



Acerca de Red Hat JBoss Enterprise Application Platform 7 [Esto es una traducción automática] 7.1

Guía de migración [Esto es una traducción automática]

Para usar con Red Hat JBoss Enterprise Application Platform 7.1 [Esto es una
traducción automática]

Acerca de Red Hat JBoss Enterprise Application Platform 7 [Esto es una traducción automática] 7.1 Guía de migración [Esto es una traducción automática]

Para usar con Red Hat JBoss Enterprise Application Platform 7.1 [Esto es una traducción automática]

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Resumen

Esta guía proporciona información sobre cómo migrar su aplicación desde versiones anteriores de Red Hat JBoss Enterprise Application Platform.

Table of Contents

CAPÍTULO 1. INTRODUCCIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	9
1.1. ACERCA DE RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	9
1.2. SOBRE LA GUÍA DE MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	9
1.3. ACERCA DE MIGRACIONES Y ACTUALIZACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	9
1.4. ACERCA DEL USO DE EAP_HOME EN ESTE DOCUMENTO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	10
CAPÍTULO 2. PREPARACIÓN PARA LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	12
2.1. RESUMEN DE LA PREPARACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	12
2.2. REVISE LAS CARACTERÍSTICAS DE JAVA EE 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	12
2.3. REVISE LAS NOVEDADES DE JBOSS EAP 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	12
Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]	13
2.4. REVISIÓN DE LA LISTA DE FUNCIONALIDADES NO SOPORTADAS Y YA NO UTILIZADAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	14
2.5. REVISE EL MATERIAL DE JBOSS EAP GETTING STARTED [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	14
2.6. ANÁLISIS Y PLANIFICACIÓN DE LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	15
2.7. COPIA DE SEGURIDAD DE DATOS IMPORTANTES Y REVISIÓN DEL ESTADO DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	16
2.8. MIGRANDO UNA INSTALACIÓN RPM [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	17
2.9. MIGRAR JBOSS EAP EJECUTÁNDOSE COMO UN SERVICIO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	17
CAPÍTULO 3. HERRAMIENTAS PARA AYUDAR EN LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	18
3.1. UTILICE EL KIT DE HERRAMIENTAS DE MIGRACIÓN DE APLICACIONES DE RED HAT PARA ANALIZAR APLICACIONES PARA LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	18
3.2. UTILICE LA HERRAMIENTA DE MIGRACIÓN DE SERVIDORES JBOSS PARA MIGRAR CONFIGURACIONES DE SERVIDORES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	18
CAPÍTULO 4. CAMBIOS DE CONFIGURACIÓN DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	20
4.1. CAMBIOS EN LA INSTALACIÓN DE RPM [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	20
4.2. OPCIONES DE MIGRACIÓN DE CONFIGURACIÓN DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	20
Cambios en la configuración del servidor EJB [Esto es una traducción automática]	20
Gestión de la migración de la CLI Operación [Esto es una traducción automática]	20
4.3. GESTIÓN DE LA MIGRACIÓN DE LA CLI OPERACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	21
Inicie el servidor y la CLI de administración	21
Migre los subsistemas JacORB, de mensajería y web	23
4.4. CAMBIOS DE REGISTRO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	26
4.4.1. Cambios en el prefijo del mensaje de registro [Esto es una traducción automática]	26
4.4.2. Cambios en el controlador de consola de Root Logger [Esto es una traducción automática]	26
4.5. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	26
4.5.1. Reemplace el subsistema web con Undertow [Esto es una traducción automática]	26
4.5.2. Migrar las condiciones de reescritura de JBoss Web [Esto es una traducción automática]	28
4.5.3. Migrar las propiedades del sistema web de JBoss [Esto es una traducción automática]	30
4.5.4. Actualizar el patrón de encabezado del registro de acceso [Esto es una traducción automática]	30
4.5.5. Migrar válvulas globales [Esto es una traducción automática]	31
Migrar válvulas globales [Esto es una traducción automática]	31
JDBCAccessLogValve Procedimiento de migración manual	32

4.5.6. Cambios en el comportamiento Set-Cookie [Esto es una traducción automática]	33
4.5.7. Cambios en el comportamiento de llamada del método HTTP [Esto es una traducción automática]	34
4.5.8. Cambios en el comportamiento del módulo web predeterminado en JBoss EAP 7.1 [Esto es una traducción automática]	35
4.6. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR JGROUPS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	36
4.6.1. JGroups se predetermina a una interfaz de red privada [Esto es una traducción automática]	36
4.6.2. Cambios de canales de JGroups [Esto es una traducción automática]	36
4.7. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR INFINISPAN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	36
4.7.1. Cambios en la configuración de caché predeterminada de Infinispan [Esto es una traducción automática]	36
4.7.2. Cambios en la estrategia de caché Infinispan [Esto es una traducción automática]	36
4.7.3. Configuración de caché de beans de sesión de estado personalizado para pasivación [Esto es una traducción automática]	37
4.7.4. Cambios en el transporte de contenedor de caché Infinispan [Esto es una traducción automática]	37
4.8. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR EJB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	38
DuplicateServiceException en el registro del servidor [Esto es una traducción automática]	38
4.9. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	39
4.9.1. Cambios en la configuración del servidor del subsistema de mensajería [Esto es una traducción automática]	39
Cambios en la administración de JMX [Esto es una traducción automática]	39
Advertencias de la operación de migración del subsistema de mensajería [Esto es una traducción automática]	40
Cambio en el comportamiento del atributo forward-when-no-consumers	40
Cambio en la política de equilibrio de carga del clúster predeterminado	40
Cambios en la configuración del servidor del subsistema de mensajería [Esto es una traducción automática]	40
4.9.2. Migrar datos de mensajería [Esto es una traducción automática]	41
4.9.2.1. Migrar datos de mensajería mediante exportación e importación [Esto es una traducción automática]	41
Ejemplo: Formato de registro de acceso en JBoss EAP 6.4 [Esto es una traducción automática]	41
Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]	42
Migrar datos de mensajería [Esto es una traducción automática]	43
Recuperación de una falla de importación de datos de mensajería	44
4.9.2.2. Migrar datos de mensajería utilizando un puente JMS [Esto es una traducción automática]	44
Configurar el servidor Source JBoss EAP 6.4	45
Configurar el servidor JBoss EAP 7.x de destino	45
Migrar datos de mensajería [Esto es una traducción automática]	48
4.9.2.3. Asignación de nombres de carpeta de mensajería [Esto es una traducción automática]	48
4.9.2.4. Copia de seguridad de los datos de la carpeta de mensajería [Esto es una traducción automática]	48
4.9.3. Migrar destinos JMS [Esto es una traducción automática]	49
4.9.4. Migrar interceptores de mensajería [Esto es una traducción automática]	49
4.9.5. Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]	50
4.9.6. Configuración de un adaptador de recursos JMS genérico [Esto es una traducción automática]	50
4.9.7. Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]	50
4.9.8. Cambios en el comportamiento de serialización JMS entre releases [Esto es una traducción automática]	51
4.10. CAMBIOS EN LA ADMINISTRACIÓN DE JMX [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	52
4.11. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR ORB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	54

4.12. MIGRAR LA CONFIGURACIÓN DEL SUBSISTEMA DE SUBPROCESOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	56
4.13. MIGRAR LA CONFIGURACIÓN DEL SUBSISTEMA REMOTO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	57
4.14. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR WEBSOCKET [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	57
4.15. CAMBIOS EN EL SERVIDOR DE SINGLE SIGN-ON [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	58
4.16. CAMBIOS EN LA CONFIGURACIÓN DE DATASOURCE [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	58
4.16.1. Nombre del controlador JDBC Datasource [Esto es una traducción automática]	58
Controlador que contiene una clase individual	58
Conductor que contiene múltiples clases	58
4.17. CAMBIOS DE CONFIGURACIÓN DEL SERVIDOR DE SEGURIDAD [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	59
4.17.1. Cambios en el comportamiento de seguridad heredado entre JBoss EAP 7.0 y JBoss EAP 7.1 [Esto es una traducción automática]	59
4.17.1.1. Cambio de estado HTTP para dominios de LDAP inalcanzables [Esto es una traducción automática]	59
4.17.1.2. Habilitación del dominio de seguridad LDAP para analizar funciones desde un DN [Esto es una traducción automática]	59
4.17.1.3. Cambios en el envío del certificado JBoss EAP SSL a un servidor LDAP [Esto es una traducción automática]	60
4.17.2. Cambios en el modo FIPS [Esto es una traducción automática]	60
4.18. CAMBIOS EN EL SUBSISTEMA DE TRANSACCIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	61
Cambios en el subsistema de transacciones [Esto es una traducción automática]	61
Cambios en el subsistema de transacciones [Esto es una traducción automática]	61
4.19. CAMBIOS A LA CONFIGURACIÓN MOD_CLUSTER [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	62
4.20. VISUALIZACIÓN DE CAMBIOS DE CONFIGURACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	62
CAPÍTULO 5. CAMBIOS EN LA MIGRACIÓN DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	63
5.1. CAMBIOS EN LA APLICACIÓN DE SERVICIOS WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	63
5.1.1. Cambios de soporte de JAX-RPC [Esto es una traducción automática]	63
5.1.2. Cambios en los servicios web de Apache CXF Spring [Esto es una traducción automática]	64
Migrar interceptores de mensajería [Esto es una traducción automática]	64
Características de Apache CXF	64
Transporte HTTP Apache CXF	65
5.1.3. Cambios en WS-Security	66
5.1.4. Cambio de estructura de JBoss Modules [Esto es una traducción automática]	66
5.1.5. Requisitos y cambios del castillo hinchable [Esto es una traducción automática]	66
5.1.6. Estrategia de selección de bus Apache CXF [Esto es una traducción automática]	67
5.1.7. Requisitos de JAX-WS 2.2 para WebServiceRef [Esto es una traducción automática]	67
5.1.8. IgnoreHttpsHost CN Check Change [Esto es una traducción automática]	67
5.1.9. Configuración del lado del servidor y carga de clases [Esto es una traducción automática]	67
5.1.10. La depreciación de las normas endosadas de Java anula el mecanismo [Esto es una traducción automática]	68
5.1.11. Especificación del descriptor en EAR Archive [Esto es una traducción automática]	68
5.2. ACTUALICE EL PUERTO Y EL CONECTOR REMOTO DE URL [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	68
5.3. CAMBIOS EN LA APLICACIÓN DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	69
5.3.1. Reemplace o actualice los descriptores de implementación de JMS [Esto es una traducción automática]	69
5.3.2. Actualizar clientes externos JMS [Esto es una traducción automática]	70

5.3.3. Reemplazar la API HornetQ [Esto es una traducción automática]	70
5.4. CAMBIOS DE APLICACIÓN JAX-RS Y RESTEASY [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	70
5.4.1. RESETClases obsoletas [Esto es una traducción automática]	70
Clases de Interceptor y Cuerpo de Mensaje	70
API cliente	73
StringConverter	74
5.4.2. Clases RESTEasy eliminadas o protegidas [Esto es una traducción automática]	74
ResteasyProviderFactory Agregar métodos	74
Clases adicionales eliminadas de RESTEasy 3	74
5.4.3. Cambios RESTEasy adicionales [Esto es una traducción automática]	74
SignedInput y SignedOutput	74
Cambios en WS-Security	74
Filtros del lado del cliente	75
Soporte HTTP asíncrono	75
Caché del lado del servidor	75
Cambios de Jackson Provider [Esto es una traducción automática]	75
Charset predeterminado UTF-8 en el encabezado de tipo de contenido	75
SerializableProvider	75
Coincidencia de solicitudes a métodos de recursos	75
5.4.4. RESTEasy SPI Cambios [Esto es una traducción automática]	76
Excepciones de SPI	76
InjectorFactory y registro	76
5.4.5. Cambios de Jackson Provider [Esto es una traducción automática]	76
5.4.6. Cambios en la integración RESTEasy de primavera [Esto es una traducción automática]	77
5.4.7. Cambios RESTEasy Jettison JSON Provider [Esto es una traducción automática]	77
5.5. CAMBIOS DE LA APLICACIÓN CDI 1.2 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	77
Archivos de Frijol	78
Aclaración de la resolución de conversación	79
Resolución del observador	79
5.6. MIGRAR DEPENDENCIAS DE MÓDULOS EXPLÍCITOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	79
Revise las dependencias para la disponibilidad	79
Dependencias que requieren exploración de anotación	79
5.7. CAMBIOS DE MIGRACIÓN DE HIBERNATE Y JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	80
5.7.1. Hibernate ORM 3.0 [Esto es una traducción automática]	80
5.7.2. Hibernate ORM 4.0 - 4.3 [Esto es una traducción automática]	80
5.7.3. Migración a Hibernate ORM 5 [Esto es una traducción automática]	80
RESETClases obsoletas [Esto es una traducción automática]	80
Otros cambios a clases y paquetes	80
Tipo de manejo	81
Cambios en el subsistema de transacciones [Esto es una traducción automática]	81
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	82
5.7.4. Migración de Hibernate ORM 5.0 a Hibernate ORM 5.1 [Esto es una traducción automática]	83
Hibernate ORM 3.0 [Esto es una traducción automática]	83
Cambios en la administración de JMX [Esto es una traducción automática]	83
Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]	83
Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]	83
5.8. CAMBIOS EN LA BÚSQUEDA DE HIBERNATE [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	84
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	84
Indexación de id. Campos de relaciones incrustadas	84
Cambios de migración de Hibernate y JPA [Esto es una traducción automática]	84
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	85
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	86

Lucene - Clases renombradas y reempaquetadas	86
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	86
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	87
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	87
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	87
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	87
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	88
Cambios en la búsqueda de Hibernate [Esto es una traducción automática]	88
Cambios que afectan a los integradores avanzados	88
5.9. MIGRAR BEANS DE ENTIDAD A JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	89
5.10. CAMBIOS EN LA PROPIEDAD DE PERSISTENCIA JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	91
5.10.1. Cambios en la propiedad de persistencia de JPA en JBoss EAP 7.1 [Esto es una traducción automática]	91
5.11. MIGRAR CÓDIGO DE CLIENTE EJB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	92
5.11.1. Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]	92
5.11.1.1. Actualice el puerto de conexión remota predeterminado [Esto es una traducción automática]	92
5.11.1.2. Actualice el conector predeterminado [Esto es una traducción automática]	93
5.11.2. Migrar código de nombre de cliente remoto [Esto es una traducción automática]	93
5.11.3. Cambios adicionales del cliente EJB introducidos en JBoss EAP 7.1 [Esto es una traducción automática]	94
5.12. MIGRE LOS CLIENTES PARA USAR EL ARCHIVO DE CONFIGURACIÓN DE WILDFLY [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	98
5.13. MIGRAR CONFIGURACIONES DE PLANES DE IMPLEMENTACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	99
5.14. MIGRAR VÁLVULAS DE APLICACIONES PERSONALIZADAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	99
Migrar configuraciones de SSL [Esto es una traducción automática]	99
Migrar válvulas de autenticador [Esto es una traducción automática]	100
5.15. CAMBIOS DE APLICACIONES DE SEGURIDAD [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	100
5.15.1. Migrar válvulas de autenticador [Esto es una traducción automática]	100
5.15.2. Cambios de PicketLink [Esto es una traducción automática]	100
5.15.3. Otros cambios en la aplicación de seguridad [Esto es una traducción automática]	100
5.16. CAMBIOS EN EL REGISTRO DE JBOSS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	100
5.17. CAMBIOS DE CÓDIGO DE JAVASERVER FACES (JSF) [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	101
Soporte descartado para JSF 1.2	101
Problema de compatibilidad entre JSF 2.1 y JSF 2.2	102
5.18. CAMBIOS DE CARGA DE CLASE DE MÓDULO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	102
5.19. CAMBIOS EN LA AGRUPACIÓN DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	103
5.19.1. Descripción general de nuevas características de agrupamiento [Esto es una traducción automática]	103
5.19.2. Cambios en clústeres de sesiones web [Esto es una traducción automática]	103
5.19.3. Cambios de agrupamiento de EJB de sesión de estado [Esto es una traducción automática]	105
5.19.4. Clustering Services Changes [Esto es una traducción automática]	106
5.19.5. Migrar agrupamiento HA Singleton [Esto es una traducción automática]	107
CAPÍTULO 6. CAMBIOS DIVERSOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	108
6.1. CAMBIOS PARA LA ENTREGA DE JBOSS EAP NATIVES Y APACHE HTTP SERVER [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	108
6.2. CAMBIOS EN IMPLEMENTACIONES EN AMAZON EC2 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	109
6.3. DESINSTALAR APLICACIONES QUE INCLUYEN MÓDULOS COMPARTIDOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	109

6.4. CAMBIOS EN LAS SECUENCIAS DE COMANDOS DE JBOSS EAP [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	110
6.5. ELIMINACIÓN DEL SOPORTE OSGI [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	110
CAPÍTULO 7. MIGRACIÓN A ELYTRON EN JBOSS EAP 7.1 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	112
7.1. DESCRIPCIÓN GENERAL DE ELYTRON [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	112
7.2. MIGRAR BÓVEDAS Y PROPIEDADES SEGURAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	113
7.2.1. Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]	113
Migre los clientes para usar el archivo de configuración de WildFly [Esto es una traducción automática]	113
Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]	114
Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]	115
7.2.2. Migre las propiedades de seguridad a Elytron [Esto es una traducción automática]	116
7.3. MIGRAR LA CONFIGURACIÓN DE AUTENTICACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	117
7.3.1. Migre la autenticación y autorización basada en las propiedades a Elytron [Esto es una traducción automática]	117
7.3.1.1. Migre la configuración basada en propiedades de PicketBox a Elytron [Esto es una traducción automática]	117
Parcialmente migrar al exponer el dominio de seguridad de PicketBox a Elytron	118
Migre la autenticación y autorización basada en las propiedades a Elytron [Esto es una traducción automática]	120
7.3.1.2. Migre la configuración heredada basada en propiedades a Elytron [Esto es una traducción automática]	122
7.3.2. Migre la configuración de autenticación LDAP a Elytron [Esto es una traducción automática]	125
7.3.2.1. Migre la autenticación LDAP heredada a Elytron [Esto es una traducción automática]	128
7.3.3. Migre la configuración de autenticación de la base de datos a Elytron [Esto es una traducción automática]	130
7.3.3.1. Migre la autenticación de base de datos heredada a Elytron [Esto es una traducción automática]	131
7.3.4. Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]	132
Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]	132
Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]	134
Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]	136
Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]	136
7.3.5. Migre tiendas compuestas a Elytron [Esto es una traducción automática]	138
Ejemplo: Configuración de PicketBox LoginModule [Esto es una traducción automática]	138
Ejemplo: Comandos de configuración de dominios de seguridad heredados [Esto es una traducción automática]	139
Ejemplo: Configuración de LDAP Security Realm [Esto es una traducción automática]	141
7.3.6. Migre los dominios de seguridad que usan almacenamiento en caché para Elytron [Esto es una traducción automática]	142
Ejemplo: configuración de dominio de seguridad en caché PicketBox [Esto es una traducción automática]	142
Ejemplo: configuración de dominio de seguridad en caché Elytron [Esto es una traducción automática]	144
7.3.7. Migre la seguridad de JACC a Elytron [Esto es una traducción automática]	145
7.4. MIGRAR CLIENTES DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	146
7.4.1. Migre una configuración de Naming Client a Elytron [Esto es una traducción automática]	146
7.4.1.1. Migre el cliente de nombres usando el enfoque de archivo de configuración [Esto es una traducción automática]	146
7.4.1.2. Migre el cliente de nombres usando el enfoque programático [Esto es una traducción automática]	147
7.4.2. Migre un cliente EJB a Elytron [Esto es una traducción automática]	148

7.4.2.1. Migre el cliente EJB utilizando el enfoque del archivo de configuración [Esto es una traducción automática]	148
7.4.2.2. Migre el cliente EJB utilizando el enfoque programático [Esto es una traducción automática]	150
7.5. MIGRAR CONFIGURACIONES DE SSL [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	150
7.5.1. Migre una configuración simple de SSL a Elytron [Esto es una traducción automática]	150
7.5.2. Migre la autenticación SSL de Client-Cert a Elytron [Esto es una traducción automática]	152
CAPÍTULO 8. MIGRACIÓN DESDE VERSIONES ANTERIORES DE JBOSS EAP [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	156
8.1. MIGRANDO DE JBOSS EAP 5 A JBOSS EAP 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	156
8.2. RESUMEN DE LOS CAMBIOS REALIZADOS EN CADA VERSIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	156
8.3. REVISE EL CONTENIDO EN LAS GUÍAS DE MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	157
8.4. REFERENCIA DE ACTUALIZACIÓN DE COMPONENTES JBOSS EAP 5 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	157
APÉNDICE A. MATERIAL DE REFERENCIA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	165
A.1. ADVERTENCIAS DE LA OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA JACORB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	165
A.2. ADVERTENCIAS DE LA OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	166
Reemplace los atributos del grupo de difusión obsoleto o del grupo de descubrimiento.	169
A.3. ADVERTENCIAS DE OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	171
Advertencias de operación de migración del subsistema web [Esto es una traducción automática]	175
Atributos del conector web SSL	175
Atributos de recursos estáticos web	175
Atributos de recursos de SSO web	176
Asignación de atributos de mensajería [Esto es una traducción automática]	176
Atributos del conector web	177
A.4. MIGRAR LA REFERENCIA DE PROPIEDADES DEL SISTEMA WEB JBOSS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	177
A.5. COMPATIBILIDAD E INTEROPERABILIDAD ENTRE VERSIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]	187
EJB remota sobre IIOP	187
EJB remoto Uso de JNDI	187
Interacción de EJB utilizando @WebService	187
Cliente autónomo de mensajería	188
Migrar datos de mensajería [Esto es una traducción automática]	188
Puentes JMS	188

CAPÍTULO 1. INTRODUCCIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

1.1. ACERCA DE RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Red Hat JBoss Enterprise Application Platform 7 (JBoss EAP) es una plataforma de middleware basada en estándares abiertos y que cumple con la especificación Java Enterprise Edition 7.

JBoss EAP incluye una nueva estructura modular que permite la habilitación de servicios únicamente cuando se requiere, mejorando así, la velocidad de arranque.

La consola administrativa y la interfaz de línea de comandos administrativa (CLI) hacen innecesaria la modificación de archivos de configuración XML y agregan la capacidad de codificar y automatizar tareas.

JBoss EAP proporciona dos modos operativos para instancias JBoss EAP: el servidor autónomo o el dominio administrado. El servidor autónomo representa la ejecución de JBoss EAP como una instancia de servidor sencilla. El modo operativo de dominio administrado permite la administración de múltiples instancias JBoss EAP desde un punto de control único.

Además, JBoss EAP incluye APIs y marcos de trabajo de desarrollo para desarrollar rápidamente aplicaciones Java EE seguras y escalables.

1.2. SOBRE LA GUÍA DE MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

El propósito de esta guía es documentar los cambios que se requieren para ejecutar e implementar con éxito Red Hat JBoss Enterprise Application Platform 6 aplicaciones en Red Hat JBoss Enterprise Application Platform 7. Proporciona información sobre las nuevas funciones disponibles en esta versión, las características desaprobadas y no compatibles, y cualquier aplicación y actualizaciones de configuración del servidor que puedan ser necesarias para evitar cambios en el comportamiento de la aplicación.

También proporciona información sobre las herramientas que pueden ayudar con la migración, como [Kit de herramientas de migración de aplicaciones de Red Hat](#), que simplifica la migración de aplicaciones Java, y [Herramienta de migración de servidor JBoss](#), que actualiza la configuración del servidor.

Una vez que la aplicación se implementa y se ejecuta con éxito, se pueden hacer planes para actualizar los componentes individuales para usar las nuevas funciones y funciones de JBoss EAP 7.

Si planea migrar sus aplicaciones de JBoss EAP 5 directamente a JBoss EAP 7, consulte [Migración desde versiones anteriores de JBoss EAP](#).

1.3. ACERCA DE MIGRACIONES Y ACTUALIZACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Actualizaciones principales [Esto es una traducción automática]

Se requiere una actualización o migración importante cuando una aplicación se mueve de una versión principal a otra, por ejemplo, de JBoss EAP 6.4 a JBoss EAP 7.0. Si una aplicación sigue las especificaciones de Java EE, no accede a API obsoletas y no contiene código propietario, es posible

ejecutar la aplicación en JBoss EAP 7 sin ningún cambio en el código de la aplicación. Sin embargo, la configuración del servidor ha cambiado en JBoss EAP 7 y requiere migración. Este tipo de migración se trata en esta guía.

Actualizaciones menores [Esto es una traducción automática]

JBoss EAP periódicamente proporciona lanzamientos de puntos, que son actualizaciones menores que incluyen correcciones de errores, soluciones de seguridad y nuevas características. La información sobre los cambios realizados en un punto de publicación se documenta en esta guía y en [7.1.0 Notas de la versión](#).

Puede utilizar la Herramienta de migración de servidores de JBoss para actualizar automáticamente de una versión de punto a otra, por ejemplo de JBoss EAP 7.0 a JBoss EAP 7.1. Para obtener información sobre cómo configurar y ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).

Si lo prefiere, puede realizar una actualización manual de la configuración del servidor. Las instrucciones sobre cómo realizar una actualización manual están documentadas en [Actualización de JBoss EAP](#) en el JBoss EAP *Guía de parches y actualizaciones*.

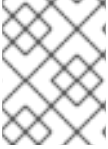
Parches acumulativos [Esto es una traducción automática]

JBoss EAP también proporciona periódicamente parches acumulativos que contienen errores y soluciones de seguridad. Los parches acumulativos incrementan la liberación en el último dígito, por ejemplo de 7.1.0 a 7.1.1. La instalación del parche se trata en el JBoss EAP [Guía de parches y actualizaciones](#).

1.4. ACERCA DEL USO DE EAP_HOME EN ESTE DOCUMENTO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En este documento, la variable **EAP_HOME** se usa para indicar el camino a la instalación de JBoss EAP. Reemplace esta variable con la ruta real a su instalación de JBoss EAP.

- Si instaló JBoss EAP utilizando el método de instalación ZIP, el directorio de instalación es el **jboss-eap-7.1** directorio donde extrajo el archivo ZIP.
- Si instaló JBoss EAP utilizando el método de instalación RPM, el directorio de instalación es **/opt/rh/eap7/root/usr/share/wildfly/**.
- Si usó el instalador para instalar JBoss EAP, la ruta predeterminada para **EAP_HOME** es **\${user.home}/EAP-7.1.0**:
 - Para Red Hat Enterprise Linux, Solaris y HP-UX: **/home/USER_NAME/EAP-7.1.0/**
 - Para Microsoft Windows: **C:\Users\USER_NAME\EAP-7.1.0**
- Si usó el instalador JBoss Developer Studio para instalar y configurar el servidor JBoss EAP, la ruta predeterminada para **EAP_HOME** es **\${user.home}/jbdevstudio/runtimes/jboss-eap**:
 - Para Red Hat Enterprise Linux: **/home/USER_NAME/jbdevstudio/runtimes/jboss-eap/**
 - Para Microsoft Windows: **C:\Users\USER_NAME\jbdevstudio\runtimes\jboss-eap** o **C:\Documents and Settings\USER_NAME\jbdevstudio\runtimes\jboss-eap**

**NOTA**

EAP_HOME no es una variable de entorno ***JBoss_HOME*** es la variable de entorno utilizada en las secuencias de comandos.

CAPÍTULO 2. PREPARACIÓN PARA LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

2.1. RESUMEN DE LA PREPARACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 7, se hizo un esfuerzo para proporcionar compatibilidad con versiones anteriores para las aplicaciones JBoss EAP 6. Sin embargo, si su aplicación usa características que fueron desaprobadas o una funcionalidad que se eliminó de JBoss EAP 7, es posible que deba realizar cambios en el código de su aplicación.

Además, varias cosas han cambiado en esta versión que podrían afectar la implementación de las aplicaciones de JBoss EAP 7. Se recomienda que investigue y planifique antes de intentar migrar su aplicación.

- Familiarízate con el [características de Java EE 7](#).
- revisión [qué hay de nuevo en JBoss EAP 7](#).
- Revisa la lista de [funciones desaprobadas y no compatibles](#).
- Revise el material en JBoss EAP 7 [Guía de inicio](#).
- Eche un vistazo a la [herramientas que pueden ayudar con las tareas de migración](#).

Una vez que se sienta cómodo con los cambios de características, los materiales de desarrollo y las herramientas que pueden ayudarlo en sus esfuerzos de migración, puede comenzar a evaluar sus aplicaciones y la configuración de su servidor para determinar los cambios necesarios para ejecutar en JBoss EAP 7.

2.2. REVISE LAS CARACTERÍSTICAS DE JAVA EE 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Java EE 7 incluye muchas mejoras para facilitar el desarrollo y la ejecución de aplicaciones ricas en funciones en nubes privadas y públicas. Incorpora nuevas funciones y los últimos estándares como HTML5, WebSocket, JSON, Batch y Concurrency Utilities. Las actualizaciones incluyen JPA 2.1, JAX-RS 2.0, Servlet 3.1, Expression Language 3.0, JMS 2.0, JSF 2.2, EJB 3.2, CDI 1.2 y Bean Validation 1.1.

Puede encontrar más información sobre Java EE 7, incluidos los tutoriales, en el sitio web de Oracle: [Java EE de un vistazo](#)

2.3. REVISE LAS NOVEDADES DE JBOSS EAP 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 7 incluye algunas mejoras notables y mejoras sobre la versión anterior.

Java EE 7

JBoss EAP 7 es una implementación certificada de Java EE 7, que cumple tanto con el perfil web como con las especificaciones completas de la plataforma. También incluye soporte para las últimas iteraciones de CDI 1.2 y Web Sockets 1.1.

Undertow

Undertow es el nuevo servidor web liviano, flexible y de alto rendimiento incluido en JBoss EAP 7, que reemplaza a JBoss Web. Escrito en Java, está diseñado para un rendimiento y una escalabilidad máximos. Es compatible con las últimas tecnologías web, como el nuevo estándar HTTP / 2.

Apache ActiveMQ Artemis

Apache ActiveMQ Artemis es el nuevo proveedor de mensajería incorporado de JBoss EAP 7. Basado en una donación de código de HornetQ, este subproyecto de Apache proporciona un rendimiento sobresaliente basado en una arquitectura comprobada de no bloqueo.

IronJacamar 1.2

El último IronJacamar proporciona un soporte estable y rico en características para JCA y DataSources.

JBossWS 5

La quinta generación de JBossWS es un gran avance, que aporta nuevas funciones y mejoras de rendimiento a los servicios web de JBoss EAP 7.

RESTEasy SPI Cambios [Esto es una traducción automática]

JBoss EAP 7 incluye la última generación de RESTEasy. Va más allá de las API REST estándar de Java EE (JAX-RS 2.0) al proporcionar una serie de extensiones útiles como JSON Web Encryption, Jackson, JSON-P y Jettison.

OpenJDK ORB

JBoss EAP 7 reemplazó la implementación de JacORB IIOP con una rama descendente de OpenJDK ORB, lo que mejoró la interoperabilidad con JVM ORB y Java EE RI.

Feature Rich Clustering

El soporte de clustering fue fuertemente refactorizado en JBoss EAP 7 e incluye varias API públicas para el acceso de las aplicaciones.

Introducción [Esto es una traducción automática]

Al utilizar la actualización HTTP, JBoss EAP 7 ha movido casi todos sus protocolos para ser multiplexados en solo dos puertos HTTP: un puerto de administración (9990) y un puerto de aplicación (8080).

Registro mejorado

La API de administración ahora es compatible con la capacidad de enumerar y ver los archivos de registro disponibles en un servidor, o incluso definir formateadores personalizados que no sean el formateador de patrones predeterminado. La configuración de inicio de sesión de implementación también se mejora en gran medida.

Para obtener una lista completa de las nuevas características introducidas en JBoss EAP 7.0, consulte [Nuevas características y mejoras](#) en el JBoss EAP 7.0.0 *Notas de la versión*.

Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]

Elitrono

Elytron, basado en el proyecto WildFly Elytron, es el nuevo marco de seguridad en JBoss EAP 7.1. Está diseñado para unificar la seguridad en todo el servidor de aplicaciones.

Cambios en la administración de JMX [Esto es una traducción automática]

La consola de administración se ha mejorado para proporcionar la capacidad de configurar más subsistemas, proporcionar mejoras **transaction** métricas de recursos de subsistema y transacción, y administrar muchas configuraciones adicionales.

Cambios en la administración de JMX [Esto es una traducción automática]

La CLI de administración proporciona compatibilidad mejorada para la respuesta y los archivos adjuntos, la configuración del módulo y el soporte de depuración a través del **echo-command** argumento.

Para obtener una lista completa de las nuevas características introducidas en JBoss EAP 7.1, consulte [Nuevas características y mejoras](#) en el *7.1.0 Notas de la versión* en el portal de clientes de Red Hat.

2.4. REVISIÓN DE LA LISTA DE FUNCIONALIDADES NO SOPORTADAS Y YA NO UTILIZADAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Antes de migrar su aplicación a JBoss EAP 7, tenga en cuenta que algunas funciones que estaban disponibles en versiones anteriores de JBoss EAP podrían haber quedado en desuso o dejar de ser compatibles.

Se eliminó el soporte para algunas tecnologías debido al alto costo de mantenimiento, el bajo interés de la comunidad y soluciones alternativas mucho mejores. El siguiente es un breve resumen de algunas de las características no compatibles.

Migrar beans de entidad a JPA [Esto es una traducción automática]

Los beans de entidad EJB ya no son compatibles. Si su aplicación utiliza beans de entidad EJB, debe migrar el código para usar JPA, que ofrece una API mucho más eficiente y flexible.

JAX-RPC

Debido a que JAX-WS ofrece una solución mucho más precisa y completa, el código escrito para JAX-RPC se debe migrar para usar JAX-WS.

JSR-88

La especificación API de implementación de aplicaciones Java EE (JSR-88), que definió un contrato para permitir que herramientas de múltiples proveedores configuren e implementen aplicaciones en cualquier producto de plataforma Java EE, no fue ampliamente adoptada. Debe usar otra opción compatible con JBoss EAP para la implementación de aplicaciones, como la consola de administración, la CLI de administración, el escáner de implementación o Maven.

Configuración de un adaptador de recursos JMS genérico [Esto es una traducción automática]

- En JBoss EAP 7.0, no se admite la capacidad de configurar un adaptador de recursos JMS genérico para conectarse a un proveedor JMS.
- En JBoss EAP 7.1, se admite la capacidad de configurar un adaptador de recursos JMS genérico para conectarse a un proveedor JMS.

Para obtener una lista completa de las características desaprobadas y no admitidas en JBoss EAP 7.0, consulte [Funcionalidad desatendida y no admitida](#) en el JBoss EAP *7.0.0 Notas de la versión* en el portal de clientes de Red Hat.

Para obtener una lista completa de las características obsoletas y no admitidas en JBoss EAP 7.1, consulte [Funcionalidad desatendida y no admitida](#) en el JBoss EAP *7.1.0 Notas de la versión* en el portal de clientes de Red Hat.

2.5. REVISE EL MATERIAL DE JBOSS EAP GETTING STARTED [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Asegúrese de revisar el JBoss EAP [Guía de inicio](#). Contiene la siguiente información importante:

- Cómo descargar e instalar JBoss EAP 7

- Cómo descargar e instalar Red Hat JBoss Developer Studio
- Cómo configurar Maven para su entorno de desarrollo, administrar dependencias de proyecto y configurar sus proyectos para utilizar los artefactos de la Lista de materiales de JBoss EAP (BOM)
- Cómo descargar y ejecutar las aplicaciones de ejemplo de inicio rápido que se entregan con el producto

2.6. ANÁLISIS Y PLANIFICACIÓN DE LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Cada aplicación y configuración del servidor es única, y debe comprender a fondo los componentes y la arquitectura de la aplicación existente y la plataforma del servidor antes de intentar la migración. Su plan de migración debe incluir una hoja de ruta detallada para probar e implementar la producción que tenga en cuenta la siguiente información.

Identificar a las personas responsables de la migración

Identifique las partes interesadas, los gerentes de proyecto, los desarrolladores, los administradores y otras personas responsables de la migración.

Revise la configuración y el hardware de la plataforma del servidor de aplicaciones

Examine el servidor de aplicaciones existente y la configuración de la plataforma para determinar cómo se ven afectados por los cambios de características en JBoss EAP 7. La revisión debe incluir los siguientes elementos.

- Sistemas operativos y versiones
- Base de datos utilizada por las aplicaciones
- Servidores web
- Arquitectura de seguridad
- Número y tipo de procesadores
- Cantidad de memoria
- Cantidad de almacenamiento en disco físico
- Migrar datos de mensajería [Esto es una traducción automática]
- Otros componentes que podrían verse afectados por la migración

Revise el entorno de producción actual

Debe planear recrear el entorno de producción lo más cerca posible para probar y organizar el proceso de migración.

- Tenga en cuenta cualquier configuración de agrupación. Ver [Actualización de un clúster](#) en el JBoss EAP *Guía de parches y actualizaciones* para obtener más información acerca de cómo migrar clústeres.
- Si actualmente está ejecutando un gran dominio administrado, considere un enfoque de migración gradual.
- Determine si necesita migrar cualquier base de datos o datos de mensajes.

Examinar y comprender la aplicación existente

Examine a fondo la aplicación existente JBoss EAP 6. Familiarícese totalmente con su arquitectura, funciones, características y componentes, que incluyen:

- La versión de JVM
- Integración con otros componentes de middleware del servidor de aplicaciones Red Hat
- Integración con software propietario de terceros
- Uso de características obsoletas que requerirán reemplazo
- Configuración de la aplicación que incluye descriptores de implementación, JNDI, persistencia, configuración y agrupamiento JDBC, temas y colas JMS y registro

Identifique cualquier código o incompatibilidades de configuración que requerirán modificaciones durante la migración a JBoss EAP 7.

Crear un plan de prueba detallado

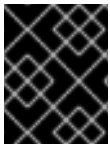
- El plan debe incluir pruebas de regresión y requisitos de criterios de aceptación.
- También debe incluir pruebas de rendimiento.
- Configure un entorno de transición lo más cercano posible al entorno de producción para probar la migración antes del lanzamiento a producción.
- ¡Asegúrese de crear un plan de respaldo y retroceso!

Revise el contenido en las guías de migración [Esto es una traducción automática]

- Evalúe las habilidades del equipo de desarrollo y planifique capacitación o ayuda de consultoría adicional.
- Tenga en cuenta que se necesitarán hardware adicional y otros recursos para organizar y probar durante el proceso de migración hasta que se complete el esfuerzo.
- Determine si se necesita alguna capacitación formal. Si es así, agréguelo al horario.

Ejecutar el plan

Reúna los recursos necesarios e implemente el plan de migración.



IMPORTANTE

Antes de realizar modificaciones en su aplicación, asegúrese de crear una copia de seguridad.

2.7. COPIA DE SEGURIDAD DE DATOS IMPORTANTES Y REVISIÓN DEL ESTADO DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Antes de migrar su aplicación, debe ser consciente de los siguientes problemas potenciales.

- La migración puede eliminar carpetas temporales. Cualquier implementación almacenada en

data/content/ el directorio se debe respaldar antes de la migración y se debe restaurar después de que se complete. De lo contrario, el servidor no podrá iniciarse debido al contenido faltante.

- Antes de la migración, maneje cualquier transacción abierta y elimine **data/tx-object-store/** directorio de transacciones
- Los datos de temporizador persistentes en **data/timer-service-data** debe verificarse para determinar si es compatible. Antes de la migración, revise el **deployment - *** archivos en ese directorio para determinar qué temporizadores están todavía en uso.

Asegúrese de hacer una copia de seguridad de la configuración y las aplicaciones del servidor actual antes de comenzar.

2.8. MIGRANDO UNA INSTALACIÓN RPM [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]



IMPORTANTE

No se admite tener más de una instancia de JBoss EAP instalada en RPM en un solo Servidor Red Hat Enterprise Linux. Como resultado, recomendamos que migre su instalación de JBoss EAP a una nueva máquina al migrar a JBoss EAP 7.

Al migrar una instalación de JBoss EAP RPM de JBoss EAP 6 a JBoss EAP 7, asegúrese de que JBoss EAP 7 esté instalado en una máquina que no tenga una instalación existente de JBoss EAP RPM.

Para instalar JBoss EAP 7 usando RPM, vea el JBoss EAP [Guía de instalación](#).

El consejo de migración en esta guía también se aplica a la migración de instalaciones de RPM de JBoss EAP, pero es posible que deba modificar algunos pasos (como, por ejemplo, cómo iniciar JBoss EAP) para adaptarlos a una instalación de RPM en comparación con una instalación de ZIP o instalador.

2.9. MIGRAR JBOSS EAP EJECUTÁNDOSE COMO UN SERVICIO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Si ejecuta JBoss EAP 6 como un servicio, asegúrese de revisar [Configurando JBoss EAP para ejecutar como un servicio](#) en el JBoss EAP *Guía de instalación* para obtener instrucciones de configuración actualizadas para JBoss EAP 7.

CAPÍTULO 3. HERRAMIENTAS PARA AYUDAR EN LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

3.1. UTILICE EL KIT DE HERRAMIENTAS DE MIGRACIÓN DE APLICACIONES DE RED HAT PARA ANALIZAR APLICACIONES PARA LA MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

El Kit de herramientas de migración de aplicaciones de Red Hat (RHAMT) es un conjunto de herramientas basado en reglas ampliable y personalizable que ayuda a simplificar la migración de aplicaciones Java. Analiza las API, tecnologías y arquitecturas utilizadas por las aplicaciones que planea migrar y proporciona informes de migración detallados para cada aplicación. Estos informes proporcionan la siguiente información.

- Explicaciones detalladas de los cambios de migración necesarios
- Si el cambio informado es obligatorio u opcional
- Si el cambio informado es complejo o trivial
- Enlaces al código que requieren el cambio de migración
- Sugerencias y enlaces a información sobre cómo realizar los cambios necesarios
- Una estimación del nivel de esfuerzo para cada problema de migración encontrado y el esfuerzo total estimado para migrar la aplicación

Puede usar RHAMT para analizar el código y la arquitectura de sus aplicaciones JBoss EAP 6 antes de migrarlos a JBoss EAP 7. La regla RHAMT configurada para la migración de JBoss EAP 6 a JBoss EAP 7 informa sobre descriptores XML y códigos de aplicaciones específicas y parámetros que debe ser reemplazado por una configuración alternativa al migrar a JBoss EAP 7.

Para obtener más información sobre cómo usar el Kit de herramientas de migración de aplicaciones de Red Hat para analizar sus aplicaciones de JBoss EAP 6, consulte el Kit de herramientas de migración de aplicaciones de Red Hat. [Guía de inicio](#).

3.2. UTILICE LA HERRAMIENTA DE MIGRACIÓN DE SERVIDORES JBOSS PARA MIGRAR CONFIGURACIONES DE SERVIDORES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La herramienta de migración de servidores de JBoss es el método preferido para actualizar la configuración de su servidor e incluir las nuevas características y configuraciones en JBoss EAP 7 mientras mantiene su configuración existente. La herramienta de migración de servidores JBoss lee sus archivos de configuración de servidores JBoss EAP existentes y agrega configuraciones para cualquier subsistema nuevo, actualiza las configuraciones existentes del subsistema con nuevas características y elimina cualquier configuración de subsistema obsoleta.

Utilice la herramienta de migración de servidores JBoss para migrar configuraciones de servidores [Esto es una traducción automática]

Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

JBoss Server Migration Tool se envía con JBoss EAP 7.1, por lo que no se requiere una descarga o instalación por separado. Ejecuta la herramienta ejecutando el **jboss-server-migration** script ubicado en **EAP_HOME/bin** directorio. Para obtener más información sobre cómo configurar y

ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).

Se recomienda que use esta versión de la herramienta de migración de servidores de JBoss para migrar la configuración de su servidor a JBoss EAP 7.1 ya que esta versión de la herramienta está [soportado](#).

Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

Para la migración a JBoss EAP 7.0, debe descargar la última distribución binaria de la Herramienta de migración de servidores JBoss desde el [Herramienta de migración de servidor GitHub](#). Para obtener información sobre cómo ejecutar la herramienta, consulte la herramienta de migración de servidores de JBoss. [Guía del usuario](#).



IMPORTANTE

Esta versión de JBoss Server Migration Tool no es compatible. En lugar de utilizar esta versión para migrar la configuración de su servidor a JBoss EAP 7.0, se recomienda que use la [versión soportada de la herramienta](#) para migrar la configuración de su servidor directamente a JBoss EAP 7.1.

CAPÍTULO 4. CAMBIOS DE CONFIGURACIÓN DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.1. CAMBIOS EN LA INSTALACIÓN DE RPM [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 6, la ruta predeterminada para la instalación de RPM fue la `/usr/share/jbossas/` directorio.

JBoss EAP 7 fue construido para [Biblioteca de colecciones de software](#) convenciones. El directorio raíz de las Colecciones de software normalmente se encuentra en el `/opt/` directorio para evitar posibles conflictos entre Colecciones de Software y la instalación del sistema base. El uso de `/opt/` el directorio es recomendado por el Estándar de jerarquía del sistema de archivos (FHS). Como resultado, la ruta predeterminada para la instalación de RPM ha cambiado a `/opt/rh/eap7/root/usr/share/wildfly/` en JBoss EAP 7.

4.2. OPCIONES DE MIGRACIÓN DE CONFIGURACIÓN DEL SERVIDOR [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Para migrar la configuración de su servidor de JBoss EAP 6 a JBoss EAP 7, puede usar el [Herramienta de migración de servidor JBoss](#) o puede realizar una migración manual con la ayuda de [CLI de gestión migrate](#) operación.

Cambios en la configuración del servidor EJB [Esto es una traducción automática]

La herramienta de migración de servidores de JBoss es el método preferido para actualizar su configuración e incluir las nuevas funciones y configuraciones en JBoss EAP 7 mientras mantiene su configuración existente. Para obtener información sobre cómo configurar y ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).

Gestión de la migración de la CLI Operación [Esto es una traducción automática]

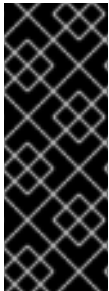
Puede usar la CLI de administración `migrate` operación para actualizar el `jacorb`, `messaging`, y `web` subsistemas en el archivo de configuración del servidor JBoss EAP 6 para permitir que se ejecuten en la nueva versión, pero tenga en cuenta que el resultado no es una configuración completa de JBoss EAP 7. Por ejemplo:

- La operación no actualiza el original `remote` protocolo y configuración de puerto para el nuevo `http-remoting` y nuevas configuraciones de puerto ahora usadas en JBoss EAP 7.
- La configuración no incluye los nuevos subsistemas o características de EAP de JBoss, como las implementaciones de singleton en clúster o el apagado ordenado.
- La configuración no incluye las nuevas características de Java EE 7, como el procesamiento por lotes.
- los `migrate` la operación no migra el `ejb3` configuración del subsistema Para obtener información sobre posibles problemas de migración de EJB, consulte [Cambios en la configuración del servidor EJB](#).

Para obtener más información sobre el uso de `migrate` operación a la migración la configuración del servidor, vea [Gestión de la migración de la CLI Operación](#).

4.3. GESTIÓN DE LA MIGRACIÓN DE LA CLI OPERACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Puede usar la CLI de administración para actualizar los archivos de configuración de su servidor JBoss EAP 6 para que se ejecuten en JBoss EAP 7. La CLI de administración proporciona una **migrate** operación para actualizar automáticamente el **jacorb**, **messaging**, y **web** subsistemas de la versión anterior a la nueva configuración. También puedes ejecutar el **describe-migration** operación para el **jacorb**, **messaging**, y **web** subsistemas para revisar los cambios de configuración de migración propuestos antes de realizar la migración. No hay reemplazos para el **cmp**, **jaxr**, o **threads** subsistemas y deben eliminarse de la configuración del servidor.



IMPORTANTE

Ver [Opciones de migración de configuración del servidor](#) para las limitaciones de la **migrate** operación. La herramienta de migración de servidores de JBoss es el método preferido para actualizar su configuración e incluir las nuevas funciones y configuraciones en JBoss EAP 7 mientras mantiene su configuración existente. Para obtener información sobre cómo configurar y ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).

Tabla 4.1. Operación CLI de Migración y Gestión de Subsistemas [Esto es una traducción automática]

JBoss EAP 6 Subsystem	JBoss EAP 7 Subsystem	Gestión de la migración de la CLI Operación [Esto es una traducción automática]
cmp	sin reemplazo	Borrar
jacorb	iiop-openjdk	emigrar
jaxr	sin reemplazo	Borrar
mensajería	messaging-activemq	emigrar
trapos	sin reemplazo	Borrar
web	resaca	emigrar

Inicie el servidor y la CLI de administración

Siga los pasos a continuación para actualizar la configuración de su servidor JBoss EAP 6 para que se ejecute en JBoss EAP 7.

1. Antes de comenzar, repase [Copia de seguridad de datos importantes y revisión del estado del servidor](#). Contiene información importante sobre cómo asegurarse de que el servidor se encuentre en buen estado y de que se hagan copias de seguridad de los archivos correspondientes.
2. Inicie el servidor JBoss EAP 7 con la configuración de JBoss EAP 6.
 - a. Haga una copia de seguridad de los archivos de configuración del servidor JBoss EAP 7.

- b. Copie el archivo de configuración de la versión anterior en el directorio de JBoss EAP 7.

```
$ cp EAP6_HOME/standalone/configuration/standalone-full.xml
EAP7_HOME/standalone/configuration
```

- c. Navegue al directorio de instalación de JBoss EAP 7 e inicie el servidor con el **--start-mode=admin-only** argumento.

```
$ bin/standalone.sh -c standalone-full.xml --start-mode=admin-only
```

NOTA

Verás lo siguiente **org.jboss.as.controller.management-operation** ERRORES en el registro del servidor cuando inicia el servidor. Se esperan estos errores e indican que las configuraciones heredadas del subsistema deben eliminarse o migrarse a JBoss EAP 7.

- WFLYCTL0402: Subsystems [cmp] provided by legacy extension 'org.jboss.as.cmp' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [jacob] provided by legacy extension 'org.jboss.as.jacob' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [jaxr] provided by legacy extension 'org.jboss.as.jaxr' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [messaging] provided by legacy extension 'org.jboss.as.messaging' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [threads] provided by legacy extension 'org.jboss.as.threads' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.
- WFLYCTL0402: Subsystems [web] provided by legacy extension 'org.jboss.as.web' are not supported on servers running this version. Both the subsystem and the extension must be removed or migrated before the server will function.

3. Abra una nueva terminal, navegue hasta el directorio de instalación de JBoss EAP 7 y comience la CLI de administración utilizando la **--controller=remote://localhost:9999** argumentos.

```
$ bin/jboss-cli.sh --connect --controller=remote://localhost:9999
```

Migre los subsistemas JacORB, de mensajería y web

1. Para revisar los cambios de configuración que se realizarán en el subsistema antes de realizar la migración, ejecute el **describe-migration** operación.
los **describe-migration** operación utiliza la siguiente sintaxis.

```
/subsystem=SUBSYSTEM_NAME:describe-migration
```

El siguiente ejemplo describe los cambios de configuración que se realizan en JBoss EAP 6.4 **standalone-full.xml** archivo de configuración cuando se migra a JBoss EAP 7. Las entradas se eliminaron de la salida para mejorar la legibilidad y ahorrar espacio.

Ejemplo: operación de descripción-migración [Esto es una traducción automática]

```
/subsystem=messaging:describe-migration
{
  "outcome" => "success",
  "result" => {
    "migration-warnings" => [],
    "migration-operations" => [
      {
        "operation" => "add",
        "address" => [("extension" =>
"org.wildfly.extension.messaging-activemq")],
        "module" => "org.wildfly.extension.messaging-
activemq"
      },
      {
        "operation" => "add",
        "address" => [("subsystem" => "messaging-
activemq")]
      },
      <!-- *** Entries removed for readability *** -->
      {
        "operation" => "remove",
        "address" => [("subsystem" => "messaging")]
      },
      {
        "operation" => "remove",
        "address" => [("extension" =>
"org.jboss.as.messaging")]
      }
    ]
  }
}
```

2. Ejecuta el **migrate** operación para migrar la configuración del subsistema al subsistema que lo reemplaza en JBoss EAP 7. La operación utiliza la siguiente sintaxis.

```
/subsystem=SUBSYSTEM_NAME:migrate
```



NOTA

los **messaging** subsistema **describe-migration** y **migrate** las operaciones le permiten pasar un argumento para configurar el acceso por clientes heredados. Para obtener más información sobre la sintaxis del comando, consulte [Migración del subsistema de mensajería y compatibilidad directa](#).

3. Revise el resultado y el resultado del comando. Asegúrese de que la operación se haya completado correctamente y de que no haya entradas de "aviso de migración". Esto significa que la configuración de migración para el subsistema está completa.

Ejemplo: operación exitosa de migración sin advertencias [Esto es una traducción automática]

```
/subsystem=messaging:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => []}
}
```

Si ve entradas de "advertencias de migración" en el registro, esto indica que la migración de la configuración del servidor se completó con éxito, pero no pudo migrar todos los elementos y atributos. Debe seguir las sugerencias proporcionadas por las "advertencias de migración" y ejecutar comandos de CLI de administración adicionales para modificar esas configuraciones. El siguiente es un ejemplo de **migrate** operación que devuelve "advertencias de migración".

Ejemplo: migrar la operación con advertencias [Esto es una traducción automática]

```
/subsystem=messaging:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYMSG0080: Could not migrate attribute group-address from
resource [
  (\"subsystem\" => \"messaging-activemq\"),
  (\"server\" => \"default\"),
  (\"broadcast-group\" => \"groupB\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
  (\"subsystem\" => \"messaging-activemq\"),
  (\"server\" => \"default\"),
  (\"broadcast-group\" => \"groupB\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute local-bind-address
from resource [
  (\"subsystem\" => \"messaging-activemq\"),
  (\"server\" => \"default\"),
  (\"broadcast-group\" => \"groupA\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute local-bind-port
```

```

from resource [
    ("subsystem\" => \"messaging-activemq\"),
    ("server\" => \"default\"),
    ("broadcast-group\" => \"groupA\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-address from
resource [
    ("subsystem\" => \"messaging-activemq\"),
    ("server\" => \"default\"),
    ("broadcast-group\" => \"groupA\")
]. Use instead the socket-binding attribute to configure this
broadcast-group.",
    "WFLYMSG0080: Could not migrate attribute group-port from
resource [
    ("subsystem\" => \"messaging-activemq\"),
    ("server\" => \"default\"),
    ("broadcast-group\" => \"groupA\")
]. Use instead the socket-binding attribute to configure this
broadcast-group."
    ]]
}

```

NOTA

La lista de **migrate** y **describe-migration** advertencias para cada subsistema se encuentra en el [Material de referencia](#) al final de esta guía.

- [Advertencias de la operación de migración del subsistema Jacorb](#)
- Advertencias de la operación de migración del subsistema de mensajería [Esto es una traducción automática]
- Advertencias de la operación de migración del subsistema de mensajería [Esto es una traducción automática]

4. Revise el archivo de configuración del servidor para verificar que la extensión, el subsistema y el espacio de nombres se actualizaron y la configuración del subsistema existente se migró a JBoss EAP 7.

NOTA

Debe repetir este proceso para cada uno de los **jacorb**, **messaging**, y **web** subsistemas usando los siguientes comandos.

```

/subsystem=jacorb:migrate
/subsystem=messaging:migrate
/subsystem=web:migrate

```

5. Eliminar el **cmp**, **jaxr**, y **threads** subsistemas y extensiones de la configuración del servidor. Mientras aún está en el indicador de CLI de administración, elimine el obsoleto **cmp**, **jaxr**, y **threads** subsistemas mediante la ejecución de los siguientes comandos.

```

/subsystem=cmp:remove
/extension=org.jboss.as.cmp:remove
/subsystem=jaxr:remove
/extension=org.jboss.as.jaxr:remove
/subsystem=threads:remove
/extension=org.jboss.as.threads:remove

```



IMPORTANTE

Debe migrar el **messaging**, **jacorb**, y **web** subsistemas y eliminar el **cmp**, **jaxr**, y **threads** extensiones y subsistemas antes de que pueda reiniciar el servidor para una operación normal. Si necesita reiniciar el servidor antes de completar este proceso, asegúrese de incluir **--start-mode=admin-only** argumento en la línea de comando de inicio del servidor. Esto le permite continuar con los cambios de configuración.

4.4. CAMBIOS DE REGISTRO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.4.1. Cambios en el prefijo del mensaje de registro [Esto es una traducción automática]

Los mensajes de registro están prefijados con el código de proyecto para el subsistema que informa el mensaje. Los prefijos de todos los mensajes de registro han cambiado en JBoss EAP 7.

Para obtener una lista completa de los nuevos prefijos de código de proyecto de mensaje de registro utilizados en JBoss EAP 7, consulte [Códigos de proyecto utilizados en JBoss EAP](#) en el *Guía de desarrollo*.

4.4.2. Cambios en el controlador de consola de Root Logger [Esto es una traducción automática]

El registrador de raíz JBoss EAP 7.0 incluía un controlador de registro de consola para todos los perfiles de servidor de dominio y para todos los perfiles independientes predeterminados, excepto el **standalone-full-ha** perfil. El registrador de raíz JBoss EAP 7.1 ya no incluye un controlador de registro de consola para los perfiles de dominio administrado. El controlador de host y el controlador de proceso se conectan a la consola de forma predeterminada. Para lograr la misma funcionalidad que se proporcionó en JBoss EAP 7.0, consulte [Configurar un controlador de registro de consola](#) en el *Guía de configuración* para JBoss EAP.

4.5. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.5.1. Reemplace el subsistema web con Undertow [Esto es una traducción automática]

Undertow reemplaza a JBoss Web como el servidor web en JBoss EAP 7. Esto significa que el legado **web** la configuración del subsistema debe migrarse al nuevo JBoss EAP 7 **undertow** configuración del subsistema.

- los **urn:jboss:domain:web:2.2** El espacio de nombre de configuración del subsistema en el archivo de configuración del servidor ha sido reemplazado por **urn:jboss:domain:undertow:4.0** espacio de nombres
- los **org.jboss.as.web** módulo de extensión, ubicado en **EAP_HOME/modules/system/layers/base/**, ha sido reemplazado con **org.wildfly.extension.undertow** módulo de extensión.

Puede usar la CLI de administración **migrate** operación para migrar el **web** subsistema a **undertow** en el archivo de configuración del servidor. Sin embargo, tenga en cuenta que esta operación no puede migrar todas las configuraciones del subsistema web de JBoss. Si ve entradas de "advertencia de migración", debe ejecutar comandos de CLI de administración adicionales para migrar esas configuraciones a Undertow. Para obtener más información acerca de la CLI de administración **migrate** operación, ver [Gestión de la migración de la CLI Operación](#).

El siguiente es un ejemplo de lo predeterminado **web** Configuración del subsistema en JBoss EAP 6.

```
<subsystem xmlns="urn:jboss:domain:web:2.2" default-virtual-
server="default-host" native="false">
  <connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
  </virtual-server>
</subsystem>
```

El siguiente es un ejemplo de lo predeterminado **undertow** configuración del subsistema en JBoss EAP 7.

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http" redirect-
socket="https"/>
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
    <websockets/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="${jboss.home.dir}/welcome-
content"/>
  </handlers>
  <filters>
    <response-header name="server-header" header-name="Server" header-
value="JBoss-EAP/7"/>
    <response-header name="x-powered-by-header" header-name="X-
```

```

    Powered-By" header-value="Undertow/1"/>
  </filters>
</subsystem>

```

4.5.2. Migrar las condiciones de reescritura de JBoss Web [Esto es una traducción automática]

La CLI de gestión **migrate** la operación no puede migrar automáticamente las condiciones de reescritura. Se informan como "advertencias de migración" y debe migrarlas manualmente. Puede crear la configuración equivalente en JBoss EAP 7 utilizando [Undertow predice los atributos y manipuladores](#).

El siguiente es un ejemplo de **web** configuración del subsistema en JBoss EAP 6 que incluye **rewrite** configuración.

```

<subsystem xmlns="urn:jboss:domain:web:2.2" default-virtual-
server="default" native="false">
  <virtual-server name="default" enable-welcome-root="true">
    <alias name="localhost"/>
    <rewrite name="test" pattern="(.*)/toberewritten/(.*)"
substitution="$1/rewritten/$2" flags="NC"/>
    <rewrite name="test2" pattern="(.*)" substitution="-" flags="F">
      <condition name="get" test="%{REQUEST_METHOD}"
pattern="GET"/>
      <condition name="andCond" test="%{REQUEST_URI}"
pattern=".*index.html" flags="NC"/>
    </rewrite>
  </virtual-server>
</subsystem>

```

Siga el [Gestión de la migración de la CLI Operación](#) instrucciones para iniciar su servidor y la CLI de administración, luego migre el **web** archivo de configuración del subsistema usando el siguiente comando.

```
/subsystem=web:migrate
```

Las siguientes "advertencias de migración" se informan cuando ejecuta el **migrate** operación en la configuración anterior.

```

/subsystem=web:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYWEB0002: Could not migrate resource {
\"pattern\" => \"(.*)\",
\"substitution\" => \"-\",
\"flags\" => \"F\",
\"operation\" => \"add\",
\"address\" => [
  (\"subsystem\" => \"web\"),
  (\"virtual-server\" => \"default-host\"),
  (\"rewrite\" => \"test2\")
]
}"],
    "WFLYWEB0002: Could not migrate resource {

```



```

    \"test\" => \"%{REQUEST_METHOD}\",
    \"pattern\" => \"GET\",
    \"flags\" => undefined,
    \"operation\" => \"add\",
    \"address\" => [
        (\"subsystem\" => \"web\"),
        (\"virtual-server\" => \"default-host\"),
        (\"rewrite\" => \"test2\"),
        (\"condition\" => \"get\")
    ]
},
    \"WFLYWEB0002: Could not migrate resource {
    \"test\" => \"%{REQUEST_URI}\",
    \"pattern\" => \".*index.html\",
    \"flags\" => \"NC\",
    \"operation\" => \"add\",
    \"address\" => [
        (\"subsystem\" => \"web\"),
        (\"virtual-server\" => \"default-host\"),
        (\"rewrite\" => \"test2\"),
        (\"condition\" => \"andCond\")
    ]
}
    ]}
}

```

Revise el archivo de configuración del servidor y verá la siguiente configuración para **undertow** subsistema.



NOTA

Configuración del lado del servidor y carga de clases [Esto es una traducción automática]

```

<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="http" socket-binding="http"/>
    <host name="default-host" alias="localhost, example.com">
      <location name="/" handler="welcome-content"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="${jboss.home.dir}/welcome-
content"/>
  </handlers>
</subsystem>

```

Use la CLI de administración para crear el filtro para reemplazar la configuración de reescritura en el **undertow** subsistema. Debería ver "{" resultado "=>" success "}" para cada comando.

```

# Create the filters
/subsystem=undertow/configuration=filter/expression-

```

```

filter="test1":add(expression="path('(.*)/toberewritten/(.*)') ->
rewrite('$1/rewritten/$2')")
/subsystem=undertow/configuration=filter/expression-
filter="test2":add(expression="method('GET') and path('.*index.html') ->
response-code(403)")

# Add the filters to the default server
/subsystem=undertow/server=default-server/host=default-host/filter-
ref="test1":add
/subsystem=undertow/server=default-server/host=default-host/filter-
ref="test2":add

```

Revise el archivo de configuración del servidor actualizado. El subsistema web JBoss ahora se migró completamente y se configuró en **undertow** subsistema.

```

<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="http" socket-binding="http"/>
    <host name="default-host" alias="localhost, example.com">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="test1"/>
      <filter-ref name="test2"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="${jboss.home.dir}/welcome-
content"/>
  </handlers>
  <filters>
    <expression-filter name="test1"
expression="path('(.*)/toberewritten/(.*)') ->
rewrite('$1/rewritten/$2')"/>
    <expression-filter name="test2" expression="method('GET') and
path('.*index.html') -> response-code(403)"/>
  </filters>
</subsystem>

```

Para obtener más información acerca de cómo configurar filtros y controladores mediante CLI de administración, consulte [Configurar el servidor web](#) en el JBoss EAP 7 *Guía de configuración*.

4.5.3. Migrar las propiedades del sistema web de JBoss [Esto es una traducción automática]

En el lanzamiento anterior de JBoss EAP, las propiedades del sistema se podían usar para modificar el comportamiento predeterminado de JBoss Web. Para obtener información sobre cómo configurar el mismo comportamiento en Undertow, consulte [Referencia de migración de propiedades del sistema web JBoss](#)

4.5.4. Actualizar el patrón de encabezado del registro de acceso [Esto es una traducción automática]

Cuando migra de JBoss EAP 6.4 a JBoss EAP 7.1, puede encontrar que los registros de acceso ya no escriben los valores esperados de "Referer" y "User-agent". Esto se debe a que JBoss Web, que se incluyó en JBoss EAP 6.4, utilizó un patrón de `%{headername}i` en el `access-log` para registrar un encabezado entrante.

Ejemplo: Formato de registro de acceso en JBoss EAP 6.4 [Esto es una traducción automática]

```
<access-log pattern="%h %l %u %t &quot;%T sec&quot; &quot;%r&quot; %s %b
&quot;%{Referer}i&quot; &quot;%{User-agent}i&quot;"/>
```

Con el cambio para usar Undertow en JBoss EAP 7.1, el patrón para un encabezado entrante ha cambiado a `%{i, headername}`.

Ejemplo: acceso al encabezado del formato en JBoss EAP 7.1 [Esto es una traducción automática]

```
<access-log pattern="%h %l %u %t &quot;%T sec&quot; &quot;%r&quot; %s %b
&quot;%{i,Referer}&quot; &quot;%{i,User-Agent}&quot;"/>
```

4.5.5. Migrar válvulas globales [Esto es una traducción automática]

Versiones anteriores de válvulas compatibles con JBoss EAP. Las válvulas son clases personalizadas insertadas en la interconexión de procesamiento de solicitudes para una aplicación antes de filtros de servlets para realizar cambios en la solicitud o realizar un procesamiento adicional.

- Las válvulas globales se insertan en la canalización de procesamiento de solicitud de todas las aplicaciones desplegadas y se configuran en el archivo de configuración del servidor.
- Las válvulas del autenticador autentican las credenciales de la solicitud.
- Las válvulas de aplicación personalizadas se crearon al extender el `org.apache.catalina.valves.ValveBase` clase y configurado en el `<valve>` elemento de la `jboss-web.xml` archivo descriptor. Estas válvulas deben migrarse manualmente.

Esta sección describe cómo migrar las válvulas globales. La migración de válvulas personalizadas y autenticadoras están cubiertas en [Migrar válvulas de aplicaciones personalizadas](#) sección de esta guía.

Undertow, que reemplaza a JBoss Web en JBoss EAP 7, no admite válvulas globales; sin embargo, debe poder lograr una funcionalidad similar mediante el uso de controladores Undertow. Undertow incluye una serie de controladores incorporados que proporcionan una funcionalidad común. También proporciona la capacidad de crear controladores personalizados, que se pueden usar para reemplazar la funcionalidad de válvula personalizada.

Si su aplicación usa válvulas, debe reemplazarlas con el código del controlador Undertow apropiado para lograr la misma funcionalidad cuando migre a JBoss EAP 7.

Para obtener más información sobre cómo configurar controladores, consulte [Configurando Manejadores](#) en el JBoss EAP 7 *Guía de configuración*.

Para obtener más información acerca de cómo configurar filtros, consulte [Configurar filtros](#) en el JBoss EAP 7 *Guía de configuración*.

Migrar válvulas globales [Esto es una traducción automática]

La siguiente tabla enumera las válvulas proporcionadas por JBoss Web en la versión anterior de JBoss EAP y el correspondiente controlador integrado de Undertow. Las válvulas JBoss Web están ubicadas en el **org.apache.catalina.valves** paquete.

Tabla 4.2. Asignación de válvulas a manipuladores [Esto es una traducción automática]

Válvula	Controlador
AccessLogValve	io.undertow.server.handlers.accesslog.AccessLogHandler
CrawlerSessionManagerValve	io.undertow.servlet.handlers.CrawlerSessionManagerHandler
ExtendedAccessLogValve	io.undertow.server.handlers.accesslog.AccessLogHandler
JDBCAccessLogValve	Ver el JDBCAccessLogValve Procedimiento de migración manual a continuación para obtener instrucciones.
RemoteAddrValve	io.undertow.server.handlers.IPAddressAccessControlHandler
RemoteHostValve	io.undertow.server.handlers.AccessControlListHandler
RemoteIpValve	io.undertow.server.handlers.ProxyPeerAddressHandler
RequestDumperValve	io.undertow.server.handlers.RequestDumpingHandler
RewriteValve	Ver Migrar las condiciones de reescritura de JBoss Web para obtener instrucciones para migrar estas válvulas manualmente.
StuckThreadDetectionValve	io.undertow.server.handlers.StuckThreadDetectionHandler

Puede usar la CLI de administración **migrate** operación para migrar automáticamente válvulas globales que cumplan los siguientes criterios:

- Están limitados a las válvulas enumeradas en la tabla anterior que no requieren procesamiento manual.
- Deben definirse en el **web** subsistema del archivo de configuración del servidor.

Para obtener más información acerca de la CLI de administración **migrate** operación, ver [Gestión de la migración de la CLI Operación](#).

JDBCAccessLogValve Procedimiento de migración manual

los **org.apache.catalina.valves.JDBCAccessLogValve** la válvula es una excepción a la regla y no se puede migrar automáticamente a **io.undertow.server.handlers.JDBCLogHandler**. Siga los pasos a continuación para migrar la siguiente válvula de ejemplo.

```
<valve name="jdbc" module="org.jboss.as.web" class-
name="org.apache.catalina.valves.JDBCAccessLogValve">
  <param param-name="driverName" param-value="com.mysql.jdbc.Driver" />
  <param param-name="connectionName" param-value="root" />
  <param param-name="connectionPassword" param-value="password" />
```

```

    <param param-name="connectionURL" param-
value="jdbc:mysql://localhost:3306/wildfly?
zeroDateTimeBehavior=convertToNull" />
    <param param-name="format" param-value="combined" />
</value>

```

1. Cree un módulo de controlador para la base de datos que almacenará las entradas de registro.
2. Configure el origen de datos para la base de datos y agregue el controlador a la lista de controladores disponibles en el **datasources** subsistema.

```

<datasources>
  <datasource jndi-name="java:jboss/datasources/accessLogDS" pool-
name="accessLogDS" enabled="true" use-java-context="true">
    <connection-url>jdbc:mysql://localhost:3306/wildfly?
zeroDateTimeBehavior=convertToNull</connection-url>
    <driver>mysql</driver>
    <security>
      <user-name>root</user-name>
      <password>Password1!</password>
    </security>
  </datasource>
  ...
<drivers>
  <driver name="mysql" module="com.mysql">
    <driver-class>com.mysql.jdbc.Driver</driver-class>
  </driver>
  ...
</drivers>
</datasources>

```

3. Configurar un **expression-filter** en el **undertow** subsistema con la siguiente expresión: **jdbc-access-log(datasource=DATASOURCE_JNDI_NAME)**.

```

<filters>
  <expression-filter name="jdbc-access" expression="jdbc-access-
log(datasource='java:jboss/datasources/accessLogDS')" />
  ...
</filters>

```

4.5.6. Cambios en el comportamiento Set-Cookie [Esto es una traducción automática]

Especificaciones anteriores para **Set-Cookie** La sintaxis del encabezado de respuesta HTTP, por ejemplo RFC2109 y RFC2965, permitía espacios en blanco y otros caracteres separadores en el valor de la cookie cuando se citaba el valor de la cookie. JBoss Web en JBoss EAP 6.4 cumplió con las especificaciones anteriores y citó automáticamente un valor de cookie cuando contenía cualquier carácter separador.

los [RFC6265](#) especificación para **Set-Cookie** La sintaxis del encabezado de respuesta HTTP indica que los valores de las cookies en el **Set-Cookie** El encabezado de respuesta debe ajustarse a las restricciones de gramática específicas. Por ejemplo, deben ser caracteres US-ASCII, pero no pueden incluir CTRL (controles), espacios en blanco, comillas dobles, comas, puntos y comas o caracteres de barra diagonal inversa.

En JBoss EAP 7.0, antes del parche acumulativo [Red Hat JBoss Enterprise Application Platform 7.0 Actualización 08](#), Undertow no restringe estos caracteres no válidos y no cita las cookies que contenían los caracteres excluidos. Si aplica este parche acumulativo o un parche acumulativo más reciente, puede habilitar la validación de cookies conforme con RFC6265 configurando el **io.undertow.cookie.DEFAULT_ENABLE_RFC6265_COOKIE_VALIDATION** propiedad del sistema a **true**.

En JBoss EAP 7.1, de forma predeterminada, Undertow no habilita la validación de cookies conforme con RFC6265. Cita cookies que contienen los caracteres excluidos. En JBoss EAP 7.1, no puede usar el **io.undertow.cookie.DEFAULT_ENABLE_RFC6265_COOKIE_VALIDATION** propiedad del sistema para habilitar la validación de cookies conforme con RFC6265. En su lugar, habilita la validación de cookies conforme a RFC6265 para un oyente HTTP, HTTPS o AJP configurando el **rfc6265-cookie-validation** atributo del oyente a **true**. El valor predeterminado para este atributo es **false**. El siguiente ejemplo habilita la validación de cookies compatibles con RFC6265 para el escucha HTTP.

```
/subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=rfc6265-cookie-validation,value=true)
```

4.5.7. Cambios en el comportamiento de llamada del método HTTP [Esto es una traducción automática]

JBoss EAP 6.4, que incluía JBoss Web como servidor web, permitió HTTP **TRACE** método llamadas por defecto.

Undertow, que reemplaza a JBoss Web como servidor web en JBoss EAP 7, no permite HTTP **TRACE** método llamadas por defecto. Esta configuración se configura usando el **disallowed-methods** atributo de la **http-listener** elemento en el **undertow** subsistema. Esto se puede confirmar revisando el resultado de la siguiente **read-resource** mando. Tenga en cuenta que el valor para el **disallowed-methods** atributo es **["TRACE"]**.

```
/subsystem=undertow/server=default-server/http-listener=default:read-resource
{
  "outcome" => "success",
  "result" => {
    "allow-encoded-slash" => false,
    "allow-equals-in-cookie-value" => false,
    "always-set-keep-alive" => true,
    "buffer-pipelined-data" => false,
    "buffer-pool" => "default",
    "certificate-forwarding" => false,
    "decode-url" => true,
    "disallowed-methods" => ["TRACE"],
    ...
  }
}
```

Para habilitar HTTP **TRACE** llamadas de método en JBoss EAP 7 y posteriores, debe eliminar la entrada "TRACE" del **disallowed-methods** lista de atributos ejecutando el siguiente comando.

```
/subsystem=undertow/server=default-server/http-listener=default:list-remove(name=disallowed-methods,value="TRACE")
```

Cuando ejecutas el **read-resource** comando nuevamente, notará el **TRACE** la llamada al método ya no se encuentra en la lista de métodos no permitidos.

```
/subsystem=undertow/server=default-server/http-listener=default:read-
resource
{
    "outcome" => "success",
    "result" => {
        "allow-encoded-slash" => false,
        "allow-equals-in-cookie-value" => false,
        "always-set-keep-alive" => true,
        "buffer-pipelined-data" => false,
        "buffer-pool" => "default",
        "certificate-forwarding" => false,
        "decode-url" => true,
        "disallowed-methods" => [],
        ...
    }
}
```

Para obtener más información sobre el comportamiento predeterminado de los métodos HTTP, consulte [Comportamiento predeterminado de los métodos HTTP](#) en el JBoss EAP *Guía de configuración*.

4.5.8. Cambios en el comportamiento del módulo web predeterminado en JBoss EAP 7.1 [Esto es una traducción automática]

En JBoss EAP 7.0, el contexto raíz de una aplicación web estaba deshabilitado por defecto en `mod_cluster`.

Este ya no es el caso en JBoss EAP 7.1. Esto puede tener consecuencias inesperadas si espera que el contexto raíz esté deshabilitado. Por ejemplo, las solicitudes pueden encaminarse erróneamente a nodos no deseados o una aplicación privada que no debe estar expuesta puede ser accesible inadvertidamente a través de un proxy público. Las ubicaciones de retroceso también ahora están registradas con el equilibrador de carga `mod_cluster` automáticamente a menos que estén explícitamente excluidas.

Use el siguiente comando CLI de administración para excluir ROOT del **modcluster** configuración del subsistema

```
/subsystem=modcluster/mod-cluster-config=configuration:write-
attribute(name=excluded-contexts,value=ROOT)
```

Utilice el siguiente comando CLI de administración para deshabilitar la aplicación web de bienvenida predeterminada.

```
/subsystem=undertow/server=default-server/host=default-
host/location=\/:remove
/subsystem=undertow/configuration=handler/file=welcome-content:remove
reload
```

Para obtener más información sobre cómo configurar la aplicación web de bienvenida predeterminada, consulte [Configurar la aplicación web de bienvenida predeterminada](#) en el *Guía de desarrollo* para JBoss EAP.

4.6. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR JGROUPS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.6.1. JGroups se predetermina a una interfaz de red privada [Esto es una traducción automática]

En la configuración predeterminada de JBoss EAP 6, JGroups utilizó el **public** interfaz definida en el **<interfaces>** sección del archivo de configuración del servidor.

Debido a que es una práctica recomendada utilizar una interfaz de red dedicada, JGroups ahora usa por defecto el nuevo **private** interfaz que se define en el **<interfaces>** sección del archivo de configuración del servidor en JBoss EAP 7.

4.6.2. Cambios de canales de JGroups [Esto es una traducción automática]

JGroups proporciona soporte de comunicación grupal para servicios de HA en forma de canales de JGroups. JBoss EAP 7 presenta **<channel>** elementos a la **jgroups** subsistema en el archivo de configuración del servidor. Puede agregar, eliminar o cambiar la configuración de los canales de JGroups utilizando la CLI de administración.

Para obtener más información acerca de cómo configurar JGroups, consulte [Comunicación de clúster con JGroups](#) en el JBoss EAP *Guía de configuración*.

4.7. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR INFINISPAN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.7.1. Cambios en la configuración de caché predeterminada de Infinispan [Esto es una traducción automática]

En JBoss EAP 6, los cachés agrupados por defecto para la replicación de sesión web y la replicación EJB se replicaron **ASYN** cachés Esto ha cambiado en JBoss EAP 7. Los cachés agrupados por defecto están ahora distribuidos **ASYN** cachés Los cachés replicados ya no están configurados de manera predeterminada. Ver [Configurar el modo de caché](#) en el JBoss EAP *Guía de configuración* para obtener información acerca de cómo agregar un caché replicado y hacerlo predeterminado.

Esto solo lo afecta cuando usa la nueva configuración predeterminada de JBoss EAP 7. Si migra la configuración de JBoss EAP 6, la configuración de **infinispan** subsistema será preservado.

4.7.2. Cambios en la estrategia de caché Infinispan [Esto es una traducción automática]

El comportamiento de **ASYN** la estrategia de caché ha cambiado en JBoss EAP 7.

En JBoss EAP 6, **ASYN** las lecturas de caché estaban libres de bloqueo. Aunque nunca bloquearían, eran propensos a lecturas sucias de datos obsoletos, por ejemplo, en la conmutación por error. Esto se debe a que permitiría que las solicitudes posteriores para el mismo usuario se inicien antes de que se complete la solicitud anterior. Esta permisividad no es aceptable cuando se utiliza el modo distribuido, ya que los cambios en la topología del clúster pueden afectar la afinidad de la sesión y dar como resultado fácilmente datos obsoletos.

En JBoss EAP 7, **ASYN** las lecturas de caché requieren bloqueos. Como ahora bloquean nuevas solicitudes del mismo usuario hasta que finaliza la replicación anterior, se evitan las lecturas sucias.

4.7.3. Configuración de caché de beans de sesión de estado personalizado para pasivación [Esto es una traducción automática]

Tenga en cuenta las siguientes restricciones al configurar una memoria caché de sesión de estado con estado personalizado (SFSB) para la pasivación en JBoss EAP 7.1.

- los **idle-timeout** atributo, que se configura en **infinispan passivation-store** del **ejb3** subsistema, está en desuso en JBoss EAP 7.1. JBoss EAP 6.4 apoyó la pasivación ansiosa, pasivado según el **idle-timeout** valor. JBoss EAP 7.1 admite pasivación lenta, pasivación cuando el **max-size** umbral se alcanza.
- En JBoss EAP 7.1, el nombre del clúster utilizado por el cliente EJB está determinado por el nombre real del clúster del canal, tal como se configuró en el **jgroups** subsistema.
- JBoss EAP 7.1 todavía le permite configurar el **max-size** atributo para controlar el umbral de pasivación.
- No debe configurar el desalojo o la expiración en la configuración de su caché EJB.
 - Debe configurar el desalojo utilizando el **max-size** atributo de la **passivation-store** en el **ejb3** subsistema.
 - Debe configurar la caducidad utilizando el **@StatefulTimeout** anotación en el código fuente SFSB Java o especificando un **stateful-timeout** valor en el **ejb-jar.xml** archivo.

4.7.4. Cambios en el transporte de contenedor de caché Infinispan [Esto es una traducción automática]

Un cambio en el comportamiento entre JBoss EAP 7.0 y JBoss EAP 7.1 requiere que cualquier actualización del protocolo de transporte del contenedor de caché se realice en modo batch o utilizando un encabezado especial. Este cambio en el comportamiento también afecta las herramientas que se utilizan para administrar el servidor JBoss EAP.

El siguiente es un ejemplo de los comandos CLI de gestión utilizados para configurar el protocolo de transporte de contenedor de caché en JBoss EAP 7.0.

```
/subsystem=infinispan/cache-container=my:add()
/subsystem=infinispan/cache-container=my/transport=jgroups:add()
/subsystem=infinispan/cache-container=my/invalidation-
cache=mycache:add(mode=SYNC)
```

El siguiente es un ejemplo de los comandos CLI de administración necesarios para realizar la misma configuración en JBoss EAP 7.1. Tenga en cuenta que los comandos se ejecutan en modo por lotes.

```
batch
/subsystem=infinispan/cache-container=my:add()
/subsystem=infinispan/cache-container=my/transport=jgroups:add()
/subsystem=infinispan/cache-container=my/invalidation-
cache=mycache:add(mode=SYNC)
run-batch
```

Si prefiere no usar el modo por lotes, puede especificar el encabezado de operación **allow-resource-service-restart=true** al definir el transporte. Tenga en cuenta que esto reinicia el servicio para que las operaciones surtan efecto y algunos servicios pueden dejar de funcionar hasta que

se reinicie el servicio.

Si usa scripts para actualizar el protocolo de transporte del contenedor de caché, asegúrese de revisarlos y agregar el modo por lotes.

4.8. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR EJB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

No hay **migrate** operación para el **ejb3** subsistema, por lo que si usa la CLI de administración **migrate** para actualizar sus otras configuraciones existentes de JBoss EAP 6.4, tenga en cuenta que **ejb3** la configuración del subsistema no se migra. Porque la configuración de **ejb3** El subsistema es ligeramente diferente en JBoss EAP 7.1 que en JBoss EAP 6.4, es posible que vea excepciones en el registro del servidor cuando despliegue sus aplicaciones EJB.



IMPORTANTE

Si usa la herramienta de migración de servidores JBoss para actualizar su configuración de servidor, **ejb3** el subsistema debería estar configurado correctamente y no debería aparecer ningún problema cuando implemente sus aplicaciones EJB. Para obtener información sobre cómo configurar y ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).

DuplicateServiceException en el registro del servidor [Esto es una traducción automática]

El seguimiento **DuplicateServiceException** es causado por cambios de caché en JBoss EAP 7.

DuplicateServiceException en el registro del servidor [Esto es una traducción automática]

```
ERROR [org.jboss.msc.service.fail] (MSC service thread 1-3) MSC000001:
Failed to start service jboss.deployment.unit."mdb-1.0-
SNAPSHOT.jar".cache-dependencies-installer:
org.jboss.msc.service.StartException in service
jboss.deployment.unit."mdb-1.0-SNAPSHOT.jar".cache-dependencies-installer:
Failed to start service
...
Caused by: org.jboss.msc.service.DuplicateServiceException: Service
jboss.infinispan.ejb."mdb-1.0-SNAPSHOT.jar".config is already registered
```

Debe volver a configurar la memoria caché para resolver este error.

1. Sigue las instrucciones para [Inicie el servidor y la CLI de administración](#).
2. Emita los siguientes comandos para volver a configurar el almacenamiento en caché en **ejb3** subsistema.

```
/subsystem=ejb3/file-passivation-store=file:remove
/subsystem=ejb3/cluster-passivation-store=infinispan:remove
/subsystem=ejb3/passivation-store=infinispan:add(cache-
container=ejb, max-size=10000)

/subsystem=ejb3/cache=passivating:remove
```

```
/subsystem=ejb3/cache=clustered:remove
/subsystem=ejb3/cache=distributable:add(passivation-
store=infinispan, aliases=[passivating, clustered])
```

4.9. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 7, ActiveMQ Artemis reemplaza a HornetQ como el proveedor de soporte JMS. Esta sección describe cómo migrar la configuración y los datos de mensajería relacionados.

4.9.1. Cambios en la configuración del servidor del subsistema de mensajería [Esto es una traducción automática]

los `org.jboss.as.messaging` módulo de extensión, ubicado en `EAP_HOME/modules/system/layers/base/`, ha sido reemplazado por `org.wildfly.extension.messaging-activemq` módulo de extensión.

los `urn:jboss:domain:messaging:3.0` El espacio de nombre de configuración del subsistema ha sido reemplazado por `urn:jboss:domain:messaging-activemq:2.0` espacio de nombres

Cambios en la administración de JMX [Esto es una traducción automática]

En la mayoría de los casos, se hizo un esfuerzo para mantener los nombres de elementos y atributos lo más similares posibles a los utilizados en lanzamientos anteriores. La siguiente tabla enumera algunos de los cambios.

Tabla 4.3. Asignación de atributos de mensajería [Esto es una traducción automática]

Nombre de HornetQ	Nombre ActiveMQ
hornetq-server	Servidor
hornetq-serverType	serverType
conectores	conector
discovery-group-name	discovery-group

Las operaciones de gestión invocadas en el nuevo `messaging-activemq` subsistema ha cambiado desde `/subsystem=messaging/hornetq-server=` a `/subsystem=messaging-activemq/server=`.

Puede migrar un JBoss EAP existente 6 `messaging` configuración del subsistema a `messaging-activemq` subsistema en un servidor JBoss EAP 7 invocando su `migrate` operación.

```
/subsystem=messaging:migrate
```

Antes de ejecutar el `migrate` operación, puede invocar el `describe-migration` operación para revisar la lista de operaciones de administración que se realizarán para migrar del JBoss EAP existente 6 `messaging` configuración del subsistema a `messaging-activemq` subsistema en el servidor JBoss EAP 7.

```
/subsystem=messaging:describe-migration
```

los **migrate** y **describe-migration** las operaciones también muestran una lista de **migration-warnings** para recursos o atributos que no se pueden migrar automáticamente.

Advertencias de la operación de migración del subsistema de mensajería [Esto es una traducción automática]

los **describe-migration** y **migrate** operaciones para el **messaging** subsistema proporciona un argumento de configuración adicional. Si desea configurar la mensajería para permitir que los clientes heredados de JBoss EAP 6 se conecten al servidor JBoss EAP 7, puede agregar la booleana **add-legacy-entries** argumento a la **describe-migration** o **migrate** operación de la siguiente manera.

```
/subsystem=messaging:describe-migration(add-legacy-entries=true)
/subsystem=messaging:migrate(add-legacy-entries=true)
```

Si el argumento booleano **add-legacy-entries** se establece en **true**, el **messaging-activemq** subsistema crea el **legacy-connection-factory** recurso y agrega **legacy-entries** al **jdbc-queue** y **jdbc-topic** recursos.

Si el argumento booleano **add-legacy-entries** se establece en **false**, no se crean recursos heredados en **messaging-activemq** el subsistema y los clientes JMS heredados no podrán comunicarse con los servidores JBoss EAP 7. Este es el valor predeterminado.

Para obtener más información sobre la compatibilidad directa e inversa, consulte la [Compatibilidad hacia atrás y adelante](#) en *Configurando la Mensajería* para JBoss EAP.

Para obtener más información acerca de la CLI de administración **migrate** y **describe-migration** operaciones, ver [Gestión de la migración de la CLI Operación](#).

Cambio en el comportamiento del atributo forward-when-no-consumers

El comportamiento de **forward-when-no-consumers** atributo ha cambiado en JBoss EAP 7.

En JBoss EAP 6, cuando **forward-when-no-consumers** fue configurado para **false** y no había consumidores en un clúster, los mensajes se redistribuyeron a todos los nodos de un clúster.

Este comportamiento ha cambiado en JBoss EAP 7. Cuando **forward-when-no-consumers** se establece en **false** y no hay consumidores en un clúster, los mensajes no se redistribuyen. En cambio, se mantienen en el nodo original al que se enviaron.

Cambio en la política de equilibrio de carga del clúster predeterminado

La política de equilibrio de carga del clúster predeterminada ha cambiado en JBoss EAP 7.

En JBoss EAP 6, la política de equilibrio de carga del clúster por defecto era similar a **STRICT**, que es como establecer el legado **forward-when-no-consumers** parámetro a **true**. En JBoss EAP 7, el valor predeterminado es ahora **ON_DEMAND**, que es como establecer el legado **forward-when-no-consumers** parámetro a **false**. Para obtener más información acerca de estas configuraciones, consulte [Atributos de conexión de clúster](#) en *Configurando la Mensajería* para JBoss EAP.

Cambios en la configuración del servidor del subsistema de mensajería [Esto es una traducción automática]

La configuración XML ha cambiado significativamente con el nuevo **messaging-activemq** subsistema, y ahora proporciona un esquema XML más consistente con otros subsistemas JBoss EAP.

Se recomienda encarecidamente que no intente modificar JBoss EAP **messaging** configuración XML

del subsistema para ajustarse a la nueva **messaging-activemq** subsistema. En cambio, invoque el subsistema heredado **migrate** operación. Esta operación escribirá la configuración XML del nuevo **messaging-activemq** subsistema como parte de su ejecución.

4.9.2. Migrar datos de mensajería [Esto es una traducción automática]

Puede usar uno de los siguientes enfoques para migrar los datos de mensajería de una versión anterior a la versión actual de JBoss EAP.

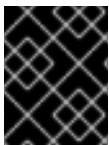
- Para sistemas de mensajería basados en archivos, puede migrar datos de mensajería desde JBoss EAP 6.4 o desde JBoss EAP 7.0 a JBoss EAP 7.1. [usando el método de exportación e importación](#). Con este método, puede exportar los datos de mensajería de la versión anterior e importarlos mediante la CLI de administración. **import-journal** operación. Tenga en cuenta que puede usar este enfoque solo para sistemas de mensajería basados en archivos.
- Puede migrar datos de mensajería de JBoss EAP 6.4 a JBoss EAP 7.1 por [configurar un puente JMS](#). Puede usar este enfoque tanto para sistemas de mensajería basados en archivos como JDBC.

Debido al cambio de HornetQ a ActiveMQ Artemis como proveedor de soporte JMS, tanto el formato como la ubicación de los datos de mensajería cambiaron en JBoss EAP 7.0 y posterior. Ver [Asignación de nombres de carpeta de mensajería](#) para detalles de los cambios en los nombres y ubicaciones de las carpetas de datos de mensajería entre las versiones 6.4 y 7.x.

4.9.2.1. Migrar datos de mensajería mediante exportación e importación [Esto es una traducción automática]

Con este enfoque, exporta los datos de mensajería de una versión anterior a un archivo XML, y luego importa ese archivo usando el **import-journal** operación.

1. Exportar los datos de mensajería a un archivo XML.
 - [Exportar datos de mensajes de JBoss EAP 6.4.](#)
 - [Exportar datos de mensajes de JBoss EAP 7.0.](#)
2. [Importe los datos de mensajería formateados en XML.](#)



IMPORTANTE

No puede usar el método de exportación e importación para mover datos de mensajería entre sistemas que usan un diario basado en JDBC para almacenamiento.

Ejemplo: Formato de registro de acceso en JBoss EAP 6.4 [Esto es una traducción automática]

Debido al cambio de HornetQ a ActiveMQ Artemis como proveedor de soporte JMS, tanto el formato como la ubicación de los datos de mensajería cambiaron en JBoss EAP 7.0 y posterior.

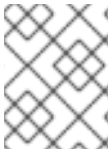
Para exportar datos de mensajería de JBoss EAP 6.4, debe usar HornetQ **exporter** utilidad. El HornetQ **exporter** la utilidad genera y exporta los datos de mensajería de JBoss EAP 6.4 a un archivo XML. Este comando requiere que especifique las rutas a los JAR HornetQ requeridos que se enviaron con JBoss EAP 6.4, pase las rutas a **messagingbindings/**, **messagingjournal/**, **messagingpaging/**, y **messaginglargemessages/** carpetas de la versión anterior como argumentos, y especifique un archivo de salida en el que escribir los datos XML exportados.

La siguiente es la sintaxis requerida por HornetQ **exporter** utilidad.

```
$ java -jar -mp MODULE_PATH org.hornetq.exporter
MESSAGING_BINDINGS_DIRECTORY MESSAGING_JOURNAL_DIRECTORY
MESSAGING_PAGING_DIRECTORY MESSAGING_LARGE_MESSAGES_DIRECTORY >
OUTPUT_DATA.xml
```

Cree un módulo personalizado para garantizar que las versiones correctas de los JAR de HornetQ, incluidos los archivos JAR instalados con parches o actualizaciones, se carguen y estén disponibles para el **exporter** utilidad. Usando su editor favorito, cree un nuevo **module.xml** archivo en el **EAP6_HOME/modules/org/hornetq/exporter/main/** directorio y copie el siguiente contenido:

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.hornetq.exporter">
  <main-class
name="org.hornetq.jms.persistence.impl.journal.XmlDataExporter"/>
  <properties>
    <property name="jboss.api" value="deprecated"/>
  </properties>
  <dependencies>
    <module name="org.hornetq"/>
  </dependencies>
</module>
```



NOTA

El módulo personalizado se crea en **modules/** directorio, no el **modules/system/layers/base/** directorio.

Siga los pasos a continuación para exportar los datos.

1. Ejemplo: nombre del controlador JBoss EAP 6 [Esto es una traducción automática]
2. Crea el módulo personalizado como se describe arriba.
3. Ejecute el siguiente comando para exportar los datos.

```
$ java -jar jboss-modules.jar -mp modules/ org.hornetq.exporter
standalone/data/messagingbindings/ standalone/data/messagingjournal/
standalone/data/messagingpaging
standalone/data/messaginglargemessages/ >
OUTPUT_DIRECTORY/OldMessagingData.xml
```

4. Asegúrese de que no haya errores o mensajes de advertencia en el registro al finalizar el comando.
5. Use herramientas disponibles para su sistema operativo para validar el XML en el archivo de salida generado.

Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]

Siga estos pasos para exportar datos de mensajería de JBoss EAP 7.0.

1. Abra una terminal, vaya al directorio de instalación de JBoss EAP 7.0 e inicie el servidor en **admin-only** modo.

```
$ EAP_HOME/bin/standalone.sh -c standalone-full.xml --start-  
mode=admin-only
```

2. Abra una nueva terminal, vaya al directorio de instalación de JBoss EAP 7.0 y conéctese a la CLI de administración.

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```

3. Use el siguiente comando CLI de administración para exportar los datos del diario de mensajería.

```
/subsystem=messaging-activemq/server=default:export-journal()
```

4. Asegúrese de que no haya errores o mensajes de advertencia en el registro al finalizar el comando.
5. Use herramientas disponibles para su sistema operativo para validar el XML en el archivo de salida generado.

Migrar datos de mensajería [Esto es una traducción automática]

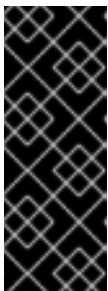
A continuación, importe el archivo XML en JBoss EAP 7.0 o posterior utilizando el **import-journal** operación de la siguiente manera.



IMPORTANTE

Si su servidor de destino ya ha realizado algunas tareas de mensajería, asegúrese de hacer una copia de seguridad de sus carpetas de mensajería antes de comenzar el **import-journal** operación para evitar la pérdida de datos en caso de una falla de importación. Ver [Copia de seguridad de los datos de la carpeta de mensajería](#) para más información.

1. Si está migrando su servidor JBoss EAP 6.4 a 7.1, asegúrese de haber completado la migración de la configuración del servidor antes de comenzar utilizando el [operación de migración CLI de gestión](#) o ejecutando la herramienta de migración de servidores JBoss. Para obtener información sobre cómo configurar y ejecutar la herramienta, consulte [Uso de la herramienta de migración de servidores JBoss](#).
2. Inicie el servidor JBoss EAP 7.x en modo normal con *no* Clientes JMS conectados.



IMPORTANTE

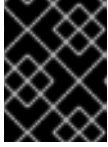
Es importante que inicie el servidor sin clientes JMS conectados. Esto es porque el **import-journal** la operación se comporta como un productor JMS. Los mensajes están disponibles de inmediato cuando la operación está en progreso. Si esta operación falla en el medio de la importación y los clientes JMS están conectados, no hay forma de recuperarlos porque los clientes JMS podrían haber consumido algunos de los mensajes.

3. Abra una nueva terminal, vaya al directorio de instalación de JBoss EAP 7.x y conéctese a la CLI de administración.

```
$ EAP_HOME/bin/jboss-cli.sh --connect
```


4. Use el siguiente comando CLI de administración para importar los datos de mensajería.

```
/subsystem=messaging-activemq/server=default:import-journal(file=OUTPUT_DIRECTORY/OldMessagingData.xml)
```



IMPORTANTE

Hacer *no* ejecute este comando más de una vez, ya que al hacerlo se generarán mensajes duplicados!



AVISO

Si está utilizando JBoss EAP 7.0, debe solicitar [Red Hat JBoss Enterprise Application Platform 7.0 Actualización 05](#) o un parche acumulativo más reciente para su instalación de JBoss EAP para evitar un problema conocido al leer mensajes grandes. Para más información, ver [JBEAP-4407: bloqueos del consumidor con IndexOutOfBoundsException al leer mensajes grandes del diario importado](#). Este problema no afecta a JBoss EAP 7.1 o posterior.

Recuperación de una falla de importación de datos de mensajería

Si el `import-journal` la operación falla, puede intentar recuperar utilizando los siguientes pasos.

1. Cierre el servidor JBoss EAP 7.x.
2. Eliminar todas las carpetas del diario de mensajes. Ver [Copia de seguridad de los datos de la carpeta de mensajería](#) para los comandos CLI de administración para determinar la ubicación de directorio correcta para las carpetas de diario de mensajería.
3. Si realizó una copia de seguridad de los datos de mensajería del servidor de destino antes de la importación, copie las carpetas de mensajería de la ubicación de la copia de seguridad al directorio de diario de mensajería determinado en el paso anterior.
4. Repita los pasos para [importar los datos de mensajería con formato XML](#).

4.9.2.2. Migrar datos de mensajería utilizando un puente JMS [Esto es una traducción automática]

Usando este enfoque, configura e implementa un puente JMS al servidor JBoss EAP 7.x. El puente JMS mueve los mensajes de la cola JBoss EAP 6.4 HornetQ a la cola JBoss EAP 7.x ActiveMQ Artemis.

Un puente JMS consume mensajes desde un tema o cola JMS fuente y los envía al tema o cola JMS destino, el cual usualmente se encuentra en un servidor diferente. Se puede utilizar como puente para los mensajes entre cualquier servidor JMS en tanto cumplan con los requerimientos de JMS 1.1. Los recursos JMS de destino y fuente se buscan utilizando JNDI y las clases clientes para la búsqueda JNDI se deben agrupar en un módulo. Luego el nombre del módulo se declara en la configuración del puente JMS.

Esta sección describe cómo configurar los servidores e implementar un puente JMS para mover los datos de mensajería de JBoss EAP 6.4 a JBoss EAP 7.x.

1. [Configure el servidor de origen JBoss EAP 6.4.](#)
2. [Configure el servidor JBoss EAP 7.x de destino.](#)
3. [Migre los datos de mensajes.](#)

Configurar el servidor Source JBoss EAP 6.4

1. Ejemplo: nombre del controlador JBoss EAP 6 [Esto es una traducción automática]
2. Haga una copia de seguridad del diario HornetQ y de los archivos de configuración.
 - Por defecto, el diario HornetQ se encuentra en el **EAP6_HOME/standalone/data/** directorio.
 - Ver [Asignación de nombres de carpeta de mensajería](#) para las ubicaciones predeterminadas de la carpeta de mensajes para cada versión.
3. Asegúrate de que **InQueue** La cola JMS que contiene los mensajes JMS está definida en el servidor JBoss EAP 6.4.
4. Asegúrate de eso **messaging** La configuración del subsistema contiene una entrada para **RemoteConnectionFactory** similar al siguiente.

```
<connection-factory name="RemoteConnectionFactory">
  <entries>
    <entry
name="java:jboss/exported/jms/RemoteConnectionFactory"/>
  </entries>
  ...
</connection-factory>
```

Si no contiene la entrada, cree una con el siguiente comando de administración CLI:

```
/subsystem=messaging/hornetq-server=default/connection-
factory=RemoteConnectionFactory:add(factory-type=XA_GENERIC,
connector=[netty], entries=
[java:jboss/exported/jms/RemoteConnectionFactory],ha=true,block-on-
acknowledge=true,retry-interval=1000,retry-interval-
multiplier=1.0,reconnect-attempts=-1)
```

Configurar el servidor JBoss EAP 7.x de destino

1. La configuración del puente JMS necesita **org.hornetq** módulo para conectarse al servidor HornetQ en la versión anterior. Este módulo y sus dependencias directas no están presentes en JBoss EAP 7.x, por lo que debe copiar los siguientes módulos de la versión anterior.
 - Copia el **org.hornetq** módulo en el JBoss EAP 7.x **EAP_HOME/modules/org/** directorio.
 - Si no aplicó revisiones a este módulo, copie esta carpeta del servidor JBoss EAP 6.4: **EAP6_HOME/modules/system/layers/base/org/hornetq/**

- Si aplicó parches a este módulo, copie esta carpeta del servidor JBoss EAP 6.4:
EAP6_HOME/modules/system/layers/base/.overlays/layer-base-jboss-eap-6.4.x.CP/org/hornetq/
- Eliminar el **<resource-root>** para el HornetQ **lib** ruta desde JBoss EAP 7.x
EAP_HOME/modules/org/hornetq/main/module.xml archivo.
 - Si no aplicó parches a JBoss EAP 6.4 **org.hornetq** módulo, elimine la siguiente línea del archivo:

```
<resource-root path="lib"/>
```

- Si aplicó parches a JBoss EAP 6.4 **org.hornetq** módulo, elimine las siguientes líneas del archivo:

```
<resource-root path="lib"/>
<resource-root path="../../../../../../org/hornetq/main/lib"/>
```



AVISO

Fallo al eliminar el HornetQ **lib** camino **resource-root** hará que el puente falle con el siguiente error en el archivo de registro.

```
2016-07-15 09:32:25,660 ERROR
[org.jboss.as.controller.management-operation]
(management-handler-thread - 2) WFLYCTL0013:
Operation ("add") failed - address: ([
  ("subsystem" => "messaging-activemq"),
  ("jms-bridge" => "myBridge")
]) - failure description: "WFLYMSGAMQ0086:
Unable to load module org.hornetq"
```

- Copia el **org.jboss.netty** módulo en el JBoss EAP 7.x
EAP_HOME/modules/org/jboss/ directorio.
 - Si no aplicó revisiones a este módulo, copie esta carpeta del servidor JBoss EAP 6.4:
EAP6_HOME/modules/system/layers/base/org/jboss/netty/
 - Si aplicó parches a este módulo, copie esta carpeta del servidor JBoss EAP 6.4:
EAP6_HOME/modules/system/layers/base/.overlays/layer-base-jboss-eap-6.4.x.CP/org/jboss/netty
2. Cree la cola JMS para contener los mensajes recibidos del servidor JBoss EAP 6.4. El siguiente es un ejemplo de un comando CLI de administración que crea el **MigratedMessagesQueue** JMS pone en cola para recibir el mensaje.

```
jms-queue add --queue-address=MigratedMessagesQueue --entries=
[jms/queue/MigratedMessagesQueue
java:jboss/exported/jms/queue/MigratedMessagesQueue]
```

Esto crea lo siguiente **jms-queue** configuración para el servidor predeterminado en el **messaging-activemq** subsistema del servidor JBoss EAP 7.x.

```
<jms-queue name="MigratedMessagesQueue"
entries="jms/queue/MigratedMessagesQueue
java:jboss/exported/jms/queue/MigratedMessagesQueue"/>
```

3. Asegúrate de eso **messaging-activemq** subsistema **default** servidor contiene una configuración para el **InVmConnectionFactory** **connection-factory** similar a lo siguiente:

```
<connection-factory name="InVmConnectionFactory" factory-
type="XA_GENERIC" entries="java:/ConnectionFactory" connectors="in-
vm"/>
```

Si no contiene la entrada, cree una con el siguiente comando de administración CLI:

```
/subsystem=messaging-activemq/server=default/connection-
factory=InVmConnectionFactory:add(factory-type=XA_GENERIC,
connectors=[in-vm], entries=[java:/ConnectionFactory])
```

4. Cree e implemente un puente JMS que lea mensajes del **InQueue** Cola JMS configurada en el servidor JBoss EAP 6.4 y la transfiere a la **MigratedMessagesQueue** configurado en el servidor JBoss EAP 7.x.

```
/subsystem=messaging-activemq/jms-bridge=myBridge:add(add-messageID-
in-header=true,max-batch-time=100,max-batch-size=10,max-retries=-
1,failure-retry-interval=1000,quality-of-
service=AT_MOST_ONCE,module=org.hornetq,source-
destination=jms/queue/InQueue,source-connection-
factory=jms/RemoteConnectionFactory,source-context=
[("java.naming.factory.initial"=>"org.wildfly.naming.client.WildFlyI
nitialContextFactory"),
("java.naming.provider.url"=>"remote://127.0.0.1:4447")],target-
destination=jms/queue/MigratedMessagesQueue,target-connection-
factory=java:/ConnectionFactory)
```

Esto crea lo siguiente **jms-bridge** configuración en el **messaging-activemq** subsistema del servidor JBoss EAP 7.x.

```
<jms-bridge name="myBridge" add-messageID-in-header="true" max-
batch-time="100" max-batch-size="10" max-retries="-1" failure-retry-
interval="1000" quality-of-service="AT_MOST_ONCE"
module="org.hornetq">
  <source destination="jms/queue/InQueue" connection-
factory="jms/RemoteConnectionFactory">
    <source-context>
      <property name="java.naming.factory.initial"
value="org.wildfly.naming.client.WildFlyInitialContextFactory"/>
      <property name="java.naming.provider.url"
value="remote://127.0.0.1:4447"/>
    </source-context>
  </source>
```

```
<target destination="jms/queue/MigratedMessagesQueue"
connection-factory="java:/ConnectionFactory"/>
</jms-bridge>
```

5. Si la seguridad está configurada para JBoss EAP 6.4, también debe configurar la configuración del puente JMS **<source>** elemento para incluir un **source-context** que especifica el nombre de usuario y la contraseña correctos para utilizar para la búsqueda JNDI al crear la conexión.

Migrar datos de mensajería [Esto es una traducción automática]

1. Verifique que la información que proporcionó para las siguientes configuraciones sea correcta.
 - Cualquier cola y nombres de tema.
 - los `java.naming.provider.url` para búsqueda JNDI.
2. Asegúrese de haber desplegado el destino JMS destino en el servidor JBoss EAP 7.x.
3. Inicie los servidores JBoss EAP 6.4 y JBoss EAP 7.x.

4.9.2.3. Asignación de nombres de carpeta de mensajería [Esto es una traducción automática]

La siguiente tabla muestra los nombres de directorio de mensajería utilizados en la versión anterior y los nombres correspondientes utilizados en la versión actual de JBoss EAP. Los directorios son relativos a `jboss.server.data.dir` directorio, que por defecto es `EAP_HOME/standalone/data/` si no está especificado

Ejemplo: nombre del controlador JBoss EAP 6 [Esto es una traducción automática]	Ejemplo: nombre del controlador JBoss EAP 7 [Esto es una traducción automática]
<code>messagingbindings/</code>	<code>activemq/bindings/</code>
<code>messagingjournal/</code>	<code>activemq/journal/</code>
<code>messaginglargemessages/</code>	<code>activemq/largemessages/</code>
<code>messagingpaging/</code>	<code>activemq/paging/</code>



NOTA

los `messaginglargemessages/` y `messagingpaging/` Es posible que los directorios no estén presentes si no hay mensajes grandes o si la búsqueda está deshabilitada.

4.9.2.4. Copia de seguridad de los datos de la carpeta de mensajería [Esto es una traducción automática]

Si su servidor de destino ya ha procesado los mensajes, es una buena idea realizar una copia de seguridad de las carpetas de mensajes de destino en una ubicación de respaldo antes de comenzar. La ubicación predeterminada de las carpetas de mensajes es

EAP_HOME/standalone/data/activemq/; sin embargo, es configurable. Si no está seguro de la ubicación de sus datos de mensajería, puede usar los siguientes comandos CLI de administración para encontrar la ubicación de las carpetas de mensajería.

```
/subsystem=messaging-activemq/server=default/path=journal-
directory:resolve-path
/subsystem=messaging-activemq/server=default/path=paging-
directory:resolve-path
/subsystem=messaging-activemq/server=default/path=bindings-
directory:resolve-path
/subsystem=messaging-activemq/server=default/path=large-messages-
directory:resolve-path
```

Una vez que sepa la ubicación de las carpetas, copie cada carpeta a una ubicación segura de respaldo.

4.9.3. Migrar destinos JMS [Esto es una traducción automática]

En versiones anteriores de JBoss EAP, las colas de destino JMS se configuraban en **<jms-destinations>** elemento bajo el **<hornetq-server>** elemento en el **messaging** subsistema.

```
<hornetq-server>
...
<jms-destinations>
  <jms-queue name="testQueue">
    <entry name="queue/test"/>
    <entry name="java:jboss/exported/jms/queue/test"/>
  </jms-queue>
</jms-destinations>
...
</hornetq-server>
```

En JBoss EAP 7, la cola de destino JMS está configurada de manera predeterminada **<server>** elemento de la **messaging-activemq** subsistema.

```
<server name="default">
...
  <jms-queue name="testQueue" entries="queue/test
java:jboss/exported/jms/queue/test"/>
...
</server>
```

4.9.4. Migrar interceptores de mensajería [Esto es una traducción automática]

Los interceptores de mensajería han cambiado significativamente en JBoss EAP 7 con el reemplazo de HornetQ por ActiveMQ Artemis como proveedor de mensajería JMS.

El HornetQ **messaging** El subsistema incluido en la versión anterior de JBoss EAP requiere que instales los interceptores HornetQ agregándolos a un JAR y luego modificando el HornetQ **module.xml** archivo.

los **messaging-activemq** subsistema incluido en JBoss EAP 7 no requiere la modificación de un **module.xml** archivo. Clases de interceptor de usuario, que ahora implementan Apache ActiveMQ Artemis [Interceptador](#) interfaz, ahora se puede cargar desde cualquier módulo de servidor. Usted

especifica el módulo desde el cual se debe cargar el interceptor en el **messaging-activemq** subsistema del archivo de configuración del servidor.

Ejemplo: configuración del interceptor [Esto es una traducción automática]

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:2.0">
  <server name="default">
    ...
    <incoming-interceptors>
      <class name="com.mycompany.incoming.myInterceptor"
module="com.mycompany" />
      <class name="com.othercompany.incoming.myOtherInterceptor"
module="com.othercompany" />
    </incoming-interceptors>
    <outgoing-interceptors>
      <class name="com.mycompany.outgoing.myInterceptor"
module="com.mycompany" />
      <class name="com.othercompany.outgoing.myOtherInterceptor"
module="com.othercompany" />
    </outgoing-interceptors>
  </server>
</subsystem>
```

4.9.5. Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]

En JBoss EAP 6, puede configurar un motor de servlet para que funcione con el transporte de Netty Servlet. Debido a que ActiveMQ Artemis reemplaza a HornetQ como el proveedor de mensajería incorporado en JBoss EAP 7, esta configuración ya no está disponible. En su lugar, debe reemplazar la configuración de servlet para usar los nuevos conectores de mensajería HTTP incorporados y los aceptores HTTP.

4.9.6. Configuración de un adaptador de recursos JMS genérico [Esto es una traducción automática]

La forma en que configura un adaptador de recursos JMS genérico para usar con un proveedor JMS de terceros ha cambiado en JBoss EAP 7. Para obtener más información, consulte [Implementación de un adaptador de recursos JMS genérico](#) en *Configurando la Mensajería* para JBoss EAP.

4.9.7. Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]

En JBoss EAP 7.0, si configuró el **replication-master** política sin especificar el **check-for-live-server** atributo, su valor predeterminado era **false**. Esto ha cambiado en JBoss EAP 7.1. El valor predeterminado para el **check-for-live-server** atributo es ahora **true**.

El siguiente es un ejemplo de un comando CLI de administración que configura el **replication-master** política sin especificar el **check-for-live-server** atributo.

```
/subsystem=messaging-activemq/server=default/ha-policy=replication-
master:add(cluster-name=my-cluster,group-name=group1)
```

Cuando lea el recurso utilizando la CLI de administración, tenga en cuenta que **check-for-live-server** el valor de atributo está configurado para **true**.

```
/subsystem=messaging-activemq/server=default/ha-policy=replication-
master:read-resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "check-for-live-server" => true,
    "cluster-name" => "my-cluster",
    "group-name" => "group1",
    "initial-replication-sync-timeout" => 30000L
  },
  "response-headers" => {"process-state" => "reload-required"}
}
```

4.9.8. Cambios en el comportamiento de serialización JMS entre releases [Esto es una traducción automática]

los **serialVersionUID** de **javax.jms.JMSEException** cambiado entre JMS 1.1 y JMS 2.0.0. Esto significa que si una instancia de **JMSEException**, o cualquiera de sus subclases, se serializa usando JMS 1.1, no se puede deserializar usando JMS 2.0.0. Lo contrario también es cierto. Si una instancia de **JMSEException** se serializa usando JMS 2.0.0, no se puede deserializar usando JMS 1.1. En ambos casos, arroja una excepción similar a la siguiente:

```
javax.jms.JMSEException: javax.jms.JMSEException; clase local incompatible:
stream classdesc serialVersionUID = 8951994251593378324, clase local
serialVersionUID = 2368476267211489441
```

Este problema se corrigió en la versión de mantenimiento JMS 2.0.1.

Implementación de JMS para cada versión de JBoss EAP [Esto es una traducción automática]

Tabla 4.4. Implementación de JMS para cada versión de JBoss EAP [Esto es una traducción automática]

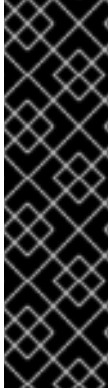
JBoss EAP Version	Implementación de JMS	La versión JMS
6.4	HornetQ	JMS 1.1
7.0	Apache ActiveMQ Artemis	JMS 2.0.0
7.1	Apache ActiveMQ Artemis	JMS 2.0.1

Tenga en cuenta que **serialVersionUID** la incompatibilidad puede generar un problema de migración en las siguientes situaciones:

- Si envía un mensaje que contiene un **JMSEException** utilizando un cliente JBoss EAP 6.4, migre sus datos de mensajería a JBoss EAP 7.0, y luego intente deserializar ese mensaje utilizando un cliente JBoss EAP 7.0, la deserialización fallará y arrojará una excepción. Esto es porque el **serialVersionUID** en JMS 1.1 es *no compatible* con el de JMS 2.0.0.

- Si envía un mensaje que contiene un **JMSEException** utilizando un cliente JBoss EAP 7.0, migre sus datos de mensajería a JBoss EAP 7.1, y luego intente deserializar ese mensaje utilizando un cliente JBoss EAP 7.1, la deserialización fallará y arrojará una excepción. Esto es porque el **serialVersionUID** en JMS 2.0.0 es *no* compatible con el de JMS 2.0.1.

Tenga en cuenta que si envía un mensaje que contiene un **JMSEException** utilizando un cliente JBoss EAP 6.4, migre sus datos de mensajería a JBoss EAP 7.1, y luego intente deserializar ese mensaje utilizando un cliente JBoss EAP 7.1, la deserialización tendrá éxito porque la **serialVersionUID** en JMS 1.1 es compatible con el de JMS 2.0.1.



IMPORTANTE

Red Hat recomienda que haga lo siguiente antes de migrar sus datos de mensajería:

- Asegúrese de consumir todos los mensajes JMS 1.1 que contengan JMSEExceptions antes de migrar los datos de mensajería de JBoss EAP 6.4 a JBoss EAP 7.0.
- Asegúrese de consumir todos los mensajes JMS 2.0.0 que contengan JMSEExceptions antes de migrar los datos de mensajería de JBoss EAP 7.0 a JBoss EAP 7.1.

4.10. CAMBIOS EN LA ADMINISTRACIÓN DE JMX [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

El componente HornetQ en JBoss EAP 6 proporcionó su propia administración JMX; sin embargo, no fue recomendado y ahora está en desuso y ya no es compatible. Si confió en esta función en JBoss EAP 6, debe migrar sus herramientas de administración para usar la CLI de administración de JBoss EAP o la administración de JM proporcionada con JBoss EAP 7.

También debe actualizar sus bibliotecas de cliente para usar **jboss-client.jar** que se envía con JBoss EAP 7.

El siguiente es un ejemplo del código de administración HornetQ JMX que se usó en JBoss EAP 6.

```
JMXConnector connector = null;
try {
    HashMap environment = new HashMap();
    String[] credentials = new String[]{"admin", "Password123!"};
    environment.put(JMXConnector.CREDENTIALS, credentials);

    // HornetQ used the protocol "remoting-jmx" and port "9999"
    JMXServiceURL beanServerUrl = new JMXServiceURL("service:jmx:remoting-
jmx://127.0.0.1:9999");

    connector = JMXConnectorFactory.connect(beanServerUrl, environment);
    MBeanServerConnection mbeanServer =
connector.getMBeanServerConnection();

    // The JMX object name pointed to the HornetQ JMX management
    ObjectName objectName = new
ObjectName("org.hornetq:type=Server,module=JMS");

    // The invoked method name was "listConnectionIDs"
    String[] connections = (String[]) mbeanServer.invoke(objectName,
```



```

"listConnectionIds", new Object[] {}, new String[] {});
    for (String connection : connections) {
        System.out.println(connection);
    }
} finally {
    if (connector != null) {
        connector.close();
    }
}
}

```

El siguiente es un ejemplo del código equivalente necesario para ActiveMQ Artemis en JBoss EAP 7.

```

JMXConnector connector = null;
try {
    HashMap environment = new HashMap();
    String[] credentials = new String[] {"admin", "Password123!"};
    environment.put(JMXConnector.CREDENTIALS, credentials);

    // ActiveMQ Artemis uses the protocol "remote+http" and port "9990"
    JMXServiceURL beanServerUrl = new
JMXServiceURL("service:jmx:remote+http://127.0.0.1:9990");

    connector = JMXConnectorFactory.connect(beanServerUrl, environment);
    MBeanServerConnection mbeanServer =
connector.getMBeanServerConnection();

    // The JMX object name points to the new JMX management in the
`messaging-activemq` subsystem
    ObjectName objectName = new ObjectName("jboss.as:subsystem=messaging-
activemq,server=default");

    // The invoked method name is now "listConnectionIds"
    String[] connections = (String[]) mbeanServer.invoke(objectName,
"listConnectionIds", new Object[] {}, new String[] {});
    for (String connection : connections) {
        System.out.println(connection);
    }
} finally {
    if (connector != null) {
        connector.close();
    }
}
}

```

Observe que los nombres y parámetros del método han cambiado en la nueva implementación. Puede encontrar los nuevos nombres de métodos en JConsole siguiendo estos pasos.

1. Conéctese a JConsole con el siguiente comando.

```
$ EAP_HOME/bin/jconsole.sh
```

2. Conéctese al proceso local de JBoss EAP. Tenga en cuenta que debe comenzar con "jboss-modules.jar".
3. En el **MBeans** pestaña, elegir **jboss.as** → **mensajería-activemq** → **defecto** → **Operaciones** para mostrar la lista de nombres y atributos de métodos.

4.11. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR ORB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La implementación de JacORB ha sido reemplazada por una rama descendente de OpenJDK ORB en JBoss EAP 7.

los `org.jboss.as.jacorb` módulo de extensión, ubicado en `EAP_HOME/modules/system/layers/base/`, ha sido reemplazado por `org.wildfly.iio-
openjdk` módulo de extensión.

los `urn:jboss:domain:jacorb:1.4` El espacio de nombre de configuración del subsistema en el archivo de configuración del servidor ha sido reemplazado por `urn:jboss:domain:iio-
openjdk:1.0` espacio de nombres

El siguiente es un ejemplo de lo predeterminado `jacorb` configuración del sistema en JBoss EAP 6.

```
<subsystem xmlns="urn:jboss:domain:jacorb:1.4">
  <orb socket-binding="jacorb" ssl-socket-binding="jacorb-ssl">
    <initializers security="identity" transactions="spec"/>
  </orb>
</subsystem>
```

El siguiente es un ejemplo de lo predeterminado `iio-openjdk` configuración del subsistema en JBoss EAP 7.

```
<subsystem xmlns="urn:jboss:domain:iio-openjdk:1.0">
  <orb socket-binding="jacorb" ssl-socket-binding="jacorb-ssl" />
  <initializers security="identity" transactions="spec" />
</subsystem>
```

El nuevo `iio-openjdk` La configuración del subsistema solo acepta un subconjunto de los elementos y atributos heredados. El siguiente es un ejemplo de `jacorb` configuración del subsistema en la versión anterior de JBoss EAP que contiene todos los elementos y atributos válidos:

```
<subsystem xmlns="urn:jboss:domain:jacorb:1.4">
  <orb name="JBoss" print-version="off" use-imr="off" use-bom="off"
cache-typecodes="off"
  cache-poa-names="off" giop-minor-version="2" socket-
binding="jacorb" ssl-socket-binding="jacorb-ssl">
    <connection retries="5" retry-interval="500" client-timeout="0"
server-timeout="0"
      max-server-connections="500" max-managed-buf-size="24" outbuf-
size="2048"
      outbuf-cache-timeout="-1"/>
    <initializers security="off" transactions="spec"/>
  </orb>
  <poa monitoring="off" queue-wait="on" queue-min="10" queue-max="100">
    <request-processors pool-size="10" max-threads="32"/>
  </poa>
  <naming root-context="JBoss/Naming/root" export-corbaloc="on"/>
  <interop sun="on" comet="off" iona="off" chunk-custom-rmi-
valuetypes="on"
    lax-boolean-encoding="off" indirection-encoding-disable="off"
strict-check-on-tc-creation="off"/>
```

```

    <security support-ssl="off" add-component-via-interceptor="on" client-
supports="MutualAuth"
    client-requires="None" server-supports="MutualAuth" server-
requires="None"/>
    <properties>
    <property name="some_property" value="some_value"/>
    </properties>
</subsystem>

```

Los siguientes atributos de elemento ya no son compatibles y deben eliminarse.

Tabla 4.5. Atributos para eliminar [Esto es una traducción automática]

Elemento	Atributos no admitidos
<orb>	<ul style="list-style-type: none"> • client-timeout • max-managed-buf-size • max-server-connections • outbuf-cache-timeout • outbuf-size • reintentos de conexión • retry-interval • Nombre • server-timeout
<poa>	<ul style="list-style-type: none"> • queue-min • queue-max • pool-size • max-hilos

El seguimiento **on/off** los atributos ya no son compatibles y no se migrarán cuando ejecute la CLI de administración **migrate** operación. Si están configurados para **on**, recibirá una advertencia de migración. Otro **on/off** atributos que no se mencionan en esta tabla, por ejemplo **<security support-ssl="on|off">**, todavía son compatibles y se migrarán con éxito. La única diferencia es que sus valores serán cambiados de **on/off** a **true/false**.

Tabla 4.6. Atributos para desactivar o eliminar [Esto es una traducción automática]

Elemento	Atributos para eliminar [Esto es una traducción automática]
----------	---

Elemento	Atributos para eliminar [Esto es una traducción automática]
<orb>	<ul style="list-style-type: none"> • cache-poa-names • cache-typecodes • print-version • uso-bom • use-imr
<interop>	<p>(todo excepto sun)</p> <ul style="list-style-type: none"> • cometa • iona • chunk-custom-rmi-valuetypes • indirection-encoding-disable • labo-boolean-encoding • strict-check-on-tc-creation
<poa>	<ul style="list-style-type: none"> • supervisión • queue-wait

4.12. MIGRAR LA CONFIGURACIÓN DEL SUBSISTEMA DE SUBPROCESOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La configuración del servidor JBoss EAP 6 incluyó una **threads** subsistema que se usó para administrar grupos de subprocesos en los distintos subsistemas del servidor.

los **threads** el subsistema ya no está disponible en JBoss EAP 7. En su lugar, cada subsistema es responsable de administrar sus propios grupos de subprocesos.

Para obtener información acerca de cómo configurar grupos de subprocesos para **infinispan** subsistema, ver [Configurar Infinispan Thread Pools](#) en el JBoss EAP *Guía de configuración*.

Para obtener información acerca de cómo configurar grupos de subprocesos para **jgroups** subsistema, ver [Configurar JGroups Thread Pools](#) en el JBoss EAP *Guía de configuración*.

En JBoss EAP 6, usted configuró grupos de hilos para conectores y oyentes para el **web** subsistema al hacer referencia a un **executor** eso fue definido en el **threads** subsistema. En JBoss EAP 7, ahora configura pools de hilos para el **undertow** subsistema haciendo referencia a un **worker** eso se define en el **io** subsistema. Para más información, ver [Configurando el subsistema IO](#) en el JBoss EAP *Guía de configuración*.

Para obtener información acerca de los cambios en la configuración del grupo de subprocesos en **remoting** subsistema, ver [Migrar la configuración del subsistema remoto](#) en esta guía, y [Configurando el punto final](#) en el JBoss EAP *Guía de configuración*.

4.13. MIGRAR LA CONFIGURACIÓN DEL SUBSISTEMA REMOTO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 6, configuró el grupo de subprocesos para **remoting** subsistema mediante el establecimiento de diversos **worker-*** atributos. El grupo de subprocesos de trabajo ya no está configurado en el **remoting** subsistema en JBoss EAP 7 y si intenta modificar la configuración existente, verá el siguiente mensaje.

```
WFLYRMT0022: Worker configuration is no longer used, please use endpoint
worker configuration
```

En JBoss EAP 7, el grupo de subprocesos de trabajo se reemplaza por una configuración de punto final que hace referencia a **worker** definido en el **io** subsistema.

Para obtener información sobre cómo configurar el punto final, consulte [Configurando el punto final](#) en el JBoss EAP *Guía de configuración*.

4.14. CAMBIOS EN LA CONFIGURACIÓN DEL SERVIDOR WEBSOCKET [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Para usar WebSockets en JBoss EAP 6, debe habilitar el protocolo de conector Java NIO2 no bloqueante para el **http** conector en el **web** subsistema del archivo de configuración del servidor JBoss EAP utilizando un comando similar al siguiente.

```
/subsystem=web/connector=http/:write-
attribute(name=protocol,value=org.apache.coyote.http11.Http11NioProtocol)
```

Para usar WebSockets en una aplicación, también tenía que crear un **<enable-websockets>** elemento en la aplicación **WEB-INF/jboss-web.xml** archivar y configurarlo para **true**.

En JBoss EAP 7, ya no es necesario configurar el servidor para el soporte predeterminado de WebSocket o configurar la aplicación para usarlo. Los WebSockets son un requisito de la especificación Java EE 7 y los protocolos necesarios están configurados por defecto. La configuración WebSocket más compleja se realiza en **servlet-container** del **undertow** subsistema del archivo de configuración del servidor JBoss EAP. Puede ver las configuraciones disponibles usando el siguiente comando.

```
/subsystem=undertow/servlet-container=default/setting=websockets:read-
resource(recursive=true)
{
  "outcome" => "success",
  "result" => {
    "buffer-pool" => "default",
    "dispatch-to-worker" => true,
    "worker" => "default"
  }
}
```

Para obtener más información sobre el desarrollo de WebSocket, consulte [Crear aplicaciones WebSocket](#) en el JBoss EAP *Guía de desarrollo*.

Los ejemplos de código WebSocket también se pueden encontrar en los inicios rápidos que se envían con JBoss EAP.

4.15. CAMBIOS EN EL SERVIDOR DE SINGLE SIGN-ON [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

los **infinispan** el subsistema aún proporciona soporte de caché distribuido para servicios de HA en forma de cachés Infinispan en JBoss EAP 7; sin embargo, el almacenamiento en caché y la distribución de la información de autenticación se manejan de forma diferente que en versiones anteriores.

En JBoss EAP 6, si el inicio de sesión único (SSO) no se proporcionó un caché Infinispan, el caché no se distribuyó.

En JBoss EAP 7, SSO se distribuye automáticamente cuando selecciona el perfil HA. Al ejecutar el perfil HA, cada host tiene su propio caché Infinispan, que se basa en el caché predeterminado del contenedor del caché web. Este caché almacena la sesión relevante y la información de cookies SSO para el host. JBoss EAP maneja la propagación de información de caché individual a todos los hosts. No hay forma de asignar específicamente un caché Infinispan al SSO en JBoss EAP 7.

En JBoss EAP 7, SSO está configurado en **undertow** subsistema del archivo de configuración del servidor.

No se requieren cambios en el código de la aplicación para SSO al migrar a JBoss EAP 7.

4.16. CAMBIOS EN LA CONFIGURACIÓN DE DATASOURCE [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

4.16.1. Nombre del controlador JDBC Datasource [Esto es una traducción automática]

Cuando configuró un origen de datos en la versión anterior de JBoss EAP, el valor especificado para el nombre del controlador depende de la cantidad de clases enumeradas en el **META-INF/services/java.sql.Driver** archivo contenido en el controlador JDBC JAR.

Controlador que contiene una clase individual

Si el **META-INF/services/java.sql.Driver** el archivo especificó solo una clase, el valor del nombre del controlador era simplemente el nombre del JAR del controlador JDBC. Esto no ha cambiado en JBoss EAP 7.

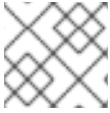
Conductor que contiene múltiples clases

En JBoss EAP 6, si había más de una clase enumerada en **META-INF/services/java.sql.Driver** archivo, especificó qué clase era la clase de controlador al agregar su nombre al nombre JAR, junto con la versión principal y secundaria, en el siguiente formato.

```
JAR_NAME + DRIVER_CLASS_NAME + "_" + MAJOR_VERSION + "_" + MINOR_VERSION
```

En JBoss EAP 7, esto ha cambiado. Ahora especifica el nombre del controlador usando el siguiente formato.

```
JAR_NAME + "_" + DRIVER_CLASS_NAME + "_" + MAJOR_VERSION + "_" + MINOR_VERSION
```

**NOTA**

Se ha agregado un guión bajo entre *JAR_NAME* y el *DRIVER_CLASS_NAME*.

El controlador JDBC de MySQL 5.1.31 es un ejemplo de un controlador que contiene dos clases. El nombre de la clase del controlador es **com.mysql.jdbc.Driver**. Los siguientes ejemplos demuestran las diferencias entre cómo se especifica el nombre del controlador en la versión anterior y actual de JBoss EAP.

Ejemplo: nombre del controlador JBoss EAP 6 [Esto es una traducción automática]

```
mysql-connector-java-5.1.31-bin.jarcom.mysql.jdbc.Driver_5_1
```

Ejemplo: nombre del controlador JBoss EAP 7 [Esto es una traducción automática]

```
mysql-connector-java-5.1.31-bin.jar_com.mysql.jdbc.Driver_5_1
```

4.17. CAMBIOS DE CONFIGURACIÓN DEL SERVIDOR DE SEGURIDAD [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Si migra a JBoss EAP 7 y planifica ejecutarlo con el Administrador de seguridad de Java habilitado, debe tener en cuenta que se realizaron cambios en la forma en que se definen las políticas y que pueden ser necesarios cambios de configuración adicionales. También tenga en cuenta que los administradores de seguridad personalizados no son compatibles con JBoss EAP 7.

Para obtener información sobre los cambios en la configuración del servidor de Java Security Manager, consulte [Consideraciones para pasar de versiones anteriores](#) en *Cómo configurar la seguridad del servidor* para JBoss EAP.

4.17.1. Cambios en el comportamiento de seguridad heredado entre JBoss EAP 7.0 y JBoss EAP 7.1 [Esto es una traducción automática]

4.17.1.1. Cambio de estado HTTP para dominios de LDAP inalcanzables [Esto es una traducción automática]

Si el servidor no puede acceder a un dominio LDAP en JBoss EAP 7.0, el **security** el subsistema devolvió un código de estado HTTP de "401 no autorizado". Esto ha cambiado en JBoss EAP 7.1. El legado **security** El subsistema en JBoss EAP 7.1 en su lugar devuelve un código de estado HTTP de "500 Error interno" para describir con mayor precisión que se produjo una situación inesperada que impidió que el servidor procesara correctamente la solicitud.

4.17.1.2. Habilitación del dominio de seguridad LDAP para analizar funciones desde un DN [Esto es una traducción automática]

En JBoss EAP 7.0, el **org.jboss.as.domain.management.security.parseGroupNameFromLdapDN** la propiedad del sistema se usó para habilitar el dominio de seguridad LDAP para analizar las funciones desde un DN. Cuando esta propiedad se configuró en **true**, los roles fueron analizados desde un DN. De lo contrario, se utilizó una búsqueda LDAP normal para buscar roles.

En JBoss EAP 7.1, esta propiedad del sistema ahora está en desuso. En su lugar, configure esta opción configurando el recién introducido **parse-group-name-from-dn** atribuir a **true** en la ruta del servicio central utilizando el siguiente comando CLI de administración:

```
/core-service=management/security-  
realm=REALM_NAME/authorization=ldap/group-search=principal-to-  
group:add(parse-group-name-from-dn=true)
```

4.17.1.3. Cambios en el envío del certificado JBoss EAP SSL a un servidor LDAP [Esto es una traducción automática]

En JBoss EAP 7.0, cuando la interfaz de administración está configurada para usar el **ldapSSL** el dominio de seguridad, la autenticación mutua entre el servidor y LDAP puede fallar, lo que da como resultado un error de autenticación en la interfaz de administración. Esto se debe a que se realizan dos conexiones LDAP diferentes, cada una con un hilo diferente, y no comparten las sesiones SSL.

JBoss EAP 7.1 presenta un nuevo booleano **always-send-client-cert** atributo de gestión en el LDAP **outbound-connection**. Esta opción permite la configuración de conexiones LDAP salientes para admitir servidores LDAP que están configurados para requerir siempre un certificado de cliente.

La autenticación LDAP ocurre en dos pasos:

1. Busca la cuenta.
2. Verifica las credenciales.

Por defecto, el **always-send-client-cert** atributo está configurado para **false**, lo que significa que el certificado SSL del cliente se envía solo con la primera solicitud de la cuenta de búsqueda. Cuando este atributo está configurado para **true**, el cliente JBoss EAP LDAP envía el certificado del cliente al servidor LDAP con las solicitudes de búsqueda y verificación.

Puede configurar este atributo para **true** utilizando el siguiente comando de gestión CLI.

```
/core-service=management/ldap-connection=my-ldap-connection:write-  
attribute(name=always-send-client-cert,value=true)
```

Esto da como resultado la siguiente conexión de salida LDAP en el archivo de configuración del servidor.

```
<management>  
  ....  
  <outbound-connections>  
    <ldap name="my-ldap-connection" url="ldap://127.0.0.1:389" search-  
dn="cn=search,dc=myCompany,dc=com" search-credential="myPass" always-send-  
client-cert="true"/>  
  </outbound-connections>  
  ....  
</management>
```

4.17.2. Cambios en el modo FIPS [Esto es una traducción automática]

Cambios en el comportamiento de seguridad heredado entre JBoss EAP 7.0 y JBoss EAP 7.1 [Esto es una traducción automática]

Al utilizar dominios de seguridad heredados, JBoss EAP 7.1 proporciona la generación automática de un certificado autofirmado para fines de desarrollo. Esta característica, que no estaba disponible en JBoss EAP 7.0, está habilitada de manera predeterminada. Esto significa que si está ejecutando en modo FIPS, debe configurar el servidor para deshabilitar la creación automática de certificados autofirmados. De lo contrario, es posible que vea el siguiente error al iniciar el servidor.

```
ERROR [org.xnio.listener] (default I/O-6) XNI0001007: A channel event
listener threw an exception: java.lang.RuntimeException: WFLYDM0114:
Failed to lazily initialize SSL context
...
Caused by: java.lang.RuntimeException: WFLYDM0112: Failed to generate self
signed certificate
...
Caused by: java.security.KeyStoreException: Cannot get key bytes, not
PKCS#8 encoded
```

Para obtener información sobre la creación automática de certificados autofirmados, consulte [Creación automática de certificados autofirmados para aplicaciones](#) en *Cómo configurar la seguridad del servidor* para JBoss EAP.

4.18. CAMBIOS EN EL SUBSISTEMA DE TRANSACCIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Algunos atributos de configuración de Transaction Manager que estaban disponibles en **transactions** el subsistema en JBoss EAP 6 ha cambiado en JBoss EAP 7.

Cambios en el subsistema de transacciones [Esto es una traducción automática]

La siguiente tabla enumera los atributos de JBoss EAP 6 que se eliminaron del **transactions** subsistema en JBoss EAP 7 y los atributos de reemplazo equivalentes.

Atributo en JBoss EAP 6	Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]
ruta	object-store-path
relativo a	object-store-relative-to

Cambios en el subsistema de transacciones [Esto es una traducción automática]

Los siguientes atributos que estaban disponibles en **transactions** el subsistema en JBoss EAP 6 está en desuso en JBoss EAP 7. Los atributos obsoletos podrían eliminarse en una versión futura del producto. La siguiente tabla enumera los atributos de reemplazo equivalentes.

Atributo en JBoss EAP 6	Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]
use-hornetq-store	use-journal-store
hornetq-store-enable-async-io-to	journal-store-enable-async-io
habilitar-estadísticas	estadísticas habilitadas

4.19. CAMBIOS A LA CONFIGURACIÓN MOD_CLUSTER [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La configuración para listas de proxy estáticas en mod_cluster ha cambiado en JBoss EAP 7.

En JBoss EAP 6, configuró el **proxy-list** atributo, que era una lista separada por comas de direcciones de proxy httpd especificadas en el formato de **hostname:port**.

los **proxy-list** atributo está en desuso en JBoss EAP 7. Ha sido reemplazado por el **proxies** atributo, que es una lista de nombres de enlace de socket de salida.

Este cambio afecta la forma en que define una lista de proxy estática, por ejemplo, al deshabilitar la publicidad de mod_cluster. Para obtener información acerca de cómo deshabilitar la publicidad de mod_cluster, consulte [Deshabilitar publicidad para mod_cluster](#) en el JBoss EAP *Guía de configuración*.

Para obtener más información sobre los atributos mod_cluster, consulte [Atributos del subsistema ModCluster](#) en el JBoss EAP *Guía de configuración*.

4.20. VISUALIZACIÓN DE CAMBIOS DE CONFIGURACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 7 proporciona la capacidad de rastrear los cambios de configuración realizados en el servidor en ejecución. Esto permite a los administradores ver un historial de cambios de configuración realizados por usuarios autorizados.

En JBoss EAP 7.0, debe usar **core-service** comando CLI de administración para configurar las opciones y para enumerar los cambios de configuración recientes.

Ejemplo: cambios de configuración de lista en JBoss EAP 7.0 [Esto es una traducción automática]

```
/core-service=management/service=configuration-changes:add(max-history=10)
/core-service=management/service=configuration-changes:list-changes
```

JBoss EAP 7.1 introdujo un nuevo **core-management** subsistema que se puede configurar para rastrear los cambios de configuración realizados en el servidor en ejecución. Este es el método preferido para configurar y ver los cambios de configuración en JBoss EAP 7.1.

Ejemplo: cambios de configuración de lista en JBoss EAP 7.1 [Esto es una traducción automática]

```
/subsystem=core-management/service=configuration-changes:add(max-history=20)
/subsystem=core-management/service=configuration-changes:list-changes
```

Para obtener más información sobre el uso de la nueva **core-management** subsistema introducido en JBoss EAP 7.1, ver [Ver cambios de configuración](#) en el JBoss EAP *Guía de configuración*.

CAPÍTULO 5. CAMBIOS EN LA MIGRACIÓN DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

5.1. CAMBIOS EN LA APLICACIÓN DE SERVICIOS WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBossWS 5 trae nuevas características y mejoras de rendimiento a los servicios web de JBoss EAP 7, principalmente a través de actualizaciones del [Apache CXF](#), [Apache WSS4J](#), y [Apache Santuario](#) componentes.

5.1.1. Cambios de soporte de JAX-RPC [Esto es una traducción automática]

La API de Java para RPC basado en XML (JAX-RPC) fue desaprobada en Java EE 6 y es opcional en Java EE 7. Ya no está disponible o no es compatible con JBoss EAP 7. Las aplicaciones que usan JAX-RPC deben migrarse para usar [JAX-WS](#), que es el marco actual de servicios web estándar de Java EE.

El uso de los servicios web JAX-RPC se puede identificar de cualquiera de las siguientes maneras:

- La presencia de un archivo de mapeo JAX-RPC, que es un archivo XML con el elemento raíz `<java-wsdl-mapping>`.
- La presencia de un `webservicex.xml` Archivo de descriptor XML que contiene un `<web-service-description>` elemento, que incluye un `<jaxrpc-mapping-file>` elemento niño. El siguiente es un ejemplo de `webservicex.xml` archivo de descriptor que define un servicio web JAX-RPC.

```
<webservicex xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://www.ibm.com/webservices/xsd/j2ee_web_services_1_1.xsd"
  version="1.1">
  <web-service-description>
    <web-service-description-name>HelloService</web-service-
description-name>
    <wsdl-file>WEB-INF/wsdl/HelloService.wsdl</wsdl-file>
    <jaxrpc-mapping-file>WEB-INF/mapping.xml</jaxrpc-mapping-file>
    <port-component>
      <port-component-name>Hello</port-component-name>
      <wsdl-port>HelloPort</wsdl-port>
      <service-endpoint-
interface>org.jboss.chap12.hello.Hello</service-endpoint-interface>
      <service-impl-bean>
        <servlet-link>HelloWorldServlet</servlet-link>
      </service-impl-bean>
    </port-component>
  </web-service-description>
</webservicex>
```

- La presencia de un `ejb-jar.xml` archivo, que contiene un `<service-ref>` que hace referencia a un archivo de mapeo JAX-RPC.

5.1.2. Cambios en los servicios web de Apache CXF Spring [Esto es una traducción automática]

En lanzamientos anteriores de JBoss EAP, podía personalizar la integración de JBossWS y Apache CXF incluyendo un `jbossws-cxf.xml` archivo de configuración con el archivo de implementación de punto final. Un caso de uso para esto fue configurar cadenas de interceptor para clientes del servicio web y puntos finales del servidor en el bus Apache CXF. Esta integración requirió que Spring se implementara en el servidor JBoss EAP.

La integración de Spring ya no es compatible con JBoss EAP 7. Cualquier aplicación que contenga un `jbossws-cxf.xml` El archivo de configuración del descriptor debe modificarse para reemplazar la configuración personalizada definida en ese archivo. Si bien aún es posible acceder directamente a la API de Apache CXF, tenga en cuenta que la aplicación no será portátil.

El enfoque sugerido es reemplazar las configuraciones personalizadas de Spring con las nuevas opciones de configuración del descriptor JBossWS siempre que sea posible. El enfoque basado en el descriptor JBossWS proporciona una funcionalidad similar sin requerir la modificación del código del punto final del cliente. En algunos casos, puede reemplazar Spring con Context Dependency Injection (CDI).

Migrar interceptores de mensajería [Esto es una traducción automática]

El descriptor JBossWS proporciona nuevas opciones de configuración que le permiten declarar los interceptores sin modificar el código de punto final del cliente. En su lugar, declara los interceptores dentro de las configuraciones predefinidas de cliente y punto final al especificar una lista de nombres de clase de interceptor para el `cxf.interceptors.in` y `cxf.interceptors.out` propiedades.

El siguiente es un ejemplo de `jaxws-endpoint-config.xml` archivo que declara interceptores usando estas propiedades.

```
<?xml version="1.0" encoding="UTF-8"?>
<jaxws-config xmlns="urn:jboss:jbossws-jaxws-config:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:javaee="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="urn:jboss:jbossws-jaxws-config:4.0 schema/jbossws-
jaxws-config_4_0.xsd">
  <endpoint-config>
    <config-
name>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointImpl</config-name>
    <property>
      <property-name>cxf.interceptors.in</property-name>
      <property-
value>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointInterceptor,org.jbo
ss.test.ws.jaxws.cxf.interceptors.FooInterceptor</property-value>
    </property>
    <property>
      <property-name>cxf.interceptors.out</property-name>
      <property-
value>org.jboss.test.ws.jaxws.cxf.interceptors.EndpointCounterInterceptor<
/property-value>
    </property>
  </endpoint-config>
</jaxws-config>
```

Características de Apache CXF

El descriptor JBossWS le permite declarar características dentro de las configuraciones predefinidas de cliente y punto final al especificar una lista de nombres de clase de entidad para el **cxf.features** propiedad.

El siguiente es un ejemplo de **jaxws-endpoint-config.xml** archivo que declara una característica que usa esta propiedad.

```
<?xml version="1.0" encoding="UTF-8"?>
<jaxws-config xmlns="urn:jboss:jbossws-jaxws-config:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:javaee="http://java.sun.com/xml/ns/javaee"
  xsi:schemaLocation="urn:jboss:jbossws-jaxws-config:4.0 schema/jbossws-
jaxws-config_4_0.xsd">
  <endpoint-config>
    <config-name>Custom FI Config</config-name>
    <property>
      <property-name>cxf.features</property-name>
      <property-value>org.apache.cxf.feature.FastInfosetFeature</property-
value>
    </property>
  </endpoint-config>
</jaxws-config>
```

Transporte HTTP Apache CXF

En Apache CXF, la configuración de transporte HTTP se logra especificando **org.apache.cxf.transport.http.HTTPConduit** opciones. La integración de JBossWS permite que los conductos se modifiquen mediante programación utilizando la API de Apache CXF de la siguiente manera.

```
import org.apache.cxf.frontend.ClientProxy;
import org.apache.cxf.transport.http.HTTPConduit;
import org.apache.cxf.transports.http.configuration.HTTPClientPolicy;

// Set chunking threshold before using a JAX-WS port client
...
HTTPConduit conduit =
(HTTPConduit)ClientProxy.getClient(port).getConduit();
HTTPClientPolicy client = conduit.getClient();

client.setChunkingThreshold(8192);
...
```

También puede controlar y anular el Apache CXF **HTTPConduit** valores predeterminados al establecer las propiedades del sistema.

Propiedad	Tipo	Descripción
cxf.client.allowChunking	Booleano	Especifica si enviar solicitudes usando fragmentación.
cxf.client.chunkingThreshold	Entero	Establece el umbral en el que se cambia del modo sin fragmentación al modo de fragmentación.

Propiedad	Tipo	Descripción
<code>cxf.client.connectionTimeout</code>	Long	Establece la cantidad de milisegundos para el tiempo de espera de la conexión.
<code>cxf.client.receiveTimeout</code>	Long	Establece el número de milisegundos para el tiempo de espera de recepción.
<code>cxf.client.connection</code>	Cadena	Especifica si usar el Keep-Alive o close tipo de conexión.
<code>cxf.tls-client.disableCNCheck</code>	Booleano	Especifica si se deshabilita la verificación del nombre de host CN.

5.1.3. Cambios en WS-Security

- Si su aplicación contiene un controlador de devolución de llamada personalizado que accede al **`org.apache.ws.security.WSPasswordCallback`** clase, tenga en cuenta que esta clase se ha movido al paquete **`org.apache.wss4j.common.ext`**.
- La mayoría de los objetos de frijol SAML de la **`org.apache.ws.security.saml.ext`** paquete se ha movido a la **`org.apache.wss4j.common.saml`** package.
- Utilice el transporte de la clave RSA v1.5 y todos los algoritmos relacionados están deshabilitados por defecto.
- El servicio de token de seguridad (STS) anteriormente solo se validó **`onBehalfOf`** tokens. Ahora también valida **`ActAs`** tokens. Como consecuencia, se debe especificar un nombre de usuario y una contraseña válidos en el **`UsernameToken`** que se proporciona para el **`ActAs`** simbólico.
- Los tokens de portador SAML ahora requieren una firma interna. los **`org.apache.wss4j.dom.validate.SamlAssertionValidator`** clase ahora tiene una **`setRequireBearerSignature()`** método para habilitar o deshabilitar la verificación de la firma.

5.1.4. Cambio de estructura de JBoss Modules [Esto es una traducción automática]

los **`cxf-api`** y **`cxf-rt-core`** Los JAR se han fusionado en uno **`cxf-core`** TARRO. Como consecuencia, el **`org.apache.cxf`** módulo en JBoss EAP ahora contiene el **`cxf-core`** JAR y expone más clases que en la versión anterior.

5.1.5. Requisitos y cambios del castillo hinchable [Esto es una traducción automática]

Si desea utilizar el cifrado AES con el modo Galois / Counter (GCM) para el cifrado simétrico en XML / WS-Security, necesita el proveedor de seguridad BouncyCastle.

JBoss EAP 7 se envía con el **`org.bouncycastle`** módulo y JBossWS ahora puede confiar en su cargador de clases para obtener y usar el proveedor de seguridad BouncyCastle. Por lo tanto, ya no es

necesario instalar estáticamente BouncyCastle en la JVM actual. Para aplicaciones que se ejecutan fuera del contenedor, el proveedor de seguridad puede estar disponible para JBossWS agregando una biblioteca BouncyCastle a la ruta de clase.

Puede deshabilitar este comportamiento configurando `org.jboss.ws.cxf.noLocalBC` valor de propiedad para `true` en el `jaxws-endpoint-config.xml` archivo descriptor de despliegue para el servidor o el `jaxws-client-config.xml` archivo descriptor para clientes.

Si desea utilizar una versión diferente a la que se envía con JBoss EAP, aún puede instalar BouncyCastle estáticamente en la JVM. En ese caso, el proveedor de seguridad BouncyCastle instalado de forma estática se elige sobre el proveedor presente en la ruta de clase. Para evitar cualquier problema, debe usar BouncyCastle 1.49, 1.51 o superior.

5.1.6. Estrategia de selección de bus Apache CXF [Esto es una traducción automática]

La estrategia de selección de bus predeterminada para los clientes que se ejecutan en el contenedor ha cambiado desde `THREAD_BUS` a `TCCL_BUS`. Para los clientes que se quedan sin contenedor, la estrategia predeterminada sigue siendo `THREAD_BUS`. Puede restablecer el comportamiento al de la versión anterior mediante uno de los siguientes métodos.

- Arranque el servidor JBoss EAP con la propiedad del sistema `org.jboss.ws.cxf.jaxws-client.bus.strategy` valor establecido en `THREAD_BUS`.
- Establezca explícitamente la estrategia de selección en el código del cliente.

5.1.7. Requisitos de JAX-WS 2.2 para WebServiceRef [Esto es una traducción automática]

Los contenedores deben usar los constructores de estilo JAX-WS 2.2, que incluyen `WebServiceFeature` clase como argumento en el constructor, para construir clientes que se inyectan en las referencias del servicio web. JBoss EAP 6.4, que se envía con JBossWS 4, oculta ese requisito. JBoss EAP 7 se envía con JBossWS 5, que ya no oculta este requisito. Esto significa que las clases de servicio proporcionadas por el usuario deben implementar JAX-WS 2.2 o posterior al actualizar el código existente para usar el `javax.xml.ws.Service` constructor que incluye uno o más `WebServiceFeature` argumentos.

```
protected Service(URL wsdlDocumentLocation,
                  QName serviceName,
                  WebServiceFeature... features)
```

5.1.8. IgnoreHttpsHost CN Check Change [Esto es una traducción automática]

En versiones anteriores, podía deshabilitar la comprobación del nombre de host de HTTPS URL con el nombre común (CN) de un servicio que figura en su certificado estableciendo la propiedad del sistema. `org.jboss.security.ignoreHttpsHost` a `true`. Este nombre de propiedad del sistema ha sido reemplazado por `cxf.tls-client.disableCNCheck`.

5.1.9. Configuración del lado del servidor y carga de clases [Esto es una traducción automática]

Como consecuencia de habilitar las inyecciones en los puntos finales del servicio y los controladores del cliente del servicio, ya no es posible cargar automáticamente clases de controladores desde el

org.jboss.as.webservices.server.integration Módulo JBoss. Si su aplicación depende de una configuración predefinida determinada, es posible que deba definir de forma explícita nuevas dependencias de módulos para su implementación. Para más información, ver [Migrar dependencias de módulos explícitos](#)

5.1.10. La depreciación de las normas endosadas de Java anula el mecanismo [Esto es una traducción automática]

los [Mecanismo de invalidación de estándares endosados por Java](#) se desaprobo en JDK 1.8_40 con la intención de eliminarlo en JDK 9. Este mecanismo permitió a los desarrolladores hacer que las bibliotecas estuvieran disponibles para todas las aplicaciones desplegadas colocando los JAR en un directorio endosado dentro del JRE.

Si su aplicación utiliza la implementación JBossWS de Apache CXF, JBoss EAP 7 garantiza que las dependencias requeridas se agreguen en el orden correcto y usted no debería verse afectado por este cambio. Si su aplicación accede a Apache CXF directamente, ahora debe proporcionar las dependencias de Apache CXF después de las dependencias de JBossWS como parte de la implementación de su aplicación.

5.1.11. Especificación del descriptor en EAR Archive [Esto es una traducción automática]

En versiones anteriores de JBoss EAP, puede configurar el **jboss-webservices.xml** archivo descriptor de despliegue para implementaciones de servicios web EJB en el **META-INF/** directorio de archivos JAR o en el **WEB-INF/** directorio para las implementaciones del servicio web POJO y los puntos finales del servicio web EJB incluidos en los archivos WAR.

En JBoss EAP 7, ahora puede configurar el **jboss-webservices.xml** archivo de descriptor de despliegue en **META-INF/** directorio de un archivo EAR. Si un **jboss-webservices.xml** archivo se encuentra tanto en el archivo EAR y el archivo JAR o WAR, los datos de configuración en el **jboss-webservices.xml** archivo para el JAR o WAR anula los datos correspondientes en el archivo de descriptor EAR.

5.2. ACTUALICE EL PUERTO Y EL CONECTOR REMOTO DE URL [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 7, el conector predeterminado ha cambiado de **remote** a **http-remoting** y el puerto de conexión remota predeterminado ha cambiado desde **4447** a **8080**. El URL del proveedor JNDI para la configuración predeterminada ha cambiado desde **remote://localhost:4447** a **http-remoting://localhost:8080**.

Si usa el JBoss EAP 7 **migrate** operación para actualizar su configuración, no necesita modificar el conector remoto, el puerto remoto o las URL de los proveedores JNDI porque la operación de migración preserva el conector remoto JBoss EAP 6 y **4447** configuraciones de puerto en la configuración del subsistema. Para más información sobre el **migrate** operación, ver [Gestión de la migración de la CLI Operación](#).

Si no usas el **migrate** operación y en su lugar se ejecuta con la nueva configuración predeterminada de JBoss EAP 7, debe cambiar el conector remoto, el puerto remoto y la URL del proveedor JNDI para usar la nueva configuración como se describe arriba.

5.3. CAMBIOS EN LA APLICACIÓN DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

5.3.1. Reemplace o actualice los descriptores de implementación de JMS [Esto es una traducción automática]

Los archivos de descriptores de distribución de recursos HornetQ JMS patentados identificados por el patrón de nomenclatura -**jms.xml** ya no funciona en JBoss EAP 7. El siguiente es un ejemplo de un archivo de descriptor de implementación de recursos JMS en JBoss EAP 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<messaging-deployment xmlns="urn:jboss:messaging-deployment:1.0">
  <hornetq-server>
    <jms-destinations>
      <jms-queue name="testQueue">
        <entry name="queue/test"/>
        <entry name="java:jboss/exported/jms/queue/test"/>
      </jms-queue>
      <jms-topic name="testTopic">
        <entry name="topic/test"/>
        <entry name="java:jboss/exported/jms/topic/test"/>
      </jms-topic>
    </jms-destinations>
  </hornetq-server>
</messaging-deployment>
```

Si usaste -**jms.xml** Los descriptores de implementación JMS en su aplicación en la versión anterior, puede convertir su aplicación para usar el descriptor de implementación estándar de Java EE 7 como se especifica en la sección EE.5.18 del [Especificación Java EE 7](#) o puede actualizar el descriptor de implementación para usar **messaging-activemq-deployment** esquema en su lugar.

Si elige actualizar el descriptor, debe hacer las siguientes modificaciones.

- Cambia el espacio de nombres de "urn: jboss: messaging-deployment: 1.0" a "urn: jboss: messaging-activemq-deployment: 1.0".
- Cambiar el **<hornetq-server>** nombre del elemento a **<server>**.

El archivo modificado debería verse como el siguiente ejemplo.

```
<?xml version="1.0" encoding="UTF-8"?>
<messaging-deployment xmlns="urn:jboss:messaging-activemq-deployment:1.0">
  <server>
    <jms-destinations>
      <jms-queue name="testQueue">
        <entry name="queue/test"/>
        <entry name="java:jboss/exported/jms/queue/test"/>
      </jms-queue>
      <jms-topic name="testTopic">
        <entry name="topic/test"/>
        <entry name="java:jboss/exported/jms/topic/test"/>
      </jms-topic>
    </jms-destinations>
  </server>
</messaging-deployment>
```

```

    </jms-destinations>
  </server>
</messaging-deployment>

```

Para obtener información sobre los cambios de configuración del servidor relacionados con la mensajería, consulte [Cambios en la configuración del servidor de mensajería](#).

5.3.2. Actualizar clientes externos JMS [Esto es una traducción automática]

JBoss EAP 7 sigue siendo compatible con la API JMS 1.1, por lo que no necesita modificar su código.

El conector remoto y el puerto predeterminados han cambiado en JBoss EAP 7. Para detalles sobre este cambio, vea [Actualice el puerto y el conector remoto de URL](#).

Si migra la configuración de su servidor usando el **migrate** operación, la configuración anterior se conserva y no necesita actualizar su **PROVIDER_URL**. Sin embargo, si ejecuta con la nueva configuración predeterminada de JBoss EAP 7, debe cambiar el **PROVIDER_URL** en el código del cliente para usar el nuevo **http-remoting://localhost:8080** ajuste. Para más información, ver [Migrar código de nombre de cliente remoto](#).

Si planea migrar su código para usar la API JMS 2.0, consulte la **helloworld-jms** inicio rápido para un ejemplo de trabajo.

5.3.3. Reemplazar la API HornetQ [Esto es una traducción automática]

JBoss EAP 6 incluyó el **org.hornetq** módulo, que le permitió usar el [API HornetQ](#) en el código fuente de tu aplicación.

Apache ActiveMQ Artemis reemplaza HornetQ en JBoss EAP 7, por lo que debe migrar cualquier código que use la API HornetQ para usar el [API Apache ActiveMQ Artemis](#). Las bibliotecas para esta API están incluidas en **org.apache.activemq.artemis** módulo.

ActiveMQ Artemis es una evolución de HornetQ, por lo que muchos de los conceptos se siguen aplicando.

5.4. CAMBIOS DE APLICACIÓN JAX-RS Y RESTEasy [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 6 incluyó RESTEasy 2, que era una implementación de JAX-RS 1.x.

JBoss EAP 7.0 incluye RESTEasy 3, que es una implementación de JAX-RS 2.0 según lo definido por el [JSR 339: JAX-RS 2.0: la API de Java para los servicios web RESTful](#) especificación. Para obtener más información sobre la API de Java para los servicios web RESTful, consulte la [Especificación JAX-RS 2.0 API](#).

La versión de Jackson incluida en JBoss EAP ha cambiado. JBoss EAP 6 incluye Jackson 1.9.9. JBoss EAP 7 y versiones posteriores ahora incluyen Jackson 2.6.3 o superior.

Esta sección describe cómo estos cambios pueden afectar las aplicaciones que usan RESTEasy o JAX-RS.

5.4.1. RESETClases obsoletas [Esto es una traducción automática]

Clases de Interceptor y Cuerpo de Mensaje

JSR 311: JAX-RS: la API de Java TM para servicios web RESTful no incluyó un marco interceptor, por lo que RESTEasy 2 proporcionó uno. JSR 339: JAX-RS 2.0: la API de Java para los servicios web RESTful introdujo un interceptor oficial y un marco de filtro, por lo que el marco de interceptor incluido en RESTEasy 2 ahora está en desuso, y se reemplazó por el recurso interceptor compatible con JAX-RS 2.0 en RESTEasy 3.x. Las interfaces relevantes se definen en `javax.ws.rs.ext` paquete de la `jaxrs-api` módulo.

- Las siguientes interfaces de interceptor están en desuso en RESTEasy 3.x.
 - `org.jboss.resteasy.spi.interception.PreProcessInterceptor`
 - `org.jboss.resteasy.spi.interception.PostProcessInterceptor`
 - `org.jboss.resteasy.spi.interception.ClientExecutionInterceptor`
 - `org.jboss.resteasy.spi.interception.ClientExecutionContext`
 - `org.jboss.resteasy.spi.interception.AcceptedByMethod`
- los `org.jboss.resteasy.spi.interception.PreProcessInterceptor` la interfaz es reemplazada por `javax.ws.rs.container.ContainerRequestFilter` interfaz en RESTEasy 3.x.
- Las siguientes interfaces y clases también están en desuso en RESTEasy 3.x.
 - `org.jboss.resteasy.spi.interception.MessageBodyReaderInterceptor`
 - `org.jboss.resteasy.spi.interception.MessageBodyWriterInterceptor`
 - `org.jboss.resteasy.spi.interception.MessageBodyWriterContext`
 - `org.jboss.resteasy.spi.interception.MessageBodyReaderContext`
 - `org.jboss.resteasy.core.interception.InterceptorRegistry`
 - `org.jboss.resteasy.core.interception.InterceptorRegistryListener`
 - `org.jboss.resteasy.core.interception.ClientExecutionContextImpl`
- los `org.jboss.resteasy.spi.interception.MessageBodyWriterInterceptor` la interfaz es reemplazada por `javax.ws.rs.ext.WriterInterceptor` interfaz.
- Además, algunos cambios en el `javax.ws.rs.ext.MessageBodyWriter` la interfaz puede no ser compatible con respecto a JAX-RS 1.x. Si su aplicación utilizó JAX-RS 1.x, revise el código de su aplicación para asegurarse de definir `@Produces` o `@Consumes` para tus puntos finales. De lo contrario, podría producirse un error similar al siguiente.

```
org.jboss.resteasy.core.NoMessageBodyWriterFoundFailure: Could not
find MessageBodyWriter for response object of type: <OBJECT> of
media type:
```

El siguiente es un ejemplo de un punto final REST que puede causar este error.

```
@Path("dates")
public class DateService {
```

```

    @GET
    @Path("daysuntil/{targetdate}")
    public long showDaysUntil(@PathParam("targetdate") String
targetDate) {
        DateLogger.LOGGER.logDaysUntilRequest(targetDate);
        final long days;

        try {
            final LocalDate date = LocalDate.parse(targetDate,
DateTimeFormatter.ISO_DATE);
            days = ChronoUnit.DAYS.between(LocalDate.now(), date);
        } catch (DateTimeParseException ex) {
            // ** DISCLAIMER **. This example is contrived.
            throw new
WebApplicationException(Response.status(400).entity(ex.getLocalizedMessage())
essage()).type(MediaType.TEXT_PLAIN)
                .build());
        }
        return days;
    }
}

```

Para solucionar el problema, agregue la importación para **javax.ws.rs.Produces** y el **@Produces** anotación de la siguiente manera.

```

...
import javax.ws.rs.Produces;
...

@Path("dates")
public class DateService {

    @GET
    @Path("daysuntil/{targetdate}")
    @Produces(MediaType.TEXT_PLAIN)
    public long showDaysUntil(@PathParam("targetdate") String
targetDate) {
        DateLogger.LOGGER.logDaysUntilRequest(targetDate);
        final long days;

        try {
            final LocalDate date = LocalDate.parse(targetDate,
DateTimeFormatter.ISO_DATE);
            days = ChronoUnit.DAYS.between(LocalDate.now(), date);
        } catch (DateTimeParseException ex) {
            // ** DISCLAIMER **. This example is contrived.
            throw new
WebApplicationException(Response.status(400).entity(ex.getLocalizedMessage())
essage()).type(MediaType.TEXT_PLAIN)
                .build());
        }
        return days;
    }
}

```



NOTA

Todos los interceptores de la versión anterior de RESTEasy pueden ejecutarse en paralelo con el nuevo filtro JAX-RS 2.0 e interfaces de interceptor.

Para obtener más información acerca de los interceptores, consulte [Interceptores RESTEasy](#) en *Desarrollo de aplicaciones de servicios web* para JBoss EAP.

Para obtener más información sobre la nueva API de reemplazo, consulte la [RESTEasy JAX-RS 3.0.13.Final API](#) o después.

API cliente

El marco de trabajo RESTEasy en **resteasy-jaxrs** ha sido reemplazado por el cumplimiento de JAX-RS 2.0 **resteasy-client** módulo. Como resultado, algunas clases y métodos de API de cliente RESTEasy están en desuso.

- Las siguientes clases están en desuso.
 - [org.jboss.resteasy.client.ClientRequest](#)
 - [org.jboss.resteasy.client.ClientRequestFactory](#)
 - [org.jboss.resteasy.client.ClientResponse](#)
 - [org.jboss.resteasy.client.ProxyBuilder](#)
 - [org.jboss.resteasy.client.ProxyConfig](#)
 - [org.jboss.resteasy.client.ProxyFactory](#)
- los [org.jboss.resteasy.client.ClientResponseFailure](#) excepción y la [org.jboss.resteasy.client.ClientExecutor](#) y [org.jboss.resteasy.client.EntityTypeFactory](#) las interfaces también están en desuso.
- Debes reemplazar el [org.jboss.resteasy.client.ClientRequest](#) y [org.jboss.resteasy.client.ClientResponse](#) clases con [org.jboss.resteasy.client.jaxrs.ResteasyClient](#) y [javax.ws.rs.core.Response](#) respectivamente.

El siguiente es un ejemplo de cómo enviar un encabezado de enlace con el cliente RESTEasy en RESTEasy 2.3.x.

```
ClientRequest request = new
ClientRequest(generateURL("/linkheader/str"));
request.addLink("previous chapter", "previous",
"http://example.com/TheBook/chapter2", null);
ClientResponse response = request.post();
LinkHeader header = response.getLinkHeader();
```

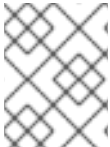
El siguiente es un ejemplo de cómo lograr la misma tarea con el cliente RESTEasy en RESTEasy 3.

```
ResteasyClient client = new ResteasyClientBuilder().build();
Response response =
client.target(generateURL("/linkheader/str")).request()
```

```
.header("Link", "<http://example.com/TheBook/chapter2>;  
rel=\"previous\";  
title=\"previous chapter\"").post(Entity.text(new String()));  
javax.ws.rs.core.Link link = response.getLink("previous");
```

Ver el **resteasy-jaxrs-client** quickstart para obtener un ejemplo de un cliente externo JAX-RS RESTEasy que interactúa con un servicio web JAX-RS.

- Las clases y las interfaces en el [org.jboss.resteasy.client.cache](#) paquete también están en desuso. Se reemplazan por clases e interfaces equivalentes en el [org.jboss.resteasy.client.jaxrs.cache](#) paquete.



NOTA

Para más información sobre el [org.jboss.resteasy.client.jaxrs](#) Clases de API, vea el [RESTEasy JAX-RS JavaDoc](#).

StringConverter

los [org.jboss.resteasy.spi.StringConverter](#) la clase está obsoleta en RESTEasy 3.x. Esta funcionalidad puede ser reemplazada usando el JAX-RS 2.0 [jax.ws.rs.ext.ParamConverterProvider](#) clase.

5.4.2. Clases RESTEasy eliminadas o protegidas [Esto es una traducción automática]

ResteasyProviderFactory Agregar métodos

La mayoría de [org.jboss.resteasy.spi.ResteasyProviderFactory](#) `add()` los métodos se han eliminado o protegido en RESTEasy 3.0. Por ejemplo, el `addBuiltInMessageBodyReader()` y `addBuiltInMessageBodyWriter()` métodos se han eliminado y el `addMessageBodyReader()` y `addMessageBodyWriter()` los métodos han sido protegidos.

Ahora debería usar el `registerProvider()` y `registerProviderInstance()` métodos.

Clases adicionales eliminadas de RESTEasy 3

los [@org.jboss.resteasy.annotations.cache.ServerCached](#) la anotación, que especificaba que la respuesta al método JAX-RS debe almacenarse en caché en el servidor, se eliminó de RESTEasy 3 y debe eliminarse del código de la aplicación.

5.4.3. Cambios RESTEasy adicionales [Esto es una traducción automática]

SignedInput y SignedOutput

- **SignedInput** y **SignedOutput** para **resteasy-crypto** debe tener el **Content-Type** ajustado a **multipart/signed** ya sea en el **Request** o **Response** objeto, o mediante el uso de **@Consumes** o **@Produces** anotación.
- **SignedOutput** y **SignedInput** se puede usar para devolver el **application/pkcs7-signature** Formato de tipo MIME en forma binaria configurando ese tipo en **@Produces** o **@Consumes** anotaciones
- Si el **@Produces** o **@Consumes** es **text/plain** Tipo MIME, **SignedOutput** se codificará en base64 y se enviará como una cadena.

Cambios en WS-Security

Los filtros de seguridad para **@RolesAllowed**, **@PermitAll**, y **@DenyAll** ahora devuelve "403 Prohibido" en lugar de "401 no autorizado".

Filtros del lado del cliente

Los nuevos filtros JAX-RS 2.0 del lado del cliente no se vincularán y ejecutarán cuando utilice la API del cliente RESTEasy de la versión anterior.

Soporte HTTP asíncrono

Porque la especificación JAX-RS 2.0 agrega soporte HTTP asíncrono usando el **@Suspended** anotación y el **AsynResponse** interfaz, la API patentada RESTEasy para HTTP asíncrono ha quedado obsoleta y podría eliminarse en una futura versión RESTEasy. Los módulos asíncronos Tomcat y Asíncronos de JBoss Web también se han eliminado de la instalación del servidor. Si no está utilizando el contenedor Servlet 3.0 o superior, se simulará el procesamiento asíncrono del lado del servidor HTTP y se ejecutará de forma síncrona en el mismo hilo de solicitud.

Caché del lado del servidor

La configuración de la memoria caché del lado del servidor ha cambiado. Por favor mira el [Documentación RESTEasy](#) para más información.

Cambios de Jackson Provider [Esto es una traducción automática]

En versiones anteriores de JBoss EAP, la configuración del proveedor RESTEasy YAML estaba habilitada de manera predeterminada. Esto ha cambiado en JBoss EAP 7. El proveedor YAML ahora está deshabilitado por defecto. Su uso no es compatible debido a un problema de seguridad en el **SnakeYAML** biblioteca utilizada por RESTEasy para desasignación y debe estar explícitamente habilitada en la aplicación. Para obtener información sobre cómo habilitar el proveedor YAML en su aplicación y agregar las dependencias Maven, consulte [Proveedor de YAML](#) en *Desarrollo de aplicaciones de servicios web* para JBoss EAP.

Charset predeterminado UTF-8 en el encabezado de tipo de contenido

En JBoss EAP 7.1, el **resteasy.add.charset** parámetro está configurado para **true** por defecto. Puedes configurar el **resteasy.add.charset** parámetro a **false** si no quieres que RESTEasy agregue **charset=UTF-8** al encabezado de tipo de contenido devuelto cuando el método de recurso devuelve un **text/*** o **application/xml*** tipo de medio sin un conjunto de caracteres explícito.

Para obtener más información acerca de los tipos de medios de texto y conjuntos de caracteres, consulte [Tipos de medios de texto y juegos de caracteres](#) en *Desarrollo de aplicaciones de servicios web* para JBoss EAP.

SerializableProvider

Deserializar objetos Java de fuentes no confiables no es seguro. Por esta razón, en JBoss EAP 7, el **org.jboss.resteasy.plugins.providers.SerializableProvider** la clase está deshabilitada de forma predeterminada, y no se recomienda utilizar este proveedor.

Coincidencia de solicitudes a métodos de recursos

En RESTEasy 3, se han realizado mejoras y correcciones en la implementación de las reglas de coincidencia, tal como se define en la especificación JAX-RS 2.0. En particular, se realizó un cambio en cómo se maneja un URI ambiguo en un método de sub-recursos y un localizador de sub-recursos.

En RESTEasy 2, era posible que un localizador de sub-recursos se ejecutara con éxito incluso cuando había otro sub-recurso con el mismo URI. Este comportamiento fue incorrecto según la especificación.

En RESTEasy 3, cuando hay un URI ambiguo para un sub-recurso y un localizador de sub-recursos, la invocación del sub-recurso será exitosa; sin embargo, llamar al localizador de sub-recursos dará como resultado un estado HTTP **405 Method Not Allowed** error.

El siguiente ejemplo contiene un ambiguo **@Path** anotación en un método de sub-recurso y un

localizador de sub-recursos. Observe que el URI para ambos extremos, **anotherResource** y **anotherResourceLocator**, es el mismo. La diferencia entre los dos puntos finales es que **anotherResource** método está asociado con el verbo REST, **POST**. los **anotherResourceLocator** método no está asociado con ningún verbo REST. De acuerdo con la especificación, el punto final con el verbo REST, en este caso el **anotherResource** método, siempre será seleccionado.

```
@Path("myResource")
public class ExampleSubResources {
    @POST
    @Path("items")
    @Produces("text/plain")
    public Response anotherResource(String text) {
        return Response.ok("ok").build();
    }

    @Path("items")
    @Produces("text/plain")
    public SubResource anotherResourceLocator() {
        return new SubResource();
    }
}
```

5.4.4. RESTEasy SPI Cambios [Esto es una traducción automática]

Excepciones de SPI

Todas las excepciones de falla de SPI han quedado obsoletas y ya no se usan internamente. Han sido reemplazados con la excepción JAX-RS 2.0 correspondiente.

Excepción obsoleta	Excepción de reemplazo en <code>jaxrs-api</code> módulo
<code>org.jboss.resteasy.spi.ForbiddenException</code>	<code>javax.ws.rs.ForbiddenException</code>
<code>org.jboss.resteasy.spi.MethodNotAllowedException</code>	<code>javax.ws.rs.NotAllowedException</code>
<code>org.jboss.resteasy.spi.NotAcceptableException</code>	<code>javax.ws.rs.NotAcceptableException</code>
<code>org.jboss.resteasy.spi.NotFoundException</code>	<code>javax.ws.rs.NotFoundException</code>
<code>org.jboss.resteasy.spi.UnauthorizedException</code>	<code>javax.ws.rs.NotAuthorizedException</code>
<code>org.jboss.resteasy.spi.UnsupportedMediaTypeException</code>	<code>javax.ws.rs.NotSupportedException</code>

InjectorFactory y registro

los **InjectorFactory** y **Registry** SPI han cambiado. Esto no debería ser un problema si usa RESTEasy como está documentado y es compatible.

5.4.5. Cambios de Jackson Provider [Esto es una traducción automática]

La versión de Jackson incluida en JBoss EAP ha cambiado. La versión anterior de JBoss EAP incluía Jackson 1.9.9. JBoss EAP 7 ahora incluye Jackson 2.6.3 o superior. Como resultado, el proveedor de Jackson ha cambiado de **resteasy-jackson-provider** a **resteasy-jackson2-provider**.

La actualización a la **resteasy-jackson2-provider** requiere algunos cambios en el paquete. Por ejemplo, el paquete de anotación de Jackson ha cambiado de **org.codehaus.jackson.annotate** a **com.fasterxml.jackson.annotation**.

Para cambiar su aplicación y usar el proveedor predeterminado que se incluyó en la versión anterior de JBoss EAP, consulte [Cambiar el proveedor predeterminado de Jackson](#) en *Desarrollo de aplicaciones de servicios web* para JBoss EAP.

5.4.6. Cambios en la integración RESTEasy de primavera [Esto es una traducción automática]

El marco de Spring 4.0 presentó soporte para Java 8. Si planea usar la integración de RESTEasy 3.x con Spring, asegúrese de especificar 4.2.x como la versión mínima de Spring en su implementación, ya que esta es la primera versión estable compatible con JBoss EAP. 7.

5.4.7. Cambios RESTEasy Jettison JSON Provider [Esto es una traducción automática]

El proveedor RESTEasy Jettison JSON está en desuso en JBoss EAP 7 y ya no se agrega a las implementaciones de forma predeterminada. Se le recomienda cambiar al proveedor recomendado de RESTEasy Jackson. Si prefiere continuar usando el proveedor de Jettison, debe definir una dependencia explícita para él en el **jboss-deployment-descriptor.xml** archivo como se demuestra en el siguiente ejemplo.

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-deployment-structure>
  <deployment>
    <exclusions>
      <module name="org.jboss.resteasy.resteasy-jackson2-provider"/>
      <module name="org.jboss.resteasy.resteasy-jackson-provider"/>
    </exclusions>
    <dependencies>
      <module name="org.jboss.resteasy.resteasy-jettison-provider"
services="import"/>
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

Para obtener más información acerca de cómo definir dependencias explícitas, consulte [Agregar una dependencia de módulo explícito a una implementación](#) en el JBoss EAP *Guía de desarrollo*.

5.5. CAMBIOS DE LA APLICACIÓN CDI 1.2 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 7 incluye soporte para CDI 1.2. Como resultado, las aplicaciones escritas usando CDI 1.0 podrían ver algunos cambios en el comportamiento cuando se migren a JBoss EAP 7. Esta sección resume solo algunos de esos cambios.

Puede encontrar más información sobre Weld y CDI 1.2 en las siguientes referencias:

- [Inyección de contextos y dependencia para la plataforma Java EE](#)
- [CDI 1.2 Javadoc](#)
- [Weld 2.3.3.Final - Implementación de referencia de CDI](#)

Archivos de Frijol

Las clases de frijoles habilitados Bean deben implementarse en archivos de frijoles para garantizar que sean escaneados por CDI para encontrar y procesar las clases de frijoles.

En CDI 1.0, un archivo se definió como *explícito* archivo de frijol si contenía un **beans.xml** archivo en el **META-INF/** directorio para un cliente de aplicación, EJB o JAR de biblioteca, o si contenía un **beans.xml** archivo en el **WEB-INF/** directorio para una GUERRA.

CDI 1.1 introducido *implícito* archivos de beans, que son archivos que contienen una o más clases de beans con una anotación que define bean, o uno o más beans de sesión. Los archivos de frijoles implícitos son escaneados por CDI y, durante el descubrimiento de tipos, solo se descubren las clases con anotaciones que definen los beans. Para más información, ver [Tipo y Bean Discovery](#) en *Inyección de contextos y dependencia para la plataforma Java EE*.

Un archivo bean tiene un modo de descubrimiento de frijoles **all**, **annotated** o **none**. Un archivo de frijoles que contiene un **beans.xml** archivo sin versión tiene un modo de descubrimiento de frijol predeterminado de **all**. Un archivo de frijoles que contiene un **beans.xml** archivo con versión **1.1** o más tarde debe especificar el **bean-discovery-mode** atributo. El valor predeterminado para el atributo es **annotated**.

Un archivo no es un archivo de beans en los siguientes casos:

- Contiene un **beans.xml** archivo con un **bean-discovery-mode** de **none**.
- Contiene una extensión CDI sin **beans.xml** archivo.

Un archivo es un *explícito* archivo de frijol en los siguientes casos:

- El archivo contiene un **beans.xml** archivo con un número de versión de 1.1 o posterior y una **bean-discovery-mode** de **all**.
- El archivo contiene un **beans.xml** archivo sin número de versión.
- Ejemplo: bandera de anotación en el **jboss-deployment-structure.xml** Archivo [Esto es una traducción automática]

Un archivo es un *implícito* archivo de frijol en los siguientes casos:

- El archivo contiene una o más clases de beans con una anotación que define bean, o uno o más beans de sesión, incluso si no contiene un **beans.xml** archivo.
- El archivo contiene un **beans.xml** archivo con un **bean-discovery-mode** de **annotated**.

CDI 1.2 limitado [bean definiendo anotaciones](#) a lo siguiente:

- **@ApplicationScoped**, **@SessionScoped**, **@ConversationScoped**, y **@RequestScoped** anotaciones
- Todos los demás tipos de alcance normal

- Ejemplo: **@DateBridge** y **@CalendarBridge** Anotación [Esto es una traducción automática]
- Todas las anotaciones de estereotipos, que son anotaciones anotadas con **@Stereotype**
- Ejemplo: **@IndexedEmbedded** Anotación [Esto es una traducción automática]

Para obtener más información acerca de los archivos de frijol, consulte [Archivos de Frijol](#) en *Inyección de contextos y dependencia para la plataforma Java EE*.

Aclaración de la resolución de conversación

El ciclo de vida del contexto de la conversación se modificó para evitar conflictos con la especificación del servlet como se describe en [CDI-411, número de especificación CDI-411](#). El ámbito de conversación está activo durante todas las solicitudes de servlet y no debe impedir que otros servlets o filtros de servlet configuren el cuerpo de solicitud o la codificación de caracteres.

Resolución del observador

La resolución del evento se ha reescrito en parte en CDI 1.2. En CDI 1.0, un evento se entrega a un método de observador si el método del observador tiene todos los calificadores de eventos. En CDI 1.2, un evento se entrega a un método de observador si el método del observador no tiene calificadores de eventos o tiene un subconjunto de los calificadores de eventos.

5.6. MIGRAR DEPENDENCIAS DE MÓDULOS EXPLÍCITOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La introducción del sistema de carga de clases modular y JBoss Modules en la versión anterior de JBoss EAP permitió un control detallado de las clases disponibles para las aplicaciones. Esta característica le permitió configurar dependencias explícitas de módulos utilizando la aplicación **MANIFEST.MF** archivo o el **jboss-deployment-structure.xml** archivo descriptor de despliegue

Si definió dependencias explícitas de módulos en su aplicación, debe tener en cuenta los siguientes cambios en JBoss EAP 7.

Revise las dependencias para la disponibilidad

Los módulos que están incluidos en JBoss EAP han cambiado. Cuando migre su aplicación a JBoss EAP 7, revise su **MANIFEST.MF** y **jboss-deployment-structure.xml** archivo de entradas para asegurarse de que no se refieran a ningún módulo que se haya eliminado de esta versión del producto.

Dependencias que requieren exploración de anotación

En el lanzamiento anterior de JBoss EAP, si su dependencia contenía anotaciones que debían procesarse durante el análisis de anotación, como al declarar Interceptor EJB, se le requería generar e incluir un índice Jandex en un nuevo archivo JAR y luego establecer un indicador en el **MANIFEST.MF** o **jboss-deployment-structure.xml** archivo descriptor de despliegue

JBoss EAP 7 ahora ofrece generación automática en tiempo de ejecución de índices de anotación para módulos estáticos, por lo que ya no es necesario generarlos manualmente. Sin embargo, aún necesita agregar el **annotations** bandera a la aplicación **MANIFEST.MF** archivo o el **jboss-deployment-structure.xml** archivo descriptor de despliegue como se muestra a continuación.

Ejemplo: bandera de anotación en el MANIFEST.MF Archivo [Esto es una traducción automática]

Dependencias: com.company.my-ejb annotations, com.company.other

Ejemplo: bandera de anotación en el `jboss-deployment-structure.xml` Archivo [Esto es una traducción automática]

```
<jboss-deployment-structure>
  <deployment>
    <dependencies>
      <module name="com.company.my-ejb" annotations="true" />
      <module name="com.company.other" />
    </dependencies>
  </deployment>
</jboss-deployment-structure>
```

5.7. CAMBIOS DE MIGRACIÓN DE HIBERNATE Y JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]**5.7.1. Hibernate ORM 3.0 [Esto es una traducción automática]**

Las clases de integración que facilitaron el uso de Hibernate ORM 3 en la versión anterior fueron eliminadas de JBoss EAP 7. Si su aplicación todavía usa las bibliotecas Hibernate ORM 3, se recomienda encarecidamente que migre su aplicación para usar Hibernate ORM 5 como Hibernate ORM 3 ya no funcionará en JBoss EAP sin mucho esfuerzo. Si no puede migrar a Hibernate ORM 5, debe definir un módulo JBoss personalizado para Hibernate ORM 3 JAR y excluir las clases Hibernate ORM 5 de su aplicación.

5.7.2. Hibernate ORM 4.0 - 4.3 [Esto es una traducción automática]

Si su aplicación necesita caché de segundo nivel habilitada, debe migrar a Hibernate ORM 5, que está integrado con Infinispan 8.x.

Las aplicaciones escritas con Hibernate ORM 4.x aún pueden usar Hibernate ORM 4.x. Debe definir un módulo JBoss personalizado para los JAR de Hibernate ORM 4.x y excluir las clases de Hibernate ORM 5 de su aplicación. Sin embargo, se recomienda encarecidamente que reescriba el código de la aplicación para usar Hibernate ORM 5. Para obtener información sobre la migración a Hibernate ORM 5, consulte [Migración a Hibernate ORM 5](#).

5.7.3. Migración a Hibernate ORM 5 [Esto es una traducción automática]

Esta sección destaca los cambios que debe realizar al migrar desde la versión 4.3 de Hibernate ORM a la versión 5. Para obtener más información acerca de los cambios implementados entre Hibernate ORM 4 e Hibernate ORM 5, consulte el [Guía de migración de Hibernate ORM 5.0](#).

RESETClases obsoletas [Esto es una traducción automática]

Las siguientes clases en desuso se eliminaron de Hibernate ORM 5:

- [org.hibernate.cfg.AnnotationConfiguration](#)
- [org.hibernate.id.TableGenerator](#)
- [org.hibernate.id.TableHiLoGenerator](#)
- [org.hibernate.id.SequenceGenerator](#)

Otros cambios a clases y paquetes

- los `org.hibernate.integrator.spi.Integrator` la interfaz cambió para dar cuenta del rediseño del bootstrap.
- Un nuevo paquete `org.hibernate.engine.jdbc.env.spi` fue creado. Contiene el `org.hibernate.engine.jdbc.env.spi.JdbcEnvironment` interfaz, que se extrajo de la `org.hibernate.engine.jdbc.spi.JdbcServices` interfaz.
- Un nuevo `org.hibernate.boot.model.relational.ExportableProducer` interfaz fue introducida que afectará `org.hibernate.id.PersistentIdentifierGenerator` implementaciones.
- La firma de `org.hibernate.id.Configurable` fue cambiado para aceptar `org.hibernate.service.ServiceRegistry` en lugar de solo `org.hibernate.dialect.Dialect`.
- los `org.hibernate.metamodel.spi.TypeContributor` la interfaz ha migrado a `org.hibernate.boot.model.TypeContributor`.
- los `org.hibernate.metamodel.spi.TypeContributions` la interfaz ha migrado a `org.hibernate.boot.model.TypeContributions`.

Tipo de manejo

- Incorporado `org.hibernate.type.descriptor.sql.SqlTypeDescriptor` las implementaciones ya no se auto-registran con `org.hibernate.type.descriptor.sql.SqlTypeDescriptorRegistry`. Aplicaciones que usan personalizadas `SqlTypeDescriptor` las implementaciones que amplían las implementaciones incorporadas y dependen de ese comportamiento deben actualizarse para llamar `SqlTypeDescriptorRegistry.addDescriptor()` sí mismos.
- Para los ID definidos como UUID generados, algunas bases de datos requieren que establezca explícitamente el `@Column(length=16)` para generar `BINARY(16)` para que las comparaciones funcionen correctamente
- por `EnumType` mapeos definidos en `hbm.xml`, Donde quieras `javax.persistence.EnumType.STRING name-mapping`, esta configuración debe establecerse explícitamente mediante el uso `useNamed(true)` configuración o especificando un `VARCHAR` valor de `12`.

Cambios en el subsistema de transacciones [Esto es una traducción automática]

- La transacción SPI experimentó un importante rediseño en Hibernate ORM 5. En Hibernate ORM 4.3, utilizó el `org.hibernate.Transaction` API para acceder directamente a diferentes estrategias de transacciones de back-end. Hibernate ORM 5 introdujo un nivel de indirección. En el back-end, el `org.hibernate.Transaction` la implementación ahora habla a un `org.hibernate.resource.transaction.TransactionCoordinator`, que representa el contexto transaccional de una sesión determinada según la estrategia de back-end. Si bien esto no tiene un impacto directo en los desarrolladores, podría afectar la configuración de arranque. Anteriormente, las aplicaciones especificarían `hibernate.transaction.factory_class` propiedad, que ahora está en desuso, y se refieren a un `org.hibernate.engine.transaction.spi.TransactionFactory` FQN (nombre completo). Con Hibernate ORM 5, especifica el `hibernate.transaction.coordinator_class` configurar y consultar un

[org.hibernate.resource.transaction.TransactionCoordinatorBuilder](#). Ver [org.hibernate.cfg.AvailableSettings.TRANSACTION_COORDINATOR_STRATEGY](#) para detalles adicionales.

- Los siguientes nombres cortos ahora son reconocidos:
 - **jdbc**: Gestione transacciones utilizando el JDBC [java.sql.Connection](#). Este es el valor predeterminado para las transacciones que no son de JPA.
 - **jta**: Gestione transacciones usando JTA.



IMPORTANTE

Si una aplicación JPA no proporciona una configuración para el **hibernate.transaction.coordinator_class** propiedad, Hibernate construirá automáticamente el coordinador de transacción adecuado en función del tipo de transacción para la unidad de persistencia.

Si una aplicación que no es JPA no proporciona una configuración para el **hibernate.transaction.coordinator_class** propiedad, Hibernate se convertirá en **jdbc** para administrar las transacciones. Este valor predeterminado causará problemas si la aplicación realmente utiliza transacciones basadas en JTA. Una aplicación no JPA que utiliza transacciones basadas en JTA debe establecer explícitamente el **hibernate.transaction.coordinator_class** valor de propiedad para **jta** o proporcionar una costumbre [org.hibernate.resource.transaction.TransactionCoordinatorBuilder](#) que construye un [org.hibernate.resource.transaction.TransactionCoordinator](#) que coordina adecuadamente con las transacciones basadas en JTA.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

- los **cfg.xml** los archivos se vuelven a analizar completamente y se integran con eventos, seguridad y otras funciones.
- Las propiedades cargadas desde **cfg.xml** utilizando el **EntityManagerFactory** previamente no prefijaba nombres con **hibernate**. Esto ahora se ha hecho consistente.
- La configuración ya no es serializable.
- los [org.hibernate.dialect.Dialect.getQuerySequencesString\(\)](#) método ahora recupera los valores de catálogo, esquema e incremento.
- los **AuditConfiguration** el modificador fue eliminado de [org.hibernate.envers.boot.internal.EnversService](#).
- los **AuditStrategy** los parámetros del método se cambiaron para eliminar el obsoleto **AuditConfiguration** y usa el nuevo **EnversService**.
- Varias clases e interfaces en el [org.hibernate.hql.spi](#) paquete y subpaquetes se han movido a la nueva [org.hibernate.hql.spi.id](#) paquete. Esto incluye el **MultiTableBulkIdStrategy** clase y el **AbstractTableBasedBulkIdHandler**, **TableBasedDeleteHandlerImpl**, y **TableBasedUpdateHandlerImpl** interfaces y sus subclases.

- Hubo un rediseño completo de los contratos de acceso a la propiedad.
- Válido `hibernate.cache.default_cache_concurrency_strategy` los valores de configuración ahora se definen usando el `org.hibernate.cache.spi.access.AccessType.getExternalName()` método en lugar de la `org.hibernate.cache.spi.access.AccessType` constantes enum. Esto es más consistente con otras configuraciones de Hibernate.

5.7.4. Migración de Hibernate ORM 5.0 a Hibernate ORM 5.1 [Esto es una traducción automática]

Mientras que JBoss EAP 7.0 incluyó Hibernate ORM 5.0, JBoss EAP 7.1 ahora incluye Hibernate ORM 5.1. Esta sección resalta las diferencias y los cambios necesarios al migrar desde la versión 5.0 de Hibernate ORM a la versión 5.1.

Hibernate ORM 3.0 [Esto es una traducción automática]

Esta versión incluye muchas mejoras de rendimiento y correcciones de errores, que se detallan en [Funciones de Hibernate ORM 5.1](#) en el *JBoss EAP 7.1.0 Notas de la versión*. Para obtener información adicional sobre los cambios implementados entre Hibernate ORM 5.0 e Hibernate ORM 5.1, consulte el [Hibernate ORM 5.1 Guía de migración](#).

Cambios en la administración de JMX [Esto es una traducción automática]

Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]

Los cambios de herramientas de gestión de esquemas en Hibernate ORM 5.1 se centran principalmente en las siguientes áreas:

- Unificando el manejo de `hbm2ddl.auto` y JPA de Hibernate `schema-generation` apoyo.
- Elimina las preocupaciones de JDBC del SPI para facilitar el reemplazo verdadero de Hibernate OGM, un motor de persistencia que proporciona compatibilidad con Java Persistence (JPA) para almacenes de datos NoSQL.

Los cambios de herramientas de administración de esquemas solo deberían ser un problema de migración para las aplicaciones que usan directamente cualquiera de las siguientes clases:

- `org.hibernate.tool.hbm2ddl.SchemaExport`
- `org.hibernate.tool.hbm2ddl.SchemaUpdate`
- `org.hibernate.tool.hbm2ddl.SchemaValidator`
- `org.hibernate.tool.schema.spi.SchemaManagementTool`, o cualquiera de sus delegados

Cambios en la configuración de mensajería en JBoss EAP 7.1 [Esto es una traducción automática]

Hibernate ORM 5.1.10 que se incluye en JBoss EAP 7.1, introdujo una nueva estrategia para recuperar tablas de bases de datos que mejora `SchemaMigrator` y `SchemaValidator` actuación. Esta estrategia ejecuta un solo `java.sql.DatabaseMetaData#getTables(String, String, String, String[])` llamar para determinar si cada `javax.persistence.Entity` tiene una tabla de base de datos mapeada. Esta es la estrategia predeterminada, y usa el `hibernate.hbm2ddl.jdbc_metadata_extraction_strategy=grouped` ajuste de propiedad. Esta estrategia puede requerir `hibernate.default_schema` y / o `hibernate.default_catalog` ser provisto.

Para usar la vieja estrategia, que ejecuta un `java.sql.DatabaseMetaData#getTables(String,`

`String, String, String[])` pedir `each javax.persistence.Entity`, utilizar el `hibernate.hbm2ddl.jdbc_metadata_extraction_strategy=individually` ajuste de propiedad.

5.8. CAMBIOS EN LA BÚSQUEDA DE HIBERNATE [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

La versión de Hibernate Search que se envía con JBoss EAP 7 ha cambiado. El lanzamiento anterior de JBoss EAP se envió con Hibernate Search 4.6.x. JBoss EAP 7 se envía con Hibernate Search 5.5.x.

Hibernate Search 5.5 está basado en Apache Lucene 5.3.1. Si usa cualquier API nativa de Lucene, asegúrese de alinearse con esta versión. los [Hibernate Search 5.5 API](#) envuelve y oculta la complejidad de muchos de los cambios de la API de Lucene realizados entre la versión 3 y la versión 5; sin embargo, algunas clases ahora están en desuso, renombradas o reempaquetadas. Esta sección describe cómo estos cambios pueden afectar el código de su aplicación.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Indexación de id. Campos de relaciones incrustadas

Cuando se usa un `@IndexedEmbedded` anotación para incluir campos de una entidad relacionada, el `id` de la entidad relacionada ya no está incluido. Puede habilitar la inclusión de `id` usando el `includeEmbeddedObjectId` atributo de la `@IndexedEmbedded` anotación.

Ejemplo: `@IndexedEmbedded` Anotación [Esto es una traducción automática]

```
@IndexedEmbedded(includeEmbeddedObjectId=true)
```

Cambios de migración de Hibernate y JPA [Esto es una traducción automática]

Los números y fechas ahora están indexados como campos numéricos de forma predeterminada. Propiedades del tipo `int`, `long`, `float`, `double`, y sus clases contenedoras correspondientes ya no se indexan como cadenas. En cambio, ahora están indexados usando la codificación numérica apropiada de Lucene. los `id` los campos son una excepción a esta regla. Incluso cuando están representados por un tipo numérico, todavía están indexados como una palabra clave de cadena de forma predeterminada. El uso de `@NumericField` ahora está obsoleto a menos que desee especificar una precisión personalizada para la codificación numérica. Puede mantener el antiguo formato de índice basado en cadenas especificando explícitamente un puente de campo de codificación de cadenas. En el caso de los enteros, este es el [org.hibernate.search.bridge.builtin.IntegerBridge](#). Comprobar el [org.hibernate.search.bridge.builtin](#) paquete para otros puentes de campo disponibles públicamente.

`Date` y `Calendar` ya no están indexados como cadenas. En cambio, las instancias se codifican como valores largos que representan el número de milisegundos desde el 1 de enero de 1970, 00:00:00 GMT. Puede cambiar el formato de indexación utilizando el nuevo [EncodingType](#) enum. Por ejemplo:

Ejemplo: `@DateBridge` y `@CalendarBridge` Anotación [Esto es una traducción automática]

```
@DateBridge(encoding=EncodingType.STRING)
@CalendarBridge(encoding=EncodingType.STRING)
```

El cambio de codificación para números y fechas es importante y puede tener un gran impacto en el comportamiento de la aplicación. Si tiene una consulta que se dirige a un campo que anteriormente estaba codificado en cadena, pero ahora está codificado numéricamente, debe actualizar la consulta.

Los campos numéricos se deben buscar con un **NumericRangeQuery**. También debe asegurarse de que todos los campos segmentados por faceting estén codificados en cadena. Si usa DSL de consulta de búsqueda, la consulta correcta se debe crear automáticamente.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

- Las opciones de ordenación han mejorado y la codificación de campo especificada incorrectamente para las opciones de clasificación ahora da como resultado excepciones de tiempo de ejecución. Lucene también ofrece una clasificación más eficiente si los campos utilizados en el género se conocen desde el principio. Hibernate Search 5.5 proporciona el nuevo **@SortableField** anotación y su compañero de múltiples valores **@SortableFields**. Ver el [Guía de migración de Hibernate Search 5.4 a 5.5](#) para más información.
- El Lucene **SortField** API requiere el siguiente cambio de código de la aplicación. En el lanzamiento anterior de JBoss EAP, estableces el tipo del campo de ordenación en la consulta de la siguiente manera.

```
fulltextQuery.setSort(new Sort(new SortField("title",
SortField.STRING)));
```

El siguiente es un ejemplo de cómo lo configuraste en JBoss EAP 7.

```
fulltextQuery.setSort(new Sort(new SortField("title",
SortField.Type.STRING)));
```

- Ya que **SearchFactory** solo debe ser utilizado por la integración de ORM, se movió de la **hibernate-search-engine** módulo a la **hibernate-search-orm** módulo. Otros integradores deberían depender exclusivamente de **SearchIntegrator**, que reemplaza al obsoleto **SearchFactoryIntegrator**.
- El valor enum **SpatialMode.GRID** fue renombrado a **SpatialMode.HASH**.
- FullTextIndexEventListener** ahora es una clase final. Si actualmente extiende esta clase, debe encontrar una solución alternativa para lograr la misma funcionalidad.
- los **hibernate-search-analyzers** módulo fue eliminado. El enfoque recomendado es usar directamente el artefacto Lucene apropiado, por ejemplo **org.apache.lucene:lucene-analyzers-common**.
- La API del controlador JMS ha cambiado. La dependencia del back-end de JMS en Hibernate ORM se eliminó para que pueda ser utilizada en otros entornos que no sean ORM. Una consecuencia es que los implementadores de **org.hibernate.search.backend.impl.jms.AbstractJMSHibernateSearchController** debe ajustarse a la nueva firma. Esta clase es una clase interna y se recomienda usarla como ejemplo en lugar de extenderla.
- los **org.hibernate.search.spi.ServiceProvider** SPI fue refactorizado. Si estuviera integrando con el antiguo contrato de servicio, consulte el [Hibernate Search 5.5 Javadoc](#) de **ServiceManager**, **Service**, **Startable** y **Stoppable** para detalles sobre el nuevo contrato.
- Si ha conservado los índices generados por Lucene 3.x y no los ha reconstruido con Hibernate Search 5.0 o posterior, obtendrá una **IndexFormatTooOldException**. Se recomienda que reconstruya los índices con el indexador masivo. Si no puede hacer eso, intente usar Lucene

IndexUpgrader. Debe actualizar cuidadosamente las asignaciones de Hibernate Search en caso de que el comportamiento predeterminado haya cambiado. Para más información, vea el [Guía de migración de Apache Lucene](#).

- Apache Lucene se actualizó de 3.6 a 5.3 en JBoss EAP 7. Si su código importa el código Lucene directamente, consulte el [Guía de migración de Apache Lucene](#) para detalles de los cambios. También se puede encontrar información adicional en [Registro de cambios de Lucene](#).
- Cuando usas **@Field(indexNullAs=)** para codificar un valor de marcador nulo en el índice, el tipo de marcador debe ser compatible con todos los demás valores que están indexados en ese mismo campo. Por ejemplo, antes era posible codificar un valor nulo para campos numéricos usando una cadena "nula". Esto ya no está permitido. En su lugar, debe elegir un número para representar el **null** valor, como **-1**.
- Se realizaron mejoras significativas en el motor de facetas. La mayoría de los cambios no afectan la API. La única excepción notable es que ahora debe anotar cualquier campo que pretenda usar para facetar con el **@Facet** o **@Facets** anotación.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

La siguiente es una lista de clases de búsqueda de Hibernate que fueron reempaquetadas o renombradas.

Paquete anterior y clase	Nuevo paquete y clase
org.hibernate.search.Environment	org.hibernate.search.cfg.Environment
org.hibernate.search.FullTextFilter	org.hibernate.search.filter.FullTextFilter
org.hibernate.search.ProjectionConstants	org.hibernate.search.engine.ProjectionConstants
org.hibernate.search.SearchException	org.hibernate.search.exception.SearchException
org.hibernate.search.Version	org.hibernate.search.engine.Version

Lucene - Clases renombradas y reempaquetadas

Los analizadores de consultas se movieron a un nuevo módulo, lo que dio como resultado un cambio de empaquetado de **org.apache.lucene.queryParser.QueryParser** a **org.apache.lucene.queryparser.classic.QueryParser**.

Muchos de los analizadores Lucene fueron refactorizados, lo que generó cambios en el empaquetado. Ver el [Documentación de Apache Lucene](#) para encontrar los paquetes de reemplazo.

Algunas clases de utilidad de Apache Solr, por ejemplo **TokenizerFactory** o **TokenFilterFactory**, fueron trasladados a Apache Lucene. Si su aplicación usa esas utilidades o analizadores personalizados, debe encontrar el nuevo nombre del paquete en Apache Lucene.

Ver el [Guía de migración de Apache Lucene](#) para más información.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Para obtener la lista completa de las interfaces, clases, enumeraciones, tipos de anotación, métodos, constructores y constantes de enumeración en desuso de Hibernate Search, consulte el [Hibernate Search API desaprobada](#) documento.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Interfaz	Descripción
org.hibernate.search.store.IndexShardingStrategy	Obsoleto a partir de Hibernate Search 4.4. Puede ser eliminado en la Búsqueda 5. Uso ShardIdentifierProvider en lugar.
org.hibernate.search.store.Workspace	Esta interfaz se moverá y se debe considerar API no pública. Para más información, ver HSEARCH-1915 .

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Clase	Descripción
org.hibernate.search.filter.FilterKey	Las claves de filtro personalizadas están en desuso y están programadas para su eliminación en Hibernate Search 6. A partir de Hibernate Search 5.1, las claves para almacenar en caché los filtros Lucene se calculan automáticamente en función de los parámetros de filtro dados.
org.hibernate.search.filter.StandardFilterKey	Las claves de filtro personalizadas están en desuso y están programadas para su eliminación en Hibernate Search 6. A partir de Hibernate Search 5.1, las claves para almacenar en caché los filtros Lucene se calculan automáticamente en función de los parámetros de filtro dados.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Enum	Descripción
org.hibernate.search.annotations.FieldCacheType	Eliminar el CacheFromIndex anotación ya que está en desuso. Ver Hibernate Buscar Anotaciones desaprobadas .

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Anotación	Descripción
org.hibernate.search.annotations.CacheFromIndex	Eliminar la anotación No es necesario un reemplazo alternativo.
org.hibernate.search.annotations.Key	Las claves de caché de filtro personalizadas son una función obsoleta y están programadas para eliminarse en Hibernate Search 6. A partir de Hibernate Search 5.1, las claves de caché de filtro se determinan automáticamente en función de los parámetros de filtro, por lo que ya no es necesario proporcionar un objeto clave.

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Método	Descripción
<code>org.hibernate.search.FullTextSharedSessionBuilder.autoClose()</code>	Sin reemplazo
<code>org.hibernate.search.FullTextSharedSessionBuilder.autoClose(boolean)</code>	Sin reemplazo
<code>org.hibernate.search.cfg.IndexedMapping.cacheFromIndex(FieldCacheType...)</code>	Esto se eliminará sin reemplazo.
<code>org.hibernate.search.cfg.EntityDescriptor.getCacheInMemory()</code>	Esto se eliminará sin reemplazo.
<code>org.hibernate.search.cfg.ContainedInMapping.numericField()</code>	Invocar <code>field().numericField()</code> en lugar.
<code>org.hibernate.search.cfg.EntityDescriptor.setCacheInMemory(Map<String, Object>)</code>	Esto se eliminará sin reemplazo.
<code>org.hibernate.search.MassIndexer.threadsForSubsequentFetching(int)</code>	Este método será eliminado.
<code>org.hibernate.search.query.dsl.FuzzyContext.withThreshold(float)</code>	Utilizar <code>FuzzyContext.withEditDistanceUpTo(int)</code> .

Cambios en la búsqueda de Hibernate [Esto es una traducción automática]

Constructor	Descripción
<code>org.hibernate.search.cfg.NumericFieldMapping(PropertyDescriptor, EntityDescriptor, SearchMapping)</code>	Utilizar <code>NumericFieldMapping.NumericFieldMapping(String, PropertyDescriptor, EntityDescriptor, SearchMapping)</code> en lugar.

Cambios que afectan a los integradores avanzados

Esta sección describe los cambios que no son parte de la API pública. No deberían afectar al desarrollador promedio, ya que los integradores que acceden al marco Hibernate Search solo deben acceder a estos artefactos.

- los [IndexWriterSetting.MAX_THREAD_STATES](#) y [IndexWriterSetting.TERM_INDEX_INTERVAL](#) las constantes enum están en desuso. Afectan qué propiedades se leen de la configuración, por lo que el hecho de que falten significa que las propiedades de configuración como **`hibernate.search.Animals.2.indexwriter.term_index_interval = default`** ahora son ignorados El único efecto secundario es que la propiedad no se aplica.
- los [SearchFactoryIntegrator](#) la interfaz ahora está en desuso. Debería migrar

inmediatamente todo el código para usar **SearchIntegrator**.

- los **SearchFactoryBuilder** clase ahora está en desuso. Utilizar **SearchIntegrationBuilder** en lugar.
- los **HSQuery.getExtendedSearchIntegrator()** método ha sido desaprobado. Podría ser posible usar **SearchIntegrator**, pero es preferible eliminarlo por completo.
- los **DocumentBuilderIndexedEntity.getFieldCacheOption()** método ha sido desaprobado. No hay reemplazo.
- los **BuildContext.getIndexingStrategy()** método está en desuso. Utilizar **BuildContext.getIndexingMode()** en lugar.
- los **DirectoryHelper.getVerifiedIndexDir(String, Properties, boolean)** método está en desuso. Utilizar **DirectoryHelper.getVerifiedIndexPath(java.lang.String, java.util.Properties, boolean)** en lugar.
- La siguiente es una lista de clases de búsqueda de Hibernate que fueron reempaquetadas o renombradas.

Paquete anterior y clase	Nuevo paquete y clase
org.hibernate.search.engine.impl.SearchMappingBuilder	org.hibernate.search.engine.spi.SearchMappingHelper
org.hibernate.search.indexes.impl.DirectoryBasedIndexManager	org.hibernate.search.indexes.spi.DirectoryBasedIndexManager
org.hibernate.search.spi.MassIndexerFactory	org.hibernate.search.batchindexing.spi.MassIndexerFactory
org.hibernate.search.spi.SearchFactoryBuilder	org.hibernate.search.spi.SearchIntegratorBuilder
org.hibernate.search.spi.SearchFactoryIntegrator	org.hibernate.search.spi.SearchIntegrator

5.9. MIGRAR BEANS DE ENTIDAD A JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

El soporte para beans de entidad EJB es opcional en Java EE 7 y no son compatibles con JBoss EAP 7. Esto significa que la persistencia manejada por contenedor (CMP) y los beans de entidad de persistencia gestionada por bean (BMP) deben reescribirse para usar Java Persistence API (JPA) entidades cuando migra a JBoss EAP 7.

En lanzamientos anteriores de JBoss EAP, se crearon beans de entidad en el código fuente de la aplicación extendiendo el **javax.ejb.EntityBean** clase e implementando los métodos requeridos. Luego se configuraron en el **ejb-jar.xml** archivo. Se especificó un bean de entidad CMP utilizando un **<entity>** elemento que contenía un **<persistence-type>** elemento hijo con un valor de **Envase**. Se especificó un bean de entidad BMP usando un **<entity>** elemento que contenía un **<persistence-type>** elemento hijo con un valor de **Frijol**.

En JBoss EAP 7, debe reemplazar cualquier entidad CMP y BMP beans en su código con entidades Java Persistence API (JPA). Las entidades JPA se crean utilizando el `javax.persistence.*` clases y se definen en el `persistence.xml` archivo.

El siguiente es un ejemplo de una clase de entidad JPA.

```
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
// User is a keyword in some SQL dialects!
@Table(name = "MyUsers")
public class MyUser {
    @Id
    @GeneratedValue
    private Long id;

    @Column(unique = true)
    private String username;
    private String firstName;
    private String lastName;

    public Long getId() {
        return id;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
}
```

El siguiente es un ejemplo de `persistence.xml` archivo.

```
<persistence version="2.1"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://xmlns.jcp.org/xml/ns/persistence
        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
    <persistence-unit name="my-unique-persistence-unit-name">
```

```

    <properties>
      // properties...
    </properties>
  </persistence-unit>
</persistence>

```

Para ejemplos de funcionamiento de entidades JPA, vea el **bmt**, **cmt**, y **hibernate5** inicios rápidos que se envían con JBoss EAP 7.

5.10. CAMBIOS EN LA PROPIEDAD DE PERSISTENCIA JPA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Una nueva propiedad de persistencia, **jboss.as.jpa.deferdetach**, se agregó para proporcionar compatibilidad con el comportamiento de persistencia en versiones anteriores de JBoss EAP.

los **jboss.as.jpa.deferdetach** La propiedad controla si el contexto de persistencia con ámbito de transacción utilizado en un subproceso de transacción no JTA separa las entidades cargadas después de cada **EntityManager** invocación o si espera hasta que se cierra el contexto de persistencia, por ejemplo, cuando finaliza la invocación del bean de sesión. El valor de la propiedad está por defecto en **false**, lo que significa que las entidades se separan o borran después de cada **EntityManager** invocación. Este es el comportamiento predeterminado correcto como se define en el [Especificación JPA](#). Si el valor de la propiedad está configurado para **true**, las entidades no se separan hasta que se cierra el contexto de persistencia.

En JBoss EAP 5, la persistencia se comportó como si **jboss.as.jpa.deferdetach** la propiedad estaba configurada para **true**. Para obtener este mismo comportamiento al migrar su aplicación de JBoss EAP 5 a JBoss EAP 7, debe configurar el **jboss.as.jpa.deferdetach** valor de propiedad para **true** en tus **persistence.xml** como se muestra en el siguiente ejemplo.

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">
  <persistence-unit name="EAP5_COMPAT_PU">
    <jta-data-source>java:jboss/datasources/ExampleDS</jta-data-source>
    <properties>
      <property name="jboss.as.jpa.deferdetach" value="true" />
    </properties>
  </persistence-unit>
</persistence>

```

En JBoss EAP 6, la persistencia se comportó como si **jboss.as.jpa.deferdetach** la propiedad estaba configurada para **false**. Este es el mismo comportamiento que el visto en JBoss EAP 7, por lo que no es necesario realizar cambios cuando migre su aplicación.

5.10.1. Cambios en la propiedad de persistencia de JPA en JBoss EAP 7.1 [Esto es una traducción automática]

En JBoss EAP 7.0, la comprobación de errores del contexto de persistencia no sincronizada no fue tan estricta como debería haber sido en las siguientes áreas.

- Se permitió que un contexto de persistencia administrado por contenedor sincronizado utilizara un contexto de persistencia extendida no sincronizada que se asoció con una transacción JTA. En cambio, debería haber arrojado un **IllegalStateException** para evitar que se use el contexto de persistencia no sincronizada.

- Un contexto de persistencia no sincronizado especificado en un descriptor de despliegue se trató como sincronizado.

En adición, **PersistenceProperty** consejos en el **@PersistenceContext** fueron ignorados por error en JBoss EAP 7.0.

Estos problemas fueron abordados y corregidos en JBoss EAP 7.1. Debido a que estas actualizaciones pueden dar como resultado un cambio no deseado en el comportamiento de la aplicación, se introdujeron dos nuevas propiedades de unidad de persistencia en JBoss EAP 7.1 para proporcionar compatibilidad con versiones anteriores y preservar el comportamiento anterior.

Propiedad	Descripción
wildfly.jpa.skipmixedsyncypechecking	Esta propiedad desactiva la comprobación de errores. Solo se debe usar como una medida temporal para la compatibilidad con versiones anteriores en situaciones donde las aplicaciones funcionaron en JBoss EAP 7.0 y fallaron en JBoss EAP 7.1. Debido a que esta propiedad podría estar obsoleta en una versión futura, se recomienda que corrija el código de la aplicación tan pronto como sea posible.
wildfly.jpa.allowjoinedunsync	Esta propiedad es una alternativa a wildfly.jpa.skipmixedsyncypechecking . Permite que la aplicación trate los contextos de persistencia no sincronizados que están asociados con una transacción JTA como si fueran contextos de persistencia sincronizados.

5.11. MIGRAR CÓDIGO DE CLIENTE EJB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

5.11.1. Cambios del cliente EJB en JBoss EAP 7 [Esto es una traducción automática]

El conector remoto y el puerto predeterminados han cambiado en JBoss EAP 7. Para detalles sobre este cambio, vea [Actualice el puerto y el conector remoto de URL](#).

Si usaste el **migrate** operación para migrar la configuración de su servidor, se conservan las configuraciones anteriores y no es necesario realizar los cambios detallados a continuación. Sin embargo, si ejecuta con la nueva configuración predeterminada de JBoss EAP 7, debe realizar los siguientes cambios.

5.11.1.1. Actualice el puerto de conexión remota predeterminado [Esto es una traducción automática]

Cambiar el valor del puerto de conexión remota de **4447** a **8080** en el **jboss-ejb-client.properties** archivo.

Ejemplo: **jboss-ejb-client.properties** Archivo de propiedades [Esto es una traducción automática]

Ejemplo: JBoss EAP 6 jboss-ejb-client.properties Archivo [Esto es una traducción automática]

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=4447
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NOANONYMOUS=false
```

Ejemplo: JBoss EAP 7 jboss-ejb-client.properties Archivo [Esto es una traducción automática]

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=localhost
remote.connection.default.port=8080
remote.connection.default.connect.options.org.xnio.Options.SASL_POLICY_NOANONYMOUS=false
```

5.11.1.2. Actualice el conector predeterminado [Esto es una traducción automática]

Si está ejecutando la nueva configuración de JBoss EAP 7, el conector predeterminado ha cambiado de **remote** a **http-remoting**. Este cambio afecta a los clientes que utilizan bibliotecas de una versión de JBoss EAP y para conectarse al servidor en una versión diferente.

- Si una aplicación cliente utiliza la biblioteca del cliente EJB de JBoss EAP 6 y desea conectarse al servidor JBoss EAP 7, el servidor debe estar configurado para exponer un **remote** conector en un puerto que no sea **8080**. El cliente debe conectarse usando ese conector recién configurado.
- Una aplicación cliente que utiliza la biblioteca de cliente EJB de JBoss EAP 7 y desea conectarse al servidor JBoss EAP 6 debe saber que la instancia del servidor no utiliza el **http-remoting** conector y en su lugar utiliza una **remote** conector. Esto se logra definiendo una nueva propiedad de conexión del lado del cliente.

Ejemplo: remote Propiedad de conexión [Esto es una traducción automática]

```
remote.connection.default.protocol=remote
```

5.11.2. Migrar código de nombre de cliente remoto [Esto es una traducción automática]

Si está ejecutando la nueva configuración predeterminada de JBoss EAP 7, debe modificar su código de cliente para usar el nuevo puerto y conector remoto predeterminado.

El siguiente es un ejemplo de cómo se especificaron las propiedades de denominación remota en el código de cliente en JBoss EAP 6.

```
java.naming.factory.initial=org.jboss.naming.remote.client.InitialContextFactory
```

```
java.naming.provider.url=remote://localhost:4447
```

El siguiente es un ejemplo de cómo especificar las propiedades de nombres remotos en el código de cliente en JBoss EAP 7.

```
java.naming.factory.initial=org.wildfly.naming.client.WildFlyInitialContextFactory
java.naming.provider.url=http-remoting://localhost:8080
```

5.11.3. Cambios adicionales del cliente EJB introducidos en JBoss EAP 7.1 [Esto es una traducción automática]

Mientras que JBoss EAP 7.0 se envía con JBoss EJB Client 2.1.4, JBoss EAP 7.1 se envía con JBoss EJB Client 4.0.x, que incluye una serie de cambios en la API.

- los **org.ejb.client.EJBClientInvocationContext** clase ha agregado los siguientes nuevos métodos.

Método	Tipo	Descripción
isBlockingCaller()	boolean	Determine si esta invocación actualmente está bloqueando el hilo de llamada.
isClientAsync()	boolean	Determine si el método está marcado como cliente asíncrono, lo que significa que la invocación debe ser asíncrona independientemente de si el método del lado del servidor es asíncronico.
isIdempotent()	boolean	Determine si el método está marcado idempotent, lo que significa que el método se puede invocar más de una vez sin ningún efecto adicional.
setBlockingCaller(boolean)	void	Establezca si esta invocación actualmente está bloqueando el hilo de llamada.
setLocator(EJBLocator<T>)	<T> void	Establezca el localizador para el destino de invocación.

- los **org.ejb.client.EJBLocator** clase ha agregado los siguientes nuevos métodos.

Método	Tipo	Descripción
asStateful()	StatefulEJBLocator<T>	Devuelve este localizador como un localizador con estado, si es uno.
asStateless()	StatelessEJBLocator<T>	Devuelve este localizador como un localizador sin estado, si es uno.

Método	Tipo	Descripción
isEntity()	boolean	Determine si este es un localizador de entidades.
isHome()	boolean	Determine si este es un localizador de casa.
isStateful()	boolean	Determine si este es un localizador con estado.
isStateless()	boolean	Determine si este es un localizador sin estado.
withNewAffinity(Affinity)	abstract EJBLocator<T>	Crea una copia de este localizador, pero con la nueva afinidad dada.

- Un nuevo **org.ejb.client.EJBClientPermission** clase, que es una subclase de [java.security.Permission](#), se ha introducido para controlar el acceso a operaciones de EJB privilegiadas.
 - Proporciona los siguientes constructores.

- **EJBClientPermission(String name)**

- **EJBClientPermission(String name, String actions)**

- Proporciona los siguientes métodos.

Método	Tipo	Descripción
equals(EJBClientPermission obj)	boolean	Controla los EJBClientPermission objetos para la igualdad.
equals(Object obj)	boolean	Controla los Permission objetos para la igualdad.
equals(Permission obj)	boolean	Controla los Permission objetos para la igualdad.
getActions()	Cadena	Devuelve las acciones como una cadena.
hashCode()	Entero	Devuelve el valor del código hash para este Permission objeto.
implies(EJBClientPermission permission)	boolean	Comprueba si las acciones del permiso especificado son <i>implicado por</i> esta EJBClientPermission las acciones del objeto.

Método	Tipo	Descripción
implies(Permission permission)	boolean	Comprueba si las acciones del permiso especificado son <i>implicado por</i> esta Permission las acciones del objeto.

- Un nuevo **org.ejb.client.EJBMethodLocator** clase ha sido introducida para localizar un método EJB específico.

- Proporciona el siguiente constructor.

■ **EJBMethodLocator(String methodName, String... parameterTypeNames)**

- Proporciona los siguientes métodos.

Método	Tipo	Descripción
equals(EJBMethodLocator other)	boolean	Determine si este objeto es igual a otro.
equals(Object other)	boolean	Determine si este objeto es igual a otro.
forMethod(Method method)	static EJBMethodLocator	Obtenga un localizador de métodos para el método de reflexión dado.
getMethodName()	Cadena	Obtener el nombre del método.
getParameterCount()	Entero	Obtenga el conteo de parámetros.
getParameterTypeName(int index)	Cadena	Obtenga el nombre del parámetro en el índice dado.
hashCode()	Entero	Obtenga el código hash.

- Un nuevo **org.jboss.ejb.client.EJBReceiverInvocationContext.ResultProducer.Failed** clase ha sido introducida para casos de falla.

- Proporciona el siguiente constructor.

■ **Failed(Exception cause)**

- Proporciona los siguientes métodos.

Método	Tipo	Descripción
discardResult()	void	Deseche el resultado, lo que indica que no se utilizará.

Método	Tipo	Descripción
getResult()	Object	Obtenga el resultado

- Un nuevo **org.jboss.ejb.client.EJBReceiverInvocationContext.ResultProducer.Immediate** clase ha sido introducida para resultados inmediatos.

- Proporciona el siguiente constructor.

- **Failed(Exception cause)**

- Proporciona los siguientes métodos.

Método	Tipo	Descripción
discardResult()	void	Deseche el resultado, lo que indica que no se utilizará.
getResult()	Object	Obtenga el resultado

- Un nuevo **org.jboss.ejb.client.URIAffinity** clase, que es una subclase de **org.jboss.ejb.client.Affinity** ha sido introducido para la especificación de afinidad de URI.

- Se crea usando **Affinity.forUri(URI)**.

- Proporciona los siguientes métodos.

Método	Tipo	Descripción
equals(Affinity other)	boolean	Indica si otro objeto es igual a este.
equals(Object other)	boolean	Indica si otro objeto es igual a este.
equals(URIAffinity other)	boolean	Indica si otro objeto es igual a este.
getURI()	URI	Obtenga el URI asociado.
hashCode()	Ent	Obtenga el código hash.
toString()	Cadena	Devuelve una representación de cadena del objeto.

- los **org.jboss.ejb.client.EJBMetaDataImpl** clase ha desaprobado los siguientes métodos.

- **toAbstractEJBMetaData()**

- `LOADSTRUCTUREJBMetaData()`
- `EJBMetaDataImpl(AbstractEJBMetaData<?, ?>)`

5.12. MIGRE LOS CLIENTES PARA USAR EL ARCHIVO DE CONFIGURACIÓN DE WILDFLY [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Antes de la versión 7.1, las bibliotecas de cliente JBoss EAP, como EJB y naming, usaban diferentes estrategias de configuración. JBoss EAP 7.1 presenta el **wildfly-config.xml** archivo con el propósito de unificar todas las configuraciones de cliente en un único archivo de configuración, de forma similar a la forma en que se maneja la configuración del servidor.

Por ejemplo, antes de JBoss EAP 7.1, podría crear un nuevo **InitialContext** para un cliente Java EJB usando un **jboss-ejb-client.properties** archivo, o al establecer programáticamente las propiedades usando un **Properties** clase.

Ejemplo: jboss-ejb-client.properties Archivo de propiedades [Esto es una traducción automática]

```
remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=one
remote.connection.one.port=8080
remote.connection.one.host=127.0.0.1
remote.connection.one.username=quickuser
remote.connection.one.password=quick-123
```

En JBoss EAP 7.1, usted crea un **wildfly-config.xml** archivo en el **META-INF/** directorio del archivo del cliente. Esta es la configuración equivalente usando un **wildfly-config.xml** archivo.

Ejemplo: configuración equivalente usando el wildfly-config.xml Archivo [Esto es una traducción automática]

```
<configuration>
  <authentication-client xmlns="urn:elytron:1.0.1">
    <authentication-rules>
      <rule use-configuration="ejb"/>
    </authentication-rules>
    <authentication-configurations>
      <configuration name="ejb">
        <set-user-name name="quickuser"/>
        <credentials>
          <clear-password password="quick-123"/>
        </credentials>
      </configuration>
    </authentication-configurations>
  </authentication-client>
  <jboss-ejb-client xmlns="urn:jboss:wildfly-client-ejb:3.0">
    <connections>
      <connection uri="remote+http://127.0.0.1:8080" />
    </connections>
  </jboss-ejb-client>
</configuration>
```

Para obtener información acerca de cómo configurar la autenticación de cliente para Elytron Client utilizando el **wildfly-config.xml** archivo, ver [Configurar la autenticación del cliente con Elytron Client](#) en *Cómo configurar la gestión de identidad* para JBoss EAP.

Para obtener más información acerca de los tipos de configuraciones de clientes que se pueden hacer utilizando el **wildfly-config.xml** archivo, ver [Configuración del cliente usando el wildfly-config.xml Archivo](#) en el JBoss EAP *Guía de desarrollo*.

5.13. MIGRAR CONFIGURACIONES DE PLANES DE IMPLEMENTACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

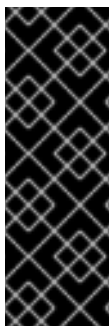
los [Especificación de implementación de aplicaciones Java EE \(JSR-88\)](#) fue diseñado para definir un contrato estándar que permita a las herramientas de múltiples proveedores configurar e implementar aplicaciones en cualquier producto de la plataforma Java EE. El contrato requería que los proveedores de productos Java EE implementaran el **DeploymentManager** y otra **javax.enterprise.deploy.spi** interfaces a las que deben acceder los proveedores de herramientas. En el caso de JBoss EAP 6, un plan de despliegue es identificado por un descriptor XML llamado **deployment-plan.xml** que se incluye en un archivo ZIP o JAR.

Esta especificación tuvo muy poca aceptación porque la mayoría de los productos de servidores de aplicaciones ofrecen sus propias soluciones de implementación más "ricas en funciones". Por este motivo, la compatibilidad con JSR-88 se eliminó de Java EE 7 y, a su vez, de JBoss EAP 7.

Si usó JSR-88 para implementar su aplicación, ahora debe usar otro método para implementar la aplicación. La CLI de administración de JBoss EAP **deploy** comando proporciona una forma estándar para implementar archivos en servidores independientes o grupos de servidores en un dominio administrado. Para obtener más información sobre la CLI de administración, consulte la [Guía CLI de gestión](#).

5.14. MIGRAR VÁLVULAS DE APLICACIONES PERSONALIZADAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Debe migrar manualmente las válvulas personalizadas o cualquier válvula que esté definida en el **jboss-web.xml** Archivo XML. Esto incluye las válvulas creadas al extender el **org.apache.catalina.valves.ValveBase** clase y configurado en el **<valve>** elemento de la **jboss-web.xml** archivo descriptor.



IMPORTANTE

Válvulas y válvulas personalizadas que están definidas en **jboss-web.xml** el archivo debe ser reescrito o reemplazado por el manejador incorporado de Undertow correspondiente. Para obtener información sobre la asignación de válvulas a los manipuladores de Undertow, consulte [Migrar las válvulas web JBoss](#).

Las válvulas de autenticación deben reemplazarse manualmente utilizando los mecanismos de autenticación integrados de Undertow.

Migrar configuraciones de SSL [Esto es una traducción automática]

En JBoss EAP 6, puede definir válvulas personalizadas en el nivel de aplicación configurándolas en **jboss-web.xml** archivo descriptor de la aplicación web. En JBoss EAP 7, también es posible hacer esto con los controladores de Undertow.

El siguiente es un ejemplo de una válvula configurada en `jboss-web.xml` archivo en JBoss EAP 6.

```
<jboss-web>
  <valve>
    <class-name>org.jboss.examples.MyValve</class-name>
    <param>
      <param-name>myParam</param-name>
      <param-value>foobar</param-value>
    </param>
  </valve>
</jboss-web>
```

Para obtener más información acerca de cómo crear y configurar manejadores personalizados en JBoss EAP, consulte [Crear manejadores personalizados](#) en el JBoss EAP *Guía de desarrollo*.

Migrar válvulas de autenticador [Esto es una traducción automática]

Para obtener información acerca de cómo migrar las válvulas de autenticación, consulte [Migrar válvulas de autenticador](#) en esta guía.

5.15. CAMBIOS DE APLICACIONES DE SEGURIDAD [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

El reemplazo de JBoss Web con Undertow requiere cambios en la configuración de seguridad en JBoss EAP 7.

5.15.1. Migrar válvulas de autenticador [Esto es una traducción automática]

Si creó una válvula autenticadora personalizada que se extendió `AuthenticatorBase` en JBoss EAP 6.4, debe reemplazarlo manualmente con una implementación de autenticación HTTP personalizada en JBoss EAP 7.1. El mecanismo de autenticación HTTP se crea en `elytron` subsistema y luego registrado con el `undertow` subsistema. Para obtener información sobre cómo implementar un mecanismo de autenticación HTTP personalizado, consulte [Implementación de un mecanismo HTTP personalizado](#) en el *Guía de desarrollo* para JBoss EAP.

5.15.2. Cambios de PicketLink [Esto es una traducción automática]

Para obtener información sobre los cambios necesarios para SSO con la configuración de SAML v2, consulte [Cambios de las versiones anteriores de JBoss EAP](#) en *Cómo configurar SSO con SAML v2* para JBoss EAP.

5.15.3. Otros cambios en la aplicación de seguridad [Esto es una traducción automática]

Para obtener información sobre las diferencias en la configuración de SSO con Kerberos, consulte [Diferencias con la configuración de versiones anteriores JBoss EAP](#) en *Cómo configurar SSO con Kerberos* para JBoss EAP.

5.16. CAMBIOS EN EL REGISTRO DE JBOSS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Si su aplicación usa JBoss Logging, tenga en cuenta que las anotaciones en el `org.jboss.logging` el paquete ahora está en desuso en JBoss EAP 7. Se han movido a la

org.jboss.logging.annotations paquete, por lo que debe actualizar su código fuente para importar el nuevo paquete.

Las anotaciones también se han movido a un Maven por separado **groupId:artifactId:version** (GAV) ID por lo que necesita agregar una nueva dependencia de proyecto para **org.jboss.logging:jboss-logging-annotations** en tu proyecto **pom.xml** archivo.



NOTA

Solo las anotaciones de registro se han movido. los **org.jboss.logging.BasicLogger** y **org.jboss.logging.Logger** todavía existen en el **org.jboss.logging** paquete.

La siguiente tabla enumera las clases de anotación desaprobadadas y las sustituciones correspondientes.

Tabla 5.1. Reemplazos de anotación de registro obsoletos [Esto es una traducción automática]

RESETClases obsoletas [Esto es una traducción automática]	Clase de reemplazo
org.jboss.logging.Cause	org.jboss.logging.annotations.Cause
org.jboss.logging.Field	org.jboss.logging.annotations.Field
org.jboss.logging.FormatWith	org.jboss.logging.annotations.FormatWith
org.jboss.logging.LoggingClass	org.jboss.logging.annotations.LoggingClass
org.jboss.logging.LogMessage	org.jboss.logging.annotations.LogMessage
org.jboss.logging.Message	org.jboss.logging.annotations.Message
org.jboss.logging.MessageBundle	org.jboss.logging.annotations.MessageBundle
org.jboss.logging.MessageLogger	org.jboss.logging.annotations.MessageLogger
org.jboss.logging.Param	org.jboss.logging.annotations.Param
org.jboss.logging.Property	org.jboss.logging.annotations.Property

5.17. CAMBIOS DE CÓDIGO DE JAVASERVER FACES (JSF) [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Soporte descartado para JSF 1.2

Ejemplo: bandera de anotación en el **jboss-deployment-structure.xml** Archivo [Esto es una traducción automática]

JBoss EAP 7 incluye JSF 2.2 y ya no admite la API JSF 1.2. Si su aplicación usa JSF 1.2, debe volver a escribirla para usar JSF 2.2.

Problema de compatibilidad entre JSF 2.1 y JSF 2.2

Las API JSF 2.1 y JSF 2.2 no son totalmente compatibles. los **FACELET_CONTEXT_KEY** valor constante cambiado de **com.sun.faces.facelets.FACELET_CONTEXT** a **javax.faces.FACELET_CONTEXT** entre los lanzamientos. Este valor está subrayado por el compilador y el código compilado contra una versión no funcionará en contra de la otra.

Las aplicaciones que contienen código similar al ejemplo siguiente y se compilan con la API JSF 2.1 pero se ejecutan en JBoss EAP 7, que utiliza la API JSF 2.2, dan como resultado un **NullPointerException**. Para solucionar el problema, debe volver a compilar la aplicación con la API de JSF 2.2.

Ejemplo: código Java que utiliza la API JSF 2.1 [Esto es una traducción automática]

```
Object obj =
FacesContext.getCurrentInstance().getAttributes().get(FaceletContext.FACELET_CONTEXT_KEY);
```

Ver [Evite que la constante FaceletContext.FACELET_CONTEXT_KEY esté en línea con el compilador](#) para más información.

5.18. CAMBIOS DE CARGA DE CLASE DE MÓDULO [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

En JBoss EAP 7, el comportamiento de carga de clase ha cambiado en los casos en que varios módulos contienen las mismas clases o paquetes.

Supongamos que hay dos módulos, **MODULE_A** y **MODULE_B**, que dependen el uno del otro y contienen algunos de los mismos paquetes. En JBoss EAP 6, las clases o paquetes que se cargaron desde las dependencias tuvieron prioridad sobre los especificados en el **resource-root** del **module.xml** archivo. Esto significaba **MODULE_A** ve los paquetes para **MODULE_B** y **MODULE_B** ve los paquetes para **MODULE_A**. Este comportamiento era confuso y podría causar conflictos. Este comportamiento ha cambiado en JBoss EAP 7. Ahora las clases o paquetes especificados por el **resource-root** en el **module.xml** archivo tiene prioridad sobre los especificados por la dependencia. Esto significa **MODULE_A** ve los paquetes para **MODULE_A** y **MODULE_B** ve los paquetes para **MODULE_B**. Esto evita conflictos y proporciona un comportamiento más apropiado.

Si ha definido módulos personalizados que incluyen **resource-root** bibliotecas o paquetes que contienen clases que están duplicadas en sus dependencias de módulos, es posible que vea **ClassCastException**, **LinkageError**, errores de carga de clases u otros cambios en el comportamiento cuando migra a JBoss EAP 7. Para resolver estos problemas, debe configurar su **module.xml** archivo para asegurar que solo se use una versión de una clase. Esto se puede lograr mediante el uso de cualquiera de los siguientes enfoques.

- Puede evitar especificar un **resource-root** que duplica las clases en la dependencia del módulo.
- Puedes usar el **include** y **exclude** subelementos de la **imports** y **exports** elementos para controlar la carga de clase en el **module.xml** archivo. El siguiente es un elemento de exportación que excluye clases está en el paquete especificado.

```
<exports>
  <exclude path="com/mycompany/duplicateclassespath"/>
</exports>
```

Si prefiere preservar su comportamiento existente, debe filtrar los paquetes de dependencia del dependiente **resource-root** en el **module.xml** archivo usando el **filter** elemento. Esto le permite conservar el comportamiento existente sin el extraño bucle que vería bajo JBoss EAP 6. El siguiente es un ejemplo de un **root-resource** que filtra las clases en un paquete específico.

```
<resource-root path="mycompany.jar">
  <filter>
    <exclude path="com/mycompany/duplicateclassespath"/>
  </filter>
</resource-root>
```

Para obtener más información acerca de los módulos y la carga de clase, consulte [Carga de clase y módulos](#) en el JBoss EAP *Guía de desarrollo*.

5.19. CAMBIOS EN LA AGRUPACIÓN DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

5.19.1. Descripción general de nuevas características de agrupamiento [Esto es una traducción automática]

La siguiente lista describe algunas de las nuevas características de clúster a tener en cuenta al migrar su aplicación de JBoss EAP 6 a JBoss EAP 7.

- JBoss EAP 7 presenta una nueva API pública para la construcción de servicios únicos que simplifica significativamente el proceso. Para obtener información sobre los servicios únicos, consulte [Servicio HA Singleton](#) en el JBoss EAP *Guía de desarrollo*
- Se puede configurar una implementación única para implementar e iniciar solo un nodo en el clúster a la vez. Para más información, ver [Despliegues HA Singleton](#) en el JBoss EAP *Guía de desarrollo*.
- Ahora puede definir MDB de singleton agrupados. Para más información, ver [Clustered Singleton MDBs](#) en *Desarrollo de aplicaciones EJB* para JBoss EAP.
- JBoss EAP 7 incluye la implementación Undertow mod_cluster. Esto ofrece una solución pura de equilibrio de carga de Java que no requiere un servidor web httpd. Para más información, ver [Configurando JBoss EAP como equilibrador de carga de front-end](#) en el JBoss EAP *Guía de configuración*.

El resto de esta sección describe cómo los cambios en la agrupación pueden afectar la migración de sus aplicaciones a JBoss EAP 7.

5.19.2. Cambios en clústeres de sesiones web [Esto es una traducción automática]

JBoss EAP 7 presenta una nueva implementación de clustering de sesión web. Reemplaza la implementación anterior, que estaba estrechamente relacionada con el código fuente del subsistema web JBoss.

La implementación de la nueva agrupación de sesiones web impacta la forma en que se configura la aplicación en el **jboss-web.xml** Archivo de descriptor XML de la aplicación web propiedad de JBoss EAP. Los siguientes son los únicos elementos de configuración de clúster que permanecen en este archivo.

■

```

<jboss-web>
...
<max-active-sessions>...</max-active-sessions>
...
<replication-config>
  <replication-granularity>...</replication-granularity>
  <cache-name>...</cache-name>
</replication-config>
...
</jboss-web>

```

La siguiente tabla describe cómo lograr un comportamiento similar para los elementos en el **jboss-web.xml** archivo que ahora está obsoleto.

Cambios de configuración del servidor [Esto es una traducción automática]	Cambios de aplicaciones de seguridad [Esto es una traducción automática]
<max-active-sessions/>	<p>Anteriormente, la creación de la sesión fallaba si causaba que el número de sesiones activas superara el valor especificado por <max-active-sessions/>.</p> <p>En la nueva implementación, <max-active-sessions/> se usa para habilitar la pasivación de la sesión. Si la creación de la sesión hará que el número de sesiones activas exceda el <max-active-sessions/>, entonces la sesión más antigua conocida por el administrador de sesiones se pasivará para dejar espacio para la nueva sesión.</p>
<passivation-config/>	Este elemento de configuración y sus subelementos ya no se usan en JBoss EAP 7.
<use-session-passivation/>	<p>Anteriormente, la pasivación se habilitó utilizando este atributo.</p> <p>En la nueva implementación, la pasivación se habilita especificando un valor no negativo para <max-active-sessions/>.</p>
<passivation-min-idle-time/>	<p>Anteriormente, las sesiones debían estar activas durante un tiempo mínimo antes de convertirse en candidatos para la pasivación. Esto podría provocar la falla en la creación de la sesión, incluso cuando la pasivación estaba habilitada.</p> <p>La nueva implementación no es compatible con esta lógica y, por lo tanto, evita esta vulnerabilidad de denegación de servicio (DoS).</p>
<passivation-max-idle-time/>	<p>Anteriormente, una sesión se pasivaba después de estar inactiva durante un período de tiempo específico.</p> <p>La nueva implementación solo admite la pasivación lenta. No es compatible con la pasivación ansiosa. Las sesiones solo son pasivadas cuando sea necesario para cumplir con <max-active-sessions/>.</p>
<replication-config/>	La nueva implementación desaprueba una serie de subelementos.

Cambios de configuración del servidor [Esto es una traducción automática]	Cambios de aplicaciones de seguridad [Esto es una traducción automática]
<replication-trigger/>	Anteriormente, este elemento se usaba para determinar cuándo se activaba la replicación de la sesión. La nueva implementación reemplaza esta opción de configuración con una estrategia única y sólida. Para más información, ver Atributos de sesión inmutables en el JBoss EAP <i>Guía de desarrollo</i> .
<use-jk/>	Anteriormente, el instance-id del nodo que maneja una solicitud dada se adjuntó a la jsessionId , para uso de balanceadores de carga como mod_jk, mod_proxy_balancer, mod_cluster, dependiendo del valor especificado para <use-jk/>. En la nueva implementación, el instance-id , si está definido, siempre se agrega al jsessionId .
<max-unreplicated-interval/>	Anteriormente, esta opción de configuración estaba pensada como una optimización para evitar la replicación de la marca de tiempo de una sesión si no se cambiaba ningún atributo de sesión. Aunque esto suena bien, en la práctica no impide ningún RPC, ya que el acceso a la sesión requiere RPC de transacciones de caché, independientemente de si se modificaron los atributos de sesión. En la nueva implementación, la marca de tiempo de una sesión se replica en cada solicitud. Esto evita los metadatos de sesión obsoleta después de una conmutación por error.
<snapshot-mode/>	Anteriormente, uno podía configurar <snapshot-mode/> como INSTANT o INTERVAL . La replicación asíncrona de Infinispan hace que esta opción de configuración sea obsoleta.
<snapshot-interval/>	Esto solo fue relevante para <snapshot-mode> INTERVAL </snapshot-mode>. Ya que <snapshot-mode/> es obsoleto, esta opción ahora también está obsoleta.
<session-notification-policy/>	Anteriormente, el valor especificado por este atributo definía una política para activar eventos de sesión. En la nueva implementación, este comportamiento depende de las especificaciones y no se puede configurar.

Esta nueva implementación también es compatible con las tiendas de caché de escritura simultánea, así como las tiendas de caché solo de pasivación. Normalmente, se usa un almacén de caché de escritura simultánea junto con un caché de invalidación. La implementación de clústeres de sesiones web en JBoss EAP 6 no funcionaba correctamente cuando se utilizaba con un caché de invalidación.

5.19.3. Cambios de agrupamiento de EJB de sesión de estado [Esto es una traducción automática]

En JBoss EAP 6, se le solicitó que habilitara el comportamiento de clúster para beans de sesión con estado (SFSB) de una de las siguientes maneras.

- Puedes agregar el **org.jboss.ejb3.annotation.Clustered** anotación en el bean de la sesión.

```
@Stateful
@Clustered
public class MyBean implements MySessionInt {

    public void myMethod() {
        //
    }
}
```

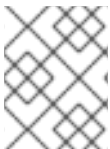
- Puedes agregar el **<clustered>** elemento a la **jboss-ejb3.xml** archivo.

```
<c:clustering>
  <ejb-name>DDBasedClusteredSFSB</ejb-name>
  <c:clustered>true</c:clustered>
</c:clustering>
```

JBoss EAP 7 ya no requiere que habilite el comportamiento de agrupamiento. De forma predeterminada, si el servidor se inicia utilizando un perfil HA, el estado de los SFSB se replicará automáticamente.

Puede deshabilitar este comportamiento predeterminado de una de las siguientes maneras.

- Puede deshabilitar el comportamiento predeterminado para un único bean de sesión con estado mediante el uso de **@Stateful(passivationCapable=false)**, que es nuevo en la especificación EJB 3.2.
- Puede desactivar este comportamiento globalmente en la configuración de **ejb3** subsistema en la configuración del servidor.



NOTA

Si el **@Clustered** la anotación no se elimina de la aplicación, simplemente se ignora y no afecta la implementación de la aplicación.

5.19.4. Clustering Services Changes [Esto es una traducción automática]

En JBoss EAP 6, las API para servicios de clustering estaban en módulos privados y no eran compatibles.

JBoss EAP 7 presenta una API de servicios públicos de clustering para uso de las aplicaciones. Los nuevos servicios están diseñados para ser livianos, fácilmente inyectables y no requieren dependencias externas.

- El nuevo **org.wildfly.clustering.group.Group** La interfaz proporciona acceso al estado actual del clúster y permite escuchar los cambios en la membresía del clúster.
- El nuevo **org.wildfly.clustering.dispatcher.CommandDispatcher** La interfaz permite ejecutar código en el clúster, en todos o en un subconjunto seleccionado de nodos.

Estos servicios reemplazan API similares que estaban disponibles en versiones anteriores, es decir, **HAPartition** de JBoss EAP 5 y **GroupCommunicationService**, **GroupMembershipNotifier**, y **GroupRpcDispatcher** en JBoss EAP 6.

Para más información, ver [API pública para servicios de clúster](#) en el JBoss EAP *Guía de desarrollo*.

5.19.5. Migrar agrupamiento HA Singleton [Esto es una traducción automática]

En JBoss EAP 6, no había una API pública disponible para el servicio de singleton HA de todo el clúster. Si usaste el privado `org.jboss.as.clustering.singleton.*` clases, debe cambiar su código para usar el nuevo público `org.wildfly.clustering.singleton.*` paquetes cuando migre su aplicación a JBoss EAP 7.

Para obtener más información acerca de los servicios de HA singleton, consulte [Servicio HA Singleton](#) en el *Guía de desarrollo* para JBoss EAP. Para obtener información acerca de las implementaciones de HA singleton, consulte [Despliegues HA Singleton](#) en el *Guía de desarrollo* para JBoss EAP.

CAPÍTULO 6. CAMBIOS DIVERSOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

6.1. CAMBIOS PARA LA ENTREGA DE JBOSS EAP NATIVES Y APACHE HTTP SERVER [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 7 nativos se entregan de forma diferente en este lanzamiento. Algunos se distribuyen con el nuevo producto Red Hat JBoss Core Services, el cual es un set de software complementario común a muchos productos Red Hat JBoss middleware. El nuevo producto permite una distribución de actualizaciones más rápida y consistente. El producto JBoss Core Services está disponible para ser descargado en un sitio diferente del Portal del cliente de Red Hat.

- La siguiente tabla presenta las diferencias en los métodos de entrega entre lanzamientos.

Paquete	JBoss EAP 6	JBoss EAP 7
Nativos AIO para Mensajería	Se distribuye con el producto en una descarga separada de "Native Utilities".	Incluidos dentro de la distribución de JBoss EAP. No se requiere descarga adicional.
Servidor Apache HTTP	Se distribuye con el producto en una descarga separada de "Apache HTTP Server"	Se distribuye con el nuevo producto JBoss Core Services
Conectores mod_cluster, mod_jk, isapi y nsapi	Se distribuye con el producto en una descarga separada de "Webserver Connector Natives"	Se distribuye con el nuevo producto JBoss Core Services
JSVC	Se distribuye con el producto en una descarga separada de "Native Utilities".	Se distribuye con el nuevo producto JBoss Core Services
OpenSSL	Se distribuye con el producto en una descarga separada de "Native Utilities".	Se distribuye con el nuevo producto JBoss Core Services
tcnatives	Se distribuye con el producto en una descarga separada de "Native Components"	Revise las novedades de JBoss EAP 7 [Esto es una traducción automática]

- Le informamos los siguientes cambios:
 - Se eliminó el soporte para los conectores mod_cluster y mod_jk utilizados con el servidor Apache HTTP de los canales Red Hat Enterprise Linux RPM. Si ejecuta Servidor Apache HTTP desde los canales RPM de Red Hat Enterprise Linux y necesita configurar el equilibrio de carga para los servidores JBoss EAP 7, puede hacer una de las siguientes cosas:
 - Use el Apache HTTP Server provisto por JBoss Core Services.

- Puede configurar JBoss EAP 7 para que actúe como un equilibrador de carga front-end. Para más información, ver [Configurando JBoss EAP como equilibrador de carga de front-end](#) en el JBoss EAP *Guía de configuración*.
- Puede implementar Apache HTTP Server en una máquina que sea compatible y esté certificada y luego ejecutar el equilibrador de carga en esa máquina. Para ver la lista de configuraciones compatibles, consulte [Descripción general de los conectores HTTP](#) en el JBoss EAP 7 *Guía de configuración*.
- Se ha dejado de ofrecer soporte para los conectores `mod_cluster` y `mod_jk` utilizados con Apache HTTP Server desde los HP-UX Web Server Suites. Si ejecuta Apache HTTP Server desde HP-UX Web Server Suites y necesita configurar el equilibrador de cargas para servidores JBoss EAP 7, puede hacer lo siguiente:
 - Puede configurar JBoss EAP 7 para que actúe como un equilibrador de carga front-end. Para más información, ver [Configurando JBoss EAP como equilibrador de carga de front-end](#) en el JBoss EAP *Guía de configuración*.
 - Puede implementar Apache HTTP Server en una máquina que sea compatible y esté certificada y luego ejecutar el equilibrador de carga en esa máquina. Para ver la lista de configuraciones compatibles, consulte [Descripción general de los conectores HTTP](#) en el JBoss EAP *Guía de configuración*.
- Puede encontrar más información sobre [Servicios básicos de JBoss](#) en el *Guía de instalación del servidor Apache HTTP*.

6.2. CAMBIOS EN IMPLEMENTACIONES EN AMAZON EC2 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Se han realizado una serie de cambios en Amazon Machine Images (AMI) en JBoss EAP 7. Esta sección resume brevemente algunos de esos cambios.

- La forma en que inicia las instancias y dominios de JBoss EAP agrupados y no agrupados en Amazon EC2 ha cambiado significativamente.
- JBoss EAP 6 utilizó el **User Data**: campo para la configuración de JBoss EAP. Los scripts AMI que analizaron la configuración en **User Data**: campo y comenzó los servidores automáticamente en el inicio de la instancia se han eliminado de JBoss EAP 7.
- El agente Red Hat JBoss Operations Network se instaló en la versión anterior de JBoss EAP. En JBoss EAP 7, debe instalarlo por separado.

Para obtener más información sobre la implementación de JBoss EAP 7 en Amazon EC2, consulte [Despliegue Red Hat JBoss Enterprise Application Platform en Amazon EC2](#).

6.3. DESINSTALAR APLICACIONES QUE INCLUYEN MÓDULOS COMPARTIDOS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Los cambios en el servidor JBoss EAP 7.1 y el complemento Maven pueden ocasionar el siguiente error cuando intente anular la implementación de su aplicación. Este error puede ocurrir si su aplicación contiene módulos que interactúan o dependen el uno del otro.

```
WFLYCTL0184:      New missing/unsatisfied dependencies
```

Por ejemplo, supongamos que tiene una aplicación que contiene dos módulos de proyecto Maven WAR, **application-A** y **application-B**, que comparten datos administrados por **data-sharing** módulo.

Cuando implementa esta aplicación, debe implementar la compartida **data-sharing** módulo primero, y luego implementar los módulos que dependen de él. La orden de implementación se especifica en `<modules>` elemento del padre **pom.xml** archivo. Esto es cierto en las versiones 6.4 a 7.1 de JBoss EAP.

En lanzamientos anteriores de JBoss EAP, puede anular la implementación de todos los archivos de esta aplicación desde la raíz del proyecto principal utilizando el siguiente comando.

```
$ mvn wildfly:undeploy
```

En JBoss EAP 7.1, primero debe anular la implementación de los archivos que utilizan los módulos compartidos y luego anular la implementación de los módulos compartidos. Dado que no hay forma de especificar el orden de des-despliegue en el proyecto **pom.xml** archivo, debe anular la implementación de los módulos de forma manual. Puede lograr esto ejecutando los siguientes comandos desde la raíz del directorio principal.

```
$ mvn wildfly:undeploy -pl application-A,application-B
$ mvn wildfly:undeploy -pl data-shared
```

Este nuevo comportamiento de anulación de implementación es más correcto y garantiza que no termine en un estado de implementación inestable.

6.4. CAMBIOS EN LAS SECUENCIAS DE COMANDOS DE JBOSS EAP [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

los **add-user** El comportamiento del script ha cambiado en JBoss EAP 7 debido a un cambio en la política de contraseñas. JBoss EAP 6 tenía una política de contraseña estricta. Como resultado, el **add-user** el script rechazó las contraseñas débiles que no cumplían con los requisitos mínimos. En JBoss EAP 7, se aceptan contraseñas débiles y se emite una advertencia. Para más información, ver [Configuración de restricciones de contraseña de la utilidad para usuarios adicionales](#) en el JBoss EAP *Guía de configuración*.

6.5. ELIMINACIÓN DEL SOPORTE OSGI [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Cuando se lanzó por primera vez JBoss EAP 6.0 GA, se incluyó JBoss OSGi, una implementación de la especificación OSGi, como característica de Vista previa de tecnología. Con el lanzamiento de JBoss EAP 6.1.0, JBoss OSGi fue degradado de Technology Preview a Unsupported.

En JBoss EAP 6.1.0, el **configadmin** y **osgi** los módulos de extensión y la configuración del subsistema para un servidor independiente se movieron a un **EAP_HOME/standalone/configuration/standalone-osgi.xml** archivo de configuración. Debido a que no debe migrar este archivo de configuración no compatible, la eliminación de la compatibilidad con JBoss OSGi no debe afectar la migración de una configuración de servidor independiente. Si modificó alguno de los otros archivos de configuración independientes para configurar **osgi** o **configadmin**, esas configuraciones deben ser eliminadas.

Para un dominio administrado, el **osgi** la extensión y la configuración del subsistema se eliminaron de la **EAP_HOME/domain/configuration/domain.xml** archivo en la versión JBoss EAP 6.1.0. sin embargo, el **configadmin** la extensión del módulo y la configuración del subsistema permanecen en

el ***EAP_HOME/domain/configuration/domain.xml*** archivo. Esta configuración ya no es compatible con JBoss EAP 7 y debe eliminarse.

CAPÍTULO 7. MIGRACIÓN A ELYTRON EN JBOSS EAP 7.1 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

7.1. DESCRIPCIÓN GENERAL DE ELYTRON [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

JBoss EAP 7.1 presenta Elytron, que proporciona un único marco unificado que puede administrar y configurar el acceso tanto para servidores independientes como para dominios administrados. También se puede usar para configurar el acceso de seguridad para aplicaciones implementadas en servidores JBoss EAP.

IMPORTANTE

Las arquitecturas de Elytron y el subsistema de seguridad heredado que se basa en PicketBox son muy diferentes. Con Elytron, se intentó crear una solución que le permita operar en los mismos entornos de seguridad en los que opera actualmente; sin embargo, esto no *no* significa que cada opción de configuración de PicketBox tiene una opción de configuración equivalente en Elytron.

Si no puede encontrar información en la documentación que lo ayude a lograr una funcionalidad similar utilizando Elytron que tenía al usar la implementación de seguridad heredada, puede encontrar ayuda de una de las siguientes maneras.

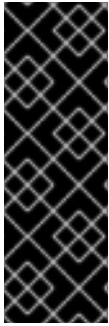
- Si tienes un [Suscripción a Red Hat Development](#), tienes acceso a [Casos de soporte](#), [Soluciones](#), y [Artículos de conocimiento](#) en el portal de clientes de Red Hat. También puedes abrir un caso con [Soporte técnico](#) y [consigue ayuda](#) de la comunidad WildFly como se describe a continuación.
- Si no tienes una suscripción a Red Hat Development, aún puedes acceder [Artículos de conocimiento](#) en el portal de clientes de Red Hat. También puedes unirte al [foros de usuarios y chat en vivo](#) para hacer preguntas a la comunidad de WildFly. La oferta de la comunidad WildFly es monitoreada activamente por el equipo de ingeniería de Elytron.

Su configuración de servidor JBoss EAP 7.0 y las implementaciones que utilizan el legado **security** el subsistema, que está basado en PicketBox, debería ejecutarse sin cambios en JBoss EAP 7.1.

PicketBox continúa admitiendo dominios de seguridad, lo que permite que las aplicaciones continúen utilizando los módulos de inicio de sesión existentes. Los dominios de seguridad, que son utilizados por la capa de administración para la seguridad, también son transferidos e imitados por Elytron. Esto le permite definir la autenticación tanto en el **elytron** y legado **security** subsistemas y usarlos en paralelo. Para obtener más información acerca de cómo configurar su aplicación para usar Elytron y la seguridad heredada, consulte [Configurar aplicaciones web para usar Elytron o Legacy Security para la autenticación](#) en *Cómo configurar la gestión de identidad* para JBoss EAP.

Aunque la autenticación PicketBox sigue siendo compatible, se recomienda que cambie a Elytron cuando esté listo para migrar sus aplicaciones. Una de las ventajas de usar la seguridad de Elytron es que proporciona una solución de seguridad consistente en todo el servidor y sus aplicaciones. Para obtener información sobre cómo migrar la autenticación y autorización de PicketBox para usar Elytron, consulte [Migrar la configuración de autenticación](#) en esta guía.

Para obtener una descripción general de los nuevos recursos que están disponibles en **elytron** subsistema, ver [Recursos en el subsistema Elytron](#) en el JBoss EAP *Arquitectura de seguridad* guía.



IMPORTANTE

Tenga en cuenta que si elige usar tanto el legado **security** subsistema y Elytron en sus implementaciones, las invocaciones entre implementaciones que usan arquitecturas de seguridad diferentes no son compatibles.

Para obtener más información sobre el uso de estos subsistemas en paralelo, consulte [Usando Elytron y los subsistemas de seguridad heredados en paralelo](#) en *Cómo configurar la gestión de identidad* para JBoss EAP.

7.2. MIGRAR BÓVEDAS Y PROPIEDADES SEGURAS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

7.2.1. Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]

La bóveda que se usó para almacenar el cifrado de cadenas de texto sin formato en el legado **security** El subsistema en JBoss EAP 7.0 no es compatible con Elytron en JBoss EAP 7.1, que utiliza un almacén de credenciales recientemente diseñado para almacenar cadenas. Las tiendas de credenciales cifran las credenciales de forma segura en un archivo de almacenamiento externo a los archivos de configuración de JBoss EAP. Puede usar la implementación provista por Elytron o puede personalizar la configuración usando las API y SPI del store de credenciales. Cada servidor JBoss EAP puede contener múltiples tiendas de credenciales.



NOTA

Si utilizó previamente expresiones de bóveda para parametrizar datos no sensibles, se recomienda que reemplace los datos con [Propiedades de seguridad Elytron](#).

Si continúa usando el legado **security** subsistema, no debería necesitar modificar o actualizar sus datos de bóveda. Sin embargo, si planea migrar su aplicación para usar Elytron, debe convertir sus bóvedas existentes en tiendas de credenciales para que puedan ser procesadas por el **elytron** subsistema. Para obtener más información acerca de las tiendas de credenciales, consulte [Tiendas de credenciales](#) en *Cómo configurar la seguridad del servidor* para JBoss EAP.

Migre los clientes para usar el archivo de configuración de WildFly [Esto es una traducción automática]

La herramienta Wildly Elytron que se envía con JBoss EAP proporciona una **vault** comando para ayudarlo a migrar el contenido de la bóveda a las tiendas de credenciales. Ejecuta la herramienta ejecutando el **elytron-tool** script, que se encuentra en **EAP_HOME/bin** directorio.

```
$ EAP_HOME/bin/elytron-tool.sh vault VAULT_ARGUMENTS
```

Si lo prefiere, puede ejecutar la herramienta ejecutando el **java -jar** mando.

```
$ java -jar EAP_HOME/bin/wildfly-elytron-tool.jar vault VAULT_ARGUMENTS
```

Puede usar el siguiente comando para obtener una descripción de todos los argumentos disponibles.

```
$ EAP_HOME/bin/elytron-tool.sh vault --help
```



NOTA

- La herramienta Wildly Elytron no puede manejar la primera versión de los archivos de datos de la bóveda de seguridad.
- Puede ingresar el **--keystore-password** argumento en formato enmascarado, como se muestra en el siguiente ejemplo para migrar una sola bóveda, o en texto claro.
- los **--salt** y **--iteration** se proporcionan argumentos para proporcionar información para descifrar la contraseña enmascarada o para generar una contraseña enmascarada en la salida. Si el **--salt** y **--iteration** los argumentos se omiten, se usan valores predeterminados.
- los **--summary** El argumento produce comandos CLI de administración formateados que se pueden usar para agregar las tiendas de credenciales convertidas a la configuración de JBoss EAP. Las contraseñas de texto plano están enmascaradas en el resultado del resumen.



IMPORTANTE

Tenga en cuenta que las tiendas de credenciales solo se pueden usar para proteger contraseñas. No son compatibles con la característica de expresión de la bóveda que podría utilizarse en cualquier lugar del modelo de gestión.

Elija una de las siguientes opciones de migración:

- [Migre una bóveda de seguridad individual a una tienda de credenciales](#)
- [Migre varias bóvedas de seguridad a una tienda de credenciales a granel](#)

Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]

El siguiente es un ejemplo del comando utilizado para convertir una única bóveda de seguridad a un almacén de credenciales.

```
$ EAP_HOME/bin/elytron-tool.sh vault --enc-dir vault_data/ --keystore
vault-jceks.keystore --keystore-password MASK-2hKo56F1a3jYGnJwhPmiF5 --
iteration 34 --salt 12345678 --alias test --location cs-v1.store --summary
```

Este comando convierte la bóveda de seguridad en un almacén de credenciales e imprime el resumen de los comandos CLI de administración que se usaron para convertirlo en la salida.

```
Vault (enc-dir="vault_data/";keystore="vault-jceks.keystore") converted to
credential store "cs-v1.store"
Vault Conversion summary:
-----
Vault Conversion Successful
CLI command to add new credential store:
/subsystem=elytron/credential-store=test:add(relative-
to=jboss.server.data.dir,create=true,modifiable=true,location="cs-
v1.store",implementation-properties={"keyStoreType"=>"JCEKS"},credential-
reference={clear-text="MASK-2hKo56F1a3jYGnJwhPmiF5;12345678;34"})
```

Migre las bóvedas para proteger el almacenamiento de credenciales [Esto es una traducción automática]

Puede convertir varias bóvedas en una tienda de credenciales utilizando **--bulk-convert** argumento y apuntando a un archivo descriptor de conversión masiva.

Los ejemplos en esta sección usan el siguiente archivo descriptor de conversión masiva.

Ejemplo: bulk-vault-conversion-descriptor.txt Archivo [Esto es una traducción automática]

```
keystore:vault-v1/vault-jceks.keystore
keystore-password:MASK-2hKo56F1a3jYGnJwhPmiF5
enc-dir:vault-v1/vault_data/
salt:12345678
iteration:34
location:v1-cs-1.store
alias:test

keystore:vault-v1/vault-jceks.keystore
keystore-password:secretsecret
enc-dir:vault-v1/vault_data/
location:v1-cs-2.store
alias:test

# different vault vault-v1-more
keystore:vault-v1-more/vault-jceks.keystore
keystore-password:MASK-2hKo56F1a3jYGnJwhPmiF5
enc-dir:vault-v1-more/vault_data/
salt:12345678
iteration:34
location:v1-cs-more.store
alias:test
```

Una nueva conversión comienza cuando cada nuevo **keystore:** línea se encuentra. Todas las opciones son obligatorias excepto por **salt**, **iteration**, y **properties**.

Para realizar la conversión masiva y generar resultados que formatean los comandos CLI de administración, ejecute el siguiente comando.

```
$ EAP_HOME/bin/elytron-tool.sh vault --bulk-convert path/to/bulk-vault-
conversion-descriptor.txt --summary
```

Este comando convierte todas las bóvedas de seguridad especificadas en el archivo en un almacén de credenciales e imprime el resumen de los comandos CLI de administración que se usaron para convertirlos en el resultado.

```
Vault (enc-dir="vault-v1/vault_data/";keystore="vault-v1/vault-
jceks.keystore") converted to credential store "v1-cs-1.store"
Vault Conversion summary:
-----
Vault Conversion Successful
CLI command to add new credential store:
/subsystem=elytron/credential-store=test:add(relative-
to=jboss.server.data.dir,create=true,modifiable=true,location="v1-cs-
1.store",implementation-properties={"keyStoreType"=>"JCEKS"},credential-
```



```
reference={clear-text="MASK-2hKo56F1a3jYGnJwhPmiF5;12345678;34"})
-----

Vault (enc-dir="vault-v1/vault_data/";keystore="vault-v1/vault-
jceks.keystore") converted to credential store "v1-cs-2.store"
Vault Conversion summary:
-----
Vault Conversion Successful
CLI command to add new credential store:
/subsystem=elytron/credential-store=test:add(relative-
to=jboss.server.data.dir,create=true,modifiable=true,location="v1-cs-
2.store",implementation-properties={"keyStoreType"=>"JCEKS"},credential-
reference={clear-text="secretsecret"})
-----

Vault (enc-dir="vault-v1-more/vault_data/";keystore="vault-v1-more/vault-
jceks.keystore") converted to credential store "v1-cs-more.store"
Vault Conversion summary:
-----
Vault Conversion Successful
CLI command to add new credential store:
/subsystem=elytron/credential-store=test:add(relative-
to=jboss.server.data.dir,create=true,modifiable=true,location="v1-cs-
more.store",implementation-properties=
{"keyStoreType"=>"JCEKS"},credential-reference={clear-text="MASK-
2hKo56F1a3jYGnJwhPmiF5;12345678;34"})
-----
```

7.2.2. Migre las propiedades de seguridad a Elytron [Esto es una traducción automática]

Los ejemplos en esta sección suponen que **group.name** y **encoding.algorithm** las propiedades de seguridad se definen como **security-properties** en el legado **security** subsistema de la siguiente manera.

Ejemplo: Propiedades de seguridad definidas en security Subsistema [Esto es una traducción automática]

```
<subsystem xmlns="urn:jboss:domain:security:2.0">
  ...
  <security-properties>
    <property name="group.name" value="engineering-group" />
    <property name="encoding.algorithm" value="BASE64" />
  </security-properties>
</subsystem>
```

Para definir las mismas propiedades de seguridad en **elytron** subsistema, configure el **security-properties** atributo de la **elytron** subsistema utilizando el siguiente comando CLI de gestión.

```
/subsystem=elytron:write-attribute(name=security-properties, value={
group.name = "engineering-group", encoding.algorithm = "BASE64" })
```

Esto configura lo siguiente **security-properties** en el **elytron** subsistema en el archivo de configuración del servidor.


```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
  <security-properties>
    <security-property name="group.name" value="engineering-group"/>
    <security-property name="encoding.algorithm" value="BASE64"/>
  </security-properties>
  ...
</subsystem>
```

los **write-attribute** la operación utilizada en el comando anterior sobrescribe las propiedades existentes. Para agregar o cambiar una propiedad de seguridad sin afectar otras propiedades de seguridad, use la **map** operación en el comando CLI de administración.

```
/subsystem=elytron:map-put(name=security-properties, key=group.name,
value=technical-support)
```

De manera similar, puede eliminar una propiedad de seguridad específica utilizando el **map-remove** operación.

```
/subsystem=elytron:map-remove(name=security-properties, key=group.name)
```

7.3. MIGRAR LA CONFIGURACIÓN DE AUTENTICACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

7.3.1. Migre la autenticación y autorización basada en las propiedades a Elytron [Esto es una traducción automática]

7.3.1.1. Migre la configuración basada en propiedades de PicketBox a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar la autenticación basada en propiedades de PicketBox a Elytron. Puedes elegir [migrar parcialmente](#) autenticación basada en propiedades exponiendo únicamente el dominio de seguridad PicketBox a Elytron o puede [migrar completamente](#) las configuraciones de autenticación basadas en propiedades para usar Elytron.

Los siguientes procedimientos suponen que la aplicación web implementada que planea migrar está configurada para requerir autenticación basada en formularios. La aplicación hace referencia a un dominio de seguridad PicketBox y está utilizando el **UsersRolesLoginModule** para cargar información del usuario desde **example-users.properties** y **example-roles.properties** archivos. Estos ejemplos también suponen que el dominio de seguridad se define en el legado **security** subsistema utilizando los siguientes comandos CLI de gestión.

Ejemplo: Comandos de configuración basados en las propiedades de PicketBox [Esto es una traducción automática]

```
/subsystem=security/security-domain=application-security:add
/subsystem=security/security-domain=application-
security/authentication=classic:add(login-modules=[{code=UsersRoles,
flag=Required, module-options=
{usersProperties=file://$${jboss.server.config.dir}/example-
```

```
users.properties,
rolesProperties=file://${jboss.server.config.dir}/example-
roles.properties}}])
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: configuración de dominio de seguridad basada en propiedades de PicketBox [Esto es una traducción automática]

```
<security-domain name="application-security">
  <authentication>
    <login-module code="UsersRoles" flag="required">
      <module-option name="usersProperties"
value="file://${jboss.server.config.dir}/example-users.properties"/>
      <module-option name="rolesProperties"
value="file://${jboss.server.config.dir}/example-roles.properties"/>
    </login-module>
  </authentication>
</security-domain>
```

Elija una de las siguientes opciones de migración:

- [Parcialmente migrar al exponer el dominio de seguridad de PicketBox a Elytron.](#)
- [Migre completamente la autenticación basada en propiedades a Elytron](#)

Parcialmente migrar al exponer el dominio de seguridad de PicketBox a Elytron

Puede exponer un dominio de seguridad PicketBox como un dominio de seguridad Elytron para que pueda conectarse a una configuración Elytron; sin embargo, hacerlo crea una dependencia en el legado **security** subsistema. Si solo está migrando la autenticación basada en propiedades, se recomienda que [migrar completamente la aplicación a Elytron](#) para evitar la dependencia innecesaria del legado **security** subsistema. Sin embargo, una migración parcial puede ser una solución intermedia cuando no es posible migrar completamente la aplicación para usar Elytron.

Siga este procedimiento para agregar una configuración de dominio de seguridad PicketBox existente como un dominio de seguridad Elytron.

1. Ejemplo: Propiedades de seguridad definidas en **security** Subsistema [Esto es una traducción automática]

```
/subsystem=security/elytron-realm=application-security:add(legacy-
jaas-config=application-security)
```

Esto configura el siguiente reino de seguridad Elytron en el **security** subsistema del archivo de configuración del servidor.

```
<subsystem xmlns="urn:jboss:domain:security:2.0">
  ...
  <elytron-integration>
    <security-realms>
      <elytron-realm name="application-security" legacy-jaas-
config="application-security"/>
    </security-realms>
  </elytron-integration>
</subsystem>
```

```

    </elytron-integration>
    ...
</subsystem>

```

2. Definir un dominio de seguridad en el **elytron** subsistema que hace referencia al reino de seguridad exportado y una fábrica de autenticación HTTP que admite la autenticación basada en formularios.

```

/subsystem=elytron/security-domain=application-security:add(realms=
[{{realm=application-security}}, default-realm=application-security,
permission-mapper=default-permission-mapper])
/subsystem=elytron/http-authentication-factory=application-security-
http:add(http-server-mechanism-factory=global, security-
domain=application-security, mechanism-configurations=[{{mechanism-
name=DIGEST,mechanism-realm-configurations=[{{realm-
name=RealmName}}}}])

```

Esto resulta en lo siguiente **elytron** configuración del subsistema en el archivo de configuración del servidor.

```

<subsystem xmlns="urn:wildfly:elytron:1.2" final-
providers="combined-providers" disallowed-providers="OracleUcrypto">
    ...
    <security-domains>
        ...
        <security-domain name="application-security" default-
realm="application-security" permission-mapper="default-permission-
mapper">
            <realm name="application-security"/>
        </security-domain>
    </security-domains>
    ...
    <http>
        ...
        <http-authentication-factory name="application-security-http"
http-server-mechanism-factory="global" security-domain="application-
security">
            <mechanism-configuration>
                <mechanism mechanism-name="DIGEST">
                    <mechanism-realm realm-name="RealmName"/>
                </mechanism>
            </mechanism-configuration>
        </http-authentication-factory>
        ...
    </http>
    ...
</subsystem>

```

3. Asigne el dominio de seguridad al que hace referencia la implementación a la fábrica de autenticación HTTP recientemente definida en **undertow** subsistema.

```

/subsystem=undertow/application-security-domain=application-
security:add(http-authentication-factory=application-security-http)

```

Esto da como resultado la siguiente configuración en **undertow** subsistema del archivo de configuración del servidor.

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  ...
  <application-security-domains>
    <application-security-domain name="application-security" http-
authentication-factory="application-security-http"/>
  </application-security-domains>
  ...
</subsystem>
```

4. Si la aplicación ya se implementó antes de esta configuración, debe volver a cargar el servidor o volver a implementar la aplicación para que la nueva asignación de dominio de seguridad de la aplicación surta efecto.
5. Verifique que la asignación se aplicó a la implementación utilizando el siguiente comando CLI de administración. La implementación utilizada en este ejemplo es **HelloWorld.war**. El resultado de este comando muestra que esta implementación hace referencia a la asignación de Elytron.

```
/subsystem=undertow/application-security-domain=application-
security:read-resource(include-runtime=true)
{
  "outcome" => "success",
  "result" => {
    "enable-jacc" => false,
    "http-authentication-factory" => "application-security-
http",
    "override-deployment-config" => false,
    "referencing-deployments" => ["HelloWorld.war"],
    "setting" => undefined
  }
}
```

En esta etapa, el dominio de seguridad previamente definido se utiliza para su **LoginModule** configuración, pero está envuelto por los componentes de Elytron, que se hacen cargo de la autenticación.

Migre la autenticación y autorización basada en las propiedades a Elytron [Esto es una traducción automática]

Siga estos pasos para migrar completamente la autenticación basada en propiedades de PicketBox a Elytron. Este procedimiento supone que está comenzando con la configuración heredada descrita en la introducción a esta sección y tiene *no* migrado a la [parcialmente migrado](#) solución. Cuando haya completado este proceso, cualquier definición de dominio de seguridad que exista en el legado **security** subsistema permanece completamente independiente de la configuración de Elytron.

1. Definir un nuevo reino de seguridad en **elytron** subsistema que hace referencia a los archivos de propiedades de PicketBox.

```
/subsystem=elytron/properties-realm=application-
properties:add(users-properties={path=example-users.properties,
relative-to=jboss.server.config.dir, plain-text=true, digest-realm-
```

```
name="Application Security"}, groups-properties={path=example-roles.properties, relative-to=jboss.server.config.dir}, groups-attribute=Roles)
```

2. Definir un subsistema de dominio de seguridad y una fábrica de autenticación HTTP en **elytron** subsistema.

```
/subsystem=elytron/security-domain=application-security:add(realms=[{realm=application-properties}], default-realm=application-properties, permission-mapper=default-permission-mapper)
/subsystem=elytron/http-authentication-factory=application-security-http:add(http-server-mechanism-factory=global, security-domain=application-security, mechanism-configurations=[{mechanism-name=FORM}])
```

Esto resulta en lo siguiente **elytron** configuración del subsistema en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-
providers="combined-providers" disallowed-providers="OracleUcrypto">
  ...
  <security-domains>
    ...
    <security-domain name="application-security" default-
realm="application-properties" permission-mapper="default-
permission-mapper">
      <realm name="application-properties"/>
    </security-domain>
  </security-domains>
  <security-realms>
    ...
    <properties-realm name="application-properties" groups-
attribute="Roles">
      <users-properties path="example-users.properties" relative-
to="jboss.server.config.dir" digest-realm-name="Application
Security" plain-text="true"/>
      <groups-properties path="example-roles.properties" relative-
to="jboss.server.config.dir"/>
    </properties-realm>
  </security-realms>
  ...
  <http>
    ...
    <http-authentication-factory name="application-security-http"
http-server-mechanism-factory="global" security-domain="application-
security">
      <mechanism-configuration>
        <mechanism mechanism-name="FORM"/>
      </mechanism-configuration>
    </http-authentication-factory>
    ...
  </http>
  ...
</subsystem>
```

3. Asigne el dominio de seguridad de la aplicación al que hace referencia la implementación a la fábrica de autenticación HTTP recientemente definida en **undertow** subsistema.

```
/subsystem=undertow/application-security-domain=application-  
security:add(http-authentication-factory=application-security-http)
```

Esto resulta en lo siguiente **undertow** configuración del subsistema en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">  
  ...  
  <application-security-domains>  
    <application-security-domain name="application-security" http-  
authentication-factory="application-security-http"/>  
  </application-security-domains>  
  ...  
</subsystem>
```

4. Debe volver a cargar el servidor o volver a implementar la aplicación para que la nueva asignación de dominio de seguridad de la aplicación surta efecto.

La autenticación ahora está configurada para ser equivalente a la configuración de PicketBox; sin embargo, los componentes de Elytron ahora se usan exclusivamente para la autenticación.

7.3.1.2. Migre la configuración heredada basada en propiedades a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar un reino de seguridad heredado que carga información de usuarios, contraseñas y grupos desde los archivos de propiedades a Elytron. Este tipo de reino de seguridad heredado se usa generalmente para proteger las interfaces de administración o los conectores remotos.

Estos ejemplos suponen que el dominio de seguridad heredado se define mediante los siguientes comandos CLI de gestión.

Ejemplo: Comandos Legacy Security Realm [Esto es una traducción automática]

```
/core-service=management/security-realm=ApplicationSecurity:add  
/core-service=management/security-  
realm=ApplicationSecurity/authentication=properties:add(relative-  
to=jboss.server.config.dir, path=example-users.properties, plain-  
text=true)  
/core-service=management/security-  
realm=ApplicationSecurity/authorization=properties:add(relative-  
to=jboss.server.config.dir, path=example-roles.properties)
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: Comandos de configuración de dominios de seguridad heredados [Esto es una traducción automática]

```
<security-realm name="ApplicationSecurity">  
  <authentication>  
    <properties path="example-users.properties" relative-  
to="jboss.server.config.dir" plain-text="true"/>  
  </authentication>  
</security-realm>
```

```

</authentication>
<authorization>
  <properties path="example-roles.properties" relative-
to="jboss.server.config.dir"/>
</authorization>
</security-realm>

```

Una de las motivaciones para agregar la seguridad de Elytron al servidor de aplicaciones es permitir que se use una solución de seguridad consistente en todo el servidor. Los pasos iniciales para migrar un dominio de seguridad heredado basado en propiedades a Elytron son similares a los utilizados para migrar una autenticación basada en propiedades de PicketBox a Elytron. Siga estos pasos para migrar un dominio de seguridad heredado basado en propiedades a Elytron.

1. Definir un nuevo reino de seguridad en **elytron** subsistema que hace referencia a los archivos de propiedades.

```

/subsystem=elytron/properties-realm=application-
properties:add(users-properties={path=example-users.properties,
relative-to=jboss.server.config.dir, plain-text=true, digest-realm-
name="Application Security"}, groups-properties={path=example-
roles.properties, relative-to=jboss.server.config.dir}, groups-
attribute=Roles)

```

2. Definir un subsistema de dominio de seguridad y una fábrica de autenticación HTTP en **elytron** subsistema.

```

/subsystem=elytron/security-domain=application-security:add(realms=
[{{realm=application-properties}}, default-realm=application-
properties, permission-mapper=default-permission-mapper)
/subsystem=elytron/http-authentication-factory=application-security-
http:add(http-server-mechanism-factory=global, security-
domain=application-security, mechanism-configurations=[{{mechanism-
name=FORM}}])

```

Esto da como resultado la siguiente configuración de Elytron.

```

<subsystem xmlns="urn:wildfly:elytron:1.2" final-
providers="combined-providers" disallowed-providers="OracleUcrypto">
  ...
  <security-domains>
    ...
    <security-domain name="application-security" default-
realm="application-properties" permission-mapper="default-
permission-mapper">
      <realm name="application-properties"/>
    </security-domain>
  </security-domains>
  <security-realms>
    ...
    <properties-realm name="application-properties" groups-
attribute="Roles">
      <users-properties path="example-users.properties" relative-
to="jboss.server.config.dir" digest-realm-name="Application
Security" plain-text="true"/>
      <groups-properties path="example-roles.properties" relative-

```



```

to="jboss.server.config.dir"/>
  </properties-realm>
</security-realms>
...
<http>
  ...
  <http-authentication-factory name="application-security-http"
http-server-mechanism-factory="global" security-domain="application-
security">
    <mechanism-configuration>
      <mechanism mechanism-name="FORM"/>
    </mechanism-configuration>
  </http-authentication-factory>
  ...
</http>
...
</subsystem>

```

3. Definir un **sasl-authentication-factory** para que el reino de seguridad heredado también se pueda usar para la autenticación de capa de seguridad de autenticación simple (SASL).

```

/subsystem=elytron/sasl-authentication-factory=application-security-
sasl:add(sasl-server-factory=elytron, security-domain=application-
security, mechanism-configurations=[{mechanism-name=PLAIN}])

```

Esto da como resultado la siguiente configuración de Elytron.

```

<subsystem xmlns="urn:wildfly:elytron:1.2" final-
providers="combined-providers" disallowed-providers="OracleUcrypto">
  ...
  <sasl>
    ...
    <sasl-authentication-factory name="application-security-sasl"
sasl-server-factory="elytron" security-domain="application-
security">
      <mechanism-configuration>
        <mechanism mechanism-name="PLAIN"/>
      </mechanism-configuration>
    </sasl-authentication-factory>
    ...
  </sasl>
</subsystem>

```

4. Configure un conector remoto para la autenticación SASL y elimine la asociación con el reino de seguridad heredado.

```

/subsystem=remoting/http-connector=http-remoting-connector:write-
attribute(name=sasl-authentication-factory, value=application-
security-sasl)
/subsystem=remoting/http-connector=http-remoting-connector:undefine-
attribute(name=security-realm)

```

Esto da como resultado la siguiente configuración en **remoting** subsistema del archivo de configuración del servidor.

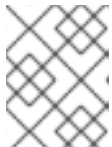

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0">
    ...
    <http-connector name="http-remoting-connector" connector-
ref="default" sasl-authentication-factory="application-security-
sasl"/>
</subsystem>
```

5. Agregue las dos fábricas de autenticación y elimine las referencias heredadas del reino de seguridad para asegurar el **http-interface** con Elytron.

```
/core-service=management/management-interface=http-interface:write-
attribute(name=http-authentication-factory, value=application-
security-http)
/core-service=management/management-interface=http-interface:write-
attribute(name=http-upgrade.sasl-authentication-factory,
value=application-security-sasl)
/core-service=management/management-interface=http-
interface:undefine-attribute(name=security-realm)
```

Esto da como resultado la siguiente configuración.

```
<management-interfaces>
    <http-interface http-authentication-factory="application-security-
http">
        <http-upgrade enabled="true" sasl-authentication-
factory="application-security-sasl"/>
        <socket-binding http="management-http"/>
    </http-interface>
</management-interfaces>
```



NOTA

Debería elegir nombres más adecuados que los utilizados en estos ejemplos para proteger las interfaces de gestión.

Migre la configuración heredada basada en propiedades a Elytron [Esto es una traducción automática]

7.3.2. Migre la configuración de autenticación LDAP a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar la autenticación LDAP heredada a Elytron para que pueda administrar la información como atributos de identidad. Gran parte de la información proporcionada en la sección titulada [Migre la autenticación y autorización basada en las propiedades a Elytron](#) se aplica aquí, particularmente con respecto a cómo definir dominios de seguridad y fábricas de autenticación, y cómo mapearlas para ser utilizadas para la autenticación. Esta sección no repite esas instrucciones, así que asegúrese de leer esa sección antes de continuar.

Los siguientes ejemplos suponen que la información de grupo o rol se carga directamente desde LDAP y que la autenticación LDAP heredada se configura de la siguiente manera.

- El servidor LDAP contiene las siguientes entradas de usuario y grupo.

Ejemplo: entradas de usuario del servidor LDAP [Esto es una traducción automática]

```
dn: uid=TestUserOne,ou=users,dc=group-to-principal,dc=wildfly,dc=org
objectClass: top
objectClass: inetOrgPerson
objectClass: uidObject
objectClass: person
objectClass: organizationalPerson
cn: Test User One
sn: Test User One
uid: TestUserOne
userPassword: {SSHA}UG8ov2rnrnBKakcARVvraZHqTa7mFWJZlWt2HA==
```

Ejemplo: entradas de grupos de servidores LDAP [Esto es una traducción automática]

```
dn: uid=GroupOne,ou=groups,dc=group-to-principal,dc=wildfly,dc=org
objectClass: top
objectClass: groupOfUniqueNames
objectClass: uidObject
cn: Group One
uid: GroupOne
uniqueMember: uid=TestUserOne,ou=users,dc=group-to-
principal,dc=wildfly,dc=org
```

Para fines de autenticación, el nombre de usuario se compara con el **uid** atributo y el nombre del grupo resultante se toma de la **uid** atributo de la entrada del grupo.

- La conexión con el servidor LDAP y el reino de seguridad relacionado se define mediante los siguientes comandos CLI de gestión.

Ejemplo: Comandos de configuración de LDAP Security Realm [Esto es una traducción automática]

```
batch
/core-service=management/ldap-
connection=MyLdapConnection:add(url="ldap://localhost:10389",
search-dn="uid=admin,ou=system", search-credential="secret")

/core-service=management/security-realm=LDAPRealm:add
/core-service=management/security-
realm=LDAPRealm/authentication=ldap:add(connection="MyLdapConnection
", username-attribute=uid, base-dn="ou=users,dc=group-to-
principal,dc=wildfly,dc=org")

/core-service=management/security-
realm=LDAPRealm/authorization=ldap:add(connection=MyLdapConnection)
/core-service=management/security-
realm=LDAPRealm/authorization=ldap/username-to-dn=username-
filter:add(attribute=uid, base-dn="ou=users,dc=group-to-
principal,dc=wildfly,dc=org")
/core-service=management/security-
realm=LDAPRealm/authorization=ldap/group-search=group-to-
```

```
principal:add(base-dn="ou=groups,dc=group-to-
principal,dc=wildfly,dc=org", iterative=true, prefer-original-
connection=true, principal-attribute=uniqueMember, search-
by=DISTINGUISHED_NAME, group-name=SIMPLE, group-name-attribute=uid)
run-batch
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: Configuración de LDAP Security Realm [Esto es una traducción automática]

```
<management>
  <security-realms>
    ...
    <security-realm name="LDAPRealm">
      <authentication>
        <ldap connection="MyLdapConnection" base-
dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org">
          <username-filter attribute="uid"/>
        </ldap>
      </authentication>
      <authorization>
        <ldap connection="MyLdapConnection">
          <username-to-dn>
            <username-filter base-dn="ou=users,dc=group-to-
principal,dc=wildfly,dc=org" attribute="uid"/>
          </username-to-dn>
          <group-search group-name="SIMPLE" iterative="true" group-
name-attribute="uid">
            <group-to-principal search-by="DISTINGUISHED_NAME" base-
dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org" prefer-
original-connection="true">
              <membership-filter principal-
attribute="uniqueMember"/>
            </group-to-principal>
          </group-search>
        </ldap>
      </authorization>
    </security-realm>
  </security-realms>
  <outbound-connections>
    <ldap name="MyLdapConnection" url="ldap://localhost:10389"
search-dn="uid=admin,ou=system" search-credential="secret"/>
  </outbound-connections>
  ...
</management>
```

- Los siguientes comandos CLI de gestión se utilizan para configurar un dominio de seguridad PicketBox, que utiliza el **LdapExtLoginModule** para verificar un nombre de usuario y contraseña.

Ejemplo: Comandos de configuración de dominio de seguridad [Esto es una traducción automática]

```
/subsystem=security/security-domain=application-security:add
```

```
/subsystem=security/security-domain=application-
security/authentication=classic:add(login-modules=
[{code=LdapExtended, flag=Required, module-options={
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory,
java.naming.provider.url=ldap://localhost:10389,
java.naming.security.authentication=simple,
bindDN="uid=admin,ou=system", bindCredential=secret,
baseCtxDN="ou=users,dc=group-to-principal,dc=wildfly,dc=org",
baseFilter="(uid={0})", rolesCtxDN="ou=groups,dc=group-to-
principal,dc=wildfly,dc=org", roleFilter="(uniqueMember={1})",
roleAttributeID="uid" }]])
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: configuración de dominio de seguridad [Esto es una traducción automática]

```
<subsystem xmlns="urn:jboss:domain:security:2.0">
  ...
  <security-domains>
    ...
    <security-domain name="application-security">
      <authentication>
        <login-module code="LdapExtended" flag="required">
          <module-option name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
          <module-option name="java.naming.provider.url"
value="ldap://localhost:10389"/>
          <module-option name="java.naming.security.authentication"
value="simple"/>
          <module-option name="bindDN" value="uid=admin,ou=system"/>
          <module-option name="bindCredential" value="secret"/>
          <module-option name="baseCtxDN" value="ou=users,dc=group-
to-principal,dc=wildfly,dc=org"/>
          <module-option name="baseFilter" value="(uid={0})"/>
          <module-option name="rolesCtxDN"
value="ou=groups,dc=group-to-principal,dc=wildfly,dc=org"/>
          <module-option name="roleFilter" value="(uniqueMember=
{1})"/>
          <module-option name="roleAttributeID" value="uid"/>
        </login-module>
      </authentication>
    </security-domain>
  </security-domains>
</subsystem>
```

7.3.2.1. Migre la autenticación LDAP heredada a Elytron [Esto es una traducción automática]

Siga estos pasos para migrar la configuración de ejemplo de autenticación LDAP anterior a Elytron. Esta sección se aplica a la migración de un [legado de seguridad LDAP reino](#) así como también [Dominio de seguridad LDAP de PicketBox](#).

1. Ejemplo: Propiedades de seguridad definidas en **security** Subsistema [Esto es una traducción automática]

```
/subsystem=elytron/dir-context=ldap-
connection:add(url=ldap://localhost:10389, principal="uid=admin,
ou=system", credential-reference={clear-text=secret})
```

2. Cree un dominio de seguridad para buscar LDAP y verifique la contraseña proporcionada.

```
/subsystem=elytron/ldap-realm=ldap-realm:add(dir-context=ldap-
connection, direct-verification=true, identity-mapping={search-base-
dn="ou=users, dc=group-to-principal, dc=wildfly, dc=org", rdn-
identifier="uid", attribute-mapping=[{filter-base-dn="ou=groups,
dc=group-to-principal, dc=wildfly, dc=org", filter="(uniqueMember=
{1})", from="uid", to="Roles"}]})
```

Estos pasos resultan en lo siguiente **elytron** configuración del subsistema en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
  ...
  <security-realms>
    ...
    <ldap-realm name="ldap-realm" dir-context="ldap-connection" direct-
verification="true">
      <identity-mapping rdn-identifier="uid" search-base-
dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org">
        <attribute-mapping>
          <attribute from="uid" to="Roles" filter="(uniqueMember={1})"
filter-base-dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org"/>
        </attribute-mapping>
      </identity-mapping>
    </ldap-realm>
  </security-realms>
  ...
  <dir-contexts>
    <dir-context name="ldap-connection" url="ldap://localhost:10389"
principal="uid=admin,ou=system">
      <credential-reference clear-text="secret"/>
    </dir-context>
  </dir-contexts>
</subsystem>
```



NOTA

Por defecto, si no **role-decoder** se define para un dado **security-domain**, el atributo de identidad "Roles" se asigna a los roles de identidad.

La información cargada desde LDAP ahora puede asociarse con identidades como atributos. Estos atributos se pueden asignar a roles, pero también se pueden cargar y usar para otros fines. El reino de seguridad recién creado se puede usar en un dominio de seguridad de la misma manera que se describe en el [Migre la autenticación y autorización basada en las propiedades a Elytron](#) sección de esta guía.

7.3.3. Migre la configuración de autenticación de la base de datos a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar la autenticación de PicketBox basada en la fuente de datos JDBC a Elytron. Gran parte de la información proporcionada en la sección titulada [Migre la autenticación y autorización basada en las propiedades a Elytron](#) se aplica aquí, particularmente con respecto a cómo definir dominios de seguridad y fábricas de autenticación, y cómo mapearlas para ser utilizadas para la autenticación. Esta sección no repite esas instrucciones, así que asegúrese de leer esa sección antes de continuar.

Los siguientes ejemplos suponen que los datos de autenticación de usuario se almacenan en una tabla de base de datos creada usando sintaxis similar al ejemplo siguiente.

Ejemplo: sintaxis para crear la tabla de usuario de la base de datos [Esto es una traducción automática]

```
CREATE TABLE User (
    id BIGINT NOT NULL,
    username VARCHAR(255),
    password VARCHAR(255),
    role ENUM('admin', 'manager', 'user'),
    PRIMARY KEY (id),
    UNIQUE (username)
)
```

Para fines de autenticación, el nombre de usuario se compara con los datos almacenados en el **username** columna, se espera que la contraseña se almacene como un hash MD5 con codificación hexadecimal en el **password** columna, y la función de usuario para fines de autorización se almacena en **role** columna.

El dominio de seguridad PicketBox está configurado para usar un origen de datos JDBC para recuperar datos de la tabla de la base de datos, y luego usarlos para verificar el nombre de usuario y la contraseña, y para asignar roles. Supongamos que el dominio de seguridad PicketBox se configura con los siguientes comandos CLI de administración.

Ejemplo: Comandos de configuración de LoginModule de la base de datos de PicketBox [Esto es una traducción automática]

```
/subsystem=security/security-domain=application-security:add
/subsystem=security/security-domain=application-
security/authentication=classic:add( login-modules=[ { code=Database,
flag=Required, module-options={
dsJndiName="java:jboss/datasources/ExampleDS", principalsQuery="SELECT
password FROM User WHERE username = ?", rolesQuery="SELECT role, 'Roles'
FROM User WHERE username = ?", hashAlgorithm=MD5, hashEncoding=base64 } } ]
)
```

Esto resulta en lo siguiente **login-module** configuración en el legado **security** subsistema.

Ejemplo: Configuración de PicketBox LoginModule [Esto es una traducción automática]

```
<subsystem xmlns="urn:jboss:domain:security:2.0">
  <security-domains>
    ...
    <security-domain name="application-security">
```

```

    <authentication>
      <login-module code="Database" flag="required">
        <module-option name="dsJndiName"
value="java:jboss/datasources/ExampleDS"/>
        <module-option name="principalsQuery" value="SELECT password
FROM User WHERE username = ?"/>
        <module-option name="rolesQuery" value="SELECT role, 'Roles'
FROM User WHERE username = ?"/>
        <module-option name="hashAlgorithm" value="MD5"/>
        <module-option name="hashEncoding" value="base64"/>
      </login-module>
    </authentication>
  </security-domain>
</security-domains>
</subsystem>

```

7.3.3.1. Migre la autenticación de base de datos heredada a Elytron [Esto es una traducción automática]

Para migrar la configuración de ejemplo de autenticación de base de datos anterior a Elytron, debe definir un dominio JDBC para habilitar el acceso a la fuente de datos JDBC por Elytron.

Use el siguiente comando de administración para definir el **jdbc-realm**.

```

/subsystem=elytron/jdbc-realm=jdbc-realm:add(principal-query=[ { data-
source=ExampleDS, sql="SELECT role, password FROM User WHERE username =
?", attribute-mapping=[{index=1, to=Roles } ] simple-digest-mapper=
{algorithm=simple-digest-md5, password-index=2} } ] )

```

Esto resulta en lo siguiente **jdbc-realm** configuración en el **elytron** subsistema del archivo de configuración del servidor.

```

<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
  ...
  <security-realms>
    ...
    <jdbc-realm name="jdbc-realm">
      <principal-query sql="SELECT role, password FROM User WHERE username
= ?" data-source="ExampleDS">
        <attribute-mapping>
          <attribute to="Roles" index="1"/>
        </attribute-mapping>
        <simple-digest-mapper password-index="2"/>
      </principal-query>
    </jdbc-realm>
    ...
  </security-realms>
  ...
</subsystem>

```

Elytron ahora administra la autenticación de la base de datos utilizando la configuración de reino JDBC. Elytron es más eficiente que PicketBox porque usa una consulta SQL para obtener todos los atributos de usuario y credenciales, y luego extrae datos de los resultados de SQL y crea una asignación de los

atributos para usar para la autenticación.

7.3.4. Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]

Cuando se trabaja con una configuración de Kerberos, el servidor JBoss EAP puede confiar en la información de configuración del entorno, o la configuración de la clave se puede especificar utilizando las propiedades del sistema. Esta sección explica cómo migrar [Kerberos HTTP](#) y [Kerberos SASL](#) autenticación.

Los ejemplos que siguen suponen que Kerberos se configura utilizando las siguientes propiedades del sistema. Estas propiedades del sistema son aplicables tanto a la configuración heredada como a la configuración migrada de Elytron.

Ejemplo: Comandos de la CLI de administración de propiedades del sistema Kerberos [Esto es una traducción automática]

```
# Enable debugging
/system-property=sun.security.krb5.debug:add(value=true)
# Identify the Kerberos realm to use
/system-property=java.security.krb5.realm:add(value=ELYTRON.ORG)
# Identify the address of the KDC
/system-property=java.security.krb5.kdc:add(value=kdc.elytron.org)
```

Ejemplo: Kerberos System Properties Server Configuration [Esto es una traducción automática]

```
<system-properties>
  <property name="sun.security.krb5.debug" value="true"/>
  <property name="java.security.krb5.realm" value="ELYTRON.ORG"/>
  <property name="java.security.krb5.kdc" value="kdc.elytron.org"/>
</system-properties>
```

Elija una de las siguientes opciones de migración:

- [Migrar autenticación Kerberos HTTP](#).
- [Migrar Kerberos Remoting SASL Autenticación](#).

Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]

En las configuraciones de seguridad heredadas, puede definir un reino de seguridad para habilitar la autenticación SPNEGO para la interfaz de administración HTTP de la siguiente manera.

Ejemplo: habilitar la autenticación SPNEGO para la interfaz de gestión HTTP [Esto es una traducción automática]

```
/core-service=management/security-realm=Kerberos:add
/core-service=management/security-realm=Kerberos/server-
identity=kerberos:add
/core-service=management/security-realm=Kerberos/server-
identity=kerberos/keytab=HTTP\test-
server.elytron.org@ELYTRON.ORG:add(path=/path/to/test-server.keytab,
```



```

debug=true)
/core-service=management/security-
realm=Kerberos/authentication=kerberos:add(remove-realm=true)

```

Ejemplo: Kerberos Security Realm Configuration [Esto es una traducción automática]

```

<security-realms>
  ...
  <security-realm name="Kerberos">
    <server-identities>
      <kerberos>
        <keytab principal="HTTP/test-server.elytron.org@ELYTRON.ORG"
path="/path/to/test-server.keytab" debug="true"/>
      </kerberos>
    </server-identities>
    <authentication>
      <kerberos remove-realm="true"/>
    </authentication>
  </security-realm>
</security-realms>

```

También puede definir un par de dominios de seguridad heredados para permitir que las aplicaciones utilicen la autenticación HTTP de Kerberos.

Ejemplo: definir múltiples dominios de seguridad [Esto es una traducción automática]

```

# Define the first security domain
/subsystem=security/security-domain=host:add
/subsystem=security/security-domain=host/authentication=classic:add
/subsystem=security/security-domain=host/authentication=classic/login-
module=1:add(code=Kerberos, flag=Required, module-options={storeKey=true,
useKeyTab=true, principal=HTTP/test-server.elytron.org@ELYTRON.ORG,
keyTab=/path/to/test-server.keytab, debug=true})

# Define the second SPNEGO security domain
/subsystem=security/security-domain=SPNEGO:add
/subsystem=security/security-domain=SPNEGO/authentication=classic:add
/subsystem=security/security-domain=SPNEGO/authentication=classic/login-
module=1:add(code=SPNEGO, flag=requisite, module-options={password-
stacking=useFirstPass, serverSecurityDomain=host})
/subsystem=security/security-domain=SPNEGO/authentication=classic/login-
module=1:write-attribute(name=module,
value=org.jboss.security.negotiation)
/subsystem=security/security-domain=SPNEGO/authentication=classic/login-
module=2:add(code=UsersRoles, flag=required, module-options={password-
stacking=useFirstPass, usersProperties= /path/to/kerberos/spnego-
users.properties, rolesProperties= /path/to/kerberos/spnego-
roles.properties, defaultUsersProperties= /path/to/kerberos/spnego-
users.properties, defaultRolesProperties= /path/to/kerberos/spnego-
roles.properties})

```

Ejemplo: Configuración usando un par de dominios de seguridad [Esto es una traducción automática]

```

<subsystem xmlns="urn:jboss:domain:security:2.0">
  <security-domains>
    ...
    <security-domain name="host">
      <authentication>
        <login-module name="1" code="Kerberos" flag="required">
          <module-option name="storeKey" value="true"/>
          <module-option name="useKeyTab" value="true"/>
          <module-option name="principal" value="HTTP/test-
server.elytron.org@ELYTRON.ORG"/>
          <module-option name="keyTab" value="/path/to/test-
server.keytab"/>
          <module-option name="debug" value="true"/>
        </login-module>
      </authentication>
    </security-domain>
    <security-domain name="SPNEGO">
      <authentication>
        <login-module name="1" code="SPNEGO" flag="requisite"
module="org.jboss.security.negotiation">
          <module-option name="password-stacking" value="useFirstPass"/>
          <module-option name="serverSecurityDomain" value="host"/>
        </login-module>
        <login-module name="2" code="UsersRoles" flag="required">
          <module-option name="password-stacking" value="useFirstPass"/>
          <module-option name="usersProperties"
value="/path/to/kerberos/spnego-users.properties"/>
          <module-option name="rolesProperties" value="
/path/to/kerberos/spnego-roles.properties"/>
          <module-option name="defaultUsersProperties" value="
/path/to/kerberos/spnego-users.properties"/>
          <module-option name="defaultRolesProperties" value="
/path/to/kerberos/spnego-roles.properties"/>
        </login-module>
      </authentication>
    </security-domain>
  </security-domains>
</subsystem>

```

Las aplicaciones heredadas luego se implementan haciendo referencia al dominio de seguridad SPNEGO y aseguradas con el mecanismo SPNEGO.

Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]

Tanto la interfaz de administración como las aplicaciones se pueden proteger en Elytron utilizando un reino de seguridad y una fábrica de seguridad Kerberos.

1. Defina un reino de seguridad que se utilizará para cargar información de identidad.

```

/subsystem=elytron/properties-realm=spnego-properties:add(users-
properties={path=path/to/spnego-users.properties, plain-text=true,
digest-realm-name=ELYTRON.ORG}, groups-properties=
{path=path/to/spnego-roles.properties})

```

2. Defina una fábrica de seguridad Kerberos que permita al servidor cargar su propia identidad Kerberos.

```
/subsystem=elytron/kerberos-security-factory=test-
server:add(path=path/to/test-server.keytab, principal=HTTP/test-
server.elytron.org@ELYTRON.ORG, debug=true)
```

3. Defina un dominio de seguridad para reunir la política, así como una fábrica de autenticación HTTP para la política de autenticación.

```
/subsystem=elytron/security-domain=SPNEGODomain:add(default-
realm=spnego-properties, realms=[{realm=spnego-properties, role-
decoder=groups-to-roles}], permission-mapper=default-permission-
mapper)
/subsystem=elytron/http-authentication-factory=spnego-http-
authentication:add(security-domain=SPNEGODomain, http-server-
mechanism-factory=global,mechanism-configurations=[{mechanism-
name=SPNEGO, credential-security-factory=test-server}])
```

Esto da como resultado la siguiente configuración en **elytron** subsistema del archivo de configuración del servidor.

Ejemplo: configuración migrada de Elytron [Esto es una traducción automática]

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-
providers="combined-providers" disallowed-providers="OracleUcrypto">
  ...
  <security-domains>
    ...
    <security-domain name="SPNEGODomain" default-realm="spnego-
properties" permission-mapper="default-permission-mapper">
      <realm name="spnego-properties" role-decoder="groups-to-
roles"/>
    </security-domain>
  </security-domains>
  <security-realms>
    ...
    <properties-realm name="spnego-properties">
      <users-properties path="path/to/spnego-users.properties"
digest-realm-name="ELYTRON.ORG" plain-text="true"/>
      <groups-properties path="path/to/spnego-roles.properties"/>
    </properties-realm>
  </security-realms>
  <credential-security-factories>
    <kerberos-security-factory name="test-server"
principal="HTTP/test-server.elytron.org@ELYTRON.ORG"
path="path/to/test-server.keytab" debug="true"/>
  </credential-security-factories>
  ...
  <http>
    ...
    <http-authentication-factory name="spnego-http-authentication"
http-server-mechanism-factory="global" security-
domain="SPNEGODomain">
      <mechanism-configuration>
        <mechanism mechanism-name="SPNEGO" credential-security-
factory="test-server"/>
      </mechanism-configuration>
    </http-authentication-factory>
  </http>
</subsystem>
```

```

        </http-authentication-factory>
        ...
    </http>
    ...
</subsystem>

```

4. Para proteger la aplicación, defina un dominio de seguridad de la aplicación en **undertow** subsistema para mapear dominios de seguridad a este **http-authentication-factory**. La interfaz de gestión de HTTP se puede actualizar para hacer referencia a la **http-authentication-factory** definido en esta configuración. Este proceso está documentado en [Migre la autenticación y autorización basada en las propiedades a Elytron](#) sección de esta guía.

Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]

Es posible definir un reino de seguridad heredado para la autenticación Kerberos / GSSAPI SASL que se utilizará para la autenticación remota, como la interfaz de administración nativa.

Ejemplo: Autenticación de Kerberos para los comandos de la CLI de administración remota [Esto es una traducción automática]

```

/core-service=management/security-realm=Kerberos:add
/core-service=management/security-realm=Kerberos/server-identity=kerberos:add
/core-service=management/security-realm=Kerberos/server-identity=kerberos/keytab=remote\test-server.elytron.org@ELYTRON.ORG:add(path=path/to/remote-test-server.keytab, debug=true)
/core-service=management/security-realm=Kerberos/authentication=kerberos:add(remove-realm=true)

```

Ejemplo: Kerberos Remoting Security Realm Configuration [Esto es una traducción automática]

```

<management>
  <security-realms>
    ...
    <security-realm name="Kerberos">
      <server-identities>
        <kerberos>
          <keytab principal="remote/test-server.elytron.org@ELYTRON.ORG"
path="path/to/remote-test-server.keytab" debug="true"/>
        </kerberos>
      </server-identities>
      <authentication>
        <kerberos remove-realm="true"/>
      </authentication>
    </security-realm>
  </security-realms>
  ...
</management>

```

Migre la autenticación de Kerberos a Elytron [Esto es una traducción automática]

Los pasos para definir la configuración de Elytron equivalente son muy similares a los descritos en [Migrar autenticación Kerberos HTTP](#).

1. Defina un reino de seguridad que se utilizará para cargar información de identidad.

```
/path=kerberos:add(relative-to=user.home, path=src/kerberos)
/subsystem=elytron/properties-realm=kerberos-properties:add(users-
properties={path=kerberos-users.properties, relative-to=kerberos,
digest-realm-name=ELYTRON.ORG}, groups-properties={path=kerberos-
groups.properties, relative-to=kerberos})
```

2. Defina la fábrica de seguridad Kerberos para la identidad del servidor.

```
/subsystem=elytron/kerberos-security-factory=test-
server:add(relative-to=kerberos, path=remote-test-server.keytab,
principal=remote/test-server.elytron.org@ELYTRON.ORG)
```

3. Ejemplo: Dominio de Seguridad Elytron y Comandos de Configuración de Fábrica de Autenticación [Esto es una traducción automática]

```
/subsystem=elytron/security-domain=KerberosDomain:add(default-
realm=kerberos-properties, realms=[{realm=kerberos-properties, role-
decoder=groups-to-roles}], permission-mapper=default-permission-
mapper)
/subsystem=elytron/sasl-authentication-factory=gssapi-
authentication-factory:add(security-domain=KerberosDomain, sasl-
server-factory=elytron, mechanism-configurations=[{mechanism-
name=GSSAPI, credential-security-factory=test-server}])
```

Esto da como resultado la siguiente configuración en **elytron** subsistema del archivo de configuración del servidor.

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
  ...
  <security-domains>
    ...
    <security-domain name="KerberosDomain" default-realm="kerberos-
properties" permission-mapper="default-permission-mapper">
      <realm name="kerberos-properties" role-decoder="groups-to-roles"/>
    </security-domain>
  </security-domains>
  <security-realms>
    ...
    <properties-realm name="kerberos-properties">
      <users-properties path="kerberos-users.properties" relative-
to="kerberos" digest-realm-name="ELYTRON.ORG"/>
      <groups-properties path="kerberos-groups.properties" relative-
to="kerberos"/>
    </properties-realm>
  </security-realms>
  <credential-security-factories>
    <kerberos-security-factory name="test-server" principal="remote/test-
server.elytron.org@ELYTRON.ORG" path="remote-test-server.keytab" relative-
to="kerberos"/>
  </credential-security-factories>
  ...
  <sasl>
```

```

...
<sasl-authentication-factory name="gssapi-authentication-factory"
sasl-server-factory="elytron" security-domain="KerberosDomain">
  <mechanism-configuration>
    <mechanism mechanism-name="GSSAPI" credential-security-
factory="test-server"/>
  </mechanism-configuration>
</sasl-authentication-factory>
...
</sasl>
</subsystem>

```

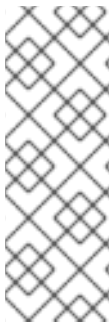
La interfaz de gestión o los conectores remotos ahora se pueden actualizar para hacer referencia a la fábrica de autenticación SASL.

Los dos ejemplos de Elytron definidos aquí también podrían combinarse para usar un dominio de seguridad y un dominio de seguridad compartidos y simplemente usar fábricas de autenticación específicas del protocolo cada una haciendo referencia a su propia fábrica de seguridad de Kerberos.

7.3.5. Migre tiendas compuestas a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar un [PicketBox](#) o [reino de seguridad heredado](#) configuración que usa múltiples almacenes de identidad para [Elitrono](#). Al utilizar PicketBox o los reinos de seguridad heredados, es posible definir una configuración donde la autenticación se realiza en un único almacén de identidades mientras que la información utilizada para la autorización se carga desde un almacén diferente. Al migrar a Elytron, esto se puede lograr mediante el uso de un reino de seguridad agregado.

Los siguientes ejemplos realizan autenticación de usuario utilizando el **example-users.properties** archivo de propiedades, y luego consulta LDAP para cargar la información del grupo y la función.



NOTA

Las configuraciones que se muestran se basan en los ejemplos de las siguientes secciones, que brindan información de fondo adicional:

- Migre la autenticación y autorización basada en las propiedades a Elytron [Esto es una traducción automática]
- [Migre la configuración de autenticación LDAP a Elytron](#)

Ejemplo: Configuración de PicketBox LoginModule [Esto es una traducción automática]

El dominio de seguridad PicketBox para este escenario se configura mediante los siguientes comandos CLI de administración.

Ejemplo: Comandos de configuración de PicketBox [Esto es una traducción automática]

```

/subsystem=security/security-domain=application-security:add

/subsystem=security/security-domain=application-
security/authentication=classic:add(login-modules=[ {code=UsersRoles,
flag=Required, module-options={ password-stacking=useFirstPass,
usersProperties=file://${jboss.server.config.dir}/example-
users.properties}} {code=LdapExtended, flag=Required, module-options={
password-stacking=useFirstPass,
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory,

```



```
java.naming.provider.url=ldap://localhost:10389,
java.naming.security.authentication=simple, bindDN="uid=admin,ou=system",
bindCredential=secret, baseCtxDN="ou=users,dc=group-to-
principal,dc=wildfly,dc=org", baseFilter="(uid={0})",
rolesCtxDN="ou=groups,dc=group-to-
principal,dc=wildfly,dc=org", roleFilter="(uniqueMember={1})",
roleAttributeID="uid" }]])
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: configuración del dominio de seguridad PicketBox [Esto es una traducción automática]

```
<security-domain name="application-security">
  <authentication>
    <login-module code="UsersRoles" flag="required">
      <module-option name="password-stacking" value="useFirstPass"/>
      <module-option name="usersProperties"
value="file://${jboss.server.config.dir}/example-users.properties"/>
    </login-module>
    <login-module code="LdapExtended" flag="required">
      <module-option name="password-stacking" value="useFirstPass"/>
      <module-option name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
      <module-option name="java.naming.provider.url"
value="ldap://localhost:10389"/>
      <module-option name="java.naming.security.authentication"
value="simple"/>
      <module-option name="bindDN" value="uid=admin,ou=system"/>
      <module-option name="bindCredential" value="secret"/>
      <module-option name="baseCtxDN" value="ou=users,dc=group-to-
principal,dc=wildfly,dc=org"/>
      <module-option name="baseFilter" value="(uid={0})"/>
      <module-option name="rolesCtxDN" value="ou=groups,dc=group-to-
principal,dc=wildfly,dc=org"/>
      <module-option name="roleFilter" value="(uniqueMember={1})"/>
      <module-option name="roleAttributeID" value="uid"/>
    </login-module>
  </authentication>
</security-domain>
```

Ver [Configuración del reino de seguridad agregada de Elytron](#) para saber cómo configurar un reino de seguridad agregado en el **elytron** subsistema para lograr esto.

Ejemplo: Comandos de configuración de dominios de seguridad heredados [Esto es una traducción automática]

La configuración del reino de seguridad heredada para este escenario se configura mediante los siguientes comandos CLI de administración.

Ejemplo: Comandos de configuración de dominios de seguridad heredados [Esto es una traducción automática]

```
/core-service=management/ldap-
connection=MyLdapConnection:add(url="ldap://localhost:10389", search-
dn="uid=admin,ou=system", search-credential="secret")
```

```

/core-service=management/security-realm=ApplicationSecurity:add
/core-service=management/security-
realm=ApplicationSecurity/authentication=properties:add(path=example-
users.properties, relative-to=jboss.server.config.dir, plain-text=true)

batch
/core-service=management/security-
realm=ApplicationSecurity/authorization=ldap:add(connection=MyLdapConnecti
on)
/core-service=management/security-
realm=ApplicationSecurity/authorization=ldap/username-to-dn=username-
filter:add(attribute=uid, base-dn="ou=users,dc=group-to-
principal,dc=wildfly,dc=org")
/core-service=management/security-
realm=ApplicationSecurity/authorization=ldap/group-search=group-to-
principal:add(base-dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org",
iterative=true, prefer-original-connection=true, principal-
attribute=uniqueMember, search-by=DISTINGUISHED_NAME, group-name=SIMPLE,
group-name-attribute=uid)
run-batch

```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: Comandos de configuración de dominios de seguridad heredados [Esto es una traducción automática]

```

<security-realms>
...
  <security-realm name="ApplicationSecurity">
    <authentication>
      <properties path="example-users.properties" relative-
to="jboss.server.config.dir" plain-text="true"/>
    </authentication>
    <authorization>
      <ldap connection="MyLdapConnection">
        <username-to-dn>
          <username-filter base-dn="ou=users,dc=group-to-
principal,dc=wildfly,dc=org" attribute="uid"/>
        </username-to-dn>
        <group-search group-name="SIMPLE" iterative="true" group-name-
attribute="uid">
          <group-to-principal search-by="DISTINGUISHED_NAME" base-
dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org" prefer-original-
connection="true">
            <membership-filter principal-attribute="uniqueMember"/>
          </group-to-principal>
        </group-search>
      </ldap>
    </authorization>
  </security-realm>
</security-realms>
<outbound-connections>
  <ldap name="MyLdapConnection" url="ldap://localhost:10389" search-
dn="uid=admin,ou=system" search-credential="secret"/>
</outbound-connections>

```


Ver [Configuración del reino de seguridad agregada de Elytron](#) para saber cómo configurar un reino de seguridad agregado en el **elytron** subsistema para lograr esto.

Ejemplo: Configuración de LDAP Security Realm [Esto es una traducción automática]

La configuración equivalente de Elytron para este escenario se configura con los siguientes comandos CLI de administración.

Ejemplo: Comandos de configuración de Elytron [Esto es una traducción automática]

```
/subsystem=elytron/dir-context=ldap-
connection:add(url=ldap://localhost:10389,
principal="uid=admin,ou=system", credential-reference={clear-text=secret}))

/subsystem=elytron/ldap-realm=ldap-realm:add(dir-context=ldap-connection,
direct-verification=true, identity-mapping={search-base-
dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org", rdn-
identifier="uid", attribute-mapping=[{filter-base-dn="ou=groups,dc=group-
to-principal,dc=wildfly,dc=org", filter="(uniqueMember=
{1})", from="uid", to="Roles"}]})

/subsystem=elytron/properties-realm=application-properties:add(users-
properties={path=example-users.properties, relative-
to=jboss.server.config.dir, plain-text=true, digest-realm-
name="Application Security"})

/subsystem=elytron/aggregate-realm=combined-realm:add(authentication-
realm=application-properties, authorization-realm=ldap-realm)

/subsystem=elytron/security-domain=application-security:add(realms=
[{realm=combined-realm}], default-realm=combined-realm, permission-
mapper=default-permission-mapper)
/subsystem=elytron/http-authentication-factory=application-security-
http:add(http-server-mechanism-factory=global, security-
domain=application-security, mechanism-configurations=[{mechanism-
name=BASIC}])
```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: configuración Elytron [Esto es una traducción automática]

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-
providers" disallowed-providers="OracleUcrypto">
  ...
  <security-domains>
    ...
    <security-domain name="application-security" default-realm="combined-
realm" permission-mapper="default-permission-mapper">
      <realm name="combined-realm"/>
    </security-domain>
  </security-domains>
  <security-realms>
    <aggregate-realm name="combined-realm" authentication-
realm="application-properties" authorization-realm="ldap-realm"/>
  </security-realms>
  ...
</subsystem>
```

```

    <properties-realm name="application-properties">
      <users-properties path="example-users.properties" relative-
to="jboss.server.config.dir" digest-realm-name="Application Security"
plain-text="true"/>
    </properties-realm>
    <ldap-realm name="ldap-realm" dir-context="ldap-connection" direct-
verification="true">
      <identity-mapping rdn-identifier="uid" search-base-
dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org">
        <attribute-mapping>
          <attribute from="uid" to="Roles" filter="(uniqueMember={1})"
filter-base-dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org"/>
        </attribute-mapping>
      </identity-mapping>
    </ldap-realm>
  </security-realms>
  ...
  <http>
    ...
    <http-authentication-factory name="application-security-http" http-
server-mechanism-factory="global" security-domain="application-security">
      <mechanism-configuration>
        <mechanism mechanism-name="BASIC"/>
      </mechanism-configuration>
    </http-authentication-factory>
    ...
  </http>
  ...
  <dir-contexts>
    <dir-context name="ldap-connection" url="ldap://localhost:10389"
principal="uid=admin,ou=system">
      <credential-reference clear-text="secret"/>
    </dir-context>
  </dir-contexts>
</subsystem>

```

En el **elytron** subsistema, un **aggregate-realm** ha sido definido que especifica qué dominios de seguridad usar para la autenticación y cuáles usar para las decisiones de autorización.

7.3.6. Migre los dominios de seguridad que usan almacenamiento en caché para Elytron [Esto es una traducción automática]

Al usar PicketBox, es posible definir un dominio de seguridad y habilitar el almacenamiento en memoria caché en memoria para su acceso. Esto le permite acceder a los datos de identidad en la memoria y evita el acceso directo adicional al almacén de identidades. Es posible lograr una configuración similar con Elytron. Esta sección describe cómo configurar el almacenamiento en caché del dominio de seguridad cuando se usa Elytron.

Ejemplo: configuración de dominio de seguridad en caché PicketBox [Esto es una traducción automática]

Los siguientes comandos muestran cómo configurar un dominio de seguridad PicketBox que permite el almacenamiento en caché.

Ejemplo: Comandos del dominio de seguridad en caché de PicketBox [Esto es una traducción automática]

```

/subsystem=security/security-domain=application-security:add(cache-
type=default)
/subsystem=security/security-domain=application-
security/authentication=classic:add(login-modules=[{code=LdapExtended,
flag=Required, module-options={
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory,
java.naming.provider.url=ldap://localhost:10389,
java.naming.security.authentication=simple, bindDN="uid=admin,ou=system",
bindCredential=secret, baseCtxDN="ou=users,dc=group-to-
principal,dc=wildfly,dc=org", baseFilter="(uid={0})",
rolesCtxDN="ou=groups,dc=group-to-principal,dc=wildfly,dc=org",
roleFilter="(uniqueMember={1})", roleAttributeID="uid" }]])

```

Esto da como resultado la siguiente configuración de servidor.

Ejemplo: configuración de dominio de seguridad en caché PicketBox [Esto es una traducción automática]

```

<subsystem xmlns="urn:jboss:domain:security:2.0">
  <security-domains>
    ...
    <security-domain name="application-security" cache-type="default">
      <authentication>
        <login-module code="LdapExtended" flag="required">
          <module-option name="java.naming.factory.initial"
value="com.sun.jndi.ldap.LdapCtxFactory"/>
          <module-option name="java.naming.provider.url"
value="ldap://localhost:10389"/>
          <module-option name="java.naming.security.authentication"
value="simple"/>
          <module-option name="bindDN" value="uid=admin,ou=system"/>
          <module-option name="bindCredential" value="secret"/>
          <module-option name="baseCtxDN" value="ou=users,dc=group-to-
principal,dc=wildfly,dc=org"/>
          <module-option name="baseFilter" value="(uid={0})"/>
          <module-option name="rolesCtxDN" value="ou=groups,dc=group-to-
principal,dc=wildfly,dc=org"/>
          <module-option name="roleFilter" value="(uniqueMember={1})"/>
          <module-option name="roleAttributeID" value="uid"/>
        </login-module>
      </authentication>
    </security-domain>
  </security-domains>
</subsystem>

```

NOTA

Este comando y la configuración resultante es similar al ejemplo que se muestra en [Migre la configuración de autenticación LDAP a Elytron](#); Sin embargo, aquí el atributo **cache-type** se define con un valor de **default**. Los **default** el tipo de caché es un caché en memoria. Al usar PicketBox, también puede especificar un **cache-type** de **infinispan**, Sin embargo, este tipo no es compatible con Elytron.

Ejemplo: configuración de dominio de seguridad en caché Elytron [Esto es una traducción automática]

Siga los pasos a continuación para crear una configuración similar que almacena en caché un dominio de seguridad cuando usa Elytron.

1. Defina un reino de seguridad y ajuste el dominio de seguridad en un reino de almacenamiento en caché. El dominio de almacenamiento en caché se puede usar en un dominio de seguridad y, posteriormente, en una fábrica de autenticación.

Ejemplo: Comandos de configuración de Elytron Security Realm [Esto es una traducción automática]

```
/subsystem=elytron/dir-context=ldap-connection:add(url=ldap://localhost:10389,principal="uid=admin,ou=system", credential-reference={clear-text=secret})
/subsystem=elytron/ldap-realm=ldap-realm:add(dir-context=ldap-connection, direct-verification=true, identity-mapping={search-base-dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org", rdn-identifier="uid", attribute-mapping=[{filter-base-dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org", filter="(uniqueMember={1})", from="uid", to="Roles"}]})
/subsystem=elytron/caching-realm=cached-ldap:add(realm=ldap-realm)
```

2. Defina un dominio de seguridad y una fábrica de autenticación HTTP que utilicen **cached-ldap** reino definido en el paso anterior.

Ejemplo: Dominio de Seguridad Elytron y Comandos de Configuración de Fábrica de Autenticación [Esto es una traducción automática]

```
/subsystem=elytron/security-domain=application-security:add(realms=[{realm=cached-ldap}], default-realm=cached-ldap, permission-mapper=default-permission-mapper)
/subsystem=elytron/http-authentication-factory=application-security-http:add(http-server-mechanism-factory=global, security-domain=application-security, mechanism-configurations=[{mechanism-name=BASIC}])
```

**NOTA**

En este paso, es importante que haga referencia al **caching-realm** en lugar del reino original. De lo contrario, el almacenamiento en caché se pasa por alto.

Estos comandos dan como resultado las siguientes adiciones a la configuración del servidor.

Ejemplo: configuración de dominio de seguridad en caché Elytron [Esto es una traducción automática]

```
<subsystem xmlns="urn:wildfly:elytron:1.2" final-providers="combined-providers" disallowed-providers="OracleUcrypto">
...
<security-domains>
...
<security-domain name="application-security" default-realm="cached-
```

```

ldap" permission-mapper="default-permission-mapper">
    <realm name="cached-ldap" />
  </security-domain>
</security-domains>
...
<security-realms>
  ....
  <ldap-realm name="ldap-realm" dir-context="ldap-connection" direct-
verification="true">
    <identity-mapping rdn-identifier="uid" search-base-
dn="ou=users,dc=group-to-principal,dc=wildfly,dc=org">
      <attribute-mapping>
        <attribute from="uid" to="Roles" filter="(uniqueMember={1})"
filter-base-dn="ou=groups,dc=group-to-principal,dc=wildfly,dc=org"/>
      </attribute-mapping>
    </identity-mapping>
  </ldap-realm>
  <cached-ldap name="cached-ldap" realm="ldap-realm"/>
</security-realms>
...
<http>
  ...
  <http-authentication-factory name="application-security-http" http-
server-mechanism-factory="global" security-domain="application-security">
    <mechanism-configuration>
      <mechanism mechanism-name="BASIC"/>
    </mechanism-configuration>
  </http-authentication-factory>
  ...
</http>
...
<dir-contexts>
  <dir-context name="ldap-connection" url="ldap://localhost:10389"
principal="uid=admin,ou=system">
    <credential-reference clear-text="secret"/>
  </dir-context>
</dir-contexts>
...

```

7.3.7. Migre la seguridad de JACC a Elytron [Esto es una traducción automática]

Por defecto, JBoss EAP usa el legado **security** subsistema para configurar el proveedor de políticas de Contrato de Autorización de Java para Contenedores (JACC) y la fábrica. La configuración predeterminada se asigna a las implementaciones de PicketBox.

los **elytron** subsistema proporciona un proveedor de políticas incorporado basado en la especificación JACC. Antes de configurar su servidor para permitir que Elytron administre las configuraciones de JACC y otras políticas, primero debe deshabilitar JACC en el legado **security** subsistema utilizando el siguiente comando de gestión CLI.

```
/subsystem=security:write-attribute(name=initialize-jacc, value=false)
```

De lo contrario, puede producirse el siguiente error en el registro del servidor: **MSC000004: Failure during stop of service org.wildfly.security.policy: java.lang.StackOverflowError.**

Para obtener información sobre cómo habilitar JACC y definir un proveedor de políticas JACC en el **elytron** subsistema, ver [Habilitando JACC Usando el elytron Subsistema](#) en el *Guía de desarrollo* para JBoss EAP.

7.4. MIGRAR CLIENTES DE APLICACIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

7.4.1. Migre una configuración de Naming Client a Elytron [Esto es una traducción automática]

Esta sección describe cómo migrar una aplicación cliente que realiza una búsqueda JNDI remota utilizando un `org.jboss.naming.remote.client.InitialContext` clase, que está respaldado por `org.jboss.naming.remote.client.InitialContextFactory` clase, a Elytron.

Los siguientes ejemplos suponen que `InitialContextFactory` la clase se crea especificando las propiedades de las credenciales del usuario y la URL del proveedor de nombres al que se conecta.

Ejemplo: InitialContext Código utilizado en la versión anterior [Esto es una traducción automática]

```
Properties properties = new Properties();
properties.put(Context.INITIAL_CONTEXT_FACTORY,
    "org.jboss.naming.remote.client.InitialContextFactory");
properties.put(Context.PROVIDER_URL, "http-remoting://127.0.0.1:8080");
properties.put(Context.SECURITY_PRINCIPAL, "bob");
properties.put(Context.SECURITY_CREDENTIALS, "secret");
InitialContext context = new InitialContext(properties);
Bar bar = (Bar) context.lookup("foo/bar");
...
```

Puede elegir uno de los siguientes enfoques de migración:

- Migre el cliente de nombres usando el enfoque de archivo de configuración [Esto es una traducción automática]
- Migre el cliente de nombres usando el enfoque programático [Esto es una traducción automática]

7.4.1.1. Migre el cliente de nombres usando el enfoque de archivo de configuración [Esto es una traducción automática]

Migre el cliente de nombres usando el enfoque de archivo de configuración [Esto es una traducción automática]

1. Crear un `wildfly-config.xml` archivo en la aplicación cliente **META-INF/** directorio. El archivo debe contener las credenciales de usuario que se utilizarán al establecer una conexión con el proveedor de nombres.

Ejemplo: configuración equivalente usando el wildfly-config.xml Archivo [Esto es una traducción automática]

```
<configuration>
  <authentication-client xmlns="urn:elytron:1.0.1">
```

```

<authentication-rules>
  <rule use-configuration="namingConfig">
    <match-host name="127.0.0.1"/>
  </rule>
</authentication-rules>
<authentication-configurations>
  <configuration name="namingConfig">
    <set-user-name name="bob"/>
    <credentials>
      <clear-password password="secret"/>
    </credentials>
  </configuration>
</authentication-configurations>
</authentication-client>
</configuration>

```

2. Crear un **InitialContext** como en el siguiente ejemplo. Tenga en cuenta que **InitialContext** está respaldado por el **org.wildfly.naming.client.WildFlyInitialContextFactory** clase.

Ejemplo: InitialContext Código utilizado en la versión anterior [Esto es una traducción automática]

```

Properties properties = new Properties();
properties.put(Context.INITIAL_CONTEXT_FACTORY, "org.wildfly.naming.c
lient.WildFlyInitialContextFactory");
properties.put(Context.PROVIDER_URL, "remote+http://127.0.0.1:8080");
InitialContext context = new InitialContext(properties);
Bar bar = (Bar) context.lookup("foo/bar");
...

```

7.4.1.2. Migre el cliente de nombres usando el enfoque programático [Esto es una traducción automática]

Con este enfoque, proporciona las credenciales de usuario que se utilizan para establecer una conexión con el proveedor de nombres directamente en el código de la aplicación.

Migre el cliente EJB utilizando el enfoque programático [Esto es una traducción automática]

```

// Create the authentication configuration
AuthenticationConfiguration namingConfig =
AuthenticationConfiguration.empty().useName("bob").usePassword("secret");

// Create the authentication context
AuthenticationContext context =
AuthenticationContext.empty().with(MatchRule.ALL.matchHost("127.0.0.1"),
namingConfig);

// Create a callable that creates and uses an InitialContext
Callable<Void> callable = () -> {
    Properties properties = new Properties();

    properties.put(Context.INITIAL_CONTEXT_FACTORY, "org.wildfly.naming.client.

```



```

WildFlyInitialContextFactory");
    properties.put(Context.PROVIDER_URL, "remote+http://127.0.0.1:8080");
    InitialContext context = new InitialContext(properties);
    Bar bar = (Bar) context.lookup("foo/bar");
    ...
    return null;
};

// Use the authentication context to run the callable
context.runCallable(callable);

```

7.4.2. Migre un cliente EJB a Elytron [Esto es una traducción automática]

Este ejemplo de migración supone que la aplicación cliente está configurada para invocar un EJB desplegado en un servidor remoto utilizando un `jboss-ejb-client.properties` archivo. Este archivo, que se encuentra en la aplicación cliente `META-INF/` directorio, contiene la siguiente información necesaria para conectarse al servidor remoto.

Ejemplo: `jboss-ejb-client.properties` Archivo [Esto es una traducción automática]

```

remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=false
remote.connections=default
remote.connection.default.host=127.0.0.1
remote.connection.default.port = 8080
remote.connection.default.username=bob
remote.connection.default.password=secret

```

El cliente busca el EJB y llama a uno de sus métodos usando un código similar al del siguiente ejemplo.

Ejemplo: Código de cliente que llama a un EJB remoto [Esto es una traducción automática]

```

// Create an InitialContext
Properties properties = new Properties();
properties.put(Context.URL_PKG_PREFIXES, "org.jboss.ejb.client.naming");
InitialContext context = new InitialContext(properties);

// Look up the EJB and invoke one of its methods
RemoteCalculator statelessRemoteCalculator = (RemoteCalculator)
context.lookup(
    "ejb:/ejb-remote-server-side//CalculatorBean!" +
    RemoteCalculator.class.getName());
int sum = statelessRemoteCalculator.add(101, 202);

```

Puede elegir uno de los siguientes enfoques de migración:

- Migre el cliente EJB utilizando el enfoque del archivo de configuración [Esto es una traducción automática]
- Migre el cliente EJB utilizando el enfoque programático [Esto es una traducción automática]

7.4.2.1. Migre el cliente EJB utilizando el enfoque del archivo de configuración [Esto es una traducción automática]

Migre el cliente de nombres usando el enfoque de archivo de configuración [Esto es una traducción automática]

1. Configura un **wildfly-config.xml** archivo en la aplicación cliente **META-INF/** directorio. El archivo debe contener las credenciales de usuario que se utilizarán al establecer una conexión con el proveedor de nombres.

Ejemplo: configuración equivalente usando el wildfly-config.xml Archivo [Esto es una traducción automática]

```
<configuration>
  <authentication-client xmlns="urn:elytron:1.0.1">
    <authentication-rules>
      <rule use-configuration="ejbConfig">
        <match-host name="127.0.0.1"/>
      </rule>
    </authentication-rules>
    <authentication-configurations>
      <configuration name="ejbConfig">
        <set-user-name name="bob"/>
        <credentials>
          <clear-password password="secret"/>
        </credentials>
      </configuration>
    </authentication-configurations>
  </authentication-client>
  <jboss-ejb-client xmlns="urn:jboss:wildfly-client-ejb:3.0">
    <connections>
      <connection uri="remote+http://127.0.0.1:8080" />
    </connections>
  </jboss-ejb-client>
</configuration>
```

2. Crear un **InitialContext** como en el siguiente ejemplo. Tenga en cuenta que **InitialContext** está respaldado por el **org.wildfly.naming.client.WildFlyInitialContextFactory** clase.

Ejemplo: InitialContext Código utilizado en la versión anterior [Esto es una traducción automática]

```
// Create an InitialContext
Properties properties = new Properties();
properties.put(Context.INITIAL_CONTEXT_FACTORY, "org.wildfly.naming.c
lient.WildFlyInitialContextFactory");
InitialContext context = new InitialContext(properties);

// Look up an EJB and invoke one of its methods
// Note that this code is the same as before
RemoteCalculator statelessRemoteCalculator = (RemoteCalculator)
context.lookup(
    "ejb:/ejb-remote-server-side//CalculatorBean!" +
    RemoteCalculator.class.getName());
int sum = statelessRemoteCalculator.add(101, 202);----
```

3. Ejemplo: `jboss-ejb-client.properties` Archivo de propiedades [Esto es una traducción automática]

7.4.2.2. Migre el cliente EJB utilizando el enfoque programático [Esto es una traducción automática]

Con este enfoque, proporciona la información necesaria para conectarse al servidor remoto directamente en el código de la aplicación.

Migre el cliente EJB utilizando el enfoque programático [Esto es una traducción automática]

```
// Create the authentication configuration
AuthenticationConfiguration ejbConfig =
AuthenticationConfiguration.empty().useName("bob").usePassword("secret");

// Create the authentication context
AuthenticationContext context =
AuthenticationContext.empty().with(MatchRule.ALL.matchHost("127.0.0.1"),
ejbConfig);

// Create a callable that invokes the EJB
Callable<Void> callable = () -> {

    // Create an InitialContext
    Properties properties = new Properties();
    properties.put(Context.INITIAL_CONTEXT_FACTORY,
"org.wildfly.naming.client.WildFlyInitialContextFactory");
    properties.put(Context.PROVIDER_URL, "remote+http://127.0.0.1:8080");
    InitialContext context = new InitialContext(properties);

    // Look up the EJB and invoke one of its methods
    // Note that this code is the same as before
    RemoteCalculator statelessRemoteCalculator = (RemoteCalculator)
context.lookup(
    "ejb:/ejb-remote-server-side//CalculatorBean!" +
RemoteCalculator.class.getName());
    int sum = statelessRemoteCalculator.add(101, 202);
    ...
    return null;
};

// Use the authentication context to run the callable
context.runCallable(callable);
```

Ejemplo: `jboss-ejb-client.properties` Archivo de propiedades [Esto es una traducción automática]

7.5. MIGRAR CONFIGURACIONES DE SSL [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

7.5.1. Migre una configuración simple de SSL a Elytron [Esto es una traducción automática]

Si aseguró conexiones HTTP al servidor JBoss EAP usando un reino de seguridad, puede migrar esa configuración a Elytron utilizando la información proporcionada en esta sección.

Los siguientes ejemplos asumen que tiene lo siguiente **keystore** configurado en el **security-realm**.

Ejemplo: Configuración de SSL con un almacén de claves de Realismo de seguridad [Esto es una traducción automática]

```
<security-realm name="ApplicationRealm">
  <server-identities>
    <ssl>
      <keystore path="server.keystore" relative-
to="jboss.server.config.dir" keystore-password="keystore_password"
alias="server" key-password="key_password" />
    </ssl>
  </server-identities>
</security-realm>
```

Siga los pasos a continuación para lograr la misma configuración usando Elytron.

1. Crear un **key-store** en el **elytron** subsistema que especifica la ubicación del almacén de claves y la contraseña mediante la cual se encripta. Este comando supone que el almacén de claves se generó utilizando el comando `keytool` y su tipo es **JKS**.

```
/subsystem=elytron/key-
store=LocalhostKeyStore:add(path=server.keystore,relative-
to=jboss.server.config.dir,credential-reference={clear-
text="keystore_password"},type=JKS)
```

2. Crear un **key-manager** en el **elytron** subsistema que especifica el **key-store** definido en el paso anterior, el alias y la contraseña de la clave.

```
/subsystem=elytron/key-manager=LocalhostKeyManager:add(key-
store=LocalhostKeyStore,alias-filter=server,credential-reference=
{clear-text="key_password"})
```


3. Crear un **server-ssl-context** en el **elytron** subsistema que hace referencia a la **key-manager** eso fue definido en el paso anterior.

```
/subsystem=elytron/server-ssl-context=LocalhostSslContext:add(key-
manager=LocalhostKeyManager)
```

4. Cambiar el **https-listener** del legado **security-realm** al recién creado Elytron **ssl-context**.

```
batch
/subsystem=undertow/server=default-server/https-
listener=https:undefine-attribute(name=security-realm)
/subsystem=undertow/server=default-server/https-
listener=https:write-attribute(name=ssl-
context,value=LocalhostSslContext)
run-batch
```

5. Recargar el servidor.

 recargar

Esto resulta en lo siguiente **elytron** configuración del subsistema en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:wildfly:elytron:1.2" ...>
  ...
  <tls>
    <key-stores>
      <key-store name="LocalhostKeyStore">
        <credential-reference clear-text="keystore_password"/>
        <implementation type="JKS"/>
        <file path="server.keystore" relative-
to="jboss.server.config.dir"/>
      </key-store>
    </key-stores>
    <key-managers>
      <key-manager name="LocalhostKeyManager" key-
store="LocalhostKeyStore" alias-filter="server">
        <credential-reference clear-text="key_password"/>
      </key-manager>
    </key-managers>
    <server-ssl-contexts>
      <server-ssl-context name="LocalhostSslContext" key-
manager="LocalhostKeyManager"/>
    </server-ssl-contexts>
  </tls>
</subsystem>
```

Esto resulta en lo siguiente **undertow** configuración del subsistema en el archivo de configuración del servidor.

```
<https-listener name="https" socket-binding="https" ssl-
context="LocalhostSslContext" enable-http2="true"/>
```

Para más información, ver [Subsistema Elytron](#) y [Cómo proteger las interfaces de administración](#) en [Cómo configurar la seguridad del servidor](#) para JBoss EAP.

7.5.2. Migre la autenticación SSL de Client-Cert a Elytron [Esto es una traducción automática]

Si configuró la autenticación SSL de Client-Cert utilizando un almacén de confianza del reino de seguridad, puede migrar esa configuración a Elytron utilizando la información proporcionada en esta sección.

Los pasos siguientes suponen que tiene lo siguiente **truststore** configurado en el **security-realm**.

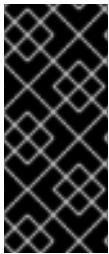
Ejemplo: Configuración de SSL utilizando Security Realm Truststore [Esto es una traducción automática]

```
<security-realm name="ApplicationRealm">
  <server-identities>
```

```

<ssl>
  <keystore path="server.keystore" relative-
to="jboss.server.config.dir" keystore-password="keystore_password"
alias="server" key-password="key_password" />
</ssl>
</server-identities>
<authentication>
  <truststore path="server.truststore" relative-
to="jboss.server.config.dir" keystore-password="truststore_password" />
  <local default-user="$local"/>
  <properties path="application-users.properties" relative-
to="jboss.server.config.dir"/>
</authentication>
<authorization>
  <properties path="application-roles.properties" relative-
to="jboss.server.config.dir"/>
</authorization>
</security-realm>

```



IMPORTANTE

Los pasos a continuación solo brindan configuración para evitar que los usuarios sin un certificado válido y una clave privada accedan al servidor. No configuran la identidad del usuario para autenticarse en el servidor. Se supone que ya ha configurado la autenticación HTTP CLIENT-CERT y la autenticación SASL externa para la autenticación de identidad del usuario.

Siga estos pasos para configurar el servidor para evitar que los usuarios sin un certificado válido y una clave privada accedan al servidor utilizando Elytron.

1. Crear un **key-store** en el **elytron** subsistema que especifica la ubicación del almacén de claves y la contraseña mediante la cual se encripta. Este comando supone que el almacén de claves se generó utilizando el comando `keytool` y su tipo es **JKS**.

```

/subsystem=elytron/key-
store=LocalhostKeyStore:add(path=server.keystore,relative-
to=jboss.server.config.dir,credential-reference={clear-
text="keystore_password"},type=JKS)

```

2. Crear un **key-store** en el **elytron** subsistema que especifica la ubicación del almacén de confianza y la contraseña mediante la cual se encripta. Este comando supone que el almacén de claves se generó utilizando el comando `keytool` y su tipo es **JKS**.

```

/subsystem=elytron/key-
store=TrustStore:add(path=server.truststore,relative-
to=jboss.server.config.dir,credential-reference={clear-
text="truststore_password"},type=JKS)

```

3. Crear un **key-manager** en el **elytron** subsistema que especifica lo previamente definido **LocalhostKeyStore** keystore, el alias y la contraseña de la clave.

```

/subsystem=elytron/key-manager=LocalhostKeyManager:add(key-
store=LocalhostKeyStore,alias-filter=server,credential-reference=
{clear-text="key_password"})

```

4. Crear un **trust-manager** en el **elytron** subsistema que especifica el **key-store** del almacén de confianza creado previamente.

```
/subsystem=elytron/trust-manager=TrustManager:add(key-
store=TrustStore)
```

5. Crear un **server-ssl-context** en el **elytron** subsistema que hace referencia a los definidos previamente **key-manager**, establece el **trust-manager** atributo, y habilita la autenticación del cliente.

```
/subsystem=elytron/server-ssl-context=LocalhostSslContext:add(key-
manager=LocalhostKeyManager,trust-manager=TrustManager,need-client-
auth=true)
```

6. Cambiar el **https-listener** del legado **security-realm** al recién creado Elytron **ssl-context**.

```
batch
/subsystem=undertow/server=default-server/https-
listener=https:undefine-attribute(name=security-realm)
/subsystem=undertow/server=default-server/https-
listener=https:write-attribute(name=ssl-
context,value=LocalhostSslContext)
run-batch
```

7. Recargar el servidor.

```
recargar
```

Esto resulta en lo siguiente **elytron** configuración del subsistema en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:wildfly:elytron:1.2" ...>
  ...
  <tls>
    <key-stores>
      <key-store name="LocalhostKeyStore">
        <credential-reference clear-text="keystore_password"/>
        <implementation type="JKS"/>
        <file path="server.keystore" relative-
to="jboss.server.config.dir"/>
      </key-store>
      <key-store name="TrustStore">
        <credential-reference clear-text="truststore_password"/>
        <implementation type="JKS"/>
        <file path="server.truststore" relative-
to="jboss.server.config.dir"/>
      </key-store>
    </key-stores>
    <key-managers>
      <key-manager name="LocalhostKeyManager" key-
store="LocalhostKeyStore" alias-filter="server">
        <credential-reference clear-text="key_password"/>
      </key-manager>
    </key-managers>
  </tls>
</subsystem>
```

```
        </key-manager>
    </key-managers>
    <trust-managers>
        <trust-manager name="TrustManager" key-store="TrustStore"/>
    </trust-managers>
    <server-ssl-contexts>
        <server-ssl-context name="LocalhostSslContext" need-client-
auth="true" key-manager="LocalhostKeyManager" trust-
manager="TrustManager"/>
    </server-ssl-contexts>
</tls>
</subsystem>
```

Esto resulta en lo siguiente **undertow** configuración del subsistema en el archivo de configuración del servidor.

```
<https-listener name="https" socket-binding="https" ssl-
context="LocalhostSslContext" enable-http2="true"/>
```

Para más información, ver [Subsistema Elytron](#) y [Usando un cliente-ssl-context](#) en *Cómo configurar la seguridad del servidor* para JBoss EAP.

CAPÍTULO 8. MIGRACIÓN DESDE VERSIONES ANTERIORES DE JBOSS EAP [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

8.1. MIGRANDO DE JBOSS EAP 5 A JBOSS EAP 7 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Esta guía se centra en los cambios que se requieren para ejecutar e implementar con éxito aplicaciones JBoss EAP 6 en JBoss EAP 7. Si planea migrar sus aplicaciones directamente de JBoss EAP 5 a JBoss EAP 7, hay una serie de recursos disponibles para ayudarlo planifica y ejecuta tu migración. Sugerimos que tome el siguiente enfoque.

1. Ver [Resumen de los cambios realizados en cada versión](#) en esta guía para una descripción general rápida y de alto nivel de los cambios realizados en cada lanzamiento de JBoss EAP.
2. Lea a través de JBoss EAP 6 [Guía de migración](#) y esta guía para familiarizarse con los contenidos de cada uno.
3. Utilizar el [Referencia de actualización de componentes JBoss EAP 5](#) como una referencia rápida a la información de migración sobre componentes y características específicos.
4. Red Hat Application Migration Toolkit sigue agregando reglas para ayudarlo a migrar directamente de JBoss EAP 5 a JBoss EAP 7. Puede usar estas herramientas para analizar su aplicación y generar informes detallados sobre los cambios necesarios para migrar a JBoss EAP. 7. Para más información, ver [Utilice el Kit de herramientas de migración de aplicaciones de Red Hat para analizar aplicaciones para la migración](#).
5. los [Base de conocimiento del portal de clientes](#) actualmente contiene artículos y soluciones para ayudar con la migración de JBoss EAP 5 a JBoss EAP 6. Existen planes para agregar contenido adicional para la migración de JBoss EAP 5 a JBoss EAP 7 a lo largo del tiempo.

8.2. RESUMEN DE LOS CAMBIOS REALIZADOS EN CADA VERSIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Antes de planificar su migración, debe conocer los cambios que se hicieron a JBoss EAP 6 y JBoss EAP 7.

los [Guía de migración JBoss EAP 6](#) cubre los cambios que se hicieron entre JBoss EAP 5 y JBoss EAP 6. La siguiente es una lista resumida de los cambios más importantes realizados en JBoss EAP 6.

- Implementado una nueva arquitectura construida en el contenedor de servicio modular
- Fue una implementación certificada de la especificación Java Enterprise Edition 6
- Se introdujo la administración de dominio, la nueva configuración de implementación y una nueva estructura de directorios de archivos y scripts
- Estandarizado en nuevos espacios de nombres JNDI portátiles

Ver [Revise lo nuevo y lo diferente en JBoss EAP 6](#) en el JBoss EAP 6 [Guía de migración](#) para obtener una lista detallada de los cambios realizados en ese lanzamiento.

JBoss EAP 7 se basa en la misma estructura modular que JBoss EAP 6 e incluye la misma administración de dominio, configuración de implementación, estructura de directorios de archivos y scripts. También sigue utilizando los mismos espacios de nombres JNDI estandarizados. Sin embargo,

JBoss EAP 7 introduce los siguientes cambios.

- Agrega soporte para la especificación Java Enterprise Edition 7
- Reemplace el subsistema web con Undertow [Esto es una traducción automática]
- Reemplaza la implementación de JacORB IIOP con una rama descendente de OpenJDK ORB
- Incluye Apache ActiveMQ Artemis como el nuevo proveedor de mensajes
- Elimina el **cmp**, **jaxr**, y **threads** subsistemas
- Elimina el soporte para beans de entidad EJB

Para obtener una lista más completa de cambios, consulte [Revise las novedades de JBoss EAP 7](#)

8.3. REVISE EL CONTENIDO EN LAS GUÍAS DE MIGRACIÓN [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Revise todo el contenido de la Guía de migración para cada versión para conocer las características que se agregaron o se desaprobaron, y para comprender la configuración del servidor y los cambios de aplicación necesarios para ejecutar las aplicaciones existentes para esa versión.

Debido a que la arquitectura subyacente no se modificó entre JBoss EAP 6 y JBoss EAP 7, muchos de los cambios documentados en JBoss EAP 6 [Guía de migración](#) siguen aplicando. Por ejemplo, los cambios documentados bajo [Cambios requeridos por la mayoría de las aplicaciones](#) están relacionados con los cambios arquitectónicos subyacentes realizados en JBoss EAP 6, que todavía se aplican a este lanzamiento. El cambio al nuevo sistema de carga de clase modular es significativo e impacta en el empaque y las dependencias de casi todas las aplicaciones de JBoss EAP 5. Muchos de los cambios enumerados en [Cambios dependientes de la arquitectura y los componentes de su aplicación](#) también son todavía válidos Sin embargo, debido a que JBoss EAP 7 reemplazó el servidor web, ORB y proveedor de mensajes, eliminó el **cmp**, **threads**, y **jaxr** subsistemas, y soporte eliminado para beans de entidad EJB, debe consultar esta guía para cualquier cambio relacionado con esas áreas componentes. Presta especial atención a la [Cambios de configuración del servidor](#) y [Cambios en la migración de aplicaciones](#) detallado en esta guía antes de comenzar.

8.4. REFERENCIA DE ACTUALIZACIÓN DE COMPONENTES JBOSS EAP 5 [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Use la siguiente tabla para buscar información sobre cómo migrar una característica o componente particular de JBoss EAP 5 a JBoss EAP 7.1.

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
---	--

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Cambios en la migración de aplicaciones [Esto es una traducción automática]	<p>En JBoss EAP 6, la estructura de carga de la clase jerárquica anterior fue reemplazada por una arquitectura modular basada en JBoss Modules. El empaquetado de la aplicación también cambió debido a la nueva estructura modular de carga de clases. Esta arquitectura aún se usa en JBoss EAP 7. Para obtener información sobre la nueva arquitectura modular, consulte el siguiente capítulo en JBoss EAP 7.1 <i>Guía de desarrollo</i>.</p> <ul style="list-style-type: none"> • Carga de clase y módulos <p>Para obtener información sobre cómo actualizar y reempaquetar aplicaciones para la nueva arquitectura modular, consulte la siguiente sección en JBoss EAP 6. <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios de carga de clase
Cambios en la migración de aplicaciones [Esto es una traducción automática]	<p>Debido a los cambios en JBoss EAP 6 para utilizar la carga de clase modular, es posible que deba crear o modificar uno o más archivos de configuración de la aplicación para agregar dependencias o evitar la carga de dependencias automáticas. Esto no ha cambiado en JBoss EAP 7. Para más detalles, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios del archivo de configuración
Almacenamiento en caché e Infinispan	<p>JBoss Cache fue reemplazado por Infinispan para uso interno del servidor solo en JBoss EAP 6. Consulte las siguientes secciones en JBoss EAP 6 <i>Guía de migración</i> para obtener información sobre cómo reemplazar JBoss Cache en el código de la aplicación.</p> <ul style="list-style-type: none"> • Cambios de caché <p>La estrategia de caché de Infinispan y los cambios de configuración para JBoss EAP 7 están documentados en la siguiente sección de esta guía.</p> <ul style="list-style-type: none"> • Cambios en la configuración del servidor Infinispan
Configuración de un adaptador de recursos JMS genérico [Esto es una traducción automática]	<p>La configuración consolidada de JBoss EAP 6 de fuentes de datos y adaptadores de recursos en un solo archivo, y esto sigue siendo cierto en JBoss EAP 7. Consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i> para más información.</p> <ul style="list-style-type: none"> • Cambios en la configuración del adaptador de recursos y fuente de datos

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Migrar configuraciones de planes de implementación [Esto es una traducción automática]	<p>En JBoss EAP 6, la estructura del directorio, los scripts y la configuración de implementación cambiaron. Estos cambios todavía son válidos en JBoss EAP 7. Consulte la siguiente sección de JBoss EAP 6 <i>Guía de migración</i> para más información.</p> <ul style="list-style-type: none"> • Revise lo nuevo y lo diferente en JBoss EAP 6
EJB	<p>La especificación Java EE 7 hizo EJB 2.xy características anteriores opcionales, por lo que se recomienda encarecidamente reescribir el código de la aplicación para utilizar la especificación EJB 3.x y JPA. Para obtener información sobre las características en desuso y los cambios necesarios para ejecutar EJB 2.x, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • EJB 2.xy cambios anteriores <p>En JBoss EAP 6, el caché EJB con estado y el tamaño del grupo de beans de sesión sin estado se configuran en ejb3 subsistema del archivo de configuración del servidor. los jboss-ejb3.xml descriptor de despliegue reemplaza el jboss.xml archivo descriptor de despliegue Para obtener más información sobre estos cambios, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios de EJB <p>El puerto y el conector remoto predeterminados han cambiado en JBoss EAP 7. Para obtener más información sobre esto y sobre los cambios de configuración del servidor, consulte las siguientes secciones de esta guía.</p> <ul style="list-style-type: none"> • Cambios en la configuración del servidor EJB • Migrar código de cliente EJB <p>Los beans de entidad EJB no son compatibles con JBoss EAP 7. Para obtener información acerca de cómo migrar beans de entidad a JPA, consulte la siguiente sección de esta guía.</p> <ul style="list-style-type: none"> • Migrar beans de entidad a JPA

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Cambios de migración de Hibernate y JPA [Esto es una traducción automática]	<p>En JBoss EAP 6, Hibernate se actualizó de la versión 3 a la versión 4. Esta versión de JBoss EAP también implementó la especificación JPA 2.0 y se realizaron cambios en las propiedades de persistencia de JPA. Para obtener información sobre cómo modificar su aplicación para estos cambios, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios de Hibernate y JPA <p>JBoss EAP 7 implementa JPA 2.1 e incluye Hibernate 5. También actualiza Hibernate Search desde la versión 4.6.x a la versión 5.5.x. Otros cambios incluyen la eliminación de soporte para beans de entidad EJB y actualizaciones adicionales a las propiedades de persistencia JPA. Para obtener información sobre cómo estos cambios afectan sus aplicaciones, consulte las siguientes secciones en esta guía.</p> <ul style="list-style-type: none"> • Cambios de migración de Hibernate y JPA • Cambios en la búsqueda de Hibernate • Migrar beans de entidad a JPA • Cambios en la propiedad de persistencia JPA
Cambios de aplicación JAX-RS y RESTEasy [Esto es una traducción automática]	<p>JBoss EAP 6 incluyó RESTEasy 2, que configuró automáticamente RESTEasy y requirió cambios en la configuración de la aplicación. Consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i> para información.</p> <ul style="list-style-type: none"> • JAX-RS y RESTEasy Cambios sencillos <p>JBoss EAP 7 incluye RESTEasy 3 y muchas clases han quedado obsoletas. La versión de Jackson cambió de la versión 1.9.9 a la versión 2.6.3 o superior. Para obtener detalles sobre estos cambios, consulte la siguiente sección de esta guía.</p> <ul style="list-style-type: none"> • Cambios de aplicación JAX-RS y RESTEasy
JBoss AOP	<p>JBoss AOP (Programación Orientada a Aspectos) se eliminó en JBoss EAP 6. Para obtener información sobre cómo refactorizar aplicaciones que usan JBoss AOP, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios JBoss AOP

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Cambios de canales de JGroups [Esto es una traducción automática]	<p>La forma en que habilita el clúster y especifica las direcciones de enlace ha cambiado en JBoss EAP 6. Consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i> para más información.</p> <ul style="list-style-type: none"> • Cambios de agrupamiento <p>En JBoss EAP 7, JGroups ahora usa de manera predeterminada una interfaz de red privada en lugar de una interfaz de red pública y también presenta <channel> elementos a la jgroups subsistema. JBoss EAP 7 también incluye la implementación Undertow mod_cluster, presenta una nueva API para construir servicios singleton y otras nuevas características de clustering. Estos cambios están documentados en las siguientes secciones de esta guía.</p> <ul style="list-style-type: none"> • Cambios en la configuración del servidor JGroups • Cambios en la agrupación de aplicaciones
JNDI	<p>JBoss EAP 6 implementó un nuevo espacio de nombres JNDI global estandarizado y una serie de espacios de nombres relacionados que se asignan a los distintos ámbitos de una aplicación Java EE. Consulte la siguiente sección de JBoss EAP 6 <i>Guía de migración</i> para obtener información sobre los cambios en las aplicaciones necesarios para usar las nuevas reglas de espacio de nombres JNDI.</p> <ul style="list-style-type: none"> • Cambios JNDI
JSF	<p>JBoss EAP 6 incluye JSF 2.0 y le permitió configurar su aplicación para usar una versión anterior. Esto ya no es posible en JBoss EAP 7, que ahora incluye JSF 2.2. Consulte la siguiente sección en esta guía para obtener más información.</p> <ul style="list-style-type: none"> • Cambios de código de JavaServer Faces (JSF)
Cambios de registro [Esto es una traducción automática]	<p>JBoss EAP 6 introdujo un nuevo marco de registro de JBoss que todavía se utiliza en JBoss EAP 7. Las aplicaciones que usan marcos de registro de terceros pueden verse afectadas por los cambios de carga de clases modulares. Repase la siguiente sección en JBoss EAP 6 <i>Guía de migración</i> para obtener información sobre estos cambios.</p> <ul style="list-style-type: none"> • Cambios de registro <p>En JBoss EAP 7, anotaciones en el org.jboss.logging el paquete ahora está en desuso, lo que afecta el código fuente y los GAV de Maven (groupId: artifactId: version). Los prefijos para todos los mensajes de registro también se cambiaron. Para obtener más información sobre estos cambios, consulte las siguientes secciones en esta guía.</p> <ul style="list-style-type: none"> • Cambios en el registro de JBoss • Cambios en el prefijo del mensaje de registro

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Mensajería y JMS	<p>En JBoss EAP 6, HornetQ reemplazó a JBoss Messaging como la implementación JMS predeterminada. Luego, en JBoss EAP 7, ActiveMQ Artemis reemplazó a HornetQ como proveedor de mensajería incorporado.</p> <p>El mejor enfoque para migrar su configuración de mensajería es comenzar con la configuración del servidor por defecto de JBoss EAP 7 y utilizar la siguiente guía para aplicar sus cambios de configuración de mensajería actuales.</p> <ul style="list-style-type: none"> • Configurando la Mensajería para JBoss EAP 7.1 <p>Si desea comprender los cambios necesarios para pasar de JBoss Messaging a HornetQ, revise la siguiente sección de JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios HornetQ <p>Luego revise la siguiente información sobre cómo migrar la configuración de HornetQ y los datos de mensajes relacionados en esta guía.</p> <ul style="list-style-type: none"> • Cambios en la configuración del servidor de mensajería • Cambios en la aplicación de mensajería
ORB	<p>En JBoss EAP 6, la configuración de JacORB se movió de la <i>EAP_HOME/server/production/conf/jacorb.properties</i> archivo al archivo de configuración del servidor. JBoss EAP 7 luego reemplazó la implementación de JacORB IIOP con una rama de flujo descendente del OpenJDK ORB.</p> <p>El mejor enfoque para migrar su configuración de ORB es comenzar con la configuración del servidor por defecto de JBoss EAP 7 y usar la siguiente sección en JBoss EAP 7.1 <i>Guía de configuración</i> para aplicar los cambios de configuración actuales de su ORB.</p> <ul style="list-style-type: none"> • Configuración de ORB

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Invocación remota	<p>Se introdujo una nueva API de cliente EJB en JBoss EAP 6 para invocaciones remotas; sin embargo, si prefirió no reescribir el código de la aplicación para usar la nueva API, podría modificar su código existente para usar el <code>ejb:BEAN_REFERENCE</code> para acceso remoto a EJB. Consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i> para más información.</p> <ul style="list-style-type: none"> • Cambios de invocación remota <p>En JBoss EAP 7, el conector predeterminado y el puerto de conexión remota predeterminado cambiaron. Para obtener más información, consulte las siguientes secciones en esta guía.</p> <ul style="list-style-type: none"> • Actualice el puerto y el conector remoto de URL • Actualizar clientes externos • Migrar código de cliente EJB
Seam 2.x	<p>Mientras que el soporte oficial para las aplicaciones Seam 2.2 se eliminó en JBoss EAP 6, aún era posible configurar dependencias para JSF 1.2 e Hibernate 3 para permitir que las aplicaciones Seam 2.2 se ejecutaran en esa versión. JBoss EAP 7, que ahora incluye JSF 2.2 e Hibernate 5, no es compatible con Seam 2.2 o Seam 2.3 debido al final de la vida útil de Red Hat JBoss Web Framework Kit. Se recomienda que reescriba sus componentes Seam utilizando frijoles Weld CDI.</p>
Seguridad	<p>Las actualizaciones de seguridad en JBoss EAP 6 incluyeron cambios en los nombres de dominio de seguridad y cambios en cómo configurar la seguridad para la autenticación básica. La configuración del reino de seguridad LDAP se movió al archivo de configuración del servidor. Vea las siguientes secciones en JBoss EAP 6 <i>Guía de migración</i> para más información.</p> <ul style="list-style-type: none"> • Cambios de seguridad • Cambios en el reino de seguridad LDAP <p>Las actualizaciones que afectan la seguridad en JBoss EAP 7 incluyen cambios en la configuración del servidor y cambios en la aplicación. La información se puede encontrar en las siguientes secciones de esta guía.</p> <ul style="list-style-type: none"> • Cambios de configuración del servidor de seguridad • Cambios de aplicaciones de seguridad

JBoss EAP 5 Característica o Componente	Resumen de cambios y Dónde encontrar información de migración
Cambios de aplicaciones de seguridad [Esto es una traducción automática]	<p>Spring 4.2.x es la primera versión estable de Spring admitida por JBoss EAP 7. Para obtener información sobre los servicios web de Apache CXF Spring y los cambios de integración de Spring RESTEasy, consulte las siguientes secciones de esta guía.</p> <ul style="list-style-type: none"> • Cambios en los servicios web de Apache CXF Spring • Cambios en la integración RESTEasy de primavera
Transacciones	<p>JBoss EAP 6 consolidó la configuración de la transacción y la movió al archivo de configuración del servidor. Otras actualizaciones incluyeron cambios en la configuración del identificador de nodos JTA y cómo habilitar JTS. Para más detalles, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • JTS y JTA Cambios <p>Algunos atributos de configuración de Transaction Manager que estaban disponibles en transactions el subsistema en JBoss EAP 6 ha cambiado en JBoss EAP 7. Para obtener más información, consulte la siguiente sección de esta guía.</p> <ul style="list-style-type: none"> • Cambios en el subsistema de transacciones
Válvulas	<p>Undertow reemplazó a JBoss Web en JBoss EAP 7 y las válvulas ya no son compatibles. Vea las siguientes secciones en esta guía.</p> <ul style="list-style-type: none"> • Migrar válvulas globales • Migrar válvulas de aplicaciones personalizadas • Migrar válvulas de autenticador
Servicios de red	<p>JBoss EAP 6 incluye JBossWS 4. Para obtener información sobre los cambios requeridos por esa actualización de versión, consulte la siguiente sección en JBoss EAP 6 <i>Guía de migración</i>.</p> <ul style="list-style-type: none"> • Cambios en los servicios web <p>JBoss EAP 7 introdujo JBossWS 5. Consulte la siguiente sección de esta guía para conocer las actualizaciones requeridas.</p> <ul style="list-style-type: none"> • Cambios en las aplicaciones de servicios web

APÉNDICE A. MATERIAL DE REFERENCIA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

A.1. ADVERTENCIAS DE LA OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA JACORB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

los **migrate** la operación no puede procesar todos los recursos y atributos. La siguiente tabla enumera algunas de las advertencias que puede ver cuando ejecuta el **migrate** o **describe-migration** operación para el **jacorb** subsistema.



NOTA

Si ve las entradas "No se pudo migrar" o "No se puede migrar" en el resultado del **migrate** operación, esto indica que la migración de la configuración del servidor se completó con éxito, pero no fue capaz de migrar automáticamente todos los elementos y atributos. Debe seguir las sugerencias proporcionadas por las "advertencias de migración" para modificar esas configuraciones.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
los iiop la migración se puede realizar cuando el servidor está en admin-only modo	<p>los migrate operación requiere iniciar el servidor en admin-only modo, que se realiza mediante la adición --start-mode=admin-only al comando de inicio del servidor:</p> <pre>\$ EAP_HOME/bin/standalone.sh --start-mode=admin-only</pre>
Propiedades X no puede ser emulado usando OpenJDK ORB y no son compatibles	<p>La configuración de la propiedad especificada no es compatible y no está incluida en el nuevo iiop-openjdk configuración del subsistema El comportamiento exhibido por esta propiedad en la versión anterior de JBoss EAP no se migra y el administrador debe verificar que el nuevo iiop-openjdk El subsistema en JBoss EAP 7 puede funcionar correctamente sin ese comportamiento.</p> <p>Las propiedades no admitidas incluyen: cache-poa-names, cache-typecodes, chunk-custom-rmi-valuetypes, client-timeout, comet, indirection-encoding-disable, iona, lax-boolean-encoding, max-managed-buf-size, max-server-connections, max-threads, outbuf-cache-timeout, outbuf-size, queue-max, queue-min, poa-monitoring, print-version, retries, retry-interval, queue-wait, server-timeout, strict-check-on-tc-creation, use-bom, use-imr.</p>

Mensaje de advertencia	Qué significa / Cómo arreglarlo
Las propiedades <i>X</i> usar expresiones Las propiedades de configuración que se utilizan para resolver esas expresiones se deben transformar manualmente a la nueva iiop-openjdk formato de subsistema	<p>Las propiedades que usan expresiones deben ser configuradas manualmente por el administrador.</p> <p>Por ejemplo, el jacorb subsistema en JBoss EAP 6 define una giop-minor-version propiedad. los iiop-openjdk subsistema en JBoss EAP 7 define una giop-version propiedad. Supongamos que jacorb atributo de versión secundaria del subsistema está configurado para \${iiop-giop-minor-version} y la propiedad del sistema está configurada en standalone.conf archivo como -Diiop-giop-minor-version=1. Después de la migrate operación, el administrador debe cambiar el valor de propiedad del sistema a 1.1 para asegurar que el nuevo subsistema esté configurado correctamente.</p>
No se puede migrar: el nuevo iiop-openjdk subsistema ya está definido	El mensaje contiene la explicación.

A.2. ADVERTENCIAS DE LA OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA DE MENSAJERÍA [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

los **migrate** la operación no puede procesar todos los recursos y atributos. La siguiente tabla enumera algunas de las advertencias que puede ver cuando ejecuta el **migrate** o **describe-migration** operación para el **messaging** subsistema.



NOTA

Si ve las entradas "No se pudo migrar" o "No se puede migrar" en el resultado del **migrate** operación, esto indica que la migración de la configuración del servidor se completó con éxito, pero no fue capaz de migrar automáticamente todos los elementos y atributos. Debe seguir las sugerencias proporcionadas por las "advertencias de migración" para modificar esas configuraciones.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
los migrate operación no se puede realizar: el servidor debe estar en admin-only modo	<p>los migrate operación requiere iniciar el servidor en admin-only modo, que se realiza mediante la adición --start-mode=admin-only al comando de inicio del servidor:</p> <pre>\$ EAP_HOME/bin/standalone.sh --start-mode=admin-only</pre>

Mensaje de advertencia	Qué significa / Cómo arreglarlo
No se puede migrar el atributo local-bind-address del recurso <i>X</i> . Use en cambio el socket-binding atributo para configurar esto broadcast-group .	El mensaje contiene la explicación y cómo solucionarlo.
No se puede migrar el atributo local-bind-port del recurso <i>X</i> . Use en cambio el socket-binding atributo para configurar esto broadcast-group .	El mensaje contiene la explicación y cómo solucionarlo.
No se puede migrar el atributo group-address del recurso <i>X</i> . Use en cambio el socket-binding atributo para configurar esto broadcast-group .	El mensaje contiene la explicación y cómo solucionarlo.
No se puede migrar el atributo group-port del recurso <i>X</i> . Use en cambio el socket-binding atributo para configurar esto broadcast-group .	los broadcast-group recurso ya no acepta la local-bind-address , local-bind-port , group-address , o group-port atributos. Solo acepta una socket-binding atributo. La advertencia es notificación de ese recurso <i>X</i> tiene un atributo no compatible. Debe configurar manualmente el socket-binding atributo en el recurso y asegurar que corresponde a un definido socket-binding recurso.
Clases que proporcionan el <i>X</i> se descartan durante la migración. Para usarlos en el nuevo messaging-activemq subsistema, tendrá que ampliar la base de Artemisa Interceptor .	El soporte de interceptores de mensajería es significativamente diferente en JBoss EAP 7. Todos los interceptores configurados en la versión anterior del subsistema se descartan durante la migración. Ver Migrar interceptores de mensajería para más información.
No se puede migrar la configuración HA de <i>X</i> . Sus shared-store y backup atributos tiene expresiones y no es posible determinar inequívocamente cómo crear el correspondiente ha-policy para el servidor messaging-activemq.	Esto significa que hornetq-server XEs shared-store o backup los atributos contenían una expresión, como \$ {xxx}, y la operación de migración no pudo resolverlo en una expresión concreta. El valor se descarta y el ha-policy Para el messaging-activemq debe actualizarse manualmente
No se puede migrar el atributo local-bind-address del recurso <i>X</i> . Use en cambio el socket-binding atributo para configurar esto discovery-group .	El mensaje contiene la explicación y cómo solucionarlo.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
<p>No se puede migrar el atributo local-bind-port del recurso <i>X</i>. Use en cambio el socket-binding atributo para configurar esto discovery-group.</p>	<p>El mensaje contiene la explicación y cómo solucionarlo.</p>
<p>No se puede migrar el atributo group-address del recurso <i>X</i>. Use en cambio el socket-binding atributo para configurar esto discovery-group.</p>	<p>El mensaje contiene la explicación y cómo solucionarlo.</p>
<p>No se puede migrar el atributo group-port del recurso <i>X</i>. Use en cambio el socket-binding atributo para configurar esto discovery-group.</p>	<p>los discovery-group los recursos ya no aceptan local-bind-address, local-bind-port, group-address, o group-port atributos. Solo acepta una socket-binding. La advertencia es notificación de ese recurso <i>X</i> tiene un atributo no compatible. Debe configurar manualmente el socket-binding atributo en el recurso y asegura que corresponde a un definido socket-binding recurso.</p>
<p>No se puede crear un legacy-connection-factory Residencia en connection-factory <i>X</i>. Utiliza un HornetQ in-vm conector que no es compatible con Artemis in-vm conector</p>	<p>El legado HornetQ remoto connection-factory los recursos se migran a legacy-connection-factory recursos para permitir que los clientes de JBoss EAP 6 se conecten a JBoss EAP 7. Sin embargo, legacy-connection-factory los recursos solo se crean cuando connection-factory está usando conectores remotos. Alguna connection-factory utilizando in-vm no se migra porque in-vm los clientes se basan en JBoss EAP 7, no en JBoss EAP 6. Esta advertencia es una notificación de que el in-vm connection-factory no fue migrado</p>
<p>No se puede migrar el atributo <i>X</i> del recurso <i>Y</i>. El atributo usa una expresión que se puede resolver de manera diferente según las propiedades del sistema. Después de la migración, este atributo se debe volver a agregar con un valor real en lugar de la expresión.</p>	<p>Esta advertencia aparece cuando la migración no puede resolver el atributo <i>X</i> a un valor concreto durante el proceso de migración. El valor se descarta y el atributo debe migrarse manualmente. Esto sucede en los siguientes casos:</p> <ul style="list-style-type: none"> • cluster-connection forward-when-no-consumers: Este atributo booleano ha sido reemplazado por message-load-balancing-type atributo, que es una enumeración con un valor de OFF, STRICT, o ON_DEMAND. • broadcast-group y discovery-groupes jgroups-stack y jgroups-channel atributos Hacen referencia a otros recursos y JBoss EAP 7 ya no acepta estas expresiones.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
<p>No se puede migrar el atributo <i>X</i> del recurso <i>Y</i>. Este atributo no es compatible con el nuevo messaging-activemq subsistema.</p>	<p>Algunos atributos ya no se admiten en el nuevo messaging-activemq subsistema y simplemente se descartan:</p> <ul style="list-style-type: none"> • Ejemplo: @DateBridge y @CalendarBridge Anotación [Esto es una traducción automática] • Ejemplo: @DateBridge y @CalendarBridge Anotación [Esto es una traducción automática] • Ejemplo: @DateBridge y @CalendarBridge Anotación [Esto es una traducción automática] • remote-connectores use-nio atributo • remote-acceptores use-nio atributo
<p>No se puede migrar el atributo failback-delay del recurso <i>X</i>. Artemis detecta el failback deterministically y ya no requiere especificar un retraso para que se produzca failback.</p>	<p>El mensaje contiene la explicación.</p>

Reemplace los atributos del grupo de difusión obsoleto o del grupo de descubrimiento.

Si se le aconseja reemplazar el obsoleto **broadcast-group** o **discovery-group** atributos con el **socket-binding** atributo, puede agregar el nuevo atributo utilizando la CLI de administración.

Este ejemplo supone que está migrando un servidor independiente que contiene lo siguiente **discovery-group** configuración en el **messaging** subsistema.

```
<discovery-groups>
  <discovery-group name="my-discovery-group">
    <group-address>224.0.1.105</group-address>
    <group-port>56789</group-port>
  </discovery-group>
</discovery-groups>
```

Cuando ejecutas el **migrate** operación para el **messaging** subsistema, verá la siguiente salida y advertencias:

```
/subsystem=messaging:migrate
{
  "outcome" => "success",
  "result" => {"migration-warnings" => [
    "WFLYMSG0084: Can not migrate attribute group-address from
resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"discovery-group\\" => \\"my-discovery-group\\")
]. Use instead the socket-binding attribute to configure this discovery-
group.",
    "WFLYMSG0084: Can not migrate attribute group-port from resource [
  (\\"subsystem\\" => \\"messaging-activemq\\"),
  (\\"server\\" => \\"default\\"),
  (\\"discovery-group\\" => \\"my-discovery-group\\")
]. Use instead the socket-binding attribute to configure this discovery-
group."
  ]}
}
```

los **migrate** operación crea una **discovery-group** llamado "mi-descubrimiento-grupo" en el nuevo **messaging-activemq** subsistema que ahora está configurado como el siguiente.

```
<discovery-group name="my-discovery-group"/>
```

Ahora debe usar el siguiente comando CLI de administración para crear un **socket-binding** elemento en el archivo de configuración del servidor denominado "mi-descubrimiento-grupo-socket-binding".

```
/socket-binding-group=standard-sockets/socket-binding=my-discovery-group-
socket-binding:add(multicast-address=224.0.1.105, multicast-port=56789)
```

A continuación, agregue el recién creado **socket-binding** al **discovery-group** llamado "mi-descubrimiento-grupo" en el **messaging-activemq** subsistema en el archivo de configuración del servidor utilizando el siguiente comando de administración CLI.

```
/subsystem=messaging-activemq/server=default/discovery-group=my-discovery-
group:write-attribute(name=socket-binding,value=my-discovery-group-socket-
binding)
```

Estos comandos crean el siguiente XML en el archivo de configuración del servidor.

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:2.0">
  <server name="default">
```

```

...
<discovery-group name="my-discovery-group" socket-binding="my-
discovery-group-socket-binding"/>
...
</server>
</subsystem>
...
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="{jboss.socket.binding.port-offset:0}">
...
<socket-binding name="my-discovery-group-socket-binding" multicast-
address="224.0.1.105" multicast-port="56789"/>
...
</socket-binding-group>

```

A.3. ADVERTENCIAS DE OPERACIÓN DE MIGRACIÓN DEL SUBSISTEMA WEB [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

los **migrate** la operación no puede procesar todos los recursos y atributos. La siguiente tabla enumera algunas de las advertencias que puede ver cuando ejecuta el **migrate** o **describe-migration** operación para el **web** subsistema.



NOTA

Si ve las entradas "No se pudo migrar" o "No se puede migrar" en el resultado del **migrate** operación, esto indica que la migración de la configuración del servidor se completó con éxito, pero no fue capaz de migrar automáticamente todos los elementos y atributos. Debe seguir las sugerencias proporcionadas por las "advertencias de migración" para modificar esas configuraciones.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
La operación de migración solo está permitida en modo solo administrador	<p>los migrate operación requiere iniciar el servidor en admin-only modo, que se hace agregando el parámetro - admin-only al comando de inicio del servidor:</p> <pre>\$ EAP_HOME/bin/standalone.sh --admin-only</pre>
No se pudo migrar el recurso X	<p>El comportamiento exhibido por este recurso en la versión anterior de JBoss EAP no se migró. El administrador debe verificar si el nuevo undertow el subsistema en JBoss EAP 7 puede funcionar correctamente sin ese comportamiento o si el comportamiento debe migrarse manualmente.</p>

Mensaje de advertencia	Qué significa / Cómo arreglarlo
No se pudo migrar el atributo <i>X</i> del recurso <i>Y</i> .	<p>El comportamiento exhibido por este atributo de recurso en la versión anterior de JBoss EAP no se migró. El administrador debe verificar si el nuevo undertow el subsistema en JBoss EAP 7 puede funcionar correctamente sin ese comportamiento o si el comportamiento debe migrarse manualmente.</p> <p>Ver Advertencias de atributos de operación de migración del subsistema web para la lista de atributos que no se migran.</p>
No se pudo migrar el conector SSL ya que no se definió ninguna configuración SSL	El mensaje contiene la explicación.
No se pudo migrar verify-client atributo <i>X</i> al equivalente de Undertow	El mensaje contiene la explicación.
No se pudo migrar verify-client expresión <i>X</i>	El mensaje contiene la explicación.

Mensaje de advertencia	Qué significa / Cómo arreglarlo
<p>No se pudo migrar la válvula <i>X</i></p>	<p>El comportamiento exhibido por esta válvula en la versión anterior de JBoss EAP no se migró. El administrador debe verificar si el nuevo undertow el subsistema en JBoss EAP 7 puede funcionar correctamente sin ese comportamiento o si el comportamiento debe migrarse manualmente.</p> <p>Esta advertencia puede ocurrir para las siguientes válvulas:</p> <ul style="list-style-type: none"> • org.apache.catalina.valves.RemoteAddrValve Debe tener al menos un valor permitido o denegado. • org.apache.catalina.valves.RemoteHostValve Debe tener al menos un valor permitido o denegado. • org.apache.catalina.authenticator.BasicAuthenticator • org.apache.catalina.authenticator.DigestAuthenticator • org.apache.catalina.authenticator.FormAuthenticator • org.apache.catalina.authenticator.SSLAuthenticator • org.apache.catalina.authenticator.SpnegoAuthenticator • válvulas personalizadas

Mensaje de advertencia	Qué significa / Cómo arreglarlo
No se pudo migrar el atributo <i>X</i> de la válvula <i>Y</i>	<p>El comportamiento exhibido por este atributo de válvula en la versión anterior de JBoss EAP no se migró. El administrador debe verificar si el nuevo undertow el subsistema en JBoss EAP 7 puede funcionar correctamente sin ese comportamiento o si el comportamiento debe migrarse manualmente. Esta advertencia puede ocurrir para los siguientes atributos de válvula:</p> <ul style="list-style-type: none"> • org.apache.catalina.valves.AccessLogValve <ul style="list-style-type: none"> ◦ resolveHosts ◦ fileDateFormat ◦ renameOnRotate ◦ encoding ◦ locale ◦ requestAttributesEnabled ◦ buffered • org.apache.catalina.valves.ExtendedAccessLogValve <ul style="list-style-type: none"> ◦ resolveHosts ◦ fileDateFormat ◦ renameOnRotate ◦ encoding ◦ locale ◦ requestAttributesEnabled ◦ buffered • org.apache.catalina.valves.RemoteIpValve <ul style="list-style-type: none"> ◦ httpServerPort ◦ httpsServerPort ◦ remoteIpHeader Si está definido pero no configurado como "x-reenviado-para" ◦ protocolHeader Si está definido pero no configurado como "x-forward-proto"

Advertencias de operación de migración del subsistema web [Esto es una traducción automática]

los **migrate** la operación no puede procesar todos los atributos de JBoss Web. Consulte las siguientes tablas de referencia para obtener información acerca de cómo migrar manualmente los atributos no procesados.

Atributos del conector web SSL

Los siguientes atributos se usaron en JBoss EAP 6 para configurar el conector SSL. Las bibliotecas nativas de OpenSSL no son compatibles con JBoss EAP 7 por lo que no hay configuraciones equivalentes.

Atributos para eliminar [Esto es una traducción automática]	Descripción	Aumentar equivalente
ca-revocation-url	El archivo o URL que contiene la lista de revocación.	Sin equivalente en Undertow.
certificate-file	Al usar el cifrado OpenSSL, la ruta al archivo que contiene el certificado del servidor.	Sin equivalente en Undertow.
ssl-protocol	La cadena de protocolo SSL.	Sin equivalente en Undertow.
verify-depth	El número máximo de emisores de certificados intermedios revisados antes de decidir que los clientes no tienen un certificado válido.	Sin equivalente en Undertow.

Atributos de recursos estáticos web

El seguimiento **static-resources** los atributos del elemento se utilizaron para describir cómo los recursos estáticos fueron manejados por el **DefaultServlet** o por el **WebdavServlet**. No hay equivalentes para estos atributos porque WebDAV no es compatible con Undertow. Para más información, ver <https://issues.jboss.org/browse/JBEAP-1036>.

Atributos para eliminar [Esto es una traducción automática]	Descripción	Aumentar equivalente
discapacitado	Habilite la asignación de servlets predeterminada.	Sin configuración equivalente en Undertow.
codificación de archivos	Codificación de archivos para usar al leer archivos estáticos.	Sin configuración equivalente en Undertow.
máxima profundidad	Recursión máxima para PROPFIND .	Esta es una configuración de WebDAV y WebDAV no es compatible con Undertow.

Atributos para eliminar [Esto es una traducción automática]	Descripción	Aumentar equivalente
solo lectura	Permitir escribir métodos HTTP (PUT, DELETE).	Esta es una configuración de WebDAV y WebDAV no es compatible con Undertow.
secreto	Secreto para las operaciones de bloqueo WebDAV.	Esta es una configuración de WebDAV y WebDAV no es compatible con Undertow.
enviar archivo	Habilite sendfile si es posible, para archivos más grandes que el tamaño de byte especificado.	Esto se establece en un valor predeterminado razonable en Undertow y no es configurable.
webdav	Habilita la funcionalidad de WebDAV.	WebDAV no es compatible con Undertow.

Atributos de recursos de SSO web

SSO se maneja de manera diferente que en la versión anterior y no hay configuraciones de atributos equivalentes en JBoss EAP 7.

JBoss Web Attribute	Descripción	Aumentar equivalente
cache-container	Nombre del contenedor de caché a usar para SSO agrupado.	Esta configuración ya no es necesaria en Undertow. Esto funciona de manera predeterminada en un entorno distribuido agrupado.
cache-name	Nombre de la memoria caché a usar para SSO agrupado.	Esta configuración ya no es necesaria en Undertow. Esto funciona de manera predeterminada en un entorno distribuido agrupado.
Migrar válvulas de autenticador [Esto es una traducción automática]	Si cada solicitud debería provocar una reautenticación.	No hay una configuración equivalente en Undertow, que se comporta de forma similar a la reauthenticate=true configuración en JBoss EAP 6. Mientras reauthenticate=false posiblemente podría mejorar el rendimiento, también podría crear problemas de seguridad.

Asignación de atributos de mensajería [Esto es una traducción automática]

JBoss Web Attribute	Descripción	Aumentar equivalente
resolver-hosts	Si se debe habilitar la resolución de hosts para el acceso de registro.	Use la configuración en el conector para lograr el mismo comportamiento.

Atributos del conector web

JBoss Web Attribute	Descripción	Aumentar equivalente
ejecutor	El nombre del ejecutor que debe usarse para procesar los hilos de este conector.	Ejemplo: Propiedades de seguridad definidas en security Subsistema [Esto es una traducción automática] Ver Migrar la configuración del subsistema de subprocesos para más información.
enlace proxy	El enlace del socket para definir el host y el puerto que se utiliza al enviar un redireccionamiento.	No hay un equivalente directo. Ver https-listener Atributos en el JBoss EAP <i>Guía de configuración</i> para las opciones de configuración disponibles.
redirect-port	El puerto para la redirección a un conector seguro.	Este atributo fue desaprobado en JBoss EAP 6 y reemplazado por redirect-binding . Undertow proporciona el redirect-socket atributo en el http-listener elemento, que es un reemplazo para redirect-binding . Ver https-listener Atributos en el JBoss EAP <i>Guía de configuración</i> para más información.

A.4. MIGRAR LA REFERENCIA DE PROPIEDADES DEL SISTEMA WEB JBOSS [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Esta referencia describe cómo mapear las propiedades del sistema previamente utilizadas para la configuración web de JBoss a la configuración equivalente para Undertow en JBoss EAP 7.

- [Propiedades del sistema de contenedor y conectores de servlets de mapa](#)
- [Mapa EL System Properties](#)
- [Propiedades del sistema JSP del mapa](#)
- [Propiedades de Map Security System](#)

Tabla A.1. Propiedades del sistema de contenedor y conectores de servlets de mapa [Esto es una traducción automática]

Propiedad del sistema JBoss EAP 6	Descripción
	Equivalent in JBoss EAP 7
jvmRoute	Proporciona un valor predeterminado para el jvmRoute atributo. No anula el valor generado automáticamente al usar el standalone-ha.xml archivo de configuración. Es compatible reload .
	Gestión de la migración de la CLI Operación [Esto es una traducción automática] <pre>/subsystem=undertow:write-attribute(name=instance-id,value=VALUE)</pre>
org.apache.tomcat.util.buf.StringCache.byte.enabled	Si true , la caché de cadenas está habilitada ByteChunk . Si el valor no está especificado, el valor predeterminado de false es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]
org.apache.tomcat.util.buf.StringCache.char.enabled	Si true , la caché de cadenas está habilitada CharChunk . Si el valor no está especificado, el valor predeterminado de false es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]
org.apache.tomcat.util.buf.StringCache.cacheSize	El tamaño de la caché de cadenas. Si el valor no está especificado, el valor predeterminado de 5000 es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]
org.apache.tomcat.util.buf.StringCache.maxStringSize	La longitud máxima de String que se almacenará en caché. Si el valor no está especificado, el valor predeterminado de 128 es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]
org.apache.tomcat.util.http.FastDateFormat.CACHE_SIZE	El tamaño de la memoria caché para usar el valor de fecha analizado y formateado. Si el valor no está especificado, el valor predeterminado de 1000 es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]

<p>org.apache.catalina.core.StandardService.DELAY_CONNECTOR_STARTUP</p>	<p>Si true, el inicio del conector no se realiza automáticamente. Es útil en modo incrustado.</p>
	<p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>
<p>org.apache.catalina.connector.Request.SSESSION_ID_CHECK</p>	<p>Si true, el contenedor Servlet verifica que exista una sesión en un contexto con la ID de sesión especificada antes de crear una sesión con esa ID.</p>
	<p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>
<p>org.apache.coyote.USE_CUSTOM_STAT US_MSG_IN_HEADER</p>	<p>Si true, los mensajes de estado HTTP personalizados se usan dentro de encabezados HTTP. Los usuarios deben asegurarse de que dicho mensaje sea ISO-8859-1 codificado, especialmente si la entrada proporcionada por el usuario se incluye en el mensaje, para evitar una posible vulnerabilidad XSS. Si no se especifica el valor, el valor predeterminado de false es usado.</p>
<p>org.apache.tomcat.util.http.Parameters.M AX_COUNT</p>	<p>Debe ser habilitado programáticamente mediante la implementación de una costumbre io.undertow.servlet.ServletExtension. Luego usa la extensión para llamar setSendCustomReasonPhraseOnError(true) sobre el io.undertow.servlet.api.DeploymentInfo instancia de estructura</p> <p>La cantidad máxima de parámetros que se pueden analizar en un cuerpo de publicación. Si se excede, el análisis falla al usar un IllegalStateException. El valor predeterminado es 512 parámetros.</p> <p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre> /subsystem=undertow/server=default- server/http-listener=default:write- attribute(name=max- parameters,value=VALUE) /subsystem=undertow/server=default- server/https-listener=default:write- attribute(name=max- parameters,value=VALUE) /subsystem=undertow/server=default- server/ajp-listener=default:write- attribute(name=max- parameters,value=VALUE) </pre>

org.apache.tomcat.util.http.MimeHeaders. MAX_COUNT	<p>La cantidad máxima de encabezados que pueden enviarse en la solicitud HTTP. Si se excede, el análisis fallará usando un IllegalStateException. El valor predeterminado es 128 encabezados</p> <p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre>/subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=max-headers,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=max-headers,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=max-headers,value=VALUE)</pre>
org.apache.tomcat.util.net.MAX_THREADS	<p>La cantidad máxima de subprocesos que un conector va a usar para procesar las solicitudes. El valor predeterminado es 32 X 512. (512 X Runtime.getRuntime().availableProcessors() Para el JIO conector)</p> <p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre>/subsystem=io/worker=default:write-attribute(name=task-max-threads,value=VALUE)</pre>
org.apache.coyote.http11.Http11Protocol. MAX_HEADER_SIZE	<p>El tamaño máximo de los encabezados HTTP, en bytes. Si se excede, el análisis fallará usando un ArrayOutOfBoundsException. El valor predeterminado es 8192 bytes.</p> <p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre>/subsystem=undertow/server=default-server/http-listener=default:write-attribute(name=max-header-size,value=VALUE) /subsystem=undertow/server=default-server/https-listener=default:write-attribute(name=max-header-size,value=VALUE) /subsystem=undertow/server=default-server/ajp-listener=default:write-attribute(name=max-header-size,value=VALUE)</pre>

<p>org.apache.coyote.http11.Http11Protocol. COMPRESSION</p>	<p>Permite usar compresión simple con el conector HTTP. El valor predeterminado es off, y la compresión se puede habilitar utilizando el valor on habilitarlo condicionalmente, o forzarlo a habilitarlo siempre.</p> <p>Configure un filtro utilizando la CLI de administración:</p> <pre># Create a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add()</pre>
<p>org.apache.coyote.http11.Http11Protocol. COMPRESSION_RESTRICTED_UA</p>	<p>Regexps de agentes de usuario que no recibirán contenido comprimido. El valor predeterminado es vacío.</p> <p>Configure un predicado en un filtro usando la CLI de administración:</p> <pre># Use a predicate in a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add(predicate="regex[patte rn='AppleWebKit',value=%{i,User-Agent}]")</pre>
<p>org.apache.coyote.http11.Http11Protocol. COMPRESSION_MIME_TYPES</p>	<p>Prefijos de tipo de contenido de contenido comprimible. El valor predeterminado es text/html, text/xml, text/plain.</p> <p>Configure un predicado en un filtro usando la CLI de administración:</p> <pre># Use a predicate in a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add(predicate="regex[patte rn='text/html',value=%{o,Content-Type}]")</pre>
<p>org.apache.coyote.http11.Http11Protocol. COMPRESSION_MIN_SIZE</p>	<p>Tamaño mínimo de contenido que se comprimirá. El valor predeterminado es 2048 bytes.</p>

	<p>Configure un predicado en un filtro usando la CLI de administración:</p> <pre># Use a predicate in a filter /subsystem=undertow/configuration=filter/ gzip=gzipfilter:add() /subsystem=undertow/server=default- server/host=default-host/filter- ref=gzipfilter:add(predicate="max- content-size[value=MIN_SIZE]")</pre>
org.apache.coyote.http11.DEFAULT_CONNECTION_TIMEOUT	<p>Tiempo de espera predeterminado del socket. El valor predeterminado es 60000 Sra.</p>
	<p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre>/subsystem=undertow/server=default- server/http-listener=default:write- attribute(name=no-request- timeout,value=VALUE) /subsystem=undertow/server=default- server/https-listener=default:write- attribute(name=no-request- timeout,value=VALUE) /subsystem=undertow/server=default- server/ajp-listener=default:write- attribute(name=no-request- timeout,value=VALUE)</pre>
org.jboss.as.web.deployment.DELETE_WORK_DIR_ONCONTEXTDESTROY	<p>Usa esta propiedad para eliminar .java y .class archivos para garantizar que las fuentes JSP se vuelvan a compilar. El valor predeterminado es false. Tiempo de espera del socket predeterminado para keep-alive. El valor predeterminado es -1 ms, lo que significa que usará el tiempo de espera predeterminado del socket.</p>
	<p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>
org.apache.tomcat.util.buf.StringCache.trimThreshold	<p>Especifica la cantidad de veces toString() debe invocarse antes de activar el caché. El valor predeterminado es 100000.</p>
	<p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>

Tabla A.2. Mapa EL System Properties [Esto es una traducción automática]

Propiedad del sistema JBoss EAP 6	Descripción
-----------------------------------	-------------

	Equivalent in JBoss EAP 7
org.apache.el.parser.COERCE_TO_ZERO	Si true , cuando se fuerzan expresiones a números, las cadenas vacías ("") y nulo serán forzadas a cero según lo requiera la especificación. Si no se especifica un valor, el valor predeterminado de true es usado.
	La propiedad del sistema aún es válida y procesada por EL

Tabla A.3. Propiedades del sistema JSP del mapa [Esto es una traducción automática]

Propiedad del sistema JBoss EAP 6	Descripción
	Equivalent in JBoss EAP 7
org.apache.jasper.compilerGenerator.VARIABLE_EXPRESSION_FACTORY	El nombre de la variable que se usará para la expresión expresión de idioma fábrica. Si no se especifica el valor, el valor predeterminado de _el_expressionfactory es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.compilerGenerator.VARIABLE_INSTANCE_MANAGER	El nombre de la variable que se usará para la fábrica del administrador de instancias. Si no se especifica el valor, el valor predeterminado de _jsp_instancemanager es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.compiler.Parser.STRICT_QUOTE_ESCAPING	Si false , los requisitos para escalar las cotizaciones en los atributos JSP se relajan de modo que una cotización requerida faltante no cause un error. Si no se especifica el valor, el valor predeterminado de la especificación de true es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE	Cualquier buffer de etiquetas que se expanda más allá org.apache.jasper.Constants.DEFAULT_TAG_BUFFER_SIZE se destruye y se crea un nuevo buffer del tamaño predeterminado. Si no se especifica el valor, el valor predeterminado de 512 es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.runtime.JspFactoryImpl.USE_POOL	Si true , se utiliza un grupo de elementos de página de ThreadLocal. Si no se especifica el valor, el valor predeterminado de true es usado.
	La propiedad del sistema no ha cambiado

org.apache.jasper.runtime.JspFactoryImpl. POOL_SIZE	El tamaño del ThreadLocal PageContext. Si no se especifica el valor, el valor predeterminado de 8 es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.JSP_SERVLET_BASE	La clase base de los Servlets generados a partir de los JSP. Si no se especifica el valor, el valor predeterminado de org.apache.jasper.runtime.HttpJspBase es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.SERVICE_METHOD_NAME	El nombre del método de servicio llamado por la clase base. Si no se especifica el valor, el valor predeterminado de _jspService es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.SERVLET_CLASSPATH	El nombre del atributo ServletContext que proporciona la ruta de clase para el JSP. Si no se especifica el valor, el valor predeterminado de org.apache.catalina.jsp_classpath es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.JSP_FILE	El nombre del atributo de solicitud para <jsp-file> elemento de una definición de servlet. Si está presente en una solicitud, esto anula el valor devuelto por request.getServletPath() para seleccionar la página JSP que se ejecutará. Si no se especifica el valor, el valor predeterminado de org.apache.catalina.jsp_file es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.PRECOMPILE	El nombre del parámetro de consulta que hace que el motor JSP simplemente pregenere el servlet pero no lo invoque. Si no se especifica el valor, el valor predeterminado de org.apache.catalina.jsp_precompile es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.JSP_PACKAGE_NAME	El nombre del paquete predeterminado para las páginas compiladas JSP. Si el valor no se especifica, el valor predeterminado de org.apache.jsp es usado.
	La propiedad del sistema no ha cambiado

org.apache.jasper.Constants.TAG_FILE_PACKAGE_NAME	El nombre del paquete predeterminado para los manejadores de etiquetas generados a partir de archivos de etiquetas. Si no se especifica el valor, el valor predeterminado de org.apache.jsp.tag es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.TEMP_VARIABLE_NAME_PREFIX	Prefijo para usar para nombres de variables temporales generados. Si no se especifica el valor, el valor predeterminado de _jspx_temp es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.USE_INSTANCE_MANAGER_FOR_TAGS	Si true , el administrador de instancias se usa para obtener instancias de controladores de etiquetas. Si el valor no está especificado, true es usado.
	La propiedad del sistema no ha cambiado
org.apache.jasper.Constants.INJECT_TAGS	Si true , las anotaciones especificadas en las etiquetas se procesarán e inyectarán. Esto puede tener un impacto en el rendimiento al usar etiquetas simples, o si la agrupación de etiquetas está deshabilitada. Si el valor no está especificado, false es usado.
	La propiedad del sistema no ha cambiado

Tabla A.4. Propiedades de Map Security System [Esto es una traducción automática]

Propiedad del sistema JBoss EAP 6	Descripción
	Equivalent in JBoss EAP 7
org.apache.catalina.connector.RECYCLE_FACADES	Si esto es true o si un administrador de seguridad está en uso, se crea un nuevo objeto de fachada para cada solicitud. Si no se especifica el valor, el valor predeterminado de false es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]
org.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH	Si esto es true el carácter '\' está permitido como un delimitador de ruta. Si no se especifica el valor, el valor predeterminado de false es usado.
	Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]

org.apache.tomcat.util.buf.UDecoder.ALL OW_ENCODED_SLASH	<p>Si esto es true, '%2F' y '%5C' está permitido como delimitadores de ruta. Si no se especifica el valor, el valor predeterminado de false es usado.</p> <p>Gestión de la migración de la CLI Operación [Esto es una traducción automática]</p> <pre> /subsystem=undertow/server=default- server/http-listener=default:write- attribute(name=allow-encoded- slash,value=VALUE) /subsystem=undertow/server=default- server/https-listener=default:write- attribute(name=allow-encoded- slash,value=VALUE) /subsystem=undertow/server=default- server/ajp-listener=default:write- attribute(name=allow-encoded- slash,value=VALUE) </pre>
org.apache.catalina.STRICT_SERVLET_ COMPLIANCE	<p>Si el valor no está especificado, true es usado. Si esto es true se producirán las siguientes acciones: se verifica cualquier solicitud envuelta u objeto de respuesta que se transfiera a un despachador de aplicaciones para garantizar que haya ajustado la solicitud o respuesta original. (SRV.8.2 / SRV.14.2.5.1) una llamada a Response.getWriter() si no se ha especificado ninguna codificación de caracteres, se generarán llamadas posteriores a Response.getCharacterEncoding() regresando ISO-8859-1 y el Content-Type El encabezado de respuesta incluirá un charset=ISO-8859-1 componente. (SRV.15.2.22.1) cada solicitud asociada a una sesión hace que se actualice la última hora accedida de la sesión, independientemente de si la expiración de la solicitud accede o no a la sesión. (SRV.7.6)</p> <p>Cumple por defecto</p>
org.apache.catalina.core.StandardWrapperValve.SERVLET_STATS	<p>Si true o si org.apache.catalina.STRICT_SERVLET_COMPLIANCE es true, el contenedor recogerá las estadísticas JSR-77 para servlets individuales. Si no se especifica el valor, el valor predeterminado de false es usado.</p> <p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>

<p>org.apache.catalina.session.StandardSession.ACTIVITY_CHECK</p>	<p>Si esto es true o si org.apache.catalina.STRICT_SERVLET_COMPLIANCE es true Tomcat rastrea el número de solicitudes activas para cada sesión. Al determinar si una sesión es válida, cualquier sesión con al menos una solicitud activa siempre se considerará válida. Si no se especifica el valor, el valor predeterminado de false es usado.</p> <p>Reemplazar la configuración del servlet de Netty [Esto es una traducción automática]</p>
---	---

A.5. COMPATIBILIDAD E INTEROPERABILIDAD ENTRE VERSIONES [ESTO ES UNA TRADUCCIÓN AUTOMÁTICA]

Esta sección describe la compatibilidad e interoperabilidad de los componentes EJB y de mensajería entre las versiones JBoss EAP 5, JBoss EAP 6 y JBoss EAP 7.

EJB remota sobre IIOP

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

EJB remoto Uso de JNDI

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

JBoss EAP 6 brindó soporte para la especificación EJB 3.1 e introdujo el uso de espacios de nombres JNDI globales estandarizados, que aún se utilizan en JBoss EAP 7. Debido al cambio en los nombres de nombres de nombres JNDI, las siguientes configuraciones no son compatibles:

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Conexión desde un cliente JBoss EAP 7 o JBoss EAP 6 a un servidor JBoss EAP 5

Para obtener más información acerca de los cambios de espacio de nombres JNDI estandarizados, consulte [Cambios JNDI](#) en el JBoss EAP 6 *Guía de migración*.

Interacción de EJB utilizando @WebService

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

Cliente autónomo de mensajería

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

En la siguiente configuración, si el cliente está utilizando la API HornetQ específica del agente de mensajería en lugar de la API JMS genérica, la conexión es posible. Sin embargo, las búsquedas JNDI deben abordarse utilizando la extensión de nombres JNDI legacy que se entrega con JBoss EAP 7.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

La mensajería incorporada de JBoss EAP 7 no puede conectarse a HornetQ 2.2.x que se envió con JBoss EAP 5 debido a problemas de compatibilidad de protocolo. Por esta razón, las siguientes configuraciones no son compatibles.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

Migrar datos de mensajería [Esto es una traducción automática]

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

En la siguiente configuración, si el cliente está utilizando la API HornetQ específica del agente de mensajería en lugar de la API JMS genérica, la conexión es posible. Sin embargo, las búsquedas JNDI deben abordarse utilizando la extensión de nombres JNDI legacy que se entrega con JBoss EAP 7.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

La mensajería incorporada de JBoss EAP 7 no puede conectarse a HornetQ 2.2.x que se envió con JBoss EAP 5 debido a problemas de compatibilidad de protocolo. Por esta razón, las siguientes configuraciones no son compatibles.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

Puentes JMS

No debería encontrar problemas con ninguna de las siguientes configuraciones.

- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]
- Migrando de JBoss EAP 5 a JBoss EAP 7 [Esto es una traducción automática]

Revised on 2018-07-26 09:16:02 EDT