



# **Red Hat JBoss Enterprise Application Platform 7.0**

## **Introduction to JBoss EAP**

For Use with Red Hat JBoss Enterprise Application Platform 7.0



# Red Hat JBoss Enterprise Application Platform 7.0 Introduction to JBoss EAP

---

For Use with Red Hat JBoss Enterprise Application Platform 7.0

## 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.

## Abstract

This document provides a high-level, conceptual overview of Red Hat JBoss Enterprise Application Platform as well as several topics on which JBoss EAP is based.

---

## Table of Contents

<b>CHAPTER 1. OVERVIEW OF GENERAL CONCEPTS</b> .....	<b>3</b>
1.1. JAVA	3
1.2. APPLICATION SERVERS	3
1.3. JAVA EE 7	3
<b>CHAPTER 2. OVERVIEW OF JBOSS EAP</b> .....	<b>4</b>
2.1. ABOUT JBOSS EAP 7	4
2.2. SUBSYSTEMS	5
2.3. HIGH AVAILABILITY	5
2.4. OPERATING MODES	5
<b>CHAPTER 3. EXAMPLES</b> .....	<b>6</b>
3.1. SIMPLE EXAMPLE	6
3.2. EXPANDED EXAMPLE	6



# CHAPTER 1. OVERVIEW OF GENERAL CONCEPTS

Before understanding how Red Hat JBoss Enterprise Application Platform can be configured and deployed, there are some important concepts to understand.

## 1.1. JAVA

Java is a programming language and a computing platform that incorporates concepts such as object-orientation, classes, and concurrency. Java applications are compiled down to bytecode and are run inside a Java Virtual Machine (JVM).

## 1.2. APPLICATION SERVERS

An application server, or app server, is software that provides an environment to run web applications. Most app servers also provide functionality to web applications running in their environment through a set of APIs. For example, an app server can provide an API for connecting to a database.

## 1.3. JAVA EE 7

Java EE (Java Platform, Enterprise Edition) is a standards-based enterprise platform that provides both an API and runtime environment for running and developing Java applications. The goal is to improve developer productivity by providing rich enterprise capabilities in easy to consume frameworks that eliminate boilerplate and reduce technical burden. The frameworks that compose Java EE are heavily tested in combination. Java EE 7, based on [JSR 342](#), is a follow up to Java EE 6, with the primary focus being additional simplification of APIs to access container services while increasing support for emerging web technologies, such as HTML5.

Java EE 7 includes support for multiple profiles, or subsets of APIs. The Java EE 7 specification defines the *full* profile and the *web* profile.

### Java EE 7 Web Profile

The Java EE 7 web profile is designed for web application development and supports a subset of the APIs defined by Java EE 7 related web-based technologies.

### Java EE 7 Full Profile

The Java EE 7 full profile contains all of the APIs defined by Java EE 7 (including all the items in the web profile). When developing EJBs, messaging applications, and web services (in contrast to web applications), use the full profile.

## CHAPTER 2. OVERVIEW OF JBOSS EAP

### 2.1. ABOUT JBOSS EAP 7

Red Hat JBoss Enterprise Application Platform 7.0 (JBoss EAP 7) is a certified implementation of the Java Enterprise Edition 7 (Java EE 7) **Full** and **Web** profile specifications.

Major versions of JBoss EAP are forked from the [Wildfly](#) community project at certain points where the community project has reached the desired feature completeness level. After that point, an extended period of testing and productization takes place, in which JBoss EAP is stabilized, certified, and enhanced for production use. During the lifetime of a JBoss EAP major version, selected features may be cherry-picked and back-ported from the community project into a series of feature enhancing minor releases within the same major version family. For example, JBoss EAP 7.0 is forked from WildFly 10.

JBoss EAP provides preconfigured options for features such as high-availability clustering, messaging, and distributed caching. It also enables users to write, deploy, and run applications using the various APIs and services that JBoss EAP provides.

JBoss EAP includes a modular structure that allows service enabling only when required, improving startup speed. The web-based management console and management command line interface (CLI) make editing XML configuration files unnecessary and add the ability to script and automate tasks. In addition, JBoss EAP includes APIs and development frameworks for quickly developing secure and scalable Java EE applications.

**Table 2.1. Features of JBoss EAP**

Feature	Description
Java Certification	Java Enterprise Edition 7 full and web profile certified.
Managed Domain	Centralized management of multiple server instances and physical hosts, while a standalone server allows for a single server instance. Per-server group management of configuration, deployment, socket bindings, modules, extensions, and system properties. Centralized and simplified management of application security (including security domains).
Management console and management CLI	New domain or standalone server management interfaces. XML configuration file editing is no longer required. The management CLI also includes a batch mode that can script and automate management tasks.
Simplified directory layout	The modules directory now contains all application server modules. The common and server-specific lib directories are deprecated. The domain and standalone directories contain the artifacts and configuration files for domain and standalone deployments respectively.



Feature	Description
Modular class-loading mechanism	Modules are loaded and unloaded on demand. This improves performance, has security benefits, and reduces start-up and restart times.
Streamlined datasource management	Database drivers are deployed like other services. In addition, datasources are created and managed using the management console and management CLI.

## 2.2. SUBSYSTEMS

Many of the APIs and capabilities that are exposed to applications deployed to JBoss EAP are organized into subsystems. These subsystems can be configured by administrators to provide different behavior, depending on the goal of the application. For instance, if an application requires a database, a datasource can be configured in the `datasources` subsystem and accessed by that application after it is deployed to that JBoss EAP server or domain.

## 2.3. HIGH AVAILABILITY

High availability (HA) in JBoss EAP refers to multiple JBoss EAP instances working together to provide applications that are more resistant to fluctuations in traffic, server load, and server failure. HA incorporates concepts such as scalability, load balancing, and fault tolerance.

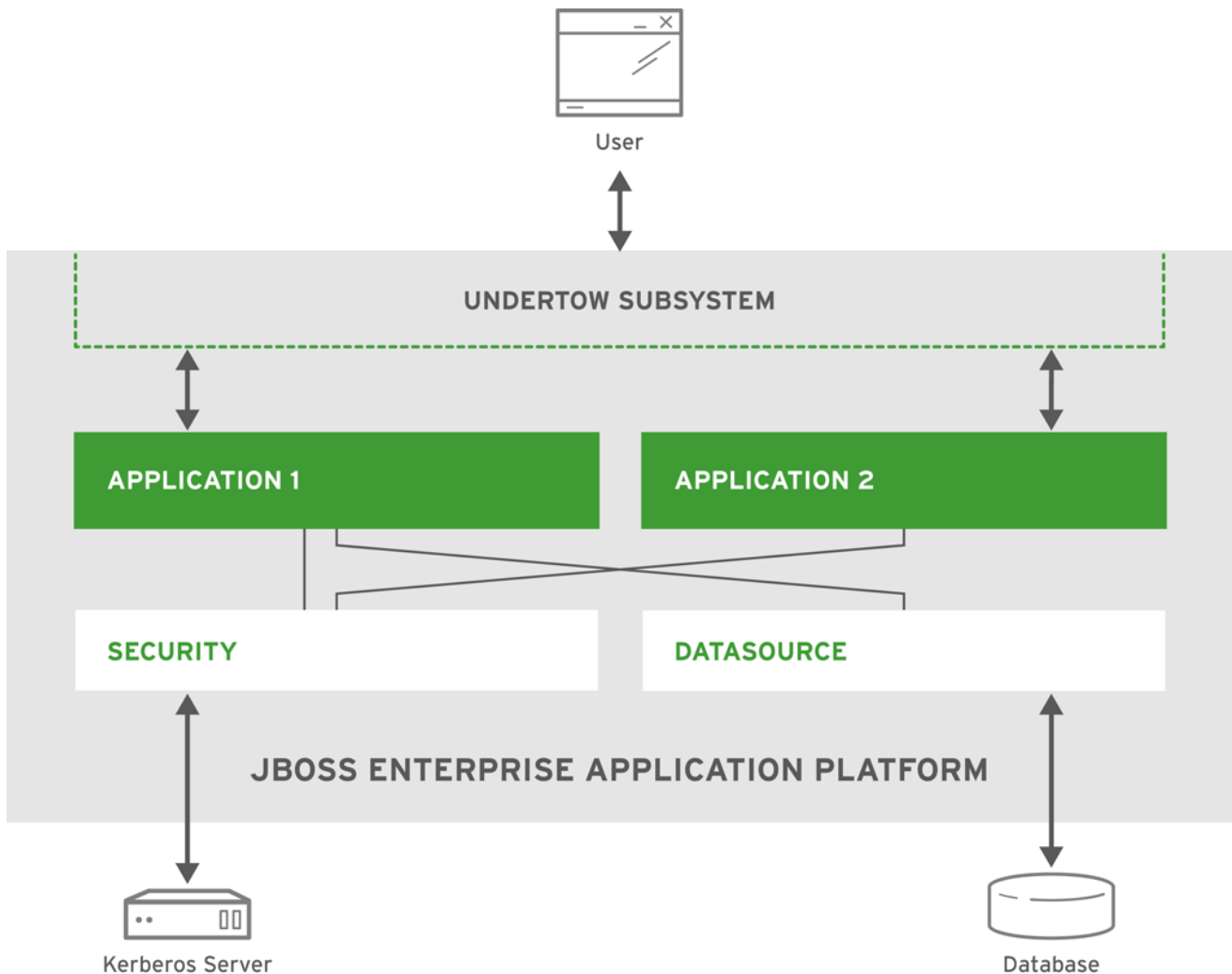
## 2.4. OPERATING MODES

In addition to providing functionality and APIs to its applications, JBoss EAP has powerful management capabilities. These management capabilities differ depending on which operating mode is used to start JBoss EAP. JBoss EAP offers a *standalone server* operating mode for managing discrete instances and a *managed domain* operating mode for managing groups of instances from a single control point.

## CHAPTER 3. EXAMPLES

Below are several examples to illustrate how JBoss EAP works and where it fits into different environments.

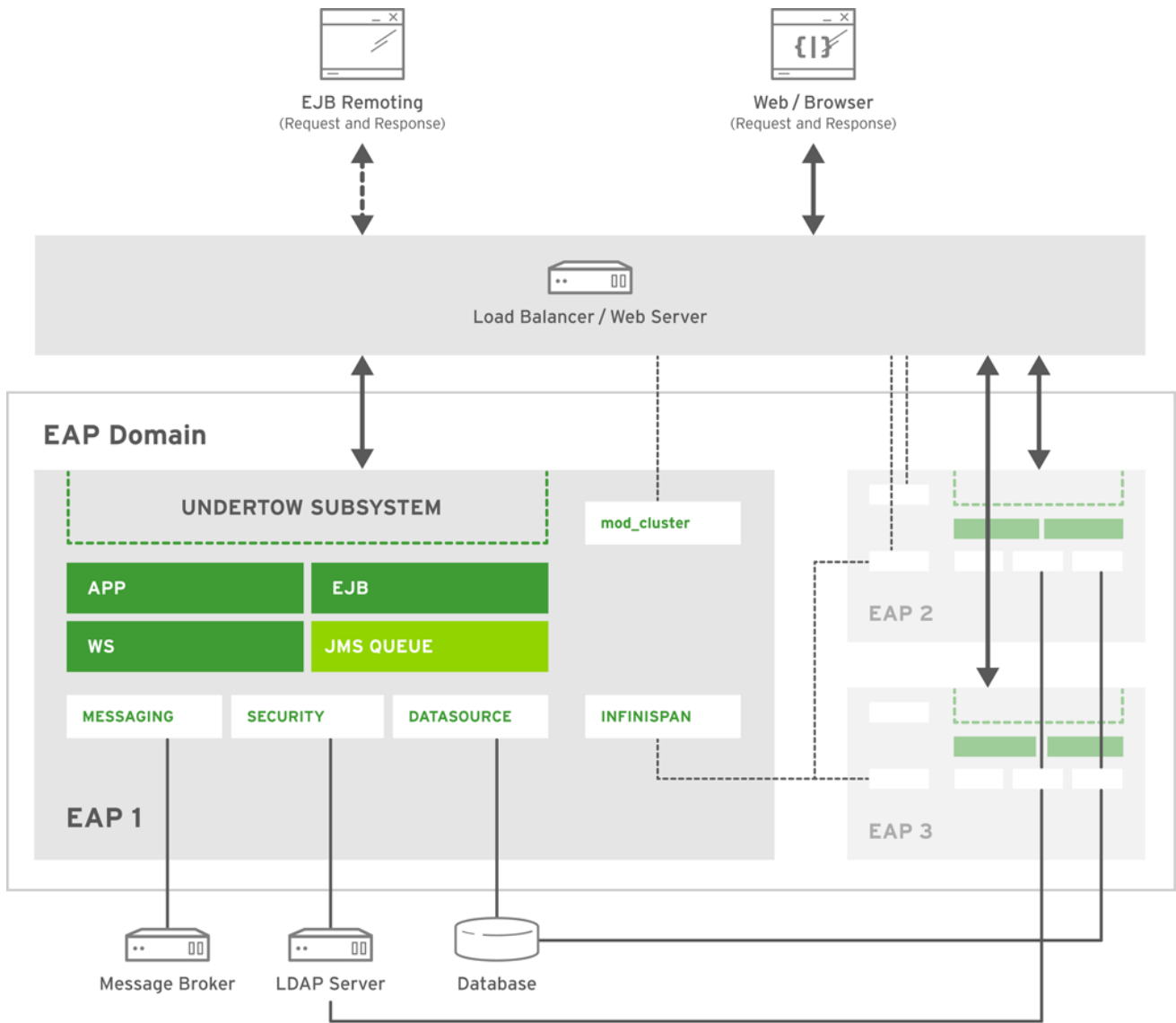
### 3.1. SIMPLE EXAMPLE



JBOSS\_380289\_1115

This example shows a simple JBoss EAP setup. The JBoss EAP instance has two applications deployed to it. It is also configured to connect to a database using the `datasources` subsystem and a Kerberos server using the `security` subsystem. These connections are exposed to the deployed applications. The JBoss EAP instance handles requests (through the `undertow` subsystem) and directs those requests to the appropriate application. The applications use the APIs exposed by JBoss EAP to connect to the database and Kerberos server, and perform their implemented business logic. After completion, the applications send a response back to the requester through the `undertow` subsystem.

### 3.2. EXPANDED EXAMPLE



JBOSS\_380289\_1115

This example illustrates a more complex setup involving three JBoss EAP instances arranged in a managed domain with either a load balancer or a web server. The three instances are also configured to support high availability through load balancing (using `mod_cluster`) and session replication (using `Infinispan`). All three JBoss EAP instances have a web application, a web service, and EJB deployed. One JBoss EAP instance has a JMS queue configured (through the `messaging-activemq` subsystem). All three JBoss EAP instances have connections to a database and an LDAP server (through the `datasource` and `security` domains, respectively) and one JBoss EAP instance is configured to connect to an external message broker (through the `messaging-activemq` subsystem). Those configured connections are exposed to the applications, web services, EJBs, and JMS queues deployed to that respective instance.

All inbound requests (intended for the application, web service, or EJB) are first received by the load balancer or web server. Based on the configured load balancing algorithm and the information provided by each JBoss EAP instance, the web server or load balancer directs that request to the appropriate JBoss EAP instance. The JBoss EAP instance handles requests (through the `undertow` subsystem) and directs those request to the appropriate application. The applications use the APIs exposed by JBoss EAP to connect to the database and Kerberos server, and perform their implemented business logic. After completion, the applications send a response back to the requester through the `undertow` subsystem. Any non-persisted information (for example, session information) is propagated among the JBoss EAP instances through the `infinispan` subsystem.

*Revised on 2018-02-08 10:17:32 EST*