de tiempo de ejecución .....	638
Eliminar un entorno de tiempo de ejecución .....	638
Application Testing .....	640
¿Qué es Application Testing? .....	640
¿Es la primera vez que usa Application Testing? .....	641
Ventajas de Application Testing .....	641
Integración con AWS CloudFormation .....	642
Cómo funciona Application Testing .....	643
Servicios relacionados .....	4
Acceso a Application Testing .....	644
Precios de Application Testing .....	645
Conceptos de Application Testing .....	645
Caso de prueba .....	646
Escenario de prueba .....	646
Proyecto de prueba .....	647
Condición inicial .....	647

Registro (captura) .....	647
Reproducción .....	648
Comparación .....	648
Comparación de bases de datos .....	648
Comparación de conjuntos de datos .....	649
Estado de comparación .....	649
Reglas de equivalencia .....	650
Comparación de conjuntos de datos de estado final .....	650
Comparaciones de bases de datos estado-progreso .....	650
Equivalencia funcional (FE) .....	651
Comparaciones de pantallas 3270 en línea .....	651
Registro .....	651
Datos de reproducción .....	651
Datos de referencia .....	651
Registro, reproducción y comparación .....	652
Diferencias .....	653
Equivalencias .....	653
Aplicación de origen .....	653
Aplicación de destino .....	653
Tutorial: Configuración CardDemo .....	653
Requisitos previos .....	654
Paso 1: Prepárese para la configuración CardDemo .....	654
Paso 2: Crear todos los recursos necesarios .....	655
Paso 3: Implementar e iniciar la aplicación .....	656
Paso 4: Importar los datos iniciales .....	656
Paso 5: Conectarse a la CardDemo aplicación .....	657
Tutorial: Reproduzca y compare en Blu Age usando AWS CardDemo .....	658
Paso 1: Obtenga la imagen de máquina Amazon EC2 de AWS Blu Age (AMI) .....	658
Paso 2: Iniciar una instancia de Amazon EC2 con la AMI de AWS Blu Age .....	659
Paso 3: Cargue CardDemo los archivos dependientes a S3 .....	660
Paso 4: Cargue las bases de datos e inicialice la aplicación CardDemo .....	660
Paso 5: Inicie el tiempo de ejecución de AWS Blu Age CloudFormation .....	663
Paso 6: Probar la instancia Amazon EC2 de AWS Blu Age .....	666
Paso 7: Validar que los pasos anteriores se hayan completado correctamente .....	667
Paso 8. Crear la condición inicial .....	667
Paso 9: Crear el caso de prueba .....	668

Paso 10: Crear un escenario de prueba .....	668
Paso 11: Registrar su escenario de prueba .....	669
Paso 12: Reproducir y comparar .....	669
Páginas de código de conjuntos de datos compatibles .....	670
File Transfer .....	681
¿Qué es File Transfer? .....	681
Ventajas de AWS Mainframe Modernization File Transfer .....	681
Cómo funciona AWS Mainframe Modernization File Transfer .....	682
Instalación de un agente de File Transfer .....	683
Paso 1: Iniciar sesión en la ISPF .....	683
Paso 2: Asignar un conjunto de datos para el z/FS .....	683
Paso 3: Formatear el conjunto de datos como z/FS .....	684
Paso 4: Definir el sistema de archivos para z/OS .....	684
Paso 5: Montar el sistema de archivos .....	684
Paso 6: Verificar el montaje .....	685
Paso 7: Introducir los OMV .....	685
Paso 8: Definir la variable de entorno del directorio de instalación del agente .....	685
Paso 9: Definir la variable de entorno del directorio de trabajo .....	685
Paso 10: Creación del directorio de trabajo .....	685
Paso 11: Copie el paquete tar de modernización de AWS mainframe al directorio de trabajo de z/OS .....	686
Paso 12: Asumir el usuario raíz .....	686
Configuración de permisos y STC .....	687
Creación de un usuario de IAM con credenciales de acceso a largo plazo .....	688
Creación de un rol de IAM para que lo asuma el agente .....	688
Configuración del agente .....	690
Puntos de conexión de transferencia de datos .....	692
Creación de un punto de conexión de transferencia de datos .....	692
Tareas de transferencia .....	694
Creación de tareas de transferencia .....	694
Visualización de las tareas de transferencia .....	697
Tutorial: Introducción a File Transfer .....	698
Información general .....	698
Paso 1: Transfiera el paquete tar de los archivos binarios del agente AWS a la partición lógica del mainframe .....	699
Paso 2: Configurar el agente de File Transfer en el mainframe de origen .....	699

Paso 3: Creación de un punto de conexión de transferencia de datos .....	699
Paso 4: Crear una tarea de transferencia .....	699
Paso 5: Ver el progreso de la tarea de transferencia .....	699
Páginas de códigos fuente y destino compatibles .....	699
Tipos de conjuntos de datos de mainframe .....	700
Páginas de códigos compatibles .....	700
Seguridad .....	702
Protección de datos .....	703
Datos que recopila AWS Mainframe Modernization .....	704
Cifrado de datos en reposo para el servicio de AWS modernización de mainframe .....	705
Cómo utiliza AWS Mainframe Modernization las subvenciones en AWS KMS .....	708
Crear una clave administrada por el cliente .....	710
Especificar una clave administrada por el cliente para AWS Mainframe Modernization .....	711
AWS Contexto de cifrado de modernización del mainframe .....	712
Supervisión de las claves de cifrado .....	714
Más información .....	729
Cifrado en tránsito .....	729
Identity and Access Management .....	730
Público .....	730
Autenticación con identidades .....	731
Administración de acceso mediante políticas .....	735
Cómo funciona la modernización AWS del mainframe con la IAM .....	738
Ejemplos de políticas basadas en identidades .....	751
Resolución de problemas .....	754
Uso de roles vinculados a servicios .....	756
Validación de conformidad .....	760
Resiliencia .....	761
Seguridad de la infraestructura .....	761
AWS PrivateLink .....	762
Consideraciones .....	762
Creación de un punto de conexión de interfaz .....	762
Creación de una política de punto de conexión .....	763
Supervisión .....	765
Monitorización con CloudWatch .....	765
Métricas del entorno de tiempo de ejecución .....	766
Métricas de aplicación .....	767

Dimensiones .....	772
CloudTrail registra .....	773
AWS Información sobre la modernización de mainframes en CloudTrail .....	773
Descripción de las entradas del archivo de registro de AWS modernización de mainframe ..	774
Resolución de problemas .....	777
Error: Time out while waiting for dataset name to be unlocked .....	777
Cómo se produce este error .....	777
¿Cómo sabe si esta es su situación? .....	778
¿Qué puede hacer? .....	778
Forzar la liberación del bloqueo .....	778
Configuración del mecanismo de reparación automática de Blusam .....	779
Administrador de bloqueos de Blusam .....	780
Cannot access an application's URL .....	780
Cómo se produce este error .....	781
¿Cómo sabe si esta es su situación? .....	781
¿Qué puede hacer? .....	781
AWS Blu Insights no se abre desde la consola .....	782
Cómo se produce este error .....	782
¿Qué puede hacer? .....	782
Historial de documentos .....	784
.....	dcclxxxvii

# ¿Qué es la modernización del AWS mainframe?

AWS La modernización de mainframe le ayuda a modernizar sus aplicaciones de mainframe para convertirlas en entornos de tiempo de ejecución gestionados. AWS Ofrece herramientas y recursos para ayudarle a planificar e implementar la migración y modernización. Puede analizar sus aplicaciones de unidad central existentes, desarrollarlas o actualizarlas mediante COBOL o PL/I e implementar un proceso automatizado para la integración y la entrega continuas (CI/CD) de las aplicaciones. Puede elegir entre patrones automatizados de refactorización y de redefinición de plataforma, en función de las necesidades de sus clientes. Si es un consultor que ayuda a un cliente a migrar sus cargas de trabajo de mainframe, puede utilizar las herramientas de modernización de AWS mainframe en todas las fases del proceso de migración y modernización, desde la planificación inicial hasta las operaciones en la nube posteriores a la migración.

Puede usar AWS Mainframe Modernization para crear y administrar de manera eficiente el entorno de ejecución de sus aplicaciones de mainframe, así como AWS para administrar y monitorear sus aplicaciones modernizadas.

## Temas

- [Características de la modernización del AWS mainframe](#)
- [Patrones](#)
- [¿Cómo empezar con la modernización del mainframe AWS](#)
- [Servicios relacionados](#)
- [Acceder a la modernización del AWS mainframe](#)
- [¿Es la primera vez que utiliza Mainframe Modernization? AWS](#)
- [Precios](#)

### Note

¿Ha contactado con socios competentes en migración de AWS mainframe o con servicios AWS profesionales para su proyecto de modernización de mainframe? De no ser así, le recomendamos que contrate a expertos para su proyecto.

- [AWS Socios competentes en materia de modernización de mainframes](#)
- [Servicios profesionales de AWS](#)

Las características y los casos de uso de la modernización de AWS mainframes respaldan un enfoque de modernización evolutivo, que proporciona beneficios a corto plazo al mejorar la agilidad y muchas oportunidades para optimizar e innovar más adelante. Para obtener más información, consulte [Enfoque de modernización](#).

## Características de la modernización del AWS mainframe

AWS Las funciones de modernización del mainframe admiten los siguientes casos de uso:

- **Evaluación:** la capacidad de evaluación de AWS Mainframe Modernization puede ayudarle a evaluar, definir y planificar un proyecto de migración y modernización.
- **Refactorización:** con la tecnología de AWS Blu Age, puede utilizar la refactorización para convertir lenguajes de programación de aplicaciones antiguos, crear macroservicios o microservicios y modernizar las interfaces de usuario (UI) y los paquetes de software de aplicaciones.

AWS Blu Insights ya está disponible mediante el inicio de sesión único. AWS Management Console Ya no tiene que gestionar credenciales de AWS Blu Insights independientes. Puede acceder a las funciones de AWS AWS Blu Age Codebase y Transformation Center directamente desde. AWS Management Console

- **Redefinición de la plataforma:** con la tecnología de la solución Micro Focus Enterprise, puede llevar a la aplicación a un lugar donde gran parte del código fuente de la aplicación se vuelva a compilar sin cambios.
- **IDE para desarrolladores:** AWS Mainframe Modernization ofrece un entorno de desarrollo integrado (IDE) bajo demanda para que los desarrolladores puedan escribir código más rápido mediante la edición y la depuración inteligentes, la compilación instantánea de código y las pruebas unitarias.
- **Tiempo de ejecución gestionado:** el entorno de tiempo de ejecución gestionado de AWS Mainframe Modernization monitorea continuamente los clústeres para mantener las cargas de trabajo empresariales en funcionamiento mediante un procesamiento autorreparable y un escalado automatizado.
- **Integración y entrega continuas (CI/CD):** la función CI/CD de AWS Mainframe Modernization ayuda a los equipos de desarrollo de aplicaciones a realizar cambios de código con mayor frecuencia y fiabilidad, lo que acelera la velocidad de migración, aumenta la calidad y ayuda a reducir el número de nuevas funciones empresariales. time-to-market



- Integraciones con otros AWS servicios: la modernización del AWS mainframe apoya AWS CloudFormation y permite una implementación repetible y una mayor seguridad y AWS Key Management Service conformidad. AWS PrivateLink
- Mayor disponibilidad: la modernización de los AWS mainframes ya está disponible en EE. UU. Este (Ohio), EE. UU. Oeste (Norte de California), Asia Pacífico (Bombay), Asia Pacífico (Seúl), Asia Pacífico (Singapur), Asia Pacífico (Tokio), Europa (Londres) y Europa (París).

Para obtener más información sobre las funciones de modernización de AWS mainframe, consulte <https://aws.amazon.com/mainframe-modernization/features/>

## Patrones

El patrón de refactorización automatizada, impulsado por AWS Blu Age, se centra en acelerar la modernización al convertir toda la pila de aplicaciones antiguas y su capa de datos en una aplicación moderna basada en Java y, al mismo tiempo, preservar la equivalencia funcional. Durante esta transformación automatizada, crea una aplicación de varios niveles con una interfaz basada en Angular, un backend de Java con API y una capa de datos que accede a los almacenes de datos modernos. El proceso de refactorización proporciona una funcionalidad equivalente a la versión anterior para aumentar la automatización de los proyectos, lo que se traduce en velocidad, calidad y menor coste, lo que permite obtener beneficios empresariales más rápidamente. Para obtener más información, consulte [AWS Mainframe Modernization Automated Refactor](#).

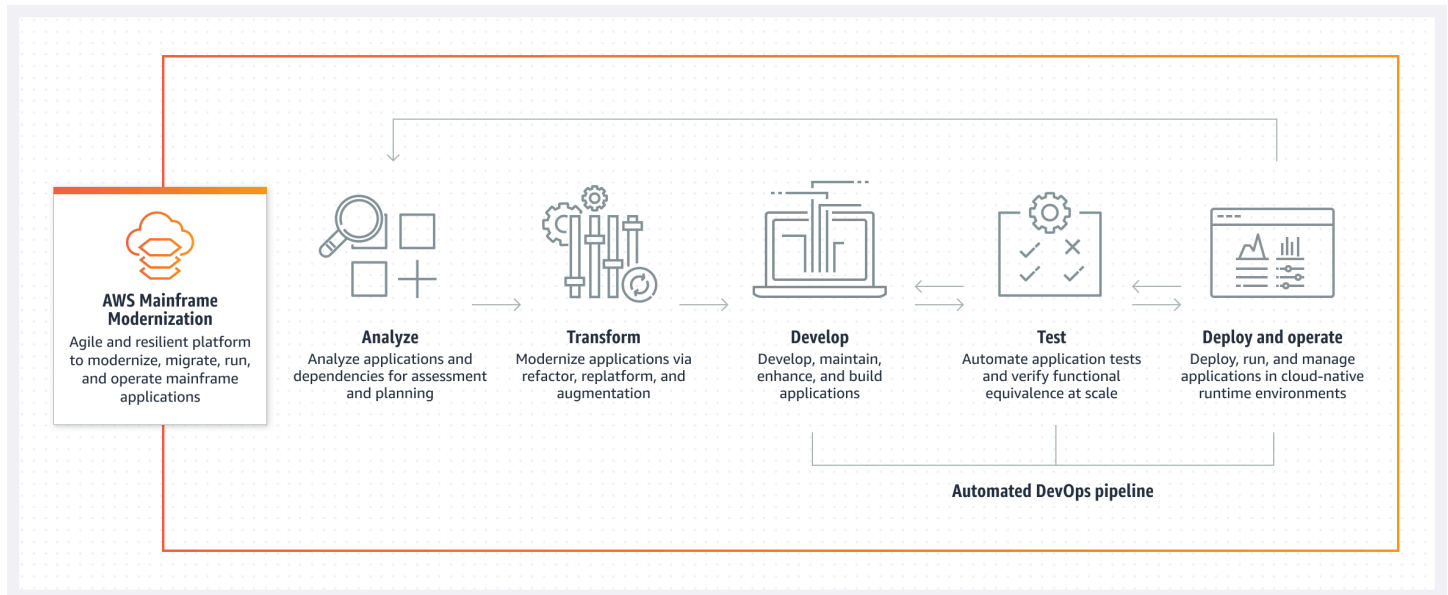
El patrón de redefinición de plataforma, basado en el conjunto de Micro Focus Enterprise, se centra en preservar el lenguaje, el código y los artefactos de la aplicación a fin de minimizar el impacto en los activos y los equipos de las aplicaciones. Ayuda a los clientes a mantener los conocimientos y las habilidades de la aplicación. Si bien los cambios en las aplicaciones son limitados, este patrón también facilita la modernización de la infraestructura y los procesos. La infraestructura se transforma en un servicio administrado moderno basado en la nube, mientras que los procesos se modifican para seguir las mejores prácticas de desarrollo de aplicaciones y operaciones de TI. Para obtener más información, consulte [AWS Mainframe Modernization Replatform](#)

## ¿Cómo empezar con la modernización del mainframe AWS

¡Pruébalo! Ofrecemos tutoriales y ejemplos de aplicaciones para ayudarle a hacerse una idea de lo que ofrece la modernización de AWS mainframes. Elija entre el tutorial [Tutorial: Tiempo de ejecución gestionado para AWS Blu Age](#) o el [Tutorial: Tiempo de ejecución administrado para Micro Focus](#) para ver un step-by-step tutorial completo.

Si está interesado en la refactorización automatizada, consulte las herramientas de AWS Blu Age en [BluInsights](#). También puede configurar la AppStream versión 2.0 para acceder al IDE para desarrolladores de AWS Blu Age o a las herramientas Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer.

Los tutoriales y las aplicaciones de muestra solo le dan una idea de lo que ofrece AWS Mainframe Modernization. Cuando esté listo para iniciar un proyecto de modernización, consulte los detalles sobre las etapas y las tareas de un proyecto de modernización en [Enfoque de modernización](#).



## Servicios relacionados

Además de Blu Insights para la refactorización automatizada, puede utilizar los siguientes AWS servicios con AWS Mainframe Modernization.

- Amazon RDS para alojar sus bases de datos migradas.
- Amazon S3 para almacenar archivos binarios de aplicaciones y de definición.
- Amazon FSx o Amazon EFS para almacenar datos de aplicaciones.
- Amazon AppStream para acceder a las herramientas Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer.
- AWS CloudFormation para la DevOps canalización automatizada que puede utilizar para configurar la CI/CD para las aplicaciones migradas.
- AWS Migration Hub
- AWS DMS para migrar sus bases de datos.

## Acceder a la modernización del AWS mainframe

[Actualmente, puede acceder a AWS Mainframe Modernization a través de la consola en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/). Para obtener una lista de las regiones en las que está disponible la modernización de AWS mainframes, consulte los [puntos finales y las cuotas de modernización de AWS mainframes](#) en la Referencia general de Amazon Web Services

## ¿Es la primera vez que utiliza Mainframe Modernization? AWS

Si es la primera vez que utiliza AWS Mainframe Modernization, le recomendamos que comience leyendo las siguientes secciones:

- [Introducción](#)
- [Configuración](#)

## Precios

AWS La modernización del mainframe cobra por el uso de instancias compatibles con los entornos de tiempo de ejecución gestionados. Además, AWS Mainframe Modernization ofrece algunas herramientas sin cargos adicionales. Usted es responsable de los cargos incurridos por otros AWS servicios que utilice en relación con la modernización del AWS mainframe. AWS avisará con 30 días de antelación antes de que entre en vigor cualquier cambio en los precios por el uso de AWS Mainframe Modernization. Para obtener más información, consulte Modernización de [mainframe](#) con AWS

Con AWS Blu Insights, usted paga por el uso del Centro de Transformación. Para obtener más información, consulte [AWS Mainframe Modernization pricing](#)

# Configuración de la modernización del AWS mainframe

Antes de empezar a utilizar la modernización AWS del mainframe, usted o su administrador deben completar algunos pasos.

## Temas

- [Inscríbese en una Cuenta de AWS](#)
- [Creación de un usuario con acceso administrativo](#)

## Inscríbese en una Cuenta de AWS

Si no tiene una Cuenta de AWS, complete los siguientes pasos para crearlo.

Para suscribirse a una Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones que se le indiquen.

Parte del procedimiento de registro consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Cuando te registras en una Cuenta de AWS, Usuario raíz de la cuenta de AWS se crea una. El usuario raíz tendrá acceso a todos los Servicios de AWS y recursos de esa cuenta. Como práctica recomendada de seguridad, asigne acceso administrativo a un usuario y utilice únicamente el usuario raíz para realizar [tareas que requieren acceso de usuario raíz](#).

AWS te envía un correo electrónico de confirmación una vez finalizado el proceso de registro. Puede ver la actividad de la cuenta y administrar la cuenta en cualquier momento entrando en <https://aws.amazon.com/> y seleccionando Mi cuenta.

## Creación de un usuario con acceso administrativo

Después de crear un usuario administrativo Cuenta de AWS, asegúrelo Usuario raíz de la cuenta de AWS AWS IAM Identity Center, habilite y cree un usuario administrativo para no usar el usuario root en las tareas diarias.

## Proteja su Usuario raíz de la cuenta de AWS

1. Inicie sesión [AWS Management Console](#) como propietario de la cuenta seleccionando el usuario root e introduciendo su dirección de Cuenta de AWS correo electrónico. En la siguiente página, escriba su contraseña.

Para obtener ayuda para iniciar sesión con el usuario raíz, consulte [Signing in as the root user](#) en la Guía del usuario de AWS Sign-In .

2. Active la autenticación multifactor (MFA) para el usuario raíz.

Para obtener instrucciones, consulte [Habilitar un dispositivo MFA virtual para el usuario Cuenta de AWS raíz \(consola\)](#) en la Guía del usuario de IAM.

## Creación de un usuario con acceso administrativo

1. Activar IAM Identity Center.

Consulte las instrucciones en [Activar AWS IAM Identity Center](#) en la Guía del usuario de AWS IAM Identity Center .

2. En IAM Identity Center, conceda acceso administrativo a un usuario.

Para ver un tutorial sobre su uso Directorio de IAM Identity Center como fuente de identidad, consulte [Configurar el acceso de los usuarios con la configuración predeterminada Directorio de IAM Identity Center en la](#) Guía del AWS IAM Identity Center usuario.

## Iniciar sesión como usuario con acceso de administrador

- Para iniciar sesión con el usuario de IAM Identity Center, utilice la URL de inicio de sesión que se envió a la dirección de correo electrónico cuando creó el usuario de IAM Identity Center.

Para obtener ayuda para iniciar sesión con un usuario del Centro de identidades de IAM, consulte [Iniciar sesión en el portal de AWS acceso](#) en la Guía del AWS Sign-In usuario.

## Concesión de acceso a usuarios adicionales

1. En IAM Identity Center, cree un conjunto de permisos que siga la práctica recomendada de aplicar permisos de privilegios mínimos.

Para conocer las instrucciones, consulte [Create a permission set](#) en la Guía del usuario de AWS IAM Identity Center .

2. Asigne usuarios a un grupo y, a continuación, asigne el acceso de inicio de sesión único al grupo.

Para conocer las instrucciones, consulte [Add groups](#) en la Guía del usuario de AWS IAM Identity Center .

# Introducción a la modernización del AWS mainframe

Para empezar con la modernización de AWS mainframes, puede seguir los tutoriales que le presentan el servicio y cada motor de ejecución.

## Temas

- [Tutorial: Tiempo de ejecución gestionado para AWS Blu Age](#)
- [Tutorial: Tiempo de ejecución administrado para Micro Focus](#)

Para seguir aprendiendo, consulte los siguientes tutoriales.

- [Tutorial: Configuración de la versión de Micro Focus para la aplicación BankDemo de muestra](#)
- [Tutorial: Configuración de una canalización de CI/CD para su uso con Micro Focus Enterprise Developer](#)

## Tutorial: Tiempo de ejecución gestionado para AWS Blu Age

Este tutorial muestra cómo implementar una aplicación modernizada de AWS Blu Age en un entorno de ejecución de AWS Mainframe Modernization.

## Temas

- [Requisitos previos](#)
- [Paso 1: Cargar la aplicación de demostración](#)
- [Paso 2: Crear la definición de aplicación](#)
- [Paso 3: Crear un entorno de tiempo de ejecución](#)
- [Paso 4: Crear una aplicación](#)
- [Paso 5: Implementar una aplicación](#)
- [Paso 6: Iniciar una aplicación](#)
- [Paso 7: Acceder a la aplicación](#)
- [Paso 8: Probar la aplicación](#)
- [Eliminar recursos](#)

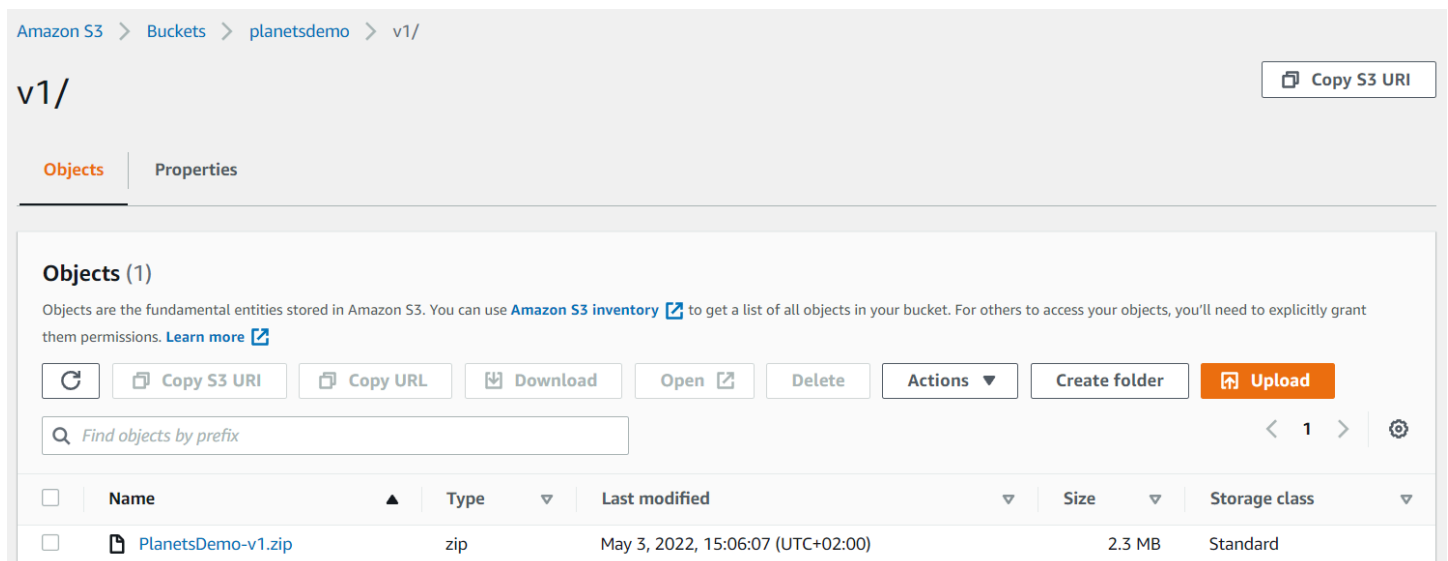
## Requisitos previos

[Para completar este tutorial, descargue el archivo PlanetsDemo de aplicaciones de demostración - v1.zip.](#)

La aplicación de demostración en ejecución requiere un navegador moderno para acceder a ella. El hecho de que ejecute este navegador desde su escritorio o desde una instancia de Amazon Elastic Compute Cloud, por ejemplo, dentro de la VPC, determinará su configuración de seguridad.

## Paso 1: Cargar la aplicación de demostración

Cargue la aplicación de demostración en un bucket de Amazon S3. Asegúrese de que este bucket esté en la misma Región de AWS en la que va a implementar la aplicación. El siguiente ejemplo muestra un bucket denominado planetsdemo, con un prefijo de clave, o carpeta, llamado v1 y un archivo llamado planetsdemo-v1.zip



The screenshot shows the Amazon S3 console interface. The breadcrumb navigation at the top reads 'Amazon S3 > Buckets > planetsdemo > v1/'. The main heading is 'v1/' with a 'Copy S3 URI' button. Below this, there are tabs for 'Objects' and 'Properties'. The 'Objects (1)' section contains a descriptive paragraph and a toolbar with buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar is present with the placeholder 'Find objects by prefix'. Below the search bar is a table with the following data:

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

### Note

La carpeta del bucket es obligatoria.

## Paso 2: Crear la definición de aplicación

Para implementar una aplicación en el entorno de ejecución gestionado, necesita una definición de aplicación de modernización AWS del mainframe. Esta definición es un archivo JSON que describe



la ubicación y la configuración de la aplicación. El siguiente ejemplo es una definición de aplicación de este tipo para la aplicación de demostración:

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

Cambie la entrada `s3-bucket` por el nombre del bucket en el que haya guardado el archivo zip de la aplicación de ejemplo.

Para obtener más información sobre la definición de aplicaciones, consulte [AWS Ejemplo de definición de aplicación de Blu Age](#).

### Paso 3: Crear un entorno de tiempo de ejecución

Para crear el entorno de ejecución de AWS Mainframe Modernization, lleve a cabo los siguientes pasos:

1. Abra la [consola de AWS Mainframe Modernization](#).
2. En el Región de AWS selector, elija la región en la que desee crear el entorno. Región de AWS Debe coincidir con la región en la que creó el bucket de S3 [Paso 1: Cargar la aplicación de demostración](#).
3. En Modernice las aplicaciones de mainframe, elija Refactorizar con Blu Age y, a continuación, elija Comenzar.

## Modernize mainframe applications

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

Choose an option to get started.






- Refactor with Blu Age
- Replatform with Micro Focus

**Get started**

4. En Cómo puede ayudarle AWS Mainframe Modernization, elija Implementar y Crear entorno en tiempo de ejecución.

### How can AWS Mainframe Modernization help?

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

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

#### Deploy [Info](#)

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

5. En el panel de navegación de la izquierda, seleccione Entornos y, a continuación, Crear entorno. En la página Especificar información básica, introduzca un nombre y una descripción para su

entorno y, a continuación, asegúrese de que está seleccionado el motor AWS Blu Age. Si lo desea, puede agregar etiquetas al recurso creado. A continuación, elija Siguiente.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
**Specify basic information**

Step 2  
Specify configurations

Step 3 - *Optional*  
Attach storage

Step 4  
Review and create

## Specify basic information [Info](#)

### Name and description

Environment name

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


Environment description - *optional*

The description can be up to 500 characters.


### Engine options

Select Engine

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



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



AWS Blu Age Version

6. En la página Especificar las configuraciones, elija Entorno en tiempo de ejecución independiente.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
**Specify configurations**

Step 3 - Optional  
Attach storage

Step 4  
Review and create

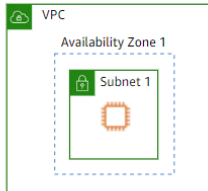
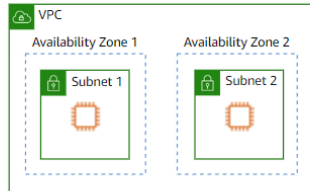
## Specify configurations Info

### Availability Info

Choose the availability pattern for your environment.

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

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

## 7. En Seguridad y red, realice los siguientes cambios:

- Seleccione Permitir que las aplicaciones implementadas en este entorno sean de acceso público. Esta opción asigna una dirección IP pública a la aplicación para que pueda acceder a ella desde el escritorio.
- Elija una VPC. Puede utilizar el valor Predeterminado.
- Seleccione dos subredes. Asegúrese de que las subredes permitan la asignación de direcciones IP públicas.
- Elija un grupo de seguridad. Puede utilizar el valor Predeterminado. Asegúrese de que el grupo de seguridad que elija permita el acceso desde una dirección IP al puerto que haya especificado en la propiedad `listener` de la definición de la aplicación. Para obtener más información, consulte [Paso 2: Crear la definición de aplicación](#).

## Security and network

Allow applications deployed to this environment to be publicly accessible.

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

Default vpc- ▼

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

Choose subnets ▼

subnet- ✕

subnet- ✕

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

Choose security groups ▼

default ✕  
default VPC security group

Si desea acceder a la aplicación desde fuera de la VPC que haya elegido, asegúrese de que las reglas de entrada de esa VPC estén configuradas correctamente. Para obtener más información, consulte [Cannot access an application's URL](#).

8. Elija Siguiente.
9. En Adjuntar almacenamiento: opcional, deje las selecciones predeterminadas y elija Siguiente.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
Specify configurations

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

Step 4  
Review and create

### Attach storage - *Optional* Info

#### EFS storage

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

No EFS associated with this environment.

**Choose EFS storage**

You can add up to 1 more EFS.

#### FSx storage

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

No EFS associated with this environment.

**Choose FSx storage**

You can add up to 1 more FSx.

Cancel Previous **Next**

10. En Mantenimiento programado, elija Sin preferencia y, a continuación, elija Siguiete.
11. En Revisar y crear, revise la información y, a continuación, elija Crear entorno.

## Paso 4: Crear una aplicación

1. Navegue hasta AWS Mainframe Modernization en AWS Management Console.
2. En el panel de navegación, elija Aplicaciones, y, a continuación, seleccione Crear aplicación. En la página Especificar información básica, introduzca un nombre y una descripción para la aplicación y, a continuación, asegúrese de que está seleccionado el motor AWS Blu Age. A continuación, elija Next.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name


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

Application description - *optional*


The maximum length is 500 characters.

### Engine type

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



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



3. En la página Especificar recursos y configuraciones, copie y pegue la definición de aplicación actualizada en formato JSON que creó en [the section called “Paso 2: Crear la definición de aplicación”](#).

AWS Mainframe Modernization > Applications > Create application

Step 1  
Specify basic information

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

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

### Resources and configurations

Choose an approach to define the application

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

Use an application definition JSON file in an Amazon S3 bucket

```

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

```

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

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

4. En Revisar y crear, revise sus opciones y, a continuación, elija Crear aplicación.

## Paso 5: Implementar una aplicación

Una vez que haya creado correctamente el entorno de ejecución y la aplicación de AWS Mainframe Modernization y ambos estén en el estado Disponible, podrá implementar la aplicación en el entorno de ejecución. Para ello, siga los pasos que se describen a continuación:

1. Navegue hasta AWS Mainframe Modernization en la consola AWS de administración. En el panel de navegación, elija Entornos. Aparece la página de lista de entornos.



AWS Mainframe Modernization > Environments

**Environments (1)** [Info](#)

Find environment

Actions [Create environment](#)

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

2. Elija el entorno de tiempo de ejecución creado anteriormente. Página de detalles del entorno.
3. Elija Implementar aplicación.

AWS Mainframe Modernization > Environments > planets-demo-env

**planets-demo-env** [Info](#)

Actions [Deploy application](#)

Summary | Configurations | Deployed applications | Tags

**Environment** [Info](#)

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

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

**No applications**  
No applications to display.  
[Deploy application](#)

4. Elija la aplicación creada anteriormente y seleccione la versión en la que desee implementar la aplicación. A continuación, elija Implementar.

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

## Deploy application Info

You have selected the following application:

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

### Available versions (0) ↻

Choose a version from the list.

< 1 > ⚙️

Version ▾

Creation time ▲

No versions  
No versions to display

### Environments (1) Info

< 1 > ⚙️

Enviro... ▾

Status ▾

Engine ▾

Version ▾

Instance type ▾

Cr

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

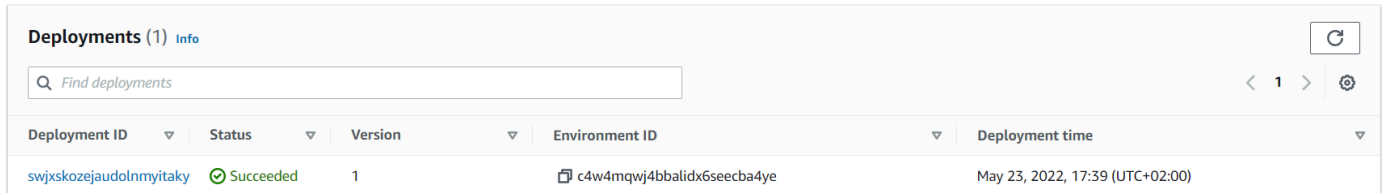
Cancel

Deploy

- Espera hasta que la aplicación finalice su implementación. Verá un banner con el mensaje Application was deployed successfully.

## Paso 6: Iniciar una aplicación

1. Navegue hasta AWS Mainframe Modernization en AWS Management Console y seleccione Aplicaciones.
2. Seleccione la aplicación y elija Implementaciones. El estado de la aplicación debe ser Finalizado con éxito.



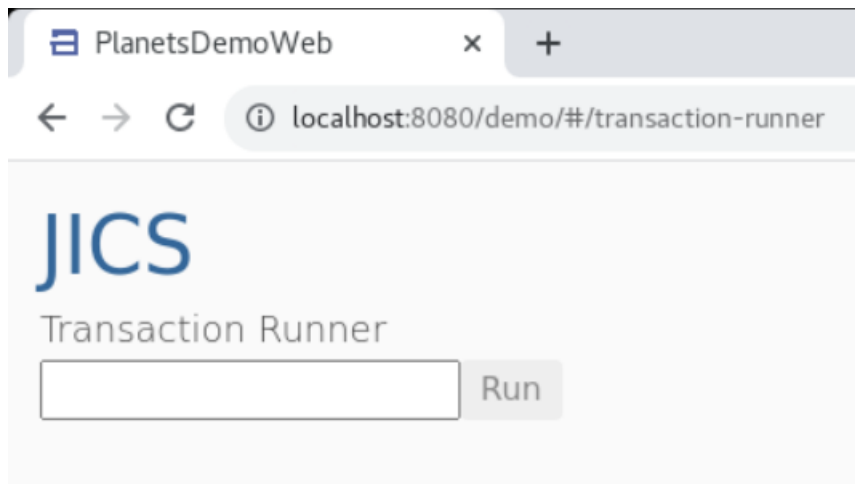
Deployment ID	Status	Version	Environment ID	Deployment time
swjxskozejaudolnmyitaky	Succeeded	1	c4w4mqwj4bbalidx6seecba4ye	May 23, 2022, 17:39 (UTC+02:00)

3. Elija Acciones y, a continuación, Iniciar aplicación.

## Paso 7: Acceder a la aplicación

1. Espere hasta que la aplicación esté en estado En ejecución. Ver un banner con el mensaje Application was started successfully.
2. Copie el nombre de host DNS de la aplicación. Puede encontrarlo en la sección de Información de la aplicación de la aplicación.
3. En un navegador, navegue hasta `http://{hostname}:{portname}/PlanetsDemo-web-1.0.0/`, donde:
  - `hostname` es el nombre de host DNS copiado anteriormente.
  - `portname` es el puerto de Tomcat especificado en la definición de la aplicación que ha creado en el [Paso 2: Crear la definición de aplicación](#).

Aparece la pantalla de JICS.



Puede acceder a la aplicación, consulte [Cannot access an application's URL](#).

**Note**

Si no se puede acceder a la aplicación y la regla de entrada del grupo de seguridad tiene seleccionada la opción “Mi IP” en el puerto 8196, especifique la regla que permita el tráfico desde LB i/p en el puerto 8196.

## Paso 8: Probar la aplicación

En este paso, ejecutará una transacción en la aplicación migrada.

1. En la pantalla de JICS, introduzca PINQ en el campo de entrada y elija Ejecutar (o pulse Intro) para iniciar la transacción de la aplicación.

Debería aparecer la pantalla de la aplicación de demostración.



2. Escriba el nombre de un planeta en el campo correspondiente y pulse Intro.



Deberían aparecer los detalles del planeta.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para evitar cargos adicionales. Para ello, complete los siguientes pasos.

- Si la aplicación de modernización del AWS mainframe sigue ejecutándose, deténgala.
- Elimine la aplicación. Para obtener más información, consulte [Elimine una aplicación de modernización AWS de mainframe](#).
- Eliminar un entorno de tiempo de ejecución Para obtener más información, consulte [Elimine un entorno de ejecución AWS de modernización de mainframe](#).

## Tutorial: Tiempo de ejecución administrado para Micro Focus

En este tutorial se muestra cómo implementar y ejecutar la aplicación de CardDemo ejemplo en un entorno de tiempo de ejecución gestionado por la modernización del AWS mainframe con el motor de ejecución Micro Focus. La aplicación CardDemo de ejemplo es una aplicación de tarjetas de crédito simplificada desarrollada para probar y mostrar la tecnología AWS y asociarla a los casos de uso de la modernización de mainframes.

En el tutorial, creará recursos en otros Servicios de AWS. Estos incluyen Amazon Simple Storage Service, Amazon Relational Database Service AWS Key Management Service y. AWS Secrets Manager

### Temas

- [Requisitos previos](#)
- [Paso 1: Crear y cargar un bucket de Amazon S3](#)
- [Paso 2: Crear y configurar una base de datos](#)
- [Paso 3: Crear y configurar un AWS KMS key](#)
- [Paso 4: Crear y configurar un secreto de AWS Secrets Manager base de datos](#)
- [Paso 5: Crear un entorno de ejecución](#)
- [Paso 6: Crear una aplicación](#)
- [Paso 7: implementar una aplicación](#)
- [Paso 8: Importar conjuntos de datos](#)
- [Paso 9: Iniciar una aplicación](#)

- [Paso 10: Conectarse a la CardDemo aplicación CICS](#)
- [Eliminar recursos](#)
- [Sigüientes pasos](#)

## Requisitos previos

- Asegúrese de tener acceso a un emulador 3270 para usar la conexión CICS. Los emuladores 3270 gratuitos y de prueba están disponibles en sitios web de terceros. Como alternativa, puede iniciar una instancia de Micro Focus de AWS Mainframe Modernization AppStream 2.0 y utilizar el emulador Rumba 3270 (no disponible de forma gratuita).

Para obtener información acerca de la AppStream versión 2.0, consulte [the section called “Tutorial: Configuración de la AppStream versión 2.0 para Enterprise Analyzer y Enterprise Developer”](#)

### Note

Al crear la pila, elija la opción Enterprise Developer (ED) y no Enterprise Analyzer (EA).

- Descargue la [aplicación CardDemo de muestra](#) y descomprima el archivo descargado en cualquier directorio local. Este directorio contendrá un subdirectorio titulado. CardDemo
- Identifique una VPC en su cuenta donde pueda definir los recursos creados en este tutorial. La VPC necesitará subredes en al menos dos zonas de disponibilidad. Para obtener más información sobre Amazon VPC, consulte [Cómo funciona Amazon VPC](#).

## Paso 1: Crear y cargar un bucket de Amazon S3

En este paso, debe crear un depósito de Amazon S3 y cargar CardDemo archivos en este depósito. Más adelante en este tutorial, utilizará estos archivos para implementar y ejecutar la aplicación de CardDemo ejemplo en un entorno de ejecución gestionado por Micro Focus para la modernización de AWS mainframe.

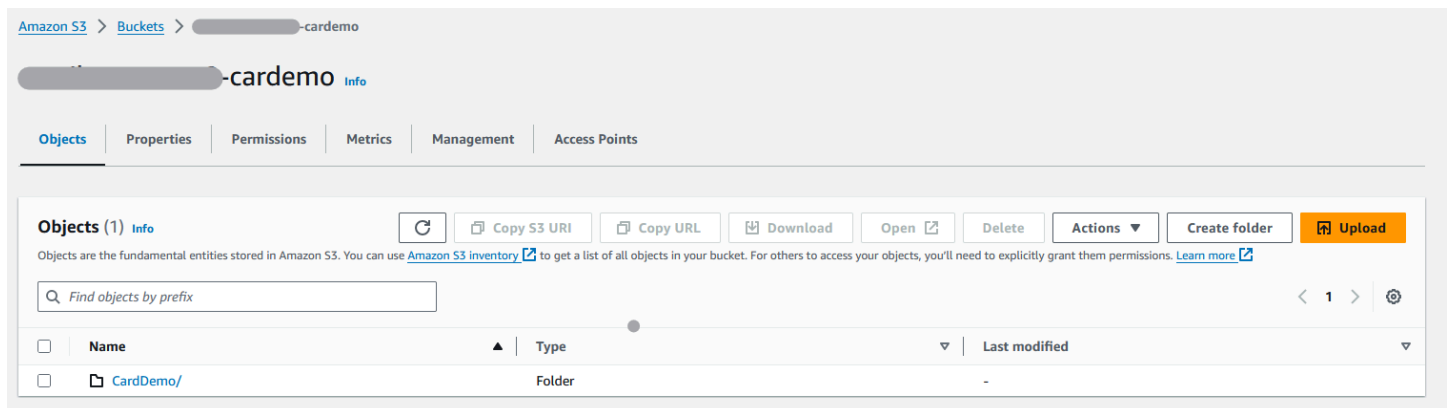
### Note

No tiene que crear un nuevo depósito de S3, pero el depósito que elija debe estar en la misma región que los demás recursos utilizados en este tutorial.

## Creación de un bucket de Amazon S3

1. Abra la [consola Amazon S3](#) y elija Create bucket.
2. En Configuración general, elija la región de AWS en la que desee crear el entorno de ejecución gestionado de Micro Focus para la modernización de AWS mainframe.
3. Introduzca el nombre de un bucket, por ejemplo, `yourname-aws-region-carddemo`. Mantenga la configuración predeterminada y elija Crear depósito. Como alternativa, también puede copiar la configuración de un bucket de Amazon S3 existente y, a continuación, elegir Create bucket.
4. Elige el bucket que acabas de crear y, a continuación, selecciona Cargar.
5. En la sección Cargar, selecciona Añadir carpeta y, a continuación, busca el CardDemo directorio desde tu ordenador local.
6. Selecciona Cargar para iniciar el proceso de carga. Los tiempos de carga varían en función de la velocidad de conexión.
7. Cuando se complete la carga, confirme que todos los archivos se han cargado correctamente y, a continuación, seleccione Cerrar.

Su bucket de Amazon S3 ahora contiene la CardDemo carpeta.



Para obtener información sobre los buckets de S3, consulte [Crear, configurar y trabajar con buckets de Amazon S3](#).

## Paso 2: Crear y configurar una base de datos


En este paso, creará una base de datos PostgreSQL en Amazon Relational Database Service (Amazon RDS). Para el tutorial, esta base de datos contiene los conjuntos de datos que la aplicación



de CardDemo muestra utiliza para las tareas de los clientes relacionadas con las transacciones con tarjetas de crédito.


Para crear una nueva base de datos en Amazon RDS.

1. Abra la [consola de Amazon RDS](#).
2. Elija la región de AWS en la que desee crear la instancia de base de datos.
3. Seleccione Databases (Bases de datos) en el panel de navegación.
4. Elija Crear base de datos y, a continuación, elija Creación estándar.
5. En Tipo de motor, elija PostgreSQL.
6. Elija una versión de motor de 15 o superior.

 Note

Guarde la versión del motor porque la necesitará más adelante en este tutorial.

7. En Plantillas, elija Nivel gratuito.
8. Cambie el identificador de la instancia de base de datos por algo significativo, por ejemplo, `MicroFocus-Tutorial`.
9. Absténgase de gestionar las credenciales maestras en AWS Secrets Manager. En su lugar, introduzca una contraseña maestra y confírmela.

 Note

Guarde el nombre de usuario y la contraseña que utiliza para la base de datos. Los almacenará de forma segura en los siguientes pasos de este tutorial.

10. En Conectividad, elija la VPC en la que desee crear el entorno de tiempo de ejecución gestionado por AWS Mainframe Modernization.
11. Elija Crear base de datos.

Para crear un grupo de parámetros personalizado en Amazon RDS

1. En el panel de navegación de la consola Amazon RDS, elija Grupos de parámetros y, a continuación, elija Crear grupo de parámetros.

2. En la ventana Crear grupo de parámetros, en Familia de grupos de parámetros, seleccione la opción Postgres que coincida con la versión de su base de datos.

 Note

Algunas versiones de Postgres requieren un tipo. Seleccione el grupo de parámetros de base de datos si es necesario. Introduzca un nombre de grupo y una descripción para el grupo de parámetros.


3. Seleccione Crear.

Para configurar el grupo de parámetros personalizado

1. Elija el grupo de parámetros recién creado.
2. Seleccione Acciones y, a continuación, Editar.
3. Active el filtro `max_prepared_transactions` y cambie el valor del parámetro a 100.
4. Seleccione Guardar cambios.

Para asociar el grupo de parámetros personalizado a la base de datos

1. En el panel de navegación de la consola de Amazon RDS, elija Databases y, a continuación, elija la instancia de base de datos que desee modificar.
2. Elija Modify (Modificar). Aparece la página Modify DB instance (Modificar instancia de base de datos).

 Note

La opción Modificar no estará disponible hasta que la base de datos haya terminado de crearse y realizar copias de seguridad, lo que puede tardar varios minutos.

3. En la página Modificar la instancia de base de datos, vaya a Configuración adicional y cambie el grupo de parámetros de base de datos por el suyo. Si su grupo de parámetros no está disponible en la lista, compruebe si se creó con la versión de base de datos correcta.
4. Seleccione Continuar y consulte el resumen de las modificaciones.
5. Seleccione Aplicar inmediatamente para aplicar los cambios al instante.
6. Seleccione Modificar la instancia de base de datos para guardar los cambios.

Para obtener más información, consulte [Trabajo con los grupos de parámetros](#).

**Note**

También puede utilizar una base de datos PostgreSQL de Amazon Aurora AWS con la modernización del mainframe, pero no hay ninguna opción de capa gratuita. Para obtener más información, consulte [Trabajar con Amazon Aurora PostgreSQL](#).

## Paso 3: Crear y configurar un AWS KMS key

Para almacenar de forma segura las credenciales de la instancia de Amazon RDS, cree primero un AWS KMS key.

Para crear un AWS KMS key

1. Abra la [consola del Servicio de administración de claves](#).
2. Elija Create Key (Crear clave).
3. Deje los valores predeterminados de Symmetric para el tipo de clave y Cifrar y descifrar para el uso de claves.
4. Elija Siguiente.
5. Proporcione a la clave un alias, por ejemplo, MicroFocus-Tutorial-RDS-Key y una descripción opcional.
6. Elija Siguiente.
7. Asigne un administrador clave marcando la casilla situada junto a su usuario o función.
8. Seleccione Siguiente y, a continuación, vuelva a seleccionar Siguiente.
9. En la pantalla de revisión, edite la política clave y, a continuación, introduzca lo siguiente:

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

Esta política otorga a AWS Mainframe Modernization permisos de descifrado mediante esta política clave específica.

10. Seleccione Finalizar para crear la clave.

Para obtener más información, consulte [Creación de claves](#) en la Guía para AWS Key Management Service desarrolladores.

## Paso 4: Crear y configurar un secreto de AWS Secrets Manager base de datos

Ahora almacene las credenciales de la base de datos de forma segura con AWS Secrets Manager y AWS KMS key.

Para crear y configurar un secreto de AWS Secrets Manager base de datos

1. Abra la [consola de Secrets Manager](#).
2. En el panel de navegación, elija Secretos.
3. En Secretos, selecciona Guardar un secreto nuevo.
4. Establezca el tipo de secreto en Credenciales para la base de datos de Amazon RDS.
5. Introduzca las credenciales que especificó al crear la base de datos.
6. En Clave de cifrado, seleccione la clave que creó en el paso 3.
7. En la sección Base de datos, seleccione la base de datos que creó para este tutorial y, a continuación, elija Siguiente.
8. En Nombre secreto, introduzca un nombre como, por ejemplo, MicroFocus-Tutorial-RDS-Secret y una descripción opcional.
9. En la sección Permisos de recursos, selecciona Editar permisos y sustituye el contenido por la siguiente política:

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

```

        "Action" : "secretsmanager:GetSecretValue",
        "Resource" : "*"
    }
]
}

```

10. Seleccione Guardar.
11. Selecciona Siguiente para las pantallas siguientes y, a continuación, selecciona Almacenar. Actualiza la lista de secretos para ver el nuevo secreto.
12. Elige el secreto recién creado y anótalo Secret ARN porque lo necesitarás más adelante en el tutorial.
13. En la pestaña Descripción general del secreto, elija Recuperar el valor secreto.
14. Selecciona Editar y, a continuación, selecciona Añadir fila.
15. Agregue una clave para sslMode con un valor de verify-full:

#### Edit secret value

Key/value | Plaintext

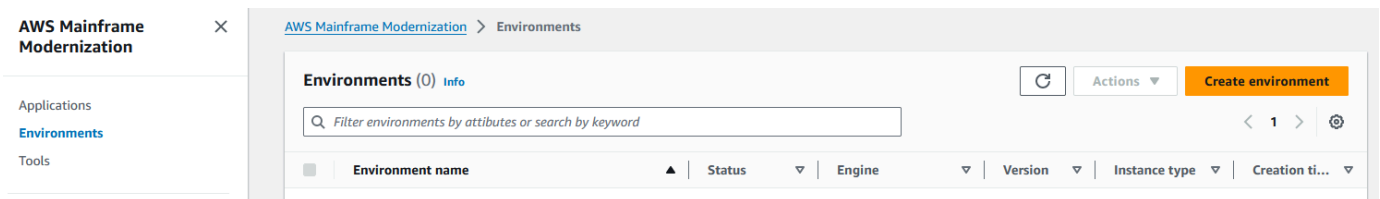
sslMode | verify-full

16. Seleccione Guardar.

## Paso 5: Crear un entorno de ejecución

Para crear un entorno de tiempo de ejecución

1. Abra la [consola de AWS Mainframe Modernization](#).
2. En el panel de navegación, elija Entornos. A continuación, selecciona Crear entorno.



3. En Especificar información básica,
  - a. Introduzca MicroFocus-Environment el nombre del entorno.
  - b. En las opciones de motor, asegúrese de que esté seleccionado Micro Focus.
  - c. Elija la versión más reciente de Micro Focus.

## d. Elija Siguiente.

**Name and description** [Info](#)

## Environment name

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

Environment description - *optional*

The description can be up to 500 characters.

**Engine options** [Info](#)

## Select Engine

**Blu Age**

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

**Micro Focus**

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



## Micro Focus Version

## 4. Configure el entorno

## a. En Disponibilidad, elija Clúster de alta disponibilidad.

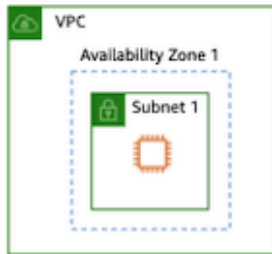
- b. En Recursos, elija `M2.c5.large` el tipo `M2.m5.large` de instancia y el número de instancias que desee. Especifica hasta dos instancias.
- c. En Seguridad y red, elija Permitir que las aplicaciones implementadas en este entorno sean de acceso público y elija al menos dos subredes públicas.
- d. Elija Siguiente.

# Specify configurations [Info](#)

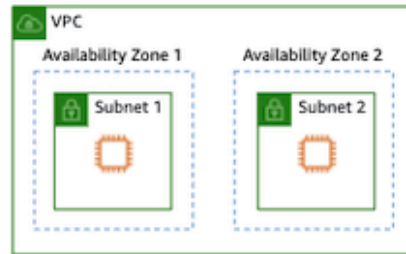
## Availability [Info](#)

Choose the availability pattern for your environment.

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



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



## Resources

### Instance type

Choose the instance type for your high availability cluster.

M2.m5.large

### Desired capacity

Specify the desired number of instances.

2

## Security and network

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

### Virtual Private Cloud (VPC)

Choose the VPC where you want to create the environment.

Default vpc-15

### Subnets

Choose one or more subnets for a high availability setup.

Choose subnets

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

subnet-6685 | us-west-2b X

### Security groups

Choose one or more security groups for the chosen VPC.



5. En la página Adjuntar almacenamiento, elija Paso siguiente.
6. En la página Programar el mantenimiento, elija Sin preferencias y, a continuación, elija Siguiente.

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

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

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

**When to apply modifications**

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

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

Cancel Previous **Next**

7. En la página Revisar y crear, revise todas las configuraciones que proporcionó para el entorno de ejecución y, a continuación, seleccione Crear entorno.

### Step 3: Attach storage Edit

#### EFS storage

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

#### FSx storage

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

### Step 4: Schedule maintenance Edit

#### Maintenance window

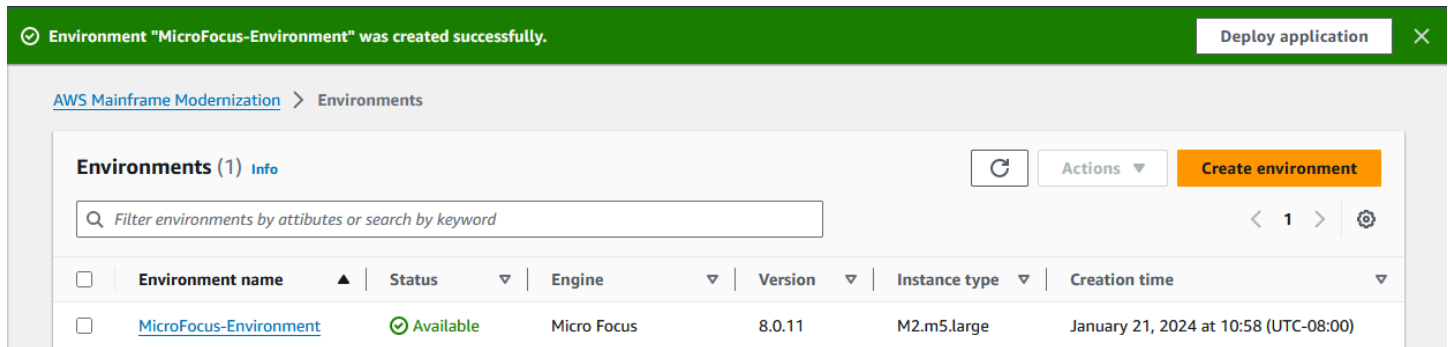
Preferred maintenance window  
No preference

Cancel Previous Create environment

Cuando haya creado su entorno, aparecerá un banner que dice `Environment name was created successfully` y el campo Estado cambiará a Disponible. El proceso de creación del entorno tarda varios minutos, pero puede continuar con los siguientes pasos mientras se ejecuta.

Paso 5: Crear un entorno de ejecución

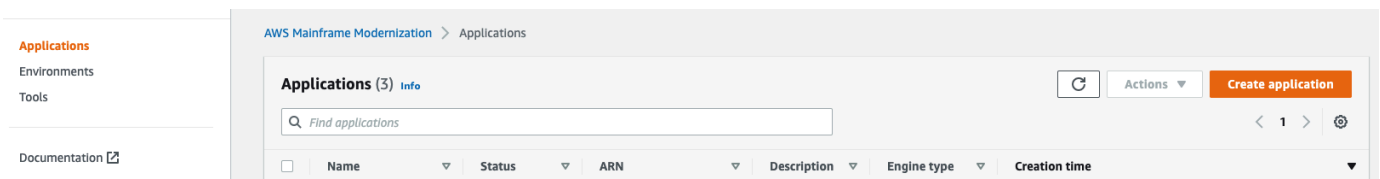
36



## Paso 6: Crear una aplicación

Para crear una aplicación

1. En el panel de navegación, elija Aplicaciones. Elija Crear aplicación.



2. En la página Crear aplicación, en Especificar información básica, introduzca el nombre MicroFocus-CardDemo de la aplicación y, en Tipo de motor, asegúrese de seleccionar Micro Focus. A continuación, elija Next.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name

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


Application description - *optional*

*Describe the application*


The maximum length is 500 characters.

### Engine type

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



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



3. En Especificar recursos y configuraciones, elija la opción para especificar la definición de la aplicación con sus recursos y configuraciones mediante el editor integrado.

AWS Mainframe Modernization > Applications > Create application

Step 1  
[Specify basic information](#)

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

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

### Resources and configurations

Choose an approach to define the application

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

1 {}

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

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

Introduzca la siguiente definición de aplicación en el editor:

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

```

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

```

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

 Note

Este archivo está sujeto a cambios.

4. Edite el JSON de la aplicación en el objeto de propiedades de las ubicaciones de origen de la siguiente manera:
  - a. Sustituya el valor for por `s3_bucket` el nombre del bucket de Amazon S3 que creó en el paso 1.
  - b. Sustituya el valor for por `s3-key-prefix` la carpeta (key prefix) en la que cargó los archivos de CardDemo ejemplo. Si has subido el CardDemo directorio directamente a un bucket de Amazon S3, `s3-key-prefix` no es necesario cambiarlo.
  - c. Sustituya ambos `secret-manager-arn` valores por el ARN del secreto de base de datos que creó en el paso 4.

## Resources and configurations

Choose an approach to define the application

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

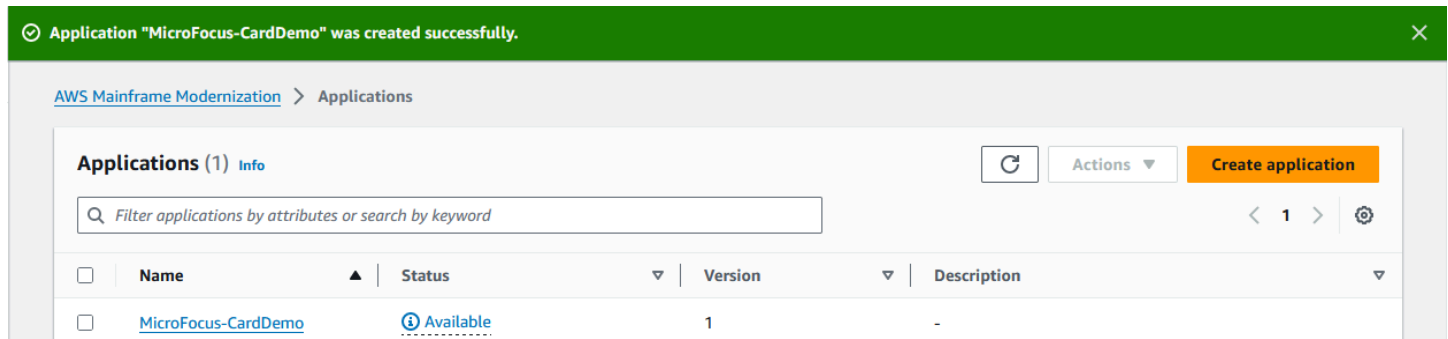
```
1 {
2   "template-version": "2.0",
3   "source-locations": [
4     {
5       "source-id": "s3-source",
6       "source-type": "s3",
7       "properties": {
8         "s3-bucket": "XXXXXXXXXXXX-cardemo",
9         "s3-key-prefix": "CardDemo"
10      }
11    }
12  ],
13  "definition": {
14    "listeners":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

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

Para obtener más información sobre la definición de aplicaciones, consulte [Definición de la aplicación Micro Focus](#).

5. Elija Siguiente para continuar.
6. En la página Revisar y crear, revise la información que ha proporcionado y, a continuación, elija Crear aplicación.



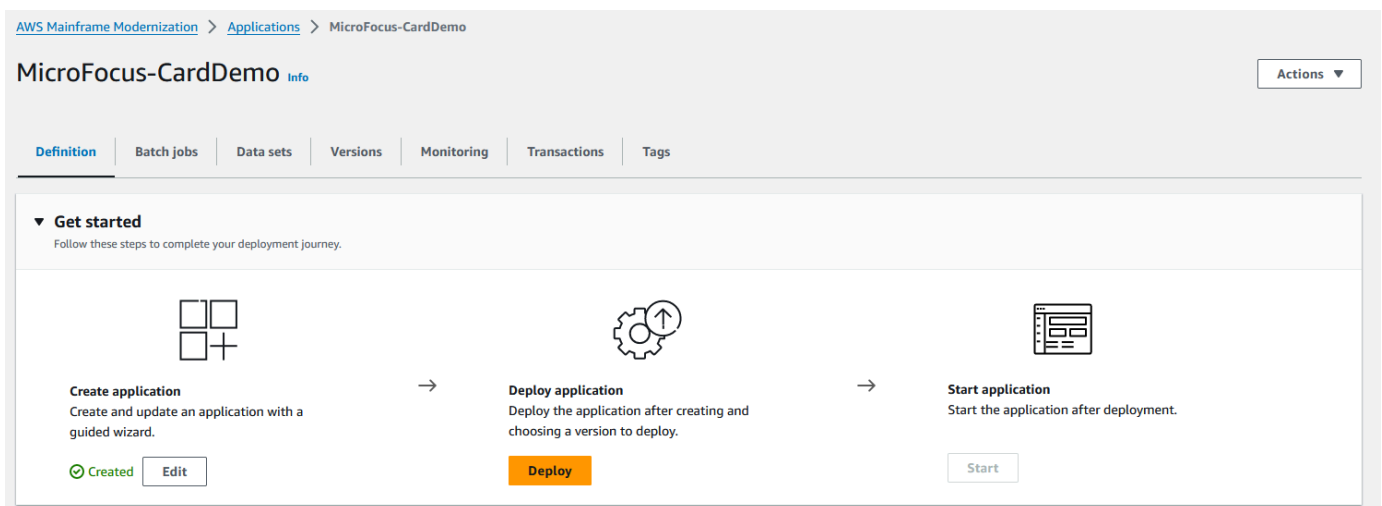


Cuando hayas creado tu aplicación, aparecerá un banner que dice `Application name was created successfully`. Y el campo Estado cambia a Disponible.

## Paso 7: implementar una aplicación

Para implementar una aplicación

1. En el panel de navegación, elija Aplicaciones y, a continuación, elija `MicroFocus-CardDemo`.
2. En Implementar aplicación, elija Implementar.



3. Elija la versión más reciente de la aplicación y el entorno que creó anteriormente y, a continuación, elija Implementar.

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

## Deploy application Info

You have selected the following application:

Name	Description	Engine
MicroFocus-CardDemo	-	Micro Focus

**Available versions (1/1)** ↻

Choose a version from the list.

< 1 > ⚙️

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

**Environments (1/1) Info**

< 1 > ⚙️

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

Cancel Deploy

Cuando la CardDemo aplicación se implementa correctamente, el estado cambia a Listo.

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

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

### Applications (1) Info

< 1 > ⚙️ ↻ Actions Create application

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

## Paso 8: Importar conjuntos de datos

Para importar conjuntos de datos

1. En el panel de navegación, elija Aplicaciones y, a continuación, elija la aplicación.
2. Elija la pestaña Conjuntos de datos. A continuación, elija Import.
3. Elija Importar y editar la configuración de JSON y, a continuación, elija la opción Copiar y pegar su propio JSON.

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

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

Choose import method.

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

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

**JSON configuration**

Import from Amazon S3 bucket.

Copy and paste your own JSON.

1		
---	--	--

4. Copia y pega el siguiente JSON, pero no selecciones «Enviar» todavía. Este JSON contiene todos los conjuntos de datos necesarios para la aplicación de demostración, pero necesita los detalles del bucket de Amazon S3.

```
{
  "dataSets": [
    {
      "dataSet": {
        "storageType": "Database",
```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

5. Sustituya cada aparición de <s3-bucket-name> (hay ocho) por el nombre del bucket de Amazon S3 que contiene la CardDemo carpeta, por ejemplo, your-name-aws-region-carddemo.

#### Note

Para copiar el URI de Amazon S3 de la carpeta de Amazon S3, seleccione la carpeta y, a continuación, elija Copiar el URI de Amazon S3.

6. Seleccione Submit (Enviar).

Cuando finalice la importación, aparecerá un banner con el siguiente mensaje: Import task with resource identifier *name* was completed successfully. Se muestra una lista de los conjuntos de datos importados.



Import task with resource identifier "Ipa6795ukmfr9" was completed successfully.

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

## MicroFocus-CardDemo Info

Actions ▾

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

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

Filter data sets by name

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

También puede ver el estado de todas las importaciones de conjuntos de datos seleccionando Historial de importaciones en la pestaña Conjuntos de datos.

## Paso 9: Iniciar una aplicación

Para iniciar una aplicación

1. En el panel de navegación, elija Aplicaciones y, a continuación, elija la aplicación.
2. Seleccione Iniciar aplicación.


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

## MicroFocus-CardDemo Info

Actions ▾


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

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




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

✔ Created Edit



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

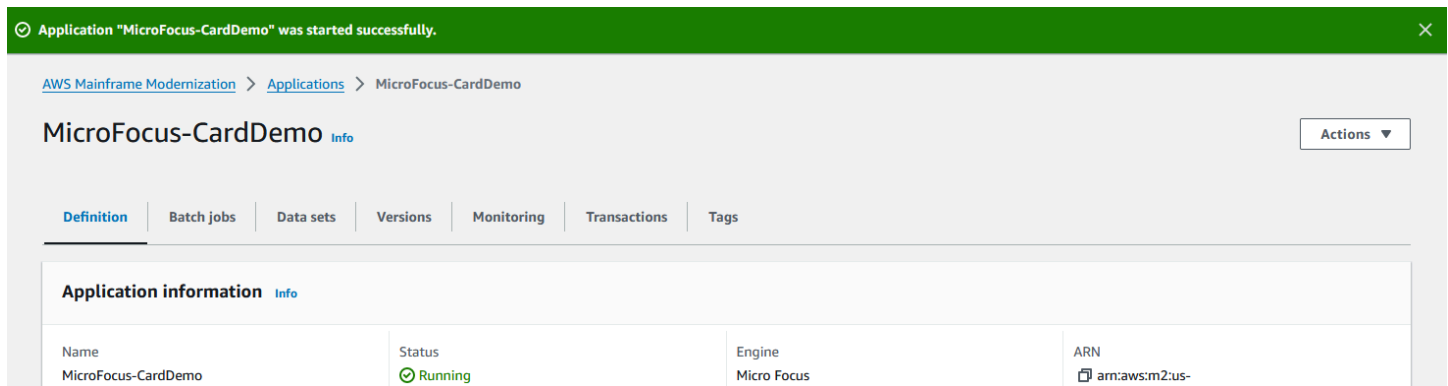
✔ Deployed Deploy



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

⊖ Stopped Start

Cuando la CardDemo aplicación comienza a ejecutarse correctamente, aparece un banner con el siguiente mensaje: `Application name was started successfully`. El campo Estado cambia a En ejecución.



## Paso 10: Conectarse a la CardDemo aplicación CICS

Antes de conectarse, asegúrese de que la VPC y el grupo de seguridad que especificó para la aplicación son los mismos que los que aplicó para la interfaz de red desde la que se conectará.

Para configurar la conexión TN3270, también necesita el nombre de host DNS y el puerto de la aplicación.

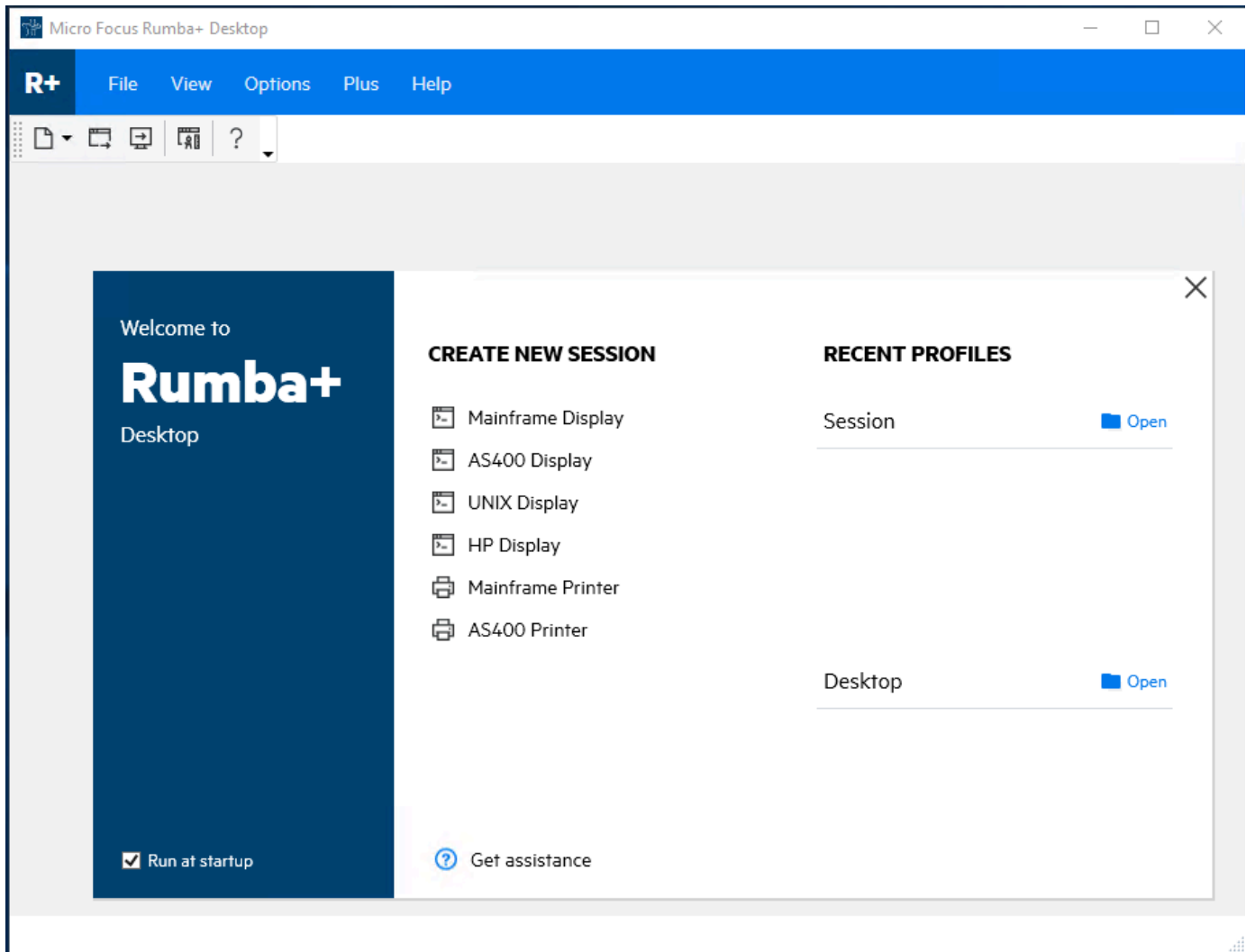
Para configurar y conectar una aplicación al mainframe mediante un emulador de terminal

1. Abra la consola de modernización del AWS mainframe, seleccione Aplicaciones y, a continuación, elija. `MicroFocus-CardDemo`
2. Seleccione el icono de copia para copiar el nombre de servidor DNS. Asegúrese también de anotar el número de puertos.
3. Inicie un emulador de terminal. En este tutorial se utiliza Micro Focus Rumba+.

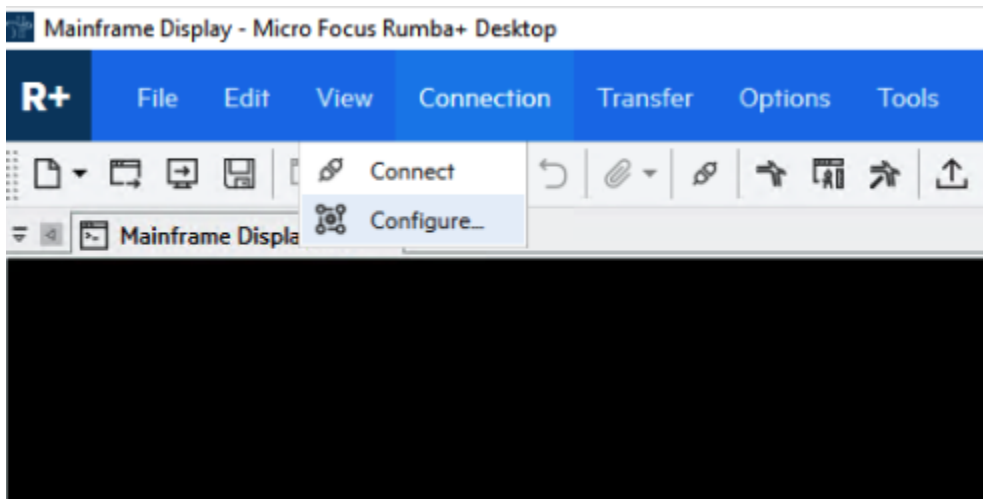
### Note

Los pasos de configuración varían según el emulador.

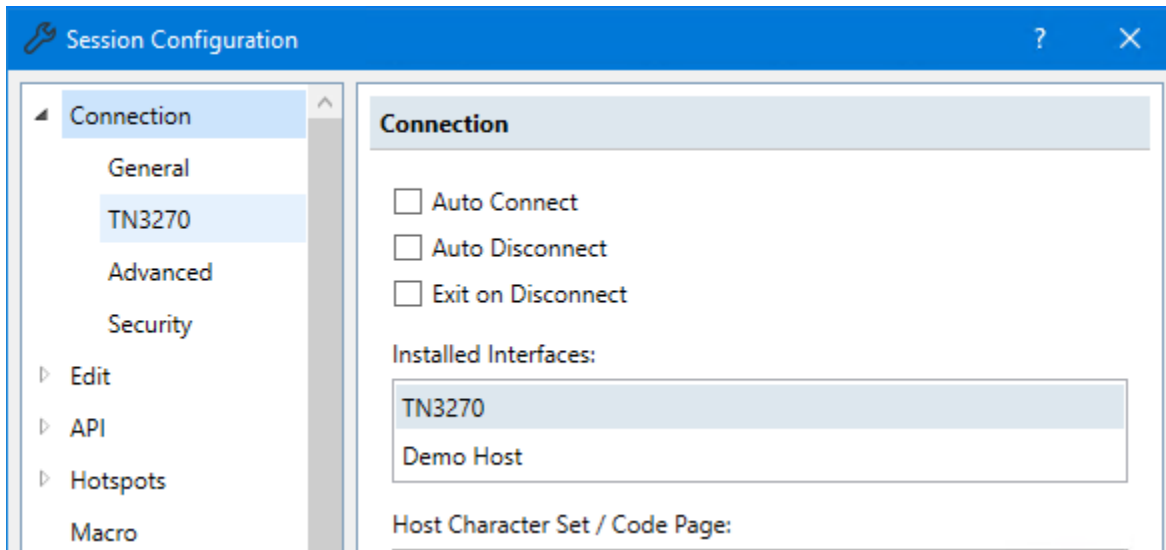
4. Elija Mainframe Display.



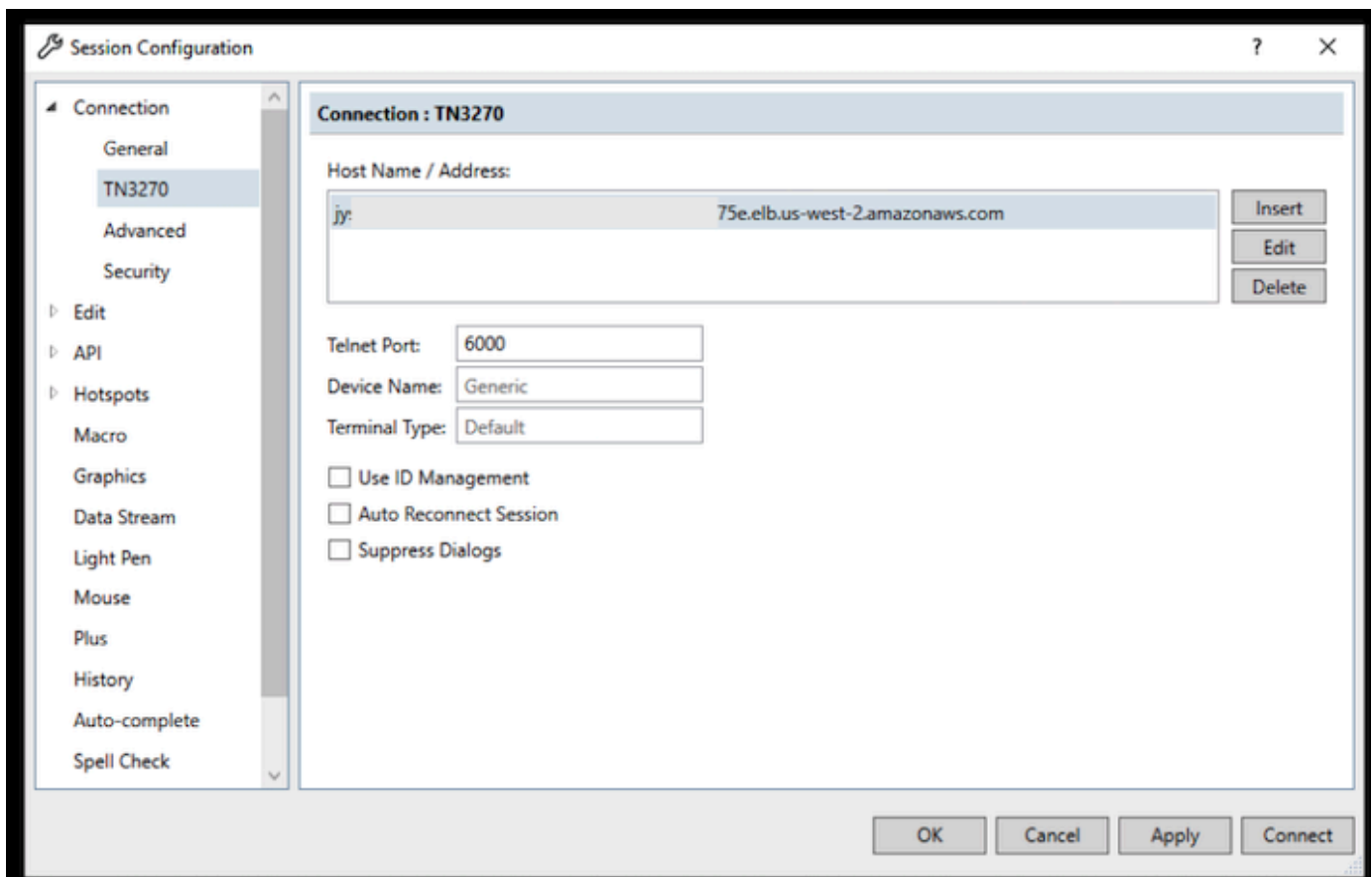
5. Selecciona Conexión y, a continuación, selecciona Configurar.




6. En Interfaces instaladas, seleccione yTN3270, a continuación, TN3270 vuelva a seleccionarla en el menú Conexión.




7. Seleccione Insertar y péguelo DNS Hostname para la aplicación. Especifique 6000 el puerto Telnet.



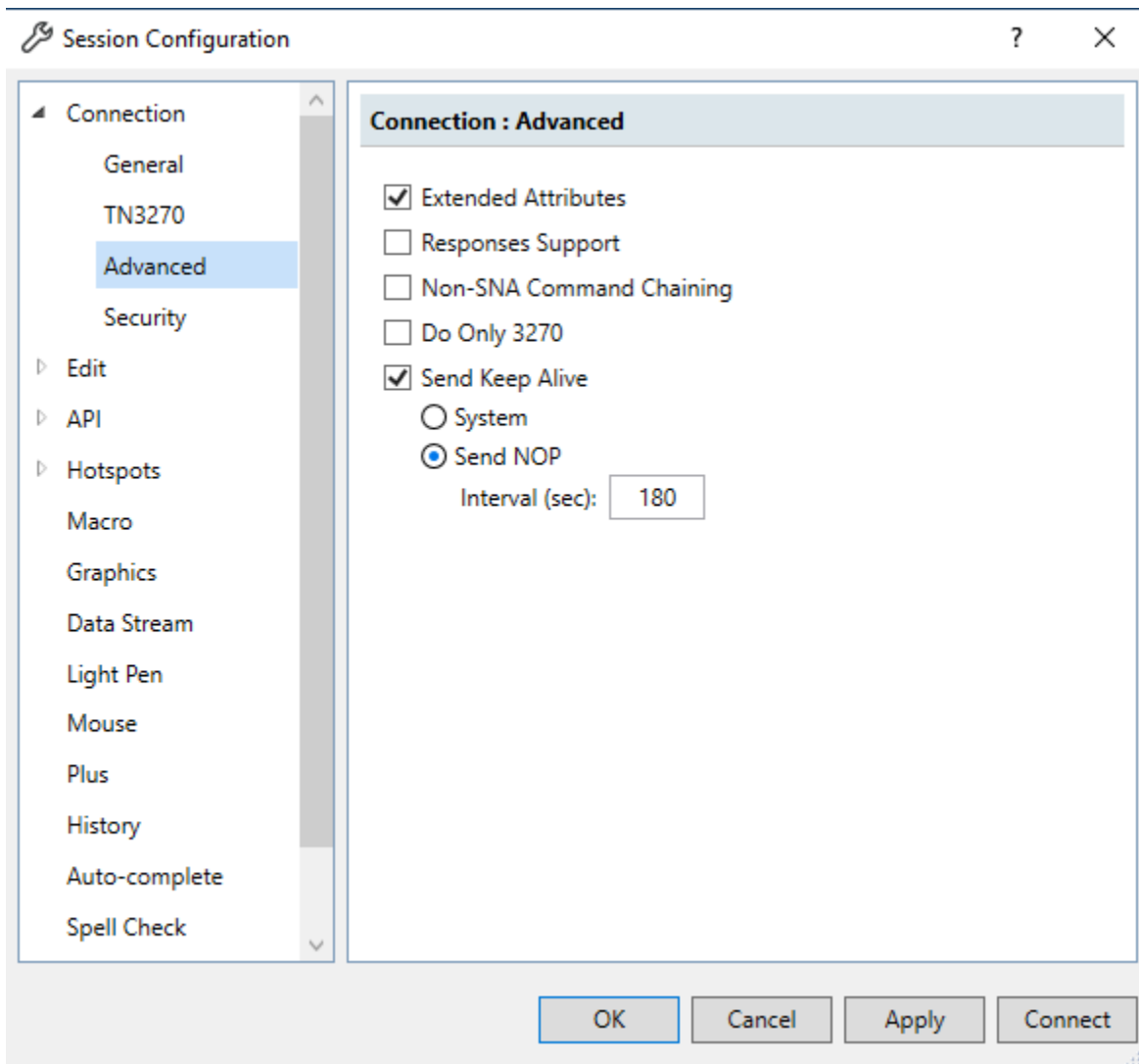
 Note

Si utiliza AWS AppStream 2.0 en un navegador y tiene dificultades para pegar valores, consulte [Solución de problemas de usuario de la AppStream versión 2.0](#).

8. En Conexión, seleccione Avanzado y, a continuación, seleccione Enviar Keep Alive y Enviar NOP e introduzca 180 para el intervalo.

 Note

Si configuras el ajuste keep alive en tu terminal TN3270 en al menos 180 segundos, te asegurarás de que el Network Load Balancer no interrumpa la conexión.



## 9. Elija Conectar.

### Note

Si se produce un error en la conexión:

- Si usa AppStream 2.0, confirme que la VPC y el grupo de seguridad especificados para el entorno de la aplicación son los mismos que los de la flota 2.0. AppStream
- Utilice el Reachability Analyzer de VPC para analizar la conexión. [Puede acceder al Reachability Analyzer a través de la consola.](#)
- Como paso de diagnóstico, intente agregar o cambiar las reglas de entrada del grupo de seguridad para que la aplicación permita el tráfico hacia el puerto 6000 desde

cualquier lugar (es decir, el bloque CIDR 0.0.0.0/0). Si se conecta correctamente, sabrá que el grupo de seguridad estaba bloqueando su tráfico. Cambie la fuente del grupo de seguridad por una más específica. Para obtener más información sobre los grupos de seguridad, consulte [Conceptos básicos sobre los grupos de seguridad](#).

10. Introduzca USER0001 el nombre de password usuario y la contraseña.

**Note**

En Rumba, el valor predeterminado para Borrar es ctrl-shift-z y el predeterminado para Restablecer es ctrl-r.

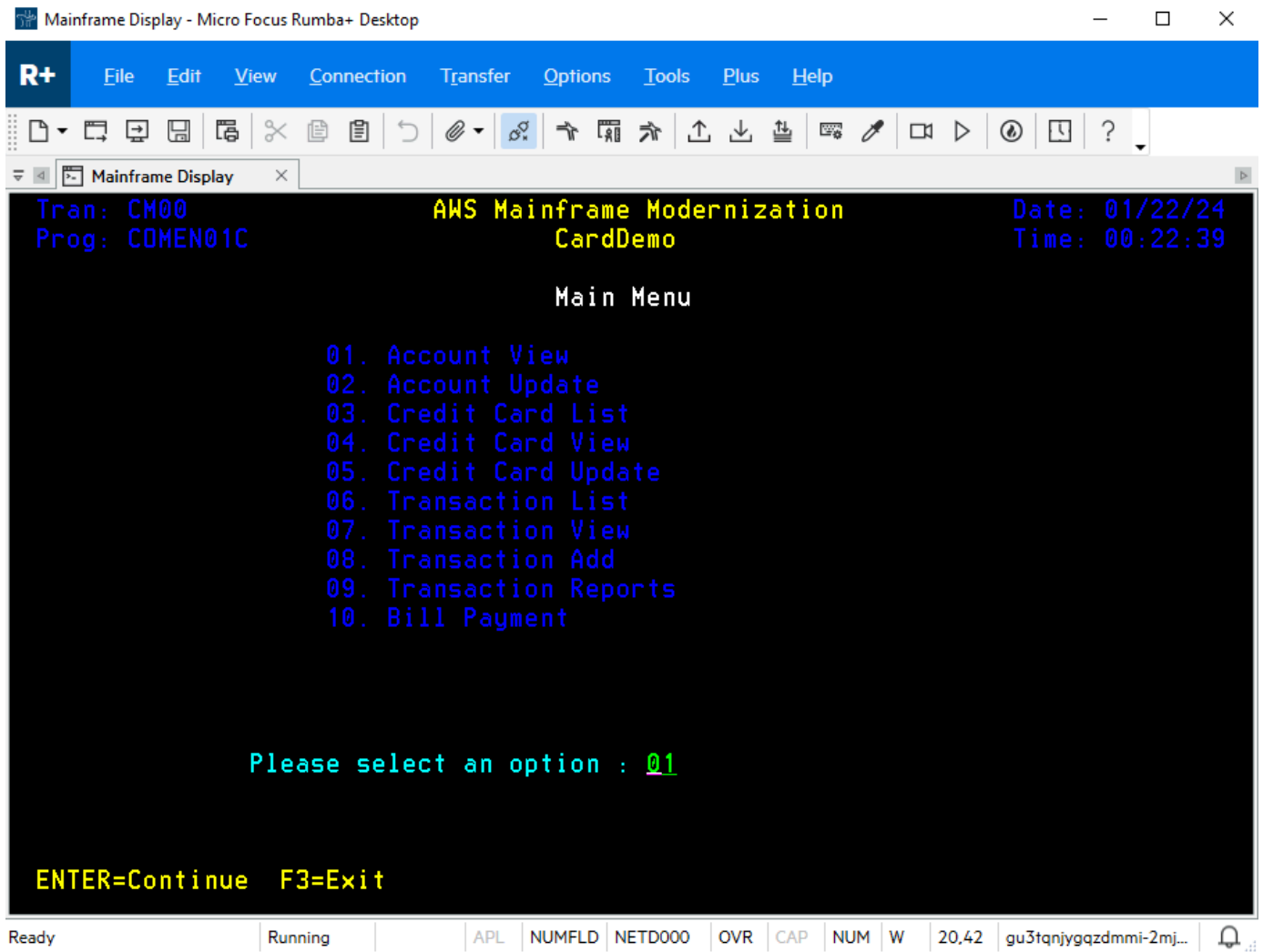
```

Mainframe Display - Micro Focus Rumba+ Desktop
R+ File Edit View Connection Transfer Options Tools Plus Help
Mainframe Display
Tran : CC00           AWS Mainframe Modernization      Date : 01/22/24
Prog : CDSGN00C      CardDemo                                               Time : 00:00:49
AppID: SBP7CMEZ      SysID: CARD
This is a Credit Card Demo Application for Mainframe Modernization
=====
|%%%%%%%% NATIONAL RESERVE NOTE %%%%%%%%%|
|%(1) THE UNITED STATES OF KICSLAND (1)%|
|%%$ $***** $$$|
|%% {x} (o o) %|
|%% ***** ( V ) ONE %|
|%(1) ---m-m--- (1)%|
|%%~~~~~ ONE DOLLAR ~~~~~%|
=====
Type your User ID and Password, then press ENTER:
User ID      : user0001 (8 Char)
Password     : (8 Char) _
ENTER=Sign-on F3=Exit
Ready | Running | APL | NUMFLD | NETB000 | OVR | CAP | NUM | W | 20.62 | gu3tqnjyqzdmml-2mj...

```

11. Después de iniciar sesión correctamente, podrá navegar por la CardDemo aplicación.

## 12. Ingresa 01 para ver la cuenta.



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

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

                                Main Menu

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

                                Please select an option : 01

                                ENTER=Continue  F3=Exit
```

At the bottom of the terminal window, there is a status bar with the following information: Ready | Running | APL | NUMFLD | NETD000 | OVR | CAP | NUM | W | 20.42 | gu3tanjyqazdmmi-2mj... | [bell icon]

## 13. Introduzca 0000000010 el número de cuenta y pulse Entrar en el teclado.

### Note

Otras cuentas válidas son 0000000011 y 0000000020.



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

```

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

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

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

                                Enter or update id of account to display

                                F3=Exit
  
```

At the bottom of the window, there is a status bar with the following information: Ready | Running | APL | NUMFLD | NETD000 | OVR | CAP | NUM | W | 5,39 | gu3tqnjyqzdmml-2mj... | [notification icon]

14. Pulse F3 para salir del menú y F3 salir de la transacción.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para evitar cargos adicionales. Para ello, complete los siguientes pasos.

- Si es necesario, detenga la aplicación.
- Elimine la aplicación. Para obtener más información, consulte [Elimine una aplicación de modernización AWS de mainframe](#).
- Eliminar un entorno de tiempo de ejecución Para obtener más información, consulte [Elimine un entorno de ejecución AWS de modernización de mainframe](#).

- Elimine los buckets de Amazon S3 que creó para este tutorial. Para obtener más información, consulte [Eliminar un bucket](#) en la Guía del usuario de Amazon S3.
- Elimine el AWS Secrets Manager secreto que creó para este tutorial. Para obtener más información, consulte [Eliminar un secreto](#).
- Elimine la clave KMS que creó para este tutorial. Para obtener más información, consulte [Eliminar claves de AWS KMS](#).
- Elimine la base de datos de Amazon RDS que creó para este tutorial. Para obtener más información, consulte [Eliminar la instancia EC2 y la instancia de base de datos](#) en la Guía del usuario de Amazon RDS.
- Si ha añadido una regla de grupo de seguridad para el puerto 6000, elimínela.

## Siguientes pasos

Para obtener información sobre cómo configurar un entorno de desarrollo para sus aplicaciones modernizadas, consulte el [tutorial: Configurar la AppStream versión 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#).

# Enfoque de modernización

La migración es compleja y tiene muchas variables. AWS La modernización de los mainframes ofrece un enfoque evolutivo que proporciona algunos beneficios a corto plazo, ya que mejora la agilidad y ofrece numerosas oportunidades para optimizar e innovar más adelante. Además, la modernización del AWS mainframe ayuda a simplificar el proceso y, al mismo tiempo, respeta los detalles de la empresa y el negocio de su cliente. Los dos enfoques principales que admite la modernización de los AWS mainframes son la refactorización automática o la replataforma. La opción que elija depende de la situación de su cliente.

La refactorización automatizada utiliza las herramientas de AWS Blu Age para convertir automáticamente el código, los datos y las dependencias a lenguajes, almacenes de datos y marcos modernos, garantizando al mismo tiempo la equivalencia funcional con las mismas funciones empresariales.

La redefinición de la plataforma utiliza las herramientas de Micro Focus para transformar las cargas de trabajo de la unidad central en servicios ágiles en AWS.

Puede pensar en el proceso de modernización por etapas. La primera etapa incluye tres fases: evaluar, movilizar y migrar y modernizar. La siguiente etapa incluye la fase de operación y optimización, en la que puede identificar más oportunidades de innovación.

## Temas

- [Fase de evaluación](#)
- [Fase de movilización](#)
- [Fase de migración y modernización](#)
- [Fase de operación y optimización](#)

## Fase de evaluación

En el nivel más alto, la fase de evaluación analiza si está preparado para migrar. Usted define un modelo de negocio y, a continuación, capacita a su equipo con talleres y un día de inmersión (demostraciones y laboratorios) ofrecido por AWS. Los talleres y los días de inmersión abordan diferentes temas. Estas tareas se llevan a cabo fuera de la modernización del mainframe. AWS

## Fase de movilización

En la fase de movilización, el proyecto comienza desde el principio y, a continuación, se ejecuta un proceso de descubrimiento que extrae los datos de las aplicaciones de su unidad central y los incorpora a una herramienta de migración. Identifica las aplicaciones que desea migrar y selecciona algunas de ellas para ponerlas a prueba. Usted perfecciona su modelo de negocio, redacta su plan de migración y decide cómo quiere gestionar la seguridad y el cumplimiento, la gobernanza de las cuentas y su modelo operativo. Configura un centro de excelencia en la nube con las personas adecuadas de su equipo. Ejecuta los programas piloto y documenta lo que ha aprendido. Es usted quien refine su plan de migración y su modelo de negocio. Muchas de estas tareas se llevan a cabo fuera de la modernización del AWS mainframe.

## Fase de migración y modernización

La fase de migración y modernización se aplica a cada aplicación y consta de varias tareas, entre las que se incluyen la asignación de personal, la realización de un descubrimiento exhaustivo, la definición de la arquitectura de la aplicación adecuada, la configuración de los entornos de ejecución de las aplicaciones AWS, la reconfiguración de la plataforma o la refactorización del código, la integración con otros sistemas y, por supuesto, las pruebas. Al final de la fase, debe implementar en producción las aplicaciones se han reconfigurado o cuya plataforma se ha redefinido y pasar al nuevo sistema de AWS. La mayoría o la totalidad de estas tareas se llevan a cabo en AWS Mainframe Modernization, en otro AWS servicio o en una herramienta a la que Mainframe Modernization dé acceso. AWS

[Si desea utilizar la refactorización automatizada, consulte Blu Insights.](#) AWS Blu Insights ya está disponible en la AWS Management Console mediante el inicio de sesión único. Ya no tiene que gestionar credenciales de AWS Blu Insights independientes. Puede acceder a las funciones de AWS Codebase y AWS Transformation Center directamente desde la AWS Management Console.

Para migrar datos del mainframe a AWS, recomendamos el [AWS SCT](#) y el [AWS Database Migration Service](#). Para obtener más información, consulte [What is the AWS Schema Conversion Tool?](#) en la guía del usuario de AWS Schema Conversion Tool y [¿Qué es AWS Database Migration Service?](#) en la Guía del usuario de AWS Database Migration Service.

## Fase de operación y optimización

En la fase de operación y optimización, usted se centra en supervisar las aplicaciones implementadas, administrar los recursos y garantizar que la seguridad y el cumplimiento estén actualizados. También evalúa las oportunidades para optimizar las cargas de trabajo migradas.

# Conceptos

AWS La modernización del mainframe proporciona herramientas y recursos para ayudarlo a migrar, modernizar y ejecutar las cargas de trabajo del mainframe. AWS

## Temas

- [Aplicación](#)
- [Definición de aplicación](#)
- [Trabajo por lotes](#)
- [Configuración](#)
- [Conjunto de datos](#)
- [Entorno](#)
- [Modernización de la unidad central](#)
- [Viaje migratorio](#)
- [Punto de montaje](#)
- [Refactorización automatizada](#)
- [Recreación de plataformas](#)
- [Recurso](#)
- [Motor de tiempo de ejecución](#)

## Aplicación

Una carga de trabajo de mainframe en ejecución en Mainframe Modernization. AWS Una aplicación está formada por un conjunto de trabajos por lotes, transacciones interactivas (CICS o IMS) u otros componentes. Usted define el alcance. Debe definir y especificar los componentes o recursos que necesite la carga de trabajo, como las transacciones del CICS o los trabajos por lotes.

## Definición de aplicación

La definición o especificación de los componentes y recursos que necesita una aplicación (carga de trabajo de mainframe) que se ejecuta en AWS Mainframe Modernization. Es importante separar la definición de la aplicación en sí, ya que es posible reutilizar la misma definición para varias etapas (preproducción, producción), representadas por diferentes entornos de tiempo de ejecución.

# Trabajo por lotes

Un programa planificado que se ha configurado para ejecutarse sin requerir la interacción del usuario. En AWS Mainframe Modernization, necesitará almacenar los archivos de JCL de los trabajos por lotes y los archivos binarios de los trabajos por lotes se preparan en un bucket de Amazon Simple Storage Service (Amazon S3), y proporcionar la ubicación de ambos en el archivo de definición de la aplicación. Al ejecutar un trabajo por lotes, AWS Mainframe Modernization informa de los siguientes valores de estado:

## Submitting

El trabajo por lotes está en proceso de enviarse.

## Holding

El trabajo por lotes está en espera.

## Dispatching

El trabajo por lotes está en proceso de enviarse.

## Running

El trabajo se está ejecutando.

## Cancelling

El trabajo por lotes está en proceso de ser cancelado.

## Cancelled

El trabajo por lotes está cancelado.

## Succeeded

El trabajo por lotes ha terminado de ejecutarse correctamente.

## Failed

Se ha producido un error en el trabajo por lotes.

## Succeeded With Warning

El trabajo por lotes ha terminado de ejecutarse correctamente y se informó de un error menor. El código de condición del trabajo devuelto como parte de la `GetBatchJobExecution` respuesta indica la causa del error.

## Configuración

Las características de un entorno o una aplicación. Las configuraciones del entorno consisten en el tipo de motor, la versión del motor, los patrones de disponibilidad, las configuraciones opcionales del sistema de archivos y más.

Las configuraciones de las aplicaciones pueden ser estáticas o dinámicas. Las configuraciones estáticas solo cambian cuando se actualiza una aplicación mediante la implementación de una nueva versión. Las configuraciones dinámicas, que suelen ser una actividad operativa, como activar o desactivar el rastreo, cambian en cuanto se actualizan.

## Conjunto de datos

Archivo que contiene datos para su uso por parte de las aplicaciones.

## Entorno

Una combinación con nombre de recursos AWS informáticos, un motor de tiempo de ejecución y detalles de configuración creada para alojar una o más aplicaciones.

## Modernización de la unidad central

El proceso de migración de aplicaciones de un entorno de mainframe heredado a AWS

## Viaje migratorio

El end-to-end proceso de migración y modernización de las aplicaciones heredadas, normalmente consta de las siguientes fases: evaluar, movilizar, migrar y modernizar, y operar y optimizar.

## Punto de montaje

Directorio de un sistema de archivos que proporciona acceso a los archivos almacenados en ese sistema.



## Refactorización automatizada

El proceso de modernizar los artefactos de las aplicaciones heredadas para que se ejecuten en un entorno de nube moderno. Puede incluir la conversión de código y datos. Para obtener más información, consulte [AWS Mainframe Modernization Automated Refactor](#).

## Recreación de plataformas

Proceso de trasladar una aplicación y sus artefactos de una plataforma informática a otra diferente. Para obtener más información, consulte [AWS Mainframe Modernization Replatform](#)

## Recurso

Componente físico o virtual de un sistema informático.

## Motor de tiempo de ejecución

Software que facilita la ejecución de una aplicación.

# Refactorización automática de aplicaciones con AWS Blu Age

La refactorización automatizada con AWS Blu Age proporciona una end-to-end solución para migrar y modernizar sus aplicaciones de mainframe. Los pasos del proceso de refactorización son los siguientes:

- Análisis del inventario
- Análisis de las dependencias
- Transformación automática del código
- Captura y administración de escenarios de prueba

Puede completar los pasos anteriores en la herramienta Blu Insights, disponible mediante el inicio de sesión único desde la consola de modernización del mainframe. AWS Para obtener más información sobre Blu Insights, consulte la [documentación de Blu Insights](#).

Cuando esté satisfecho con el código fuente transformado, es el momento de continuar AWS y completar los siguientes pasos:

- Cree e implemente la aplicación refactorizada.
- Implemente y supervise su aplicación en AWS Mainframe Modernization.

AWS Blu Age Runtime (no gestionado) es una de las ofertas del servicio de modernización de AWS mainframe junto con el servicio gestionado de AWS Blu Age. Con la gestión de AWS Blu Age, puede implementar su aplicación modernizada en un entorno AWS gestionado que simplifica su experiencia, de modo que no tendrá que gestionar la infraestructura subyacente en la que se ejecuta la aplicación modernizada. Por el contrario, con AWS Blu Age Runtime (no administrado), puede implementar su aplicación modernizada en su propia AWS cuenta, de modo que pueda administrar su propia infraestructura. Con AWS Blu Age Runtime (no gestionado), tiene la flexibilidad de utilizar todos los componentes técnicos necesarios para ejecutar su aplicación modernizada de la forma que desee.

AWS Blu Age Runtime (no administrado) está disponible para su implementación en:

- Amazon EC2

- Amazon ECS en Amazon EC2
- Amazon EKS en Amazon EC2
- Amazon ECS gestionado por AWS Fargate

La implementación en Amazon EC2 (las tres primeras opciones de la lista anterior) se puede realizar directamente en la instancia o mediante una aplicación contenerizada de Docker, que es la forma preferida cuando se utiliza Amazon ECS o Amazon EKS.

#### Temas

- [AWS Notas de lanzamiento de Blu Age](#)
- [AWS Instrucciones de actualización de Blu Age](#)
- [AWS Conceptos de Blu Age Runtime](#)
- [AWS Archivos de configuración y configuración de Blu Age Runtime](#)
- [AWS API de ejecución de Blu Age](#)
- [AWS Configuración de Blu Age Runtime \(no gestionado\)](#)
- [Modificación del código fuente con el IDE para desarrolladores de Blu Age](#)

## AWS Notas de lanzamiento de Blu Age

Esta sección contiene las notas de la versión de AWS Blu Age Runtime and Modernization Tools a partir de la versión 3.5.0, primero la más reciente, organizadas por número de versión.

#### Note

Para ver las notas de la versión anteriores a este documento, ponte en contacto con los servicios de entrega de AWS Blu Age. Para obtener información sobre las últimas características de Blu Insights, consulte las [versiones de Blu Insights](#).

#### Temas

- [Notas de la versión 4.1.0](#)
- [Runtime, versión 4.1.0](#)
- [Herramientas de modernización, versión 4.1.0](#)
- [Notas de la versión 4.0.0](#)

- [Runtime, versión 4.0.0](#)
- [Herramientas de modernización, versión 4.0.0](#)
- [Notas de la versión 3.10.0](#)
- [Runtime, versión 3.10.0](#)
- [Herramientas de modernización, versión 3.10.0](#)
- [Notas de la versión 3.9.0](#)
- [Runtime, versión 3.9.0](#)
- [Herramientas de modernización, versión 3.9.0](#)
- [Notas de la versión 3.8.0](#)
- [Runtime, versión 3.8.0](#)
- [Herramientas de modernización, versión 3.8.0](#)
- [Notas de la versión 3.7.0](#)
- [Runtime, versión 3.7.0](#)
- [Herramientas de modernización, versión 3.7.0](#)
- [Notas de la versión 3.6.0](#)
- [Runtime, versión 3.6.0](#)
- [Herramientas de modernización, versión 3.6.0](#)
- [Notas de la versión 3.5.0](#)
- [Runtime, versión 3.5.0](#)
- [Herramientas de modernización, versión 3.5.0](#)

## Notas de la versión 4.1.0

Esta versión de AWS Blu Age Runtime and Modernization Tools se centra en el rendimiento y la seguridad. Algunas funciones y cambios clave de esta versión son:

- Transformación y rendimiento: para permitir que los proyectos con una base de código grande (más de 50 millones de líneas de código) se transformen con éxito, hemos optimizado el rendimiento y el consumo de memoria de todo el mecanismo de transformación.
- BAC/JAC: la seguridad es la máxima prioridad. AWS Las aplicaciones modernizadas con AWS Blu Age deben cumplir con los estándares de seguridad. Hemos realizado algunas mejoras importantes en la Consola de BluSam administración (BAC) y la Consola de administración del JICS (JAC) para hacerlas más seguras:

- Se actualizó la aplicación a Angular v17.
- Además del soporte nativo para AWS Cognito, añadimos soporte genérico para OAuth, que ofrecerá una mayor flexibilidad para permitir a los clientes utilizar el proveedor de identidades que prefieran.
- Configuramos y ampliamos las funciones de seguridad mediante los encabezados adecuados.
- AS400: soporte de varios nodos para el mecanismo de bloqueo de la base de datos. Proporcionó la posibilidad de conectar un servidor de almacenamiento en caché externo y compartido (Redis) para ejecutar una aplicación por lotes en varias instancias, como la modernización gestionada del mainframe. AWS

Esta versión del motor de ejecución de Blu Age se probó con la siguiente pila. Es posible que otras versiones también sean compatibles.

Componente	Versión probada
Java	Java 17
Capa de presentación	Nodo JS 18.18, Npm 9.8, Angular 16.1
Capa de servicio	Spring Boot 3.2.5, Spring Core 6.1.5, Spring Statemachine 4.0.0
Capa de persistencia	PostgreSQL 14, Oracle 21c
Servidor de aplicaciones	Apache Tomcat 10.1.17

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 4.1.0

### zOS

#### Nuevas características

- Se agregó una configuración para el manejo dinámico de proveedores de OAuth2. Se introdujo `SECRET_OAUTH2_PROVIDER_NAME_KEY` para especificar el proveedor. Se actualizó el

método de recuperación de secretos para gestionar varios proveedores. Los secretos garantizados se recuperan de forma segura de AWS Secrets Manager.

- Se ha añadido la compatibilidad con las propiedades del SSL de DB2 para poder definir un certificado SSL (TrustStoreubicación ssl) y una contraseña (TrustStorecontraseña ssl) para desbloquear el archivo del almacén de claves. AWS Secrets Manager
- Se agregó soporte para fuentes de datos empresariales externas.
- JCL: Se agregó soporte para el mecanismo de puntos de control para el reinicio por lotes.
- JCL: se agregó soporte para los parámetros DCB, tamaño de registro y RDW.
- JCL: se agregó una configuración dinámica de nombres de carpetas para los archivos temporales generados.
- REDIS: se agregó una configuración de grupo en la configuración de Redis para JICS.
- REDIS: índice de base de datos agregado en la configuración de Redis para Catalog y JICS.
- BatchScript - Se agregó la propagación del nombre del paso para ejecutar el programa.
- CICS: se agregó soporte para el comando ADDRESS SET.
- CICS: se agregó soporte para PURGE MESSAGE y JUSTIFY.

## Mejoras

- JCL - INFUTILB - Soporte mejorado para deshabilitar el indicador nulo en función de la propiedad YML.
- JCL - INFUTILB: compatibilidad mejorada con el tipo de datos CHAR/BPCHAR.
- JCL - ICEGENER - Se agregó soporte para copiar flujos de entrada multilínea en archivos.
- JCL - IEBGENER - Soporte mejorado para gestionar la conversión de archivos de bloque variable a bloque fijo.
- JCL - DFSORT: compatibilidad mejorada con parámetros de varios dígitos en la fecha de operación.
- JCL - DFSORT: se agregó soporte para la cláusula INCLUDE=ALL.
- JCL: compatibilidad mejorada con la utilidad SORT para gestionar el campo BDW en la salida.
- JCL: soporte mejorado para la concatenación de DD.
- JCL: soporte mejorado para Input Stream.
- JCL - DSNUTILB - Soporte mejorado para la sentencia NULLIF ().
- JCL - INFUTILB: se agregó soporte para descargar datos con la opción NOPAD.

- JCL - INFUTILB: compatibilidad mejorada con la fecha actual en INFUTILB.
- JCL - Se han añadido comprobaciones de tamaño y existencia de archivos antes de utilizarlos.
- JCL - GDG - Se ha mejorado el manejo de los subdirectorios de GDG.
- MQ: apertura de conexión mejorada en la implementación de JMS.
- MQ: configuración de longitud de datos mejorada del mensaje GET para la fuente de datos XA.
- MQ: cuaderno estándar CMQV descompuesto para evitar errores de compilación y usos de refactorización.
- BluSam - Soporte mejorado para las solicitudes de eliminación de conjuntos de datos inexistentes.
- Soporte mejorado para la declaración ALLOCATE.
- Se ha mejorado la solidez de la nomenclatura TS-QUEUE.
- BatchScript - Se ha mejorado la conservación del código de retorno del paso anterior al volver a ejecutar un trabajo.
- Conjunto de datos: se mejoró la comprobación de la existencia del archivo cuando existe un archivo y es temporal.
- Conjunto de datos: se ha mejorado la simultaneidad a la hora de localizar los archivos GDG que se van a eliminar.
- Conjunto de datos: se agregó soporte para obtener el tamaño de registro del conjunto de datos GDG.
- CICS: compatibilidad mejorada con la opción SUSPENDED del comando INQUIRE TASK LIST.
- CICS: soporte mejorado para LOAD SET mediante la dirección de la sentencia.
- CICS: Se mejoró el sistema remoto de argumentos CICS no controlados cuando CICS CONSULTABA.
- CICS: compatibilidad mejorada con el comando GETMAIN para gestionar la opción SET con un puntero definido con la palabra clave OF.
- JICS: se mejoró la solidez del método jicsXAPRepare () al agregar la verificación del estado de la transacción.
- JICS XA: se agregó una verificación del estado de la transacción y se mejoró la terminación del hilo de transacciones.
- BAC: se mejoró la autenticación basada en roles en el lado del cliente y se refactorizaron y centralizaron todas las llamadas a la API.
- BAC: se implementó una función para bloquear el acceso público a BAC y JAC según la configuración

- BAC - Actualización de las dependencias: Angular 17.
- BAC: integración de seguridad mejorada con OAuth2 - /FIDIS. StateFarm
- BAC: DDL mejorado generado por hibernación.
- BAC: mecanismo mejorado de exportación de conjuntos de datos.
- JAC: actualizado a Angular 17 e informa sobre todos los trabajos específicos del BAC (ROLE, sadmin conf, XSRF, logout).
- COBOL: se agregó soporte para las funciones CHAR y ORD-MIN.
- Se ha mejorado FileFactory para mantener el tamaño del registro del catálogo al estar a disposición de los MOD.
- Se habilitó el registro mediante MDC para las transacciones de JICS.
- SQLCA > SQLSTATE mejorado creado para procedimientos almacenados que generan conjuntos de resultados ad hoc.
- Se ha mejorado la compatibilidad con la programación de tareas relacionada con la última actualización de primavera.

## AS400

### Nuevas características

- Se ha añadido compatibilidad con varios nodos para bloquear los registros de bases de datos mediante Redis.
- Se agregó soporte para caracteres binarios para el tipo DDS.
- CL: se agregó soporte para la generación de archivos de informes personalizados.
- RPG: se ha añadido compatibilidad con la palabra clave RENAME en los archivos primarios/ secundarios.

### Mejoras

- Soporte de base de datos mejorado para gestionar la columna CTID con una cláusula JOIN.
- Se ha mejorado la posición del cursor para varios DSPATR (PC).
- Excepción de registro al leer mejorada.
- Se mejoró el registro de trabajos de Quartz para incluir las propiedades de los trabajos en MDC.
- Soporte mejorado para la pantalla de ayuda del AS400.



- CL: compatibilidad mejorada con el comando RMVJOBSCDE para aceptar números de entrada con espacios al final.
- CL - Se ha mejorado la compatibilidad con el comando RMVJOBSCDE para eliminar un cronograma de tareas utilizando un nombre de trabajo genérico.
- CL: compatibilidad mejorada con el comando SAVOBJ para ordenar los registros por clave de tabla.
- CL: compatibilidad mejorada con el comando CPYF para establecer una nueva conexión para las consultas a la base de datos.
- CL: se ha mejorado la inserción de los mensajes de consulta en los mensajes de cola con SNDPGMMSG.
- CL - Se ha mejorado la configuración de la cola de trabajos para especificar la cola de trabajos predeterminada.
- CL: se mejoró el comando CRTPF para que sea compatible con la biblioteca QTEMP y el parámetro RCDLEN.
- CL - Soporte mejorado para el comando CHKOBJ - Compruebe si hay particiones con la biblioteca.
- CL: RTVMGS mejorado para enviar CPF2407 y CPF2419 cuando no se encuentra el archivo/ID.
- CL - Se ha mejorado la interpretación de los parámetros de formato antiguos con CPYTOIMPF y CPYFRMIMPF.
- CL: se agregó soporte para el parámetro USRDTA de OVRPRTF.
- CL: se mejoró el comando CL CPYTOIMPF para establecer una nueva conexión y evitar el cierre de los conjuntos de resultados existentes.
- CL: se mejoró el CHGDTAARA para que ya no modifique la longitud del área de datos al actualizar el contenido.
- CL: manejo mejorado CCommand de la conexión a la base de datos.
- Interacción optimizada entre el front-end y el back-end.
- COBOL: se ha actualizado la transformación para utilizar FILLER en los cuadernos.
- Se ha mejorado la visualización adicional de la información de los mensajes para los mensajes personalizados que se envían a la interfaz.
- Se actualizó el valor predeterminado del selector en app.component.ts.
- Se ha mejorado la división del texto en la pantalla. split-dynamic-field
- Se ha mejorado la visualización de los mensajes de error con varias escrituras seguidas de una lectura.

## Capacidades transversales

### Nuevas características

Se ha añadido soporte para la configuración dinámica del secreto del proveedor OAuth2.

### Mejoras

- Impresión: se ha mejorado la compatibilidad con los parámetros QCMDEXC para gestionar las comillas y se ha mejorado la formación de los nombres de los informes
- Se ha mejorado la compatibilidad con la sintaxis delimitada activada. RecordAdaptable
- Se ha mejorado InspectBuilder el registro de errores para añadir contexto a la cadena de origen.
- DataSimplifier - mayor robustez para ByteArray la afectación.
- Registro MDC mejorado con nuevos atributos de tiempo de ejecución.

## Herramientas de modernización, versión 4.1.0

### zOS

#### Nuevas características

- Se agregó soporte para múltiples transformaciones de archivos CSD
- COBOL: Se agregó soporte para la declaración CICS ALLOCATE.
- COBOL: Se ha añadido la compatibilidad con el error ON SIZE en la sentencia ADD CORRESPONDIENTE.
- COBOL: se ha añadido soporte para el párrafo EXIT.

#### Mejoras

- COBOL: soporte mejorado para el libro de texto -INC.
- COBOL: soporte mejorado para la inicialización de FILLER.
- COBOL: soporte mejorado para la comparación de valores figurativos.
- COBOL: compatibilidad mejorada con WHEN ANY en cláusulas WHEN consecutivas que carecen de bloques de códigos intermedios.
- COBOL: soporte mejorado para la constante figurativa.
- COBOL: soporte mejorado para el cálculo del tamaño de letra empaquetado.

- COBOL: se ha mejorado el argumento CICS no controlado KEEP para SPOOLCLOSE.
- COBOL: generación mejorada de la función TEST-NUMVAL.
- COBOL: se han mejorado los argumentos de generación de Java gracias a la compatibilidad con el marco INSPECT.
- CICS: soporte mejorado para definir DFHCOMMAREA.

## AS400

### Nuevas características

- RPG: se agregó un mecanismo de detección de errores para generar el DDS (incompleto) y no bloquear la generación de programas.
- Se agregó soporte para la palabra clave de especificación de descripción del archivo INCLUDE.

### Mejoras

- RPG: análisis completamente gratuito mejorado.
- RPG: mayor robustez con la detección de errores.
- RPG: inicialización mejorada de Field/DS con la palabra clave de exportación.
- RPG: funcionamiento DAO mejorado para gestionar los indicadores.
- RPG: manejó el valor predeterminado de PERRCD con CTDATA.
- RPG: se actualizó el analizador Free-RPG para registrar un error único por regla de análisis.
- PRTF: gestionó la colisión de nombres entre PRTF y JRXML.
- COBOL: compatibilidad mejorada con la palabra clave LIKE.

## Capacidades transversales

### Mejoras

- Se agregó solidez a la API ErrorID
- Optimización del rendimiento para la transformación de grandes proyectos. Por ejemplo: tiempo de espera para omitir archivos bloqueados, reutilización de la clasificación de Blu Insights y mejores asignaciones de memoria.
- Se optimizó el consumo de memoria durante la transformación a COBOL/PL1.

- Se corrigió el CVE en un tercero (jQuery y bootstrap).
- Se administraron las opciones de TimeoutParser en TC.
- Se mejoró la reescritura de múltiples espacios en las consultas SQL.
- Cursor de solo lectura mejorado con atributo de sensibilidad.

## Notas de la versión 4.0.0

Para obtener instrucciones sobre cómo migrar de AWS Blu Age Runtime 3.10.0 a 4.0.0, consulte [the section called “Migración de la versión 3.10.0 a la 4.0.0”](#)

Esta versión de AWS Blu Age Runtime and Modernization Tools se centra en actualizar las dependencias críticas y las tecnologías compatibles, a la vez que se mejora el rendimiento en múltiples funcionalidades. Algunas funciones y cambios clave de esta versión son:

- Actualice Spring Boot 2.7 a 3.2.4, Spring Core 5.3 a 6.1.5 y Tomcat 9.0 a 10.1.17 para mejorar la seguridad, el rendimiento y la facilidad de mantenimiento mediante el uso de versiones que están siendo parcheadas y mantenidas de forma activa.
- La carga diferida de la aplicación front-end permite crear proyectos grandes más rápidos con más de 2000 pantallas y reducir la inicialización de la pantalla de 10 s a 300 ms.
- Support para la visualización DBCS en la aplicación front-end. Se ha mejorado la compatibilidad con los caracteres de doble byte para ofrecer una nueva fuente que permita gestionar los caracteres de un byte y de un byte, evitar la introducción de un solo byte en un campo de doble byte y gestionar los campos en los que se mezclan caracteres de un byte y de un byte.
- Función de monitoreo de subprocesos para la aplicación AS400 Online para ejecutar la aplicación AS400 con paralelización.
- Se mejoró el rendimiento en función del contexto y RunUnit la inicialización al añadir un mecanismo configurable para preinicializar el contexto del programa, lo que reduce el impacto de cargar estructuras complejas inherentes a la complejidad heredada.

Esta versión de AWS Blu Age Runtime se probó con la siguiente pila. Es posible que otras versiones también sean compatibles.

Componente	Versión probada
Java	Java 17

Capa de presentación	Nodo JS 18.18
	Npm 9.8
	Angular 16.1
Capa de servicio	Spring Boot 3.2.4
	Spring Core 6.1.5
	Spring Statemachine 4.0.0
Capa de persistencia	Motor PostgreSQL 14
	Oracle 21
Servidor de aplicaciones	Apache Tomcat 10.1.17

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 4.0.0

### zOS

#### Nuevas características

- Se agregó soporte para la declaración de inclusión «-INC CPYNAME».
- CICS: se agregó soporte para la declaración PUSH/POP HANDLE.
- COBOL: se agregó soporte para «ASSIGN TO DYNAMIC».
- Se agregó soporte para DB2 UNLOAD mediante INFUTILB.
- Se agregó soporte para la palabra clave SEQNUM en una superposición de la sentencia INREC.

#### Mejoras

- SORT: se agregó soporte para caracteres especiales (paréntesis y asteriscos) en los literales de cadena de clasificación C'... '.
- SORT: compatibilidad mejorada con el argumento UTFIL NOMATCH- (..).

- SORT: se agregó soporte para la definición de datos de SYMNames.
- SORT: manejo mejorado de los argumentos TO= y LENGTH=.
- SORT: manejo mejorado de la disposición de los MOD.
- SORT: se agregó soporte para el argumento HIT=NEXT.
- ICEGENER mejorado para añadir soporte para la codificación específica de archivos de salida.
- INFUTILB: compatibilidad mejorada con la cláusula WITH UR.
- INFUTILB: soporte mejorado para descargar cuando es falso. writeNullIndicator
- DSNUTILB: robustez mejorada para cargar el paso cuando la palabra clave NULLIF aparece después de una palabra clave SQL opcional.
- DSNUTILB: compatibilidad mejorada para aislar el nombre de la columna.
- DSNUTILB: se agregó soporte para cargar un archivo vacío en una tabla.
- DNSUTILB: Se agregó soporte para la disposición de mods en el archivo SYSDISC DNSUTILB.
- IDCAMS: soporte de comentarios mejorado.
- JCL: se agregó soporte para columnas con comillas dobles. LoadTask
- JCL: manejo mejorado de consultas SQL de UNLOAD relacionadas con la eliminación de espacios en blanco.
- JCL: respuesta mejorada del script Groovy cuando se produce una excepción en el procesamiento para garantizar un formato JSON.
- JCL: se ha mejorado la disposición de los archivos de verificación en los casos de DISP=NEW y DISP=OLD.
- JCL: soporte mejorado para gestionar múltiples referencias de generación de GDG con caracteres especiales en el nombre base del GDG.
- JCL: soporte mejorado para cargar un archivo ficticio.
- JCL: soporte mejorado para tempFilesDirectory el parámetro YML.
- JCL: se ha mejorado el retorno de JSON cuando es necesario evitar las comillas dobles dentro de un elemento de cadena.
- JCL: mejorado FileUtils para admitir el nombre base GDG.
- JCL: programa DSNTEP mejorado para la ejecución de múltiples consultas de DB2.
- Se agregó soporte para Spring beans.
- SQLConverter mejorado para evitar rectificar fechas incorrectas.
- JicsTimeBuilder Manejo mejorado de YYYYDDD.

- Se permitió acceder a los tarros personalizados desde Groovy.
- IMS: navegación mejorada entre los registros en la implementación de la base de datos IMS.
- IMS: CBLTDLI mejorado para poder iniciar el programa use purge.
- IMS: el DFSRRC00 es capaz de pasar los parámetros de groovy al programa de fondo.
- Se agregó soporte para el comando JICS que no se invocó a través de un TransactionRunner.
- JICS: rendimiento mejorado mediante el uso de una caché configurable.
- BluSam - Se ha añadido soporte para deshabilitar el calentamiento BluSam al abrir, a fin de mejorar el rendimiento en conjuntos de datos de gran tamaño.
- BluSam- Se ha mejorado el comportamiento de borrar/cambiar el nombre en conjuntos de datos normales. BluSam
- BluSam - Rendimiento mejorado en las operaciones de registro.
- Simplificador de datos mejorado para los métodos que determinan si una cadena tiene un valor bajo.
- Soporte mejorado para problemas de orden de clasificación y decimales empaquetados.
- Configuración mejorada de DB2 como fuente de datos principal con AWS Secrets.
- FileSystem API mejorada para mostrar el estado del archivo.
- Entrada de flujo de DynamicFileBuilder lectura mejorada con LineSeparator.
- Simplificador de datos mejorado para los métodos que determinan si una cadena tiene un valor bajo cuando se trata del juego de caracteres CUSTOM930.
- SQL: procesamiento mejorado de la salida de los procedimientos almacenados de SQL.
- SQL: mapeo lambda mejorado para múltiples tablas con alias.
- COBOL: compatibilidad mejorada con la sentencia LENGTH OF.
- COBOL: se agregó soporte para la declaración TRANSFORM.
- COBOL: se agregó soporte para 9 nuevas funciones matemáticas.
- COBOL: soporte mejorado para la función de números enteros del día.
- COBOL: soporte mejorado para 88 niveles que implican valores figurativos.
- COBOL: transformación mejorada para la declaración SET ADDRESS.

## AS400

### Nuevas características

- Se eliminaron las entidades indicadoras duplicadas.
- Se ha añadido soporte para los personajes del DBCS.
- Se introdujo el manejo de la palabra clave HELP para el control de registros de subarchivos.
- Se agregó un parámetro de configuración para alternar el uso de mayúsculas en los nombres de las columnas y dividir el contenido de las columnas de comentarios en el carácter vertical.
- Se ha añadido la posibilidad de utilizar 0x0c como último fragmento en los campos de tipo empaquetado.
- RPG: manipulaba prototipos declarados con ExtProc ('sistema').
- CL: el parámetro «CLEAR» del cl-command RMVMSG + introduce colas de mensajes en la memoria que no están programadas.
- CL: gestionaba las declaraciones genéricas que se pasaban a las llamadas de SBMJOB CMD ().
- CL: se agregó el comando STRCMTCTL y ENDCMTCTL. Se modificó el mecanismo de bloqueo y se eliminaron las transacciones y los bloqueos.
- CL: Se agregó soporte para el parámetro RCDDLML para el comando CPYTOIMPF.
- CL: se agregó el manejo del relleno de ceros en el comando SAVOBJ.
- CL: se agregó el manejo de las bibliotecas incluidas en el nombre completo del parámetro OBJ para RTVOBJD.
- CL: Se agregó soporte para los parámetros de comando CPYTOIMPF STRDLM, STRESCCHR y RMVBLANK.
- CL: RTVMGS mejorado para enviar CPF2407 y CPF2419 cuando no se encuentra el archivo/identificador.
- CL: comando RCVF mejorado para recibir registros de cualquier biblioteca proporcionada en el parámetro DEV.

## Mejoras

- Se modificaron los valores predeterminados del ejecutor de tareas Blu4iv para permitir un mejor escalado de forma predeterminada.
- Parameterhelper modificado para convertir la lista de cadenas y en String.  
ElementaryRangeReference
- Se ha mejorado el CTID para gestionar las columnas que no existen en POSTGRE.
- Se agregó solidez para admitir la API de espacio de usuario «QUSPTRUS».
- Se agregó soporte para las API de espacios de usuario QUSRUSAT y QUSCUSAT.



- Soporte mejorado para la API de espacio de usuario (QUSPTRUS) sin código de error.
- Se agregó soporte para CRON Job Scheduling usando Quartz.
- Soporte mejorado del ciclo de programación RPG.
- Gestión mejorada de las transacciones de Blu4iv.
- Se ha mejorado el bloqueo de registros de archivos bajo control de compromiso dentro de una misma transacción.
- Se ha mejorado el manejo de la inicialización de los subarchivos.
- Se ha mejorado la visualización de los indicadores de desplazamiento de las líneas de mensajes.
- Se ha impedido que aparezcan ceros finales en los números enviados a través de la cola de datos.
- Pantalla de información de mensajes adicional mejorada.
- Se han mejorado las operaciones de escritura en JPA para tener en cuenta la biblioteca actual.
- Se ha mejorado el comportamiento ProgramJobExecutor al ejecutar programas sin parámetros.
- Se agregó la funcionalidad para pasar directamente los argumentos de los enlaces frontales a los scripts de back-end.
- Gestión mejorada de las transacciones de los metadatos de los trabajos.
- CL: se agregó soporte para el parámetro SECLVL en RTVMSG.
- CL: se agregó una implementación vacía para CLRLIB.
- CL: compatibilidad mejorada con CPYFRMIMPF para copiar tanto de bases de datos como de CSV.
- CL: se mejoró la implementación de CPYFRMIMPF para ignorar las columnas adicionales.
- CL: interpretación mejorada de los parámetros de formato antiguos por parte de CPYTOIMPF y CPYFRMIMPF.
- CL: se agregó un parámetro para formatear valores numéricos en SAVOBJ. removeDecimalPoint
- CL: comando RCVF mejorado para manejar correctamente la condición de EOF.
- CL - RTVSYSVAL - Implementación SYSVAL = QDATETIME.
- CL: comando OVRDBF modificado para obtener Field como nombre de tabla predeterminado.
- CL - RTVJOBA Valor no disponible para el parámetro: USRLIBL.
- CL - Se manejan las barras diagonales iniciales en el parámetro SNDPGMMMSG MSGF.
- CL - Se ha mejorado la compatibilidad con los caracteres comodín en el archivo fuente del comando DSPFFD.

- CL: manejo mejorado del parámetro PGMQ en RCVMSG y SNDPGMMMSG.
- CL: hizo que RTVMSG param MSG fuera opcional para alinearlos con los documentos antiguos.

## Capacidades transversales

### Nuevas características

- Se ha mejorado la capacidad al pasar un parámetro a la cláusula USING del cursor OPEN.
- Rendimiento: se ha mejorado la preinicialización del contexto y el ajuste RunUnit del rendimiento.

### Mejoras

- Se mejoró el mecanismo para descargar los valores bajos del comando UNLOAD del programa de utilidad INFUTILB.
- Se agregó la compatibilidad con la opción de esquema actual en el administrador secreto de las fuentes de datos.
- Se ha mejorado el tiempo de ejecución para no tener en cuenta los parámetros que se pasan al abrir el cursor cuando no son necesarios.
- Se ha mejorado la validación del formato numérico de los campos numéricos.
- Diagnóstico SQL mejorado en un entorno de ejecución altamente paralelo.
- Se introdujo el código unicode para la secuencia de bytes de la página de códigos (FE FD).
- DataSimplifier optimización del rendimiento: instrucciones de asignación mejoradas.
- DataSimplifier optimización del rendimiento: mejore el valor predeterminado para la inicialización de tipos numéricos para evitar un uso inútil. BigDecimal

## Herramientas de modernización, versión 4.0.0

### zOS

#### Nuevas características

- Se agregó soporte para el manejo del PROGRAMA Abend.
- Soporte mejorado para generar conjuntos de datos AIX.
- COBOL: Se agregó soporte para la cláusula JUSTIFIED en los campos ALFANUMÉRICOS, ALFABÉTICOS/GRÁFICOS.

## Mejoras

- Se mejoró el manejo de atributos de PURGETHRESH para las definiciones de recursos de TRANSCLASS.
- Soporte mejorado para la definición de datos y la declaración MOVE.
- CICS: compatibilidad mejorada con el comando DELAY en la opción MILLISECS.
- Se ha mejorado el mapeo lambda de SQL para varias tablas con alias.
- Soporte mejorado para la búsqueda de campos principales.
- Se ha mejorado el conjunto sqlstate de SQLCA para las operaciones COMMIT y ROLLBACK.
- COBOL: mejore el análisis comentando los párrafos obsoletos
- COBOL: soporte mejorado para la cláusula REPLACING.
- COBOL: se agregó soporte para funciones matemáticas ASIN ACOS LOG TAN.
- COBOL: Se agregó soporte para múltiples sentencias AFTER en PERFORM VARIING.
- COBOL: compatibilidad mejorada con los campos RENAMES (nivel 66).
- COBOL: método LENGTH OF mejorado para obtener la longitud de un índice específico en un campo matricial.
- COBOL: Se agregó soporte para múltiples cláusulas AFTER en las sentencias PERFORM VARIING.
- COBOL: compatibilidad mejorada con la cláusula RENAMES.
- COBOL: compatibilidad mejorada con la palabra clave PICTURE.
- COBOL: soporte mejorado para el análisis de campos de nivel 88.
- COBOL: se mejoró la condición de función de acceso con elementos de datos de tablas.

## AS400

### Nuevas características

- Se ha añadido la funcionalidad de pasar argumentos a las llamadas java directas del front-end.
- CL: generación mejorada de %SST, incluida la compatibilidad con \*LDA con CL→Java.
- RPG: se agregó un registro descrito por el programa de soporte para archivos DISK.

## Mejoras

- Archivo de visualización mejorado, resuelve los campos referenciados con la palabra clave «REFFLD».
- Se ha mejorado la compatibilidad con la palabra clave SETOF-CSRLOC del archivo de visualización.
- Se eliminaron los archivos del control de compromisos después del cierre.
- Se garantizó un comportamiento uniforme para las operaciones simultáneas de lectura y escritura en una tabla cuando las realizaba el mismo programa.
- Se gestionó la asignación a la subcadena de. `SizePrefixedAlphanumericType`
- Se gestionó el paso de la estructura de datos al procedimiento con un parámetro de cadena de longitud variable.
- Se mejoró la retención de valores numéricos no válidos en el evento `OnBlur` y la creación de detectores de eventos solo para los campos válidos.
- Se han mejorado los mensajes de error en las pantallas y el resaltado de los campos con entradas no válidas.
- Se ha mejorado el manejo de los campos de pantalla condicionados a los indicadores.
- Se ha habilitado el desplazamiento con la rueda del ratón.
- Se ha añadido soporte para las teclas de función de la pantalla de ayuda.
- Se ha mejorado el soporte para texto largo en el `split-dynamic-field` componente.
- Se ha mejorado el manejo de los archivos LF de varios registros al cambiar el nombre de los registros.
- CL: comando `RTVJOB` mejorado para gestionar archivos LF (vistas).
- CL: comando `OVRDBF` mejorado cuando se usa en un LF de varios registros.
- RPG: escenario gestionado en el que el procedimiento define una variable con el mismo nombre que el parámetro renombrado.
- RPG: se mejoró el manejo de `*ZEROS` al inicializar un `BinaryInteger` firmado.
- RPG: manejo mejorado de los punteros a variables no locales (de referencia).
- RPG: manejo mejorado de las sentencias `ELSEIF` después de las sentencias `iFxx`.
- RPG: se agregó soporte para los campos definidos con `LIKE` en el prototipo.
- RPG: se ha mejorado la compatibilidad con la palabra clave `LIKE` de un campo creado por `LIKEREC`.
- RPG: generación mejorada de operadores con caracteres figurativos.
- RPG: se ha mejorado el análisis de la expresión matricial `xxx (\ *)` y se admite en `%lookup`.

- RPG: código de LookUp operación mejorado con indicadores altos e iguales (o bajos e iguales).
- RPG: análisis de formato libre mejorado.
- RPG: análisis mejorado de las constantes con nombre de i-Card que siguen los formatos de registro de i-Card.
- RPG: compatibilidad mejorada con los tipos INTEGER y UNSIGNED.
- COBOL: Se agregó la cláusula INDIC compatible con el formato DSPF en la instrucción COPY DDS.
- COBOL: Se ha mejorado la gramática de las sentencias DISPLAY y ACCEPT para facilitar la transformación y la generación.
- COBOL: se agregó soporte para archivos DISK.
- COBOL: programas mejorados de soporte para archivos de visualización DDS.
- COBOL: se agregó soporte para la cláusula LIKE.
- COBOL: se agregó soporte para el archivo DISK descrito por el programa.
- COBOL: se agregó soporte para el nombre de archivo con sufijo.

## Capacidades transversales

### Nuevas características

- Manejó la carga lenta de los componentes del mapa de los proyectos web.

### Mejoras

- Se mejoró la generación en Java de los parámetros de los indicadores SQL.
- Capacidad mejorada para manejar las variables involucradas en la sentencia SET DB2.
- Se ha mejorado el aumento del error al final del cursor obtenido cuando la salida es una matriz de una sola entidad.
- Ruta gestionada en Linux.
- Data Migrator gestiona las vulnerabilidades y elimina las dependencias no utilizadas.

## Notas de la versión 3.10.0

Esta versión de las herramientas de tiempo de ejecución y modernización de AWS Blu Age se centra en las actualizaciones y mejoras básicas de todo el producto, con el objetivo de aumentar

el rendimiento y la solidez en todos los pasos de transformación y ejecución. Algunas funciones y cambios clave de esta versión son:

- Actualización de la versión de Java 8 a Java 17, lo que aumenta la seguridad y el rendimiento, y permite a los clientes implementar y ejecutar aplicaciones implementadas en un lenguaje más moderno y utilizar versiones recientes de marcos de terceros.
- Soporte adicional para administrar grandes espacios de memoria compartida entre usuarios o trabajos, almacenando datos reutilizables tras el reinicio de la aplicación o la instancia.
- Acceso más rápido a grandes conjuntos de datos en Blusam mediante un mecanismo de paginación que permite recuperar un subconjunto de registros de forma incremental.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.10.0

Este tiempo de ejecución se basa en Java17, Spring2.7 y Angular16.

### zOS

#### Nuevas características

- Blusam: Se agregó soporte para grandes conjuntos de datos a través de un mecanismo paginado en el que los índices se almacenan y cargan mediante páginas

#### Mejoras

- DataUtils.compare mejorado para gestionar la conversión de menor prioridad de cadena a número
- Se ha añadido soporte para comprobar si ByteRange se crea un no con valores incorrectos mediante la propiedad DataSimplifier.byte Check de YML RangeBounds
- RemoveSOSI () mejorado para admitir la inicialización de a con un carácter vacío GraphicAlphanumericType
- Se ha añadido solidez al funcionamiento del trabajo y se ha añadido una lectura segura del estado del GDG
- Blusam: Se agregó soporte para borrar Ecache de los conjuntos de datos de Blusam mediante un nuevo método denominado .removeCache () CoreBluesamManager

- Blusam: comportamiento mejorado de borrar/cambiar el nombre de los conjuntos de datos normales de Blusam
- Redis: soporte mejorado para desbloquear conjuntos de datos y borrar el bloqueo de registros
- JICS: se mejoró el mensaje de error para las solicitudes fallidas
- JCL: se agregó soporte para la concatenación de variables ControlM basada en un carácter de punto
- JCL: se agregó soporte para Write ADVANCING (ADV) para archivos GDG
- JCL: soporte mejorado para el número de generación actual después de eliminar todos los archivos GDG
- JCL: compatibilidad mejorada con la lectura RDW/RecordSize del catálogo al crear el conjunto de datos
- JCL: Se ha añadido el soporte para actualizar el objeto de recurso (desde AbstractSequentialFile) al abrir el archivo con el tamaño del registro de salida de datos
- JCL: rendimiento mejorado de IDCAMS
- JCL - Se ha mejorado la compatibilidad con PRINT STATEMENT al añadir «CHAR» como alias de «CHARACTER»
- SORT: soporte mejorado para la operación de copia de un conjunto de datos de longitud fija de Blusam a un conjunto de datos de longitud variable
- SORT: gramática de clasificación mejorada para gestionar algunas declaraciones específicas

## AS400

### Nuevas características

- Se agregó soporte para los espacios de usuario y sus API relacionadas
- Se agregó compatibilidad con el parámetro TOMSGQ de SNDPGMMSG y se implementaron colas de mensajes
- CL: Se agregó soporte para los parámetros FILE y SPLFNAME para el comando OVRPRTF
- CL - Se agregó soporte para manejar las bibliotecas de la tabla de particiones correspondiente con el comando CPYF
- CL: se agregó soporte para manejar el comando CHGCURLIB y tener en cuenta la biblioteca actual al crear consultas
- CL: se agregó soporte para manejar el comando cl como parte de la llamada stacktrace

## Mejoras

- Se ha mejorado MessageHandlingBuilder para gestionar mejor la entrada de seguimiento de la pila de llamadas
- Ejecución paralela mejorada de la función ContextPreconstruct
- Se han mejorado los atributos de visualización cuando SFLINZ crea un registro
- Se ha mejorado el SAVOBJ para permitir el manejo de varios archivos de salida
- Se ha mejorado el manejo de los programas geniales al añadirlos programCallStack cuando se invocan desde un programa Java
- Se ha mejorado la detección del posicionamiento superior del modal de ayuda
- Se ha mejorado la funcionalidad TopGMQ cuando se proporciona el parámetro TomSGQ para SNDPGMMSG
- Mejora de la obtención de mensajes predefinidos y de la funcionalidad del cargador de mensajes
- Se ha mejorado el manejo por parte de CPYTOIMPF de los caracteres delimitadores del contenido
- Se ha mejorado el bloqueo de publicación del registro READ

## Capacidades transversales

### Nuevas características

- Se agregó una traducción para los mensajes del sistema en el front-end
- Se agregó un nuevo método ExecutionContext para devolver la pila de llamadas del programa
- Establezca un separador de líneas (para simplificar los datos) independientemente del entorno real
- Se agregó la posibilidad de configurar la ruta JSON del modelo SQL

## Mejoras

- Se mejoró el método de comparación DataUtils .compare AlphInt () cuando se trata de relleno
- Creación de un indicador para permitir un comportamiento personalizado en caso de excepción en las consultas con el cursor
- Conversión gráfica mejorada de LOWVALUES db

## Tercero



- Actualización para mitigar los errores CVE-2024-21634, CVE-2023-34055, CVE-2023-34462, IN1-JAVA-ORGSPRINGFRAMEWORK SECURITY-5905484, CVE-2023-46120, CVE-2023-6481, CVE-2023-6378, CVE-2023-5072)

## Herramientas de modernización, versión 3.10.0

### zOS

#### Mejoras

- COBOL: soporte agregado para la función ABS
- JCL: alcance variable mejorado: se conecta a STEP en lugar de a JOB
- Inyección mejorada de parámetros de cursor para valores bajos/altos
- Análisis de CSD mejorado, especialmente para TRANSACCIONES remotas

### AS400

#### Mejoras

- Se ha eliminado la marca en blanco del indicador de nivel de control
- Se agregó soporte para nombres externos para las palabras clave de IMPORTACIÓN/EXPORTACIÓN
- Se ha añadido compatibilidad con %LEN en los campos
- CL: se agregó soporte para nuevos operadores para el lenguaje CLLE
- CL: se agregó soporte para IF anidados
- COBOL: manejo mejorado del comando START cuando se usa con varias teclas
- DSPF: manejo mejorado de la posición del cursor con un número de registro
- DSPF: se mejoró el formato de los campos numéricos firmados, los campos solo numéricos y los campos a gran escala
- DSPF: se ha mejorado la definición del título para la Ayuda general de Screen
- DSPF: compatibilidad mejorada con las especificaciones de entrada/salida
- DSPF: manejo mejorado de los separadores de agrupamiento durante la validación de campos numéricos

- Resultados de mapeo mejorados y registros DDS
- Se ha mejorado la capacidad de la palabra clave REFFLT del archivo de impresión para resolver los campos referenciados
- RPG: compatibilidad mejorada con las frases «TOTALMENTE gratuitas»
- RPG: se mejoró el análisis de condiciones y se agregó soporte para manejar CABXX sin etiqueta de resultado
- RPG: manejo mejorado de las especificaciones de entrada de los campos numéricos
- RPG: gestión mejorada de las llamadas a procedimientos en condiciones IF/ELSEIF/WHEN
- RPG: manejo mejorado del comando READ cuando se invoca en un archivo dspf
- RPG: mejora la compatibilidad con los archivos que hacen referencia a un DDS inexistente
- Mejore el manejo de REFFLD cuando se le pase un nombre en formato de registro físico
- Se agregó soporte para usar 'return' como nombre de columna de base de datos

## Capacidades transversales

### Nuevas características

- Oracle: permitió definir usuarios a diferencia de SYS para almacenar las funciones integradas

### Mejoras

- Versión Java actualizada de la v8 a la v17
- Condición SQL mejorada con el nombre de la columna del clúster
- Se ha añadido compatibilidad con las cláusulas ORDER BY desde la vista

## Notas de la versión 3.9.0

Esta versión de AWS Blu Age Runtime and Modernization Tools se centra en múltiples mejoras transversales en todo el producto, con el objetivo de aumentar el rendimiento en arquitecturas de alta disponibilidad, junto con nuevas capacidades para llevar la ejecución de los trabajos al siguiente nivel. Algunas funciones y cambios clave de esta versión son:

- Actualización de la versión de Angular 13 a Angular 16, lo que aumenta la seguridad y proporciona acceso a nuevas funciones que mejoran el rendimiento de las aplicaciones en línea de los clientes.

- Incorpora compatibilidad con características de varios trabajos en AS400, con la ventaja principal de que los trabajos pueden enviar mensajes de consulta de forma sincrónica entre ellos, lo que permite disociarlos en los trabajos modernizados.
- Mejoras en el rendimiento del uso de Redis, como la optimización del conjunto de conexiones, la alta seguridad de la conexión y la mejora del mecanismo de bloqueo de los conjuntos de datos.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.9.0

### zOS

#### Nuevas características

- Programa de clasificación: entradas VSAM actualizadas con longitud fija
- JHDB DB: incorporación de tiempo de espera configurable

#### Mejoras

- Compatibilidad mejorada con el separador de líneas para la transmisión si se usa en la concatenación de archivos
- Compatibilidad mejorada para abrir archivos secuenciales concatenados. Inicialice DataSetIndex después de abrir el archivo
- Compatibilidad mejorada con el separador decimal virtual cuando a NumericEditedType se ve afectada por un valor numérico
- Soporte mejorado para NumericEditedType valores no negativos
- IDCAMS: Las tarjetas SYSIN ahora se leen con la propiedad de «codificación» definida en .yml application-utility-pgm
- IDCAMS: se ha actualizado la gramática para admitir el argumento FILE(..) en la instrucción DEFINE CLUSTER
- INFUTILB: se ha agregado compatibilidad con el argumento DFSIGDCB para anular los parámetros DCB de DD SYSREC
- INFUTIL: se ha mejorado la compatibilidad para el parámetro “DFSIGDCB YES”
- Se ha mejorado SPLICE para gestionar archivos de entrada de gran tamaño

- DFSORT: se ha mejorado la gestión de los campos de comentarios
- DFSORT: se ha agregado compatibilidad con el formato numérico de formato libre (firmado o no firmado) (SFF/UFF)
- SORT: se ha agregado compatibilidad con el análisis de las sentencias OPTION PRINT y OPTION ROUTE
- SORT/ICEMAN: se ha agregado compatibilidad para operaciones de división incluidas (campo con operador DIV)
- Se ha mejorado la compatibilidad para CICS READ mediante una clave genérica
- Se corrigió la función StringUtils .chgraphic para eliminar el SOSI de un tipo de gráfico
- Mejore el rendimiento en la codificación .is DataUtils DoubleByte
- JCL: se ha mejorado la compatibilidad con el modo de disposición KEEP para un conjunto de datos temporal. El sistema cambia la disposición a PASS
- JCL: gestiona los parámetros DCB de forma dinámica
- JCL: salidas de SUM FIELDS mejoradas para valores incorrectos
- JCL: CommonDDUtils::getContent ahora busca el recordSize en el catálogo
- JCL: lee los atributos rdw/recordSize del catálogo al crear el conjunto de datos
- JCL: se ha agregado compatibilidad con DCB=.MYDD para copiar los parámetros DCB de un DD en otro en el mismo paso de trabajo
- JCL: mejora en el sistema de herencia de tamaño de registros
- JCL: Se agregó un bloqueo de conjunto de datos exclusivo (Redis)
- Redis: se ha agregado compatibilidad SSL para el modo independiente
- Redis: se ha agregado el recuento sincronizado de Redis con bloqueo
- Redis: parámetros de grupo compatibles con el bloqueo de Redis
- Redis: actualización de metadatos optimizada con Redis
- Redis: compatibilidad mejorada con clústeres de Redis
- Mejora en los bloqueos abiertos con modo IO
- Mejora en el rendimiento de los bloqueos de conjuntos de datos y el borrado de bloqueos no utilizados
- Mejora en la ruta del conjunto de datos al anular el registro del archivo
- Mejora en la invalidación de la caché de la ventana de recuperación previa
- Se ha agregado compatibilidad para el uso seguro de subprocessos por proveedores de orígenes de datos de utilidades

- Mejora en la comprobación de nulidad de datasetState
- Mejora en la compatibilidad de la no reapertura de conjuntos de datos ya abiertos
- Mayor robustez para la operación final del trabajo
- Mejora en la compatibilidad para los índices y el orden de las claves, lo que permite duplicarlos
- Mejora en la compatibilidad para el orden de serialización de las listas de omisión
- Se ha agregado compatibilidad para la característica de descarga de depuración para ayudar a diagnosticar problemas con el orden de los índices
- Mejora en la compatibilidad con la actualización de metadatos
- Soporte mejorado para la lectura masiva de Blusam

## AS400

### Nuevas características

- Crea un nuevo registro application-context
- Compatibilidad para palabras clave DSPF CLRL(NO) Compatibilidad con la supervisión de bloqueos de registros
- Support for keyed DataQueue
- Compatibilidad con mensajes INQUIRY para trabajos por lotes
- Se ha agregado compatibilidad con el archivo de impresora descrito por el programa para el AS400 COBOL
- Gestión del comando RMVJOBSCDE cl
- Mejora para RUNSQL/DLYJOB
- CHKOBJ: generación de un código de error heredado para el parámetro LIB
- SNDPGMMMSG: compatibilidad con parámetros de cadena
- RTVDTAARA: subcadena mejorada en LDA
- DSPFD: se ha agregado un parámetro FILE compatible para el nombre de archivo específico
- RUNQRY: compatibilidad para archivos sql en QRY PARAM
- CRTDUPOB: compatibilidad para copiar los datos entre áreas de datos
- SBMJOB: Convierte la instrucción en uso JobQueueManager
- OPNQRYF: se ha agregado compatibilidad para la biblioteca Qtemp
- CRTDUPOBJ: lógica mejorada para copiar el contenido de la partición

- CRTDUPOBJ: se ha agregado compatibilidad para Qtemp para las vistas
- RTVSYSVAL: compatibilidad para el valor SYSVAL, QDATFMT en el comando CL
- CHKOBJ: se ha agregado compatibilidad para OUTQ
- RTVJOBA: compatibilidad con el parámetro SWS
- SNDPGMMMSG y RCVMSG: compatibilidad de parámetros adicionales como MSGF, MSGFLIB, MSGDTA, MSGTYPE, KEYVAR, MSGKEY, MSGID

## Mejoras

- Mejora de la compatibilidad con tarjetas WORKSTATION I/O
- Mejora de la gestión del mensaje configurado superpuesto al mensaje anterior
- Compatibilidad con información de mensajes adicional en array-messageline
- Mejora del acceso al contenedor de matrices independiente dentro de EVAL, SortA y figuratives
- Mejora de la limpieza de los DAO cuando finaliza la aplicación en línea
- Se ha agregado compatibilidad para formatos de fecha adicionales y se ha mejorado la gestión de las entradas de cadenas
- Se mejoró el manejo CVTDAT de SYSVAL al agregar valores del sistema, la clase auxiliar Decode y compilar parámetros a partir del comando CL SbmJob
- Se ha eliminado el paquete com.netfective.bluage.gapwalk.rt.blu4iv del escaneo de componentes gapwalk-cl-command
- Se ha mejorado la compatibilidad de mensajes predefinidos para la API de cola de mensajes
- Se ha mejorado el soporte para registros escritos en otro programa retrieveSubfileRecord
- Se ha mejorado la compatibilidad de mensajes inmediatos para la API de cola de mensajes
- Se ha mejorado la gestión del área de datos local al enviar un trabajo
- Se inicia JobQueues automáticamente cuando se inicia el servidor
- Utiliza la configuración de applicationContext para descodificar los parámetros de SBMJOB
- Mejora de los mensajes de error proporcionados por el sistema
- Permite a RTVMSG buscar archivos .properties en subdirectorios anidados
- Gestiona el restablecimiento de entidades vinculadas a punteros incorrectos o no válidos
- Se ha mejorado MessageHandlingBuilder la visualización del MsgID y el MsgFile nombre como cadenas para RCVMSG
- Método de withMsgFile nombre mejorado de la API de cola de mensajes

- Mejora en el mecanismo de bloqueo del área de datos
- RTVMBRD: compatibilidad para minúsculas y mayúsculas para el parámetro FILE
- CRTDUPOBJ: mejora en la gestión de las vistas
- CPYTOSTMF: mejora en la gestión de la conexión
- CPYF: mejora en la gestión del nombre del directorio al copiarlo desde un archivo plano
- RCVF: gestión correcta de los parámetros DEV/RCDFMT y la transformación de RCDFMT para groovy y java
- RCVF: gestiona las llamadas posteriores y evita restablecer el cursor
- CPYF: se ha agregado compatibilidad para escribir desde archivos planos
- CRTDUPOBJ: se ha agregado la gestión del nuevo obj con la biblioteca Qtemp
- CHGDTAARA: se ha aumentado la longitud máxima del área de datos de 256 a 2000
- SAVOBJ: se asegura de que los registros guardados estén en orden de inserción
- RTVDTAARA: valores recuperados (no deben recortarse)
- CHKOBJ: devuelve los mensajes de supervisión correctos cuando el miembro no existe
- RTVDTAARA: se ha agregado compatibilidad para la subcadena LDA
- RTVDTAARA: devuelve espacios en blanco de hasta la longitud de la variable especificada en el parámetro RTNVAR
- RTVDTAARA: admite parámetros enteros para el inicio y la longitud y admite el formato de transformación más reciente
- CHGDTAARA: se ha agregado compatibilidad para parámetros que incluyen límites inferior y superior
- CHKOBJ: gestiona el valor VIEW del tipo de objeto de parámetro
- CHKOBJ: el resultado se establece en true independientemente del miembro si la vista existe

## Capacidades transversales

### Nuevas características

- Gestiona la generación de informes en archivos .txt
- Se ha agregado la propiedad de origen de datos currentSchema XA al administrador de secretos
- Añada la propiedad YAML database.cursor.raise.already.opened.error para permitir que el marco genere el error 502 de SQLCODE cuando se abra el cursor ya abierto

## Mejoras

- Se agregaron pompones Gapwalk a los empaques de AWS Blu Age en Amazon EC2.
- Utiliza el nuevo paradigma de manejo de señales de forma predeterminada
- Se ha agregado compatibilidad para el bloqueo cuando la disposición sea MOD u OLD
- Se ha agregado una caché para almacenar los patrones de fecha y hora de la base de datos
- Función de comprobación mejorada de PackedType
- Mejore las funciones DataUtils de .setTo para registros con VariableSizeArray
- Gestiona la opción MQ SYNCPOINT como si fuese una unidad de ejecución
- Marco habilitado para configurar SQLCODE en una transacción de reversión
- Se ha agregado el nombre de clase de controlador automático según el secreto de la clave del motor
- Tiempo de espera del programa/transacción
- Se restaura la posición del cursor después de la reversión al acceder al cursor

## Tercero

- Actualice SnakeyAML, Redisson y Amazon SDK, elimine YamIBeans (mitigue CVE-2023-24621, CVE-2023-42809, CVE-2023-44487)

## Herramientas de modernización, versión 3.9.0

### zOS

#### Mejoras

- Compatibilidad mejorada para XML-TEXT como origen para el destino de tipo String
- Se ha mejorado el flujo de trabajo de STM a UML para admitir el patrón de división X/(Y/Z)
- JHDB DB: acepta la llamada ROLLBACK antes de cualquier actualización de la base de datos
- JHDB DB: acepta el ROLLBACK incluso si la transacción está terminada (NOP)
- JCL: función de validación por pasos mejorada
- SORT: gestiona la función SUM con valores negativos decimales de zona
- COBOL: agrega soporte para el escape de comillas simples o dobles en literales de cadena



## AS400

### Mejoras

- Se ha mejorado la función integrada %editc que gestiona el código de edición X añadiendo ceros a la izquierda
- Se ha mejorado la gestión del valor inicial de los campos solo de entrada
- Se han agregado teclas de acción para facilitar los cuadros de diálogo
- Registro de pie de página de la tabla dinámica que aparece en la parte inferior
- Se gestiona el comando START sin KEY PHASE para los archivos que especifiquen una RECORD-KEY real
- Se agregó un valor predeterminado para los tipos float y NumberUtils: :pow
- Se ha agregado compatibilidad para definir una variable usando LIKE(IN)
- Se ha actualizado la gestión de bucles FOR para permitir la omisión de elementos opcionales
- Se ha actualizado el análisis de RPG para asociar los registros con el nombre de la matriz CTDATA
- Se ha mejorado la gestión de los indicadores para las instrucciones CABxx
- Compatibilidad del parámetro opcional de la palabra clave COMMIT
- Compatibilidad mejorada con la palabra clave FORMAT en LF
- Código de operación LOOKUP administrado con indicadores altos e iguales (o bajos e iguales)
- Se gestiona el nombre de clave PF declarado entre comillas dobles
- Se ha mejorado la gestión de EDTCDE X para no suprimir los ceros a la izquierda
- Se ha mejorado la compatibilidad con MSGCON en archivos de impresora que no generan etiquetas sin nombre
- El CONTENT del campo lo comparten varias estructuras de datos
- Se gestiona el parámetro ERRSFL en combinación con SFLMSG/SFLMSGID
- Se ha mejorado el código principal antes del ámbito de declaración de proceso del rpg libre completo
- Se ha agregado la especificación de control condicionado al análisis
- Se ha mejorado la compatibilidad con el método setErrSfl () en dataholdermapper
- Se ha mejorado la resolución de tipos para las variables creadas internamente

- Se ha mejorado la compatibilidad para el código de operación Z-ADD
- Se ha mejorado la gestión del campo constante con un valor DFT
- Se ha mejorado la compatibilidad con campos enteros dentro del estado del programa ds
- Se gestiona la asignación de indicadores en los parámetros ENTRY
- Se ha mejorado el filtro de palabras clave propagadas a través de la palabra clave ref/reffield
- Estructura de datos sin nombre compatible DataArea
- Se ha mejorado la gestión del tipo de datos del puntero
- Los elementos gestionados de la matriz utilizados para definir variables con compatibilidad con la palabra clave LIKE admiten el acceso a la matriz en el campo de salida
- Se ha mejorado la compatibilidad para campos numéricos firmados, que solo muestran dígitos
- Compatibilidad con la relación lógica en la tarjeta O
- Caso de prueba para %CHAR en formato alfanumérico
- Compatibilidad con la entidad principal de la palabra clave de especificación de control
- EDTCDE con dos parámetros en el archivo de impresión
- Análisis de FullFree RPG mejorado
- Se ha mejorado la tabla dinámica para garantizar que el pie de página esté colocado correctamente
- Se ha agregado compatibilidad para inicializar tipos numéricos con TODAS las constantes figurativas
- Se ha mejorado la gestión de varios archivos lógicos RPG que hacen referencia al mismo archivo físico
- Se ha mejorado la detección de campos modificados en una pantalla moderna
- Sincronización modal con campos dinámicos
- Se ha mejorado la gestión del campo numérico firmado de solo salida
- Mejora de la compatibilidad con tarjetas WORKSTATION I/O

## Capacidades transversales

### Nuevas características

- Herramienta de migración de datos: se agregó la propiedad ebcDicFilesWith VarcharIn VB para permitir tener en cuenta la longitud de 2 bytes de VARCHAR al leer los bytes

- Se ha implementado una API común para registrar los errores
- Implementación BluAgeErrorDictionaryUtils y uso de una API común para registrar errores y/o información en COBOL2Model, RPG, Definitions2Model y CycleBuilder FieldsProcessor
- Se ha mejorado la gramática SQL para admitir diferentes definiciones de cláusulas de aislamiento

## Mejoras

- Se ha actualizado la versión de Angular a la v16
- Angular: se ha mejorado la versión ajv de la 6 a la 8.9

## Tercero

- Se ha actualizado Groovy a la versión 2.4.15

## Notas de la versión 3.8.0

Esta versión de AWS Blu Age Runtime and Modernization Tools se centra en múltiples mejoras transversales en todo el producto para mejorar su calidad y seguridad, además de mejorar el rendimiento del almacenamiento en caché y la unificación de los soportes de comandos en una sola distribución. Algunas funciones y cambios clave de esta versión son:

- Actualización de la versión de Spring 2.5 a Spring 2.7, lo que aumenta el soporte de mantenimiento, el rendimiento y la seguridad de la plataforma.
- La unificación de más de 82 comandos CL es compatible con la over-the-counter distribución para facilitar el uso y el despliegue de aplicaciones modernizadas que anteriormente utilizaban scripts CL.
- Hay nuevas API disponibles para operar e interactuar mejor con los conjuntos de datos de BluSAM, como la importación integrada al servicio gestionado y la capacidad de listar la información de los metadatos de los conjuntos de datos.
- Mejoras en el rendimiento y ampliación del uso de Redis, incluida la disponibilidad en modo clúster, la recuperación de datos de alta disponibilidad y la estandarización del uso de datos secretos.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.8.0

### zOS

#### Nuevas características

- Manejar la definición clave como una cadena para DynamicFileBuilder
- DFSORT: se ha agregado compatibilidad con varios elementos en la inicialización gramatical de OUTFIL TRAILER1 + DFSORT
- Herramienta CommonDDUtils: gestiona el tamaño de los registros en los datos integrados
- Archivo indexado: gestión de la opción GENKEY

#### Mejoras

- Servicios de carga de BluSAM externalizados en un contenedor separado
- Se ha agregado compatibilidad para configurar la ubicación a fin de almacenar archivos temporales
- Mecanismos de caché compartida mejorados para casos con varios nodos
- Uso de caché compartida: los IDCAMS verifican la optimización
- Mejora de la inyección de ROWID para la selección integrada
- JCL: cada procedimiento de trabajo continuo se genera ahora en un archivo Groovy distinto
- Garantice una cobertura de card-demo-v 2 en las tarjetas IDCAMS JCL
- BluSAM: evite duplicar el calentamiento cuando utilice varias instancias
- Menor consumo de memoria al hidratar la caché
- Compatibilidad con la configuración de Jedis Pool
- Se ha agregado un separador de líneas a la transmisión si se usa en la concatenación de archivos
- Compatibilidad con tarjetas EBCDIC + bloquear comentarios (/.../) en la utilidad IDCAMS
- Consulta de compatibilidad con bases de datos: compatibilidad con cadenas de doble byte en la conversión del nivel 49 a SQL
- Gramática DFSORT: implementa 17 instrucciones de control más la integración de 2 de ellas (OMIT/INCLUDE)
- Mejora de las columnas gráficas, búsqueda de INFUTILB

- Compatibilidad para leer archivos con tabla de tamaños variables
- Support para ZonedType con signo nibble donde el primer bit del último byte es 'E'
- DFSORT/ICETOOL añade compatibilidad con el argumento NOMATCH =(..) si un registro no coincide con ninguna de las constantes de búsqueda de CHANGE
- Compatibilidad con clústeres de Redis
- Gestión del estado del trabajo (fallido) según el código de salida groovy
- Compatibilidad mejorada con CICS SYNCPOINT ROLLBACK.
- Ventana de recuperación previa para optimizar el uso de la caché de Redis
- JCL/GROOVY: hereda la propiedad isRDW del conjunto de datos del paso anterior cuando DISP =(, PASS)
- Gestión de copias parciales de datos con una matriz de tamaño variable

## AS400

### Nuevas características

- Compatibilidad con tarjetas de E/S para archivos de visualización
- Soporte para información de mensajes adicional para las palabras clave DSPF ERRMSGID y CHKMSGID
- Compatibilidad con múltiples mensajes de error en la pantalla de la interfaz
- Se agregó o mejoró la compatibilidad con 82 comandos CL dentro de la aplicación gapwalk-cl-command

### Mejoras

- Compatibilidad mejorada para DELETE y READ bajo control de compromiso
- ConvertDate dentro del %dec incorporado
- Encabezados de seguridad XSS forzados
- Mejora de la solidez y la coherencia de la generación de STM (mejor gestión de: línea de continuación en formato rpg libre, comas para la parte decimal, bloques de formato libre en la definición o declaración)
- Generación mejorada DataHolderMapper
- Mayor robustez y margen de cambio en DataAreaFactory

- Se ha mejorado el cambio de enfoque en la tecla de tabulación
- Rendimiento mejorado en la generación de informes de Jasper
- Visualización de decimales mejorada con un relleno de 0
- Compatibilidad mejorada para el campo ROW/COL en INFDS
- Mejora de la compatibilidad con los campos modificados desde la pantalla
- Se han agregado captadores para el nombre y la ruta del informe generado
- Se ha mejorado la longitud de Dataqueue
- Configuración automática mejorada de Job Queues para adaptarla a los nuevos estándares de Spring Boot 2.7
- Actualizaciones de estaciones de trabajo mejoradas para múltiples sesiones simultáneas

## Capacidades transversales

### Nuevas características

- Compatibilidad con tolerancia a datos no válidos para empaquetados
- Se ha agregado paginación o filtrado para enumerar los puntos de conexión del conjunto de datos

### Mejoras

- Estrategia mejorada de transformación de consultas de ORACLE en la comparación de columnas con cadenas vacías
- Gestión de BLOB DB2 con los programas de utilidades DSNTEP e INFUTILB. Los BLOB DB2 ahora están modernizados a postgres de tipo BYTEA.
- Mejora de la eliminación del último elemento del cursor
- Compatibilidad con la eliminación de archivos RRDS mejorada
- Se mejoró el rendimiento secreto de AWS Blusam
- Gestión mejorada de las conexiones de bases de datos en el marco SQL
- Claves de administrador secreto estandarizadas AWS de múltiples fuentes de datos
- Correcciones de regresión del rendimiento
- Función de comprobación mejorada para PackedType
- Manejo mejorado de LOW-VALUE para PackedType

- Paquete de seguridad Spring mejorado para la conexión de Cognito
- No se aplica la codificación y decodificación por puntos de cambio de código en las bases de datos de destino DB2

### Tercero

- Actualización de Spring Boot de 2.5 a 2.7

## Herramientas de modernización, versión 3.8.0

### zOS

#### Nuevas características

- JCL: gestión del flujo con retorno de carro “\ r”

#### Mejoras

- Se ha mejorado el registro para evitar la división por cero al modernizar una cláusula DIVIDE con ON SIZE ERROR
- JCL: compatibilidad mejorada para llamar a un procedimiento dentro de un procedimiento
- Compatibilidad con la palabra clave OF en el comando FORMATTIME CICS cuando hay campos ambiguos
- JCL: compatibilidad con el carácter Â¥ en variables
- JCL: cálculo de RC en función de los pasos anteriores
- Comparación de bytes en lugar de cadenas cuando se utiliza PL1 SUBSTR
- Mejora de la inicialización de matrices multidimensionales desde un solo origen
- Se ha mejorado el análisis de COBOL cuando se trata de una única consulta SQL en un bloque IF

### AS400

#### Nuevas características

- Compatibilidad con la instrucción IF anidada en CL
- Compatibilidad mejorada para la instrucción ENDDO en formato libre de RPG

## Mejoras

- Compatibilidad mejorada para el acondicionamiento del nivel de control
- El prototipo mejorado regresa con LIKE
- Compatibilidad mejorada para gestionar las funciones %months, %year, %days
- Compatibilidad con la función de ayuda para toda la pantalla
- Gestión de los BLANKS figurativos pasados como parámetro
- Mejora de la expresión EVAL con el operador ""
- Gestión del comando START sin KEY PHASE
- Mejora en la gestión de la palabra clave LIKEREK
- Mejora en los subcampos sin nombre
- Mejora en el procedimiento de devolución de un tipo sin signo
- Se ha mejorado la compatibilidad con el funcionamiento de RESET (RPG gratuito) y las funciones integradas %CHAR y %DEC.
- Mejora de la función integrada %LOOKUPXX
- Compatibilidad mejorada para la palabra clave LIKEDS en un procedimiento sin prototipo
- Gestión del tipo de matriz de palabras clave Dim (VAR, AUTO)
- Compatibilidad mejorada para XFOOT
- COBOL: compatibilidad mejorada para los campos RENAME
- CL: admite la condición while(true)
- Se ha mejorado la gestión de matrices independientes con la palabra clave LIKE
- Mejora de la función integrada %INT
- Análisis RPG Full Free mejorado
- Compatibilidad mejorada para la matriz en el enlace
- CL2GROOVY: compatibilidad con la instrucción Select
- Mejora en la palabra clave del DSPF "ERRMSGID"
- Se ha mejorado la gestión de la inicialización de bytes con ceros a la izquierda
- Mejora de authorizedValues para los campos numéricos
- Gestión del extensor H para una instrucción EVAL de formato libre
- CL para Groovy: compatibilidad con la subcadena de LDA



- Compatibilidad mejorada para RESET en un registro
- Se ha mejorado la gestión de EDTCDE y EDTWRD con referencias
- Se ha mejorado el mapeo de campos de entrada con campos DDS
- Se ha mejorado la compatibilidad para caracteres MOVEA en la matriz IN
- Mejora en el prototipo con la palabra clave LIKEDS
- Se ha mejorado la compatibilidad con la palabra clave DSPATR del DSPF
- Análisis mejorado de la tarjeta D con +/-
- Mayor robustez en las llamadas de los programas
- Mayor solidez en el proceso de resolución de campo

## Capacidades transversales

### Mejoras

- FrontEnd: Simula un evento de pegado para una entrada de IME

### Tercero

- Actualización de Spring Boot de 2.5 a 2.7

## Notas de la versión 3.7.0

Esta versión de las herramientas de modernización y tiempo de ejecución de AWS Blu Age incluye principalmente mejoras para una mejor compatibilidad con los comandos y las utilidades, capacidades para integrarse con AWS Secrets Manager y nuevas funciones de supervisión. Algunos de los cambios clave de esta versión son:

- Varios componentes de tiempo de ejecución ahora pueden usar AWS Secrets Manager para aumentar la configuración de seguridad de las aplicaciones modernizadas, principalmente relacionadas con las fuentes de datos de las utilidades, las colas de Redis para TS, la BluSam memoria caché y los bloqueos.
- Punto de conexión que permite recuperar métricas de transacciones, lotes y JVM para optimizar el uso de los recursos y gestionar las operaciones, como el estado, la duración, el volumen y otras.
- Nuevas funciones para admitir las llamadas de IBM MQ en RPG y una mayor cobertura de transformación de JCL SORT e IDCAMS.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.7.0

### Temas

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

## zOS

### Nuevas características

- Mejora del análisis de las consultas incluidas en la aplicación de utilidades del programa mediante el uso de una gramática similar a la de SQL. (V7-9401)
- Gestión de una matriz de tamaño variable indexada cuando está desplazada (V7-9904)
- Compatibilidad de INSERT SQL TIME en DB2 con formato de 24:00:00 horas (V7-10023)
- Compatibilidad con consultas INSERT SQL desde matrices con las opciones FOR ROWS y ATOMIC (V7-10105)
- JCL SORT: mejorada TranscodeTool para que sea compatible con OUTREC con IFTHEN (V7-10124)
- JCL SORT: agregue compatibilidad con la palabra clave DATE en el comando OUTREC (V7-10125)
- JCL: añade compatibilidad con los procedimientos In-Stream (V7-10223)

### Mejoras

- Debería haber un conjunto de datos marcado con la disposición "PASS" disponible en todas las etapas del trabajo (V7-9504)
- Compatibilidad del atributo SCHENV de JCL (V7-9570)
- Compatibilidad de SEND con la opción CTLCHAR (V7-9714)
- COBOL: gestione diferentes conjuntos de caracteres separadores de líneas en las instrucciones ACCEPT (V7-9875)

- Evite la reversión múltiple (V7-9958)
- Permita el uso de la disposición MOD para anexar al final de los archivos GDG (V7-10031)
- Optimización: refactorización de putAll (V7-10063)
- PutAll refactorización: añadir paginación (V7-10063)
- Haga que el tiempo de espera de lectura del cliente Jedis sea configurable (V7-10063)
- UseSsl soporte para modo independiente (V7-10114)
- Compatibilidad con EIBDS después de abrir el archivo correctamente (V7-10147)
- Compatibilidad con EIBDS tras una solicitud de control de archivos (V7-10147)
- Mejore la compatibilidad con CICS SYNCPOINT (V7-10187)
- BluesamRedisSerializador: problema con MetadataPersistence (V7-10202)
- Compatibilidad con AWS Secrets Manager de Redis para colas TS (V7-10204)
- Compatibilidad con JCLBCICS en la personalización del tamaño del nombre DD (V7-10224)
- Añade compatibilidad con la ruta absoluta en la instrucción DELETE de IDCAMS (V7-10308)

## AS400

### Nuevas características

- Implementación de la característica de ayuda para las pantallas AS400 (V7-9673)

### Mejoras

- Número de registros en el INFDS (V7-9377)

## Capacidades transversales

### Nuevas características

- Support for Runtime on EC2 para enviar registros a Amazon CloudWatch (D87990246)
- Se ha agregado un nuevo punto de conexión para recuperar métricas sobre lotes, transacciones y JVM (D88393832)

### Mejoras

- Compatibilidad con AWS Secrets Manager de orígenes de datos para la utilidad pgm (V7-9570)
- Se ha agregado compatibilidad con Db2 para DSNUTILB DISCARD (V7-9798)
- Compatibilidad para escribir en el registrador en lugar del flujo de salida predeterminado del sistema en los archivos SYSPRINT y SYSPUNCH predeterminados (V7-10098)
- Support BluSam Redis almacena en caché y bloquea las propiedades de conexión en AWS Secrets Manager (V7-10238)
- Compatibilidad para conexión SSL en secreto de AWS de Db2 XA (V7-10258)
- Metadatos actualizados para IDCAMS REPRO y VERIFY (V7-10281)
- Administración mejorada de códigos de devolución de IDCAMS Abend (V7-10307)

## Herramientas de modernización, versión 3.7.0

### Temas

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

### zOS

#### Nuevas características

- PLI: asignación mejorada para matrices bidimensionales y de sección transversal (V7-9830)

### AS400

#### Nuevas características

- Gestión de los indicadores de nivel de control (V7-9227)
- Compatibilidad con el parámetro EXTNAME con valor \*INPUT (V7-9897)
- Reescritura mejorada de Goto: compatibilidad con etiquetas ubicadas en instrucciones SELECT OTHER (V7-9973)
- Compatibilidad con la palabra clave REFSHIT en el DSPF (V7-10049)

### Mejoras

- Mejora en la gestión de la palabra clave de descripción del archivo EXTIND (\*INUx) (V7-7404)
- Transformación mejorada de archivos SQLDDS (V7-7687)
- Los objetos de archivo ya no se generan para los archivos AS400 (V7-9062)
- Se ha mejorado la gestión de la palabra clave de descripción del archivo EXTDESC (V7-9268)
- Se ha mejorado la gestión de la versión integrada %CHAR (V7-9311)
- Se ha mejorado la compatibilidad con la reducción de páginas en el último registro sin SFLEND (V7-9322)
- Compatibilidad mejorada con estructuras de datos con prefijos (V7-9436)
- Compatibilidad con dimensiones definidas con %SIZE (V7-9472)
- Compatibilidad para gestionar el nombre de campo PF declarado entre comillas dobles (V7-9557)
- Funcionamiento de archivos mejorado: no se distingue entre mayúsculas y minúsculas (V7-9785)
- Compatibilidad con el campo inicializado como \*USER (V7-9806)
- Compatibilidad con el tipo COMP en AS400 (V7-9840)
- Análisis mejorado de COBOL400 en (Not) (V7-9922) InvalidKey
- Gestión mejorada de la operación SCAN (V7-9971)
- Compatibilidad mejorada con el código de operación GOTO (V7-9973)
- Gestión mejorada de la operación EXCEPT (V7-9977)
- Compatibilidad con prefijos mejorada (V7-10000)
- Compatibilidad con llamadas MQ en RPG (V7-10007)
- %LOOKUP integrado mejorado (estructura de datos de matriz con claves) (V7-10022)
- Compatibilidad con Close \*All operation (V7-10036)
- Compatibilidad con la instrucción UPDATE AS ROW CHANGE SQLDDS (V7-10051)
- Mejora para gestionar el tipo de valor literal Long (V7-10073)
- Mejora en la gramática de RPG (uso de la palabra clave INZ como nombre de la subrutina) (V7-10074)
- Se ha mejorado la gramática en RPG para admitir valores numéricos con una parte fraccionaria vacía (V7-10077)
- Compatibilidad mejorada para los campos compartidos entre CL y un archivo externo (V7-10081)
- Compatibilidad mejorada para los indicadores condicionales DDS (V7-10084)

- Compatibilidad con el tipo binario DDS con programas COBOL (V7-10100)
- Se ha mejorado la colisión de nombres con el enlace (V7-10109)
- Compatibilidad para mezclar procedimientos principales y de exportación (V7-10112)
- Soporte mejorado para DataStructure un subprocedimiento (V7-10113)
- Compatibilidad mejorada de CLEAR (V7-10126)
- Compatibilidad mejorada del bucle DO (V7-10134)
- Compatibilidad con SQLTYPE en RPG completamente gratuito (V7-10151)
- Se ha mejorado el análisis de las condiciones de la palabra clave DDS (V7-10155)
- Generación de DSL mejorada (V7-10163)
- Mejora de processIndicators cuando la condición es una expresión binaria. (V7-10164)
- GOTOs con la condición Else mejorada (V7-10168)
- Compatibilidad con los tipos Time y Timestamp en DSPF (V7-10173)
- Análisis mejorado de la línea de continuación para el DDS (V7-10183)
- Compatibilidad con COBOL para RENAMES FLD OF RECORD (V7-10195)
- Se ha mejorado el análisis de indicadores condicionales en los campos DSPF (V7-10221)
- Compatibilidad para el análisis de la palabra clave DDS NOALTSEQ (V7-10288)
- Compatibilidad con el menú de ayuda y campos ocultos (V7-10314)
- Comprobación mejorada del estado de las palabras clave de ayuda del DSPF (V7-10328)
- Ya no se propagan todas las palabras clave en el campo Ref (V7-10347)

## Capacidades transversales

### Nuevas características

- Migrador de datos: gestión de datos CLOB (V7-9665)

### Mejoras

- Propagación de la propiedad SCHENV de JCL desde la definición de JOB hasta la de PROC GROOVY (V7-10225) JobContext
- FrontEnd - Ajustar el tamaño de la ventana en caso de que no haya bordes (V7-10358)

## Notas de la versión 3.6.0

Esta versión de las herramientas de tiempo de ejecución y modernización de AWS Blu Age ofrece nuevas funciones para las migraciones antiguas de zOS y AS400, orientadas principalmente a ampliar los mecanismos de soporte del CICS, complementar las capacidades de JCL, optimizar el rendimiento de las funciones simultáneas y de gran volumen y añadir capacidades. multi-data-source Algunos de los cambios clave de esta versión son:

- Mejora de la gestión dinámica de archivos de JCL, expansión de las instrucciones actuales y administración de conjuntos de datos concatenados, ejecución de múltiples instrucciones en un solo bloque y transferencia de datos de lotes a programas.
- Compatibilidad mejorada con varios comandos del CICS, incluida la consulta de varios tipos de recursos del CICS.
- La capacidad de tener diferentes bases de datos cuando se utilizan las utilidades de Blu Age Runtime, que resulta ideal para situaciones en las que los datos empresariales se distribuyen en varios orígenes.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.6.0

### Temas

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

## zOS

### Nuevas características

- JCL - DynamicFileBuilder - Gestión mejorada de los gestores de archivos (V7-9408)
- Conversión de formato mejorada en algunas funciones integradas de SQL DB2 al llamar a la utilidad INFUTILB UNLOAD (V7-9554)
- Asignaciones de matrices multidimensionales PLI mejoradas (V7-9592)

- Gestión de la redirección de sysout al archivo (V7-9992)

## Mejoras

- Añada la activación de los procedimientos almacenados para el RDBMS de DB2 (V7-9155)
- SORT gestiona la conversión a formato PDF (V7-9286)
- JCL/GROOVY: mejore la instrucción REPRO para que sea compatible con conjuntos de datos DUMMY (V7-9424)
- Mejora de la compatibilidad con CICS UNLOCK (V7-9606)
- Gestione el tamaño del valor predeterminado para Union (V7-9648)
- JCL/GROOVY gestionan diferentes terminaciones/disposiciones en conjuntos de datos concatenados (V7-9653)
- Haga que PageSize sea configurable para los conjuntos de datos de Blusam (V7-9680)
- DSNUTIL: permita cargar las 24:00:00 como valor válido para TIME en DB2LUW (V7-9697)
- Support la comparación de VALORES ALTOS (0xff) en NumberUtils .ne (/) NumberUtils.eq () (V7-9731)
- JCL/GROOVY: compatible con DO... Palabras clave THEN en las cláusulas IF-THEN-ELSE de IDCAMS para ejecutar varias instrucciones en un solo bloque (V7-9750)
- Programa llamado JHDB no válido fuera de JHDB (V7-9782) BatchRunner
- Compatibilidad para espacios en blanco en la tarjeta de control SORT OUTFIL (V7-9808)
- Mejora de la compatibilidad con CICS READ PREV (V7-9845)
- Mejora del acceso simultáneo a los índices de los conjuntos de datos (V7-9864)
- Mejora de la compatibilidad con CICS REWRITE (V7-9873)
- COBOL: las sentencias ACCEPT admiten SYSIN de líneas múltiples para pasar datos de un lote (JCL) a un programa (COBOL) (V7-9875)
- Groovy: mejor manejo de los archivos en la etapa de creación (V7-9876)  
ConcatenatedFileConfiguration
- IDCAMS UTILITY: gestión de la instrucción DEFINE PATH (V7-9878)
- SORT BUILD: ajuste la opción TRAN y gestione los espacios en blanco implícitos (V7-9925)
- Mejora de CICS DELETE con la compatibilidad con la opción GENERIC (V7-9939)
- Mejora de la compatibilidad con CICS STARTBR y ENDBR (V7-9952)



- Mejora del rendimiento cercano en el acceso simultáneo (V7-9953)
- Mejora de la gestión del estado de los archivos al inicio (V7-9991)
- Groovy: permite llamar a `getDisposition () /get () /get NormalTermination ()` el `()` (V7-10012)  
`AbnormalTermination ConcatenatedFileConfiguration`

## AS400

### Nuevas características

- Compatibilidad con indicadores externos en palabras clave COMMIT (V7-6035)
- Restablezca el bucle ReadC después de escribir en SFLCTL (V7-8061)
- Compatibilidad con el indicador LR en CALL (V7-9250)
- Añada un nuevo tipo de campo dinámico (dividido) para gestionar el campo de entrada en varias líneas (V7-9370)
- Compatibilidad con archivos primarios/secundarios (V7-9390)
- El área de datos local ahora se transfiere al trabajo solicitado al enviar un trabajo (V7-9775)
- Compatibilidad de QTEMP para el área de datos y compatibilidad para la creación de valor del área de datos. (V7-9916)
- Control de compromisos: compatibilidad para activar/desactivar el control de compromisos (V7-9956)
- Compatibilidad con los indicadores externos en las palabras clave COMMIT

### Mejoras

- Mejora de la visualización del valor 0 y EDTWRD (V7-8933)
- Compatibilidad con la palabra clave "CHKMSGID" del DSPF (V7-9125)
- Transacción de confirmación de SQL al finalizar el lote (V7-9232)
- Mejora de la compatibilidad de las palabras clave EXPORT e IMPORT para el campo y la estructura de datos (V7-9265)
- Soporta entrada de minúscula DateHelper (V7-9461)
- Admite la conversión de \*CYMD a \*ISO (numérico) (V7-9488)
- Mejora de la gestión del %len incorporado para un campo variable (lado izquierdo y derecho de una expresión) (V7-9733)

- Mejora de la compatibilidad con las funciones integradas '%LOOKUPXX' XX ("LE","LT","GE","GT") (V7-10064)

## Capacidades transversales

### Nuevas características

- CICS: mejora de la transacción Inquire para comprobar el estado de la opción (V7-9712)
- JCL: mejora de la carga de sysprint con el archivo de salida del sistema (V7-9797)
- CICS: mejora de INQUIRE TSQUEUE (V7-9823)
- CICS: mejora del terminal Inquire para el identificador de usuario de la opción (V7-9906)

### Mejoras

- Mejora de la gestión de la comparación con blank (V7-8047)
- Mejore el registro para Jics y Blusam (V7-8847)
- Compatibilidad con atributos extendidos SOSI y símbolo programado F8 para campos dinámicos en BMS (V7-8857)
- Controle el desbordamiento del búfer en el parámetro del programa (V7-9138)
- Mejora de la simultaneidad de escritura de subprocesos para el registro de bloqueos de Blusam (V7-9505)
- Compatibilidad para la configuración de múltiples orígenes de datos para Utility-pgm (V7-9570)
- Modo de bloqueo de nivel de registro de Blusam únicamente (V7-9626)
- Asegúrese de que la persistencia de los metadatos se resista al reinicio del servidor (V7-9748)
- Mejora de la limpieza de DAO en caso de excepción (cierre del navegador) (V7-9790)
- Soporte DummyFile para INFUTILB SYSPUNCH (V7-9799)
- Mejore la compatibilidad con valores negativos en el (V7-9935) NumericEditedType

## Herramientas de modernización, versión 3.6.0

### Temas

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

- [Capacidades transversales](#)

## zOS

### Nuevas características

- JCL: mejore el registro al final del procedimiento (V7-8509)
- PL1: Mejore la generación de bolsas para cada tipo de datos PakedLong (V7-8917)
- JCL: mejore el registro al final del procedimiento cuando el archivo contenga el marcador de “fin” // (V7-9509)
- PL1: mejore la compatibilidad con GET EDIT con flujos de punto fijo y SYSIN (V7-9593)
- DB2: mejore la compatibilidad con el tipo VARGRAPHIC DB2 (V7-9809)
- CICS: mejora del comando QUERY SECURITY para la opción LOGMESSAGE (V7-9969)
- PL1: mejora de la generación de bolsas para CHARG/chargraphic integrado (V7-9989)

### Mejoras

- PL1: mejore la compatibilidad con la palabra clave INCLUDEX (V7-9588)
- PL/I: gestione la palabra clave CHARGRAPHIC como un parámetro válido de cualquier llamada a un método (V7-9589)
- Mejora de la resolución de la variable host de PL1 cuando se nombra con caracteres específicos @ # \$ §. (V7-9654)
- COBOL: compatibilidad de las palabras clave C01... C12 y S01... S05 como parámetro de la instrucción WRITE ADVANCING en la etapa de análisis (V7-9669)

## AS400

### Nuevas características

- Compatibilidad para la transformación de SQL-DDS en Analyzer (V7-7687)
- Automatice la detección de archivos SQL-DDS (V7-7687)
- Implementación del preprocesamiento de SQL-DDS (V7-7687)
- Admite la palabra clave ALIGN (V7-9254)
- Support ExtName para DSPF y arreglos multidim (V7-9663)

- InvalidKey Declaraciones de apoyo sobre COBOL WRITE (V7-9793)

## Mejoras

- Mejora del código de operación TESTB (V7-8865)
- Mejora de la compatibilidad con DECFMT en el enfoque (V7-8933)
- Gestión del indicador resultante en MOVE (V7-9224)
- Mejora de la compatibilidad de la palabra clave TEMPLATE para el campo y la estructura de datos (V7-9278)
- Mejora de LIKEDS (el DS definido con LIKEDS se califica automáticamente) (V7-9302)
- COBOL: mejora de la estructura de generación de indicadores (V7-9423)
- El parámetro constante del prototipo no es de solo lectura (V7-9437)
- Mejora de la palabra clave EDTCDE con el código de edición "Y" (V7-9443)
- Compatibilidad para la generación del campo \*ROUTINE en PSDS e INFDS (V7-9487)
- Mejora del campo de reescritura XXX para que sea independiente (el valor predeterminado se pierde al reescribir) (V7-9522)
- Mejora de la compatibilidad con palabras clave del DSPF (V7-9658)
- Gestión del valor predeterminado ZEROES en formato binario (V7-9666)
- Compatibilidad con puntero implícito (V7-9719)
- Mejora de la gestión del %size de llamadas integrado con un parámetro (V7-9730)
- Mejora del manejo de las referencias a la estructura de datos en las llamadas integradas (%ELEM) (V7-9736)
- Mejora de la gestión de la longitud firmada de los campos con referencia LIKE en la especificación de definición (V7-9738)
- Mejora en REWRITE (V7-9791)
- Mejora de la generación de índices a partir de archivos DDS (V7-9803)
- Mejora de la solidez de los mapeadores con un valor numérico no válido (V7-9813)
- Mejora de la generación de archivos SQLModel y AllIndexes (V7-9818)
- Mejora de la compatibilidad cualificada de DS (V7-9863)
- Mejora de la compatibilidad con LOOKUP (con un campo independiente como un parámetro DS) (V7-9961)
- Mejora de LIKE en el indicador (V7-9985)

- Gestión del indicador resultante en MVR (V7-9995)
- Compatibilidad con el carácter N con tilde (V7-10021)
- Mejora de la generación moderna de archivos DDL a partir de archivos antiguos de SQLDDS (V7-10067)

## Capacidades transversales

### Nuevas características

- Personalización de la ubicación de los recursos con una propiedad yml (D88816105)
- COBOL: compatibilidad de la instrucción EXIT PERFORM para salir de un PERFORM en línea sin utilizar un GO TO / PERFORM ... HASTA (V7-9582)
- Especificación de la codificación heredada predeterminada para considerarla en los metadatos globales. (V7-9883)

### Mejoras

- Mejora en la generación de máscaras (V7-9602)
- Mejora de la preparación anticipada del contexto (V7-9621)
- Hacer que los subprocesos Charset CUSTOM930 sean seguros. (V7-9674)
- Mejora en MOVEA (V7-9773)

## Notas de la versión 3.5.0

Esta versión de las herramientas de tiempo de ejecución y modernización de AWS Blu Age ofrece nuevas funciones para las migraciones antiguas de zOS y AS400, orientadas principalmente a la optimización de los conjuntos de datos y la mensajería, así como capacidades ampliadas de Java como un activo derivado del proceso de transformación. Algunos de los cambios clave de esta versión son:

- Capacidad de migrar programas de CL a Java, además de la característica preexistente de groovy scripts, para facilitar su integración con otros programas modernizados y simplificar la curva de aprendizaje de los clientes mediante la unificación del lenguaje de programación resultante.
- Reducción del tiempo y optimización del rendimiento de las cargas de conjuntos de datos en Redis con la nueva característica de almacenamiento masivo de datos.

- Capacidad para operar y transferir conjuntos de datos según las etapas del trabajo a fin de modernizar los comportamientos tradicionales de los conjuntos de datos.
- Extensión de la migración a SQL para admitir archivos de entrada de VB y migración simplificada a Java 11.
- Varios mecanismos nuevos para una integración más rápida con IBM MQ, incluidos encabezados adicionales, compatibilidad ampliada para GET/PUT y recuperación automática de los metadatos de las colas.
- REST Endpoint para metadatos de conjuntos de datos e importación de conjuntos de datos desde buckets de S3.

Para obtener más información acerca de los cambios incluidos en esta versión, consulte las siguientes secciones.

## Runtime, versión 3.5.0

### Temas

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

## zOS

### Nuevas características

- JCL SORT: gestiona la superposición de nuevas palabras clave (V7-9409)
- ZOS COBOL: mejora la compatibilidad con caracteres flotantes (V7-9404)
- Puerto de RedisJics TSQueue a RedisTemplate & ListOperations (V7-9212)
- ZOS JCL: mejore la ruta del directorio temporal con el directorio de archivos si está definido mediante (V7-9012) UserDefinedParameters
- Gestión de la FUNCIÓN ORD-MAX con ALL (todos los elementos de la matriz) (V7-9366)
- Ahora se utilizan claves prefijadas y en lenguaje natural al almacenar TS Queues en Redis (V7-9212)
- Agregue el punto final get dataset para la API de Blusam

- JCL: compatibilidad con ADD para trabajos por lotes con un nombre que incluya un carácter especial como # (V7-9136)
- La búsqueda de TSMModel ahora se realiza sin problemas bajo demanda (V7-9212)

## Mejoras

- Compatibilidad con INCLUDE sin versiones en archivos LNK (V7-6022)
- MQ: mejora de la compatibilidad de codificación (V7-9652)
- Mejora de la compatibilidad con bytes dobles o conjuntos de caracteres mixtos para distintos tipos de caracteres (V7-9596)
- JCL: la compatibilidad de configuración de filesDirectory en IDCAMS borra instrucciones NONVSAM (V7-9609)
- Compatibilidad con el modo masivo para la carga de conjuntos de datos ESDS y RRDS desde archivos (V7-8639)
- Gestión de la apertura de un ESDS vacío en el modo de entrada. (V7-9287)
- Mejora de la instrucción DEFINE CLUSTER con la compatibilidad para abreviaturas ORD/UNORD (V7-9451)
- Mejoras en el rendimiento del bloqueo de Blusam Redis (V7-8639)
- Mejora de la instrucción DEFINE CLUSTER para que sea compatible con el RECORDSIZE proporcionado en el ámbito del argumento DATA() (V7-9337)
- Añade compatibilidad con los atributos BUFFERSPACE/UNIQUE en las instrucciones DEFINE CLUSTER (V7-9419)
- Mejore la operación de lectura de Blusam para conjuntos de datos de registros de longitud variable. (V7-9391)
- La CICS ADDRESS representa correctamente la CWA que falta como nulo (V7-9491)
- Eliminación de la escritura innecesaria en los bloqueos de los extremos (V7-8639)
- Gestión de la inyección de plantillas de caché de Redis en la caché (V7-9510)
- Decodificación correcta del parámetro BPXWDYN (V7-9417)
- Mejora del consumo de exportación de LISTCAT (V7-9201)
- El nombre de las colas TS de Blusam admite caracteres no imprimibles (V7-9212)
- Gestión de la creación de mapas de recepción para un campo con el mapset nulo (V7-9486)
- Mejore la operación BluesamRelativeFile de eliminación y reescritura para el modo de acceso dinámico. (V7-8989)

## AS400

### Nuevas características

- Añadida una función para generar archivos CL como programas Java mediante un sistema dinámico estándar DS/STM (V7-9427)
- Compatibilidad para el archivo de entrada con modo ADD (V7-9378)
- Se mejoró la gestión del orden de clasificación y la recuperación para admitir el comando cl OPNQRYF (Abrir archivo de consulta) y se agregó la compatibilidad con el parámetro SHARE en. Overrideltem (V7-9364)

### Mejoras

- Support SFLNXTCHG en (V7-8061) UpdateSubfile
- Modificación del alcance del contexto CL al ejecutar el comando CL (V7-9624)
- Gestión del código de retorno del programa BPXWDYN (V7-9417)
- Borrado de los monitores locales. (V7-9624)
- Compatibilidad con la palabra clave RTNCSRLOC del DSPF (V7-9389)
- set OnGreater OrEqual () no establece igual a 1 (V7-9342)
- Actualice la caché de campos en UpdateSubfileRecord (V7-9376)
- Mejora de la compatibilidad con SFLNXTCHG (V7-8061)

## Capacidades transversales

### Nuevas características

- Se ignora el prefijo G en la cadena gráfica literal. (V7-9420)
- ZOS COBOL: mejora de la compatibilidad de Fiedl.initialize() para algunas estructuras especiales (V7-9485)
- Se permite la inicialización del contexto de forma asíncrona para mejorar el rendimiento del inicio del programa (V7-9446)
- SQL publique explícitamente la sentencia de preparación abierta y. ResultSet (V7-9422)
- Mejora del JMS MQ: compatibilidad con MQRFH2 para MQ PUT/V7-7085, compatibilidad con el administrador de colas predeterminado (V7-9400)



- Administración de SQL: se habilitan las conversiones de Lambda en los parámetros de los comandos SET (V7-9492)
- ZOS MQ JMS: se añade compatibilidad con MQCOMIT y MQBACK (V7-9399)
- ZOS IBMMQ: mejora de la compatibilidad con MQINQ (V7-9544)
- Gestión de la operación CONCAT con bytes en lugar de cadenas cuando se utiliza la codificación de doble byte. (V7-8932)
- ZOS IBMMQ: mejora de la compatibilidad con el comando PUT con las opciones SET\_ALL\_CONTEXT (V7-9544)

## Mejoras

- Gestión de los nombres de los archivos gdg con el carácter \$ (V7-9066)
- El diagnóstico de SQL devuelve 1 como cláusula NUMBER cuando la sentencia SQL anterior es correcta. (V7-9410)
- Esquema de un campo con una longitud no nula (V7-7536)
- Compatibilidad integrada con la función PL1 GRAPHIC (V7-9245)
- MQ: se ha agregado compatibilidad para la configuración de campos de MQGMO (V7-9500)
- JMS MQ GET: mejora de la longitud de datos del mensaje devuelto (V7-9502)
- Definición de sqlerrd (3) con el número de elementos recuperados en el contexto de ROWSET. (V7-9371)

## Herramientas de modernización, versión 3.5.0

### Temas

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

## zOS

### Nuevas características

- ZOS PLI: compatibilidad del índice de asteriscos en la asignación con expresión binaria (V7-9178)
- De JCL a BatchScript - Un «//» marca el final de la ejecución del trabajo (V7-9304)

- ZOS PLI: mejora la compatibilidad con caracteres flotantes y el tipo de registro editado numéricamente (V7-8982)
- COBOL: compatibilidad con la función SUM integrada (V7-9367)
- JCL: opcionalmente, se puede comentar el código inactivo después de una sentencia nula (//) (V7-9202)
- JCL: compatibilidad del operador '|' en la instrucción de estado (V7-9499)
- PL/I: comentario sobre las directivas de precompilación en la etapa de preprocesamiento para evitar excepciones de análisis (V7-9507)

## Mejoras

- Gestión de la definición de flujo con un delimitador (V7-9615)
- Mejora de la gestión de las exportaciones de LISTCAT. (V7-9201)
- PL/I: mejora para admitir argumentos “null” implícitos (V7-9204)

## AS400

### Nuevas características

- Compatibilidad de la palabra clave CONCAT de DDS (V7-9439)
- Refactorización del código java generado para las palabras clave del DSPF. (V7-7700)
- Se admite variar las palabras clave en los campos de una definición de estructura de datos (V7-9029)

### Mejoras

- Mejora del análisis de la relación lógica Y/O (V7-9352)
- COBOL Mejora del mapeo entre vo y dEntity (V7-9449)
- Se muestra un valor vacío si la entrada numérica está enfocada (V7-9374)
- Variable local en SQL Declare Cursor (V7-9456)
- Problema de alcance con un DS vacío (V7-9466)
- Truncamiento de la columna 80 antes del análisis (V7-9632)
- Mejora de la gestión de las referencias de campo y las llamadas integradas en las palabras clave (DIM, LIKE,...) en la especificación de definiciones (V7-9358)

- Comentarios sobre compatibilidad con SQL (-- ) (V7-9632)
- FullFree analizando, escriba Fecha/Hora/Marca de tiempo (V7-9542)
- Incluya el FullFree SQLCA del análisis (V7-9333)
- Mejora de la compatibilidad con el nivel de control. (V7-9610)
- Gestión de la comparación de DS con \*BLANKS (V7-9668)
- Mejora de la compatibilidad con varios indicadores en el DDS (V7-9318)
- Mejora de la compatibilidad con varios programas de DSPF (V7-9657)
- Mejora de la gestión del campo con LIKE (caso de estructura de datos similar y caso de estructura de datos similar en una matriz) (V7-9213)
- RPG libre, gestión de continuación en literales (V7-9686)
- Mejora de la compatibilidad con los registros de fin de programa (V7-9452)
- Compatibilidad de la frase LINKAGE en la instrucción CALL. (V7-9685)
- Código de operación CASXX (CASBB sin grupo CASXX) (V7-9357)
- Mejore el análisis de los juegos de rol (V7-9457) FullFree
- El %LEN incorporado no admite DS como argumento (V7-9267)
- Mejoras en MOVEA cuando el factor 2 es \*ALL'X... ' (V7-9228)
- Admite la asignación con el campo RENAME (V7-9385)

## Capacidades transversales

### Nuevas características

- Herramienta SQL Migrator: se ha agregado la opción OID para una longitud de registro variable en el paso de carga de ebclic. (V7-9380)
- Herramienta SQL Migrator: compatibilidad con Java 11 en la opción OID (V7-9599)

### Mejoras

- Mejora de la compatibilidad con matrices anidadas (V7-9595)
- Se sustituye el carácter  $\hat{\neg}$  por ! en caso de que  $\hat{\neg}$  sea compatible con la codificación original. (V7-9465)
- JCL: compatibilidad con la terminación normal PASS para compartir conjuntos de datos entre las etapas del trabajo (V7-9504)

- se aplica ON NULL a la definición de columnas en ORACLE cuando se trate de columnas tipo VARCHAR y bases de datos que admiten valores NULL. (V7-9681)
- Mejora de la conformidad con la inyección de Spring (V7-9635)

## AWS Instrucciones de actualización de Blu Age

Esta página contiene instrucciones para actualizar la versión AWS Blu Age.

### Migración de la versión 3.10.0 a la 4.0.0

El principal cambio de la versión 4.0.0 es la migración de Spring Boot 2.7 a Spring Boot 3.2 y de Tomcat 9 a Tomcat 10.

### Cambios en el código

En esta sección se enumeran los cambios necesarios para que el código modernizado sea compatible con AWS Blu Age Runtime 4.0.0. Puedes saltarte esta sección si decides lanzar una nueva generación con la versión 4.0.0 en Blu Insights (Transformation Center).

### Cambios en POM

Grupo	ArtifactId	Cambio
org.slf4j	slf4j-api	Eliminar (es una dependencia transitiva)
org.yaml	snakeyaml	Eliminar (es una dependencia transitiva)
org.springframework.boot	spring-boot-starter-web	- Actualice spring.boot.version a la versión 3.2.4 - Elimine la exclusión de log4j-to-slf
org.springframework.boot	spring-boot-starter-jta-atomikos	Cambiar a com.atomikos:3-starter:6.0.0 transactions-spring-boot
org.apache.commons	commons-dbcp2	Actualice a 2.10.0

Grupo	ArtifactId	Cambio
org.postgresql	postgresql	Actualice a 42.7.2
com.microsoft.sqlserver	mssql-jdbc	Actualice a 12.4.2.jre11
com.oracle.database.jdbc	ojdbc8	Cambie a la versión 23.3.0.23.09 de ojdbc11

## Migre de Javax a Yakarta

La actualización de Tomcat incluye una migración del paquete Java Javax a Yakarta. Asegúrese de actualizar las importaciones en consecuencia desde `javax.*` a `Yakarta.*`.

Casi todas las clases antiguas a las que se hace referencia en el paquete Javax se encuentran en Yakarta. Las excepciones conocidas a esto son los `javax.xml` paquetes `javax.sql` y, que siguen sin cambios.

## El cambio de Atomikos

Debido al cambio de dependencia mencionado anteriormente, las referencias a `org.springframework.boot.jta.atomikos.AtomikosDataSourceBean` deben cambiarse a `com.atomikos.spring.AtomikosDataSourceBean`

## Eliminación del dialecto de PostgreSQL

Se elimina la clase personalizada `PostgreSQLDialect.java`. También se deben eliminar las referencias a ella en el lanzador principal.

## Despliegue (AWS Blu Age Runtime (no gestionado))

### Tomcat

Esta versión es compatible con 10.1.17 Tomcat. Es necesario actualizar el servidor Tomcat a esta versión para ejecutar Blu Age Runtime. 4.0.0 Asegúrese de portar los cambios de configuración anteriores (especialmente las propiedades de Catalina).

### Dependencias compartidas

La carpeta compartida en tiempo de ejecución contiene las up-to-date dependencias.

### Dependencias adicionales

Si usaste dependencias adicionales (no incluidas en el tiempo de ejecución), es posible que tengas que actualizarlas. El archivo readme de la carpeta adicional muestra las versiones compatibles.

## AWS Conceptos de Blu Age Runtime

Comprender los conceptos básicos del tiempo de ejecución de AWS Blu Age puede ayudarle a comprender cómo se modernizan sus aplicaciones mediante la refactorización automatizada.

### Temas

- [AWS Arquitectura de alto nivel de ejecución de Blu Age](#)
- [AWS Estructura de la era azul de una aplicación modernizada](#)
- [Data Simplifier](#)

## AWS Arquitectura de alto nivel de ejecución de Blu Age

Como parte de la solución de AWS Blu Age para modernizar los programas antiguos a Java, el AWS Blu Age Runtime proporciona un punto de entrada unificado y basado en REST para las aplicaciones modernizadas y un marco de ejecución para dichas aplicaciones, a través de bibliotecas que proporcionan construcciones heredadas y una estandarización de la organización del código de los programas.

Estas aplicaciones modernizadas son el resultado del proceso de refactorización automatizada de AWS Blu Age para modernizar los programas de mainframe y rango medio (denominados «heredados» en el siguiente documento) a una arquitectura basada en la web.

Los objetivos de AWS Blu Age Runtime son la reproducción del comportamiento de los programas heredados (isofuncionalidad), el rendimiento (con respecto al tiempo de ejecución de los programas y el consumo de recursos) y la facilidad de mantenimiento de los programas modernizados por parte de los desarrolladores de Java, mediante el uso de entornos y modismos familiares como tomcat, Spring, getters/setters y API fluidas.

### Temas

- [AWS Componentes de ejecución de Blu Age](#)
- [Entornos de ejecución](#)
- [Ausencia de estado y gestión de sesiones](#)
- [Alta disponibilidad y ausencia de estado](#)

## AWS Componentes de ejecución de Blu Age

El entorno de ejecución de AWS Blu Age se compone de dos tipos de componentes:

- Un conjunto de bibliotecas java (archivos jar), a las que a menudo se hace referencia como “la carpeta compartida”, y que proporcionan construcciones e instrucciones heredadas.
- Un conjunto de aplicaciones web (archivos war) que contienen aplicaciones web basadas en Spring y que proporcionan un conjunto común de marcos y servicios para los programas modernizados.

En las siguientes secciones se detalla el rol de estos dos componentes.

### AWS Bibliotecas de Blu Age

Las bibliotecas AWS Blu Age son un conjunto de archivos jar almacenados en una shared/subcarpeta que se añade a la ruta de clases estándar de Tomcat, para que estén disponibles para todos los programas Java modernizados. Su objetivo es proporcionar características que no están disponibles de forma nativa ni fácilmente disponibles en el entorno de programación Java, sino que son típicas de los entornos de desarrollo tradicionales. Estas características se exponen de una forma que resulte lo más familiar posible para los desarrolladores de Java (getters/setters, API fluidas y basadas en clases). Un ejemplo importante es la biblioteca Data Simplifier, que proporciona a los programas Java estructuras antiguas de diseño y manipulación de la memoria (que se encuentran en los lenguajes COBOL, PL1 o RPG). Estos archivos jar son una dependencia fundamental para el código Java modernizado generado a partir de programas heredados. Para obtener más información sobre Data Simplifier, consulte [Data Simplifier](#).

### Aplicación web

Los archivos de aplicaciones web (WAR) son una forma estándar de implementar código y aplicaciones en el servidor de aplicaciones Tomcat. Las que se proporcionan como parte del entorno de ejecución de AWS Blu Age tienen como objetivo proporcionar un conjunto de marcos de ejecución que reproduzcan los entornos y monitores de transacciones tradicionales (lotes de JCL, CICS, IMS...) y los servicios necesarios asociados.

El más importante es gapwalk-application (a menudo abreviado como “Gapwalk”), que proporciona un conjunto unificado de puntos de entrada basados en REST para activar y controlar la ejecución de transacciones, programas y lotes. Para obtener más información, consulte [AWS API de ejecución de Blu Age](#).

Esta aplicación web asigna subprocesos y recursos de ejecución de Java para ejecutar programas modernizados en el contexto para el que fueron diseñados. En la siguiente sección se detallan ejemplos de dichos entornos reproducidos.

Otras aplicaciones web añaden al entorno de ejecución (con más precisión, al “registro de programas”, que se describe a continuación) programas que emulan los programas antiguos disponibles y a los que se puede acceder desde ellos. Dos categorías importantes son las siguientes:

- Emulación de programas proporcionados por el sistema operativo: los lotes impulsados por JCL esperan poder llamar a una variedad de programas de manipulación de archivos y bases de datos como parte de su entorno estándar. Entre los ejemplos se incluyen SORT/DFSORT y IDCAMS. Para ello, se proporcionan programas Java que reproducen dicho comportamiento y se pueden invocar utilizando las mismas convenciones que los programas antiguos.
- Los “controladores”, que son programas especializados proporcionados por el marco de ejecución o el middleware como puntos de entrada. Un ejemplo es CBLTDLI, de qué dependen los programas COBOL que se ejecutan en el entorno IMS para acceder a los servicios relacionados con el IMS (IMS DB, diálogo con el usuario a través de MFS, etc.).

## Registro de programas

Para participar y aprovechar esas constructos, marcos y servicios, los programas Java modernizados a partir de los antiguos se adhieren a una estructura específica documentada en [AWS Estructura de la era azul de una aplicación modernizada](#). Al iniciarse, el motor de ejecución de AWS Blu Age recopilará todos estos programas en un «registro de programas» común para poder invocarlos (y llamarlos entre sí) posteriormente. El registro de programas ofrece un acoplamiento flexible y posibilidades de descomposición (ya que los programas que se llaman entre sí no tienen que modernizarse simultáneamente).

## Entornos de ejecución

Están disponibles los entornos y coreografías tradicionales que se encuentran con más frecuencia:

- Los lotes impulsados por JCL, una vez modernizados a programas Java y scripts Groovy, se pueden iniciar de forma sincrónica (bloqueo) o asincrónica (separada). En este último caso, su ejecución se puede supervisar a través de puntos de conexión REST.
- Un subsistema AWS Blu Age proporciona un entorno de ejecución similar al CICS mediante:
  - un punto de entrada utilizado para iniciar una transacción del CICS y ejecutar los programas asociados, respetando la coreografía de los “niveles de ejecución” del CICS,



- un almacenamiento externo para las definiciones de recursos,
- un conjunto homogéneo de API fluidas en Java que reproducen instrucciones EXEC CICS,
- un conjunto de clases conectables que reproducen los servicios del CICS, como las colas de almacenamiento temporal, las colas temporales de datos o el acceso a los archivos (normalmente hay varias implementaciones disponibles, como Amazon Managed Service para Apache Flink, Amazon Simple Queue Service o RabbitMQ para TD Queues),
- para las aplicaciones orientadas al usuario, el formato de descripción de pantalla BMS se moderniza para convertirse en una aplicación web angular y se admite el correspondiente cuadro de diálogo “pseudoconversacional”.
- Del mismo modo, otro subsistema proporciona una coreografía basada en mensajes IMS y admite la modernización de las pantallas de interfaz de usuario en formato MFS.
- Además, un tercer subsistema permite la ejecución de programas en un entorno similar al de iSeries, incluida la modernización de las pantallas especificadas por el DSPF (Display File).

Todos estos entornos se basan en servicios comunes a nivel de sistema operativo, tales como:

- la emulación de la asignación y el diseño de la memoria tradicionales (Data Simplifier),
- reproducción basada en subprocesos de Java del mecanismo de ejecución y paso de parámetros de las “unidades de ejecución” de COBOL (declaración CALL),
- emulación de organizaciones planas, concatenadas, VSAM (a través del conjunto de bibliotecas Blusam) y de conjuntos de datos GDG,
- acceso a almacenes de datos, como RDBMS (declaraciones EXEC SQL).

## Ausencia de estado y gestión de sesiones

Una característica importante del tiempo de ejecución de AWS Blu Age es permitir escenarios de alta disponibilidad (HA) y escalabilidad horizontal al ejecutar programas modernizados.

La piedra angular de esto es la ausencia de estado, un ejemplo importante de lo cual es la gestión de las sesiones HTTP.

### Gestión de sesiones

Como Tomcat se basa en web, un mecanismo importante para ello es la gestión de sesiones HTTP (tal como lo proporcionan Tomcat y Spring) y el diseño sin estado. El diseño de la ausencia de estado se basa en lo siguiente:

- los usuarios se conectan a través de HTTPS,
- los servidores de aplicaciones se implementan detrás de un equilibrador de carga,
- cuando un usuario se conecte por primera vez a la aplicación, se autenticará y el servidor de la aplicación creará un identificador (normalmente dentro de una cookie)
- este identificador se utilizará como clave para guardar y recuperar el contexto del usuario hacia/desde una caché externa (almacén de datos).

La gestión de las cookies se realiza automáticamente mediante el marco AWS Blu Age y el servidor Tomcat subyacente, de forma transparente para el usuario. El navegador de Internet del usuario lo gestionará automáticamente.

La aplicación web Gapwalk puede almacenar el estado de la sesión (el contexto) en varios almacenes de datos:

- Amazon ElastiCache para Redis
- Clúster de Redis
- en un mapa de memoria (solo para entornos de desarrollo e independientes, no apto para HA).

## Alta disponibilidad y ausencia de estado

En términos más generales, uno de los principios de diseño del marco de la era AWS azul es la apatridia: la mayoría de los estados no transitorios necesarios para reproducir el comportamiento de los programas heredados no se almacenan en los servidores de aplicaciones, sino que se comparten a través de una «fuente única de información» externa y común.

Algunos ejemplos de estos estados son las colas de almacenamiento temporal o las definiciones de recursos del CICS, y los almacenamientos externos típicos de esos estados son los servidores o las bases de datos relacionales compatibles con Redis.

Este diseño, combinado con el equilibrador de carga y las sesiones compartidas, permite que la mayor parte del diálogo orientado al usuario (OLTP, “procesamiento transaccional en línea”) se pueda distribuir entre varios “nodos” (en este caso, instancias de Tomcat).

De hecho, un usuario puede ejecutar una transacción en cualquier servidor sin importarle si la siguiente llamada a la transacción se realiza en un servidor diferente. Luego, cuando se genera un nuevo servidor (debido al escalado automático o para reemplazar un servidor que no esté en buen estado), podemos garantizar que cualquier servidor accesible y en buen estado pueda ejecutar la

transacción según lo esperado con los resultados adecuados (valor devuelto esperado, cambio de datos esperado en la base de datos, etc.).

## AWS Estructura de la era azul de una aplicación modernizada

Este documento proporciona detalles sobre la estructura de las aplicaciones modernizadas (mediante las herramientas de refactorización de la modernización AWS del mainframe), de modo que los desarrolladores puedan realizar diversas tareas, como:

- Navegar por las aplicaciones sin problemas.
- Desarrollar programas personalizados a los que se pueda acceder desde las aplicaciones modernizadas.
- Refactorizar de forma segura las aplicaciones modernizadas.

Damos por supuesto que ya tiene conocimientos básicos sobre lo siguiente:

- conceptos de codificación comunes heredados, como los registros, los conjuntos de datos y sus modos de acceso a los registros (indexados, secuenciales), VSAM, unidades de ejecución, scripts jcl, conceptos de CICS, etc.
- codificación Java utilizando el [marco de Spring](#)
- A lo largo del documento, utilizamos `short class names` para facilitar la lectura. Para obtener más información, consulte [AWS Asignaciones de nombres totalmente calificadas de Blu Age](#) para recuperar los nombres completos correspondientes a los elementos de tiempo de ejecución de AWS Blu Age y [Asignaciones de nombres totalmente cualificadas de terceros](#) para recuperar los nombres completos correspondientes para los elementos de terceros.
- [Todos los artefactos y muestras se toman de los resultados del proceso de modernización de la aplicación CardDemo COBOL/CICS de muestra.](#)

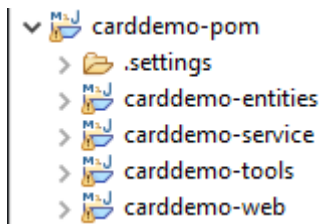
### Temas

- [Organización de artefactos](#)
- [Ejecutar y llamar a programas](#)
- [Escriba su propio programa](#)
- [Asignaciones de nombres totalmente cualificadas](#)

## Organización de artefactos

AWS Las aplicaciones modernizadas de Blu Age se empaquetan como aplicaciones web java (.war), que se pueden implementar en un servidor JEE. Normalmente, el servidor es una instancia de [Tomcat](#) que incorpora el tiempo de ejecución de AWS Blu Age, que actualmente se basa en los marcos [Springboot y Angular \(para la parte de la interfaz de usuario\)](#).

La aplicación war agrega varios artefactos de componentes (.jar). Cada artefacto jar es el resultado de la compilación (con la herramienta [maven](#)) de un proyecto Java específico cuyos elementos son el resultado del proceso de modernización.



La organización básica se basa en la siguiente estructura:

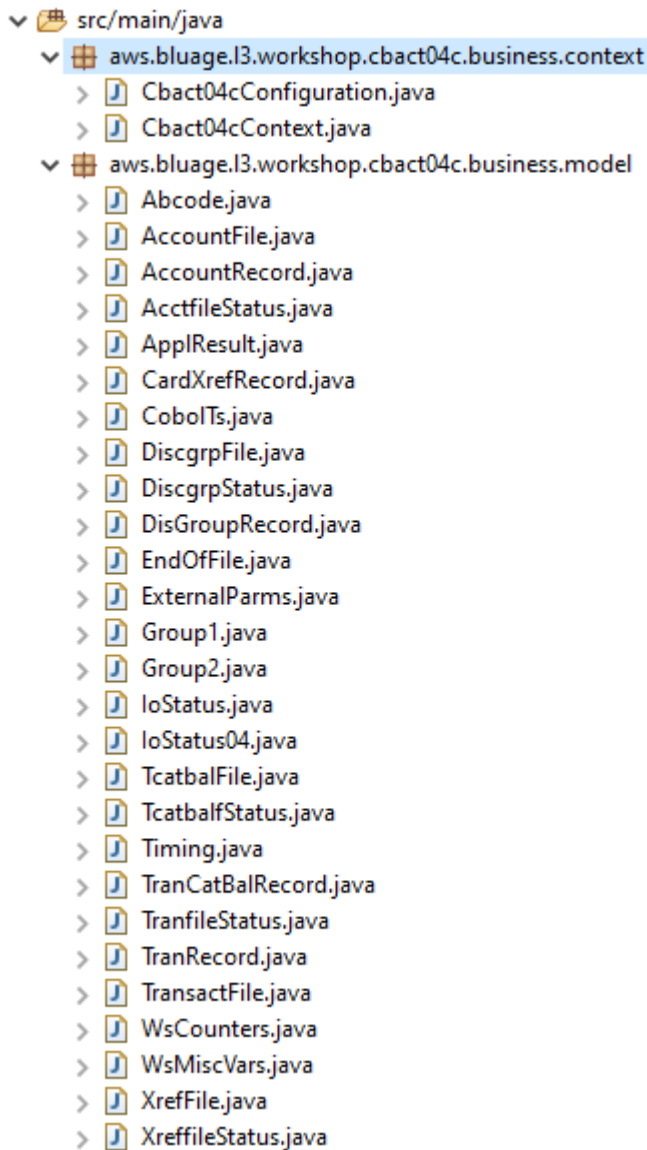
- Proyecto de entidades: contiene elementos del modelo de negocio y del contexto. El nombre del proyecto generalmente termina con “-entities”. Normalmente, en el caso de un programa COBOL anterior, esto corresponde a la modernización de la sección de E/S (conjuntos de datos) y de la división de datos. Puede tener más de un proyecto de entidades.
- Proyecto de servicio: contiene elementos de modernización de la lógica empresarial heredados. Por lo general, la división de procedimientos de un programa COBOL. Puede tener más de un proyecto de servicio.
- Proyecto de utilidad: contiene herramientas y utilidades comunes compartidas, utilizadas por otros proyectos.
- Proyecto web: contiene la modernización de los elementos relacionados con la interfaz de usuario cuando corresponde. No se utiliza para proyectos de modernización únicamente por lotes. Estos elementos de la interfaz de usuario pueden provenir de mapas BMS de CICS, componentes de IMS MFS y otras fuentes de interfaz de usuario de unidad central. Puede tener más de un proyecto web.

## Contenido de proyecto de entidades

### Note

Las siguientes descripciones solo se aplican a las salidas de modernización de COBOL y PL/I. Las salidas de la modernización de RPG se basan en un diseño diferente.

Antes de cualquier refactorización, la organización de los paquetes en el proyecto de la entidad está vinculada a los programas modernizados. Puede lograr esto de dos maneras diferentes. La forma preferida es utilizar la caja de herramientas de refactorización, que funciona antes de activar el mecanismo de generación de código. Se trata de una operación avanzada, que se explica en las capacitaciones. BluAge Para obtener más información, consulte el taller [Refactorings](#). Este enfoque le permite conservar la capacidad de volver a generar el código Java más adelante, para beneficiarse de nuevas mejoras en el futuro, por ejemplo). La otra forma es realizar una refactorización de Java regular, directamente sobre el código fuente generado, utilizando cualquier enfoque de refactorización de Java que desee aplicar, bajo su propia responsabilidad.



## Clases relacionadas con el programa

Cada programa modernizado está relacionado con dos paquetes, un paquete `business.context` y un paquete `business.model`.

- `base package.program.business.context`

El subpaquete `business.context` contiene dos clases, una clase de configuración y una clase de contexto.

- Una clase de configuración para el programa, que contiene detalles de configuración específicos para el programa en cuestión, como el juego de caracteres que se utilizará para representar los elementos de datos basados en caracteres, el valor

de bytes predeterminado para rellenar los elementos de la estructura de datos, etc. El nombre de la clase termina con "Configuration". Está marcado con la anotación `@org.springframework.context.annotation.Configuration` y contiene un único método que debe devolver un objeto `Configuration` configurado correctamente.

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

- Una clase de contexto, que sirve de puente entre las clases de servicios del programa (véase más abajo) y las estructuras de datos (`Record`) y los conjuntos de datos (`File`) del subpaquete del modelo (véase más abajo). El nombre de la clase termina con "Context" y es una subclase de la clase `RuntimeContext`.

```

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

```

- *base package.program.business.model*

El subpaquete del modelo contiene todas las estructuras de datos que puede utilizar el programa en cuestión. Por ejemplo, cualquier estructura de datos COBOL de nivel 01 corresponde a una clase del subpaquete del modelo (las estructuras de datos de nivel inferior son propiedades de la estructura de nivel 01 que poseen). Para obtener más información sobre cómo modernizamos estructuras de datos 01, consulte [Data Simplifier](#).

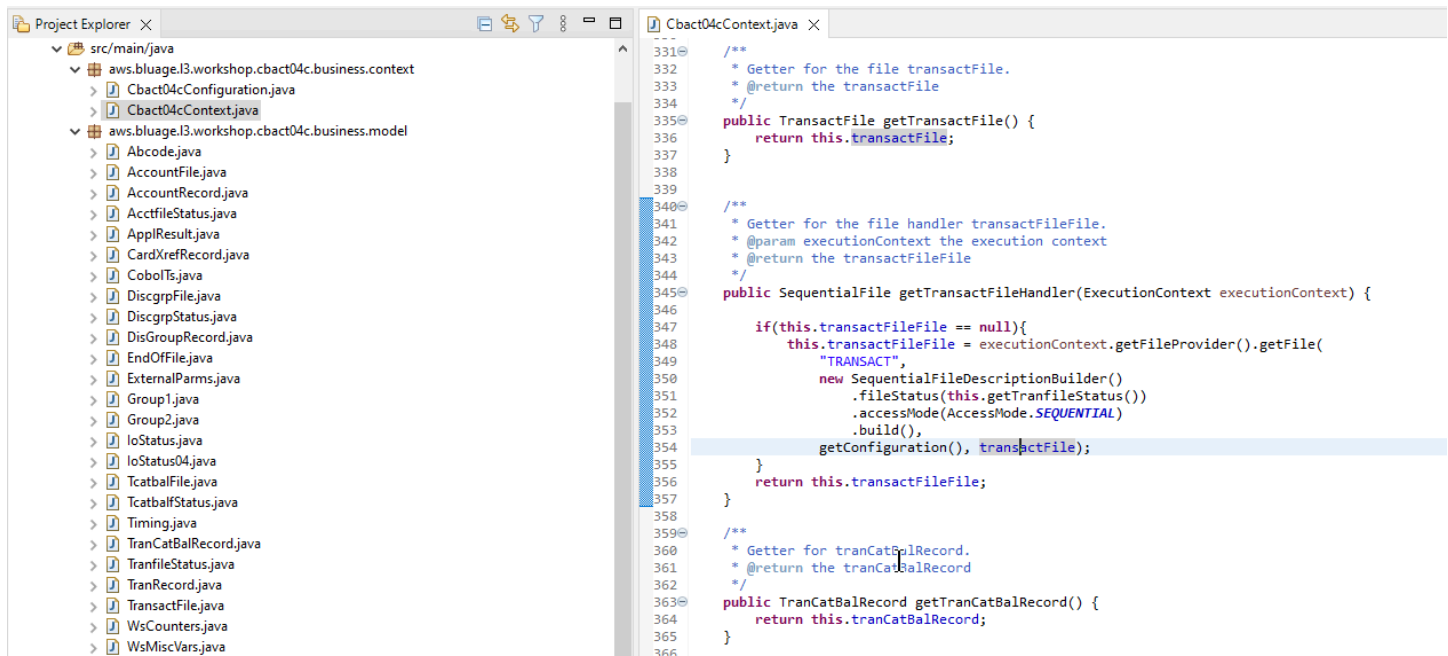


```

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

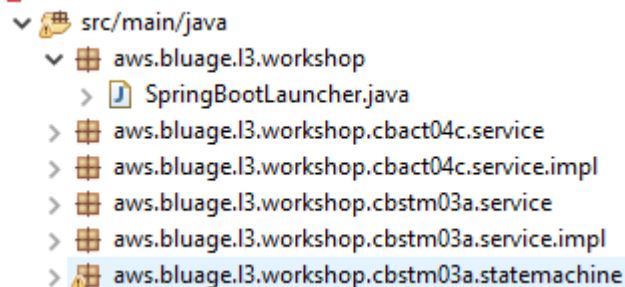
```

Todas las clases amplían la clase `RecordEntity`, que representa el acceso a una representación de un registro empresarial. Algunos de los registros tienen un propósito especial, ya que están vinculados a un `File`. El enlace entre a `Record` y a `File` se realiza en los `FileHandler` métodos\* correspondientes que se encuentran en la clase `context` al crear el objeto de fichero. Por ejemplo, en la siguiente lista se muestra cómo `TransactfileFile File` está enlazado al `TransactFile Record` (del subpaquete del modelo).



## Contenido del proyecto de servicio

Cada proyecto de servicio incluye una aplicación [Springboot](#) dedicada, que se utiliza como columna vertebral de la arquitectura. Esto se materializa a través de la clase denominada `SpringBootLauncher`, ubicada en el paquete base de los orígenes Java del servicio:



Esta clase es, en particular, responsable de:

- Hacer de enlace entre las clases de programas y los recursos administrados (orígenes de datos / administradores de transacciones / asignaciones de conjuntos de datos / etc...).
- Proporcionar un `ConfigurableApplicationContext` a los programas.
- Descubrir todas las clases marcadas como componentes de Spring (`@Component`).
- Garantizar que los programas están registrados correctamente en el `ProgramRegistry`. Consulte el método de inicialización responsable de este registro.

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

## Artefactos relacionados con el programa

Sin necesidad de una refactorización previa, las salidas de la modernización de la lógica empresarial se organizan en dos o tres paquetes por programa tradicional:

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

El caso más exhaustivo constará de tres paquetes:

- *base package.program.service*: contiene una interfaz denominada `ProgramProcess`, que cuenta con métodos empresariales para gestionar la lógica empresarial y preservar el flujo de control de ejecución tradicional.
- *base package.program.service.impl*: contiene una clase denominada `ProgramProcessImpl`, que es la implementación de la interfaz de procesos descrita anteriormente. Aquí es donde las declaraciones heredadas se «traducen» a declaraciones de Java, basándose en el marco de AWS Blu Age:

```

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

```

- *base package.program.statemachine*: es posible que este paquete no esté siempre presente. Es necesaria cuando la modernización del flujo de control tradicional requiere utilizar un enfoque de máquina de estados (es decir, utilizar el [StateMachine marco Spring](#)) para cubrir adecuadamente el flujo de ejecución tradicional.

En ese caso, el subpaquete `statemachine` contiene dos clases:

- **ProgramProcedureDivisionStateMachineController**: una clase que amplía una clase que implementa las interfaces `StateMachineController` (define las operaciones necesarias para controlar la ejecución de una máquina de estados) y `StateMachineRunner` (define las operaciones necesarias para ejecutar una máquina de estados), utilizadas para controlar la mecánica de la máquina de estados de Spring; por ejemplo, `SimpleStateMachineController` como en el caso de muestra.

```

1 package aws.bluage.l3.workshop.cocrdslc.statemachine;
2
3 import aws.bluage.l3.workshop.cocrdslc.business.context.CocrdslcContext;
4
5 /**
6  * Controller managing the state machine "CocrdslcProcedureDivisionStateMachine" execution.
7  */
8 @Component("aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineController")
9 @Import({
10     aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineService.class
11 })
12 @Lazy
13 public class CocrdslcProcedureDivisionStateMachineController extends SimpleStateMachineController<States, Events> {
14
15     /**
16      * State machine states.
17      */
18     public enum States {
19         _0000_MAIN_1, _0000_MAIN, ABEND_ROUTINE, FINAL, LOCAL_FINAL
20     }
21
22     /**
23      * State machine events.
24      */
25     public enum Events {
26         TO_0000_MAIN_1, TO_0000_MAIN, TO_ABEND_ROUTINE, TO_FINAL, TO_LOCAL_FINAL
27     }
28
29     /**
30      * State machine state process service provider.
31      */
32     @Autowired
33     @Lazy
34     private CocrdslcProcedureDivisionStateMachineService stateProcess;
35
36     @Override
37     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
38         throw new UnsupportedOperationException("Please use the four arguments configureStateMachine method instead: configureStateMachine(StateMachineStateConfigurer<States, Events> states, "
39             + "StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl)");
40     }
41
42     @Override
43     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl) throws Exception {
44         StateConfigurer<States, Events> configurer = states.withStates();
45         configurer.initial(States._0000_MAIN_1).end(States.FINAL);
46         configurer.state(States._0000_MAIN_1);
47         configurer.state(States.FINAL);
48
49         StateConfigurer<States, Events> subConfigurer = states.withStates().parent(States._0000_MAIN_1);
50         subConfigurer.initial(States._0000_MAIN).end(States.LOCAL_FINAL);
51         CocrdslcContext lctx = (CocrdslcContext) ctx;
52         subConfigurer.state(States._0000_MAIN, buildAction(() -> {stateProcess._0000Main(lctx, ctrl)}), null);
53         subConfigurer.state(States.ABEND_ROUTINE, buildAction(() -> {stateProcess.abendRoutine(lctx, ctrl)}), null);
54
55         configureTransitions(transitions);
56     }
57
58     /**
59      * Declare state machine transitions.
60      * @param transitions the transitions configuration helper
61      */
62     private void configureTransitions(StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
63         transitions.withLocal().source(States._0000_MAIN_1).target(States.ABEND_ROUTINE).event(Events.TO_ABEND_ROUTINE);
64         transitions.withExternal().source(States.ABEND_ROUTINE).target(States.FINAL).event(Events.TO_FINAL);
65     }
66 }
67
68
69
70
71
72
73
74
75
76
77
78
79
80

```

El controlador de la máquina de estados define los diferentes estados posibles y las transiciones entre ellos, lo que reproduce el flujo de control de ejecución heredado para el programa dado.

Al crear la máquina de estados, el controlador hace referencia a los métodos que se definen en la clase de servicio asociada ubicada en el paquete de la máquina de estados y que se describen a continuación:

```

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

```

- *ProgramProcedureDivisionStateMachineService*: esta clase de servicio representa una parte de la lógica empresarial que debe estar vinculada a la máquina de estados que crea el controlador de la máquina de estados, como se describió anteriormente.

El código de los métodos de esta clase utiliza los eventos definidos en el controlador de la máquina de estados:

```

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

```

```

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

```

El servicio statemachine también realiza llamadas a la implementación del servicio de procesos descrita anteriormente:

```

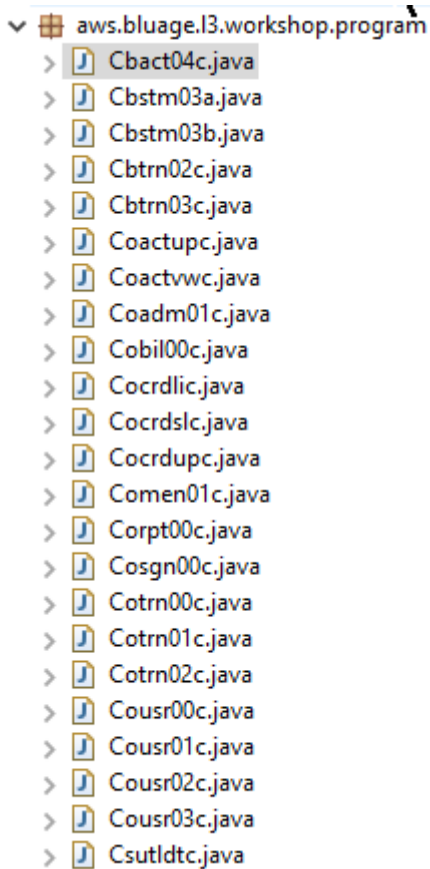
CocrdslcProcedureDivisionStateMachineService.java X
166
167      /*
168      .....
169      COMING FROM CREDIT CARD LIST SCREEN
170      SELECTION CRITERIA ALREADY VALIDATED
171      ..... */
172  } else if (ctx.getCarddemoCommarea().isCdemoPgmEnter() && DataUtils.compare(ctx.getCarddemoCommarea().getCdemoFromProgramReference(),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

```

Además, un paquete *base package*.program desempeña un papel importante, ya que reúne una clase por programa, que servirá como punto de entrada al programa (más detalles sobre esto



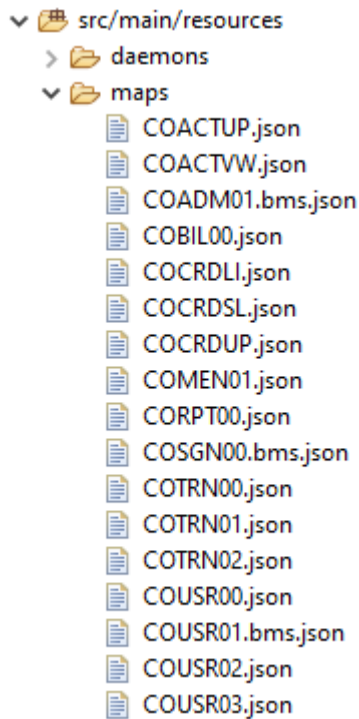
más adelante). Cada clase implementa la interfaz de `Program`, marcador del punto de entrada del programa.



## Otros artefactos

- Artefactos complementarios MAP de BMS

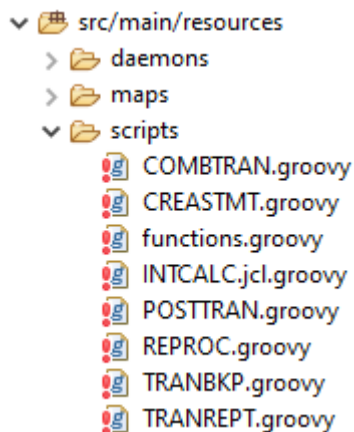
Además de los artefactos relacionados con el programa, el proyecto de servicio puede contener otros artefactos para diversos fines. En el caso de la modernización de una aplicación en línea del CICS, el proceso de modernización genera un archivo json y lo coloca en la carpeta de mapas de la carpeta `/src/main/resources`:



El motor de tiempo de ejecución de Blu Age consume esos archivos json para vincular los registros utilizados por la instrucción SEND MAP con los campos de la pantalla.

- Scripts groovy

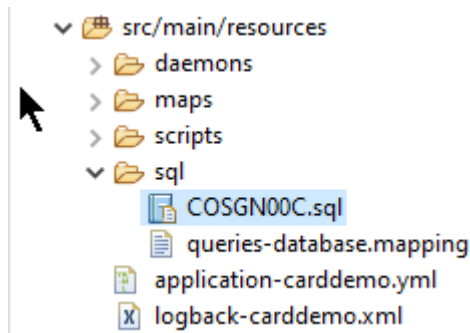
Si la aplicación anterior tenía scripts JCL, estos se han modernizado como scripts [groovy](#) y se guardan en la carpeta /src/main/resources/scripts (hablaremos de esa ubicación específica más adelante):



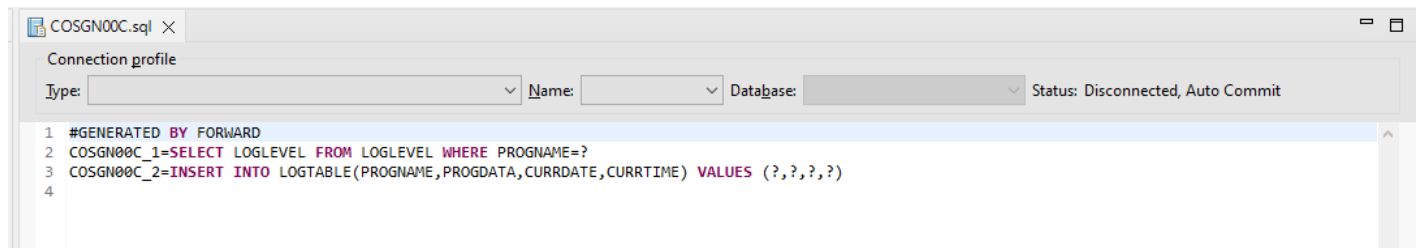
Estos scripts se utilizan para lanzar trabajos por lotes (cargas de trabajo de procesamiento de datos dedicadas, no interactivas y con un uso intensivo de la CPU).

- Archivos SQL

Si la aplicación antigua utilizaba consultas SQL, las consultas SQL modernizadas correspondientes se habrán recopilado en archivos de propiedades dedicados, con el patrón de nomenclatura program.sql, donde program es el nombre del programa que utiliza esas consultas.



El contenido de esos archivos sql es una colección de entradas (clave=consulta), donde cada consulta está asociada a una clave única, que el programa modernizado utiliza para ejecutar la consulta en cuestión:



Por ejemplo, el programa COSGN00C ejecuta la consulta con la clave "COSGN00C\_1" (la primera entrada del archivo sql):

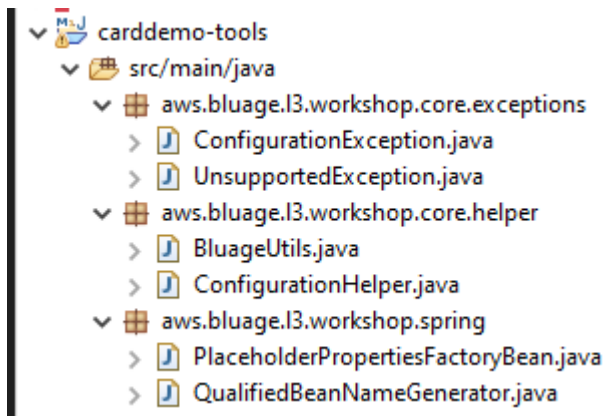
```

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

```

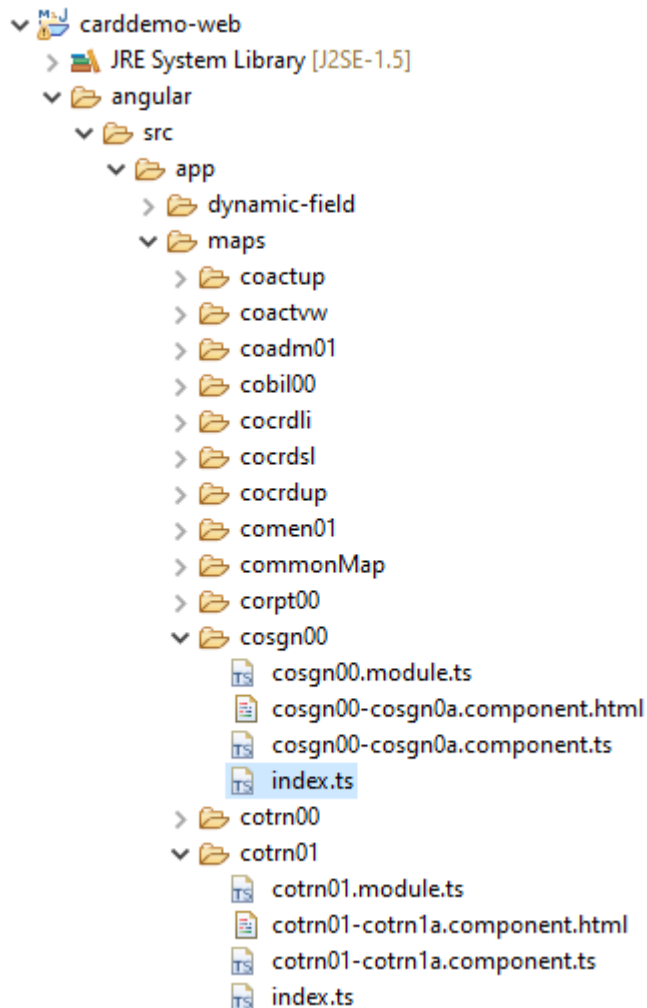
## Contenido de proyecto de utilidades

El proyecto de utilidades, cuyo nombre termina con "-tools", contiene un conjunto de utilidades técnicas que podrían ser utilizadas por todos los demás proyectos.



## Contenido de los proyectos web

El proyecto web solo está presente al modernizar los elementos de la interfaz de usuario heredados. Los elementos de la interfaz de usuario moderna que se utilizan para crear la interfaz de usuario modernizada de la aplicación se basan en [Angular](#). La aplicación de ejemplo utilizada para mostrar los artefactos de modernización es una aplicación COBOL/CICS que se ejecuta en una unidad central. El sistema CICS utiliza MAP para representar las pantallas de la interfaz de usuario. Los elementos modernos correspondientes serán, para cada mapa, un archivo html acompañado de archivos [TypeScript](#):



El proyecto web solo se ocupa del aspecto inicial de la aplicación. El proyecto de servicio, que se basa en los proyectos de utilidades y entidades, proporciona los servicios de backend. El enlace entre el front-end y el backend se realiza a través de la aplicación web llamada Gapwalk-Application, que forma parte de la distribución estándar de tiempo de ejecución de Blu Age. AWS

## Ejecutar y llamar a programas

En los sistemas heredados, los programas se compilan como ejecutables independientes que pueden autodenominarse mediante un mecanismo CALL, como la sentencia CALL de COBOL, pasando argumentos cuando es necesario. Las aplicaciones modernizadas ofrecen la misma capacidad, pero utilizan un enfoque diferente, ya que la naturaleza de los artefactos involucrados difiere de la de los antiguos.

Desde el punto de vista modernizado, los puntos de entrada de los programas son clases específicas que implementan la interfaz de `Program`, son componentes de Spring (`@Component`) y se encuentran en proyectos de servicio, en un paquete denominado `base package.program`.

## Registro de programas

Cada vez que se inicia el servidor [Tomcat](#) que aloja las aplicaciones modernizadas, también se inicia la aplicación Springboot del servicio, lo que desencadena el registro del programa. Un nombre de registro específico `ProgramRegistry` se rellena con entradas de programa, y cada programa se registra con sus identificadores, una entrada por cada identificador de programa conocido, lo que significa que si un programa es conocido por varios identificadores diferentes, el registro contiene tantas entradas como identificadores.

El registro de un programa determinado se basa en la colección de identificadores devueltos por el método `()`: `getProgramIdentifiers`

```

Cbact04c.java ×
1 package aws.bluage.13.workshop.program;
2
3 import aws.bluage.13.workshop.SpringBootLauncher;
24
25 /**
26  * Reference the spring application of program CBACT04C.
27  * Provides an access to the contained program for the run unit.
28  */
29 @Component
30 @Import({
31     aws.bluage.13.workshop.cbact04c.business.context.Cbact04cConfiguration.class,
32     aws.bluage.13.workshop.cbact04c.business.context.Cbact04cContext.class,
33     aws.bluage.13.workshop.cbact04c.service.impl.Cbact04cProcessImpl.class
34 })
35 public class Cbact04c implements Program {
36     /**
37      * Unique identifiers for the contained program.
38      */
39     private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("CBACT04C").collect(Collectors.toSet()));
40
41     /**
42      * Main program identifier for the contained program.
43      */
44     private static final String programIdentifier = "CBACT04C";
45     @Autowired
46     PlatformTransactionManager transactionManager;
47
48     @Autowired
49     Map<String, DataSource> datasources;
50     @Autowired
51     BeanFactory beanFactory;
52     /**
53      * {@inheritDoc}
54      */
55     @Override
56     public ConfigurableApplicationContext getSpringApplication() {
57         return SpringBootLauncher.getCac();
58     }
59
60     /**
61      * {@inheritDoc}
62      */
63     @Override
64     public void updateExecutionContext(ExecutionContext executionContext) {
65         executionContext.setDatasources(datasources);
66         executionContext.setDatabaseSupport(ExecutionContext.DatabaseSupport.POSTGRE);
67         executionContext.setSqlcaVersion(ExecutionContext.SqlcaVersion.getEnum("ansi-comp5"));
68         executionContext.setTransactionManager(transactionManager);
69         executionContext.setUseSQLDateNewParadigm(true);
70         executionContext.setUseSQLTrimStringType(false);
71     }
72
73     /**
74      * {@inheritDoc}
75      */
76     @Override
77     public Set<String> getProgramIdentifiers() {
78         return programIdentifiers;
79     }
80

```

En este ejemplo, el programa se registra una vez, con el nombre “CBACT04C” (mire el contenido de la colección de programIdentifiers). Los registros tomcat muestran todos los registros del programa. El registro del programa solo depende de los identificadores del programa declarados y no del nombre de la clase del programa en sí (aunque normalmente los identificadores y los nombres de las clases del programa se corresponden).

El mismo mecanismo de registro se aplica a los programas utilitarios incluidos en las distintas aplicaciones web utilitarias de AWS Blu Age, que forman parte de la distribución en tiempo de ejecución de AWS Blu Age. Por ejemplo, la aplicación web Gapwalk-Utility-Pgm proporciona los

equivalentes funcionales de las utilidades del sistema z/OS (IDCAMS, ICEGENER, SORT, etc.) y puede invocarse mediante programas o scripts modernizados. Todos los programas de utilidades disponibles que se registran al iniciar Tomcat se registran en los registros de Tomcat.

## Registro de scripts y daemons

En el momento de iniciar Tomcat, se produce un proceso de registro similar para los scripts groovy que se encuentran en la jerarquía de carpetas `/src/main/resources/scripts`. Se recorre la jerarquía de carpetas de scripts y todos los scripts groovy que se descubren (excepto el script reservado `functions.groovy` especial) se registran en el `ScriptRegistry`, utilizando su nombre abreviado (la parte del nombre del archivo de script que aparece antes del primer punto) como clave para la recuperación.

### Note

- Si varios scripts tienen nombres de archivo que dan como resultado la misma clave de registro, solo se registra la última, lo que sobrescribe cualquier registro que se haya encontrado anteriormente para esa clave determinada.
- Teniendo en cuenta lo anterior, preste atención al utilizar subcarpetas, ya que el mecanismo de registro aplanar la jerarquía y podría provocar sobrescrituras inesperadas. La jerarquía no se tiene en cuenta en el proceso de registro: lo normal es que `/scripts/A/myscript.groovy` y `/scripts/B/myscript.groovy` provoquen que `/scripts/B/myscript.groovy` sobrescriba `/scripts/A/myscript.groovy`.

Los scripts groovy de la carpeta `/src/main/resources/daemons` se gestionan de forma un poco diferente. Se siguen registrando como scripts normales, pero además, se lanzan una vez, directamente al iniciar Tomcat, de forma asíncrona.

Una vez registrados los scripts en el `ScriptRegistry`, se pueden ejecutar mediante una llamada REST, utilizando los puntos de conexión dedicados que expone la aplicación Gapwalk. Para obtener más información, consulte la documentación correspondiente.

## Programas que llaman a programas

Un programa puede llamar a otro programa como subprograma, pasándole parámetros. Para ello, los programas utilizan una implementación de la interfaz de `ExecutionController` (la mayoría de las veces, se trata de una instancia de `ExecutionControllerImpl`), junto con un mecanismo de API fluido llamado por el `CallBuilder` para crear los argumentos de llamada al programa.



Todos los métodos de los programas utilizan `RuntimeContext` y `ExecutionController` como argumentos de método, por lo que `ExecutionController` siempre está disponible para llamar a otros programas.

Véase, por ejemplo, el siguiente diagrama, que muestra cómo el programa CBST03A llama al programa CBST03B como subprograma y le pasa los parámetros:

```

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

```

- El primer argumento de `ExecutionController.callSubProgram` es un identificador del programa al que se va a llamar (es decir, uno de los identificadores utilizados para el registro del programa; consulte los párrafos anteriores).
- El segundo argumento, que es el resultado de la creación de `CallBuilder`, es una matriz de `Record`, que corresponde a los datos que se transmiten del programa que hace la llamada al programa que recibe la llamada.
- El tercer y último argumento es la instancia `RuntimeContext` del programa que hace la llamada.

Los tres argumentos son obligatorios y no pueden ser nulos, pero el segundo argumento puede ser una matriz vacía.

El programa que llama solo podrá procesar los parámetros transmitidos si se ha diseñado originalmente para ello. En el caso de un programa COBOL antiguo, esto significa disponer de una

sección LINKAGE y una cláusula USING para que la división de procedimientos pueda utilizar los elementos de LINKAGE.

Por ejemplo, consulte el archivo de origen de COBOL [CBSTM03B.CBL](https://github.com/aws-samples/aws-mainframe-modernization-carddemo/blob/main/app/cbl/CBSTM03B.CBL) correspondiente:

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

```

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

```

Por lo tanto, el programa CBSTM03B toma un solo Record como parámetro (una matriz de tamaño 1). Esto es lo que CallBuilder está creando mediante el encadenamiento de los métodos byReference() y getArguments().

La clase de API fluida CallBuilder tiene varios métodos disponibles para rellenar la matriz de argumentos y pasarlos al programa que hace la llamada:

- asPointer (RecordAdaptable): agrega un argumento tipo puntero, como referencia. El puntero representa la dirección de una estructura de datos de destino.
- byReference (RecordAdaptable): añade un argumento por referencia. El programa que hace la llamada verá las modificaciones que realiza el programa que recibe la llamada.
- byReference (RecordAdaptable): variante varargs del método anterior.
- byValue(Object): agrega un argumento, transformado en Record, por valor. El programa que hace la llamada no verá las modificaciones que realiza el programa que recibe la llamada.

- `byValue (RecordAdaptable)`: igual que el método anterior, pero el argumento está disponible directamente como `RecordAdaptable`
- `by ValueWith Bounds (Object, int, int)`: agrega un argumento, transformado en `aRecord`, extrayendo la parte de la matriz de bytes definida por los límites dados, por valor.

Por último, el método `getArguments` recopilará todos los argumentos agregados y los devolverá como una matriz de `Record`.

### Note

Es responsabilidad del programa que hace la llamada asegurarse de que la matriz de argumentos tenga el tamaño requerido, que los elementos estén ordenados correctamente y que sean compatibles, en términos de diseño de memoria, con los diseños esperados para los elementos de enlace.

## Scripts que llaman a programas

Para llamar a programas registrados desde scripts groovy, es necesario usar una instancia de clase que implemente la interfaz de `MainProgramRunner`. Por lo general, obtener una instancia de este tipo se logra mediante el uso de Spring: `ApplicationContext`

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

Cuando haya una interfaz de `MainProgramRunner` disponible, utilice el método `runProgram` para llamar a un programa y pasar el identificador del programa de destino como parámetro:

```

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

```

En el ejemplo anterior, un paso de trabajo llama a IDCAMS (programa de utilidades de gestión de archivos) y asigna las definiciones reales de los conjuntos de datos a sus identificadores lógicos.

Cuando se trata de conjuntos de datos, los programas antiguos utilizan principalmente nombres lógicos para identificar los conjuntos de datos. Cuando se llama al programa desde un script, el script debe hacer corresponder los nombres lógicos con los conjuntos de datos físicos reales. Estos conjuntos de datos pueden estar en el sistema de archivos, en un almacenamiento de Blusam o incluso estar definidos por una secuencia en línea, la concatenación de varios conjuntos de datos o la generación de un GDG.

Utilice el `withFileConfiguration` método para crear un mapa lógico a físico de conjuntos de datos y ponerlo a disposición del programa llamado.

## Escriba su propio programa

Escribir un programa propio para ejecutar scripts u otros programas modernizados es una tarea habitual. Por lo general, en los proyectos de modernización, se escriben programas propios cuando un programa ejecutable antiguo está escrito en un lenguaje que el proceso de modernización no admite, o cuando los orígenes se han perdido (sí, eso puede suceder), o cuando el programa es una utilidad cuyos orígenes no están disponibles.

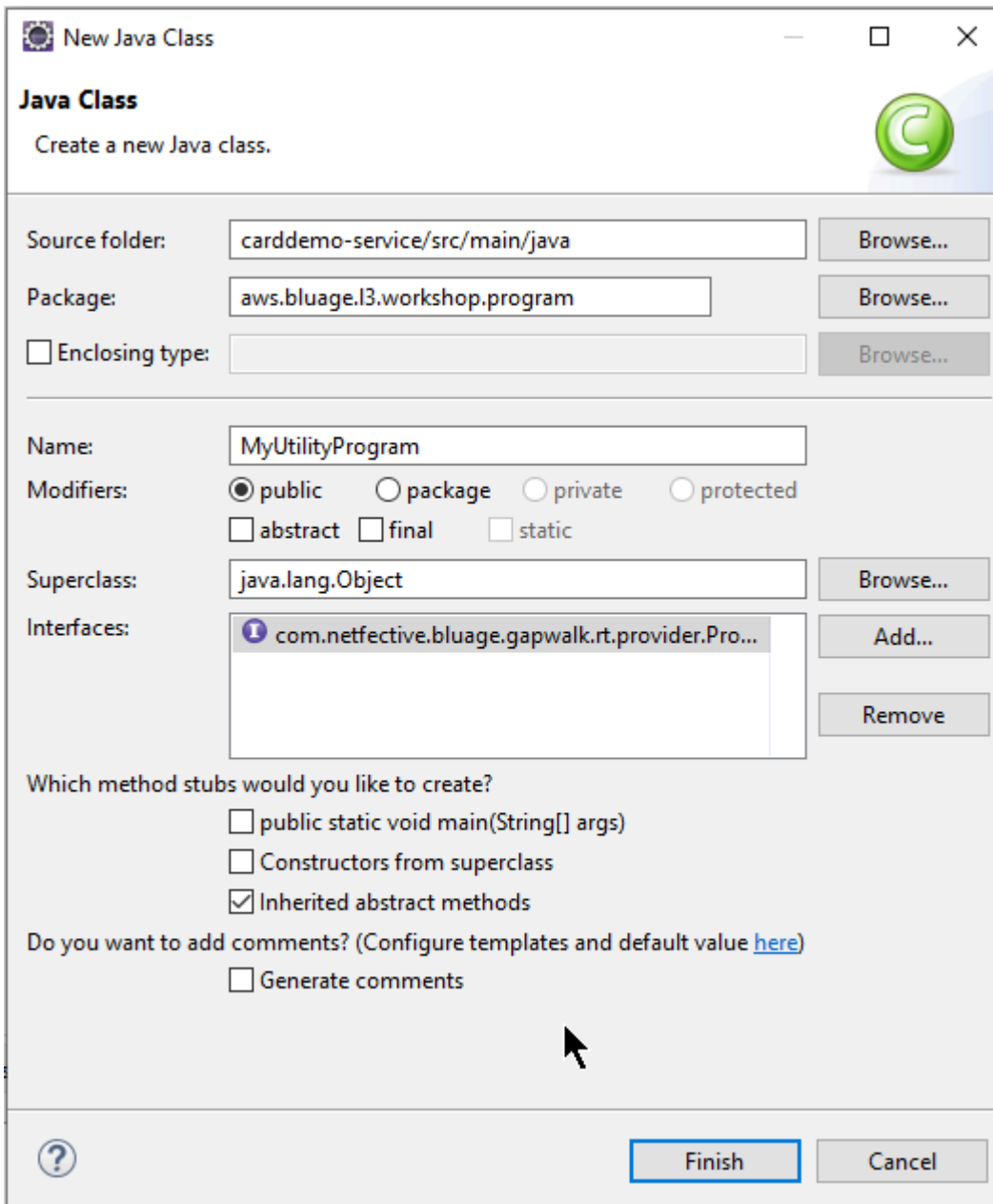
En ese caso, puede que tenga que escribir el programa que falta, en Java, usted mismo (suponiendo que tenga suficiente conocimiento sobre cuál debería ser el comportamiento esperado del programa, el diseño de memoria de los argumentos del programa, si los hubiera, etc.) Su programa Java debe cumplir con la mecánica del programa descrita en este documento, de modo que otros programas y scripts puedan ejecutarlo.

Para asegurarse de que el programa se puede utilizar, debe completar dos pasos obligatorios:

- Escriba una clase que implemente la interfaz de `Program` correctamente, de modo que se pueda registrar y llamar a ella.
- Asegúrese de que el programa esté registrado correctamente, de modo que sea visible desde otros programas/scripts.

### Escribir la implementación del programa

Use su IDE para crear una nueva clase Java que implemente la interfaz de `Program`:



La siguiente imagen muestra el IDE de Eclipse, que se encarga de crear todos los métodos obligatorios que deben implementarse:

```

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

```

## Integración con Spring

En primer lugar, la clase debe declararse como un componente de Spring. Anote la clase con la anotación `@Component`:

```

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 {

```

A continuación, implemente los métodos requeridos correctamente. En el contexto de este ejemplo, añadimos el paquete `MyUtilityProgram` que ya contiene todos los programas modernizados. Esa

ubicación permite al programa usar la aplicación Springboot existente para proporcionar lo necesario `ConfigurableApplicationContext` para la implementación del `getSpringApplication` método:

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

Puede elegir una ubicación diferente para su programa. Por ejemplo, puede ubicar el programa en otro proyecto de servicio dedicado. Asegúrese de que el proyecto de servicio en cuestión tenga su propia aplicación Springboot, que permita recuperar el `ApplicationContext` (que debería ser `unConfigurableApplicationContext`).

### Dar una identidad al programa

Para que otros programas y scripts puedan invocarlo, el programa debe tener al menos un identificador, que no debe colisionar con ningún otro programa registrado existente en el sistema. La elección del identificador puede estar motivada por la necesidad de sustituir un programa antiguo existente; en ese caso, tendrás que usar el identificador esperado, tal y como se encuentra en las apariciones de CALL que se encuentran en los programas antiguos. La mayoría de los identificadores del programa tienen una longitud de 8 caracteres en los sistemas antiguos.

Crear un conjunto de identificadores no modificables en el programa es una forma de hacerlo. En el siguiente ejemplo, se muestra la elección de "MYUTILPG" como identificador único:

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

    public ConfigurableApplicationContext getSpringApplication() {
        I

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

### Asocie el programa a un contexto

El programa necesita una instancia de `RuntimeContext` complementaria. En el caso de los programas modernizados, AWS Blu Age genera automáticamente el contexto complementario, utilizando las estructuras de datos que formaban parte del programa anterior.

Si está escribiendo su propio programa, también debe escribir el contexto complementario.



Consulte [Clases relacionadas con el programa](#) para ver que un programa requiere al menos dos clases complementarias:

- una clase de configuración
- una clase de contexto que usa la configuración

Si el programa de utilidades usa alguna estructura de datos adicional, también debería escribirse y usarse según el contexto.

Esas clases deben estar en un paquete que forme parte de una jerarquía de paquetes que se escaneará al iniciar la aplicación, para garantizar que el marco de Spring gestione el componente de contexto y la configuración.

A continuación, escribamos una configuración y un contexto mínimos en el paquete de *base package*.myutilityprogram.business.context recién creado en el proyecto de entidades:

```
▼ aws.bluage.l3.workshop.csutldtc.business.model
  > FeedbackCode.java
  > LsDate.java
  > LsDateFormat.java
  > LsResult.java
  > OutputLillian.java
  > WsDateFormat.java
  > WsDateToTest.java
  > WsMessage.java
▼ aws.bluage.l3.workshop.myutilityprogram.business.context
  > MyUtilityProgramConfiguration.java
  > MyUtilityProgramContext.java
```

Este es el contenido de la configuración: Utiliza una compilación de configuración similar a la de otros programas (modernizados) cercanos. Probablemente tengas que personalizarlo para que se adapte a tus necesidades específicas.

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

## Notas:

- La convención de nomenclatura general es ProgramNameConfiguración.
- Debe usar las anotaciones `@org.springframework.context.annotation.Configuration` y `@Lazy`
- El nombre del bean suele seguir la ProgramNameContextConfiguration convención, pero no es obligatorio. Asegúrese de evitar conflictos de nombres de beans en todo el proyecto.
- El único método a implementar debe devolver un objeto `Configuration`. Use la API fluida de `ConfigurationBuilder` como ayuda para crear una.

Y el contexto asociado:

```

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

```

## Notas

- La clase de contexto debe ampliar una implementación de interfaz de Context existente (RuntimeContext o JicsRuntimeContext, que es una RuntimeContext mejorada con elementos específicos de JICS).
- La convención general de nomenclatura es ProgramNameel contexto.
- Debe declararlo como un componente de Prototype y utilizar la anotación @Lazy.
- El constructor hace referencia a la configuración asociada y utiliza la anotación @Qualifier para dirigirse a la clase de configuración adecuada.
- Si el programa de utilidades utiliza algunas estructuras de datos adicionales, deberían ser:
  - escritas y añadidas al paquete `base package.business.model`
  - referenciadas en el contexto. Eche un vistazo a otras clases de contexto existentes para ver cómo hacer referencia a las clases de estructuras de datos y adaptar los métodos de contexto (constructor, limpieza o restablecimiento) según sea necesario.

Ahora que hay un contexto específico disponible, deje que el nuevo programa lo use:

```

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

```

## Notas:

- El método `getContext` debe implementarse estrictamente como se muestra, utilizando una delegación al `getOrCreate` método de la `ProgramContextStore` clase y al Spring `BeanFactory` cableado automáticamente. Se utiliza un único identificador de programa para almacenar el contexto del programa en el `ProgramContextStore`; este identificador se denomina “identificador principal del programa”.
- Se debe hacer referencia a las clases complementarias de configuración y contexto mediante la anotación `@Import` de Spring.

## Implementación de la lógica empresarial

Cuando el esquema del programa esté completo, implemente la lógica empresarial para el nuevo programa de utilidades.

Hágalo según el método `run` del programa. Este método se ejecutará cada vez que se llame al programa, ya sea mediante otro programa o mediante un script.

¡Feliz programación!

## Gestionar el registro del programa

Por último, asegúrese de que el nuevo programa esté registrado correctamente en el `ProgramRegistry`. Si ha agregado el nuevo programa al paquete que ya contiene otros programas, no hay nada más que hacer. El nuevo programa se selecciona y se registra en todos sus programas vecinos al iniciar la aplicación.

Si elige otra ubicación para el programa, debe asegurarse de que el programa esté registrado correctamente al iniciar Tomcat. Para inspirarse sobre cómo hacerlo, consulte el método de inicialización de las `SpringbootLauncher` clases generadas en los proyectos de servicio (consulte [Contenido del proyecto de servicio](#)).

Compruebe los registros de inicio de Tomcat. Se registran todos los registros de programas. Si el programa se ha registrado correctamente, encontrará la entrada de registro correspondiente.

Cuando esté seguro de que su programa está registrado correctamente, puede empezar a iterar con la codificación de la lógica empresarial.

## Asignaciones de nombres totalmente calificadas

Esta sección contiene listas de la era AWS azul y de mapeos de nombres totalmente calificados de terceros para utilizarlos en sus aplicaciones modernizadas.

### AWS Asignaciones de nombres totalmente calificadas de Blu Age

Nombre corto	Nombre completo calificado
<code>CallBuilder</code>	<code>com.netfective.bluage.gapwalk.runtime.statements.CallBuilder</code>
<code>Configuration</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration</code>

Nombre corto	Nombre completo cualificado
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

Nombre corto	Nombre completo cualificado
SimpleStateMachineController	com.netfective.bluage.gapwalk.rt.statemachine.SimpleStateMachineController
StateMachineController	com.netfective.bluage.gapwalk.rt.statemachine.StateMachineController
StateMachineRunner	com.netfective.bluage.gapwalk.rt.statemachine.StateMachineRunner

### Asignaciones de nombres totalmente cualificadas de terceros

Nombre corto	Nombre completo cualificado
@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

## Data Simplifier

En los sistemas de unidad central y de gama media (denominados en el siguiente tema sistemas “heredados”), los lenguajes de programación de uso frecuente, como COBOL, PL/I o RPG, proporcionan un acceso de bajo nivel a la memoria. Este acceso se centra en el diseño de la memoria, al que se accede mediante tipos nativos, como zonificados, empaquetados o alfanuméricos, y es posible que también se agreguen mediante grupos o matrices.

En un programa determinado coexiste una combinación de accesos a un determinado fragmento de memoria, a través de campos de tipos definidos y acceso directo a bytes (memoria sin procesar). Por ejemplo, los programas COBOL pasarán los argumentos a las personas que llaman como conjuntos de bytes contiguos (LINKAGE) o leerán y escribirán datos de los archivos de la misma manera (registros), al tiempo que interpretarán dichos rangos de memoria con campos mecanografiados organizados en cuadernos.

Estas combinaciones de acceso sin proceso y estructurado a la memoria, la dependencia de un diseño de memoria preciso a nivel de bytes y los tipos heredados, como zonificados o empaquetados, son características que no están disponibles de forma nativa ni son de fácil acceso en el entorno de programación Java.

Como parte de la solución AWS Blu Age para modernizar los programas heredados a Java, la biblioteca Data Simplifier proporciona estas construcciones a los programas Java modernizados y las expone de una manera que resulta lo más familiar posible para los desarrolladores de Java (captres/ setters, matrices de bytes, basadas en clases). Se trata de una dependencia fundamental para el código Java modernizado generado a partir de esos programas.

Para simplificar, la mayoría de las siguientes explicaciones se basan en constructos de COBOL, pero puede utilizar la misma API para la modernización del diseño de datos de PL1 y RPG, ya que la mayoría de los conceptos son similares.

### Temas

- [Clases principales](#)
- [Enlace y acceso a los datos](#)
- [Nombres completos calificados de los tipos de Java descritos](#)



## Clases principales

Para facilitar la lectura, en este documento se utilizan los nombres abreviados en Java de las interfaces y clases de la API de Blu Age. AWS Para obtener más información, consulte [Nombres completos calificados de los tipos de Java descritos](#).

### Representación de la memoria de bajo nivel

En el nivel más bajo, la memoria (un rango contiguo de bytes al que se puede acceder de forma rápida y aleatoria) está representada por la interfaz de `Record`. Esta interfaz es esencialmente una abstracción de una matriz de bytes de un tamaño fijo. Como tal, proporciona setters y getters capaces de acceder o modificar los bytes subyacentes.

### Representación de datos estructurados

Para representar datos estructurados, como “01 elementos de datos” o “01 cuadernos”, como se encuentran en COBOL DATA DIVISION, se utilizan subclases de la clase `RecordEntity`. Por lo general, no se escriben a mano, sino que se generan con las herramientas de modernización de AWS Blu Age a partir de las correspondientes construcciones heredadas. Sigue siendo útil conocer su estructura principal y su API, para poder entender cómo los usa el código de un programa modernizado. En el caso de COBOL, ese código es Java generado a partir de su PROCEDURE DIVISION.

El código generado representa cada “01 elemento de datos” con una subclase `RecordEntity`; cada campo elemental o agregado que lo compone se representa como un campo Java privado, organizado en forma de árbol (cada elemento tiene un elemento principal, excepto el raíz).

A modo ilustrativo, he aquí un ejemplo de elemento de datos de COBOL, seguido del correspondiente código generado por AWS Blu Age que lo moderniza:

```
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);
    }
}
```

## Campos elementales

Los campos de clase `Elementary` (o `Filler`, cuando no tienen nombre) representan una “hoja” de la estructura de datos heredada. Están asociados a un espacio contiguo de bytes subyacentes (“rango”) y, por lo general, tienen un tipo (posiblemente parametrizado) que expresa cómo interpretar y modificar esos bytes (mediante la “decodificación” y la “codificación”, respectivamente, de un valor desde/hacia una matriz de bytes).

Todos los tipos elementales son subclases de `RangeType`. Los tipos frecuentes son:

Tipo COBOL	Tipo de Data Simplifier
PIC X(n)	AlphanumericType
PIC 9(n)	ZonedType
PIC 9(n) COMP-3	PackedType
PIC 9(n) COMP-5	BinaryType

## Campos agregados

Los campos agregados organizan el diseño de memoria de sus contenidos (otros agregados o campos elementales). No tienen un tipo elemental en sí mismos.

Los campos Group representan campos contiguos en la memoria. Cada uno de sus campos contenidos está dispuesto en el mismo orden en la memoria, el primer campo tiene un desplazamiento 0 con respecto a la posición del campo de grupo en la memoria, el segundo campo tiene un desplazamiento 0 + (size in bytes of first field), etc. Se utilizan para representar secuencias de campos COBOL en el mismo campo contenedor.

Los campos Union representan varios campos que acceden a la misma memoria. Cada uno de sus campos contenidos está dispuesto con un desplazamiento 0 con respecto a la posición del campo de unión en la memoria. Por ejemplo, se utilizan para representar el constructo COBOL "REDEFINE" (el primer elemento secundario de Union es el elemento de datos redefinido, el segundo elemento secundario es su primera redefinición, etc.).

Los campos matriciales (subclases de Repetition) representan la repetición, en la memoria, del diseño de su campo secundario (ya sea un agregado en sí mismo o un elemento elemental). Diseñan un número determinado de estos diseños secundarios en la memoria, cada uno de los cuales tiene un desplazamiento  $index * (size\ in\ bytes\ of\ child)$ . Se utilizan para representar los constructos "OCCURS" de COBOL.

## Elementos primitivos

En algunos casos de modernización, las «primitivas» también se pueden utilizar para presentar elementos de datos «raíz» independientes. Su uso es muy similar `RecordEntity`, pero no provienen de él, ni se basan en el código generado. En cambio, el motor de ejecución de AWS Blu Age los

proporciona directamente como subclases de la `Primitive` interfaz. Algunos ejemplos de estas clases proporcionadas son `Alphanumeric` o `ZonedDecimal`.

## Enlace y acceso a los datos

La asociación entre los datos estructurados y los datos subyacentes se puede realizar de varias maneras.

Una interfaz importante para este propósito es `RecordAdaptable`, la cual se utiliza para obtener un `Record` que ofrece una “vista modificable” de los datos subyacentes de `RecordAdaptable`. Como veremos a continuación, varias clases implementan `RecordAdaptable`. Recíprocamente, las API de AWS Blu Age y los códigos que manipulan la memoria de bajo nivel (como los argumentos de los programas, los registros de E/S de los archivos, el área de comunicación CICS, la memoria asignada, etc.) suelen necesitar a `RecordAdaptable` como identificador de esa memoria.

En el caso de la modernización de COBOL, la mayoría de los elementos de datos están asociados a una memoria que permanecerá fija durante el tiempo de ejecución del programa correspondiente. Para ello, `RecordEntity` las subclases se instancian una vez en un objeto principal generado (el programa `Context`) y se encargan de instanciar su subyacente, en función del tamaño del byte. `RecordEntity`

En otros casos de COBOL, como cuando se asocian elementos `LINKAGE` a los argumentos del programa o se moderniza el constructo `SET ADDRESS OF`, una instancia `RecordEntity` debe estar asociada a un `RecordAdaptable` proporcionado. Para ello, existen dos mecanismos:

- Si la instancia de `RecordEntity` ya existe, se puede usar el método `RecordEntity.bind(RecordAdaptable)` (heredado de `Bindable`) para hacer que esta instancia “apunte” a este `RecordAdaptable`. Cualquier getter o setter al que se invoque en `RecordEntity` estará respaldado (lectura o escritura de bytes) por los bytes del `RecordAdaptable` subyacente.
- Si se va a crear una instancia de `RecordEntity`, hay disponible un constructor generado que acepte a `RecordAdaptable`.

Por el contrario, se puede acceder al `Record` actualmente vinculado a los datos estructurados. Para ello, `RecordEntity` implementa `RecordAdaptable`, por lo que se `getRecord()` puede invocar en cualquier instancia de este tipo.

Por último, muchos verbos de COBOL o CICS requieren acceso a un único campo para poder leerlos o escribirlos. La clase `RangeReference` se usa para representar dicho acceso. Sus

instancias se pueden obtener a partir de los métodos `getXXXReference()` generados para `RecordEntity` (XXX es el campo al que se accede) y pasarlas a los métodos de tiempo de ejecución. `RangeReference` se utiliza normalmente para acceder a campos enteros `RecordEntity` o `Group`, mientras que su subclase `ElementaryRangeReference` representa los accesos a los campos `Elementary`.

Tenga en cuenta que la mayoría de las observaciones anteriores se aplican a `Primitive` las subclases, ya que se esfuerzan por implementar un comportamiento similar al que proporciona el `RecordEntity` tiempo de ejecución de AWS Blu Age (en lugar del código generado). Con este propósito, todas las subclases de `Primitive` implementan las interfaces `RecordAdaptable`, `ElementaryRangeReference` y `Bindable` de modo que se puedan utilizar en lugar de las subclases `RecordEntity` y los campos elementales.

## Nombres completos calificados de los tipos de Java descritos

La siguiente tabla muestra los nombres completos calificados de los tipos de Java analizados en esta sección.

Nombre corto	Nombre completo calificado
<code>Alphanumeric</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.Alphanumeric</code>
<code>AlphanumericType</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType</code>
<code>BinaryType</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.BinaryType</code>
<code>Bindable</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.data.Bindable</code>
<code>Elementary</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary</code>

Nombre corto	Nombre completo calificado
ElementaryRangeReference	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference</code>
Filler	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Filler</code>
Group	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group</code>
PackedType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.PackedType</code>
Primitive	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.Primitive</code>
RangeReference	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference</code>
RangeType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.RangeType</code>
Record	<code>com.netfective.bluage.gapwalk.datasimplifier.data.Record</code>
RecordAdaptable	<code>com.netfective.bluage.gapwalk.datasimplifier.data.RecordAdaptable</code>

Nombre corto	Nombre completo calificado
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 Archivos de configuración y configuración de Blu Age Runtime

El tiempo de ejecución de AWS Blu Age y el código del cliente son aplicaciones web que utilizan el [marco Spring Boot](#). Aproveche las capacidades de Spring para proporcionar la configuración, con varias ubicaciones posibles y reglas de prioridad. También existen reglas de prioridad similares para el suministro de muchos otros archivos, como scripts groovy, sql, etc.

El AWS Blu Age Runtime también contiene aplicaciones web opcionales adicionales, que se pueden activar si es necesario.

### Temas

- [Principios básicos de la configuración de aplicaciones](#)
- [Prioridad de la aplicación](#)
- [JNDI para bases de datos](#)



- [Uso de secretos AWS](#)
- [Otros archivos \(groovy, sql, etc.\)](#)
- [Aplicación web adicional](#)
- [Habilitación de propiedades](#)
- [Configuración de la seguridad para las aplicaciones de Gapwalk](#)

## Principios básicos de la configuración de aplicaciones

La forma predeterminada de gestionar la configuración de la aplicación es mediante el uso de archivos YAML dedicados que se proporcionan en la carpeta del servidor de aplicaciones. `config` Hay dos archivos principales de configuración de YAML:

- `application-main.yaml`
- `application-profile.yaml` (donde el valor *profile* se configura durante la generación de la aplicación).

El primer archivo configura el marco, es decir `Gapwalk-application.war`, mientras que el segundo es para opciones adicionales específicas para la aplicación cliente. Esto funciona con el uso de perfiles de Spring: la aplicación de Gapwalk usa el perfil `main`, mientras que la aplicación cliente usa el perfil *profile*.

A continuación se muestra un ejemplo de archivo YAML principal típico.

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

A continuación se muestra un ejemplo de archivo YAML de cliente típico.

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

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

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

Para obtener más información sobre el contenido de los archivos YAML, consulte [Habilitación de propiedades](#).

## Prioridad de la aplicación

Para estos archivos de configuración, se aplican las reglas de prioridad de Spring. En particular:

- El archivo `application-main` YAML aparece en el archivo de guerra principal de Gapwalk con los valores predeterminados y el archivo de la carpeta lo reemplaza. `config`
- Lo mismo debería hacerse con la configuración de la aplicación cliente
- Es posible que se transfieran parámetros adicionales a través de la línea de comandos en el momento del lanzamiento del servidor. Anularían los de YAML.

Para obtener más información, consulte la [documentación oficial de Spring Boot](#).

## JNDI para bases de datos

La configuración de la base de datos puede proporcionarse con JNDI en el archivo `context.xml` de Tomcat. Cualquier configuración de este tipo anularía la de YAML. Pero tenga en cuenta que, si usa eso, no podrá guardar sus credenciales en un administrador secreto (consultar a continuación).

El siguiente ejemplo muestra ejemplos de configuraciones para JICS y bases de datos. BluSam

```
<Resource auth="Container" driverClassName="org.postgresql.Driver" initialSize="0"
  maxIdle="5"
  maxOpenPreparedStatements="-1" maxTotal="10" maxWaitMillis="-1" name="jdbc/jics"
  poolPreparedStatements="true" testOnBorrow="false" type="javax.sql.DataSource"
  url="jdbc:postgresql://XXXX.rds.amazonaws.com:5432/XXXX" username="XXXX"
  password="XXXX" />
```

`jdbc/jics`

Sería `jdbc/jics` para la base de datos JICS y `jdbc/bluesam` (preste atención a la «e») para la base de datos bluesam.

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

URL, nombre de usuario y contraseña de la base de datos.

## Uso de secretos AWS

Algunas de las configuraciones de recursos que contienen credenciales se pueden proteger aún más mediante el uso de secretos de AWS . La idea es almacenar los datos críticos en un lugar AWS

secreto y tener una referencia al secreto en la configuración de YAML para que el contenido secreto se recoja sobre la marcha al iniciar Tomcat.

## Secretos de Aurora

La configuración de la base de datos Aurora (para jics, blusam, base de datos de cliente, etc.) utilizará el [secreto de base de datos](#) integrado, que rellenará automáticamente todos los campos relevantes de la base de datos correspondiente.

### Note

La clave dbname es opcional y, según la configuración de la base de datos, se incluirá en el secreto o no. Puede añadirlos allí manualmente o proporcionando el nombre del archivo YAML.

## Otros secretos

Hay otros secretos para los recursos que tienen una sola contraseña (en particular, las cachés de Redis protegidas con contraseña). En este caso, se debe usar el valor de [Otro tipo de secreto](#), con una sola clave. password

## Referencias en YAML a secretos

application-main.yaml Pueden hacer referencia al ARN secreto para varios recursos. Los más importantes son:

- Credenciales de la base de datos JICS con `spring.aws.jics.db.secret`
- JICS TS pone en cola las credenciales de Redis con `spring.aws.client.jics.queues.ts.redis.secret`
- Las credenciales de la base de datos de Blusam con `spring.aws.client.bluesam.db.secret`
- La contraseña de caché de Blusam con `spring.aws.client.bluesam.redis.secret`
- La contraseña de caché de bloqueos de Blusam con `spring.aws.client.bluesam.locks.redis.secret`

En el siguiente ejemplo, se muestra cómo declarar estos secretos en un archivo YAML.

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

dbname: bluesam

En este ejemplo, el nombre de la base de datos no está en el secreto, sino que se proporciona aquí.

El `application-profile.yaml` del cliente puede hacer referencia al ARN secreto de la base de datos del cliente. Esto requiere una propiedad adicional para enumerar los orígenes de datos, como se muestra en el siguiente ejemplo:

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

names: primary,host

Un ejemplo con dos orígenes de datos de clientes denominados principal y host, cada uno con su base de datos y sus credenciales.

dbname: mydb

En este ejemplo, el nombre de la base de datos “host” no está en el secreto, sino que se proporciona aquí, mientras que la base de datos “primary” sí está en el secreto.

También puede configurar `application-main.yaml` para recuperar el secreto del cliente OAuth2 AWS Secrets Manager especificando el proveedor y el ARN. El valor predeterminado de la propiedad del proveedor es Amazon Cognito. El siguiente es un ejemplo de configuración para el proveedor de OAuth2 Keycloak:

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

En este ejemplo, el secreto del cliente del proveedor de OAuth2 Keycloak se recupera del ARN especificado en. AWS Secrets Manager Esta configuración admite varios proveedores al resolver dinámicamente el nombre del proveedor y el ARN secreto correspondiente.

`application-utility-pgm.yml` Pueden hacer referencia al ARN secreto para varios recursos.

- `spring.aws.client.datasources.primary`
  - `secretARN` secreto para la base de datos de la aplicación.

Tipo: cadena

- `type`

Nombre completo de la implementación del grupo de conexiones que se va a utilizar.

Tipo: cadena

Valor predeterminado: `com.zaxxer.hikari.HikariDataSource`

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

Lista de nombres de fuentes de datos.

Tipo: cadena

- `dsname`
  - `dbname`

Nombre del host.

Tipo: cadena

- `secret`

ARN secreto de la base de datos host.

Tipo: cadena

- `type`

Nombre completo de la implementación del grupo de conexiones que se va a utilizar.

Tipo: cadena

Valor predeterminado: `com.zaxxer.hikari.HikariDataSource`

Para un secreto de varias fuentes de datos:

```
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
```

No hay claves secretas compatibles con XA

- motor (postgres/oracle/db2/mssql)
- port
- dbname
- Esquema actual
- username
- password
- url
- Conexión SSL
- Ubicación ssl TrustStore
- TrustStoreContraseña ssl

postgresSolo para la clave sslMode secreta valorada (disable/allow/prefer/require/verify-ca/verify-full), además de la propiedad `spring.aws.rds.ssl.cert-path` YAML, se puede conectar con SSL.



## XA admitía claves secretas

Si la base de datos del cliente utiliza XA, las propiedades subxa se admiten mediante valores secretos.

- host
- port
- dbname
- Esquema actual
- username
- password
- url
- Conexión SSL (verdadero/falso)
- TrustStoreUbicación ssl
- TrustStoreContraseña ssl

Sin embargo, para otras propiedades xa (por ejemplo, `maxPoolSize` o `driverType`), se `spring.jta.atomikos.datasource.XXXX.unique-resource-name` debe seguir proporcionando la clave YAML normal.

El valor secreto anula las propiedades de YAML.

## Otros archivos (groovy, sql, etc.)

Los demás archivos utilizados por el proyecto del cliente utilizan reglas de prioridad similares a las de la configuración de Spring. Ejemplos:

- Los scripts groovy son los archivos `.groovy` de la carpeta o las subcarpetas `scripts`.
- Los scripts SQL son los archivos `.sql` de la carpeta o las subcarpetas `sql`.
- Los scripts daemon son los archivos `.groovy` de la carpeta o las subcarpetas `daemons`.
- Los archivos de asignación de base de datos de consultas se llaman `queries-database.mapping` y se encuentran en las subcarpetas de la carpeta `sql`.
- Las plantillas de Jasper son los archivos `.jrxml` de la carpeta o las subcarpetas `templates`.
- Los catálogos de conjuntos de datos son los archivos `.json` de la carpeta `catalog`.
- Los archivos Lnk son los archivos `.json` de la carpeta `lnk`.

Todas estas ubicaciones se pueden anular mediante una propiedad del sistema o una propiedad YAML del cliente.

- Para los scripts de Groovy: `configuration.scripts`
- Para los scripts de SQL: `configuration.sql`
- Para los scripts de Daemon: `configuration.daemons`
- Para el archivo de asignación de bases de datos de consultas: `configuration.databaseMapping`
- Para las plantillas de Jasper: `configuration.templates`
- Para los catálogos de conjuntos de datos: `configuration.catalog`
- Para los archivos Lnk: `configuration.lnk`

Si no se encuentra la propiedad, los archivos se tomarán de la ubicación predeterminada mencionada anteriormente. La búsqueda se realizará primero con el directorio de trabajo de Tomcat como raíz y, por último, en el archivo war de la aplicación.

## Aplicación web adicional

El AWS Blu Age Runtime contiene aplicaciones web adicionales en su `webapps-extra` carpeta. El servidor Tomcat no sirve estas aplicaciones de forma predeterminada.

La aceptación de estas aplicaciones web depende del proyecto de modernización y se realiza moviendo el archivo war deseado de la carpeta `webapps-extra` a la carpeta `webapps`. Después de eso, el servidor Tomcat servirá el archivo war la próxima vez que se inicie.

También se pueden añadir algunas configuraciones adicionales específicas del proyecto en un archivo de configuración YAML para cada guerra adicional, tal y como se ha hecho en el `application-main.yml` archivo y se ha explicado anteriormente. Los archivos war adicionales son:

- `gapwalk-utility-pgm.war`: contiene soporte para los programas de utilidades de ZOS y utiliza `application-utility-pgm.yaml` como configuración.
- `gapwalk-cl-command.war`: contiene soporte para los programas de utilidades de AS/400 y utiliza `application-cl-command.yaml` como configuración.
- `gapwalk-hierarchical-support.war`: contiene soporte para transacciones IMS/MFS y utiliza `application-jhdb.yaml` como configuración.

## Habilitación de propiedades

En las aplicaciones Spring Boot, `application-main.yml` es el archivo de configuración en el que definimos diferentes tipos de propiedades, como el puerto de escucha, la conectividad de la base de datos y muchas más.

### Temas

- [Notación YML](#)
- [Inicio rápido y casos de uso](#)
- [Propiedades disponibles para la aplicación principal](#)
- [Propiedades disponibles para las aplicaciones web opcionales](#)

### Notación YML

En la siguiente documentación, una propiedad como esta `parent.child1.child2=true` está escrita de la siguiente manera en formato YAML.

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

### Inicio rápido y casos de uso

Los siguientes casos de uso muestran ejemplos de las claves y valores aplicables.

- Archivo `application-main.yml` predeterminado

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

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

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

```

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

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

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

#####
##### Level 2 cache #####
#####
#       cache:
#         use_second_level_cache: true
#         use_query_cache: true

```

```

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

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

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

#####
##### Embedded Bluesam configuration #####
#####

```

```

bluesam :
  remote : false
  cache : ehcache
  persistence : pgsql #pgsql, mssql, xodus...
  ehcache:
    resource-pool:
      size: 4GB
  write-behind:
    enabled: true
  pgsql :
    dataSource : bluesamDs

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

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

```

```
#jics.runUnitLauncherPool.validationInterval: 1000

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

```



```
# query.useConcatCondition: true
----
```

- Utilice archivos de longitud variable con los comandos LISTCAT

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

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

- Proporcione un valor indicador de bytes null en la utilidad LOAD/UNLOAD

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

## Propiedades disponibles para la aplicación principal

Esta tabla proporciona una vista exhaustiva de los parámetros clave/valores.

Clave	Tipo	Valor predeterminado	Descripción
<code>logging.config</code>	Ruta	<code>classpath:logback-main.xml</code>	Clave estándar para la referencia al archivo de configuración del logback. También hay disponibles otras claves de registro estándar.
<code>spring.jta.enabled</code>	valor booleano	<code>false</code>	Clave estándar. Si el modo de soporte del origen de datos no es <code>static-xa</code> , la configuración automática de las transacciones JTA de Spring debe estar deshabilitada.
<code>datasource.jicsDs + -driver-class-name + -url + -username + -password + -type</code>	Origen de datos estándar de Spring con subclaves		Contiene la información de conexión de la base de datos de Jics. Como alternativa, se recomienda encarecidamente el uso de secretos de AWS, tal y como se explica en <a href="#">xref:../configuration/configuration.adoc[Configuration]</a> .

Clave	Tipo	Valor predeterminado	Descripción
<code>datasource.bluesamDs</code> <code>+ -driver-class-name + -url</code> <code>+ -username + -password + -type</code>	Origen de datos estándar de Spring con subclaves		Contiene la información de conexión de la base de datos de Blusam. Como alternativa, se recomienda encarecidamente el uso de secretos de AWS, tal y como se explica en <code>xref:../configuration/configuration.adoc[Configuration]</code> .
<code>bluesam.disabled</code>	valor booleano	false	Si se debe deshabilitar Blusam por completo.
<code>bluesam.cache</code>	cadena		Si no se establece, no se utilizará la caché de Blusam. Los valores posibles (implementaciones de caché) son cache y redis.
<code>forcedDate</code>	cadena		Fuerza la fecha a la fecha proporcionada, si la hay.
<code>frozenDate</code>	booleano	true	Especifica si se debe congelar la fecha. Solo se aplica si también <code>forcedDate</code> está establecida.

Clave	Tipo	Valor predeterminado	Descripción
<code>externalSort.threshold</code>	tamaño de datos (ejemplo: 12 MB)		El umbral de clasificación: cuándo cambiar a una ordenación externa (fusión).
<code>jics.parameters.datform</code>	cadena	MMDDYY	El formato de fecha.
<code>jics.initList</code>	cadena		La lista JICS de inicialización, separada por comas. Si está presente, define los nombres de las listas separados por comas para activarlas al iniciar Apache Tomcat entre las listas CICS. Ejemplo de valor: \$UUU,DFH\$IVPL,PEZ1 Esto se extenderá en cascada a los grupos contenidos en esas listas y a sus definiciones de recursos subyacentes, que luego estarán visibles en el entorno de tiempo de ejecución. Vacío de forma predeterminada.

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.parameters.applid</code>	cadena	VELOCITY	Se aplica para identificar la aplicación en el JICS (4 caracteres como mínimo, sin longitud máxima).
<code>jics.parameters.sysid</code>	cadena	CICS	La identificación del sistema (SYSID).
<code>jics.parameters.eibtrmid</code>	cadena	TERM	El identificador del terminal (4 caracteres como máximo, 1 como mínimo).
<code>jics.parameters.userid</code>	cadena		El identificador de usuario (8 caracteres como máximo, sin mínimo). Si no se proporciona ningún valor (en blanco de forma predeterminada), el identificador de sesión HTTP se utiliza como identificador de usuario.
<code>jics.parameters.username</code>	cadena	MYUSERNAME	El nombre de usuario (10 caracteres como máximo, 1 como mínimo).
<code>jics.parameters.netname</code>	cadena	MYNETNAME	El nombre de la red (8 caracteres como máximo, 1 como mínimo).

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.parameters.opid</code>	cadena	XXX	La identificación del operador de 3 caracteres.
<code>jics.parameters.jobname</code>	cadena	MJOBNAME	El nombre del trabajo.
<code>jics.parameters.sysname</code>	cadena	SYSNAME	El nombre del sistema AS400 (sysname).
<code>jics.parameters.cw a.length</code>	number	0	La longitud del área de trabajo común (CWA).
<code>jics.parameters.charset</code>	cadena	CP037	Juego de caracteres utilizado globalmente por JICS.
<code>jics.parameters.tsqimpl</code>	cadena	bluesam	Implementación de la cola de almacenamiento temporal (TSQ) de JICS (los valores permitidos son/bluesam/memory/redis)
<code>jics.queues.ts.redis.hostname</code>	cadena	127.0.0.1	El nombre de host del servidor de caché redis de jics.
<code>jics.queues.ts.redis.port</code>	number	6379	El puerto del servidor de caché redis de jics.

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.queue es.ts.redis.password</code>	cadena	redis	La contraseña del servidor de caché redis de jics.
<code>jics.queue es.ts.redis.username</code>	cadena		El nombre de usuario del servidor de caché redis de jics. El valor predeterminado está en blanco (sin nombre de usuario).
<code>jics.queue es.ts.redis.mode</code>	cadena	individual	El modo de caché de jics. Los valores posibles son <code>standalone</code> o <code>cluster</code> . El valor predeterminado es <code>standalone</code> .
<code>lockTimeout</code>	number	500	El tiempo de espera del bloqueo, en milisegundos.

Clave	Tipo	Valor predeterminado	Descripción
sqlCodePointShift	number		Opcional. El cambio de punto del código sql. Cambia el punto de código de los caracteres de control que podemos encontrar al migrar datos del RDBMS heredado a un RDBMS moderno. Por ejemplo, puede especificar que coincida con un carácter Unicode. 384 \u0180
sqlIntegerOverflowAllowed	valor booleano	false	Especifica si se permite el desbordamiento de enteros de SQL, es decir, si se permite colocar valores más grandes en la variable host.



Clave	Tipo	Valor predeterminado	Descripción
database.cursor.overflow.allowed	booleano	true	Especifica si se debe permitir el desbordamiento del cursor. Configúrelo en true para realizar una siguiente llamada en el cursor, sea cual sea su posición. Configúrelo en false para comprobar si el cursor está en la última posición antes de realizar una siguiente llamada con el cursor. Actívela solo si el cursor es desplazable (sensible o INSENSIBLE).
reportOutputPath	cadena	/reports	La ruta de salida del informe.
spring.session.store-type	cadena	Ninguno	La memoria caché de sesiones para entornos de alta disponibilidad. Los valores posibles son none o redis. El valor predeterminado es none.

Clave	Tipo	Valor predeterminado	Descripción
<code>stopExecutionWhenProgramNotFound</code>	booleano	true	Especifica si se debe detener la ejecución si no se encuentra un programa. Si se establece en true, interrumpe la ejecución si no se encuentra ningún programa.
<code>forceHR</code>	valor booleano	false	Especifica si se debe utilizar SYSPRINT inteligible, ya sea en la consola o en la salida de un archivo.
<code>rollbackOnRTE</code>	valor booleano	false	Especifica si se debe revertir la transacción implícita de la unidad de ejecución en las excepciones de tiempo de ejecución.
<code>sctThreadLimit</code>	long	5	El límite de subprocesos para activar scripts.

Clave	Tipo	Valor predeterminado	Descripción
<code>dataSimplifier.onInvalidNumericData</code>	cadena	<code>reject</code>	Cómo reaccionar al decodificar datos numéricos no válidos. Los valores permitidos son <code>reject</code> / <code>tolerates paces</code> / <code>tolerates paceslowvalues</code> / <code>toleratemoost</code> . El valor predeterminado es <code>reject</code> .
<code>filesDirectory</code>	cadena		El directorio de los archivos de entrada/salida de los lotes.
<code>ims.messages.extendedSize</code>	valor booleano	<code>false</code>	Especifica si se debe establecer el tamaño extendido de los mensajes IMS.
<code>defaultKeepExistingFiles</code>	valor booleano	<code>false</code>	Especifica si se debe establecer el valor anterior predeterminado del conjunto de datos.

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.db.ddlScriptLocation</code>	cadena		La ubicación del script DDL de Jics. Permite iniciar el esquema de la base de datos de Jics mediante un script .sql. Está en blanco por defecto. Por ejemplo, <code>./jics/sql/jics.sql</code> .
<code>jics.db.schemaTestQueryLocation</code>	cadena		Ubicación del archivo sql que debe contener una consulta única que devuelva el número de objetos del esquema jics (si lo hay).
<code>jics.db.dataScriptLocation</code>	cadena		Ubicación del script <code>initJics.sql</code> , preparado por Analyzer a partir del análisis de las exportaciones de CSD desde el ordenador principal.

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.db.dataTestQueryLocation</code>	cadena		Ubicación de un script sql que contiene una única consulta sql que se espera que devuelva un recuento de objetos (por ejemplo: contar el número de registros de la tabla del programa jics). Si el recuento es igual a 0, la base de datos se cargará mediante el <code>jics.db.dataScriptLocation</code> script; de lo contrario, se omitirá la carga de la base de datos.
<code>jics.data.dataJsonInitLocation</code>	cadena		
<code>jics.xagent.timeout</code>	number		
<code>query.useConcatCondition</code>	valor booleano	false	Especifica si la condición clave se crea mediante la concatenación de claves o no.
<code>system.qdecfmt</code>	cadena		

Clave	Tipo	Valor predeterminado	Descripción
<code>dispositi on.checke xistence</code>	valor booleano	false	Especifica si se debe publicar una comprobación de la existencia del archivo para el conjunto de datos con DISP SHR u OLD.
<code>useContro IMVariable</code>	valor booleano	false	Especifica si se debe utilizar la especificación Control-M para el reemplazo de variables.
<code>card.encoding</code>	cadena	CP1145	Codificación de tarjetas: para usar con. useControIMVariable
<code>mapTransf o.prefixes</code>	cadena	<code>&amp;,@,%%</code>	Lista de prefijos que se utilizarán al transformar las variables de ControlM. Cada uno separado por comas.
<code>checkinpu tfilesize</code>	valor booleano	false	Especifica si se debe activar una comprobación si el tamaño del archivo es un múltiplo del tamaño del registro.

Clave	Tipo	Valor predeterminado	Descripción
<code>stepFailWhenAbend</code>	booleano	true	Especifica si se debe generar una curva si un paso falla o completa la ejecución.
<code>bluesam.fileLoading.commitInterval</code>	number	100000	El intervalo de confirmación de Bluesam.
<code>uppercaseUserInput</code>	booleano	true	Especifica si la entrada del usuario debe estar en mayúsculas.
<code>jhdb.lterm</code>	cadena		Le permiten forzar un ID de terminal lógico común en el caso de una emulación de IMS. Si no se establece, se utiliza SessionID.
<code>jhdb.identificationCardData</code>	cadena		Se utiliza para codificar algunos «datos de la tarjeta de identificación del operador» en el campo MID designado por el parámetro CARD. Está en blanco por defecto, sin restricciones de entrada.

Clave	Tipo	Valor predeterminado	Descripción
<code>encoding</code>	cadena	ASCII	La codificación utilizada en los proyectos (no en los archivos groovy). Espera una codificación válida CP1047,IBM930,ASCII,UTF-8.
<code>cl.configuration.context.encoding</code>	cadena	CP297	La codificación de los archivos CL. Espera una codificación válida CP1047,IBM930,ASCII,CP297. El valor predeterminado es CP297.
<code>cl.zonedMode</code>	cadena	EBCDIC_STRICT	El modo para codificar o decodificar los comandos del lenguaje de control (CL). Los valores permitidos son EBCDIC_STRICT /EBCDIC_MODIFIED /AS400.



Clave	Tipo	Valor predeterminado	Descripción
<code>ims.programs</code>	cadena		Lista de programas IMS que se van a utilizar. Separe cada parámetro con punto y coma (;) y cada transacción con una coma (,) . Por ejemplo: PCP008,PC T008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;
<code>jhdb.configuration.context.encoding</code>	cadena	CP297	La codificación JHDB (base de datos jerárquica de Java). Espera una cadena de codificación válidaCP1047,IBM930,ASCII,
<code>jhdb.metadata.extrapath</code>	cadena	file:./setup/	Parámetro de configuración que especifica una carpeta raíz adicional específica para el tiempo de ejecución para las carpetas psbs y dbds.

Clave	Tipo	Valor predeterminado	Descripción
jhdb.checkpointPersistence	cadena	Ninguno	<p>El modo de persistencia del punto de control. Los valores permitidos son none /add /end. Utilice add para conservar los puntos de control cuando se crea y se agrega uno nuevo al registro. Utilice end para conservar el punto de control al cerrar el servidor. Cualquier otro valor deshabilita la persistencia. Tenga en cuenta que cada vez que se añada un nuevo punto de control al registro, todos los puntos de comprobación existentes se serializarán y el archivo se borrará. No es un anexo a los datos existentes en el archivo. Por lo tanto, dependiendo del número de puntos de control, puede tener algunos efectos en el rendimiento.</p>

Clave	Tipo	Valor predeterminado	Descripción
jhdb.chkpointPath	cadena	archivo:./configuración/	Si jhdb.chkpointPeristence no es none, este parámetro le permite configurar la ruta de persistencia del punto de control (ubicación de almacenamiento del archivo checkpoint.dat). Todos los datos de los puntos de control contenidos en el registro se serializan y se copian en un archivo (checkpoint.dat) ubicado en la carpeta proporcionada. Tenga en cuenta que esta copia de seguridad solo afecta a los datos de los puntos de control (ScriptID, StepID, posición de la base de datos y área del punto de control).
jhdb.navigacion.cacheNexts	number	5000	La duración de la memoria caché (en milisegundos) utilizada en la navegación jerárquica de un RDBMS.

Clave	Tipo	Valor predeterminado	Descripción
<code>jhdb.use-db-prefix</code>	booleano	true	Especifica si se debe habilitar un prefijo de base de datos en la navegación jerárquica de un RDBMS.
<code>jhdb.query.limitJoinUsage</code>	booleano	true	Especifica si se debe utilizar el parámetro límite de uso de uniones en los gráficos del RDBMS.
<code>taskExecutor.corePoolSize</code>	number	5	Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño del grupo de núcleos.
<code>taskExecutor.maxPoolSize</code>	number	10	Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño máximo del pool (número máximo de subprocesos paralelos).

Clave	Tipo	Valor predeterminado	Descripción
<code>taskExecutor.queueCapacity</code>	number	50	Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño de la cola. (= número máximo de transacciones pendientes cuando se alcanza <code>taskExecutor.maxPoolSize</code> )
<code>taskExecutor.allowCoreThreadTimeOut</code>	valor booleano	false	Especifica si se debe permitir que se agote el tiempo de espera de los subprocesos principal es en JCIS. Esto permite un crecimiento y una reducción dinámicos incluso en combinación con una cola distinta de cero (ya que el tamaño máximo del grupo solo aumentará una vez que la cola esté llena).

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.runUnitLauncherPool.enable</code>	valor booleano	false	Especifica si se debe activar el grupo de lanzadores de unidades de ejecución en JICS.
<code>jics.runUnitLauncherPool.size</code>	number	20	El tamaño del conjunto de lanzadores de unidades de ejecución en JICS.
<code>jics.runUnitLauncherPool.validationInterval</code>	number	1 000	El intervalo entre cada ejecución de la tarea que ajusta el tamaño del grupo.
<code>jics.runUnitLauncherPool.parallelism</code>	number	2	El número de subprocesos que se utilizan para generar las instancias que faltan en la cola cuando se ejecuta la tarea de ajuste.
<code>context.preconstruct.enable</code>	valor booleano	false	Especifica si se debe activar la construcción previa del contexto del programa.
<code>context.preconstruct.frequencyInMillis</code>	number	100	El intervalo entre cada ejecución de la tarea que ajusta el tamaño del grupo.

Clave	Tipo	Valor predeterminado	Descripción
<code>context.parallelism</code>	number	5	El número de subprocesos que se utilizan para generar las instancias que faltan en la cola cuando se ejecuta la tarea de ajuste.
<code>context.minInstances</code>	number	2	El número de instancias que se crearán la primera vez que se necesite un contexto.
<code>spring.aws.application.credentials</code>	cadena	null	Cargue las credenciales de AWS del archivo de perfiles de credenciales en JICS.
<code>jics.queues.sqs.region</code>	cadena	eu-west-1	La región de AWS para AWS Simple Queue Service, utilizada en JICS.
<code>mq.queues.sqs.region</code>	cadena	eu-west-3	La región de AWS para el servicio AWS SQS MQ.

Clave	Tipo	Valor predeterminado	Descripción
<code>quartz.scheduler.standby-if-error</code>	valor booleano	false	Especifica si se debe activar la ejecución del trabajo si el programador de trabajos está en modo de espera. Si es verdadero, cuando está habilitada, no se activa la ejecución de tareas.
<code>databaseStatistics</code>	valor booleano	false	Especifica si se debe permitir que los compiladores de SQL recopilen y muestren información estadística.
<code>dbDateFormat</code>	cadena	aaaa-MM-dd	El formato de fecha objetivo de la base de datos.
<code>dbTimeFormat</code>	cadena	HH:mm:ss	El formato de hora objetivo de db.
<code>dbTimestampFormat</code>	cadena	yyyy-MM-dd HH:mm:ss.SSSSSS	El formato de marca de tiempo objetivo de la base de datos.



Clave	Tipo	Valor predeterminado	Descripción
dateTimeFormat	cadena	ISO	dateTimeFormat Describe cómo incluir el tipo de fecha, hora y hora de la base de datos en las entidades simplificadas de datos. Los valores permitidos son ISO /EUR /EUR /USA /LOCAL
localDateFormat	cadena		Lista de formatos de fecha local. Separe cada formato con \.
localTimeFormat	cadena		Lista de formatos de hora local. Separe cada formato con \
localTime stampFormat	cadena		Lista de formatos de marcas de tiempo locales. Separe cada formato con \.
pgmDateFormat	cadena	aaaa-MM-dd	formato de fecha hora
pgmTimeFormat	cadena	Hh.mm.ss	Formato de hora utilizado para la ejecución de los pgm (programas).
pgmTimestampFormat	cadena	yyyy-MM-dd-HH.mm.s s.SSSSSS	Formato de marca temporal

Clave	Tipo	Valor predeterminado	Descripción
cacheMetadata	booleano	true	Especifica si se deben almacenar en caché los metadatos de la base de datos
forceDisableSQLTrimStringType	valor booleano	false	Especifica si se debe deshabilitar el recorte de todos los parámetros de la cadena SQL.
fetchSize	number		El valor fetchSize de los cursores. Se usa cuando se obtienen datos mediante fragmentos a través de utilidades de carga o descarga.
check-groovy-file	booleano	true	Especifica si se debe comprobar el contenido de los archivos groovy antes de registrarlos.
qtemp.uuid.length	number	9	La longitud del identificador único de QTEMP.
qtemp.dblog	valor booleano	false	Si se debe habilitar el registro de la base de datos QTEMP.

Clave	Tipo	Valor predeterminado	Descripción
qtemp.cle anup.thre shold.hours	number	0	Para especificar cuándo qtemp.db1 og está habilitad o. La vida útil de la partición de base de datos (en horas).
sort.function	cadena		El nombre de la función de ordenació n de la base de datos blu4iv.
invalidDa taTolerance	booleano	true	Especifica si se toleran datos no válidos para el tipo empaquetado.
program.timeout	number	-1	Especifica un tiempo de espera en segundos para la ejecución de cualquier programa o transacci ón. Transcurrido este tiempo, el sistema intentará interrumpir el programa.

Clave	Tipo	Valor predeterminado	Descripción
gapwalk.1 ine.separator	cadena	null	Especifica el tipo de separador de líneas en Gapwalk. Los valores permitidos son WIN (CRLF)/UNIX (LF)/LINUX (LF). Los demás valores se ignoran y se utiliza la propiedad System line.sepatator.
enableActivePgmIdCache	valor booleano	false	Especifica si se debe habilitar la caché local del ID del programa activo. Utilice esta función con cuidado porque los recursos del JICS se pueden compartir entre los programas y los usuarios. Cualquier administrador puede cambiar esos recursos externamente y la memoria caché local instalada podría invalidarse.

Clave	Tipo	Valor predeterminado	Descripción
<code>mq.queues.default.syncpoint</code>	valor booleano	false	Especifica el comportamiento por defecto de los comandos MQ PUT cuando no están configurados ni MQPMO_SYNCPPOINT ni MQPMO_NO_SYNCPOINT. Cuando se establece en true, actúa como si los mensajes NO se confirmaran directamente durante el comando PUT. MQPMO_SYNCPOINT Si se establece en false, actúa como si MQPMO_NO_SYNCPOINT los mensajes se confirmaran directamente durante el comando PUT.
<code>dataSimplifier.byteRangeBoundsCheck</code>	valor booleano	false	Si se establece en verdadero, garantiza que no ByteRange se cree con valores incorrectos. El valor predeterminado es false.

Clave	Tipo	Valor predeterminado	Descripción
<code>file.stdo utIntoLogger</code>	valor booleano	false	Especifica si se debe habilitar la escritura en el registrador en lugar del flujo de salida predeterminado del sistema en los SYSPUNCH archivos SYSPRINT and predeterminados.
<code>tempFiles Directory</code>	cadena	null	Especifica el nombre de la ubicación de la carpeta de los archivos temporales que se generan.

Clave	Tipo	Valor predeterminado	Descripción
tempFolderPattern	cadena	null	<p>Especifica un patrón que se utilizará para crear dinámicamente el nombre de la carpeta temporal en función de la siguiente información predefinida y personalizable.</p> <p>HOST: el nombre del host.</p> <p>JOBID: el ID del trabajo.</p> <p>HASHCODE: el código hash del contexto laboral.</p> <p>TIMESTAMP: el patrón que se utilizará para obtener la marca de tiempo. El nombre de destino de la carpeta temporal es TMP_DIR_{temp}. FolderPattern Por ejemplo, en el caso del siguiente patrón, el nombre empezará con el identificador del trabajo y terminará con la «marca de tiempo» tempFolderPattern: JOBID, host=XXXX</p>

Clave	Tipo	Valor predeterminado	Descripción
			X, HASHCODE, timestamp =YYYYMMDD HHMMSS. Si la propiedad no tempFolderPattern se agrega al archivo YAML o está vacía, el nombre de la carpeta temporal será «TMP_DIR_» + this.hashCode () (DefaultJobContext).
database.cursor.raise.already.opened.error	valor booleano	false	Especifica si se debe activar el error 502 de SQLCODE cuando se abre un cursor ya abierto.
jics.spool.smtp.hostname	cadena	null	Especifica el host del servidor SMTP. Ejemplo: smtp.xxx.com
jics.spool.smtp.port	cadena	null	Especifica el puerto del servidor SMTP. Ejemplo: 25
jics.spool.smtp.password	cadena	null	Especifica la contraseña de inicio de sesión del servidor SMTP.



Clave	Tipo	Valor predeterminado	Descripción
<code>jics.spool.smtp.username</code>	cadena	null	Especifica el nombre de usuario del servidor SMTP.
<code>jics.spool.smtp.debug</code>	valor booleano	false	Especifica el modo de depuración del servidor SMTP.
<code>gapwalk-application.security</code>	cadena	disabled	Cambia la configuración de seguridad global (autenticación XSS, CORS, CSRF, OAUTH...). Los valores permitidos son disabled y enabled.
<code>gapwalk-application.identity</code>	cadena	null	Método de autenticación global. El valor recomendado es oauth. Los valores permitidos son json y oauth. Esta opción es necesaria cuando <code>gapwalk-application.security</code> es enabled.

Clave	Tipo	Valor predeterminado	Descripción
<code>gapwalk-application.security.issuerUri</code>	cadena	null	El URI del emisor del proveedor de identidad (IdP). Esta opción es necesaria cuando <code>gapwalk-application.identity</code> lo es. <code>oauth</code>
<code>gapwalk-application.security.allowedOrigins</code>	cadena []	null	La lista de orígenes que se van a permitir. Esta opción debe <code>gapwalk-application.identity</code> estar configurada en <code>oauth</code> .
<code>gapwalk-application.security.claimGroupName</code>	cadena	<code>cognito:groups</code>	El atributo de notificación que contiene la lista de todos los grupos a los que pertenece un usuario. Úselo <code>cognito:groups</code> para Amazon Cognito o cualquier otra cadena para un IdP extranjero.

Clave	Tipo	Valor predeterminado	Descripción
gapwalk-application.security.userNameAttribute	cadena	username	El nombre del atributo de notificación utilizado para identificar la solicitud de un usuario. Úselo username para Amazon Cognito, Keycloak o cualquier otra cadena preferred_username para un IdP extranjero.
gapwalk-application.localhostWhitelistingEnabled	booleano	true	Especifica si se debe habilitar la autenticación a partir de cualquier solicitud. localhost
gapwalk-application.defaultSuperAdminUserName	cadena	sadmin	Cuando gapwalk-application.security está deshabilitado, especifica el nombre de superusuario local predeterminado.

Clave	Tipo	Valor predeterminado	Descripción
<code>gapwalk-application.defaultSuperAdminUserPwd</code>	cadena	<code>sadmin</code>	Cuando <code>gapwalk-application.security</code> está deshabilitado, especifica la contraseña de superusuario local predeterminada.
<code>gapwalk-application.security.filterURIs</code>	cadena	<code>disabled</code>	Cambia la configuración de los URIs de filtrado. Los valores permitidos son <code>disabled</code> y <code>enabled</code> .
<code>gapwalk-application.security.blockedURIs</code>	cadena []	<code>null</code>	La lista de URIs que se van a bloquear. Esta opción es necesaria cuando <code>gapwalk-application.security.filterURIs</code> es <code>enabled</code> .
<code>jics.redis.databases</code>	number	<code>0</code>	Especifica el índice de base de datos de la fábrica de conexiones del servidor Redis, que va de 0 a 15. El valor predeterminado es 0.

Clave	Tipo	Valor predeterminado	Descripción
<code>jics.redis.maxTotal</code>	number	32	Redis agrupa el número máximo de conexiones activas.
<code>jics.redis.maxIdle</code>	number	32	Redis agrupa el número máximo de conexiones inactivas.
<code>jics.redis.minIdle</code>	number	8	Redis agrupa el número mínimo de conexiones inactivas.

## Propiedades disponibles para las aplicaciones web opcionales

Según la aplicación modernizada, es posible que necesite configurar una o más aplicaciones web opcionales que admitan dependencias como z/OS, AS/400 o IMS/MFS. Las siguientes tablas contienen listas de los parámetros clave/valor disponibles para configurar cada aplicación web opcional.

### gapwalk-utility-pgm.war

Esta aplicación web opcional contiene soporte para los programas de utilidades de Z/OS.

Esta tabla proporciona una vista exhaustiva de los parámetros clave/valores para esta aplicación.

Clave	Tipo	Valor predeterminado	Descripción
<code>logging.config</code>	Ruta	<code>classpath:logback-utility.xml</code>	Clave estándar para la referencia al archivo de configuración del logback. También hay disponibles otras claves de registro estándar.

Clave	Tipo	Valor predeterminado	Descripción
<code>spring.jta.enabled</code>	valor booleano	false	Clave estándar. Si el modo de soporte de la fuente de datos no es static-xa, la configuración automática de las transacciones JTA de Spring debe estar deshabilitada.
<code>spring.datasource.primary.jndi-name</code>	cadena	jdbc/primary	El nombre JNDI (interfaz de nombres y directorios de Java) de la fuente de datos principal, si se utiliza JNDI.
<code>primary.datasource-driver-class-name</code> <code>-url</code> <code>-username</code> <code>-password</code>	Origen de datos estándar de Spring con subclaves		<p>Contiene la información de conexión de la base de datos de la aplicación, si no se utiliza JNDI. Debe tener la misma configuración que en el archivo YAML de la aplicación modernizada.</p> <p>Como alternativa, se recomienda encarecidamente el uso de secretos de AWS, tal y como se explica en <a href="#">xref:../configuration/configuration.adoc[Configuration]</a>.</p>

Clave	Tipo	Valor predeterminado	Descripción
encoding	cadena	ASCII	La codificación utilizada en los programas de utilidades. Espera una codificación válida CP1047IBM930ASCIIUTF-8.
sysPunchEncoding	cadena	ASCII	El juego de caracteres que codifica el syspunch. Espera una codificación válida CP1047IBM930ASCIIUTF-8.
zonedMode	cadena	EBCDIC_STRICT	El modo para codificar o decodificar tipos de datos zonificados. Los valores permitidos son EBCDIC_STRICT /EBCDIC_MODIFIED /AS400.
unload.chunkSize	number	0	Tamaño de fragmento utilizado para la utilidad de descarga.
unload.sqlCodePointShift	number	0	El punto de cambio en el código SQL para la utilidad de descarga. Ejecuta el proceso de cambio de caracteres. Se requiere cuando la base de datos de destino de DB2 es Postgresql.

Clave	Tipo	Valor predeterminado	Descripción
<code>unload.columnFiller</code>	cadena	espacio	El rellenedor de columnas de la utilidad de descarga.
<code>unload.variableCharIsNull</code>	valor booleano	false	Utilice este parámetro en el programa INFTILB; si true se establece en, todos los campos que no admiten valores nulos con valores en blanco (espacio) devolverán una cadena vacía.
<code>unload.useDatabaseConfiguration</code>	valor booleano	false	Especifica si se debe usar la configuración de fecha u hora de <code>application-main.yml</code> en la utilidad de descarga.
<code>unload.format.date</code>	cadena	MM/dd/aaaa	Si <code>unload.useDatabaseConfiguration</code> está activado, el formato de fecha que se va a utilizar en la utilidad de descarga.



Clave	Tipo	Valor predeterminado	Descripción
<code>unload.format.time</code>	cadena	Hh.mm.ss	Si <code>unload.us</code> eDatabase Configuration está activado, el formato de hora que se utilizará en la utilidad de descarga.
<code>unload.format.timestamp</code>	cadena	yyyy-MM-dd-HH.mm.ss.SSSSSS	Si <code>unload.us</code> eDatabase Configuration está activado, el formato de marca de tiempo que se utilizará en la utilidad de descarga.
<code>unload.nb i.whenNull</code>	hexadecimal	6F	El valor del indicador de bytes nulos (NBI) que se añade cuando el valor de la base de datos es nulo.
<code>unload.nb i.whenNotNull</code>	hexadecimal	00	El valor del indicador de bytes nulos (NBI) que se añade cuando el valor de la base de datos no es nulo.
<code>unload.nb i.writeNullIndicator</code>	valor booleano	false	Especifica si se debe escribir el indicador nulo en el archivo de salida de descarga.

Clave	Tipo	Valor predeterminado	Descripción
<code>unload.fetchSize</code>	number	0	Permite ajustar el tamaño de la búsqueda al manipular los cursores en la utilidad de descarga.
<code>treatLargeNumbersAsInteger</code>	valor booleano	false	Especifica si se deben tratar los números grandes como Integer. Se tratan como BigDecimal por defecto.
<code>load.batchSize</code>	number	0	El tamaño del lote de la utilidad de carga.
<code>load.format.localDate</code>	cadena	dd.MM.aaaa\dd/MM/aaaa\aaaa-MM-dd	El formato de hora local de la utilidad de carga que se va a utilizar.
<code>load.format.localTime</code>	cadena	HH:mm:ss\ HH.mm.ss	El formato de hora local de la utilidad de carga que se va a utilizar.
<code>load.format.dbDate</code>	cadena	aaaa-MM-dd	El formato de base de datos de la utilidad de carga que se va a utilizar.
<code>load.format.dbTime</code>	cadena	HH:mm:ss	La hora de la base de datos de la utilidad de carga que se va a utilizar.

Clave	Tipo	Valor predeterminado	Descripción
load.sqlCodePointShift	number	0s	El punto de cambio en el código SQL para la utilidad de carga. Ejecuta el proceso de cambio de caracteres. Se requiere cuando la base de datos de destino de DB2 es Postgresql.
forcedDate	cadena		Fuerza la fecha a la fecha proporcionada, si la hay.
frozenDate	booleano	true	Especifica si se debe congelar la fecha. Solo se aplica si también forcedDate está establecida.
jcl.type	cadena	mvs	Tipo de archivo .jcl. Los valores permitidos son jcl /vse. Los comandos PRINT/REPRO de la utilidad IDCAMS devuelven 4 si el archivo está vacío en el caso de jcl que no es vse.
hasGraphic	valor booleano	false	Si la utilidad INFUTILB necesita gestionar columnas de GRAPHIC DB2.

## gapwalk-cl-command.war

Esta aplicación web opcional contiene soporte para los programas de utilidades del AS/400.

Esta tabla proporciona una vista exhaustiva de los parámetros clave/valores para esta aplicación.

Clave	Tipo	Valor predeterminado	Descripción
logging.config	Ruta	classpath:logback-utility.xml	Clave estándar para la referencia al archivo de configuración del logback. También hay disponibles otras claves de registro estándar.
spring.jta.enabled	valor booleano	false	Clave estándar. Si el modo de soporte de la fuente de datos no es static-xa, la configuración automática de las transacciones JTA de Spring debe estar deshabilitada.
spring.datasource.primary.jndi-name	cadena	jdbc/primary	El nombre JNDI (interfaz de nombres y directorios de Java) de la fuente de datos principal, si se utiliza JNDI.
primary.datasource + -driver-class-name + -url	Origen de datos estándar de Spring con subclaves		Contiene la información de conexión de la base de datos de la aplicación, si no se utiliza JNDI.

Clave	Tipo	Valor predeterminado	Descripción
+ -username + -password			<p>Debe tener la misma configuración que en el archivo yml de la aplicación modernizada.</p> <p>Como alternativa, se recomienda encarecidamente el uso de secretos de AWS, tal y como se explica en <code>xref:../configurat ion/configuration. adoc[Configuration]</code>.</p>
encoding	cadena	ASCII	La codificación utilizada en los programas de utilidades. Espera una codificación válida <code>CP1047IBM930ASCIITF-8</code> .
zonedMode	cadena	EBCDIC_STRICT	El modo para codificar o decodificar tipos de datos zonificados. Los valores permitidos son <code>EBCDIC_STRICT /EBCDIC_MODIFIED /AS400</code> .

Clave	Tipo	Valor predeterminado	Descripción
commands-off	cadena		Lista de comandos a desactivar, separados por comas. Los valores permitidos son PGM_BASIC, RCVMMSG, SNDRCVF, CHGVAR, Q. Útil cuando se desea deshabilitar o sobrescribir un programa existente. PGM_BASIC es un programa específico de AWS Blu Age Runtime diseñado con fines de depuración.
forcedDate	cadena		Fuerza la fecha a la fecha proporcionada, si la hay.

### gapwalk-hierarchical-support.war

Esta aplicación web opcional contiene soporte para transacciones de IMS/MFS.

Esta tabla proporciona una vista exhaustiva de los parámetros clave/valores para esta aplicación.

Clave	Tipo	Valor predeterminado	Descripción
logging.config	Ruta	classpath:logback-utility.xml	Clave estándar para la referencia al archivo de configuración del logback. También hay disponibles otras

Clave	Tipo	Valor predeterminado	Descripción
			claves de registro estándar.
<code>spring.jta.enabled</code>	valor booleano	false	Clave estándar. Si el modo de soporte de la fuente de datos no es static-xa, la configuración automática de las transacciones JTA de Spring debe estar deshabilitada.
<code>jhdb.configuration.context.encoding</code>	cadena		La codificación JHDB (base de datos jerárquica de Java). Espera una cadena de codificación válida CP1047, IBM930, ASCII,

Clave	Tipo	Valor predeterminado	Descripción
jhdb.checkpointPersistence	cadena	Ninguno	<p>El modo de persistencia del punto de control. Los valores permitidos son none /add /end. Utilice add para conservar los puntos de control cuando se crea y se agrega uno nuevo al registro. Utilice end para conservar el punto de control al cerrar el servidor. Cualquier otro valor deshabilita la persistencia. Tenga en cuenta que cada vez que se añada un nuevo punto de control al registro, todos los puntos de comprobación existentes se serializarán y el archivo se borrará. No es un anexo a los datos existentes en el archivo. Por lo tanto, dependiendo del número de puntos de control, puede tener algunos efectos en el rendimiento.</p>



## Configuración de la seguridad para las aplicaciones de Gapwalk

En los siguientes temas se describe cómo proteger las aplicaciones de Gapwalk.

Es su responsabilidad proporcionar la configuración correcta para garantizar que el uso del marco AWS Blu Age sea seguro.

Todas las funciones relacionadas con la seguridad están deshabilitadas de forma predeterminada. Para habilitar la autenticación (y CSRF, XSS, CSP, etc.), defina en `y. gapwalk-application.security enabled gapwalk-application.security.identity oauth`

### Temas

- [Configuración de la accesibilidad de los URI para las aplicaciones de Gapwalk](#)
- [Configurar la autenticación para las aplicaciones de Gapwalk](#)

## Configuración de la accesibilidad de los URI para las aplicaciones de Gapwalk

En este tema se describe cómo configurar el filtrado de los URI para las aplicaciones de Gapwalk. No requiere un proveedor de identidad (IdP).

Para bloquear una lista de URIs, añada las dos líneas siguientes a la `application-main.yml` de su aplicación modernizada y sustituya la *URI-1*, la *URI-2*, etc., por las URIs que desee bloquear.

```
gapwalk-application.security.filterURIs: enabled
gapwalk-application.security.blockedURIs: URI-1, URI-2, URI-3
```

## Configurar la autenticación para las aplicaciones de Gapwalk

En los siguientes temas se describe cómo configurar la autenticación OAuth2 para las aplicaciones de Gapwalk mediante un proveedor de identidad (IdP) como Cognito o Keycloak.

### Temas

- [Autenticación Gapwalk OAuth2 con Amazon Cognito](#)
- [Autenticación OAuth2 de Gapwalk con Keycloak](#)

## Autenticación Gapwalk OAuth2 con Amazon Cognito

En este tema se describe cómo configurar la autenticación OAuth2 para las aplicaciones de Gapwalk que utilizan Amazon Cognito como proveedor de identidad (IdP).

## Requisitos previos

En este tutorial utilizaremos Amazon Cognito como IdP y PlanetDemo como proyecto modernizado.

Puede utilizar cualquier otro proveedor de identidades externo. La ClientRegistration información debe obtenerse de su IDP y es necesaria para la autenticación de Gapwalk. Para obtener más información, consulte la [Guía para desarrolladores de Amazon Cognito](#).

La información: ClientRegistration

client-id

El ID de la ClientRegistration. En nuestro ejemplo, lo será PlanetsDemo.

client-secret

El secreto de tu cliente.

authorization endpoint

El URI del punto de conexión de autorización para el servidor de autorizaciones.

token endpoint

El URI del punto final del token del servidor de autorización.

jwt endpoint

El URI utilizado para obtener la clave web JSON (JWK) que contiene las claves para validar la firma web JSON emitida por el servidor de autorización.

redirect URI

El URI al que el servidor de autorización redirige al usuario final si se le concede el acceso.

## Configuración de Amazon Cognito

En primer lugar, crearemos y configuraremos un grupo de usuarios y usuarios de Amazon Cognito que utilizaremos con nuestra aplicación Gapwalk implementada con fines de prueba.

### Note

Si utilizas otro IdP, puedes saltarte este paso.

## Creación de un grupo de usuarios

1. Vaya a Amazon Cognito en AWS Management Console y auténtíquese con sus credenciales. AWS
2. Elija Grupos de usuarios.
3. Elija Crear un grupo de usuarios.
4. En Configurar la experiencia de inicio de sesión, mantenga el tipo de proveedor predeterminado Grupo de usuarios de Cognito. Puede elegir una o varias Opciones de inicio de sesión del grupo de usuarios de Cognito; por ahora, elija Nombre de usuario y, a continuación, Siguiente.

[Amazon Cognito](#) > [User pools](#) > Create user pool

Step 1  
**Configure sign-in experience**

Step 2  
Configure security requirements

Step 3  
Configure sign-up experience

Step 4  
Configure message delivery

Step 5  
Integrate your app

Step 6  
Review and create

### Configure sign-in experience [Info](#)

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

#### Authentication providers

Configure the providers that are available to users when they sign in.

#### Provider types

Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. [Learn more about pricing](#)

**Cognito user pool**  
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.

**Federated identity providers**  
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

#### Cognito user pool sign-in options [Info](#)

Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

**User name**

Email

Phone number

#### User name requirements

Allow users to sign in with a preferred user name

Make user name case sensitive

**⚠ Cognito user pool sign-in options can't be changed after the user pool has been created.**

[Cancel](#) [Next](#)

5. En Configurar requisitos de seguridad, mantenga los valores predeterminados y desactive la autenticación multifactor seleccionando Sin MFA y, a continuación, elija Siguiente.

Advanced security features can protect your production user accounts from malicious sign-in attempts. Activate it today from [App Integration](#). [Learn more](#)

**Configure security requirements**

Step 2  
Configure sign-up experience

Step 4  
Configure message delivery

Step 5  
Integrate your app

Step 6  
Review and create

**Password policy** [Info](#)  
Create a password policy to define the length and complexity of the passwords your users can set.

**Password policy mode** [Info](#)

**Cognito defaults**  
Use default password requirements.

**Custom**  
Use password requirements that you define.

**Password minimum length**  
8 character(s)

**Password requirements**  
Contains at least 1 number  
Contains at least 1 special character  
Contains at least 1 uppercase letter  
Contains at least 1 lowercase letter

**Temporary passwords set by administrators expire in**  
7 day(s)

**Multi-factor authentication**  
Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

**MFA enforcement** [Info](#)

**Require MFA - Recommended**  
Users must provide an additional authentication factor when signing in.

**Optional MFA**  
Users can sign in with a single authentication factor, and can choose to add additional authentication factors.

**No MFA**  
Users can only sign in with a single authentication factor. This is the least secure option.

**User account recovery**  
Configure how users will recover their account when they forget their password. Recipient message and data rates apply.

**Self-service account recovery** [Info](#)

**Enable self-service account recovery - Recommended**  
Allow forgot-password operations in your user pool. In the hosted UI sign-in page, a "Forgot your password?" link is displayed. When this feature is not enabled, administrators reset passwords with the Cognito API.

**Delivery method for user account recovery messages** [Info](#)

Select how your user pool will deliver messages when users request an account recovery code. SMS messages are charged separately by Amazon SNS. Email messages are charged separately by Amazon SES. [Learn more about pricing](#)

**Email only**

**SMS only**

**Email if available, otherwise SMS**

**SMS if available, otherwise email**

**SMS if available, otherwise email, and allow a user to reset their password via SMS if they are also using it for MFA**

Cancel [Previous](#) [Next](#)

6. Como medida de seguridad, desactive la opción **Habilitar el registro automático** y, a continuación, seleccione **Siguiente**.

**Self-service sign-up** [Info](#)  
Choose whether new users of your app can register for an account themselves.

**Self-registration** [Info](#)

**Enable self-registration**  
Display a "Sign up" link on the sign-in page in the hosted UI, and allow the use of public APIs to create new user accounts. When this feature is not enabled, federation and administrative API operations create user profiles.

7. Elija **Enviar correo electrónico con Cognito** y, a continuación, elija **Siguiente**.



## Email

Configure how your user pool sends email messages to users.

Email provider [Info](#)



Send email with Amazon SES - Recommended  
Send emails using an Amazon SES verified identity in your account. We recommend this option for higher email volume and production workloads.

Send email with Cognito  
Use Cognito's default email address as a temporary start for development. You can use it to send up to 50 emails a day.

You must have configured a verified sender with [Amazon SES](#)  to use the SES feature. [Learn more](#) 

SES Region [Info](#)  
Europe (Ireland)

FROM email address [Info](#)  
By default "no-reply@verificationemail.com" will be used. You can also choose a different email address that you have previously verified with Amazon SES.

no-reply@verificationemail.com  

REPLY-TO email address - *optional* [Info](#)  
If you set an invalid reply-to address, sending restrictions may be imposed on your account.

8. En Integre su aplicación, especifique un nombre para su grupo de usuarios. En Páginas de autenticación alojadas, elija Usar la interfaz de usuario alojada de Cognito.

Advanced security features can protect your production user accounts from malicious sign-in attempts. Activate it today from [App Integration](#). [Learn more](#)

Amazon Cognito > User pools > Create user pool

Step 1  
Configure sign-in experience

Step 2  
Configure security requirements

Step 3  
Configure sign-up experience

Step 4  
Configure message delivery

Step 5  
**Integrate your app**

Step 6  
Review and create

### Integrate your app info

Set up app integration for your user pool with Cognito's built-in authentication and authorization flows.

**User pool name**  
Create a friendly name for your user pool.

**User pool name**

User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + - . @ -

⚠
Your user pool name can't be changed once this user pool is created.

**Hosted authentication pages**

Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.

**Use the Cognito Hosted UI**  
Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

**Domain** info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. To use the Hosted UI, you must choose a domain where authentication endpoints will be created.

**Domain type**

**Use a Cognito domain**  
Enter an identifying prefix to use in an Amazon-owned domain. For production apps, we recommend using a custom domain instead.

**Use a custom domain**  
Enter a domain that you own for Cognito-hosted sign-up and sign-in pages. You must provide a DNS record and an AWS Certificate Manager (ACM) certificate to use a custom domain. We recommend using a custom domain for production workloads.

**Cognito domain**

Enter a domain prefix.

Domain prefixes may only include lowercase, alphanumeric characters, and hyphens. You can't use the text aws, amazon, or cognito in the domain prefix. Your domain prefix must be unique within the current Region.

✔ **Available**

**Initial app client**

Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

**App type** info

Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

9. Para simplificar, en Dominio, elija Usar un dominio de Cognito e introduzca un prefijo de dominio; por ejemplo, `https://planetsdemo` La aplicación de demostración debe añadirse como cliente.

1. En Cliente de aplicación inicial, elija Cliente confidencial. Introduzca el nombre de un cliente de aplicación, por ejemplo, `yp1anetsdemo`, a continuación, elija Generar un secreto de cliente.
2. En URL de devolución de llamada permitida, introduce la URL a la que redirigir al usuario tras la autenticación. La URL debe terminar en `/login/oauth2/code/cognito` Por ejemplo, para nuestras aplicaciones y aplicaciones de backend Gapwalk y BAC:

```

http://localhost:8080/bac
http://localhost:8080/bac/login/oauth2/code/cognito
http://localhost:8080/gapwalk-application
http://localhost:8080/gapwalk-application/login/oauth2/code/cognito
http://localhost:8080/planetsdemo
http://localhost:8080/planetsdemo/login/oauth2/code/cognito

```

Puedes editar la URL más adelante.

**Initial app client**  
Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

**App type** | [Info](#)  
Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

**Public client**  
A native, browser or mobile-device app. Cognito API requests are made from user systems that are not trusted with a client secret.

**Confidential client**  
A server-side application that can securely store a client secret. Cognito API requests are made from a central server.

**Other**  
A custom app. Choose your own grant, auth flow, and client-secret settings.

**App client name** | [Info](#)  
Enter a friendly name for your app client.  
planetsdemo  
App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + , . @ -

**Client secret** | [Info](#)  
Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API requests. Using a client secret can prevent a third party from impersonating your client.

**Generate a client secret**  
 Don't generate a client secret

**⚠ You cannot change or remove a client secret after you allow Amazon Cognito to generate it for your app client.**

**Allowed callback URLs** | [Info](#)  
Enter at least one callback URL to redirect the user back to after authentication. This is typically the URL for the app receiving the authorization code issued by Cognito. You may use HTTPS URLs, as well as custom URL schemes.

**URL**

Length of callback URL must be between 1 and 1024 characters. Valid characters are letters, marks, numbers, symbols, and punctuations. Amazon Cognito requires HTTPS over HTTP except for http://localhost for testing purposes only. App callback URLs such as myapp://example are also supported. Must not contain a fragment.

You can add 94 more URLs

► **Advanced app client settings**

- En URL de cierre de sesión permitidas, introduzca la URL de la página de cierre de sesión a la que desea que Amazon Cognito redirija cuando la aplicación cierre sesión de los usuarios. Por ejemplo, para las aplicaciones de backend de Gapwalk y BAC:

```
http://localhost:8080/bac/logout
http://localhost:8080/gapwalk-application/logout
http://localhost:8080/planetsdemo/logout
```

Puedes editar la URL más adelante.

- Mantén los valores predeterminados en las secciones de configuración avanzada del cliente de aplicaciones y de permisos de lectura y escritura de atributos.
  - Elija Siguiente.
- En Revisar y crear, compruebe sus opciones y, a continuación, seleccione Crear grupo de usuarios.

Para obtener más información, consulte [Creación de un grupo de usuarios](#).

## Creación de usuarios

Como el registro automático está desactivado, cree un usuario de Amazon Cognito. Vaya a la consola de Amazon Cognito. Elija el grupo de usuarios que creó y, a continuación, en Usuarios, elija Crear usuario.

En Información del usuario, elija Enviar una invitación por email, introduzca un nombre de usuario y una dirección de correo electrónico y, a continuación, seleccione Generar una contraseña. Seleccione la opción Crear usuario.

## Creación de roles

En la pestaña Grupos, cree 3 grupos (SUPER\_ADMIN, ADMIN y USER) y asocie su usuario a uno o más de estos grupos. Posteriormente, la aplicación Gapwalk asigna estas funciones a ROLE\_SUPER\_ADMIN, ROLE\_ADMIN y ROLE\_USER para poder acceder a algunas llamadas REST de API restringidas.

## Integración de Amazon Cognito en la aplicación Gapwalk

Ahora que el grupo de usuarios y los usuarios de Amazon Cognito están preparados, vaya al `application-main.yml` archivo de la aplicación modernizada y añada el siguiente código:

```
gapwalk-application.security: enabled
gapwalk-application.security.identity: oauth
gapwalk-application.security.issuerUri: https://cognito-idp.<region-id>.amazonaws.com/
<pool-id>
gapwalk-application.security.domainName: <your-cognito-domain>
gapwalk-application.security.localhostWhitelistingEnabled: false

spring:
  security:
    oauth2:
      client:
        registration:
          cognito:
            client-id: <client-id>
            client-name: <client-name>
            client-secret: <client-secret>
            provider: cognito
            authorization-grant-type: authorization_code
            scope: openid
            redirect-uri: "<redirect-uri>"
        provider:
          cognito:
```



```
issuer-uri: ${gapwalk-application.security.issuerUri}
authorization-uri: ${gapwalk-application.security.domainName}/oauth2/
authorize
jwks.json    jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/
token-uri: ${gapwalk-application.security.domainName}/oauth2/token
user-name-attribute: username
resourceserver:
  jwt:
    jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/jwks.json
```

Reemplace los siguientes marcadores de posición como se indica:

1. Vaya a Amazon Cognito en AWS Management Console y auténtíquese con sus credenciales.  
AWS
2. Elija grupos de usuarios y elija el grupo de usuarios que creó. Puede encontrar el *pool-id* en ID de grupo de usuarios.
3. Elija Integración de aplicaciones, donde podrá encontrar *su dominio cognito*, y luego vaya a Clientes y análisis de aplicaciones y elija su aplicación.
4. ***En el cliente de la aplicación: YourApp, puedes encontrar el nombre del cliente, el ID del cliente y el secreto del cliente (Mostrar el secreto del cliente).***
5. *region-id corresponde al ID* de AWS región en el que creó su usuario y grupo de usuarios de Amazon Cognito. Ejemplo:eu-west-3.
6. En ***redirect-uri, introduzca el URI*** que especificó para la URL de devolución de llamada permitida. En nuestro ejemplo, lo es. http://localhost:8080/planetsdemo/login/oauth2/code/cognito

Ahora puedes implementar tu aplicación Gapwalk y usar el usuario creado anteriormente para iniciar sesión en tu aplicación.

## Autenticación OAuth2 de Gapwalk con Keycloak

En este tema se describe cómo configurar la autenticación OAuth2 para las aplicaciones de Gapwalk que utilizan Keycloak como proveedor de identidad (IdP). En este tutorial, utilizamos Keycloak 24.0.0.

### Requisitos previos

- [Keycloak](#)

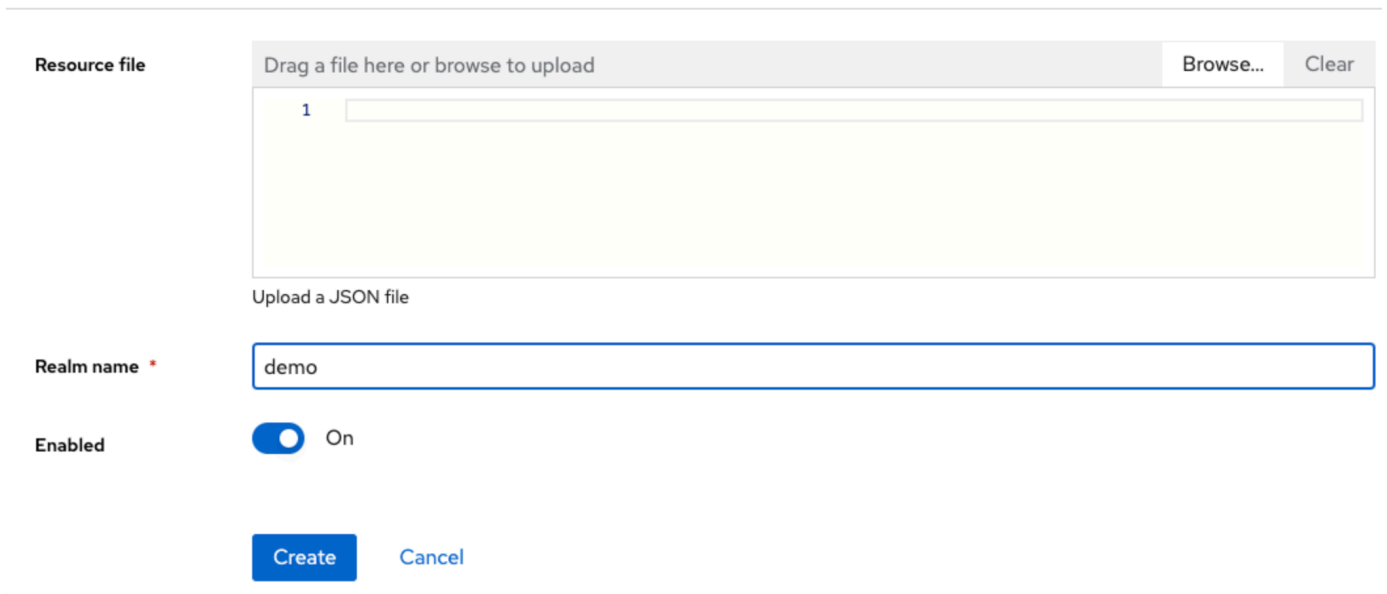
- Aplicación Gapwalk

## Configuración de Keycloak

1. Ve al panel de control de Keycloak en tu navegador web. Las credenciales predeterminadas son admin/admin. Vaya a la barra de navegación superior izquierda y cree un dominio con ese nombre **demo**, como se muestra en la siguiente imagen.

### Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and c



Resource file

Drag a file here or browse to upload Browse... Clear

1

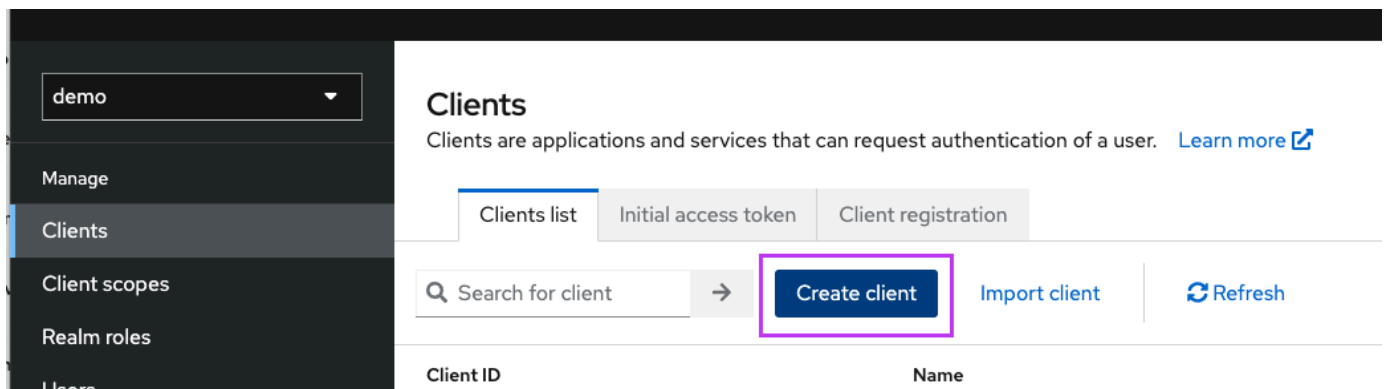
Upload a JSON file

Realm name \*

Enabled  On

Create Cancel

2. Cree un cliente con el nombre **app-demo**.



demo

Manage

Clients

Client scopes

Realm roles

Users

## Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list Initial access token Client registration

→ Create client Import client Refresh






Client ID Name

localhost:8080 Sustitúyalo por la dirección de tu aplicación Gapwalk

## General settings

Client ID * ?	<input type="text" value="app-demo"/>
Name ?	<input type="text"/>
Description ?	<input type="text"/>
Always display in UI ?	<input type="checkbox"/> Off

## Access settings

Root URL ?	<input type="text" value="http://localhost:8080"/>
Home URL ?	<input type="text"/>
Valid redirect URIs ?	<input type="text" value="http://localhost:8080/*"/>  <input type="text" value="https://localhost:8080/*"/>  <a href="#">+ Add valid redirect URIs</a>
Valid post logout redirect URIs ?	<input type="text" value="http://localhost:8080/*"/>  <input type="text" value="https://localhost:8080/*"/>  <a href="#">+ Add valid post logout redirect URIs</a>
Web origins ?	<input type="text" value="+"/>  <a href="#">+ Add web origins</a>

## Capability config

**Client authentication**  On

**Authorization**  Off

**Authentication flow**

- Standard flow [?](#)
- Direct access grants [?](#)
- Implicit flow [?](#)
- Service accounts roles [?](#)
- OAuth 2.0 Device Authorization Grant [?](#)
- OIDC CIBA Grant [?](#)

3. Para obtener el secreto de tu cliente, selecciona Clients, app-demo y Credentials.

**app-demo** OpenID Connect  Enabled [?](#) Action ▾

Clients are applications and services that can request authentication of a user.

Settings | Keys | **Credentials** | Roles | Client scopes | Service accounts roles | Sessions | Advanced

**Client Authenticator** [?](#) Client Id and Secret

**Save**

**Client Secret** [?](#) 5wfK2WyAPQ2Sap732p2Jf39LitIDzYk 🗑️ 📄 **Regenerate**

4. Selecciona Clientes, Ámbitos de cliente y, a continuación, Añadir mapeador predefinido. Elija los roles de dominio.

## Add predefined mappers

Choose any of the predefined mappings from this table

× → Refresh

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	groups	Map a user realm role to a token claim.
<input checked="" type="checkbox"/>	realm roles	Map a user realm role to a token claim.

Add

Cancel

5. Edite su rol de dominio con la configuración que se muestra en la siguiente imagen.


[Clients](#) > [Client details](#) > [Dedicated scopes](#) > Mapper details

## User Realm Role



ab8791fd-964d-48d2-89e7-c7234da3604e

Mapper type

User Realm Role

Name \* 


realm roles

Realm Role prefix Multivalued  OnToken Claim Name 

keycloak:groups

Claim JSON Type 


String

Add to ID token  On

Add to access token

 On

Add to lightweight

access token  OnAdd to userinfo  On

Add to token

introspection  On

6. Recuerde el nombre de reclamación del token definido. Necesitará este valor en la definición de configuración de Gapwalk para la `gapwalk-application.security.claimGroupName` propiedad.

The screenshot shows the 'Realm roles' page in the Keycloak admin console. The left sidebar is dark with a 'demo' dropdown menu. The main content area has a white background with the title 'Realm roles' and a subtitle 'Realm roles are the roles that you define for use in the current realm. [Lea](#)'. Below the subtitle is a search bar with a magnifying glass icon and the text 'Search role by name', followed by a right arrow button, a blue 'Create role' button, and a 'Refresh' button with a circular arrow icon. Below these buttons is a table with the following content:

Role name
ADMIN
SADMIN
USER

7. Elige los roles de Realms y crea 3 roles: **SUPER\_ADMIN**, **ADMIN** y **USER**. Estos roles se asignan posteriormente a `ROLE_SUPER_ADMIN`, `ROLE_ADMIN`, y `ROLE_USER` por la aplicación Gapwalk para poder acceder a algunas llamadas REST de API restringidas.

The screenshot shows the 'User details' page in the Keycloak admin console. The left sidebar is dark with a 'demo' dropdown menu. The main content area has a white background with the title 'User' and a subtitle 'Users > User details'. Below the subtitle are several tabs: 'Details', 'Credentials', 'Role mapping' (selected), 'Groups', 'Consents', and 'Identity'. Below the tabs is a search bar with a magnifying glass icon and the text 'Search by name', followed by a right arrow button, a checked checkbox labeled 'Hide inherited roles', and a blue 'Assign role' button. Below these buttons is a table with the following content:

<input type="checkbox"/>	Name
<input type="checkbox"/>	default-roles-demo
<input type="checkbox"/>	USER
<input type="checkbox"/>	ADMIN
<input type="checkbox"/>	SADMIN

## Integración de Keycloak en la aplicación Gapwalk

Edita el tuyo de la siguiente manera: `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
```



Sustituya <KEYCLOAK\_SERVER\_HOSTNAME><YOUR\_REALM\_NAME>,<YOUR\_CLIENT\_ID>, y por el nombre de <YOUR\_CLIENT\_SECRET>host del servidor Keycloak, el nombre de su dominio, su ID de cliente y su secreto de cliente.

## AWS API de ejecución de Blu Age

El AWS Blu Age Runtime utiliza varias aplicaciones web para exponer los puntos finales REST, lo que proporciona formas de interactuar con las aplicaciones modernizadas mediante clientes REST (por ejemplo, para llamar a las tareas mediante un programador).

El propósito de este documento es enumerar los puntos de conexión REST disponibles, proporcionando detalles sobre:

- ¿Cuál es su función
- ¿La forma de utilizarlos correctamente

La lista de puntos de conexión está organizada en categorías, según la naturaleza del servicio prestado y la aplicación web que muestre los puntos de conexión.

Suponemos que ya tiene conocimientos básicos sobre el uso de puntos de conexión REST (utilizando herramientas específicas como [POSTMAN](#), [Thunder Client](#), [CURL](#), [navegadores web](#), etc.) o escribir su propio código para realizar una llamada a la API.

### Temas

- [Creación de direcciones URL](#)
- [Aplicación Gapwalk](#)
- [Puntos de conexión REST de la consola de aplicaciones de Blusam](#)
- [Consola de la aplicación JICS](#)
- [Estructuras de base de datos](#)

## Creación de direcciones URL

Cada aplicación web que aparece a continuación define una ruta raíz, compartida por todos los puntos de conexión. Luego, cada punto de conexión agrega su propia ruta dedicada. La URL resultante que se utilizará es el resultado de la concatenación de las rutas. Por ejemplo, si consideramos el primer punto de conexión de la aplicación Gapwalk, tenemos:

- `/gapwalk-application` para la ruta raíz de la aplicación web.
- `/scripts` para la ruta de punto de conexión dedicada.

La URL resultante que se utilizará será `http://server:port/gapwalk-application/scripts`

`server`

apunta al nombre del servidor (el que aloja la aplicación web en cuestión).

`port`

el puerto expuesto por el servidor.

## Aplicación Gapwalk

Los puntos de conexión de la aplicación web Gapwalk utilizan la ruta raíz `/gapwalk-application`.

Temas

- [Puntos de conexión relacionados con trabajos por lotes \(JCL modernizados y similares\)](#)
- [Métricas para puntos de conexión](#)
- [Otros puntos de conexión](#)
- [Puntos de conexión relacionados con las colas de trabajos](#)

### Puntos de conexión relacionados con trabajos por lotes (JCL modernizados y similares)

Los trabajos por lotes se pueden ejecutar de forma sincrónica o asincrónica (consulte los detalles a continuación). Los trabajos por lotes se ejecutan mediante scripts groovy que son el resultado de la modernización de los scripts heredados (JCL).

Temas

- [Enumere los scripts implementados](#)
- [Lanzar un script de forma sincrónica](#)
- [Lanzar un script de forma asincrónica](#)
- [Listado de scripts activados](#)

- [Recuperar los detalles de la ejecución del trabajo](#)
- [Lista los scripts lanzados de forma asincrónica que se pueden eliminar](#)
- [Lista los scripts lanzados de forma sincrónica que se pueden eliminar](#)
- [Eliminar una ejecución de trabajo determinada](#)
- [Listado de los puntos de comprobación existentes para la reiniciabilidad](#)
- [Reiniciar un trabajo \(de forma sincrónica\)](#)
- [Reiniciar un trabajo \(de forma asincrónica\)](#)
- [Establecer el límite de subprocesos para la ejecución de trabajos asincrónica](#)

Enumere los scripts implementados

- Método compatible: GET
- Ruta: /scripts
- Argumentos: ninguno
- Este punto de conexión devuelve la lista de scripts groovy desplegados en el servidor, en forma de cadena. Este punto de conexión está diseñado principalmente para ser utilizado desde un navegador web, ya que la cadena resultante es una página HTML, con enlaces activos (un enlace por script que se pueda iniciar; consulte el ejemplo siguiente).

Respuesta de ejemplo:

```
<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>
```

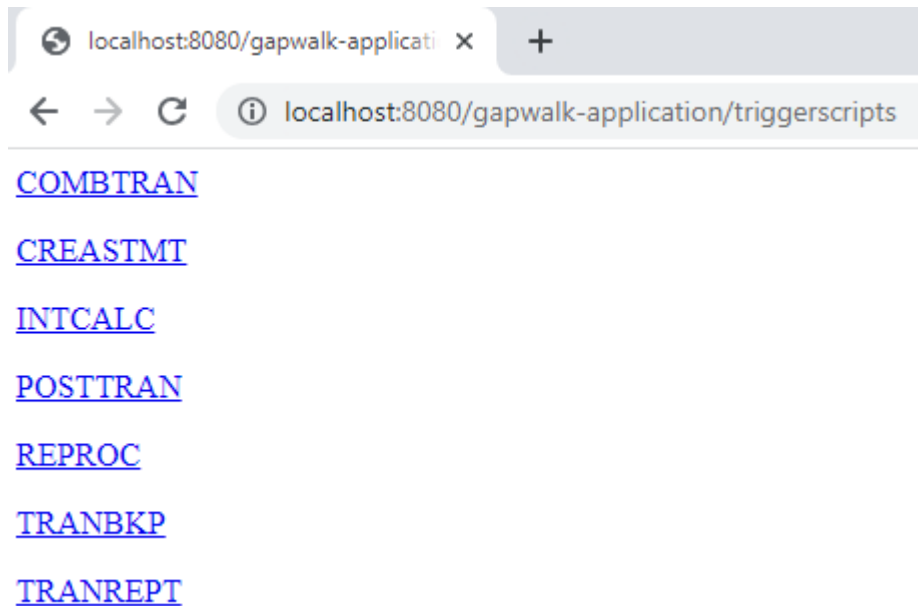
#### Note

Los enlaces representan la URL que se utilizará para lanzar cada script de la lista de forma sincrónica.

- Método compatible: GET
- Ruta: /triggerscripts

- Argumentos: ninguno
- Este punto de conexión devuelve la lista de scripts groovy desplegados en el servidor, en forma de cadena. Este punto de conexión está diseñado principalmente para ser utilizado desde un navegador web, ya que la cadena resultante es una página HTML, con enlaces activos (un enlace por script que se pueda iniciar; consulte el ejemplo siguiente).

A diferencia de la respuesta anterior del punto de conexión, los enlaces representan la URL que se debe utilizar para lanzar cada script de la lista de forma asincrónica.



Lanzar un script de forma sincrónica

Este punto de conexión tiene dos variantes con rutas dedicadas para el uso de GET y POST (ver más abajo).

- Método compatible: GET
- Ruta: /script/{scriptId:..+}
- Método compatible: POST
- Ruta: /post/script/{scriptId:..+}
- Argumentos:
  - identificador del script que se va a lanzar

- opcionalmente: parámetros para pasarlos al script, utilizando los parámetros de solicitud (vistos como un `Map<String,String>`). Los parámetros dados se añadirán automáticamente a los [enlaces](#) del script groovy invocado.
- La llamada lanzará el script con el identificador dado, utilizando parámetros adicionales si se proporcionan y esperará a que se complete la ejecución del script antes de devolver un mensaje (`String`) que será:
  - “Done” (si la ejecución del trabajo se realizó sin problemas).
  - Un mensaje de error de JSON con detalles sobre lo que ha fallado durante la ejecución del trabajo. Se pueden obtener más detalles de los registros del servidor para saber qué ha fallado en la ejecución del trabajo.

```
{
  "exitCode": -1,
  "stepName": "STEP15",
  "program": "CBACT04C",
  "status": "Error"
}
```

Al observar los registros del servidor, podemos darnos cuenta de que se trata de un problema de implementación (el programa esperado no se ha implementado correctamente, por lo que no se puede encontrar, lo que provoca un error en la ejecución del trabajo):

```
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.JJobExecutor - a65c2791-864f-43c9-972a-b5f2353389e6 - worker :Thread-26 [1547512424]
2023-06-09 10:27:28 default INFO - c.n.b.g.r.j.s.JJobExecutor - 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

Las llamadas sincrónicas deben reservarse para tareas de corta duración. Los trabajos que se ejecutan durante mucho tiempo deberían lanzarse más bien de forma asincrónica (consulte el punto de conexión específico a continuación).

## Lanzar un script de forma asincrónica

- Métodos compatibles: GET/POST
- Ruta: `/triggerscript/{scriptId:.+}`
- Argumentos:
  - identificador del script que se va a lanzar
  - opcionalmente: parámetros para pasarlos al script, utilizando los parámetros de solicitud (vistos como un `Map<String, String>`). Los parámetros indicados se añadirán automáticamente a la página <https://docs.groovy-lang.org/latest/html/api/groovy/lang/Binding.html>[bindings] del script groovy invocado.
- A diferencia del modo sincrónico anterior, el punto de conexión no espera a que finalice la ejecución del trabajo para enviar una respuesta. La ejecución de la tarea se inicia al mismo tiempo, si se encuentra un subproceso disponible para hacerlo, y se envía inmediatamente una respuesta al programa que hace la llamada, con el identificador de ejecución de la tarea, un identificador único que representa la ejecución de la tarea, que se puede utilizar para consultar el estado de la ejecución de la tarea o forzar la finalización de una ejecución de tareas que se supone que no funciona correctamente. El formato de la respuesta es:

```
Triggered script <script identifier> [unique job execution id] @ <date and time>
```

- Dado que la ejecución asincrónica del trabajo se basa en un número fijo y limitado de subprocesos, es posible que la ejecución del trabajo no se inicie si no se encuentra ningún subproceso disponible. En ese caso, el mensaje devuelto se ve del siguiente modo:

```
Script [<script identifier>] NOT triggered - Thread limit reached (<actual thread limit>) - Please retry later or increase thread limit.
```

Consulte el siguiente punto de conexión `settriggerthreadlimit` para obtener información sobre cómo aumentar el límite de subprocesos.

### Respuesta de ejemplo:

```
Triggered script INTCALC [d43cbf46-4255-4ce2-aac2-79137573a8b4] @ 06-12-2023 16:26:15
```

El identificador único de ejecución de tareas permite recuperar rápidamente las entradas de registro relacionadas en los registros del servidor, si es necesario. También lo utilizan varios otros puntos de conexión que se detallan a continuación.

### Listado de scripts activados

- Método compatible: GET
- Rutas: `/triggeredscripts/{status:.+}`, `/triggeredscripts/{status:.+}/  
{namefilter}`
- Argumentos:
  - Estado (obligatorio): el estado de los scripts activados para su recuperación. Los valores posibles son los siguientes:
    - `all`: muestra todos los detalles de la ejecución de los trabajos, independientemente de si los trabajos siguen ejecutándose o no.
    - `running`: muestra solo los detalles de los trabajos que se están ejecutando actualmente.
    - `done`: muestra solo los detalles de los trabajos cuya ejecución ha finalizado.
    - `killed`: solo muestra los detalles de los trabajos cuya ejecución se ha interrumpido forzosamente mediante el punto final específico (ver más abajo).
    - `triggered`: muestra solo los detalles de los trabajos que se han activado pero que aún no se han lanzado.
    - `failed`: muestra solo los detalles de los trabajos cuya ejecución se ha marcado como fallida.
    - `_namefilter` (opcional) `_`: recupera solo las ejecuciones del identificador de script dado.
  - Devuelve una colección de detalles de las ejecuciones de los trabajos en formato JSON. Para obtener más información, consulte [Estructura de mensajes de detalles de ejecución del trabajo](#).

### Respuesta de ejemplo:

```
[
  {
    "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"  
  }  
]
```

## Recuperar los detalles de la ejecución del trabajo

- Método compatible: GET
- Ruta: `/getjobexecutioninfo/{jobexecutionid:.+}`
- Argumentos:
  - `jobexecutionid` (obligatorio): el identificador único de ejecución de tareas para recuperar los detalles de ejecución de tareas correspondientes.
- Devuelve: una cadena JSON que representa los detalles de la ejecución de una sola tarea (consulte [Estructura de mensajes de detalles de ejecución del trabajo](#)) o una respuesta vacía si no se pudieron encontrar detalles de ejecución de la tarea para el identificador dado.

## Lista los scripts lanzados de forma asincrónica que se pueden eliminar

- Método compatible: GET
- Ruta: `/killablescripts`
- Devuelve una colección de identificadores de ejecución de tareas que se han lanzado de forma asincrónica y que aún se están ejecutando y que se pueden cerrar por la fuerza (consulte el punto de conexión `/kill` a continuación).

## Lista los scripts lanzados de forma sincrónica que se pueden eliminar

- Método compatible: GET
- Ruta: `/killablesyncscripts`
- Devuelve una colección de identificadores de ejecución de tareas que se han lanzado de forma sincrónica y que aún se están ejecutando y que se pueden cerrar por la fuerza (consulte el punto de conexión `/kill` a continuación).

## Eliminar una ejecución de trabajo determinada

- Método compatible: GET



- Ruta: `/kill/{identifíer:.+}`
- argumento: identificador de ejecución de la tarea (obligatorio): el identificador único de ejecución de la tarea que apunta a la ejecución de la tarea que se va a anular por la fuerza.
- Devuelve: un mensaje textual en el que se detalla el resultado del intento de ejecución de la tarea; el mensaje contendrá el identificador del script, el identificador único de la ejecución de la tarea y la fecha y hora en que se produjo la ejecución. Si no se encuentra ninguna ejecución de trabajo en ejecución para el identificador indicado, se devolverá un mensaje de error en su lugar.

#### Warning

- El motor de ejecución hace todo lo posible para acabar de forma adecuada con la ejecución de la tarea objetivo. Por lo tanto, la respuesta del punto final `/kill` puede tardar un poco en llegar a la persona que llama, ya que el tiempo de ejecución de AWS Blu Age intentará minimizar el impacto empresarial que supone interrumpir el trabajo.
- Eliminar por la fuerza la ejecución de un trabajo no debe hacerse a la ligera, ya que puede tener consecuencias comerciales directas, incluida la posible pérdida o corrupción de datos. Debe reservarse para los casos en los que la ejecución de una determinada tarea no se haya realizado correctamente y los medios de corrección de datos estén claramente identificados.
- Destruir un trabajo debería dar lugar a nuevas investigaciones (análisis post mortem) para averiguar qué fue lo que salió mal y tomar las medidas correctivas adecuadas.
- En cualquier caso, el intento de interrumpir un trabajo en ejecución se anotará en los registros del servidor con mensajes de nivel de advertencia.

#### Listado de los puntos de comprobación existentes para la reiniciabilidad

La reiniciabilidad de los trabajos se basa en la capacidad de los scripts de registrar puntos de comprobación en el `CheckpointRegistry` para seguir el progreso de la ejecución del trabajo. Si la ejecución de una tarea no finaliza correctamente y se han registrado los puntos de comprobación de reinicio, basta con reiniciar la ejecución de la tarea desde el último punto de comprobación registrado conocido (sin tener que ejecutar los pasos anteriores al punto de comprobación).

- Método compatible: GET
- Ruta: `/restarts`

- Devuelve la lista de puntos de reinicio existentes, que se pueden utilizar para reiniciar un trabajo cuya ejecución no ha finalizado correctamente, en forma de página html. Si ningún script ha registrado ningún punto de comprobación, el contenido de la página será "No registered checkpoints."

#### Reiniciar un trabajo (de forma sincrónica)

- Método compatible: GET
- Ruta: `/restart/{hashcode}`
- Argumentos: código hash (entero: obligatorio): reinicia una ejecución de trabajo abortada anteriormente, utilizando el código hash proporcionado como valor de punto de comprobación (consulte el punto de conexión `/restarts` anterior para obtener información sobre cómo recuperar un valor de punto de comprobación válido).
- Devoluciones: consulte la descripción de devolución de `script` anterior.

#### Reiniciar un trabajo (de forma asincrónica)

- Método compatible: GET
- Ruta: `/triggerrestart/{hashcode}`
- Argumentos: código hash (entero: obligatorio): reinicia una ejecución de trabajo abortada anteriormente, utilizando el código hash proporcionado como valor de punto de comprobación (consulte el punto de conexión `/restarts` anterior para obtener información sobre cómo recuperar un valor de punto de comprobación válido).
- Devoluciones: consulte la descripción de devolución de `triggerscript` anterior.

#### Establecer el límite de subprocesos para la ejecución de trabajos asincrónica

La ejecución asincrónica del trabajo se basa en un grupo dedicado de subprocesos en la JVM. Ese grupo tiene un límite fijo en cuanto al número de subprocesos disponibles. El usuario tiene la capacidad de ajustar el límite de acuerdo con las capacidades del host (número de CPU, memoria disponible, etc.). De forma predeterminada, el límite de subprocesos está establecido en 5 subprocesos.

- Método compatible: GET
- Ruta: `/settriggerthreadlimit/{threadlimit:.+}`

- Argumento (entero): el nuevo límite de subprocesos que se aplicará. Debe ser un entero estrictamente positivo.
- Devuelve un mensaje (String) con el nuevo límite de subprocesos y el anterior, o un mensaje de error si el valor límite de subprocesos proporcionado no es válido (no es un entero estrictamente positivo).

Respuesta de ejemplo:

```
Set thread limit for Script Tower Control to 10 (previous value was 5)
```

Contar las ejecuciones de trabajos desencadenados en curso

- Método compatible: GET
- Ruta: `/countrunningtriggeredscripts`
- Devuelve un mensaje que indica el número de trabajos en ejecución lanzados de forma asíncrona y el límite de subprocesos (es decir, el número máximo de trabajos activados que se pueden ejecutar simultáneamente).

Respuesta de ejemplo:

```
0 triggered script(s) running (limit =10)
```

#### Note

Se puede utilizar para comprobar, antes de lanzar un trabajo, si no se ha alcanzado el límite de subprocesos (lo que impediría lanzar el trabajo).

Purgar la información sobre las ejecuciones de trabajos

La información sobre las ejecuciones de los trabajos permanece en la memoria del servidor mientras el servidor esté activo. Puede ser conveniente purgar la información más antigua de la memoria, pues ya no es relevante; este es el propósito de este punto de conexión.

- Método compatible: GET
- Ruta: `/purgejobinformation/{age:.+}`

- Argumentos: un valor entero estrictamente positivo que representa la antigüedad en horas de la información que se va a purgar.
- Devuelve un mensaje con la siguiente información:
  - Nombre del archivo de purga en el que se almacena la información de ejecución de los trabajos purgados con fines de archivado.
  - Número de información de ejecución de tareas purgadas.
  - Número de información restante sobre la ejecución del trabajo en la nota

## Métricas para puntos de conexión

### JVM

Este punto de conexión devuelve las métricas disponibles relacionadas con la JVM.

- Método compatible: GET
- Ruta: `/metrics/jvm`
- Argumentos: ninguno
- Devuelve un mensaje con la siguiente información:
  - `hiloActiveCount`: número de hilos activos.
  - `jvmMemoryUsed`: memoria utilizada activamente por la máquina virtual Java.
  - `jvmMemoryMax`: memoria máxima permitida para la máquina virtual Java.
  - `jvmMemoryFree`: memoria disponible que la máquina virtual Java no está utilizando actualmente.

### Sesión

Este punto de conexión devuelve las métricas relacionadas con las sesiones HTTP abiertas en ese momento.

- Método compatible: GET
- Ruta: `/metrics/session`
- Argumentos: ninguno
- Devuelve un mensaje con la siguiente información:
  - `sessionCount`: número de sesiones de usuario activas que actualmente mantiene el servidor.

## Lote

- Método compatible: GET
- Ruta: `/metrics/batch`
- Argumentos:
  - `startTimestamp` (opcional, número): marca de tiempo inicial para el filtrado de datos.
  - `endTimestamp` (opcional, número): marca de tiempo final para el filtrado de datos.
  - `page` (opcional, número): número de página para la paginación.
  - `pageSize` (opcional, número): número de elementos por página en la paginación.
- Devuelve un mensaje con la siguiente información:
  - `content`: lista de métricas de ejecución por lotes.
  - `pageNumber`: número de página actual en la paginación.
  - `pagesize`: número de elementos mostrados por página.
  - `totalPages`: número total de páginas disponibles.
  - `numberOfElements`: número de elementos de la página actual.
  - `last`: marca booleana para la última página.
  - `first`: marca booleana para la primera página.

## Transacción

- Método compatible: GET
- Ruta: `/metrics/transaction`
- Argumentos:
  - `startTimestamp` (opcional, número): marca de tiempo inicial para el filtrado de datos.
  - `endTimestamp` (opcional, número): marca de tiempo final para el filtrado de datos.
  - `page` (opcional, número): número de página para la paginación.
  - `pageSize` (opcional, número): número de elementos por página en la paginación.
- Devuelve un mensaje con la siguiente información:
  - `content`: lista de métricas de ejecución de transacciones.
  - `pageNumber`: número de página actual en la paginación.
  - `pagesize`: número de elementos mostrados por página.
  - `totalPages`: número total de páginas disponibles.

- `númeroOfElements`: número de elementos de la página actual.
- `last`: marca booleana para la última página.
- `first`: marca booleana para la primera página.

## Otros puntos de conexión

Utilice estos puntos de conexión para listar los programas o servicios registrados, conocer el estado y administrar las transacciones del JICS.

### Temas

- [Visualización de los programas registrados](#)
- [Listado de servicios registrados](#)
- [Estado](#)
- [Listado de las transacciones JICS disponibles](#)
- [Lanzar una transacción de JICS](#)
- [Lanzar una transacción de JICS \(alternativa\)](#)

### Visualización de los programas registrados

- Método compatible: GET
- Ruta: `/programs`
- Devuelve la lista de programas registrados, en forma de página html. Cada programa se designa mediante su identificador de programa principal. Se devuelven a la lista tanto los programas antiguos modernizados como los programas de utilidades (IDCAMS, IEBGENER, etc.). Tenga en cuenta que los programas de utilidades disponibles dependerán de las aplicaciones web de utilidades que se hayan implementado en su servidor Tomcat. Por ejemplo, es posible que los programas de compatibilidad de utilidades de z/OS no estén disponibles para los activos modernizados de iSeries, ya que no son relevantes.

### Listado de servicios registrados

- Método compatible: GET
- Ruta: `/services`

- Devuelve la lista de servicios de tiempo de ejecución registrados, en forma de página html. El motor de ejecución de AWS Blu Age ofrece estos servicios como utilidades, que se pueden utilizar, por ejemplo, en scripts geniales. Los servicios de carga de Blusam (para crear conjuntos de datos de Blusam a partir de conjuntos de datos antiguos) entran en esa categoría.

Respuesta de ejemplo:

```
<p>BluesamESDSFileLoader</p><p>BluesamKSDSFileLoader</p><p>BluesamRRDSFileLoader</p>
```

Estado

- Método compatible: GET
- Ruta: /
- Devuelve un mensaje sencillo que indica que la aplicación Gapwalk está activa y en ejecución (Jics application is running.)

Listado de las transacciones JICS disponibles

- Método compatible: GET
- Ruta: /transactions
- Devuelve una página html con una lista de todas las transacciones JICS disponibles. Esto solo tiene sentido para entornos con elementos JICS (modernización de elementos CICS heredados).

Respuesta de ejemplo:

```
<p>INQ1</p><p>MENU</p><p>MNT2</p><p>ORD1</p><p>PRNT</p>
```

Lanzar una transacción de JICS

- Métodos compatibles: GET,POST
- Ruta: /jicstransrunner/{jtrans:.+}
- Argumentos:
  - Identificador de transacción JICS (cadena, obligatorio): identificador de la transacción JICS que se va a lanzar (8 caracteres como máximo)

- obligatorio: datos de entrada adicionales para pasarlos a la transacción, en forma de `Map<String, Object>`. El contenido de este mapa se utilizará para alimentar la [COMMAREA](#) que consumirá la transacción del JICS. El mapa puede estar vacío si no se requieren datos para ejecutar la transacción.
- opcional: entradas de encabezados HTTP para personalizar el entorno de ejecución de la transacción en cuestión. Se admiten las siguientes claves de encabezado:
  - `jics-channel`: el nombre del JICS CHANNEL que utilizará el programa que se lanzará al lanzar esta transacción.
  - `jics-container`: el nombre del JICS CONTAINER que se utilizará para el lanzamiento de esta transacción de JICS.
  - `jics-startcode`: el STARTCODE (cadena, de hasta 2 caracteres) que se utilizará al iniciar la transacción con el JICS. Consulte [STARTCODE](#) para ver los valores posibles (desplácese hacia abajo en la página).
  - `jicxa-xid`: el XID (estructura XID del identificador de transacción X/Open) de una “transacción global” ([XA](#)), iniciada por el programa que hace la llamada, en la que participará el lanzamiento actual de la transacción del JICS.
- Devuelve: una serialización de JSON `com.netfective.bluage.gapwalk.rt.shared.web.TransactionResultBean`, que representa la salida del lanzamiento de la transacción del JICS.

Para obtener más información sobre los detalles de la estructura, consulte [Estructura de resultados del lanzamiento de la transacción](#).

Lanzar una transacción de JICS (alternativa)

- Métodos compatibles: GET,POST
- Ruta: `/jicstransaction/{jtrans:.+}`
- Argumentos:  
identificador de transacción JICS (cadena, obligatorio)

identificador de la transacción JICS que se va a lanzar (8 caracteres como máximo)



obligatorio: datos de entrada adicionales para pasarlos a la transacción, en forma de Map<String, Object>.

El contenido de este mapa se utilizará para alimentar la [COMMAREA](#) que consumirá la transacción del JICS. El mapa puede estar vacío si no se requieren datos para ejecutar la transacción.

opcional: entradas de encabezados HTTP para personalizar el entorno de ejecución de la transacción en cuestión.

Se admiten las siguientes claves de encabezado:

- `jics-channel`: el nombre del JICS CHANNEL que utilizará el programa que se lanzará al lanzar esta transacción.
  - `jics-container`: el nombre del JICS CONTAINER que se utilizará para el lanzamiento de esta transacción de JICS.
  - `jics-startcode`: el STARTCODE (cadena, de hasta 2 caracteres) que se utilizará al iniciar la transacción con el JICS. Para ver los valores posibles, consulte [STARTCODE](#) (navigue hacia abajo en la página).
  - `jicxa-xid`: el XID (estructura XID del identificador de transacción X/Open) de una “transacción global” ([XA](#)), iniciada por el programa que hace la llamada, en la que participará el lanzamiento actual de la transacción del JICS.
- Devuelve: una serialización de JSON `com.netfective.bluage.gapwalk.rt.shared.web.RecordHolderBean`, que representa la salida del lanzamiento de la transacción del JICS. Los detalles de la estructura se pueden consultar en [Estructura de resultados del registro de lanzamiento de la transacción](#).

## Puntos de conexión relacionados con las colas de trabajos

Las colas de trabajos son el soporte de AWS Blu Age para el mecanismo de presentación de trabajos AS400. Las colas de trabajos se utilizan en AS400 para ejecutar trabajos en grupos de subprocesos específicos. Una cola de trabajos se define mediante un nombre y un número máximo de subprocesos que corresponde al número máximo de programas que se pueden ejecutar simultáneamente en esa cola. Si se envían más trabajos a la cola que el número máximo de subprocesos, los trabajos esperarán a que haya un subproceso disponible.

Para obtener una lista exhaustiva del estado de un trabajo en cola, consulte [Los posibles estados de un trabajo en una cola son:](#)

Las operaciones en las colas de trabajos se gestionan a través de los siguientes puntos de conexión específicos. Puede invocar estas operaciones desde la URL de la aplicación Gapwalk con la siguiente URL raíz: `http://server:port/gapwalk-application/jobqueue`

## Temas

- [Listado de las colas disponibles](#)
- [Iniciar o reiniciar una cola de trabajo](#)
- [Enviar un trabajo para su lanzamiento](#)
- [Enumere todos los trabajos enviados](#)
- [Liberación de todos los trabajos que están “en espera”](#)
- [Liberación de todos los trabajos que estén “en espera” de un nombre de trabajo determinado](#)
- [Liberación de un trabajo determinado de un número de trabajo](#)
- [Envíe un trabajo siguiendo un cronograma repetido](#)
- [Enumere todos los trabajos repetitivos enviados](#)
- [Cancela la programación de un trabajo repetido](#)

## Listado de las colas disponibles

- Método compatible: GET
- Ruta: `list-queues`
- Devuelve la lista de las colas disponibles junto con su estado, como una lista JSON de valores clave.

## Respuesta de ejemplo:

```
{"Default": "STAND_BY", "queue1": "STARTED", "queue2": "STARTED"}
```

Los posibles estados de una cola de trabajos son:

### STAND\_BY

la cola de trabajos está esperando a que se inicie.

### STARTED

la cola de trabajos está activa y funcionando.

## UNKNOWN

no se puede determinar el estado de la cola de trabajos.

### Iniciar o reiniciar una cola de trabajo

- Método compatible: POST
- Ruta: `/restart/{name}`
- Argumento: el nombre de la cola que se va a iniciar o reiniciar, en forma de cadena: obligatorio.
- El punto de conexión no devuelve nada, sino que se basa en el estado de http para indicar el resultado de la operación de inicio o reinicio:

HTTP 200

la operación de inicio o reinicio ha ido bien: la cola de trabajos indicada ahora está STARTED.

HTTP 404

la cola de trabajos no existe.

HTTP 503

se ha producido una excepción durante el intento de inicio o reinicio (se deben inspeccionar los registros del servidor para averiguar qué ha fallado).

### Enviar un trabajo para su lanzamiento

- Método compatible: POST
- Ruta: `/submit`
- Argumento: obligatorio como cuerpo de la solicitud, una serialización JSON de un objeto `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage`. Para obtener más información, consulte [Envíe el trabajo y programe la entrada del trabajo](#).
- Devuelve: un JSON que contiene el `SubmitJobMessage` original y un registro que indica si el trabajo se ha enviado o no.

### Enumere todos los trabajos enviados

- Método compatible: GET
- Ruta: `/list-jobs?status={status}&size={size}&page={page}&sort={sort}`

- Argumentos:
  - página: número de página que se va a recuperar (predeterminado = 1)
  - tamaño: tamaño de la página (predeterminado = 50, máximo = 300)
  - ordenar: el orden de los trabajos. (predeterminado = «ExecutionID»). «ExecutionID» es actualmente el único valor admitido
  - estado: (opcional) Si está presente, filtrará el estado.
- Devuelve: una lista de todos los trabajos programados, en forma de cadena JSON. Para ver un ejemplo de respuesta, consulte [Lista de respuestas a los trabajos programados](#).

Liberación de todos los trabajos que están “en espera”

- Método compatible: POST
- Ruta: /release-all
- Devuelve: un mensaje que indica el resultado de la operación de intento de liberación. Aquí hay dos posibles casos:
  - HTTP 200 y el mensaje "All job released with success!" si todos los trabajos se han publicado correctamente.
  - HTTP 503 y un mensaje "Jobs not released. An unknown error occurred. See log for more details" si algo ha salido mal en el intento de publicación.

Liberación de todos los trabajos que estén “en espera” de un nombre de trabajo determinado

Para un nombre de trabajo determinado, se pueden enviar varios trabajos con diferentes números de trabajo (la unicidad de una ejecución de trabajo viene determinada por un par <job name, job number>). El punto de conexión intentará publicar todos los trabajos presentados con el nombre indicado, que estén “en espera”.

- Método compatible: POST
- Ruta: /release/{name}
- Argumentos: el nombre del trabajo que se va a buscar, en forma de cadena. Obligatorio.
- Devuelve: un mensaje que indica el resultado de la operación de intento de liberación. Aquí hay dos posibles casos:
  - HTTP 200 y el mensaje "Jobs in group <nombre> (<número de trabajos publicados>) released with success!" si los trabajos se han publicado correctamente.

- HTTP 503 y un mensaje "Jobs in group <nombre> not released. An unknown error occurred. See log for more details" si algo ha salido mal en el intento de publicación.

## Liberación de un trabajo determinado de un número de trabajo

El punto de conexión intentará liberar la publicación de trabajo única que está "en espera" para el par en cuestión <job name, job number>.

- Método compatible: POST
- Ruta: /release/{name}/{number}
- Argumentos:

name

el nombre del trabajo que se va a buscar, en forma de cadena. Obligatorio.

number

el número de trabajo que se va a buscar, como un entero. Obligatorio.

returns

un mensaje que indica la salida de la operación de intento de liberación. Aquí hay dos posibles casos:

- HTTP 200 y el mensaje ""Job <nombre/número> released with success!" si el trabajo se ha publicado correctamente.
- HTTP 503 y el mensaje "Job <nombre/número> not released. An unknown error occurred. See log for more details" si algo ha salido mal en el intento de publicación.

## Envíe un trabajo siguiendo un cronograma repetido

Programar un trabajo que se ejecutará con un programa repetido.

- Método compatible: POST
- Ruta: /schedule
- Argumento: el cuerpo de la solicitud debe contener una serialización JSON de un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` objeto.

## Enumere todos los trabajos repetitivos enviados

- Método compatible: GET
- Ruta: `/schedule/list?status={status}&size={size}&page={page}&sort={sort}`
- Argumentos:
  1. página: número de página que se va a recuperar (predeterminado = 1)
  2. tamaño: tamaño de la página (predeterminado = 50, máximo = 300)
  3. ordenar: el orden de los trabajos. (predeterminado = «id»). «id» es el único valor admitido por ahora.
  4. estado: (opcional) Si está presente, filtrará el estado. Los valores posibles son los que se mencionan en la sección 1.
  5. estado: (opcional) Si está presente, filtrará el estado. Los valores posibles son los que se mencionan en la sección 1.
  6. Devuelve: una lista de todos los trabajos programados, en forma de cadena JSON.

## Cancela la programación de un trabajo repetido

Elimina un trabajo que se creó según un programa de repetición. El estado de la programación del trabajo está establecido en INACTIVO.

- Método compatible: GET
- Ruta: `/schedule/remove/{schedule_id}`
- Argumento: `schedule_id`, el identificador del trabajo programado que se va a eliminar.

## Puntos de conexión REST de la consola de aplicaciones de Blusam

La consola de aplicaciones de Blusam es una API diseñada para simplificar la administración de conjuntos de datos de VSAM modernizados. Los puntos de conexión de la aplicación web de Blusam utilizan la ruta raíz `/bac`.

### Temas

- [Puntos de conexión relacionados con conjuntos de datos](#)
- [Puntos de conexión relacionados con conjuntos de datos en bloque](#)
- [Registros](#)

- [Máscaras](#)
- [Otro](#)
- [Usuarios](#)

## Puntos de conexión relacionados con conjuntos de datos

Utilice los siguientes puntos de conexión para crear o administrar un conjunto de datos específico.

### Temas

- [Creación de un conjunto de datos](#)
- [Carga de un archivo](#)
- [Cargue un conjunto de datos \(POST\)](#)
- [Cargue un conjunto de datos \(GET\)](#)
- [Puede cargar un conjunto de datos desde un bucket de Amazon S3.](#)
- [Exportar el conjunto de datos a un bucket de Amazon S3.](#)
- [Borrar un conjunto de datos](#)
- [Eliminar un conjunto de datos](#)
- [Contar los registros del conjunto de datos](#)

### Creación de un conjunto de datos

Puede usar este punto final para crear una definición de conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: `/api/services/rest/bluesamservice/createDataSet`
- Argumentos:

name

(obligatorio, cadena): el nombre del conjunto de datos.

type

(obligatorio, cadena): el tipo de conjunto de datos. Los valores posibles son ESDS, KSDS, RRDS.

**recordSize**

(opcional, cadena): tamaño máximo de cada registro del conjunto de datos.

**fixedLength**

(opcional, booleano): indica si la longitud de los registros es fija.

**compression**

(opcional, booleano): indica si el conjunto de datos está comprimido.

**cacheEnable**

(opcional, booleano): indica si el almacenamiento en caché está habilitado para el conjunto de datos.

**alternativeKeys**

(opcional, lista de claves):

- offset (obligatorio, número)
  - length (obligatorio, número)
  - name (obligatorio, número)
- Devuelve un archivo json que representa el conjunto de datos recién creado.

**Solicitud de ejemplo:**

```
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  
}
```

### Respuesta de ejemplo:

```
{  
  "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  
}
```

## Carga de un archivo

Puede usar este punto final para cargar archivos al servidor. El archivo se almacena en una carpeta temporal que corresponde a cada usuario específico. Utilice este punto final cada vez que necesite cargar un archivo.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/upload
- Argumentos:  
archivo

(obligatorio, datos de varias partes/formulario): el archivo que se va a cargar.

- Devuelve un valor booleano que refleja el estado de la carga

## Cargue un conjunto de datos (POST)

Después de `createDataSet` crear la definición del conjunto de datos, puede cargar los registros asociados al archivo cargado en un conjunto de datos específico.

- Método compatible: POST
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/loadDataSet
- Argumentos:  
name

(obligatorio, cadena): el nombre del conjunto de datos.

- Devuelve el estado de la solicitud y del conjunto de datos cargado.

## Cargue un conjunto de datos (GET)

- Método compatible: GET
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/loadDataSet

- Argumentos:

name

(obligatorio, cadena): el nombre del conjunto de datos.

archivo de conjunto de datos

(obligatorio, cadena): el nombre del archivo del conjunto de datos.

- Devuelve el estado de la solicitud y del conjunto de datos cargado.

Puede cargar un conjunto de datos desde un bucket de Amazon S3.

Carga un conjunto de datos mediante un archivo listcat de un bucket de Amazon S3.

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/loadDataSetFromS3

- Argumentos:

listcatFileS3Location

(obligatorio, cadena): la ubicación de Amazon S3 del archivo listcat.

datasetFileS3Location

(obligatorio, cadena): la ubicación de Amazon S3 del archivo de conjunto de datos.

región

(obligatorio, cadena): el Amazon S3 Región de AWS donde se almacenan los archivos.

- Devuelve el conjunto de datos recién creado

Solicitud de ejemplo:

```
/BAC/api/services/rest/bluesamservice/loadDataSetFromS3?region=us-east-1&listcatFileS3Location=s3://bucket-name/listcat.json&datasetFileS3Location=s3://bucket-name/dataset.DAT
```

Exportar el conjunto de datos a un bucket de Amazon S3.

Exporta un conjunto de datos al bucket de Amazon S3 especificado.

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/exportDataSetToS3
- Argumentos:
  - s3Location

(obligatorio, cadena): la ubicación de Amazon S3 a la que exportar el conjunto de datos.

datasetName

(obligatorio, cadena): el nombre del conjunto de datos que se va a exportar.

región

(obligatorio, cadena): el Región de AWS del bucket de Amazon S3.

- Devuelve el conjunto de datos exportado

Solicitud de ejemplo:

```
/BAC/api/services/rest/bluesamservice/exportDataSetToS3?region=eu-west-1&s3Location=s3://bucket-name/dump&datasetName=dataset
```

Borrar un conjunto de datos

Borra todos los registros de un conjunto de datos.

- Métodos compatibles: POST, GET
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/clearDataSet
- Argumentos:

## name

(obligatorio, cadena): el nombre del conjunto de datos que se va a borrar.

- Devuelve: el estado de la solicitud.

## Eliminar un conjunto de datos

Elimina la definición y los registros del conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/deleteDataSet
- Argumentos:

## name

(obligatorio, cadena): el nombre del conjunto de datos que se va a eliminar.

- Devuelve el estado de la solicitud y del conjunto de datos eliminado.

## Contar los registros del conjunto de datos

Este punto final devuelve el número de registros asociados a un conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/countRecords
- Argumentos:

## name

(obligatorio, cadena): el nombre del conjunto de datos.

- Devuelve: el número de registros

## Puntos de conexión relacionados con conjuntos de datos en bloque

Utilice los siguientes puntos de conexión para crear o administrar varios conjuntos de datos a la vez.

## Temas

- [Exportar conjuntos de datos \(GET\)](#)
- [Exportar conjuntos de datos \(POST\)](#)
- [Creación de varios conjuntos de datos](#)
- [Lista de todos los conjuntos de datos](#)
- [Enumere directamente todos los conjuntos de datos](#)
- [Enumere directamente todos los conjuntos de datos por página](#)
- [Transmita el conjunto de datos](#)
- [Eliminación de todos los conjuntos de datos](#)
- [Obtenga las definiciones de los conjuntos de datos del archivo list.cat](#)
- [Obtener las definiciones de conjuntos de datos del archivo listcat cargado](#)
- [Obtenga un conjunto de datos](#)
- [Cargue listcat desde un archivo JSON](#)

## Exportar conjuntos de datos (GET)

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/exportDataSet
- Argumentos:  
datasetName

(obligatorio, cadena): el nombre del conjunto de datos que se va a exportar.

conjunto de datos OutputFile

(obligatorio, cadena): la ruta de la carpeta en la que desea almacenar el conjunto de datos exportado en el servidor.

rdw

(obligatorio, booleano): si desea que la palabra descriptora del registro (RDW) forme parte de los registros exportados. Si el conjunto de datos tiene registros de longitud fija, se omite el valor de este parámetro.

- Devuelve el estado de la solicitud y la ruta al archivo que contiene el conjunto de datos exportado (si lo hubiera). Si el conjunto de datos es nulo en la respuesta, significa que el sistema no ha podido localizar un conjunto de datos con el nombre indicado.

## Exportar conjuntos de datos (POST)

- Método compatible: POST
- Requiere autenticación y la función ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/exportDataSet

- Argumentos:

Parámetros de volcado

(obligatorio, BACReadParameters): parámetros de lectura de Bluesam.

- Devuelve el estado del conjunto de datos exportado.

## Creación de varios conjuntos de datos

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/createAllDataSets

- Argumentos:

- Lista de conjuntos de datos

name

(obligatorio, cadena): el nombre del conjunto de datos.

type

(obligatorio, cadena): el tipo de conjunto de datos. Los valores posibles son ESDS, KSDS, RRDS.

recordSize

(opcional, cadena): tamaño máximo de cada registro del conjunto de datos.

fixedLength

(opcional, booleano): indica si la longitud de los registros es fija.

compression

(opcional, booleano): indica si el conjunto de datos está comprimido.

## cacheEnable

(opcional, booleano): indica si el almacenamiento en caché está habilitado para el conjunto de datos.

- Devuelve: el estado de la solicitud y el conjunto de datos recién creado.

## Lista de todos los conjuntos de datos

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/listDataSet
- Argumentos: ninguno
- Devuelve el estado de la solicitud y de la lista de conjuntos de datos.

## Enumere directamente todos los conjuntos de datos

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/directListDataSet
- Argumentos: ninguno
- Devuelve el estado de la solicitud y de la lista de conjuntos de datos.

## Enumere directamente todos los conjuntos de datos por página

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/directListDataSetByPage
- Argumentos:

### datasetName

(obligatorio, cadena): el nombre del conjunto de datos.

### Número de página

(obligatorio, int): el número de página.



## pageSize

(obligatorio, int): el tamaño de la página.

- Devuelve el estado de la solicitud y de la lista de conjuntos de datos.

## Transmita el conjunto de datos

- Método compatible: GET
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/streamDataset
- Argumentos:

### datasetName

(obligatorio, cadena): el nombre del conjunto de datos.

- Devuelve: un flujo de los conjuntos de datos solicitados.

## Eliminación de todos los conjuntos de datos

- Método compatible: POST
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/removeAll
- Argumentos: ninguno
- Devuelve: un valor booleano que representa el estado de la solicitud.

## Obtenga las definiciones de los conjuntos de datos del archivo list.cat

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/getDataSetsDefinitionFromListcat
- Argumentos:

### parámetro FilePath

(obligatorio, cadena): la ruta al archivo listcat.

- Devuelve: una lista de conjuntos de datos

## Obtener las definiciones de conjuntos de datos del archivo listcat cargado

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: `/api/services/rest/bluesamservice/getDataSetsDefinitionFromUploadedListcat`
- Argumentos: ninguno
- Devuelve: una lista de conjuntos de datos

## Obtenga un conjunto de datos

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: `/api/services/rest/bluesamservice/getDataSet`
- Argumentos:
  - name  
  
(obligatorio, cadena): el nombre del conjunto de datos.
- Devuelve el conjunto de datos solicitado.

## Cargue listcat desde un archivo JSON

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: `/api/services/rest/bluesamservice/loadListcatFromJsonFile`
- Argumentos:
  - filePath  
  
(obligatorio, cadena): la ruta al archivo listcat.
- Devuelve: una lista de conjuntos de datos

## Registros

Utilice los siguientes puntos de conexión para crear o administrar registros en un conjunto de datos.

## Temas

- [Creación de un registro](#)
- [Lectura de un conjunto de datos](#)
- [Eliminación de un registro](#)
- [Actualización de un registro](#)
- [Guardar un registro](#)
- [Validar un registro](#)
- [Obtenga un árbol de registros](#)

### Creación de un registro

Puede usar este punto final para crear un registro nuevo.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/createRecord
- Argumentos:  
conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

mask

(obligatorio, mask): el objeto de máscara.

- Devuelve: el estado de la solicitud y el registro creado.

### Lectura de un conjunto de datos

Puede utilizar este punto final para leer un conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/readDataSet
- Argumentos:

## conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos.

- Devuelve el estado de la solicitud y del conjunto de datos con los registros.

## Eliminación de un registro

Puede utilizar este punto final para eliminar un registro de un conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/deleteRecord
- Argumentos:

### conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

### record

(obligatorio, Record): el registro que se va a eliminar

- Devuelve el estado de la eliminación.

## Actualización de un registro

Puede utilizar este punto final para actualizar un registro asociado a un conjunto de datos.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/updateRecord
- Argumentos:

### conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

### record

(obligatorio, Record): el registro que se va a actualizar

- Devuelve el estado de la solicitud y del conjunto de datos con los registros.

## Guardar un registro

Puede usar este punto final para guardar un registro en un conjunto de datos y usar una máscara.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: `/api/services/rest/crud/saveRecord`
- Argumentos:

conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

record

(obligatorio, Record): el registro que se va a guardar

- Devuelve el estado de la solicitud y del conjunto de datos con los registros.

## Validar un registro

Utilice este punto final para validar un registro.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: `/api/services/rest/crud/validateRecord`
- Argumentos:

conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

- Devuelve el estado de la solicitud y del conjunto de datos con los registros.

## Obtenga un árbol de registros

Utilice este punto final para obtener el árbol jerárquico de un registro.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: `/api/services/rest/crud/getRecordTree`

- Argumentos:

conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

record

(obligatorio, registro): el registro a buscar

- Devuelve el estado de la solicitud y el árbol jerárquico del registro solicitado.

## Máscaras

Utilice los siguientes puntos de conexión para cargar o aplicar máscaras a un conjunto de datos.

### Temas

- [Cargar máscaras](#)
- [Aplicar máscara](#)
- [Aplicar filtro de máscara](#)

### Cargar máscaras

Puede utilizar este punto final para recuperar todas las máscaras asociadas a un conjunto de datos específico.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: `/api/services/rest/crud/loadMasks`
- Argumentos:  
conjunto de datos

(obligatorio, DataSet): el objeto del conjunto de datos

- Devuelve el estado de la solicitud y de la lista de máscaras.

### Aplicar máscara

Puede utilizar este punto final para aplicar una máscara a un conjunto de datos específico.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/applyMask
- Argumentos:
  - conjunto de datos
    - (obligatorio, DataSet): el objeto del conjunto de datos
  - mask
    - (obligatorio, máscara): el objeto del conjunto de datos
- Devuelve el estado de la solicitud y el conjunto de datos con la máscara aplicada.

### Aplicar filtro de máscara

Puede utilizar este punto final para aplicar una máscara y un filtro a un conjunto de datos específico.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/crud/applyMaskFilter
- Argumentos:
  - conjunto de datos
    - (obligatorio, DataSet): el objeto del conjunto de datos
  - mask
    - (obligatorio, máscara): el objeto del conjunto de datos
- Devuelve el estado de la solicitud y el conjunto de datos con la máscara y el filtro aplicados.

### Otro

Utilice los siguientes puntos de conexión para administrar la caché de un conjunto de datos o comprobar las características del conjunto de datos

### Temas

- [Comprobación de la memoria caché de preparación](#)
- [Comprobar que la caché está habilitada](#)

- [Habilitar caché](#)
- [Compruebe la memoria RAM asignada](#)
- [Comprobar la persistencia](#)
- [Comprobar los tipos de conjuntos de datos compatibles](#)
- [Comprobar el estado del servidor](#)

### Comprobación de la memoria caché de preparación

Comprueba si la caché de preparación está habilitada para un conjunto de datos específico.

- Método compatible: POST
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/warmupCache
- Argumentos:  
name  
  
(obligatorio, cadena): el nombre del conjunto de datos.
- Devuelve: true si la memoria caché de preparación está habilitada y false en caso contrario.

### Comprobar que la caché está habilitada

Comprueba si la caché está habilitada para un conjunto de datos específico.

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/isEnableCache
- Argumentos: ninguno
- Devuelve true si el almacenamiento en caché está habilitado.

### Habilitar caché

- Método compatible: GET
- Requiere autenticación y las funciones ROLE\_ADMIN y ROLE\_SUPER\_ADMIN.
- Ruta: /api/services/rest/bluesamservice/enableDisableCache/{enable}
- Argumentos:



## enable

(obligatorio, booleano): si se establece en true, habilitará el almacenamiento en caché.

- Devuelve: nada

## Compruebe la memoria RAM asignada

Puede utilizar este punto final para recuperar la memoria caché RAM asignada.

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/allocatedRamCache
- Argumentos: ninguno
- Devuelve: el tamaño de la memoria en forma de cadena

## Comprobar la persistencia

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_USER.
- Ruta: /api/services/rest/bluesamservice/persistence
- Argumentos: ninguno
- Devuelve: la persistencia utilizada como cadena

## Comprobar los tipos de conjuntos de datos compatibles

- Método compatible: GET
- Ruta: /api/services/rest/bluesamservice/getDataSetTypes
- Requiere autenticación y el rol ROLE\_USER.
- Argumentos: ninguno
- Devuelve: la lista de tipos de conjuntos de datos admitidos en forma de lista de cadenas.

## Comprobar el estado del servidor

- Método compatible: GET

- Ruta: `/api/services/rest/bluesamserver/serverIsUp`
- Argumentos: ninguno
- Devoluciones: ninguna. El código de estado de respuesta HTTP 200 indica que el servidor está en funcionamiento.

## Usuarios

Utilice los siguientes puntos de conexión para administrar las interacciones de los usuarios.

### Temas

- [Login \(Iniciar sesión\)](#)
- [Comprobar la cuenta de usuario](#)
- [Inicio de sesión](#)
- [Obtenga información de usuario](#)
- [Lista de todos los usuarios](#)
- [Delete user \(Eliminar usuario\)](#)
- [Cerrar sesión](#)

### Login (Iniciar sesión)

- Método compatible: POST
- Ruta: `/api/services/security/servicelogin/login`
- Argumentos:
  - username
    - (obligatorio, cadena)
  - password
    - (obligatorio, cadena)
- Devuelve el nombre de usuario y los roles del usuario que ha iniciado la sesión

### Respuesta de ejemplo

```
{"login":"some-user","roles":[{"id":0,"roleName":"ROLE_ADMIN"}]}
```

## Comprobar la cuenta de usuario

- Método compatible: POST
- Ruta: `/api/services/security/servicelogin/hasAccount`
- Argumentos: ninguno
- Devuelve: true si el usuario ya ha iniciado sesión

## Inicio de sesión

- Método compatible: POST
- Requiere autenticación y la función ROLE\_ADMIN.
- Ruta: `/api/services/security/servicelogin/recorduser`
- Argumentos: ninguno
- Devuelve: true si el usuario ya ha iniciado sesión

## Obtenga información de usuario

- Método compatible: GET
- Ruta: `/api/services/security/servicelogin/userInfo`
- Argumentos: ninguno
- Devuelve: el nombre de usuario y el rol del usuario actualmente conectado

## Lista de todos los usuarios

- Método compatible: GET
- Requiere autenticación y el rol ROLE\_ADMIN.
- Ruta: `/api/services/security/servicelogin/listusers`
- Argumentos: ninguno
- Devuelve: la lista de todos los usuarios

## Delete user (Eliminar usuario)

- Método compatible: POST
- Ruta: `/api/services/security/servicelogin/deleteuser`

- Argumentos:

login

(obligatorio, cadena)

- Devuelve: verdadero si el usuario se ha eliminado correctamente

#### Cerrar sesión

- Método compatible: POST
- Ruta: `/api/services/security/service/logout/logout`
- Argumentos: ninguno
- Devuelve: true si la sesión del usuario se ha cerrado correctamente.

## Consola de la aplicación JICS

El componente JICS es el soporte de la era AWS azul para la modernización de los recursos antiguos del CICS. La aplicación web JICS Application Console está dedicada a administrar los recursos del JICS. Los siguientes puntos de conexión permiten realizar las tareas de administración sin tener que interactuar con la interfaz de usuario de JAC. Siempre que un punto de conexión requiera autenticación, la solicitud deberá incluir los detalles de autenticación (normalmente, nombre de usuario/contraseña, tal y como exige la autenticación básica). Los puntos finales de la aplicación web JICS Application Console utilizan la ruta raíz. `/jac/`

#### Temas

- [Administración de recursos del JICS](#)
- [Otro](#)
- [Puntos de conexión de administración de usuarios de JAC](#)

## Administración de recursos del JICS

Todos los puntos de conexión siguientes están relacionados con la administración de los recursos del JICS, lo que permite a los administradores del JICS gestionar los recursos a diario.

#### Temas

- [Muestre las listas y los grupos del JICS](#)

- [Recupera los recursos de JICS](#)
- [Lista los grupos de JICS](#)
- [Lista de grupos de JICS para una lista determinada](#)
- [Lista de recursos del JICS para un grupo determinado](#)
- [Lista de los recursos del JICS para un grupo determinado \(también puede usar un nombre\)](#)
- [Edición de los grupos propios de varias listas](#)
- [Eliminar una lista](#)
- [Eliminación de un grupo](#)
- [Eliminar una transacción](#)
- [Eliminar un programa](#)
- [Eliminar un archivo](#)
- [Eliminar un objeto TDQUEUE](#)
- [Eliminar un objeto TSMODEL](#)
- [Eliminar elementos](#)
- [Crear una lista](#)
- [Crear un grupo](#)
- [Consideraciones comunes sobre la creación de recursos](#)
- [Crear una transacción](#)
- [Crear un programa](#)
- [Crear un archivo](#)
- [Crear un objeto TDQUEUE](#)
- [Crear un objeto TSMODEL.](#)
- [Crea elementos](#)
- [Actualizar una lista](#)
- [Actualizar un grupo](#)
- [Consideraciones sobre la actualización de los recursos comunes](#)
- [Actualizar una transacción](#)
- [Actualizar un programa](#)
- [Actualizar un archivo](#)
- [Actualice un objeto TDQUEUE](#)

- [Actualizar un objeto TSMODEL](#)
- [Elementos de actualización](#)
- [Elementos alterados](#)
- [Recupera elementos](#)
- [Operación CRUD del JICS](#)

Muestre las listas y los grupos del JICS

La lista y los grupos son los principales recursos de contenedores propietarios del componente JICS. Todos los recursos del JICS deben pertenecer a un grupo. Los grupos pueden pertenecer a listas, pero no es obligatorio. Es posible que las listas ni siquiera existan en un entorno JICS determinado, pero la mayoría de las veces, las listas están ahí para ofrecer una capa adicional de organización de los recursos. Para obtener más información sobre la organización de los recursos del CICS, consulte los [recursos del CICS](#).

- Método compatible: GET
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/listJicsListsAndGroups
- Argumentos: ninguno
- Devuelve: una lista de objetos serializados, tanto LISTAS como GRUPOS, en formato JSON.  
JicsContainer

Respuesta de ejemplo:

```
[
  {
    "name": "Resources",
    "children": [
      {
        "jacType": "JAList",
        "name": "MURACHS",
        "isActive": true,
        "children": [
          {
            "jacType": "JACGroup",
            "name": "MURACHS",
```

```

        "isActive": true,
        "children": []
    }
  ],
  {
    "jacType": "JACGroup",
    "name": "TEST",
    "isActive": true,
    "children": []
  }
],
"isExpanded": true
}
]

```

### Recupera los recursos de JICS

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/retrieveJicsResources`
- Argumentos: una carga útil de JSON que representa los recursos de JICS que desea recuperar. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.request.RetrieveOperationRequest`.
- Devuelve: una lista de objetos `JicsResource` serializados. Los objetos se devuelven sin ningún orden en particular y son de diferentes tipos, como PROGRAM, TRANSACTION, FILE, etc.

### Lista los grupos de JICS

- Método compatible: GET
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/listJicsGroups`
- Devuelve: una lista de objetos serializados (GRUPOS) en formato JSON. `JicsContainer` Los grupos se devuelven sin su propia información de lista.

### Respuesta de ejemplo:

```
[
  {
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  },
  {
    "jacType": "JACGroup",
    "name": "TEST",
    "isActive": true,
    "children": []
  }
]
```

### Lista de grupos de JICS para una lista determinada

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/listGroupsForList
- Argumentos: una carga de JSON, que representa la lista de JICS cuyos grupos estamos buscando. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACList`.

### Solicitud de ejemplo:

```
{
  "jacType": "JACList",
  "name": "MURACHS",
  "isActive": true
}
```

- Devuelve: una lista de JicsContainer objetos serializados (GRUPOS) en formato JSON, que se adjuntan a la LISTA dada. Los grupos se devuelven sin su propia información de lista.

### Respuesta de ejemplo:

```
[
  {
```



```
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  }
]
```

### Lista de recursos del JICS para un grupo determinado

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/listResourcesForGroup
- Argumentos: una carga de JSON, que representa el grupo de JICS cuyos grupos estamos buscando. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACGroup`. No es necesario especificar todos los campos para el grupo, pero el nombre sí es obligatorio.

Solicitud de ejemplo:

```
{
  "jacType": "JACGroup",
  "name": "MURACHS",
  "isActive": true
}
```

- Devuelve: una lista de objetos serializados, propiedad del GRUPO en cuestión. `JicsResource` Los objetos se devuelven sin ningún orden en particular y son de diferentes tipos (PROGRAM, TRANSACTION, FILE, etc.).

### Lista de los recursos del JICS para un grupo determinado (también puede usar un nombre)

- Método compatible: POST
- Requiere autenticación
- Ruta: /api/services/rest/jicsservice/listResourcesForGroupName
- Argumentos: el nombre del grupo propietario de los recursos que estamos buscando.

- Devuelve: una lista de JicsResource objetos serializados, propiedad del GRUPO dado. Los objetos se devuelven sin ningún orden en particular y son de diferentes tipos (PROGRAM, TRANSACTION, FILE, etc.)

### Edición de los grupos propios de varias listas

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/editGroupsList
- Argumentos: una representación en JSON de una colección de listas con grupos secundarios;

Solicitud de ejemplo:

```
[
  {
    "jacType": "JACList",
    "name": "MURACHS",
    "isActive": true,
    "children": [
      {
        "jacType": "JACGroup",
        "name": "MURACHS",
        "isActive": true,
        "children": []
      },
      {
        "jacType": "JACGroup",
        "name": "TEST",
        "isActive": true,
        "children": []
      }
    ]
  }
]
```

Antes de esta edición, solo el grupo denominado “MURACHS” pertenecía a la lista denominada “MURACHS”. Con esta edición, “añadimos” el grupo denominado “TEST” a la lista denominada “MURACHS”.

- Devuelve un valor booleano. Si el valor es “true”, las modificaciones de la lista se han conservado correctamente en el almacenamiento JICS subyacente.

### Eliminar una lista

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/deleteList`
- Argumentos: una carga JSON, que representa la lista de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACList`.
- Devuelve un valor booleano. Si el valor es “true”, la eliminación de la lista se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminación de un grupo

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/deleteGroup`
- Argumentos: una carga JSON, que representa el grupo de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACGroup`.
- Devuelve un valor booleano. Si el valor es “true”, la eliminación del grupo se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar una transacción

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/deleteTransaction`
- Argumentos: una carga JSON, que representa la transacción de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACTransaction`.

- Devuelve un valor booleano. Si el valor es “true”, la eliminación de la transacción se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar un programa

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/deleteProgram
- Argumentos: una carga JSON, que representa el programa de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACProgram`.
- Devuelve un valor booleano. Si el valor es “true”, la eliminación del programa se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar un archivo

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/deleteFile
- Argumentos: una carga JSON, que representa el archivo de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACFile`.
- Devuelve un valor booleano. Si el valor es “true”, la eliminación del archivo se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar un objeto TDQUEUE

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/deleteTDQueue
- Argumentos: una carga JSON que representa el TDQUEUE del JICS que se va a eliminar. Se trata de la serialización en JSON de un objeto ``com.netfective.bluage.jac.entities.JACTDQueue``.

- Devuelve un valor booleano. Si el valor es “true”, la eliminación del objeto TDQUEUE se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar un objeto TSMODEL

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/deleteTSMoDel`
- Argumentos: una carga JSON que representa el TSMODEL del JICS que se va a eliminar. Se trata de la serialización en JSON de un objeto ``com.netfective.bluage.jac.entities.JACTSMoDel``.
- Devuelve un valor booleano. Si el valor es “true”, la eliminación del TSMODEL se ha realizado correctamente en el almacenamiento JICS subyacente.

### Eliminar elementos

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/deleteElements`
- Argumentos: carga útil de JSON que representa los elementos del JICS que se van a eliminar.
- Devuelve un valor booleano que `true` indica que la eliminación se realizó correctamente en el almacenamiento JICS subyacente.

### Crear una lista

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/createList`
- Argumentos: una carga JSON, que representa la lista de JICS que se va a eliminar. Esta es la serialización en JSON de un objeto ``com.netfective.bluage.jac.entities.JACList``.
- Devuelve un valor booleano. Si el valor es “true”, la lista se ha creado correctamente en el almacenamiento JICS subyacente.

**Note**

La lista siempre se creará vacía. Adjuntar grupos a la lista requerirá otra operación.

## Crear un grupo

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/createGroup`
- Argumentos: una carga JSON, que representa el grupo de JICS que se va a crear. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACGroup`.
- Devuelve un valor booleano. Si el valor es "true", el grupo se ha creado correctamente en el almacenamiento JICS subyacente.

**Note**

El grupo siempre se creará vacío. Adjuntar recursos al grupo requerirá operaciones adicionales (la creación de recursos los asociará automáticamente a un grupo determinado).

## Consideraciones comunes sobre la creación de recursos

Todos los puntos de conexión siguientes están relacionados con la creación de recursos de JICS y comparten algunas limitaciones comunes: en la carga de la solicitud que se va a enviar al punto de conexión, se debe valorar el campo `groupName`.

### Restricción de propiedad del grupo:

No se puede crear ningún recurso sin estar asociado a un grupo existente, y el punto de conexión utiliza el `groupName` para recuperar el grupo al que se adjuntará este recurso. El `groupName` debe apuntar al nombre de un grupo existente. Se enviará un mensaje de error con el estado HTTP 400 si el `groupName` no apunta a un grupo existente en el almacenamiento subyacente del JICS.

### Restricción de unicidad dentro de un grupo:

Un recurso determinado con un nombre determinado tiene que ser único en un grupo determinado. La comprobación de la unicidad la realizará cada punto de conexión de creación de recursos. Si la carga dada no respeta la restricción de unicidad, el punto de conexión enviará una respuesta HTTP STATUS 400 (BAD REQUEST); consulte el ejemplo de respuesta que aparece a continuación.

Ejemplo de carga: intentamos crear la transacción "ARIT" en el grupo "TEST", pero ya existe una transacción con ese nombre en este grupo.

```
{
  "jacType": "JACTransaction",
  "name": "ARIT",
  "groupName": "TEST",
  "isActive": true
}
```

Hemos recibido la siguiente respuesta de error:

```
{
  "timestamp": 1686759054510,
  "status": 400,
  "error": "Bad Request",
  "path": "/jac/api/services/rest/jicservice/createTransaction"
}
```

Al inspeccionar los registros de los servidores, se confirmará el origen del problema:

```
2023-06-14 18:10:54 default          TRACE - o.s.w.m.HandlerMethod
      - Arguments: [java.lang.IllegalArgumentException: Transaction already
present in the group, org.springframework.security.web.header.HeaderWriterFilter
$HeaderWriterResponse@e34f6b8]
2023-06-14 18:10:54 default          ERROR - c.n.b.j.a.WebConfig          -
400
java.lang.IllegalArgumentException: Transaction already present in the group
at
com.netfactive.bluage.jac.server.services.rest.impl.JicsServiceImpl.createElement(JicsServiceI
```

## Crear una transacción

- Método compatible: POST

- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/createTransaction`
- Argumentos: una carga JSON, que representa la transacción de JICS que se va a crear. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACTransaction`.
- Devuelve un valor booleano. Si el valor es “true”, la transacción se ha creado correctamente en el almacenamiento JICS subyacente.

### Crear un programa

- Método compatible: `POST`
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/createProgram`
- Argumentos: una carga JSON, que representa el programa de JICS que se va a crear. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACProgram`.
- Devuelve un valor booleano. Si el valor es “true”, el programa se ha creado correctamente en el almacenamiento JICS subyacente.

### Crear un archivo

- Método compatible: `POST`
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/createFile`
- Argumentos: una carga JSON, que representa el archivo de JICS que se va a crear. Se trata de la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACFile``.
- Devuelve un valor booleano. Si el valor es “true”, el archivo se ha creado correctamente en el almacenamiento JICS subyacente.

### Crear un objeto TDQUEUE

- Método compatible: `POST`



- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/createTDQueue`
- Argumentos: una carga JSON, que representa el objeto TDQUEUE de JICS que se va a crear. Se trata de la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACTDQueue``.
- Devuelve un valor booleano. Si el valor es “true”, el objeto TDQUEUE se ha creado correctamente en el almacenamiento JICS subyacente.

#### Crear un objeto TSMODEL.

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/createTSModel`
- Argumentos: una carga JSON, que representa el objeto TSMODEL de JICS que se va a crear. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACTSModel`.
- Devuelve un valor booleano que true indica que la creación de elementos se realizó correctamente en el almacenamiento JICS subyacente.

#### Crea elementos

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: `/api/services/rest/jicsservice/createElements`
- Argumentos: una carga útil de JSON que representa los elementos del JICS que se van a crear.
- Devuelve un valor booleano. Si el valor es “true”, el objeto TSMODEL se ha creado correctamente en el almacenamiento JICS subyacente.

#### Actualizar una lista

- Método compatible: POST

- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateList`
- Argumentos: una carga JSON, que representa la lista de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACList`. No es necesario proporcionar los elementos secundarios de la LISTA, ya que el mecanismo de actualización de la LISTA no lo tendrá en cuenta.
- Devuelve un valor booleano. Si el valor es “true”, la lista se ha actualizado correctamente en el almacenamiento JICS subyacente.

Si se actualiza el indicador “isActive” de la lista, se propagará a todos los elementos que sean propiedad de la lista, es decir, a todos los grupos que sean propiedad de la lista y a todos los recursos que pertenezcan a dichos grupos. Esta es una forma cómoda de desactivar una gran cantidad de recursos con una sola operación, en varios grupos.

#### Actualizar un grupo

- Método compatible: `POST`
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateGroup`
- Argumentos: una carga JSON, que representa el grupo de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACGroup`. No es necesario proporcionar los elementos secundarios del GRUPO, ya que el mecanismo de actualización del GRUPO no lo tendrá en cuenta.
- Devuelve un valor booleano. Si el valor es “true”, el grupo se ha actualizado correctamente en el almacenamiento JICS subyacente.

#### Note

Si se actualiza el indicador “isActive” del grupo, se propagará a todos los elementos que sean propiedad del grupo, es decir, a todos los recursos que pertenezcan al grupo. Esta es una forma cómoda de desactivar muchos recursos con una sola operación dentro de un GRUPO determinado.

## Consideraciones sobre la actualización de los recursos comunes

Los siguientes puntos de conexión están relacionados con la actualización de recursos de JICS. Con `groupName` este campo, puede cambiar el GRUPO propietario de cualquier RECURSO DE JICS, siempre que el valor del campo apunte a un GRUPO existente en el almacenamiento JICS subyacente (de lo contrario, recibirá una RESPUESTA DE SOLICITUD ERRÓNEA (HTTP STATUS 400) del punto final).

### Actualizar una transacción

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateTransaction`
- Argumentos: una carga JSON, que representa la transacción de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACTransaction`.
- Devuelve un valor booleano. Si el valor es “true”, la transacción se ha actualizado correctamente en el almacenamiento JICS subyacente.

### Actualizar un programa

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateProgram`
- Argumentos: una carga JSON, que representa el programa de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.JACProgram`.
- Devuelve un valor booleano. Si el valor es “true”, el programa se ha actualizado correctamente en el almacenamiento JICS subyacente.

### Actualizar un archivo

- Método compatible: POST

- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateFile`
- Argumentos: una carga JSON, que representa el archivo de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACFile`.
- Devuelve un valor booleano. Si el valor es “true”, el archivo se ha actualizado correctamente en el almacenamiento JICS subyacente.

### Actualice un objeto TDQUEUE

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateTDQueue`
- Argumentos: una carga JSON, que representa el objeto TDQUEUE de JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACTDQueue`.
- Devuelve un valor booleano. Si el valor es “true”, el objeto TDQUEUE se ha actualizado correctamente en el almacenamiento JICS subyacente.

### Actualizar un objeto TSMODEL

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Ruta: `/api/services/rest/jicsservice/updateTSModel`
- Argumentos: una carga útil de JSON, que representa el TSMODEL del JICS que se va a actualizar. Esta es la serialización en JSON de un objeto `com.netfactive.bluage.jac.entities.JACTSModel`.
- Devuelve un valor booleano. Si el valor es “true”, el objeto TSMODEL se ha actualizado correctamente en el almacenamiento JICS subyacente.

## Elementos de actualización

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/updateElements
- Argumentos: carga útil de JSON que representa los elementos que se van a actualizar.
- Devuelve un valor booleano que true indica que la actualización de los elementos se realizó correctamente en el almacenamiento JICS subyacente.

## Elementos alterados

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/upsertElements
- Argumentos: una carga útil de JSON que representa los elementos que se van a utilizar.
- Devuelve un valor booleano que true indica que la configuración de los elementos se ha utilizado correctamente en el almacenamiento JICS subyacente.

## Recupera elementos

- Método compatible: GET
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Ruta: /api/services/rest/jicsservice/retrieveElements
- Argumentos: ninguno
- Devuelve una lista de todos los recursos JICS serializados.

## Operación CRUD del JICS

- Método compatible: POST
- Requiere autenticación y las siguientes funciones: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER

- Ruta: `/api/services/rest/jicsservice/jicsCrudOperation`
- Argumentos: una carga útil de JSON que representa los recursos de JICS que estamos buscando. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.request.JicsCrudOperationRequest`.
- Devuelve una carga útil JSON que representa la respuesta. Esta es la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.request.JicsCrudOperationResponse`.

## Otro

### Temas

- [Estado de salud del servidor JICS](#)

### Estado de salud del servidor JICS

- Método compatible: GET
- Ruta: `/api/services/rest/jicsserver/serverIsUp`
- Argumentos: ninguno
- Devoluciones: ninguna. Una respuesta HTTP STATUS 200 indica que el servidor está en funcionamiento.

## Puntos de conexión de administración de usuarios de JAC

Utilice los siguientes puntos de conexión para administrar las interacciones de los usuarios.

### Temas

- [Registrar a un usuario](#)
- [Probar si existe al menos un usuario en el sistema](#)
- [Grabar a un usuario nuevo](#)
- [Información de usuario](#)
- [Mostrar usuarios](#)
- [Eliminar un usuario](#)
- [Cierre de sesión del usuario actual](#)

## Registrar a un usuario .

- Método compatible: POST
- Ruta: `/api/services/security/servicelogin/login`
- Argumentos: ninguno
- Devuelve la serialización JSON de un objeto `com.netfactive.bluage.jac.entities.SignOn`, que representa al usuario cuyas credenciales se proporcionan en la solicitud actual. La contraseña está oculta en la vista del objeto devuelto. Se muestran las funciones asignadas al usuario.

## Respuesta de ejemplo:

```
{
  "login": "some-admin",
  "password": null,
  "roles": [
    {
      "id": 0,
      "roleName": "ROLE_ADMIN"
    }
  ]
}
```

## Probar si existe al menos un usuario en el sistema

- Método compatible: GET
- Ruta: `/api/services/security/servicelogin/hasAccount`
- Argumentos: ninguno
- Devuelve el valor booleano `true` si se ha creado al menos un usuario distinto del superadministrador predeterminado. Devuelve `false` lo contrario.

## Grabar a un usuario nuevo

- Método compatible: POST
- Requiere autenticación y la función `ROLE_ADMIN`.
- Ruta: `/api/services/security/servicelogin/recorduser`

- Argumentos: la serialización en JSON de un objeto `com.netfective.bluage.jac.entities.SignOn`, que representa al usuario que se va a añadir al almacenamiento. Se deben definir los roles del usuario; de lo contrario, es posible que el usuario no pueda utilizar las funciones y los puntos de conexión del JAC.
- Devuelve el valor booleano `true` si el usuario se creó correctamente. Devuelve `false` lo contrario.

Solicitud de ejemplo:

```
{
  "login": "simpleuser",
  "password": "simplepassword",
  "roles": [
    {
      "id": 2,
      "roleName": "ROLE_USER"
    }
  ]
}
```

Al grabar a un usuario nuevo, solo se pueden usar los siguientes roles:

- `ROLE_ADMIN`: puede administrar los recursos y usuarios de JICS.
- `ROLE_USER`: puede administrar los recursos del JICS pero no los usuarios.

Información de usuario

- Método compatible: `GET`
- Requiere autenticación y derechos de administrador
- Ruta: `/api/services/security/servicelogin/userInfo`
- Devuelve el nombre de usuario y las funciones del usuario actualmente conectado.

Mostrar usuarios

- Método compatible: `GET`
- Requiere autenticación y el rol `ROLE_ADMIN`.
- Ruta: `/api/services/security/servicelogin/listusers`



- Argumentos: ninguno
- Devuelve una lista de `com.netfactive.bluage.jac.entities.SignOn`, serializada como JSON.

### Eliminar un usuario

- Método compatible: POST
- Ruta: `/api/services/security/servicelogin/deleteuser`
- Argumentos: la serialización en JSON de un `com.netfactive.bluage.jac.entities.SignOn` objeto que representa al usuario que se va a eliminar del almacenamiento.
- Devuelve el valor booleano `true` si el usuario se ha eliminado correctamente.

#### Important

Esta acción no se puede deshacer. El usuario eliminado no podrá volver a conectarse a la aplicación JAC.

### Cierre de sesión del usuario actual

- Método compatible: GET
- Ruta: `/api/services/security/servicelogout/logout`
- Argumentos: ninguno
- Devuelve el mensaje JSON `{"success": true}` si la sesión del usuario actual se cerró correctamente. La sesión HTTP relacionada se invalidará.

## Estructuras de base de datos

En esta sección se describen los detalles de las distintas estructuras de datos.

### Temas

- [Estructura de mensajes de detalles de ejecución del trabajo](#)
- [Estructura de resultados del lanzamiento de la transacción](#)
- [Estructura de resultados del registro de lanzamiento de la transacción](#)

- [Los posibles estados de un trabajo en una cola son:](#)
- [Envíe el trabajo y programe la entrada del trabajo](#)
- [Lista de respuestas a los trabajos programados](#)
- [Respuesta a la lista de trabajos repetidos](#)

## Estructura de mensajes de detalles de ejecución del trabajo

Los detalles de la ejecución de cada trabajo tendrán los siguientes campos:

### ScriptId

el identificador del script llamado.

### caller

dirección IP del programa que llama.

### identifier

identificador único de ejecución del trabajo.

### startTime

la fecha y hora de inicio de la ejecución del trabajo.

### endTime

la fecha y hora de fin de la ejecución del trabajo.

### estado

un estado de la ejecución del trabajo. Un valor posible entre:

- DONE: la ejecución del trabajo ha finalizado con normalidad.
- TRIGGERED: la ejecución del trabajo se ha activado pero aún no se ha iniciado.
- RUNNING: la ejecución del trabajo se está ejecutando.
- KILLED: la ejecución del trabajo ha sido cancelada.
- FAILED: la ejecución del trabajo ha fallado.

### executionResult

un mensaje para resumir el resultado de la ejecución del trabajo. Este mensaje puede ser un mensaje simple si la ejecución del trabajo aún no ha finalizado o una estructura JSON con los siguientes campos:

- `exitCode`: código de salida numérico; los valores negativos indican situaciones de fallo.
- `program`: último programa lanzado por el trabajo.
- `status`: un valor posible entre:
  - `Error`: cuando `exitCode = -1`; corresponde a un error (técnico) que se produce durante la ejecución del trabajo.
  - `Failed`: cuando `exitcode = -2`; corresponde a un fallo que se produce durante la ejecución de un programa de servicio (como una situación ABEND).
  - `Succeeded`: cuando `exitCode >= 0`;
- `stepName`: nombre del último paso ejecutado en el trabajo.

#### executionMode

SYNCHRONOUS o ASYNCHRONOUS, según la forma en que se haya iniciado el trabajo.

#### Salida de ejemplo:

```
{
  "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"
}
```

## Estructura de resultados del lanzamiento de la transacción

La estructura podría contener los siguientes campos:

#### outCome

una cadena que representa el resultado de la ejecución de la transacción. Los valores posibles son los siguientes:

- `Success`: la ejecución de la transacción ha finalizado correctamente.
- `Failure`: la ejecución de la transacción no ha finalizado correctamente, se han producido algunos problemas.

## commarea

una cadena que representa el valor final de COMMAREA, como una matriz de bytes codificada en byte64. Puede ser una cadena vacía.

## containerRecord

(opcional) una cadena que representa el contenido del registro del CONTAINER como una matriz de bytes codificada en byte64.

## serverDescription

Puede contener información sobre el servidor que ha atendido la solicitud (con fines de depuración). Puede ser una cadena vacía.

## abendCode

(opcional) si el programa al que se hace referencia en la transacción lanzada está ausente, el valor del código de abend se devolverá en forma de cadena en este campo.

## Respuestas de ejemplo

### Success

```
{
  "outcome": "Success",
  "commarea": "",
  "serverDescription": ""
}
```

### Failure

```
{
  "outcome": "Failure",
  "commarea": "",
  "serverDescription": "",
  "abendCode": "AEIA"
}
```

## Estructura de resultados del registro de lanzamiento de la transacción

La estructura podría contener los siguientes campos:

## recordContent

una cadena que representa el contenido del registro del COMMAREA como una matriz de bytes codificada en byte64.

## containerRecord

una cadena que representa el contenido del registro del CONTAINER como una matriz de bytes codificada en byte64.

## serverDescription

Puede contener información sobre el servidor que ha atendido la solicitud (con fines de depuración). Puede ser una cadena vacía.

## Respuestas de ejemplo

### Success

```
{
  "recordContent": "",
  "serverDescription": ""
}
```

Los posibles estados de un trabajo en una cola son:

En una cola, los trabajos pueden tener el siguiente estado:

### ACTIVE

El trabajo se está ejecutando actualmente en la cola.

### EXECUTION\_WAIT

El trabajo está esperando a que haya un subprocesso disponible.

### SCHEDULED

La ejecución del trabajo está programada en una fecha y hora específicas.

### HOLD

El trabajo está esperando a ser publicado antes de ejecutarse.

### COMPLETED

El trabajo se ha ejecutado correctamente.

## ERROR

La ejecución del trabajo ha fallado.

## UNKNOWN

Se desconoce el estado.

## Envíe el trabajo y programe la entrada del trabajo

La entrada del trabajo de envío y el trabajo de programación es la serialización en JSON de un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` objeto. El ejemplo de entrada que aparece a continuación muestra todos los campos de dicho bean.

Ejemplo de entrada para enviar un trabajo:

```
{
  "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
}
```

Ejemplo de entrada para programar un trabajo:

```
{
  "scheduleCron": "* / 2 * * * * ? ",
  "programName": "LOGPGM",
  "programParams": {
    "cl_sbmjob_param_json": "[\"./output/schedule-job-log.txt\", \"Every 2 seconds!\"]"
  },
  "localDataAreaValue": ""
}
```

```
"userName": "PV0",
"jobName": "LOGGERJOB",
"jobPriority": 5,
"jobQueue": "queue1",
"scheduleMisfirePolicy": 4,
"startTime": "2003/05/04 07:00:00.000 GMT-06:00",
"endTime": "2003/05/04 07:00:07.000 GMT-06:00"
}
```

## jobNumber

si el número de trabajo es 0, el número de trabajo se generará automáticamente utilizando el siguiente número de la secuencia de números de trabajo. Ese valor debe establecerse en 0 (excepto para fines de prueba).

## jobPriority

La prioridad de trabajo predeterminada en el AS400 es 5. El rango válido es de 0 a 9, siendo 0 la prioridad más alta.

## trabajo OnHold

Si un trabajo se envía en espera, no se ejecutará de inmediato, sino cuando alguien lo “publique”. Se puede publicar un trabajo mediante la API de REST (/release o /release-all).

## scheduleDate y scheduleTime

Si estos valores no son null, el trabajo se ejecutará en la fecha y hora especificadas.

## Date

Se puede proporcionar con el formato mmddyy o ddmMyyyy (el tamaño de la entrada determinará qué formato se utilizará)

## Tiempo

Se puede proporcionar con el formato HHmm o HHmmss (el tamaño de la entrada determinará qué formato se utilizará)

## programParams

Se pasará al programa en forma de mapa.

## horario MisfirePolicy

Define la estrategia utilizada cuando se produce un error en el gatillo. A continuación se muestran los posibles valores:

1. Libera el primer fallo de encendido y descarta los demás fallos de encendido.
2. Deje en suspenso el primer fallo de encendido y deseche los demás fallos de encendido.
3. Deseche el fallo de encendido.
4. Libera todos los fallos de encendido. La cola de trabajos incluirá todos los trabajos.

## Lista de respuestas a los trabajos programados

Esta es la estructura del punto de conexión de la cola de trabajos de la lista de trabajos. El mensaje de envío de trabajos que se utilizó para enviar ese trabajo forma parte de la respuesta. Este se puede utilizar para realizar un seguimiento, realizar pruebas o volver a enviarlo. Cuando se complete un trabajo, también se rellenarán la fecha de inicio y la fecha de finalización.

```
[
  {
    "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
  }
]

```

## Respuesta a la lista de trabajos repetidos

Esta es la estructura del punto final de la cola de trabajos `/schedule/list`.

```
[
  {
    "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 Configuración de Blu Age Runtime (no gestionado)

En esta sección se explican los pasos para configurar AWS Blu Age Runtime (no gestionado) en su AWS infraestructura.

### Temas

- [AWS Requisitos previos de tiempo de ejecución de Blu Age](#)

- [AWS Incorporación de Blu Age Runtime](#)
- [Requisitos de configuración de infraestructura para AWS Blu Age Runtime \(no gestionado\)](#)
- [AWS Implementación de Blu Age Runtime en Amazon ECS gestionada por AWS Fargate](#)
- [AWS Implementación de Blu Age Runtime en Amazon EC2](#)
- [Pruebe la aplicación PlanetsDemo](#)

## AWS Requisitos previos de tiempo de ejecución de Blu Age

AWS Blu Age Runtime (no gestionado) está disponible en varias [versiones](#). Si tiene proyectos de modernización en curso, es posible que necesite versiones incrementales del motor de ejecución para fines de implementación y pruebas. Para definir sus necesidades, póngase en contacto con su gerente de entrega de AWS Blu Age.

Antes de comenzar el proceso de incorporación de AWS Blu Age Runtime (no gestionado), haga lo siguiente:

- Asegúrese de tener una AWS cuenta.
- Asegúrese de tener una aplicación modernizada rediseñada con Blu Age. AWS
- Elija una AWS región y una de las opciones de procesamiento compatibles con AWS Blu Age Runtime (no administrado).
- Elige la versión de AWS Blu Age Runtime que quieras usar.
- Revise [the section called “Requisitos de configuración de infraestructura”](#) y valide los componentes adicionales necesarios para ejecutar el AWS Blu Age Runtime (no administrado).

### Note

[Si desea probar las funciones de AWS Blu Age Runtime \(no gestionado\), puede utilizar la aplicación de demostración Planets Demo, que puede descargar desde PlanetsDemo - v1.zip.](#)

## AWS Incorporación de Blu Age Runtime

Para empezar, crea un AWS Support estuche para solicitar la incorporación y acceder a AWS Blu Age Runtime. Incluya en su solicitud su Cuenta de AWS ID, la AWS región que desea usar y una

opción de procesamiento y una versión de tiempo de ejecución. Si no estás seguro de qué versión necesitas, ponte en contacto con tu responsable de entrega de AWS Blu Age.

### Note

El AWS Blu Age Runtime está disponible en dos variedades principales: versiones preliminares alfa y versiones oficiales. Para determinar qué versión utilizar, consulta [Cómo empezar](#) en el sitio de Blu Insights o ponte en contacto con tu responsable de entregas de AWS Blu Age.

## AWS Blu Age Runtime (no gestionado) en Amazon EC2

Almacenamos los artefactos de AWS Blu Age Runtime (no gestionados) en diferentes depósitos de Amazon S3 por región y por elección de procesamiento. Para acceder al bucket de AWS Blu Age Runtime (no gestionado) en Amazon EC2, utilice el nombre que aparece en la siguiente tabla.

Región de AWS

### Note

Esta tabla se aplica a Amazon EC2 y a las instancias de Amazon EC2 utilizadas en Amazon ECS y Amazon EKS.

Región de AWS	Depósito de liberación	Cubeta previa al lanzamiento
Este de EE. UU. (Ohio)	aws-bluage-runtime-artifacts-055777665268-us-east-2	aws-bluage-runtime-artifacts-dev-055777665268-us-east-2
Este de EE. UU. (Norte de Virginia)	aws-bluage-runtime-artifacts-139023371234-us-east-1	aws-bluage-runtime-artifacts-dev-139023371234-us-east-1
Oeste de EE. UU. (Norte de California)	aws-bluage-runtime-artifacts-788454048782-us-west-1	aws-bluage-runtime-artifacts-dev-788454048782-us-west-1
Oeste de EE. UU. (Oregón)	aws-bluage-runtime-artifacts-836771190483-us-west-2	aws-bluage-runtime-artifacts-dev-836771190483-us-west-2

Región de AWS	Depósito de liberación	Cubeta previa al lanzamiento
Europa (Irlanda)	aws-bluage-runtime-artifacts-925278190477-eu-west-1	aws-bluage-runtime-artifacts-dev-925278190477-eu-west-1
Europa (París)	aws-bluage-runtime-artifacts-673009995881-eu-west-3	aws-bluage-runtime-artifacts-dev-673009995881-eu-west-3
Europa (Fráncfort)	aws-bluage-runtime-artifacts-485196800481-eu-central-1	aws-bluage-runtime-artifacts-dev-485196800481-eu-central-1
América del Sur (São Paulo)	aws-bluage-runtime-artifacts-737536804457-sa-east-1	aws-bluage-runtime-artifacts-dev-737536804457-sa-east-1
Asia-Pacífico (Tokio)	aws-bluage-runtime-artifacts-445578176276-ap-northeast-1	aws-bluage-runtime-artifacts-dev-445578176276-ap-northeast-1
Asia-Pacífico (Sídney)	aws-bluage-runtime-artifacts-726160321909-ap-southeast-2	aws-bluage-runtime-artifacts-dev-726160321909-ap-southeast-2

## AWS Blu Age Runtime (no gestionado) en Amazon ECS gestionado por Fargate

Almacenamos los artefactos de AWS Blu Age Runtime (no gestionados) en diferentes depósitos de Amazon S3 por región y por elección de procesamiento. Para acceder al bucket de AWS Blu Age Runtime (no gestionado) en Amazon ECS gestionado por Fargate, utiliza el nombre que aparece en la siguiente tabla. Región de AWS

Región de AWS	Libera el bucket	Cubeta previa al lanzamiento
Este de EE. UU. (Ohio)	aws-bluage-runtime-fargate-rel-483416914331-us-east-2	aws-bluage-runtime-fargate-dev-483416914331-us-east-2
Este de EE. UU. (Norte de Virginia)	aws-bluage-runtime-fargate-rel-308472162679-us-east-1	aws-bluage-runtime-fargate-dev-308472162679-us-east-1

Región de AWS	Libera el bucket	Cubeta previa al lanzamiento
Oeste de EE. UU. (Norte de California)	aws-bluage-runtime-fargate-rel-343763094578-us-west-1	aws-bluage-runtime-fargate-dev-343763094578-us-west-1
Oeste de EE. UU. (Oregón)	aws-bluage-runtime-fargate-rel-688933007849-us-west-2	aws-bluage-runtime-fargate-dev-688933007849-us-west-2
Europa (Irlanda)	aws-bluage-runtime-fargate-rel-140138033705-eu-west-1	aws-bluage-runtime-fargate-dev-140138033705-eu-west-1
Europa (París)	aws-bluage-runtime-fargate-rel-339712948211-eu-west-3	aws-bluage-runtime-fargate-dev-339712948211-eu-west-3
Europa (Fráncfort)	aws-bluage-runtime-fargate-rel-339712918892-eu-central-1	aws-bluage-runtime-fargate-dev-339712918892-eu-central-1
América del Sur (São Paulo)	aws-bluage-runtime-fargate-rel-767397998881-sa-east-1	aws-bluage-runtime-fargate-dev-767397998881-sa-east-1
Asia-Pacífico (Tokio)	aws-bluage-runtime-fargate-rel-891377400849-ap-northeast-1	aws-bluage-runtime-fargate-dev-891377400849-ap-northeast-1
Asia-Pacífico (Sídney)	aws-bluage-runtime-fargate-rel-533267435478-ap-southeast-2	aws-bluage-runtime-fargate-dev-533267435478-ap-southeast-2

## Se usa para enumerar el contenido del bucket AWS CLI

Una vez incorporado, puede enumerar el contenido del depósito ejecutando el siguiente AWS CLI comando en una terminal.

```
aws s3 ls bucket-name
```

*bucket-name* Sustitúyelo por el nombre del depósito correspondiente Región de AWS a usted de la tabla anterior.

Este comando devuelve una lista de carpetas que corresponden a diferentes versiones del motor de ejecución (no administrado) de AWS Blu Age Runtime, como la siguiente para un bucket de versiones:

```
PRE 3.10.0/  
PRE 4.0.0/
```

O lo siguiente para un bucket de compilación:

```
PRE 4.1.0-alpha.8/  
PRE 4.1.0-alpha.9/
```

Le recomendamos que utilice la última versión disponible. Si eso no es posible, utilice la versión en tiempo de ejecución que se validó durante la fase de refactorización de la aplicación. Para ver una lista de los marcos disponibles para una versión específica, ejecute el siguiente comando:

```
aws s3 ls s3://bucket-name/version/Framework/
```

*bucket-name*Sustitúyalo por el nombre del bucket correspondiente a su versión Región de AWS y *version* por el nombre que desee. A continuación se muestran dos ejemplos.

Para un depósito de lanzamiento:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/  
Framework/
```

El comando devuelve una lista de marcos, como:

```
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
```

Para un bucket de compilación:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-dev-139023371234-us-  
east-1/4.1.0-alpha.9/Framework/
```

El comando devuelve una lista de marcos, como:

```
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
```

## Descarga del marco

Puede descargar el marco, por ejemplo, para actualizar la versión de AWS Blu Age Runtime en una instancia Amazon EC2 existente.

```
aws s3 cp s3://bucket-name/version/Framework/ folder-of-your-choice --recursive
```

Donde:

*folder-of-your-choice*

ruta de la carpeta en la que vaya a descargar el marco.

```
Por ejemplo: aws s3 cp s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/Framework/ . --recursive
```

Este comando produce el resultado siguiente:

```
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
```

Puede obtener una lista de los archivos del marco de la siguiente manera:

```
ls -l
```

Este comando produce el resultado siguiente:



```
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
```

## Requisitos de configuración de infraestructura para AWS Blu Age Runtime (no gestionado)

En este tema se describe la configuración de infraestructura mínima necesaria para ejecutar AWS Blu Age Runtime (no gestionado). Los siguientes procedimientos describen cómo configurar AWS Blu Age Runtime (no administrado) en el equipo que prefiera para implementar una aplicación modernizada en AWS Blu Age Runtime. Los recursos que cree deben estar en una Amazon VPC que tenga una subred dedicada al dominio de su aplicación.

### Temas

- [Requisitos de infraestructura](#)
- [Tipos de instancias de Amazon EC2 para AWS Blu Age Runtime \(en Amazon EC2\)](#)
- [Ejecución de AWS Blu Age Runtime en Amazon EC2](#)
- [Ejecución de AWS Blu Age Runtime en Amazon ECS en Amazon EC2](#)
- [Ejecución de AWS Blu Age Runtime en Amazon EKS en Amazon EC2](#)
- [Ejecución de AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate](#)

## Requisitos de infraestructura

### Creación de un grupo de seguridad

Si planea trabajar en instancias de Amazon EC2 en Amazon EKS, omita este procedimiento porque el proceso de creación del clúster de Amazon EKS crea un grupo de seguridad en su nombre. Utilice ese grupo de seguridad en los siguientes procedimientos en lugar de crear uno nuevo.

1. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
2. En el panel de navegación izquierdo, en Seguridad, elija Grupos de seguridad.

3. En el panel central, elija Crear grupos de seguridad.
4. En el campo Nombre del grupo de seguridad, escriba **M2BluagePrivateLink-SG**.
5. En la sección Inbound rules (Reglas de entrada), elija Add rule (agregar regla).
6. En Tipo, seleccione HTTPS.
7. En Origen, introduzca el CIDR de su VPC.
8. En la sección Reglas salientes, elija Agregar regla.
9. En Tipo, seleccione HTTPS.
10. En Destino, escriba **0.0.0.0/0**.
11. Elija Crear grupo de seguridad.

### Crear un punto de conexión de VPC de Amazon

1. Abra la consola de Amazon VPC en <https://console.aws.amazon.com/vpc/>.
2. En el panel de navegación izquierdo, en Nube virtual privada, seleccione Puntos de conexión.
3. En el panel central, elija Crear punto de conexión.
4. En la sección Servicios, introduzca **SQS** en el campo de búsqueda y, a continuación, seleccione el servicio Amazon SQS que corresponda a su región.
5. En VPC, seleccione la Amazon VPC que creó en el paso anterior.
6. En la sección Subredes, seleccione la subred que creó para el dominio de su aplicación.
7. En la sección Grupos de seguridad, seleccione el grupo de seguridad del procedimiento anterior.
8. Seleccione Crear punto de conexión.

### Creación de una política de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación izquierdo, en Administración de acceso, seleccione Políticas.
3. En el panel central, elija Crear política.
4. En la sección Editor de políticas, elija JSON.
5. Sustituya todo el JSON que vea en el editor por el siguiente JSON.

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

```
{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": [
    "sqs:GetQueueUrl",
    "sqs:ReceiveMessage",
    "sqs:SendMessage"
  ],
  "Resource": "*"
}
```

#### Note

Si necesita más detalles para personalizar su política, póngase en contacto con su gerente de entrega o administrador de cuentas de AWS Blu Age.

6. Elija Siguiente.
7. Escriba un nombre para la política y elija Crear política.

### Creación de un rol de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, en Administración del acceso, elija Roles.
3. En el panel central, elija Crear rol.
4. En la sección Casos de uso, en función del ordenador que elija, elija una de las siguientes opciones:
  - EC2 (para Amazon EC2 y Amazon EKS en Amazon EC2)
  - Elastic Container Service y, a continuación, Rol de EC2 para Elastic Container Service (para Amazon ECS en Amazon EC2)
  - Elastic Container Service y, después, Elastic Container Service Task (para Amazon ECS gestionado por Fargate)
5. Elija Siguiente.
6. En el campo de búsqueda, introduzca el nombre de la política que creó antes.
7. Seleccione la casilla de verificación a la izquierda de su política.

**Note**

Si no puede agregar una política, termine de crear el rol y, a continuación, actualícelo para agregar la política.

8. Elija Siguiente.
9. Ingrese un nombre para el rol y, a continuación, seleccione Crear rol.

## Tipos de instancias de Amazon EC2 para AWS Blu Age Runtime (en Amazon EC2)

La siguiente es una lista de los tipos de instancias de Amazon EC2 que puede usar para AWS Blu Age Runtime (en Amazon EC2) al crear instancias de Amazon EC2 o al definir los nodos de trabajo de Amazon EKS.

```
t3.xlarge
t3.small
t3.large
t2.small
t2.large
r7a.medium
r7a.large
r7a.xlarge
r7a.2xlarge
r7a.4xlarge
r7a.8xlarge
r7a.12xlarge
r7a.16xlarge
r7a.24xlarge
r7a.32xlarge
r7a.48xlarge
r7a.metal-48xl
r7i.large
r7i.xlarge
r7i.2xlarge
r7i.4xlarge
r7i.8xlarge
r7i.12xlarge
r7i.16xlarge
r7i.24xlarge
r7i.48xlarge
```

r7i.metal-24xl  
r7i.metal-48xl  
r6i.xlarge  
r6i.large  
r6i.4xlarge  
r6i.2xlarge  
r5b.xlarge  
r5b.large  
r5b.2xlarge  
r3.xlarge  
m6i.xlarge  
m6i.large  
m6i.8xlarge  
m6i.4xlarge  
m6i.2xlarge  
m6i.16xlarge  
m5zn.xlarge  
m5zn.large  
m5zn.3xlarge  
m5zn.2xlarge  
m5.xlarge  
m5.large  
m5.8xlarge  
m5.4xlarge  
m5.2xlarge  
m5.16xlarge  
m5.12xlarge  
c6i.xlarge  
c6i.large  
c6i.8xlarge  
c6i.4xlarge  
c6i.2xlarge  
c6i.16xlarge  
c5.xlarge  
c5.large  
c5.9xlarge  
c5.4xlarge  
c5.2xlarge  
c5.18xlarge  
c5.12xlarge

## Ejecución de AWS Blu Age Runtime en Amazon EC2

Para crear una instancia de Amazon EC2, siga estos pasos.

### Crear una instancia de Amazon EC2

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. Seleccione Iniciar instancia.
3. Para el tipo de instancia, elija uno de los tipos que aparecen en [the section called “Tipos de instancias de Amazon EC2 para AWS Blu Age Runtime \(en Amazon EC2\)”](#).
4. En la sección Par de claves, seleccione o bien un par de claves existente o bien cree uno nuevo.
5. En la sección Configuración de red, elija Seleccionar grupo de seguridad existente.
6. En Grupos de seguridad comunes, elija M2 BluagePrivate Link-SG.
7. Amplíe la sección Detalles avanzados.
8. En Perfil de instancia de IAM, seleccione el rol de IAM que creó antes.
9. Seleccione Iniciar instancia.

### Instale la aplicación en la instancia de Amazon EC2

1. Cuando el estado de la instancia de Amazon EC2 cambie a En ejecución, conéctese a la instancia.
2. Instale los siguientes componentes de software en la instancia:
  - Java Runtime Environment (JRE) 17.
  - Apache Tomcat 10.
  - AWS Blu Age Runtime (en Amazon EC2). Instale el motor de ejecución AWS Blu Age en la raíz de la carpeta de instalación de Apache Tomcat (algunos archivos se añadirán y otros se sobrescribirán).

Para instalar las aplicaciones web adicionales que se incluyen con el archivo AWS Blu Age Runtime, configure una instancia secundaria del servidor Apache Tomcat y descomprima el archivo de aplicaciones web en esa ubicación.

## Ejecución de AWS Blu Age Runtime en Amazon ECS en Amazon EC2

1. Cree un clúster de Amazon ECS con instancias de Amazon EC2 como infraestructura subyacente. Consulte [Introducción a Windows en Amazon EC2 en la Guía para desarrolladores de Amazon Elastic Container Service](#).
2. Especifique el rol de IAM que creó en los pasos anteriores.
3. Elija uno de los tipos de instancias que se muestran en [the section called “Tipos de instancias de Amazon EC2 para AWS Blu Age Runtime \(en Amazon EC2\)”](#).
4. En Configuración de red para instancias de Amazon EC2, elija el grupo de seguridad que creó en los pasos anteriores.

## Ejecución de AWS Blu Age Runtime en Amazon EKS en Amazon EC2

1. Cree un clúster de Amazon EKS. Consulte [Creación de un clúster de Amazon EKS](#) en la Guía del usuario de Amazon EKS.
2. Como se mencionó anteriormente, se crea un grupo de seguridad en su nombre. Puede usar ese grupo de seguridad al crear el punto de enlace de Amazon VPC.
3. Cree un grupo de nodos. Especifique el rol de IAM que creó en los pasos anteriores.
4. Elija uno de los tipos de instancias que se muestran en [the section called “Tipos de instancias de Amazon EC2 para AWS Blu Age Runtime \(en Amazon EC2\)”](#).
5. Amazon EKS asignará automáticamente el grupo de seguridad a las instancias de Amazon EC2 generadas.

## Ejecución de AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate

Cree un clúster de Amazon ECS con AWS Fargate (sin servidor) como infraestructura subyacente. Consulte [Introducción a los contenedores de Linux en AWS Fargate en la Guía para desarrolladores de Amazon Elastic Container Service](#).

## AWS Implementación de Blu Age Runtime en Amazon ECS gestionada por AWS Fargate

En los temas de esta sección se describe cómo configurar AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate, cómo actualizar la versión en tiempo de ejecución, cómo supervisar

la implementación mediante CloudWatch alarmas de Amazon y cómo añadir dependencias con licencia.

## Temas

- [Configuración de AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate](#)
- [Actualización del tiempo de ejecución de AWS Blu Age en Amazon ECS gestionado por AWS Fargate](#)
- [Amazon CloudWatch Alarms for AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate](#)
- [Configuración de dependencias con licencia en AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate](#)

## Configuración de AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate

En este tema se explica cómo configurar e implementar la aplicación de PlanetsDemo ejemplo mediante AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate.

AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate está disponible para Linux/X86.

## Temas

- [Requisitos previos](#)
- [Configuración](#)
- [Pruebe la aplicación implementada](#)

## Requisitos previos

Antes de comenzar, asegúrese de que cumple los requisitos previos que se indican a continuación:

- Configure el AWS CLI archivo siguiendo los pasos de [Configuración de la AWS CLI](#).
- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Descargue el tiempo de ejecución de AWS Blu Age en Amazon ECS gestionado por AWS Fargate binarios. Para ver instrucciones, consulte [the section called “AWS Incorporación de Blu Age Runtime”](#).



- Descargue los archivos binarios de Apache Tomcat 10.
- Descargue el archivo de la [PlanetsDemo aplicación](#).
- Cree una base de datos PostgreSQL de Amazon Aurora para JICS y ejecute `PlanetsDemo-v1/jics/sql/initJics.sql` la consulta en ella. Para obtener información sobre cómo crear una base de datos PostgreSQL de Amazon Aurora, [consulte Creación y conexión a un clúster de base de datos PostgreSQL de Aurora](#).

## Configuración

Para configurar la aplicación de PlanetsDemo muestra, complete los siguientes pasos.

1. Tras descargar los archivos binarios de Apache Tomcat, extraiga el contenido y vaya a la `conf` carpeta. Abra el `catalina.properties` archivo para editarlo y sustituya la línea que comienza `common.loader` por la siguiente línea.

```
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. Comprima la carpeta Apache Tomcat mediante el comando `tar` para crear un archivo ``tar.gz``.
3. Prepare un [Dockerfile](#) para crear su imagen personalizada a partir de los binarios de tiempo de ejecución proporcionados y los binarios del servidor Apache Tomcat. Consulta el siguiente ejemplo de Dockerfile. El objetivo es instalar Apache Tomcat 10, seguido de AWS Blu Age Runtime (para Amazon ECS gestionado por AWS Fargate) extraído de la raíz del directorio de instalación de Apache Tomcat 10 y, a continuación, instalar la aplicación modernizada de muestra denominada. PlanetsDemo

### Note

El contenido de los scripts `install-gapwalk.sh` y `install-app.sh`, que se utilizan en este ejemplo de Dockerfile, aparece después del Dockerfile.

```
FROM --platform=linux/x86_64 amazonlinux:2  
  
RUN mkdir -p /workdir/apps  
WORKDIR /workdir
```

```
COPY install-gapwalk.sh .
COPY install-app.sh .
RUN chmod +x install-gapwalk.sh
RUN chmod +x install-app.sh

# Install Java and AWS CLI v2-y
RUN yum install sudo java-17-amazon-corretto unzip tar -y
RUN sudo yum remove awscli -y
RUN curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
  "awscliv2.zip"
RUN sudo unzip awscliv2.zip
RUN sudo ./aws/install

# Installation dir
RUN mkdir -p /usr/local/velocity/installation/gapwalk
# Copy PlanetsDemo archive to a dedicated apps dir
COPY PlanetsDemo-v1.zip /workdir/apps/

# Copy resources (tomcat, blu age runtime) to installation dir
COPY tomcat.tar.gz /usr/local/velocity/installation/tomcat.tar.gz
COPY aws-bluage-on-fargate-runtime-4.x.x.tar.gz /usr/local/velocity/installation/
gapwalk/gapwalk-bluage-on-fargate.tar.gz

# run relevant installation scripts
RUN ./install-gapwalk.sh
RUN ./install-app.sh

EXPOSE 8080
EXPOSE 8081
# ...

# Run Command to start Tomcat server
CMD ["sh", "-c", "sudo /bluage-on-fargate/tomcat.gapwalk/velocity/startup.sh
  $ECS_CONTAINER_METADATA_URI_V4 $AWS_CONTAINER_CREDENTIALS_RELATIVE_URI"]
```

El siguiente es el contenido de install-gapwalk.sh.

```
#!/bin/sh

# Vars
TEMP_DIR=/bluage-on-fargate/tomcat.gapwalk/temp

# Install
```

```

echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage-on-fargate
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk-bluage-on-fargate.tar.gz
  ${TEMP_DIR}
sudo cp /usr/local/velocity/installation/tomcat.tar.gz ${TEMP_DIR}
#•Create velocity dir
mkdir -p /bluage-on-fargate/tomcat.gapwalk/velocity
#•Extract tomcat files
tar -xvf ${TEMP_DIR}/tomcat.tar.gz -C ${TEMP_DIR}
# Copy all tomcat files to velocity dir
cp -fr ${TEMP_DIR}/apache-tomcat-10.x.x/* /bluage-on-fargate/tomcat.gapwalk/
velocity
# Remove default webapps of Tomcat
rm -f /bluage-on-fargate/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk-bluage-on-fargate.tar.gz -C /bluage-on-fargate/
tomcat.gapwalk
# Remove temporary extraction dir
sudo rm -rf ${TEMP_DIR}

```

El siguiente es el contenido de 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/

```

- Proporcione la información de conexión de la base de datos que creó como parte de los requisitos previos en el siguiente fragmento del application-main.yml archivo, que se encuentra en la carpeta. {TOMCAT\_GAPWALK\_DIR}/config Para obtener más información, consulte [Creación de un clúster de base de datos de Aurora PostgreSQL y conexión a él.](#)

```

datasource:
  jicsDs:
    driver-class-name :
    url:

```

```
username:
password:
type :
```

5. Cree e inserte la imagen en su repositorio de Amazon ECR. Para obtener instrucciones, consulte [Cómo insertar una imagen de Docker](#) en la Guía del usuario de Amazon Elastic Container Registry.
6. Abra la consola en <https://console.aws.amazon.com/ecs/v2>.
7. En el panel de navegación izquierdo, elija Definiciones de tareas.
8. En Tipo de lanzamiento, elija AWS Fargate.
9. Seleccione el rol de tarea del que creó como parte [the section called “Requisitos de configuración de infraestructura”](#).
10. Adjunta tu imagen al contenedor.
11. Termina de rellenar el formulario y, a continuación, selecciona Crear.
12. En el panel de navegación izquierdo, selecciona Clústeres y, a continuación, selecciona tu clúster de la lista.
13. En la página de detalles del clúster, en la pestaña Servicios, selecciona Crear.
14. Seleccione la definición de la tarea.
15. Amplíe la sección Redes y configure la VPC, las subredes y el grupo de seguridad que creó como parte de ellos. [the section called “Requisitos de configuración de infraestructura”](#)
16. Implemente su servicio Amazon ECS.

Si la implementación falla, compruebe los registros. Para encontrarlos, vaya a la página de tareas de Amazon ECS gestionado por y AWS Fargate, a continuación, seleccione la pestaña Logs. Si encuentra códigos de error que comiencen por una C seguida de un número, como CXXXX, anote los mensajes de error. Por ejemplo, el código de error C5102 es un error común que indica una configuración de infraestructura incorrecta. También puede navegar dentro de la tarea en ejecución y ejecutar algunos comandos, de forma similar a AWS Blu Age Runtime (en Amazon EC2). Para obtener más información, consulte [Uso de Amazon ECS Exec para la depuración](#) en la Guía para desarrolladores de Amazon Elastic Container Service.

Para abrir un shell interactivo, ejecute el siguiente comando desde su máquina local.

```
aws ecs execute-command --cluster your_cluster_name --container your_container_name --
task task_id --interactive --command /bin/sh
```

## Pruebe la aplicación implementada

Para ver un ejemplo de cómo probar la PlanetsDemo aplicación, consulte [the section called “Pruebe la aplicación PlanetsDemo”](#).

## Actualización del tiempo de ejecución de AWS Blu Age en Amazon ECS gestionado por AWS Fargate

Esta guía describe cómo actualizar el tiempo de ejecución de AWS Blu Age en Amazon ECS administrado por AWS Fargate.

### Temas

- [Requisitos previos](#)
- [Actualice el tiempo de ejecución de AWS Blu Age](#)

### Requisitos previos

Antes de comenzar, asegúrese de que cumple los siguientes requisitos previos:

- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Descargue la versión de AWS Blu Age Runtime a la que desee actualizar. Para obtener más información, consulte [the section called “AWS Incorporación de Blu Age Runtime”](#). El marco consta de dos archivos binarios: `aws-bluage-runtime-x.x.x.x.tar.gz` y `yaws-bluage-webapps-x.x.x.x.tar.gz`.

### Actualice el tiempo de ejecución de AWS Blu Age

Complete los siguientes pasos para actualizar el tiempo de ejecución de AWS Blu Age.

1. Reconstruye tu imagen de Docker con la versión de AWS Blu Age Runtime que desees. Para ver instrucciones, consulte [the section called “Configuración de AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate”](#).
2. Envía tu imagen de Docker a tu repositorio de Amazon ECR.
3. Detenga y reinicie el servicio Amazon ECS.
4. Verifique los registros.

El tiempo de ejecución de AWS Blu Age se actualizó correctamente.

## Amazon CloudWatch Alarms for AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate

Para tener notificaciones más visibles cada vez que las aplicaciones implementadas encuentren excepciones, configúrelo CloudWatch para recibir el registro de su aplicación y añada una alarma que le avise de posibles errores.

### Configuración de alarmas

Con CloudWatch los registros, puede configurar cualquier cantidad de métricas y alarmas, en función de su aplicación y sus necesidades.

En concreto, puede configurar alarmas proactivas para las alertas de uso directamente durante la creación del clúster de Amazon ECS, de forma que reciba notificaciones cuando se produzcan errores. Para resaltar los errores en la conexión al sistema de control AWS Blu Age, añada una métrica relativa a la cadena «Error C» en los registros. A continuación, puede definir una alarma que reaccione a esta métrica.

## Configuración de dependencias con licencia en AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate

En este tema se describe cómo configurar dependencias con licencia adicionales que puede utilizar con AWS Blu Age Runtime en Amazon ECS gestionado por AWS Fargate.

### Temas

- [Requisitos previos](#)
- [Información general](#)

### Requisitos previos

Antes de comenzar, asegúrese de que cumple los requisitos previos que se indican a continuación:

- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Obtenga las siguientes dependencias de su origen.

## Base de datos de Oracle

Proporcione un [controlador de base de datos de Oracle](#). Por ejemplo, ojdbc11-23.3.0.23.09.jar.

## Conexión MQ de IBM

Suministre un [cliente MQ de IBM](#). Por ejemplo, com.ibm.mq.jakarta.client-9.3.4.1.jar.

Con esta versión de dependencias, suministre también las siguientes dependencias transitivas:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15to18-1.76.jar
- bcutil-jdk15 a 18-1.76.jar

## Archivos de impresora DDS

Proporcione la [biblioteca de informes de Jasper](#). Por ejemplo,.jasperreports-6.16.0.jar, pero puede que sea compatible con una versión más reciente.

Con esta versión de dependencias, suministre también las siguientes dependencias transitivas:

- castor-core-1.4.1.jar
- castor-core-1.4.1.jar
- commons-digester-2.1.jar
- ecj-3.21.0.jar
- itext-2.1.7.js8.jar
- javax.inject-1.jar
- jcommon-1.0.23.jar
- jfreechart-1.0.19.jar
- commons-beanutils-1.9.4.jar
- commons-collections-3.2.2.jar

## Información general

Para instalar las dependencias, complete los pasos siguientes.

1. Copia cualquiera de las dependencias anteriores según sea necesario en la carpeta de compilación de imágenes de Docker.
2. Si su base de datos de JICS o Blusam está alojada en Oracle, proporcione el controlador de la base de datos de Oracle. *your-tomcat-path*/extra
3. En su Dockerfile, copie estas dependencias a. *your-tomcat-path*/extra
4. Cree su imagen de Docker y, a continuación, envíela a Amazon ECR.
5. Detenga y reinicie el servicio Amazon ECS.
6. Consulte los registros.

## AWS Implementación de Blu Age Runtime en Amazon EC2

En los temas de esta sección se describe cómo configurar AWS Blu Age Runtime (no gestionado) en Amazon EC2, cómo actualizar la versión de tiempo de ejecución, cómo supervisar la implementación mediante alarmas de CloudWatch Amazon y cómo añadir dependencias con licencia. Estas instrucciones se aplican al crear instancias de Amazon EC2 y al utilizar Amazon ECS en Amazon EC2 o Amazon EKS en Amazon EC2.

### Temas

- [Configuración de AWS Blu Age Runtime \(no gestionado\) en Amazon EC2](#)
- [Uso de contenedores en Amazon EC2 para Amazon ECS y Amazon EKS](#)
- [Actualización del tiempo de ejecución de AWS Blu Age en Amazon EC2](#)
- [AWS Tiempo de ejecución de Blu Age \(en Amazon EC2\) Amazon Alarms CloudWatch](#)
- [Configuración de dependencias con licencia en AWS Blu Age Runtime en Amazon EC2](#)

## Configuración de AWS Blu Age Runtime (no gestionado) en Amazon EC2

En este tema se explica cómo configurar e implementar la aplicación de PlanetsDemo ejemplo mediante AWS Blu Age Runtime (no administrado) en Amazon EC2.

### Temas

- [Requisitos previos](#)
- [Configuración](#)
- [Pruebe la aplicación implementada](#)



## Requisitos previos

Antes de comenzar, asegúrese de que cumple los requisitos previos que se indican a continuación:

- Configure el AWS CLI archivo siguiendo los pasos que se indican en [Configuración de la AWS CLI](#).
- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Cree una instancia de Amazon EC2 con uno de los tipos de instancias compatibles. Para obtener más información, consulte [Introducción a las instancias de Amazon EC2 Linux](#).
- Asegúrese de poder conectarse correctamente a la instancia de Amazon EC2, por ejemplo, mediante SSM.
- Descargue y extraiga AWS Blu Age Runtime (en Amazon EC2) en. *your-tomcat-path*/\* Asegúrese de colocar el bluage.bin archivo exactamente en la ubicación especificada por la variable de CATALINA\_HOME entorno descrita en [CATALINA\\_HOME y CATALINA\\_BASE](#) en la documentación de Apache Tomcat. Para obtener instrucciones sobre cómo recuperar el tiempo de ejecución de Blu Age, consulte. AWS [the section called “AWS Incorporación de Blu Age Runtime”](#)
- Descargue el [archivo PlanetsDemo de la aplicación](#).
- Descomprima el archivo y cargue la aplicación en el bucket de Amazon S3 de su elección.
- Cree una base de datos PostgreSQL de Amazon Aurora para JICS y ejecute P1anetsDemo-v1/jics/sql/initJics.sql la consulta en ella. Para obtener información sobre cómo crear una base de datos PostgreSQL de Amazon Aurora, [consulte Creación y conexión a un clúster de base de datos PostgreSQL de Aurora](#).

## Configuración

Para configurar la aplicación de PlanetsDemo muestra, complete los siguientes pasos.


1. Conéctese a su instancia de Amazon EC2 y vaya a la conf carpeta situada debajo de la carpeta de instalación de Apache Tomcat 10. Abra el catalina.properties archivo para editarlo y sustituya la línea que comienza common.loader por la siguiente línea.

```
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. Vaya a la carpeta *<your-tomcat-path>/webapps*.

3. Copie los PlanetsDemo archivos binarios disponibles en la PlanetsDemo carpeta -v1/webapps/ del bucket de Amazon S3 mediante el siguiente comando.

```
aws s3 cp s3://path-to-demo-app-webapps/ . --recursive
```

 Note

*path-to-demo-app-webapps* Sustitúyalo por el URI de Amazon S3 correcto para el depósito en el que descomprimió el PlanetsDemo archivo anteriormente.

4. Copie el contenido de la carpeta PlanetsDemo-v1/config/ a *<your-tomcat-path>/config/*.
5. Proporcione la información de conexión de la base de datos que creó como parte de los requisitos previos en el siguiente fragmento del archivo. `application-main.yml` Para obtener más información, consulte [Creación de un clúster de base de datos de Aurora PostgreSQL y conexión a él](#).

```
datasource:  
  jicsDs:  
    driver-class-name :  
    url:  
    username:  
    password:  
    type :
```

6. Inicie el servidor Apache Tomcat y compruebe los registros.

```
your-tomcat-path/startup.sh  
  
tail -f your-tomcat-path/logs/catalina.log
```

Si encuentra códigos de error que comiencen por una C seguida de un número, como CXXXX, anote los mensajes de error. Por ejemplo, el código de error C5102 es un error común que indica una configuración de infraestructura incorrecta.

## Pruebe la aplicación implementada

Para ver un ejemplo de cómo probar la PlanetsDemo aplicación, consulte [the section called “Pruebe la aplicación PlanetsDemo”](#).

## Uso de contenedores en Amazon EC2 para Amazon ECS y Amazon EKS

En este tema se explica cómo configurar e implementar la aplicación de PlanetsDemo ejemplo utilizando AWS Blu Age Runtime (no administrado) en Amazon EC2 como contenedor.

### Temas

- [Requisitos previos](#)
- [Configuración](#)
- [Pruebe la aplicación implementada](#)

### Requisitos previos

Antes de comenzar, asegúrese de que cumple los requisitos previos que se indican a continuación:

- Configure el AWS CLI archivo siguiendo los pasos que se indican en [Configuración de la AWS CLI](#).
- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Descargue AWS Blu Age Runtime (en Amazon EC2). Para obtener instrucciones sobre cómo recuperar el tiempo de ejecución, consulte [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Descargue el [archivo PlanetsDemo de la aplicación](#).
- Cree una base de datos PostgreSQL de Amazon Aurora para JICS y ejecute PlanetsDemo-v1/jics/sql/initJics.sql la consulta en ella. Para obtener información sobre cómo crear una base de datos PostgreSQL de Amazon Aurora, [consulte Creación y conexión a un clúster de base de datos PostgreSQL de Aurora](#).

### Configuración

Para configurar la aplicación de PlanetsDemo muestra, complete los siguientes pasos.

1. Prepare un [Dockerfile](#) para crear su imagen personalizada a partir de los binarios de tiempo de ejecución proporcionados y los binarios del servidor Apache Tomcat. Consulta el siguiente ejemplo de Dockerfile. El objetivo es instalar Apache Tomcat 10, seguido de AWS Blu Age

Runtime (en Amazon EC2) extraído de la raíz del directorio de instalación de Apache Tomcat 10 y, a continuación, instalar la aplicación modernizada de ejemplo denominada. PlanetsDemo Los `install-app.sh` scripts `install-gapwalk.sh` y que se utilizan en este ejemplo de Dockerfile aparecen después del Dockerfile.

```
FROM --platform=linux/x86_64 amazonlinux:2

RUN mkdir -p /workdir/apps
WORKDIR /workdir
COPY install-gapwalk.sh .
COPY install-app.sh .
RUN chmod +x install-gapwalk.sh
RUN chmod +x install-app.sh

# Install Java and AWS CLI v2-y
RUN yum install sudo java-17-amazon-corretto unzip tar -y
RUN sudo yum remove awscli -y
RUN curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
  "awscliv2.zip"
RUN sudo unzip awscliv2.zip
RUN sudo ./aws/install

#•Installation dir
RUN mkdir -p /usr/local/velocity/installation/gapwalk

# Copy PlanetsDemo archive to a dedicated apps dir
COPY PlanetsDemo-v1.zip /workdir/apps/

# Copy resources (tomcat, blu age runtime) to installation dir
COPY tomcat.tar.gz /usr/local/velocity/installation/tomcat.tar.gz
COPY aws-bluage-runtime-4.x.x.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"]
```

A continuación se muestra el contenido de `install-gapwalk.sh`

```
#!/bin/sh

# Vars
TEMP_DIR=/bluage/tomcat.gapwalk/temp

# Install
echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk-bluage.tar.gz ${TEMP_DIR}
sudo cp /usr/local/velocity/installation/tomcat.tar.gz ${TEMP_DIR}
# Create velocity dir
mkdir -p /bluage/tomcat.gapwalk/velocity
# Extract tomcat files
tar -xvf ${TEMP_DIR}/tomcat.tar.gz -C ${TEMP_DIR}
# Copy all tomcat files to velocity dir
cp -fr ${TEMP_DIR}/apache-tomcat-10.x.x/* /bluage/tomcat.gapwalk/velocity
# Remove default webapps of Tomcat
rm -f /bluage/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk-bluage.tar.gz -C /bluage/tomcat.gapwalk
# Remove temporary extraction dir
sudo rm -rf ${TEMP_DIR}
```

Los siguientes son los contenidos de `install-app.sh`.

```
#!/bin/sh

APP_DIR=/workdir/apps
TOMCAT_GAPWALK_DIR=/bluage/tomcat.gapwalk

unzip ${APP_DIR}/PlanetsDemo-v1.zip -d ${APP_DIR}
cp -r ${APP_DIR}/webapps/* ${TOMCAT_GAPWALK_DIR}/velocity/webapps/
cp -r ${APP_DIR}/config/* ${TOMCAT_GAPWALK_DIR}/velocity/config/
```

2. Proporcione la información de conexión de la base de datos que creó como parte de los requisitos previos en el siguiente fragmento del `application-main.yml` archivo, que se

encuentra en la carpeta. {TOMCAT\_GAPWALK\_DIR}/config Para obtener más información, consulte [Creación de un clúster de base de datos de Aurora PostgreSQL y conexión a él](#).

```
datasource:
  jicsDs:
    driver-class-name :
    url:
    username:
    password:
    type :
```

3. Cree e inserte la imagen en su repositorio de Amazon ECR. Para obtener instrucciones, consulte [Cómo insertar una imagen de Docker](#) en la Guía del usuario de Amazon Elastic Container Registry. A continuación, en función de su situación, cree un pod de Amazon EKS o una definición de tarea de Amazon ECS con su imagen de Amazon ECR e impleméntela en su clúster. Por ejemplo, consulte [Crear una definición de tarea mediante la consola en la](#) Guía para desarrolladores de Amazon Elastic Container Service e [Implementar una aplicación de muestra](#) en la Guía del usuario de Amazon EKS.

Pruebe la aplicación implementada

Para ver un ejemplo de cómo probar la PlanetsDemo aplicación, consulte [the section called “Pruebe la aplicación PlanetsDemo”](#).

## Actualización del tiempo de ejecución de AWS Blu Age en Amazon EC2

Esta guía describe cómo actualizar el tiempo de ejecución de AWS Blu Age en Amazon EC2.

### Temas

- [Requisitos previos](#)
- [Actualice el tiempo de ejecución de AWS Blu Age en la instancia de Amazon EC2](#)
- [Actualice el tiempo de ejecución de AWS Blu Age en un contenedor](#)

### Requisitos previos

Antes de comenzar, asegúrese de que cumple los siguientes requisitos previos:

- Para comprobar si hay instrucciones específicas para su versión, consulte [the section called “AWS Instrucciones de actualización de Blu Age”](#).

- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Asegúrese de tener una instancia de Amazon EC2 que contenga la versión más reciente de AWS Blu Age Runtime. Para obtener más información, consulte [Introducción a las instancias de Amazon EC2 Linux](#).
- Asegúrese de poder conectarse correctamente a la instancia de Amazon EC2, por ejemplo, mediante SSM.
- Descargue la versión de AWS Blu Age Runtime a la que desee actualizar. Para obtener más información, consulte [the section called “AWS Configuración de Blu Age Runtime \(no gestionado\)”](#) El marco consta de dos archivos binarios: `aws-bluage-runtime-x.x.x.x.tar.gz` y `aws-bluage-webapps-x.x.x.x.tar.gz`.

Actualice el tiempo de ejecución de AWS Blu Age en la instancia de Amazon EC2

Complete los siguientes pasos para actualizar el tiempo de ejecución de AWS Blu Age.

1. Conéctese a su instancia de Amazon EC2 y cambie el usuario a su ejecutando el siguiente comando.

```
sudo su
```

Necesita privilegios de superusuario para ejecutar los comandos de este tutorial.

2. Cree dos carpetas, una para cada archivo binario.
3. Dé a cada carpeta el nombre del archivo binario.
4. Copie cada archivo binario en la carpeta correspondiente.

#### Warning

Al extraer cada binario se crea una carpeta con el mismo nombre. Por lo tanto, si extrae ambos archivos binarios en la misma ubicación uno tras otro, sobrescribirá el contenido.

5. Para extraer los archivos binarios, utilice los siguientes comandos: Ejecute los comandos en cada carpeta:

```
tar xvf aws-bluage-runtime-x.x.x.x.tar.gz
tar xvf aws-bluage-webapps-x.x.x.x.tar.gz
```

6. Detenga los servicios de Apache Tomcat mediante los siguientes comandos.

```
systemctl stop tomcat.service
systemctl stop tomcat-webapps.service
```

7. Sustituya el contenido de <your-tomcat-path>/shared/ por el contenido de aws-bluage-runtime-x.x.x.x/velocity/shared/.
8. Sustituya <your-tomcat-path>/webapps/gapwalk-application.war por aws-bluage-runtime-x.x.x.x/velocity/webapps/gapwalk-application.war.
9. Reemplace los archivos war de <your-tomcat-path>/webapps/, es decir bac.war y jac.war, por los mismos archivos de aws-bluage-webapps-x.x.x.x/velocity/webapps/.
10. Inicie los servicios de Apache Tomcat ejecutando los siguientes comandos.

```
systemctl start tomcat.service
systemctl start tomcat-webapps.service
```

11. Consulte los registros.

Para comprobar el estado de la aplicación implementada, ejecute los siguientes comandos.

```
curl http://localhost:8080/gapwalk-application/
```

Debería aparecer el siguiente mensaje.

```
Jics application is running
```

```
curl http://localhost:8181/jac/api/services/rest/jicsservice/
```

Debería aparecer el siguiente mensaje.

```
Jics application is running
```

```
curl http://localhost:8181/bac/api/services/rest/bluesamserver/serverIsUp
```

La respuesta debe estar vacía.



El tiempo de ejecución de AWS Blu Age se actualizó correctamente.

Actualice el tiempo de ejecución de AWS Blu Age en un contenedor

Complete los siguientes pasos para actualizar el tiempo de ejecución de AWS Blu Age.

1. Reconstruye tu imagen de Docker con la versión de AWS Blu Age Runtime que desees. Para ver instrucciones, consulte [the section called “Configuración de AWS Blu Age Runtime \(no gestionado\) en Amazon EC2”](#).
2. Envía tu imagen de Docker a tu repositorio de Amazon ECR.
3. Detenga y reinicie el servicio Amazon ECS o Amazon EKS.
4. Consulte los registros.

El tiempo de ejecución de AWS Blu Age se actualizó correctamente.

## AWS Tiempo de ejecución de Blu Age (en Amazon EC2) Amazon Alarms CloudWatch

Para tener notificaciones más visibles cuando las aplicaciones implementadas encuentren excepciones que pongan a la aplicación en un período de gracia, puede configurarla CloudWatch para recibir el registro de la aplicación y añadir una alarma que le avise de posibles errores.

### Despliegue del CloudWatch registro

De forma predeterminada, el archivo `application-main.yml` incluye una referencia a otro archivo de configuración de registro denominado `logback-cloudwatch.yml`.

```
logging:  
  config: classpath:logback-cloudwatch.xml
```

Ambos archivos se encuentran en la carpeta de configuración y así es como se configura el CloudWatch registro, como se explica en las siguientes secciones.

### Configuración del CloudWatch registro

El archivo `logback-cloudwatch.xml` predeterminado tiene el siguiente contenido.

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE configuration>  
<configuration>  
  
  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
```

```

    <encoder>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </encoder>
  </appender>

  <appender name="cloudwatch"
class="com.netfactive.bluage.runtime.cloudwatchlogger.CloudWatchAppender">
    <logGroup>BluAgeRuntimeOnEC2-Logs</logGroup>
    <logStream>%date{yyyy-MM-dd,UTC}.%instanceId.%uuid</logStream>
    <layout>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </layout>
    <appender-ref ref="console" />
  </appender>

  <root level="INFO">
    <appender-ref ref="cloudwatch" />
  </root>
</configuration>

```

Todo lo que está fuera del elemento `<appender name="cloudwatch"/>` es la configuración estándar de logback. Hay dos anexadores en este archivo: un anexador de consola para enviar los registros a la consola y un CloudWatch anexador al que enviar los registros. CloudWatch

El atributo de `level` del elemento `root` especifica el nivel de registro de toda la aplicación.

Los valores obligatorios dentro de la etiqueta son: `<appender name="cloudwatch"/>`

- `<logGroup/>`: Establece el nombre del grupo de registros en CloudWatch. Si no se especifica ningún valor, el predeterminado es `BluAgeRuntimeOnEC2-Logs`. Si el grupo de registro no existe, se creará automáticamente. Este comportamiento se puede cambiar mediante la configuración, que se describe a continuación.
- `<logStream/>`: Establece el nombre del LogStream (dentro del grupo de registros) en CloudWatch

Valores opcionales:

- `<region/>`: sobrescribe la región en la que se escribirá el flujo de registro. De forma predeterminada, los registros van a la misma región que la instancia de EC2.
- `<layout/>`: el patrón que utilizarán los mensajes de registro.

- `<maxbatchsize/>`: El número máximo de mensajes de registro a los que se debe enviar CloudWatch por operación.
- `<maxbatchtimemillis/>`: el tiempo en milisegundos que se permite escribir CloudWatch los registros.
- `<maxqueuwaittimemillis/>`: el tiempo en milisegundos que se tarda en intentar insertar las solicitudes en la cola de registros interna.
- `<internalqueuesize/>`: el tamaño máximo de la cola interna.
- `<createlogdests/>`: crea un grupo y un flujo de registro si no existen.
- `<initialwaittimemillis/>`: la cantidad de tiempo que el subproceso debe permanecer en reposo al startup. Esta espera inicial permite una acumulación inicial de registros.
- `<maxeventmessagesize/>`: el tamaño máximo de un evento de registro. Los registros que superen este tamaño no se enviarán.
- `<truncateeventmessages/>`: trunca los mensajes que son demasiado largos.
- `<printrejectionevents/>`: habilita el appender de emergencia.

## CloudWatch configuración

Para que la configuración anterior envíe los registros correctamente CloudWatch, actualice la función de perfil de su instancia de IAM de Amazon EC2 para concederle permisos adicionales para el grupo de registros `BluAgeRuntimeOn `EC2-Logs`` y sus flujos de registros:

- `logs:CreateLogStream`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:DescribeLogGroups`

## Configuración de alarmas

Gracias a CloudWatch los registros, puede configurar diferentes métricas y alarmas, en función de su aplicación y sus necesidades. En concreto, puede configurar alarmas proactivas para las alertas de uso, a fin de recibir alertas en caso de que se produzcan errores que puedan poner a su aplicación en un período de gracia (y, en última instancia, impedir que funcione). Para ello, puede añadir en los registros una métrica relativa a la cadena «Error C5001», que muestre los errores en la conexión con

el sistema de control AWS Blu Age. A continuación, puede definir una alarma que reaccione a esta métrica.

## Configuración de dependencias con licencia en AWS Blu Age Runtime en Amazon EC2

Esta guía describe cómo configurar dependencias con licencia adicionales que puede utilizar con AWS Blu Age Runtime en Amazon EC2.

### Temas

- [Requisitos previos](#)
- [Información general](#)
- [Configure las dependencias para las aplicaciones web JAC y BAC](#)

### Requisitos previos

Antes de comenzar, asegúrese de que cumple los requisitos previos que se indican a continuación:

- Complete [the section called “AWS Requisitos previos de tiempo de ejecución de Blu Age”](#) y [the section called “AWS Incorporación de Blu Age Runtime”](#).
- Asegúrese de tener una instancia de Amazon EC2 que contenga la última versión de AWS Blu Age Runtime (en Amazon EC2). Para obtener más información, consulte [Introducción a las instancias de Amazon EC2 Linux](#).
- Asegúrese de poder conectarse correctamente a la instancia de Amazon EC2, por ejemplo, mediante SSM.
- Obtenga las siguientes dependencias de sus fuentes.

### Base de datos de Oracle

Proporcione un [controlador de base de datos de Oracle](#). Probamos la funcionalidad de AWS Blu Age Runtime (en Amazon EC2) con la versión ojdbc11-23.3.0.23.09.jar, pero podría ser compatible con una versión más reciente.

### Conexión MQ de IBM

Suministre un [cliente MQ de IBM](#). Probamos la funcionalidad de AWS Blu Age Runtime (en Amazon EC2) con la versión com.ibm.mq.jakarta.client-9.3.4.1.jar, pero podría ser compatible con una versión más reciente.

Con esta versión de dependencias, suministre también las siguientes dependencias transitivas:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15to18-1.76.jar
- bcutil-jdk15 a 18-1.76.jar

### Archivos de impresora DDS

Proporcione la [biblioteca de informes de Jasper](#). Probamos la funcionalidad de AWS Blu Age Runtime (en Amazon EC2) con jasperreports-6.16.0.jar, pero es posible que una versión más reciente sea compatible.

Con esta versión de dependencias, suministre también las siguientes dependencias transitivas:

- castor-core-1.4.1.jar
- castor-core-1.4.1.jar
- commons-digester-2.1.jar
- ecj-3.21.0.jar
- itext-2.1.7.js8.jar
- javax.inject-1.jar
- jcommon-1.0.23.jar
- jfreechart-1.0.19.jar
- commons-beanutils-1.9.4.jar
- commons-collections-3.2.2.jar

### Información general

Para instalar las dependencias, complete los pasos siguientes.

1. Conéctese a su instancia de Amazon EC2 y cambie el usuario a su ejecutando el siguiente comando.

```
sudo su
```

En este tutorial, necesitará el privilegio de superusuario para ejecutar los comandos.

2. Vaya a la carpeta `<your-tomcat-path>/extra/`.

```
cd <your-tomcat-path>/extra/
```

3. Copie cualquiera de las dependencias anteriores según sea necesario en esta carpeta.
4. Pare e inicie el `tomcat.service` ejecutando los siguientes comandos.

```
systemctl stop tomcat.service
```

```
systemctl start tomcat.service
```

5. Compruebe el estado del servicio para asegurarse de que se está ejecutando.

```
systemctl status tomcat.service
```

6. Verifique los registros.

Configure las dependencias para las aplicaciones web JAC y BAC

1. Si su base de datos de JICS o Blusam está alojada en Oracle, debe proporcionar el controlador de base de datos de Oracle en `<your-tomcat-path>/extra`.
2. Cree la carpeta si aún no está presente.
3. Detenga y reinicie el servidor Apache Tomcat.
4. Verifique los registros.

## Pruebe la aplicación PlanetsDemo

Para comprobar el estado de la PlanetsDemo aplicación implementada, ejecute los siguientes comandos después de reemplazarla `load-balancer-DNS-name` y `web-binary-name` con los valores correctos para la configuración. `listener-port`

```
curl http://load-balancer-DNS-name:listener-port/gapwalk-application/
```

Si la aplicación se está ejecutando, verá el siguiente mensaje de salida: `Jics application is running`.

A continuación, ejecute el siguiente comando.

```
curl http://load-balancer-DNS-name:listener-port/jac/api/services/rest/jicsservice/
```

Si la aplicación se está ejecutando, verá el siguiente mensaje de salida: Jics application is running.

```
Jics application is running
```

Si ha configurado Blusam, puede esperar una respuesta vacía cuando ejecute el siguiente comando.

```
curl http://load-balancer-DNS-name:listener-port/bac/api/services/rest/bluesamserver/serverIsUp
```

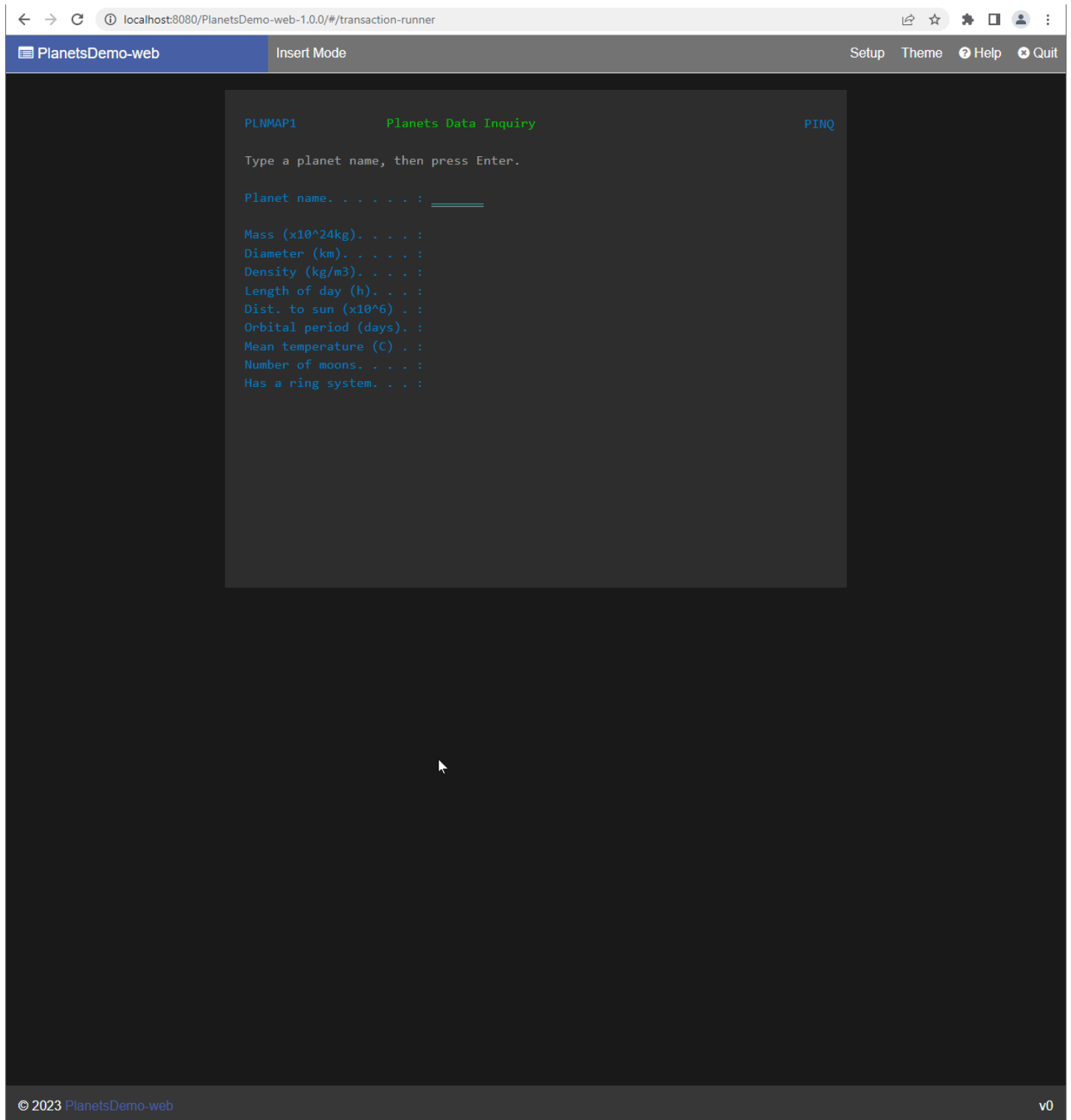
Anote el nombre del binario web (PlanetsDemo-web-1.0.0, si no ha cambiado). Para acceder a la PlanetsDemo aplicación, utilice una URL con el siguiente formato.

```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

Cuando se inicie la PlanetsDemo aplicación, aparecerá la página de inicio.



En el cuadro de texto, escriba PINQ y, a continuación, pulse Intro. Aparece la página de consulta de datos.



Por ejemplo, escriba EARTH en el campo PlanetsDemo del nombre y, a continuación, presione Entrar. Aparece la página del planeta que haya introducido.



The screenshot shows a web browser window with the address bar at localhost:8080/PlanetsDemo-web-1.0.0/#/transaction-runner. The browser tabs include 'PlanetsDemo-web' and 'Insert Mode'. The main content area displays a terminal window with the following text:

```
PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

Planet name. . . . . : EARTH

Mass (x10^24kg). . . . . : 0005.970
Diameter (km). . . . . : 012756
Density (kg/m3). . . . . : 5514
Length of day (h). . . . . : 0024.0
Dist. to sun (x10^6). . . . . : 0149.6
Orbital period (days). . . . . : 00365.2
Mean temperature (C) . . . . . : +015
Number of moons. . . . . : 01
Has a ring system. . . . . : N
```

At the bottom of the browser window, there is a footer with the text '© 2023 PlanetsDemo-web' on the left and 'v0' on the right.

# Modificación del código fuente con el IDE para desarrolladores de Blu Age

Si utilizas el motor AWS de ejecución AWS Blu Age gestionado, puedes utilizar Blu Age Developer para modificar el código fuente generado. Puede que quiera hacerlo si necesita actualizar el código modernizado por alguna razón o si una parte del código fuente heredado no se ha podido modernizar. Puedes acceder a Blu Age Developer a través de Amazon AppStream 2.0. En esta sección se describe cómo configurar Blu Age Developer en la AppStream versión 2.0. También se explica cómo utilizar Blu Age Developer para actualizar el código fuente mediante la aplicación de ejemplo PlanetsDemo.

## Temas

- [Tutorial: Configurar AppStream 2.0 para el IDE de desarrolladores de AWS Blu Age](#)
- [Tutorial: Utilice AWS Blu Age Developer en AppStream 2.0](#)

## Tutorial: Configurar AppStream 2.0 para el IDE de desarrolladores de AWS Blu Age

AWS La modernización del mainframe proporciona varias herramientas a través de Amazon AppStream 2.0. AppStream 2.0 es un servicio de streaming de aplicaciones seguro y totalmente gestionado que permite transmitir aplicaciones de escritorio a los usuarios sin tener que volver a escribirlas. AppStream La versión 2.0 proporciona a los usuarios acceso instantáneo a las aplicaciones que necesitan, con una experiencia de usuario fluida y con gran capacidad de respuesta en el dispositivo que elijan. El uso de la AppStream versión 2.0 para alojar herramientas específicas del motor de ejecución ofrece a los equipos de aplicaciones de los clientes la posibilidad de utilizar las herramientas directamente desde sus navegadores web e interactuar con los archivos de las aplicaciones almacenados en depósitos o repositorios de Amazon S3. CodeCommit

Para obtener información sobre la compatibilidad de los navegadores en la AppStream versión 2.0, consulte [System Requirements and Feature Support \(navegador web\)](#) en la Guía de administración de Amazon AppStream 2.0. Si tiene problemas al utilizar la AppStream versión 2.0, consulte [Solución de problemas de usuario de la AppStream versión 2.0](#) en la Guía de administración de Amazon AppStream 2.0.

Este documento describe cómo configurar el IDE para desarrolladores de AWS Blu Age en una flota AppStream 2.0.

## Temas

- [Requisito previo](#)
- [Paso 1: Crear un bucket de Amazon S3](#)
- [Paso 2: Adjuntar una política al bucket de S3](#)
- [Paso 3: Cargar los archivos a un bucket de Amazon S3](#)
- [Paso 4: Descarga AWS CloudFormation las plantillas](#)
- [Paso 5: Crea la flota con AWS CloudFormation](#)
- [Paso 6: Acceder a una instancia](#)
- [Eliminar recursos](#)

## Requisito previo

Descargue el [archivo comprimido](#) que contiene los artefactos que necesita para configurar el IDE para desarrolladores de AWS Blu Age en la versión AppStream 2.0.

### Note

Es un archivo grande. Si tiene problemas con el tiempo de espera de la operación, le recomendamos que utilice una instancia de Amazon EC2 para mejorar el rendimiento de carga y descarga.

## Paso 1: Crear un bucket de Amazon S3

Cree un bucket de Amazon S3 Región de AWS igual que la flota AppStream 2.0 que va a crear. Este bucket contendrá los artefactos que necesita para completar este tutorial.

## Paso 2: Adjuntar una política al bucket de S3

Adjunte la siguiente política al bucket que ha creado para este tutorial. Asegúrese de reemplazar MYBUCKET por el nombre real del bucket que ha creado.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowAppStream2.0ToRetrieveObjects",
```

```
    "Effect": "Allow",
    "Principal": {
      "Service": "appstream.amazonaws.com"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::MYBUCKET/*"
  ]
}
```

### Paso 3: Cargar los archivos a un bucket de Amazon S3

Descomprima los archivos que ha descargado en el Requisito previo y suba la carpeta `appstream` a su bucket. Al cargar esta carpeta, se crea la estructura correcta en el bucket. Para obtener más información, consulte [Carga de objetos](#) en la Guía del usuario de Amazon S3.

### Paso 4: Descarga AWS CloudFormation las plantillas

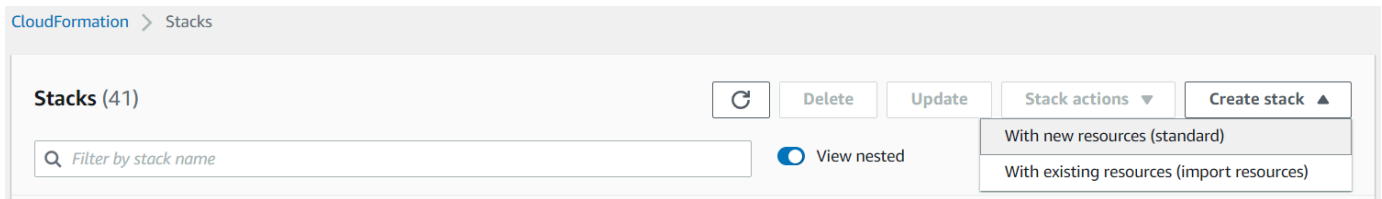
Descarga las siguientes AWS CloudFormation plantillas. Necesita estas plantillas para crear y completar la flota AppStream 2.0.

- [cfn-m2-.yaml appstream-elastic-fleet-linux](#)
- [cfn-m2- appstream-bluage-dev-tools -linux.yaml](#)
- [cfn-m2 appstream-bluage-shared-linux - .yaml](#)
- [appstream-chrome-linuxcfn-m2-.yaml](#)
- [appstream-eclipse-jee-linuxcfn-m2-.yaml](#)
- [appstream-pgadmin-linuxcfn-m2-.yaml](#)

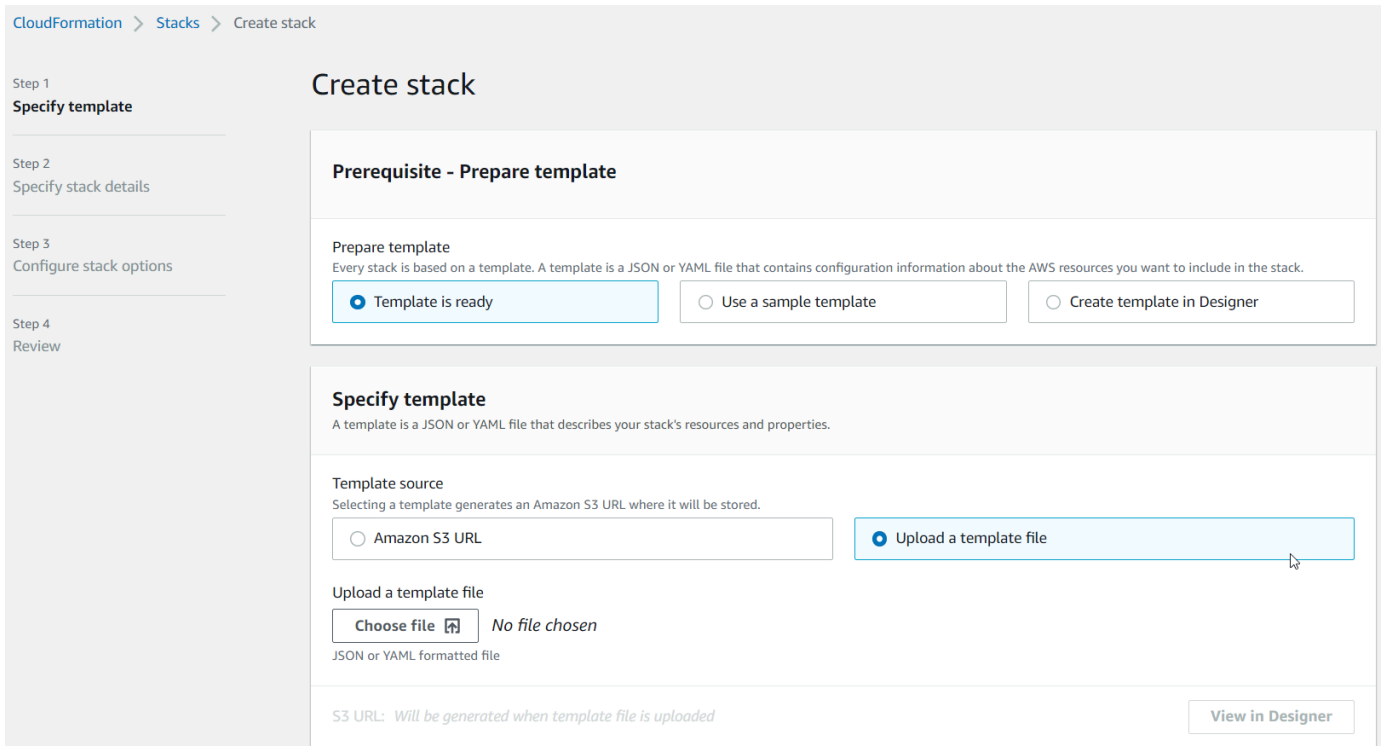
### Paso 5: Crea la flota con AWS CloudFormation

En este paso, utilizarás la `cfn-m2-appstream-elastic-fleet-linux.yaml` AWS CloudFormation plantilla para crear una flota AppStream 2.0 y apilarla para alojar el IDE para desarrolladores de AWS Blu Age. Tras crear la flota y la pila, ejecutará las demás AWS CloudFormation plantillas que descargó en el paso anterior para instalar el IDE para desarrolladores y otras herramientas necesarias.

1. Ve a AWS CloudFormation la consola AWS de administración y selecciona Stacks.
2. Seleccione Pilas, seleccione Crear pila y elija la opción Con recursos nuevos (estándar).



3. En Crear pila, seleccione La plantilla está lista y Cargar un archivo de plantilla:



4. Seleccione Elegir archivo y navegue al archivo `cfn-m2-appstream-elastic-fleet-linux.yaml`. Elija Siguiente.

5. En Especificar los detalles de la pila, indique la siguiente información.

- Un nombre para la pila.
- Su grupo de seguridad predeterminado y dos subredes de ese grupo de seguridad.

**Note**

Las dos subredes del grupo de seguridad deben estar en zonas de disponibilidad diferentes.

6. Elija Siguiente y después elija Siguiente otra vez.

7. Elija Acepto que AWS CloudFormation podría crear recursos de IAM con nombres personalizados. y, a continuación, selecciona Enviar.

- Tras crear la flota, cree CloudFormation pilas con las demás plantillas descargadas para terminar de configurar las aplicaciones. Asegúrese de actualizarlas BucketName cada vez para que apunten al bucket S3 correcto. Puedes editarlo BucketName en la CloudFormation consola. Como alternativa, puede editar los archivos de plantilla directamente y actualizar la propiedad S3Bucket.

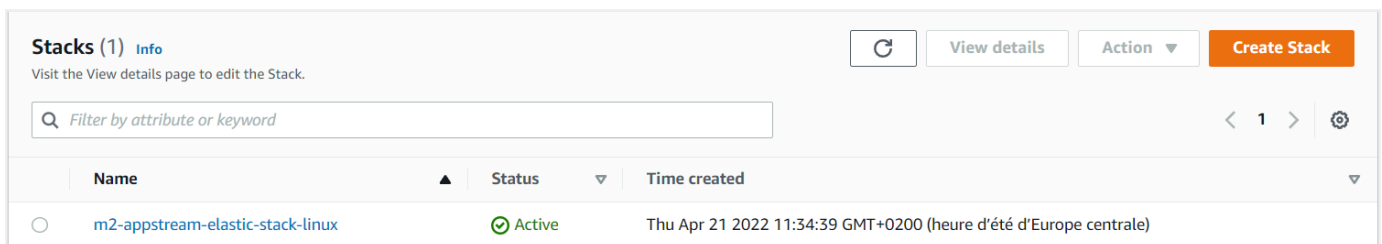
### Note

Las plantillas descargadas esperan encontrar los activos en un bucket de S3 con una estructura de carpetas denominada `appstream/bluage/developer-ide/`. El depósito debe estar en el Región de AWS mismo lugar que la flota que has creado.

## Paso 6: Acceder a una instancia

Tras crear e iniciar la flota, puede crear un enlace temporal para acceder a la flota a través del cliente nativo.

- Navega hasta la AppStream versión 2.0 en AWS Management Console y elige la pila creada anteriormente:



- En la página de detalles de la pila, elija Acción y Create URL en streaming:

### Create Streaming URL: m2-appstream-elastic-stack-linux ✕

User ID \*

This is the User ID the URL will be associated to.

URL Expiration \*

Set the amount of time the URL will be active before expiration.

30 Minutes ▼

Cancel Get URL

3. En Crear URL en streaming, introduzca un ID de usuario arbitrario y una hora de caducidad de la URL y, a continuación, seleccione Obtener URL. Obtendrá una URL que puede usar para transmitir en streaming a un navegador o al cliente nativo. Le recomendamos que haga streaming en el cliente nativo.

## Eliminar recursos

Para conocer el procedimiento para limpiar la pila y las flotas creadas, consulte [Crear una flota y una pila AppStream 2.0](#).

Cuando hayas eliminado los objetos AppStream 2.0, tú o el administrador de la cuenta también podéis limpiar los depósitos de S3 para guardar la configuración de la aplicación y las carpetas de inicio.

### Note

La carpeta principal de un usuario determinado es única en todas las flotas, por lo que es posible que tengas que conservarla si hay otras pilas AppStream 2.0 activas en la misma cuenta.

No puedes usar la consola AppStream 2.0 para eliminar usuarios. Para ello, debe utilizar la API de servicio con la AWS CLI. Para obtener más información, consulte [Administración de grupos de usuarios](#) en la Guía de administración de Amazon AppStream 2.0.

## Tutorial: Utilice AWS Blu Age Developer en AppStream 2.0

Este tutorial te muestra cómo acceder a AWS Blu Age Developer en la AppStream versión 2.0 y cómo usarlo con una aplicación de muestra para que puedas probar sus funciones. Cuando haya acabado este tutorial, podrá seguir los mismos pasos con sus propias aplicaciones.

### Temas

- [Paso 1: Crear una base de datos](#)
- [Paso 2: Acceder al entorno](#)
- [Paso 3: Configurar el tiempo de ejecución](#)
- [Paso 4: Iniciar el IDE de Eclipse](#)
- [Paso 5: Configurar un proyecto de Maven](#)
- [Paso 6: Configurar un servidor de Tomcat](#)
- [Paso 7: Implementar en Tomcat](#)
- [Paso 8: Crear la base de datos de JICS](#)
- [Paso 8: iniciar y probar la aplicación](#)
- [Paso 10: Depurar la aplicación](#)
- [Eliminar recursos](#)

### Paso 1: Crear una base de datos

En este paso, utilizará Amazon RDS para crear una base de datos PostgreSQL administrada que la aplicación de ejemplo utilizará para almacenar la información de configuración.

1. Abra la consola de Amazon RDS.
2. Seleccione Bases de datos > Crear base de datos.
3. Elija Creación estándar > PostgreSQL, deje la versión predeterminada y, a continuación, elija Capa gratuita.
4. Elija el identificador de la instancia de base de datos.
5. En Configuración de credenciales, seleccione Administrar credenciales principales en AWS Secrets Manager. Para obtener más información, consulte [Administración de contraseñas con Amazon RDS y AWS Secrets Manager](#) en la Guía del usuario de Amazon RDS.
6. Asegúrese de que la VPC sea la misma que la que utiliza para la instancia AppStream 2.0. Puede solicitar este valor a su administrador.



7. En Nuevo nombre del grupo de seguridad de VPC, elija Crear nuevo.
8. Establezca Acceso público en Sí.
9. Deje los demás valores predeterminados. Revise estos valores.
10. Elija Crear base de datos.

Para que el servidor de base de datos sea accesible desde su instancia, seleccione el servidor de base de datos en Amazon RDS. En Conectividad y seguridad, elija el grupo de seguridad de VPC para el servidor de base de datos. Este grupo de seguridad se ha creado previamente para usted y debe tener una descripción similar a la de Creado por la consola de administración de RDS. Seleccione Acción > Editar reglas de entrada, elija Agregar regla y cree una regla de tipo PostgreSQL. Para el origen de la regla, utilice el grupo de seguridad predeterminado. Puede empezar a escribir el nombre del origen en el campo Origen y aceptar el ID sugerido. Por último, seleccione Guardar reglas.

## Paso 2: Acceder al entorno

En este paso, accederá al entorno de desarrollo de AWS Blu Age en la AppStream versión 2.0.

1. Póngase en contacto con su administrador para conocer la forma correcta de acceder a su instancia AppStream 2.0. Para obtener información general sobre posibles clientes y configuraciones, consulte [Métodos y clientes de acceso AppStream 2.0](#) en la Guía de administración de Amazon AppStream 2.0. Considere la posibilidad de utilizar el cliente nativo para disfrutar de la mejor experiencia.
2. En la AppStream versión 2.0, elija Escritorio.

## Paso 3: Configurar el tiempo de ejecución

En este paso, configuras el tiempo de ejecución de AWS Blu Age. Debe configurar el tiempo de ejecución en el primer lanzamiento y de nuevo si se le notifica una actualización del tiempo de ejecución. Este paso rellena la carpeta .m2.

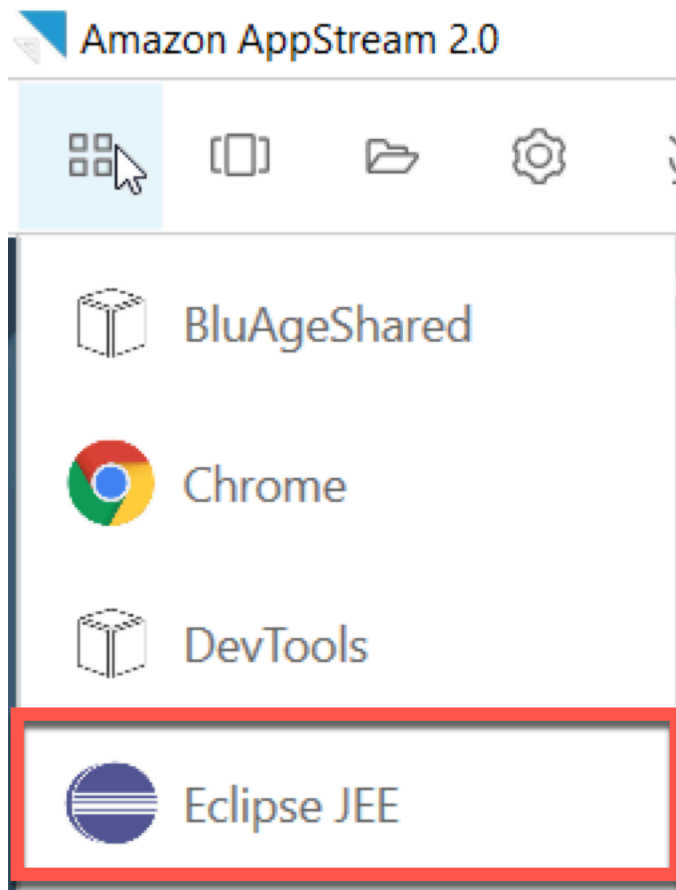
1. Seleccione Aplicaciones en la barra de menús y, a continuación, elija Terminal.
2. Escriba el siguiente comando:

```
~/_install-velocity-runtime.sh
```

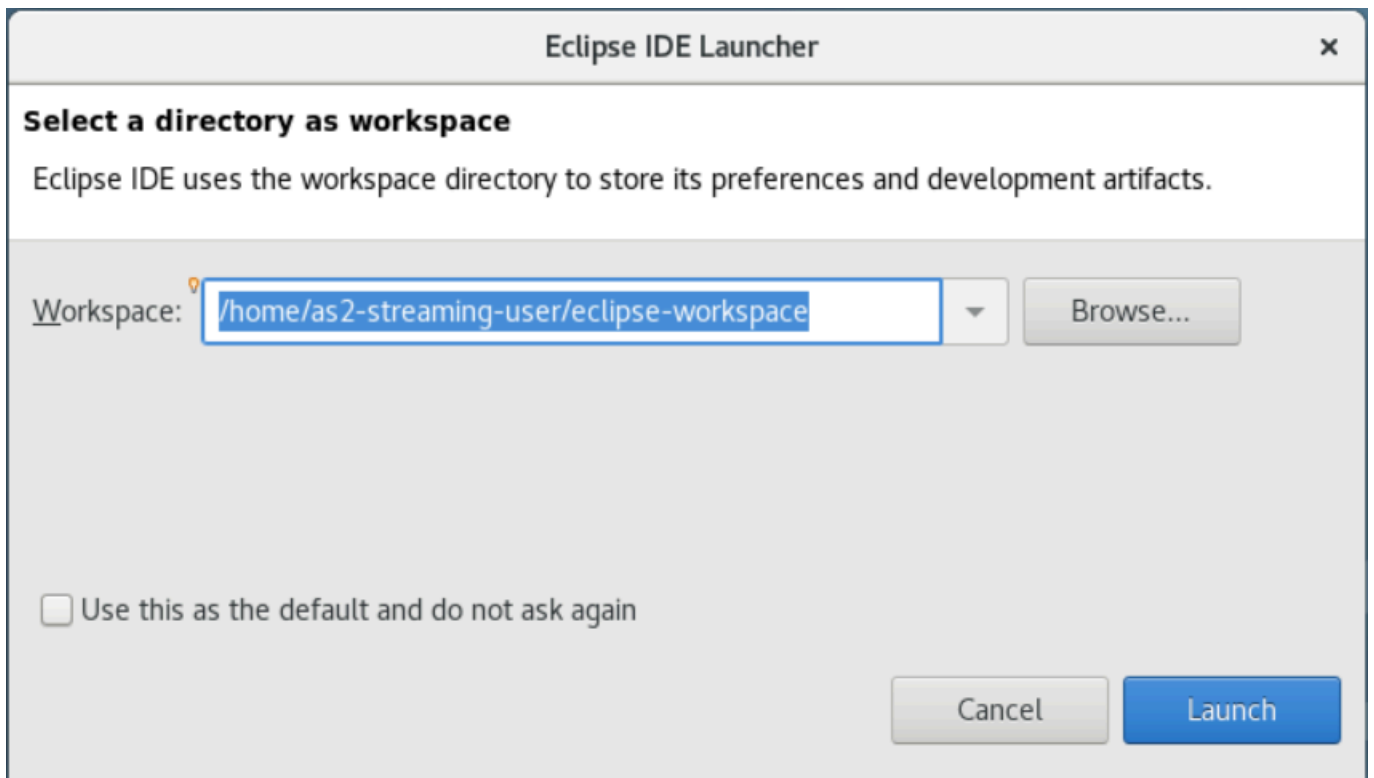
## Paso 4: Iniciar el IDE de Eclipse

En este paso, iniciará el IDE de Eclipse y elegirá la ubicación en la que desee crear un espacio de trabajo.

1. En la AppStream versión 2.0, elija el icono de iniciar aplicación en la barra de herramientas y, a continuación, elija Eclipse JEE.



2. Cuando se abra el lanzador, introduzca la ubicación en la que quiere crear su espacio de trabajo y elija Launch.



Si lo desea, puede lanzar Eclipse desde la línea de comandos, de la siguiente manera:

```
~/eclipse &
```

## Paso 5: Configurar un proyecto de Maven


En este paso, importará un proyecto de Maven para la aplicación de demostración Planets.

1. Sube el [PlanetsDemoarchivo -pom.zip](#) a tu carpeta de inicio. Para ello, puede utilizar la característica “My Files” del cliente nativo.
2. Utilice la herramienta de línea de comandos de unzip para extraer los archivos.
3. Desplácese por la carpeta descomprimida y abra el `pom.xml` raíz de su proyecto en un editor de texto.
4. Edite la propiedad `gapwalk.version` para que coincida con el motor de tiempo de ejecución de AWS Blu Age instalado.

Si no conoce con seguridad la versión instalada, ejecute el siguiente comando en un terminal:

```
cat ~/runtime-version.txt
```

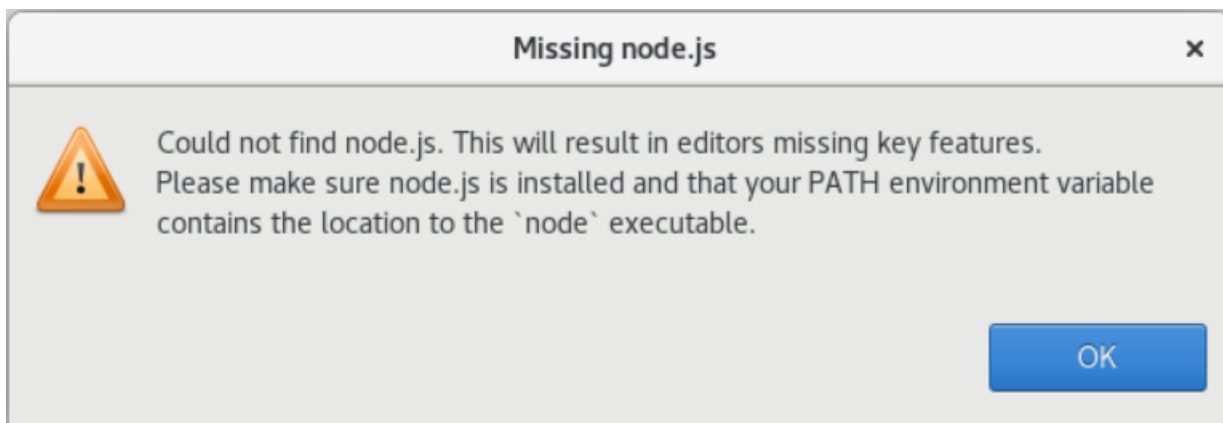
Este comando imprime la versión de tiempo de ejecución disponible actualmente, por ejemplo `3.1.0-b3257-dev`.

 Note

No incluya el sufijo `-dev` en `gapwalk.version`. Por ejemplo, un valor válido sería `<gapwalk.version>3.1.0-b3257</gapwalk.version>`.

5. En Eclipse, elija File e Import. En la ventana de diálogo Import, expanda Maven y elija Existing Maven Projects. Elija Next.
6. En Import Maven Projects, proporcione la ubicación de los archivos extraídos y elija Finish.

Puede omitir la siguiente ventana emergente de forma segura. Maven descarga una copia local de `node.js` para compilar la parte angular (`*-web`) del proyecto:



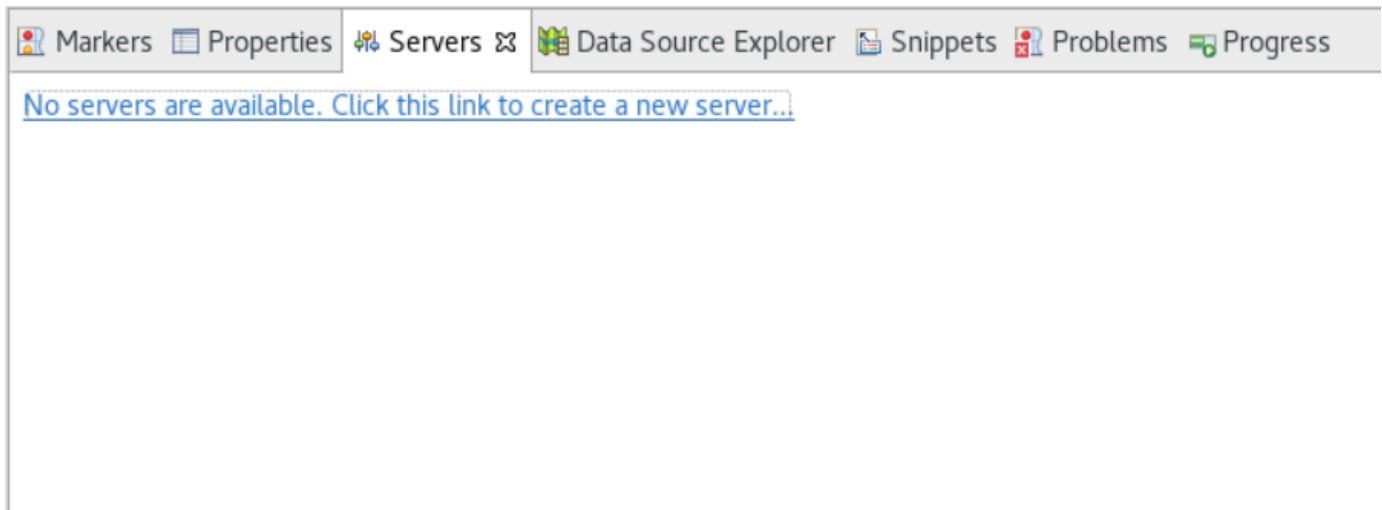
Espere hasta el final de la compilación. Puede seguir la compilación en la vista de Progreso.

7. En Eclipse, seleccione el proyecto y elija Run as. Después, seleccione Maven install. Una vez que la instalación de Maven se haya realizado correctamente, se creará el archivo `war` en `PlanetsDemoPom/PlanetsDemo-web/target/PlanetsDemo-web-1.0.0.war`.

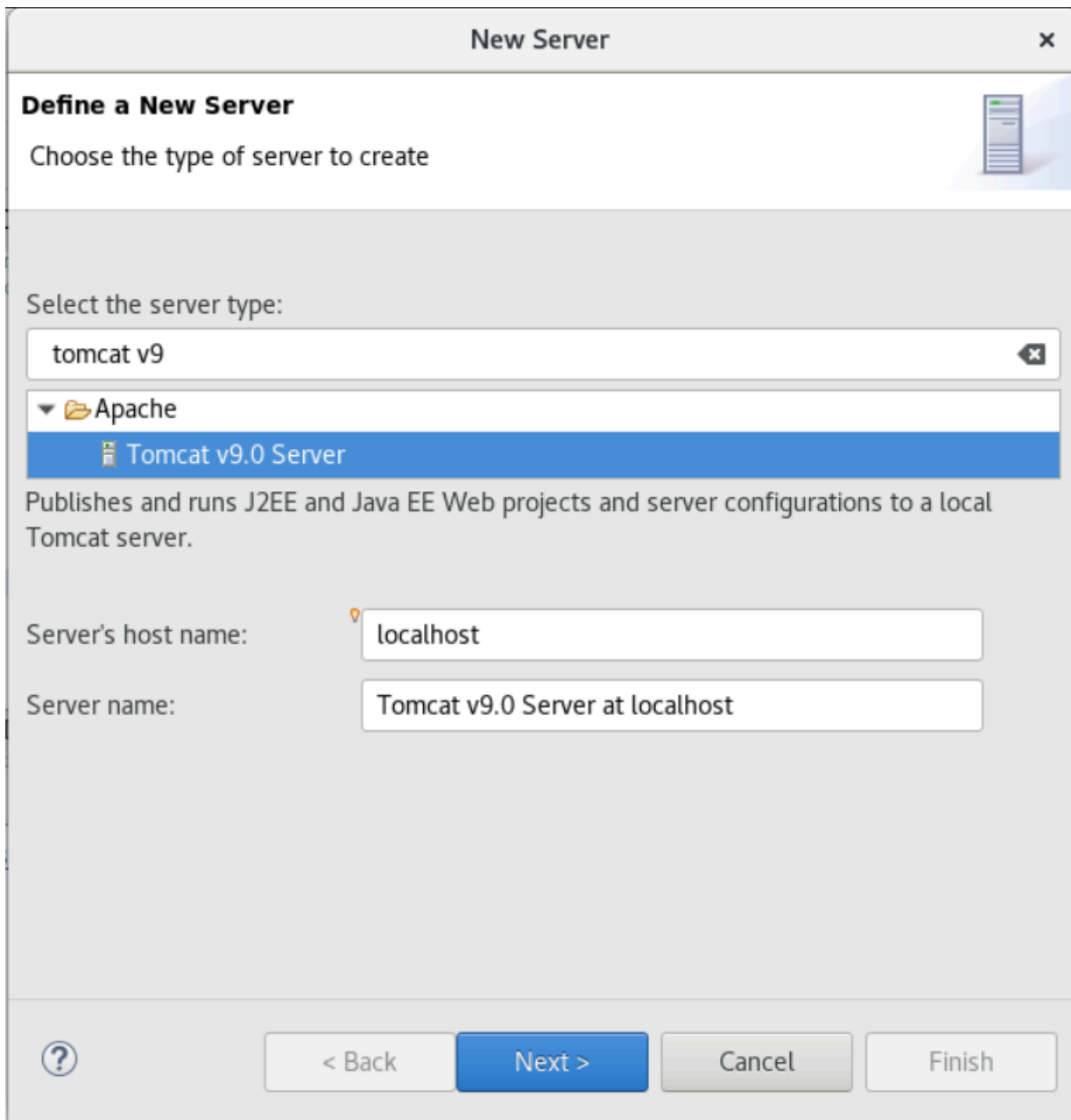
## Paso 6: Configurar un servidor de Tomcat

En este paso, configurará un servidor de Tomcat en el que implementará e iniciará la aplicación compilada.

1. En Eclipse, seleccione Window > Show View > Servers para mostrar la vista Servers:

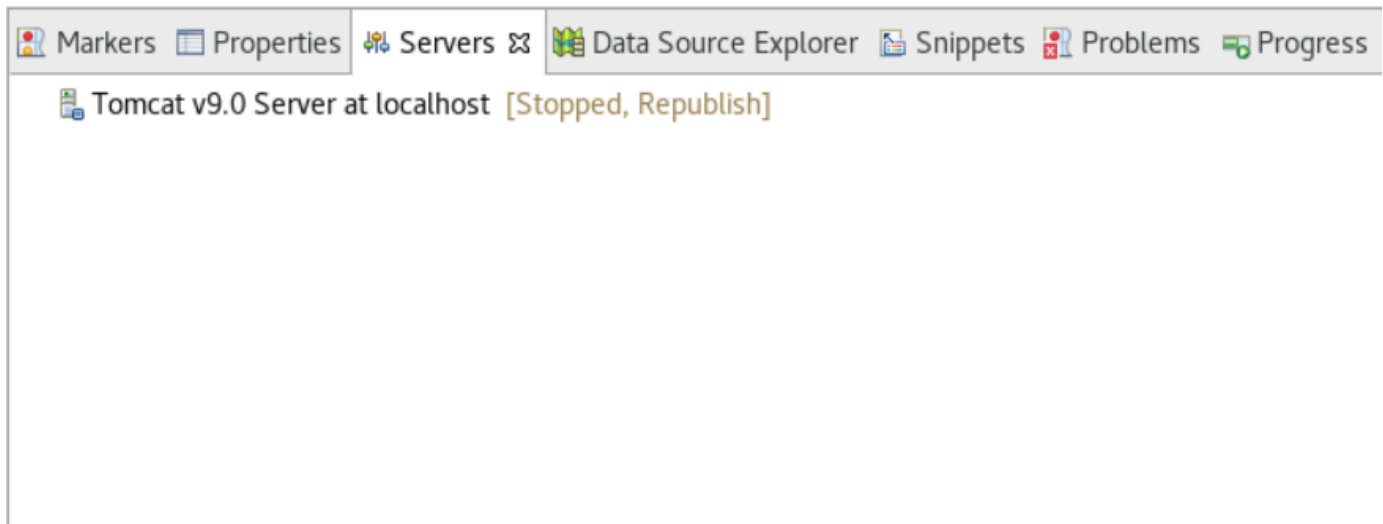


2. Seleccione No servers are available. Haga clic en este enlace para crear un nuevo servidor. Se abrirá el asistente New Server. En el campo Select the server type del asistente, escriba tomcat v9 y elija Tomcat v9.0 Server. A continuación, elija Next.



3. Seleccione Browse y elija la carpeta tomcat situada en la raíz de la carpeta de inicio. Deje el JRE en su valor predeterminado y seleccione Finish.

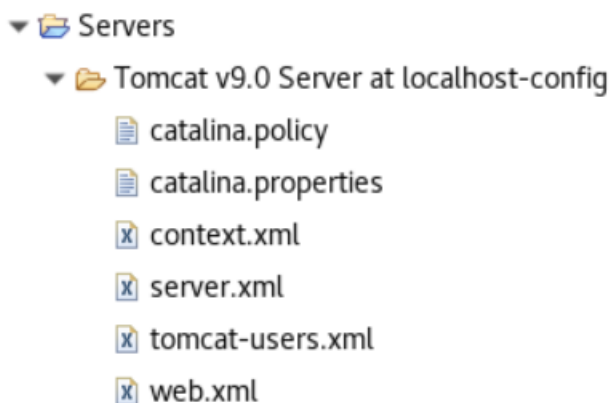
Se ha creado un proyecto Servers en el espacio de trabajo y ahora hay un servidor Tomcat v9.0 disponible en la vista Servers. Aquí es donde se implementará e iniciará la aplicación compilada:



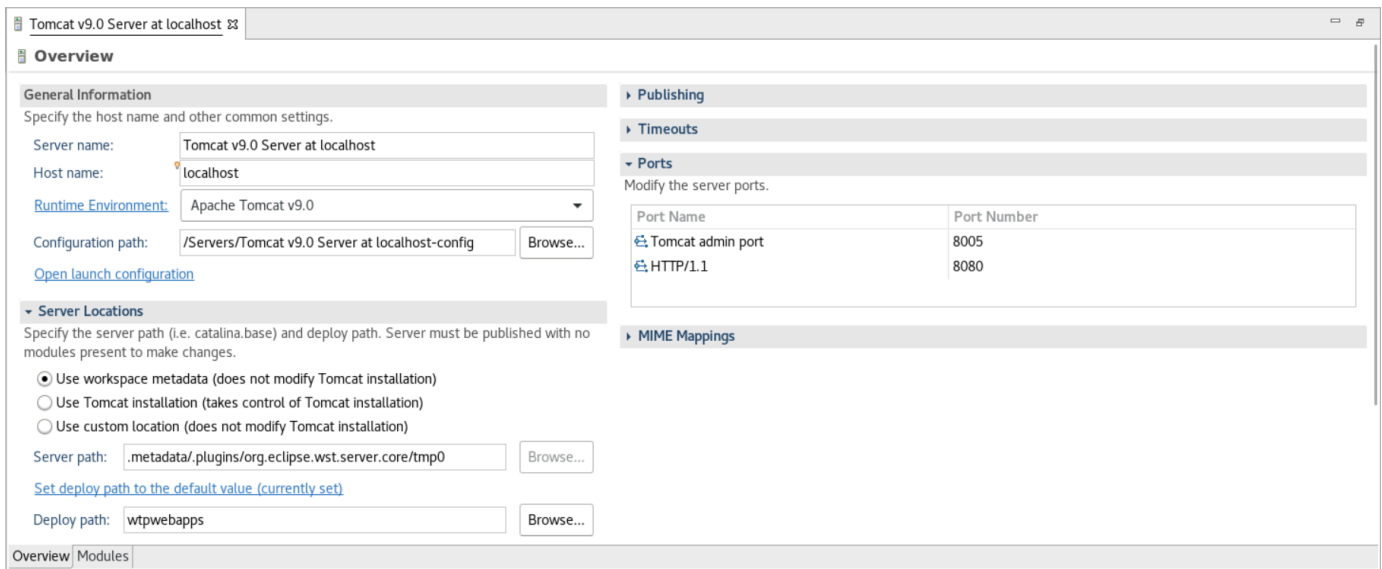
## Paso 7: Implementar en Tomcat

En este paso, implementará la aplicación de ejemplo Planets en el servidor de Tomcat para poder ejecutar la aplicación.

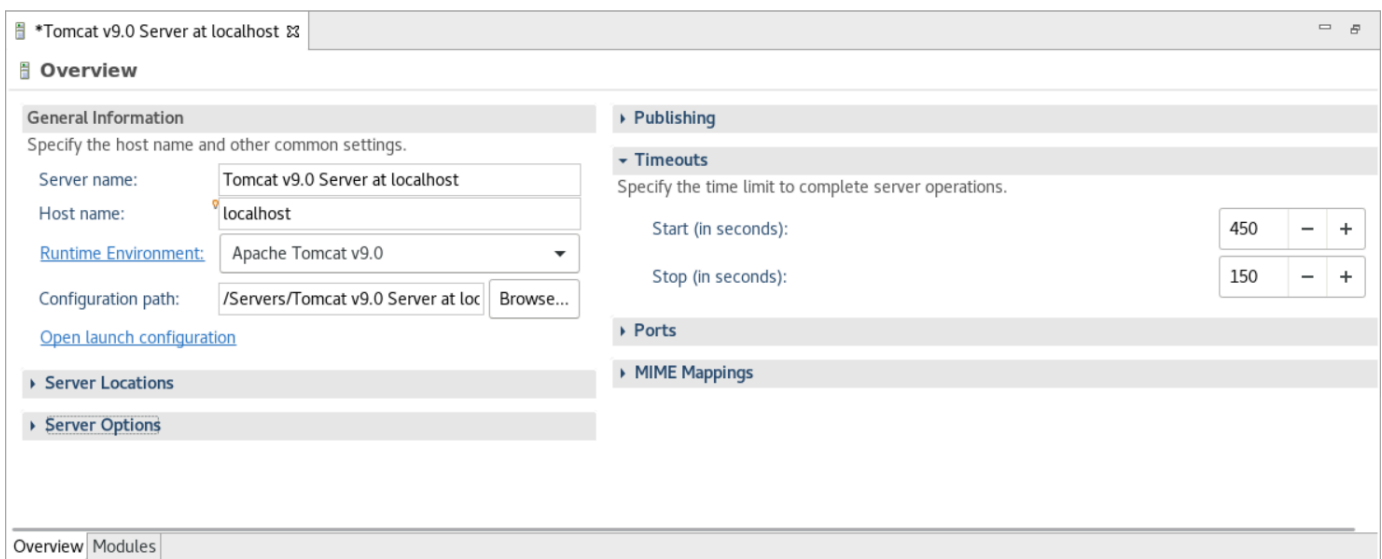
1. Seleccione el archivo PlanetsDemo-web y elija Run As > Maven install. Vuelva a seleccionar PlanetsDemo-web y elija Refresh para asegurarse de que la interfaz compilada con npm esté correctamente compilada en un archivo .war y Eclipse la detecte.
2. Cargue el [PlanetsDemoarchivo -runtime.zip](#) a la instancia y descomprima el archivo en una ubicación accesible. Esto garantiza que la aplicación de demostración pueda acceder a las carpetas y archivos de configuración que necesite.
3. Copie el contenido de PlanetsDemo-runtime/tomcat-config en la subcarpeta Servers/Tomcat v9.0... que ha creado para su servidor Tomcat:



- Abra la entrada del servidor tomcat v9.0 en la vista Servers. Aparece el editor de propiedades del servidor:

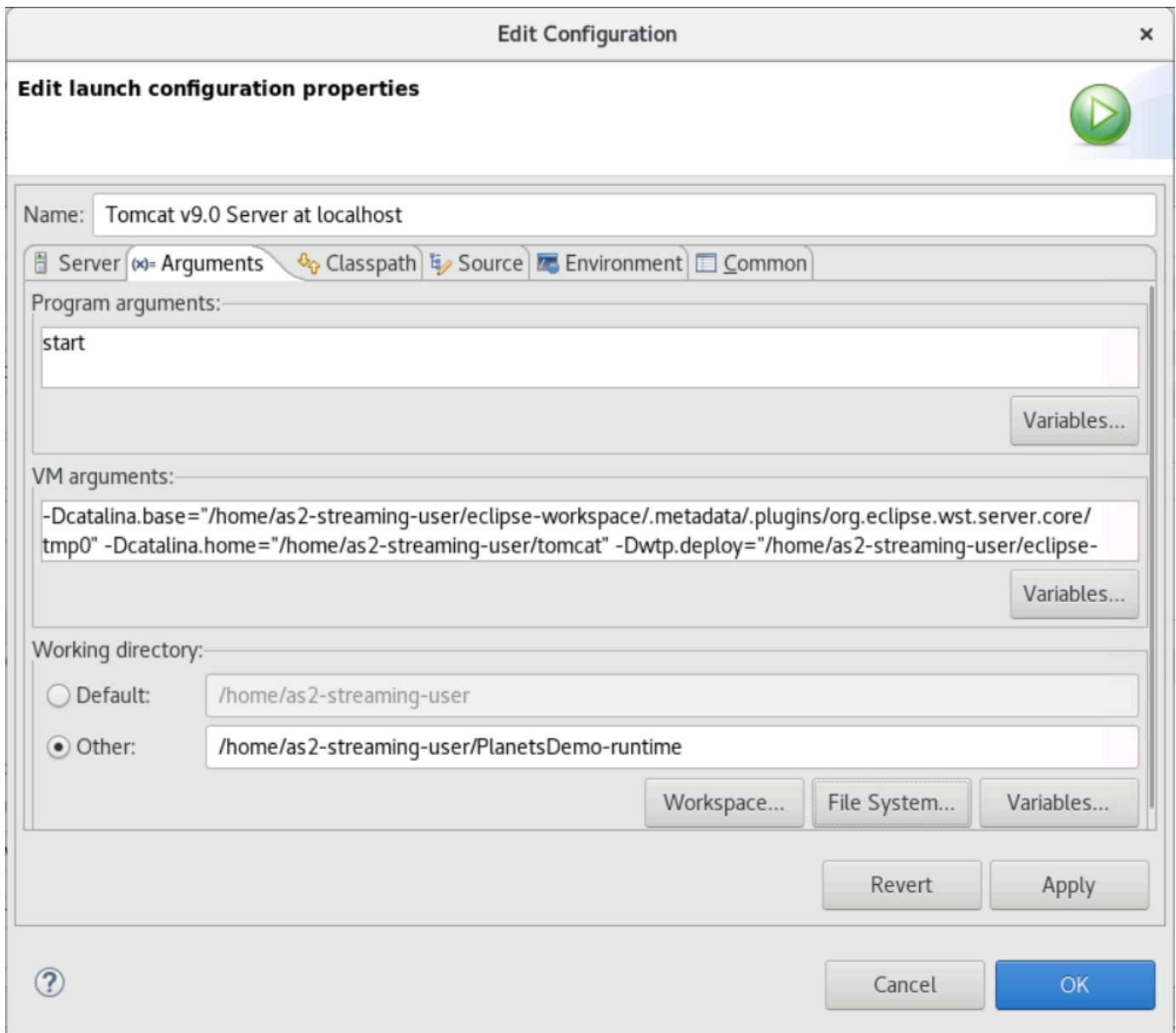


- En la pestaña Overview, aumente los valores de Timeouts a 450 segundos para Start y 150 segundos para Stop, como se muestra a continuación:



- Seleccione Open launch configuration. Aparecerá el asistente. En el asistente, vaya a la carpeta Arguments y, en Working directory, elija Other. Elija File System y navegue hasta la carpeta PlanetsDemo-runtime descomprimida anteriormente. Esta carpeta debe contener una subcarpeta directa llamada config.

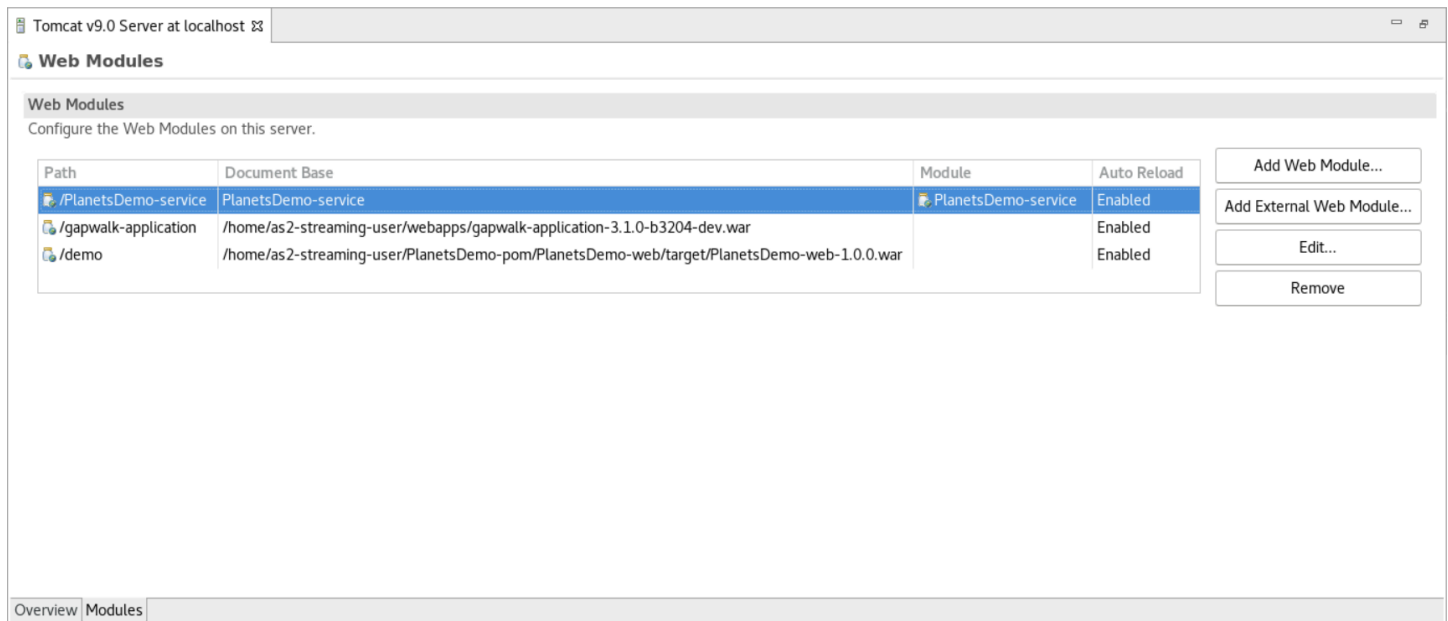




7. Seleccione la pestaña Modules del editor de propiedades del servidor y realice los siguientes cambios:
  - Seleccione Add Web Module and añada PlanetsDemo-service.
  - Elija Add External Web Module. Aparece la ventana Add Web Module. Realice los siguientes cambios:
    - En Document base, elija Browse y vaya a ~/webapps/gapwalk-application...war
    - En Path, introduzca /gapwalk-application.
  - Seleccione Aceptar.
  - Vuelva a seleccionar Add External Web Module y realice los siguientes cambios:

- En Document base, introduzca la ruta a la interfaz .war (in PlanetsDemo-web/target)
- En Path, introduzca /demo
- Seleccione Aceptar.
- Guarde las modificaciones del editor (Ctrl + S).

El editor debe de mostrar un contenido similar al del siguiente ejemplo.



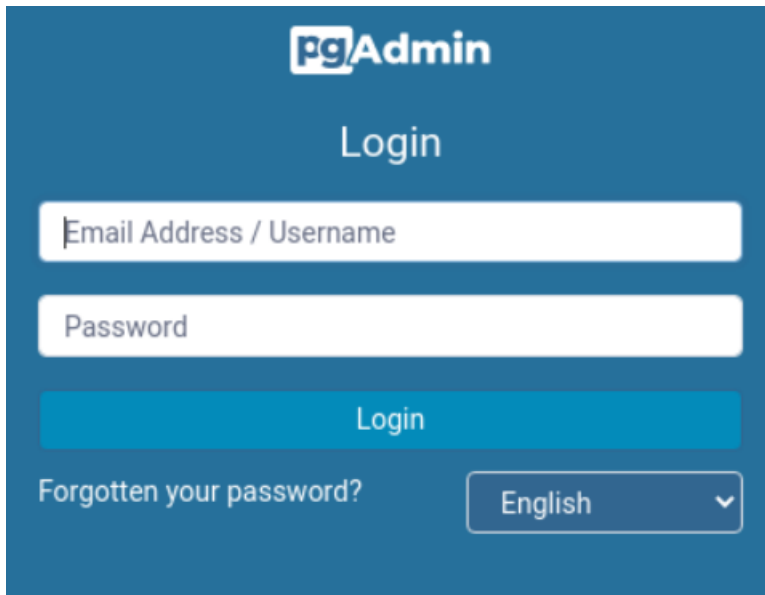
## Paso 8: Crear la base de datos de JICS

En este paso, se conectará a la base de datos que ha creado en [Paso 1: Crear una base de datos](#).

1. Desde la instancia AppStream 2.0, ejecute el siguiente comando en una terminal para iniciar: `pgAdmin`

```
./pgadmin-start.sh
```

2. Elija una dirección de correo electrónico y una contraseña como identificadores para el inicio de sesión. Tome nota de la URL proporcionada (normalmente es `http://127.0.0.1:5050`). Lance Google Chrome en la instancia, copie y pegue la URL en el navegador e inicie sesión con sus identificadores.



pgAdmin

Login

Email Address / Username

Password

Login

Forgotten your password?

English

- Después de iniciar sesión, elija Add New Server e introduzca la información de conexión a la base de datos creada anteriormente de la siguiente manera.

The image shows a 'Register - Server' dialog box with the 'Connection' tab selected. The fields are as follows:

Field	Value
Host name/address	xxx.yyy.zzz.rds.amazonaws.com
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	<input type="checkbox"/>
Password	.....
Save password?	<input type="checkbox"/>
Role	
Service	

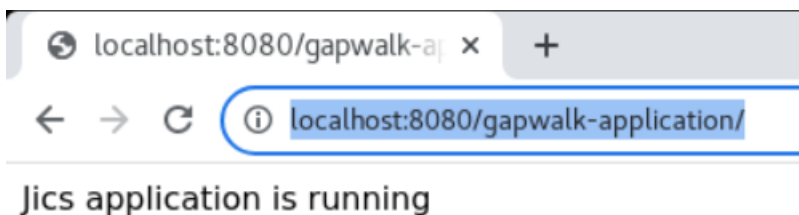
Buttons at the bottom: Close, Reset, Save.

4. Cuando se conecte al servidor de base de datos, utilice Object > Create > Database y cree una nueva base de datos llamada jics.
5. Edite la información de conexión a la base de datos que ha utilizado la aplicación de demostración. Esta información se define en `PlanetsDemo-runtime/config/application-main.yml`. Busque la entrada `jicsDs`. Para recuperar los valores de `username` y `password`, vaya a la base de datos en la consola de Amazon RDS. En la pestaña Configuración, en ARN de credenciales maestras, seleccione Administrar en Secrets Manager. A continuación, en la consola de Secrets Manager, en el secreto, selecciona Recuperar valor del secreto.

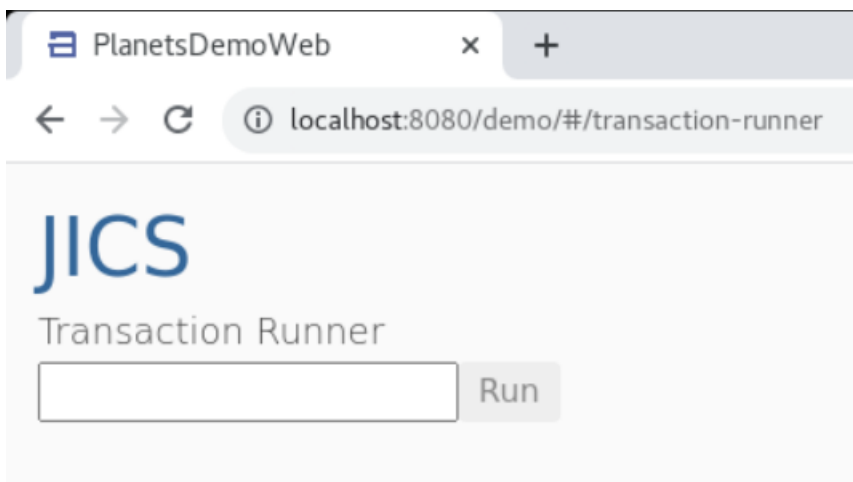
## Paso 8: iniciar y probar la aplicación

En este paso, iniciará el servidor de Tomcat y la aplicación de demostración para poder probarla.

1. Para iniciar el servidor de Tomcat y las aplicaciones implementadas anteriormente, seleccione la entrada del servidor en la vista Servers y elija Start. Aparece una consola que muestra los registros de inicio.
2. Compruebe el estado del servidor en la vista Servers o espere a que aparezca el mensaje de Server startup in [xxx] milliseconds en la consola. Cuando se inicie el servidor, compruebe que gapwalk-application esté correctamente implementada. Para ello, acceda a la URL `http://localhost:8080/gapwalk-application` en un navegador Google Chrome. Debería ver lo siguiente:

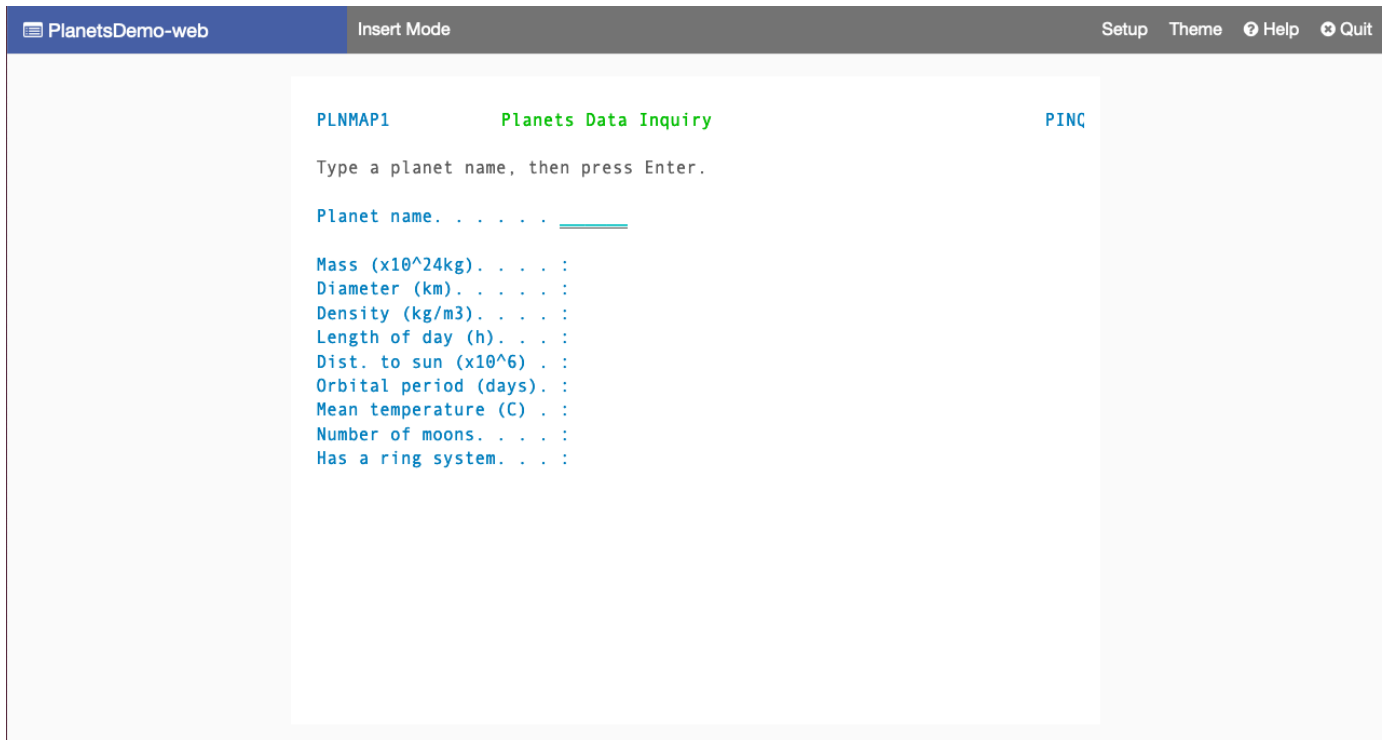


3. Acceda a la interfaz de la aplicación implementada desde Google Chrome en `http://localhost:8080/demo`. Debería aparecer la página Transaction Launcher siguiente.



4. Para iniciar la transacción de la aplicación, introduzca PINQ en el campo de entrada y elija Run (o presione Intro).

Debería aparecer la pantalla de la aplicación de demostración.



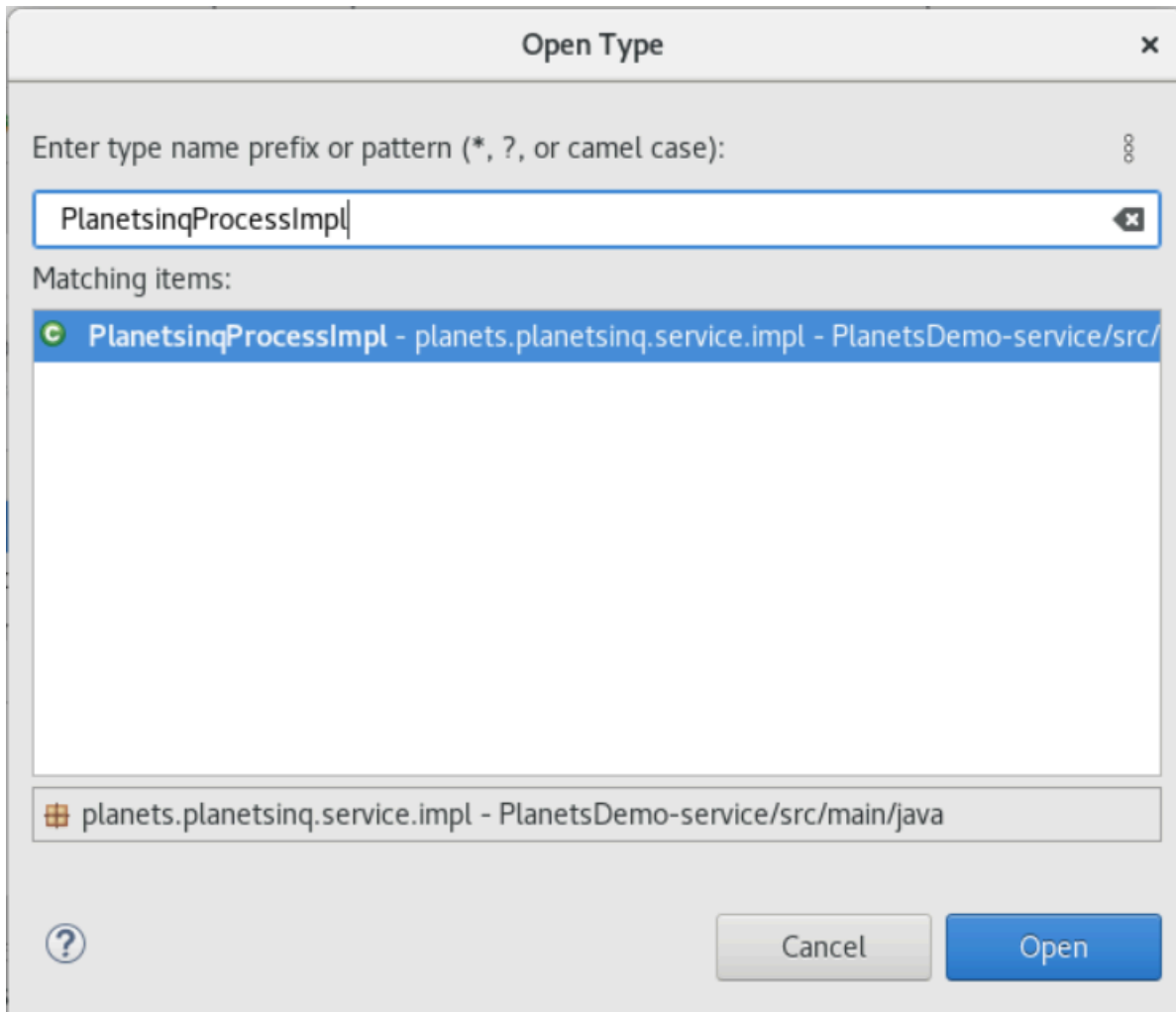
5. Escriba el nombre de un planeta en el campo correspondiente y pulse Intro.



## Paso 10: Depurar la aplicación

En este paso, probará las características de depuración estándar de Eclipse. Estas características están disponibles cuando se trabaja en una aplicación modernizada.

1. Para abrir la clase de servicio principal, pulse **Ctrl + Mayús + T**. Después, introduzca `PlanetsinqProcessImpl`.



2. Navegue hasta el método `searchPlanet` y coloque allí un punto de ruptura.
3. Seleccione el nombre del servidor y elija **Restart in Debug**.
4. Repita los pasos anteriores. Es decir, acceda a la aplicación, introduzca el nombre de un planeta y pulse **Intro**.

Eclipse detendrá la aplicación en el método `searchPlanet`. Ahora puede examinarlo.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para que no se le cobre por ellos. Realice los siguientes pasos:

- Si la aplicación Planets sigue ejecutándose, deténgala.
- Elimine la base de datos que ha creado en [Paso 1: Crear una base de datos](#). Para obtener más información, consulte [Eliminación de una instancia de base de datos](#).



# Redefinición de plataformas de aplicaciones con Micro Focus

En esta sección se describe cada paso del proceso de redefinición de plataformas. Describe todas las tareas e incluye información sobre la configuración y el funcionamiento del entorno de ejecución de modernización de AWS mainframe en Amazon EC2.

## Temas

- [Configuración de tiempo de ejecución de Micro Focus \(en Amazon EC2\)](#)
- [Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#)
- [Tutorial: Configurar Enterprise Analyzer en 2.0 AppStream](#)
- [Tutorial: Configurar Micro Focus Enterprise Developer en AppStream 2.0](#)
- [Configuración de la automatización para las sesiones en streaming de Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#)
- [Ver conjuntos de datos como tablas y columnas en Enterprise Developer](#)
- [Tutorial: Uso de plantillas con Micro Focus Enterprise Developer](#)
- [Tutorial: Configuración de la versión de Micro Focus para la aplicación BankDemo de muestra](#)
- [Tutorial: Configuración de una canalización de CI/CD para su uso con Micro Focus Enterprise Developer](#)
- [Utilidades por lotes en la modernización de AWS mainframes](#)

## Configuración de tiempo de ejecución de Micro Focus (en Amazon EC2)

AWS Mainframe Modernization proporciona varias Amazon Machine Images (AMI) que incluyen productos con licencia de Micro Focus. Estas AMI le permiten aprovisionar rápidamente instancias de Amazon Elastic Compute Cloud (Amazon EC2) para que sean compatibles con los entornos de Micro Focus que usted controla y administra. En este tema se proporcionan los pasos necesarios para acceder a estas AMI y lanzarlas. El uso de estas AMI es totalmente opcional y no es obligatorio para completar los tutoriales de esta guía del usuario.

## Temas

- [Requisitos previos](#)
- [Cree puntos de conexión de Amazon VPC para Amazon S3](#)
- [Solicitud de actualización de la lista de permitidos de la cuenta](#)
- [Crear el rol AWS Identity and Access Management](#)
- [Concesión de permisos necesarios con License Manager](#)
- [Suscríbase a Amazon Machine Images](#)
- [Lance una instancia de Micro Focus para la modernización de AWS mainframe](#)
- [Subred o VPC sin acceso a Internet](#)
- [Solución de problemas de licencia](#)

## Requisitos previos

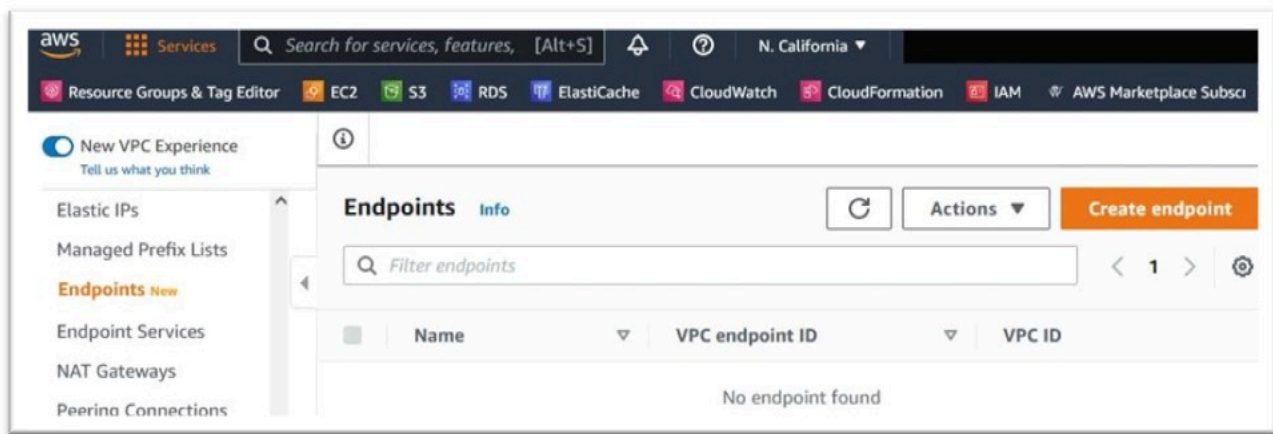
Asegúrese de cumplir los siguientes requisitos previos:

- Acceso de administrador a la cuenta donde se van a crear las instancias de Amazon EC2.
- Identifique el Región de AWS lugar donde se crearán las instancias de Amazon EC2 y compruebe que el servicio de modernización del AWS mainframe esté disponible. Consulte [Servicios de AWS por región](#). Asegúrese de elegir una región en la que el servicio esté disponible.
- Identifique la Amazon Virtual Private Cloud (Amazon VPC) donde se van a crear las instancias de Amazon EC2.

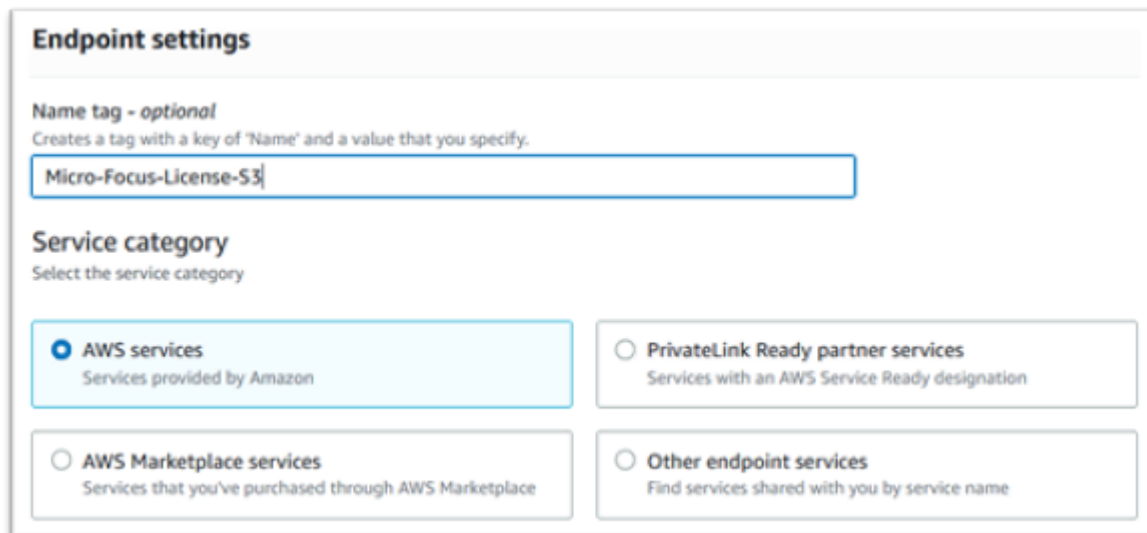
## Cree puntos de conexión de Amazon VPC para Amazon S3

En esta sección, se crea un punto de conexión de Amazon VPC para Amazon S3.

1. Vaya a Amazon VPC en la AWS Management Console.
2. En el panel de navegación, elija Puntos de conexión.
3. Elija Crear punto de conexión.



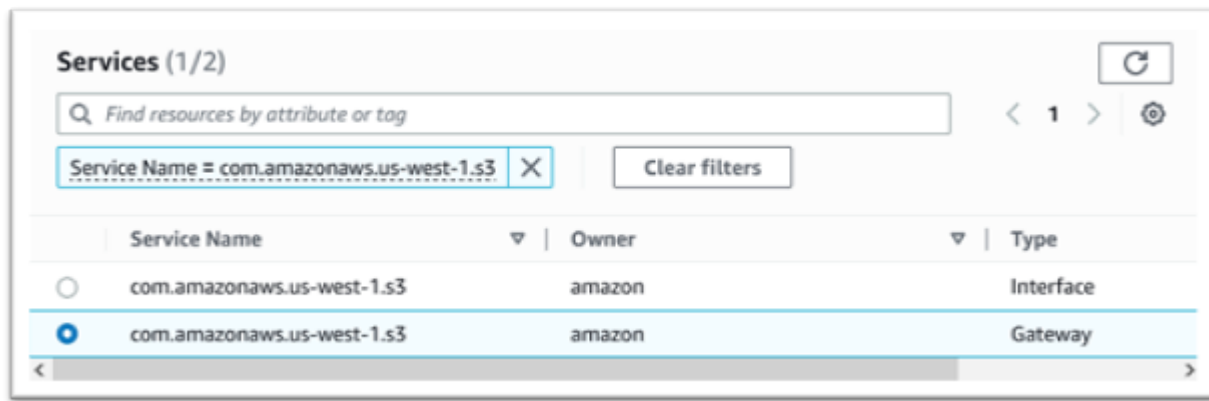
4. Introduzca una etiqueta con un nombre significativo, por ejemplo: “Micro-Focus-License-S3”.
5. Elija Servicios de AWS en Categoría de servicio.



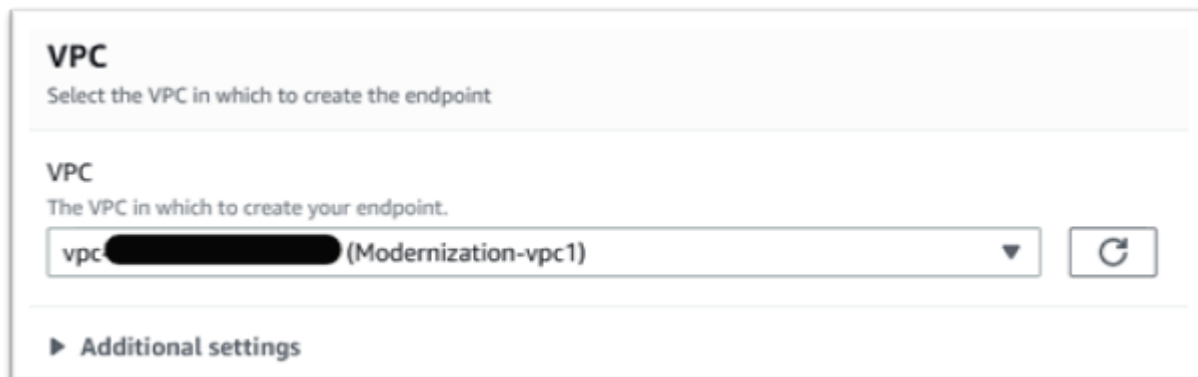
6. En Servicios, busque el servicio de puerta de enlace de Amazon S3: com.amazonaws.[región].s3.

Para us-west-1 sería: com.amazonaws.us-west-1.s3.

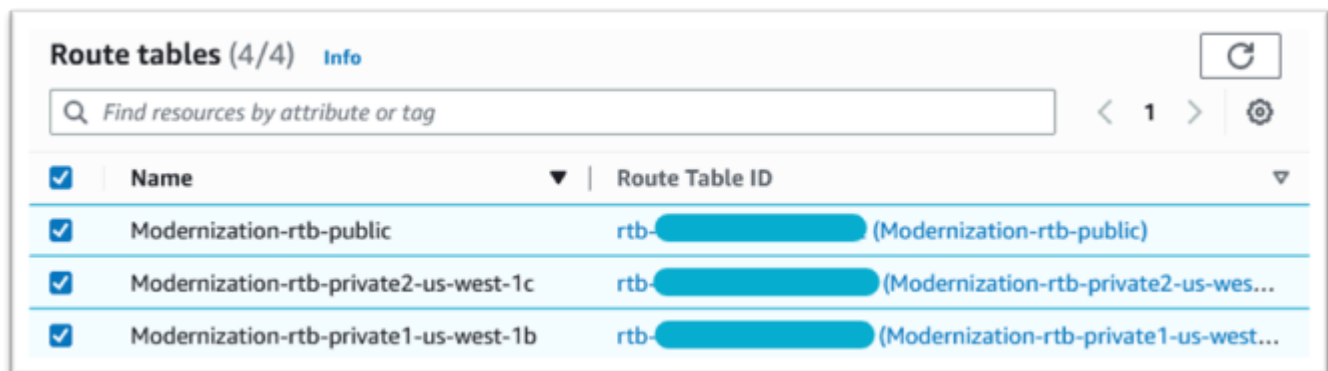
7. Elija el servicio Puerta de enlace.



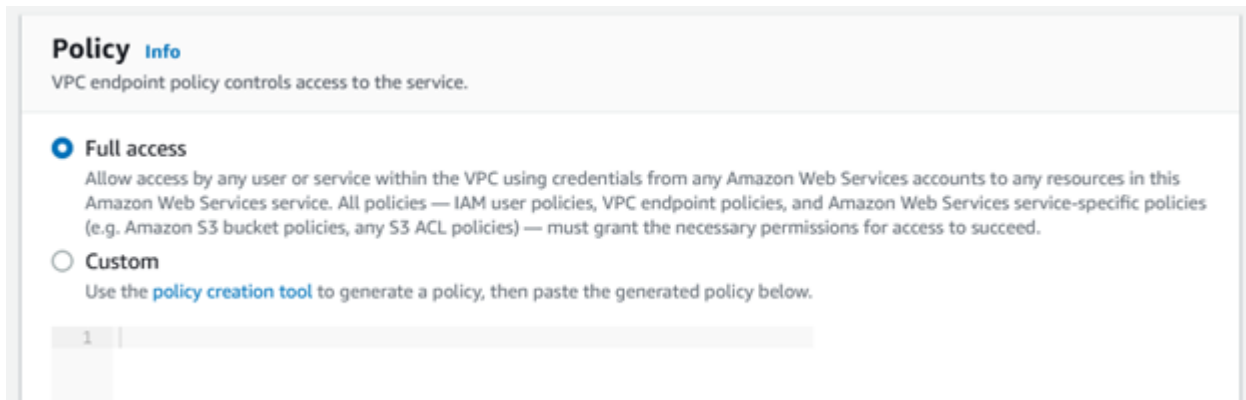
8. Como VPC, elija la VPC que vaya a utilizar.



9. Elija todas las tablas de enrutamiento de la VPC.



10. En Política, seleccione Full Acces.



11. Elija Crear punto de conexión.

## Solicitud de actualización de la lista de permitidos de la cuenta

Hable con su AWS representante para que su cuenta figure en la lista de permitidos para las AMI de modernización de AWS mainframes. Proporcione la siguiente información:

- El Cuenta de AWS ID.
- El Región de AWS lugar donde se creó el punto final de Amazon VPC.
- El ID del punto de conexión de Amazon VPC para Amazon S3 que se creó en [Cree puntos de conexión de Amazon VPC para Amazon S3](#). Este es el identificador vpce-xxxxxxxxxxxxxxxx del punto de conexión com.amazonaws.[region].s3 Gateway
- El número de licencias necesarias en todas las instancias de AMI de Amazon EC2 de Micro Focus Enterprise Suite.

Se requiere una licencia por núcleo de CPU (por cada 2 vCPU para la mayoría de las instancias de Amazon EC2).

Para obtener más información, consulte [Optimización de las opciones de CPU](#).

El número solicitado se puede ajustar en el futuro mediante AWS.

### Note

El AWS representante debe abrir el ticket de soporte para la solicitud de la lista de permitidos. No se puede solicitar directamente y la solicitud puede tardar varios días en completarse.

## Crear el rol AWS Identity and Access Management

Cree una AWS Identity and Access Management política y un rol para que los utilicen las instancias Amazon EC2 de modernización del AWS mainframe. Al crear el rol a través de la consola de IAM, se creará un perfil de instancia asociado con el mismo nombre. Al asignar este perfil de instancia a las instancias de Amazon EC2, se pueden asignar licencias de Micro Focus. Para obtener más información sobre los perfiles de instancia, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias de Amazon EC2](#).

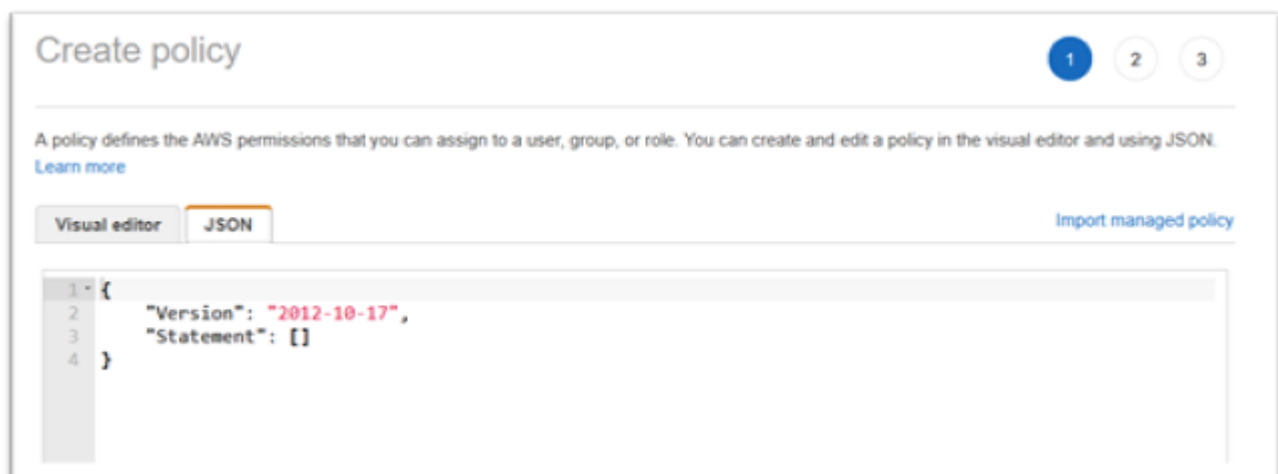
### Crear una política de IAM

Primero se crea una política de IAM y después se asocia al rol.

1. Navegue hasta AWS Identity and Access Management . AWS Management Console
2. Elija Políticas y, a continuación, Crear política.



3. Seleccione la pestaña JSON.



4. Sustituya us-west-1 el siguiente JSON por el Región de AWS lugar donde se definió el punto de conexión de Amazon S3 y, a continuación, copie y pegue el JSON en el editor de políticas.

```
{
  "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**

Las acciones bajo el Sid `OtherRequiredActions` no admiten permisos de recursos y deben especificar `*` en el elemento de recurso.

1 {  
2 "Version": "2012-10-17",  
3 "Statement": [  
4 {  
5 "Sid": "S3WriteObject",  
6 "Effect": "Allow",  
7 "Action": [  
8 "s3:PutObject"  
9 ],  
10 "Resource": [  
11 "arn:aws:s3::aws-supernova-marketplace-us-west-1-prod/\*\*"  
12 ]  
13 },  
14 {  
15 "Sid": "OtherRequiredActions",  
16 "Effect": "Allow",  
17 "Action": [  
18 "sts:GetCallerIdentity",  
19 "ec2:DescribeInstances",  
20 "license-manager:ListReceivedLicenses"  
21 ],  
22 "Resource": [  
23 "\*" ]  
24 ]  
25 } ]  
}

Character count: 339 of 6,144

Cancel Next: Tags

5. Elija Siguiente: etiquetas.

Create policy

1 2 3

**Add tags - optional**  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Previous Next: Review

6. Si lo desea, introduzca cualquier etiqueta y, a continuación, seleccione Siguiente: Revisar.

7. Introduzca un nombre para la política, por ejemplo, "Microfocus Licensing-Policy". Si lo desea, introduzca una descripción, por ejemplo: «Debe adjuntarse un rol que incluya esta política a cada instancia de Amazon EC2 de modernización de AWS mainframe».



**Create policy**

1 2 3

### Review policy

**Name\***

Use alphanumeric and '+', '@', '\_' characters. Maximum 128 characters.

**Description**

Maximum 1000 characters. Use alphanumeric and '+', '@', '\_' characters.

**Summary**

Service	Access level	Resource	Request condition
Allow (4 of 369 services) Show remaining 365			
EC2	Limited: List	All resources	None
License Manager	Limited: List	All resources	None
S3	Limited: Write	BucketName   string like   aws-supernova-marketplace-us-west-1-prod, ObjectPath   string like   All	None
STS	Limited: Read	All resources	None

**Tags**

Key	Value
No tags associated with the resource.	

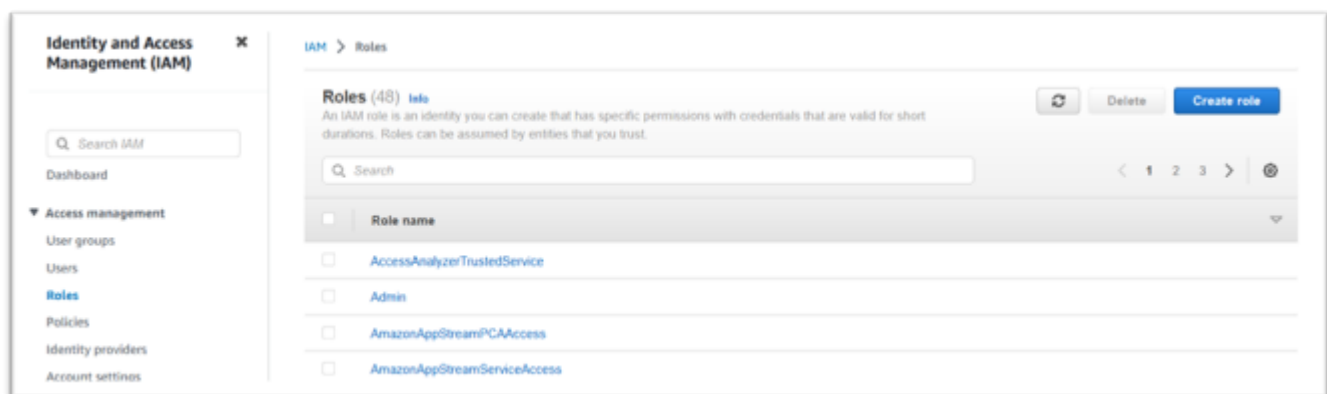
\* Required

Cancel Previous **Create policy**

## 8. Elija Crear política.

## Cree el rol de IAM.

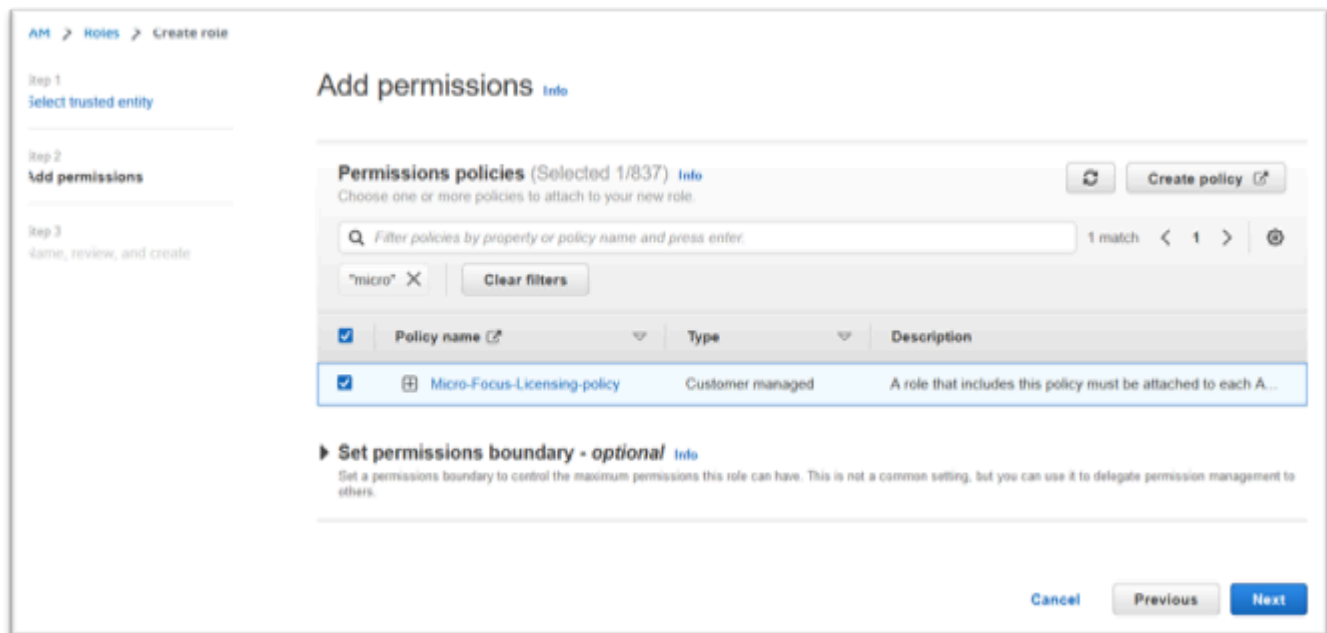
1. En la AWS Management Console, vaya a IAM.
2. Elija Roles y, a continuación, Crear rol.



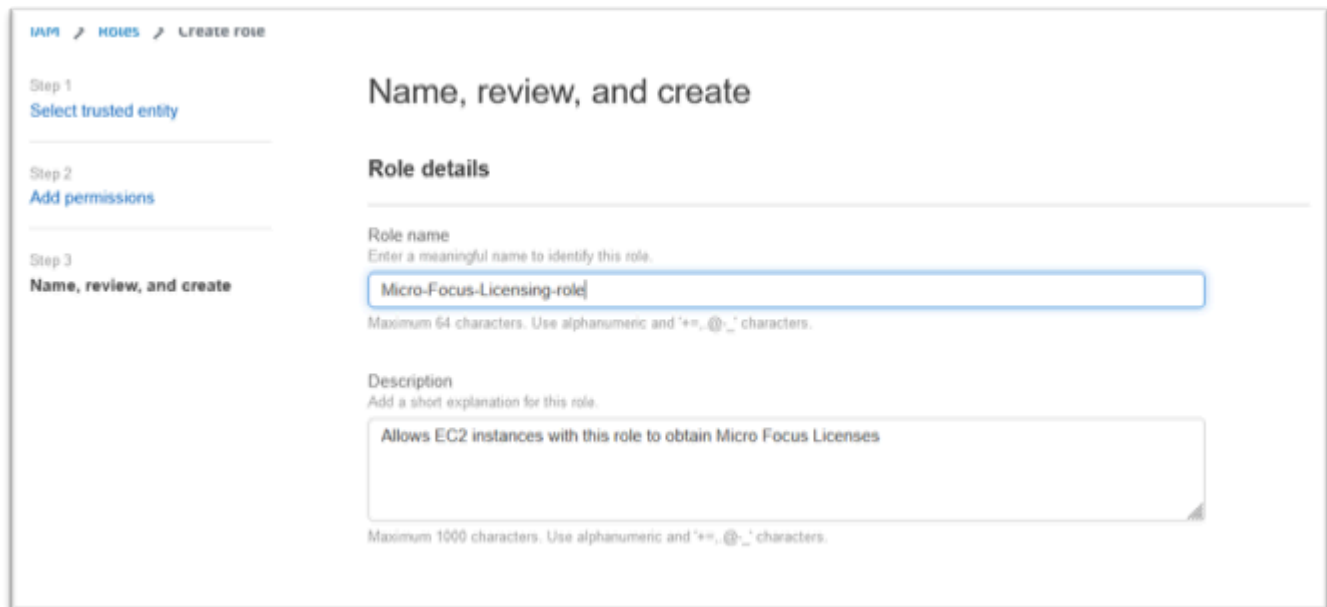
- Deje el Tipo de entidad de confianza como Servicio de AWS y elija el caso de uso común de EC2.

The screenshot shows the 'Create role' wizard in the AWS IAM console. The breadcrumb is 'IAM > Roles > Create role'. The current step is 'Step 1: Select trusted entity'. The 'Trusted entity type' section contains five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. The 'Use case' section is titled 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' and includes 'Common use cases' with 'EC2' (selected) and 'Lambda'. Below this is a dropdown menu for 'Use cases for other AWS services' with the placeholder text 'Choose a service to view use case'. At the bottom right are 'Cancel' and 'Next' buttons.

- Elija Siguiente.
- Introduzca "Micro" en el filtro y pulse Intro para aplicar el filtro.
- Elija la política que se acaba de crear, por ejemplo, la "Micro-Focus-Licensing-policy".
- Elija Siguiente.



8. Introduzca el nombre del rol, por ejemplo, "Micro-focus-Licensing-Role".
9. Sustituya la descripción por una propia, por ejemplo, "Permite que las instancias de Amazon EC2 con este rol obtengan licencias de Micro Focus".



10. En el Paso 1: Seleccionar entidades de confianza, revise el JSON y confirme que tiene los siguientes valores:

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

```

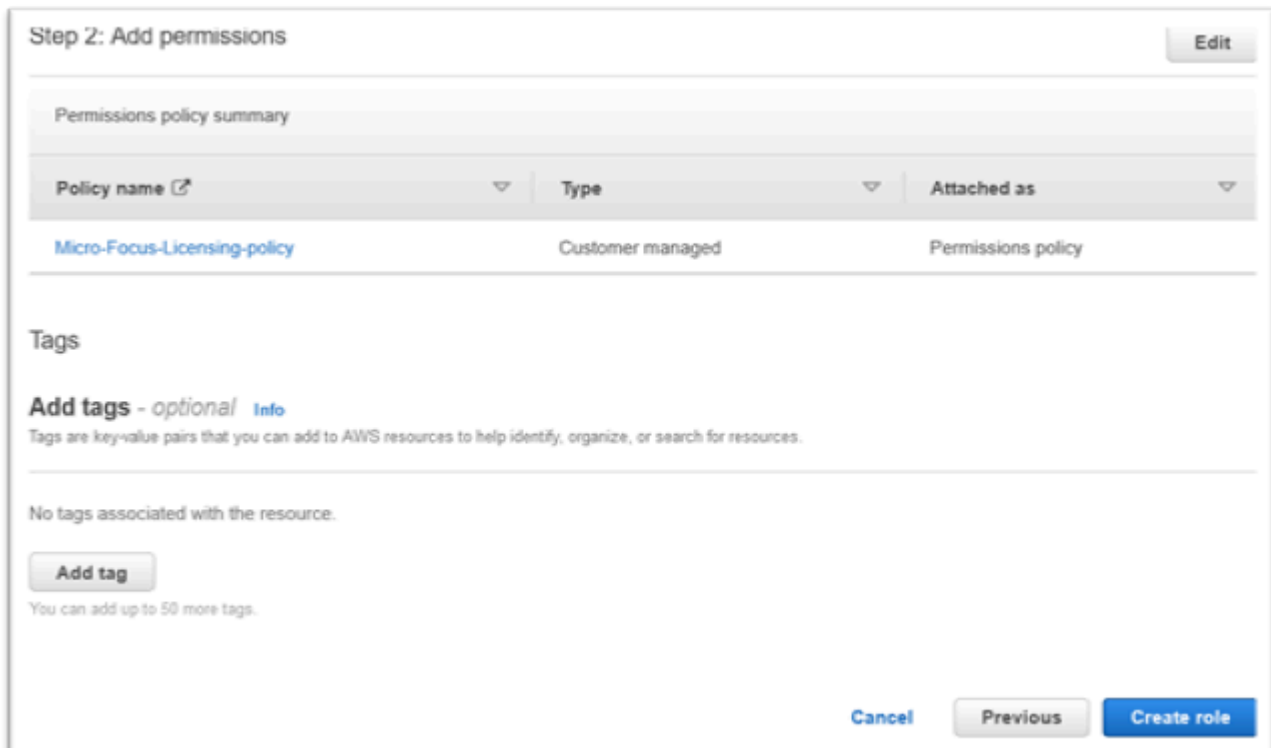
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "sts:AssumeRole"
    ],
    "Principal": {
      "Service": [
        "ec2.amazonaws.com"
      ]
    }
  }
]
}

```

### Note


El orden del efecto, la acción y el principio no son significativos.

11. Confirme que el Paso 2: Agregar permisos muestre su política de licencias.



Step 2: Add permissions Edit

Permissions policy summary

Policy name 	Type	Attached as
<a href="#">Micro-Focus-Licensing-policy</a>	Customer managed	Permissions policy

Tags

**Add tags - optional** [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Previous Create role

12. Elija Create role.

Una vez que se haya completado la solicitud de lista de permitidos, continúe con los siguientes pasos.

## Concesión de permisos necesarios con License Manager

1. Navegue hasta AWS License Manager . AWS Management Console

The screenshot shows the AWS License Manager console page. The main heading is "AWS License Manager" with the sub-heading "Manage, discover, and report software license usage". Below this, there is a "Get started" section with a "Start using AWS License Manager" button. A "Pricing" section indicates that there is no additional charge for AWS License Manager and lists links for Amazon pricing, Amazon EC2 pricing, Amazon EBS pricing, Amazon Systems Manager pricing, and Amazon SNS pricing. A "How it works" section contains a diagram with five steps: 1. Define rules for your licensed software, 2. Attach licensing rules during launch and proactively control usage, 3. Search inventory and track licenses brought in post-launch, 4. Use alerts to control and centrally manage licenses across all AWS accounts and on-premises, and 5. AWS License Manager.

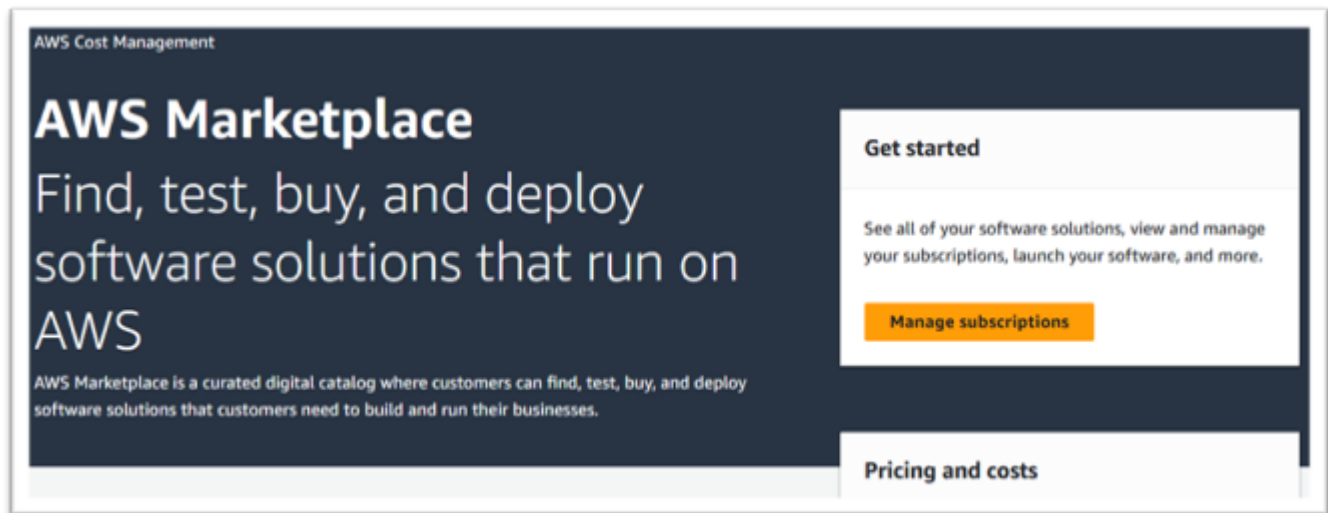
2. Elija Comenzar a usar AWS License Manager.
3. Si ve la siguiente ventana emergente, consulte los detalles, seleccione la casilla de verificación y pulse Conceder permisos.

The screenshot shows the "IAM permissions (one-time setup)" dialog box. The title is "IAM permissions (one-time setup)" with a close button (X). The text inside says "AWS License Manager requires permissions to manage licenses used by resources." Below this, there is a checkbox labeled "I grant AWS License Manager the required permissions" which is currently unchecked. A "View details" link is provided below the checkbox. At the bottom, there are two buttons: "Cancel" and "Grant permissions".

## Suscríbase a Amazon Machine Images

Tras suscribirse a un AWS Marketplace producto, puede lanzar una instancia desde la AMI del producto.

1. Navegue hasta AWS Marketplace Suscripciones en. AWS Management Console
2. Elija Administrar suscripciones.



3. Copie y pegue uno de los siguientes enlaces en la barra de direcciones del navegador.

### Note

Elija únicamente un enlace para uno de los productos para los que esté autorizado a utilizar.

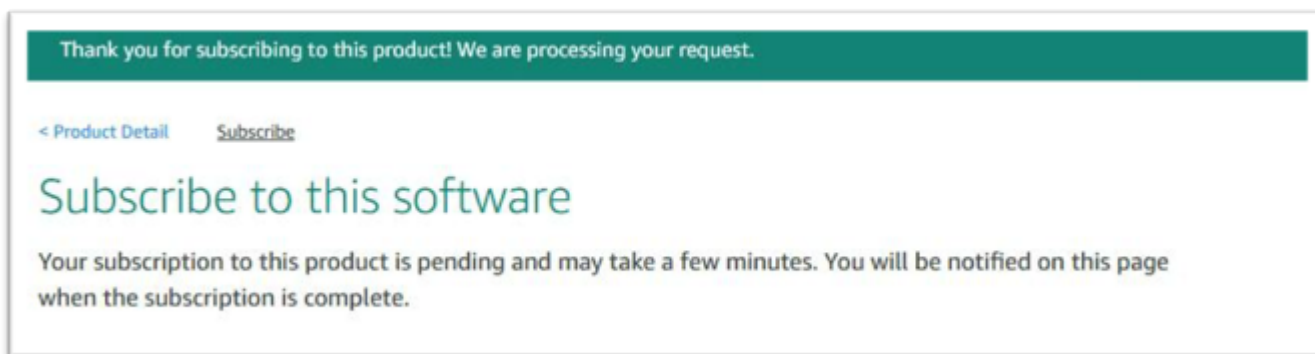
- Enterprise Server: <https://aws.amazon.com/marketplace/pp/prodview-g5emev63l7blc>
- Enterprise Server para Windows: <https://aws.amazon.com/marketplace/pp/prodview-lwybsiykbhc2>
- Enterprise Developer: <https://aws.amazon.com/marketplace/pp/prodview-77qmpr42yzxwk>
- Enterprise Developer con Visual Studio 2022: <https://aws.amazon.com/marketplace/pp/prodview-m4l3lqiszo6cm>
- Enterprise Analyzer: <https://aws.amazon.com/marketplace/pp/prodview-tttheylcmcihm>
- Enterprise Build Tools for Windows: <https://aws.amazon.com/marketplace/pp/prodview-2rw35bbt6uozi>

- Enterprise Stored Procedures: <https://aws.amazon.com/marketplace/pp/prodview-zoeyqnsdsj6ha>
- Enterprise Stored Procedures con SQL Server 2019: <https://aws.amazon.com/marketplace/pp/prodview-ynfklquwubnz4>

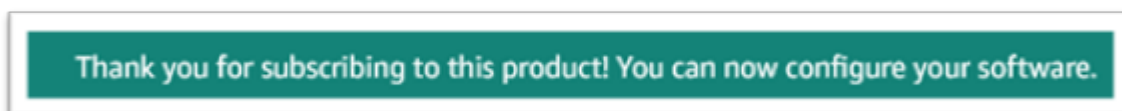
4. Elija Continuar con la suscripción.

5. Si los términos y condiciones son aceptables, seleccione Aceptar términos.

6. El proceso de suscripción puede tardar unos minutos.



7. Cuando aparezca el mensaje de agradecimiento, copie y pegue el siguiente enlace del paso 3 para seguir añadiendo suscripciones.

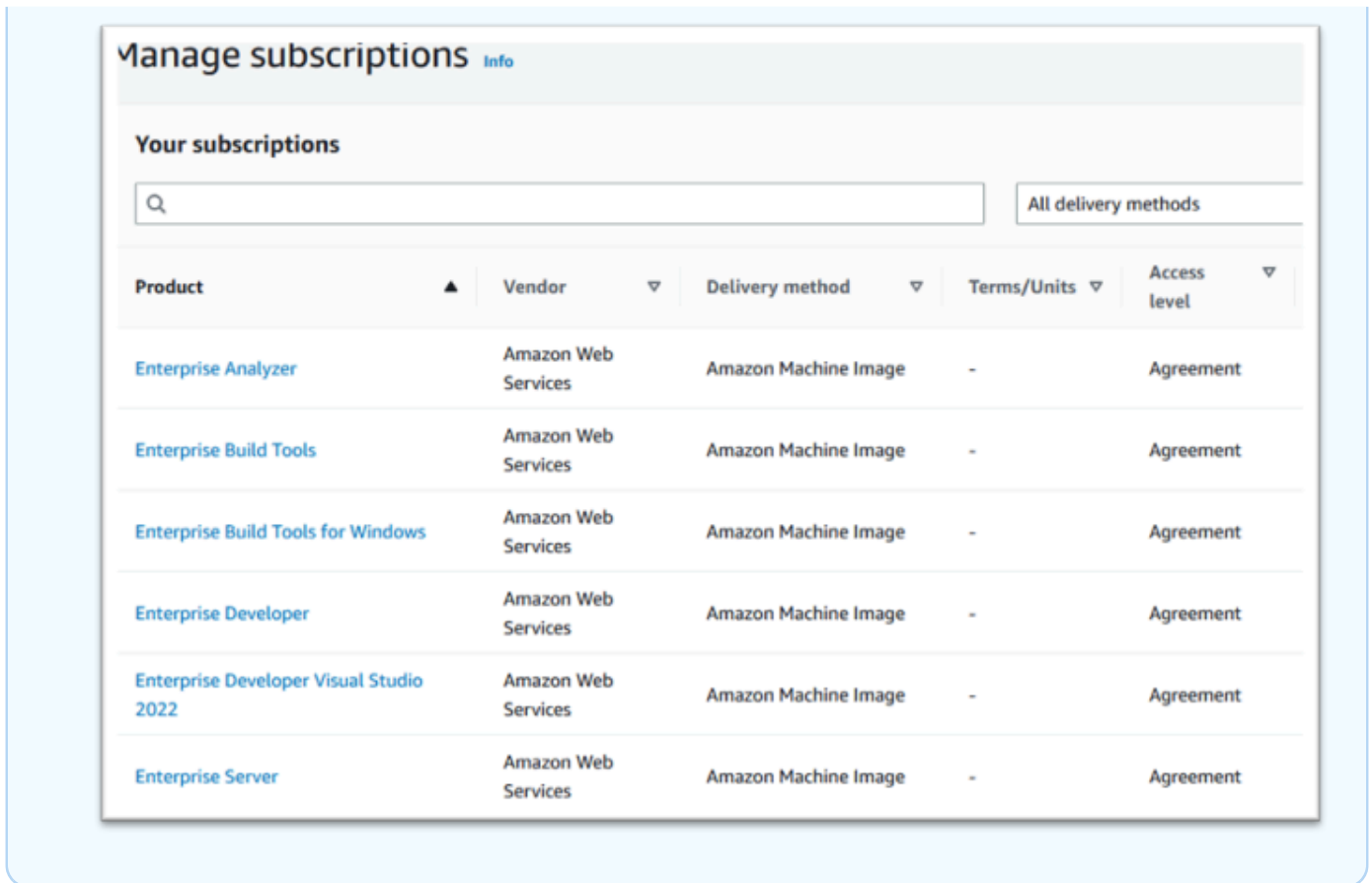


8. Pare cuando la opción Administrar suscripciones muestre todas las AMI suscritas.

**Note**

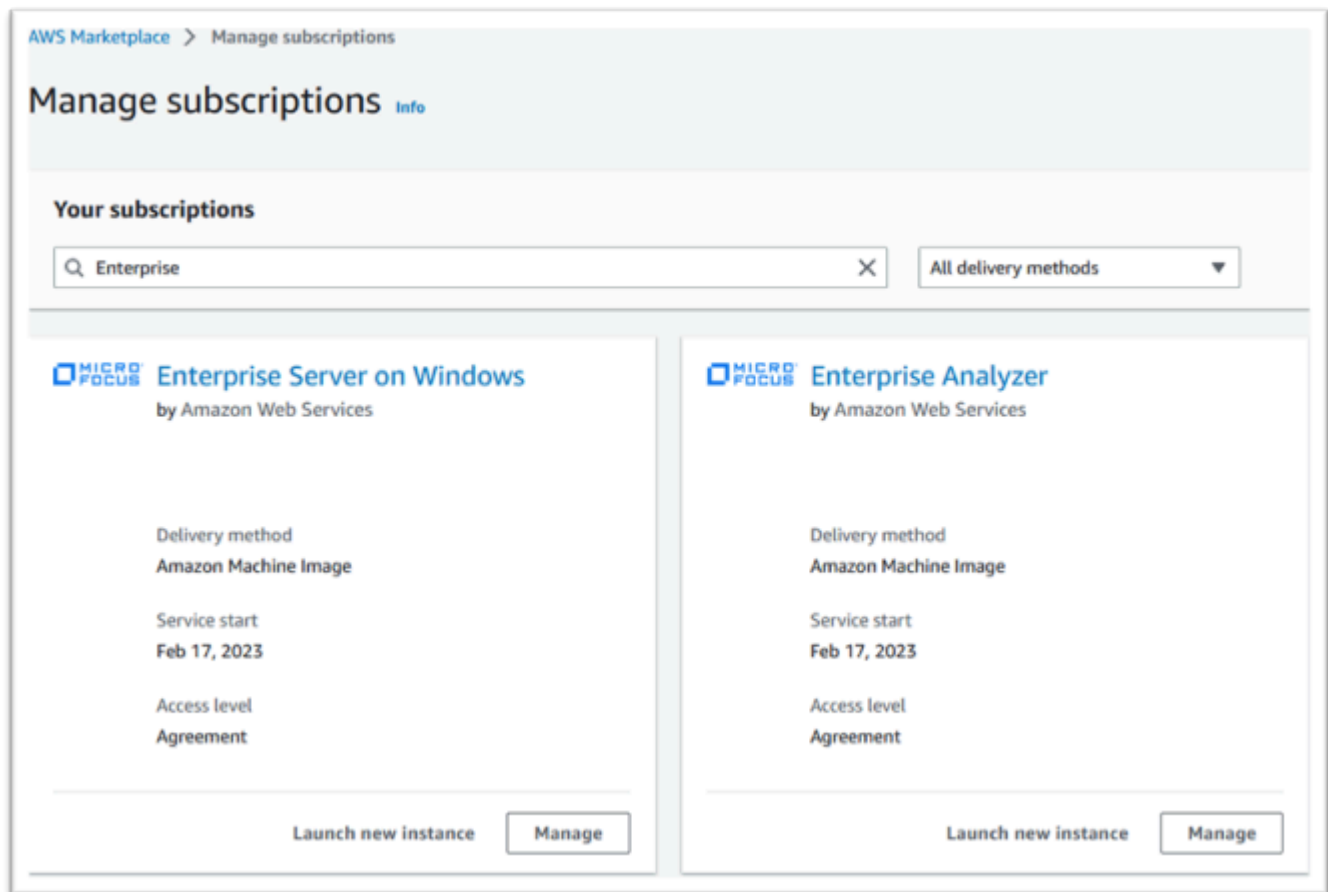
Las preferencias del panel (icono con forma de engranaje) están configuradas para mostrar la vista en forma de tabla.





## Lance una instancia de Micro Focus para la modernización de AWS mainframe

1. Navegue hasta AWS Marketplace Suscripciones en. AWS Management Console
2. Localice la AMI que se va a lanzar y elija Lanzamiento de una instancia nueva.



3. En el cuadro de diálogo de lanzamiento de una nueva instancia, asegúrese de seleccionar la región de la lista de permitidos.
4. Pulse Continue to launch through EC2.

**Note**

El siguiente ejemplo muestra el lanzamiento de una AMI para desarrolladores empresariales, pero el proceso es el mismo para todas las AMI de modernización de AWS mainframe.

AWS Marketplace > Manage subscriptions > Enterprise Developer > Launch new instance

## Launch new instance

### Configure this software

Choose a fulfillment option below to select how you wish to deploy the software, then enter the information required to configure the deployment.

**Delivery method**  
64-bit (x86) Amazon Machine Image ▼

**Software version**  
v8.0.1 (Oct 26, 2022) ▼

For older software versions, please visit the [full AWS Marketplace website](#) .

**Region**  
us-west-1 ▼

**AMI ID:** ami-0f199167bc5fce009

Cancel **Continue to launch through EC2**

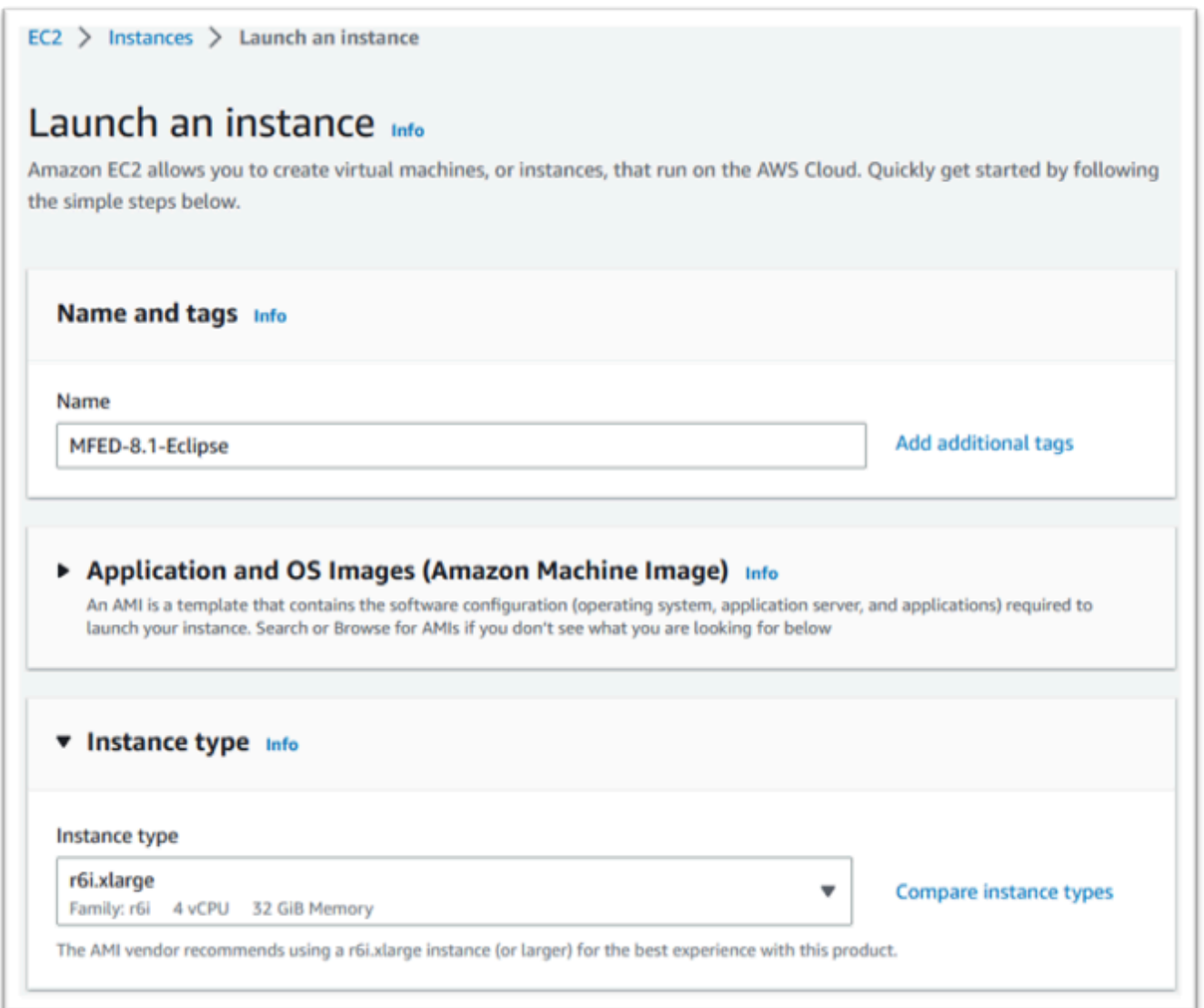
5. Escriba un nombre del servidor.
6. Elija un tipo de instancia.

El tipo de instancia seleccionado debe determinarse en función de los requisitos de rendimiento y coste del proyecto. Se sugieren los siguientes puntos de partida:

- Para Enterprise Analyzer, un r6i.xlarge
- Para Enterprise Developer, un r6i.large
- Para una instancia independiente de Enterprise Server, un r6i.xlarge
- Para el clúster de disponibilidad de rendimiento (PAC) de Micro Focus con capacidad de ampliación, un r6i.large

**Note**

La sección Application and OS Images se ha reducido para la captura de pantalla.



EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags Info

Name

 [Add additional tags](#)

### ▶ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

### ▼ Instance type Info

Instance type

 [Compare instance types](#)  
Family: r6i 4 vCPU 32 GiB Memory

The AMI vendor recommends using a r6i.xlarge instance (or larger) for the best experience with this product.


7. Elija o cree (y guarde) un par de claves (no se muestran).

Para obtener más información sobre pares de claves para instancias de Linux, consulte [Pares de claves de Amazon EC2 e instancias Linux](#).

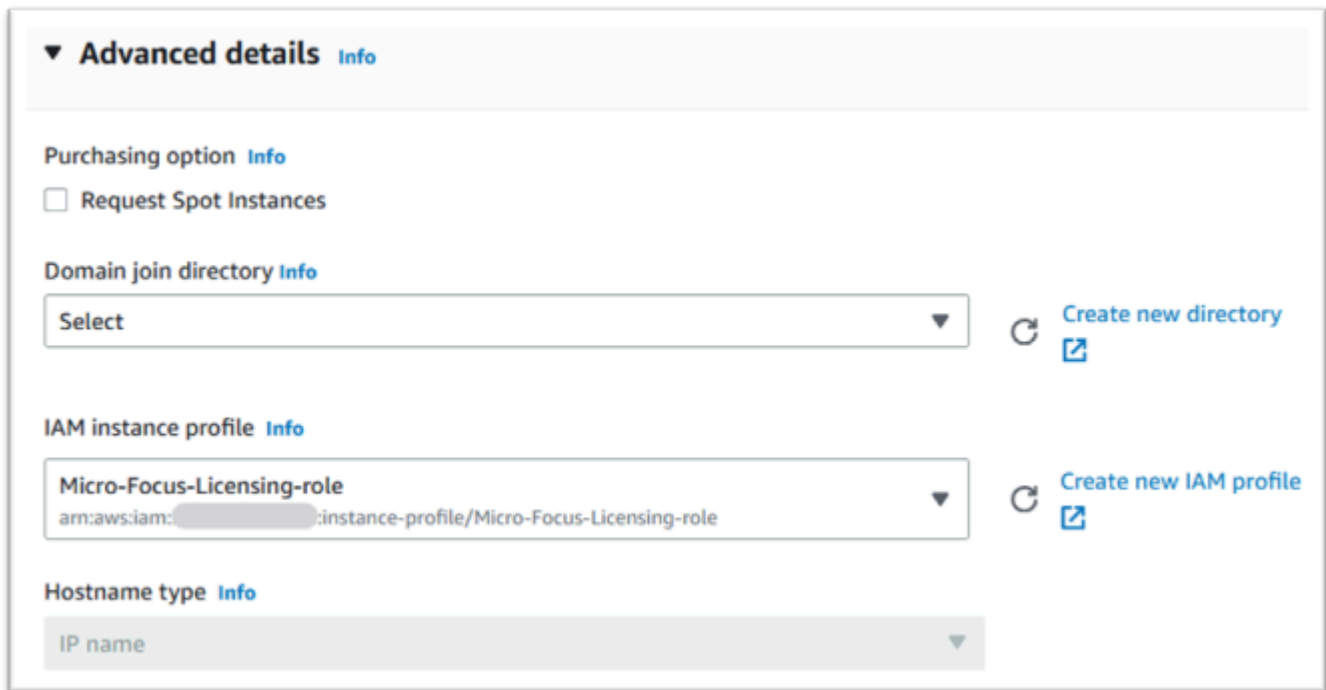
Para obtener más información acerca de los pares de claves para las instancias de Windows, consulte [Pares de claves de Amazon EC2 e instancias de Windows](#).

8. Edite la configuración de red y elija la VPC de la lista de permitidos y la subred adecuada.
9. Elija o cree un nuevo grupo de seguridad. Si se trata de una instancia EC2 de Enterprise Server, lo habitual es permitir que el tráfico TCP llegue a los puertos 86 y 10086 para administrar la configuración de Micro Focus.

10. Si lo desea, configure el almacenamiento de la instancia de Amazon EC2.
11. Importante: amplíe los detalles avanzados y, en el perfil de instancia de IAM, elija la función de licencia creada anteriormente, por ejemplo, "Micro-focus-Licensing-role".

 Note

Si omite este paso, una vez creada la instancia, puede modificar la función de IAM desde la opción Seguridad del menú de acciones de la instancia de EC2.



**Advanced details** [Info](#)

**Purchasing option** [Info](#)

Request Spot Instances

**Domain join directory** [Info](#)

Select [Create new directory](#)

**IAM instance profile** [Info](#)

Micro-Focus-Licensing-role  
arn:aws:iam:[:instance-profile/Micro-Focus-Licensing-role] [Create new IAM profile](#)

**Hostname type** [Info](#)

IP name

12. Revise el resumen y pulse Lanzar instancia.

**▼ Summary**

Number of instances [Info](#)

1

Software Image (AMI)

Distribution Configuration for...[read more](#)  
ami-0f199167bc5fce009

Virtual server type (instance type)

r6i.xlarge

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 100 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance

13. El lanzamiento de la instancia fallará si se elige un tipo de servidor virtual no válido.

Si esto ocurre, seleccione Editar configuración de instancias y cambie el tipo de instancia.

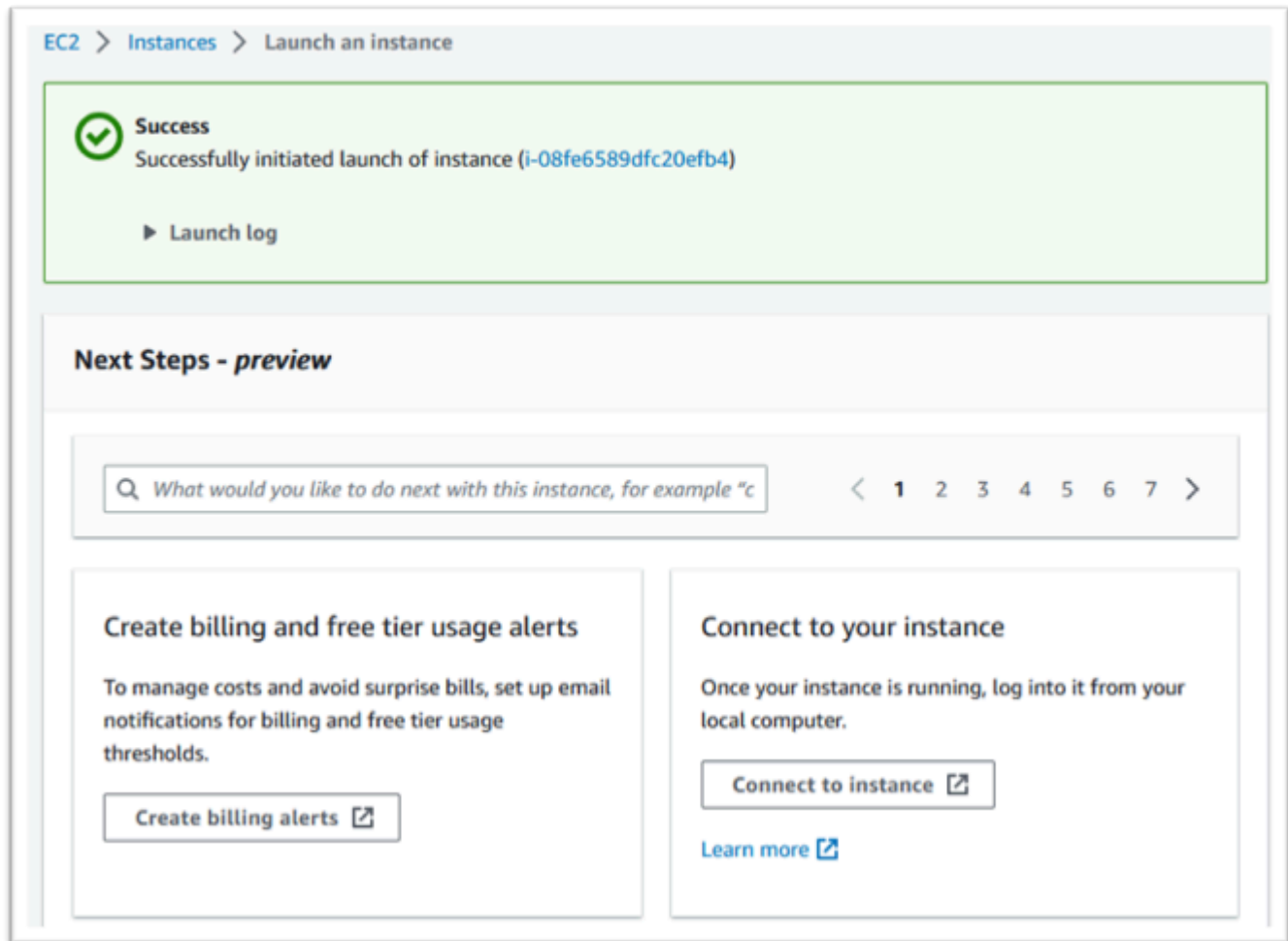
**Launching instance**

Please wait while we launch your instance.  
Do not close your browser while this is loading.

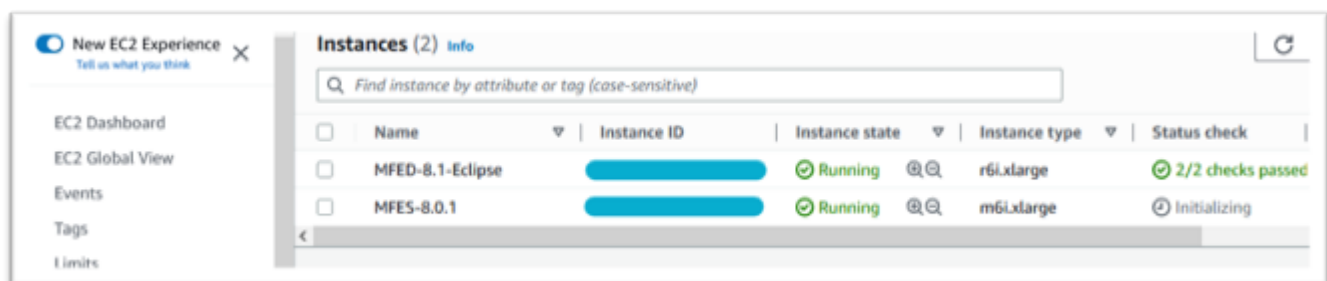
Subscribing to Marketplace AMI 73%

► Details

14. Cuando aparezca el mensaje “Correcto”, seleccione Conectarse a la instancia para obtener los detalles de la conexión.



15. Como alternativa, navegue hasta EC2 en la AWS Management Console.  
 16. Elija Instancias para ver el estado de la nueva instancia.



## Subred o VPC sin acceso a Internet

Realice estos cambios adicionales si la subred o la VPC no tienen acceso a Internet de salida.

El administrador de licencias requiere acceso a los siguientes servicios de AWS:

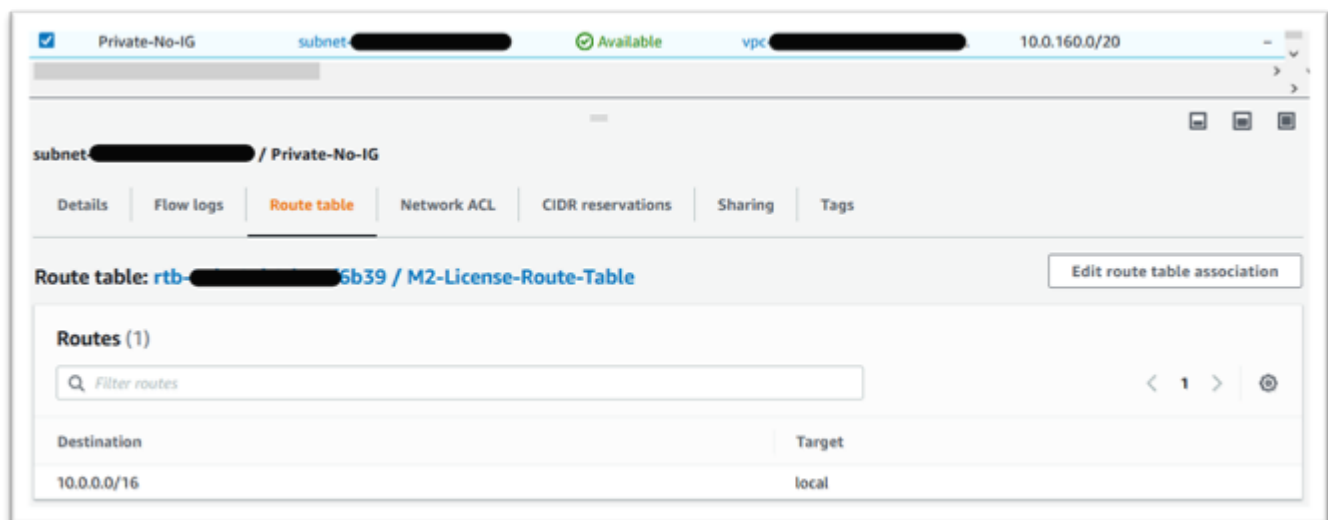
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ec2`
- `com.amazonaws.region.license-manager`
- `com.amazonaws.region.sts`

Los pasos anteriores definieron el el servicio `com.amazonaws.region.s3` como punto de conexión de puerta de enlace. Este punto de conexión necesita una entrada en la tabla de enrutamiento para cualquier subred sin acceso a Internet.

Los tres servicios adicionales se definirán como puntos de conexión de interfaz.

### Añadir la entrada de la tabla de enrutamiento para el punto de conexión Amazon S3

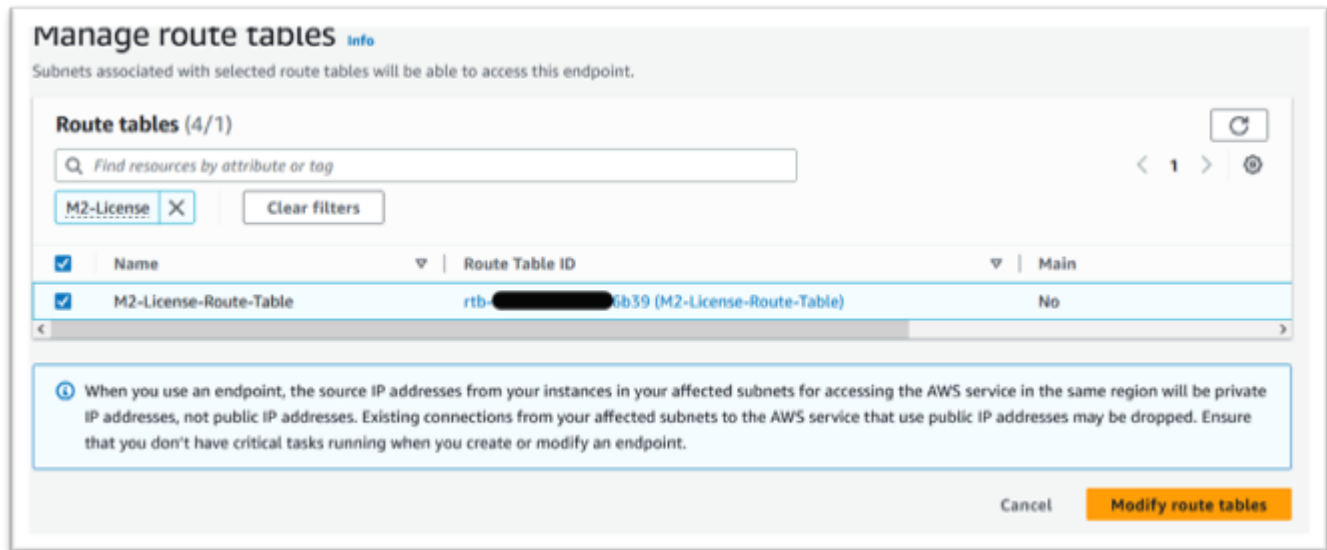
1. Navegue hasta VPC en AWS Management Console y elija Subredes.
2. Elija la subred en la que se vayan a crear las instancias de Amazon EC2 y elija la pestaña Tabla de enrutamiento.
3. Anote algunos dígitos finales del identificador de la tabla de enrutamiento. Por ejemplo, 6b39 de la imagen de abajo.



4. En el panel de navegación, elija Puntos de conexión.
5. Elija el punto de conexión creado anteriormente y, a continuación, Administrar tablas de enrutamiento en la pestaña Tablas de enrutamiento del punto de conexión o en el menú desplegable Acciones.



6. Elija la tabla de enrutamiento con los dígitos identificados anteriormente y pulse Modificar tablas de enrutamiento.



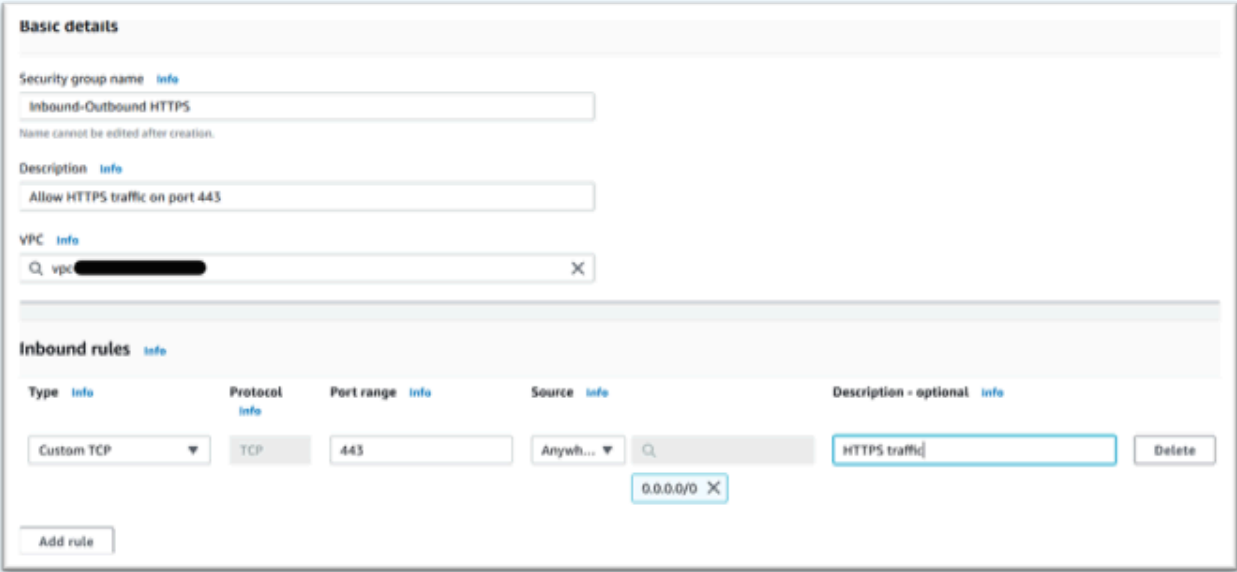
## Definición del grupo de seguridad necesario

Los servicios Amazon EC2 y License Manager se comunican a través de HTTPS a través del puerto 443. AWS STS Esta comunicación es bidireccional y requiere reglas de entrada y salida para permitir que la instancia se comunique con los servicios.

1. Vaya a Amazon VPC en la AWS Management Console.
2. En el panel de navegación, en Grupos de seguridad, elija Crear grupo de seguridad.
3. Introduzca el nombre y la descripción de un grupo de seguridad, por ejemplo, "Inbound-Outbound HTTPS".
4. Pulse la X en el área de selección de VPC para eliminar la VPC predeterminada y elija la VPC que contenga el punto de conexión de S3.
5. Agregue una regla de entrada que permita el tráfico TCP en el puerto 443 desde cualquier lugar.

### Note

Las reglas de entrada (y de salida) se pueden restringir aún más limitando el origen. Para obtener más información, consulte [Controle el tráfico a sus AWS recursos mediante grupos de seguridad](#) en la Guía del usuario de Amazon VPC.



The screenshot displays the AWS Security Groups console interface. Under the "Basic details" section, the "Security group name" is "Inbound-Outbound HTTPS" and the "Description" is "Allow HTTPS traffic on port 443". The "VPC" is selected as "vpc-...". The "Inbound rules" section shows a table with columns for Type, Protocol, Port range, Source, and Description - optional. A rule is configured with Type "Custom TCP", Protocol "TCP", Port range "443", Source "Anywh...", and Description "HTTPS traffic". A search box below the source field shows "0.0.0.0/0". An "Add rule" button is visible at the bottom left of the rules section.

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	443	Anywh...	HTTPS traffic

6. Pulse Crear grupo de seguridad.

## Creación de puntos de conexión del servicio

Repita este proceso tres veces, una para cada servicio.

1. Diríjase a Amazon VPC en AWS Management Console y seleccione Endpoints.
2. Pulse Crear punto de conexión.
3. Introduzca un nombre, por ejemplo, "Micro-Focus-License-EC2", "Micro-focus-License-STS" o "Micro-Focus-License-Manager".
4. Elija Servicios de AWS en Categoría de servicio.

**Endpoint settings**

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

Micro-Focus-License-EC2

**Service category**  
Select the service category

**AWS services**  
Services provided by Amazon

**PrivateLink Ready partner services**  
Services with an AWS Service Ready designation

**AWS Marketplace services**  
Services that you've purchased through AWS Marketplace

**Other endpoint services**  
Find services shared with you by service name

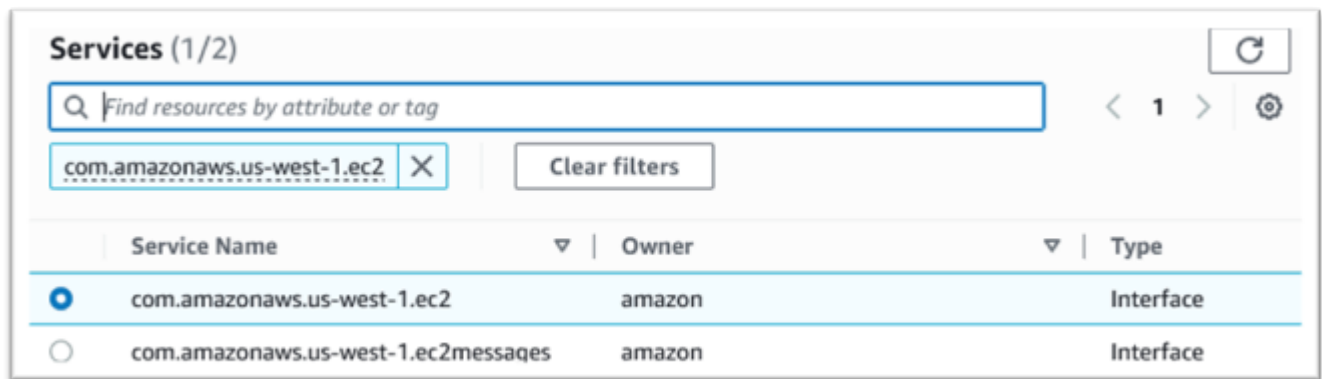
5. En Servicios, busque el servicio de interfaz correspondiente, que sea uno de los siguientes:
- “com.amazonaws.*región*.ec2”
  - “com.amazonaws.*región*.sts”
  - “com.amazonaws.*región*.license-manager”

Por ejemplo:

- “com.amazonaws.us-west-1.ec2”
- “com.amazonaws.us-west-1.sts”
- “com.amazonaws.us-west-1.license-manager”

6. Elija el servicio de interfaz correspondiente.

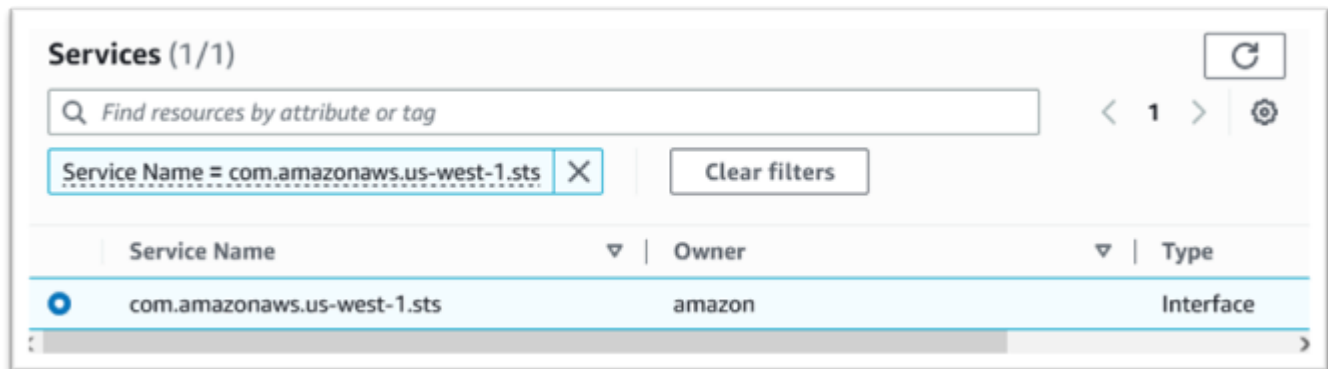
com.amazonaws.*región*.ec2:



The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'com.amazonaws.us-west-1.ec2'. The table below shows two services:

Service Name	Owner	Type
com.amazonaws.us-west-1.ec2	amazon	Interface
com.amazonaws.us-west-1.ec2messages	amazon	Interface

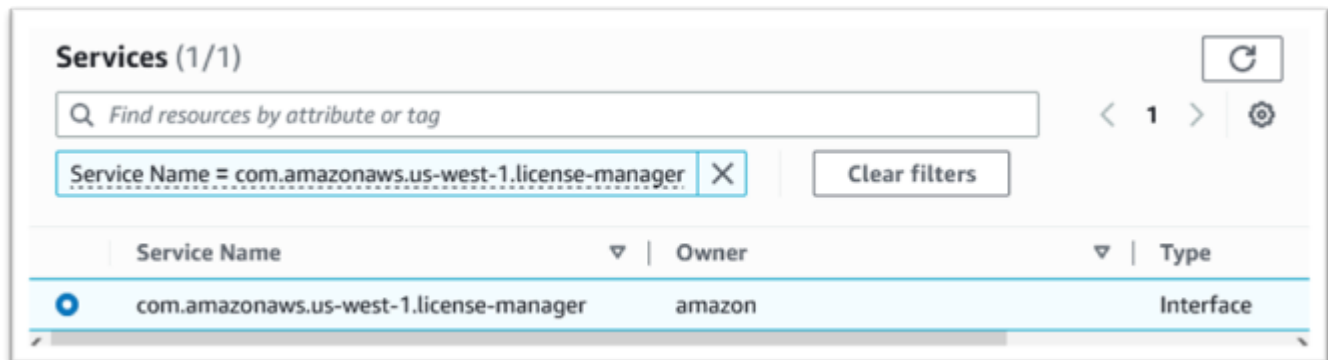
com.amazonaws.**región**.sts:



The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'Service Name = com.amazonaws.us-west-1.sts'. The table below shows one service:

Service Name	Owner	Type
com.amazonaws.us-west-1.sts	amazon	Interface

com.amazonaws.**región**.license-manager:



The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'Service Name = com.amazonaws.us-west-1.license-manager'. The table below shows one service:

Service Name	Owner	Type
com.amazonaws.us-west-1.license-manager	amazon	Interface

7. En VPC, elija la VPC para su instancia.

**VPC**  
Select the VPC in which to create the endpoint

**VPC**  
The VPC in which to create your endpoint.

vpc-[redacted] (Modernization-vpc1) [Refresh]

▶ Additional settings

8. Elija la Zona de disponibilidad y las Subredes de la VPC.

**Subnets (1/2)** Info

Availability Zone	Subnet ID
<input checked="" type="checkbox"/> us-west-1b (usw1-az3)	subnet-[redacted]
<input type="checkbox"/> us-west-1c (usw1-az1)	

subnet-[redacted] X  
Private-No-IG

IP address type

IPv4  
 IPv6  
 Dualstack

9. Elija el Grupo de seguridad creado anteriormente.

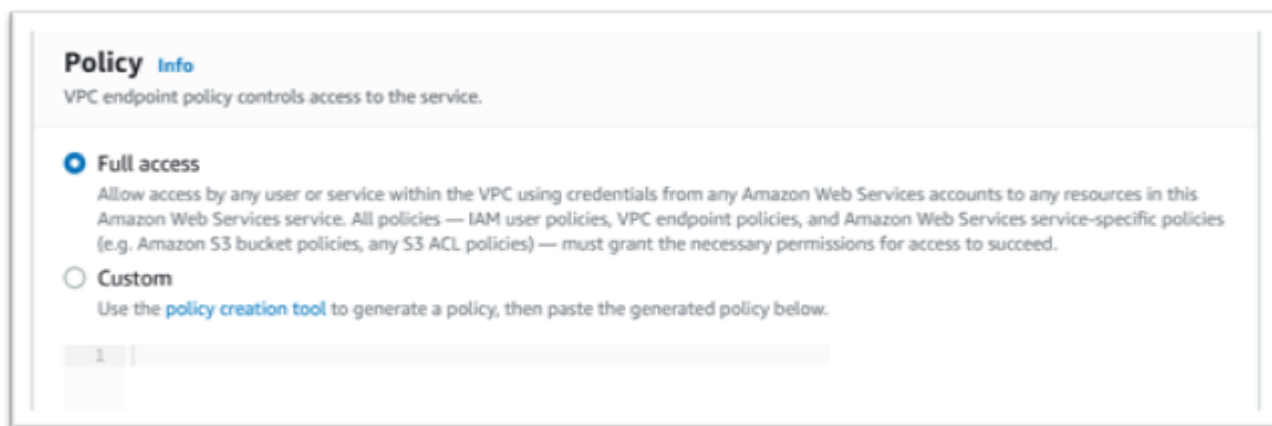
**Security groups (1)** Info

Find resources by attribute or tag

Group name = Inbound-Outbound HTTPS X Clear filters

Group ID	Group name
<input type="checkbox"/> sg-[redacted]	Inbound-Outbound HTTPS

10. En Política, seleccione Full Access.



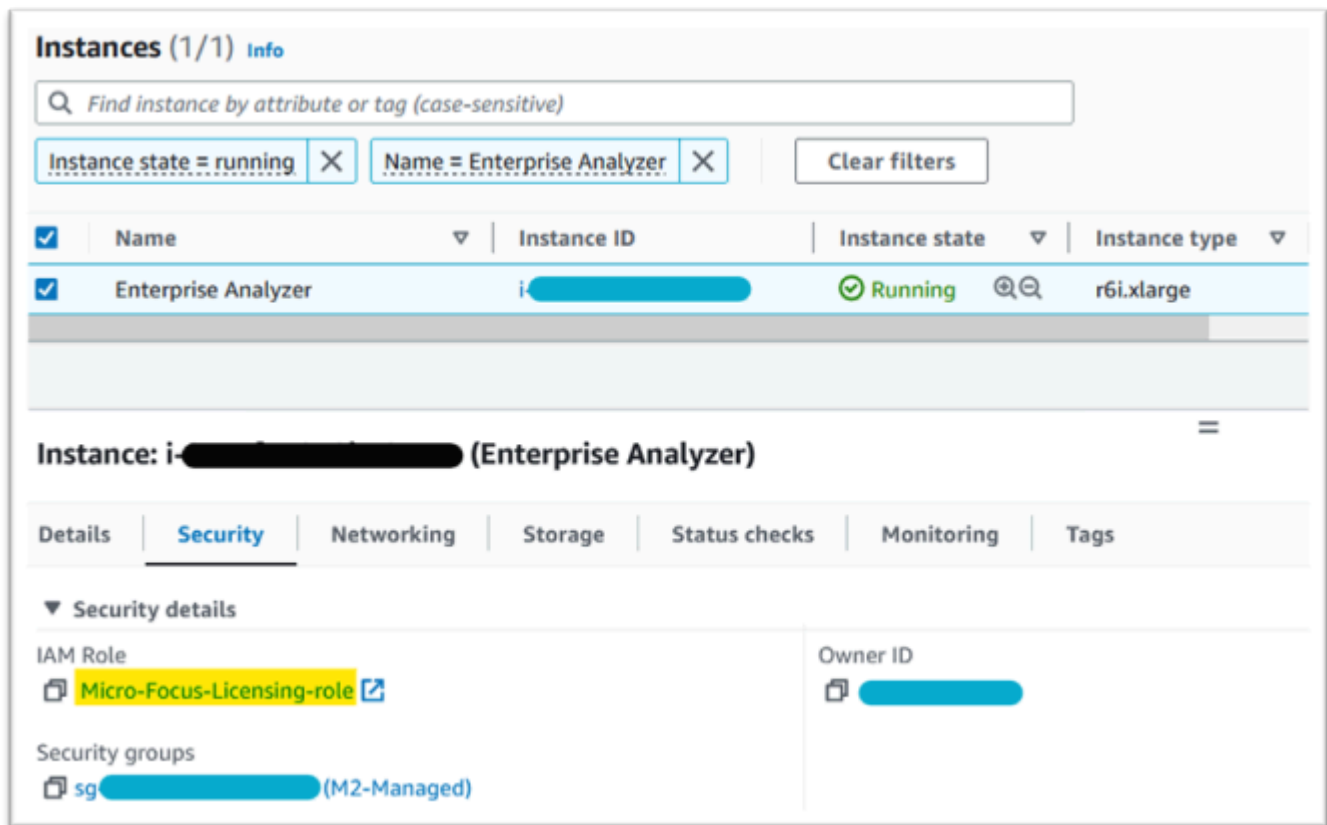
11. Elija Crear punto de conexión.
12. Repita este proceso para el resto de las interfaces.

## Solución de problemas de licencia

Si tiene problemas para acceder a las AMI o utilizarlas, la siguiente información puede resultarle útil.

Compruebe que la instancia de Amazon EC2 tenga el rol de licencia de IAM

Esto se puede comprobar en la pestaña Seguridad de los detalles de la instancia de Amazon EC2. Se puede cambiar mediante la opción de seguridad del menú desplegable Acciones.



## Uso del Analizador de accesibilidad

Busque el Reachability Analyzer en la página de la consola. AWS Network Manager

Cree y analice una ruta entre la instancia de Amazon EC2 creada a partir de la AMI y el punto de conexión de VPC de Amazon S3.

Si la instancia de Amazon EC2 no tiene acceso a Internet, repita el análisis de la ruta de los 4 puntos de conexión.

Para obtener más información sobre el Analizador de accesibilidad, consulte [Getting started with Reachability Analyzer](#) en la guía del Analizador de accesibilidad.

## Ejecute el daemon de licencias

En Windows Enterprise Developer, utilice el siguiente comando desde una línea de comandos:

```
"C:\Program Files (x86)\Micro Focus\Enterprise Developer\AdoptOpenJDK\bin\java" -jar
"C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

y examine la salida. Ignore los mensajes del SLF4J y busque la primera excepción.

En Enterprise Analyzer, utilice el siguiente comando desde una línea de comandos:

```
"C:\Program Files (x86)\Micro Focus\AdoptOpenJDK\bin\java" -jar "C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

y examine la salida. Ignore los mensajes del SLF4J y busque la primera excepción.

En Linux, ejecute .

```
java -jar /var/microfocuslicensing/bin/aws-license-daemon.jar
```

Ignore los mensajes del SLF4J y busque la primera excepción.

Por ejemplo, si el recurso de Amazon S3 no está disponible, la excepción es la siguiente:

```
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
```

El mensaje de excepción indica qué recurso no está disponible. Compare los valores de configuración con los que se muestran en este tema.

## Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer

AWS La modernización del mainframe proporciona varias herramientas a través de Amazon AppStream 2.0. AppStream 2.0 es un servicio de streaming de aplicaciones seguro y totalmente gestionado que permite transmitir aplicaciones de escritorio a los usuarios sin tener que volver a escribirlas. AppStream La versión 2.0 proporciona a los usuarios acceso instantáneo a las aplicaciones que necesitan, con una experiencia de usuario fluida y con gran capacidad de respuesta en el dispositivo que elijan. El uso de la AppStream versión 2.0 para alojar herramientas específicas del motor de ejecución ofrece a los equipos de aplicaciones de los clientes la posibilidad de utilizar las herramientas directamente desde sus navegadores web e interactuar con los archivos de las aplicaciones almacenados en depósitos o repositorios de Amazon S3. CodeCommit



Para obtener información sobre la compatibilidad de los navegadores en la AppStream versión 2.0, consulte [System Requirements and Feature Support \(navegador web\)](#) en la Guía de administración de Amazon AppStream 2.0. Si tiene problemas al utilizar la AppStream versión 2.0, consulte [Solución de problemas de usuario de la AppStream versión 2.0](#) en la Guía de administración de Amazon AppStream 2.0.

Este documento está dirigido a los miembros del equipo de operaciones del cliente. Describe cómo configurar las flotas y pilas de Amazon AppStream 2.0 para alojar las herramientas Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer que se utilizan con AWS la modernización del mainframe. Micro Focus Enterprise Analyzer se suele utilizar durante la fase de evaluación y Micro Focus Enterprise Developer se suele utilizar durante la fase de migración y modernización del enfoque de modernización del mainframe. AWS Si piensa utilizar Enterprise Analyzer y Enterprise Developer, debe crear flotas y pilas independientes para cada herramienta. Cada herramienta requiere su propia flota y su propia pila, ya que las condiciones de licencia son diferentes.

#### Important

Los pasos de este tutorial se basan en la AWS CloudFormation plantilla descargable [cfn-m2-appstream-fleet-ea-ed.yml](#).

## Temas

- [Requisitos previos](#)
- [Paso 1: Obtenga las imágenes AppStream 2.0](#)
- [Paso 2: Cree la pila con la plantilla AWS CloudFormation](#)
- [Paso 3: Crea un usuario en la AppStream versión 2.0](#)
- [Paso 4: Inicie sesión en AppStream 2.0](#)
- [Paso 5: Verificar los buckets de Amazon S3 \(opcional\)](#)
- [Sigüientes pasos](#)
- [Eliminar recursos](#)

## Requisitos previos

- Descargue la plantilla: [cfn-m2-appstream-fleet-ea-ed.yml](#).

- Obtenga el ID de la VPC y el grupo de seguridad predeterminados. Para obtener más información sobre la VPC predeterminada, consulte [VPC predeterminadas](#) en la Guía del usuario de Amazon VPC. Para obtener más información sobre el grupo de seguridad predeterminado, consulte [Grupos de seguridad predeterminados y personalizados](#) en la Guía del usuario de Amazon EC2.
- Asegúrese de que tiene permisos para:
  - cree pilas, flotas y usuarios en la versión 2.0. AppStream
  - cree pilas AWS CloudFormation utilizando una plantilla.
  - Crear buckets y cargar archivos a buckets de Amazon S3.
  - Descargar credenciales (access\_key\_id y secret\_access\_key) de IAM.

## Paso 1: Obtenga las imágenes AppStream 2.0

En este paso, comparte las imágenes AppStream 2.0 de Enterprise Analyzer y Enterprise Developer con su AWS cuenta.

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el panel de navegación izquierdo, seleccione Herramientas.
3. En Análisis, desarrollo y creación de activos, seleccione Compartir activos con mi AWS cuenta.

## Paso 2: Cree la pila con la plantilla AWS CloudFormation

En este paso, utilizará la AWS CloudFormation plantilla descargada para crear una pila AppStream 2.0 y una flota para ejecutar Micro Focus Enterprise Analyzer. Puede repetir este paso más adelante para crear otra pila y flota AppStream 2.0 para ejecutar Micro Focus Enterprise Developer, ya que cada herramienta requiere su propia flota y pila en la AppStream versión 2.0. Para obtener más información sobre las AWS CloudFormation pilas, consulte [Trabajar con pilas](#) en la Guía del AWS CloudFormation usuario.

### Note

AWS La modernización de mainframe añade una tarifa adicional al precio estándar de la AppStream versión 2.0 por el uso de Enterprise Analyzer y Enterprise Developer. Para obtener más información, consulte [AWS Mainframe Modernization Pricing](#).

1. Descargue la plantilla [cfn-m2-appstream-fleet-ea-ed.yml](#), si es necesario.
2. Abra la AWS CloudFormation consola y elija Create Stack and with new resources (estándar).
3. En Requisito previo: preparar la plantilla, elija La plantilla está lista.
4. En Especificar plantilla, elija Cargar un archivo de plantilla.
5. En Cargar un archivo de plantilla, seleccione Elegir archivo y cargue la plantilla [cfn-m2-appstream-fleet-ea-ed.yml](#).
6. Elija Siguiente.

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The current step is 'Specify template'. The interface includes a sidebar with steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main content area is titled 'Create stack' and contains two sections: 'Prerequisite - Prepare template' and 'Specify template'. In the 'Prerequisite' section, the 'Template is ready' radio button is selected. In the 'Specify template' section, the 'Upload a template file' radio button is selected. Below this, the 'Upload a template file' section shows a 'Choose file' button with the filename 'cfn-m2-appstream-fleet-ea-ed.yml' entered. At the bottom, there is a 'View in Designer' button and 'Cancel' and 'Next' buttons.

7. En Especificar los detalles de la pila, introduzca la siguiente información:
  - En Nombre de pila, introduzca un nombre de su elección. Por ejemplo, **m2-ea**.
  - AppStreamEn Aplicación, elija ea.
  - En AppStreamFleetSecurityGrupo, elige el grupo de seguridad predeterminado de la VPC predeterminada.
  - En AppStreamFleetVpcSubred, elige una subred dentro de tu VPC predeterminada.

- En `AppStreamImageName`, elige la imagen que empieza por `m2-enterprise-analyzer`. Esta imagen contiene la versión actualmente compatible de la herramienta Micro Focus Enterprise Analyzer.
- Acepte los valores predeterminados para el resto de los campos y después elija **Siguiente**.

**Specify stack details**

**Stack name**

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

**Parameters**

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**AppStreamApplication**

AppStream application

**AppStreamFleetSecurityGroup**

AppStream fleet security group

**AppStreamFleetType**

AppStream fleet type

**AppStreamFleetVpcSubnet**

AppStream fleet subnet

**AppStreamImageName**

AppStream machine image name: m2-enterprise-analyzer-v7.0.1.R1 or m2-enterprise-developer-v7.0.3.R1

**AppStreamInstanceType**

AppStream instance type

**AppStreamInstances**

AppStream desired instances

**AppStreamView**

AppStream view

Cancel Previous **Next**

8. Acepte los valores predeterminados y, a continuación, elija **Siguiente**.
9. En **Revisar**, asegúrese de que todos los parámetros son los que desea.

10. Desplázate hasta el final, selecciona Acepto que AWS CloudFormation podría crear recursos de IAM con nombres personalizados y selecciona Create Stack.

La pila y la flota tardan entre 20 y 30 minutos en crearse. Puede elegir Actualizar para ver los AWS CloudFormation eventos a medida que se producen.

### Paso 3: Crea un usuario en la AppStream versión 2.0

Mientras esperas AWS CloudFormation a terminar de crear la pila, puedes crear uno o más usuarios en la AppStream versión 2.0. Estos usuarios son los que utilizarán Enterprise Analyzer en la AppStream versión 2.0. Deberá especificar una dirección de correo electrónico para cada usuario y asegurarse de que cada uno tenga los permisos suficientes para crear buckets en Amazon S3, cargar archivos en un bucket y vincular a un bucket para asignar su contenido.

1. Abra la consola AppStream 2.0.
2. En el panel de navegación izquierdo, elija Grupo de usuarios.
3. Seleccione la opción Crear usuario.
4. Proporcione una dirección de correo electrónico en la que el usuario pueda recibir una invitación por correo electrónico para usar la AppStream versión 2.0, un nombre y apellidos, y seleccione Crear usuario.
5. Repita este procedimiento si es necesario para crear más usuarios. La dirección de correo electrónico de cada usuario debe ser única.

Para obtener más información sobre la creación de usuarios AppStream 2.0, consulte [Grupos de usuarios AppStream 2.0](#) en la Guía de administración de Amazon AppStream 2.0.

Cuando AWS CloudFormation termine de crear la pila, puede asignar el usuario que creó a la pila de la siguiente manera:

1. Abre la consola AppStream 2.0.
2. Elija el nombre de usuario.
3. Elija Acciones y luego Asignar pila.
4. En Asignar pila, elija la pila que empieza por m2-appstream-stack-ea.
5. Elija Asignar pila.

**Assign stack** ✕

Select a stack to enable access to the user(s) below.

User(s) being assigned

- Mary Major (mary.major@example.com)

Stack

m2-appstream-stack-ea-c92d75b0 ▼

Send email notification to user

Cancel **Assign stack**

Al asignar un usuario a una pila, AppStream 2.0 envía un correo electrónico al usuario a la dirección que ha proporcionado. Este correo electrónico contiene un enlace a la página de inicio de sesión de AppStream 2.0.

## Paso 4: Inicie sesión en AppStream 2.0

En este paso, inicia sesión en AppStream 2.0 mediante el enlace del correo electrónico enviado por AppStream 2.0 al usuario que creó [Paso 3: Crea un usuario en la AppStream versión 2.0](#).

1. Inicie sesión en AppStream 2.0 mediante el enlace que se proporciona en el correo electrónico enviado por AppStream 2.0.
2. Cambie la contraseña si se le solicita. La pantalla AppStream 2.0 que aparece es similar a la siguiente:



3. Seleccione Escritorio.
4. En la barra de tareas, seleccione Buscar e introduzca **D :** para navegar hasta la carpeta de inicio.

**Note**

Si omite este paso, es posible que aparezca el error `Device not ready` al intentar acceder a la carpeta de inicio.

En cualquier momento, si tienes problemas para iniciar sesión en la AppStream versión 2.0, puedes reiniciar tu flota de versiones AppStream 2.0 e intentar iniciar sesión de nuevo siguiendo estos pasos.

1. Abre la consola AppStream 2.0.
2. En el panel de navegación izquierdo, elija Flotas.
3. Elija la flota que está intentando utilizar.
4. Seleccione Acción y, a continuación, Detener.
5. Espere a que la flota se detenga.
6. Seleccione Acción y, a continuación, seleccione Iniciar.

Este proceso puede tardar unos 10 minutos.

## Paso 5: Verificar los buckets de Amazon S3 (opcional)

Una de las tareas que llevó a cabo la AWS CloudFormation plantilla que utilizó para crear la pila consistía en crear dos depósitos en Amazon S3, que son necesarios para guardar y restaurar los datos de los usuarios y la configuración de la aplicación en las sesiones de trabajo. Estos buckets son los siguientes:

- Su nombre comienza por `appstream2-`. Este bucket asigna los datos a su carpeta de inicio en la AppStream versión 2.0 (`D:\PhotonUser\My Files\Home Folder`).

### Note

La carpeta de inicio es única para una dirección de correo electrónico determinada y se comparte entre todas las flotas y pilas de una cuenta determinada AWS. El nombre de la carpeta de inicio es un hash SHA256 de la dirección de correo electrónico del usuario y se almacena en una ruta basada en ese hash.

- Su nombre comienza por `appstream-app-settings-`. Este depósito contiene información sobre las sesiones de usuario de la AppStream versión 2.0 e incluye ajustes como los favoritos del navegador, los perfiles de conexión del IDE y de las aplicaciones y las personalizaciones de la interfaz de usuario. Para obtener más información, consulte [Cómo funciona la persistencia de la configuración de aplicaciones](#) en la Guía de administración de Amazon AppStream 2.0.

Para comprobar que se han creado los buckets, siga estos pasos:

1. Abra la consola de Amazon S3.
2. En el panel de navegación izquierdo, elija Buckets.
3. En Buscar buckets por nombre, introduzca **appstream** para filtrar la lista.

Si ve los buckets, no es necesario realizar ninguna otra acción. Solo es para saber que los buckets existen. Si no ve los cubos, significa que la AWS CloudFormation plantilla no ha terminado de ejecutarse o que se ha producido un error. Ve a la AWS CloudFormation consola y revisa los mensajes de creación de la pila.



## Siguientes pasos

Ahora que la infraestructura AppStream 2.0 está configurada, puede configurar y empezar a utilizar Enterprise Analyzer. Para obtener más información, consulte [Tutorial: Configurar Enterprise Analyzer en 2.0 AppStream](#). También puede configurar Enterprise Developer. Para obtener más información, consulte [Tutorial: Configurar Micro Focus Enterprise Developer en AppStream 2.0](#).

## Eliminar recursos

El procedimiento para limpiar la pila y las flotas creadas se describe en [Crear una flota y una pila AppStream 2.0](#).

Cuando se hayan eliminado los objetos AppStream 2.0, el administrador de la cuenta también puede, si procede, limpiar los depósitos de Amazon S3 para la configuración de la aplicación y las carpetas de inicio.

### Note

La carpeta de inicio de un usuario determinado es única en todas las flotas, por lo que es posible que tengas que conservarla si hay otras pilas AppStream 2.0 activas en la misma cuenta.

Por último, la AppStream versión 2.0 actualmente no permite eliminar usuarios mediante la consola. Para ello, debe usar la API de servicio con la CLI. Para obtener más información, consulte [Administración de grupos de usuarios](#) en la Guía de administración de Amazon AppStream 2.0.

## Tutorial: Configurar Enterprise Analyzer en 2.0 AppStream

Este tutorial describe cómo configurar Micro Focus Enterprise Analyzer para analizar una o más aplicaciones de unidad central. La herramienta Enterprise Analyzer proporciona varios informes basados en su análisis del código fuente de la aplicación y las definiciones del sistema.

Esta configuración está diseñada para fomentar la colaboración en equipo. La instalación utiliza un bucket de Amazon S3 para compartir el código fuente con los discos virtuales. Para ello, se utiliza [Rclone](#) en el equipo de Windows. Con una instancia de Amazon RDS común que ejecute [PostgreSQL](#), cualquier miembro del equipo puede acceder a todos los informes solicitados.

Los miembros del equipo también pueden montar el disco virtual respaldado por Amazon S3 en sus equipos personales y actualizar el bucket de origen desde sus estaciones de trabajo. También pueden usar scripts o cualquier otra forma de automatización en sus equipos si están conectados a otros sistemas internos en las instalaciones.

La configuración se basa en las imágenes de Windows AppStream 2.0 que AWS Mainframe Modernization comparte con el cliente. La configuración también se basa en la creación de flotas y pilas AppStream 2.0, tal como se describe en [Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#)

#### Important

En los pasos de este tutorial se supone que configuró la AppStream versión 2.0 con la AWS CloudFormation plantilla descargable [cfn-m2-appstream-fleet-ea-ed.yml](#). Para obtener más información, consulte [Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#).

Para realizar los pasos en este tutorial, debe haber configurado la flota y la pila de Enterprise Analyzer y deben estar en funcionamiento.

Para obtener una descripción completa de las características y prestaciones de Enterprise Analyzer, consulte la [documentación de Enterprise Analyzer en el sitio web](#) de Micro Focus.

## Contenido de la imagen

Además de la propia aplicación Enterprise Analyzer, la imagen contiene las siguientes herramientas y bibliotecas.

### Herramientas de terceros

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [git-scm](#)
- [Controlador ODBC para PostgreSQL](#)

### Bibliotecas en C:\Users\Public

- BankDemo código fuente y definición m2-bankdemo-template.zip del proyecto para Enterprise Developer:.
- Paquete de instalación de MFA para el sistema central: mfa.zip. Para obtener más información, consulte [Mainframe Access Overview](#) en la documentación de Micro Focus Enterprise Developer.
- Archivos de comandos y configuración para Rclone (instrucciones para su uso en los tutoriales): m2-rclone.cmd y m2-rclone.conf.

## Temas

- [Requisitos previos](#)
- [Paso 1: Configuración](#)
- [Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows](#)
- [Paso 3: Crear un origen de ODBC para la instancia de Amazon RDS](#)
- [Sesiones posteriores](#)
- [Solución de problemas de conexión de espacio de trabajo](#)
- [Eliminar recursos](#)

## Requisitos previos

- Cargue el código fuente y las definiciones del sistema de la aplicación del cliente que desee analizar en un bucket de S3. Las definiciones del sistema incluyen el CICS, el CSD, las definiciones de objetos del DB2, etc. Puede crear una estructura de carpetas dentro del bucket que se adapte a la forma en que desea organizar los artefactos de la aplicación. Por ejemplo, cuando descomprima la BankDemo muestra, tiene la siguiente estructura:

```
demo
  |--> jcl
  |--> RDEF
  |--> transaction
  |--> xa
```

- Cree una instancia de Amazon RDS que ejecute PostgreSQL e iníciela. Esta instancia almacenará los datos y los resultados que produzca Enterprise Analyzer. Puede compartir esta instancia con todos los miembros del equipo de la aplicación. Además, cree un esquema vacío llamado m2\_ea (o cualquier otro nombre adecuado) en la base de datos. Defina las credenciales para los usuarios autorizados que les permitan crear, insertar, actualizar y eliminar elementos de este esquema.

Puede obtener el nombre de la base de datos, la URL del punto de conexión del servidor y el puerto TCP desde la consola de Amazon RDS o a través del administrador de la cuenta.

- Asegúrese de haber configurado el acceso programático a su. Cuenta de AWS Para obtener más información, consulte [Acceso programático](#) en Referencia general de Amazon Web Services.

## Paso 1: Configuración

1. Inicie una sesión con la AppStream versión 2.0 con la URL que recibió en el mensaje de correo electrónico de bienvenida de la AppStream versión 2.0.
2. Utilice su correo electrónico como ID de usuario y defina su contraseña permanente.
3. Seleccione la pila de Enterprise Analyzer.
4. En la página del menú AppStream 2.0, selecciona Escritorio para acceder al escritorio de Windows que está transmitiendo la flota.

## Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows

### Note

Si ya utilizó Rclone durante la versión preliminar de AWS Mainframe Modernization, debe actualizar `m2-rclone.cmd` a la versión más reciente que se encuentra en. `C:\Users\Public`

1. Copie los archivos `m2-rclone.conf` y `m2-rclone.cmd` proporcionados en `C:\Users\Public` a su carpeta de inicio `C:\Users\PhotonUser\My Files\Home Folder` mediante el Explorador de archivos.
2. Actualice los parámetros de `m2-rclone.conf` configuración con su clave de AWS acceso y el secreto correspondiente, así como con su. Región de AWS

```
[m2-s3]
type = s3
provider = AWS
access_key_id = YOUR-ACCESS-KEY
secret_access_key = YOUR-SECRET-KEY
region = YOUR-REGION
acl = private
```

```
server_side_encryption = AES256
```

3. En `m2-rc1one.cmd`, realice los siguientes cambios:

- Cambie `your-s3-bucket` por el nombre de su bucket de Amazon S3. Por ejemplo, `m2-s3-mybucket`.
- Cambie `your-s3-folder-key` por el nombre de su clave de bucket de Amazon S3. Por ejemplo, `myProject`.
- Cambie `your-local-folder-path` por la ruta del directorio en el que desee sincronizar los archivos de la aplicación desde el bucket de Amazon S3 que los contiene. Por ejemplo, `D:\PhotonUser\My Files\Home Folder\m2-new`. Este directorio sincronizado debe ser un subdirectorio de la carpeta principal para que la AppStream versión 2.0 pueda realizar copias de seguridad y restaurarlo correctamente al iniciar y finalizar la sesión.

```
:loop
timeout /T 10
"C:\Program Files\rc1one\rc1one.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rc1one.conf"
goto :loop
```

4. Abra una línea de comandos de Windows, ejecute un `cd` a `C:\Users\PhotonUser\My Files\Home Folder` si es necesario y ejecute `m2-rc1one.cmd`. Este script de comandos ejecuta un bucle continuo y sincroniza el bucket y la clave de Amazon S3 con la carpeta local cada 10 segundos. Puede ajustar el tiempo de espera según sea necesario. Debería ver el código fuente de la aplicación ubicado en el bucket de Amazon S3 en el Explorador de archivos de Windows.

Para añadir nuevos archivos al conjunto en el que está trabajando o actualizar los existentes, cargue los archivos en el bucket de Amazon S3 y se sincronizarán con su directorio en la siguiente iteración definida en `m2-rc1one.cmd`. Del mismo modo, si desea eliminar algunos archivos, elimínelos del bucket de Amazon S3. La siguiente operación de sincronización los eliminará de su directorio local.

## Paso 3: Crear un origen de ODBC para la instancia de Amazon RDS

1. Para iniciar la herramienta `EA_Admin`, navegue hasta el menú de selección de aplicaciones situado en la esquina superior izquierda de la ventana del navegador y seleccione `MF EA_Admin`.

2. En el menú Administrar, seleccione Orígenes de datos de ODBC y, en la pestaña DSN de usuario, seleccione Agregar.
3. En el cuadro de diálogo Crear nuevo origen de datos, elija el controlador PostgreSQL Unicode y, a continuación, elija Finalizar.
4. En el cuadro de diálogo Configuración de controlador ODBC Unicode PostgreSQL (psqlODBC), defina y anote el nombre del origen de datos que desee. Complete los siguientes parámetros con los valores de la instancia de RDS que ha creado anteriormente:

#### Descripción

Descripción opcional para ayudarle a identificar rápidamente esta conexión a la base de datos.

#### Database

La base de datos de Amazon RDS que ha creado anteriormente.

#### Server

El punto de conexión de Amazon RDS.

#### Puerto

El puerto de Amazon RDS.

#### User Name

El que se ha definido en la instancia de Amazon RDS.

#### Password

La que se ha definido en la instancia de Amazon RDS.

5. Elija Prueba para validar que la conexión a Amazon RDS se ha realizado correctamente y, a continuación, seleccione Guardar para guardar su nuevo DSN de usuario.
6. Espere a que aparezca el mensaje que confirme la creación del espacio de trabajo adecuado y, a continuación, pulse Aceptar para terminar con los orígenes de datos de ODBC y cerrar la herramienta EA\_Admin.
7. Vuelva al menú de selección de aplicaciones y elija Enterprise Analyzer para iniciar la herramienta. Elija Crear nuevo.
8. En la ventana de configuración del espacio de trabajo, introduzca el nombre del espacio de trabajo y defina su ubicación. El espacio de trabajo puede ser el disco basado en Amazon S3 si trabaja con esta configuración, o su carpeta de inicio si lo prefiere.

9. Seleccione Elegir otra base de datos para conectarse a su instancia de Amazon RDS.
10. Elija el icono de Postgre de entre las opciones y, a continuación, pulse Aceptar.
11. Para la configuración de Windows, en Opciones: Definir parámetros de conexión, introduzca el nombre del origen de datos que ha creado. Introduzca también el nombre de la base de datos, el nombre de esquema, el nombre de usuario y la contraseña. Seleccione Aceptar.
12. Espere a que Enterprise Analyzer cree todas las tablas, índices, etc. que necesite para almacenar los resultados. Es posible que tarde un par de minutos. Enterprise Analyzer confirma cuándo la base de datos y el espacio de trabajo están listos para su uso.
13. Vuelva al menú de selección de aplicaciones y elija Enterprise Analyzer para iniciar la herramienta.
14. La ventana de inicio de Enterprise Analyzer aparece en la nueva ubicación del espacio de trabajo seleccionada. Seleccione Aceptar.
15. Navegue hasta el repositorio en el panel izquierdo, seleccione el nombre del repositorio y elija Añadir archivos/carpetas al espacio de trabajo. Seleccione la carpeta en la que está almacenado el código de la aplicación para añadirlo al espacio de trabajo. Si lo desea, puede usar el código del BankDemo ejemplo anterior. Cuando Enterprise Analyzer le pida que verifique esos archivos, elija Verificar para iniciar el informe de verificación inicial de Enterprise Analyzer. Este proceso puede tardar unos minutos, dependiendo del tamaño de la aplicación.
16. Amplíe su espacio de trabajo para ver los archivos y carpetas que ha agregado al espacio de trabajo. Los tipos de objetos y los informes de complejidad ciclomática también están visibles en el cuadrante superior del panel Visor de gráficos.

Ahora puede usar Enterprise Analyzer para todas las tareas necesarias.

## Sesiones posteriores

1. Inicie una sesión con AppStream 2.0 con la URL que recibió en el mensaje de correo electrónico de bienvenida de AppStream 2.0.
2. Inicie sesión con su correo electrónico y contraseña permanente.
3. Seleccione la pila de Enterprise Analyzer.
4. Inicie Rclone para conectarse al disco respaldado por Amazon S3 si utiliza esta opción para compartir los archivos del espacio de trabajo.
5. Inicie Enterprise Analyzer para hacer sus tareas.

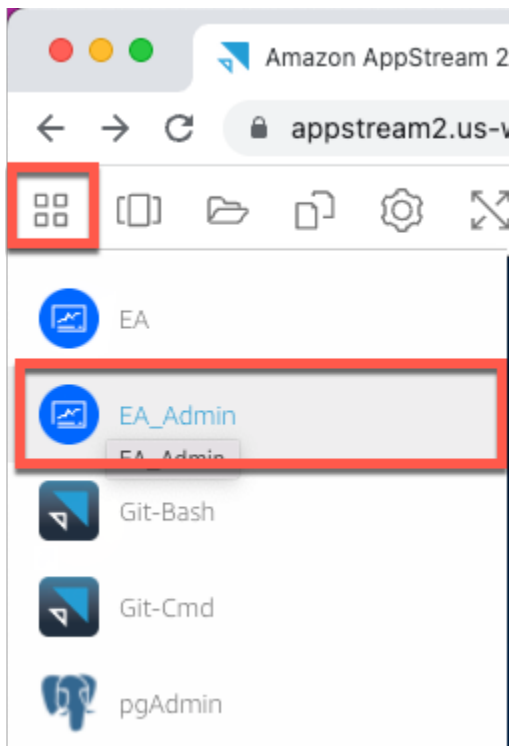
## Solución de problemas de conexión de espacio de trabajo

Cuando intente volver a conectarse a su espacio de trabajo de Enterprise Analyzer, es posible que aparezca un error como el siguiente:

```
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?
```

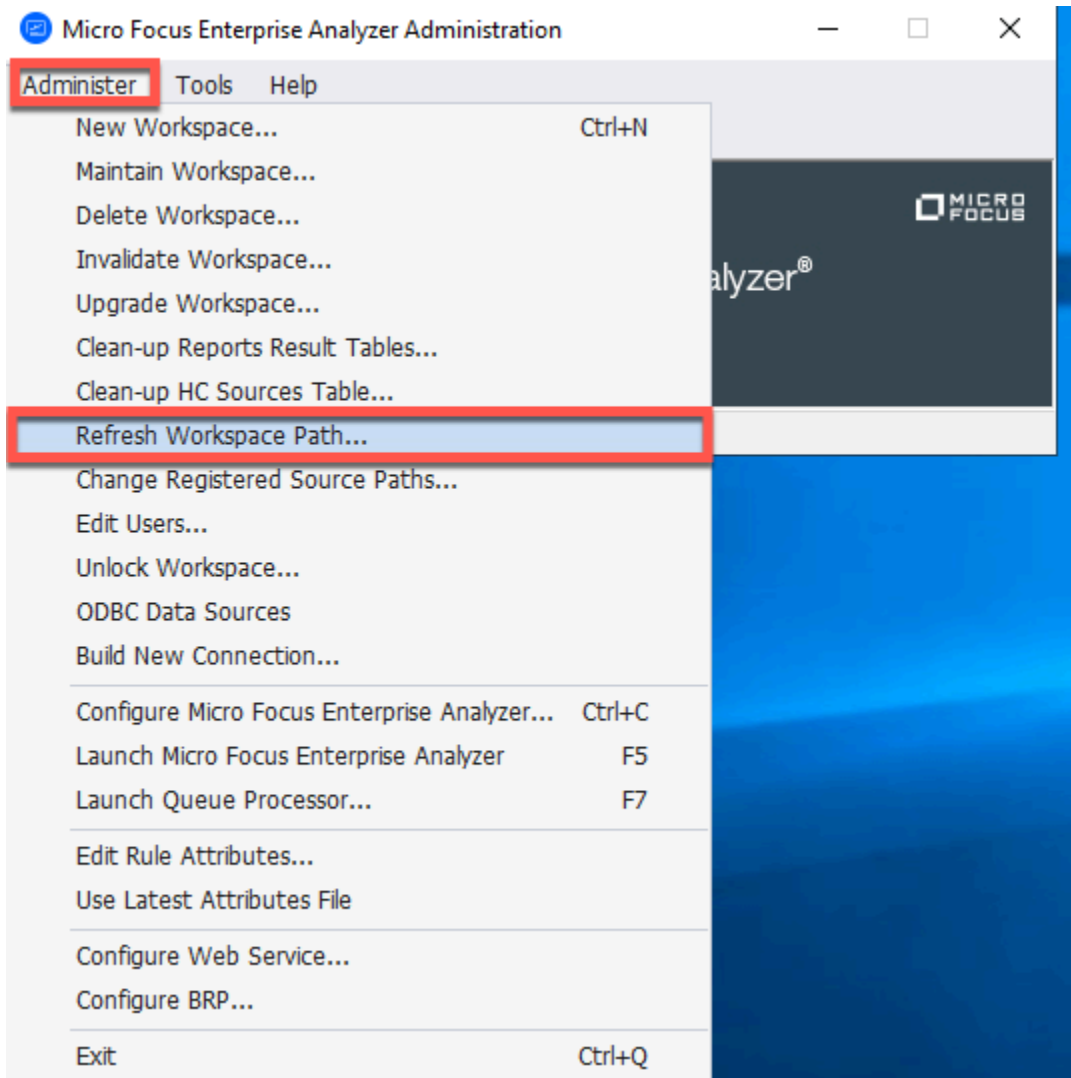
Para resolver este problema, pulse Aceptar para borrar el mensaje y, a continuación, complete los pasos siguientes.

1. En la AppStream versión 2.0, pulse el icono Iniciar aplicación en la barra de herramientas y, a continuación, elija EA\_Admin para iniciar la herramienta de administración de Micro Focus Enterprise Analyzer.

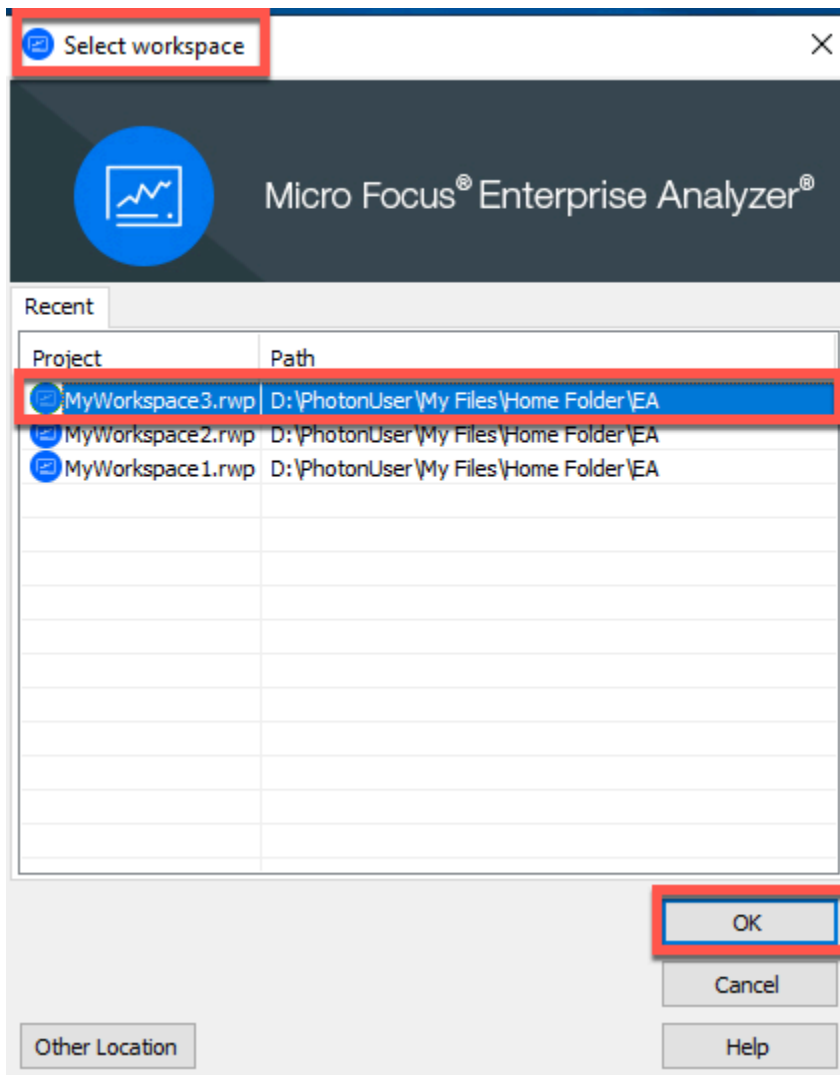


2. En el menú Administrar, seleccione Actualizar ruta de espacio de trabajo....

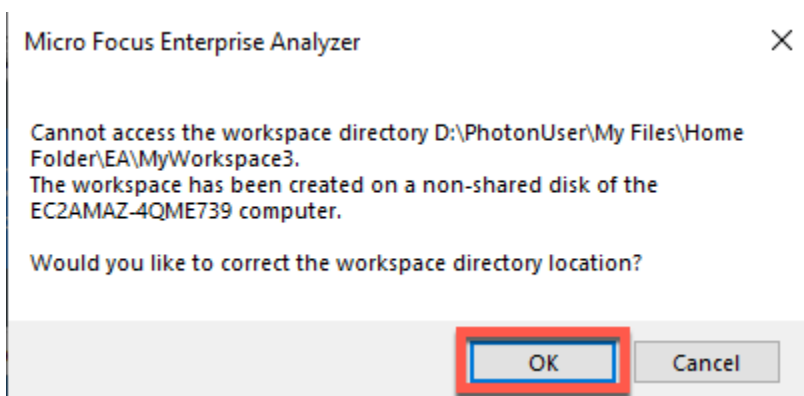




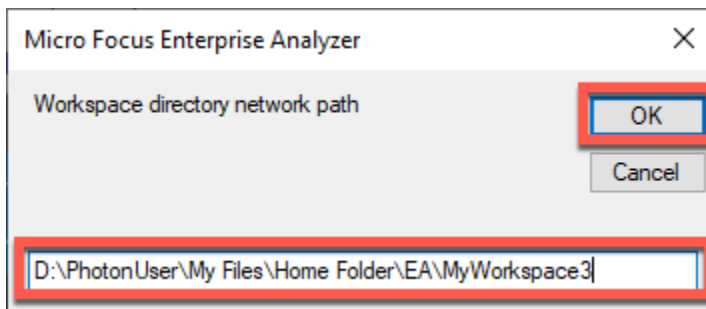
3. En Seleccionar espacio de trabajo, elija el espacio de trabajo que desee y, a continuación, pulse Aceptar.



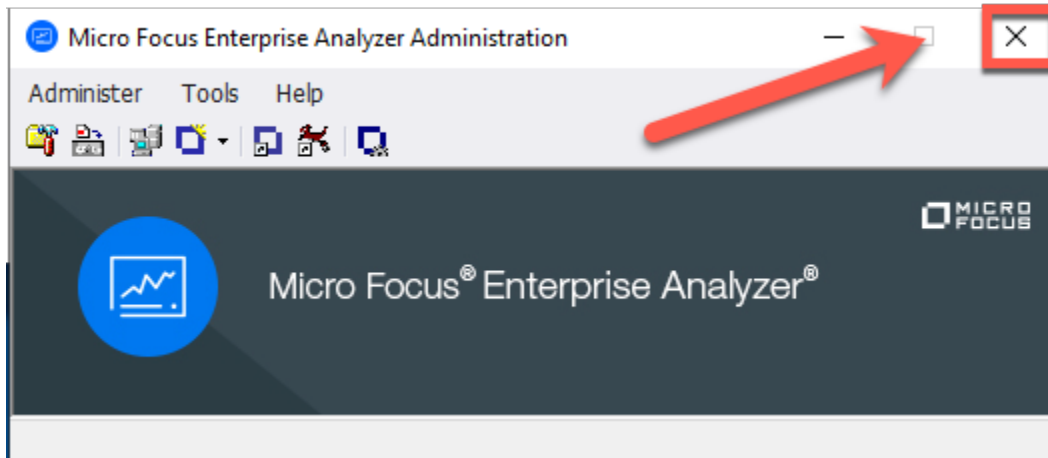
4. Pulse Aceptar para confirmar el mensaje de error.



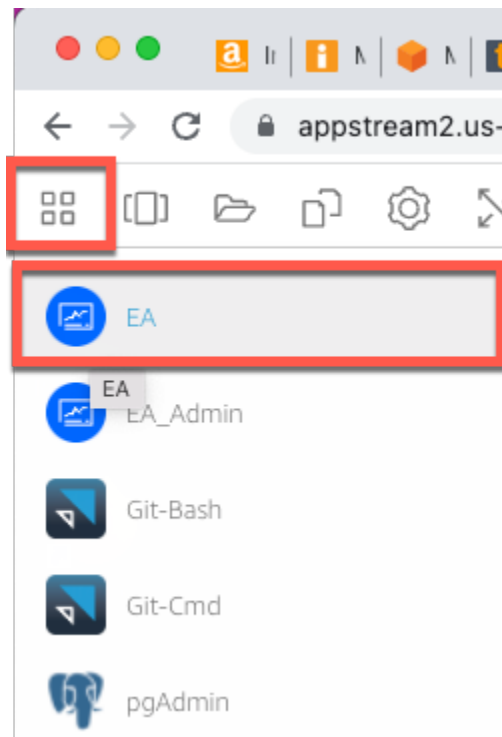
5. En Ruta de red del directorio de espacio de trabajo, introduzca la ruta correcta a su espacio de trabajo, por ejemplo D:\PhotonUser\My Files\Home Folder\EA\MyWorkspace3.



6. Cierre la herramienta Micro Focus Enterprise Analyzer Administration.



7. En la AppStream versión 2.0, pulse el icono Iniciar aplicación en la barra de herramientas y, a continuación, seleccione EA para iniciar Micro Focus Enterprise Analyzer.



## 8. Repita los pasos 3 a 5.

Ahora Micro Focus Enterprise Analyzer debería abrirse con el espacio de trabajo existente.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para que no se le cobre por ellos. Realice los siguientes pasos:

- Use la herramienta EA\_Admin para eliminar el espacio de trabajo.
- Elimine los buckets de S3 que ha creado para este tutorial. Para obtener más información, consulte [Eliminar un bucket](#) en la Guía del usuario de Amazon S3.
- Elija la base de datos que ha creado para este tutorial. Para obtener más información, consulte [Eliminación de una instancia de base de datos](#).

## Tutorial: Configurar Micro Focus Enterprise Developer en AppStream 2.0

Este tutorial describe cómo configurar Micro Focus Enterprise Developer para una o más aplicaciones de unidad central con el fin de mantenerlas, compilarlas y probarlas mediante las características de Enterprise Developer. La configuración se basa en las imágenes de Windows AppStream 2.0 que AWS Mainframe Modernization comparte con el cliente y en la creación de flotas y pilas AppStream 2.0, tal como se describe en [Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#)

### Important

En los pasos de este tutorial se supone que configuras la AppStream versión 2.0 con la AWS CloudFormation plantilla descargable [cfn-m2-appstream-fleet-ea-ed.yaml](#). Para obtener más información, consulte [Tutorial: Configurar AppStream 2.0 para su uso con Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer](#).

Debe realizar los pasos de esta configuración cuando la flota y la pila de Enterprise Developer estén en ejecución.

[Para obtener una descripción completa de las funciones y prestaciones de Enterprise Developer v7, consulte su documentación en línea \(v7.0\) en el sitio de Micro Focus. up-to-date](#)

## Contenido de la imagen

Además del propio Enterprise Developer, la imagen contiene la imagen que contiene Rumba (un emulador TN3270). También contiene las siguientes herramientas y bibliotecas.

### Herramientas de terceros

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [git-scm](#)
- [Controlador ODBC para PostgreSQL](#)

### Bibliotecas en C:\Users\Public

- BankDemo código fuente y definición de proyecto para Enterprise Developer: `m2-bankdemo-template.zip`
- Paquete de instalación de MFA para el sistema central: `mfa.zip`. Para obtener más información, consulte [Mainframe Access Overview](#) en la documentación de Micro Focus Enterprise Developer.
- Archivos de comandos y configuración para Rclone (instrucciones para su uso en los tutoriales): `m2-rclone.cmd` y `m2-rclone.conf`.

Si necesita acceder a un código fuente que aún no está cargado en CodeCommit los repositorios, pero que está disponible en un bucket de Amazon S3, por ejemplo, para realizar la carga inicial del código fuente en git, siga el procedimiento para crear un disco virtual de Windows tal y como se describe en [Tutorial: Configurar Enterprise Analyzer en 2.0 AppStream](#).

### Temas

- [Requisitos previos](#)
- [Paso 1: Configuración por parte de usuarios individuales de Enterprise Developer](#)
- [Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows \(opcional\)](#)
- [Paso 3: Clonar el repositorio](#)

- [Sesiones posteriores](#)
- [Eliminar recursos](#)

## Requisitos previos

- Uno o más CodeCommit repositorios cargados con el código fuente de la aplicación que se va a mantener. La configuración del repositorio debe cumplir con los requisitos de la canalización de CI/CD anterior para crear sinergias mediante la combinación de ambas herramientas.
- Cada usuario debe tener credenciales para el CodeCommit repositorio o los repositorios definidos por el administrador de la cuenta de acuerdo con la información de [Autenticación y control de acceso para AWS CodeCommit](#). La estructura de esas credenciales se analiza en [Autenticación y control de acceso para AWS CodeCommit](#) y la referencia completa para las autorizaciones de IAM CodeCommit se encuentra en la [referencia de CodeCommit permisos](#): el administrador puede definir políticas de IAM distintas para funciones distintas con credenciales específicas para la función de cada repositorio y limitar las autorizaciones del usuario al conjunto específico de tareas que debe realizar en un repositorio determinado. Por lo tanto, para cada responsable del CodeCommit repositorio, el administrador de la cuenta generará un usuario principal y le concederá permisos para acceder al repositorio o repositorios necesarios seleccionando la política o políticas de CodeCommit acceso de IAM adecuadas.

## Paso 1: Configuración por parte de usuarios individuales de Enterprise Developer

1. Obtenga sus credenciales de IAM:
  1. Conéctese a la AWS consola en <https://console.aws.amazon.com/iam/>.
  2. Siga el procedimiento descrito en el paso 3 de [Setup for HTTPS users using Git credentials](#) en la guía del usuario de AWS CodeCommit .
  3. Copie las credenciales CodeCommit de inicio de sesión específicas que IAM generó para usted, ya sea mostrando, copiando y pegando esta información en un archivo seguro de su equipo local, o seleccionando Descargar credenciales para descargar esta información como un archivo.CSV. Necesitas esta información para conectarte a él. CodeCommit
2. Inicie una sesión con AppStream 2.0 en función de la URL recibida en el correo electrónico de bienvenida. Utilice su correo electrónico como nombre de usuario y cree su contraseña.
3. Seleccione su pila de Enterprise Developer.

4. En la página de menú, elija Escritorio para acceder al escritorio de Windows que está transmitiendo la flota.

## Paso 2: Crear la carpeta virtual basada en Amazon S3 en Windows (opcional)

Si necesita Rclone (consulte más arriba), cree la carpeta virtual basada en Amazon S3 en Windows: (opcional si todos los artefactos de la aplicación provienen exclusivamente de Access). CodeCommit

### Note

Si ya utilizó Rclone durante la versión preliminar de AWS Mainframe Modernization, debe actualizar `m2-rclone.cmd` a la versión más reciente que se encuentra en `C:\Users\Public`

1. Copie los archivos `m2-rclone.conf` y `m2-rclone.cmd` proporcionados en `C:\Users\Public` a su carpeta de inicio `C:\Users\PhotonUser\My Files\Home Folder` mediante el Explorador de archivos.
2. Actualice los parámetros de `m2-rclone.conf` configuración con su clave de AWS acceso y el secreto correspondiente, así como con su. Región de AWS

```
[m2-s3]
type = s3
provider = AWS
access_key_id = YOUR-ACCESS-KEY
secret_access_key = YOUR-SECRET-KEY
region = YOUR-REGION
acl = private
server_side_encryption = AES256
```

3. En `m2-rclone.cmd`, realice los siguientes cambios:
  - Cambie `your-s3-bucket` por el nombre de su bucket de Amazon S3. Por ejemplo, `m2-s3-mybucket`.
  - Cambie `your-s3-folder-key` por el nombre de su clave de bucket de Amazon S3. Por ejemplo, `myProject`.

- Cambie `your-local-folder-path` por la ruta del directorio en el que desee sincronizar los archivos de la aplicación desde el bucket de Amazon S3 que los contiene. Por ejemplo, `D:\PhotonUser\My Files\Home Folder\m2-new`. Este directorio sincronizado debe ser un subdirectorio de la carpeta principal para que la AppStream versión 2.0 pueda realizar copias de seguridad y restaurarlo correctamente al iniciar y finalizar la sesión.

```
:loop
timeout /T 10
"C:\Program Files\rclone\rclone.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rclone.conf"
goto :loop
```

4. Abra una línea de comandos de Windows, ejecute un `cd` a `C:\Users\PhotonUser\My Files\Home Folder` si es necesario y ejecute `m2-rclone.cmd`. Este script de comandos ejecuta un bucle continuo y sincroniza el bucket y la clave de Amazon S3 con la carpeta local cada 10 segundos. Puede ajustar el tiempo de espera según sea necesario. Debería ver el código fuente de la aplicación ubicado en el bucket de Amazon S3 en el Explorador de archivos de Windows.

Para añadir nuevos archivos al conjunto en el que está trabajando o actualizar los existentes, cargue los archivos en el bucket de Amazon S3 y se sincronizarán con su directorio en la siguiente iteración definida en `m2-rclone.cmd`. Del mismo modo, si desea eliminar algunos archivos, elimínelos del bucket de Amazon S3. La siguiente operación de sincronización los eliminará de su directorio local.

### Paso 3: Clonar el repositorio

1. Navegue hasta el menú selector de aplicaciones en la esquina superior izquierda de la ventana del navegador y seleccione Enterprise Developer.
2. Complete la creación del espacio de trabajo requerida por Enterprise Developer en su carpeta de inicio eligiendo `C:\Users\PhotonUser\My Files\Home Folder` (también conocida como `D:\PhotonUser\My Files\Home Folder`) como ubicación del espacio de trabajo.
3. En Enterprise Developer, clona tu CodeCommit repositorio yendo al Explorador de proyectos, haz clic con el botón derecho y selecciona Importar, Importar..., Git, Projects from Git Clone URI. A continuación, introduce tus credenciales CodeCommit de inicio de sesión específicas y completa el cuadro de diálogo de Eclipse para importar el código.



El repositorio de CodeCommit git ahora está clonado en tu espacio de trabajo local.

Su espacio de trabajo de Enterprise Developer ya está listo para iniciar los trabajos de mantenimiento de su aplicación. En concreto, puede utilizar la instancia local de Microfocus Enterprise Server (ES) integrada con Enterprise Developer para depurar y ejecutar la aplicación de forma interactiva y validar los cambios a nivel local.

#### Note

El entorno de Enterprise Developer local, incluida la instancia local de Enterprise Server, se ejecuta en Windows, mientras que AWS Mainframe Modernization se ejecuta en Linux. Le recomendamos que ejecute pruebas complementarias en el entorno Linux proporcionado por AWS Mainframe Modernization después de instalar la nueva aplicación CodeCommit y reconstruirla para este objetivo y antes de lanzar la nueva aplicación a producción.

## Sesiones posteriores

Al seleccionar una carpeta que esté gestionada por la AppStream versión 2.0, como la carpeta principal, para clonar el CodeCommit repositorio, se guardará y restaurará de forma transparente en todas las sesiones. Complete los siguientes pasos la próxima vez que necesite trabajar con la aplicación:

1. Inicie una sesión con la AppStream versión 2.0 en función de la URL recibida en el correo electrónico de bienvenida.
2. Inicie sesión con su correo electrónico y contraseña permanente.
3. Seleccione la pila de Enterprise Developer.
4. Inicie Rc1one para conectarse (véase anteriormente) al disco respaldado por Amazon S3 si utiliza esta opción para compartir los archivos del espacio de trabajo.
5. Inicie Enterprise Developer para hacer su trabajo.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para que no se le siga cobrando por ellos. Realice los siguientes pasos:

- Elimine el CodeCommit repositorio que creó para este tutorial. Para obtener más información, consulte [Eliminar un CodeCommit repositorio](#) en la Guía del AWS CodeCommit usuario.
- Elimine la base de datos que ha creado para este tutorial. Para obtener más información, consulte [Eliminación de una instancia de base de datos](#).

## Configuración de la automatización para las sesiones en streaming de Micro Focus Enterprise Analyzer y Micro Focus Enterprise Developer

Puede ejecutar automáticamente un script al inicio y al final de la sesión para permitir una automatización específica del contexto de su cliente. Para obtener más información sobre esta función AppStream 2.0, consulte [Uso de scripts de sesión para gestionar la experiencia de streaming de los usuarios de la AppStream versión 2.0](#) en la Guía de administración de Amazon AppStream 2.0.

Esta característica requiere que tenga al menos las siguientes versiones de las imágenes de Enterprise Analyzer y Enterprise Developer:

- `m2-enterprise-analyzer-v8.0.4.R1`
- `m2-enterprise-developer-v8.0.4.R1`

### Temas

- [Configuración de la automatización al inicio de la sesión](#)
- [Configuración de la automatización al final de la sesión](#)

## Configuración de la automatización al inicio de la sesión

Si desea ejecutar un script de automatización cuando los usuarios se conecten a la AppStream versión 2.0, cree el suyo y asígnele un nombre `m2-user-setup.cmd`. Guarde el script en la carpeta principal de la AppStream versión 2.0 para el usuario. Las imágenes AppStream 2.0 que proporciona AWS Mainframe Modernization buscan un script con ese nombre en esa ubicación y lo ejecutan si existe.

**Note**

La duración del guion no puede superar el límite establecido en AppStream 2.0, que actualmente es de 60 segundos. Para obtener más información, consulte [Ejecutar scripts antes de que comiencen las sesiones de streaming](#) en la Guía de administración de Amazon AppStream 2.0.

## Configuración de la automatización al final de la sesión

Si desea ejecutar un script de automatización cuando los usuarios se desconecten de la AppStream versión 2.0, cree el suyo y asígnele un nombre `2-user-teardown.cmd`. Guarde el script en la carpeta principal de la AppStream versión 2.0 para el usuario. Las imágenes AppStream 2.0 que proporciona AWS Mainframe Modernization buscan un script con ese nombre en esa ubicación y lo ejecutan si existe.

**Note**

La duración del guion no puede superar el límite establecido en AppStream 2.0, que actualmente es de 60 segundos. Para obtener más información, consulte [Ejecutar scripts tras finalizar las sesiones de streaming](#) en la Guía de administración de Amazon AppStream 2.0.

## Ver conjuntos de datos como tablas y columnas en Enterprise Developer

Puede acceder a los conjuntos de datos de mainframe que se implementan en AWS Mainframe Modernization mediante el entorno de ejecución de Micro Focus. Puede ver los conjuntos de datos migrados como tablas y columnas desde una instancia de Micro Focus Enterprise Developer. Ver los conjuntos de datos de esta manera le permite:

- Realizar operaciones SQL `SELECT` en los archivos de datos migrados.
- Exponer los datos fuera de la aplicación de unidad central migrada sin cambiar la aplicación.
- Filtrar fácilmente los datos y guardarlos como CSV u otros formatos de archivo.

**Note**

Los pasos 1 y 2 son actividades que se realizan una sola vez. Repita los pasos 3 y 4 para en conjunto de datos para crear las vistas de base de datos.

## Temas

- [Requisitos previos](#)
- [Paso 1: Configurar la conexión ODBC al almacén de datos de Micro Focus \(base de datos de Amazon RDS\)](#)
- [Paso 2: Crear el archivo MFDBFH.cfg](#)
- [Paso 3: Crear un archivo de estructura \(STR\) para el diseño del cuaderno](#)
- [Paso 4: Crear una vista de base de datos con el archivo de estructura \(STR\)](#)
- [Paso 5: Ver los conjuntos de datos de Micro Focus como tablas y columnas](#)

## Requisitos previos

- Debe tener acceso a Micro Focus Enterprise Developer Desktop a través de la versión 2.0. AppStream
- Debe tener una aplicación implementada y ejecutada en el marco de AWS Mainframe Modernization mediante el motor de ejecución Micro Focus.
- Va a almacenar los datos de la aplicación en la edición de PostgreSQL compatible con Aurora.

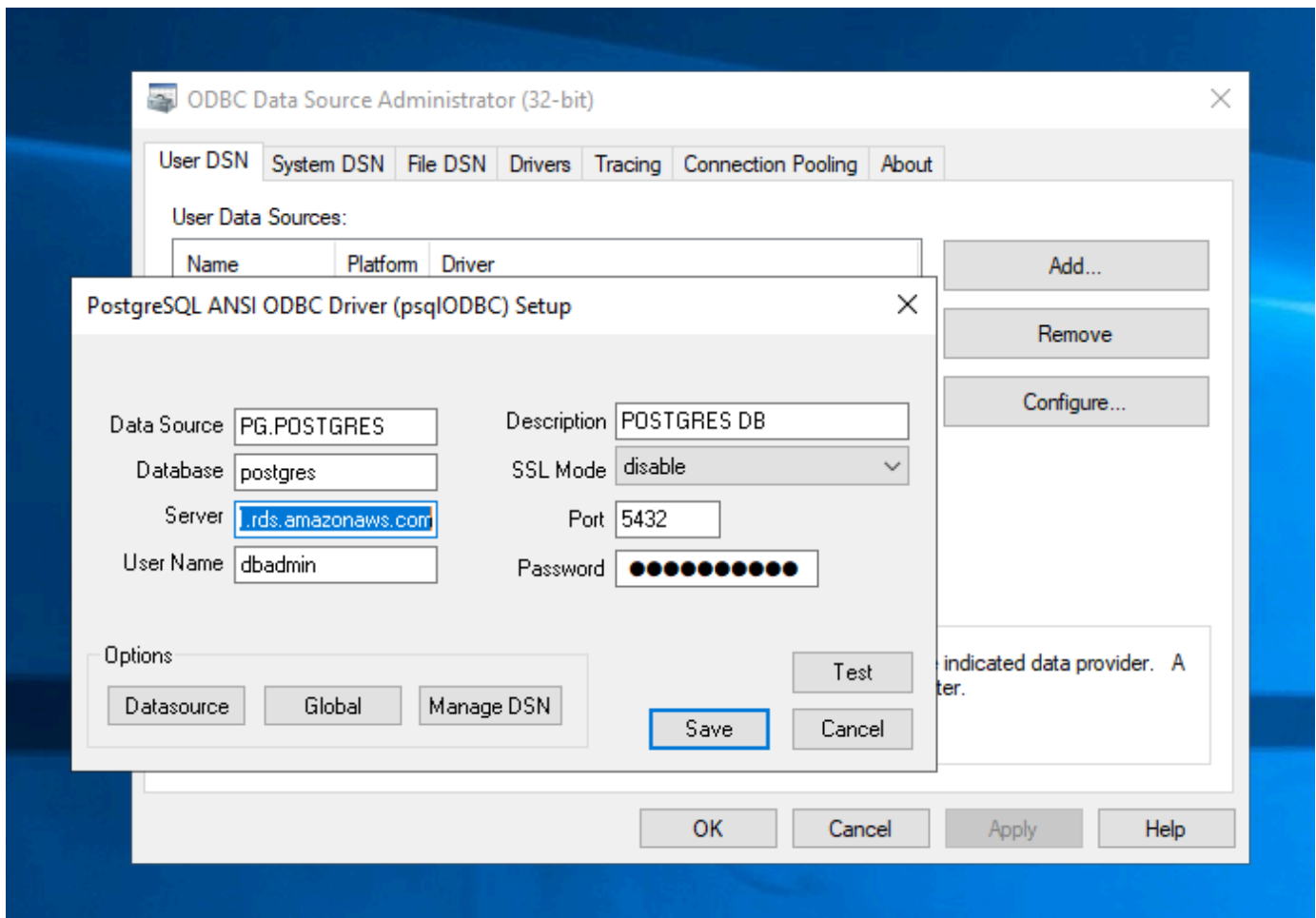
## Paso 1: Configurar la conexión ODBC al almacén de datos de Micro Focus (base de datos de Amazon RDS)

En este paso, configurará una conexión ODBC a la base de datos que contiene los datos que desee ver en forma de tablas y columnas. Solo tendrá que hacerlo una vez.

1. Inicie sesión en Micro Focus Enterprise Developer Desktop mediante la URL de transmisión AppStream 2.0.
2. Abra el Administrador de orígenes de datos ODBC, elija DSN de usuario y, a continuación, seleccione Agregar.
3. En Crear nuevo origen de datos, elija PostgreSQL ANSI y, a continuación, Finalizar.

4. Cree un origen de datos para PG.POSTGRES proporcionando la información de base de datos necesaria, de la siguiente manera:

```
Data Source : PG.POSTGRES
Database    : postgres
Server      : rds_endpoint.rds.amazonaws.com
Port        : 5432
User Name   : user_name
Password    : user_password
```



5. Elija Probar para asegurarse de que la conexión funciona. Si la prueba se realiza correctamente, aparecerá el mensaje Connection successful.

Si la prueba no tiene éxito, lea la siguiente información.

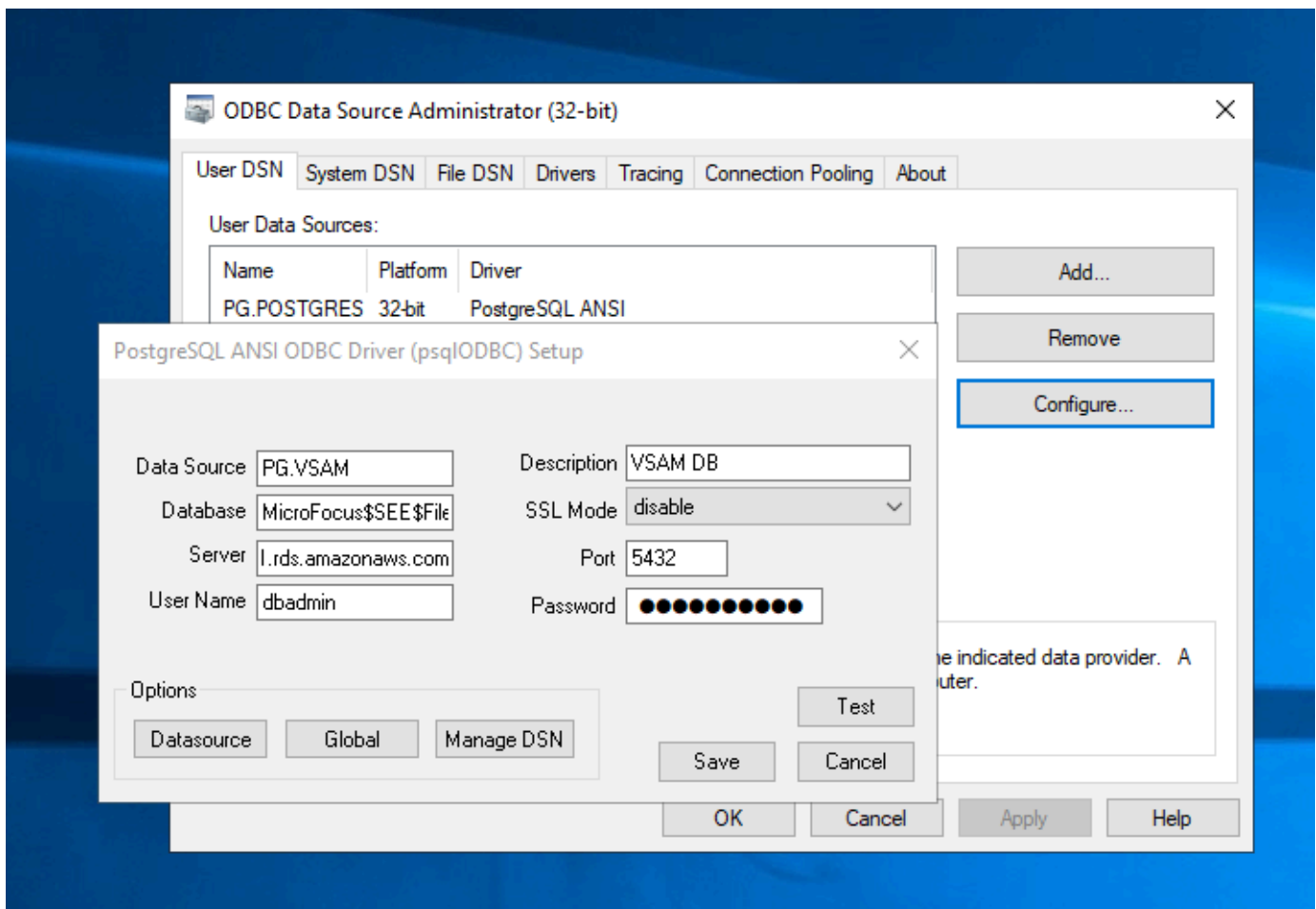
- [Solución de problemas de Amazon RDS](#)

- [¿Cómo resuelvo los problemas al conectarme a mi instancia de base de datos de Amazon RDS?](#)

6. Guarde el origen de datos.

7. Cree un origen de datos para PG.VSAM, pruebe la conexión y guarde el origen de datos. Proporcione la siguiente información de la base de datos:

Data Source : PG.VSAM  
 Database : MicroFocus\$SEE\$Files\$VSAM  
 Server : *rds\_endpoint*.rds.amazonaws.com  
 Port : 5432  
 User Name : *user\_name*  
 Password : *user\_password*



## Paso 2: Crear el archivo MFDBFH.cfg

En este paso, creará un archivo de configuración que describa el almacén de datos de Micro Focus. Este paso de configuración solo se tiene que realizar una vez.

1. En su carpeta de inicio, por ejemplo, en D:\PhotonUser\My Files\Home Folder\MFED\cfg\MFDBFH.cfg, cree el archivo MFDBFH.cfg con el siguiente contenido.

```
<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. Compruebe la configuración del MFDBFH ejecutando los siguientes comandos para consultar el almacén de datos de Micro Focus:

```
***
*** Test the connection by running the following commands*
***

set MFDBFH_CONFIG="D:\PhotonUser\My Files\Home Folder\MFED\cfg\MFDBFH.cfg"

dbfhdeploy list sql://ESPACDatabase/VSAM?folder=/DATA
```

## Paso 3: Crear un archivo de estructura (STR) para el diseño del cuaderno

En este paso, creará un archivo de estructura para el diseño del cuaderno de texto que utilizará más adelante para crear vistas de bases de datos a partir de los conjuntos de datos.

1. Compile el programa asociado al cuaderno. Si ningún programa utiliza el cuaderno, cree y compile un programa simple como el siguiente con una instrucción COPY para el cuaderno.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. TESTPGM1.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
```

```

DATA DIVISION.
WORKING-STORAGE SECTION.

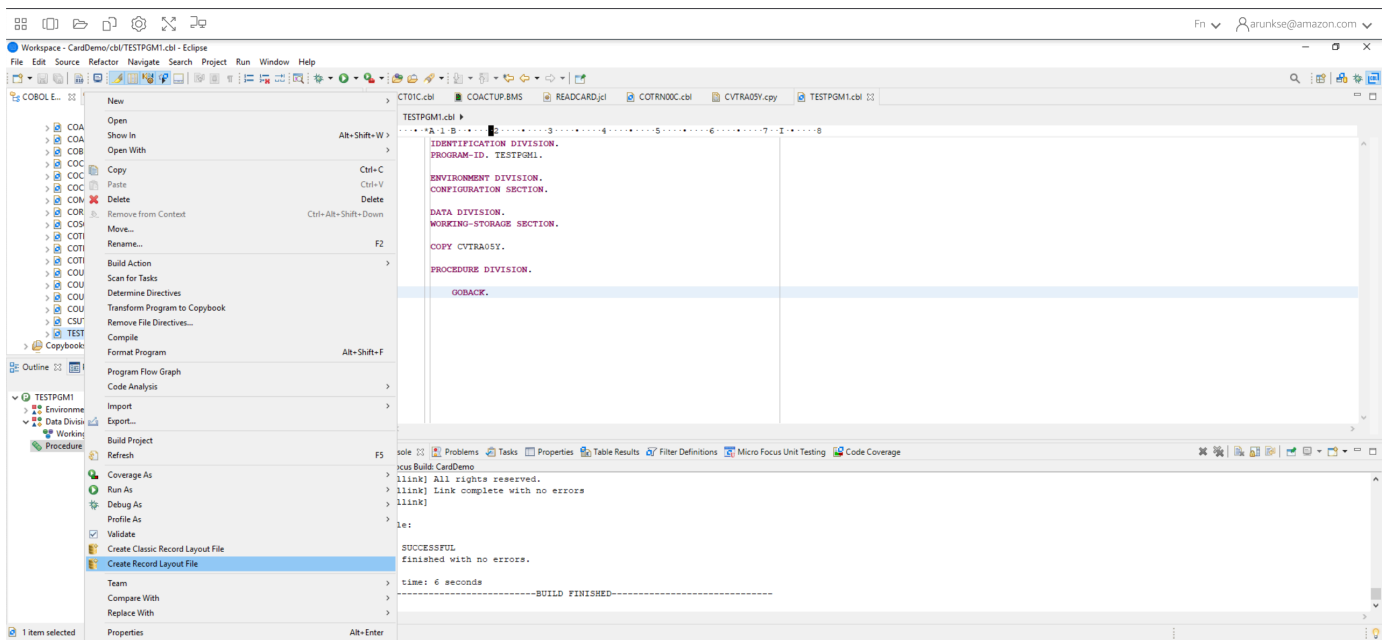
COPY CVTRA05Y.

PROCEDURE DIVISION.

GOBACK.

```

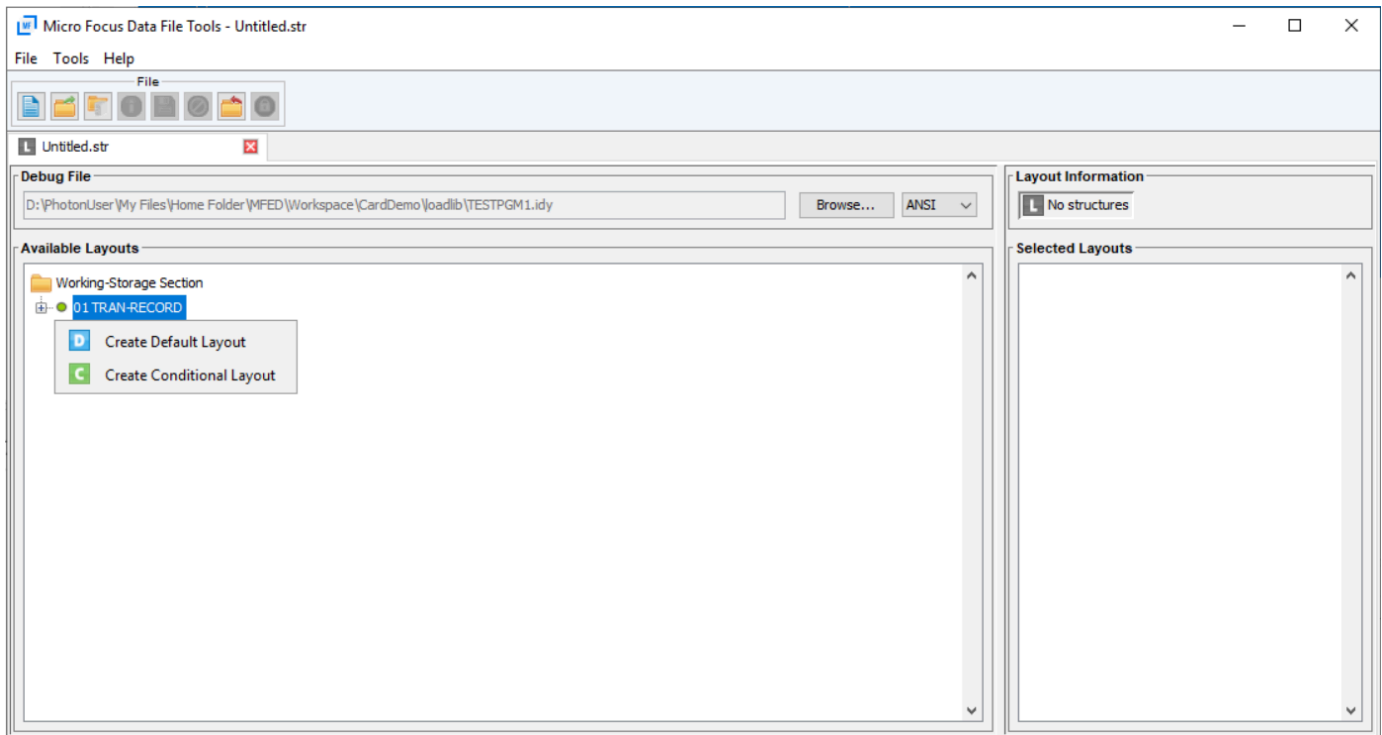
- Después de una compilación correcta, haga clic con el botón derecho en el programa y elija Crear archivo de diseño de registro. Se abrirán las herramientas de archivos de datos de Micro Focus que utilizan el archivo .idy generado durante la compilación.



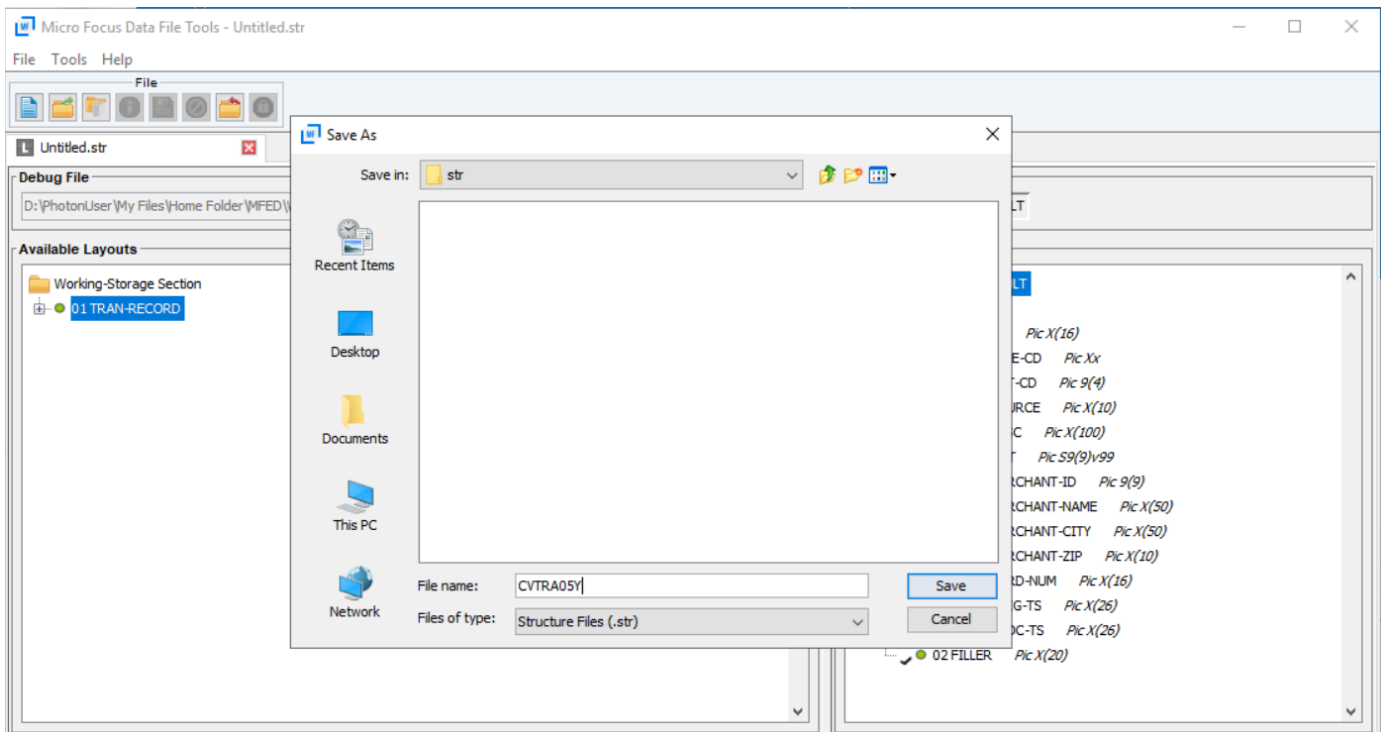
- Haga clic con el botón derecho en la estructura del registro y elija Crear diseño predeterminado (estructura única) o Crear diseño condicional (estructura múltiple) según el diseño.

Para obtener más información, consulte [Creating Structure Files and Layouts](#) en la documentación de Micro Focus.





- Tras crear el diseño, seleccione Archivo en el menú y, a continuación, elija Guardar como. Busque y guarde el archivo en su carpeta principal con el mismo nombre de archivo que su cuaderno. Puede optar por crear una carpeta llamada `str` y guardar allí todos sus archivos de estructura.



## Paso 4: Crear una vista de base de datos con el archivo de estructura (STR)

En este paso, utilizará el archivo de estructura creado anteriormente para crear una vista de base de datos para un conjunto de datos.

- Utilice el comando `dbfhview` para crear una vista de base de datos para un conjunto de datos que ya se encuentra en el almacén de datos de Micro Focus, como se muestra en el siguiente ejemplo.

```
##
## 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
```

## Paso 5: Ver los conjuntos de datos de Micro Focus como tablas y columnas

En este paso, se conectará a la base de datos utilizando pgAdmin para poder ejecutar consultas y ver los conjuntos de datos, como tablas y columnas.

- Conéctese a la base de datos `MicroFocus$SEE$Files$VSAM` mediante pgAdmin y consulte la vista de base de datos que ha creado en el paso 4.

```
SELECT * FROM public."V_AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT";
```

The screenshot shows the pgAdmin 4 interface. The query editor contains the SQL statement: `SELECT * FROM public."V_AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT";`. The results pane displays a table with 19 rows and 13 columns. The columns are: `tran_id`, `tran_type_cd`, `tran_cat_cd`, `tran_source`, `tran_desc`, `tran_amt`, `tran_merchant_id`, `tran_merchant_name`, `tran_merchant_city`, `tran_merchant_zip`, `tran_card_num`, and `tran_orig_ts`. The data represents various transactions, including purchases and return items from different merchants.

tran_id	tran_type_cd	tran_cat_cd	tran_source	tran_desc	tran_amt	tran_merchant_id	tran_merchant_name	tran_merchant_city	tran_merchant_zip	tran_card_num	tran_orig_ts	
1	000000000683580	01	0001	POS TERM	Purchase at Abshire-Lowe	0000005	800000000	Abshire-Lowe	North Enoshaven	72112	485945261287...	2022-06-10
2	0000000001774260	03	0001	OPERATOR	Return item at Nitzsche, Nic...	0000009	800000000	Nitzsche, Nicolas an...	Fidelshire	53378	092798710863...	2022-06-10
3	0000000006292564	01	0001	POS TERM	Purchase at Emser, Roob an...	0000000	800000000	Emser, Roob and Gle...	North Makenziemo...	78487-7965	600961915067...	2022-06-10
4	0000000009101861	01	0001	POS TERM	Purchase at Guann LLC	0000002	800000000	Guann LLC	South Lynn	51508-9166	804058041034...	2022-06-10
5	0000000010142252	01	0001	POS TERM	Purchase at Kertzmann-Scho...	0000004	800000000	Kertzmann-Schoen	East Eulahstad	98754-1089	565683050498...	2022-06-10
6	0000000010229018	01	0001	POS TERM	Purchase at Giasason-Medhu...	0000008	800000000	Giasason-Medhurst	Colleenburgh	23712-2080	737933563466...	2022-06-10
7	0000000016259484	03	0001	OPERATOR	Return item at Sipes Inc	0000000	800000000	Sipes Inc	Emilioside	93329	401150089177...	2022-06-10
8	0000000017874199	01	0001	POS TERM	Purchase at Legros Group	0000003	800000000	Legros Group	Carmeloborough	34849-5127	804058041034...	2022-06-10
9	0000000019065428	03	0001	OPERATOR	Return item at Turcotte Group	0000005	800000000	Turcotte Group	Andrewfurt	41346-3789	590353518179...	2022-06-10
10	00000000021711604	01	0001	POS TERM	Purchase at Gleason, Shana...	0000004	800000000	Gleason, Shanahan a...	Myrticeport	21768-0823	590173372142...	2022-06-10
11	0000000025430891	01	0001	POS TERM	Purchase at Beatty-Hessel	0000000	800000000	Beatty-Hessel	Simonisport	52595	326676361233...	2022-06-10
12	0000000028097268	01	0001	POS TERM	Purchase at Wolf, Cruicksha...	0000002	800000000	Wolf, Cruickshank an...	Fritchester	20195-5156	709414275105...	2022-06-10
13	00000000302975266	01	0001	POS TERM	Purchase at Rathe LLC	0000008	800000000	Rathe LLC	Brendenfort	35302-6495	376628198415...	2022-06-10
14	0000000032979555	01	0001	POS TERM	Purchase at Treutel-Leffler	0000000	800000000	Treutel-Leffler	New Niciolette	65014-0045	650923036255...	2022-06-10
15	0000000033681127	01	0001	POS TERM	Purchase at Schinner-Steuber	0000009	800000000	Schinner-Steuber	Schmittchester	50777-5535	376628198415...	2022-06-10
16	0000000040455859	01	0001	POS TERM	Purchase at Brekke, Bradtie ...	0000007	800000000	Brekke, Bradtie and ...	Vuummouth	18481-5013	114216769287...	2022-06-10
17	0000000043636099	03	0001	OPERATOR	Return item at Nader-Bayer	0000009	800000000	Nader-Bayer	Goyetteville	35324	294013936230...	2022-06-10
18	0000000051205286	01	0001	POS TERM	Purchase at Goodwin, Von a...	0000006	800000000	Goodwin, Von and Kr...	Ericmouth	03874	709414275105...	2022-06-10
19	0000000054788966	01	0001	POS TERM	Purchase at Cramin and Sons	0000005	800000000	Cramin and Sons	Bartonside	06677	453478410771...	2022-06-10

## Tutorial: Uso de plantillas con Micro Focus Enterprise Developer

Este tutorial describe cómo utilizar plantillas y proyectos predefinidos con Micro Focus Enterprise Developer. Incluye tres casos de uso. Todos los casos de uso utilizan el código de ejemplo que se proporciona en el BankDemo ejemplo. Para descargar el ejemplo, elija [bankdemo.zip](#).

### ⚠ Important

Si utiliza la versión de Enterprise Developer para Windows, los binarios generados por el compilador solo se pueden ejecutar en el Enterprise Server suministrado con Enterprise Developer. No puede ejecutarlos en el entorno de ejecución de modernización del AWS mainframe, que está basado en Linux.

### Temas

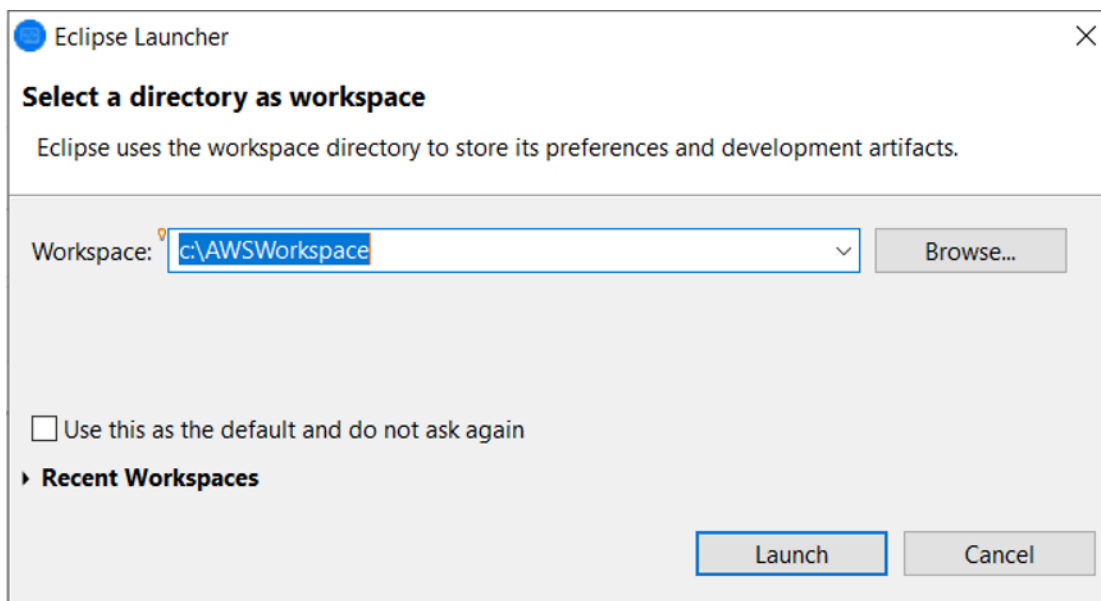
- [Caso de uso 1: Uso de la plantilla de proyecto de COBOL que contiene los componentes de origen](#)
- [Caso de uso 2: Uso de la plantilla de proyecto de COBOL sin los componentes de origen](#)
- [Caso de uso 3: Uso del proyecto de COBOL predefinido que enlaza con las carpetas de origen](#)

- [Uso de la plantilla JSON de definición de región](#)

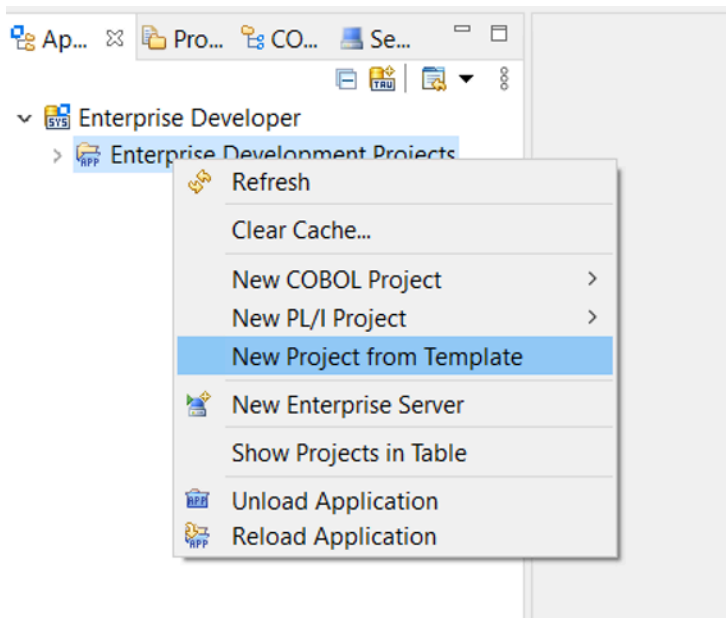
## Caso de uso 1: Uso de la plantilla de proyecto de COBOL que contiene los componentes de origen

Este caso de uso requiere que copie los componentes de origen en la estructura de directorios de la plantilla como parte de los pasos de preconfiguración de la demostración. En el [bankdemo.zip](#), se ha cambiado con respecto al del `AWSTemplates.zip` original para evitar tener dos copias del origen.

1. Inicie Enterprise Developer y especifique el espacio de trabajo elegido.



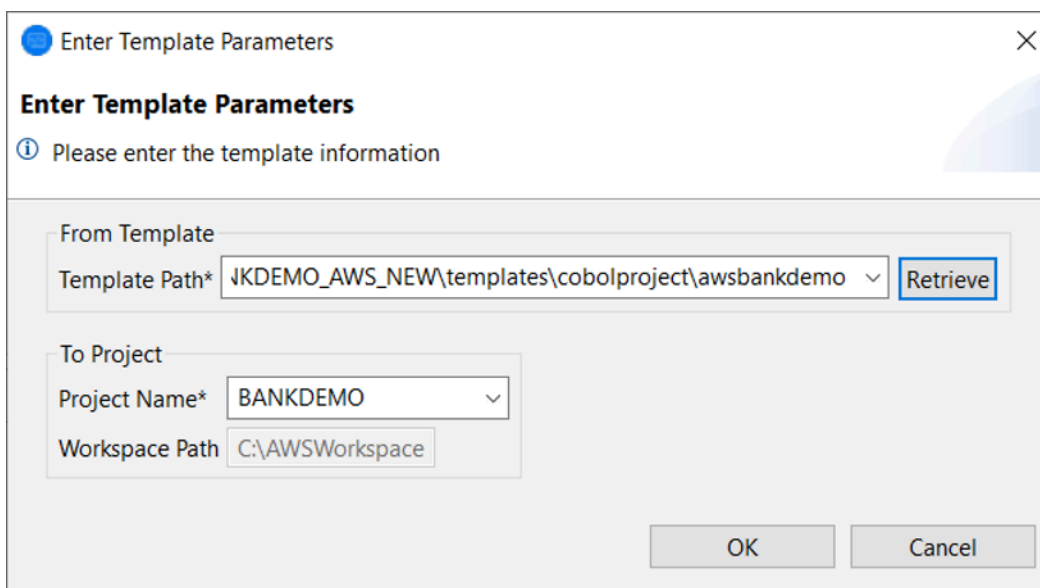
2. En la vista del explorador de aplicaciones, en el elemento de vista en árbol Enterprise Development Project, seleccione New Project from Template en el menú contextual.



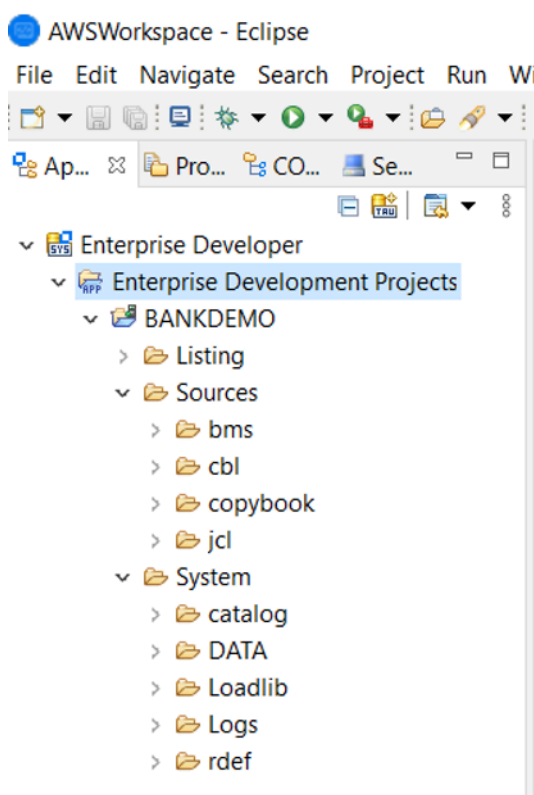
3. Introduzca los parámetros de la plantilla como se muestra.

**Note**

La ruta de la plantilla hará referencia al lugar donde se haya extraído el ZIP.



4. Si selecciona OK, se creará un proyecto de Eclipse de desarrollo local basado en la plantilla proporcionada, con una estructura completa de origen y entorno de ejecución.



La estructura de `System` contiene un archivo completo de definición de recursos con las entradas requeridas para `BANKDEMO`, el catálogo requerido con las entradas añadidas y los archivos de datos ASCII correspondientes.

Como la estructura de la plantilla de origen contiene todos los elementos de origen, estos archivos se copian en el proyecto local y, por lo tanto, se incorporan automáticamente a Enterprise Developer.

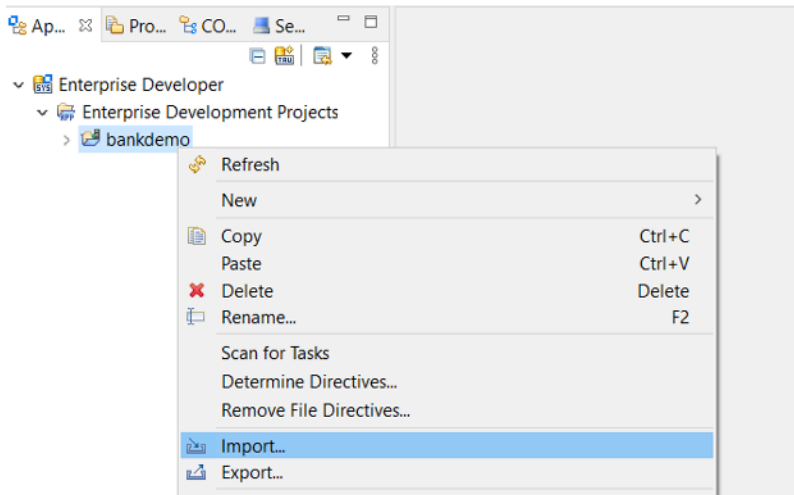
## Caso de uso 2: Uso de la plantilla de proyecto de COBOL sin los componentes de origen

Los pasos 1 a 3 son idénticos al [Caso de uso 1: Uso de la plantilla de proyecto de COBOL que contiene los componentes de origen](#).

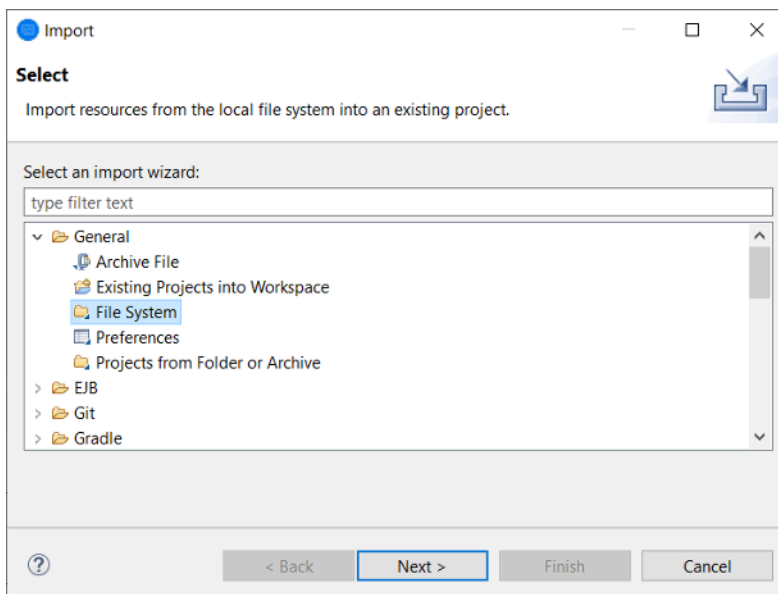
La `System` estructura de este caso de uso también contiene un archivo completo de definición de recursos con las entradas necesarias BankDemo, el catálogo necesario con las entradas añadidas y los archivos de datos ASCII correspondientes.

Sin embargo, la estructura de origen de la plantilla no contiene ningún componente. Debe importarlos al proyecto desde el repositorio de origen que utilice.

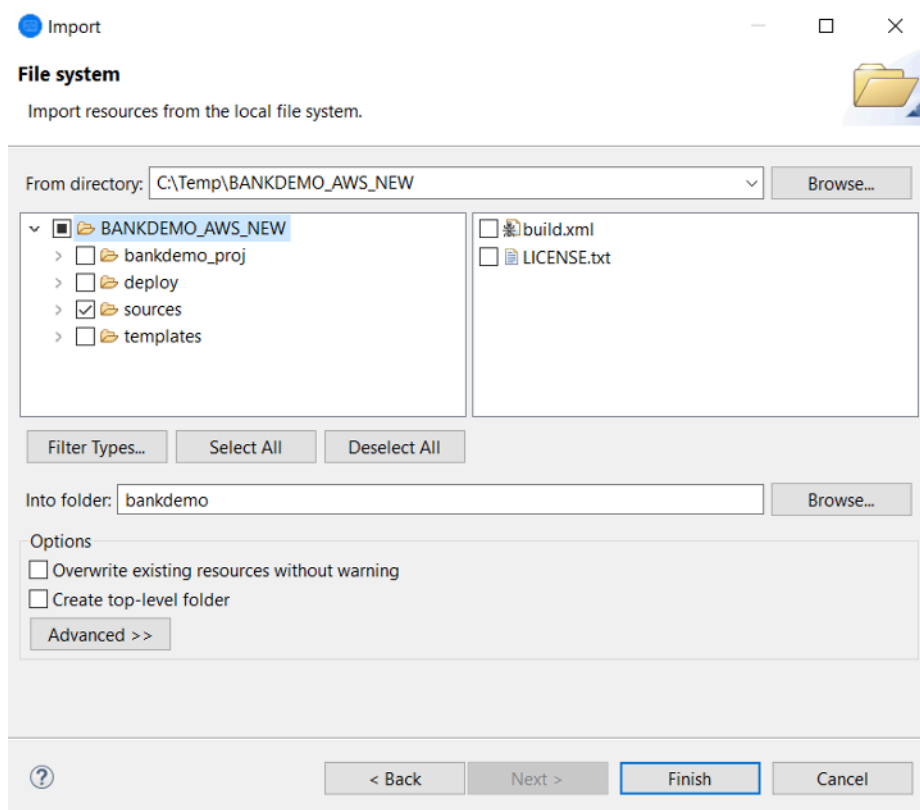
1. Elija el nombre del proyecto. En el menú contextual relacionado, seleccione Import.



2. En el cuadro de diálogo resultante, en la sección General, elija File System y, a continuación, elija Next.



3. Rellene el campo From directory navegando por el sistema de archivos hasta que señale a la carpeta del repositorio. Seleccione todas las carpetas que desee importar, por ejemplo sources. El campo Into folder se rellenará automáticamente. Elija Finish.

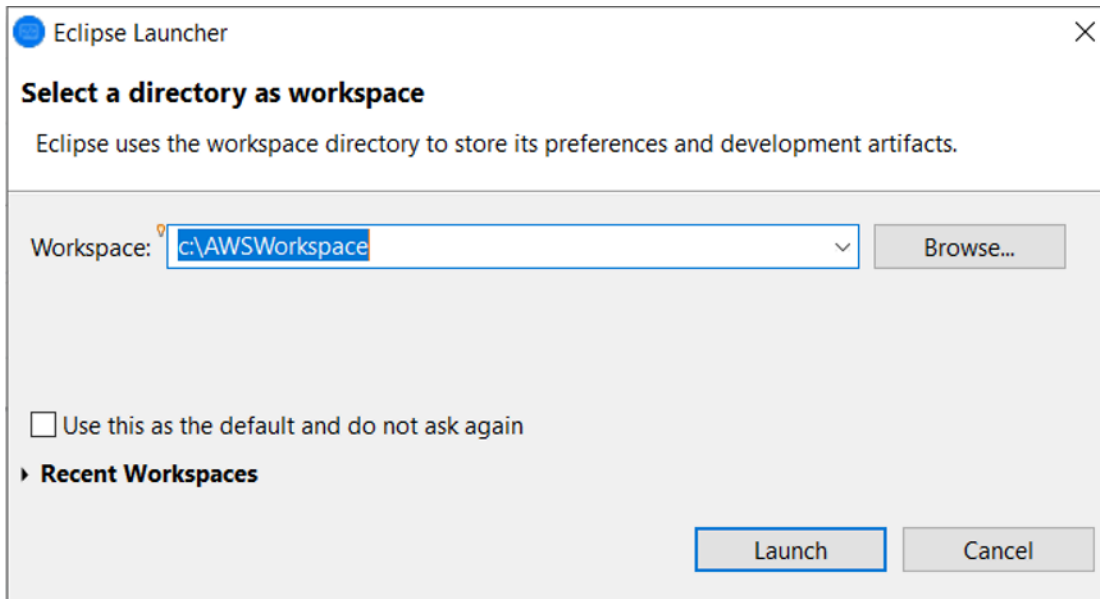


Una vez que la estructura de la plantilla de origen contiene todos los elementos de origen, estos se crean automáticamente en Enterprise Developer.

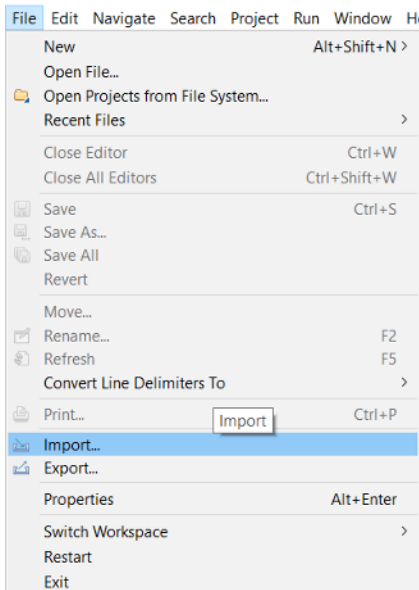
## Caso de uso 3: Uso del proyecto de COBOL predefinido que enlaza con las carpetas de origen

1. Inicie Enterprise Developer y especifique el espacio de trabajo elegido.

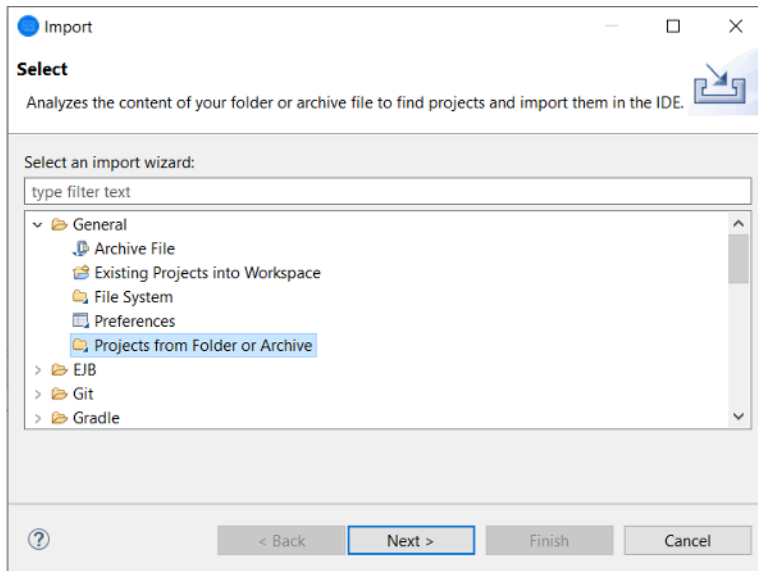




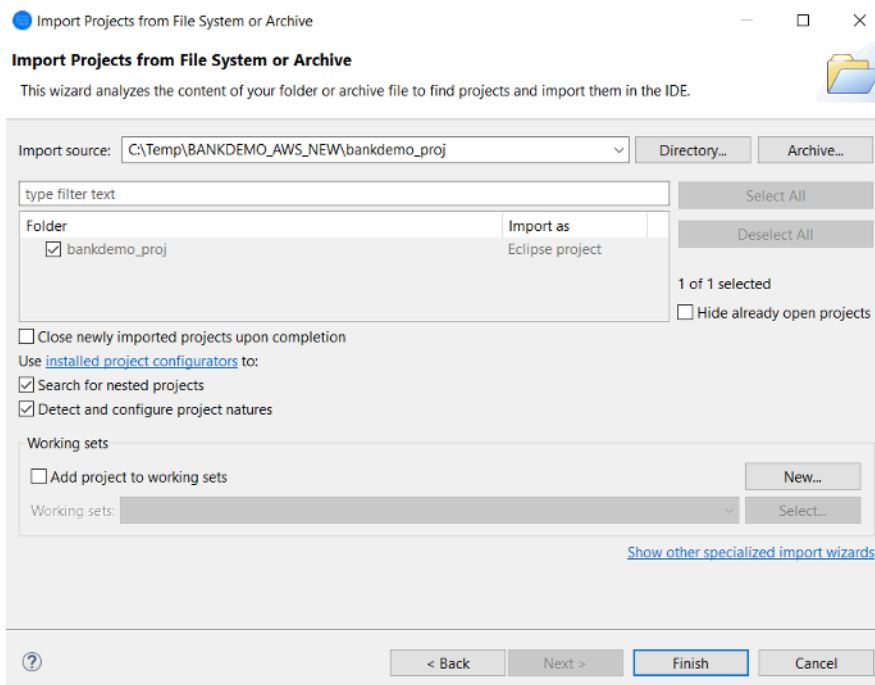
## 2. En el menú File, elija Import.



## 3. En el cuadro de diálogo resultante, en General, seleccione Projects from Folder or Archive y pulse Next.



- Introduzca el Import source, elija el Directory y navegue por el sistema de archivos para seleccionar la carpeta de proyecto predefinida. El proyecto que contiene tiene enlaces a las carpetas de origen del mismo repositorio.

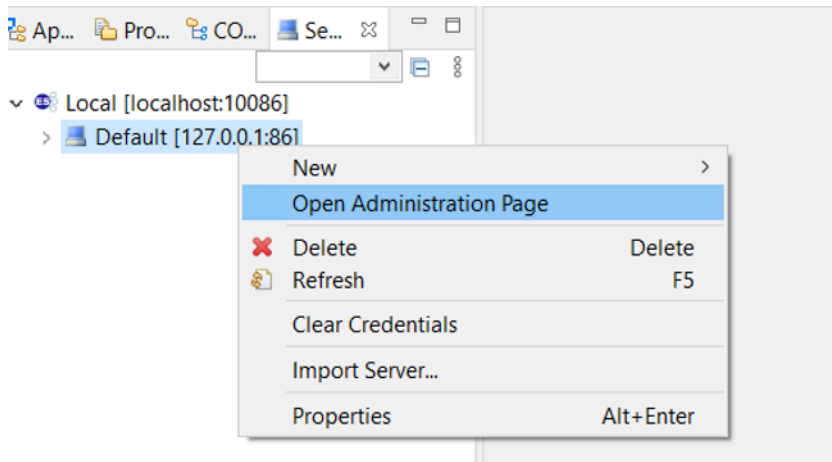


Seleccione Finalizar.

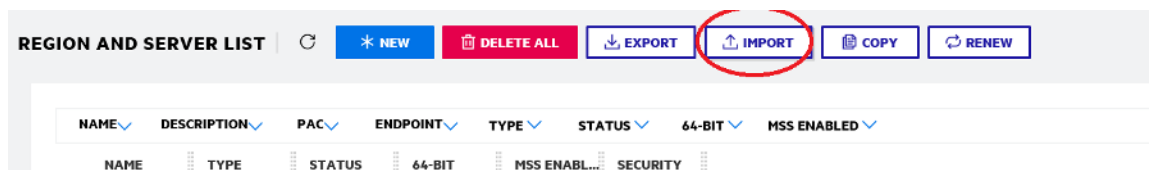
Como el proyecto se rellena con los enlaces a la carpeta de origen, el código se genera automáticamente.

## Uso de la plantilla JSON de definición de región

1. Cambie a la vista del explorador de servidores. En el menú contextual correspondiente, seleccione Open Administration Page, que iniciará el navegador predeterminado.



2. En la pantalla resultante de Enterprise Server Common Web Administration (ESCWA), seleccione Import.



3. Seleccione el tipo de importación JSON y, después, Next.

### CHOOSE IMPORT TYPE



#### JSON

Import a .json file by selecting a file on the host where the client browser is running.



#### XML

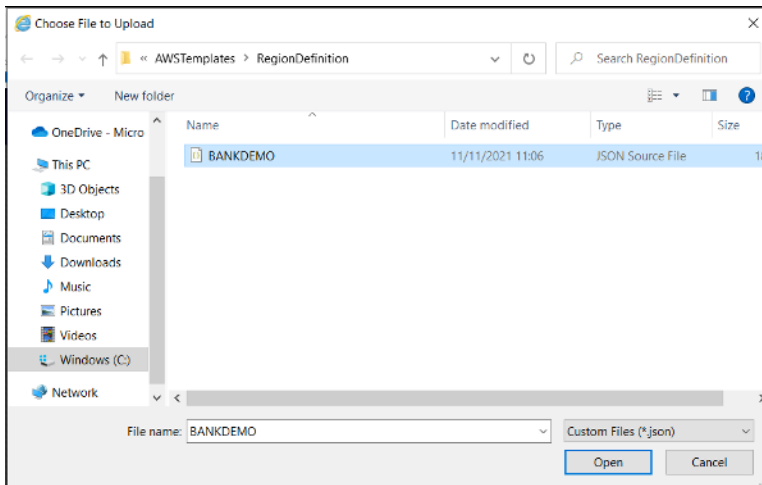
Import a .xml file by selecting a file on the host where the client browser is running.



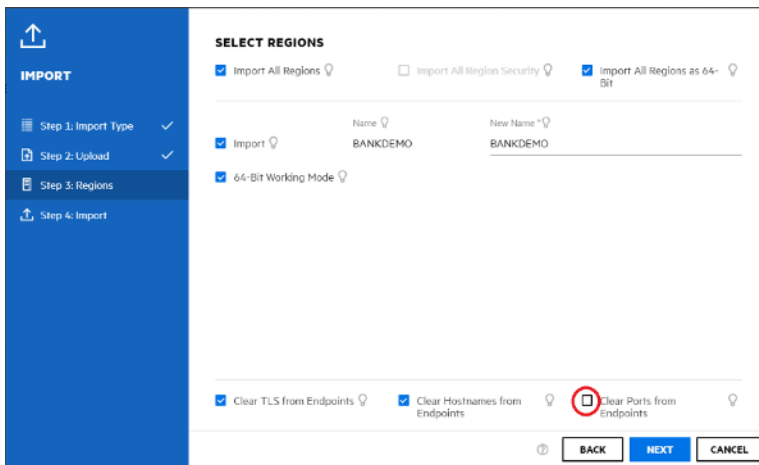
#### Legacy

Import a legacy repository (directory of .dat files) by selecting the directory location on the host where the Directory Server is running.

4. Cargue el archivo BANKDEMO.JSON suministrado.



Una vez seleccionado, elija Next.



En el panel Select Regions, asegúrese de que la opción Clear Ports from Endpoints no esté seleccionada y, a continuación, siga seleccionando Next en los paneles hasta que aparezca el panel Perform Import. A continuación, elija Import.



Por último, haga clic en Finish. La región BANKDEMO se añadirá a la lista de servidores.

**REGION AND SERVER LIST** ↻ \* NEW 🗑️ DELETE ALL 📄 EXPORT 📄 IMPORT 📄 COPY 🔄 RENEW

NAME	DESCRIPTION	PAC	ENDPOINT	TYPE	STATUS	64-BIT	MSS ENABLED	SECURITY
BANKDEMO	Region				Stopped		✓	Default
ESDEMO	Region				Stopped			Default
ESDEMO64	Region				Stopped	✓		Default

- Navegue hasta las General Properties de la región BANKDEMO.
- Desplácese a la sección Configuration.
- La variable de entorno ESP debe configurarse en la carpeta System correspondiente al proyecto de Eclipse creado en los pasos anteriores. El valor debería ser `workspacefolder/projectname/System`.

ADDITIONAL

```

Configuration Information ⓘ
[ES-Environment]
ESP={Enter Project System Folder Here}
MF_CHARSET=A
EXTFH=$ESP/EXTFH.cfg

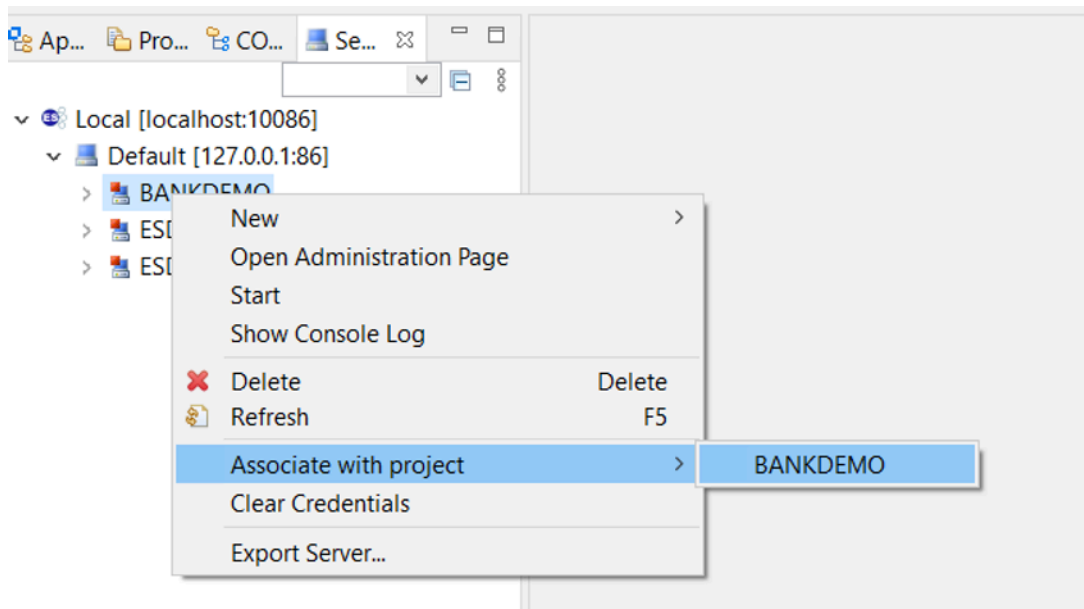
```

- Haga clic en Apply.

**APPLY**

La región ahora está completamente configurada para ejecutarse junto con el proyecto COBOL de Eclipse.

- Por último, vuelva a Enterprise Developer y asocie la región importada al proyecto.



El entorno Enterprise Developer ya está listo para usarse, con una versión funcional completa de BankDemo. Puede editar, compilar y depurar el código de la región.

#### Important

Si utiliza la versión de Enterprise Developer para Windows, los binarios generados por el compilador solo se pueden ejecutar en el Enterprise Server suministrado con Enterprise Developer. No se pueden ejecutar en el entorno de ejecución AWS Mainframe Modernization, que está basado en Linux.

## Tutorial: Configuración de la versión de Micro Focus para la aplicación BankDemo de muestra

AWS La modernización del mainframe le permite configurar compilaciones y canalizaciones de integración y entrega continuas (CI/CD) para sus aplicaciones migradas. Estas compilaciones y canalizaciones utilizan AWS CodeBuild y proporcionan estas capacidades. AWS CodeCommit AWS CodePipeline CodeBuild es un servicio de compilación totalmente gestionado que compila el código fuente, ejecuta pruebas unitarias y produce artefactos listos para su despliegue. CodeCommit es un servicio de control de versiones que te permite almacenar y gestionar de forma privada los repositorios de Git en la nube. AWS CodePipeline es un servicio de entrega continua que te permite modelar, visualizar y automatizar los pasos necesarios para lanzar tu software.

En este tutorial se muestra cómo AWS CodeBuild compilar el código fuente de la aplicación de BankDemo muestra desde Amazon S3 y, a continuación, exportar el código compilado a Amazon S3.

AWS CodeBuild es un servicio de integración continua totalmente gestionado que compila el código fuente, ejecuta pruebas y produce paquetes de software listos para su implementación. Con CodeBuild, puede usar entornos de compilación preempaquetados o puede crear entornos de compilación personalizados que usen sus propias herramientas de compilación. En esta situación de demostración se utiliza la segunda opción. Consiste en un entorno de CodeBuild compilación que utiliza una imagen de Docker preempaquetada.

### Important

Antes de iniciar su proyecto de modernización de unidad central, le recomendamos que conozca el [Programa de aceleración de la migración de AWS](#) o que se ponga en contacto con [especialistas en unidades centrales de AWS](#) para obtener información sobre los pasos necesarios para modernizar una aplicación de unidad central.

## Temas

- [Requisitos previos](#)
- [Paso 1: Crear un bucket de Amazon S3](#)
- [Paso 2: Crear el archivo de especificaciones de compilación](#)
- [Paso 3: Cargar los archivos de origen](#)
- [Paso 4: Crear políticas de IAM](#)
- [Paso 5: Crear un rol de IAM](#)
- [Paso 6: Asociar las políticas de IAM al rol de IAM.](#)
- [Paso 7: Crea el CodeBuild proyecto](#)
- [Paso 8: Iniciar la compilación](#)
- [Paso 9: Descargar los artefactos de salida](#)
- [Eliminar recursos](#)

## Requisitos previos

Antes de comenzar este tutorial, complete los siguientes requisitos previos:

- Descarga la [aplicación de BankDemo muestra](#) y descomprímela en una carpeta. La carpeta de origen contiene programas COBOL y cuadernos, así como definiciones de BMS del CICS. También contiene una carpeta JCL como referencia, aunque no es necesario compilar JCL. La carpeta también contiene los metaarchivos necesarios para la compilación.
- En la consola de modernización AWS del mainframe, elija Herramientas. En Análisis, desarrollo y compilación de activos, seleccione Compartir activos con mi cuenta de AWS.

## Paso 1: Crear un bucket de Amazon S3

En este paso, creará dos buckets de Amazon S3. El primero es un bucket de entrada para almacenar el código fuente y el otro es un depósito de salida para almacenar la salida de la compilación. Para obtener más información, consulte [Creación, configuración y trabajo con buckets de Amazon S3](#) en la Guía del usuario de Amazon S3.

1. Para crear el bucket de entrada, inicie sesión en la consola de Amazon S3 y elija Crear bucket.
2. En la configuración general, proporcione un nombre para el depósito y especifique el Región de AWS lugar en el que desea crearlo. Un ejemplo Región de AWS de nombre `regionId` es `codebuild-regionId-accountId-input-bucket` dónde está el depósito y `accountId` es su Cuenta de AWS ID.

### Note

Si va a crear el depósito en un lugar diferente al Región de AWS de EE. UU. Este (Virginia del Norte), especifique el `LocationConstraint` parámetro. Para obtener más información, consulte [CreateBucket](#) en Amazon Simple Storage Service API Referenc.

3. Conserve todos los demás valores y elija Crear bucket.
4. Repita los pasos 1 a 3 para crear el bucket de salida. Un ejemplo de nombre es `codebuild-regionId-accountId-output-bucket` dónde `regionId` está Región de AWS el depósito y `accountId` es su Cuenta de AWS ID.

Utilice los mismos nombres que elija para los buckets en todo este tutorial.

## Paso 2: Crear el archivo de especificaciones de compilación

En este paso, creará un archivo de especificaciones de compilación. Este archivo proporciona los comandos de compilación y la configuración relacionada, en formato YAML, CodeBuild para ejecutar



la compilación. Para obtener más información, consulta la [referencia sobre las especificaciones de compilación CodeBuild](#) en la Guía del AWS CodeBuild usuario.

1. Cree un archivo con el nombre `buildspec.yml` en el directorio que haya utilizado para descomprimir como se indicaba en los requisitos previos.
2. Añada el siguiente contenido al archivo y guarde. No se requieren cambios en este archivo.

```
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/**
```

Aquí `CODEBUILD_BUILD_ID`, `CODEBUILD_BUILD_ARN`, `$CODEBUILD_SRC_DIR/source`, y `$CODEBUILD_SRC_DIR/target` están las variables de entorno disponibles en él CodeBuild. Para obtener más información, consulte [Variables de entorno en los entornos de compilación](#).

En este punto, la estructura de directorios debería ser similar a la siguiente.

```
(root directory name)
|-- build.xml
```

```
|-- buildspec.yml
|-- LICENSE.txt
|-- source
   |... etc.
```

3. Comprima el contenido de la carpeta en un archivo con el nombre `BankDemo.zip`. En este tutorial, no puede comprimir la carpeta. Así es que, comprima el contenido de la carpeta en un archivo con el nombre `BankDemo.zip`.

## Paso 3: Cargar los archivos de origen

En este paso, debe cargar el código fuente de la aplicación de BankDemo muestra en su bucket de entrada de Amazon S3.

1. En la consola de Amazon SNS, en el panel de navegación izquierdo, seleccione Buckets. A continuación, elija el bucket de entrada que ha creado anteriormente.
2. En Objetos, selecciona Cargar.
3. En la sección Archivos y carpetas, elija Añadir archivos.
4. Vaya al archivo `BankDemo.zip` y elíjalo.
5. Seleccione Cargar.

## Paso 4: Crear políticas de IAM

En este paso, creará dos [políticas de IAM](#). Una política concede permisos para que AWS Mainframe Modernization pueda acceder y utilizar la imagen de Docker que contiene las herramientas de compilación de Micro Focus. Esta política no está personalizada para los clientes. La otra política otorga permisos para que AWS Mainframe Modernization interactúe con los buckets de entrada y salida y con los [CloudWatch registros de Amazon que se CodeBuild generan](#).

Para obtener más información sobre la creación de una política de IAM, consulte [Crear políticas de IAM](#) en la Guía del usuario de IAM.

Para crear una política de acceso a las imágenes de Docker

1. En la consola de IAM, copie el siguiente documento de política y péguelo en el editor de políticas.

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage"
    ],
    "Resource": "arn:aws:ecr:*:673918848628:repository/m2-enterprise-build-
tools"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::aws-m2-repo-*/*"
  }
]
}

```

2. Escriba un nombre para la política, por ejemplo, m2CodeBuildPolicy.

Crear una política que permita a la modernización del AWS mainframe interactuar con los depósitos y los registros

1. En la consola de IAM, copie el siguiente documento de política y péguelo en el editor de políticas. Asegúrese de actualizar `regionId` a Región de AWS, y `accountId` a su. Cuenta de AWS

```

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

```

```

        "Action": [
            "logs:CreateLogGroup",
            "logs:CreateLogStream",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/
codebuild-bankdemo-project",
            "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/
codebuild-bankdemo-project:*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetBucketAcl",
            "s3:GetBucketLocation",
            "s3:List*"
        ],
        "Resource": [
            "arn:aws:s3:::codebuild-regionId-accountId-input-bucket",
            "arn:aws:s3:::codebuild-regionId-accountId-input-bucket/*",
            "arn:aws:s3:::codebuild-regionId-accountId-output-bucket",
            "arn:aws:s3:::codebuild-regionId-accountId-output-bucket/*"
        ],
        "Effect": "Allow"
    }
]
}

```

2. Escriba un nombre para la política, por ejemplo, BankdemoCodeBuildRolePolicy.

## Paso 5: Crear un rol de IAM

En este paso, creará un nuevo [rol de IAM](#) que le CodeBuild permita interactuar con AWS los recursos, después de asociar las políticas de IAM que creó anteriormente a este nuevo rol de IAM.

Para obtener información sobre la creación de un rol de servicio, consulte [Creación de un rol para delegar permisos a un AWS servicio](#) en la Guía del usuario de IAM,.

1. Inicie sesión en la consola de IAM; y elija Roles en el panel de navegación de la izquierda.
2. Elija Crear rol.
3. En Tipo de entidad de confianza, seleccione Servicio de AWS.
4. En Casos de uso de otros servicios de AWS, elija y CodeBuild, a continuación, CodeBuildvuelva a elegir.
5. Elija Siguiente.
6. Elija Siguiente en la página Agregar permisos. Posteriormente, asignará una política al rol.
7. En Detalles del rol, proporcione un nombre para el rol, por ejemplo, BankdemoCodeBuildServiceRole.
8. En Seleccionar entidades de confianza, compruebe que el documento de política tenga el siguiente aspecto:

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

9. Elija Crear rol.

## Paso 6: Asociar las políticas de IAM al rol de IAM.

En este paso, asociará las políticas de IAM que ha creado antes al rol de IAM BankdemoCodeBuildServiceRole.

1. Inicie sesión en la consola de IAM; y elija Roles en el panel de navegación de la izquierda.
2. En Roles, elija el rol que ha creado anteriormente, por ejemplo, BankdemoCodeBuildServiceRole.
3. En la sección Políticas de permisos, elija Agregar permisos y, después, Asociar políticas.

4. En Otras políticas de permisos, elija las políticas que ha creado anteriormente, por ejemplo, `m2CodeBuildPolicy` y `BankdemoCodeBuildRolePolicy`.
5. Seleccione Asociar políticas.

## Paso 7: Crea el CodeBuild proyecto

En este paso, crearás el CodeBuild proyecto.

1. Inicie sesión en la CodeBuild consola y elija Crear proyecto de compilación.
2. En la sección Configuración del proyecto, proporciona un nombre para el proyecto, por ejemplo, `codebuild-bankdemo-project`.
3. En la sección Origen, en Proveedor del origen, elija Amazon S3 y, a continuación, elija el bucket de entrada que ha creado anteriormente, por ejemplo, `codebuild-regionId-accountId-input-bucket`.
4. En el campo Clave de objeto de S3 o carpeta de S3, introduzca el nombre del archivo zip que ha cargado en el bucket de S3. En este caso, el nombre de archivo es `bankdemo.zip`.
5. En la sección Entorno, seleccione Imagen personalizada.
6. En el campo Tipo de entorno, seleccione Linux.
7. En Registro de imágenes, seleccione Otro registro.
8. En el campo URL de registro externo,
  - Para Micro Focus v9: Introduzca `673918848628.dkr.ecr.us-west-1.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1`. Si utiliza una AWS región diferente con Micro Focus v9, también puede especificar `673918848628.dkr.ecr.<m2-region>.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1` dónde `<m2-region>` hay una AWS región en la que esté disponible el servicio de modernización de AWS mainframe (por ejemplo, `eu-west-3`).
  - Para Micro Focus v8: Introduzca `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:8.0.9.R1`
  - Para Micro Focus v7: Introduzca `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:7.0.R10`
9. En Rol de servicio, elija Rol de servicio existente y, en el campo ARN de rol, elija el rol de servicio que ha creado anteriormente; por ejemplo, `BankdemoCodeBuildServiceRole`.
10. En Especificaciones de la compilación, elija Utilizar un archivo de especificación de compilación.

11. En la sección Artefactos, en Tipo, elija Amazon S3 y, a continuación, elija su bucket de salida, por ejemplo, `codebuild-regionId-accountId-output-bucket`.
12. En el campo Nombre, introduzca el nombre de una carpeta del bucket en la que desee que se incluyan los artefactos de salida de la compilación, por ejemplo, `bankdemo-output.zip`.
13. En Empaquetado de artefactos, elija Zip.
14. Elija Crear el proyecto de compilación.

## Paso 8: Iniciar la compilación

En este paso, iniciará la compilación.

1. Inicie sesión en la CodeBuild consola.
2. En el panel de navegación, elija Proyectos de compilación.
3. Elija el proyecto de compilación que ha creado antes, por ejemplo, `codebuild-bankdemo-project`.
4. Seleccione Iniciar la compilación.

Este comando inicia la compilación. La compilación se ejecuta de forma asincrónica. La salida del comando es un JSON que incluye el ID del atributo. Este identificador de atributo es una referencia al identificador de CodeBuild compilación de la compilación que acabas de iniciar. Puedes ver el estado de la compilación en la CodeBuild consola. También puede ver los registros detallados sobre la ejecución de la compilación en la consola. Para obtener más información, consulte [View detailed build information](#) en AWS CodeBuild User Guide.

Cuando se complete la fase actual, significa que la compilación ha finalizado correctamente y que los artefactos compilados están listos en Amazon S3.

## Paso 9: Descargar los artefactos de salida

En este paso, descargará los artefactos de salida de Amazon S3. La herramienta de compilación de Micro Focus puede crear varios tipos de ejecutables diferentes. En este tutorial, generará objetos compartidos.

1. Inicie sesión en la consola de Amazon S3.
2. En la sección Buckets **>**, elige el nombre del depósito de salida, por ejemplo, `codebuild-regionId-accountId-output-bucket`

3. Seleccione **Descargar**.
4. Descomprima el archivo descargado. Navegue hasta la carpeta de destino para ver los artefactos de la compilación. Estos incluyen los objetos compartidos de Linux .so.

## Eliminar recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para evitar cargos adicionales. Para ello, complete los siguientes pasos.

- Elimine los depósitos de S3 que ha creado para este tutorial. Para obtener más información, consulte [Eliminar un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.
- Elija las políticas que ha creado para este tutorial. Para obtener más información, consulte [Eliminación de políticas de IAM](#) en la Guía del usuario de IAM.
- Elija el rol de IAM que ha creado para este tutorial. Para obtener más información, consulte [Eliminación de roles o perfiles de instancia](#) en la Guía del usuario de IAM.
- Elimine el CodeBuild proyecto que creó para este tutorial. Para obtener más información, consulte [Eliminar un proyecto de compilación CodeBuild en](#) la Guía del AWS CodeBuild usuario.

## Tutorial: Configuración de una canalización de CI/CD para su uso con Micro Focus Enterprise Developer

En este tutorial, se muestra cómo importar, editar, compilar y ejecutar la aplicación de BankDemo ejemplo en Micro Focus Enterprise Developer y, a continuación, cómo confirmar los cambios para activar una canalización de CI/CD.

### Contenido

- [Requisitos previos](#)
- [Creación de una infraestructura básica de canalización de CI/CD](#)
- [Cree un AWS CodeCommit repositorio y una canalización de CI/CD](#)
  - [Ejemplo de archivo de activación de YAML: config\\_git.yml](#)
- [Creación de Enterprise Developer 2.0 AppStream](#)
- [Configuración y prueba de Enterprise Developer](#)
  - [Clona el BankDemo CodeCommit repositorio en Enterprise Developer](#)
  - [Cree un proyecto COBOL para BankDemo mainframe y cree una aplicación](#)



- [Cree un BankDemo CICS local y un entorno por lotes para las pruebas](#)
- [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
- [Inicie el terminal Rumba 3270](#)
- [Ejecute una BankDemo transacción](#)
- [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)
- [Ejercicio 1: Mejorar el cálculo de los préstamos en la aplicación de BANKDEMO](#)
  - [Agregue la regla de análisis de préstamos a Enterprise Developer Code Analysis](#)
  - [Paso 1: Realizar un análisis de código para calcular el préstamo](#)
  - [Paso 2: Modificar el mapa CICS BMS y el programa COBOL y pruébelo](#)
  - [Paso 3: Agregar el cálculo del importe total en el programa COBOL](#)
  - [Paso 4: Confirmar los cambios y ejecutar la canalización de CI/CD](#)
- [Ejercicio 2: Extraiga el cálculo del préstamo en BankDemo la solicitud](#)
  - [Paso 1: Refactorizar la rutina de cálculo de préstamos en una sección de COBOL](#)
  - [Paso 2: Extraer la rutina de cálculo de préstamos a un programa COBOL independiente](#)
  - [Paso 3: Confirmar los cambios y ejecutar la canalización de CI/CD](#)
- [Limpieza de recursos](#)

## Requisitos previos

Descargue los siguientes archivos.

- `basic-infra.yaml`
  - [Descarga desde la región Europa \(Fráncfort\).](#)
  - [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)
- `pipeline.yaml`
  - [Descarga desde la región Europa \(Fráncfort\).](#)
  - [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)
- `m2-code-sync-function.zip`
  - [Descarga desde la región Europa \(Fráncfort\).](#)
  - [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)

- [Descarga desde la región Europa \(Fráncfort\).](#)
- [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)
- BANKDEMO-source.zip
  - [Descarga desde la región Europa \(Fráncfort\).](#)
  - [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)
- BANKDEMO-exercise.zip
  - [Descarga desde la región Europa \(Fráncfort\).](#)
  - [Descarga desde la región Este de EE. UU. \(Norte de Virginia\).](#)

El propósito de cada archivo es el siguiente:

#### basic-infra.yaml

Esta AWS CloudFormation plantilla crea la infraestructura básica necesaria para la canalización de CI/CD: VPC, buckets de Amazon S3, etc.

#### pipeline.yaml

Una función Lambda utiliza esta AWS CloudFormation plantilla para lanzar la pila de canalización. Asegúrese de que la plantilla se encuentre en un bucket de Amazon S3 de acceso público. Añada el enlace a este bucket como valor predeterminado para el parámetro PipelineTemplateURL de la plantilla basic-infra.yaml.

#### m2-code-sync-function.zip

Esta función Lambda crea el CodeCommit repositorio, la estructura de directorios basada en config\_git.yaml, y lanza la pila de canalizaciones mediante pipeline.yaml. Asegúrese de que este archivo zip esté disponible en un bucket de Amazon S3 de acceso público en todos los Regiones de AWS lugares en los que se admita la modernización de AWS mainframe. Le recomendamos que almacene el archivo en un depósito en una Región de AWS y lo replique en todos los depósitos. Regiones de AWS Utilice una convención de nomenclatura para el depósito con un sufijo que identifique lo específico Región de AWS (por ejemplo m2-cicd-deployment-source-eu-west-1) y añada el prefijo m2-cicd-deployment-source como valor predeterminado para el parámetro DeploymentSourceBucket y forme el depósito completo mediante la función de AWS CloudFormation sustitución !Sub {DeploymentSourceBucket}-\${AWS::Region} mientras hace referencia a ese depósito en la basic-infra.yaml plantilla como recurso. SourceSyncLambdaFunction

## config\_git.yml

CodeCommit definición de la estructura de directorios. Para obtener más información, consulte [Ejemplo de archivo de activación de YAML: config\\_git.yml](#).

## BANKDEMO-source.zip

BankDemo código fuente y archivo de configuración creados a partir del CodeCommit repositorio.

## BANKDEMO-exercise.zip

BankDemo fuente de los ejercicios tutoriales creados a partir del CodeCommit repositorio.

## Creación de una infraestructura básica de canalización de CI/CD

Utilice la AWS CloudFormation plantilla `basic-infra.yaml` para crear la pila de infraestructura básica de la canalización de CI/CD a través de la AWS CloudFormation consola. Esta pila crea buckets de Amazon S3 en los que puede cargar el código y los datos de su aplicación, y una AWS Lambda función de apoyo para crear otros recursos necesarios, como un AWS CodeCommit repositorio y una AWS CodePipeline canalización.

### Note

Para lanzar esta pila, necesita permisos para administrar IAM, Amazon S3, Lambda AWS CloudFormation y permisos de uso. AWS KMS

1. [Inicie sesión en la AWS CloudFormation consola AWS Management Console y ábrala en https://console.aws.amazon.com/cloudformation](https://console.aws.amazon.com/cloudformation).
2. Cree una nueva pila utilizando una de las siguientes opciones:
  - Elija Crear pila. Esta es la única opción si tiene una pila actualmente en activo.
  - En la página Pilas, elija Crea pila. Esta opción sólo es visible si no hay pilas en ejecución.
3. En la página Especificar la plantilla:
  - En Preparar la plantilla, elija La plantilla está lista.
  - En Especificar plantilla, elija la Amazon S3 URL como origen de la plantilla e introduzca una de las siguientes URL en función de su Región de AWS.

- <https://m2-us-east-1.s3.us-east-1.amazonaws.com/cicd/mf/basic-infra.yaml>
- <https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/basic-infra.yaml>
- Para aceptar la configuración elija Siguiente.

Se abre la página Crear pila.

## Specify stack details

### Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

### Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

#### Networking Configuration

Do you want to use an existing VPC in your account?

If you select 'Yes', then you must provide the VPC ID and the Subnet IDs.

Which VPC ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID in a different AZ should be used for HA?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Enter the CIDR block that should be used for the new VPC

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

CIDR bits for creating subnets. Choose 5 for /27, 6 for /26, 7 for /25, 8 for /24 range

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

#### Deployment Configuration

Name of the S3 bucket which contains the source files for this stack deployment

Don't change unless you know what you are doing.

Name of the source package file for the infrastructure Lambda function

Don't change unless you know what you are doing.

Full URL of the pipeline CloudFormation template file


Don't change unless you know what you are doing.

What name prefix to use for the new S3 buckets?

A name prefix for the S3 buckets that will be created by this stack.


Realice los siguientes cambios:

- Proporcione los valores adecuados para el nombre y los parámetros de la pila para la configuración de red.
- La mayoría de los parámetros de Configuraciones de la implementación se rellenan previamente de forma adecuada, por lo que no es necesario modificarlos. En función de su Región de AWS ubicación, cambie la AWS CloudFormation plantilla de canalización a una de las siguientes URL de Amazon S3.
  - <https://m2-us-east-1.s3.amazonaws.com/cicd/mf/pipeline.yaml>
  - <https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/pipeline.yaml>
- Elija Siguiente.

 Note

No cambie los valores de los parámetros predeterminados a menos que haya modificado la AWS CloudFormation plantilla usted mismo.

4. En Configurar opciones de pila, elija Siguiente.
5. En Capacidades, elija Acepto que AWS CloudFormation podría crear recursos de IAM AWS CloudFormation para permitir la creación de un rol de IAM en su nombre. Seleccione Crear pila.

 Note

El aprovisionamiento de la pila puede tardar entre 3 y 5 minutos.

6. Una vez que la pila se haya creado correctamente, diríjase a la sección Salidas de la pila recién aprovisionada. Allí encontrará el bucket de Amazon S3 en el que debe cargar el código de la unidad central y los archivos dependientes.

Stack info	Events	Resources	Outputs	Parameters	Template	Change sets
<b>Outputs (7)</b>						
<input type="text" value="Search outputs"/>						
Key	Value	Description				
M2CICDNewPrivateSubnet1	subnet-0e1dda3ae86f025da	Subnet 1 for M2 CI/CD				
M2CICDNewPrivateSubnet2	subnet-0b89e607975284f8f	Subnet 2 for M2 CI/CD				
M2CICDNewVPC	vpc-034cbfc880b73dd28	VPC Id for M2 CI/CD				
MainframeCodeBucketS3URI	s3://mf-code-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Code S3 Bucket				
MainframeCodeBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Code S3 Bucket				
MainframeDataBucketS3URI	s3://mf-data-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Test Data S3 Bucket				
MainframeDataBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Test Data S3 Bucket				

## Cree un AWS CodeCommit repositorio y una canalización de CI/CD

En este paso, se crea un CodeCommit repositorio y se aprovisiona una pila de canalizaciones de CI/CD llamando a una función de Lambda que llama AWS CloudFormation para crear la pila de canalizaciones.

1. Descargue la [aplicación de BankDemo muestra](#) en su máquina local.
2. Cargue `bankdemo.zip` desde el equipo local al bucket de Amazon S3 creado en [Creación de una infraestructura básica de canalización de CI/CD](#).
3. Descargue `config_git.yml`.
4. Modifique el `config_git.yml` si es necesario, de la siguiente manera:
  - Añada su propio nombre de repositorio de destino, rama de destino y mensaje de confirmación.

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
```

- Escriba la dirección de correo electrónico que desea utilizar para las notificaciones.

```
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. Cargue el `config_git.yml` archivo que contiene la definición de la estructura de carpetas del CodeCommit repositorio en el bucket de Amazon S3 creado en [Creación de una infraestructura básica de canalización de CI/CD](#). Esto invocará la función de Lambda que aprovisionará automáticamente el repositorio y la canalización.

Esto creará un CodeCommit repositorio con el nombre indicado en `target-repository` el `config_git.yml` archivo; por ejemplo, `bankdemo-repo`.

La función Lambda también creará la pila de canalizaciones de CI/CD a través de ella. AWS CloudFormation La pila AWS CloudFormation tendrá el mismo prefijo que el nombre de `target-repository` proporcionado seguido de una cadena aleatoria (por ejemplo, `bankdemo-repo-01234567`). Puede encontrar la URL del CodeCommit repositorio y la URL para acceder a la canalización creada en la AWS consola de administración.

**bankdemo-repo-mcdilnof** [Delete] [Update] [Stack actions ▼] [Create stack ▼]

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

**Outputs (2)** [Refresh] [Settings]

Search outputs

Key	Value	Description
CodeCommitRepo	<a href="https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo">https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo</a>	HTTPS endpoint to clone the CodeCommit repository
PipelineURL	<a href="https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2">https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2</a>	URL to access the pipeline on AWS Management Console

6. Si se completa la creación del CodeCommit repositorio, la canalización de CI/CD se activará inmediatamente para realizar una CI/CD completa.



7. Una vez insertado el archivo, se activará automáticamente la canalización, que se construirá, se desplegará en el entorno provisional, realizará algunas pruebas y esperará a la aprobación manual antes de implementarlo todo en el entorno de producción.

## Ejemplo de archivo de activación de YAML: config\_git.yml

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
  directory-structure:
    - '/':
      files:
        - build.xml
        - '*.yaml'
        - '*.yml'
        - '*.xml'
        - 'LICENSE.txt'
      readme: |
        # Root Folder
        - 'build.xml' : Build configuration for the application
    - tests:
      files:
        - '*.py'
      readme: |
        # Test Folder
        - '*.py' : Test scripts
    - config:
      files:
        - 'BANKDEMO.csd'
        - 'BANKDEMO.json'
        - 'BANKDEMO_ED.json'
        - 'dfhldrdat'
        - 'ESPGSQLXA.dll'
        - 'ESPGSQLXA64.so'
        - 'ESPGSQLXA64_S.so'
        - 'EXTFH.cfg'
        - 'm2-2021-04-28.normal.json'
        - 'MFDBFH.cfg'
        - 'application-definition-template-config.json'
      readme: |
        # Config Folder
```

This folder contains the application configuration files.

- 'BANKDEMO.csd' : CICS Resource definitions export file
- 'BANKDEMO.json' : Enterprise Server configuration
- 'BANKDEMO\_ED.json' : Enterprise Server configuration for ED
- 'dfhdrdat' : CICS resource definition file
- 'ESPGSQLXA.dll' : XA switch module Windows
- 'ESPGSQLXA64.so' : XA switch module Linux
- 'ESPGSQLXA64\_S.so' : XA switch module Linux
- 'EXTFH.cfg' : Micro Focus File Handler configuration
- 'm2-2021-04-28.normal.json' : M2 request document
- 'MFDBFH.cfg' : Micro Focus Database File Handler
- 'application-definition-template-config.json' : Application definition for

M2

- source:
  - subdirs:
    - .settings:
      - files:
        - '.bms.mfdirset'
        - '.cbl.mfdirset'
    - copybook:
      - files:
        - '\*.cpy'
        - '\*.inc'
      - readme: |
        - # Copy folder
        - This folder contains the source for COBOL copy books, PLI includes, ...
        - .cpy COBOL copybooks
        - .inc PLI includes
  - # - ctlcards:
    - # files:
      - # - '\*.ctl'
      - # - 'KBNKSRT1.txt'
    - # readme: |
      - # # Control Card folder
      - # This folder contains the source for Batch Control Cards
      - # - .ctl Control Cards
  - ims:
    - files:
      - '\*.dbd'
      - '\*.psb'
    - readme: |
      - # ims folder
      - This folder contains the IMS DB source files with the extensions
      - .dbd for IMS DBD source

```
    - .psb for IMS PSB source
- jcl:
  files:
    - '*.jcl'
    - '*.ctl'
    - 'KBNKSRT1.txt'
    - '*.prc'
  readme: |
    # jcl folder
    This folder contains the JCL source files with the extensions
    - .jcl
#
# - proclib:
#   files:
#     - '*.prc'
#   readme: |
#     # proclib folder
#     This folder contains the JCL procedures referenced via PROCLIB
#     statements in the JCL with extensions
#     - .prc
- rdbms:
  files:
    - '*.sql'
  readme: |
    # rdbms folder
    This folder contains any DB2 related source files with extensions
    - .sql for any kind of SQL source
- screens:
  files:
    - '*.bms'
    - '*.mfs'
  readme: |
    # screens folder
    This folder contains the screens source files with the extensions
    - .bms for CICS BMS screens
    - .mfs for IMS MFS screens
  subdirs:
    - .settings:
      files:
        - '*.bms.mfdirset'
- cobol:
  files:
    - '*.cbl'
    - '*.pli'
  readme: |
```

```
# source folder
This folder contains the program source files with the extensions
- .cbl for COBOL source
- .pli for PLI source
subdirs:
- .settings:
  files:
    - '*.cbl.mfdirset'
- tests:
  files:
  - 'test_script.py'
  readme: |
    # tests Folder
    This folder contains the application test scripts
pipeline-config:
  alert-notifications:
  - myname@mycompany.com
  approval-notifications:
  - myname@mycompany.com
```

## Creación de Enterprise Developer 2.0 AppStream

Para configurar Micro Focus Enterprise Developer en la AppStream versión 2.0, consulte [Tutorial: Configurar Micro Focus Enterprise Developer en AppStream 2.0](#).

Para conectar el CodeCommit repositorio a Enterprise Developer, utilice el nombre especificado `target-repository` en [Ejemplo de archivo de activación de YAML: config\\_git.yml](#).


## Configuración y prueba de Enterprise Developer

### Temas

- [Clona el BankDemo CodeCommit repositorio en Enterprise Developer](#)
- [Cree un proyecto COBOL para BankDemo mainframe y cree una aplicación](#)
- [Cree un BankDemo CICS local y un entorno por lotes para las pruebas](#)
- [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
- [Inicie el terminal Rumba 3270](#)
- [Ejecute una BankDemo transacción](#)
- [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)

Conéctese a la instancia de Enterprise Developer AppStream 2.0 en la que creó [Creación de Enterprise Developer 2.0 AppStream](#) .

1. Inicie Enterprise Developer desde el menú de inicio de Windows. Elija Micro Focus Enterprise Developer y, a continuación, Enterprise Developer for Eclipse. Si es la primera vez que lo inicia, es posible que tarde algún tiempo.
2. En el lanzador de Eclipse, en el espacio de trabajo: introduzca `C:\Users\\workspace` y seleccione Lanzar.


 Note

Asegúrese de elegir la misma ubicación después de volver a conectarse a la instancia AppStream 2.0. La selección del espacio de trabajo no es persistente.

3. En Welcome, elija Open COBOL Perspective. Esto solo se mostrará la primera vez en un espacio de trabajo nuevo.

## Clona el BankDemo CodeCommit repositorio en Enterprise Developer

1. Elija Window / Perspective / Open Perspective / Other ... / Git.
2. Elija Clone a Git repository.
3. En Clone Git Repository, introduzca la siguiente información:
  - En el URI de ubicación, introduzca la URL HTTPS del CodeCommit repositorio.

 Note

Copie la URL HTTPS del clon del CodeCommit repositorio en la consola AWS de administración y péguela aquí. La URI se dividirá en las rutas de Host y Repository.

- El usuario CodeCommit deposita las credenciales en Authentication User and Password y selecciona Guardar en almacenamiento seguro.
4. En Branch Selection, elija Main branch y, a continuación, Next .
  5. En Destino local, en Directorio, ingresa `C:\Users\\workspace` y selecciona Finalizar.

El proceso de clonación finaliza cuando `BANKDEMO [main]` se muestra en la vista Repositorios de Git.

## Cree un proyecto COBOL para BankDemo mainframe y cree una aplicación

1. Cambie a COBOL Perspective.
2. En Project, desactive Build Automatically.
3. En File, seleccione New y, a continuación, Mainframe COBOL Project.
4. En New Mainframe COBOL Project, introduzca la siguiente información:
  - En Project name, introduzca BankDemo.
  - Elija la Micro Focus template [64 bit]
  - Elija Finish.
5. En COBOL Explorer, amplíe el nuevo proyecto. BankDemo

### Note

[`BANKDEMO main`]entre corchetes indica que el proyecto está conectado con el BankDemo CodeCommit repositorio local.

6. Si la vista en árbol no muestra las entradas de los programas COBOL, los libros de texto, el código fuente de BMS y los archivos JCL, seleccione Actualizar en el menú contextual del BankDemo proyecto.
7. BankDemo En el menú contextual, seleccione Propiedades/Micro Focus/Configuración del proyecto/COBOL:
  - Seleccione Character Set - ASCII.
  - Elija Apply y, después, Close.
8. Si la compilación del origen BMS y COBOL no se inicia inmediatamente, compruebe en el menú Project que la opción Build Automatically esté habilitada.

La salida de la compilación se mostrará en la vista de la consola y lo normal es que se complete después de unos minutos con los mensajes `BUILD SUCCESSFUL` y `Build finished with no errors`.

La BankDemo aplicación debería estar ahora compilada y lista para su ejecución local.

## Cree un BankDemo CICS local y un entorno por lotes para las pruebas

1. En COBOL Explorer, expanda BANKDEMO / config.
2. En el editor, abra BANKDEMO\_ED.json.
3. Busque la cadena ED\_Home= y cambie la ruta para que apunte al proyecto de Enterprise Developer, de la siguiente manera: D:\\<username>\\workspace\\BANKDEMO. Observe el uso de barras dobles (\\) en la definición de la ruta.
4. Guarde y cierre el archivo.
5. Elija Server Explorer.
6. En el menú contextual predeterminado, elija Open Administration Page. La página de administración del servidor Micro Focus Enterprise se abre en el navegador predeterminado.
7. Solo para las sesiones AppStream 2.0, realice los siguientes cambios para poder conservar su región local de Enterprise Server para las pruebas locales:
  - En Directory Server / Default, elija PROPERTIES / Configuration.
  - Sustituya la Repository Location por D:\\<username>\\My Files\\Home Folder\\MFDS.

### Note

Debe completar los pasos 5 a 8 después de cada nueva conexión a una instancia AppStream 2.0.

8. En Directory Server / Default, elija Import y, a continuación, complete los siguientes pasos:
  - En Step 1: Import Type, elija JSON y Next.
  - En Step 2: Upload, haga clic para cargar el archivo del cuadrado azul.
  - En Choose File to Upload, introduzca:
    - File name: D:\\<username>\\workspace\\BANKDEMO\\config\\BANKDEMO\_ED.json.
    - Elija Open.
  - Elija Next.
  - En Step 3: Regions elimine Clear Ports from Endpoints.
  - Elija Siguiente.
  - En Step 4: Import, elija Import.
  - Seleccione Finalizar.

La lista mostrará ahora un nuevo nombre de servidor BANKDEMO.

## Inicie el servidor de BANKDEMO desde Enterprise Developer

1. Elija Enterprise Developer.
2. En el explorador de servidores, elija Default y, a continuación, elija Refresh en el menú contextual.

La lista de servidores ahora también debería mostrar BANKDEMO.

3. Elija BANKDEMO.
4. En el menú contextual, elija Associate with project y, a continuación, elija BANKDEMO.
5. En el menú contextual, elija Start.

La vista de la consola debería mostrar el registro del inicio del servidor.

Si aparece el mensaje BANKDEMO CASSI5030I PLTPI Phase 2 List(PI) Processing Completed, el servidor está listo para probar la aplicación BANKDEMO de CICS.

## Inicie el terminal Rumba 3270

1. Desde el menú de inicio de Windows, inicie Micro Focus Rumba+ Desktop o Rumba+ Desktop.
2. En Welcome, elija CREATE NEW SESSION / Mainframe Display.
3. En Mainframe Display, elija Connection / Configure.
4. En Session Configuration, elija Connection / TN3270.
5. En Host Name / Address, elija Insert and enter IP address 127.0.0.1.
6. En Telnet Port, introduzca el puerto 6000.
7. Seleccione Apply.
8. Elija Connect.

La pantalla de bienvenida del CICS muestra una pantalla con el mensaje de la fila 1: This is the Micro Focus MFE CICS region BANKDEMO.

9. Pulse Ctrl+Shift+z para borrar la pantalla.



## Ejecute una BankDemo transacción

1. En una pantalla vacía, introduzca BANK.
2. En la pantalla BANK10, en el campo de entrada para el ID de usuario... :, escriba guest y pulse Intro.
3. En la pantalla BANK20, en el campo de entrada situado antes de Calculate the cost of a loan, introduzca / (barra inclinada) y pulse Intro.
4. En la pantalla BANK70:
  - En The amount you would like to borrow..., introduzca 10000.
  - En At an interest rate of.....:, introduzca 5.0.
  - En For how many months.....:, introduzca 10.
  - Pulse Intro.

Se debe mostrar el siguiente resultado:

```
Resulting monthly payment.....: $1023.06
```

Esto completa la configuración de la aplicación de BANKDEMO en Enterprise Developer.

## Detenga el servidor de BANKDEMO desde Enterprise Developer

1. En el explorador de servidores, elija Default y, a continuación, elija Refresh en el menú contextual.
2. Elija BANKDEMO.
3. En el menú contextual, elija Stop.

La vista de la consola debería mostrar el registro de la detención del servidor.

Si se muestra el mensaje `Server: BANKDEMO stopped successfully`, el servidor se ha apagado correctamente.

# Ejercicio 1: Mejorar el cálculo de los préstamos en la aplicación de BANKDEMO

## Temas

- [Agregue la regla de análisis de préstamos a Enterprise Developer Code Analysis](#)
- [Paso 1: Realizar un análisis de código para calcular el préstamo](#)
- [Paso 2: Modificar el mapa CICS BMS y el programa COBOL y pruébelo](#)
- [Paso 3: Agregar el cálculo del importe total en el programa COBOL](#)
- [Paso 4: Confirmar los cambios y ejecutar la canalización de CI/CD](#)

En este escenario, explicará el proceso de realizar un ejemplo de cambio en el código, implementarlo y probarlo.

El departamento de préstamos quiere que aparezca un nuevo campo en la pantalla de cálculo del préstamo de BANK70 para mostrar el importe total del préstamo. Para ello, es necesario cambiar la pantalla MBANK70.CBL del BMS y añadir un nuevo campo y el programa de gestión de pantalla correspondiente, el SBANK70P.CBL, con los cuadernos correspondientes. Además, la rutina de cálculo de préstamos del BBANK70P.CBL debe ampliarse con la fórmula adicional.

Para completar este ejercicio, asegúrese de completar los siguientes requisitos previos.

- Descargue [BANKDEMO-exercise.zip](#) en D:\PhotonUser\My Files\Home Folder.
- Descomprima el archivo .zip en D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise.
- Crea la carpeta D:\PhotonUser\My Files\Home Folder\AnalysisRules.
- Copie el archivo de reglas Loan+Calculation+Update.General-1.xml de la carpeta BANKDEMO-exercise a D:\PhotonUser\My Files\Home Folder\AnalysisRules.

### Note

Para este ejercicio, los cambios de código en \*.CBL y \*.CPY se marcan con EXER01 en las columnas 1 a 6.

## Agregue la regla de análisis de préstamos a Enterprise Developer Code Analysis

Las reglas de análisis definidas en Micro Focus Enterprise Analyzer se pueden exportar desde Enterprise Analyzer e importar a Enterprise Developer para ejecutar las mismas reglas de análisis en todos los orígenes del proyecto de Enterprise Developer.

1. Abra `Window/Preferences/Micro Focus/COBOL/Code Analysis/Rules`.
2. Elija `Edit...` e introduzca el nombre de la carpeta `D:\PhotonUser\My Files\Home Folder\AnalysisRules` que contiene el archivo de reglas `Loan+Calculation+Update.General-1.xml`.
3. Seleccione `Finalizar`.
4. Seleccione `Apply` y, a continuación, seleccione `Close`.
5. En el menú contextual del proyecto de `BANKDEMO`, seleccione `Code Analysis`.

Debería ver una entrada para `Loan Calculation Update`.

### Paso 1: Realizar un análisis de código para calcular el préstamo

Con la nueva regla de análisis, queremos identificar los programas COBOL y las líneas de código que contienen que coinciden con los patrones de búsqueda `*PAYMENT*`, `*LOAN*` y `*RATE*` en las expresiones, instrucciones y variables. Esto nos ayudará a navegar por el código e identificar los cambios de código necesarios.

1. En el menú contextual del proyecto de `BANKDEMO`, seleccione `Code Analysis/Loan Calculation Update`.

Esto ejecutará la regla de búsqueda y mostrará los resultados en una nueva pestaña llamada `Code Analysis`. La ejecución del análisis finaliza cuando desaparece la barra de progreso verde situada en la parte inferior derecha.

La pestaña `Code Analysis` debería mostrar una lista expandida de `BBANK20P.CBL`, `BBANK70P.CBL` y `SBANK70P.CBL`, cada uno con un listado de las instrucciones, expresiones y variables que coincidan con los patrones de búsqueda.

Al observar el resultado de `BBANK20P.CBL`, solo se mueven los literales que coinciden con el patrón de búsqueda. Por lo tanto, este programa se puede ignorar.

2. En la barra de menú de pestañas, elija `Icon` para colapsar todo.

3. Expanda SBANK70P.CBL y seleccione las líneas en cualquier orden haciendo doble clic para ver cómo se abrirá el origen y se resaltará la línea seleccionada en el código fuente. También verá que todas las líneas fuente identificadas están marcadas.

## Paso 2: Modificar el mapa CICS BMS y el programa COBOL y pruébelo

Primero cambiaremos el mapa del BMS MBANK70.BMS, el programa de gestión de pantalla SBANK70P.CBL y el cuaderno CBANKDAT.CPY para mostrar el nuevo campo. Para evitar codificación innecesaria en este ejercicio, los módulos de origen modificados están disponibles en la carpeta D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01. Normalmente, un desarrollador utilizaría los resultados del análisis de código para navegar y modificar los orígenes. Si tiene tiempo y desea realizar los cambios manualmente, hágalo con la información proporcionada en \*Manual change in MBANK70.BMS and SBANK70P.CBL (Optional)\*.

Para realizar cambios rápidos, copie los siguientes archivos:

1. ..\BANKDEMO-exercise\Exercise01\screens\MBANK70.BMS a D:\PhotonUser\workspace\bankdemo\source\screens.
2. .\BANKDEMO-exercise\Exercise01\cobol\SBANK70P.CBL a D:\PhotonUser\workspace\bankdemo\source\cobol.
3. ..\BANKDEMO-exercise\Exercise01\copybook\CBANKDAT.CPY a D:\PhotonUser\workspace\bankdemo\source\copybook.
4. Para asegurarse de que todos los programas afectados por los cambios estén compilados, elija Project/Clean.../Clean all project.

Para realizar cambios manuales en MBANK70.BMS y SBANK70P.CBL, complete los siguientes pasos:

- Para cambiar manualmente el origen MBANK70.BMS del BMS, añada después del campo PAYMENT:
  - TXT09 con los mismos atributos que el TXT08 y el valor INITIAL "Total Loan Amount"
  - TOTAL con los mismos atributos que PAYMENT

Pruebe los cambios

Para probar los cambios, repita los pasos de las siguientes secciones:

1. [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
2. [Inicie el terminal Rumba 3270](#)
3. [Ejecute una BankDemo transacción](#)

Además, ahora también debería ver el texto Total Loan Amount.....:.

4. [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)

### Paso 3: Agregar el cálculo del importe total en el programa COBOL

En el segundo paso, cambiaremos BBANK70P.CBL y sumaremos el cálculo del importe total del préstamo. El origen preparado con los cambios necesarios está disponible en la carpeta D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01. Si tiene tiempo y desea realizar los cambios manuales, hágalo con la información proporcionada en \*Cambio manual en BBANK70P.CBL (opcional)\*.

Para realizar cambios rápidos, copie el siguiente archivo:

- ..\BANKDEMO-exercise\Exercise01\source\cobol\BBANK70P.CBL a D:\PhotonUser\workspace\bankdemo\source\cobol.

Para realizar un cambio manual en BBANK70P.CBL, complete los siguientes pasos:

- Utilice el resultado del análisis de código para identificar los cambios necesarios.

Pruebe los cambios

Para probar los cambios, repita los pasos de las siguientes secciones:

1. [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
2. [Inicie el terminal Rumba 3270](#)
3. [Ejecute una BankDemo transacción](#)

Además, ahora también debería ver el texto Total Loan Amount.....: \$10230.60.

4. [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)

## Paso 4: Confirmar los cambios y ejecutar la canalización de CI/CD

Guarda los cambios en el CodeCommit repositorio central y active la canalización de CI/CD para crear, probar e implementar los cambios.

1. En el menú contextual del proyecto de BANKDEMO, seleccione Team/Commit.
2. En la pestaña Git Staging, introduzca el siguiente mensaje de confirmación: `Added Total Amount Calculation`.
3. Elija Commit and Push....
4. Abra la CodePipeline consola y compruebe el estado de la ejecución de la canalización.

### Note

En caso de que tenga algún problema con la característica Commit o Push de Enterprise Developer o Teams, utilice la interfaz de línea de comandos de Git Bash.

## Ejercicio 2: Extraiga el cálculo del préstamo en BankDemo la solicitud

### Temas

- [Paso 1: Refactorizar la rutina de cálculo de préstamos en una sección de COBOL](#)
- [Paso 2: Extraer la rutina de cálculo de préstamos a un programa COBOL independiente](#)
- [Paso 3: Confirmar los cambios y ejecutar la canalización de CI/CD](#)

En el siguiente ejercicio, analizará otro ejemplo de solicitud de cambio. En este escenario, el departamento de préstamos quiere reutilizar la rutina de cálculo de préstamos de forma independiente. WebService La rutina debería permanecer en COBOL y también debería poder invocarse desde el programa CICS COBOL existente BBANK70P.CBL.

### Paso 1: Refactorizar la rutina de cálculo de préstamos en una sección de COBOL

En el primer paso, extraemos la rutina de cálculo del préstamo en una sección de COBOL. Este paso es necesario para extraer el código en un programa COBOL independiente en el siguiente paso.

1. Abra BBANK70P.CBL en el editor de COBOL.

2. En el menú contextual del proyecto de BANKDEMO, seleccione Code Analysis/Loan Calculation Update. Esto solo escaneará el origen actual en busca de los patrones definidos en la regla de análisis.
3. En el resultado, en la pestaña Code Analysis, busque la primera instrucción aritmética DIVIDE WS-LOAN-INTEREST BY 12.
4. Haga doble clic en la instrucción para ir a la línea de origen en el editor. Esta es la primera instrucción de la rutina de cálculo de préstamos.
5. Marque el siguiente bloque de código para extraer la rutina de cálculo de préstamos en una sección.

```

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. En el menú contextual del editor, seleccione Refactor/Extract to Section....
7. Introduzca el New section name: LOAN-CALCULATION.
8. Seleccione Aceptar.

El bloque de código marcado se ha extraído ahora a la nueva sección LOAN-CALCULATION y el bloque de código se ha sustituido por la instrucción PERFROM LOAN-CALCULATION.

Pruebe los cambios

Para probar los cambios, repita los pasos de las siguientes secciones:

1. [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
2. [Inicie el terminal Rumba 3270](#)
3. [Ejecute una BankDemo transacción](#)

Además, ahora también debería ver el texto Total Loan Amount.....: \$10230.60.

4. [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)

**Note**

Si quiere evitar los pasos anteriores para extraer el bloque de código a una sección, puede copiar el origen modificado del paso 1 de `..\BANKDEMO-exercise\Exercis02\Step1\cobo1\BBANK70P.CBL` a `D:\PhotonUser\workspace\bankdemo\source\cobo1`.

## Paso 2: Extraer la rutina de cálculo de préstamos a un programa COBOL independiente

En el paso 2, el bloque de código de la sección LOAN-CALCULATION se extraerá a un programa independiente y el código original se sustituirá por el código para llamar al nuevo subprograma.

1. Abra `BBANK70P.CBL` en el editor y busque la nueva instrucción `PERFORM LOAN-CALCULATION` creada en el paso 1.
2. Coloque el cursor dentro del nombre de la sección. Se marcará en gris.
3. En el menú contextual, seleccione `Refactor->Extract Section/Paragraph to Program....`
4. En `Extract Section/Paragraph to Program`, introduzca `New file name: LOANCALC.CBL`.
5. Seleccione `OK`.

El nuevo programa `LOANCALC.CBL` se abrirá en el editor.

6. Desplácese hacia abajo y revise el código que se está extrayendo y generando para la interfaz de llamada.
7. Seleccione el editor con `BBANK70P.CBL` y vaya a `LOAN-CALCULATION SECTION`. Revise el código que se está generando para llamar al nuevo subprograma `LOANCALC.CBL`.

**Note**

La instrucción `CALL` utiliza `DFHEIBLK` y `DFHCOMMAREA` para llamar a `LOANCALC` con bloques de control de CICS. Como queremos denominar al nuevo subprograma `LOANCALC.CBL` como programa ajeno al CICS, tenemos que eliminar `DFHEIBLK` y `DFHCOMMAREA` de la llamada comentando o borrando.



## Pruebe los cambios

Para probar los cambios, repita los pasos de las siguientes secciones:

1. [Inicie el servidor de BANKDEMO desde Enterprise Developer](#)
2. [Inicie el terminal Rumba 3270](#)
3. [Ejecute una BankDemo transacción](#)

Además, ahora también debería ver el texto Total Loan Amount.....: \$10230.60.

4. [Detenga el servidor de BANKDEMO desde Enterprise Developer](#)

### Note

Si quiere evitar los pasos anteriores para extraer el bloque de código a una sección, puede copiar el origen modificado del paso 1 de ..\BANKDEMO-exercise \Exercis02\Step2\cobl\BBANK70P.CBL y LOANCALC.CBL a D:\PhotonUser \workspace\bankdemo\source\cobl.

## Paso 3: Confirmar los cambios y ejecutar la canalización de CI/CD

Guarda los cambios en el CodeCommit repositorio central y active la canalización de CI/CD para crear, probar e implementar los cambios.

1. En el menú contextual del proyecto de BANKDEMO, seleccione Team/Commit.
2. En la pestaña Git Staging
  - Añada Unstaged Stages LOANCALC.CBL y LOANCALC.CBL.mfdirset.
  - Introduzca un mensaje de confirmación: Added Total Amount Calculation.
3. Elija Commit and Push....
4. Abra la CodePipeline consola y compruebe el estado de la ejecución de la canalización.

**Note**

En caso de que tenga algún problema con la característica Commit o Push de Enterprise Developer o Teams, utilice la interfaz de línea de comandos de Git Bash.

## Limpieza de recursos

Si ya no necesita los recursos que ha creado para este tutorial, elimínelos para que no se le siga cobrando por ellos. Realice los siguientes pasos:

- Elimine la CodePipeline canalización. Para obtener más información, consulte [Eliminar una canalización CodePipeline en](#) la Guía del AWS CodePipeline usuario.
- Elimine el CodeCommit repositorio. Para obtener más información, consulte [Eliminar un CodeCommit repositorio](#) en la Guía del AWS CodeCommit usuario.
- Elimine el bucket de S3. Para obtener más información, consulte [Eliminar un bucket](#) en la Guía del usuario de Amazon Simple Storage Service.
- Elimine la AWS CloudFormation pila. Para obtener más información, consulte [Eliminar una pila de la AWS CloudFormation consola](#) en la Guía del AWS CloudFormation usuario.

## Utilidades por lotes en la modernización de AWS mainframes

Las aplicaciones de unidad central suelen utilizar programas de utilidades por lotes para realizar funciones específicas, como ordenar datos, transferir archivos mediante FTP, cargar datos en bases de datos como DB2, descargar datos de bases de datos, etc.

Al migrar sus aplicaciones a la modernización del AWS mainframe, necesitará utilidades de reemplazo funcionalmente equivalentes que puedan realizar las mismas tareas que las que utilizaba en el mainframe. Es posible que algunas de estas utilidades ya estén disponibles como parte de los motores de ejecución de AWS Mainframe Modernization, pero ofrecemos las siguientes utilidades de reemplazo:

- M2SFTP: permite la transferencia segura de archivos mediante el protocolo SFTP.
- M2WAIT: espera un período de tiempo específico antes de continuar con el siguiente paso de un trabajo por lotes.
- TXT2PDF: convierte archivos de texto a formato PDF.

- M2DFUTIL: proporciona funciones de copia de seguridad, restauración, eliminación y copia en conjuntos de datos, de forma similar al soporte que proporciona la utilidad ADRDSSU del mainframe.
- M2RUNCMD: permite ejecutar comandos, scripts y llamadas al sistema de Micro Focus directamente desde JCL.

Desarrollamos estas utilidades por lotes basándonos en los comentarios de los clientes y las diseñamos para que proporcionaran la misma funcionalidad que las utilidades de unidad central. El objetivo es hacer que la transición de la modernización del mainframe a la modernización del AWS mainframe sea lo más fluida posible.

## Temas

- [Ubicación binaria](#)
- [Utilidad por lotes M2SFTP](#)
- [Utilidad por lotes M2WAIT](#)
- [Utilidad TXT2PDF por lotes](#)
- [Utilidad por lotes M2DFUTIL](#)
- [Utilidad por lotes M2RUNCMD](#)

## Ubicación binaria

Estas utilidades se encuentran preinstaladas en los productos Micro Focus Enterprise Developer (ED) y Micro Focus Enterprise Server (ES). Puede encontrarlas en la siguiente ubicación de todas las variantes de ED y ES:

- Linux: `/opt/aws/m2/microfocus/utilities/64bit`
- Windows de 32 bits: `C:\AWS\M2\MicroFocus\Utilities\32bit`
- Windows de 64 bits: `C:\AWS\M2\MicroFocus\Utilities\64bit`

## Utilidad por lotes M2SFTP

M2SFTP es un programa de utilidades de JCL diseñado para realizar transferencias de archivos seguras entre sistemas que utilizan el protocolo de transferencia segura de archivos (SFTP). El programa utiliza el cliente SFTP de Putty, `psftp`, para realizar las transferencias reales de archivos.

El programa funciona de manera similar a un programa de utilidades de FTP para unidades centrales y utiliza la autenticación de usuario y contraseña.

 Note

No se admite la autenticación de clave pública.

Para convertir los JCL de FTP de su unidad central a SFTP, cambie PGM=FTP por PGM=M2SFTP.

## Temas

- [Plataformas admitidas](#)
- [Instalación de dependencias](#)
- [Configuración de M2SFTP para la modernización del mainframe gestionada AWS](#)
- [Configuración de M2SFTP para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 \(incluida la versión 2.0\) AppStream](#)
- [JCL de ejemplo](#)
- [Referencia de comandos del cliente SFTP de Putty \(PSFTP\)](#)
- [Sigüientes pasos](#)

## Plataformas admitidas

Puede utilizar M2SFTP en cualquiera de las siguientes plataformas:

- AWS Modernización de mainframe gestionada por Micro Focus
- Micro Focus Runtime (en Amazon EC2)
- Todas las variantes de los productos Micro Focus Enterprise Developer (ED) y Micro Focus Enterprise Server (ES).

## Instalación de dependencias

Para instalar el cliente SFTP de Putty en Windows

- Descargue el cliente [SFTP de PuTTY](#) e instálelo.

Para instalar el cliente SFTP de Putty en Linux:

- Ejecute el siguiente comando para instalar el cliente SFTP de Putty.

```
sudo yum -y install putty
```

## Configuración de M2SFTP para la modernización del mainframe gestionada AWS

Si sus aplicaciones migradas se ejecutan en AWS Mainframe Modernization Managed, tendrá que configurar M2SFTP de la siguiente manera.

- Configure las variables de entorno de Micro Focus Enterprise Server adecuadas para el MFFTP. A continuación se muestran algunos ejemplos:
  - MFFTP\_TEMP\_DIR
  - MFFTP\_SENDEOL
  - MFFTP\_TIME
  - MFFTP\_ABEND

Puede configurar tantas variables como desee. Puede configurarlas en su JCL mediante la instrucción ENVAR DD. Para obtener más información sobre estas variables, consulte [MFFTP Control Variables](#) en la documentación de Micro Focus.

Para probar la configuración, consulte [JCL de ejemplo](#).

## Configuración de M2SFTP para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 (incluida la versión 2.0) AppStream

Si las aplicaciones migradas se ejecutan en el entorno de ejecución de modernización de AWS mainframe en Amazon EC2, configure M2SFTP de la siguiente manera.

1. Cambie la [ruta del programa JES de Micro Focus](#) para incluir la ubicación binaria de las utilidades por lotes. Si necesita especificar varias rutas, utilice dos puntos (:) para separarlas en Linux y punto y coma (;) en Windows.

- Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bits): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bits): C:\AWS\M2\MicroFocus\Utilities\64bit
2. Configure las variables de entorno de Micro Focus Enterprise Server adecuadas para el MFFTP. A continuación se muestran algunos ejemplos:

- MFFTP\_TEMP\_DIR
- MFFTP\_SENDEOL
- MFFTP\_TIME
- MFFTP\_ABEND

Puede configurar tantas variables como desee. Puede configurarlas en su JCL mediante la instrucción ENVAR DD. Para obtener más información sobre estas variables, consulte [MFFTP Control Variables](#) en la documentación de Micro Focus.

Para probar la configuración, consulte [JCL de ejemplo](#).

## JCL de ejemplo

Para probar la instalación, puede utilizar uno de los siguientes archivos JCL de ejemplo.

### M2SFTP1.jcl

Este JCL muestra cómo llamar a M2SFTP para enviar un archivo a un servidor SFTP remoto. Observe las variables de entorno que se establecen en la instrucción 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
MBSSENDEOL CRLF
TRAILINGBLANKS FALSE
/*
//NETRC DD *
machine 127.0.0.1 login sftpuser password sftppass
/*
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD *
type a
locsite notrailingblanks
cd files
put 'AWS.M2.TXT2PDF1.PDF' AWS.M2.TXT2PDF1.pdf
put 'AWS.M2.CARDDEMO.CARDDATA.PS' AWS.M2.CARDDEMO.CARDDATA.PS1.txt
quit
/*
//ENVVAR DD *
MFFTP_VERBOSE_OUTPUT=ON
MFFTP_KEEP=N
/*
/**
//

```

## M2SFTP2.jcl

Este JCL muestra cómo llamar a M2SFTP para recibir un archivo de un servidor SFTP remoto. Observe las variables de entorno establecidas en la instrucción 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

Recomendamos encarecidamente almacenar las credenciales de FTP en un archivo NETRC y restringir el acceso únicamente a los usuarios autorizados.

## Referencia de comandos del cliente SFTP de Putty (PSFTP)

El cliente PSFTP no admite todos los comandos de FTP. La siguiente lista muestra todos los comandos compatibles con PSFTP.

Comando	Descripción
!	Ejecuta un comando local
bye	Finaliza su sesión de SFTP
cd	Cambia su directorio de trabajo remoto
chmod	Cambia los permisos y modos de los archivos



Comando	Descripción
close	Finaliza la sesión de SFTP pero no cierra PSFTP
del	Elimina archivos en el servidor remoto
dir	Lista los archivos remotos
exit	Finaliza su sesión de SFTP
get	Descarga el archivo del servidor en el equipo local.
help	Brinda ayuda
lcd	Cambia el directorio de trabajo local
lpwd	Imprime el directorio de trabajo local
ls	Lista los archivos remotos
mget	Descarga varios archivos a la vez
mkdir	Crea directorios en el servidor remoto
mput	Carga varios archivos a la vez
mv	Mueve o cambia el nombre de los archivos en el servidor remoto
open	Conectarse a un host
put	Carga un archivo del equipo local al servidor.
pwd	Imprime su directorio de trabajo remoto
quit	Finaliza su sesión de SFTP
reget	Continúa descargando archivos

Comando	Descripción
ren	Mueve los archivos en el servidor remoto o les cambia el nombre
reput	Continúa descargando archivos
rm	Elimina archivos en el servidor remoto
rmdir	Elimina directorios en el servidor remoto

## Siguientes pasos

Para cargar y descargar archivos en Amazon Simple Storage Service mediante SFTP, puede utilizar M2SFTP junto con el AWS Transfer Family, tal y como se describe en las siguientes entradas del blog.

- [Uso de directorios lógicos de AWS SFTP para crear un servicio de distribución de datos sencillo](#)
- [Habilite la autenticación por contraseña para su uso AWS Transfer for SFTP AWS Secrets Manager](#)

## Utilidad por lotes M2WAIT

M2WAIT es un programa de utilidad para unidades centrales que le permite introducir un periodo de espera en los scripts de JCL especificando una duración en segundos, minutos u horas. Puede llamar a M2WAIT directamente desde JCL pasando el tiempo de espera como parámetro de entrada. Internamente, el programa M2WAIT llama al módulo C\$SLEEP suministrado por Micro Focus para que espere un tiempo específico.

### Note

Puede utilizar los alias de Micro Focus para reemplazar los que tiene en sus scripts de JCL. Para obtener más información, consulte [JES Alias](#) en la documentación de Micro Focus.

## Temas

- [Plataformas admitidas](#)
- [Configure M2WAIT para gestionar la modernización del AWS mainframe](#)

- [Configure M2WAIT para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 \(incluida la versión 2.0\) AppStream](#)
- [JCL de ejemplo](#)

## Plataformas admitidas

Puede utilizar M2WAIT en cualquiera de las siguientes plataformas:

- AWS Modernización de mainframe gestionada por Micro Focus
- Micro Focus Runtime (en Amazon EC2)
- Todas las variantes de los productos Micro Focus Enterprise Developer (ED) y Micro Focus Enterprise Server (ES).

## Configure M2WAIT para gestionar la modernización del AWS mainframe

Si sus aplicaciones migradas se ejecutan en AWS Mainframe Modernization Managed, tendrá que configurar M2WAIT de la siguiente manera.

- Utilice el programa M2WAIT en su JCL pasando el parámetro de entrada como se muestra en [JCL de ejemplo](#).

## Configure M2WAIT para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 (incluida la versión 2.0) AppStream

Si las aplicaciones migradas se ejecutan en el entorno de ejecución de modernización de AWS mainframe en Amazon EC2, configure M2WAIT de la siguiente manera.

1. Cambie la [ruta del programa JES de Micro Focus](#) para incluir la ubicación binaria de las utilidades por lotes. Si necesita especificar varias rutas, utilice dos puntos (:) para separarlas en Linux y punto y coma (;) en Windows.
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bits): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bits): C:\AWS\M2\MicroFocus\Utilities\64bit
2. Utilice el programa M2WAIT en su JCL pasando el parámetro de entrada como se muestra en [JCL de ejemplo](#).

## JCL de ejemplo

Para probar la instalación, puede utilizar el programa M2WAIT1.jcl.

Este ejemplo de JCL muestra cómo llamar a M2WAIT y pasarle varias duraciones diferentes.

```
//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=*
/**
//
```

## Utilidad TXT2PDF por lotes

El TXT2PDF es un programa de utilidad para ordenadores centrales que se utiliza habitualmente para convertir un archivo de texto en un archivo PDF. Esta utilidad utiliza el mismo código fuente para TXT2PDF (software gratuito de z/OS). Lo modificamos para que se ejecute en el entorno de ejecución Micro Focus de AWS Mainframe Modernization.

### Temas

- [Plataformas admitidas](#)
- [Configure el TXT2PDF para gestionar la modernización del mainframe AWS](#)
- [Configurar TXT2PDF para el tiempo de ejecución de modernización de AWS mainframe en Amazon EC2 \(incluida la versión 2.0\) AppStream](#)
- [ACL de ejemplo](#)
- [Modificaciones](#)
- [Referencias](#)

## Plataformas admitidas

Puede utilizar TXT2PDF en cualquiera de las siguientes plataformas:

- AWS Modernización del mainframe gestionada por Micro Focus
- Micro Focus Runtime (en Amazon EC2)
- Todas las variantes de los productos Micro Focus Enterprise Developer (ED) y Micro Focus Enterprise Server (ES).

## Configure el TXT2PDF para gestionar la modernización del mainframe AWS

Si sus aplicaciones migradas se ejecutan en AWS Mainframe Modernization Managed, configure TXT2PDF de la siguiente manera.

- Cree una biblioteca REXX EXEC llamada `AWS.M2.REXX.EXEC`. Descargue estos [módulos REXX](#) y cópielos en la biblioteca.
  - `TXT2PDF.rex`- Software gratuito TXT2PDF z/OS (modificado)
  - `TXT2PDFD.rex`- Software gratuito TXT2PDF z/OS (modificado)
  - `TXT2PDFX.rex`- Software gratuito TXT2PDF z/OS (modificado)
  - `M2GETOS.rex`- Para comprobar el tipo de sistema operativo (Windows o Linux)

Para probar la configuración, consulte [ACL de ejemplo](#).

## Configurar TXT2PDF para el tiempo de ejecución de modernización de AWS mainframe en Amazon EC2 (incluida la versión 2.0) AppStream

Si las aplicaciones migradas se ejecutan en el entorno de ejecución de modernización de AWS mainframe en Amazon EC2, configure TXT2PDF de la siguiente manera.

1. Defina la variable de entorno MFREXX\_CHARSET de Micro Focus en el valor adecuado, como "A" para los datos ASCII.

### Important

Si introduce un valor incorrecto, podrían producirse problemas de conversión de datos (de EBCDIC a ASCII), lo que haría que el PDF resultante fuera ilegible o inoperable. Recomendamos configurar MFREXX\_CHARSET para que coincida con MF\_CHARSET.

2. Cambie la [ruta del programa JES de Micro Focus](#) para que incluya la ubicación binaria de las utilidades por lotes. Si necesita especificar varias rutas, utilice dos puntos (:) para separarlas en Linux y punto y coma (;) en Windows.
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bits): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bits): C:\AWS\M2\MicroFocus\Utilities\64bit
3. Cree una biblioteca REXX EXEC llamada AWS.M2.REXX.EXEC`. Descargue estos [módulos REXX](#) y cópielos en la biblioteca.
  - TXT2PDF.rex- Software gratuito TXT2PDF z/OS (modificado)
  - TXT2PDFD.rex- Software gratuito TXT2PDF z/OS (modificado)
  - TXT2PDFX.rex- Software gratuito TXT2PDF z/OS (modificado)
  - M2GETOS.rex- Para comprobar el tipo de sistema operativo (Windows o Linux)

Para probar la configuración, consulte [ACL de ejemplo](#).

### ACL de ejemplo

Para probar la instalación, puede utilizar uno de los siguientes archivos JCL de ejemplo.

TXT2PDF1.jcl

Este archivo JCL de ejemplo usa un nombre DD para la conversión TXT2PDF.

```
//TXT2PDF1 JOB 'TXT2PDF1',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** PRE DELETE*
/**-----**
/**
//PREDEL EXEC PGM=IEFBR14
/**
//DD01 DD DSN=AWS.M2.TXT2PDF1.PDF.VB,
// DISP=(MOD,DELETE,DELETE)
/**
//DD02 DD DSN=AWS.M2.TXT2PDF1.PDF,
// DISP=(MOD,DELETE,DELETE)
/**
/**-----**
/** CALL TXT2PDF TO CONVERT FROM TEXT TO PDF (VB)*
/**-----**
/**
//STEP01 EXEC PGM=IKJEFT1B
/**
//SYSEXEC DD DISP=SHR,DSN=AWS.M2.REXX.EXEC
/**
//INDD DD *
1THIS IS THE FIRST LINE ON THE PAGE 1
0THIS IS THE THIRD LINE ON THE PAGE 1
-THIS IS THE 6TH LINE ON THE PAGE 1
THIS IS THE 7TH LINE ON THE PAGE 1
+_____ - OVERSTRIKE 7TH LINE
1THIS IS THE FIRST LINE ON THE PAGE 2
0THIS IS THE THIRD LINE ON THE PAGE 2
-THIS IS THE 6TH LINE ON THE PAGE 2
THIS IS THE 7TH LINE ON THE PAGE 2
+_____ - OVERSTRIKE 7TH LINE
/*
/**
//OUTDD DD DSN=AWS.M2.TXT2PDF1.PDF.VB,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=256,DSORG=PS,RECFM=VB,BLKSIZE=0)
/**
```

```

//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DDNAME=SYSIN
//*
//SYSIN    DD *
%TXT2PDF BROWSE Y IN DD:INDD +
OUT DD:OUTDD +
CC YES
/*
/**
/**-----**
/** CONVERT PDF (VB) TO PDF (LSEQ - BYTE STREAM)*
/**-----**
/**
//STEP02 EXEC PGM=VB2LSEQ
/**
//INFILE   DD DSN=AWS.M2.TXT2PDF1.PDF.VB,DISP=SHR
/**
//OUTFILE  DD DSN=AWS.M2.TXT2PDF1.PDF,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=256,DSORG=PS,RECFM=LSEQ,BLKSIZE=0)
/**
//SYSOUT   DD SYSOUT=*
/**
//

```

## TXT2PDF2.jcl

Este archivo JCL de ejemplo usa un nombre DSN para la conversión TXT2PDF.

```

//TXT2PDF2 JOB 'TXT2PDF2',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** PRE DELETE*
/**-----**
/**
//PREDEL EXEC PGM=IEFBR14
/**
//DD01   DD DSN=AWS.M2.TXT2PDF2.PDF.VB,
//        DISP=(MOD,DELETE,DELETE)
/**
//DD02   DD DSN=AWS.M2.TXT2PDF2.PDF,

```



```

//          DISP=(MOD,DELETE,DELETE)
//*
//*-----**
//* CALL TXT2PDF TO CONVERT FROM TEXT TO PDF (VB)*
//*-----**
//*
//STEP01 EXEC PGM=IKJEFT1B
//*
//SYSEXEC  DD DISP=SHR,DSN=AWS.M2.REXX.EXEC
//*
//INDD      DD *
1THIS IS THE FIRST LINE ON THE PAGE 1
0THIS IS THE THIRD LINE ON THE PAGE 1
-THIS IS THE   6TH LINE ON THE PAGE 1
THIS IS THE   7TH LINE ON THE PAGE 1
+_____ - OVERSTRIKE 7TH LINE
1THIS IS THE FIRST LINE ON THE PAGE 2
0THIS IS THE THIRD LINE ON THE PAGE 2
-THIS IS THE   6TH LINE ON THE PAGE 2
THIS IS THE   7TH LINE ON THE PAGE 2
+_____ - OVERSTRIKE 7TH LINE
/*
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DDNAME=SYSIN
//*
//SYSIN    DD *
%TXT2PDF BROWSE Y IN DD:INDD +
OUT 'AWS.M2.TXT2PDF2.PDF.VB' +
CC YES
/*
//*
//*-----**
//* CONVERT PDF (VB) TO PDF (LSEQ - BYTE STREAM)*
//*-----**
//*
//STEP02 EXEC PGM=VB2LSEQ
//*
//INFILE   DD DSN=AWS.M2.TXT2PDF2.PDF.VB,DISP=SHR
//*
//OUTFILE  DD DSN=AWS.M2.TXT2PDF2.PDF,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=256,DSORG=PS,RECFM=LSEQ,BLKSIZE=0)
//*

```

```
//SYSOUT DD SYSOUT=*  
//*  
//
```

## Modificaciones

Para que el programa TXT2PDF se ejecute en el entorno de ejecución Micro Focus de AWS Mainframe Modernization, realizamos los siguientes cambios:

- Cambios en el código fuente para garantizar la compatibilidad con el entorno de tiempo de ejecución REXX de Micro Focus
- Cambios para garantizar que el programa pueda ejecutarse en los sistemas operativos Windows y Linux
- Modificaciones para admitir el tiempo de ejecución en EBCDIC y ASCII

## Referencias

Referencias y código fuente del TXT2PDF:

- [Convertidor de texto a PDF](#)
- [Herramientas gratuitas de correo y TCP/IP de z/OS](#)
- [Guía de referencia para el usuario de TXT2PDF](#)

## Utilidad por lotes M2DFUTIL

M2DFUTIL es un programa de utilidad de JCL que proporciona funciones de copia de seguridad, restauración, eliminación y copia en conjuntos de datos, de forma similar al soporte que proporciona la utilidad ADRDSSU del mainframe. Este programa conserva muchos de los parámetros SYSIN de ADRDSSU, lo que agiliza el proceso de migración a esta nueva utilidad.

### Temas

- [Plataformas admitidas](#)
- [Requisitos de la plataforma](#)
- [Compatibilidad futura planificada](#)
- [Ubicaciones de los activos](#)

- [Configure M2DFUTIL o el tiempo de ejecución de modernización de AWS mainframe en Amazon EC2 \(incluida la versión 2.0\) AppStream](#)
- [Sintaxis general](#)
- [JCL de ejemplo](#)

## Plataformas admitidas

Puede utilizar M2DFUTIL en cualquiera de las siguientes plataformas:

- Micro Focus ES en Windows (64 y 32 bits)
- Micro Focus ES en Linux (64 bits)

## Requisitos de la plataforma

M2DFUTIL depende de llamar a un script para realizar una prueba de expresión regular. En Windows, debe instalar los Windows Services for Linux (WSL) para que se ejecute este script.

## Compatibilidad futura planificada

Entre las funciones que actualmente no están disponibles en la utilidad ADRDSSU del mainframe, pero que estarán disponibles en el futuro, se incluyen:

- M2 administrado
- VSAM
- Compatibilidad con COPY para cambiar el nombre de los archivos
- Compatibilidad con RENAME para RESTORE
- Múltiples INCLUDE y EXCLUDE
- Cláusula BY para la subselección mediante DSORG, CREDIT, EXPDT
- Cláusula MWAIT para reintentar errores en la cola
- Compatibilidad con el almacenamiento de S3 para DUMP/RESTORE

## Ubicaciones de los activos

El módulo de carga de esta utilidad se llama M2DFUTIL .so en Linux y M2DFUTIL .dll en Windows. Este módulo de carga se puede encontrar en las siguientes ubicaciones:

- Linux: /opt/aws/m2/microfocus/utilities/64bit
- Windows de 32 bits: C:\AWS\M2\MicroFocus\Utilities\32bit
- Windows de 64 bits: C:\AWS\M2\MicroFocus\Utilities\64bit

El script utilizado para las pruebas de expresiones regulares se llama `compare.sh`. Este script se puede encontrar en las siguientes ubicaciones:

- Linux: /opt/aws/m2/microfocus/utilities/scripts
- Windows de 32 bits: C:\AWS\M2\MicroFocus\Utilities\scripts

Configure M2DFUTIL o el tiempo de ejecución de modernización de AWS mainframe en Amazon EC2 (incluida la versión 2.0) AppStream

Configure su región de Enterprise Server con lo siguiente:

- Agregue las siguientes variables en [ES-Environment]
  - M2DFUTILS\_BASE\_LOC: la ubicación predeterminada para la salida de DUMP
  - M2DFUTILS\_SCRIPTPATH: la ubicación del script `compare.sh` documentada en Ubicaciones de activos
  - M2DFUTILS\_VERBOSE: [DETALLADO o NORMAL]. Esto controla el nivel de detalle de la salida SYSPRINT
- Compruebe que la ruta del módulo de carga se haya agregado a la configuración de JES \Configuration\JES Program Path
- Compruebe que los scripts del directorio de utilidades tengan permisos de ejecución. Puede agregar un permiso de ejecución mediante el comando `chmod + x <script name>`, en el entorno de Linux

## Sintaxis general

### DUMP

Ofrece la posibilidad de copiar archivos desde la ubicación catalogada actual a una ubicación de copia de seguridad. Actualmente, esta ubicación debe ser un sistema de archivos.

## Proceso

DUMP realizará las siguientes tareas:

1. Crear el directorio de ubicaciones de destino.
2. Catalogar el directorio de ubicaciones de destino como miembro del PDS.
3. Determinar los archivos que se van a incluir procesando el parámetro INCLUDE.
4. Anular la selección de los archivos incluidos procesando el parámetro EXCLUDE.
5. Determinar si los archivos que se van a volcar deben eliminarse con DELETED.
6. Colocar en la cola los archivos que se procesarán.
7. Copiar los archivos.
8. Exportar la información DCB catalogada de los archivos copiados a un archivo lateral en la ubicación de destino para facilitar las futuras operaciones de RESTORE.

## Sintaxis

```
DUMP
TARGET ( TARGET LOCATION ) -
INCLUDE ( DSN. )
[ EXCLUDE ( DSN ) ]
[ CANCEL | IGNORE ]
[ DELETE ]
```

## Parámetros necesarios

A continuación se enumeran los parámetros necesarios para el DUMP:

- **SYSPRINT DD NAME:** para contener información de registro adicional
- **TARGET:** ubicación de destino. Puede ser alguno de los siguientes:
  - Ruta completa de la ubicación del volcado
  - Nombre del subdirectorio creado en la ubicación definida en la variable M2DFUTILS\_BASE\_LOC
- **INCLUDE:** una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida
- **EXCLUDE:** una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida

## Parámetros opcionales

- CANCEL: se cancela si se produce algún error. Los archivos que se procesaron se conservarán
- (Predeterminado) IGNORE: ignora cualquier error y proceso hasta el final
- DELETE: si no se produce ningún error de ENQ, el archivo se elimina y no se cataloga

## DELETE

Ofrece la posibilidad de eliminar y descatalogar archivos de forma masiva. No se hace una copia de seguridad de los archivos.

## Proceso

DELETE realizará las siguientes tareas:

1. Determinar los archivos que se van a incluir procesando el parámetro INCLUDE.
2. Anular la selección de los archivos incluidos procesando el parámetro EXCLUDE.
3. Colocar en la cola los archivos que se procesarán. Establecer la disposición como OLD, DELETE, KEEP.

## Sintaxis

```
DELETE  
INCLUDE ( DSN )  
[ EXCLUDE ( DSN ) ]  
[ CANCEL | IGNORE ]  
[ DELETE ]
```

## Parámetros necesarios

A continuación aparecen los parámetros necesarios para DELETE:

- SYSPRINT DD NAME: para contener información de registro adicional
- INCLUDE: una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida
- EXCLUDE: una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida

## Parámetros opcionales

- CANCEL: se cancela si se produce algún error. Los archivos que se procesen se conservarán
- (Predeterminado) IGNORE: ignora cualquier error y proceso hasta el final

## RESTORE

Ofrece la posibilidad de restaurar archivos de los que se haya hecho una copia de seguridad previa mediante DUMP. Los archivos se restauran en la ubicación catalogada original, a menos que se utilice RENAME para modificar el DSNNAME restaurado.

### Proceso

RESTORE realizará las siguientes tareas:

1. Valide el directorio de ubicaciones de origen.
2. Determine los archivos que se van a incluir procesando el archivo de exportación del catálogo.
3. Anular la selección de los archivos incluidos procesando el parámetro EXCLUDE.
4. Colocar en la cola los archivos que se procesarán.
5. Archivos de catálogo que no están catalogados en función de su información de exportación.
6. Si un archivo ya está catalogado y la información del catálogo de exportación es la misma, RESTORE reemplazará el conjunto de datos catalogado si se ha activado la opción REPLACE.

### Sintaxis

```
RESTORE
SOURCE ( TARGET LOCATION )
INCLUDE ( DSN )
[ EXCLUDE ( DSN ) ]
[ CANCEL | IGNORE ]
[ REPLACE]
```

### Parámetros necesarios

Los siguientes son los parámetros necesarios para RESTORE:

- SYSPRINT DD NAME: para contener información de registro adicional
- SOURCE: ubicación del origen. Puede ser alguno de los siguientes:

- Ruta completa de la ubicación del volcado
- Nombre del subdirectorio creado en la ubicación definida en la variable M2DFUTILS\_BASE\_LOC
- INCLUDE: una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida
- EXCLUDE: una cadena de búsqueda DSNAME única con nombre o una cadena de búsqueda DSN de mainframe válida

### Parámetros opcionales

- CANCEL: se cancela si hay algún error. Archivos procesados retenidos
- (Predeterminado) IGNORE: ignora cualquier error y proceso hasta el final
- REPLACE: si el archivo que se está restaurando ya está catalogado y los registros del catálogo son los mismos, sustituye el archivo catalogado

### JCL de ejemplo

#### Trabajo DUMP

Este trabajo creará un subdirectorio llamado TESTDUMP. Esta es la ubicación de copia de seguridad predeterminada especificada por la variable M2DFUTILS\_BASE\_LOC. Creará una biblioteca PDS para esta copia de seguridad llamada M2DFUTILS.TESTDUMP. Los datos del catálogo exportados se almacenan en un archivo secuencial de líneas en el directorio de copias de seguridad denominado CATDUMP.DAT. Todos los archivos seleccionados se copiarán en este directorio de copia de seguridad.

```
//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
/*
//
```

#### Trabajo DELETE



Este trabajo eliminará todos los archivos del catálogo que coincidan con el parámetro INCLUDE.

```

/M2DFDEL JOB 'M2DFDEL',CLASS=A,MSGCLASS=X
//STEP001 EXEC PGM=M2DFUTIL
//SYSPRINT DD DSN=TESTDEL.SYSPRINT,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=LSEQ,LRECL=256)
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  DELETE                                     -
      INCLUDE(TEST.FB.FILE*.ABC)           -
CANCEL
/*
//

```

## Trabajo RESTORE

Este trabajo restaurará los archivos que coincidan con el parámetro INCLUDE desde la ubicación de la copia de seguridad de TESTDUMP. Los archivos catalogados se reemplazarán si el archivo catalogado es el mismo que el de la exportación a CATDUMP y se especifica la opción REPLACE.

```

//M2DFREST JOB 'M2DFREST',CLASS=A,MSGCLASS=X
//STEP001 EXEC PGM=M2DFUTIL
////SYSPRINT DD DSN=TESTREST.SYSPRINT,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=LSEQ,LRECL=256)
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
RESTORE SOURCE(TESTDUMP)                   -
      INCLUDE(TEST.FB.FILE*.ABC)           -
IGNORE
REPLACE
/*
//

```

## Utilidad por lotes M2RUNCMD

Puede utilizar M2RUNCMD, un programa de utilidad por lotes, para ejecutar comandos, scripts y llamadas al sistema de Micro Focus directamente desde JCL, en lugar de ejecutarlos desde una terminal o una línea de comandos. El resultado de los comandos se registra en el registro spool del trabajo por lotes.

## Temas

- [Plataformas admitidas](#)
- [Configuración de M2RUNCMD para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 \(incluida la versión 2.0\) AppStream](#)
- [JCL de ejemplo](#)

## Plataformas admitidas

Puede utilizar M2RUNCMD en las siguientes plataformas:

- Micro Focus Runtime (en Amazon EC2)
- Todas las variantes de los productos Micro Focus Enterprise Developer (ED) y Micro Focus Enterprise Server (ES).

## Configuración de M2RUNCMD para el tiempo de ejecución de modernización AWS del mainframe en Amazon EC2 (incluida la versión 2.0) AppStream

Si las aplicaciones migradas se ejecutan en el entorno de ejecución de modernización de AWS mainframe en Amazon EC2, configure M2RUNCMD de la siguiente manera.

- Cambie la [ruta del programa JES de Micro Focus](#) para incluir la ubicación binaria de las utilidades por lotes. Si necesita especificar varias rutas, utilice dos puntos (:) para separarlas en Linux y punto y coma (;) en Windows.
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bits): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bits): C:\AWS\M2\MicroFocus\Utilities\64bit

## JCL de ejemplo

Para probar la instalación, puede utilizar uno de los siguientes archivos JCL de ejemplo.

RUNSCRL1.jcl

En este JCL de ejemplo se crea un script y se ejecuta. El primer paso crea un script denominado /tmp/TEST\_SCRIPT.sh y con contenido a partir de datos de SYSUT1 internos. El segundo paso establece el permiso de ejecución y ejecuta el script creado en el primer paso. También puede

optar por realizar solo el segundo paso para ejecutar los comandos de Micro Focus y del sistema ya existentes.

```
//RUNSCL1 JOB 'RUN SCRIPT',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
/*-----*
/*  CREATE SCRIPT (LINUX)
/*-----*
//*
//STEP0010 EXEC PGM=IEBGENER
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//SYSUT1   DD *
#!/bin/bash

set -x

## ECHO PATH 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

El resultado del comando o script que se ejecuta se escribe en el registro SYSOUT. Para cada comando ejecutado, muestra el comando, la salida y el código de retorno.

```

***** 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

Este ejemplo de JCL usa RUNCMD para ejecutar varios comandos.

```
//RUNCMDL1 JOB 'RUN CMD',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
//*-----*
//*  RUN SYSTEM COMMANDS                               *
//*-----*
//*
//STEP0001 EXEC PGM=RUNCMD
//*
//SYSOUT DD SYSOUT=*
//*
//SYSIN  DD *
*LIST DIRECTORY
  ls
*ECHO PATH ENVIRONMNET VARIABLE
  echo $PATH
/*
//
```

# AWS Modernización de mainframe y replicación de datos con Precisely

AWS Mainframe Modernization ofrece una variedad de Amazon Machine Images (AMI). Estas AMI facilitan el aprovisionamiento rápido de las instancias de Amazon EC2, lo que crea un entorno personalizado para la replicación de datos desde sistemas AWS mainframe hasta el uso de Precisely. En esta guía se proporcionan los pasos necesarios para acceder a estas AMI y usarlas.

## Requisitos previos

- Asegúrese de tener acceso de administrador a una AWS cuenta en la que pueda crear instancias de Amazon EC2.
- Compruebe que el servicio de modernización del AWS mainframe esté disponible en la región en la que planea crear las instancias de Amazon EC2. Consulte la [lista de servicios de AWS disponibles por región](#).
- Identifique la Amazon Virtual Private Cloud (Amazon VPC) donde se van a crear las instancias de Amazon EC2.
- Al crear instancias de Amazon EC2 en una Amazon VPC, asegúrese de que la tabla de enrutamiento asociada tenga una puerta de enlace de Internet o una puerta de enlace NAT.

### Note

La replicación de datos correcta requiere que la instancia EC2 de AWS tenga acceso de comunicación a AWS Marketplace. Si hay algún problema de conectividad con AWS Marketplace, el proceso de replicación fallará.

## Suscripción a imágenes de máquina de Amazon

Al suscribirse a un producto del AWS Marketplace, puede lanzar una instancia desde la AMI del producto.

1. [Inicie sesión en la AWS Marketplace consola AWS Management Console y ábrala en https://console.aws.amazon.com/marketplace](https://console.aws.amazon.com/marketplace).

2. Elija Administrar suscripciones.
3. Copie y pegue el siguiente enlace en la barra de direcciones del navegador: <https://aws.amazon.com/marketplace/pp/prodview-en3xrbgzbs3dk>
4. Elija Continuar con la suscripción.
5. Si los términos y condiciones son aceptables, seleccione Aceptar términos. El proceso de suscripción puede tardar unos minutos.
6. Espere a que aparezca el mensaje de agradecimiento, como se muestra a continuación. Este mensaje confirma que se ha suscrito correctamente al producto.



## AWS Mainframe Modernization service Data Replication with Precisely

Thank you for subscribing to this product! You can now configure your software.

7. En el panel de navegación izquierdo, elija Administrar suscripciones. Esta vista muestra todas las suscripciones a las que se ha suscrito.

## Lance la replicación de datos AWS de Mainframe Modernization con Precisely

1. Abra la AWS Marketplace consola en <https://console.aws.amazon.com/marketplace>.
2. En el panel de navegación izquierdo, elija Administrar suscripciones.
3. Busque la AMI que desee lanzar y elija Lanzar nueva instancia.
4. En Región, seleccione la región incluida en la lista de permitidos.
5. Elija Continuar para lanzar mediante EC2. Esta acción le llevará a la consola de Amazon EC2.
6. Escriba un nombre del servidor.
7. Seleccione un tipo de instancia que se ajuste a los requisitos de rendimiento y costo de su proyecto. El punto de partida sugerido para el tamaño de la instancia es c5.2xLarge.
8. Elija un par de claves existente o cree y guarde uno nuevo. Para obtener información sobre los pares de claves, consulte los [pares de claves de Amazon EC2 y las instancias de Linux](#) en la Guía del usuario de Amazon EC2.
9. Edite la configuración de red y elija la VPC de la lista de permitidos y la subred adecuada.

10. Seleccione un grupo de seguridad existente o cree uno nuevo. Además de permitir el acceso SSH (de forma predeterminada en el puerto 22), para la replicación de datos con una instancia EC2 de Precisely Server, lo habitual es permitir que el tráfico TCP llegue a su puerto predeterminado 2626.
11. Configure el almacenamiento de la instancia de Amazon EC2.
12. Revise el resumen y seleccione Lanzar instancia. Para que el lanzamiento se realice correctamente, el tipo de instancia debe ser válido. Si se produce un error en el lanzamiento, seleccione Editar la configuración de la instancia y elija un tipo de instancia distinto.
13. Cuando vea el mensaje de éxito, seleccione Conectarse a la instancia.
14. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
15. En el panel de navegación, bajo Instancias elija Instancias.
16. En el panel principal, compruebe el estado de la instancia.

## Creación de una política de IAM

Para utilizar correctamente las instancias EC2 de modernización de AWS mainframe implementadas a través de nuestra AWS Marketplace lista, debe configurar una función y una política de IAM. Esta configuración de IAM específicamente diseñada no es opcional; autoriza a las instancias de Amazon EC2 a interactuar con el servicio. AWS Marketplace La función y la política de IAM permiten a AWS Mainframe Modernization registrar con precisión los datos de uso, lo cual es esencial para una facturación precisa. Si no se implementa esta configuración, es posible que se produzcan intentos fallidos de replicación de datos e interrupciones operativas.

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación de la izquierda, elija Políticas.
3. Si es la primera vez que elige Políticas, aparecerá la página Bienvenido a políticas administradas. Elija Comenzar.
4. En la parte superior de la página, seleccione Crear política.
5. En la sección Editor de políticas, seleccione la opción JSON.
6. Introduzca la siguiente política JSON:

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



```
        "Action": ["aws-marketplace:MeterUsage"],
        "Effect": "Allow",
        "Resource": "*"
    }
]
}
```

## Creación de un rol de IAM

1. Abra la consola de IAM en <https://console.aws.amazon.com/iam/>.
2. En el panel de navegación, seleccione Roles y luego seleccione Crear rol.
3. En la sección Tipo de entidad de confianza, elija Servicio de AWS .
4. En la sección Caso de uso, en Servicio o caso de uso, elija Amazon EC2.
5. Elija Siguiente.
6. En la lista de políticas, seleccione Gestionado por el cliente en el menú desplegable Filtrar por tipo e introduzca el nombre de la política que creó. Seleccione la casilla de verificación que hay junto al nombre de la política.
7. Elija Siguiente.
8. Indique un nombre para el rol y, opcionalmente, una descripción.
9. Revise la política de confianza y los permisos y, a continuación, seleccione Crear rol.

## Asociación del rol de IAM a la instancia de Amazon EC2

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el panel de navegación, seleccione Instancias.
3. Seleccione la instancia de Amazon EC2.
4. En el menú Acciones, elija Seguridad, Modificar rol de IAM.
5. Seleccione el rol que desee asociar a la instancia y luego elija Actualizar rol de IAM.

# Integración de Charon

## Introducción a Charon-SSP

En 1987, Sun Microsystems lanzó el procesador SPARC V7, un procesador RISC de 32 bits. Le siguió el SPARC V8 en 1990, una revisión del SPARC V7 original, cuya inclusión más notable fueron las instrucciones de dividir y multiplicar por hardware. Los procesadores SPARC V8 sirvieron de base para varios servidores y estaciones de trabajo, como las SPARCstation 5, 10 y 20. En 1993, al SPARC V8 le siguió el procesador SPARC V9 de 64 bits. Esta fue también la base de varios servidores y estaciones de trabajo, como los Enterprise 250 y 450.

Debido a la obsolescencia del hardware y a la falta de piezas de repuesto o reacondicionadas, el software y los sistemas desarrollados para estas antiguas estaciones de trabajo y servidores basados en SPARC son cada vez más difíciles de mantener. Para cubrir la necesidad continua de determinados sistemas end-of-life basados en SPARC, Stromasys S.A. desarrolló la línea Charon-SSP de productos emuladores de SPARC. Los siguientes productos son sustitutos de máquinas virtuales basados en software para los sistemas SPARC de hardware nativo especificados. A continuación presentamos una descripción general de las gamas de hardware emulado.

Charon-SSP/4M emula el siguiente hardware SPARC:

- Gama Sun-4m (representada por la Sun SPARCstation 20): originalmente, era una variante Sun-4 multiprocesador, basada en el bus de módulos del procesador MBus introducido en la serie SPARCServer 600MP. Posteriormente, la arquitectura de la Sun-4m también incluyó sistemas uniprocador que no eran MBus, como el SPARCstation 5, que utilizaba procesadores de arquitectura SPARC V8. Compatible a partir de SunOS 4.1.2 y de Solaris 2.1 a Solaris 9. La compatibilidad con SPARCServer 600MP se suspendió después de Solaris 2.5.1.

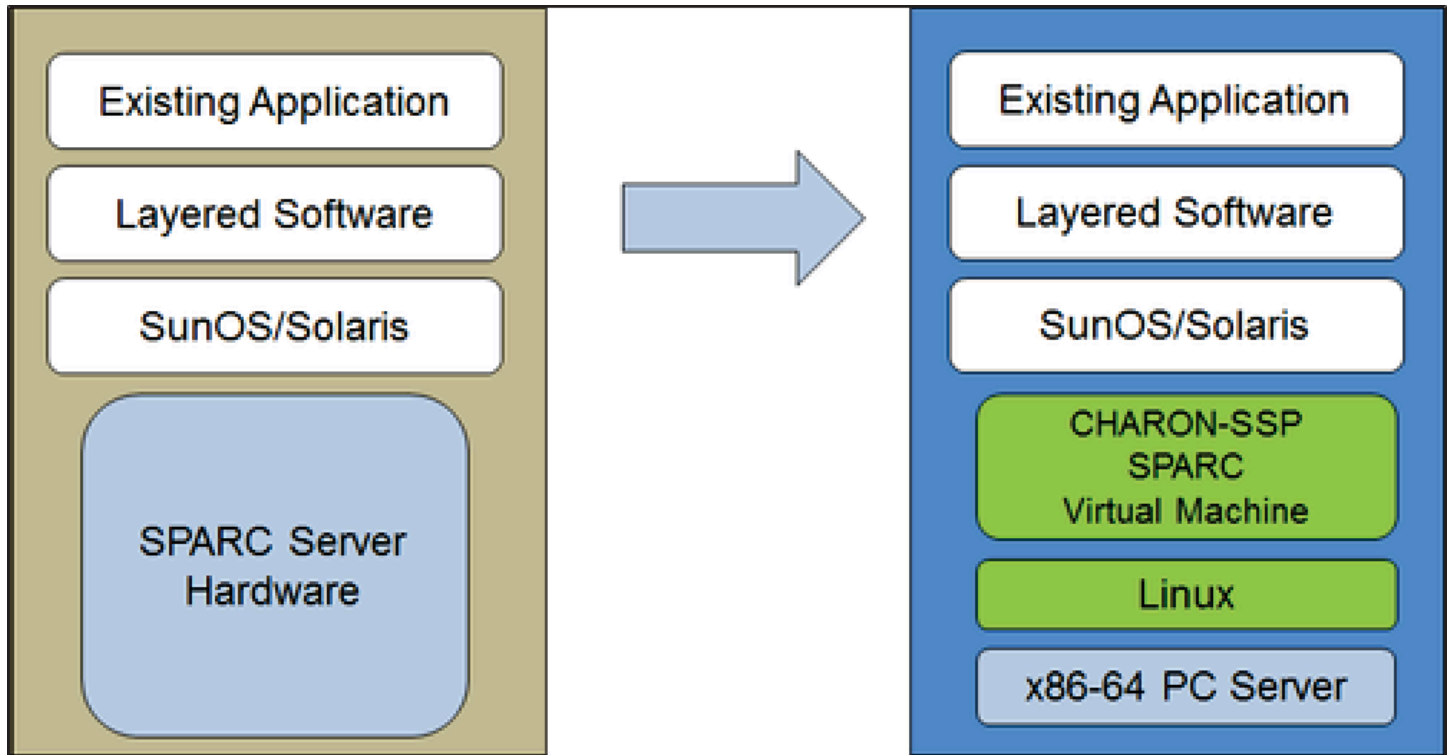
Charon-SSP/4U(+) emula el siguiente hardware SPARC:

- Gama Sun-4u (representada por el Sun Enterprise 450): (U para UltraSPARC): esta variante introdujo la arquitectura del procesador SPARC V9 de 64 bits y la interconexión de procesadores UPA utilizada por primera vez en la serie Sun Ultra. Compatible con las versiones de 32 bits de Solaris a partir de la versión 2.5.1 La primera versión de Solaris de 64 bits para Sun-4u fue la Solaris 7. La compatibilidad con UltraSPARC I se eliminó tras Solaris 9. Solaris 10 admite las implementaciones de Sun-4u, desde UltraSPARC II hasta UltraSPARC IV.

Charon-SSP/4V(+) emula el siguiente hardware SPARC:

- Gama Sun-4v (representada por los SPARC T2 y T4): esta variante incorporó la virtualización del procesador con hipervisor al Sun-4u; se introdujo en el procesador multinúcleo Ultra SPARC T1. El hardware seleccionado era compatible con la versión de Solaris 10 a partir de la versión 3/05 HW2 (la mayoría de los modelos, incluido el hardware emulado por Charon-SSP, requieren versiones más recientes de Solaris 10). También se admiten varias versiones de Solaris 11.

La siguiente imagen muestra el concepto básico de la migración de hardware físico a un emulador.



Las máquinas virtuales Charon-SSP permiten a los usuarios de equipos basados en Sun y Oracle SPARC reemplazar su hardware nativo de una manera que requiere pocos o ningún cambio en la configuración original del sistema. Esto significa que puede seguir ejecutando sus aplicaciones y datos sin necesidad de cambiar o migrar a otra plataforma. El software Charon-SSP se ejecuta en sistemas Intel de 64 bits básicos, lo que garantiza la protección continua de su inversión.

El Charon-SSP/4U+ es compatible con las mismas plataformas SPARC virtuales que el Charon-SSP/4U, y el Charon-SSP/4V+ lo mismo que el Charon-SSP/4V. Sin embargo, las versiones 4U+ y 4V+ aprovechan la tecnología de virtualización asistida por hardware VTx/EPT de Intel y AMD-v/NPT de AMD en las CPU modernas para ofrecer un mejor rendimiento de la CPU virtual. Los Charon-SSP/4U+ y Charon-SSP/4V+ requieren CPU compatibles con VT-x/EPT o AMD-v/NPT y deben

instalarse en un sistema de host dedicado. No se admite la ejecución de estas variantes de producto en una máquina virtual (por ejemplo, en VMware).

 Note

Si planea ejecutar Charon-SSP/4U+ o 4V+ en un entorno de nube, póngase en contacto con Stromasys o con un VAR de Stromasys para analizar sus requisitos.

## Sistemas operativos huéspedes compatibles

Las máquinas virtuales Charon-SSP/4M son compatibles con las siguientes versiones de sistemas operativos huéspedes:

- SunOS 4.1.3 - 4.1.4
- De Solaris 2.3 a Solaris 9

Las máquinas virtuales Charon-SSP/4U(+) son compatibles con las siguientes versiones de sistemas operativos huéspedes:

- De Solaris 2.5.1 a Solaris 10

Las máquinas virtuales Charon-SSP/4V(+) son compatibles con las siguientes versiones de sistemas operativos huéspedes:

- Solaris 10 (a partir de la actualización 4, 08/07) y de Solaris 11.1 a Solaris 11.4

Para Charon-SSP/4V(+), tenga en cuenta lo siguiente:

- Para el SPARC T4 emulado, las versiones de Solaris 10 compatibles son: Oracle Solaris 10 1/13, Oracle Solaris 10 8/11 y Solaris 10 9/10, o Solaris 10 10/09 con el set de parches Oracle Solaris 10 8/11.
- El modelo SPARC T4 emulado es un requisito previo para ejecutar Solaris 11.4 en el emulador.
- No se admiten las zonas de núcleos de Solaris.

## Requisitos previos de la instancia en la nube de Charon-SSP

Al seleccionar un tipo o forma de instancia, seleccione el hardware virtual que se utilizará para la instancia de host de Charon-SSP en la nube. Por lo tanto, la selección de un tipo o forma de instancia determina las características del hardware del host virtual de Charon-SSP (por ejemplo, cuántos núcleos de CPU y cuánta memoria tendrá su sistema de host virtual Charon).

### Note

Si utiliza una imagen de mercado de Charon-SSP para lanzar la instancia, se cumplen todos los requisitos del sistema operativo de host Linux.

Los requisitos mínimos de hardware se describen a continuación.

Puntos importantes relacionados con las pautas de tamaño:

- Las siguientes pautas de tamaño, en particular en lo que respecta al número de núcleos de CPU del host y la memoria del host, ilustran los requisitos mínimos. Se debe revisar cada situación de implementación y el tamaño real del host debe adaptarse según sea necesario. Por ejemplo, se debe aumentar la cantidad de núcleos de CPU disponibles para la E/S si las aplicaciones del huésped producen una carga de E/S elevada. Además, un sistema con muchas CPU emuladas suele ser capaz de generar una mayor carga de E/S y, por lo tanto, puede que sea necesario aumentar la cantidad de núcleos de CPU disponibles para la E/S. En un entorno de Hyper-Threading, para obtener el mejor rendimiento, la cantidad de núcleos de CPU (es decir, CPU reales o físicas) debe ser suficiente para cumplir con los requisitos de CPU de los emuladores activos, lo que evita que los subprocesos con altas cargas de trabajo compartan un núcleo de CPU físico.
- La configuración determina la asignación de núcleos de CPU para las CPU emuladas y los núcleos de CPU para el procesamiento de E/S. Consulte la configuración de la CPU en la guía general del usuario de Charon-SSP para obtener más información sobre esto y la asignación predeterminada de los núcleos de CPU para el procesamiento de E/S.

### Información general importante

- Para facilitar la transferencia rápida de los datos del emulador de una instancia en la nube a otra, se recomienda encarecidamente almacenar todos los datos relevantes del

emulador en un volumen de disco independiente que pueda separarse fácilmente de la instancia anterior y adjuntarse a una nueva instancia.

- Asegúrese de dimensionar la instancia correctamente desde el principio (consulte los requisitos mínimos que se indican a continuación). La licencia de Charon-SSP para Charon-SSP AL se crea cuando la instancia se lanza por primera vez. Si se cambia posteriormente a otro tamaño/tipo de instancia y, por lo tanto, se modifica el número de núcleos de la CPU, se invalidará la licencia y, por lo tanto, se impedirá que las instancias de Charon se inicien (se requiere una nueva instancia). Si planea utilizar la instancia de Charon-SSP AL en modo AutoVE, asegúrese de incluir la información del servidor de AutoVE antes del primer lanzamiento; de lo contrario, se utilizarán los servidores de licencias públicas. La licencia de Charon-SSP VE se crea en función de la huella digital capturada en el servidor de licencias. Si el servidor de licencias se ejecuta directamente en el host del emulador y, posteriormente, el host del emulador requiere, por ejemplo, un cambio en el número de núcleos de la CPU, la licencia se invalidará (se necesitará una nueva licencia y, posiblemente, una nueva instancia).

## Requisitos previos para las instancias

Requisitos generales de CPU: Charon-SSP es compatible con procesadores de arquitectura x86-64 modernos basados en instancias de Amazon EC2.

Requisitos mínimos para Charon-SSP:

- Número mínimo de núcleos de CPU del sistema host:
  - Al menos un núcleo de CPU para el sistema operativo host, además de:
  - Para cada sistema SPARC emulado:
    - Un núcleo de CPU por cada CPU emulada de la instancia, además de:
    - Al menos un núcleo de CPU adicional para el procesamiento de E/S (al menos dos, si se utiliza la optimización JIT del servidor). Consulte la sección de configuración de la CPU mencionada anteriormente para ver las opciones de configuración. De forma predeterminada, Charon asignará al procesamiento de E/S 1/3 (mínimo 1, redondeado a la baja) del número de CPU visibles para el host de Charon.
- Requisitos mínimos de memoria:
  - 4 GB o más de RAM para el sistema operativo del host Linux. Los requisitos reales pueden ser mayores y dependerán de los requisitos de los servicios no emuladores que se ejecuten en el

host Linux. La recomendación anterior de al menos 2 GB de RAM para el host Linux seguirá siendo válida para muchos sistemas, pero los requisitos cada vez mayores del sistema operativo y las aplicaciones Linux han llevado a actualizar la recomendación para nuevas instalaciones.

Además:

- Para cada sistema SPARC emulado:
  - La memoria configurada de la instancia emulada, además de:
  - 2 GB de RAM (6 GB de RAM si se utiliza el JIT del servidor) para permitir la optimización del DIT, los requisitos del emulador, los búferes en tiempo de ejecución, el SMP y la emulación de gráficos.
- Si la tecnología Hyper-Threading está habilitada en las CPU x86-64 modernas, se pueden ejecutar dos subprocesos en un núcleo de CPU físico, lo que proporciona dos CPU lógicas al sistema operativo host. Si es posible, deshabilite el Hyper-Threading en el host Charon-SSP. Sin embargo, esto no suele ser posible en VMware y en los entornos de nube, o no está claro si se utiliza el Hyper-Threading o no. La opción de Hyper-Threading de Charon-SSP permite a Charon-SSP adaptarse a dichos entornos. Consulte la sección de configuración de la CPU en la guía general del usuario de Charon-SSP mencionada anteriormente para obtener información detallada sobre la configuración. Tenga en cuenta que, para obtener el mejor rendimiento, los subprocesos Charon-SSP no deben compartir un núcleo de CPU físico; debe haber suficientes núcleos físicos en el sistema host para cumplir con los requisitos de los emuladores configurados.
- Una o más instancias de red, en función de los requisitos del cliente.
- Charon-SSP/4U+ y Charon-SSP/4V+ deben ejecutarse en hardware físico compatible con los procesadores Intel VT-x/EPT o AMD-v/NPT (instancias básicas) y, por lo tanto, no pueden ejecutarse en todos los entornos de nube. Consulte la documentación de su proveedor de servicios en la nube para conocer la disponibilidad de dicho hardware. Además, tenga en cuenta los siguientes puntos:
  - Charon-SSP/4U+ y Charon-SSP/4V+ solo están disponibles cuando se utiliza un núcleo de Linux compatible con Stromasys.
  - Si necesita este tipo de hardware SPARC emulado, póngase en contacto con Stromasys o con su VAR de Stromasys para analizar sus requisitos en detalle.

# Creación y configuración de una instancia en la nube para Charon (nueva AWS GUI)

Esta sección refleja lo ocurrido AWS Management Console en la primavera de 2022. Si aún utilizas la consola anterior, consulta el apéndice de la guía de introducción de Charon-SSP AWS .

## Requisitos previos generales

Esta descripción muestra la configuración básica de una instancia de Linux en AWS. No incluye requisitos previos específicos. Sin embargo, en función de su caso de uso, tenga en cuenta los siguientes requisitos previos:

- Cuenta y AWS Marketplace suscripciones de Amazon
  - Para configurar una instancia de Linux AWS, necesitas una AWS cuenta con acceso de administrador.
  - Identifique la AWS región en la que planea lanzar la instancia. Asegúrese de que los servicios de AWS que planea usar estén disponibles en esa región. Consulte [Servicios de AWS por región](#).
  - Identifique la VPC en la que planea lanzar la instancia.
  - Si la instancia requiere acceso a Internet, asegúrese de que la tabla de enrutamiento asociada a la VPC tenga una puerta de enlace de Internet. Si su instancia requiere acceso mediante VPN a su red en las instalaciones, asegúrese de que haya una puerta de enlace VPN disponible. La configuración exacta de la VPC y sus subredes dependerá del diseño de la red y de los requisitos de la aplicación.
  - Para suscribirse a un AWS Marketplace servicio específico, elija Suscripciones a AWS Marketplace en AWS Management Console y, a continuación, elija Administrar suscripciones.
  - Busque el servicio que planea usar y suscríbase a él. Cuando la suscripción se haya realizado correctamente, la encontrará en la sección Administrar suscripciones. Desde allí, puede lanzar directamente una nueva instancia.
- Los requisitos previos de hardware y software de la instancia variarán en función del uso previsto para la instancia:
  - Opción 1: la instancia se utilizará como sistema host del emulador de Charon:
    - Consulte las secciones de requisitos previos de hardware y software de la guía del usuario o la guía de introducción de su producto Charon para determinar los requisitos previos exactos de hardware y software que debe cumplir la instancia de Linux. La imagen que utilice para



lanzar la instancia y el tipo de instancia que elija determinan el software y el hardware de la instancia en la nube.

- Se requiere una licencia de producto de Charon para ejecutar sistemas emulados heredados. Consulte la información de licencia en la documentación de su producto Charon o póngase en contacto con su representante de Stromasys o con el VAR de Stromasys para obtener más información.
- Opción 2: la instancia se utilizará como servidor de licencias de VE dedicado:
  - Consulte la guía del servidor de licencias de VE para conocer los requisitos previos detallados.
- Algunos sistemas operativos heredados que pueden ejecutarse en los sistemas emulados proporcionados por los productos emuladores de Charon requieren una licencia del proveedor original del sistema operativo. El usuario es responsable de cualquier obligación de licencia relacionada con el sistema operativo heredado y debe proporcionar las licencias correspondientes.

## AWS Management Console Utilizándolo para lanzar una nueva instancia


Para crear una nueva instancia

1. [Inicie sesión en la consola Amazon EC2 AWS Management Console y ábrala en https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. Seleccione Iniciar instancia.
3. Escriba el nombre de la instancia.
4. Seleccione una AMI. Una AMI es una imagen preempaquetada que se utiliza para lanzar instancias en la nube. Incluye el sistema operativo y el software de aplicación correspondiente. La elección de la AMI depende de cómo vaya a utilizar la instancia:
  - Si la instancia se va a utilizar como sistema host del emulador de Charon, son posibles varias opciones de AMI:
    - Instalación del sistema del host de Charon a partir de una imagen de mercado de Charon preempaquetada: contienen el sistema operativo subyacente y el software de Charon preinstalado.
    - Consulte con su representante de Stromasys qué opciones están disponibles actualmente en su mercado de proveedores de servicios en la nube.
    - Según el proveedor de servicios en la nube y los planes de lanzamiento de los productos de Stromasys, puede haber dos variantes:

- Licencias automáticas (AL) para su uso con un servidor de licencias público operado por Stromasys o con un servidor de licencias AutoVE privado operado por el cliente
- Entorno virtual (VE) para su uso con un servidor de licencias de VE privado operado por el cliente
- Instalación del sistema host de Charon mediante una instalación de emulador de Charon convencional con los paquetes RPM de instalación del emulador de Charon para Linux:
  - Elija una AMI de Linux de una distribución compatible con el producto y la versión de Charon que haya seleccionado. Consulte la guía del usuario de su producto en el sitio de documentación de Stromasys.
- Si la instancia se va a utilizar como un servidor de licencias de VE dedicado, consulte la Guía del servidor de licencias de VE en la documentación de licencias para conocer los requisitos de la instancia de Linux.

Después de decidir qué AMI es necesaria, seleccione una AMI de producto de Linux o Charon coincidente. Si no ve la AMI que necesita aquí, elija Buscar más AMI. Elija la AMI de Linux que coincida con la forma en que tenga previsto usar la instancia. Puede ser una de las siguientes opciones:


- Una imagen de mercado de Charon VE preempaquetada. El nombre de la AMI incluirá la cadena "ve".
  - Una imagen de mercado de Charon AL preempaquetada para Automatic Licensing o AutoVE.
  - Una versión de Linux compatible con la instalación de un producto de RPM.
  - Una versión de Linux compatible con el servidor de licencias de VE.
5. Seleccione un tipo de instancia. Amazon EC2 ofrece tipos de instancias con distintas combinaciones de CPU, memoria, almacenamiento y capacidad de red. Seleccione un tipo de instancia que coincida con los requisitos del producto Charon que desee utilizar. Algunas imágenes de mercado tienen una selección restringida de tipos de instancias.
  6. Seleccione un par de claves existente o cree y guarde uno nuevo. Si selecciona un par de claves existente, asegúrese de tener la clave privada correspondiente. De lo contrario, no podrá conectarse a su instancia.

 Note

Si su sistema de administración lo admite, para RHEL 9.x, Rocky Linux 9.x y Oracle Linux 9.x, utilice la clave SSH de tipo ECDSA o ED25519. Estos tipos le permiten

conectarse a estos sistemas de host Linux de Charon mediante un túnel SSH sin necesidad de cambiar la configuración de política criptográfica predeterminada del host Charon por una configuración menos segura. Por ejemplo, esto es importante para el administrador de Charon-SSP. Consulte [Uso de políticas criptográficas en todo el sistema](#) en la documentación de Red Hat.

7. En la sección Configuración de red, elija Editar. Elija la configuración que corresponda a su entorno.
  - Especifique una VPC.
  - Especifique una subred existente o cree una nueva.
  - Habilite o deshabilite la asignación automática de una dirección IP pública a la interfaz principal. La asignación automática solo es posible si la instancia tiene una única interfaz de red.
  - Asigne un grupo de seguridad personalizado nuevo o existente. El grupo de seguridad debe permitir al menos el acceso de SSH a la instancia. También se deben permitir todos los puertos que requieran las aplicaciones que planea ejecutar en la instancia. Puede modificar el grupo de seguridad en cualquier momento después de crear la instancia.
8. En la sección Almacenamiento, para el volumen raíz (el disco del sistema), elija un tamaño adecuado para su entorno. El tamaño mínimo de disco de sistema recomendado para el sistema Linux es de 30 GiB. Para proporcionar espacio para los contenedores de discos virtuales y otros requisitos de almacenamiento, puede agregar más almacenamiento ahora o después de lanzar la instancia. Sin embargo, el tamaño del disco del sistema debe cumplir los requisitos del sistema Linux, incluidas las aplicaciones y utilidades que planea instalar.

 Note

Le recomendamos que cree volúmenes de almacenamiento independientes para los datos de las aplicaciones de Charon (por ejemplo, imágenes de disco). Si es necesario, más adelante podrá migrar dichos volúmenes a otra instancia.

9. Expanda la sección Detalles avanzados y active la casilla Especificar opciones de CPU. En la siguiente imagen se muestran a modo de ejemplo tres que tienen más probabilidades de ser útiles en un entorno de emulador de Charon.



Specify CPU options

Core count

2

Threads per core

2

Number of vCPUs

4

10. En el caso de un sistema de servidor de licencias de VE con una versión anterior a la 1.1.23, debe asignar el rol de IAM necesario a la instancia. Debe ser un rol que permita la acción `ListUsers`. Para asignar un rol, en la sección ampliada Detalles avanzados, seleccione un rol en Perfil de instancia de IAM o elija Crear un nuevo perfil de IAM. Para obtener más información, consulte [Roles de IAM para Amazon EC2](#).
11. Si su instancia está basada en una AWS Marketplace imagen de Charon AL y planea usar los servidores de licencias públicas operados por Stromasys, debe agregar la información correspondiente a la configuración de la instancia antes de lanzarla.

Introduzca la información del servidor de licencias de AutoVE, tal y como se muestra en la siguiente imagen.

The screenshot shows a configuration interface with the following sections:

- Metadata accessible Info**: A dropdown menu set to "Enabled".
- Metadata version Info**: A dropdown menu set to "V1 and V2 (token optional)".
- Metadata response hop limit Info**: A dropdown menu set to "Select".
- Allow tags in metadata Info**: A dropdown menu set to "Select".
- User data Info**: A text input field containing the text "primary\_server=172.31.34.235:8083".
- User data has already been base64 encoded**

Las siguientes opciones de configuración son válidas:

- **primary\_server**=<ip-address>[:<port>]
- **backup\_server**=<ip-address>[:<port>]

Donde

- <ip-address> representa la dirección IP del servidor principal y del servidor de copia de seguridad, según corresponda.
- <port> representa un puerto TCP no predeterminado que se utiliza para comunicarse con el servidor de licencias (predeterminado: TCP/8083).

**Note**

Se debe configurar al menos un servidor de licencias en el lanzamiento inicial para habilitar el modo AutoVE. De lo contrario, la instancia se vinculará a uno de los servidores de licencias públicos operados por Stromasys.

12. En el panel Resumen, elija Lanzar instancia. Después de un tiempo, verá el siguiente mensaje de éxito:

The screenshot shows the AWS Management Console interface. At the top, a dark navigation bar contains a hamburger menu icon, the text 'EC2 > Instances > Launch an instance', and a search icon. Below this is a green success notification banner with a checkmark icon, the text 'Success', and 'Successfully initiated launch of instance (i-0130460018e01000d)'. Underneath the banner is a 'Launch log' section with a right-pointing arrow. Below the log is a 'Next Steps' section with a search bar containing the text 'What would you like to do next with this instance, for example "create alarm" or "create backup"'. At the bottom, there are two white cards. The left card is titled 'Create billing and free tier usage alerts' and contains the text 'To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.' and a blue button labeled 'Create billing alerts' with an external link icon. The right card is titled 'Connect to your instance' and contains the text 'Once your instance is running, log into it from your local computer.' and a blue button labeled 'Connect to instance' with an external link icon. Below this button is a link labeled 'Learn more' with an external link icon.

13. En la esquina inferior derecha de la pantalla, elija Ver todas las instancias.
14. Para ver los detalles de la instancia, seleccione la casilla de verificación situada a la izquierda de la fila que representa la instancia en la tabla Instancias. Los detalles de la instancia se mostrarán en la parte inferior de la pantalla. Para obtener información sobre cómo conectarse a la instancia, consulte [Connect](#) en la Guía del usuario de Amazon EC2.

# AWS Modernización de mainframe y cambio de plataforma con NTT DATA

AWS Mainframe Modernization ofrece una variedad de Amazon Machine Images (AMI). Estas AMI facilitan el aprovisionamiento rápido de las instancias de Amazon EC2, lo que crea un entorno personalizado para el realojamiento y la replataforma de las aplicaciones de mainframe mediante NTT Data. AWS En esta guía se proporcionan los pasos necesarios para acceder a estas AMI y usarlas.

## Requisitos previos

- Asegúrese de tener acceso de administrador a una AWS cuenta en la que pueda crear instancias de Amazon EC2.
- Compruebe que el servicio de modernización del AWS mainframe esté disponible en la región en la que planea crear las instancias de Amazon EC2. Consulte la [lista de servicios de AWS disponibles por región](#).
- Identifique la Amazon VPC en la que desee crear las instancias de Amazon EC2.

## Suscripción a imágenes de máquina de Amazon

Al suscribirse a un producto del AWS Marketplace, puede lanzar una instancia desde la AMI del producto.

1. [Inicie sesión en la AWS Marketplace consola AWS Management Console y ábrala en https://console.aws.amazon.com/marketplace](https://console.aws.amazon.com/marketplace).
2. Elija Administrar suscripciones.
3. Copie y pegue el siguiente enlace en la barra de direcciones del navegador: <https://aws.amazon.com/marketplace/pp/prodview-eg227ymlsdx2>
4. Elija Continuar con la suscripción.
5. Si los términos y condiciones son aceptables, seleccione Aceptar términos. El proceso de suscripción puede tardar unos minutos.
6. Espere a que aparezca un mensaje de agradecimiento. Este mensaje confirma que se ha suscrito correctamente al producto.

7. En el panel de navegación izquierdo, elija Administrar suscripciones. Esta vista muestra todas las suscripciones.

## Lance la replataforma AWS de modernización del mainframe con una instancia de NTT DATA

1. [Abra la AWS Marketplace consola en https://console.aws.amazon.com/marketplace.](https://console.aws.amazon.com/marketplace)
2. En el panel de navegación izquierdo, elija Administrar suscripciones.
3. Busque la AMI que desee lanzar y elija Lanzar nueva instancia.
4. En Región, seleccione la región incluida en la lista de permitidos.
5. Elija Continuar para lanzar mediante EC2. Esta acción le llevará a la consola de Amazon EC2.
6. Escriba un nombre del servidor.
7. Seleccione un tipo de instancia que se ajuste a los requisitos de rendimiento y costo de su proyecto. El punto de partida sugerido para el tamaño de la instancia es c5.2xLarge.
8. Elija un par de claves existente o cree y guarde uno nuevo. Para obtener información sobre los pares de claves, consulte los [pares de claves de Amazon EC2 y las instancias de Linux](#) en la Guía del usuario de Amazon EC2.
9. Edite la configuración de red y elija la VPC de la lista de permitidos y la subred adecuada.
10. Seleccione un grupo de seguridad existente o cree uno nuevo. Si se trata de una instancia de Amazon EC2 de Enterprise Server, lo habitual es permitir que el tráfico TCP llegue a los puertos 86 y 10086 para administrar la configuración de Micro Focus.
11. Configure el almacenamiento de la instancia de Amazon EC2.
12. Revise el resumen y seleccione Lanzar instancia. Para que el lanzamiento se realice correctamente, el tipo de instancia debe ser válido. Si se produce un error en el lanzamiento, seleccione Editar la configuración de la instancia y elija un tipo de instancia distinto.
13. Cuando vea el mensaje de éxito, seleccione Conectarse a la instancia.
14. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
15. En el panel de navegación, bajo Instancias elija Instancias.
16. En el panel principal, compruebe el estado de la instancia.





- Administrador central de NTT DATA UniKix

La carpeta `software` que aparece en la imagen anterior contiene los binarios de los componentes que se muestran arriba.

Tras validar correctamente la instancia de Amazon EC2, comience a utilizar AWS Mainframe Modernization Replatform con NTT DATA siguiendo la documentación de NTT Data.

# Aplicaciones en la modernización de AWS mainframes

Si es la primera vez que utiliza la modernización de AWS mainframes, consulte los siguientes temas para empezar:

- [¿Qué es la modernización del AWS mainframe?](#)
- [Configuración de la modernización del AWS mainframe](#)
- [Tutorial: Tiempo de ejecución gestionado para AWS Blu Age](#)
- [Tutorial: Tiempo de ejecución administrado para Micro Focus](#)

Una aplicación de AWS Mainframe Modernization contiene una carga de trabajo de mainframe migrada. La aplicación es análoga a una carga de trabajo en la unidad central y está asociada a un entorno de tiempo de ejecución. Puede añadir archivos por lotes y conjuntos de datos a las aplicaciones y supervisarlas a medida que se ejecutan. Las aplicaciones de modernización del AWS mainframe se crean para cada carga de trabajo que se migra. Al crear una aplicación de modernización de AWS mainframe, debe especificar el motor en el que se ejecutará la aplicación al crearla. Elija AWS Blu Age si utiliza el patrón de refactorización automática y Micro Focus si utiliza el patrón de replataforma.

## Temas

- [Cree una aplicación de modernización de mainframe AWS](#)
- [Implemente una aplicación de modernización AWS de mainframe](#)
- [Actualice una aplicación de modernización AWS de mainframe](#)
- [Elimine una aplicación de modernización de AWS mainframe de un entorno](#)
- [Elimine una aplicación de modernización AWS de mainframe](#)
- [Envíe trabajos por lotes para las aplicaciones de modernización AWS de mainframes](#)
- [Importe conjuntos de datos para aplicaciones de modernización de AWS mainframes](#)
- [Administre las transacciones de las aplicaciones de modernización AWS de mainframe](#)
- [Cree AWS recursos para una aplicación migrada](#)
- [Configuración de la aplicación administrada.](#)
- [AWS Referencia de definición de aplicaciones de modernización de mainframe](#)
- [AWS Referencia de definición de conjuntos de datos de modernización de mainframe](#)

# Cree una aplicación de modernización de mainframe AWS

Utilice la consola de modernización de AWS mainframe para crear una aplicación de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

## Crear una aplicación

Para crear una aplicación

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que desee crear la aplicación.
3. En la página Aplicaciones, seleccione Crear aplicación.
4. En la página Especificar información básica, en la sección Nombre y descripción, introduzca un nombre para la aplicación.
5. En el campo Descripción de la aplicación, escriba una descripción de la aplicación. Esta descripción ayudarle a usted y a otros usuarios a identificar el propósito de la aplicación.
6. En la sección Tipo de motor, seleccione Blu Age para la refactorización automática o Micro Focus para redefinir la plataforma.
7. En la sección de claves de KMS, elija Personalizar la configuración de cifrado si desea utilizar una AWS KMS clave gestionada por el cliente. Para obtener más información, consulte [Cifrado de datos en reposo para el servicio de AWS modernización de mainframe](#).

### Note

De forma predeterminada, AWS Mainframe Modernization cifra sus datos con una AWS KMS clave que AWS Mainframe Modernization posee y administra por usted. Sin embargo, puede optar por utilizar una clave gestionada por el cliente. AWS KMS

8. (Opcional) Elija una AWS KMS clave por nombre o por nombre de recurso de Amazon (ARN), o bien elija Crear una AWS KMS clave para ir a la AWS KMS consola y crear una clave nueva AWS KMS .
9. (Opcional) En la sección Etiquetas, seleccione Agregar nueva etiqueta para añadir una o varias etiquetas a la aplicación. Una etiqueta de aplicación es una etiqueta de atributo personalizada que le ayuda a organizar y administrar sus AWS recursos.

10. Elija Siguiente.
11. En la sección Recursos y configuraciones, utilice el editor en línea para introducir la definición de la aplicación. También puede elegir Use an application definition JSON file in an Amazon S3 bucket e indicar la ubicación de la definición de aplicación que desea usar. Para obtener más información, consulte [AWS Ejemplo de definición de aplicación de Blu Age](#) o [Definición de la aplicación Micro Focus](#).
12. Elija Siguiente.
13. En la página Revisar y crear, revise la información que ha proporcionado y, a continuación, elija Crear origen de datos.

## Implemente una aplicación de modernización AWS de mainframe

Utilice la consola de modernización de AWS mainframe para implementar una aplicación de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Implementar una aplicación

Para ejecutar una aplicación de modernización de AWS mainframe, primero debe implementarla en un entorno de ejecución. Una aplicación puede tener más de una versión. Cada versión de una aplicación tiene su propia definición de aplicación. Para implementar una aplicación, debe especificar la versión que desea implementar.

Solo puede implementar una versión de una aplicación determinada a la vez. Si implementa una versión de una aplicación y decide implementar una versión diferente, primero debe detener la aplicación si está en ejecución.

Para implementar una aplicación

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que desee crear la aplicación.
3. En la página Aplicaciones, seleccione la aplicación que desea implementar.
4. Elija Implementar aplicación.
5. En la sección Versiones disponibles, elija la versión que desea implementar.

6. En la sección Entornos, elija un entorno de tiempo de ejecución en el que desea que se ejecute la aplicación.
7. Elija Implementar.

Para implementar una versión diferente de una aplicación implementada

1. Abra la consola de modernización de AWS mainframe en <https://console.aws.amazon.com/m2/>.
2. En el Región de AWS selector, elija la región en la que desee crear la aplicación.
3. En la página Aplicaciones, seleccione la aplicación que desea implementar.
4. En el menú Acciones, elija Detener aplicación.
5. Cuando la aplicación se detenga, elija Implementar aplicación.
6. En la sección Versiones disponibles, elija la versión que desea implementar. En la sección Entornos, el entorno en el que ya está desplegada la aplicación aparece preseleccionado.
7. Elija Implementar.

## Actualice una aplicación de modernización AWS de mainframe

Utilice la consola de modernización de AWS mainframe para actualizar una aplicación de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Actualización de una aplicación

Una aplicación de modernización de AWS mainframe puede tener varias versiones, cada una con su propia definición de aplicación. Para actualizar una aplicación, proporcione una nueva definición de aplicación. Así se creará una versión nueva de la aplicación.

Para actualizar una aplicación

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación que desea actualizar.
3. En la página Aplicaciones, seleccione la aplicación que desea actualizar.

4. En la página de detalles de la aplicación, en la sección Definición actual, elija Editar para actualizar la definición de la aplicación actual.
5. En la página Actualizar aplicación, utilice el editor en línea para actualizar la definición de la aplicación actual.

También puede elegir Use an application definition JSON file in an Amazon S3 bucket e indicar la ubicación de la definición de aplicación que desea usar. Para obtener más información, consulte [AWS Ejemplo de definición de aplicación de Blu Age](#) o [Definición de la aplicación Micro Focus](#).

6. Cuando haya terminado de actualizar la definición de la aplicación, seleccione Actualizar.

#### Note

Tras actualizar la aplicación, debe volver a implementarla. Para obtener más información, consulte [Implemente una aplicación de modernización AWS de mainframe](#).

## Elimine una aplicación de modernización de AWS mainframe de un entorno

Puede eliminar una aplicación de modernización de AWS mainframe de un entorno mediante la consola de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Eliminar una aplicación de un entorno.

Si necesita eliminar una aplicación de modernización de AWS mainframe y ya está en ejecución, asegúrese de detenerla primero. Puede ver el estado de la aplicación en la página Aplicaciones.

Para eliminar una aplicación de un entorno.

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación que desea eliminar del entorno.

3. En la página Aplicaciones, elija la aplicación que desea eliminar del entorno y, a continuación, elija Acciones.
4. (Opcional) Si el estado de la aplicación es `Running`, seleccione Detener aplicación.
5. Seleccione Eliminar del entorno.

El proceso de eliminación se inicia de forma inmediata.

## Elimine una aplicación de modernización AWS de mainframe

Utilice la consola de modernización de AWS mainframe para eliminar una aplicación de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Eliminar una aplicación

Si necesita eliminar una aplicación de modernización de AWS mainframe y ya está en ejecución, asegúrese de detenerla primero. Puede ver el estado de la aplicación en la página Aplicaciones.

#### Eliminación de una aplicación

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación que desea eliminar.
3. En la página Aplicaciones, elija la aplicación que desea eliminar y, a continuación, elija Acciones.
4. (Opcional) Si el estado de la aplicación es `Running`, seleccione Detener aplicación.
5. Seleccione Eliminar aplicación.
6. En la ventana Eliminar aplicación, introduzca `delete` para confirmar que desea eliminar la aplicación y, a continuación, seleccione Eliminar.

## Envíe trabajos por lotes para las aplicaciones de modernización AWS de mainframes

En AWS Mainframe Modernization, puede enviar trabajos por lotes para sus aplicaciones. Puede enviar o cancelar trabajos por lotes y revisar los detalles sobre la ejecución de los trabajos por lotes.



Cada vez que envía un trabajo por lotes, AWS Mainframe Modernization crea una ejecución de trabajo por lotes independiente. Puede supervisar la ejecución de este trabajo. Puede buscar los trabajos por lotes por su nombre y suministrar archivos JCL o de script a los trabajos por lotes.

#### Important

Si cancela un trabajo por lotes, no se elimina el trabajo, se cancela una ejecución concreta del trabajo por lotes. Los registros del trabajo por lotes permanecen disponibles para que pueda consultarlos en los detalles de la ejecución del trabajo por lotes.

Si su trabajo por lotes requiere acceso a uno o más conjuntos de datos, utilice la consola de modernización de AWS mainframe o el AWS Command Line Interface (AWS CLI) para importar los conjuntos de datos. Para obtener más información, consulte [Importe conjuntos de datos para aplicaciones de modernización de AWS mainframes](#).

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#) y en [Cree una aplicación de modernización de mainframe AWS](#).

#### Temas

- [Envío de un trabajo por lotes](#).
- [Reinicie un trabajo por lotes](#)

## Envío de un trabajo por lotes.

Para enviar un trabajo por lotes.

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación para la que desea enviar un trabajo por lotes.
3. En la página Aplicaciones, seleccione la aplicación para la que desea enviar un trabajo por lotes.

#### Note

Para poder enviar un trabajo por lotes a una aplicación, debe implementar la aplicación de forma correcta.

4. En la página de detalles de la aplicación, seleccione Trabajos por lotes.
5. Seleccione Enviar el trabajo.
6. En la sección Seleccionar un script, elija un script. Puede buscar el script que desee por nombre.
7. Seleccione Enviar el trabajo.

## Reinicie un trabajo por lotes

Para reiniciar un trabajo por lotes

### Important

El reinicio de un trabajo por lotes solo está disponible en las versiones 8.0.6 o posteriores del motor de entorno Micro Focus. También debe tener un sistema de archivos EFS o FSx conectado a su entorno.

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se crearon la aplicación y el trabajo por lotes.
3. En la página Aplicaciones, elija la aplicación en la que desee reiniciar un trabajo por lotes.
4. En la página de detalles de la aplicación, seleccione Trabajos por lotes.
5. Seleccione el trabajo por lotes que desee reiniciar de la lista generada. Navegue hasta el menú Acciones y elija Reiniciar trabajo.
6. Especifique cómo desea reiniciar el trabajo por lotes. Puede elegir entre reiniciar desde el principio o reiniciar mediante pasos o pasos continuos.
  - La opción Reiniciar desde el principio le permite reiniciar todos los pasos de un trabajo por lotes desde el principio.
  - Con la opción Reiniciar mediante pasos o procsteps, puede elegir un paso específico o procstep (paso del procedimiento) que desee reiniciar y, opcionalmente, un paso o procstep después del cual finalizar.

**Note**

El paso final o procstep debe ser mayor o igual que el número del paso inicial o procstep.

7. Seleccione Enviar el trabajo.

## Importe conjuntos de datos para aplicaciones de modernización de AWS mainframes

Con AWS Mainframe Modernization, puede importar conjuntos de datos para usarlos con sus aplicaciones. Puede especificar los conjuntos de datos en un archivo JSON almacenado en un bucket de Amazon S3, o bien, puede especificar valores de configuración de conjuntos de datos por separado. Tras importar los conjuntos de datos, puede revisar los detalles de la tarea de importación para confirmar que se importaron los conjuntos de datos que quería. Todos los conjuntos de datos catalogados de una aplicación se muestran juntos en la consola.

Utilice la consola de modernización de AWS mainframe para importar conjuntos de datos para una aplicación de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#) y en [Cree una aplicación de modernización de mainframe AWS](#).

### Importar un conjunto de datos

Para importar un conjunto de datos

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación para la que desea importar los conjuntos de datos.
3. En la página Aplicaciones, seleccione la aplicación para la que desea importar conjuntos de datos.
4. En la página de detalles de la aplicación, elija Conjuntos de datos.
5. Seleccione Importar.

## 6. Realice una de las siguientes acciones siguientes:

- Elija Use data set configuration JSON file in an Amazon S3 bucket e indique la ubicación de la configuración del conjunto de datos.
- Elija Specify the data set configuration values separately con la configuración indicada. Consulte [the section called “Referencia de definición de conjuntos de datos”](#) para conocer los detalles de la definición específica.

Introduzca el nombre, la organización del conjunto de datos (VSAM, GDG, PO, PS), la ubicación y la ubicación externa de Amazon S3, así como los ajustes de los parámetros de cada valor de configuración del conjunto de datos. En la configuración guiada, también puede elegir Generar JSON para revisar la configuración de JSON a partir de sus datos.

## 7. Elija Enviar.

# Administre las transacciones de las aplicaciones de modernización AWS de mainframe

Con AWS Mainframe Modernization, puede ejecutar una aplicación, previa solicitud, al mismo tiempo que muchos otros usuarios que envían solicitudes para ejecutar la misma aplicación con los mismos archivos y programas. Una transacción consta de uno o más programas de aplicación que llevan a cabo el procesamiento necesario.


En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#) y en [Cree una aplicación de modernización de mainframe AWS](#).

## Gestión de las transacciones de las aplicaciones

Para administrar las transacciones de las aplicaciones


1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó la aplicación que desea ejecutar.
3. En la página Aplicaciones, elija la aplicación en la que desee gestionar las transacciones.
4. En la pestaña Transacciones, en Transaction resources, elija cómo quiere que se muestren sus recursos en la lista desplegable. Puede mostrar los recursos por recursos de transacciones, grupos, listas o SIT.

- Los recursos de transacciones le permiten elegir el tipo de recurso según las definiciones de archivos, las definiciones de transacciones, las definiciones de programas o las definiciones de colas de datos transitorios.

 Note

El servicio de modernización de AWS mainframe admite tipos de recursos adicionales para gestionar las transacciones de las aplicaciones y se puede acceder a él desde la consola.

- Los grupos son una colección de recursos de transacciones. Puede elegir los grupos que desee asociar a su recurso de transacciones.
- Las listas son una colección ordenada de grupos. Puede ver todos sus recursos y grupos de transacciones en una vista de lista. La lista de startup determina qué recursos se cargan cuando se inicializa el servidor.
  - Con el motor de refactorización AWS Blu Age, usted especifica las listas que se incluirán al inicio. No hay límite en el número de listas.
  - Con el motor de redefinición de plataforma de Micro Focus, puede especificar hasta cuatro listas en un SIT.
- La SIT (tabla de inicialización del sistema) muestra todas las configuraciones de transacciones disponibles. Puede buscar las SIT según sus propiedades (nombre, descripción y listas de startup). También puede elegir listas para asociarlas a la SIT que haya elegido.

 Note

Las SIT solo se aplican al motor de redefinición de plataforma de Micro Focus.

5. Elija un recurso de transacción para mostrar toda la información del recurso. También puede ver todos los atributos asociados a su recurso de transacciones.

## Cree AWS recursos para una aplicación migrada

Para ejecutar la aplicación migrada AWS, debe crear algunos AWS recursos con otros Servicios de AWS. Los recursos que debe crear incluyen lo siguiente:

- Un depósito de S3 para almacenar el código de la aplicación, la configuración, los archivos de datos y otros elementos necesarios.
- Una base de datos de Amazon RDS o Amazon Aurora para almacenar los datos que requiere la aplicación.
- Y AWS KMS key, que es necesaria AWS Secrets Manager para crear y almacenar secretos.
- Un secreto de Secrets Manager para guardar las credenciales de la base de datos

#### Note

Cada aplicación migrada requiere su propio conjunto de estos recursos. Se trata de un conjunto mínimo. Es posible que la aplicación también requiera recursos adicionales, como secretos de Amazon Cognito o colas de MQ.

## Permisos necesarios

Asegúrese de que tiene los siguientes permisos:

- `s3:CreateBucket`, `s3:PutObject`
- `rds:CreateDBInstance`
- `kms:CreateKey`
- `secretsmanager:CreateSecret`

## Bucket de Amazon S3

Tanto las aplicaciones refactorizadas como las que han redefinido su plataforma requieren un bucket de S3 que se configura de la siguiente manera:

```
bucket-name/root-folder-name/application-name
```

bucket-name

Cualquier nombre dentro de las restricciones de nomenclatura de Amazon S3. Le recomendamos que incluya el nombre de la Región de AWS como parte del nombre de su bucket. Asegúrese de crear el bucket en la misma región en la que tiene previsto implementar la aplicación migrada.

## root-folder-name

Nombre necesario para cumplir con las restricciones de la definición de la aplicación, que se crea como parte de la aplicación de modernización del AWS mainframe. Puede utilizar el `root-folder-name` para distinguir entre las distintas versiones de una aplicación, por ejemplo, la V1 y la V2.

## application-name

El nombre de la aplicación migrada, por ejemplo, PlanetsDemo o BankDemo.

## Database

Tanto las aplicaciones refactorizadas como las reconfiguradas pueden requerir una base de datos. Debe crear, configurar y administrar la base de datos de acuerdo con los requisitos específicos de cada motor de ejecución. AWS La modernización del mainframe admite el cifrado en tránsito en esta base de datos. Si habilita el SSL en su base de datos, asegúrese de especificar `sslMode` en el secreto de la base de datos junto con los detalles de conexión de la base de datos. Para obtener más información, consulte [AWS Secrets Manager secreto](#).

Si utiliza el patrón de refactorización de AWS Blu Age y necesita una BluSam base de datos, el motor de ejecución de AWS Blu Age espera una base de datos PostgreSQL de Amazon Aurora, que debe crear, configurar y administrar. La BluSam base de datos es opcional. Cree esta base de datos solo si su aplicación lo requiere. Para crear la base de datos, siga los pasos que se indican en [Creación de un clúster de base de datos de Amazon Aurora](#) en la Guía del usuario de Amazon Aurora.

Si utiliza el patrón de redefinición de plataforma de Micro Focus, puede crear una base de datos de Amazon RDS o Amazon Aurora PostgreSQL. Para crear la base de datos, siga los pasos que se indican en [Creación de una instancia de base de datos de Amazon RDS](#) en la Guía del usuario de Amazon RDS o en [Creación de un clúster de base de datos de Amazon Aurora](#) en la Guía del usuario de Amazon Aurora.

Para ambos motores de ejecución, debe almacenar las credenciales de la base de datos AWS KMS key para AWS Secrets Manager utilizarlas y cifrarlas.

## AWS Key Management Service clave

Debe almacenar las credenciales de la base de datos de la aplicación de forma segura en AWS Secrets Manager. Para crear un secreto en Secrets Manager, debe crear una AWS KMS key.

Para crear una nueva clave de KMS, siga los pasos de [Creación de claves](#) en la Guía para desarrolladores de AWS Key Management Service .

Tras crear la clave, debe actualizar la política de claves para conceder permisos de descifrado a AWS Mainframe Modernization. Añada las siguiente instrucciones de política:

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

## AWS Secrets Manager secreto

Debe almacenar las credenciales de la base de datos de la aplicación de forma segura en AWS Secrets Manager. Para crear un secreto, siga los pasos de [Creación de un secreto](#) en la Guía del usuario de AWS Secrets Manager .

AWS La modernización del mainframe admite el cifrado en tránsito en esta base de datos. Si habilita el SSL en su base de datos, asegúrese de especificar `sslMode` en el secreto de la base de datos junto con los detalles de conexión de la base de datos. Puede especificar uno de los siguientes valores para `sslMode`: `verify-full`, `verify-ca` o `disable`.

Durante el proceso de creación de la clave, elija Permisos de recursos. Opcional y, a continuación, elija Editar permisos. En el editor de políticas, añada una política basada en recursos, como la siguiente, para recuperar el contenido de los campos cifrados.

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



## Configuración de la aplicación administrada.

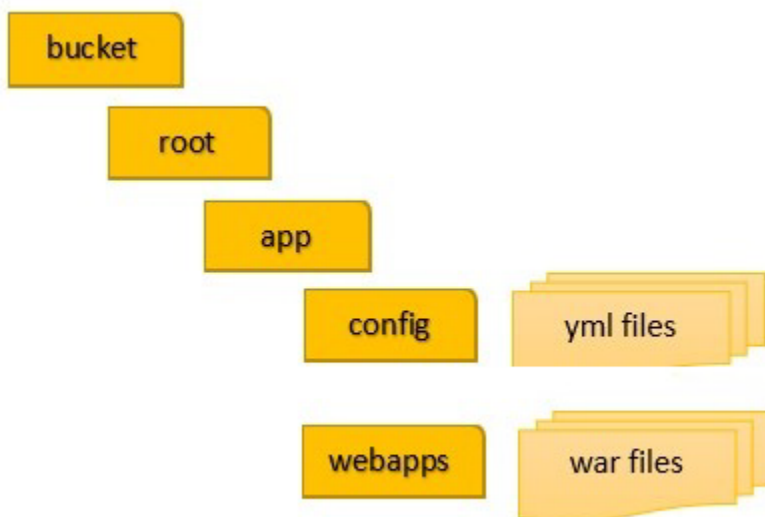
Puede configurar la aplicación para que incluya acceso a utilidades heredadas. También puede personalizar propiedades adicionales. Para entender qué se puede configurar y dónde, es útil entender la estructura general de una aplicación modernizada de AWS Blu Age.

### Temas

- [Estructura de las aplicaciones gestionadas de AWS Blu Age](#)
- [Configuración del acceso a las utilidades para las aplicaciones administradas](#)
- [Añadir propiedades de configuración para el motor AWS Blu Age](#)

## Estructura de las aplicaciones gestionadas de AWS Blu Age

Si utiliza el patrón de refactorización de AWS Blu Age, el motor de ejecución de AWS Blu Age espera la siguiente estructura dentro de la `application-name` carpeta de su depósito de S3:



### config

Contiene los archivos YAML del proyecto. Se trata de los archivos YAML específicos de su aplicación, que suelen tener un nombre similar `application-planetsdemo.yaml` y no el `application-main.yaml` archivo que AWS Mainframe Modernization proporciona y configura automáticamente para usted.

## webapps

Contiene los archivos `war` de la aplicación. Estos archivos son el resultado del proceso de modernización.

Una aplicación también puede tener las siguientes carpetas opcionales:

### jics/sql

Contiene el script `initJics.sql` que inicializa la base de datos JICS para su aplicación.

### scripts

Contiene scripts de la aplicación, que también puede proporcionar directamente dentro de los archivos `war`.

### sql

Contiene archivos de SQL de la aplicación, que también puede proporcionar directamente dentro de los archivos `war`.

### lnk

Contiene archivos de LNK de la aplicación, que también puede proporcionar directamente dentro de los archivos `war`.

### extra

Contiene tarros que pueden proporcionar capacidades adicionales para la aplicación modernizada.

## Administración del consumo de memoria Java de una aplicación

Para administrar el consumo de memoria de Java de la aplicación, añada un archivo de propiedades llamado `tomcat.properties` a la carpeta `application-name`. Este archivo puede tener dos propiedades: `xms`, que especifica el consumo mínimo de memoria de Java y `xmx`, que especifica el consumo máximo de memoria de Java. El siguiente ejemplo muestra el contenido de un archivo `tomcat.properties` válido:

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

Los valores que especifique para estas dos propiedades pueden estar en cualquiera de las siguientes unidades:

- Bytes: no especifique una unidad.
- Kilobytes: añada una K al valor.
- Megabytes: añada una M al valor.
- Gigabytes: añada una G al valor.

## Configuración del acceso a las utilidades para las aplicaciones administradas

Al refactorizar una aplicación de mainframe con AWS Blu Age, es posible que necesites proporcionar soporte para varios programas utilitarios de plataformas antiguas, como IDCAMS, INFUTILB, SORT, etc., si tu aplicación depende de ellos. AWS La refactorización de Blu Age proporciona este acceso con una aplicación web dedicada que se implementa junto con las aplicaciones modernizadas. Esta aplicación web requiere un archivo de configuración, `application-utility-pgm.yml`, que usted debe proporcionar. Si no proporciona este archivo de configuración, la aplicación web no se podrá implementar junto con la suya y no estará disponible.

### Temas

- [Propiedades de configuración](#)

En este tema se describen todas las propiedades posibles que puede especificar en el archivo de configuración `application-utility-pgm.yml`, junto con sus valores predeterminados. En este tema se describen las propiedades obligatorias y opcionales. A continuación se muestra un archivo de configuración completo de ejemplo. Incluye las propiedades en el orden que recomendamos. Puede utilizar este ejemplo como punto de partida para su propio archivo de configuración.

```
# 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
```

## Propiedades de configuración

Puede especificar las siguientes propiedades en su archivo de configuración.

### spring.jta.enabled

Opcional. Controla si está habilitada la compatibilidad con JTA. Para las utilidades, recomendamos que establezca este valor en `false`.

```
spring.jta.enabled : false
```

### logging.config

Obligatorio. Especifica la ruta al archivo de configuración del registrador dedicado. Le recomendamos utilizar el nombre `logback-utility.xml` y proporcionar este archivo como parte de la aplicación modernizada. La forma habitual de organizar estos archivos es colocar todos los archivos de configuración del registrador en el mismo lugar, normalmente en la subcarpeta `/config/logback` donde se `/config` encuentra la carpeta que contiene los archivos de configuración de YAML. Para obtener más información, consulte [Chapter 3: Logback configuration](#) en la documentación de Logback.

```
logging.config : classpath:logback-utility.xml
```

### encoding

Obligatorio. Especifica el conjunto de caracteres que utiliza el programa de utilidades. En la mayoría de los casos, al migrar desde plataformas z/OS, este conjunto de caracteres es una variante del EBCDIC y debe coincidir con el conjunto de caracteres configurado para las aplicaciones modernizadas. Si no se establece, el valor predeterminado es ASCII.

```
encoding : cp1047
```

### sys PunchEncoding

Opcional. Especifica el conjunto de caracteres que utilizan INFUTILB y DSNUTILB para generar y leer los archivos SYSPUNCH. Si utiliza los archivos SYSPUNCH de la plataforma anterior tal como están, este valor debe ser una variante EBCDIC. Si no se establece, el valor predeterminado es ASCII.

```
sysPunchEncoding : cp1047
```

## Configuración del origen de datos principal

Algunas utilidades relacionadas con las bases de datos, como LOAD y UNLOAD, requieren acceder a una base de datos de destino a través de un origen de datos. Al igual que otras definiciones de fuentes de datos de AWS Mainframe Modernization, este acceso requiere que utilice AWS Secrets Manager. Las propiedades que señalan a los secretos adecuados de Secrets Manager son las siguientes:

```
spring.aws.client.datasources.primary.secret
```

Opcional. Especifica el secreto de Secrets Manager que contiene las propiedades del origen de datos.

```
spring.aws.client.datasources.primary.secret: datasource-secret-ARN
```

```
spring.aws.client.datasources.primary.dbname
```

Opcional. Especifica el nombre de la base de datos de destino si el nombre de la base de datos no se proporciona directamente en el secreto de la base de datos, junto con la propiedad dbname.

```
spring.aws.client.datasources.primary.dbname: target-database-name
```

```
tratar LargeNumber AsInteger
```

Opcional. Relacionado con las especificaciones del motor de base de datos Oracle y el uso de las utilidades DSNTEP2/DSNTEP4. Si establece este indicador en true, los números grandes que provienen de la base de datos Oracle (NUMBER (38,0)) se tratan como números enteros. Predeterminado: false

```
treatLargeNumberAsInteger : false
```

```
zonedMode
```

Opcional. Establece el modo zonificado para codificar o decodificar los tipos de datos zonificados. Esta configuración influye en la forma en que se representan los dígitos de los signos. Los siguientes valores son válidos:

- **EBCDIC\_STRICT**: predeterminado Utilice una definición estricta para la gestión de signos. Dependiendo de si el conjunto de caracteres es EBCDIC o ASCII, la representación de dígitos con signo utiliza los siguientes caracteres:
  - Caracteres EBCDIC que corresponden a bytes (Cn+Dn) para representar rangos de dígitos positivos y negativos (+0 a +9, -0 a -9). Los caracteres se muestran como {,A a I, }, J a R
  - Caracteres ASCII que corresponden a bytes (3n+7n) para representar rangos de dígitos positivos y negativos (+0 a +9, -0 a -9). Los caracteres se muestran como 0 a 9, p a y
- **EBCDIC\_MODIFIED**: utilice una definición modificada para la gestión de los signos. Tanto para EBDIC como para ASCII, la misma lista de caracteres representa los dígitos del signo, es decir, +0 a +9 se asignan a { + A a I y -0 a -9 se asignan a } + J a R. \
- **AS400**: utilícelo para activos heredados modernizados que provienen de las plataformas iSeries (AS400).

```
zonedMode:EBCDIC_STRICT
```

## jcl.type

Opcional. Indica el tipo heredado de los scripts JCL modernizados. La utilidad IDCAMS utiliza esta configuración para personalizar el código de retorno si el JCL que invoca es de tipo vse. Los valores válidos son los siguientes:

- **mvs** (predeterminado)
- **vse**

```
jcl.type : mvs
```

## Propiedades relacionadas con las utilidades de descarga de bases de datos

Utilice estas propiedades para configurar las utilidades que descargan tablas de bases de datos en conjuntos de datos. Todas las demás propiedades son opcionales.

En este ejemplo se muestran todas las propiedades de descarga posibles.

```
# Unload properties
# For date/time: if use database configuration is enabled, formats are ignored
# For nbi; use hexadecimal syntaxe to specify the byte value
unload:
sqlCodePointShift: 0
nbi:
```

```

whenNull: "6F"
whenNotNull: "00"
useDatabaseConfiguration: false
format:
date: MM/dd/yyyy
time: HH.mm.ss
timestamp: yyyy-MM-dd-HH.mm.ss.SSSSSS
chunkSize: 0
fetchSize: 0
varCharIsNull: false
columnFiller: space

```

## SQL CodePoint Shift

Opcional. Especifica un valor entero que representa el desplazamiento de puntos del código SQL utilizado en los datos. El valor predeterminado es 0. Esto significa que no se produce ningún cambio de punto de código. Alinee esta configuración con el parámetro de cambio de punto del código SQL que se utiliza en las aplicaciones modernizadas. Cuando se utiliza el desplazamiento de puntos de código, el valor más común para este parámetro es 384.

```
unload.sqlCodePointShift: 0
```

## nbi

Opcional. Especifica un byte indicador nulo. Se trata de un valor hexadecimal (en forma de cadena) que se añade a la derecha del valor de los datos. Los dos valores posibles son los siguientes:

- whenNull: agrega el valor hexadecimal cuando el valor de los datos es nulo. El valor predeterminado es 6`. A veces se utiliza el valor alto FF en su lugar.

```
unload.nbi.whenNull: "6F"
```

- cuando NotNull: añade el valor hexadecimal cuando el valor de los datos no sea nulo, pero la columna sí lo sea. Valor predeterminado: 00 (valor bajo).

```
unload.nbi.whenNotNull: "00"
```

## usar DatabaseConfiguration

Opcional. Especifica las propiedades de formato de fecha y hora. Se utiliza para tratar los objetos de fecha y hora en las consultas UNLOAD. El valor predeterminado es false.



- Si se establece en `true`, utiliza las propiedades `pgmDateFormat`, `pgmTimeFormat` y `pgmTimestampFormat` del archivo de configuración principal (`application-main.yml`).
- Si se establece en `false`, utiliza las siguientes propiedades de formato de fecha y hora:
  - `unload.format.date`: especifica un patrón de formato de fecha. El valor predeterminado es `MM/dd/yyyy`.
  - `unload.format.time`: especifica un patrón de formato de hora. El valor predeterminado es `HH.mm.ss`.
  - `unload.format.timestamp`: especifica un patrón de formato de marca de tiempo. El valor predeterminado es `yyyy-MM-dd-HH.mm.ss.SSSSSS`.

### chunkSize

Opcional. Especifica el tamaño de los fragmentos de datos que se utilizan para crear conjuntos de datos `SYSREC`. Estos conjuntos de datos son el objetivo de la operación de descarga de conjuntos de datos, con operaciones en paralelo. El valor predeterminado es `0` (sin fragmentos).

```
unload.chunkSize:0
```

### fetchSize

Opcional. Especifica el tamaño de la recuperación de datos. El valor es el número de registros que se van a recuperar al mismo tiempo cuando se utiliza una estrategia de fragmentos de datos. Predeterminado: `0`.

```
unload.fetchSize:0
```

### var CharIs Nulo

Opcional. Especifica cómo gestionar una columna `varchar` que no admite valores `null` con contenido en blanco. El valor predeterminado es `false`.

Si establece este valor en `true`, el contenido de la columna se trata como una cadena vacía a efectos de descarga, en lugar de como una cadena de espacios individuales. Establezca este indicador como `true` solo para el caso del motor de base de datos Oracle.

```
unload.varCharIsNull: false
```

## columnFiller

Opcional. Especifica el valor que se utilizará para rellenar las columnas descargadas en las columnas de varchar. Los valores posibles son espacios o valores bajos. Valor predeterminado: space

```
unload.columnFiller: space
```

## Propiedades relacionadas con la carga de base de datos

Utilice estas propiedades para configurar las utilidades que cargan registros de conjuntos de datos en una base de datos de destino, por ejemplo, DSNUTILB. Todas las demás propiedades son opcionales.

En este ejemplo se muestran todas las propiedades de carga posibles.

```
# 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
```

## CodePointCambio sql

Opcional. Especifica un valor entero que representa el desplazamiento de puntos del código SQL utilizado en los datos. El valor predeterminado es 0, lo que significa que las aplicaciones no desplazan los puntos de código. Alinee esta configuración con el parámetro de cambio de punto del código SQL que se utiliza en las aplicaciones modernizadas. Cuando se utiliza el desplazamiento de puntos de código, el valor más común para este parámetro es 384.

```
load.sqlCodePointShift : 384
```

## batchSize

Opcional. Especifica un valor entero que representa el número de registros que se van a tratar antes de enviar una declaración de lote real a la base de datos. El valor predeterminado es 0.

```
load.batchSize: 500
```

## format

Opcional. Especifica los patrones de formato de fecha y hora que se utilizarán para las conversiones de fecha y hora durante las operaciones de carga de la base de datos.

- `load.format.localDate`: patrón de formato de fecha local. El valor predeterminado es `dd.MM.yyyy|dd/MM/yyyy|yyyy-MM-dd`.
- `load.format.dbDate`: patrón de formato de fecha de la base de datos. El valor predeterminado es `yyyy-MM-dd`.
- `load.format.localTime`: patrón de formato de hora local. El valor predeterminado es `HH:mm:ss|HH.mm.ss`.
- `load.format.dbTime`: patrón de formato de hora de la base de datos. El valor predeterminado es `HH:mm:ss`.

## table-mappings

Opcional. Especifica una colección de asignaciones proporcionadas por el cliente entre nombres de tablas antiguos y modernos. El programa de utilidades DSNUTILB usa estas asignaciones.

Especifique los valores en el siguiente formato: MODERN\_TABLE\_NAME :  
LEGACY\_TABLE\_NAME

A continuación se muestra un ejemplo:

```
table-mappings:  
TABLE_1_NAME : LEGACY_TABLE_1_NAME  
TABLE_2_NAME : LEGACY_TABLE_2_NAME  
...  
TABLE_*N*_NAME : LEGACY_TABLE_*N*_NAME
```

**Note**

Cuando se inicia la aplicación de la utilidad, registra de forma explícita todas las asignaciones proporcionadas.

## Añadir propiedades de configuración para el motor AWS Blu Age

Puedes añadir un archivo a la `config` carpeta de tu aplicación refactorizada que te dará acceso a las nuevas funciones del motor de ejecución Blu Age. AWS Debe asignar un nombre a este archivo `user-properties.yml`. Este archivo no reemplaza la definición de la aplicación, sino que la amplía. En este tema se describen las propiedades que puede incluir en el archivo `user-properties.yml`.

**Note**

No puede cambiar algunos parámetros porque están controlados por la modernización del AWS mainframe o por la definición de la aplicación. Todos los parámetros especificados en la definición de la aplicación para su aplicación tienen prioridad sobre los parámetros que especifique en `user-properties.yml`.

Para obtener más información sobre las aplicaciones refactorizadas, consulte [Estructura de las aplicaciones gestionadas de AWS Blu Age](#).

El siguiente diagrama muestra dónde ubicar el `user-properties.yml` archivo dentro de la estructura de la aplicación de ejemplo AWS Blu Age, PlanetsDemo.

```
PlanetsDemo-v1/  
  ## config/  
  # ## application-PlanetsDemo.yml  
  # ## user-properties.yml  
  ## jics/  
  ## webapps/
```

## Referencia de propiedades de configuración

Esta es la lista de propiedades disponibles. Todos los parámetros son opcionales.

## Temas

- [Propiedades de la aplicación Gapwalk](#)
- [Propiedades del script de procesamiento por lotes de Gapwalk](#)
- [Propiedades de Gapwalk Blugen](#)
- [Propiedades del comando Gapwalk CL](#)
- [Propiedades del ejecutor Gapwalk CL](#)
- [Propiedades JHDB de Gapwalk](#)
- [Propiedades JICS de Gapwalk](#)
- [Propiedades de tiempo de ejecución de Gapwalk](#)
- [Propiedades del programa de utilidades Gapwalk](#)
- [Otras propiedades](#)

### Propiedades de la aplicación Gapwalk

#### bluesam.fileLoading.commitInterval

Opcional. El intervalo de confirmación de BluSam.

Tipo: número

Predeterminado: 100000

#### card.encoding

Opcional. Codificación de tarjetas: para usar con `useControlMVariable`.

Tipo: cadena

Predeterminado: CP1145

#### checkinputfilesize

Opcional. Especifica si se debe activar una comprobación si el tamaño del archivo es múltiplo del tamaño del registro.

Tipo: booleano

Predeterminado: false

`database.cursor.overflow.allowed`

Opcional. Especifica si se debe permitir que el cursor se desborde. Configúrelo en `true` para realizar una siguiente llamada en el cursor, sea cual sea su posición. Configúrelo en `false` para comprobar si el cursor está en la última posición antes de realizar una siguiente llamada con el cursor. Actívela solo si el cursor es SCROLLABLE (SENSITIVE o INSENSITIVE)

Tipo: booleano

Predeterminado: `true`

`DataSimplifier.OnDataInvalidNumeric`

Opcional. Cómo reaccionar al decodificar datos numéricos no válidos. Los valores permitidos son: `reject`, `toleratespaces`, `toleratespaceslowvalues`, `toleratemoost`.

Tipo: cadena

Predeterminado: rechazar

`Archivos predeterminados KeepExisting`

Opcional. Especifica si se debe establecer el valor anterior predeterminado del conjunto de datos.

Tipo: booleano

Predeterminado: `false`

`disposition.checkexistence`

Opcional. Especifica si se debe publicar una verificación de la existencia del archivo para el conjunto de datos con DISP SHR u OLD.

Tipo: booleano

Predeterminado: `false`

`externalSort.threshold`

Opcional. El umbral de clasificación: cuándo cambiar a una ordenación externa (fusión).

Tipo: cadena

Predeterminado: `null`

`externalSort.threshold: 12MB`

## forceHR

Opcional. Especifica si se debe utilizar SYSPRINT inteligible, ya sea en la consola o en la salida de un archivo.

Tipo: booleano

Predeterminado: false

## forcedDate

Opcional. Fuerza una fecha y hora específicas en la base de datos. Úselo solo durante el desarrollo y las pruebas.

Predeterminado: null

`forcedDate: 2022-08-26T12:59:58.123456+01:57`

## frozenDate

Opcional. Congela la fecha y la hora de la base de datos. Úselo solo durante el desarrollo y las pruebas.

Predeterminado: false

`frozenDate: false`

## ims.messages.extendedSize

Opcional. Especifica si se debe establecer el tamaño extendido en los mensajes ims.

Tipo: booleano

Predeterminado: false

## lockTimeout

Opcional. El tiempo de espera en milisegundos de una transacción cuando no se puede adquirir un bloqueo dentro de un período de tiempo específico.

Tipo: número

Predeterminado: 500

## mapTransfo.prefixes

Opcional. Lista de prefijos que se utilizarán al transformar las variables de ControlM. Cada uno separado por comas.

Tipo: cadena

Predeterminado: `&,@,%%`

`consulta.use ConcatCondition`

Opcional. Especifica si la condición clave se crea mediante concatenación de claves o no.

Tipo: booleano

Predeterminado: `false`

`rollbackOnRTE`

Opcional. Especifica si se debe anular la transacción implícita de la unidad de ejecución en las excepciones de tiempo de ejecución.

Tipo: booleano

Predeterminado: `false`

`actuar ThreadLimit`

Opcional. El límite de subprocessos para activar scripts.

Tipo: número

Predeterminado: `5`

`cambio sql CodePoint`

Opcional. El cambio de punto del código sql. Cambia el punto de código de los caracteres de control que podemos encontrar al migrar datos de rdbms antiguos a un rdbms moderno. Por ejemplo, puede especificar `384` para que coincida con un carácter `\u0180`.

Tipo: número

Predeterminado: `0`

`sql IntegerOverflow permitido`

Opcional. Especifica si se permite el desbordamiento de enteros de SQL, es decir, si se permite colocar valores más grandes en la variable `host`.

Tipo: booleano



Predeterminado: false

paso FailWhen Abend

Opcional. Especifica si se debe generar un abend si un paso falla o completa la ejecución.

Tipo: booleano

Predeterminado: true

detener Encontrado ExecutionWhen ProgNot

Opcional. Especifica si se debe detener la ejecución si no se encuentra un programa. Si se establece en `true`, interrumpe la ejecución si no se encuentra ningún programa.

Tipo: booleano

Predeterminado: true

mayúscula UserInput

Opcional. Especifica si la entrada del usuario debe estar en mayúsculas.

Tipo: booleano

Predeterminado: true

useControlMVariable

Opcional. Especifica si se debe utilizar la especificación control-M para el reemplazo de variables.

Tipo: booleano

Predeterminado: false

Propiedades del script de procesamiento por lotes de Gapwalk

encoding

Opcional. La codificación utilizada en los proyectos de scripts de procesamiento por lotes (no con groovy). Espera una codificación válida CP1047, IBM930, ASCII, UTF-8...

Tipo: cadena

Predeterminado: ASCII

## Propiedades de Gapwalk Blugen

### managers.trancode

Opcional. El mapeo de transcodificación del administrador de diálogos. Le permite asignar un código de transacción del JICS a un administrador de diálogos. El formato esperado es `trancode1:dialogManager1;trancode2:dialogManager2;`.

Tipo: cadena

Predeterminado: null

`managers.trancode: OR12:MYDIALOG1`

## Propiedades del comando Gapwalk CL

### commands-off

Opcional. Lista de comandos a desactivar, separados por comas. Los valores permitidos son `PGM_BASIC`, `RCVMSG`, `SNDRCVF`, `CHGVAR`, `QCLRDTAQ`, `RTVJOB`, `ADDLFM`, `ADDPFM`, `RCVF`, `OVRDBF`, `DLTOVR`, `CPYF`, `SNDDTAQ`. Útil cuando se desea deshabilitar o sobrescribir un programa existente. `PGM_BASIC` es un programa específico de AWS Blu Age Runtime diseñado con fines de depuración.

Tipo: cadena

Predeterminado: null

### spring.datasource.primary.jndi-name

Opcional. La fuente de datos principal de la interfaz de nombres y directorios de Java (JNDI).

Tipo: cadena

Predeterminado: `jdbc/primary`

### zonedMode

Opcional. El modo para codificar o decodificar tipos de datos zonificados. Los valores permitidos son `EBCDIC_STRICT` / `EBCDIC_MODIFIED` / `AS400`.

Tipo: cadena

Predeterminado: `EBCDIC_STRICT`

## Propiedades del ejecutor Gapwalk CL

### cl.configuration.context.encoding

Opcional. La codificación de los archivos CL. Espera una codificación válida CP1047, IBM930, ASCII, UTF-8...

Tipo: cadena

Predeterminado: CP297

### cl.zonedMode

Opcional. El modo para codificar o decodificar los comandos del lenguaje de control (CL). Los valores permitidos son EBCDIC\_STRICT / EBCDIC\_MODIFIED / AS400.

Tipo: cadena

Predeterminado: EBCDIC\_STRICT

## Propiedades JHDB de Gapwalk

### ims.programas

Opcional. Lista de programas de IMS que se van a utilizar. Separe cada parámetro con punto y coma (;) y cada transacción con una coma (,) . Por ejemplo: `ims.programas: PCP008, PCT008; PCP054, PCT054; PCP066, PCT066; PCP068, PCT068;`

Tipo: cadena

Predeterminado: null

### jhdb.checkpointPath

Opcional. Si `jhdb.checkpointPersistence` no es `none`, este parámetro le permite configurar la ruta de persistencia del punto de comprobación (ubicación de almacenamiento del archivo `checkpoint.dat`), todos los datos de los puntos de comprobación contenidos en el registro se serializan y se hace una copia de seguridad en un archivo (`checkpoint.dat`) ubicado en la carpeta proporcionada. Tenga en cuenta que esta copia de seguridad solo afecta a los datos de los puntos de control (`ScriptID`, `StepID`, posición de la base de datos y área del punto de control).

Tipo: cadena

Predeterminado: `file:./setup/`

## jhdb.checkpointPersistence

Opcional. El modo de persistencia del punto de comprobación. Los valores permitidos son none / add / end. Utilice add para conservar los puntos de comprobación cuando se crea y se agrega uno nuevo al registro. Utilice end para conservar el punto de comprobación al cerrar el servidor. Cualquier otro valor deshabilita la persistencia. Tenga en cuenta que cada vez que se añada un nuevo punto de control al registro, todos los puntos de comprobación existentes se serializarán y el archivo se borrará. No es un anexo a los datos existentes en el archivo. Por lo tanto, dependiendo del número de puntos de comprobación, puede tener algún efecto en el rendimiento.

Tipo: cadena

Predeterminado: none

## jhdb.configuration.context.encoding

Opcional. La codificación JHDB (base de datos jerárquica de Java). Espera una cadena de codificación válida CP1047, IBM930, ASCII, UTF-8...

Tipo: cadena

Predeterminado: CP297

## jhdb.identification CardData

Opcional. Se utiliza para codificar algunos “datos de la tarjeta de identificación del operador” en el campo MID designado por el parámetro CARD.

Tipo: cadena

Predeterminado: ""

## jhdb.lterm

Opcional. Le permiten forzar un ID de terminal lógico común en el caso de una emulación de IMS. Si no se establece, se utiliza sessionId.

Tipo: cadena

Predeterminado: null

## jhdb.metadata.extrath

Parámetro de configuración que especifica una carpeta raíz adicional específica para el tiempo de ejecución para las carpetas psbs y dbds.

Tipo: cadena

Predeterminado: file:./setup/

**Note**

Actualmente, debido a las restricciones de implementación, debe copiar sus directorios dbds y psbs en el directorio config de su aplicación o en un subdirectorío del directorio config: por ejemplo, config/setup

```
config
|- setup
  |- dbds
  |- psbs
```

y configurarlo en application-jhdb.yml

```
jhdb.metadata.extrapath: file: ./config/setup/
```

jhdb.navigation.cachenexts

Opcional. La duración de la caché (en milisegundos) utilizada en la navegación jerárquica de un RDBMS.

Tipo: número

Predeterminado: 5000

jhdb.query.limit JoinUsage

Opcional. Especifica si se debe utilizar el parámetro de límite de uso de uniones en los gráficos del RDBMS.

Tipo: booleano

Predeterminado: true

jhdb.use-db-prefix

Opcional. Especifica si se debe habilitar un prefijo de base de datos en la navegación jerárquica de un RDBMS.

Tipo: booleano

Predeterminado: true

## Propiedades JICS de Gapwalk

### Ubicación de `jics.data.data JsonInit`

Opcional. Ubicación del archivo json preparado por el analizador a partir del análisis del CSD y utilizado para inicializar la base de datos jics,

Tipo: cadena

Predeterminado: ""

### `jics.db.data ScriptLocation`

Opcional. Ubicación del script `initJics.sql`, preparado por el analizador a partir del análisis de las exportaciones de CSD desde el ordenador central.

Tipo: cadena

Predeterminado: ""

### Ubicación de `jics.db.data TestQuery`

Opcional. Ubicación de un script sql que contiene una única consulta sql que se espera que devuelva un recuento de objetos (por ejemplo: contar el número de registros de la tabla del programa jics). Si el recuento es 0, la base de datos se cargará mediante el script `jics.db.dataScriptLocation`; de lo contrario, se omitirá la carga de la base de datos.

Tipo: cadena

Predeterminado: ""

### `jics.db.ddl ScriptLocation`

Opcional. La ubicación del script ddl de Jics. Permite iniciar el esquema de la base de datos jics mediante un script `.sql`.

Tipo: cadena

Predeterminado: ""

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

## Ubicación de `jics.db.schema TestQuery`

Opcional. Ubicación del archivo sql que debe contener una consulta única que devuelva el número de objetos del esquema jics (si los hay).

Tipo: cadena

Predeterminado: ""

## `jics.run.pool.enable UnitLauncher`

Opcional. Especifica si se debe activar el grupo de lanzadores de unidades de ejecución en JICS.

Tipo: booleano

Predeterminado: false

## `jics.run.UnitLauncher.pool.Size`

Opcional. El tamaño del conjunto de lanzadores de unidades de ejecución en JICS.

Tipo: número

Predeterminado: 20

## `jics.run.UnitLauncher.pool.ValidationInterval`

Opcional: el intervalo de validación del conjunto de lanzadores de unidades de ejecución en JICS, expresado en milisegundos.

Tipo: número

Predeterminado: 1000

## `jics.queues.sqs.region`

Opcional. El Región de AWS para Amazon SQS, utilizado en JICS. Se recomienda configurar la misma región de la aplicación desplegada para garantizar el rendimiento, pero no es obligatorio.

Tipo: cadena

Predeterminado: eu-west-1

## `jics.xa.agent.timeout`

Opcional. Define la duración máxima para que el agente xa responsable de gestionar las transacciones distribuidas complete sus operaciones.

Tipo: número

Predeterminado: null

`mq.queues.sqs.region`

Opcional. El Región de AWS para el servicio Amazon SQS MQ.

Tipo: cadena

Predeterminado: eu-west-3

Ejecutor de tareas. ¿Permitir CoreThread TimeOut

Opcional. Especifica si se debe permitir que se agote el tiempo de espera de los subprocesos principales en JCIS. Esto permite un crecimiento y una reducción dinámicos incluso en combinación con una cola distinta de cero (ya que el tamaño máximo del grupo solo aumentará una vez que la cola esté llena).

Tipo: booleano

Predeterminado: false

`TaskExecutor.core PoolSize`

Opcional. Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño del grupo de núcleos.

Tipo: número

Predeterminado: 5

`TaskExecutor.max PoolSize`

Opcional. Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño máximo del grupo (número máximo de subprocesos paralelos).

Tipo: número

Predeterminado: 10

`taskExecutor.queueCapacity`

Opcional. Cuando se inicia una transacción en un terminal mediante un script groovy, se crea un nuevo subproceso. Utilice este parámetro para configurar el tamaño de la cola. (= número máximo de transacciones pendientes cuando se alcanza `taskExecutor.maxPoolSize`)



Tipo: número

Predeterminado: 50

## Propiedades de tiempo de ejecución de Gapwalk

### cacheMetadata

Opcional. Especifica si se deben almacenar en caché los metadatos de la base de datos.

Tipo: booleano

Predeterminado: true

### check-groovy-file

Opcional. Especifica si se debe comprobar el contenido de los archivos groovy antes de registrarlos.

Tipo: booleano

Predeterminado: true

### databaseStatistics

Opcional. Especifica si se permite a los compiladores de SQL recopilar y mostrar información estadística.

Tipo: booleano

Predeterminado: false

### fecha TimeFormat

Opcional. `dateTimeFormat` Describe cómo incluir el tipo de fecha, hora y hora de la base de datos en las entidades simplificadoras de datos. Los valores permitidos son ISO / EUR / USA / LOCAL

Tipo: cadena

Predeterminado: ISO

### db DateFormat

Opcional. El formato de fecha de destino de la base de datos.

Tipo: cadena

Predeterminado: aaaa-MM-dd

#### db TimeFormat

Opcional. El formato de hora de destino de la base de datos.

Tipo: cadena

Predeterminado: HH:mm:ss

#### db TimestampFormat

Opcional. El formato de hora de destino de la base de datos.

Tipo: cadena

Predeterminado: aaaa-MM-dd HH:mm:ss.SSSSSS

#### fetchSize

Opcional. El valor fetchSize de los cursores. Se usa cuando se obtienen datos con fragmentos mediante utilidades de carga o descarga.

Tipo: número

Predeterminado: 10

#### Tipo ForceDisableSQL TrimString

Opcional. Especifica si se debe deshabilitar el recorte de todos los parámetros de cadena SQL.

Tipo: booleano

Predeterminado: false

#### local DateFormat

Opcional. Lista de formatos de fecha locales. Separe cada formato con |.

Tipo: cadena

#### local TimeFormat

Opcional. Lista de formatos de hora locales. Separe cada formato con |.

Tipo: cadena

#### local TimestampFormat

Opcional. Lista de formatos de marca temporal locales. Separe cada formato con |.

Tipo: cadena

Predeterminado:

pgm DateFormat

Opcional. El formato de fecha y hora utilizado en los programas.

Tipo: cadena

Predeterminado: aaaa-MM-dd

pgm TimeFormat

Opcional. Formato de hora utilizado para la ejecución de los pgm (programas).

Tipo: cadena

Predeterminado: HH.mm.ss

pgm TimestampFormat

Opcional. Formato de marca temporal

Tipo: cadena

Predeterminado: aaaa-MM-dd-HH.mm.ss.SSSSSS

Propiedades del programa de utilidades Gapwalk

jcl.type

Opcional. tipo de archivo . jcl. Los valores permitidos son jcl / vse. Los comandos PRINT/ REPRO de la utilidad IDCAMS devuelven 4 si el archivo está vacío en el caso de jcl que no es vse.

Tipo: cadena

Predeterminado: mvs

listcat.variablelengthpreprocessor.enabled

Opcional. Especifica si se debe habilitar el preprocesador de longitud variable para el comando LISTCAT.

Tipo: booleano

Predeterminado: false

`listcat.variablelengthpreprocessor.type`

Opcional. El tipo de objetos que contiene el archivo `listcat`, si está activado

`listcat.variablelengthpreprocessor.enabled`. Los valores permitidos son `rdw` / `bdw`.

Tipo: cadena

Predeterminado: `rdw`

`load.batchSize`

Opcional. El tamaño del lote de la utilidad de carga.

Tipo: número

Predeterminado: 0

`load.format.dbDate`

Opcional. El formato de base de datos de la utilidad de carga que se va a utilizar.

Tipo: cadena

Predeterminado: `aaaa-MM-dd`

`load.format.dbTime`

Opcional. La hora de base de datos de la utilidad de carga que se va a utilizar.

Tipo: cadena

Predeterminado: `HH:mm:ss`

`load.format.localDate`

Opcional. El formato de fecha local de la utilidad de carga que se va a utilizar.

Tipo: cadena

Predeterminado: `dd.MM.aaaa|dd/MM/aaaa|aaaa-MM-dd`

`load.format.localTime`

Opcional. El formato de hora local de la utilidad de carga que se va a utilizar.

Tipo: cadena

Predeterminado: hh:mm:ss|hh.mm.ss

load.sql Shift CodePoint

Opcional. El punto de cambio en el código SQL para la utilidad de carga. Ejecuta el proceso de cambio de caracteres. Se requiere cuando la base de datos de destino de DB2 es Postgresql.

Tipo: número

Predeterminado: 0

sys PunchEncoding

Opcional. El conjunto de caracteres de codificación de syspunch. Los valores admitidos son Cp1047 / ASCII.

Tipo: cadena

Predeterminado: ASCII

tratar LargeNumber AsInteger

Opcional. Especifica si se deben tratar los números grandes como Integer. Se tratan como BigDecimal por defecto.

Tipo: booleano

Predeterminado: false

unload.chunkSize

Opcional. Tamaño de fragmento utilizado para la utilidad de descarga.

Tipo: número

Predeterminado: 0

unload.columnFiller

Opcional. El rellenedor de columnas de la utilidad de descarga.

Tipo: cadena

Predeterminado: space

unload.fetchSize

Opcional. Permite ajustar el tamaño de la búsqueda al manipular los cursores en la utilidad de descarga.

Tipo: número

Predeterminado: 0

`unload.format.date`

Opcional. Si `unload.useDatabaseConfiguration` está activado, el formato de fecha que se utilizará en la utilidad de descarga. Para obtener más información, consulte [unload.format.date](#).

Tipo: cadena

Predeterminado: aaaa-MM-dd

`unload.format.time`

Opcional. Si `unload.useDatabaseConfiguration` está activado, el formato de hora que se utilizará en la utilidad de descarga.

Tipo: cadena

Predeterminado: HH.mm.ss

`unload.format.timestamp`

Opcional. Si `unload.useDatabaseConfiguration` está activado, el formato de marca de tiempo que se utilizará en la utilidad de descarga.

Tipo: cadena

Predeterminado: aaaa-MM-dd-HH.mm.ss.SSSSSS

`descargar.nbi.when NotNull`

Opcional. El valor del indicador de bytes nulos (nbi) que se añade cuando el valor de la base de datos no es nulo.

Tipo: hexadecimal

Predeterminado: 00

`unload.nbi.whenNull`

Opcional. El valor del indicador de bytes nulos (nbi) que se añade cuando el valor de la base de datos es nulo.

Tipo: hexadecimal

Predeterminado: 6F

`descargar.nbi.write NullIndicator`

Opcional. Especifica si se debe escribir el indicador nulo en el archivo de salida de descarga.

Tipo: booleano

Predeterminado: false

`unload.sql Shift CodePoint`

Opcional. El punto de cambio en el código SQL para la utilidad de descarga. Ejecuta el proceso de cambio de caracteres. Se requiere cuando la base de datos de destino de DB2 es Postgresql.

Tipo: número

Predeterminado: 0

`descargar.usar DatabaseConfiguration`

Opcional. Especifica si se debe usar la configuración de fecha u hora de `application-main.yml` en la utilidad de descarga.

Tipo: booleano

Predeterminado: false

`unload.var Nulo Charls`

Opcional. Utilice este parámetro en el programa INFTILB, si se establece en `true`, y todos los campos que no admiten valores NULL con valores en blanco (espacios) devolverán una cadena vacía.

Tipo: booleano

Predeterminado: false

Otras propiedades

`qtemp.cleanup.threshold.hours`

Opcional. Para especificar cuándo está activado `qtemp.dblog`. La vida útil de la partición de base de datos (en horas).

Tipo: número

Predeterminado: 0

qtemp.dblog

Opcional. Si se debe habilitar el registro de la base de datos QTEMP.

Tipo: booleano

Predeterminado: false

qtemp.uuid.length

Opcional. La longitud de identificación única de QTEMP.

Tipo: número

Predeterminado: 9

quartz.scheduler.stand-by-if error

Opcional. Especifica si se debe activar la ejecución del trabajo si el programador de trabajos está en modo de espera. Si es verdadero, cuando está habilitada, no se activa la ejecución de tareas.

Tipo: booleano

Predeterminado: false

cálido UpCache

Opcional. Especifica si se deben cargar todos los datos de la tabla de comunicación de datos en una memoria caché de calentamiento al iniciar el servidor.

Tipo: booleano

Predeterminado: false

## AWS Referencia de definición de aplicaciones de modernización de mainframe

En AWS Mainframe Modernization, las aplicaciones de mainframe migradas se configuran en un archivo JSON de definición de aplicaciones, específico para el motor de ejecución que elija. La definición de una aplicación contiene tanto información general como información específica del



motor. En este tema se describen las definiciones de las aplicaciones AWS Blu Age y Micro Focus e identifica todos los elementos obligatorios y opcionales.

## Temas

- [Sección de cabecera general](#)
- [Descripción general de la sección de definiciones](#)
- [AWS Ejemplo de definición de aplicación de Blu Age](#)
- [AWS Detalles de la definición de Blu Age](#)
- [Definición de la aplicación Micro Focus](#)
- [Detalles de definición de Micro Focus](#)

## Sección de cabecera general

Cada definición de aplicación comienza con información general sobre la versión de la plantilla y las ubicaciones de origen. La versión actual de la definición de la aplicación es la 2.0.

Utilice la siguiente estructura para especificar la versión de la plantilla y las ubicaciones de origen.

```
"template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket",
        "s3-key-prefix": "v1"
      }
    }
  ]
```

### Note

Puede usar la siguiente sintaxis si desea introducir el ARN de S3 como s3-bucket:

```
"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"
    }
  }
]
```

### template-version

Obligatorio. Especifica la versión del archivo de definición de aplicaciones. No cambie este valor. La única permitida actualmente es la 2.0. Especifique `template-version` con una cadena.

### source-locations

Especifica las ubicaciones de los archivos y otros recursos que la aplicación necesita durante el tiempo de ejecución.

### id de origen

Especifica un nombre para la ubicación. Este nombre se utiliza para hacer referencia a la ubicación de origen según sea necesario en el JSON de definición de la aplicación.

### tipo de fuente

Especifica el tipo de fuente. Actualmente, el único valor permitido es `s3`.

### properties

Proporciona los detalles de la ubicación de origen. Cada propiedad se especifica con una cadena.

- `s3-bucket`: obligatorio. Especifica el nombre del bucket de Amazon S3 donde se almacenan los archivos.
- `s3-key-prefix`: obligatorio. Especifica el nombre de la carpeta del bucket de Amazon S3 donde se almacenan los archivos.

## Descripción general de la sección de definiciones

Especifica las definiciones de recursos de los servicios, la configuración, los datos y otros recursos típicos que la aplicación necesita para ejecutarse. Al actualizar la definición de una aplicación, AWS Mainframe Modernization detecta los cambios comparando las listas `source-locations` y `definition` de las versiones anterior y actual del archivo JSON de definición de la aplicación.

La sección de definición es específica del motor y está sujeta a cambios. En las siguientes secciones, se muestran ejemplos de definiciones de aplicaciones específicas de los motores para ambos motores.

## AWS Ejemplo de definición de aplicación de Blu Age

```
{
  "template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ],
  "definition" : {
    "listeners": [{
      "port": 8194,
      "type": "http"
    }],
    "ba-application": {
      "app-location": "${s3-source}/murachs-v6/"
    },
    "blusam": {
      "db": {
        "nb-threads": 8,
        "batch-size": 10000,
        "name": "blusam",
        "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:blusam-FfmXLG"
      },
      "redis": {
        "hostname": "blusam.c3geul.ng.0001.usw2.cache.amazonaws.com",
        "port": 6379,
        "useSsl": true,
        "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:bluesamredis-nioefm"
      }
    }
  }
}
```

```
}
```

## AWS Detalles de la definición de Blu Age

Oyente(s): obligatorio

Especifique el puerto que utilizará para acceder a la aplicación a través del Elastic Load Balancing creado por AWS Mainframe Modernization. Utilice la estructura siguiente:

```
"listeners": [{  
    "port": 8194,  
    "type": "http"  
}],
```

port

Obligatorio. Puede usar cualquier puerto disponible, excepto los puertos bien conocidos del 0 al 1023. Le recomendamos que utilice valores entre 8192 y 8199. Asegúrese de que no haya otros oyentes o aplicaciones funcionando en este puerto.

type

Obligatorio. En la actualidad, solo se admite http.

## AWS Aplicación Blue Age: necesaria

Especifique la ubicación en la que el motor recoge el archivo de imagen de la aplicación mediante la siguiente estructura.

```
"ba-application": {  
    "app-location": "${s3-source}/murachs-v6/",  
    "files-directory": "/m2/mount/myfolder",  
    "enable-jics": <true|false>,  
    "shared-app-location": "${s3-source}/shared/"  
},
```

app-location

La ubicación específica en Amazon S3 donde está almacenado el archivo de imagen de la aplicación.

## files-directory

Opcional. La ubicación de los archivos de entrada/salida de los lotes. Debe ser una subcarpeta de la configuración del punto de montaje de Amazon EFS o Amazon FSx a nivel de entorno. La subcarpeta debe ser propiedad de un usuario adecuado para que la utilice la aplicación Blu Age que se ejecuta en AWS Mainframe Modernization. Para ello, al conectar la unidad a una instancia Amazon EC2 de Linux, se debe crear un grupo con 101 ID y un usuario con 3001 ID, y la carpeta deseada debe ser propiedad de este usuario. Por ejemplo, de esta forma, Blu Age AWS Mainframe Modernization Managed puede utilizar la *testclient* carpeta.

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

## enable-jics

Opcional. Especifica si se habilita JICS. El valor predeterminado es true (verdadero). Si se establece en false, se evita que se genere la base de datos de JICS.

## shared-app-location

Opcional. Otra ubicación en Amazon S3 donde se almacenan los elementos compartidos de aplicaciones. Puede contener el mismo tipo de estructura de aplicación que la ubicación de la aplicación.

## BluSAM: opcional

Especifique la base de datos de BluSAM y la caché de Redis mediante la siguiente estructura.

```
"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

Especifica las propiedades de la base de datos utilizada con la aplicación. La base de datos debe ser una base de datos de Aurora PostgreSQL. Puede especificar las propiedades siguientes:

- `nb-threads`: opcional. Especifica cuántos subprocesos dedicados se utilizan para el mecanismo de escritura trasera en el que se basa el motor BluSAM. El valor predeterminado es 8.
- `batch-size`: opcional. Especifica el umbral que utiliza el mecanismo de escritura diferida para iniciar las operaciones de almacenamiento por lotes. El umbral representa el número de registros modificados que iniciarán una operación de almacenamiento por lotes para garantizar que los registros modificados se conserven. El desencadenante en sí se basa en una combinación del tamaño del lote y un tiempo transcurrido de un segundo, lo que se alcance primero. El valor predeterminado es 10000.
- `name`: opcional. Especifica el nombre de la base de datos.
- `secret-manager-arn`: especifica el nombre de recurso de Amazon (ARN) del secreto que contiene las credenciales de la base de datos. Para obtener más información, consulte [Paso 4: Crear y configurar un secreto de AWS Secrets Manager base de datos](#).

## Redis

Especifica las propiedades de la caché de Redis que la aplicación utiliza para almacenar los datos temporales que necesita en una ubicación central para mejorar el rendimiento. Le recomendamos que cifre y proteja con contraseña la caché de Redis.

- `hostname`: especifica la ubicación de la caché de Redis.
- `port`: especifica el puerto, normalmente el 6379, al que la caché de Redis envía y recibe las comunicaciones.
- `useSsl`: especifica si la caché de Redis está cifrada. Si lo está, establezca `useSsl` en `false`.
- `secret-manager-arn`: especifica el nombre de recurso de Amazon (ARN) del secreto que contiene la contraseña de la caché de Redis. Si la caché de Redis no está protegida con contraseña, no especifique `secret-manager-arn`. Para obtener más información, consulte [Paso 4: Crear y configurar un secreto de AWS Secrets Manager base de datos](#).

## AWS Colas de mensajes de Blu Age: opcionales

Especifique los detalles de la conexión JMS-MQ para AWS la aplicación Blu Age.

```
"message-queues": [  
  {  
    "product-type": "JMS-MQ",  
    "queue-manager": "QMGr1",  
    "channel": "mqChannel1",  
    "hostname": "mqserver-host1",  
    "port": 1414,  
    "user-id": "app-user1",  
    "secret-manager-arn": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:sample/mq/test-279PTa"  
  },  
  {  
    "product-type": "JMS-MQ",  
    "queue-manager": "QMGr2",  
    "channel": "mqChannel2",  
    "hostname": "mqserver-host2",  
    "port": 1412,  
    "user-id": "app-user2",  
    "secret-manager-arn": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:sample/mq/test-279PTa"  
  }  
]
```

### product-type

Obligatorio. Especifica el tipo de producto. Actualmente, solo puede ser «JMS-MQ» para AWS aplicaciones de Blu Age.

### queue-manager

Obligatorio. Especifica el nombre del administrador de colas.

### channel

Obligatorio. Especifica el nombre del canal de conexión al servidor.

### hostname

Obligatorio. Especifica el nombre de host del servidor de cola de mensajes.

**port**

Obligatorio. Especifica el número de puerto de oyente en el que escucha el servidor.

**user-id**

Opcional. Especifica el ID de cuenta de usuario autorizado para realizar operaciones de cola de mensajes en el canal especificado.

**secret-manager-arn**

Opcional. Especifica el nombre de recurso de Amazon (ARN) de Secrets Manager que proporciona la contraseña del usuario especificado.

**AWS Configuración de EFS de almacenamiento de aplicaciones Blu Age: opcional**

Especifique los detalles del punto de acceso EFS de almacenamiento de la aplicación mediante la siguiente estructura.

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

**ID del sistema de archivos**

Obligatorio. El ID del sistema de archivos EFS al que se aplica el punto de acceso. Patrón: «fs-([0-9a-f] {8,40}) {1,128} \$»

**punto de montaje**

Obligatorio. El punto de montaje del sistema de archivos a nivel de aplicación. Debe ser diferente del punto de montaje del almacenamiento a nivel de entorno.

**identificador del punto de acceso**

Obligatorio. El ID del punto de acceso, asignado por Amazon EFS. Patrón: «^fsap- ([0-9a-f] {8,40}) {1,128} \$»



## file-permission-mask

Opcional. Define la máscara de creación de archivos para los archivos creados por el proceso de solicitud. Por ejemplo, si el valor se establece en `UMASK006`, todos los archivos tendrán el permiso 660. Esto significa que solo el propietario del archivo y el grupo de archivos tendrán acceso de lectura y escritura, mientras que los demás usuarios no tendrán ningún permiso.

### Note

El valor establecido para este campo solo se tiene en cuenta cuando se utiliza el almacenamiento EFS a nivel de aplicación.

### Note

Cuando se proporciona la configuración `efs`, se debe especificar el directorio de archivos en la sección de definición de la aplicación. Debe ser una subcarpeta del punto de montaje de Amazon EFS configurada a nivel de aplicación.

## Definición de la aplicación Micro Focus

La siguiente sección de ejemplos de definición está dedicada al motor de tiempo de ejecución de Micro Focus y contiene elementos obligatorios y opcionales.

```
{
  "template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ],
  "definition" : {
    "listeners": [{
      "port": 5101,
```

```

    "type": "tn3270"
  ]],
  "dataset-location": {
    "db-locations": [{
      "name": "Database1",
      "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456"
    }]
  },
  "cognito-auth-handler": {
    "user-pool-id": "cognito-idp.us-west-2.amazonaws.com/us-west-2_rvYFnQIxL",
    "client-id": "58k05jb8grukjjsudm5hhn1v87",
    "identity-pool-id": "us-west-2:64464b12-0bfb-4dea-ab35-5c22c6c245f6"
  },
  "ldap-ad-auth-handler": {
    "ldap-ad-connection-secrets": [LIST OF AD-SECRETS]
  },
  "batch-settings": {
    "initiators": [{
      "classes": ["A", "B"],
      "description": "initiator...."
    }],
    "jcl-file-location": "${s3-source}/batch/jcl",
    "program-path": "/m2/mount/libs/loadlib:$EFS_MOUNT/emergency/loadlib",
    "system-procedure-libraries": "SYS1.PROCLIB;SYS2.PROCLIB",
    "aliases": [
      { "alias": "FDSSORT", "program": "SORT" },
      { "alias": "MFADRDSU", "program": "ADRDSU" }
    ]
  },
  "cics-settings": {
    "binary-file-location": "${s3-source}/cics/binaries",
    "csd-file-location": "${s3-source}/cics/def",
    "system-initialization-table": "BNKCICV"
  },
  "xa-resources" : [{
    "name": "XASQL",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
    "module": "${s3-source}/xa/ESPGSQLXA64.so"
  }],
  "runtime-settings": {
    "environment-variables": {
      "ES_JES_RESTART": "N",
      "EFS_MOUNT": "/m2/mount/efs"
    }
  }
}

```

```

    }
  }
}

```

## Detalles de definición de Micro Focus

El contenido de la sección de definición del archivo de definición de la aplicación de Micro Focus varía en función de los recursos que la aplicación de la unidad central migrada requiera en tiempo de ejecución.

### Oyente (es): obligatorio

Especifique un oyente mediante la siguiente estructura:

```

"listeners": [{
  "port": 5101,
  "type": "tn3270"
}],

```

#### port

En el caso de tn3270, el valor predeterminado es 5101. Para otros tipos de oyentes de servicios, el puerto varía. Puede usar cualquier puerto disponible, excepto los puertos bien conocidos del 0 al 1023. Cada oyente debe tener un puerto distintivo. Los oyentes no deben compartir puertos. Para obtener más información, consulte [Listener Control](#) en la documentación de Micro Focus Enterprise Server.

#### type

Especifica el tipo de oyente del servicio. Para obtener más información, consulte [Listeners](#) en la documentación de Micro Focus Enterprise Server.

## Ubicaciones de los conjuntos de datos: obligatorias

Especifique la ubicación del conjunto de datos mediante la siguiente estructura.

```

"dataset-location": {
  "db-locations": [{
    "name": "Database1",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456"
  }],
}

```

```
}

```

## db-locations

Especifica la ubicación de los conjuntos de datos que crea la aplicación migrada. Actualmente, la modernización AWS del mainframe solo admite conjuntos de datos de una única base de datos de VSAM.

- `name`: especifica el nombre de la instancia de base de datos que contiene los conjuntos de datos que crea la aplicación migrada.
- `secret-manager-arn`: especifica el nombre de recurso de Amazon (ARN) del secreto que contiene las credenciales de la base de datos.

## Controlador de autenticación y autorización de Amazon Cognito: opcional

AWS Mainframe Modernization utiliza Amazon Cognito para la autenticación y autorización de las aplicaciones migradas. Especifique el controlador de autenticación de Amazon Cognito mediante la siguiente estructura.

```
"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

Especifica el grupo de usuarios de Amazon Cognito que AWS Mainframe Modernization utiliza para autenticar a los usuarios de la aplicación migrada. El del grupo Región de AWS de usuarios debe coincidir con el de la aplicación Región de AWS de modernización del AWS mainframe.

## client-id

Especifica la aplicación migrada a la que puede acceder el usuario autenticado.

## identity-pool-id

Especifica el grupo de identidades de Amazon Cognito en el que el usuario autenticado intercambia un token del grupo de usuarios por credenciales que permiten al usuario acceder AWS a la modernización del mainframe. El del grupo Región de AWS de identidades debe coincidir con el de la aplicación Región de AWS de modernización del AWS mainframe.

## Controlador de LDAP y Active Directory: opcional

Puede integrar la aplicación con Active Directory (AD) o cualquier tipo de servidor LDAP para que los usuarios de la aplicación puedan utilizar sus credenciales de LDAP/AD para la autorización y la autenticación.

Para integrar su aplicación con AD

1. Siga los pasos descritos en [Configuring Active Directory for Enterprise Server Security](#), en la documentación de Micro Focus Enterprise Server.
2. Cree un AWS Secrets Manager secreto con los detalles de AD/LDAP para cada servidor de AD/LDAP que desee utilizar con la aplicación. Para obtener información sobre cómo crear un secreto, consulte [Crear un secreto de AWS Secrets Manager](#) en la Guía del AWS Secrets Manager usuario. Para el tipo de secreto, elija Otro tipo de secreto e incluya los siguientes pares clave-valor.

```
{
  "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>"
}
```

### Recomendaciones de seguridad

- **connectionPath** En efecto, la modernización del AWS mainframe es compatible con los protocolos LDAP y LDAP over SSL (LDAPS). Recomendamos utilizar el LDAPS porque es más seguro y evita que las credenciales aparezcan en las transmisiones de red.
- Para **authorizedId** y **password**, le recomendamos que especifique las credenciales de un usuario sin más permisos que los permisos de verificación y de solo lectura más restrictivos que se requieren para que la aplicación se ejecute.
- Le recomendamos que rote las credenciales AD/LDAP periódicamente.

- No cree usuarios de AD con el nombre de usuario `awsuser` o `mfuser`. Estos dos nombres de usuario están reservados para el uso de AWS .

A continuación, se muestra un ejemplo.

```
{
  "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"
}
```

Cree el secreto con una clave de KMS administrada por el cliente. Debe conceder a AWS Mainframe Modernization los `DescribeSecret` permisos `GetSecretValue` y sobre el secreto y los permisos sobre la clave KMS. `Decrypt DescribeKey` Para obtener más información, consulte [los permisos de la clave KMS](#) en la Guía del AWS Secrets Manager usuario.

3. Agregue el código siguiente a la definición de la aplicación.

```
"ldap-ad-auth-handler": {
  "ldap-ad-connection-secrets": [LIST OF AD/LDAP SECRETS]
}
```

A continuación, se muestra un ejemplo.

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

El controlador de autenticación LDAP/AD está disponible para Micro Focus 8.0.11 y versiones posteriores.

## Configuración de lotes: obligatoria

Especifique los detalles necesarios para los trabajos por lotes que se ejecutan como parte de la aplicación mediante la siguiente estructura.

```
"batch-settings": {
  "initiators": [{
    "classes": ["A", "B"],
    "description": "initiator...."
  }],
  "jcl-file-location": "${s3-source}/batch/jcl",
  "program-path": "/m2/mount/libs/loadlib:$EFS_MOUNT/emergency/loadlib",
  "system-procedure-libraries": "SYS1.PROCLIB;SYS2.PROCLIB",
  "aliases": [
    {"alias": "FDSSORT", "program": "SORT"},
    {"alias": "MFADRDSU", "program": "ADRDSU"}
  ]
}
```

### initiators

Especifica un iniciador de lotes que se inicia cuando la aplicación migrada se inicia correctamente y continúa ejecutándose hasta que la aplicación se detiene. Puede definir una o más clases por iniciador. También puede definir varios iniciadores. Por ejemplo:

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

Para obtener más información, consulte [To define a batch initiator or printer SEP](#) en la documentación de Micro Focus Enterprise Server.

- `classes`: especifica las clases de trabajo que puede ejecutar el iniciador. Puede usar hasta 36 caracteres. Puede usar los siguientes caracteres: A-Z o 0-9.
- `description`: describe para qué sirve el iniciador.

### jcl-file-location

Especifica la ubicación de los archivos JCL (Job Control Language) que requieren los trabajos por lotes que ejecuta la aplicación migrada.

### ruta del programa

Especifica la ruta necesaria para ejecutar trabajos por lotes cuando un programa de una JCL no se encuentra en la ubicación predeterminada. Los nombres de las distintas rutas se separan con dos puntos (:).

#### Note

La ruta del programa solo puede ser una ruta EFS.

### bibliotecas de procedimientos del sistema

Especifica los conjuntos de datos particionados predeterminados en los que se buscarán los procedimientos de JCL. Sin embargo, el procedimiento no se encuentra en la JCL ni en las sentencias de la JCLLIB. Estos conjuntos de datos deben estar catalogados y debe usarse el nombre del catálogo. Y las entradas se separan con punto y coma (;).

### alias

Define un mapeo entre los nombres de las utilidades y los programas utilizados en JCL y el nombre de implementación de la utilidad. AWS y las utilidades por lotes de terceros (por ejemplo, M2SFTP, M2WAIT, Syncsort, etc.) pueden tener alias de forma opcional para eliminar la necesidad de cambiar el JCL. Por ejemplo:

- FDSSORT Alias FDSSORT para SORT y Alias FDSICET para ICETOOL
- ADRDSSU Alias MFADRDSU para ADRDSSU
- Alias de Syncsort DMXMFSRT para SORT



## Configuración de CICS: necesaria

Especifique los detalles necesarios para las transacciones de CICS que se ejecutan como parte de la aplicación mediante la siguiente estructura.

```
"cics-settings": {
  "binary-file-location": "${s3-source}/cics/binaries",
  "csd-file-location": "${s3-source}/cics/def",
  "system-initialization-table": "BNKCICV"
}
```

### binary-file-location

Especifica la ubicación de los archivos del programa de transacciones CICS.

### csd-file-location

Especifica la ubicación del archivo de definición de recursos (CSD) del CICS para esta aplicación. Para obtener más información, consulte [CICS Resource Definitions](#) en la documentación de Micro Focus Enterprise Server.

### system-initialization-table

Especifica la tabla de inicialización del sistema (SIT) que utiliza la aplicación migrada. El nombre de la tabla SIT puede tener hasta 8 caracteres. Puede usar A-Z, 0-9, \$, @ y #. Para obtener más información, consulte [CICS Resource Definitions](#) en la documentación de Micro Focus Enterprise Server.

## Recursos de XA: obligatorio

Especifique los detalles necesarios para los recursos de XA que requiere la aplicación mediante la siguiente estructura.

```
"xa-resources" : [{
  "name": "XASQL",
  "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
  "module": "${s3-source}/xa/ESPGSQLXA64.so"
}]
```

### name

Obligatorio. Especifica el nombre del recurso de XA.

## secret-manager-arn

Especifica el nombre de recurso de Amazon (ARN) del secreto que contiene las credenciales para conectarse a la base de datos.

## module

Especifica la ubicación del archivo ejecutable del módulo conmutador RM. Para obtener más información, consulte [Planning and Designing XARs](#) en la documentación de Micro Focus Enterprise Server.

## Configuración de tiempo de ejecución: opcional

Especifique los detalles necesarios para que la configuración del tiempo de ejecución gestione las variables de entorno permitidas mediante la siguiente estructura.

```
"runtime-settings": {
  "environment-variables": {
    "ES_JES_RESTART": "N",
    "EFS_MOUNT": "/m2/mount/efs"
  }
}
```

## variables de entorno

Especifica las variables de entorno compatibles con Micro Focus que se aplican al tiempo de ejecución de esta aplicación.

- **ES\_JES\_RESTART** Es una variable de entorno de Micro Focus que permite reiniciar el procesamiento de JCL. Opcionalmente, también se puede utilizar **ES\_ALLOC\_OVERRIDE** como variable de entorno de Micro Focus.
- **EFS\_MOUNT** Es una variable de entorno personalizada que la aplicación puede usar para identificar dónde se encuentra el montaje de EFS del entorno.

Puede acceder a todas las [variables de entorno de Micro Focus](#) en la guía Micro Focus Enterprise Server para UNIX.

# AWS Referencia de definición de conjuntos de datos de modernización de mainframe

Si su aplicación requiere más de unos pocos conjuntos de datos para su procesamiento, introducirlos uno por uno en la consola de modernización del AWS mainframe resulta ineficiente. En lugar de eso, se recomienda crear un archivo JSON para especificar cada conjunto de datos. Los distintos tipos de conjuntos de datos se especifican de forma diferente en el JSON, aunque muchos parámetros son comunes. En este documento se describen los detalles del JSON necesarios para importar diferentes tipos de conjuntos de datos.

## Note

Antes de importar un conjunto de datos, debe transferirlo de la unidad central a AWS. Luego, debe asegurarse de que los conjuntos de datos se conviertan del formato de mainframe a un formato que AWS pueda utilizar. Si es necesario, transforme los datos según sea necesario y almacene los conjuntos de datos transformados en Amazon S3. Especifique el nombre del bucket y la carpeta en el archivo JSON de definición del conjunto de datos.

Si utiliza el motor de tiempo de ejecución de Micro Focus, puede utilizar la utilidad DFCONV para convertir los conjuntos de datos. Incluimos esta utilidad en nuestras imágenes de Micro Focus Enterprise Developer y Enterprise Server. Para obtener más información, consulte [Conversión de archivos de lote DFCONV](#) en la documentación de Micro Focus Enterprise Developer.

## Temas

- [Propiedades comunes](#)
- [Ejemplo de formato de solicitud de conjunto de datos para VSAM](#)
- [Ejemplo de formato de solicitud de conjunto de datos para base GDG](#)
- [Ejemplo de formato de solicitud de conjunto de datos para las generaciones PS o GDG](#)
- [Ejemplo de formato de solicitud de conjunto de datos para PO](#)

## Propiedades comunes

Varios parámetros son comunes en todos los conjuntos de datos. Estos parámetros cubren las siguientes áreas:

- Información sobre el conjunto de datos (`datasetName`, `datasetOrg`, `recordLength`, `encoding`).
- Información sobre la ubicación desde la que está importando, es decir, la ubicación de origen del conjunto de datos. Esta no es la ubicación de la unidad central. Es la ruta a la ubicación de Amazon S3 en la que cargó el conjunto de datos (`externalLocation`).
- Información sobre la ubicación a la que está importando, es decir, la ubicación de destino del conjunto de datos. Esta ubicación puede ser una base de datos o un sistema de archivos, según el motor de tiempo de ejecución. (`storageType` y `relativePath`).
- Información sobre el tipo de conjunto de datos (tipo de conjunto de datos específico, formato, codificación, etc.).

Cada definición de conjunto de datos tiene la misma estructura JSON. El siguiente ejemplo de JSON muestra todos estos parámetros comunes.

```
{
  "dataSet": {
    "storageType": "Database",
    "datasetName": "MFI01V.MFIDEMO.BNKACC",
    "relativePath": "DATA",
    "datasetOrg": {
      "type": {
        type-specific properties
        ...
      },
    },
  },
},
}
```

Las siguientes propiedades son comunes a todos los conjuntos de datos.

### storageType

Obligatorio. Se aplica a la ubicación de destino. Especifica si el conjunto de datos se almacena en una base de datos o en un sistema de archivos. Los valores posibles son `Database` o `FileSystem`.

- AWS Motor de ejecución Blu Age: los sistemas de archivos no son compatibles. Debe utilizar una base de datos.

- Motor de ejecución de Micro Focus: se admiten bases de datos y sistemas de archivos. Puede utilizar Amazon Relational Database Service o Amazon Aurora para las bases de datos, y Amazon Elastic File System o Amazon FSx para Lustre para los sistemas de archivos.

#### datasetName

Obligatorio. Especifica el nombre completo del conjunto de datos tal como aparece en la unidad central.

#### relativePath

Obligatorio. Se aplica a la ubicación de destino. Especifica la ubicación relativa del conjunto de datos en la base de datos o el sistema de archivos.

#### datasetOrg

Obligatorio. Especifica el tipo de conjunto de datos. Los valores posibles son vsam, gdg, ps, po o unknown.

- AWS Motor de ejecución Blu Age: solo se admiten conjuntos de datos de tipo VSAM.
- Motor de tiempo de ejecución de Micro Focus: se admiten conjuntos de datos VSAM, GDG, PS, PO o de tipo desconocido.

#### Note

Si la aplicación requiere archivos que no sean archivos de datos COBOL, sino archivos PDF u otros archivos binarios, puede especificarlos de la siguiente manera:

```
"datasetOrg": {  
  "type": PS {  
    "format": U  
  },  
}
```

## Ejemplo de formato de solicitud de conjunto de datos para VSAM

- AWS Motor de tiempo de ejecución Blu Age: compatible.
- Motor de tiempo de ejecución de Micro Focus: compatible.

Si va a importar conjuntos de datos de VSAM, especifique `vsam` como `datasetOrg`. El JSON debe parecerse al siguiente ejemplo:

```
{
  "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"
}
}
```

Los conjuntos de datos de VSAM admiten las siguientes propiedades.

#### encoding

Obligatorio. Especifica la codificación de conjunto de caracteres del conjunto de datos. Los valores posibles son ASCII (A), EBCDIC (E) y Unknown (?).

#### format

Obligatorio. Especifica el tipo de conjunto de datos de VSAM y el formato de registro.

- AWS Motor de ejecución Blu Age: los valores posibles son ESDS (ES), KSDS (KS) y RRDS (). RR El formato de registro puede ser fijo o variable.
- Motor de tiempo de ejecución de Micro Focus: los valores posibles son ESDS (ES), KSDS (KS) y RRDS (RR). La definición de VSAM incluye el formato de registro, por lo que no es necesario especificarlo por separado.

## primaryKey

Se aplica únicamente a los conjuntos de datos KSDS de VSAM. Especifica la clave principal. Consta del nombre de la clave principal, el desplazamiento y la longitud de la clave. `name` es opcional; `offset` y `length` son obligatorias.

## recordLength

Obligatorio. Especifica la longitud de un registro. Para los formatos de registro de longitud fija, estos valores deben coincidir.

- AWS Motor de tiempo de ejecución Blu Age: para VSAM ESDS, KSDS y RRDS, es opcional y obligatorio. `min` `max`
- Motor de tiempo de ejecución de Micro Focus: `min` y `max` son obligatorios.

## externalLocation

Obligatorio. Especifica la ubicación de origen: es decir, el bucket de Amazon S3 en el que cargó el conjunto de datos.

## Propiedades específicas del motor de Blue Age

El motor de tiempo de ejecución AWS Blu Age admite la compresión de conjuntos de datos VSAM. En el siguiente ejemplo se muestra cómo se puede especificar esta propiedad en JSON.

```
{
  common properties
  ...
  "datasetOrg": {
    "vsam": {
      common properties
      ...
      "compressed": boolean,
      common properties
      ...
    }
  }
}
```

Especifique la propiedad de compresión de la siguiente manera:

## compression

Opcional. Especifica si los índices de este conjunto de datos se almacenan como valores comprimidos. Si tiene un conjunto de datos grande (normalmente > 100 Mb), considere configurar este indicador en `true`.

## Ejemplo de formato de solicitud de conjunto de datos para base GDG

- AWS Motor de tiempo de ejecución Blu Age: no compatible.
- Motor de tiempo de ejecución de Micro Focus: compatible.

Si va a importar conjuntos de datos base GDG, especifique `gdg` como `datasetOrg`. El JSON debe parecerse al siguiente ejemplo:

```
{
  "storageType": "Database",
  "datasetName": "AWS.M2.GDG",
  "relativePath": "DATA",
  "datasetOrg": {
    "gdg": {
      "limit": "3",
      "rollDisposition": "Scratch and No Empty"
    }
  }
}
```

Los conjuntos de datos base de GDG admiten las siguientes propiedades.

### límite

Obligatorio. Especifica el número de generaciones activas, o sesgos. Para un clúster base GDG, el máximo es 255.

### rollDisposition

Opcional. Especifica cómo gestionar los conjuntos de datos de generación cuando se alcanza o se supera el máximo. Los valores posibles son `No Scratch and No Empty`, `Scratch and No Empty`, `Scratch and Empty` o `No Scratch and Empty`. El valor predeterminado es `Scratch and No Empty`.



## Ejemplo de formato de solicitud de conjunto de datos para las generaciones PS o GDG

- AWS Motor de tiempo de ejecución Blu Age: no compatible.
- Motor de tiempo de ejecución de Micro Focus: compatible.

Si va a importar conjuntos de datos de generación PS o GDG, especifique `ps` como `datasetOrg`. El JSON debe parecerse al siguiente ejemplo:

```
{
  "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"
  }
}
```

Los conjuntos de datos de generaciones de PS o GDG admiten las siguientes propiedades.

### format

Obligatorio. Especifica formato de los registros del conjunto de datos. Los posibles valores son F, FA, FB, FBA, FBM, FBS, FM, FS, LSEQ, U, V, VA, VB, VBA, VBM, VBS, VM y VS.

### encoding

Obligatorio. Especifica la codificación de conjunto de caracteres del conjunto de datos. Los valores posibles son ASCII (A), EBCDIC (E) y Unknown () ?.

## recordLength

Obligatorio. Especifica la longitud de un registro. Debe especificar la longitud mínima (min) y máxima (max) del registro. Para los formatos de registro de longitud fija, estos valores deben coincidir.

## externalLocation

Obligatorio. Especifica la ubicación de origen: es decir, el bucket de Amazon S3 en el que cargó el conjunto de datos.

## Ejemplo de formato de solicitud de conjunto de datos para PO

Si va a importar conjuntos de datos de PO, especifique po como datasetOrg. El JSON debe parecerse al siguiente ejemplo:

```
{
  "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/"
}
}
```

Los conjuntos de datos de PO admiten las siguientes propiedades.

## format

Obligatorio. Especifica formato de los registros del conjunto de datos. Los posibles valores son F, FA, FB, FBA, FBM, FBS, FM, FS, LSEQ, U, V, VA, VB, VBA, VBM, VBS, VM y VS.

## encoding

Obligatorio. Especifica la codificación de conjunto de caracteres del conjunto de datos. Los valores posibles son ASCII (A), EBCDIC (E) y Unknown (?).

## memberFileExtensions

Obligatorio. Especifica una matriz que contiene una o más extensiones de nombre de archivo, lo que le permite especificar qué archivos se van a incluir como miembros del PDS.

## recordLength

Opcional. Especifica la longitud de un registro. Tanto la longitud mínima (min) como la máxima (max) del registro son opcionales. Para los formatos de registro de longitud fija, estos valores deben coincidir.

## externalLocation

Obligatorio. Especifica la ubicación de origen: es decir, el bucket de Amazon S3 en el que cargó el conjunto de datos.

### Note

La implementación actual del motor de tiempo de ejecución de Micro Focus añade entradas de PDS como conjuntos de datos dinámicos.

# Entornos de tiempo de ejecución gestionados en la AWS modernización de mainframes

Si no tiene experiencia en la modernización de AWS mainframes, consulte los siguientes temas para empezar:

- [¿Qué es la modernización del AWS mainframe?](#)
- [Configuración de la modernización del AWS mainframe](#)
- [Introducción a la modernización del AWS mainframe](#)
- [Tutorial: Tiempo de ejecución gestionado para AWS Blu Age](#)
- [Tutorial: Tiempo de ejecución administrado para Micro Focus](#)

Un entorno de tiempo de ejecución en la modernización de AWS mainframe es una combinación determinada de recursos de AWS procesamiento, un motor de tiempo de ejecución y los detalles de configuración que especifique. El entorno de tiempo de ejecución aloja una o más aplicaciones. Las aplicaciones de AWS Mainframe Modernization contienen cargas de trabajo de mainframe migradas. Puede elegir el motor de tiempo de ejecución para los entornos que cree. Elija AWS Blu Age si utiliza el patrón de refactorización automática y Micro Focus si utiliza el patrón de replataforma. También puede elegir la cantidad de recursos informáticos adecuada para su aplicación y, de forma opcional, conectar el almacenamiento a los entornos de tiempo de ejecución. AWS La modernización del mainframe le permite CloudWatch utilizar las métricas y el registro de Amazon para que pueda supervisar su entorno de ejecución.

## Temas

- [Cree un entorno de ejecución de modernización AWS de mainframe](#)
- [Actualice un entorno de AWS ejecución de modernización de mainframe](#)
- [Detenga un entorno de AWS ejecución de modernización de mainframe](#)
- [Reinicie un entorno de ejecución AWS de modernización de mainframe](#)
- [Elimine un entorno de ejecución AWS de modernización de mainframe](#)

# Cree un entorno de ejecución de modernización AWS de mainframe

Utilice la consola de modernización de AWS mainframe para crear un entorno de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

## Creación de un entorno de tiempo de ejecución

Para crear un entorno de tiempo de ejecución

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que desee crear el entorno.
3. En la página Entornos, elija Crear entorno.
4. En la página Especificaciones de inferencia proporcione la siguiente información:
  - a. En la sección Nombre y descripción, introduzca un nombre para el entorno.
  - b. (Opcional) En el campo Descripción del entorno, introduzca una descripción para el entorno. Esta descripción ayudarle a usted y a otros usuarios a identificar el propósito del entorno de tiempo de ejecución.
  - c. En la sección Opciones del motor, seleccione Blu Age para la refactorización automática o Micro Focus para redefinir la plataforma.
  - d. Elija una versión para el motor que ha seleccionado.
  - e. (Opcional) En la sección Etiquetas, seleccione Agregar nueva etiqueta para añadir una o varias etiquetas al entorno. Una etiqueta de entorno es un atributo personalizado que le ayuda a organizar y administrar los recursos de AWS .
  - f. Elija Siguiente.
5. En la página Especificaciones de ajuste, proporcione la siguiente información:
  - a. En la sección Disponibilidad, elija un Entorno en tiempo de ejecución independiente o un Clúster de alta disponibilidad.

El patrón de disponibilidad determina la disponibilidad de la aplicación cuando se ejecute. La versión independiente está bien para fines de desarrollo. La de alta disponibilidad es para las aplicaciones que deben estar disponibles en todo momento.


- b. En Recursos, elija un tipo de instancia y la capacidad deseada.

Estos recursos son las instancias Amazon EC2 gestionadas por la modernización del AWS mainframe que alojarán su entorno de ejecución. Los entornos de tiempo de ejecución independientes ofrecen dos opciones de tipo de instancia y solo permiten una instancia. Los entornos de tiempo de ejecución de alta disponibilidad ofrecen dos opciones para el tipo de instancia y permiten hasta dos instancias.

Para obtener más información, consulte los [tipos de instancias de Amazon EC2](#) y póngase en contacto con un especialista en AWS mainframe para obtener orientación.

6. En la sección Seguridad y red, realice lo siguiente:

- a. Si desea que las aplicaciones sean de acceso público, seleccione Allow applications deployed to this environment to be publicly accessible.
- b. Elija una nube virtual privada (VPC).
- c. Si utiliza el patrón de alta disponibilidad, elija dos o más subredes. Si utiliza el patrón independiente con el motor AWS Blu Age, elija dos o más subredes. Si usa un patrón independiente con el motor Micro Focus, especifique una subred.
- d. Elija un grupo de seguridad para la VPC que ha seleccionado.

 Note

AWS La modernización del mainframe crea un Network Load Balancer para que pueda distribuir las conexiones a su entorno de ejecución. Asegúrese de que las reglas de entrada de su grupo de seguridad permitan el acceso desde una dirección IP al puerto que especificó en la propiedad `listener` de la definición de la aplicación. Para obtener más información, consulte [Registro de destinos](#) en la Guía del usuario de Equilibradores de carga de red.

- e. En el campo de clave KMS, seleccione Personalizar la configuración de cifrado si desea utilizar un sistema gestionado por el cliente. AWS KMS key Para obtener más información, consulte [Cifrado de datos en reposo para el servicio de AWS modernización de mainframe](#).

**Note**

De forma predeterminada, AWS Mainframe Modernization cifra sus datos con un archivo AWS KMS key que AWS Mainframe Modernization posee y administra por usted. Sin embargo, puede optar por utilizar una AWS KMS key administrada por el cliente.

- f. (Opcional) Elija un AWS KMS key por nombre o nombre de recurso de Amazon (ARN). También puede elegir Crear un AWS KMS key para ir a la AWS KMS consola y crear uno nuevo. AWS KMS key
  - g. Elija Siguiente.
7. (Opcional) En la página Adjuntar almacenamiento, elija uno o más sistemas de archivos de Amazon EFS o Amazon FSx y, a continuación, elija Siguiente.
  8. En la sección Periodo de mantenimiento, elija cuándo desea aplicar los cambios pendientes al entorno.
    - Si selecciona Ninguna preferencia, AWS Mainframe Modernization seleccionará un intervalo de mantenimiento optimizado para usted.
    - Si desea especificar un periodo de mantenimiento concreto, elija Seleccionar nuevo periodo de mantenimiento. Puede elegir el día de la semana, una hora de inicio y la duración del periodo de mantenimiento.

Para obtener más información sobre los períodos de mantenimiento, consulte [AWS Ventana de mantenimiento y modernización del mainframe](#).

Elija Siguiente.

9. En la página Revisar y crear, revise la información que ha proporcionado y, a continuación, elija Crear entorno.

## Actualice un entorno de AWS ejecución de modernización de mainframe

Utilice la consola de modernización de AWS mainframe para actualizar un entorno de ejecución de modernización de AWS mainframe. Puede actualizar la versión secundaria del motor de tiempo de

ejecución o el tipo de instancia que aloja el entorno de tiempo de ejecución. Puede elegir si aplicar las actualizaciones inmediatamente o durante el periodo de mantenimiento que prefiera.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

## Actualizar un entorno de tiempo de ejecución

Para actualizar un entorno de tiempo de ejecución

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó el entorno que desea actualizar.
3. En la página Entornos, elija el nombre del entorno que desea actualizar.
4. En la página de detalles del entorno, elija Acciones y, a continuación, seleccione Editar entorno.
5. Realice cualquiera de los siguientes cambios:
  - En la sección Opciones de motor, elija la versión de motor que desee.
  - En la sección Recursos, elija el tipo de instancia que desee.
  - En la sección de la Periodo de mantenimiento, elija el día, la hora y la duración que desee.

### Note

Los únicos cambios que puede optar por aplicar durante el periodo de mantenimiento son los cambios en la versión del motor. Debe aplicar todos los demás cambios inmediatamente.

6. Elija Siguiente.
7. En When to apply these changes, elija Inmediatamente o Durante el siguiente periodo de mantenimiento. A continuación, seleccione Actualizar entorno.

Si elige Inmediatamente, verá un mensaje cuando el entorno haya terminado de actualizarse.

## AWS Ventana de mantenimiento y modernización del mainframe

Cada entorno de ejecución tiene un período de mantenimiento semanal de dos horas. Todos los cambios en el sistema se aplican durante este tiempo. El período de mantenimiento es su



oportunidad de controlar cuándo se realizan las modificaciones y los parches de software y seguridad. Si un evento de mantenimiento está programado para una semana determinada, comienza durante ese período de mantenimiento de dos horas. La mayoría de los eventos de mantenimiento también se completan durante el período de mantenimiento de dos horas, aunque los eventos de mantenimiento más grandes pueden tardar más de un par de horas en completarse.

El período de mantenimiento de dos horas se selecciona al azar de un bloque de tiempo de 8 horas por región. Si no especifica un período de mantenimiento al crear un entorno de ejecución, AWS Mainframe Modernization asigna un período de mantenimiento de 2 horas en un día de la semana seleccionado al azar.

AWS La modernización del mainframe consume algunos de los recursos de la instancia de su entorno mientras se realiza el mantenimiento. Es posible que observe un efecto mínimo en el rendimiento o algunas interrupciones en las aplicaciones durante el mantenimiento.

En la siguiente tabla se muestran los bloques de tiempo de cada región desde los que se asignan los periodos de mantenimiento predeterminados.

Nombre de la región	Región	Bloque de tiempo
Este de EE. UU. (Norte de Virginia)	us-east-1	03:00–11:00 UTC
Oeste de EE. UU. (Oregón)	us-west-2	06:00–14:00 UTC
Asia-Pacífico (Bombay)	ap-south-1	06:00–14:00 UTC
Asia Pacífico (Singapur)	ap-southeast-1	14:00 — 22:00 UTC
Asia Pacífico (Sídney)	ap-southeast-2	12:00 — 20:00 UTC
Asia Pacífico (Tokio)	ap-northeast-1	13:00 — 21:00 UTC
Canadá (centro)	ca-central-1	03:00 — 11:00 UTC
Europa (Fráncfort)	eu-central-1	21:00 — 05:00 UTC

Nombre de la región	Región	Bloque de tiempo
Europe (Ireland)	eu-west-1	22:00 — 06:00 UTC
Europe (London)	eu-west-2	22:00 — 06:00 UTC
Europe (Paris)	eu-west-3	23:59–07:29 UTC
América del Sur (São Paulo)	sa-east-1	00:00–08:00 UTC

## Detenga un entorno de AWS ejecución de modernización de mainframe

Utilice la consola de modernización de AWS mainframe para detener un entorno de ejecución de modernización de AWS mainframe. Al detener un entorno, se conservan las implementaciones de aplicaciones actuales y no se le cobrará por el entorno hasta que se reinicie.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Detener un entorno de tiempo de ejecución

Si necesita detener un entorno de ejecución de modernización de AWS mainframe, siga los mismos pasos que en la sección de actualización del entorno.

Utilice la consola de modernización de AWS mainframe para detener un entorno de ejecución de modernización de AWS mainframe. Al detener un entorno, se conservan las implementaciones de aplicaciones actuales y no se le cobrará por el entorno hasta que se reinicie.

### Detener un entorno de tiempo de ejecución

Para detener un entorno de ejecución de modernización de AWS mainframe, siga pasos similares a los de la sección de actualización del entorno.

#### Note

Debe detener todas las aplicaciones antes de detener el entorno.

## Para detener un entorno de tiempo de ejecución

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó el entorno que desea detener.
3. En la página Entornos, elija el nombre del entorno que desea detener.
4. En la página de detalles del entorno, elija Acciones y, a continuación, seleccione Editar entorno.
5. En la página Editar entorno, busque la sección Recursos y actualice la capacidad deseada a cero.

### Note

Para detener un entorno, solo puede optar por detenerlo inmediatamente.

6. Elija Siguiente.
7. En When to apply these changes, elija Inmediatamente. A continuación, seleccione Actualizar entorno.

Aparece un mensaje cuando se actualiza la capacidad del entorno.

## Reinicie un entorno de ejecución AWS de modernización de mainframe

Utilice la consola de modernización de AWS mainframe para reiniciar un entorno de ejecución de modernización de AWS mainframe. Al reiniciar un entorno de ejecución, se reanuda la facturación del entorno.

## Reiniciar un entorno de tiempo de ejecución

Para reiniciar un entorno de ejecución de modernización de AWS mainframes, siga pasos similares a los de la sección de parada del entorno.

Para reiniciar un entorno de tiempo de ejecución

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó el entorno que desea reiniciar.
3. En la página Entornos, elija el nombre del entorno que desea reiniciar.

4. En la página de detalles del entorno, elija Acciones y, a continuación, seleccione Editar entorno.

 Note

La capacidad deseada para un entorno independiente solo se puede actualizar a 1. Para reiniciar un entorno de tiempo de ejecución, solo puede elegir reiniciarlo inmediatamente.

5. En la página Editar entorno, busque la sección Recursos y actualice la capacidad deseada de cero a la que desee.
6. Elija Siguiente.
7. En When to apply these changes, elija Inmediatamente. A continuación, seleccione Actualizar entorno.

Aparece un mensaje cuando se actualiza la capacidad del entorno y se reinicia.

## Elimine un entorno de ejecución AWS de modernización de mainframe

Utilice la consola de modernización de AWS mainframe para eliminar un entorno de ejecución de modernización de AWS mainframe.

En estas instrucciones se presupone que ha completado los pasos que se detallan en [Configuración de la modernización del AWS mainframe](#).

### Eliminar un entorno de tiempo de ejecución

Si necesita eliminar un entorno de ejecución de AWS Mainframe Modernization, asegúrese de eliminar primero todas las aplicaciones implementadas del entorno. No puede eliminar un entorno de tiempo de ejecución en el que se implementan las aplicaciones.

Para eliminar un entorno

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región en la que se creó el entorno que desea eliminar.
3. En la página Entornos, elija el entorno que desee eliminar y, a continuación, elija Acciones y Eliminar entorno.

4. En la ventana Eliminar entorno, introduzca `delete` para confirmar que desea eliminar el entorno de tiempo de ejecución y, a continuación, seleccione Eliminar.

# Pruebas de aplicaciones en la modernización de AWS mainframes

AWS Las pruebas de aplicaciones se encuentran en la versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.

AWS Las pruebas de aplicaciones de modernización de mainframes proporcionan pruebas automatizadas de equivalencia funcional para sus proyectos de migración.

## Temas

- [¿Qué son las pruebas de aplicaciones de modernización de AWS mainframe?](#)
- [AWS Conceptos de pruebas de aplicaciones de modernización de mainframe](#)
- [Tutorial: configurar la aplicación CardDemo de muestra](#)
- [Tutorial: Prueba de aplicaciones de modernización de AWS mainframe, reproducción y comparación con el uso CardDemo de AWS Blu Age implementado en Amazon EC2](#)
- [AWS Modernización de mainframe, pruebas de aplicaciones, conjuntos de datos compatibles y páginas de códigos](#)

## ¿Qué son las pruebas de aplicaciones de modernización de AWS mainframe?

AWS Las pruebas de aplicaciones se encuentran en una versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.

Las pruebas tienen un impacto significativo en los proyectos de migración. Puede consumir hasta un 70% del tiempo y el esfuerzo de su proyecto de migración, modernización o ampliación. AWS Las pruebas de aplicaciones, una función de la modernización del AWS mainframe, proporcionan pruebas automatizadas de equivalencia funcional para las aplicaciones migradas. Las pruebas

de equivalencia funcional le ayudan a validar que las aplicaciones del mainframe Nube de AWS son equivalentes a las aplicaciones del mainframe. AWS Las pruebas de aplicaciones comparan automáticamente los cambios en los conjuntos de datos, los registros de bases de datos y las 3270 pantallas en línea entre su mainframe y. AWS Además, Application Testing permite realizar pruebas repetibles, por lo que puede ejecutar sus escenarios de prueba muchas veces a medida que actualice la arquitectura de destino, resuelva problemas y avance hacia una aplicación completamente migrada. Tras la migración, puede seguir utilizando Application Testing para realizar pruebas de regresión, a fin de garantizar que las actualizaciones de los motores del tiempo de ejecución u otros componentes no provoquen regresiones. Las pruebas de aplicaciones son rentables: los entornos de prueba objetivo se crean utilizando las CloudFormation plantillas proporcionadas por el usuario, aprovechando los conceptos de infraestructura como código (IaC). Application Testing acelera los proyectos de migración gracias a la elasticidad de la nube. Puede ejecutar escenarios de prueba independientes en tantos entornos paralelos como sea necesario, lo que reduce los plazos de las pruebas.

## Temas

- [¿Es la primera vez que usa Application Testing?](#)
- [Ventajas de Application Testing](#)
- [Integración con AWS CloudFormation](#)
- [Cómo funciona Application Testing](#)
- [Servicios relacionados](#)
- [Acceso a Application Testing](#)
- [Precios de Application Testing](#)

## ¿Es la primera vez que usa Application Testing?

Si es la primera vez que usa Application Testing, le recomendamos que empiece leyendo las siguientes secciones:

- [Conceptos de Application Testing](#)
- [Tutorial: Configuración CardDemo](#)

## Ventajas de Application Testing

Application Testing ofrece varias ventajas que le ayudarán en el proceso de migración:

- Aceleración de las pruebas, agilidad y flexibilidad
- Conceptos de prueba: “Registrar una vez en el mainframe y reproducir varias veces en AWS”
- IaC: creación de CloudFormation entornos de destino mediante plantillas proporcionadas por los usuarios
- Altos grados de repetibilidad de las pruebas
- Diseñado para la nube, pensando en la escalabilidad y la elasticidad
- Pruebas a gran escala con un alto grado de automatización
- Rentabilidad

## Integración con AWS CloudFormation

Las pruebas de aplicaciones utilizan la infraestructura como código con AWS CloudFormation. Esta elección de diseño simplifica y mejora su experiencia de prueba. AWS CloudFormation le brinda autonomía e independencia para definir la mejor infraestructura para sus necesidades. Puede seleccionar o definir varios parámetros (tamaño de instancia, instancia de RDS, grupo de seguridad óptimo) de forma independiente. Puede agregar recursos, como una cola de Amazon SQS, que necesite para que su aplicación funcione correctamente en condiciones de prueba.

En las AWS CloudFormation plantillas que se proporcionan para su descarga, observará algunas características comunes:

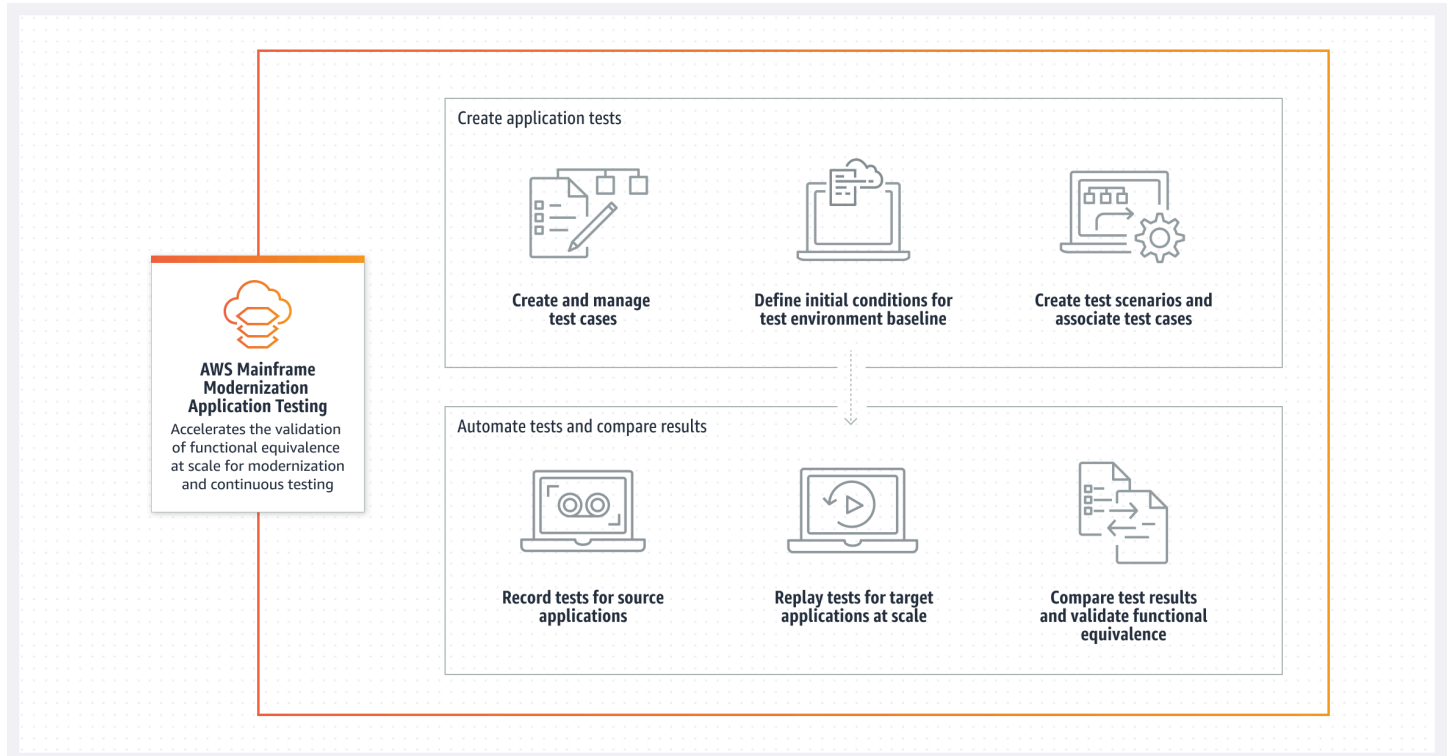
- Las pruebas de aplicaciones crean una pila completamente aislada, que incluye una aplicación y un entorno de ejecución de modernización del AWS mainframe, con sus propias definiciones de red y seguridad. Este conjunto aislado proporciona resiliencia, ya que otros actores del mismo sistema Cuenta de AWS no pueden interferir con la actividad de prueba. También evita situaciones en las que los operadores del sistema modifiquen la VPC o el grupo de seguridad predeterminados, lo que puede provocar errores en las actividades de prueba.
- El grupo de seguridad también permite controlar el acceso externo a los recursos utilizados en las pruebas. Por ejemplo, una base de datos puede contener datos confidenciales.
- El aislamiento total evita que otros actores que comparten la VPC espíen el tráfico.
- Mejora el rendimiento. Por ejemplo, la comunicación entre la aplicación de modernización del AWS mainframe que crea la plantilla y su base de datos de Amazon RDS se produce en una red independiente (una VPC privada), lo que evita que otros actores ralenticen el tráfico.



Le recomendamos que implemente también estas funciones en las AWS CloudFormation plantillas que cree.

## Cómo funciona Application Testing

La siguiente figura es una descripción general de cómo funcionan las pruebas de equivalencia funcional en Application Testing.



- - Puede transferir los datos de entrada desde la fuente AWS mediante la transferencia de [archivos de modernización de AWS mainframe](#) o sus herramientas preferidas para la transferencia de datos de mainframe.
- Aplica la misma lógica empresarial tanto en el origen como en el destino.
- Application Testing compara automáticamente los datos de salida (conjuntos de datos, cambios en las bases de datos relacionales, pantallas 3270 e interacciones de los usuarios) tanto del origen como del destino. Tras ejecutar el escenario de prueba en el ordenador central, se capturan los datos de salida y se transfieren al mismo y AWS, a continuación, se reproduce el escenario de prueba en el servidor de destino. Las pruebas de aplicaciones comparan automáticamente los datos de salida de la prueba AWS con los datos de salida de la fuente. Puede ver rápidamente qué registros son idénticos, equivalentes, diferentes o faltantes. Además, puede definir reglas de equivalencia para que los registros que no sean idénticos pero tengan el mismo significado empresarial se entiendan como equivalentes.

El flujo de trabajo que se sigue en Application Testing consta de los siguientes pasos:

1. Crear conjuntos de prueba. Los casos de prueba son la unidad más pequeña de las acciones de prueba. Al crear un caso de prueba, también se identifican los tipos de datos que se van a comparar y que representan mejor la equivalencia funcional entre el origen y el destino.
2. Crear escenarios de prueba. Los escenarios de prueba agrupan los casos de prueba relacionados en una secuencia específica para su ejecución.
3. Crear las condiciones iniciales. Las condiciones iniciales explican cómo grabar una prueba en el mainframe y cómo AWS CloudFormation crear automáticamente el mismo estado en AWS.
4. Registro en el origen y reproducción en el destino. Capture los conjuntos de datos de entrada y salida en el ordenador central y cárguelos allí. AWS A continuación, se vuelve a reproducir el escenario de prueba en AWS.
5. Comparar los conjuntos de datos de origen y destino. Application Testing compara automáticamente los conjuntos de datos de salida, tanto de origen como de destino, para que pueda ver rápidamente qué es correcto y qué no.

Tanto la acción final de un escenario de prueba como el objetivo de todo el proceso es identificar las discrepancias entre las pruebas de origen y de destino. Application Testing compara la versión de origen y la versión de destino de los datos capturados en todos los canales de interacción durante la ejecución de la prueba. También compara los estados finales de los datos relevantes (tal como se definen en los casos de prueba).

## Servicios relacionados

Las pruebas de aplicaciones son una característica de la modernización del AWS mainframe. También utiliza la infraestructura como código AWS CloudFormation para garantizar la repetibilidad, la automatización y la rentabilidad de las pruebas. Para obtener más información, consulte:

- [AWS Modernización del mainframe](#)
- [AWS CloudFormation](#)

## Acceso a Application Testing

Para acceder a las pruebas de aplicaciones desde la consola de modernización del AWS mainframe, seleccione Application Testing en el menú de navegación de la izquierda.

## Precios de Application Testing

Los precios de Application Testing se encuentran en [Precios de AWS Mainframe Modernization](#).

## AWS Conceptos de pruebas de aplicaciones de modernización de mainframe

AWS Las pruebas de aplicaciones se encuentran en una versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.

AWS Las pruebas de aplicaciones utilizan términos que otros servicios de pruebas o paquetes de software podrían utilizar con un significado ligeramente diferente. En las siguientes secciones se explica cómo las pruebas de aplicaciones de modernización de AWS mainframes utilizan esta terminología.

### Temas

- [Caso de prueba](#)
- [Escenario de prueba](#)
- [Proyecto de prueba](#)
- [Condición inicial](#)
- [Registro \(captura\)](#)
- [Reproducción](#)
- [Comparación](#)
- [Comparación de bases de datos](#)
- [Comparación de conjuntos de datos](#)
- [Estado de comparación](#)
- [Reglas de equivalencia](#)
- [Comparación de conjuntos de datos de estado final](#)
- [Comparaciones de bases de datos estado-progreso](#)
- [Equivalencia funcional \(FE\)](#)
- [Comparaciones de pantallas 3270 en línea](#)

- [Registro](#)
- [Datos de reproducción](#)
- [Datos de referencia](#)
- [Registro, reproducción y comparación](#)
- [Diferencias](#)
- [Equivalencias](#)
- [Aplicación de origen](#)
- [Aplicación de destino](#)

## Caso de prueba

Un caso de prueba es la unidad de acción individual más básica del flujo de trabajo de pruebas. Por lo general, un caso de prueba se utiliza para representar una unidad independiente de lógica empresarial que modifica los datos. Se realizarán comparaciones para cada caso de prueba. Los casos de prueba se agregan a un escenario de prueba. Los casos de prueba contienen metadatos sobre los artefactos de datos (conjuntos de datos, bases de datos) que modifica el caso de prueba y sobre las funciones empresariales que se activan durante la ejecución del caso de prueba: trabajos por lotes, cuadros de diálogo interactivos de tipo 3270 y otros. Por ejemplo, los nombres y las páginas de códigos de los conjuntos de datos.

Datos de entrada → Caso de prueba → Datos de salida

Los casos de prueba pueden ser en línea o por lotes:

- Los casos de prueba en línea son casos de prueba en los que el usuario ejecuta cuadros de diálogo interactivos en pantalla (3270) para leer, modificar o generar nuevos datos empresariales (registros de bases de datos o conjuntos de datos).
- Los casos de prueba por lotes son casos de prueba que requieren enviar un lote para leer, procesar y modificar o producir nuevos datos empresariales (conjuntos de datos o registros de bases de datos).

## Escenario de prueba

Los escenarios de prueba son una serie de casos de prueba que se ejecutan en orden secuencial, uno por uno. La reproducción se realiza en el nivel del escenario de prueba. Todos los casos de

prueba del escenario de prueba se ejecutan en el entorno de prueba de destino cuando se reproduce un escenario de prueba. Si hay diferencias después de comparar los artefactos de las pruebas de referencia y de reproducción, las diferencias se mostrarán en el nivel del caso de prueba.

Por ejemplo, el escenario de prueba A:

Caso de prueba 1, caso de prueba 2, caso de prueba 3, etc.

## Proyecto de prueba

Los proyectos de prueba representan un conjunto de escenarios de prueba para alcanzar el hito de prueba deseado. Por ejemplo, la migración de una aplicación específica se puede considerar como un proyecto de prueba único. Al agrupar los escenarios de prueba en proyectos de prueba, los responsables de las pruebas pueden realizar un seguimiento del estado del proyecto de prueba, incluidas las pruebas aprobadas o no aprobadas.

## Condición inicial

Una condición inicial contiene un conjunto de recursos (de computación, almacenes de datos, etc.) que debe crear y datos de aplicación que debe restaurar en esos recursos creados antes de poder ejecutar escenarios de prueba. Esto crea la referencia del entorno de prueba objetivo. Le permite proporcionar una AWS CloudFormation plantilla. La plantilla se utiliza para crear el entorno de pruebas de destino y, de forma opcional, se extrae el DDL de la base de datos de origen si el escenario de prueba modifica los registros de la base de datos. Cada escenario de prueba se asociará a una condición inicial. Una condición inicial puede asociarse con varios escenarios de prueba. Para garantizar la repetibilidad y la coherencia de los resultados y evitar falsos positivos debido a datos ya modificados, debe restablecer las condiciones iniciales antes de ejecutar cada escenario de prueba.

En los escenarios de prueba que contienen casos de prueba que modifican los registros de la base de datos, la condición inicial también hace referencia a una exportación a DDL de los esquemas y tablas de la base de datos de origen.

## Registro (captura)

El registro se realiza en el nivel del escenario de prueba. Durante el registro, debe proporcionar una ubicación de Amazon S3 que contenga los artefactos, los conjuntos de datos y los diarios CDC de las bases de datos relacionales del mainframe de origen con los que se compararán. Se considerarán datos de referencia del mainframe de origen. Durante la reproducción, los datos de

reproducción generados se compararán con los datos de referencia registrados para garantizar la equivalencia de las aplicaciones.

## Reproducción

La reproducción se realiza en el nivel del escenario de prueba. Durante la reproducción, las pruebas de aplicaciones de modernización del AWS mainframe utilizan el CloudFormation script al que se hace referencia en la condición inicial asociada para crear el entorno de prueba de destino y ejecutar la aplicación. Los conjuntos de datos y los registros de las bases de datos que se modifican durante la reproducción se capturan y se comparan con los datos de referencia del mainframe. Por lo general, se graba en el mainframe una vez y luego se reproduce varias veces, hasta que se alcance la equivalencia funcional.

## Comparación

Las comparaciones se realizan automáticamente cuando la reproducción finaliza satisfactoriamente. Durante las comparaciones, los datos de referencia que cargó y capturó durante la fase de registro se comparan con los datos de reproducción generados durante la fase de reproducción. Las comparaciones se realizan en el nivel de caso de prueba individual para conjuntos de datos, registros de bases de datos y pantallas en línea por separado.

## Comparación de bases de datos

Application Testing emplea una función de comparación del progreso del estado al comparar los cambios en los registros de las bases de datos entre las aplicaciones de origen y destino. La comparación estado-progreso compara las diferencias en cada ejecución individual de las instrucciones INSERT, UPDATE y DELETE, a diferencia de comparar las filas de la tabla al final del proceso. La comparación estado-progreso es más eficiente que las alternativas, ya que proporciona comparaciones más rápidas y precisas, pues solo se comparan los datos modificados y se detectan los errores que se corrigen automáticamente en el flujo de transacciones. Al utilizar la tecnología CDC (Changed Data Capture), Application Testing puede detectar cambios en las bases de datos de relaciones individuales y compararlos entre la de origen y la de destino.

Los cambios en la base de datos relacional se generan en el origen y el destino mediante el código de la aplicación probada con sentencias DML (lenguaje de modificación de datos) como SQL INSERT, UPDATE o DELETE, pero también indirectamente cuando la aplicación utiliza procedimientos almacenados, cuando los activadores de la base de datos se configuran en algunas tablas o cuando se utiliza CASCADE DELETE para garantizar la integridad referencial, lo que desencadena automáticamente eliminaciones adicionales.

## Comparación de conjuntos de datos

Application Testing compara automáticamente los conjuntos de datos de referencia y reproducción producidos en los sistemas de origen (registro) y de destino (reproducción).

Para comparar conjuntos de datos:

1. Comience con los mismos datos de entrada (conjuntos de datos, base de datos) tanto en el origen como en el destino.
2. Ejecute sus casos de prueba en el sistema de origen (mainframe).
3. Capture los conjuntos de datos producidos y cargue los datos en un bucket de Amazon S3. Puede transferir los conjuntos de datos de entrada de la fuente a otros AWS mediante revistas, pantallas y conjuntos de datos de los CDC.
4. Especifique la ubicación del bucket de Amazon S3 en el que se cargaron los conjuntos de datos del mainframe al registrar el caso de prueba.

Una vez finalizada la reproducción, Application Testing compara automáticamente los conjuntos de datos de referencia y de destino de salida y muestra si los registros son idénticos, equivalentes, diferentes o faltan. Por ejemplo, los campos de fecha relativos al momento de la ejecución de la carga de trabajo (día + 1, final del mes en curso, etc.) se consideran automáticamente equivalentes. Además, si lo desea, puede definir reglas de equivalencia para que los registros que no sean idénticos sigan teniendo el mismo significado empresarial y se marquen como equivalentes.

## Estado de comparación

Application Testing utiliza los siguientes estados de comparación: IDENTICAL, EQUIVALENT y DIFFERENT.

### IDENTICAL

Los datos de origen y destino son exactamente los mismos.

### EQUIVALENT

Los datos de origen y destino contienen diferencias falsas consideradas equivalencias, como fechas o marcas horarias, que no afectan a la equivalencia funcional cuando son relativas al momento de la ejecución de la carga de trabajo. Puede definir reglas de equivalencia para identificar cuáles son estas diferencias. Cuando todos los escenarios de prueba repetidos en

comparación con sus escenarios de prueba de referencia muestran el estado de IDENTICAL o EQUIVALENT, el escenario de prueba demuestra la equivalencia funcional.

## DIFFERENT

Los datos de origen y de destino contienen diferencias, como un número diferente de registros en un conjunto de datos o valores diferentes en el mismo registro.

## Reglas de equivalencia

Conjunto de reglas para identificar diferencias falsas que pueden considerarse resultados equivalentes. Las pruebas de equivalencia funcional offline (OFET) provocan inevitablemente diferencias en algunos resultados entre los sistemas de origen y de destino. Por ejemplo, las marcas de tiempo de actualización son diferentes según el diseño. Las reglas de equivalencia explican cómo ajustar esas diferencias y evitar los falsos positivos en el momento de la comparación. Por ejemplo, si una fecha es el tiempo de ejecución más 2 días en una columna de datos concreta, la regla de equivalencia lo describe y acepta un tiempo en el sistema de destino que tenga un tiempo de ejecución en el objetivo más 2 días en lugar de un valor que sea estrictamente igual a la misma columna del registro de referencia.

## Comparación de conjuntos de datos de estado final

El estado final de los conjuntos de datos que se han creado o modificado, incluidos todos los cambios o actualizaciones realizados en los conjuntos de datos desde su estado inicial. En el caso de los conjuntos de datos, Application Testing examina los registros de esos conjuntos de datos al final de la ejecución de un caso de prueba y compara los resultados.

## Comparaciones de bases de datos estado-progreso

Comparaciones de los cambios realizados en los registros de la base de datos como una secuencia de instrucciones DML (eliminar, actualizar, insertar) individuales. Application Testing compara los cambios individuales (insertar, actualizar o eliminar la fila de una tabla) de la base de datos de origen con la de destino e identifica las diferencias para cada cambio individual. Por ejemplo, se puede usar una instrucción INSERT individual para insertar en una tabla una fila con valores diferentes en la base de datos de origen en comparación con la base de datos de destino.



## Equivalencia funcional (FE)

Dos sistemas se consideran funcionalmente equivalentes si producen los mismos resultados en todas las operaciones observables, con los mismos datos de entrada. Por ejemplo, dos aplicaciones se consideran funcionalmente equivalentes si los mismos datos de entrada producen datos de salida idénticos (a través de pantallas, cambios en el conjunto de datos o cambios en la base de datos).

## Comparaciones de pantallas 3270 en línea

Compara la salida de las 3270 pantallas del ordenador central con la salida de las pantallas web de aplicaciones modernizadas cuando el sistema de destino se ejecuta en un entorno de ejecución de AWS Blu Age. Nube de AWS Además, compara la salida de las pantallas 3270 del mainframe con las pantallas 3270 de la aplicación que se ha vuelto a alojar cuando el sistema de destino se ejecuta en el tiempo de ejecución de Micro Focus en la Nube de AWS.

## Registro

Acción que consiste en restaurar un estado de datos conocido y, a continuación, capturar o registrar los datos de referencia de un escenario de prueba de referencia (para uno o varios casos de prueba de forma secuencial) en un sistema de origen.

## Datos de reproducción

Los datos de reproducción se utilizan para describir los datos generados al reproducir un escenario de prueba en el entorno de prueba de destino. Por ejemplo, los datos de reproducción se generan cuando se ejecuta un escenario de prueba en una aplicación del servicio de modernización de AWS mainframe. A continuación, los datos de reproducción se comparan con los datos de referencia capturados en el origen. Cada vez que reproduce la carga de trabajo en el entorno de destino, se genera una nueva generación de datos de reproducción.

## Datos de referencia

Los datos de referencia se utilizan para describir los datos capturados en el mainframe de origen. Es la referencia con la que se compararán los datos generados por la reproducción (objetivo). Por lo general, por cada registro del mainframe que cree datos de referencia, habrá muchas repeticiones. Esto se debe a que los usuarios suelen capturar el estado correcto de la aplicación en el mainframe y reproducir los casos de prueba en la aplicación modernizada de destino para validar la equivalencia. Si se encuentran errores, se corrigen, y los casos de prueba se vuelven a reproducir. A menudo, se

repite varios ciclos, se corrigen errores y se vuelve a reproducir para validar la incidencia. Esto se conoce como el paradigma de prueba de capturar una vez y reproducir varias veces.

## Registro, reproducción y comparación

Application Testing opera en tres pasos:

- **Registro:** captura los datos de referencia creados en el mainframe para cada caso de prueba de un escenario de prueba. Pueden incluir pantallas 3270 en línea, conjuntos de datos y registros de bases de datos.
  - En el caso de pantallas 3270 en línea, debe utilizar el emulador de terminal Blu Insights para capturar la carga de trabajo de origen. Para obtener más información, consulte la [documentación de Blu Insights](#).
  - En el caso de los conjuntos de datos, tendrá que capturar los conjuntos de datos generados por cada caso de prueba en el mainframe mediante herramientas habituales, como el FTP o el servicio de transferencia de conjuntos de datos que forma parte de AWS Mainframe Modernization.
  - Para los cambios en la base de datos, utilice la documentación de [AWS Mainframe Modernization Data Replication with Precisely](#) para recopilar y generar diarios de CDC que contengan los cambios.
- **Reproducción:** reproduce el escenario de prueba en el entorno de destino. Todos los casos de prueba especificados en el escenario de prueba. Los tipos de datos específicos creados por los casos de prueba individuales, como conjuntos de datos, cambios en las bases de datos relacionales o pantallas 3270, se capturarán de forma automática. Estos datos se conocen como datos de reproducción y se compararán con los datos de referencia capturados durante la fase de registro.

### Note

Los cambios en la base de datos relacional requerirán opciones de configuración específicas del DMS en la plantilla de estado inicial. CloudFormation

- **Comparación:** se comparan los datos de referencia de la prueba de origen y los datos de reproducción de destino, y los resultados se muestran como datos idénticos, diferentes, equivalentes o faltantes.

## Diferencias

Indica que se han detectado diferencias entre los conjuntos de datos de referencia y de reproducción mediante la comparación de datos. Por ejemplo, un campo de una pantalla 3270 en línea que muestre valores diferentes desde el punto de vista de la lógica empresarial entre el mainframe de origen y la aplicación modernizada de destino se considerará una diferencia. Otro ejemplo es un registro en un conjunto de datos que no es idéntico entre las aplicaciones de origen y destino.

## Equivalencias

Los registros equivalentes son registros que difieren entre los conjuntos de datos de referencia y los de reproducción, pero que no deben tratarse de forma diferente desde el punto de vista de la lógica empresarial. Por ejemplo, un registro que contenga la marca temporal del momento en que se produjo el conjunto de datos (tiempo de ejecución de la carga de trabajo). Mediante reglas de equivalencia personalizables, puede indicar a Application Testing que trate esa diferencia de falsos positivos como una equivalencia, incluso aunque muestre valores diferentes entre los datos de referencia y los de reproducción.

## Aplicación de origen

La aplicación de mainframe de origen con la que se va a comparar.

## Aplicación de destino

La aplicación nueva o modificada en la que se realizan las pruebas y que se comparará con la aplicación de origen para detectar cualquier defecto y lograr la equivalencia funcional entre las aplicaciones de origen y de destino. La aplicación de destino normalmente se ejecuta en la nube. AWS

## Tutorial: configurar la aplicación CardDemo de muestra

AWS Las pruebas de aplicaciones se encuentran en una versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.

Para este tutorial, debe crear una AWS CloudFormation pila que le ayude a configurar la [aplicación de CardDemo ejemplo](#) para cambiarla de plataforma con el servicio gestionado Micro Focus on

AWS Mainframe Modernization y con funciones como las pruebas de aplicaciones de modernización de AWS mainframes. El tutorial describe un ejemplo de AWS CloudFormation plantilla que puede utilizar para crear la pila. También proporcionamos un archivo comprimido con los artefactos de aplicación necesarios. La plantilla de ejemplo proporciona una base de datos, un entorno de tiempo de ejecución, una aplicación y un entorno de red completamente aislado.

Esta plantilla crea varios AWS recursos. Estos recursos se le facturarán si crea una pila a partir de esta plantilla.

## Requisitos previos

- Descargue y descomprima [IC3-card-demo-zip](#) y [datasets\\_Mainframe\\_ebcdic.zip](#). Estos archivos contienen la CardDemo muestra y los conjuntos de datos de muestra para su uso en las pruebas de AWS aplicaciones.
- Cree un bucket de Amazon S3 para almacenar los CardDemo archivos y otros artefactos. Por ejemplo, `my-carddemo-bucket`.

## Paso 1: Prepárese para la configuración CardDemo

Cargue los archivos de CardDemo muestra y edite la AWS CloudFormation plantilla que creará la CardDemo aplicación.

1. Cargue las carpetas `datasets_Mainframe_ebcdic` y `IC3-card-demo` que descomprimió anteriormente en su bucket.
2. Descarga la `aws-m2-math-mf-carddemo.yaml` AWS CloudFormation plantilla de tu paquete. Está en la carpeta `IC3-card-demo`.
3. Edita la `aws-m2-math-mf-carddemo.yaml` AWS CloudFormation plantilla de la siguiente manera:
  - Cambie el parámetro `BucketName` por el nombre del bucket que definió anteriormente, por ejemplo `my-carddemo-bucket`.
  - Cambie la `ImportJsonPath` a la ubicación del archivo `mf-carddemo-datasets-import.json` en su bucket. Por ejemplo, `s3://my-carddemo-bucket/IC3-card-demo/mf-carddemo-datasets-import.json`. Al actualizar este valor se asegura que la salida `M2ImportJson` tenga el valor correcto.
  - (Opcional) Adapte los parámetros `EngineVersion` y `InstanceType` para que coincidan con sus estándares.

**Note**

No modifique los resultados `M2EnvironmentId` y `M2ApplicationId`. `Application Testing` utiliza esos valores para localizar los recursos con los que interactuará.

## Paso 2: Crear todos los recursos necesarios

Ejecute su AWS CloudFormation plantilla personalizada para crear todos los recursos que necesita para completar este tutorial correctamente. Esta plantilla configura la `CardDemo` aplicación para que pueda utilizarla en las pruebas.

1. Inicie sesión en la AWS CloudFormation consola y elija `Crear pila` y, a continuación, elija `Con nuevos recursos (estándar)`.
2. En `Requisito previo`: preparar la plantilla, elija `La plantilla está lista`.
3. En `Especificar plantilla`, seleccione `Cargar un archivo de plantilla` y, a continuación, elija `Elegir archivo`.
4. Navegue hasta el lugar donde descargó `aws-m2-math-mf-carddemo.yaml`, elija ese archivo y, a continuación, elija `Siguiente`.
5. En `Especificar los detalles de la pila`, proporcione un nombre para la pila para que pueda encontrarla fácilmente en una lista y, a continuación, seleccione `Siguiente`.
6. En `Configurar opciones de pila`, mantenga los valores predeterminados y seleccione `Siguiente`.
7. En `Revisar`, compruebe qué AWS CloudFormation es lo que te está creando y, a continuación, selecciona `Enviar`.

Se tarda entre 10 y 15 minutos en AWS CloudFormation crear la pila.

**Note**

La plantilla está configurada para agregar un sufijo único a los nombres de los recursos que crea. Esto significa que puede crear varias instancias de esta plantilla de pila en paralelo, una característica clave para `Application Testing` que le permite ejecutar varios escenarios de prueba al mismo tiempo.

## Paso 3: Implementar e iniciar la aplicación

Implemente la CardDemo aplicación que AWS CloudFormation creó para usted y asegúrese de que se esté ejecutando.

1. Abra la consola de modernización del AWS mainframe y seleccione Aplicaciones en el menú de navegación de la izquierda.
2. Elija la CardDemo aplicación, que tendrá un nombre parecido `aws-m2-math-mf-carddemo-abc1d2e3` a.
3. Elija Acciones y, a continuación, Implementar aplicación.
4. En Entornos, seleccione el entorno de tiempo de ejecución que corresponda a la aplicación. Tendrá el mismo identificador único adjunto al final del nombre. Por ejemplo, `aws-m2-math-mf-carddemo-abc1d2e3`.
5. Elija Implementar. Espere hasta que la aplicación se implemente correctamente y tenga el estado Listo.
6. Seleccione la aplicación, elija Acciones y, a continuación, Iniciar aplicación. Espere hasta que la aplicación esté en estado En ejecución.
7. En la página de detalles de la aplicación, copie el Puerto y el Nombre de host DNS que necesita para conectarse a la aplicación en ejecución.

## Paso 4: Importar los datos iniciales

Para utilizar la aplicación de CardDemo ejemplo, debe importar un conjunto inicial de datos. Siga estos pasos:

1. Descargue el archivo `mf-carddemo-datasets-import.json`.
2. Abra el archivo con el editor de texto que prefiera.
3. Localice el parámetro `s3Location` y actualice el valor para que apunte al bucket de Amazon S3 que creó.
4. Realice este mismo cambio todas las veces que aparezca `s3Location` y, a continuación, guarde el archivo.
5. Inicie sesión en la consola de Amazon S3 y vaya al bucket que creó anteriormente.
6. Cargue el archivo `mf-carddemo-datasets-import.json` personalizado.
7. Abra la consola de modernización del AWS mainframe y seleccione Aplicaciones en el menú de navegación de la izquierda.

8. Elija la CardDemo aplicación.
9. Elija Conjuntos de datos y, a continuación, Importar.
10. Navegue hasta la ubicación en Amazon S3 en la que cargó el archivo JSON personalizado y seleccione Enviar.

Este trabajo importa 23 conjuntos de datos. Para supervisar el resultado del trabajo de importación, compruebe la consola. Cuando todos los conjuntos de datos se hayan importado correctamente, conéctese a la aplicación.

#### Note

Al utilizar esta plantilla en Application Testing, la salida M2ImportJson gestiona automáticamente el proceso de importación.

## Paso 5: Conectarse a la CardDemo aplicación

Conéctese a la aplicación de CardDemo muestra con el emulador 3270 de su elección.

- Cuando la aplicación esté en ejecución, utilice el emulador 3270 para conectarse a la aplicación y especifique el nombre del servidor DNS y el nombre del puerto, si es necesario.

Por ejemplo, si utiliza el [emulador c3270](#) de código abierto, el comando tendrá el siguiente aspecto:

```
c3270 -port port-number DNS-hostname
```

port

El puerto especificado en la página de detalles de la aplicación. Por ejemplo, 6000.

Hostname

El nombre de host de DNS especificado en la página de detalles de la aplicación.

En la siguiente figura se muestra dónde encontrar el puerto y el nombre de host de DSN.

AWS Mainframe Modernization > Applications > aws-m2-math-mf-carddemo-7f28a650

**aws-m2-math-mf-carddemo-7f28a650** [Info](#) Actions ▾

[Definition](#) | [Batch jobs](#) | [Data sets](#) | [Tags](#)

**Application information** [Info](#)

Name aws-m2-math-mf-carddemo-7f28a650	Status <span style="color: green;">✔ Running</span>	Ports 7000	Logs <a href="#">ConsoleLog</a> <a href="#">BatchJobLogs</a>
ARN arn:aws:m2:us-west-2:██████████:app/efzlb7ocfb5zi7fwfcxfusw4	Creation time May 2, 2023 at 10:50 (UTC-04:00)	KMS key AWS owned key	Description m2 application: aws-m2-math-mf-carddemo-7f28a650
Engine Micro Focus	DNS Hostname <a href="#">haytgmjvgazteoi-ibgcq4di.m2.us-west-2.amazonaws.com</a>		

## Tutorial: Prueba de aplicaciones de modernización de AWS mainframe, reproducción y comparación con el uso CardDemo de AWS Blu Age implementado en Amazon EC2

AWS Las pruebas de aplicaciones se encuentran en una versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.

En este tutorial, completará los pasos necesarios para reproducir y comparar las cargas de trabajo de prueba con la CardDemo aplicación que se ejecuta en AWS Blu Age implementada en Amazon EC2.

### Paso 1: Obtenga la imagen de máquina Amazon EC2 de AWS Blu Age (AMI)

Siga las instrucciones del tutorial de [configuración de AWSAWS Blu Age Runtime \(en Amazon EC2\)](#) para conocer los pasos de incorporación necesarios para acceder a la AMI de AWS Blu Age en Amazon EC2.



## Paso 2: Iniciar una instancia de Amazon EC2 con la AMI de AWS Blu Age

1. Configure sus AWS credenciales.
2. Identifique la ubicación del archivo binario de la AMI de Amazon EC2 3.5.0 (solo AWS CLI o versión Blu Age) del bucket de Amazon S3:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/  
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/3.5.0/AMI/
```

### Note

La característica de Application Testing solo está disponible para su uso en 4 regiones de producción (us-east-1, sa-east-1, eu-central-1 y ap-southeast-2).

3. Restaure la AMI en su cuenta con el siguiente comando:

```
aws ec2 create-restore-image-task --object-key 3.5.0/AMI/ami-0182ffe3b9d63925b.bin  
--bucket aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1 --region eu-west-1 --name  
"AWS BLUAGE RUNTIME AMI"
```

### Note

Cambie el nombre del archivo bin de la AMI y la región en la que quiera crear la AMI.

4. Tras crear una instancia de Amazon EC2, puede encontrar el ID de AMI correcto que se restauró desde el bucket de Amazon S3 en el catálogo de imágenes de Amazon EC2.

### Note

En este tutorial, el ID de AMI es ami-0d0fafcc636fd1e6d y debe cambiarlo en los distintos archivos de configuración por el que se le proporcionó.

1. Si se produce un create-restore-image-task error en aws ec2, compruebe su versión de Python y CLI mediante el siguiente comando:

```
aws --version
```

**Note**

La versión de Python debe ser  $\geq 3$  y la versión de la CLI debe ser  $\geq 2$ .

2. Si estas versiones están obsoletas, se debe actualizar la CLI. Para actualizar la CLI:
  - a. Siga las instrucciones en [Instalación o actualización de la versión más reciente de la CLI de AWS](#).
  - b. Elimine la CLI v1 con el siguiente comando:

```
sudo yum remove awscli
```

- c. E instale la CLI v2 con el siguiente comando:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

- d. Por último, compruebe la versión de Python y la CLI con el siguiente comando:

```
aws --version
```

3. A continuación, puede rehacer el `aws create-restore-image-task ec2`.

## Paso 3: Cargue CardDemo los archivos dependientes a S3

Copie el contenido de las carpetas `databases`, `file-system` y `userdata`. Descarga y descomprime las CardDemo aplicaciones. Estas tres carpetas deben copiarse en uno de sus buckets denominado `your-s3-bucket` en esta documentación.

## Paso 4: Cargue las bases de datos e inicialice la aplicación CardDemo

Cree una instancia Amazon EC2 temporal que utilizará como recurso informático para generar las instantáneas de base de datos necesarias para la aplicación. CardDemo Esta instancia EC2 no ejecutará la CardDemo aplicación por sí misma, sino que generará las instantáneas de la base de datos que se utilizarán más adelante.

Comience por editar la CloudFormation plantilla proporcionada denominada 'load-and-create-ba-snapshots.yml.' Esta es la CloudFormation plantilla que se utiliza para crear la instancia de Amazon EC2 que se utiliza para generar las instantáneas de la base de datos.

1. Genere y proporcione el par de claves de EC2 que se utilizará para la instancia de EC2. Para obtener más información, consulte [Crear pares de claves](#).

Ejemplo:

```
Ec2KeyPair:
  Description: 'ec2 key pair'
  Default: 'm2-tests-us-west-2'
  Type: String
```

2. Especifique la ruta de Amazon S3 de la carpeta en la que haya colocado la carpeta database del paso anterior:

```
S3DBScriptsPath:
  Description: 'S3 DB scripts folder path'
  Type: String
  Default: 's3://your-s3-bucket/databases'
```

3. Especifique la ruta de Amazon S3 de la carpeta en la que haya colocado la carpeta file-system del paso anterior:

```
S3ApplicationFilePath:
  Description: 'S3 application files folder path'
  Type: String
  Default: 's3://your-s3-bucket/file-system'
```

4. Especifique la ruta de Amazon S3 de la carpeta en la que haya colocado la carpeta userdata del paso anterior:

```
S3UserDataPath:
  Description: 'S3 userdata folder path'
  Type: String
  Default: 's3://your-s3-bucket/userdata'
```

5. Especifique también una ruta de Amazon S3 en la que guardará los archivos de resultados para utilizarlos en el siguiente paso.

**S3SaveProducedFilesPath:**

Description: 'S3 path folder to save produced files'

Type: String

Default: 's3://your-s3-bucket/post-produced-files'

6. Cambie el ID de la AMI por el correcto obtenido anteriormente en este tutorial mediante la siguiente plantilla:

**BaaAmiId:**

Description: 'ami id (AL2) for ba anywhere'

Default: 'ami-0bd41245734fd20d9'

Type: String

- Si lo desea, puede cambiar el nombre de las tres instantáneas que se crearán al ejecutar las bases de datos de carga. `withCloudFormation` Estarán visibles en la CloudFormation pila a medida que se vaya creando y se utilizarán más adelante en este tutorial. Recuerde anotar los nombres utilizados para las instantáneas de base de datos.

**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**

En este documento, damos por sentado que el nombre de las instantáneas sigue siendo coherente.

7. Ejecute CloudFormation con CLI o AWS consola mediante el botón y el asistente Create Stack. Al final del proceso, debería ver tres instantáneas en la consola de RDS con el nombre que eligió, seguido de un ID único. Necesitará estos nombres en el siguiente paso.

 Note

RDS agregará sufijos a los nombres de las instantáneas definidos en la plantilla. AWS CloudFormation asegúrese de obtener el nombre completo de la instantánea desde RDS antes de ir al paso siguiente.

### Ejemplo de comando de la CLI

```
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
```

También puede comprobar en la ruta de Amazon S3 que proporcionó para S3 SaveProducedFilePath que los conjuntos de datos se han creado correctamente.

## Paso 5: Inicie el tiempo de ejecución de AWS Blu Age CloudFormation

Se utiliza CloudFormation para ejecutar la instancia de Amazon EC2 con la aplicación CardDemo AWS Blu Age. Debe reemplazar algunas variables del CloudFormation nombre `m2-with-ba-using-snapshots-https-authentication.yml` editando el archivo YAML o modificando los valores de la consola durante el lanzamiento del CFN.

1. Modifique `AllowedVpcEndpointPrincipals` para especificar qué cuenta llegará al punto final de la VPC para acceder al tiempo de ejecución de AWS Blu Age mediante los siguientes comandos:


```
AllowedVpcEndpointPrincipals:
  Description: 'comma-separated list of IAM users, IAM roles, or AWS accounts'
  Default: 'apptest.amazonaws.com'
  Type: String
```

2. Cambie el valor de SnapshotPrimaryDb las SnapshotBlusamDb variables y SnapshotJicsDb el nombre de las instantáneas. Obtenga también los nombres de las instantáneas de RDS después de crearlas en el paso anterior.

```
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 agregará su propio sufijo a los nombres de las instantáneas.

3. Proporcione el par de claves de Amazon EC2 para la instancia de EC2 mediante este comando:

```
Ec2KeyPair:
  Description: 'ec2 key pair'
  Default: 'm2-tests-us-west-2'
  Type: String
```

4. Proporcione el ID de AMI que obtuvo durante el proceso de registro de la AMI para la variable BaaAmildmediante:

```
BaaAmiId:
  Description: 'ami id (AL2) for ba anywhere'
  Default: 'ami-0d0fafcc636fd1e6d'
  Type: String
```


5. Proporcione la ruta de la carpeta Amazon S3 que utilizó en el paso anterior para guardar los archivos generados mediante el siguiente comando:

```
S3ApplicationFilePath:  
  Description: 'bucket name'  
  Type: String  
  Default: 's3://your-s3-bucket/post-produced-files'
```

6. Por último, indique la ruta de la carpeta s3- userdata-folder-path:

```
S3UserDataPath:  
  Description: 'S3 userdata folder path'  
  Type: String  
  Default: 's3://your-s3-bucket/userdata'
```

- (Opcional) Puede activar el modo HTTPS y la autenticación HTTP básica para tomcat. Aunque la configuración predeterminada también funcionaría.

 Note

De forma predeterminada, el modo HTTPS está deshabilitado y configurado en modo HTTP en el parámetro BacHttpsMode:

Por ejemplo:

```
BacHttpsMode:  
  Description: 'http or https for Blue Age Runtime connection mode '  
  Default: 'http'  
  Type: String  
  AllowedValues: [http, https]
```

- (Opcional) Para habilitar el modo HTTPS, debe cambiar el valor a HTTPS y proporcionar el ARN de su certificado ACM cambiando el valor de la variable ACM: CertArn

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

- (Opcional) La autenticación básica está deshabilitada de forma predeterminada con el parámetro `WithBacBasicAuthentication` establecido en `false`. Puede habilitarla, estableciendo el valor como `true`.

```
WithBacBasicAuthentication:
  Description: 'false or true for Blue Age Runtime Basic Authentication '
  Default: false
  Type: String
  AllowedValues: [true, false]
```

7. Cuando haya completado la configuración, podrá crear la pila mediante la CloudFormation plantilla editada.

## Paso 6: Probar la instancia Amazon EC2 de AWS Blu Age

Ejecute manualmente la CloudFormation plantilla para crear la instancia Amazon EC2 de AWS Blu Age para la CardDemo aplicación y asegurarse de que se inicia sin errores. Esto se hace para comprobar que la CloudFormation plantilla y todos los requisitos previos son válidos antes de utilizar la CloudFormation plantilla con la función de prueba de aplicaciones. A continuación, puede utilizar las pruebas de aplicaciones para crear automáticamente la instancia Amazon EC2 de AWS Blu Age de destino durante la reproducción y compararla con una condición inicial.

1. Ejecuta el comando CloudFormation `create stack` para crear la instancia Amazon EC2 de AWS Blu Age y proporciona la plantilla `m2- with-ba-using-snapshots CloudFormation -https-authentication.yml` que editaste en el paso anterior:

```
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

Recuerde especificar la región correcta en la que se restauró la AMI de AWS Blu Age.

2. Asegúrese de que todo funciona correctamente buscando en la consola la instancia de Amazon EC2 en ejecución. Conexión mediante el Administrador de sesiones
3. Una vez que se haya conectado a la instancia de Amazon EC2, utilice los siguientes comandos:



```
sudo su
cd /m2-anywhere/tomcat.gapwalk/velocity/logs
cat catalina.log
```

4. Asegúrese de que no haya excepciones ni errores en el registro.
5. A continuación, compruebe que la aplicación responde mediante este comando:

```
curl http://localhost:8080/gapwalk-application/
```

Ahí podrá ver el mensaje: “Jics application is running”.

## Paso 7: Validar que los pasos anteriores se hayan completado correctamente

En los siguientes pasos, utilizaremos las pruebas de la aplicación de modernización AWS del mainframe para reproducir y comparar los conjuntos de datos creados por la aplicación. CardDemo Estos pasos se basan en la finalización satisfactoria de todos los pasos anteriores de este tutorial. Valide lo siguiente antes de continuar:

1. Ha creado correctamente la instancia de AWS Blu Age en Amazon EC2 a través de la AWS CloudFormation plantilla.
2. El servicio Tomcat de la AWS Blu Age de Amazon EC2 ya está en funcionamiento, sin excepciones.

Cuando ejecute la instancia EC2 con la CardDemo aplicación, complete los siguientes pasos en la consola de pruebas de aplicaciones para reproducir y comparar los conjuntos de datos por lotes.

## Paso 8. Crear la condición inicial

En este paso, crea una condición inicial proporcionando la CloudFormation plantilla que utilizó para implementar la CardDemo aplicación AWS Blu Age en Amazon EC2.

1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el panel de navegación de la izquierda, elija Application Testing.
3. En Application Testing, elija Crear condición inicial.

4. Utilice su copia local con la ruta de Amazon S3 modificada y los ID de instantáneas que apunten a sus recursos.

## Paso 9: Crear el caso de prueba

En este paso, se crea el caso de prueba que se utilizará para comparar los conjuntos de datos creados en la aplicación de CardDemo.

1. Cree un nuevo caso de prueba. Póngale un nombre y una descripción.
2. Especifique CRESTMT . JCL como nombre de JCL.
3. Agregue los siguientes conjuntos de datos a la definición del caso de prueba:

Name	CCSID	RecordFormat	RecordLength
AWS.M2.CA RDDEMO.ST ATEMNT.PS	"037"	FB	80
AWS.M2.CA RDDEMO.ST ATEMNT.HTML	"037"	FB	100

### Note

El nombre de JCL y los detalles del conjunto de datos deben coincidir.

## Paso 10: Crear un escenario de prueba

1. Cree un nuevo escenario de prueba y póngale un nombre y una descripción.
2. Agregue el caso de prueba que creó en el paso anterior a su escenario de prueba.
3. Una vez creado el escenario de prueba, seleccione la condición inicial creada en el paso 1 en la página de descripción general del escenario de prueba.

## Paso 11: Registrar su escenario de prueba

En este paso, ejecutará casos de prueba en el origen. Para ello:

1. Descargue y ejecute los conjuntos de datos que se originaron en la ejecución de la aplicación en el ordenador central. CardDemo
2. Suba la carpeta descomprimida a su bucket de Amazon S3. Este bucket de Amazon S3 debe estar en la misma región de que los demás recursos de Application Testing.

### Note

Debe haber dos archivos con nombres coincidentes con los nombres de los conjuntos de datos aprobados en el caso de prueba anterior.

3. En la página Descripción general del escenario de prueba, pulse el botón Registrar.
4. En la página Registro del escenario de prueba, especifique la ubicación de Amazon S3 en la que cargó los conjuntos de datos obtenidos del mainframe de origen.
5. Haga clic en Enviar para iniciar el proceso de registro.

### Note

Espere a que se complete el registro antes de reproducir y comparar.

## Paso 12: Reproducir y comparar

Ejecute el escenario de prueba y los casos de prueba en el entorno de destino de AWS AWS Blu Age en Amazon EC2. Application Testing capturará los conjuntos de datos producidos por la reproducción y los comparará con los conjuntos de datos de referencia que se registraron en el mainframe.

1. Elija Reproducir y comparar. Crear la CloudFormation pila, realizar la comparación y eliminar la pila debería tardar unos tres minutos.

Una vez que todo esté completo, debería disponer de los resultados de la comparación con algunas diferencias creadas intencionadamente para esta demostración.

## AWS Modernización de mainframe, pruebas de aplicaciones, conjuntos de datos compatibles y páginas de códigos

AWS Las pruebas de aplicaciones se encuentran en una versión preliminar de AWS Mainframe Modernization y están sujetas a cambios. Se recomienda utilizar esta característica solo con datos y aplicaciones de prueba y no en entornos de producción.


Utilice la siguiente tabla para determinar si el identificador de conjunto de caracteres codificado (CCSID) de sus datos es compatible con las pruebas de AWS aplicaciones. Si sus datos utilizan un CCSID no compatible, le recomendamos que lo convierta en un CCSID compatible o que se ponga en [contacto](#) con nosotros para obtener ayuda.

CCSID	Juegos de caracteres	Descripción
37	IBM037, IBM-037, Cp037	Host: EE. UU., Canadá (ESA), Países Bajos, Portugal, Brasil, Australia, Nueva Zelanda
273	IBM273, IBM-273, Cp273	Host: Austria, Alemania
277	IBM277, IBM-277, Cp277	Host: Dinamarca, Noruega
278	IBM278, IBM-278, Cp278	Host: Finlandia, Suecia
280	IBM280, IBM-280, Cp280	Host: Italia
284	IBM284, IBM-284, Cp284	Host: España, Latinoamérica (español)
285	IBM285, IBM-285, Cp285	Host: Reino Unido
297	IBM297, IBM-297, Cp297	Host: Francia
300	IBM-300	JAPAN DB EBCDIC
301	IBM-301	Datos de PC: DB de Japón

CCSID	Juegos de caracteres	Descripción
437	IBM437, IBM-437, US-ASCII, ASCII, Cp437, US-ASCII	Datos de PC: base de PC EE. UU., muchos otros países
500	IBM500, IBM-500, Cp500	Host: Bélgica, Canadá (AS/400), Suiza, Latin-1 internacional
720	IBM-720	MSDOS ÁRABE
737	IBM-737, x-IBM737	MSDOS GRIEGO
775	IBM775, IBM-775	MSDOS BÁLTICO
808	IBM-808	Datos de PC: cirílico, ruso, con euro
813	ISO-8859-7, ISO8859_7	ISO 8859-7: Grecia
819	ISO-8859-1, ISO8859_1	ISO 8859-1: países Latin-1
833	IBM-833	EBCDIC COREANO
834	IBM-834, x-IBM834	EBCDIC DE DB COREANO
835	IBM-835	EBCD DE DB CHINO-T
836	IBM-836	EBCDIC CHINO-S
837	IBM-837	EBCDIC CHINO-S
850	IBM850, IBM-850, Cp850	Datos de PC: países Latin-1
855	IBM855, IBM-855, Cp855	Datos de PC: cirílico
856	IBM-856, x-IBM856, Cp856	Datos de PC: hebreo
858	IBM00858, IBM-858, Cp858	Datos de PC: países Latin-1, con euro

CCSID	Juegos de caracteres	Descripción
859	IBM-859	Datos de PC: LATIN-9
860	IBM860, IBM-860	Datos de PC: portugués
861	IBM861, IBM-861	Datos de PC: Islandia
862	IBM862, IBM-862, Cp862	Datos de PC: hebreo (migración)
863	IBM863, IBM-863	Datos de PC: Canadá
865	IBM865, IBM-865, Cp865	Datos de PC: Din./Noruega
866	IBM866, IBM-866, Cp866	Datos de PC: cirílico, Rusia
867	IBM-867	Datos de PC: hebreo con euro
870	IBM870, IBM-870, Cp870	Host: Latin-2 multilingüe
871	IBM871, IBM-871, Cp871	Host: Islandia
874	x-IBM874	Datos de PC: tailandés
875	IBM-875, x-IBM875, Cp875	Host: Grecia
897	IBM-897	Datos de PC: SB de Japón
912	ISO-8859-2, ISO8859_2	ISO 8859-2: multilingüe Latin-2
915	ISO-8859-5, ISO8859_5	ISO 8859-5: cirílico
916	ISO-8859-8, ISO8859_8	ISO 8859-8: hebreos
918	IBM918, IBM-918, Cp918	Host: urdu
920	ISO-8859-9, ISO8859_9	ISO 8859-9: Latin-5 (ECMA-128, Turquía TS-5881)
921	IBM-921, x-IBM921, Cp921	Datos de PC: Letonia, Lituania

CCSID	Juegos de caracteres	Descripción
922	IBM-922, x-IBM922, Cp922	Datos de PC: Estonia
923	ISO-8859-15, Cp923, ISO8859_15_FDIS	ISO 8859-15: Latin-9
924	IBM-924	ISO 8859-15: Latin-9
927	IBM-927	Datos de PC: chino-T
930	IBM-930, x-IBM930, Cp930	Host katakana: SBCS extendido. Host kanji: DBCS que incluye 4370 caracteres definidos por el usuario
932	IBM-932	Datos de PC: mix de Japón
933	IBM-933, x-IBM933, Cp933	Host: SBCS extendido. Host: DBCS, que incluye 1880 caracteres definidos por el usuario y 11 172 caracteres en hangul completo
935	IBM-935, x-IBM935, Cp935	Host: SBCS extendido. Host: DBCS que incluye 1880 caracteres definidos por el usuario.
937	IBM-937, x-IBM937, Cp937	Host: SBCS extendido. Host: DBCS que incluye 6204 caracteres definidos por el usuario
939	IBM-939, x-IBM939, Cp939	Host latino: SBCS extendido. Host kanji: DBCS que incluye 4370 caracteres definidos por el usuario.

CCSID	Juegos de caracteres	Descripción
942	IBM-942, IBM-942C, x-IBM942, x-IBM942C, Cp942, Cp942C	Datos de PC: SBCS extendido . Datos de PC: DBCS que incluye 1880 caracteres definidos por el usuario.
943	IBM-943, IBM-943C, Shift_JIS, windows-31j, windows-932, x-IBM943, x-IBM943C, Cp943, Cp943C, MS932	Datos de PC: SBCS. Datos de PC: DBCS para entornos abiertos que incluye 1880 caracteres IBM  definidos por el usuario.
947	IBM-947	CHINO-T BIG-5
948	IBM-948, x-IBM948, Cp948	Datos de PC: SBCS extendido . Datos de PC: DBCS que incluye 6204 caracteres definidos por el usuario
949	IBM-949, IBM-949C, x-IBM949, x-IBM949C, Cp949, Cp949C	Código IBM KS - datos de PC: SBCS. Código IBM KS - datos de PC: DBCS que incluye 1880 caracteres definidos por el usuario
950	Big5, IBM-950, x-IBM950, Cp950	Datos de PC: SBCS (IBM BIG5). Datos de PC: DBCS, incluidos 13 493 caracteres CNS, 566 caracteres seleccionados por IBM y 6204 caracteres definidos por el usuario
951	IBM-951	Datos de PC: IBM KS



CCSID	Juegos de caracteres	Descripción
954	EUC-JP, IBM-954, IBM-954C	G0: JIS X201 latino. G1: JIS X208-1990. G1: JIS X201 katakana. G1: JIS X212
964	EUC-TW, IBM-964, x-IBM964, Cp964	G0: ASCII. G1: CNS 11 643 plano 1. G1: CNS 11 643 plano 2.
970	EUC-KR, x-IBM970, Cp970	G0: ASCII. G1: KSC X5601-1989 que incluye 1880 caracteres definidos por el usuario
971	IBM-971	EUC COREANO
1006	IBM-1006, x-IBM1006, Cp1006	ISO-8: urdu
1025	IBM-1025, x-IBM1025, Cp1025	Host: cirílico multilingüe
1026	IBM1026, IBM-1026, Cp1026	Host: Latin-5 (Turquía)
1027	IBM-1027	EBCD JAPONÉS LATINO
1041	IBM-1041	Datos de PC: Japón
1043	IBM-1043	Datos de PC: chino-T
1046	IBM-1046, IBM-1046S, x-IBM1046	ÁRABE - PC
1047	IBM1047, IBM-1047	Host: Latin-1
1051	hp-roman8	EMULACIÓN HP
1088	IBM-1088	Datos de PC: Corea KS

CCSID	Juegos de caracteres	Descripción
1089	ISO-8859-6, ISO8859_6	ISO 8859-6: árabe
1097	IBM-1097, x-IBM1097, Cp1097	Host: farsi
1098	IBM-1098, x-IBM1098, Cp1098	Datos de PC: farsi
1112	IBM-1112, x-IBM1112, Cp1112	Host: Letonia, Lituania
1114	IBM-1114	Datos de PC: T-CH SB
1115	IBM-1115	Datos de PC: S-CH SB
1122	IBM-1122, x-IBM1122, Cp1122	Host: Estonia
1123	IBM-1123, x-IBM1123, Cp1123	Host: Ucrania cirílico
1124	IBM-1124, x-IBM1124, Cp1124	8 bits: cirílico, Bielorrusia
1140	IBM01140, IBM-1140, Cp1140	Host: EE. UU., Canadá (ESA), Países Bajos, Portugal, Brasil, Australia, Nueva Zelanda, con euro
1141	IBM01141, IBM-1141, Cp1141	Host: Austria, Alemania, con euro
1142	IBM01142, IBM-1142, Cp1142	Host: Dinamarca, Noruega, con euro
1143	IBM01143, IBM-1143, Cp1143	Host: Finlandia, Suecia, con euro

CCSID	Juegos de caracteres	Descripción
1144	IBM01144, IBM-1144, Cp1144	Host: Italia, con euro
1145	IBM01145, IBM-1145, Cp1145	Host: España, Latinoamérica (español), con euro
1146	IBM01146, IBM-1146, Cp1146	Host: Reino Unido, con euro
1147	IBM01147, IBM-1147, Cp1147	Host: Francia, con euro
1148	IBM01148, IBM-1148, Cp1148	Host: Bélgica, Canadá (AS/400), Suiza, Latin-1, con euro
1149	IBM01149, IBM-1149, Cp1149	Host: Islandia, con euro
1200	UTF-16BE	Unicode con conjunto de caracteres 65535. En ausencia de una marca de orden de bytes (BOM), se supone que es UTF-16 BE (big-endian).
1202	UTF-16LE	UTF-16 LE con IBM PUA
1204	UTF-16	UTF-16 con IBM PUA
1208	UTF-8, UTF-8J, UTF8	Unicode con conjunto de caracteres 65535. UTF-8.
1232	UTF-32BE	UTF-32 BE con IBM PUA
1234	UTF-32LE	UTF-32 LE con IBM PUA
1236	UTF-32	UTF-32 con IBM PUA
1351	IBM-1351	JAPÓN ABIERTO
1362	IBM-1362	COREANO MS-WIN

CCSID	Juegos de caracteres	Descripción
1363	IBM-1363, IBM-1363C, windows-949, MS949	Datos de PC: MS Windows coreano SBCS. Datos de PC: MS Windows coreano DBCS, incluido el hangul completo 11 172
1364	IBM-1364	Host: SBCS extendido. Host: DBCS, que incluye 1880 caracteres definidos por el usuario y 11 172 caracteres en hangul completo
1370	IBM-1370	Datos de PC: SBCS extendido , con euro. Datos de PC: DBCS que incluye 6204 caracteres definidos por el usuario, con euro
1371	IBM-1371	Host: SBCS extendido, con euro. Host: DBCS que incluye 6204 caracteres definidos por el usuario, con euro
1375	Big5-HKSCS	Big-5 Ext mixto para HKSCS
1380	IBM-1380	Datos de PC: S-CH SB
1381	IBM-1381, x-IBM1381, Cp1381	Datos de PC: SBCS extendido (IBM GB). Datos de PC: DBCS (IBM GB), incluidos 31 caracteres seleccionados por IBM y 1880 definidos por el usuario
1382	IBM-1382	EUC CHINO-S

CCSID	Juegos de caracteres	Descripción
1383	EUC-CN, GB2312, IBM-1383, x-IBM1383, Cp1383	G0: ASCII. G1: conjunto GB 2312-80
1385	IBM-1385	Datos de PC: S-CH GBK
1386	GBK, IBM-1386, windows-936, MS936	Datos de PC: GBK chino-S y IBM BIG-5 chino-T. Datos de PC: S-CH GBK
1388	IBM-1388	Host: SBCS extendido. Host: DBCS, que incluye 1880 caracteres definidos por el usuario
1390	IBM-1390	Host katakana: SBCS extendido, con euro. Host kanji: DBCS que incluye 6205 caracteres definidos por el usuario
1399	IBM-1399	Host latino: SBCS extendido , con euro. Host kanji: DBCS que incluye 4370 caracteres definidos por el usuario, con euro
5050	JIS0201, JIS0208, JIS0212, JIS0201, JIS0208, JIS0212	G0: JIS X201 latino. G1: JIS X208-1990. G1: JIS X201 katakana. G1: JIS X212
5054	ISO-2022-JP	TCP JAPONÉS
5346	windows-1250, Cp1250	MS Windows: Latin-2, versión 2 con euro
5347	windows-1251, Cp1251	MS Windows: cirílico, versión 2 con euro

CCSID	Juegos de caracteres	Descripción
5348	windows-1252, Cp1252	MS Windows: países Latin-1, versión 2 con euro
5349	windows-1253, Cp1253	MS Windows: Grecia, versión 2 con euro
5350	windows-1254, Cp1254	MS Windows: Turquía, versión 2 con euro
5351	windows-1255, Cp1255	MS Windows: hebreo, versión 2 con euro
5352	windows-1256, windows-1256S, Cp1256	MS Windows: árabe, versión 2 con euro
5353	windows-1257, Cp1257	MS Windows: región báltica, versión 2 con euro
5354	windows-1258, Cp1258	MS Windows: vietnamita, versión 2 con euro
5488	GB18030	GB18030, datos de 1 byte GB18030, datos de 2 bytes GB18030, datos de 4 bytes
9030	IBM-838, Cp838	Host: SBCS extendido tailandés.
9066	IBM-874, Cp874	Datos de PC: SBCS extendido tailandés
9400	CESU-8	CESU-8 con IBM PUA
25546	ISO-2022-KR	TCP COREANO
33722	IBM-33722, IBM-33722C	IBMeucJP

# Transferencia de archivos en la modernización del AWS mainframe

AWS Mainframe Modernization File Transfer le permite transferir y convertir conjuntos de datos de mainframe a Amazon S3 para casos de uso de modernización, migración y aumento de mainframes.

## Temas

- [¿Qué es AWS Mainframe Modernization File Transfer?](#)
- [Instalación de un agente de File Transfer](#)
- [Puntos de conexión de transferencia de datos](#)
- [Tareas de transferencia](#)
- [Tutorial: Introducción a AWS Mainframe Modernization File Transfer](#)
- [Codificaciones de origen y destino compatibles con la transferencia de archivos de AWS Mainframe Modernization](#)

## ¿Qué es AWS Mainframe Modernization File Transfer?

Con AWS Mainframe Modernization File Transfer, puede transferir y convertir conjuntos de datos y archivos con un servicio totalmente gestionado para acelerar y simplificar los casos de uso de modernización, migración y aumento al servicio AWS Mainframe Modernization y Amazon S3.

## Temas

- [Ventajas de AWS Mainframe Modernization File Transfer](#)
- [Cómo funciona AWS Mainframe Modernization File Transfer](#)

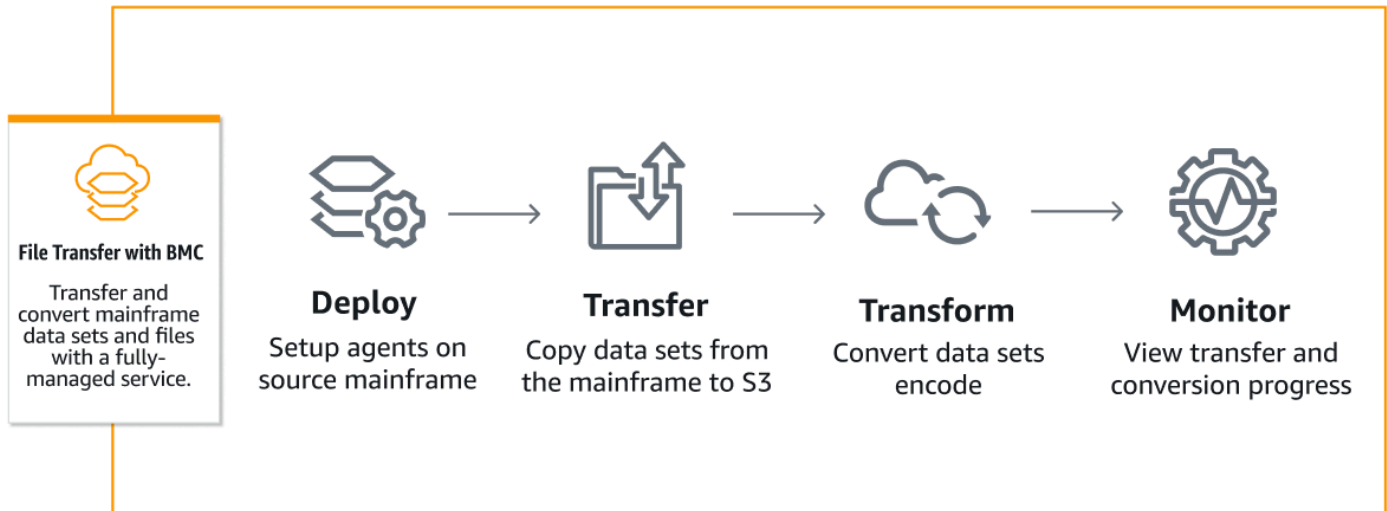
## Ventajas de AWS Mainframe Modernization File Transfer

AWS Mainframe Modernization File Transfer le ayuda a transferir conjuntos de datos del mainframe a Amazon S3. Algunas de sus ventajas son:

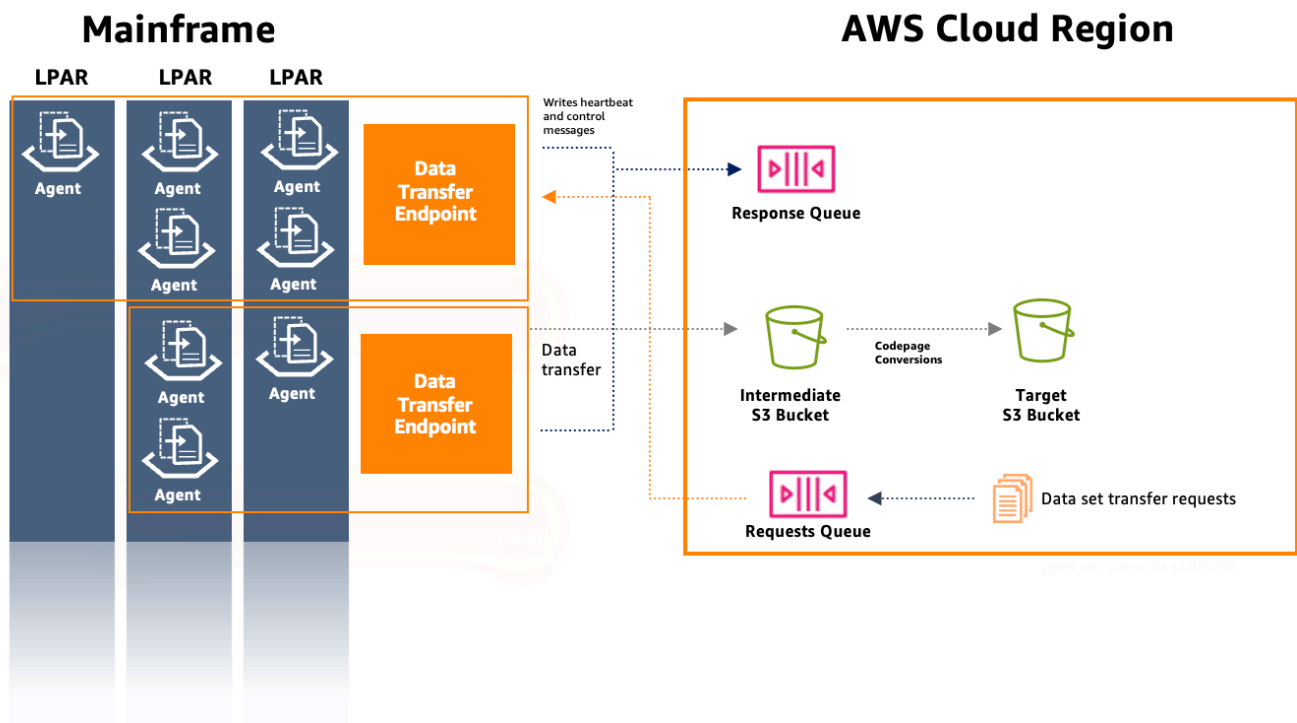
- Detección de conjuntos de datos y artefactos de mainframe de origen
- Transferencia y conversión de conjuntos de datos automatizada
- Escalabilidad, eficiencia y velocidad para lograr transferencias de conjuntos de datos a AWS más rápidas

# Cómo funciona AWS Mainframe Modernization File Transfer

En la siguiente figura se muestra una descripción general de cómo funciona AWS Mainframe Modernization File Transfer en un nivel conceptual.



La siguiente figura es una descripción general de la arquitectura de la característica AWS Mainframe Modernization File Transfer.





# Instalación de un agente de File Transfer

Siga esta guía step-by-step para completar los requisitos previos para instalar un agente en el mainframe de origen y configurar el agente.

## Temas

- [Paso 1: Iniciar sesión en la ISPF](#)
- [Paso 2: Asignar un conjunto de datos para el z/FS](#)
- [Paso 3: Formatear el conjunto de datos como z/FS](#)
- [Paso 4: Definir el sistema de archivos para z/OS](#)
- [Paso 5: Montar el sistema de archivos](#)
- [Paso 6: Verificar el montaje](#)
- [Paso 7: Introducir los OMV](#)
- [Paso 8: Definir la variable de entorno del directorio de instalación del agente](#)
- [Paso 9: Definir la variable de entorno del directorio de trabajo](#)
- [Paso 10: Creación del directorio de trabajo](#)
- [Paso 11: Copie el paquete tar de modernización de AWS mainframe al directorio de trabajo de z/OS](#)
- [Paso 12: Asumir el usuario raíz](#)
- [Configuración de permisos y STC](#)
- [Creación de un usuario de IAM con credenciales de acceso a largo plazo](#)
- [Creación de un rol de IAM para que lo asuma el agente](#)
- [Configuración del agente](#)

## Paso 1: Iniciar sesión en la ISPF

Inicie sesión en la ISPF (Interactive System Productivity Facility). Por lo general, esto se hace a través de un emulador de terminal 3270.

## Paso 2: Asignar un conjunto de datos para el z/FS

Con la utilidad de conjuntos de datos del ISPF, asigne un nuevo conjunto de datos para el z/FS. Normalmente, esto se hace en la utilidad de lista de conjuntos de datos del paso 1.

1. Vaya a la opción 3.4 (Utilidades -> Conjunto de datos).
2. Pulse la tecla “C” para crear un nuevo conjunto de datos.
3. Introduzca el nombre del conjunto de datos (por ejemplo, “yourhlq.M2AGENT.ZFS”).
4. Especifique el tipo de conjunto de datos como “Large format” con un tamaño principal de 1000 cilindros y un tamaño secundario de 200.
5. Establezca la organización del conjunto de datos (DSORG) como PS y el formato de registro (RECFM) como “U” (indefinido).
6. Completar el proceso de creación.

### Paso 3: Formatear el conjunto de datos como z/FS

Después de crear el conjunto de datos, formateéelo como un sistema de archivos z/FS.

Una forma de hacerlo es mediante el siguiente lenguaje de control de tareas (JCL):

```
//FORMAT EXEC PGM=IOEAGFMT,PARM='AGGRNAME(yourhlq.M2AGENT.ZFS),FORMAT,AGGRSIZE(1200)'  
//SYSPRINT DD SYSOUT=A
```

Envíe este trabajo y compruebe si se ha completado correctamente.

### Paso 4: Definir el sistema de archivos para z/OS

Defina de z/FS a z/OS mediante el comando DEF FILESYSTEM o mediante un trabajo por lotes.

Utilice el siguiente comando:

```
DEF FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(R/W) MOUNTPOINT('/usr/lpp/aws/m2-agent')
```

#### Note

Este paso requiere una autoridad en el nivel del sistema.

### Paso 5: Montar el sistema de archivos

Para montar el sistema de archivos, utilice el comando MOUNT. Puede montar el sistema de archivos en la línea de comandos, en la ISPF o por lotes.

Por ejemplo:

```
MOUNT FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/usr/lpp/aws/m2-agent')
```

## Paso 6: Verificar el montaje

Verifique que el sistema de archivos esté montado correctamente mediante un comando `D OMVS, A` o comprobándolo en el Servicio de Sistema Unix (USS).

## Paso 7: Introducir los OMV

Utiliza el siguiente comando para introducir los OMV:

```
TSO OMVS
```

## Paso 8: Definir la variable de entorno del directorio de instalación del agente

Utilice el siguiente comando para definir el entorno del directorio de instalación del agente:

```
export AGENT_DIR=/usr/lpp/aws/m2-agent
```

## Paso 9: Definir la variable de entorno del directorio de trabajo

Utilice el siguiente comando para definir la variable de entorno del directorio de trabajo:

```
export WORK_DIR=$AGENT_DIR/tmp
```

## Paso 10: Creación del directorio de trabajo

Utilice el siguiente comando para definir el entorno del directorio de trabajo:

```
mkdir -p $WORK_DIR
```

## Paso 11: Copie el paquete tar de modernización de AWS mainframe al directorio de trabajo de z/OS

Cuando utilice FTP o cualquier método de transferencia, asegúrese de que el archivo tar se transfiera en modo binario.

## Paso 12: Asumir el usuario raíz

Utilice el siguiente comando para asumir el usuario raíz:

```
su
```

Siga estos pasos para finalizar la instalación del agente:

### Note

Debe asumir el usuario raíz antes de continuar con estos pasos.

1. Establezca la variable de entorno de la versión m2-agent en la versión que se esté instalando actualmente mediante el siguiente comando:

```
export M2_AGENT_VERSION=1.0.0
```

2. Extraiga el paquete tar del agente con el siguiente comando:

```
tar -xpf m2-agent-package- $\$M2\_AGENT\_VERSION$ .tar -C  $\$AGENT\_DIR$ 
```

3. Cree un enlace simbólico `current-version` al directorio de instalación del agente actual con el siguiente comando:

```
ln -s  $\$AGENT\_DIR/m2-agent-v\mathit{\$M2\_AGENT\_VERSION}$   $\$AGENT\_DIR/current-version$ 
```

4. Actualice y envíe CPY#PDS para crear los conjuntos de datos del agente de File Transfer.

### Note

JCL usa el HLQ SYS2.AWS.M2.

Para crear el agente de File Transfer, defina las líneas de parámetros 000006-000012. Además, actualice las tres variables simbólicas HLQ, VOLSER y AGNTPATH usarlas más adelante en el JCL:

```
oedit $AGENT_DIR/current-version/installation/CPY#PDS
submit $AGENT_DIR/current-version/installation/CPY#PDS
```

### Note

Este JCL está diseñado para configurar ciertos aspectos de la instalación del agente en el mainframe. Asigna los conjuntos de datos necesarios y, a continuación, copia archivos específicos del sistema de archivos Unix a estos conjuntos de datos.

## Configuración de permisos y STC

1. Actualice y envíe o bien `SYS2.AWS.M2.SAMPLIB(SEC#RACF)` (para configurar los permisos de RACF) o bien `SYS2.AWS.M2.SAMPLIB(SEC#TSS)` (para configurar los permisos de TSS) de acuerdo con sus instrucciones. Estos miembros se crearon en el paso CPY#PDS anterior.
2. Actualice la exportación de PWD en el `SYS2.AWS.M2.SAMPLIB(M2AGENT)` STC JCL si se ha cambiado la ruta de directorio predeterminada del agente de File Transfer (`/usr/lpp/aws/m2-agent`).
3. Actualice y copie el `SYS2.AWS.M2.SAMPLIB(M2AGENT)` JCL a `SYS1.PROCLIB`.
4. Añada `SYS2.AWS.M2.LOADLIB` a la lista de APF utilizando el siguiente comando:

```
SETPROG APF ADD DSNAME(SYS2.AWS.M2.LOADLIB) SMS
```

5. Defina el logs del agente y el grupo de carpetas `diag` en el usuario/grupo del agente (`M2USER/M2GROUP`). Utilice el siguiente comando:

```
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/logs
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/diag
```

## Creación de un usuario de IAM con credenciales de acceso a largo plazo

Debe crear el agente de mainframe que necesita un usuario de IAM para usar las colas de respuestas y solicitudes y guardar los conjuntos de datos en los buckets de Amazon S3.

Al crear este usuario:

1. Seleccione Asociar políticas directamente en las opciones de Permisos.
2. Una vez creado el usuario, abra la pestaña de Credenciales de seguridad y cree una clave de acceso. A fin de obtener más información acerca de la creación de claves de acceso de IAM, consulte [Administración de claves de acceso para usuarios de IAM](#).
3. En la sección Clave de acceso, seleccione Otro cuando se le pregunte por el Caso de uso.

### Note

Guarde la Clave de acceso y la Clave de acceso secreta que aparecen en la última página del asistente de creación de claves de acceso antes de seleccionar Listo. Estas claves se utilizan para configurar el agente de mainframe.

### Note

Guarde el ARN del usuario de IAM utilizado para configurar una relación de confianza con un rol de IAM.

## Creación de un rol de IAM para que lo asuma el agente

Cree un nuevo rol de IAM con una Política de confianza personalizada para el Tipo de entidad de confianza. La política utilizará la siguiente plantilla:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DataTransferEndpointAgentSqsReceive",
      "Effect": "Allow",
      "Action": [
```

```

        "sqs:DeleteMessage",
        "sqs:ReceiveMessage"
    ],
    "Resource": "<data-transfer-endpoint-request-queue-arn>"
  },
  {
    "Sid": "DataTransferEndpointS3",
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "<data-transfer-endpoint-intermediate-bucket-arn>/*"
  },
  {
    "Sid": "DataTransferEndpointAgentSqsSend",
    "Effect": "Allow",
    "Action": "sqs:SendMessage",
    "Resource": "<data-transfer-endpoint-response-queue-arn>"
  },
  {
    "Sid": "DataTransferEndpointAgentKmsDecrypt",
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "<kms-key-id>"
  }
]
}

```

Donde:

- `request-queue-arn` y `response-queue-arn` son el ARN de la cola de solicitudes de Amazon SQS creada durante la inicialización del punto de conexión de transferencia de datos.
- `transfer-bucket-arn` es el ARN del bucket de transferencia creado anteriormente.

#### Note

Puede buscar todos estos valores mediante la consola de AWS.

#### Note

Guarde el nombre de rol, que utilizará más adelante para configurar el agente de mainframe.

## Configuración del agente

Para configurar el agente de File Transfer:

1. Vaya a `$AGENT_DIR/current-version/config`.
2. Edite el archivo de configuración del agente `application.properties` para agregar una configuración de entorno mediante el siguiente comando:

```
oedit $AGENT_DIR/current-version/config/application.properties
```

Por ejemplo:

```
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>
```

Donde:

- `AWS_ACCOUNT_ID` es el ID de la cuenta del cliente.
- `AWS_IAM_ROLE_NAME` es el nombre del rol de IAM creado en la [the section called “Creación de un rol de IAM para que lo asuma el agente”](#).
- `AWS_IAM_ROLE_ACCESS_KEY` es la clave de acceso del usuario de IAM creado en [the section called “Creación de un usuario de IAM con credenciales de acceso a largo plazo”](#).
- `AWS_IAM_ROLE_SECRET_KEY` es la clave secreta de acceso del usuario de IAM creado en [the section called “Creación de un usuario de IAM con credenciales de acceso a largo plazo”](#).
- `AWS_S3_BUCKET_NAME` es el nombre del bucket de transferencia creado con el punto de conexión de transferencia de datos.
- `AWS_REGION` es la región en la que se configura el agente de File Transfer.



**Note**

Puede haber varias secciones de este tipo, siempre que el índice entre corchetes, `[0]`, se incremente para cada una de ellas.

Debe reiniciar el agente para que los cambios surtan efecto.

**Requisitos**

1. Cuando se agrega o elimina un parámetro, hay que detener e iniciar el agente. Inicie el agente de File Transfer mediante el siguiente comando de la CLI:

```
/S M2AGENT
```

Para detener el agente de M2, ejecute el comando siguiente en la CLI:

```
/P M2AGENT
```

2. Puede hacer que el agente de transferencia de archivos se transfiera a varias regiones y cuentas AWS definiendo varios entornos.

```
#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
```

## Puntos de conexión de transferencia de datos

Los puntos de enlace de transferencia de datos permiten la conectividad con el mainframe de origen y admiten una alta disponibilidad, escalabilidad y una administración simplificada de los agentes. Los agentes individuales se instalan en los LPAR del mainframe y se pueden agrupar en un punto de conexión de transferencia de datos. Cuando se solicita la transferencia de un conjunto de datos, un agente del punto final de transferencia de datos se encargará de esa transferencia específica. Para iniciar las transferencias de datos, al menos un agente del punto de conexión de transferencia de datos debe estar en línea.

Este procedimiento supone que ha completado los pasos de [Configuración de la modernización del AWS mainframe](#) y de [Configurar el agente de File Transfer en el mainframe de origen](#).

### Creación de un punto de conexión de transferencia de datos

Para crear puntos finales de transferencia de datos para la transferencia de archivos, siga estos pasos en la consola de modernización del AWS mainframe.

Para crear un punto de conexión de transferencia de datos

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el Región de AWS selector, elija la región a la que quiere transferir los archivos del mainframe a un bucket de Amazon S3.
3. En la página Puntos de conexión de transferencia de datos, en File Transfer, elija Crear punto de conexión de transferencia de datos.
4. En la página de requisitos previos del punto final de transferencia de datos, lea todas las instrucciones para asegurarse de haber completado estos pasos en el mainframe de origen. Una vez confirmado, seleccione Siguiente.
5. En la página Configurar punto de conexión de transferencia de datos, agregue la información básica del punto de conexión de transferencia de datos.
  1. En la sección de información básica, introduzca el nombre del punto final de transferencia de datos.

**Note**

El nombre del punto final de transferencia de datos debe coincidir con el nombre de Sysplex, a menos que especifique un nombre complejo en la configuración del agente.

2. Una descripción opcional.
3. La clave KMS utilizada para cifrar el secreto.

**Note**

Debe agregar la siguiente política basada en recursos para KMS para que el servicio de AWS Mainframe Modernization pueda leer y usar estas claves para el cifrado y el descifrado:

```
{
  "Sid" : "Enable AWS M2 Permissions",
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : [
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Resource" : "*"
}
```

4. Especifique la ubicación S3 para los datos intermedios, que es la ubicación S3 intermedia donde se almacenan los conjuntos de datos transferidos desde el mainframe antes de convertirlos y transferirlos al bucket de Amazon S3 de destino.

**Note**

Se recomienda crear un nuevo bucket de Amazon S3 para las tareas de transferencia. Para obtener más información, consulte [Creación de un bucket](#).

También puede explorar sus buckets de Amazon S3 existentes seleccionando la opción Explorar S3.

5. Tras introducir los campos obligatorios, seleccione Siguiente.
6. En la página Revisar y crear un punto de conexión de transferencia de datos, compruebe si ha cumplido los requisitos previos y revise la información básica. Una vez confirmado, elija Crear punto final de transferencia de datos.

Se le redirigirá a la página de Información general de los puntos finales de transferencia de datos, donde podrá ver la lista de todos los puntos de conexión de transferencia de datos. También podrá ver los puntos de conexión de transferencia de datos que están disponibles o que han fallado.

También puede buscar los puntos de conexión de transferencia de datos por nombre y acceder a información adicional sobre cada agente disponible.

## Tareas de transferencia

Las tareas de transferencia se utilizan para especificar los conjuntos de datos que se van a transferir del mainframe a Amazon S3 y le permiten elegir las opciones de conversión de páginas de códigos.

En estas instrucciones se da por sentado que ha completado los pasos [Configuración de la modernización del AWS mainframe](#) y que los ha creado [the section called “Puntos de conexión de transferencia de datos”](#).

### Temas

- [Creación de tareas de transferencia](#)
- [Visualización de las tareas de transferencia](#)

## Creación de tareas de transferencia

Para crear tareas de transferencia en File Transfer, siga estos pasos en la consola de modernización del AWS mainframe.

## Para crear una tarea de transferencia

### Important

Debe tener al menos un punto de conexión de transferencia de datos para crear nuevas tareas de transferencia.


1. [Abra la consola de modernización AWS de mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
  2. En el Región de AWS selector, elija la región a la que desea transferir los archivos de su mainframe a un bucket de Amazon S3.
  3. En la página de tareas de transferencia, puede elegir cualquier punto final de transferencia de datos para crear las tareas de transferencia.
  4. En la página Crear tarea de transferencia, configure las propiedades de la tarea de transferencia. Si no ha creado ninguna tarea de transferencia anteriormente, puede crear la primera tarea seleccionando la opción Crear tarea de transferencia.
- En esta página, introduce la información básica de la tarea de transferencia, incluidos el nombre, la descripción y la clave secreta de la tarea de transferencia.

### Note

- Cifre el secreto con la clave KMS definida en el punto de conexión de transferencia de datos. El secreto debe contener las credenciales del ordenador central necesarias para acceder a los conjuntos de datos del ordenador central mediante las claves `userId` y `password`. Para obtener más información, consulte el [secreto de AWS Secrets Manager](#).
- Debe configurar la clave secreta con la siguiente política basada en recursos para que el servicio de modernización del AWS mainframe pueda acceder a ella para realizar tareas de transferencia de datos.

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "m2.amazonaws.com"
    }
  },
```

```
"Action" : [ "secretsmanager:GetSecretValue",  
            "secretsmanager:DescribeSecret" ],  
"Resource" : "*" ]  
} ]  
}
```

 Note

El tamaño máximo de conjunto de datos actualmente admitido para la transferencia es de 90 GB.

- A continuación, seleccione la ubicación del depósito Amazon S3 de destino a la que se transferirán los conjuntos de datos de destino del mainframe.
  - Se seleccionará el punto final de transferencia de datos previamente elegido. También puede seleccionar otro punto final entre los puntos finales disponibles.
5. Elija Siguiente.
  6. En la página Añadir conjuntos de datos, introduzca la consulta en el servidor central de búsqueda de conjuntos de datos para buscar conjuntos de datos en el ordenador central para incluirlos en la tarea de transferencia. Seleccione Ver conjuntos de datos.

Los siguientes símbolos comodín se pueden utilizar como parte de los criterios de búsqueda de conjuntos de datos para el ordenador central:

- Un solo asterisco (\*) como clasificador (entre períodos o después del último período) coincide con un solo calificador en esa posición.
- Un solo asterisco (\*) dentro de un calificador coincide con cero o más caracteres en esa posición.
- Un asterisco doble (\*\*) como clasificador (entre períodos o después del último período) coincide con cero o más clasificatorios en esa posición.
- Un asterisco doble (\*\*) dentro de un calificador no es una consulta válida.
- Un único signo de porcentaje (%) coincide con cualquier carácter alfanumérico o nacional que se encuentre en esa posición. Puedes usar hasta un ocho por ciento en cada calificador.

**⚠ Important**

Te sugerimos que termines siempre tus criterios de búsqueda con un punto seguido de un asterisco doble (.\*\*) y, si es necesario, afinar aún más la búsqueda.

Para obtener más información sobre las reglas comodín, consulte la documentación de IBM sobre el [filtrado de nombres de conjuntos de datos](#).

- Estos conjuntos de datos se cargarán en la sección de conjuntos de datos de mainframe, donde puede buscar o elegir uno o más conjuntos de datos para los que desee configurar las conversiones de páginas de códigos. Los conjuntos de datos elegidos se mostrarán en la sección Conjuntos de datos agregados.

**ℹ Note**

Puede seleccionar conjuntos de datos de varias consultas de búsqueda y añadirlos a la tarea de transferencia.

- En la sección Conjuntos de datos agregados, debe seleccionar manualmente la página de códigos fuente y la página de códigos de destino para cada uno de los conjuntos de datos que elija. La página de códigos fuente es el formato del conjunto de datos de origen y la página de códigos de destino es el formato del conjunto de datos de destino que se utiliza para convertir los conjuntos de datos y almacenarlos en el bucket de Amazon S3 de destino.
- Tras confirmar las páginas de códigos fuente y de destino, seleccione Siguiente.
- En la página Revisar y crear, puede revisar o editar la información para la tarea de transferencia.
- A continuación, selecciona Crear tarea de transferencia.

## Visualización de las tareas de transferencia

Para ver las tareas de transferencia en File Transfer, debe seguir estos pasos en la consola de modernización del AWS mainframe.

Para ver las tareas de transferencia

- [Abra la consola de modernización AWS del mainframe en https://console.aws.amazon.com/m2/.](https://console.aws.amazon.com/m2/)

2. En el Región de AWS selector, elija la región a la que desea transferir los archivos de su mainframe a un bucket de Amazon S3.
3. En la página Tareas de transferencia, seleccione el punto final de transferencia de datos para ver sus tareas de transferencia.
4. En el caso de los puntos finales que tienen tareas de transferencia preexistentes, estas se mostrarán en la sección Tareas de transferencia. Puede elegir ver los detalles de cualquier tarea de transferencia de esta lista.

## Tutorial: Introducción a AWS Mainframe Modernization File Transfer

AWS Mainframe Modernization File Transfer le permite transferir y convertir conjuntos de datos de mainframe para casos de uso de modernización, migración y aumento de mainframes.

Siga los pasos de este tutorial para entender cómo funciona AWS Mainframe Modernization File Transfer.

### Información general

File Transfer consta de lo siguiente:

1. Un agente que se instalará en el mainframe de origen.
2. Acceda a las funciones de descubrimiento, transferencia y conversión de conjuntos de datos directamente desde la consola del servicio de administración de AWS Mainframe Modernization.

Como usuario, puede transferir conjuntos de datos desde el mainframe a su bucket de Amazon S3.

### Temas

- [Paso 1: Transfiera el paquete tar de los archivos binarios del agente AWS a la partición lógica del mainframe](#)
- [Paso 2: Configurar el agente de File Transfer en el mainframe de origen](#)
- [Paso 3: Creación de un punto de conexión de transferencia de datos](#)
- [Paso 4: Crear una tarea de transferencia](#)
- [Paso 5: Ver el progreso de la tarea de transferencia](#)



## Paso 1: Transfiera el paquete tar de los archivos binarios del agente AWS a la partición lógica del mainframe

Descargue los archivos tar desde el enlace de [M2-agent tar](#).

## Paso 2: Configurar el agente de File Transfer en el mainframe de origen

En este paso, debe configurar e iniciar el agente de AWS Mainframe Modernization File Transfer en el mainframe de origen. El agente debe facilitar las comunicaciones entre la característica del servicio de File Transfer y el mainframe de origen. Se requiere al menos un agente por mainframe. Se puede iniciar más de un agente para obtener una alta disponibilidad y una escalabilidad mejorada.

Siga las instrucciones de la guía de [the section called “Instalación de un agente de File Transfer”](#) para completar la instalación del agente de File Transfer en el mainframe.

## Paso 3: Creación de un punto de conexión de transferencia de datos

Siga los pasos de la página [the section called “Puntos de conexión de transferencia de datos”](#) para crear un nuevo punto de conexión de transferencia de datos.

## Paso 4: Crear una tarea de transferencia

Siga los pasos de la página [the section called “Tareas de transferencia”](#) para crear y gestionar las tareas de transferencia.

## Paso 5: Ver el progreso de la tarea de transferencia

Puede ver el progreso de la tarea de transferencia en la consola de modernización del AWS mainframe. Para obtener más detalles, consulte la sección [the section called “Visualización de las tareas de transferencia”](#).

## Codificaciones de origen y destino compatibles con la transferencia de archivos de AWS Mainframe Modernization

AWS Mainframe Modernization File Transfer admite varios tipos de conjuntos de datos y opciones de conversión de páginas de códigos.

## Tipos de conjuntos de datos de mainframe

La transferencia de archivos de modernización de mainframe de AWS admite los siguientes tipos de conjuntos de datos de mainframe:

- No VSAM: secuencial (PS), PDS, GDS, GDG
- Tipos de VSAM: KDS

## Páginas de códigos compatibles

La transferencia de archivos de modernización de mainframe de AWS admite las siguientes páginas de códigos para la conversión de conjuntos de datos (de/a):

«BIG5», «BIG5\_HKSCS», «CESU\_8», «EUC\_JP», «EUC\_KR», «GB18030», «GB2312», «GBK», «IBM00858», «IBM01140», «IBM01141», «IBM01142», «IBM01143», «IBM01145», «IBM01146», «IBM01147», «IBM01148», «IBM01149», «IBM037», «IBM1026», «IBM1047», «IBM273», «IBM277», «IBM278», «IBM280», «IBM284», «IBM284" 5», «IBM290», «IBM297», «IBM420», «IBM424», «IBM437», «IBM500», «IBM775», «IBM850», «IBM852», «IBM855», «IBM857», «IBM860», «IBM861», «IBM862», «IBM863», «IBM863», «IBM863" 864», «IBM865», «IBM866», «IBM868», «IBM869», «IBM870», «IBM871», «IBM918», «IBM\_THAI», «ISO\_2022\_CN», «ISO\_2022\_JP», «ISO\_2022\_JP\_2», «ISO\_2022\_KR», «ISO\_8859\_1», «ISO\_8859\_13», «ISO\_8859\_15», «ISO\_8859\_16», «ISO\_8859\_2», «ISO\_8859\_2" \_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\_32», «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», «X\_BIG5\_HKSCS\_2001», «X\_BIG5\_SOLARIS», «X\_EUCJP\_OPEN», «X\_EUC\_JP\_LINUX», «X\_EUC\_TW», «X\_IBM1006», «X\_IBM1025», «X\_IBM1046», «X\_IBM1097», «X\_IBM1098», «X\_IBM1112», «X\_IBM1122», «X\_IBM1123», «X\_IBM1124», «X\_IBM1129», «X\_IBM1166», «X\_IBM1364», «X\_IBM1381», «X\_IBM1381", «X\_IBM1381" M1383», «X\_IBM29626C», «X\_IBM300», «X\_IBM33722», «X\_IBM737», «X\_IBM833», «X\_IBM834», «X\_IBM856», «X\_IBM874», «X\_IBM875», «X\_IBM921», «X\_IBM921" 922», «X\_IBM930», «X\_IBM933», «X\_IBM935», «X\_IBM937», «X\_IBM939», «X\_IBM942», «X\_IBM\_942C», «X\_IBM943», «X\_IBM943C», «X\_IBM948», «X\_IBM949», «X\_IBM949C», «X\_IBM950», «X\_IBM964», «X\_IBM970», «X\_ISCII91», «X\_ISO\_2022\_CN\_CNS», «X\_ISO\_2022\_CN\_GB», «X\_ISO\_8859\_11», «X\_JIS0208», «X\_JISAUTODETECT», «X\_JOHAB», «X\_MACARABIC», «X\_MACCENTRALEUROPE», «X\_MACCROATIAN», «X\_MACCIRÍLICO»,

«X\_MACDINGBAT», «XMACDINGBAT», «XMAC\_DINGBAT» GREEK», «X\_MACHEBREW»,  
«X\_MACICELAND», «X\_MACROMAN», «X\_MACROMANIA», «X\_MACSYMBOL», «X\_MACTHAI»,  
«X\_MACTURKISH», «X\_MACUKRAINE», «X\_MS932\_0213", «X\_MS950\_HKSCS»,  
«X\_MS950\_HKSCS\_XP», «X\_MSWIN\_936", «X\_PCK», «X\_SJIS\_0213", «X\_UTF\_16LE\_BOM»,  
«X\_UTF\_32BE\_BOM», «X\_UTF\_32LE\_BOM», «X\_WINDOWS\_50220", «X\_WINDOWS\_50221",  
«X\_WINDOWS\_874", «X\_WINDOWS\_949", «X\_WINDOWS\_950", «X\_Windows\_ISO2022J»

# La seguridad en la modernización del mainframe AWS

La seguridad en la nube AWS es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad.

La seguridad es una responsabilidad compartida entre usted AWS y usted. El [modelo de responsabilidad compartida](#) la describe como seguridad de la nube y seguridad en la nube:

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta AWS los servicios en la AWS nube. AWS también le proporciona servicios que puede utilizar de forma segura. Los auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los [AWS programas](#) de de . Para obtener más información sobre los programas de conformidad que se aplican a la modernización del AWS mainframe, consulte [Servicios de AWS por programa de conformidad](#) .
- Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos aplicables

Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida al utilizar la modernización del AWS mainframe. Le muestra cómo configurar la modernización del AWS mainframe para cumplir sus objetivos de seguridad y conformidad. También aprenderá a utilizar otros AWS servicios que le ayudan a supervisar y proteger sus recursos de modernización del AWS mainframe.

AWS La modernización del mainframe proporciona sus propios recursos protegidos por la IAM (aplicación, entorno, despliegue, etc.), que son los recursos administrativos de modernización del AWS mainframe, sobre los que las políticas de IAM deben permitir cualquier acción.

AWS La modernización del mainframe para la reconfiguración de plataformas también está asegurada por la IAM. IAM concede o deniega el permiso a una entidad principal para realizar una acción específica en un recurso definido, derivado del entorno de la unidad central original, también mediante políticas de IAM estándar. El tiempo de ejecución de la replataforma de AWS Mainframe Modernization llama al servicio de autorización de IAM cuando una aplicación intenta realizar dicha acción en un recurso protegido. IAM devolverá la opción de permitir o denegar en función de los mecanismos estándar de evaluación de las políticas de IAM.

## Contenido

- [La protección de datos en la modernización del mainframe AWS](#)
- [Identity and Access Management para la modernización del AWS mainframe](#)
- [Validación de la conformidad para la modernización del mainframe AWS](#)
- [Resiliencia en la modernización de los AWS mainframes](#)
- [Seguridad de la infraestructura en AWS Mainframe Modernization](#)
- [Acceda AWS Mainframe Modernization mediante un punto final de interfaz \(AWS PrivateLink\)](#)

## La protección de datos en la modernización del mainframe AWS

El [modelo de](#) se aplica a protección de datos en la modernización del AWS mainframe. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global en la que se ejecutan todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog sobre el [Modelo de responsabilidad compartida de AWS y GDPR](#) en el Blog de seguridad de AWS .

Con fines de protección de datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactor (MFA) en cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos. AWS Se recomienda el uso de TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios con. AWS CloudTrail
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-2 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información

sobre los puntos de conexión de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS Mainframe Modernization u otro tipo de Servicios de AWS uso de la consola, la API o los SDK. AWS CLI AWS Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

## Datos que recopila AWS Mainframe Modernization

AWS La modernización del mainframe recopila varios tipos de datos sobre usted:

- `Application configuration`: Se trata de un archivo JSON que se crea para configurar la aplicación. Contiene sus opciones para las diferentes opciones que ofrece AWS Mainframe Modernization. El archivo también contiene información sobre los AWS recursos dependientes, como las rutas de Amazon Simple Storage Service donde se almacenan los artefactos de la aplicación o el nombre de recurso de Amazon (ARN) AWS Secrets Manager donde se almacenan las credenciales de la base de datos.
- `Application executable (binary)`: Se trata de un binario que usted compila y que pretende implementar en la modernización del AWS mainframe.
- `Application JCL or scripts`: Este código fuente gestiona los trabajos por lotes u otro tipo de procesamiento en nombre de su aplicación.
- `User application data`: Al importar conjuntos de datos, AWS Mainframe Modernization los almacena en la base de datos relacional para que la aplicación pueda acceder a ellos.
- `Application source code`: A través de Amazon AppStream 2.0, AWS Mainframe Modernization proporciona un entorno de desarrollo para escribir y compilar código.

AWS La modernización del mainframe almacena estos datos de forma nativa en. AWS Los datos que recopilamos sobre usted se almacenan en un bucket de Amazon S3 gestionado por AWS Mainframe Modernization. Al implementar una aplicación, AWS Mainframe Modernization descarga los datos en una instancia de Amazon Elastic Compute Cloud respaldada por Amazon Elastic Block Store.

Cuando se activa la limpieza, los datos se eliminan del volumen de Amazon EBS y de Amazon S3. Los volúmenes de Amazon EBS son de un solo inquilino, lo que significa que se utiliza una instancia para un cliente. Las instancias nunca se comparten. Al eliminar un entorno de tiempo de ejecución, también se elimina el volumen de Amazon EBS. Al eliminar una aplicación, los artefactos y la configuración se eliminan de Amazon S3.

Los registros de las aplicaciones se almacenan en Amazon CloudWatch. Los mensajes de registro de las aplicaciones del cliente también CloudWatch se exportan a. Los CloudWatch registros pueden contener datos confidenciales del cliente, como datos empresariales o información de seguridad (en los mensajes de depuración). Para obtener más información, consulte [Supervisión de la modernización AWS del mainframe con Amazon CloudWatch](#).

Además, si decide adjuntar uno o más sistemas de archivos de Amazon Elastic File System o Amazon FSx a su entorno de tiempo de ejecución, los datos de esos sistemas se almacenarán en AWS. Tendrá que limpiar esos datos si decide dejar de usar los sistemas de archivos.

Puede utilizar todas las opciones de cifrado de Amazon S3 disponibles para proteger sus datos al colocarlos en el depósito de Amazon S3 que AWS Mainframe Modernization utiliza para la implementación de aplicaciones y la importación de conjuntos de datos. Además, puede utilizar las opciones de cifrado de Amazon EFS y Amazon FSx si adjunta uno o más de estos sistemas de archivos a su entorno de tiempo de ejecución.

## Cifrado de datos en reposo para el servicio de AWS modernización de mainframe

AWS La modernización del mainframe se integra AWS Key Management Service para proporcionar un cifrado transparente del lado del servidor (SSE) en todos los recursos dependientes que almacenan datos de forma permanente, a saber, Amazon Simple Storage Service, Amazon DynamoDB y Amazon Elastic Block Store. AWS Mainframe Modernization crea y administra claves de cifrado AWS KMS simétricas para usted. AWS KMS

El cifrado de los datos en reposo de forma predeterminada ayuda a reducir la sobrecarga operativa y la complejidad que implica la protección de los datos confidenciales. Al mismo tiempo, le permite migrar aplicaciones seguras que requieren estrictos requisitos normativos y de conformidad del cifrado.

No puede deshabilitar esta capa de cifrado ni seleccionar un tipo de cifrado alternativo al crear aplicaciones y entornos de ejecución.

Puede utilizar su propia clave gestionada por el cliente para las aplicaciones de modernización de AWS mainframe y los entornos de ejecución para cifrar los recursos de Amazon S3 y Amazon EBS.

Para sus aplicaciones de modernización de AWS mainframe, puede utilizar esta clave para cifrar la definición de la aplicación, así como otros recursos de la aplicación, como los archivos JCL, que se guardan en el depósito de Amazon S3 que se crea en la cuenta del servicio. Para obtener más información, consulte [Crear una aplicación](#).

Para sus entornos de ejecución de AWS Mainframe Modernization, AWS Mainframe Modernization utiliza la clave administrada por el cliente para cifrar el volumen de Amazon EBS que crea y adjunta a su instancia Amazon EC2 de Mainframe AWS Modernization, que también se encuentra en la cuenta del servicio. Para obtener más información, consulte [Creación de un entorno de tiempo de ejecución](#).

#### Note

Los recursos de DynamoDB siempre se cifran mediante Clave administrada de AWS una cuenta de servicio de modernización AWS de mainframe. No se pueden cifrar los recursos de DynamoDB con una clave administrada por el cliente.

AWS Mainframe Modernization utiliza la clave gestionada por el cliente para las siguientes tareas:

- Reimplementación de una aplicación.
- Sustitución de una AWS instancia Amazon EC2 de modernización de mainframe.

AWS Mainframe Modernization no utiliza la clave gestionada por el cliente para cifrar las bases de datos de Amazon Relational Database Service o Amazon Aurora, las colas de Amazon Simple Queue Service y las ElastiCache cachés de Amazon que se crean para dar soporte a una aplicación de modernización de AWS mainframe, ya que ninguna de ellas contiene datos de clientes.


Para más información, consulte las [claves administradas por el cliente](#) en la Guía para desarrolladores de AWS Key Management Service.

En la siguiente tabla se resume cómo Mainframe Modernization cifra los datos confidenciales. AWS

Tipo de datos	Clave administrada de AWS cifrado	Clave de cifrado gestionada por el cliente
Definition	Habilitado	Habilitado



Tipo de datos	Clave administrada de AWS cifrado	Clave de cifrado gestionada por el cliente
<p>Contiene la definición de una aplicación en particular.</p>		
<p><code>EnvironmentSummary</code></p> <p>Contiene información sobre el entorno de tiempo de ejecución.</p>	Habilitado	Habilitado
<p><code>ApplicationSummary</code></p> <p>Contiene información sobre la aplicación de modernización AWS del mainframe.</p>	Habilitado	Habilitado
<p><code>DeploymentSummary</code></p> <p>Contiene información sobre la implementación de una aplicación de modernización de AWS mainframe.</p>	Habilitado	Habilitado

 Note

AWS La modernización del mainframe permite automáticamente el cifrado en reposo Claves administradas por AWS para proteger sus datos confidenciales sin coste alguno. Sin embargo, se aplican AWS KMS cargos por el uso de una clave administrada por el cliente. Para obtener más información sobre los precios, consulte [Precios de AWS Key Management Service](#).

Para obtener más información al respecto AWS KMS, consulte AWS Key Management Service.

# Cómo utiliza AWS Mainframe Modernization las subvenciones en AWS KMS

AWS La modernización del mainframe requiere una [subvención](#) para utilizar la clave gestionada por el cliente.

Cuando crea una aplicación o un entorno de ejecución, o despliega una aplicación en AWS Mainframe Modernization cifrada con una clave gestionada por el cliente, AWS Mainframe Modernization crea una subvención en su nombre enviando una [CreateGrant](#) solicitud a. AWS KMS Las subvenciones AWS KMS se utilizan para permitir que AWS Mainframe Modernization acceda a una clave KMS de la cuenta de un cliente.

AWS La modernización del mainframe requiere la autorización para utilizar la clave gestionada por el cliente en las siguientes operaciones internas:

- Envíe [DescribeKey](#) solicitudes AWS KMS para comprobar que el identificador de clave simétrico gestionado por el cliente introducido al crear una aplicación, un entorno de ejecución o la implementación de una aplicación es válido.
- Envíe solicitudes de [GenerateDataclave](#) AWS KMS a para cifrar el volumen de Amazon EBS adjunto a las instancias de Amazon EC2 que AWS alojan los entornos de ejecución de Mainframe Modernization.
- Envíe solicitudes de [descifrado](#) a AWS KMS para descifrar contenido cifrado en Amazon EBS.

AWS Mainframe AWS KMS Modernization utiliza las subvenciones para descifrar los secretos almacenados en Secrets Manager y al crear un entorno de ejecución, crear o volver a implementar una aplicación y crear una implementación. Las subvenciones que crea AWS Mainframe Modernization respaldan las siguientes operaciones:

- Crear o actualizar una concesión de entorno de tiempo de ejecución:
  - Decrypt
  - Encrypt
  - ReEncryptDe
  - ReEncryptPara
  - GenerateDataClave
  - DescribeKey
  - CreateGrant

- Crear o reimplementar una concesión para una aplicación:
  - GenerateDataClave
- Crear una concesión para una implementación:
  - Decrypt

Puede revocar el acceso a la concesión o eliminar el acceso del servicio a la clave administrada por el cliente en cualquier momento. Si lo hace, AWS Mainframe Modernization no podrá acceder a ninguno de los datos cifrados por la clave administrada por el cliente, lo que afectará a las operaciones que dependen de los datos. Por ejemplo, si AWS Mainframe Modernization intentara acceder a una definición de aplicación cifrada por una clave gestionada por el cliente sin conceder esa clave, la operación de creación de la aplicación fallaría.

AWS La modernización del mainframe recopila las configuraciones de las aplicaciones de los usuarios (archivos JSON) y los artefactos (archivos binarios y ejecutables). También crea metadatos que rastrean varias entidades utilizadas para el funcionamiento de AWS Mainframe Modernization y crea registros y métricas. Los registros y las métricas que los clientes pueden ver incluyen:

- CloudWatch registros que reflejan la aplicación y el motor de ejecución ( AWS Blu Age o Micro Focus).
- CloudWatch métricas para los cuadros de mando de operaciones.

Además, AWS Mainframe Modernization recopila datos y métricas de uso para la medición, los informes de actividad, etc. sobre los servicios. Estos datos no son visibles para los clientes.

AWS La modernización del mainframe almacena estos datos en diferentes lugares según el tipo de datos. Los datos del cliente que cargue se almacenan en un bucket de Amazon S3. Los datos del servicio se almacenan en Amazon S3 y DynamoDB. Al implementar una aplicación, tanto los datos como los datos de servicio se descargan en los volúmenes de Amazon EBS. Si decide adjuntar el almacenamiento de Amazon EFS o Amazon FSx a su entorno de tiempo de ejecución, los datos almacenados en esos sistemas de archivos también se descargan en el volumen de Amazon EBS.

El cifrado en reposo está configurado de forma predeterminada. No puede deshabilitarlo ni cambiarlo. Actualmente tampoco se puede cambiar su configuración.

## Crear una clave administrada por el cliente

Puede crear una clave simétrica gestionada por el cliente mediante las API AWS Management Console o las AWS KMS API.

Para crear una clave simétrica administrada por el cliente

Siga los pasos para [crear una clave simétrica administrada por el cliente](#) que se indican en la AWS Key Management Service Guía para desarrolladores.

### Política de claves

Las políticas de clave controlan el acceso a la clave administrada por el cliente. Cada clave administrada por el cliente debe tener exactamente una política de clave, que contiene instrucciones que determinan quién puede usar la clave y cómo puede utilizarla. Cuando crea la clave administrada por el cliente, puede especificar una política de clave. Para obtener más información, consulte [Administración del acceso a las claves](#) en la Guía para desarrolladores de AWS Key Management Service .

Para utilizar la clave gestionada por el cliente con los recursos de modernización del AWS mainframe, la política de claves debe permitir las siguientes operaciones de API:

- [kms:CreateGrant](#): añade una concesión a una clave administrada por el cliente. Otorga el acceso de control a una clave de KMS específica, que permite acceder a [las operaciones que requiere](#) la modernización AWS del mainframe. Para obtener más información sobre el [Uso de concesiones](#), consulte la Guía para desarrolladores de AWS Key Management Service .

Esto permite que la modernización AWS del mainframe haga lo siguiente:

- Llamar a `GenerateDataKey` para generar una clave de datos cifrada y almacenarla, ya que la clave de datos no se utiliza inmediatamente para cifrar.
- Llamar a `Decrypt` para usar la clave de datos cifrados almacenada para acceder a los datos cifrados.
- Configurar una entidad principal que se retire para permitir que el servicio `RetireGrant`.
- [kms:DescribeKey](#)— Proporciona los detalles clave gestionados por el cliente para permitir que AWS Mainframe Modernization valide la clave.

AWS La modernización del mainframe requiere `kms:CreateGrant` y `kms:DescribeKey` autoriza en la política clave del cliente. AWS Mainframe Modernization utiliza esta política para crear una subvención para sí misma.

```
{
  "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

La función que se muestra `Principal` en el ejemplo anterior es la que se utiliza para las operaciones de modernización AWS del mainframe, como `CreateApplication` y `CreateEnvironment`.

Para obtener más información sobre [cómo especificar permisos en una política](#), consulte la Guía para desarrolladores de AWS Key Management Service .

Para obtener información sobre la [solución de problemas de acceso a las claves](#), consulte la Guía para desarrolladores de AWS Key Management Service .

## Especificar una clave administrada por el cliente para AWS Mainframe Modernization

Puede especificar una clave administrada por el cliente para los siguientes recursos:

- Aplicación
- Entorno

Al crear un recurso, puede especificar la clave introduciendo un ID de KMS, que AWS Mainframe Modernization utiliza para cifrar los datos confidenciales almacenados en el recurso.

- KMS ID: un [identificador de clave](#) para una clave administrada por el cliente. Introduzca el ID de la clave, el ARN de la clave, el nombre de alias o el ARN del alias.

Puede especificar una clave administrada por el cliente mediante el AWS Management Console o el AWS CLI

Para especificar la clave gestionada por el cliente al crear un entorno de ejecución en el AWS Management Console, consulte [Cree un entorno de ejecución de modernización AWS de mainframe](#). Para especificar la clave gestionada por el cliente al crear una aplicación en el AWS Management Console, consulte [Cree una aplicación de modernización de mainframe AWS](#).

Para añadir la clave gestionada por el cliente al crear un entorno de ejecución con la AWS CLI, especifique el `kms-key-id` parámetro de la siguiente manera:

```
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
```

Para añadir la clave gestionada por el cliente al crear una aplicación con AWS CLI, especifique el `kms-key-id` parámetro de la siguiente manera:

```
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 Contexto de cifrado de modernización del mainframe

Un [contexto de cifrado](#) es un conjunto opcional de pares clave-valor que pueden contener información contextual adicional sobre los datos.

AWS KMS utiliza el contexto de cifrado como [datos autenticados adicionales](#) para respaldar el cifrado [autenticado](#). Al incluir un contexto de cifrado en una solicitud de cifrado de datos, AWS KMS vincula el contexto de cifrado a los datos cifrados. Para descifrar los datos, debe incluir el mismo contexto de cifrado en la solicitud.

## AWS Modernización del mainframe: contexto de cifrado

AWS La modernización del mainframe utiliza el mismo contexto de cifrado en todas las operaciones AWS KMS criptográficas relacionadas con una aplicación (crear la aplicación y crear el despliegue), donde está la clave `aws:m2:app` y el valor es el identificador único de la aplicación.

### Example

```
"encryptionContextSubset": {
  "aws:m2:app": "a1bc2defabc3defabc4defabcd"
}
```

### Uso del contexto de cifrado para la supervisión

Si utiliza una clave simétrica administrada por el cliente para cifrar sus aplicaciones o entornos de tiempo de ejecución, también puede utilizar el contexto de cifrado en los registros y registros de auditoría para identificar cómo se está utilizando la clave administrada por el cliente.

Utilizar el contexto de cifrado para controlar el acceso a la clave administrada por el cliente

Puede utilizar el contexto de cifrado en las políticas de claves y las políticas de IAM como `conditions` para controlar el acceso a la clave simétrica administrada por el cliente. Puede usar también una restricción de contexto de cifrado en una concesión.

AWS La modernización del mainframe utiliza una restricción del contexto de cifrado en las subvenciones para controlar el acceso a la clave gestionada por el cliente en su cuenta o región. La restricción de concesión requiere que las operaciones que permite la concesión utilicen el contexto de cifrado especificado. El siguiente ejemplo es una subvención que AWS Mainframe Modernization aprovecha para cifrar los artefactos de una aplicación al crear una aplicación.

```
//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"
    }
  }
}
```

```
}
```

## Supervisión de las claves de cifrado de AWS Mainframe Modernization

Cuando utiliza una clave gestionada por el AWS KMS cliente con sus recursos de modernización de AWS mainframe, puede utilizar [AWS CloudTrailAmazon CloudWatch Logs](#) para realizar un seguimiento de las solicitudes que envía AWS Mainframe Modernization. AWS KMS

### Ejemplos de entornos de tiempo de ejecución

Los siguientes ejemplos son AWS CloudTrail eventos para DescribeKey CreateGrantGenerateDataKey, y para monitorear las operaciones de KMS Decrypt solicitadas por AWS Mainframe Modernization para acceder a los datos cifrados por su clave administrada por el cliente:

#### DescribeKey

AWS Mainframe Modernization utiliza la DescribeKey operación para comprobar si la clave gestionada por el AWS KMS cliente asociada a su entorno de ejecución existe en la cuenta y la región.

El siguiente evento de ejemplo registra la operación DescribeKey:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
```



```

        "creationDate": "2022-12-06T19:40:26Z",
        "mfaAuthenticated": "false"
    }
}
},
"eventTime": "2022-12-06T20:23:43Z",
"eventSource": "kms.amazonaws.com",
"eventName": "DescribeKey",
"awsRegion": "us-west-2",
"sourceIPAddress": "205.251.233.182",
"userAgent": "ExampleDesktop/1.0 (V1; OS)",
"requestParameters": {
    "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management",
"tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}

```

## CreateGrant

Cuando utiliza una clave administrada por el AWS KMS cliente para cifrar su entorno de ejecución, AWS Mainframe Modernization envía varias CreateGrant solicitudes en su nombre para realizar las operaciones de KMS necesarias. Algunas de las subvenciones que crea AWS

Mainframe Modernization se retiran inmediatamente después de su uso. Otras se retiran al eliminar el entorno de tiempo de ejecución.

El siguiente evento de ejemplo registra la operación CreateGrant del rol de ejecución de Lambda asociado al flujo de trabajo de Create Environment.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:11:45Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T20:23:09Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
  "userAgent": "m2.us-west-2.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "operations": [
      "Encrypt",
      "Decrypt",
      "ReEncryptFrom",

```

```

        "ReEncryptTo",
        "GenerateDataKey",
        "GenerateDataKey",
        "DescribeKey",
        "CreateGrant"
    ],
    "granteePrincipal": "m2.us-west-2.amazonaws.com",
    "retiringPrincipal": "m2.us-west-2.amazonaws.com"
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

El siguiente evento de ejemplo registra la operación `CreateGrant` del rol vinculado al servicio del grupo Auto Scaling. El rol de ejecución de Lambda asociado al flujo de trabajo de `Create Environment` llama a esta operación `CreateGrant`. Concede permiso para que el rol de ejecución cree una subconcesión para el rol vinculado a servicios del grupo Auto Scaling.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "ARO3YPCLM65MZFUPM4J0:EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",

```

```

    "arn": "arn:aws:sts::111122223333:assumed-role/EnvironmentWorkflow-
alpha-CreateEnvironmentLambdaS-1AU4A8VNQEEKN/EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN",
        "accountId": "111122223333",
        "userName": "EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:22:28Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-12-06T20:23:09Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "54.148.236.160",
  "userAgent": "aws-sdk-java/2.18.21 Linux/4.14.255-276-224.499.amzn2.x86_64
OpenJDK_64-Bit_Server_VM/11.0.14.1+10-LTS Java/11.0.14.1 vendor/Amazon.com_Inc. md/
internal exec-env/AWS_Lambda_java11 io/sync http/Apache cfg/retry-mode/legacy",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "operations": [
      "Encrypt",
      "Decrypt",
      "ReEncryptFrom",
      "ReEncryptTo",
      "GenerateDataKey",
      "GenerateDataKey",
      "DescribeKey",
      "CreateGrant"
    ],
    "granteePrincipal": "m2.us-west-2.amazonaws.com",

```

```

    "retiringPrincipal": "m2.us-west-2.amazonaws.com"
  },
  "responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  }
}
}

```

## GenerateDataKey

Cuando habilita una clave administrada por el AWS KMS cliente para su recurso de entorno de ejecución, Auto Scaling crea una clave única para cifrar el volumen de Amazon EBS asociado al entorno de ejecución. Envía una GenerateDataKey solicitud a la AWS KMS que se especifica la clave administrada por el AWS KMS cliente para el recurso.

El siguiente evento de ejemplo registra la operación GenerateDataKey:

```

{
  "eventVersion": "1.08",
  "userIdentity": {

```

```

    "type": "AssumedRole",
    "principalId": "AROA3YPCLM65EEXVIEH7D:AutoScaling",
    "arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForAutoScaling/
AutoScaling",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling",
        "accountId": "111122223333",
        "userName": "AWSServiceRoleForAutoScaling"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:23:16Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "autoscaling.amazonaws.com"
  },
  "eventTime": "2022-12-06T20:23:18Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "autoscaling.amazonaws.com",
  "userAgent": "autoscaling.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:ebs:id": "vol-080f7a32d290807f3"
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "numberOfBytes": 64
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",

```

```

        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

## Decrypt

Cuando accede a un entorno de tiempo de ejecución cifrado, Amazon EBS llama a la operación Decrypt que utiliza la clave de datos cifrados almacenada para acceder a los datos cifrados.

El siguiente evento de ejemplo registra la operación Decrypt:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "ebs.amazonaws.com"
  },
  "eventTime": "2022-12-06T20:23:22Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "ebs.amazonaws.com",
  "userAgent": "ebs.amazonaws.com",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "encryptionContext": {
      "aws:ebs:id": "vol-080f7a32d290807f3"
    }
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",

```

```

      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventCategory": "Management"
}

```

## Ejemplos de aplicaciones

Los siguientes ejemplos son AWS CloudTrail eventos que AWS Mainframe GenerateDataKey Modernization solicita CreateGrant y supervisan las operaciones de KMS para acceder a los datos cifrados por la clave gestionada por el cliente:

### CreateGrant

Cuando utiliza una clave gestionada por el AWS KMS cliente para cifrar los recursos de la aplicación, la función de ejecución de Lambda envía CreateGrant una solicitud en su nombre para acceder a la clave KMS de su cuenta. AWS La concesión permite el rol de ejecución de Lambda cargar los recursos de las aplicaciones del cliente en Amazon S3 mediante la clave administrada por el cliente. Esta concesión se retira inmediatamente después de crear la aplicación.

El siguiente evento de ejemplo registra la operación CreateGrant:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",

```



```

        "accountId": "111122223333",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "creationDate": "2022-12-06T21:51:45Z",
        "mfaAuthenticated": "false"
    }
},
"invokedBy": "m2.us-west-2.amazonaws.com"
},
"eventTime": "2022-12-06T22:47:04Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "m2.us-west-2.amazonaws.com",
"userAgent": "m2.us-west-2.amazonaws.com",
"requestParameters": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "constraints": {
        "encryptionContextSubset": {
            "aws:m2:app": "a1bc2defabc3defabc4defabcd"
        }
    }
},
"retiringPrincipal": "m2.us-west-2.amazonaws.com",
"operations": [
    "GenerateDataKey"
],
"granteePrincipal": "m2.us-west-2.amazonaws.com"
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",

```

```

      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

## GenerateDataKey

Cuando habilita una clave gestionada por el AWS KMS cliente para el recurso de su aplicación, la función de ejecución de Lambda crea una clave que utiliza para cifrar y cargar los datos de los clientes en Amazon Simple Storage Service. La función de ejecución de Lambda envía una `GenerateDataKey` solicitud a la AWS KMS que se especifica la clave gestionada por el AWS KMS cliente para el recurso.

El siguiente evento de ejemplo registra la operación `GenerateDataKey`:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0A3YPCLM65CLCEKKC7Z:ApplicationWorkflow-alpha-
CreateApplicationVersion-CstWZUn5R4u6",
    "arn": "arn:aws:sts::111122223333:assumed-role/ApplicationWorkflow-
alpha-CreateApplicationVersion-1IZRBZYDG20B/ApplicationWorkflow-alpha-
CreateApplicationVersion-CstWZUn5R4u6",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AR0AIGDTESTANDEXAMPLE: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"
}
```

## Ejemplos de implementaciones

Los siguientes ejemplos son AWS CloudTrail eventos que AWS Mainframe Decrypt Modernization solicita CreateGrant y supervisan las operaciones de KMS para acceder a los datos cifrados por la clave gestionada por el cliente:

### CreateGrant

Cuando utiliza una clave gestionada por el AWS KMS cliente para cifrar los recursos de implementación, AWS Mainframe Modernization envía dos CreateGrant solicitudes en su nombre. La primera concesión se destina a la función de ejecución de Lambda a la que se deba llamar ListBatchJobScriptFiles y se retira inmediatamente después de que finalice la implementación. La segunda concesión se destina al rol de instancia limitada de Amazon EC2 para que Amazon EC2 pueda descargar los recursos de las aplicaciones de los clientes desde Amazon S3. Esta concesión se retira cuando la aplicación se elimina del entorno de tiempo de ejecución.

El siguiente evento de ejemplo registra la operación CreateGrant:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T21:51:45Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "invokedBy": "m2.us-west-2.amazonaws.com"
```

```

},
"eventTime": "2022-12-06T23:40:07Z",
"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

Cuando se accede a una implementación, Amazon EC2 solicita a la operación Decrypt que utilice la clave de datos cifrados almacenada para descifrar y descargar los datos cifrados de los clientes de Amazon S3.

El siguiente evento de ejemplo registra la operación Decrypt:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAZYPCLM65BSPZ37E6G:m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "arn": "arn:aws:sts::111122223333:assumed-role/SupernovaEnvironmentInstanceScopeDownRole/m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/SupernovaEnvironmentInstanceScopeDownRole",
        "accountId": "111122223333",
        "userName": "SupernovaEnvironmentInstanceScopeDownRole"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T23:19:29Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "invokedBy": "m2.us-west-2.amazonaws.com",
  "eventTime": "2022-12-06T23:40:15Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
  "userAgent": "m2.us-west-2.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:m2:app": "a1bc2defabc3defabc4defabcdn",

```

```

    "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"
}

```

## Más información

Los siguientes recursos proporcionan más información sobre cifrado de datos en reposo.

- Para obtener más información acerca de [conceptos básicos de AWS Key Management Service](#), consulte la Guía para desarrolladores de AWS Key Management Service .
- Para obtener más información sobre las [prácticas recomendadas de seguridad de AWS Key Management Service](#), consulte la Guía para desarrolladores de AWS Key Management Service .

## Cifrado en tránsito

En el caso de las aplicaciones interactivas que forman parte de cargas de trabajo transaccionales, los intercambios de datos entre el emulador de terminal y el punto final del servicio de modernización del AWS mainframe para el protocolo TN3270 no se cifran durante el tránsito. Si la aplicación requiere el cifrado en tránsito, es posible que desee implementar algunos mecanismos de tunelización adicionales.

AWS La modernización del mainframe utiliza HTTPS para cifrar las API del servicio. Todas las demás comunicaciones dentro de la modernización del AWS mainframe están protegidas por la VPC del servicio o el grupo de seguridad, así como por HTTPS. AWS La modernización del mainframe transfiere los artefactos, las configuraciones y los datos de las aplicaciones. Los artefactos de la aplicación se copian de un bucket de Amazon S3 del que es propietario, al igual que los datos de la aplicación. Puede proporcionar las configuraciones de las aplicaciones mediante un enlace a Amazon S3 o cargando un archivo localmente.

El cifrado básico en tránsito está configurado de forma predeterminada, pero no se aplica al protocolo TN3270. AWS La modernización del mainframe utiliza HTTPS para los puntos finales de la API, que también están configurados de forma predeterminada.

## Identity and Access Management para la modernización del AWS mainframe

AWS Identity and Access Management (IAM) es una herramienta Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a los recursos. AWS Los administradores de IAM controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar los recursos de modernización de AWS mainframe. La IAM es una herramienta Servicio de AWS que puede utilizar sin coste adicional.

### Temas

- [Público](#)
- [Autenticación con identidades](#)
- [Administración de acceso mediante políticas](#)
- [Cómo funciona la modernización AWS del mainframe con la IAM](#)
- [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)
- [Solución de problemas de AWS identidad y acceso a la modernización del mainframe](#)
- [Uso de roles vinculados a servicios para Mainframe Modernization](#)

### Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo que se realice en la modernización del AWS mainframe.



Usuario del servicio: si utiliza el servicio de modernización del AWS mainframe para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más funciones de modernización de AWS mainframe para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una característica de AWS Mainframe Modernization, consulte [Solución de problemas de AWS identidad y acceso a la modernización del mainframe](#).

Administrador de servicios: si está a cargo de los recursos de modernización del AWS mainframe en su empresa, probablemente tenga pleno acceso a la modernización del AWS mainframe. Su tarea es determinar a qué funciones y recursos de modernización AWS del mainframe deben acceder los usuarios del servicio. Luego, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar la IAM con la modernización del AWS mainframe, consulte. [Cómo funciona la modernización AWS del mainframe con la IAM](#)

Administrador de IAM: si es administrador de IAM, puede que desee obtener información detallada sobre cómo redactar políticas para administrar el acceso a la modernización de mainframe. AWS Para ver ejemplos de políticas de modernización de AWS mainframes basadas en la identidad que puede utilizar en la IAM, consulte. [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)

## Autenticación con identidades

La autenticación es la forma de iniciar sesión con sus AWS credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como usuario de IAM o asumiendo una función de IAM. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (Centro de identidades de IAM), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, su administrador habrá configurado previamente la federación de identidades mediante roles de IAM. Cuando accedes AWS mediante la federación, estás asumiendo un rol de forma indirecta.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte [Cómo iniciar sesión Cuenta de AWS en su](#) Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS proporciona un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte [Firmar las solicitudes de la AWS API](#) en la Guía del usuario de IAM.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Autenticación multifactor](#) en la Guía del usuario de AWS IAM Identity Center y [Uso de la autenticación multifactor \(MFA\) en AWS](#) en la Guía del usuario de IAM.

## Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de las tareas que requieren que inicie sesión como usuario raíz, consulte [Tareas que requieren credenciales de usuario raíz](#) en la Guía del usuario de IAM.

## Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios de su empresa, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en el Centro de identidades de IAM, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en

todas sus Cuentas de AWS aplicaciones. Para obtener más información, consulte [¿Qué es el Centro de identidades de IAM?](#) en la Guía del usuario de AWS IAM Identity Center .

## Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos emplear credenciales temporales, en lugar de crear usuarios de IAM que tengan credenciales de larga duración como contraseñas y claves de acceso. No obstante, si tiene casos de uso específicos que requieran credenciales de larga duración con usuarios de IAM, recomendamos rotar las claves de acceso. Para más información, consulte [Rotar las claves de acceso periódicamente para casos de uso que requieran credenciales de larga duración](#) en la Guía del usuario de IAM.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

## Roles de IAM

Un [rol de IAM](#) es una identidad dentro de usted Cuenta de AWS que tiene permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente una función de IAM en el AWS Management Console [cambiando](#) de función. Puede asumir un rol llamando a una operación de AWS API AWS CLI o utilizando una URL personalizada. Para más información sobre los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

Los roles de IAM con credenciales temporales son útiles en las siguientes situaciones:

- **Acceso de usuario federado:** para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información acerca de roles para

federación, consulte [Creación de un rol para un proveedor de identidades de terceros](#) en la Guía del usuario de IAM. Si utiliza IAM Identity Center, debe configurar un conjunto de permisos. IAM Identity Center correlaciona el conjunto de permisos con un rol en IAM para controlar a qué pueden acceder las identidades después de autenticarse. Para obtener información acerca de los conjuntos de permisos, consulte [Conjuntos de permisos](#) en la Guía del usuario de AWS IAM Identity Center .

- **Permisos de usuario de IAM temporales:** un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- **Acceso entre cuentas:** puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunas Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.
- **Acceso entre servicios:** algunos Servicios de AWS utilizan funciones en otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
- **Sesiones de acceso directo (FAS):** cuando utilizas un usuario o un rol de IAM para realizar acciones en ellas AWS, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. El FAS utiliza los permisos del principal que llama Servicio de AWS y los solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).
- **Rol de servicio:** un rol de servicio es un [rol de IAM](#) que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.
- **Función vinculada al servicio:** una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar

una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

- Aplicaciones que se ejecutan en Amazon EC2: puede usar un rol de IAM para administrar las credenciales temporales de las aplicaciones que se ejecutan en una instancia EC2 y realizan AWS CLI solicitudes a la API. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar una AWS función a una instancia EC2 y ponerla a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en lugar de un usuario\)](#) en la Guía del usuario de IAM.

## Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte [Información general de políticas JSON](#) en la Guía del usuario de IAM.

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Las políticas de IAM definen permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de la API AWS Management Console AWS CLI, la o la AWS API.

## Políticas basadas en identidades

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y roles de su Cuenta de AWS empresa. Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para más información sobre cómo elegir una política administrada o una política insertada, consulte [Elegir entre políticas administradas y políticas insertadas](#) en la Guía del usuario de IAM.

## Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

## Listas de control de acceso (ACL)

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

Amazon S3 y Amazon VPC son ejemplos de servicios que admiten las ACL. AWS WAF Para obtener más información sobre las ACL, consulte [Información general de Lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

## Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- **Límites de permisos:** un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- **Políticas de control de servicios (SCP):** las SCP son políticas de JSON que especifican los permisos máximos para una organización o unidad organizativa (OU). AWS Organizations es un servicio para agrupar y gestionar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o a todas sus cuentas. El SCP limita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una. Usuario raíz de la cuenta de AWS Para obtener más información acerca de Organizations y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations .
- **Políticas de sesión:** las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

## Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo AWS determinar si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la [lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

## Cómo funciona la modernización AWS del mainframe con la IAM

Antes de usar la IAM para administrar el acceso a la modernización del AWS mainframe, averigüe qué funciones de IAM están disponibles para usar con la modernización del mainframe. AWS

Funciones de IAM que puede utilizar con la modernización de mainframes AWS

Característica de IAM	AWS Soporte para la modernización de mainframes
<a href="#">Políticas basadas en identidades</a>	Sí
<a href="#">Políticas basadas en recursos</a>	No
<a href="#">Acciones de políticas</a>	Sí
<a href="#">Recursos de políticas</a>	Sí
<a href="#">Claves de condición de política</a>	Sí
<a href="#">ACL</a>	No
<a href="#">ABAC (etiquetas en políticas)</a>	Sí
<a href="#">Credenciales temporales</a>	Sí
<a href="#">Sesiones de acceso directo (FAS)</a>	Sí
<a href="#">Roles de servicio</a>	Sí
<a href="#">Roles vinculados al servicio</a>	Sí

Para obtener una visión general de cómo funcionan la modernización del AWS mainframe y otros AWS servicios con la mayoría de las funciones de IAM, consulte los [AWS servicios que funcionan con IAM en la Guía del usuario de IAM](#).



## Políticas basadas en la identidad para la modernización del mainframe AWS

Compatibilidad con las políticas basadas en identidad      Sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en identidad, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para más información sobre los elementos que puede utilizar en una política de JSON, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la Guía del usuario de IAM.

### Ejemplos de políticas basadas en la identidad para la modernización del mainframe AWS

Para ver ejemplos de políticas de modernización de AWS mainframes basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)

## Políticas basadas en recursos dentro de la modernización del mainframe AWS

Compatibilidad con las políticas basadas en recursos      No

Las políticas basadas en recursos son documentos de política JSON que se asocian a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el administrador de IAM de la cuenta de confianza también debe conceder a la entidad principal (usuario o rol) permiso para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para más información, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

## Acciones políticas para la modernización del AWS mainframe

Admite acciones de política

Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Las acciones políticas suelen tener el mismo nombre que la operación de AWS API asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de las acciones de modernización del AWS mainframe, consulte las [acciones definidas por la modernización del AWS mainframe](#) en la Referencia de autorización de servicios.

Las acciones políticas de modernización de AWS mainframes utilizan el siguiente prefijo antes de la acción:

```
m2
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

```
"Action": [
```

```
"m2:StartApplication",  
"m2:StopApplication"  
]
```

Puede utilizar caracteres comodín (\*) para especificar varias acciones. Por ejemplo, para especificar todas las acciones que comiencen con la palabra List, incluya la siguiente acción:

```
"Action": "m2:List*"
```

Para ver ejemplos de políticas de modernización de AWS mainframes basadas en la identidad, consulte [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)

## Recursos de políticas para la modernización de mainframes AWS

Admite recursos de políticas

Sí

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento Resource de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento Resource o NotResource. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (\*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Puede restringir el acceso a recursos específicos de modernización AWS del mainframe utilizando sus ARN para identificar el recurso al que se aplica la política de IAM. Para obtener más información acerca del formato de los ARN, consulte [Nombres de recursos de Amazon \(ARN\)](#) en la Referencia general de AWS.

Por ejemplo, un entorno de modernización de AWS mainframe tiene el siguiente ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:env/service-generated-unique-identifier"
```

Una aplicación de modernización de AWS mainframe tiene el siguiente ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:app/service-generated-unique-identifier"
```

No todas las acciones de modernización AWS del mainframe admiten permisos a nivel de recursos. Para las acciones que no admiten permisos de recursos, debe utilizar el comodín (\*).

Las siguientes acciones de modernización del AWS mainframe no admiten permisos a nivel de recursos.

```
ListApplications
    ListApplicationVersions
    ListBatchJobDefinitions
    ListBatchJobExecutions
    ListDataSetImportHistory
    ListDataSets
    ListDeployments
    ListEngineVersions
    ListEnvironments
    ListTagsForResource
```

Para ver una lista de los tipos de recursos de modernización del AWS mainframe y sus ARN, consulte los [recursos definidos por la modernización del AWS mainframe](#) en la referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte [Acciones definidas por la modernización del AWS mainframe](#).

Para ver ejemplos de políticas de modernización de AWS mainframes basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)

## AWS Permisos de la API de modernización de mainframes: referencia sobre acciones, recursos y condiciones

Puede usar la siguiente tabla como referencia cuando escriba políticas de permisos que vaya a asociar a una identidad de IAM (políticas basadas en identidades). La tabla incluye lo siguiente:

- Cada operación de la API de modernización AWS de mainframe
- Las acciones correspondientes para las que puede conceder permisos para realizar la acción

- El AWS recurso para el que puede conceder los permisos

Las acciones se especifican en el campo `Action` de la política y el valor del recurso se especifica en el campo `Resource` de la política.

Puede utilizar claves de condición AWS globales en sus políticas de modernización de AWS mainframe para expresar las condiciones. Para obtener una lista completa de AWS las claves, consulte las [claves de condición globales disponibles](#) en la Guía del usuario de IAM.

#### Note

Para especificar una acción, use el prefijo `m2:` seguido del nombre de operación de la API (por ejemplo, `m2:CreateApplication`).

AWS API de modernización de mainframe y permisos necesarios para realizar acciones

AWS Operaciones de la API de modernización de mainframes	Permisos necesarios (acciones de la API)	Recursos
<a href="#">CancelBatchJobExecution</a>		Aplicación
<a href="#">CreateApplication</a>	<code>s3:GetObject</code> <code>s3:ListBucket</code>	Aplicación
<a href="#">CreateDataSetImportTask</a>	<code>m2:CreateDataSetImportTask</code> <code>s3:GetObject</code>	Aplicación
<a href="#">CreateDeployment</a>	<code>elasticloadbalancing:AddTags</code> <code>elasticloadbalancing:CreateListener</code> <code>elasticloadbalancing:CreateTargetGroup</code>	Aplicación

AWS Operaciones de la API de modernización de mainframes	Permisos necesarios (acciones de la API)	Recursos
	elasticloadbalancing:RegisterTargets	
<a href="#">CreateEnvironment</a>	ec2:CreateNetworkInterface ec2:CreateNetworkInterfacePermission ec2:DescribeNetworkInterfaces ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:DescribeVpcAttribute ec2:DescribeVpcs ec2:ModifyNetworkInterfaceAttribute elasticfilesystem:DescribeMountTargets elasticloadbalancing:AddTags elasticloadbalancing:CreateLoadBalancer fsx:DescribeFileSystems iam:CreateServiceLinkedRole	Entorno

AWS Operaciones de la API de modernización de mainframes	Permisos necesarios (acciones de la API)	Recursos
<a href="#">DeleteApplication</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup  logs:DeleteLogDelivery	Aplicación
<a href="#">DeleteApplicationFromEnvironment</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup	Aplicación  Entorno
<a href="#">DeleteEnvironment</a>	elasticloadbalancing:DeleteLoadBalancer	Entorno
<a href="#">GetApplication</a>		Aplicación
<a href="#">GetApplicationVersión</a>		Aplicación
<a href="#">GetBatchJobExecution</a>		Aplicación
<a href="#">GetDataSetDetails</a>		Aplicación
<a href="#">GetDataSetImportTask</a>		Aplicación
<a href="#">GetDeployment</a>		Aplicación
<a href="#">GetEnvironment</a>		Entorno
<a href="#">ListApplications</a>		*
<a href="#">ListApplicationVersiones</a>		*

AWS Operaciones de la API de modernización de mainframes	Permisos necesarios (acciones de la API)	Recursos
<a href="#">ListBatchJobDefinitions</a>		*
<a href="#">ListBatchJobExecutions</a>		*
<a href="#">ListDataSetImportHistory</a>		*
<a href="#">ListDataConjuntos</a>		*
<a href="#">ListDeployments</a>		*
<a href="#">ListEngineVersiones</a>		*
<a href="#">ListEnvironments</a>		*
<a href="#">ListTagsForResource</a>		*
<a href="#">StartApplication</a>		Aplicación
<a href="#">StartBatchJob</a>		Aplicación
<a href="#">StopApplication</a>		Aplicación
<a href="#">TagResource</a>		*
<a href="#">UntagResource</a>		*
<a href="#">UpdateApplication</a>	s3:GetObject s3:ListBucket	Aplicación
<a href="#">UpdateEnvironment</a>		Entorno



## Condiciones clave de la política para la modernización del AWS mainframe

Admite claves de condición de políticas específicas del servicio	Sí
--	----

Los administradores pueden usar las políticas de AWS JSON para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una OR operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de [contexto de condición AWS globales en la Guía](#) del usuario de IAM.

Las siguientes claves de condición son específicas de la modernización del AWS mainframe

```
m2:EngineType
m2:InstanceType
```

Para ver una lista de las claves de condición de la modernización del AWS mainframe, consulte las claves de [condición para la modernización del AWS mainframe](#) en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte [Acciones definidas por la modernización del AWS mainframe](#).

Para ver ejemplos de políticas de modernización de AWS mainframes basadas en la identidad, consulte. [Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS](#)

## Listas de control de acceso (ACL) en AWS Mainframe Modernization

Admite las ACL

No

Las listas de control de acceso (ACL) controlan qué entidades principales (miembros de cuentas, usuarios o roles) tienen permisos para acceder a un recurso. Las ACL son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

## Control de acceso basado en atributos (ABAC) con modernización del mainframe AWS

Admite ABAC (etiquetas en las políticas)

Sí

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a las entidades de IAM (usuarios o roles) y a muchos AWS recursos. El etiquetado de entidades y recursos es el primer paso de ABAC. A continuación, designa las políticas de ABAC para permitir operaciones cuando la etiqueta de la entidad principal coincida con la etiqueta del recurso al que se intenta acceder.

ABAC es útil en entornos que crecen con rapidez y ayuda en situaciones en las que la administración de las políticas resulta engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información sobre ABAC, consulte [¿Qué es ABAC?](#) en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte [Uso del control de acceso basado en atributos \(ABAC\)](#) en la Guía del usuario de IAM.

## Uso de credenciales temporales con la modernización del AWS mainframe

Compatible con el uso de credenciales temporales	Sí
--	----

Algunos Servicios de AWS no funcionan cuando se inicia sesión con credenciales temporales. Para obtener información adicional, incluidas las que Servicios de AWS funcionan con credenciales temporales, consulta Cómo [Servicios de AWS funcionan con IAM](#) en la Guía del usuario de IAM.

Utiliza credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para más información sobre el cambio de roles, consulte [Cambio a un rol \(consola\)](#) en la Guía del usuario de IAM.

Puedes crear credenciales temporales manualmente mediante la AWS CLI API o. AWS A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para más información, consulte [Credenciales de seguridad temporales en IAM](#).

## Sesiones de acceso directo para la modernización del AWS mainframe

Admite Forward access sessions (FAS)	Sí
--------------------------------------	----

Cuando utiliza un usuario o un rol de IAM para realizar acciones en él AWS, se le considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FAS utiliza los permisos del principal que llama y los que solicita Servicio de AWS para realizar solicitudes a los servicios descendentes. Servicio de AWS Las solicitudes de FAS solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros Servicios de AWS recursos para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información sobre las políticas a la hora de realizar solicitudes de FAS, consulte [Reenviar sesiones de acceso](#).

**⚠ Important**

Estos tokens permiten a AWS Mainframe Modernization acceder a los datos de los clientes sin su consentimiento explícito; por ejemplo, AWS Mainframe Modernization despliega artefactos de aplicaciones con datos empresariales asociados desde un bucket de Amazon S3 sin obtener el permiso explícito del cliente. Es posible que necesite actualizar la documentación de cumplimiento en consecuencia.

## Roles de servicio para AWS Mainframe Modernization

Compatible con roles de servicio

Sí

Un rol de servicio es un [rol de IAM](#) que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

AWS La modernización del mainframe respalda las funciones de servicio para los grupos de actividad (transacciones, tareas canceladas o finalizadas, etc.).

**⚠ Warning**

Cambiar los permisos de un rol de servicio podría interrumpir la funcionalidad de modernización del AWS mainframe. Edite las funciones de servicio solo cuando AWS Mainframe Modernization proporcione instrucciones para hacerlo.

## Elegir una función de IAM en AWS la modernización del mainframe

Si ha creado anteriormente una función de IAM que puedan asumir sus aplicaciones que se ejecutan en Amazon EC2, puede elegir esta función al crear una plantilla de lanzamiento o una configuración de lanzamiento. AWS La modernización del mainframe le proporciona una lista de funciones entre las que puede elegir. Al crear estos roles, es importante asociar políticas de menos privilegios de IAM que restrinjan el acceso a las llamadas a la API específicas que requiere la aplicación. Para obtener más información, consulte el tema [Rol de IAM para las aplicaciones que se ejecutan en instancias Amazon EC2](#) en la guía del usuario de Amazon EC2 Auto Scaling.

## Funciones vinculadas al servicio para la modernización de mainframes AWS

Compatible con roles vinculados al servicio	Sí
---	----

Un rol vinculado al servicio es un tipo de rol de servicio que está vinculado a un servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en su Cuenta de AWS y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Para obtener más información sobre la creación o administración de funciones vinculadas al servicio de modernización de AWS mainframe, consulte [Uso de roles vinculados a servicios para Mainframe Modernization](#)

Para más información sobre cómo crear o administrar roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#). Busque un servicio en la tabla que incluya Yes en la columna Rol vinculado a un servicio. Seleccione el vínculo Sí para ver la documentación acerca del rol vinculado a servicios para ese servicio.

## Ejemplos de políticas basadas en la identidad para la modernización de mainframes AWS

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar los recursos de modernización del AWS mainframe. Tampoco pueden realizar tareas mediante la AWS Management Console, AWS Command Line Interface (AWS CLI) o la AWS API. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan. A continuación, el administrador puede añadir las políticas de IAM a roles y los usuarios pueden asumirlos.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte [Creación de políticas de IAM](#) en la Guía del usuario de IAM.

Para obtener más información sobre las acciones y los tipos de recursos definidos por la modernización del AWS mainframe, incluido el formato de los ARN para cada uno de los tipos de recursos, consulte [las acciones, los recursos y las claves de condición para la modernización del AWS mainframe](#) en la Referencia de autorización de servicios.

### Temas

- [Prácticas recomendadas sobre las políticas](#)
- [Uso de la consola de modernización de mainframe AWS](#)
- [Cómo permitir a los usuarios consultar sus propios permisos](#)

## Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear, acceder o eliminar los recursos de modernización del AWS mainframe de su cuenta. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están disponibles en su Cuenta de AWS. Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Con el fin de obtener más información, consulte las [políticas administradas por AWS](#) o las [políticas administradas por AWS para funciones de trabajo](#) en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte [Políticas y permisos en IAM](#) en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la Guía del usuario de IAM.
- Utilice el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para

más información, consulte [Política de validación de Analizador de acceso de IAM](#) en la Guía de usuario de IAM.

- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para solicitar la MFA cuando se invocan las operaciones de la API, agregue las condiciones de la MFA a sus políticas. Para más información, consulte [Configuración del acceso a una API protegido por MFA](#) en la Guía de usuario de IAM.

Para obtener más información sobre las prácticas recomendadas de IAM, consulte las [Prácticas recomendadas de seguridad en IAM](#) en la Guía del usuario de IAM.

## Uso de la consola de modernización de mainframe AWS

Para acceder a la consola de modernización del AWS mainframe, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los recursos de modernización del AWS mainframe que tiene. Cuenta de AWS Si crea una política basada en identidades que sea más restrictiva que el mínimo de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario conceder permisos mínimos de consola a los usuarios que solo realizan llamadas a la API AWS CLI o a la AWS API. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la consola de modernización del AWS mainframe, adjunte también la política ReadOnlY AWS gestionada ConsoleAccess o de modernización del AWS mainframe a las entidades. Para más información, consulte [Adición de permisos a un usuario](#) en la Guía del usuario de IAM:

## Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API o. 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": "*"
}
]
}

```

## Solución de problemas de AWS identidad y acceso a la modernización del mainframe

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con la modernización de AWS mainframes y la IAM.

### Temas

- [No estoy autorizado a realizar la iam: PassRole](#)
- [Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de modernización de AWS mainframe](#)



## No estoy autorizado a realizar la iam: PassRole

Si recibe un mensaje de error que indica que no está autorizado a realizar la `iam:PassRole` acción, sus políticas deben actualizarse para que pueda transferir una función a AWS Mainframe Modernization.

Algunos Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en AWS Mainframe Modernization. Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con su administrador. AWS El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

## Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis recursos de modernización de AWS mainframe

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para más información, consulte lo siguiente:

- Para saber si la modernización de AWS mainframe admite estas funciones, consulte [Cómo funciona la modernización AWS del mainframe con la IAM](#)
- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte [Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM](#).

- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta [Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

## Uso de roles vinculados a servicios para Mainframe Modernization

AWS Mainframe Modernization utiliza funciones AWS Identity and Access Management vinculadas al [servicio](#) (IAM). Un rol vinculado a un servicio es un tipo único de rol de IAM que se encuentra vinculado directamente a Mainframe Modernization. Las funciones vinculadas al servicio están predefinidas por Mainframe Modernization e incluyen todos los permisos que el servicio requiere para llamar a otros servicios en su nombre. AWS

Un rol vinculado a un servicio simplifica la configuración de Mainframe Modernization porque ya no tendrá que agregar manualmente los permisos necesarios. Mainframe Modernization define los permisos de sus roles vinculados a servicios y, a menos que esté definido de otra manera, solo Mainframe Modernization puede asumir sus roles. Los permisos definidos incluyen las políticas de confianza y de permisos, y que la política de permisos no se pueda adjuntar a ninguna otra entidad de IAM.

Solo es posible eliminar un rol vinculado a un servicio después de eliminar sus recursos relacionados. De esta forma, se protegen los recursos de Mainframe Modernization, ya que se evita que se puedan eliminar accidentalmente permisos de acceso a los recursos.

Para obtener información sobre otros servicios que admiten roles vinculados a servicios, consulte [Servicios de AWS que funcionan con IAM](#) y busque los servicios que muestran Yes (Sí) en la columna Roles vinculados a servicios. Elija una opción Sí con un enlace para ver la documentación acerca del rol vinculado al servicio en cuestión.

## Permisos de roles vinculados a servicios de Mainframe Modernization

La modernización del mainframe utiliza la función vinculada al servicio denominada `AWSServiceRoleForAWSM2`: configura la red para que se conecte a tu VPC y acceda a recursos como los sistemas de archivos.

La función `AWSServiceRoleForAWSM2` vinculada al servicio confía en los siguientes servicios para que la asuman:

- `m2.amazonaws.com`

La política de permisos de roles denominada `AWSM2ServicePolicy` permite a Mainframe Modernization realizar las siguientes acciones en los recursos especificados:

- Crear, eliminar, describir y asociar permisos de las interfaces de red de Amazon EC2 para el entorno de Mainframe Modernization a fin de establecer la conectividad con la VPC del cliente.
- Registrar o anular el registro de las entradas de Elastic Load Balancing, que es la forma en que los clientes se conectan al entorno de modernización de la unidad central.
- Describir el sistema de archivos de Amazon EFS o Amazon FSx, si se utiliza.
- Emite métricas al cliente CloudWatch desde el entorno de ejecución.

```
{
  "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"
      ]
    }
  }
}
]
}

```

Debe configurar permisos para permitir a una entidad de IAM (como un usuario, grupo o rol) crear, editar o eliminar un rol vinculado a servicios. Para obtener más información, consulte [Permisos de roles vinculados a servicios](#) en la Guía del usuario de IAM.

## Creación de roles vinculados al servicio para Mainframe Modernization

No necesita crear manualmente un rol vinculado a servicios. Al crear un entorno de ejecución en la AWS Management Console, la API o la AWS API AWS CLI, Mainframe Modernization crea automáticamente la función vinculada al servicio.

Si elimina este rol vinculado a servicios y necesita crearlo de nuevo, puede utilizar el mismo proceso para volver a crear el rol en su cuenta. Cuando crea un entorno de tiempo de ejecución, Mainframe Modernization vuelve a crear el rol vinculado a servicios en su nombre.

## Edición de un rol vinculado al servicio para Mainframe Modernization

La modernización del mainframe no permite editar los dos roles vinculados a los AWSServiceRoleForAWSM servicios. Después de crear un rol vinculado al servicio, no podrá cambiar el nombre del rol, ya que varias entidades podrían hacer referencia al rol. Sin embargo, sí puede editar la descripción del rol con IAM. Para obtener más información, consulte [Editar un rol vinculado a servicios](#) en la Guía del usuario de IAM.

## Eliminación de un rol vinculado a un servicio para Mainframe Modernization

Si ya no necesita usar una característica o servicio que requieran un rol vinculado a un servicio, le recomendamos que elimine dicho rol. Así no tendrá una entidad no utilizada que no se monitoree ni mantenga de forma activa. Sin embargo, debe limpiar los recursos de su rol vinculado al servicio antes de eliminarlo manualmente.

### Note

Si el servicio de Mainframe Modernization está utilizando el rol cuando intenta eliminar los recursos, la eliminación podría producir un error. En tal caso, espere unos minutos e intente de nuevo la operación.

Para eliminar los recursos de modernización de mainframe utilizados por los 2 AWSServiceRoleForAWSM

- Elimine los entornos de tiempo de ejecución en Mainframe Modernization Asegúrese de eliminar las aplicaciones de un entorno antes de eliminar el entorno en sí.

Eliminación manual del rol vinculado a servicios mediante IAM

Utilice la consola de IAM AWS CLI, la o la AWS API para eliminar las AWSServiceRoleForAWSDos funciones vinculadas a los servicios. Para obtener más información, consulte [Eliminación de un rol vinculado a servicios](#) en la Guía del usuario de IAM.

## Regiones admitidas para roles vinculados al servicio de Mainframe Modernization

Mainframe Modernization admite el uso de roles vinculados a servicios en todas las regiones en las que el servicio esté disponible. Para obtener más información, consulte [Puntos de enlace y regiones de AWS](#).

## Validación de la conformidad para la modernización del mainframe AWS

Los auditores externos evalúan la seguridad y el cumplimiento de la modernización del AWS mainframe como parte de varios programas de AWS cumplimiento. Estos incluyen SOC, PCI, FedRAMP, HIPAA y otros.

Para ver una lista de AWS los servicios incluidos en el ámbito de los programas de conformidad específicos, consulte [Servicios de AWS incluidos](#) . Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#) .

Su responsabilidad en materia de cumplimiento al utilizar la modernización de AWS mainframes viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- [Security and Compliance Quick Start Guides](#) (Guías de inicio rápido de seguridad y conformidad) (Guías de inicio rápido de seguridad y conformidad): Estas guías de implementación analizan las consideraciones en materia de arquitectura y proporcionan los pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad en AWS.
- Documento técnico sobre [cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: en este documento técnico](#) se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- [AWS Recursos de cumplimiento Recursos](#) de de trabajo y guías puede aplicarse a su sector y ubicación.

- [Evaluación de los recursos con las reglas](#) de la guía para AWS Config desarrolladores: AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- [AWS Security Hub](#)— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar el cumplimiento de los estándares y las mejores prácticas del sector de la seguridad.

## Resiliencia en la modernización de los AWS mainframes

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. Las regiones proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja demora. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte [Infraestructura AWS global](#).

## Seguridad de la infraestructura en AWS Mainframe Modernization

Como servicio gestionado, AWS Mainframe Modernization está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte [Seguridad AWS en la nube](#). Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte [Protección de infraestructuras en un marco](#) de buena AWS arquitectura basado en el pilar de la seguridad.

Utilice las llamadas a la API AWS publicadas para acceder a Mainframe Modernization a través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad de seguridad de IAM principal. También puede utilizar [AWS Security Token Service](#) (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

## Acceda AWS Mainframe Modernization mediante un punto final de interfaz ()AWS PrivateLink

Puede usarlo AWS PrivateLink para crear una conexión privada entre su VPC y AWS Mainframe Modernization. Puede acceder a la modernización del mainframe como si estuviera en su VPC, sin utilizar una puerta de enlace a Internet, un dispositivo NAT, una conexión VPN o una conexión AWS Direct Connect. Las instancias de la VPC no necesitan direcciones IP públicas para acceder a Mainframe Modernization.

Esta conexión privada se establece mediante la creación de un punto de conexión de interfaz alimentado por AWS PrivateLink. Creamos una interfaz de red de punto de conexión en cada subred habilitada para el punto de conexión de interfaz. Se trata de interfaces de red administradas por el solicitante que sirven como punto de entrada para el tráfico destinado a Mainframe Modernization.

Para obtener más información, consulte [Acceso directo AWS PrivateLink en la Servicios de AWS](#) [AWS PrivateLink guía](#).

## Consideraciones para Mainframe Modernization

Antes de configurar un punto de conexión de interfaz para Mainframe Modernization, consulte las [consideraciones](#) en la Guía de AWS PrivateLink .

Mainframe Modernization admite la realización de llamadas a todas las acciones de la API a través del punto de conexión de interfaz.

## Crear un punto de conexión de interfaz para Mainframe Modernization

Puede crear un punto final de interfaz para la modernización del mainframe mediante la consola Amazon VPC o AWS Command Line Interface el AWS CLI(). Para obtener más información, consulte [Creación de un punto de conexión de interfaz](#) en la Guía de AWS PrivateLink .

Cree un punto de conexión para Mainframe Modernization utilizando el siguiente nombre de servicio:

```
com.amazonaws.region.m2
```



Si habilita DNS privado para el punto de conexión de interfaz, puede realizar solicitudes a la API para Mainframe Modernization usando su nombre de DNS predeterminado para la región. Por ejemplo, `m2.us-east-1.amazonaws.com`.

## Creación de una política de puntos de conexión para el punto de conexión de interfaz

Una política de punto de conexión es un recurso de IAM que puede adjuntar al punto de conexión de su interfaz. La política de punto de conexión predeterminada permite acceso completo a Mainframe Modernization a través del punto de conexión de interfaz. Para controlar el acceso permitido a Mainframe Modernization desde la VPC, adjunte una política de punto de conexión personalizada al punto de conexión de interfaz.

Una política de punto de conexión especifica la siguiente información:

- Las entidades principales (Cuentas de AWS, usuarios y roles de IAM) que puede realizar acciones
- Las acciones que se pueden realizar.
- El recurso en el que se pueden realizar las acciones.

Para obtener más información, consulte [Control del acceso a los servicios con políticas de punto de conexión](#) en la Guía del usuario de AWS PrivateLink .

Ejemplo: política de punto de conexión de la VPC para acciones de Mainframe Modernization

El siguiente es un ejemplo de una política de un punto de conexión personalizado. Cuando se asocia con un punto de conexión, esta política concede acceso a las acciones de Mainframe Modernization mostradas para todas las entidades principales en todos los recursos.

```
//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": "*"
  }
]
}
```

# Supervisión de la AWS modernización del mainframe

La supervisión es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de la modernización del AWS mainframe y del resto de las soluciones de AWS. AWS proporciona las siguientes herramientas de supervisión para supervisar la modernización del AWS mainframe, informar cuando algo va mal y tomar medidas automáticas cuando sea necesario:

- Amazon CloudWatch monitorea tus AWS recursos y las aplicaciones en las que AWS ejecutas en tiempo real. Puede recopilar métricas y realizar un seguimiento de las métricas, crear paneles personalizados y definir alarmas que le advierten o que toman medidas cuando una métrica determinada alcanza el umbral que se especifique. Por ejemplo, puede CloudWatch hacer un seguimiento del uso de la CPU u otras métricas de sus instancias de Amazon EC2 y lanzar automáticamente nuevas instancias cuando sea necesario. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).
- Amazon CloudWatch Logs le permite supervisar, almacenar y acceder a sus archivos de registro desde instancias de Amazon EC2 y otras fuentes. CloudTrail CloudWatch Los registros pueden monitorear la información de los archivos de registro y notificarle cuando se alcancen ciertos umbrales. También se pueden archivar los datos del registro en un almacenamiento de larga duración. Para obtener más información, consulta la [Guía del usuario CloudWatch de Amazon Logs](#).
- AWS CloudTrail captura las llamadas a la API y los eventos relacionados realizados por su AWS cuenta o en su nombre y entrega los archivos de registro a un bucket de Amazon S3 que especifique. Puede identificar qué usuarios y cuentas llamaron AWS, la dirección IP de origen desde la que se realizaron las llamadas y cuándo se produjeron. Para obtener más información, consulte la [Guía del usuario de AWS CloudTrail](#).

## Supervisión de la modernización AWS del mainframe con Amazon CloudWatch

Puede monitorizar la modernización AWS del mainframe mediante CloudWatch una herramienta que recopila datos sin procesar y los procesa para convertirlos en métricas legibles prácticamente en tiempo real. Estas estadísticas se mantienen durante 15 meses, de forma que pueda obtener acceso a información histórica y disponer de una mejor perspectiva sobre el desempeño de su aplicación web o servicio. También puede establecer alarmas que vigilen determinados umbrales

y enviar notificaciones o realizar acciones cuando se cumplan dichos umbrales. Para obtener más información, consulta la [Guía del CloudWatch usuario de Amazon](#).

En las siguientes tablas se enumeran las métricas y las dimensiones de la modernización AWS del mainframe. El namespace para estas métricas es AWS/M2.

## Métricas del entorno de tiempo de ejecución

Métrica	Descripción
CPUUtilization	<p>El uso de la CPU de las instancias del entorno.</p> <p>Dimensión: environmentId</p> <p>Unidades: porcentaje</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
InboundNetworkThroughput	<p>Rendimiento de red entrante de las instancias del entorno.</p> <p>Dimensión: environmentId</p> <p>Unidades: bytes por segundo</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
MemoryUtilization	<p>El uso de la memoria de las instancias del entorno.</p> <p>Dimensión: environmentId</p> <p>Unidades: porcentaje</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
OutboundNetworkThroughput	<p>Rendimiento de red saliente de las instancias del entorno.</p>

Métrica	Descripción
	<p>Dimensión: environmentId</p> <p>Unidades: bytes por segundo</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>

## Métricas de aplicación

Métrica	Descripción
BatchJobCompletedCount	<p>El número de trabajos completados durante el intervalo de tiempo.</p> <p>Esta métrica está disponible para Micro Focus y para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: Sum</p>
BatchJobFailedCount	<p>El número de trabajos con error durante el intervalo de tiempo.</p> <p>Esta métrica está disponible para Micro Focus y para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: Sum</p>

Métrica	Descripción
JvmMemoryFree	<p data-bbox="831 226 1487 352">La cantidad de memoria disponible que la máquina virtual Java no está utilizando en ese momento.</p> <p data-bbox="831 403 1497 579">Esta métrica solo está disponible para el motor de ejecución AWS Blu Age. Está disponible para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p data-bbox="831 625 1179 659">Dimensión: applicationId</p> <p data-bbox="831 705 1058 739">Unidades: bytes</p> <p data-bbox="831 785 1390 861">Estadísticas válidas: promedio, mínimo, máximo</p>
JvmMemoryMax	<p data-bbox="831 911 1507 987">La cantidad máxima de memoria permitida para la máquina virtual Java.</p> <p data-bbox="831 1037 1500 1213">Esta métrica solo está disponible para el motor de tiempo de ejecución AWS Blu Age. Está disponible para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p data-bbox="831 1260 1179 1293">Dimensión: applicationId</p> <p data-bbox="831 1339 1058 1373">Unidades: bytes</p> <p data-bbox="831 1419 1390 1495">Estadísticas válidas: promedio, mínimo, máximo</p>

Métrica	Descripción
JvmMemoryUsed	<p>La cantidad de memoria utilizada activamente por la máquina virtual Java.</p> <p>Esta métrica solo está disponible para el motor de tiempo de ejecución AWS Blu Age. Está disponible para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: bytes</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
ProcessesActiveCount	<p>El número activo de procesos de ejecución de servicios simultáneos que procesan solicitudes.</p> <p>Esta métrica solo está disponible para el motor de tiempo de ejecución de Micro Focus.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: Sum</p>

Métrica	Descripción
SessionCount	<p>El número de sesiones HTTP de la aplicación.</p> <p>Esta métrica solo está disponible para el motor de tiempo de ejecución AWS Blu Age. Está disponible para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
SharedMemoryFree	<p>La memoria disponible para que el servidor empresarial almacene toda la información que necesita para ejecutar transacciones y trabajos.</p> <p>Esta métrica solo está disponible para el motor de tiempo de ejecución de Micro Focus.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>



Métrica	Descripción
ThreadActiveCount	<p>El número de subprocesos del motor que procesan las solicitudes.</p> <p>Esta métrica solo está disponible para el motor de tiempo de ejecución AWS Blu Age. Está disponible para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>
TransactionCompletedCount	<p>El número de transacciones confirmadas durante el intervalo de tiempo.</p> <p>Esta métrica está disponible para Micro Focus y para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: Sum</p>

Métrica	Descripción
TransactionFailedCount	<p>El número de transacciones con error durante el intervalo de tiempo.</p> <p>Esta métrica está disponible para Micro Focus y para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: recuento</p> <p>Estadísticas válidas: Sum</p>
TransactionResponseTime	<p>El tiempo transcurrido desde el momento en que un usuario envía una solicitud hasta el momento en que la aplicación indica que la solicitud se ha completado.</p> <p>Esta métrica está disponible para Micro Focus y para AWS Blu Age 3.7.0 y versiones posteriores.</p> <p>Dimensión: applicationId</p> <p>Unidades: milisegundos</p> <p>Estadísticas válidas: promedio, mínimo, máximo</p>

## Dimensiones

Dimensión	Descripción
applicationId	Esta dimensión filtra la métrica hasta la aplicación identificada por ID.

Dimensión	Descripción
environmentId	Esta dimensión filtra la métrica hasta el entorno identificado por ID.

## Registro de llamadas a la API de modernización de AWS mainframe mediante AWS CloudTrail

AWS La modernización del mainframe está integrada con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o un AWS servicio en el marco de la modernización del AWS mainframe. CloudTrail captura todas las llamadas a la API para la modernización del AWS mainframe como eventos. Las llamadas capturadas incluyen llamadas desde la consola de modernización del AWS mainframe y llamadas en código a las operaciones de la API de modernización del AWS mainframe. Si crea una ruta, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon S3, incluidos los eventos para la modernización del AWS mainframe. Si no configura una ruta, podrá ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó a AWS Mainframe Modernization, la dirección IP desde la que se realizó la solicitud, quién la realizó, cuándo se realizó y detalles adicionales.

Para obtener más información CloudTrail, consulte la [Guía del AWS CloudTrail usuario](#).

## AWS Información sobre la modernización de mainframes en CloudTrail

CloudTrail está habilitada en su AWS cuenta al crearla. Cuando se produce una actividad en la modernización AWS del mainframe, esa actividad se registra en un CloudTrail evento junto con otros eventos de AWS servicio en el historial de eventos. Puede ver, buscar y descargar los eventos recientes en su AWS cuenta. Para obtener más información, consulte [Visualización de eventos con el historial de CloudTrail eventos](#).

Para obtener un registro continuo de los eventos de su AWS cuenta, incluidos los relacionados con la modernización AWS del mainframe, cree un registro. Un rastro permite CloudTrail entregar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las regiones de AWS. La ruta registra los eventos de todas las regiones de la AWS partición y envía los archivos de registro al bucket de Amazon S3 que especifique. Además, puede configurar otros AWS servicios para analizar

más a fondo los datos de eventos recopilados en los CloudTrail registros y actuar en función de ellos. Para más información, consulte los siguientes temas:

- [Introducción a la creación de registros de seguimiento](#)
- [CloudTrail servicios e integraciones compatibles](#)
- [Configuración de las notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir archivos de CloudTrail registro de varias regiones](#)
- [Recibir archivos de CloudTrail registro de varias cuentas](#)

Todas las acciones de modernización AWS del mainframe se registran CloudTrail y se documentan en la referencia de la [API de modernización del AWS mainframe](#). Por ejemplo, las llamadas a `CreateEnvironment` y `CreateDeployment` las acciones generan entradas en los archivos de CloudTrail registro. `CreateApplication`

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario lo ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales del usuario raíz o del usuario.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro AWS servicio.

Para obtener más información, consulte el elemento [CloudTrail UserIdentity](#).

## Descripción de las entradas del archivo de registro de AWS modernización de mainframe

Un rastro es una configuración que permite la entrega de eventos como archivos de registro a un bucket de Amazon S3 que usted especifique. CloudTrail Los archivos de registro contienen una o más entradas de registro. Un evento representa una solicitud única de cualquier fuente e incluye información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro no son un registro ordenado de las llamadas a la API pública, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra una entrada de CloudTrail registro que demuestra la `CreateApplication` acción.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAI6WZTHGYAEXAMPLE",
    "arn": "arn:aws:sts::444455556666:assumed-role/Admin/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAI6WZTHGYAEXAMPLE",
        "arn": "arn:aws:iam::444455556666:role/Admin",
        "accountId": "444455556666",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-06-01T20:38:22Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-06-01T20:40:39Z",
  "eventSource": "m2.amazonaws.com",
  "eventName": "CreateApplication",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.196.65",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101
Firefox/91.0",
  "requestParameters": {
    "clientToken": "1abc23de-f45g-6789-h01i-jkl2m3456789",
    "name": "MyApp",
    "description": "",
    "engineType": "microfocus",
    "definition": {
      "content": "{}"
    }
  },
  "tags": {},
  "responseElements": {
    "applicationVersion": 1,

```

```
    "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
    "applicationArn": "arn:aws:m2:us-east-1:444455556666:app/
lsfhw7fffrosff2lncwqcu",
    "applicationId": "lsfhw7fffrosff2lncwqcu"
  },
  "requestID": "36982d38-fcde-4bfe-a89a-7bd78d43c926",
  "eventID": "d7f0fc36-46ae-4157-9a79-c79f385fda98",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "444455556666",
  "eventCategory": "Management"
}
```

# Resolución de problemas

Utilice la información de esta sección como ayuda para solucionar los errores más comunes en las aplicaciones de modernización de AWS mainframes y en los entornos de ejecución mediante los motores AWS Blu Age y Micro Focus.

## Temas

- [Error: Time out while waiting for dataset name to be unlocked](#)
- [Cannot access an application's URL](#)
- [AWS Blu Insights no se abre desde la consola](#)

## Error: Time out while waiting for dataset name to be unlocked

- Motor: Blu Age AWS
- Componente: Blusam

Si ve este error en los CloudWatch registros de Amazon de una aplicación de modernización de AWS mainframe que utiliza el motor AWS Blu Age y se ejecuta en un entorno con el patrón de alta disponibilidad, indica que otra aplicación está bloqueando un conjunto de datos compartido. Por lo general, esta situación se produce si la otra aplicación se bloquea o se produce un error y no libera el bloqueo.

## Cómo se produce este error

La aplicación `example-app-1` intenta bloquear un registro `example-record-1` para una operación de escritura. Esta operación crea un bloqueo en el conjunto de datos `example-dataset-1`, que es propietario de `example-record-1`, y un bloqueo en el mismo `example-record-1`. Ahora, otra aplicación, `example-app-2`, intenta bloquear el mismo registro `example-record-1`. El conjunto de datos y el registro ya están bloqueados, por lo que `example-app-2` espera a que se libere el bloqueo. Si `example-app-1` falla, el conjunto de datos bloqueado `example-dataset-1` sigue existiendo, lo que provoca que `example-app-2` cancele su intento de escritura y se genere una excepción de tiempo de espera. Esta situación de bloqueo impide el acceso de todas las aplicaciones a `example-dataset-1`.

## ¿Cómo sabe si esta es su situación?

Busque una aplicación fallida y compruebe si utiliza el mismo conjunto de datos mencionado en el mensaje de error. Compruebe si la aplicación se ejecuta en un entorno de tiempo de ejecución con el patrón de alta disponibilidad. La aplicación que ha generado la excepción de tiempo de espera no puede continuar y mostrará el estado Failed.

## ¿Qué puede hacer?

Para resolver la situación de inmediato, puede forzar la liberación del bloqueo. Para evitar que ocurra una situación similar en el futuro, puede configurar dos parámetros que controlan el mecanismo de reparación automática de Blusam.

## Forzar la liberación del bloqueo

El administrador de bloqueos Blusam usa Amazon ElastiCache for Redis para proporcionar bloqueos compartidos entre aplicaciones. Para liberar los bloqueos ElastiCache, utilice la utilidad CLI de Redis. No puede eliminar bloqueos de registros individuales. Debe eliminar todos los bloqueos del conjunto de datos propietario. Realice los siguientes pasos:

1. Conéctese a su ElastiCache mediante el siguiente comando:


```
redis-cli -h hostname -p port
```

Encontrará sus detalles ElastiCache en la ElastiCache consola en <https://console.aws.amazon.com/elasticache/>.

2. Introduzca su contraseña.
3. Introduzca el comando que desee ejecutar, de la siguiente manera:

Comando	Finalidad
KEYS *	Obtener todas las claves existentes.
KEYS * <i>YOUR_DATASET_NAME</i>	Obtener una clave de bloqueo de conjunto de datos.
DEL <i>THE_RETURNED_KEY</i>	Eliminar un bloqueo de conjunto de datos.



Comando	Finalidad
FLUSHDB	Limpiar todo el Redis.  <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p> <b>Warning</b></p> <p>Se perderán todos los datos de la memoria caché de Redis. Si el Redis se utiliza para otros fines, como gestionar sesiones http, puede que no desee utilizar FLUSHDB.</p> </div>

## Configuración del mecanismo de reparación automática de Blusam

El administrador de bloqueos de Blusam incluye un mecanismo de reparación automática para evitar bloqueos en conjuntos de datos o registros. Puede ajustar los siguientes parámetros en la definición de la aplicación (`application-main.yml`) para configurar el mecanismo de reparación automática:

- `locksDeadTime`: se refiere al tiempo máximo que una aplicación puede mantener un bloqueo. Cuando pasa este tiempo, el bloqueo se declara caducado y se libera inmediatamente. El valor `locksDeadTime` se expresa en milisegundos y el valor predeterminado es 1000.
- `locksCheck`: define la estrategia de administrador de bloqueos de Blusam para comprobar los bloqueos. Todos los bloqueos de Blusam ElastiCache tienen una marca de tiempo y una fecha de caducidad. El valor del parámetro `locksCheck` determina si se eliminan los bloqueos caducados.
- `off`: no se ejecuta ninguna comprobación en ningún momento. Se pueden producir puntos muertos. (No recomendado)
- `reboot`: las comprobaciones se ejecutan cuando se inicia o se reinicia una instancia de aplicación de modernización de AWS mainframe que se ejecuta en un entorno de ejecución de modernización de AWS mainframe. Todos los bloqueos caducados se liberan inmediatamente. (predeterminado)
- `timeout`: las comprobaciones se ejecutan cuando se inicia o reinicia una AWS instancia de aplicación de modernización de AWS mainframe que se ejecuta en un entorno de ejecución de modernización de mainframe, o cuando se agota el tiempo de espera durante un intento de bloquear un conjunto de datos. Los bloqueos caducados se liberan inmediatamente.

Para obtener más información sobre la definición de una aplicación de AWS Blu Age, consulte. [AWS Ejemplo de definición de aplicación de Blu Age](#)

## Administrador de bloqueos de Blusam

En el contexto de un entorno de ejecución de modernización de AWS mainframe que utiliza el patrón de alta disponibilidad, una aplicación de AWS Blu Age puede implementarse varias veces. En el caso de las aplicaciones que gestionan conjuntos de datos de Blusam, pueden producirse problemas de acceso simultáneo. El administrador de bloqueos Blusam garantiza la integridad de los datos y gestiona el acceso de lectura y escritura a los registros y conjuntos de datos al proporcionar bloqueos compartidos entre las aplicaciones que utilizan. ElastiCache Este mecanismo permite que más de una aplicación lea el registro de manera concurrente y garantiza que solo una aplicación escriba el registro simultáneamente.

### Bloqueos de escritura

Para actualizar o eliminar un registro específico, la aplicación debe bloquear primero el conjunto de datos propietario del registro y, a continuación, bloquear el propio registro. Cuando el registro está bloqueado, se libera el bloqueo del conjunto de datos y quedan disponibles para su uso otros registros del mismo conjunto de datos. Cuando se completa la operación de actualización o eliminación, se libera el bloqueo de registro retenido. Solo una aplicación a la vez puede actualizar el registro, lo que impide que otras aplicaciones lean o escriban hasta que se libere el bloqueo, si la política de aplicación definida permite esperar a que se libere.

### Bloqueos de lectura

Mientras no se mantenga ningún bloqueo de escritura en el registro o el conjunto de datos, varias aplicaciones pueden leer los mismos registros al mismo tiempo. Para bloquear un registro para una operación de escritura, se deben liberar todos los bloqueos de lectura.

#### Note

El gestor de bloqueos de Blusam gestiona el acceso desde varios subprocesos de una aplicación determinada mediante el mismo mecanismo de bloqueo.

## Cannot access an application's URL

- Motor: AWS Blu Age y Micro Focus

- Componentes: aplicaciones

Si no puede acceder a la URL de una aplicación de modernización de AWS mainframe en ejecución que creó e implementó en un entorno de ejecución de AWS Mainframe Modernization, es posible que necesite configurar las reglas de entrada en el grupo de seguridad que asoció al entorno de ejecución.

## Cómo se produce este error

Al crear un entorno de tiempo de ejecución, el grupo de seguridad que proporcione, incluido el grupo de seguridad predeterminado, debe tener reglas de entrada configuradas para permitir el tráfico a las aplicaciones implementadas desde fuera de la VPC, si desea permitir este tipo de acceso.

## ¿Cómo sabe si esta es su situación?

La aplicación se ha iniciado correctamente y se ejecuta con normalidad, pero no puede conectarse a ella mediante su URL.

## ¿Qué puede hacer?

Compruebe si el grupo de seguridad de Amazon VPC asociado al entorno de tiempo de ejecución permite el tráfico al entorno en los puertos de aplicación correspondientes. Para comprobar las reglas de grupo de seguridad, siga estos pasos:

1. [Abra la consola de modernización de AWS mainframe en https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. En el panel de navegación izquierdo, seleccione Entornos.
3. Haga clic en el entorno de tiempo de ejecución que aloja la aplicación a la que desea conectarse.
4. Elija Configuración.
5. En Seguridad y red, elija el grupo de seguridad. El enlace abre los detalles del grupo de seguridad en la consola de Amazon VPC.
6. Si es necesario, elija Edición de reglas entrantes y añada la siguiente regla si aún no está presente:

Tipo

TCP personalizada

## Puerto

8196 o el puerto que coincida con las propiedades del oyente especificadas en la definición de la aplicación. Para obtener más información, consulte [Paso 2: Crear la definición de aplicación](#).

## Origen

La dirección IP desde la que se llama a la aplicación. Puede elegir myIP en el menú desplegable. Si sigue teniendo problemas con el tiempo de espera, pruebe a elegir Anywhere IPV4 o Anywhere IPV6. Asegúrese de detener la aplicación y volver a iniciarla después de añadir la regla de entrada al grupo de seguridad.

Para obtener más información, consulte [Trabajar con reglas de grupos de seguridad](#) en la Guía del usuario de Amazon VPC.

# AWS Blu Insights no se abre desde la consola

- Motor: AWS Blu Age
- Componente: Blu Insights

Cuando intenta acceder a Blu Insights desde la consola de modernización del AWS mainframe, no se abre y la nueva pestaña se cierra inmediatamente.

## Cómo se produce este error

El rol que está utilizando para acceder a Blu Insights no tiene permisos suficientes.

## ¿Qué puede hacer?

Adjunte una política de IAM al rol para permitirle acceder a Blu Insights. Asegúrese de que la política incluya los siguientes permisos.

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

```
    "Action": [  
      "m2:GetSignedBluinsightsUrl"  
    ],  
    "Resource": "arn:aws:m2:region:account:*"  
  }  
]  
}
```

Asegúrese de sustituir `region` y por `account` el Región de AWS y Cuenta de AWS correcto.

# Historial de documentos de la Guía del usuario de modernización de AWS mainframe

En la siguiente tabla se describen las versiones de la documentación sobre la modernización de AWS mainframes.

Cambio	Descripción	Fecha
<a href="#">Tutorial actualizado sobre Managed Runtime para Micro Focus</a>	Este tutorial muestra cómo implementar y ejecutar la aplicación de CardDemo ejemplo en un entorno de tiempo de ejecución gestionado o por AWS Mainframe Modernization con el motor de tiempo de ejecución Micro Focus.	5 de febrero de 2024
<a href="#">Notas de publicación de la versión 3.9.0 de AWS Blu Age Runtime and Modernization Tools.</a>	Esta versión de las herramientas de modernización y tiempo de ejecución de AWS Blu Age se centra en múltiples mejoras transversales en todo el producto, con el objetivo de aumentar el rendimiento en arquitecturas de alta disponibilidad, junto con nuevas capacidades para llevar la ejecución de los trabajos al siguiente nivel.	18 de diciembre de 2023
<a href="#">Transfiera archivos entre el mainframe y AWS</a>	Se ha lanzado una nueva función para transferir archivos desde el mainframe de origen a AWS.	27 de noviembre de 2023

[Gestión de las transacciones de las aplicaciones](#)

Se ha lanzado una nueva característica para mostrar y editar las transacciones de las aplicaciones de AWS Mainframe Modernization.

16 de octubre de 2023

[Notas de publicación de la versión 3.6.0 de AWS Blu Age Runtime and Modernization Tools.](#)

Esta versión de las herramientas de modernización y tiempo de ejecución de AWS Blu Age incluye nuevas funciones para las migraciones antiguas de zOS y AS400, orientadas principalmente a ampliar los mecanismos de soporte del CICS, complementar las capacidades de JCL, optimizar el rendimiento en funciones simultáneas y de gran volumen y añadir funciones. multi-data-source

4 de agosto de 2023

[Ahora puede implementar una nueva versión de una aplicación cuando la aplicación esté detenida.](#)

Anteriormente, para implementar una nueva versión de una aplicación, tenía que eliminar la versión implementada. Ahora puede simplemente detener la versión implementada e implementar una nueva versión.

26 de julio de 2023

[AWS Paquete de ejecución de Blu Age para facilitar la implementación de Amazon EC2](#)

AWS La modernización del mainframe con el tiempo de ejecución de AWS Blu Age ahora está disponible con más flexibilidad para configurar la pila completa y el despliegue en las instancias de Amazon EC2 de su empresa. Cuenta de AWS

6 de julio de 2023

[Inicio de sesión único en Blu Age AWSAWS Blu Insights.](#)

AWS AWS Blu Age Blu Insights está disponible a AWS Management Console través del inicio de sesión único.

31 de marzo de 2023

[Versión de disponibilidad general](#)

Versión general de la guía del usuario de modernización de AWS mainframes.

8 de junio de 2022

[Versión inicial](#)

Versión inicial (versión preliminar pública) de la Guía del usuario de modernización de AWS mainframes.

30 de noviembre de 2021



Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.