



Red Hat JBoss Enterprise Application Platform 7.1

Introducción a JBoss EAP

Para usar con Red Hat JBoss Enterprise Application Platform 7.1

Red Hat JBoss Enterprise Application Platform 7.1 Introducción a JBoss EAP

Para usar con Red Hat JBoss Enterprise Application Platform 7.1

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

En este documento se proporciona una descripción conceptual de alto nivel de Red Hat JBoss Enterprise Application Platform, así como varios temas en los que se basa JBoss EAP.

Table of Contents

CAPÍTULO 1. DESCRIPCIÓN DE CONCEPTOS GENERALES	3
1.1. JAVA	3
1.2. SERVIDORES DE APLICACIONES	3
1.3. JAVA EE 7	3
CAPÍTULO 2. DESCRIPCIÓN GENERAL DE JBOSS EAP	4
2.1. ACERCA DE JBOSS EAP 7	4
2.2. SUBSISTEMAS	5
2.3. ALTA DISPONIBILIDAD	5
2.4. MODOS OPERATIVOS	5
CAPÍTULO 3. EJEMPLOS	6
3.1. EJEMPLO SIMPLE	6
3.2. EJEMPLO AMPLIADO	6

CAPÍTULO 1. DESCRIPCIÓN DE CONCEPTOS GENERALES

Antes de comprender cómo se puede configurar e implementar Red Hat JBoss Enterprise Application Platform, debe comprender algunos conceptos importantes.

1.1. JAVA

Java es un lenguaje de programación y una plataforma de computación que incorpora conceptos como orientación de objetos, clases y concurrencia. Las aplicaciones de Java se recopilan en bytecode y se ejecutan dentro de una máquina virtual Java (JVM).

1.2. SERVIDORES DE APLICACIONES

Un servidor de aplicaciones es un software que proporciona un entorno para ejecutar aplicaciones web. La mayoría de los servidores de aplicaciones también proporcionan funcionalidad a aplicaciones web que se ejecutan en su entorno a través de un conjunto de API. Por ejemplo, un servidor de aplicaciones puede brindar una API para conectarse a una base de datos.

1.3. JAVA EE 7

Java EE (Plataforma Java, Edición empresarial) es una plataforma empresarial basada en estándares que ofrece un entorno de API y de tiempo de ejecución para ejecutar y desarrollar aplicaciones Java. La meta es mejorar la productividad de los desarrolladores mediante capacidades empresariales integrales en marcos de fácil consumo que eliminan el texto reutilizable y reducen la carga técnica. Los marcos que componen Java EE pasaron por pruebas estrictas en combinación. Java EE 7, basado en [JSR 342](#) es una extensión de Java EE 6, con el foco principal en la simplificación adicional de API para acceder a servicios de contenedores, al tiempo que se incrementa el soporte de tecnologías web emergentes, como HTML5.

Java EE 7 incluye soporte para varios perfiles o subconjuntos de API. La especificación de Java EE 7 define el perfil *full* y el perfil *web*.

Perfil web de Java EE 7

El perfil web de Java EE 7 está diseñado para el desarrollo de aplicaciones web y respalda un subconjunto de API definidas por tecnologías basadas en la Web relacionadas con Java EE 7.

Perfil completo de Java EE 7

El perfil completo de Java EE 7 contiene todas las API definidas por Java EE 7 (incluidos todos los elementos en el perfil web). Al desarrollar EJB, las aplicaciones de mensajería y los servicios web usan el perfil completo (en contraste con las aplicaciones web).

CAPÍTULO 2. DESCRIPCIÓN GENERAL DE JBOSS EAP

2.1. ACERCA DE JBOSS EAP 7

Red Hat JBoss Enterprise Application Platform 7.1 (JBoss EAP 7) es una implementación certificada de las especificaciones completas y del perfil web de Java Enterprise Edition 7 (Java EE 7).

Las principales versiones de JBoss EAP se desarrollan a partir del proyecto comunitario [WildFly](#) en ciertos puntos, cuando el proyecto comunitario ha alcanzado el nivel deseado de terminación de funciones. A partir de aquí, continúa un período extendido de pruebas y puesta en producción, en el que JBoss EAP se estabiliza, se certifica y se mejora para su uso en producción. Durante el ciclo de vida de una versión principal de JBoss EAP, se pueden seleccionar funciones y enviarlas del proyecto comunitario a una serie de versiones menores de mejora de funciones dentro de la misma familia de versiones principales. Por ejemplo, JBoss EAP 7.1 se desarrolló a partir de una rama en funcionamiento de WildFly 11.

JBoss EAP ofrece opciones preconfiguradas para funciones como agrupamiento de alta disponibilidad, mensajería y almacenamiento en caché distribuido. También permite que los usuarios escriban, implementen y ejecuten aplicaciones mediante la diversidad de API y servicios que ofrece JBoss EAP.

JBoss EAP incluye una estructura modular que permite la habilitación de servicios solo cuando es requerido, lo que aumenta la velocidad de inicio. La consola de administración basada en la Web y la interfaz de línea de comando (CLI) de administración hacen innecesaria la edición de los archivos de configuración de XML y añaden la capacidad de crear scripts y automatizar tareas. Asimismo, JBoss EAP incluye las API y marcos de desarrollo para crear rápidamente aplicaciones Java EE seguras y escalables.

Tabla 2.1. Funciones de JBoss EAP

Característica	Descripción
Cumple con Java EE 7	Perfil completo y web de Java Enterprise Edition 7 certificado.
Dominio administrado	Gestión centralizada de varias instancias de servidores y hosts físicos, al tiempo que el servidor autónomo permite una única instancia de servidor. Gestión de configuración, implementación, vinculación de sockets, módulos, extensiones y propiedades del sistema por grupo de servidores. Gestión centralizada y simplificada de la seguridad de las aplicaciones (incluidos los dominios de seguridad).
Consola de administración y CLI de administración	Nuevas interfaces de administración de dominios o servidores autónomos. La CLI de administración también incluye un modo de lote que puede generar scripts y automatizar las tareas de administración. No se recomienda editar directamente los archivos de configuración de JBoss EAP XML.

Característica	Descripción
Distribución de directorio simplificada	El directorio de módulos contiene todos los módulos del servidor de aplicaciones. Los directorios de dominio y autónomo contienen los artefactos y archivos de configuración para implementaciones de dominio y autónomas respectivamente.
Mecanismo modular de carga de clases	Los módulos se cargan y se descargan a pedido. Esto mejora el rendimiento, otorga beneficios para la seguridad y reduce los tiempos de inicio e reinicio.
Gestión optimizada de fuentes de datos	Los controladores de la base de datos se implementan como otros servicios. Asimismo, las fuentes de datos se crean y se gestionan mediante la consola de administración y la CLI de administración.

2.2. SUBSISTEMAS

Muchas de las API y capacidades expuestas a aplicaciones implementadas en JBoss EAP se organizan en subsistemas. Estos subsistemas pueden ser configurados por administradores para proporcionar un comportamiento diferente, según la meta de la aplicación. Por ejemplo, si una aplicación requiere una base de datos, se puede configurar una fuente de datos en el subsistema **datasources** a la que puede acceder la aplicación después de ser implementada en ese servidor o dominio JBoss EAP.

2.3. ALTA DISPONIBILIDAD

La alta disponibilidad (HA) en JBoss EAP se refiere a varias instancias de JBoss EAP que trabajan a la par para proporcionar aplicaciones más resistentes a fluctuaciones en el tráfico, la carga del servidor y la falla del servidor. HA incorpora conceptos como escalabilidad, balanceo de carga y tolerancia a fallas.

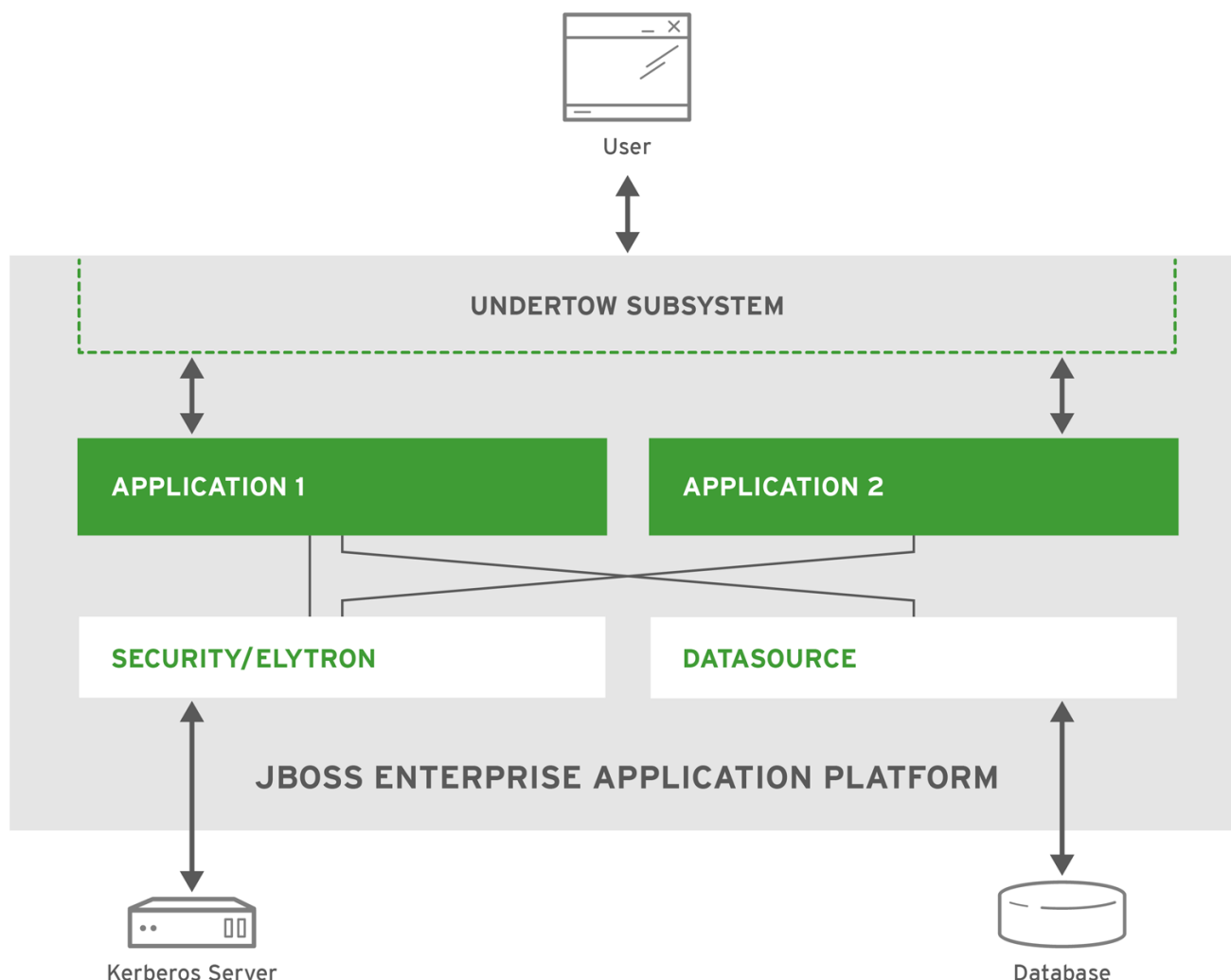
2.4. MODOS OPERATIVOS

Además de ofrecer funcionalidad y API a sus aplicaciones, JBoss EAP tiene capacidades de administración potentes. Estas capacidades de administración difieren según el modo operativo que se usa para iniciar JBoss EAP. JBoss EAP ofrece un modo operativo de *servidor autónomo* para gestionar instancias discretas y un modo operativo de *servidor administrado* para gestionar grupos de instancias desde un único punto de control.

CAPÍTULO 3. EJEMPLOS

A continuación, se brindan varios ejemplos para ilustrar cómo trabaja JBoss EAP y con qué entornos diferentes es compatible.

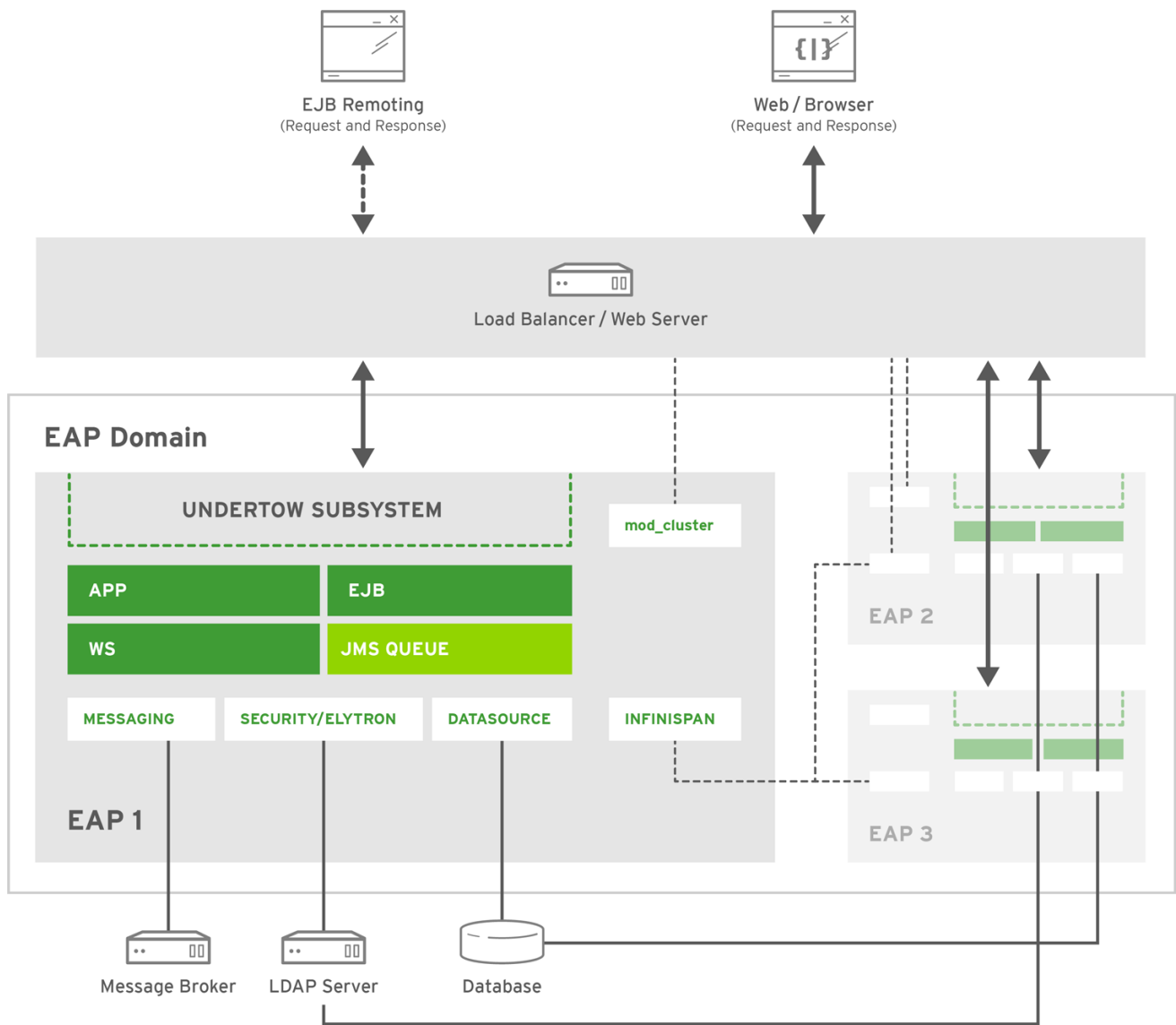
3.1. EJEMPLO SIMPLE



JBoss_430110_1216

En este ejemplo se muestra una configuración simple de JBoss EAP. La instancia de JBoss EAP cuenta con dos aplicaciones implementadas. También está configurada para conectarse a una base de datos mediante el subsistema **datasources** y el servidor Kerberos, el cual puede usar el subsistema **security** de legado o el subsistema **elytron**. Estas conexiones se exponen a las aplicaciones implementadas. La instancia JBoss EAP gestiona solicitudes a través del subsistema **undertow** y dirige estas solicitudes a la aplicación adecuada. Las aplicaciones usan las API expuestas por JBoss EAP para conectarse a la base de datos y el servidor Kerberos, y llevan a cabo su lógica empresarial implementada. Después de finalizar, las aplicaciones envían una respuesta al solicitante a través del subsistema **undertow**.

3.2. EJEMPLO AMPLIADO



JBoss_430110_1216

En este ejemplo se ilustra una configuración más compleja que implica tres instancias de JBoss EAP, dispuestas en un dominio gestionado con un balanceador de carga o un servidor web. Las tres instancias también están configuradas para respaldar una alta disponibilidad a través del balanceo de carga mediante `mod_cluster` y de la replicación de sesiones mediante Infinispan. Las tres instancias de JBoss EAP tienen una aplicación web, un servicio web y EJB implementados. Una instancia de JBoss EAP tiene una cola JMS configurada a través del subsistema **messaging-activemq**. Las tres instancias de JBoss EAP tienen conexiones a una base de datos a través de la fuente de datos. También tienen una conexión con el servidor LDAP mediante el subsistema **security** de legado o el subsistema **elytron**. Asimismo, una instancia de JBoss EAP está configurada para conectarse a un agente de mensajes externo a través del subsistema **messaging-activemq**. Estas conexiones configuradas se exponen a las aplicaciones, los servicios web, EJB y colas JMS implementadas en esa respectiva instancia.

Todas las solicitudes entrantes destinadas a la aplicación, el servicio web o EJB son recibidas en primer lugar por el balanceador de carga o el servidor web. En función del algoritmo de balanceo de carga configurado y la información provista por cada instancia de JBoss EAP, el servidor web o el balanceador de carga dirige esa solicitud a la instancia de JBoss EAP adecuada. La instancia de JBoss EAP maneja las solicitudes a través del subsistema **undertow** y dirige estas solicitudes a la aplicación adecuada. Las aplicaciones usan las API expuestas por JBoss EAP para conectarse a la base de datos y el servidor Kerberos, y llevar a cabo su lógica empresarial implementada. Después de finalizar, las

aplicaciones envían una respuesta al solicitante a través del subsistema **undertow**. Toda información no persistida, como, por ejemplo, la información de la sesión, se propaga entre las instancias de JBoss EAP a través del subsistema **infinispan**.

Revised on 2018-01-11 05:29:10 EST