Introduction to JBoss EAP

Descriptions of general Red Hat JBoss Enterprise Application Platform concepts, including its subsystems and operating modes.
Descriptions of general Red Hat JBoss Enterprise Application Platform concepts, including its subsystems and operating modes.
Abstract

Learn about Red Hat JBoss Enterprise Application Platform (JBoss EAP). The document also introduces you to JBoss EAP subsystems and the different operating modes that JBoss EAP offers.
# Table of Contents

PROVIDING FEEDBACK ON RED HAT 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 7 ............................................................... 5
   1.2. JBOSS EAP 7 FEATURES ............................................................. 5
   1.3. APPLICATION SERVERS ............................................................... 6
   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 JBOSS EAP SETUP .......................................................... 8
   2.2. COMPLEX JBOSS EAP SETUP ....................................................... 9
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments. Follow the steps in the procedure to learn about submitting feedback on Red Hat documentation.

Prerequisites

- Log in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in Multi-page HTML format.

Procedure

1. Click Feedback to see existing reader comments.

   NOTE
   The feedback feature is enabled only in the Multi-page HTML format.

2. Highlight the section of the document where you want to provide feedback.

3. In the prompt menu that displays near the text you selected, click Add Feedback.
   A text box opens in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.
   You have created a documentation issue.

5. To view the issue, click the issue tracker link in the feedback view.
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. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, 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 used by JBoss EAP.

When you understand these components, you can enhance both your use of Red Hat JBoss Enterprise Application Platform and your ability to configure Red Hat JBoss Enterprise Application Platform.

1.1. USES OF JBOSS EAP 7

Red Hat JBoss Enterprise Application Platform (JBoss EAP) 7.4 is compatible with Jakarta EE 8 specifications, such as web 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. This can improve the startup speed of your JBoss EAP 7.4 instance.
- You can script and automate tasks by using the web-based management console and 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 life cycle 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 information about the WildFly community project see WildFly.

1.2. JBOSS EAP 7 FEATURES

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

Table 1.1. Features of JBoss EAP

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jakarta EE compatible</td>
<td>JBoss EAP 7.4 is Jakarta EE 8 compatible implementation for both Web Profile and Full Platform specifications.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Managed Domain</td>
<td>Centralized management of multiple server instances and physical hosts, compared to a standalone server that supports just a single server instance. Provides server-group management of configuration, deployment, socket bindings, modules, extensions, and system properties. Centralized and simplified management of application security and security domains.</td>
</tr>
<tr>
<td>Management console and management CLI</td>
<td>New domain or standalone server management interfaces. The management CLI includes a batch mode that scripts and automates management tasks.</td>
</tr>
<tr>
<td></td>
<td>NOTE: To avoid making changes to your system configuration while a domain is active, do not edit the config.xml file for the domain.</td>
</tr>
<tr>
<td></td>
<td>Do not directly edit the JBoss EAP XML configuration files. Use the management CLI to modify configurations.</td>
</tr>
<tr>
<td>Simplified directory layout</td>
<td>The modules directory contains application server modules.</td>
</tr>
<tr>
<td></td>
<td>The domain directories contain the artifacts and configuration files for domain deployments. The standalone directories contain the standalone deployments.</td>
</tr>
<tr>
<td>Modular class-loading mechanism</td>
<td>Modules are loaded and unloaded on demand. This practice improves security performance and reduces startup and restart times.</td>
</tr>
<tr>
<td>Streamlined datasource management</td>
<td>Database drivers are deployed similarly to other JBoss EAP services.</td>
</tr>
<tr>
<td></td>
<td>Datasources are created and managed with the management console and management CLI.</td>
</tr>
<tr>
<td>Unified security framework</td>
<td>Elytron provides a single unified framework for managing and configuring access for both standalone servers and managed domains. Additionally, Elytron is used to configure security access for applications deployed on JBoss EAP servers.</td>
</tr>
</tbody>
</table>

### 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.

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

You can use the JBoss EAP HA functionality to enhance any running applications by providing improved data sharing among multiple running JBoss EAP instances.

HA in JBoss EAP refers to multiple JBoss EAP instances working together to deliver enhanced applications, which are most resistant to fluctuations in data flow, server load, and server failure. HA incorporates numerous qualities, including scalability, load balancing, 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
- Managed domain for managing groups of instances from a single control point
CHAPTER 2. RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM SETUPS

You can configure JBoss EAP with either a simple or a complex setup.

A simple setup includes a JBoss EAP instance that consists of a standalone server running one or more deployed applications.

A complex setup includes multiple JBoss EAP instances that contain numerous components that facilitate communication among the instances.

2.1. SIMPLE JBOSS EAP SETUP

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

A standalone server cannot support the complex configurations that a managed domain server can, such as using a domain controller to configure each server instance according to the management policy of the domain. This standalone server use case is considered a simple JBoss EAP setup.

Figure 2.1. Image of a simple JBoss EAP setup

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

- A database
A Kerberos server

JBoss EAP uses the **security** subsystem or the **elytron** 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 then sends a response to the requester by using the **undertow** subsystem.

### 2.2. COMPLEX JBOSS EAP SETUP

JBoss EAP can handle complex use cases. For example, you can use a load balancer to distribute processing load across JBoss EAP instances in a managed domain.

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

**Figure 2.2. Image of a complex JBoss EAP setup**
In this example, the administrator has 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 legacy **security** subsystem or the **elytron** subsystem

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

- JBoss EAP 1 has a **messaging-activemq** subsystem that is configured with a Jakarta Messaging queue that connects to an external message broker.
- The connection exposes the applications, web services, enterprise beans, and Jakarta Messaging queues running on all 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 application uses 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.