

Guided Exercise: Deploying to a Managed Domain

In this lab work, you will create two new host controllers to be slaves to the previously created domain controller.

Resources	
Files:	/home/student/JB248/labs/host
	/home/student/JB248/labs/domain-deploy
App URL:	http://172.25.250.254:8230/
	http://172.25.250.254:9080/
	http://172.25.250.254:8080/

Results

Must be able to boot from host controllers slaves of a domain controller.

before you start

Before beginning the guided exercise, run the following command to verify that EAP was installed to /opt/jboss-eap-7.0, that no EAP instances are running, and that the previous lab work established a domain with two Jboss servers. host at /home/student/JB248/labs/host:

```
[student@workstation ~]$ lab domain-deploy setup
```

1. Start the domain and hosts on the workstation virtual machine.

- 1.1. Use the following command to start the domain controller that was configured in a previous exercise:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./domain.sh \
-Djboss.domain.base.dir=/home/student/JB248/labs/domain/machine1/domain/ \ --host-config=host-master.xml
```

- 1.2. Open a new terminal window and start host2 created in the previous exercise using the following command:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./domain.sh \
-Djboss.domain.base.dir=/home/student/JB248/labs/host/machine2/domain/ \ --host-config=host-slave.xml \
-Djboss.domain.master.address=172.25.250.254
```

- 1.3. Open a new terminal window and start host3 created in the previous exercise using the following command:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./domain.sh \
```

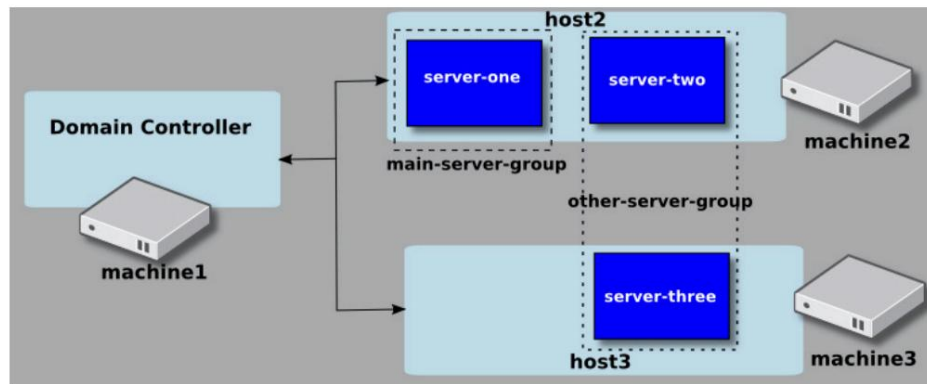
Chapter 5. Configuring servers in a managed domain

```
-Djboss.domain.base.dir=/home/student/JB248/labs/host/machine3/domain/ \ --host-
config=host-slave.xml \
-Djboss.domain.master.address=172.25.250.254
```

Wait for the domain controller and the two host controllers to start before proceeding.

2. Deploy an application to the main-server-group server group.

Here is a reminder of the managed domain architecture:



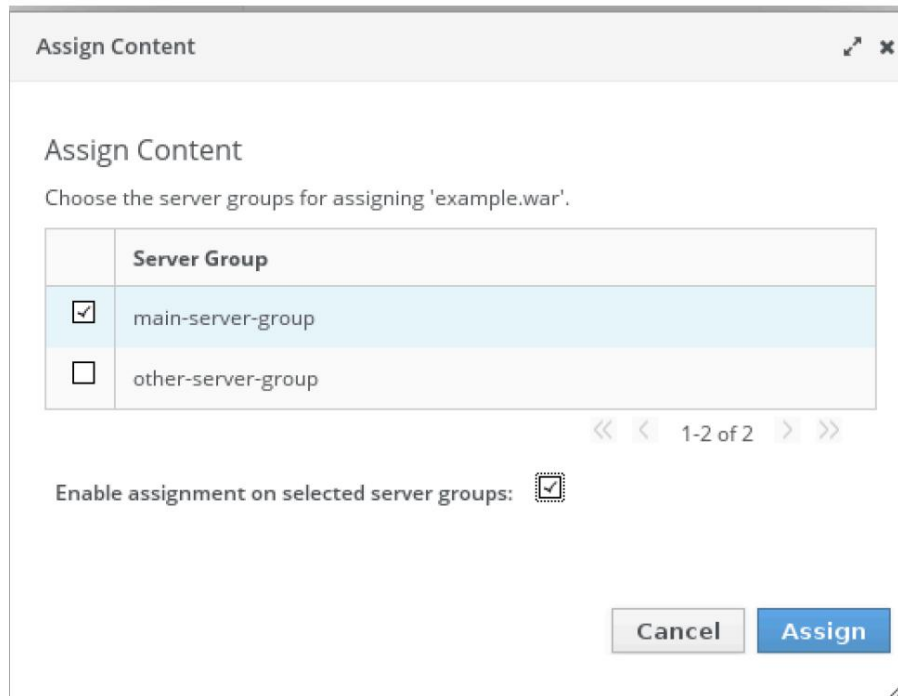
- 2.1. On the workstation virtual machine, open the management console by pointing the web browser to 172.25.250.254:9990. The administrator username is jbossadm and the password is JBoss@RedHat123.
- 2.2. Click Deployments at the top of the page, and then click Content Repository in the first column.
- 2.3. Click Add, select Upload a new deployment, and click Next. Click Browse and select the /home/student/JB248/labs/domain-deploy/example.war WAR file to upload.
- 2.4. On the Verify Upload screen, leave the Name and Runtime Name as example.war; then click Finish.



use

The Runtime Name refers to the name of the application, without the war extension, in which the sample application will be accessed.

- 2.5. Click on example.war in the second column and then on the Assign button.
- 2.6. Select main-server-group to deploy example.war to main server-group. Select Enable mapping on selected server groups, and then click Map:



Assign Content

Choose the server groups for assigning 'example.war'.

	Server Group
<input checked="" type="checkbox"/>	main-server-group
<input type="checkbox"/>	other-server-group

1-2 of 2

Enable assignment on selected server groups: ☒

Cancel Assign

3. Check the implementation.

- 3.1. Observe the terminal window of host2. You should see a message similar to the following, indicating that example.war has been deployed on server-one:

```
[Server:server-one] 22:11:10,423 INFO [org.jboss.as.server] (ServerService Thread Pool -- 75)
WFLYSRV0010: Deployed "example.war" (runtime-name : "example.war")
```

- 3.2. Observe the terminal window of host3. The application should not be deployed on host3, since there are no servers on host3 in the main-server-group.
- 3.3. Navigate to server-one at <http://172.25.250.254:8080/example/>. Must see the "Welcome to EAP 7" page of the example application.
- 3.4. Similarly, navigate to server-two at <http://172.25.250.254:8230/example/>. You should get a 404 error, because application example is not deployed on server-two.
- 3.5. Open the domain.xml file in the /home/student/JB248/labs/domain/machine1/domain/ configuration folder. Notice that example.war is now listed as an available deployment in the <deployments> section. Also notice that example.war appears in the <deployments> section of the main server-group in the <server-group> section.

4. Deploy an application to the other-server-group server group.

- 4.1. Return to the Deployments page of your administration console. domain. Click Content Repository, and then click Add.

Chapter 5. Configuring servers in a managed domain

- 4.2. Select Upload a new deployment on the first screen, and then select Next. Click Browse, upload the file /home/student/JB248/labs/domain-deploy/version.war, and click Next.

On the Verify Upload screen, leave the Name and Runtime Name as version.war; then click Finish.

- 4.3. Select version.war from the Content Repository menu and click Assign. Choose the other-server-group server group, which contains the server instances server-two and server-three. Be sure to select Enable mapping on selected server groups in other-server-group.

5. Check the implementation.

- 5.1. Observe the terminal windows of host2 and host3. The version.war application was deployed to server-two on host2 and server-three on host3. The output on host2 should be similar to the following:

```
[Server:server-two] 06:37:11,608 INFO [org.jboss.as.server] (ServerService Thread Pool -- 78)
WFLYSRV0010: Deployed "version.war" (runtime-name : "version.war")
```

- 5.2. Navigate to server-three, located at <http://172.25.250.254:9080/version/>. You should see the output of the version application.
- 5.3. Navigate to server-two, located at <http://172.25.250.254:8230/version/>. The version app should also appear at this URL.
- 5.4. Navigate to server-one, located at <http://172.25.250.254:8080/version/>. You should get a 404 error, because server-one is not a member of other-server-group.

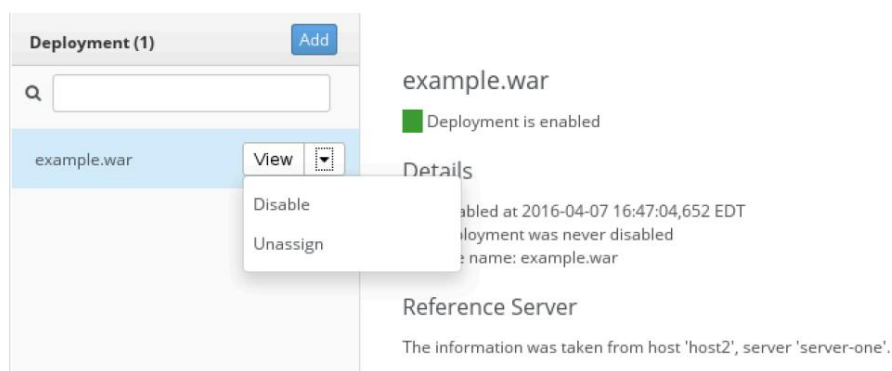
6. Undeploy an application.

- 6.1. Go to the Management Console Deployments page for your domain controller located at <http://172.25.250.254:9990> with username jbossadm and password JBoss@RedHat123.
- 6.2. Click the Server Groups tab in the first column.
- 6.3. Notice that the deployed applications are listed by server group. Click main-server-group in the Server Groups list.

Home	Deployments	Configuration	Runtime	Access Control	Patching
Browse By		Server Group (2)		Deployment (1)	
Content Repository		Q		Q Deployment is disabled	
Unassigned Content		main-server-group		example.war	
Server Groups		other-server-group		View	

- 6.4. Click main-server-group on the Server Group (2) tab.

- 6.5. Click `example.war` in the Deployments list, click the arrow next to View, and select **Disable** to disable the application. Click **Confirm** on the popup dialog.



- 6.6. View the `example.war` application on `server-one` in the browser by going to `http://172.25.250.254:8080/example/` and verify that you now get a 404 error.
- 6.7. Click the same arrow in the Deployment column and select **Deallocate** to make the deployment disappear, which will make `example.war` no longer assigned to `main-server-group`.
- 6.8. Follow the same steps to disable and remove `version.war` from `other-server-group`.
- 6.9. The applications have been removed from the EAP servers. To verify this, navigate to `http://172.25.250.254:9080/version/` for `server-three` and `http://172.25.250.254:8230/version/` for `server-two`. You should get 404 errors.

7. Remove a deployment.

- 7.1. Click the Content Repository tab on the Deployments page.
- 7.2. Select `example.war` in the Content Repository list and click the arrow next to `example.war`. Select **Delete**. Then click **Confirm** when the confirmation window appears.
- 7.3. Similarly, remove `version.war` from the Content Repository.

8. Press **Ctrl+C** to stop the domain controller and both domain controllers. host.

This concludes the guided exercise.

Lab Work: Configuring Servers in a Managed Domain

In this lab work, you will configure servers and server groups in a managed domain and deploy the bookstore application in the managed domain.

Resources	
Files	/opt/domain /home/student/JB248/labs/managed-domain-servers
app url	http://172.25.250.10:8080/bookstore http:// 172.25.250.11:8080/bookstore

Result

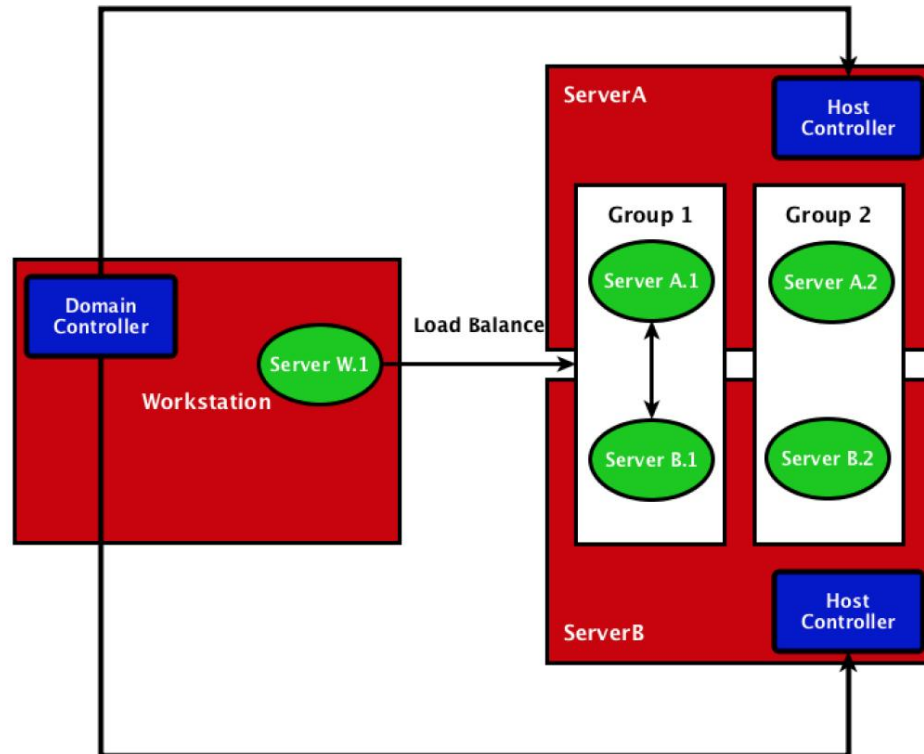
You should be able to configure an EAP 7 managed domain with two server groups and four servers, and deploy the bookstore application.

before you start

Use the following command to download the relevant lab files, ensuring that the managed domain is set up correctly:

```
[student@workstation ~]$ lab managed-domain-servers setup
```

1. An EAP administrator has configured a managed domain with two host controllers running on the servera and serverb virtual machines, respectively, and the domain controller on the workstation virtual machine. The domain and host configuration files are stored in the /opt/domain folder on all three machines. You need to start the managed domain and then create new servers and server groups, based on the graphic below:



As a final step, you should deploy the bookstore application WAR file to one of the newly created server groups. Bookstore is an application that supports high availability features and requires EAP messaging subsystems and high availability features. The EAP full-ha profile must be used to deploy the application to EAP. You can use the EAP 7 management console or the JBoss EAP CLI to achieve your goals, keeping in mind that the EAP CLI is the preferred option in production environments.

2. Briefly review the domain controller settings on the workstation.
Review the `/opt/domain/configuration/host-master.xml` file and the `/opt/domain/configuration/domain.xml` file to understand how the managed domain is configured. Verify that there are no server groups defined in the `domain.xml` file.
3. Start the domain controller on workstation. Because the files domain controller configuration are kept in the `/opt/domain` folder on workstation, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the domain controller is named `host-master.xml` and is located in the `/opt/domain/configuration` folder. (Tip: send the `--host-config=host master.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so you must start the domain controller using `sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh ...`

Chapter 5. Configuring servers in a managed domain

4. The two host controllers on `servera` and `serverb` connect to the host controller domain in the previous step and get the latest configuration of the domain. Start the two host controllers on `servera` and `serverb`.

- 4.1. Briefly review the host controller configuration on `servera` and `serverb`. Review the `/opt/domain/configuration/host-slave.xml` file to understand how hosts are configured. Verify that there are no servers defined on either host.

- 4.2. Start the host controller on `servera`. Because the host controller configuration files are kept in the `/opt/domain` folder on `servera`, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the host controller is named `host-slave.xml` and is located in the `/opt/domain/configuration` folder. (Hint: Pass the `--host config=host-slave.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so start the host controller using `sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...`

- 4.3. Start the host controller on `serverb`. Because the host controller configuration files are kept in the `/opt/domain` folder on `serverb`, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the host controller is named `host-slave.xml` and is located in the `/opt/domain/configuration` folder. (Hint: Pass the `--host config=host-slave.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so start the host controller using `sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...`

- 4.4. Verify that both host controllers connect to the domain controller and form a managed domain. Look at the console window in which you started the domain controller and verify that both `servera` and `serverb` are registered as slaves to the domain controller.

5. Access the JBoss EAP CLI. In a new terminal window on workstation, start the EAP CLI and connect to the domain controller as the `jboss` user:

```
[student@workstation bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/jboss-cli.sh \
--connect --controller=172.25.250.254:9990
```

6. By default, there are no server groups or servers defined in the managed domain. (These were removed in the final lab work of Chapter 4.) To deploy the bookstore application (WAR file), you must create server groups and define servers.

Create two new server groups named `Group1` and `Group2` in the managed domain.

6.1. The bookstore application uses the messaging subsystem and high availability capabilities of EAP, and these features are only available in the full-ha profile. Create a new server group named Group1, which is used to emulate a development environment with the following characteristics:

- Name: Group1
- Profile: full-ha
- Socket binding group: full-ha-sockets

6.2. Create another server group named Group2 to deploy bookstore in a production environment with the same characteristics as the development environment:

- Name: Group2
- Profile: full-ha
- Socket binding group: full-ha-sockets

6.3. Verify the server groups that you created in the previous steps using the JBoss EAP CLI or management console.

7. You must define four servers (two each in servera and serverb). Create the servers and assign them to the appropriate groups. When running multiple servers on one host, be sure to set the port offsets correctly to avoid port clashing. Also make sure that the servers are configured to start automatically when the host controller starts or restarts.

7.1. Create a new server named servera.1 in servera with the following characteristics:

- Name: servera.1
- Server Group: Group1
- Socket Binding Port Offset: Leave the default value of 0
- Autostart: true

7.2. Create a new server named servera.2 on servera with the following characteristics:

- Name: servera.2
- Server Group: Group2
- Offset of socket binding ports: 100
- Autostart: true

Chapter 5. Configuring servers in a managed domain

7.3. Create a new server named serverb.1 in serverb with the following characteristics:

- **Name:** serverb.1
- **Server Group:** Group1
- **Socket Binding Port Offset:** Leave the default value of 0
- **Autostart:** true

7.4. Create a new server named serverb.2 in serverb with the following characteristics:

- **Name:** serverb.2
- **Server group:** Group2
- **Offset of socket binding ports:** 100
- **Autostart:** true

8. You have now created four servers and associated them with the appropriate server groups. It will now start the newly created servers. Instead of starting servers one by one, you can start all servers together in a server group.

8.1. Start the servers in Group1

8.2. Start the servers in Group2

8.3. Verify that the servers have started successfully by looking at the console windows in which you started the host controllers. Verify that there are no port clashes and that the ports are bound according to the port offset values defined in the previous steps.

8.4. Because there is a firewall running on servera and serverb, you should unblock all ports used by servera.1, servera.2, serverb.1 and serverb.2 for public access. To unlock them, run the following command in a new terminal window on serverA:

```
[student@servera ~]$ sudo firewall-cmd --zone=public --add-port 8080/tcp \ --permanent
[student@servera ~]$ sudo firewall-cmd --zone=public --add-port 8180/tcp \ --permanent [student@servera ~]$ sudo
firewall-cmd
--reload
```

8.5. To unblock the firewall ports on serverb, run the following commands in a new terminal window on serverb:

```
[student@serverb ~]$ sudo firewall-cmd --zone=public --add-port 8080/tcp \ --permanent
[student@serverb ~]$ sudo firewall-cmd --zone=public --add-port 8180/tcp \
```

```
--permanent
[student@serverb ~]$ sudo firewall-cmd --reload
```

8.6. Verify that you can access the default welcome page for all four servers that you created in the previous steps. Go to the following URLs and verify that you can see the EAP Welcome page:

- **serve.1:** `http://172.25.250.10:8080`
- **serve.2:** `http://172.25.250.10:8180`
- **serverb.1:** `http://172.25.250.11:8080`
- **serverb.2:** `http://172.25.250.11:8180`

9. You are now ready to deploy the bookstore application WAR file to the managed domain. The bookstore.war file is available in the workstation /tmp folder.

9.1. Application deployments to a managed domain are always done at the server group level. Each application belongs to one or more server groups. Deploy the bookstore.war file to Group1.

Verify that the bookstore.war file is deployed to servera.1 and serverb.1 as they are part of the Group1 server group. Monitor the console window of servera and serverb for any error messages or warnings.

9.2. Verify that you can access the bookstore application at `http://172.25.250.10:8080/bookstore` y `http://172.25.250.11:8080/ bookstore`.

9.3. Verify that you can NOT access the bookstore application at `http://172.25.250.10:8180/bookstore` or `http://172.25.250.11:8180/ bookstore` because these servers are part of Group2 and you did not deploy the application to Group2 .

10. Using the JBoss EAP administrative console or CLI, undeploy the bookstore application and power off the servers, pools, host controllers, and the entire managed domain.

10.1. Stop servera.1.

10.2. Verify that the bookstore application is no longer accessible from servera.1, but that is still accessible from serverb.1.

10.3. Undeploy the bookstore application.

10.4. Stop all servers in Group1. Verify that the bookstore app is no longer be accessible.

10.5. Stop the servers in Group2.

10.6. Shut down the host controller on servera. Observe the servera console window and verify that the host controller has been shut down.

Chapter 5. Configuring servers in a managed domain

10.7. Shut down the host controller in serverb. Look at the serverb console window and verify that the host controller has been shut down.

11. Perform cleaning and grading.

11.1. Press Ctrl+C to stop the domain controller. (Alternatively, you can shut down the domain controller using the JBoss EAP CLI.)

11.2 Run the following workstation command to grade the assignment:

```
[student@workstation bin]$ lab managed-domain-servers grade
```

This concludes the lab work.

Solution

In this lab work, you will configure servers and server groups in a managed domain and deploy the bookstore application in the managed domain.

Resources	
Files	/opt/domain /home/student/JB248/labs/managed-domain-servers
app url	http://172.25.250.10:8080/bookstore http:// 172.25.250.11:8080/bookstore

Result

You should be able to configure an EAP 7 managed domain with two server groups and four servers, and deploy the bookstore application.

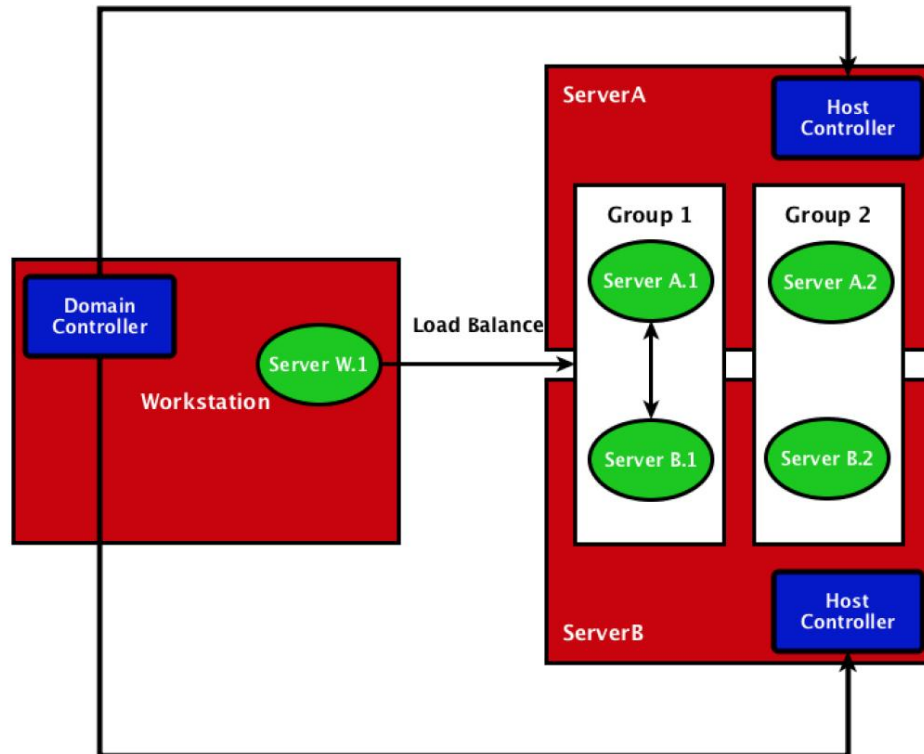
before you start

Use the following command to download the relevant lab files, ensuring that the managed domain is set up correctly:

```
[student@workstation ~]$ lab managed-domain-servers setup
```

1. An EAP administrator has configured a managed domain with two host controllers running on the servera and serverb virtual machines, respectively, and the domain controller on the workstation virtual machine. The domain and host configuration files are stored in the /opt/domain folder on all three machines. You need to start the managed domain and then create new servers and server groups, based on the graphic below:

Chapter 5. Configuring servers in a managed domain



As a final step, you should deploy the bookstore application WAR file to one of the newly created server groups. Bookstore is an application that supports high availability features and requires EAP messaging subsystems and high availability features. The EAP full-ha profile must be used to deploy the application to EAP. You can use the EAP 7 management console or the JBoss EAP CLI to achieve your goals, keeping in mind that the EAP CLI is the preferred option in production environments.

2. Briefly review the domain controller settings on the workstation.

Review the `/opt/domain/configuration/host-master.xml` file and the `/opt/domain/configuration/domain.xml` file to understand how the managed domain is configured. Verify that there are no server groups defined in the `domain.xml` file.

3. Start the domain controller on workstation. Because the files domain controller configuration are kept in the `/opt/domain` folder on workstation, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the domain controller is named `host-master.xml` and is located in the `/opt/domain/configuration` folder. (Tip: send the `--host-config=host master.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so you must start the domain controller using `sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh ...`

```
[student@workstation bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \
```

```
-Djboss.domain.base.dir=/opt/domain/ --host-config=host-master.xml
```

4. The two host controllers on `servera` and `serverb` connect to the host controller domain in the previous step and get the latest configuration of the domain. Start the two host controllers on `servera` and `serverb`.

- 4.1. Briefly review the host controller configuration on `servera` and `serverb`. Review the `/opt/domain/configuration/host-slave.xml` file to understand how hosts are configured. Verify that there are no servers defined on either host.
- 4.2. Start the host controller on `servera`. Because the host controller configuration files are kept in the `/opt/domain` folder on `servera`, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the host controller is named `host-slave.xml` and is located in the `/opt/domain/configuration` folder. (Hint: Pass the `--host config=host-slave.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so start the host controller using `sudo -u jboss /opt/jboss/eap-7.0/bin/domain.sh ...`

Open a new terminal window on the server virtual machine and run the following command:

```
[student@servera bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \  
-Djboss.domain.base.dir=/opt/domain/ \  
-Djboss.domain.master.address=172.25.250.254 \ --host-  
config=host-slave.xml
```

- 4.3. Start the host controller on `serverb`. Because the host controller configuration files are kept in the `/opt/domain` folder on `serverb`, use `/opt/domain` as the value of the `jboss.domain.base.dir` argument that you pass to the `domain.sh` startup script. Also notice that the host file for the host controller is named `host-slave.xml` and is located in the `/opt/domain/configuration` folder. (Hint: Pass the `--host config=host-slave.xml` argument to `domain.sh`.)

Note that the `/opt/domain` directory is owned by the `jboss` user, so start the host controller using `sudo -u jboss /opt/jboss/eap-7.0/bin/domain.sh ...`

Open a new terminal window on the `serverb` virtual machine and run the following command:

```
[student@serverb bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \  
-Djboss.domain.base.dir=/opt/domain/ \  
-Djboss.domain.master.address=172.25.250.254 \ --host-  
config=host-slave.xml
```

- 4.4. Verify that both host controllers connect to the domain controller and form a managed domain. Look at the console window you started in

Chapter 5. Configuring servers in a managed domain

the domain controller and verify that both `servera` and `serverb` are registered as slaves to the domain controller.

5. Access the JBoss EAP CLI. In a new terminal window on workstation, start the EAP CLI and connect to the domain controller as the `jboss` user:

```
[student@workstation bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/jboss-cli.sh \
--connect --controller=172.25.250.254:9990
```

6. By default, there are no server groups or servers defined in the managed domain. (These were removed in the final lab work of Chapter 4.) To deploy the bookstore application (WAR file), you must create server groups and define servers.

Create two new server groups named `Group1` and `Group2` in the managed domain.

- 6.1. The bookstore application uses the messaging subsystem and high availability capabilities of EAP, and these features are only available in the `full-ha` profile. Create a new server group named `Group1`, which is used to emulate a development environment with the following characteristics:

- Name: `Group1`
- Profile: `full-ha`
- Socket binding group: `full-ha-sockets`

```
[domain@172.25.250.254:9990 /] /server-group=Group1:add\ (profile=full-
ha,socket-binding-group=full-ha-sockets) {

    "outcome" => "success", "result"
    => undefined, "server-groups"
    => undefined
}
```

- 6.2. Create another server group named `Group2` to deploy bookstore in a production environment with the same characteristics as the development environment:

- Name: `Group2`
- Profile: `full-ha`
- Socket binding group: `full-ha-sockets`

```
[domain@172.25.250.254:9990 /] /server-group=Group2:add\ (profile=full-
ha,socket-binding-group=full-ha-sockets) {

    "outcome" => "success", "result"
    => undefined, "server-groups"
    => undefined
}
```


6.3. Verify the server groups that you created in the previous steps using the JBoss EAP CLI or management console.

```
[domain@172.25.250.254:9990 /] /server-group=Group1:read-resource {

  "outcome" => "success",
  "result" =>
    { "profile" => "full-ha", "socket-
      binding-group" => "full-ha-sockets",
      ...
    }
}
```

```
[domain@172.25.250.254:9990 /] /server-group=Group2:read-resource {

  "outcome" => "success",
  "result" =>
    { "profile" => "full-ha", "socket-
      binding-group" => "full-ha-sockets",
      ...
    }
}
```

7. You must define four servers (two each in servera and serverb). Create the servers and assign them to the appropriate groups. When running multiple servers on one host, be sure to set the port offsets correctly to avoid port clashing. Also make sure that the servers are configured to start automatically when the host controller starts or restarts.

7.1. Create a new server named servera.1 in servera with the following characteristics:

- **Name:** servera.1
- **Server Group:** Group1
- **Socket Binding Port Offset:** Leave the default value of 0
- **Autostart:** true

```
[domain@172.25.250.254:9990 /] /host=servera/server-config=servera.1:add\ (auto-
start=true,group=Group1,socket-binding-port-offset=0) {

  "outcome" => "success",
  "result" => {
    ....
  }
}
```

7.2. Create a new server named servera.2 on servera with the following characteristics:

- **Name:** servera.2

Chapter 5. Configuring servers in a managed domain

- **Server Group: Group2**
- **Offset of socket binding ports: 100**
- **Autostart: true**

```
[domain@172.25.250.254:9990 /] /host=servera/server-config=servera.2:add\ (auto-
start=true,group=Group2,socket-binding-port-offset=100) {

    "outcome" => "success",
    "result" => {
        ....
    }
}
```

7.3. Create a new server named serverb.1 in serverb with the following characteristics:

- **Name: serverb.1**
- **Server Group: Group1**
- **Socket Binding Port Offset: Leave the default value of 0**
- **Autostart: true**

```
[domain@172.25.250.254:9990 /] /host=serverb/server-config=serverb.1:add\ (auto-
start=true,group=Group1,socket-binding-port-offset=0) {

    "outcome" => "success",
    "result" => {
        ....
    }
}
```

7.4. Create a new server named serverb.2 in serverb with the following characteristics:

- **Name: serverb.2**
- **Server Group: Group2**
- **Offset of socket binding ports: 100**
- **Autostart: true**

```
[domain@172.25.250.254:9990 /] /host=serverb/server-config=serverb.2:add\ (auto-
start=true,group=Group2,socket-binding-port-offset=100) {

    "outcome" => "success",
    "result" => {
        ....
    }
}
```

- 8. You have now created four servers and associated them with the server groups suitable. It will now start the newly created servers. Instead of starting servers one by one, you can start all servers together in a server group.**

8.1. Start the servers in Group1

```
[domain@172.25.250.254:9990 /] /server-group=Group1:\ start-
servers(blocking=true) {

    "outcome" => "success",
    ...
}
```

8.2. Start the servers in Group2

```
[domain@172.25.250.254:9990 /] /server-group=Group2:\ start-
servers(blocking=true) {

    "outcome" => "success",
    ...
}
```

- 8.3. Verify that the servers have started successfully by looking at the console windows in which you started the host controllers. Verify that there are no port clashes and that the ports are bound according to the port offset values defined in the previous steps.**

In the servera virtual machine, a successful start of the servers should look like this:

```
[Host Controller] 03:27:22,221 INFO [org.jboss.as.host.controller]
(Controller Boot Thread) WFLYHC0148: Connected to master host controller at remote://
172.25.250.254:9999
[Host Controller] 03:27:22,252 INFO [org.jboss.as.host.controller] (Controller Boot Thread) WFLYHC0023:
Starting server servera.1 03:27:22,309 INFO
[org.jboss.as.process.Server:servera.1.status]
(ProcessController-threads - 3) WFLYPC0018: Starting process 'Server:servera.1'
[Server:servera.1] 03:27:22,957 INFO [org.jboss.modules] (main) JBoss Modules
version 1.5.1.Final-redhat-1
[Server:servera.1] 03:27:23,170 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final-redhat-1

[Server:servera.1] 03:27:23,246 INFO [org.jboss.as] (MSC service thread 1-1)
WFLYSRV0049: JBoss EAP 7.0.0.GA (WildFly Core 2.1.2.Final-redhat-1) starting [Server:servera.1]
03:27:23,332 INFO [org.xnio] (MSC service thread 1-4) XNIO version 3.3.6.Final-redhat-1

[Server:servera.1] 03:27:23,337 INFO [org.xnio.nio] (MSC service thread 1-4)
XNIO NIO Implementation Version 3.3.6.Final-redhat-1 [Server:servera.1]
03:27:23,364 INFO [org.jboss.remoting] (MSC service thread 1-4) JBoss Remoting version 4.0.18.Final-
redhat-1

[Host Controller] 03:27:23,801 INFO [org.jboss.as.host.controller] (management task-2)
WFLYHC0021: Server [Server:servera.1] connected using connection [Channel ID 15ed583d
(inbound) of Remoting connection 146a0d36 to /127.0.0.1:58069]

[Host Controller] 03:27:23,854 INFO [org.jboss.as.host.controller] (Controller Boot Thread) WFLYHC0023:
Starting server servera.2
```

Chapter 5. Configuring servers in a managed domain

```
[Host Controller] 03:27:23,885 INFO [org.jboss.as.host.controller] (server registration-threads - 1)
WFLYHC0020: Registering server servera.1 03:27:23,929 INFO
[org.jboss.as.process.Server:servera.2.status]
(ProcessController-threads - 3) WFLYPC0018: Starting process 'Server:servera.2'

[Server:servera.2] 03:27:24,974 INFO [org.jboss.modules] (main) JBoss Modules version 1.5.1.Final-
redhat-1
[Server:servera.2] 03:27:25,308 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final-redhat-1

[Server:servera.2] 03:27:25,429 INFO [org.jboss.as] (MSC service thread 1-4)
WFLYSRV0049: JBoss EAP 7.0.0.GA (WildFly Core 2.1.2.Final-redhat-1) starting
...
```

In the serverb virtual machine, a successful start of the servers should look like this:

```
Host Controller] 03:30:30,534 INFO [org.jboss.as.host.controller]
(Controller Boot Thread) WFLYHC0148: Connected to master host controller at remote://
172.25.250.254:9999
[Host Controller] 03:30:30,562 INFO [org.jboss.as.host.controller] (Controller Boot Thread) WFLYHC0023:
Starting server serverb.1 03:30:30,637 INFO
[org.jboss.as.process.Server:serverb.1.status]
(ProcessController-threads - 3) WFLYPC0018: Starting process 'Server:serverb.1'
[Server:serverb.1] 03:30:31,111 INFO [org.jboss.modules] (main) JBoss Modules version 1.5.1.Final-
redhat-1
[Server:serverb.1] 03:30:31,307 INFO [org.jboss.msc] (main) JBoss MSC version
1.2.6.Final-redhat-1
[Server:serverb.1] 03:30:31,383 INFO [org.jboss.as] (MSC service thread 1-1)
WFLYSRV0049: JBoss EAP 7.0.0.GA (WildFly Core 2.1.2.Final-redhat-1) starting [Server:serverb.1]
03:30:31,453 INFO [org.xnio] (MSC service thread 1-2) XNIO version 3.3.6.Final-redhat-1

[Server:serverb.1] 03:30:31,458 INFO [org.xnio.nio] (MSC service thread 1-2)
XNIO NIO Implementation Version 3.3.6.Final-redhat-1 [Server:serverb.1]
03:30:31,479 INFO [org.jboss.remoting] (MSC service thread 1-2) JBoss Remoting version 4.0.18.Final-
redhat-1

[Host Controller] 03:30:31,878 INFO [org.jboss.as.host.controller] (management task-2)
WFLYHC0021: Server [Server:serverb.1] connected using connection [Channel ID 02443242
(inbound) of Remoting connection 146a0d36 to /127.0.0.1:34993]

[Host Controller] 03:30:31,905 INFO [org.jboss.as.host.controller] (Controller Boot Thread) WFLYHC0023:
Starting server serverb.2 [Host Controller] 03:30:31,928 INFO
[org.jboss.as.host.controller] (server registration-threads - 1) WFLYHC0020: Registering server
serverb.1

03:30:31,980 INFO [org.jboss.as.process.Server:serverb.2.status]
(ProcessController-threads - 3) WFLYPC0018: Starting process 'Server:serverb.2'
[Server:serverb.2] 03:30:32,830 INFO [org.jboss.modules] (main) JBoss Modules version 1.5.1.Final-
redhat-1
[Server:serverb.2] 03:30:33,260 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final-redhat-1

[Server:serverb.2] 03:30:33,401 INFO [org.jboss.as] (MSC service thread 1-4)
WFLYSRV0049: JBoss EAP 7.0.0.GA (WildFly Core 2.1.2.Final-redhat-1) starting
...
```

8.4. Because there is a firewall running on servera and serverb, you should unblock all ports used by servera.1, servera.2, serverb.1 and serverb.2 for public access. To unlock them, run the following command in a new terminal window on serverA:

```
[student@servera ~]$ sudo firewall-cmd --zone=public --add-port 8080/tcp \ --permanent
[student@servera ~]$ sudo firewall-cmd --zone=public --add-port 8180/tcp \ --permanent [student@servera ~]$ sudo
firewall-cmd
--reload
```

8.5. To unblock the firewall ports on serverb, run the following commands in a new terminal window on serverb:

```
[student@serverb ~]$ sudo firewall-cmd --zone=public --add-port 8080/tcp \ --permanent
[student@serverb ~]$ sudo firewall-cmd --zone=public --add-port 8180/tcp \ --permanent [student@serverb ~]$ sudo
firewall-cmd
--reload
```

8.6. Verify that you can access the default welcome page for all four servers that you created in the previous steps. Go to the following URLs and verify that you can see the EAP Welcome page:

- **serve.1:** <http://172.25.250.10:8080>
- **serve.2:** <http://172.25.250.10:8180>
- **serverb.1:** <http://172.25.250.11:8080>
- **serverb.2:** <http://172.25.250.11:8180>

9. You are now ready to deploy the bookstore application WAR file to the managed domain. The bookstore.war file is available in the workstation /tmp folder.

9.1. Application deployments to a managed domain are always done at the server group level. Each application belongs to one or more server groups. Deploy the bookstore.war file to Group1.

```
[domain@172.25.250.254:9990 /] deploy \ /tmp/
bookstore.war --server-groups=Group1
```

Verify that the bookstore.war file is deployed to servera.1 and serverb.1 as they are part of the Group1 server group. Monitor the console window of servera and serverb for any error messages or warnings.

9.2. Verify that you can access the bookstore application at <http://172.25.250.10:8080/bookstore> y <http://172.25.250.11:8080/bookstore>.

9.3. Verify that you can NOT access the bookstore application at <http://172.25.250.10:8180/bookstore> or <http://172.25.250.11:8180/bookstore> because these servers are part of Group2 and you did not deploy the application to Group2 .

Chapter 5. Configuring servers in a managed domain

- 10. Using the JBoss EAP administrative console or CLI, undeploy the bookstore application and power off the servers, pools, host controllers, and the entire managed domain.**

10.1. Stop servera.1.

```
[domain@172.25.250.254:9990 /] /host=servera/server-config=servera.1:stop {
    "outcome" => "success",
    "result" => "STOPPING"
}
```

- 10.2. Verify that the bookstore application is no longer accessible from servera.1, but that is still accessible from serverb.1.**

Verify that the bookstore application is NOT accessible at the URL <http://172.25.250.10:8080/bookstore>. The bookstore application should still be accessible at <http://172.25.250.11:8080/bookstore>, since serverb.1 is still running.

- 10.3. Undeploy the bookstore application.**

```
[domain@172.25.250.254:9990 /] undeploy bookstore.war \ --all-relevant-
server-groups
```

- 10.4. Stop all servers in Group1. Verify that the bookstore app is no longer be accessible.**

```
[domain@172.25.250.254:9990 /] /server-group=Group1:stop-servers {
    "outcome" => "success",
    "result" => undefined, "server-
groups" => undefined
}
```

- 10.5. Stop the servers in Group2.**

```
[domain@172.25.250.254:9990 /] /server-group=Group2:stop-servers {
    "outcome" => "success",
    "result" => undefined, "server-
groups" => undefined
}
```

- 10.6. Shut down the host controller on servera. Observe the servera console window and verify that the host controller has been shut down.**

```
[domain@172.25.250.254:9990 /] /host=servera:shutdown {
    "outcome" => "success",
    "result" => undefined
}
```

10.7. Shut down the host controller in serverb. Look at the serverb console window and verify that the host controller has been shut down.

```
[domain@172.25.250.254:9990 /] /host=serverb:shutdown {  
  
    "outcome" => "success", "result"  
    => undefined  
}
```

11. Perform cleaning and grading.

11.1. Press Ctrl+C to stop the domain controller. (Alternatively, you can shut down the domain controller using the JBoss EAP CLI.)

11.2 Run the following workstation command to grade the assignment:

```
[student@workstation bin]$ lab managed-domain-servers grade
```

This concludes the lab work.

Summary

In this chapter, you learned the following:

- An EAP managed domain consists of a domain controller, host controllers, server groups, and servers.
- Server groups are configured at the domain controller level in `domain.xml`, while servers are configured at the host controller level in `host.xml`.
- A managed domain can have any number of servers and user groups. servers. A server belongs to a single server group.
- A host can be configured to run any number of servers, as long as it has the hardware capacity to run the server instances.
- Applications (EAR, WAR, JAR files, etc.) are always deployed in groups of servers.
- Server groups and servers can be configured and managed from the EAP management console, as well as from the EAP CLI. The underlying `domain.xml` and `host.xml` files are automatically updated and the tools reflect the changes immediately.
- If multiple servers are configured to run on a single host, they must be used. port offsets to avoid network port collisions.
- A server group is associated with a profile, which defines the subsystems, the configurations and network sockets to be used by applications assigned to a group of servers.