



Red Hat JBoss Enterprise Application Platform 8.0

Introduction to Red Hat JBoss Enterprise Application Platform

Learn about Red Hat JBoss Enterprise Application Platform (JBoss EAP), JBoss EAP subsystems, and the different operating modes that JBoss EAP offers

Red Hat JBoss Enterprise Application Platform 8.0 Introduction to Red Hat JBoss Enterprise Application Platform

Learn about Red Hat JBoss Enterprise Application Platform (JBoss EAP), JBoss EAP subsystems, and the different operating modes that JBoss EAP offers

Legal Notice

Copyright © 2024 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, the Red Hat 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 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

Learn about Red Hat JBoss Enterprise Application Platform (JBoss EAP), JBoss EAP subsystems, and the different operating modes that JBoss EAP offers.

Table of Contents

PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. INTRODUCTION TO RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM	5
1.1. USES OF JBOSS EAP	5
1.2. JBOSS EAP FEATURES	5
1.3. APPLICATION SERVERS	7
1.4. JBOSS EAP SUBSYSTEMS	7
1.5. HIGH AVAILABILITY (HA) FUNCTIONALITY OF JBOSS EAP	7
1.6. SUPPORTED OPERATING MODES IN JBOSS EAP	7
CHAPTER 2. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM SETUPS	8
2.1. SIMPLE SETUP USING A SINGLE JBOSS EAP SERVER INSTANCE	8
2.2. COMPLEX SETUP USING MULTIPLE JBOSS EAP SERVER INSTANCES	9
CHAPTER 3. PACKAGE NAMESPACE CHANGE FOR JBOSS EAP 8.0	11
3.1. JAVAX TO JAKARTA NAMESPACE CHANGE	11

PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

Procedure

1. Click the following link to [create a ticket](#).
2. Enter a brief description of the issue in the **Summary**.
3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.
4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our [CTO Chris Wright's message](#).

CHAPTER 1. INTRODUCTION TO RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Before you start working with Red Hat JBoss Enterprise Application Platform, you must understand some general components that are used by JBoss EAP.

When you understand these components, you can enhance both your use of JBoss EAP and your ability to configure JBoss EAP.

1.1. USES OF JBOSS EAP

Red Hat JBoss Enterprise Application Platform (JBoss EAP) 8.0 is compatible with Jakarta EE 10 specifications, such as Web Profile, Core Profile, and Full Platform specifications. Each major version of JBoss EAP provides you with a tested, stabilized, and certified product.

JBoss EAP provides preconfigured options for features such as high-availability clustering, messaging, and distributed caching.

You can use JBoss EAP to deploy and run applications using supported APIs and services. Additionally, you can configure JBoss EAP to meet your needs, for example:

- You can customize JBoss EAP to include only the subsystems required to meet your needs.
- You can script and automate tasks by using the management command line interface (CLI) to avoid having to edit XML configuration files.

Major versions of JBoss EAP are forked from the WildFly community project at intervals when the community project has reached the desired feature completeness level. The major version is tested until it is stabilized, certified, and enhanced for production use.

During the lifecycle of a JBoss EAP major version, selected features are cherry-picked and back-ported from the community project into minor releases within the major release. Each minor release introduces feature enhancements to the major release.

Additional resources


- For more information about the WildFly community project, see the [WildFly community page](#).
- For more information about Jakarta EE 10 specifications, see [Red Hat JBoss Enterprise Application Platform Supported Standards](#).

1.2. JBOSS EAP FEATURES

JBoss EAP includes a variety of features to meet the needs of your organization.

Table 1.1. Features of JBoss EAP

Feature	Description
Jakarta EE compatible	JBoss EAP 8.0 is a Jakarta EE 10 compatible implementation for Web Profile, Core Profile, and Full Platform specifications.

Feature	Description
Managed domain	<p>Provides centralized management of multiple server instances and physical hosts, compared to a standalone server that supports just a single server instance.</p> <p>Provides server-group management of configuration, deployment, socket bindings, modules, extensions, and system properties.</p> <p>Provides centralized and simplified management of application security and security domains.</p>
Management console and management CLI	<p>Used for configuration and administration of EAP, such as for deploying and undeploying applications, configuring system settings, and performing other administrative tasks.</p> <p>The management CLI includes a batch mode that scripts and automates management tasks.</p> <div data-bbox="815 996 922 1191" style="display: inline-block; vertical-align: middle;">  </div> <p style="margin-left: 20px;">IMPORTANT</p> <p style="margin-left: 20px;">Do not directly edit the JBoss EAP XML configuration files while JBoss EAP is running. Use the management CLI to modify configurations.</p>
Simplified directory layout	<p>The modules directory contains application server modules.</p> <p>The domain directories contain the configuration for the managed domain.</p> <p>The standalone directories contain the configuration for a standalone server instance.</p> <p>Deployments, logs, tmp, and more are also contained under both the domain and standalone directories.</p>
Modular class-loading mechanisms	<p>JBoss EAP uses JBoss Modules which is a thread-safe, fast, and highly concurrent, delegating class loader model, that provide exact control over the classes visible to a given module or application.</p>
Streamlined datasource management	<p>Database driver deployment is similar to other JBoss EAP services.</p> <p>The management console and management CLI create and manage datasources.</p>

Feature	Description
Unified security framework	<p>Elytron provides a single unified framework for managing and configuring access for both standalone servers and servers in managed domains.</p> <p>Additionally, Elytron is used to configure security access for applications deployed on JBoss EAP servers.</p>

1.3. APPLICATION SERVERS

An application server, or app server, is software that provides an environment to run web applications. Most app servers use a set of APIs to provide functionality to web applications. For example, an app server can use an API to connect to a database.

1.4. JBOSS EAP SUBSYSTEMS

JBoss EAP organizes APIs into subsystems. You can configure these subsystems to enhance the capabilities of your JBoss EAP instance. For example, you can tune subsystems to improve performance, configure security, and to configure connections to external resources such as databases, identity providers, messaging brokers, and more.

Administrators can configure these subsystems to support different behavior, depending on the goal of the application.

For instance, if an application requires a database, you must configure a datasource so that a deployed application on a JBoss EAP server or a domain server can access the database.

1.5. HIGH AVAILABILITY (HA) FUNCTIONALITY OF JBOSS EAP

HA services are used to guarantee availability of deployed Jakarta EE application(s) by protecting against a single point of failure (failover) and preventing long delays during times of high request volumes (load balancing).

HA in JBoss EAP refers to multiple JBoss EAP instances working together to deliver applications that are most resistant to fluctuations in data flow, server load, and server failure. In addition to load balancing, HA also incorporates scalability and fault tolerance.

1.6. SUPPORTED OPERATING MODES IN JBOSS EAP

JBoss EAP has powerful management capabilities for deployed applications. These capabilities differ depending on which operating mode is used to start JBoss EAP.

JBoss EAP offers the following operating modes:

- Standalone server to manage instances individually
- Servers in a managed domain for managing groups of instances from a single control point

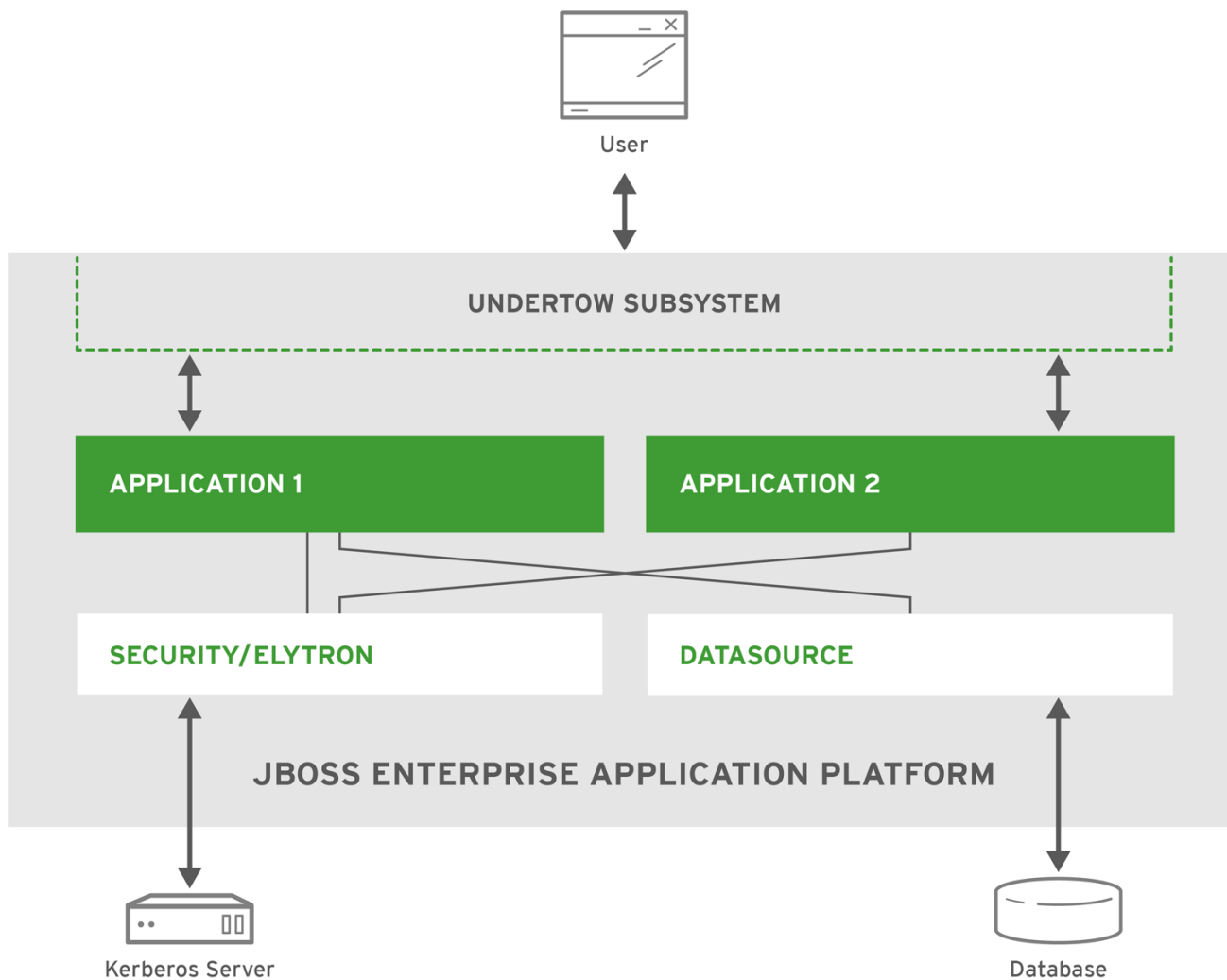
CHAPTER 2. RED HAT JOSS ENTERPRISE APPLICATION PLATFORM SETUPS

You can configure a JBoss EAP instance on single servers that run one or more applications, or you can configure multiple JBoss EAP server instances that are clustered together with an external load balancer for load balancing and failover.

2.1. SIMPLE SETUP USING A SINGLE JOSS EAP SERVER INSTANCE

A simple JBoss EAP setup consists of a single server running one or more deployed applications.

Figure 2.1. Simple setup using a single JBoss EAP server instance



JBOSS_430110_1216

The JBoss EAP instance uses the **datasources** subsystem to connect to the following components:

- A database
- A Kerberos server

JBoss EAP uses the **elytron** (security) subsystem to connect to the Kerberos server and expose the server to the two deployed applications.

JBoss EAP uses the **undertow** subsystem to handle requests from the client server and send requests to an appropriate application.

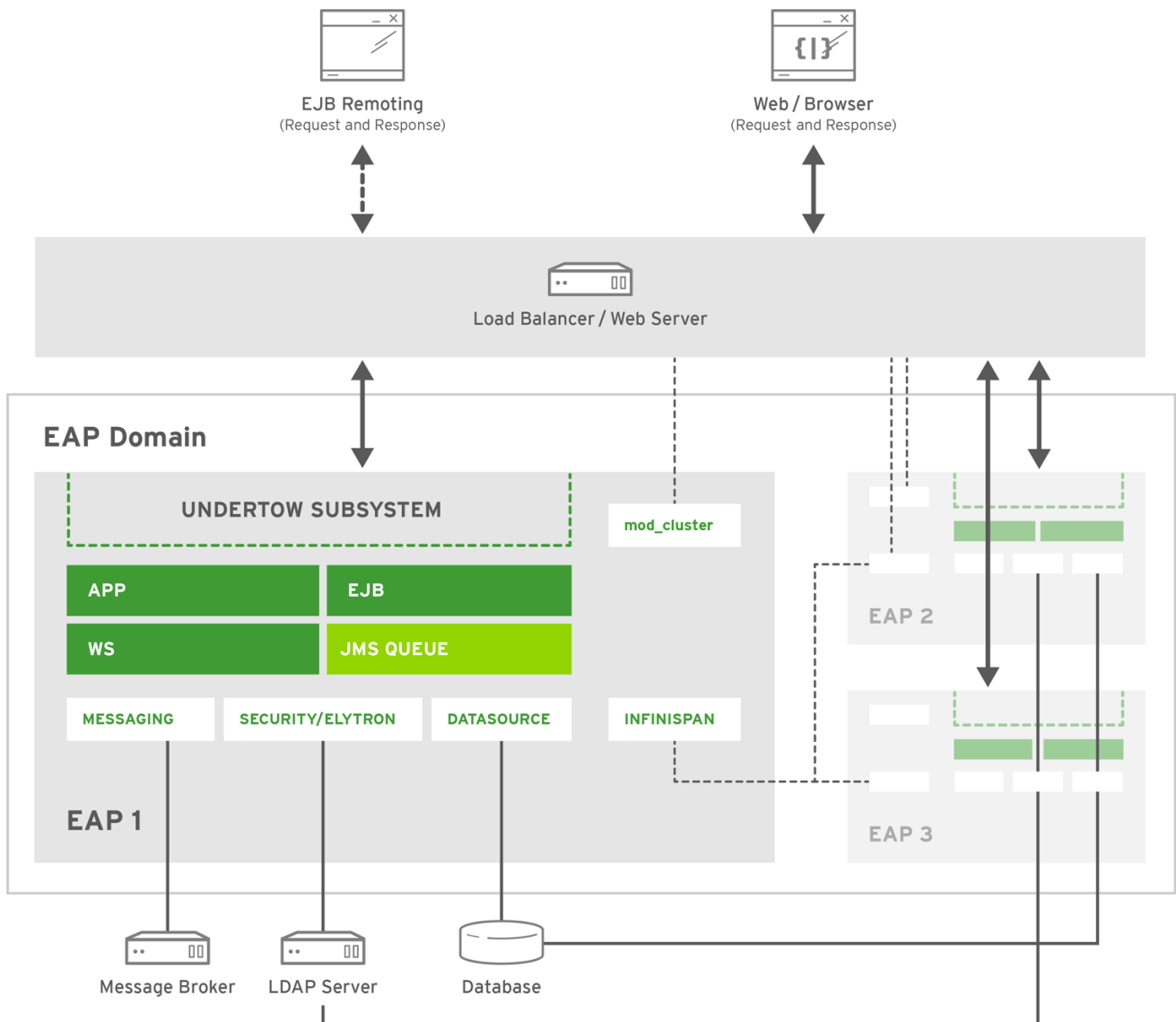
The application uses the APIs exposed by JBoss EAP to connect to the database and Kerberos server. The application completes its task and the **undertow** subsystem sends the response back to requester.

2.2. COMPLEX SETUP USING MULTIPLE JBOSS EAP SERVER INSTANCES

A complex setup may involve multiple JBoss EAP server instances. For example, you can use a load balancer to distribute the processing load across {JBoss EAP} instances in a managed domain.

The following diagram displays three JBoss EAP instances that are arranged by a load balancer in a managed domain:

Figure 2.2. Complex setup using multiple JBoss EAP server instances



JBOSS_430110_1216

In this example, the administrator configured each instance to use **mod_cluster** and **infinispan** session replication to provide high availability (HA) support for applications. Each instance includes the following components:

- A web application
- A web service

- A deployed enterprise bean
- A database connection that was established with the **datasources** subsystem
- A connection with the LDAP server that was established with the **elytron** (security) subsystem

The diagram displays the following configurations associated with a complex JBoss EAP setup:

- **EAP 1** has a **messaging-activemq** subsystem that is configured with a Jakarta Messaging queue that connects to an external message broker.
- The external message broker is shared among all running JBoss EAP instances.
- All inbound requests go through the load balancer.
 - Depending on the configured load-balancing algorithm and the information provided by each JBoss EAP instance, the load balancer directs the requests to the appropriate JBoss EAP instance.
- Each JBoss EAP instance uses the **undertow** subsystem to direct the requests to the appropriate application.
- Each application uses the APIs exposed by JBoss EAP to connect to the database and Kerberos server.
- After an application performs its work, the **undertow** subsystem to send a response to the requester.



NOTE

The **infinispan** subsystem propagates non-persisted information, such as session information, among the JBoss EAP instances.

CHAPTER 3. PACKAGE NAMESPACE CHANGE FOR JBOSS EAP 8.0

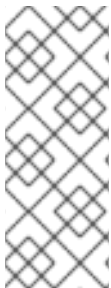
This section provides additional information for the package namespace changes in JBoss EAP 8.0. JBoss EAP 8.0 provides full support for Jakarta EE 10 and many other implementations of the Jakarta EE 10 APIs. An important change supported by Jakarta EE 10 for JBoss EAP 8.0 is the package namespace change.

3.1. JAVAX TO JAKARTA NAMESPACE CHANGE

A key difference between Jakarta EE 8 and EE 10 is the renaming of the EE API Java packages from **javax.*** to **jakarta.***. This follows the move of Java EE to the Eclipse Foundation and the establishment of Jakarta EE.

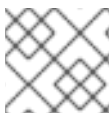
Adapting to this namespace change is the biggest task of migrating an application from JBoss EAP 7 to JBoss EAP 8. To migrate applications to Jakarta EE 10, you must complete the following steps:

- Update any import statements or other source code uses of EE API classes from the **javax** package to the **jakarta** package.
- Update the names of any EE-specified system properties or other configuration properties that begin with **javax** to begin with **jakarta**.
- For any application-provided implementations of EE interfaces or abstract classes that are bootstrapped using the **java.util.ServiceLoader** mechanism, change the name of the resource that identifies the implementation class from **META-INF/services/javax.[rest_of_name]** to **META-INF/services/jakarta.[rest_of_name]**.



NOTE

The Red Hat Migration Toolkit can assist in updating the namespaces in the application source code. For more information, see [How to use Red Hat Migration Toolkit for Auto-Migration of an Application to the Jakarta EE 10 Namespace](#). In cases where source code migration is not an option, the Open Source [Eclipse Transformer](#) project provides bytecode transformation tooling to transform existing Java archives from the **javax** namespace to the **jakarta** namespace.



NOTE

This change does not affect **javax** packages that are part of Java SE.

Additional resources

- For more information, see [The javax to jakarta Package Namespace Change](#).

Revised on 2024-02-21 14:02:54 UTC

