Running the standalone JBoss EAP server

Goals

After completing this section, students should be able to do the following:

- Configure an instance of the Enterprise Application Platform (EAP) to run a server independent.

- List the features of a standalone EAP server.

Standalone Server Overview

Standalone servers are ideal for running a single EAP instance as a single server. On a standalone server, all settings appear in a single configuration file: JBOSS_HOME/standalone/configuration/standalone.xml. Within this file, users can configure different subsystems that comprise the functions of this server, such as logging, messaging, and data source management.

use

It is generally not recommended to directly edit this file. Instead, you should use the EAP Command Line Interface (CLI) or the management console, which provides a more secure exit for configuration.
Changes made using the CLI or the management console take place immediately, while changes to the XML do not take place immediately and can be overwritten by someone making changes to the CLI/management console.

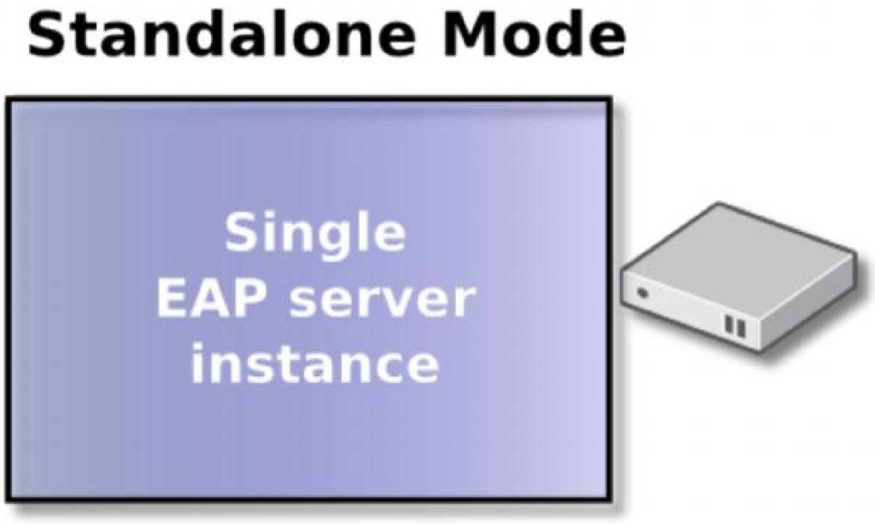

Figure 2.1: Standalone server installed on a single server

A standalone server has only one profile defined by default, while a managed domain has four defined by default. A profile refers to a set of subsystems and their settings used by the server.

Running multiple instances of independent EAP servers can create various complexities, such as requiring administrators to manage port conflicts, having multiple directories with duplicate files, and keeping the configuration synchronized for each server. To resolve these issues, EAP can be run in a managed domain which will be discussed in more depth later in this course.

The standalone EAP server provides the capabilities for clustering and high availability. These features can be configured using several of the other configuration files provided by EAP, in addition to standalone.xml.

The independent directory structure

After installing an EAP distribution, the standalone folder initially contains the following subfolders:

```
configuration/
deployments/
lib/
```

Below is a description of each of them:

- configuration – contains the configuration files for a server standalone and the standalone_xml_history subfolder, which maintains a version history of the configuration files. This folder also contains the logging.properties file for configuring loggers, a users file, mgmt users.properties, for defining login credentials to secure management interfaces, and a groups file, mgmt-groups.properties, to define the login roles.

- deployments: Contains the deployment files for applications used by Java EE, such as EAR, WAR, and JAR files. Marker files also appear in this folder (discussed later in this section).

- lib (libraries): is intended for implementations of common JAR files.

use

It is recommended to install Java libraries (JAR files) as modules, so this folder will often be empty.

These directories are critical to starting EAP. Without them, EAP could not start.

After running EAP for the first time, the following standalone subfolders are created:

```
data/
log/
tmp/
```

Below is a description of each of them:

- data (data): A location available to subsystems that store content on the file system, such as the message queue or an in-memory database.

- log - The default location for server log files, including gc.log.0.current and server.log, which contain the EAP startup logs.

- tmp (temporary): For temporary files, such as the shared key mechanism used by the Command Line Interface (CLI), to authenticate local users to the server.

The EAP base directory

You can separate the configuration and data folders of an EAP instance from the folder where the EAP application is installed. This allows users to keep configurations separate from the configuration, making it easier to upgrade and adopt a higher version of EAP. It also allows users to install EAP once and use it multiple times on the same computer.

There are two command line properties that allow users to create this separation between the installation folder and the folders used by the running instance:

1. jboss.home.dir – This property represents the root directory where EAP is installed. If this property is not defined, JBOSS_HOME is set as the default value.

2. jboss.server.base.dir – This property represents the base directory for the server configuration files. If you do not define this property, JBOSS_HOME/standalone is set as the default value.

For example, these properties can be set when running the standalone.sh script to start the EAP server:

```
$ ./standalone.sh -Djboss.server.base.dir=/path/to/base/directory  -Djboss.home.dir=/
path/to/home/directory
```

use

The port-offset attribute can be changed without editing standalone.xml by using the jboss.socket.binding.port-offset property on the command line. For example:

```
$  ./standalone.sh  -Djboss.socket.binding.port-offset=10000
```

Using a custom configuration file

By default, EAP will use BASE_DIR/configuration/standalone.xml to start a standalone server. However, an alternate configuration file can be used by passing the file name as a parameter to the standalone.sh script.
This approach is useful for customizing the configuration file without touching the files

of default settings. To use a different configuration file, the --server-config parameter can be used as follows:

```
$ ./standalone.sh ---server-config standalone-full.xml
```

An alternative parameter (-c) can also be used:

```
$ ./standalone.sh --c standalone-full.xml
```

Demo: Run JBoss EAP from a custom location

1. Users often want to be able to run multiple standalone instances without needing to install EAP multiple times on the same host. This demo will create an alternate directory to customize the EAP instance and leave the original installation intact.

   Copy the following folders from /opt/jboss-eap-8.0/standalone to the new EAP location /home/student/labs/custom-eap:

   - configuration
   - deployments
   - lib

   ```
   $ cd /opt/jboss-eap-8.0/standalone
   $ cp -r configuration deployments lib  /home/student/labs/custom-eap
   ```

   ### use

   Do not copy the data, log and tmp folders. These are created automatically when EAP is first started.

2. open /home/student/labs/custom-eap/configuration/ standalone.xml with a text editor. Subsequent steps will customize the server configuration by editing the standalone.xml file, to verify that EAP is using the configuration files in the new folder.

3. In the standard-sockets socket binding group, change the value The port-offset attribute defaults from 0 to 10000. This action will add 10000 to each port number used in the standard-sockets bonding group.

```
 ...
<socket-binding-group name="standard-sockets" default-interface="public" port  offset="${jboss.socket.binding.port-
offset:10000}">
 ...
```

4. Save the changes to standalone.xml.

5. The jboss.server.base.dir variable allows users to submit a directory path to use as an alternate base directory for server content. Run the following command to start the EAP server using the standalone.sh script in the original EAP installation, while using the new EAP configuration files:

```
$ cd /opt/jboss-eap-8.0/bin
$ ./standalone.sh -Djboss.server.base.dir=/home/student/labs/custom-eap/
```

The server should start without issue with the following message:

```
17:03:30,497 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: JBoss EAP 8.0.0.GA (WildFly Core
  2.1.2.Final-redhat-1) started in 3313ms -Started 261 of 509 services (332 services are lazy, passive or on-demand)
```

6. On your workstation, point your browser to http://localhost:18080 to see the page for Welcome to EAP with the new port offset.

7. On workstation, point the browser to http://localhost:19990 to see the EAP management console running with the same port offset.

use

The administrator username is jbossadm and the password is JBoss@RedHat123.

8. Browse the contents of the /home/student/labs/custom-eap folder on a new terminal window.

```
$ ls /home/student/labs/custom-eap/
configuration data deployments lib log tmp
```

Notice the three new folders: data, log, tmp. The folders are automatically created when the EAP server starts.

9. In the new terminal window, start a second EAP server without specifying a user jboss.server.base.dir with the user jboss:

```
$ sudo  -u  jboss  /opt/jboss-eap-8.0/bin/standalone.sh
```

use

The /opt/jboss-eap-8.0 directory is owned by the jboss user. Other users only have read and execute privileges. Since starting the EAP instance inside that directory creates folders and writes files, you must be the jboss user starting the server with that base directory.

10. Observe that the server starts without any port conflicts by checking that there are no ERROR messages in the server log. For this EAP instance, the original configuration files still have a port offset of 0, which means that this instance will not conflict with the other instance that has a port offset of 10000.

11. Open a browser on your workstation and go to http://localhost:8080. must see the welcome page on ports 8080 and 18080. EAP is used twice on the same machine with the same installation folders, but each server has its own configuration and implementations.

12. Using a browser on the workstation, verify that the management console on port 9990 is accessible by visiting http://localhost:9990. Use the administrator credentials to log in. Click Runtime, Standalone Server, Environment, and then click View.

    Filter the environment variables by looking for jboss.server.base.dir to see the base directory for this instance. Repeat these steps on the other EAP instance that is running by visiting the management console at http://localhost:19990.

    Each server is managed separately in standalone mode; they do not share settings. If you want multiple servers to be configured the same, you must run EAP in a managed domain.

13. Stop both instances of EAP by pressing Ctrl+C in each terminal window that you are running EAP.

This concludes the demo.