Lab 03_01: Domain Controllers

Performance Checklist

Lab Overview:

In this exercise, you will provision and configure a Host Controller that runs as a Domain Controller.

Lab Resources/Configuration:

Lab Files Location:	n/a
Application URL:	`http://192.168.0.xx:9990/`

Success Criteria: After completing this exercise, you will be able to login to the Management Console of your Domain Controller.

Outcome: A Domain Controller deployed and running as if it were on its own separate machine.

Lab Outline:

1. Create a New Domain Base Directory
2. Configure machine1 as the Domain Controller
3. Start the Domain Controller
4. Verify the Domain Controller is Running
5. Navigating the Management Console in Domain Mode

Before you begin...

Stop any running instances of EAP.

□ 1. Create a New Domain Base Directory

□ 1.1. In your ***STUDENT_HOME*/opt** folder, create a new folder named **machine1**.

□ 1.2. Copy the folder ***EAP_HOME*/domain** into your **machine1** folder. This will create a folder **machine1/domain**, which should contain three subfolders: **configuration**, **data** and **tmp**.

Note

Eventually (after you complete Lab 03_02) you are going to have a Domain Controller and two Host Controllers all running on your student machine. In reality, you probably would run these three controllers on separate machines, so we are going to simulate separate machines by using subfolders named **machine1**, **machine2** and **machine3**. In this lab, you are going to configure **machine1** to run as the master controller. In the next lab, you will create and configure **machine2** and **machine3** as slaves connecting to **machine1**.

□ 2. Configure machine1 as the Domain Controller

□ 2.1. Using a text editor, open the **host-master.xml** file in the **machine1/domain/configuration** folder. This host configuration file is specifically configured to run a Domain Controller that does not manage any local Servers.

□ 2.2. What is the name of this host? ________________

□ 2.3. Notice there is only one interface defined in **host-master.xml**, named **management**. Why is there not a public HTTP interface?

____

Insight

A host acting as the Domain Controller *must* expose a native (i.e. non-HTTP) management interface on an address that is accessible to the other hosts in the Domain. But if a Domain Controller is not going to have any Servers running on it, then it does not need an HTTP interface, since it will not be handling HTTP requests from clients.

□ 2.4. What specific setting in this **host-master.xml** file configures it as a master?

____

□ 2.5. Which security realm is the native management interface using?

____

□ 2.6. The labs are going to simulate multiple machines, and binding to 127.0.0.1 is not going to make the Domain Controller on machine1 visible to outside machines. We could specify the **jboss.bind.address.management** property at runtime, but instead you will manually edit the XML file. Modify the "management" interface's **inet-address** to bind to the IP address of your machine. The **<interfaces>** section of **host-master.xml** should look like the following (with 192.168.0.14 replaced with your machine's IP address):

```
    <interfaces>
        <interface name="management">
            <inet-address
value="${jboss.bind.address.management:192.168.0.14}"/>
        </interface>
    </interfaces>
```

□ 2.7. Save your changes to **host-master.xml** and close the text editor.

□ 2.8. Open the **domain.xml** in **machine1/domain/configuration**.

□ 2.9. Inside the messaging subsystem of the full-ha profile (on line 1,091) after the **<clustered>true</clustered>** element, add the following tag:

```
<cluster-password>jboss</cluster-password>
```

Insight

Why add this? Because messaging uses a separate clustering system, outside of JGroups, with it's own security settings. If clustering is enabled, a password is expected. To avoid bogus errors, we are adding a password now.

☐ 2.10. Save your changes to domain.xml.

☐ 3. Start the Domain Controller

☐ 3.1. Open a terminal window and change directories to your ***EAP_HOME*/bin** folder.

☐ 3.2. To start machine1 in Domain mode using the **host-master.xml** file in your **opt/machine1/domain** folder, enter the following command. On RHEL:

```
./domain.sh --host-config=host-master.xml -Djboss.domain.base.dir=/home/
student/JB248/opt/machine1/domain/
```

On Windows:

```
domain.bat --host-config=host-master.xml -Djboss.domain.base.dir=c:
\JB248\opt\machine1\domain\
```

Since you are only starting a master controller (and no actual EAP Servers), it should start up fairly quickly!

Insight

If you do not specify a **host-config** property, then **host.xml** is used by default. In our example, we want to use **host-master.xml**, so the **host-config** property is necessary. If you do not specify a **--domain-config** property, then **domain.xml** is used for the Domain configuration file.

Important

You copied the **`/domain`** folder from a clean installation of EAP 6 before you had ever started EAP in domain mode. This is considered a best practice, as it separates your unique configurations, deployments, log files and temporary folders from the out-of-the-box installation folders. There are two key benefits gained from this practice:

1. You can run multiple instances of EAP in Domain mode on the same machine using the same installation files.
2. You can upgrade to a newer version of EAP without affecting or overwriting your configuration.

It is the **`jboss.domain.base.dir`** runtime property that makes this best practice possible.

□ 4. Verify the Domain Controller is Running

□ 4.1. Point your web browser to `http://192.168.0.`*`xx`*`:9990/`, replacing *xx* with your actual IP address.

□ 4.2. You will be prompted for credentials - use **`admin/jboss`**. How come the username and password combination of **`admin/jboss`** works?

____

□ 4.3. On the **Runtime** page, notice there are no Server instances running. Why?

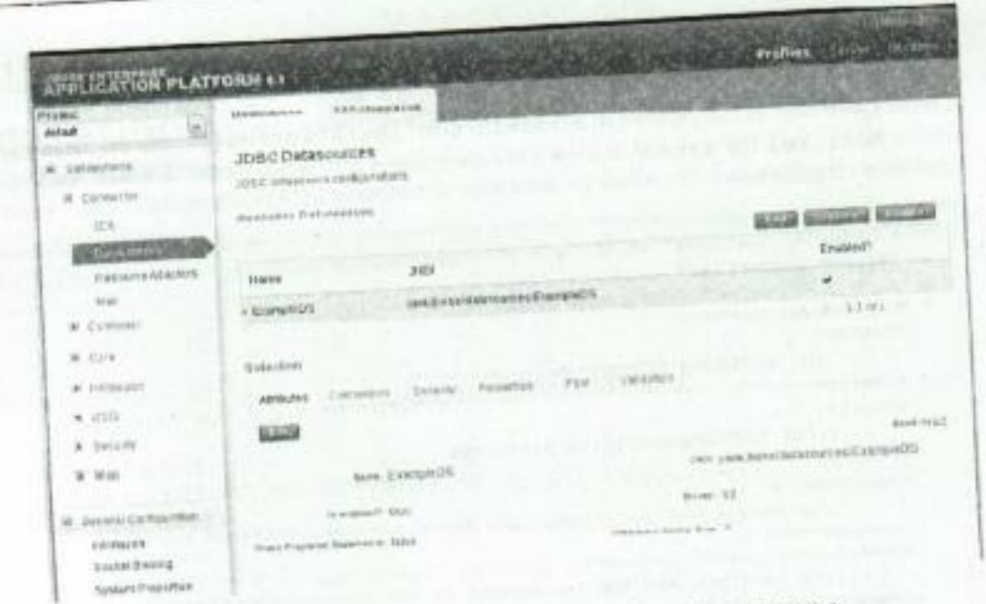
____

□ 5. Navigating the Management Console in Domain Mode

□ 5.1. The Management Console is a little different in Domain mode compared to Standalone mode. For example, the Standalone mode Management Console had two main pages: **Profiles** and **Runtime**. Notice that in Domain mode there is an additional page called **Server**, which is used for defining and configuring Servers instances. You will learn how to do that in the next Unit.

□ 5.2. Note that since there are no Servers running right now (because you have not defined and started any yet), the **Runtime** page does not have any useful information to display yet because it shows the runtime information of all of the Servers in your Domain.

□ 5.3. Click on the **Profiles** page. You saw some of the details of profiles when you had a Standalone server running. Notice this page looks similar to Standalone mode:



☐ 5.4. Notice in the upper-left corner of the **Profiles** page is a drop-down list labeled **Profile:**. The drop-down list contains all of the profiles defined in your **`domain.xml`** configuration file.

Note

Recall that in Domain mode you can have multiple profiles defined, and each profile has a unique name. In Standalone. there is only profile available so it does not have a name.

☐ 5.5. Stop here! You are done with this hands-on exercise and your Domain Controller is now up and running! In the next lab you will define two slaves that are managed by this Domain Controller.

Lab 03_02: Host Controllers

Performance Checklist

Lab Overview:

In this exercise, you will provision two Host Controllers that are slaves to the Domain Controller you started in the previous Lab.

Lab Resources/Configuration:

Lab Files Location:	n/a
Application URL:	`http://192.168.0.xx:9990/`

Success Criteria: After completing this exercise, you should see three hosts running on your machine, with one of them acting as the master.

Outcome: Three Host Controllers running on your student machine.

Lab Outline:

1. Simulate Two More Machines
2. Configure a Distinct Host Name
3. Specify the IP Address of the Domain Controller
4. Expose a Native Management Interface
5. Configure the Host's Servers
6. Start host2
7. Verify host2 is in the Domain
8. Configure the machine3 Host Controller
9. Start host3
10. Verify host3 is in the Domain
11. Verify the Servers are Running
12. Stop a Slave Host
13. Stop the Master Host

Before you begin...

The Domain Controller provisioned in the previous lab should be running.

□ 1. Simulate Two More Machines

□ 1.1. In your ***STUDENT_HOME*/opt** folder, create two new folders: **machine2** and **machine3**.

□ 1.2. Copy the folder ***EAP_HOME*/domain** into both the **machine2** folder and the **machine3** folder. These two folders symbolize two separate machines that will connect to your Domain Controller.

□ 2. Configure a Distinct Host Name

□ 2.1. EAP ships with a version of **host.xml** named **host-slave.xml** that is configured as a slave. Using a text editor, open the **host-slave.xml** file in the **machine2/domain/configuration** folder.

□ 2.2. Notice this host does not have a name. Each host in a Domain needs a distinct name, so add the **name** attribute to the **<host>** element and assign it to "**host2**", as follows:

```
<host name="host2" xmlns="urn:jboss:domain:1.3">
```

□ 3. Specify the IP Address of the Domain Controller

□ 3.1. Notice that **host-slave.xml** is configured to be a slave. What specific setting makes this host a slave? ______________________________

□ 3.2. What runtime property do you set to specify the master's IP address that this host should connect to? ______________________________

Note

You will specify this property when you start **host2** in Domain mode later in the lab.

□ 4. Expose a Native Management Interface

□ 4.1. Each slave must expose a native (non-HTTP) management interface so the master can communicate with it. Study the **host-slave.xml** file of **host2**. Does it define a native management interface? ____________________

□ 4.2. What port does this host's native management interface listen on by default?

____

□ 4.3. Because the Domain Controller from the previous lab is running and already bound to port **9999**, you need to change the native interface's port number for **host2**. Within the **<native-interface>** section, change the default port to **20000**:

```
<native-interface security-realm="ManagementRealm">
     <socket interface="management"
 port="${jboss.management.native.port:29999}"/>
</native-interface>
```

□ 4.4. Notice in the **<interfaces>** section that the IP addresses are 127.0.0.1 by default. This will not work for **jboss.bind.address.management** in a multi-machine environment, because the master will not be able to connect to this host. And it will not work for **jboss.bind.address** if you want your Servers to be accessible to the outside world. Replace the **127.0.0.1** in both of the default values with the IP address of your machine:

```
<interface name="management">
     <inet-address value="${jboss.bind.address.management:192.168.0.xx}"/>
</interface>
<interface name="public">
    <inet-address value="${jboss.bind.address:192.168.0.xx}"/>
</interface>
```

□ 5. Configure the Host's Servers

□ 5.1. In the **host-slave.xml** for **host2**, how many servers are defined?

____

□ 5.2. What server group does **server-one** belong to?

____

□ 5.3. What server group does **server-two** belong to?

____

□ 5.4. The **port-offset** of **server-two** is 150. What port will the HTTP interface be listening on for **server-two**? ____________________

□ 5.5. You can now save your changes to **host-slave.xml** and close your text editor. You are ready to start your Host Controller.

□ 6. Start host2

□ 6.1. Run the following command from your ***EAP_HOME*/bin** folder to start **host2** in Domain mode using the **host-slave.xml** configuration file and having its management interface bind to 192.168.0.*xx* on port **29999** (replacing the *xx* with your machine's actual IP address):

Enter the following command on RHEL all on a single line:

```
./domain.sh
    -Djboss.domain.base.dir=/home/student/JB248/opt/machine2/domain/
    --host-config=host-slave.xml
    -Djboss.domain.master.address=192.168.0.xx
```

On Windows, enter the following command on a single line:

```
domain.bat
    -Djboss.domain.base.dir=c:\JB248\opt\machine2\domain\
    --host-config=host-slave.xml
    -Djboss.domain.master.address=192.168.0.xx
```

> **Note**
>
> Running the **domain.sh** (or **domain.bat**) script starts three processes: the Host Controller, **server-one**, and **server-two**. Also notice that the prefix of each log entry in the terminal window is either [HostController] or the name of the server that caused the log event, which is either [Server:server-one] or [Server:server-two] in your deployment.

□ 6.2. Look in the terminal window of the Host Controller of **machine2**. Looking closely at the log output - you should see the Host Controller connecting to the master, and also **server-one** and **server-two** starting up.

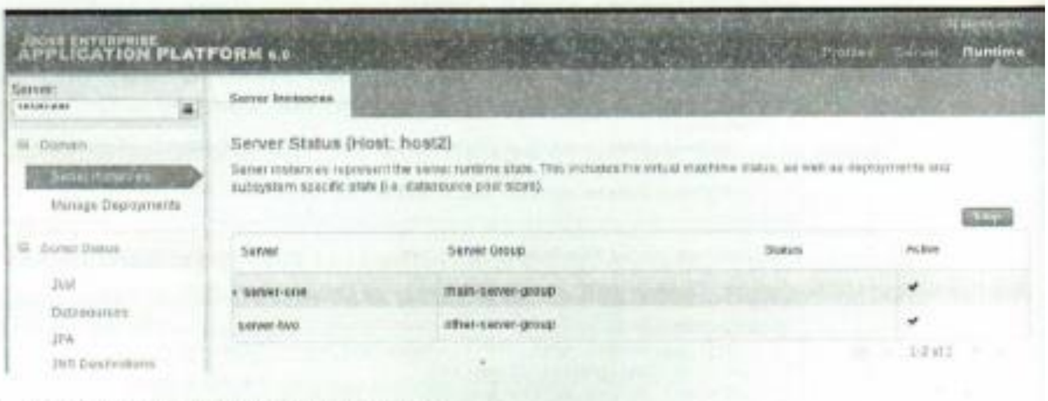
□ 6.3. Look in the terminal window of the Domain Controller. You should a log entry showing the slave connecting:

```
[Host Controller] 20:33:26,176 INFO  [org.jboss.as.domain] (domain-mgmt-
handler-thread -1) JBAS010918: Registered remote slave host "host2", JBoss
 EAP 6.0.0 (AS 7.1.1.Final-redhat-1)
```

□ 7. Verify host2 is in the Domain

□ 7.1. Point your browser to `http://192.168.0.xx:9990/`, which is the URL for the Domain Controller's management tool.

□ 7.2. Verify that **server-one** and **server-two** are displayed in the list of **Server Instances** on the **Runtime** page. (If necessary, refresh the **Runtime** page in your web browser.)



Server Status (Host: host2)

Server	Server Group	Status	Active
server-one	main-server-group		✓
server-two	other-server-group		✓

1-2 of 2

□ 8. Configure the machine3 Host Controller.

□ 8.1. Using a text editor, open the file **host-slave.xml** in your **machine3/domain/configuration** folder.

□ 8.2. Add a name attribute to the **<host>** element and assign it the value **host3**:

```
<host name="host3" xmlns="urn:jboss:domain:1.3">
```

- □ 8.3. Change the native management interface port to **39999**.
- □ 8.4. In the **<interfaces>** section, replace 127.0.0.1 in the default value of **jboss.bind.address.management** and **jboss.bind.address** with the IP address of your machine.
- □ 8.5. Delete **server-one** from the **<servers>** section.
- □ 8.6. Rename **server-two** to **server-three** and change its **port-offset** to be **1000**. Your **<servers>** section should now look like:

```
<servers>
    <server name="server-three" group="other-server-group">
        <socket-bindings port-offset="1000"/>
    </server>
</servers>
```

- □ 8.7. Save your changes to **host-slave.xml** and exit your text editor.

□ 9. Start host3

- □ 9.1. Run the following command from your **EAP_HOME/bin** folder to start **host3** as a slave in your Domain:

On RHEL:

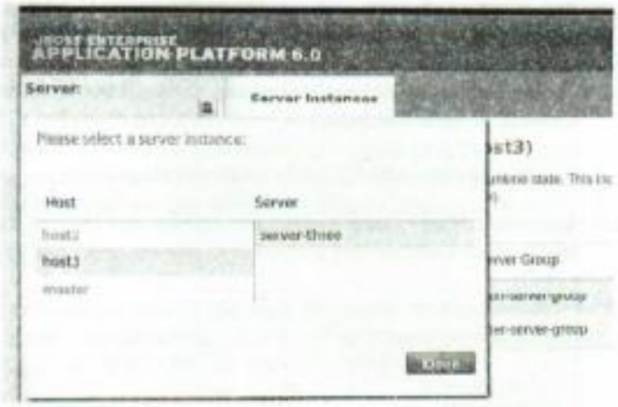
```
./domain.sh
    -Djboss.domain.base.dir=/home/student/JB248/opt/machine3/domain/
    --host-config=host-slave.xml
    -Djboss.domain.master.address=192.168.0.xx
```

On Windows:

```
domain.bat
    -Djboss.domain.base.dir=c:\JB248\opt\machine2\domain\
    --host-config=host-slave.xml
    -Djboss.domain.master.address=192.168.0.xx
```

□ 10. Verify host3 is in the Domain

- □ 10.1. Go back to your Management Console and refresh the **Runtime** page.
- □ 10.2. In the upper-left corner of the page is a drop-down list labeled **Server:**. Click on the drop-down list, select **host3**, then click the **Done** button to view the servers running on **host3**.



JBOSS ENTERPRISE APPLICATION PLATFORM 6.0

Server:

Server Instances

Please select a server instance:

Host	Server
host2	server-three
host3	
master	

st3)

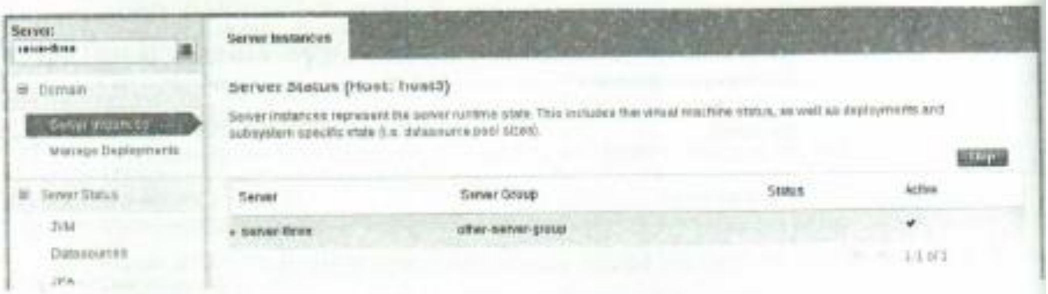
untime state. This inc

rver Group

n-server-group

er-server-group

□ 10.3. You should see **server-three** in the list of **Server Instances**, and **server-three** should be in the **other-server-group**.



Server: server-three

Domain

Server Instances

Manage Deployments

Server Status

JVM

Datasources

JPA

Server Instances

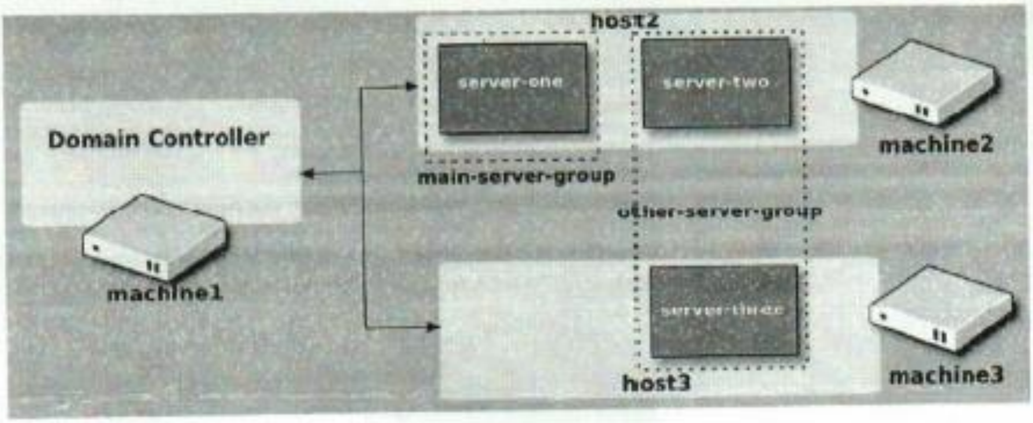
Server Status (Host: host3)

Server instances represent the server runtime state. This includes the virtual machine status, as well as deployments and subsystem specific state (i.e. datasource pool sizes).

Server	Server Group	Status	Active
server-three	other-server-group		✓

1-1 of 1

□ 10.4. The following diagram shows what your Domain now looks like:



□ 11. Verify the Servers are Running

□ 11.1. Point your web browser to `http://192.168.0.xx:8080/`. You should see the default EAP Welcome page. Which Server is displaying this web page?

____

□ 11.2. Point your web browser to `http://192.168.0.xx:8230/`. Again, you should see the default EAP Welcome page. Which Server is displaying this web page?

____

□ 11.3. What is the URL for accessing the Welcome page of **server-three**?

____

□ 11.4. Verify your answer to the previous question by entering the URL in your web browser.

□ 12. Stop a Slave Host

□ 12.1. Enter **Ctrl+c** in the terminal window of host2, which will start the shutdown of host2.

□ 12.2. Watch the output in the terminal window and notice that the **server-one** and **server-two** processes are stopped, followed by the Host Controller process.

□ 12.3. Look in the terminal window of the Domain Controller. You should see a log event similar to the following, stating that **host2** has been removed from the Domain:

```
[Host Controller] 22:26:33,209 INFO  [org.jboss.as.domain] (Remoting
 "instructor.example.com:MANAGEMENT" task-3) JBAS010925: Unregistered
 remote slave host "host2"
```

□ 12.4. Start **host2** again. Once it is started, you should see a log event in the **machine1** terminal window showing **host2** being registered again with the Domain.

□ 13. Stop the Master Host

□ 13.1. Enter **Ctrl+c** in the terminal window of **machine1**, which will shutdown your Domain Controller.

□ 13.2. Refresh the web page at `http://192.168.0.xx:8080/`. The page should display fine, even though the master controller is no longer running.

Insight

A Server is not affected by the presence of a Domain Controller once the Server has started. The slave controllers will simply attempt to reconnect to the master, and Servers continue to run without any loss of service.

□ 13.3. Start the Domain Controller back up again.

□ 13.4. Watch the output in the terminal window and wait. You should see **host2** and **host3** reconnect to the master after a few seconds.

□ 13.5. Leave your master controller and two slave hosts running, and continue on to the next Lab. You will deploy an application onto your Servers.

Lab 03_03: Application Deployment on a Domain

Lab Overview:

In this exercise, you will deploy an application onto the Servers in your Domain.

Lab Resources/Configuration:

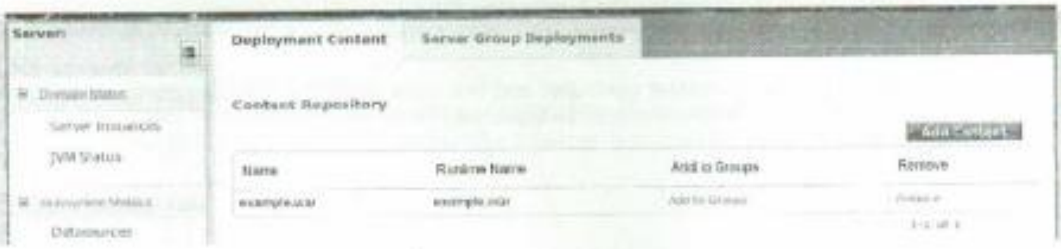
Lab Files Location:	n/a
Application URL:	`http://192.168.0.xx:8080/example` and `http://192.168.0.xx:9080/version`

Success Criteria: After completing this exercise, the **`example.war`** application will be deployed onto all three of your Servers.

Lab Outline:

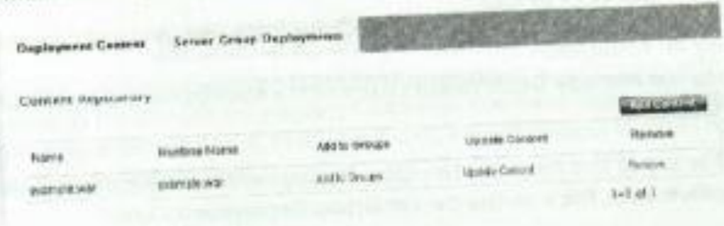
1. Deploy an Application to a Server Group
2. Verify the Deployment
3. Deploy another Application
4. Verify the Deployment
5. Undeploy an Application
6. Remove a Deployment

1. Deploy an Application to a Server Group
   1. Go to the **Runtime** page of the Management Console of the Domain Controller.
   2. Click on the **Manage Deployments** link in the **Domain** section to view the **Content Repository**.
   3. Click the **Add Content** button and upload the WAR file ***LABS*/`Lab03_03/example.war`**.
   4. Notice in the list of deployments next to **`example.war`** there is a link named "**Add to Groups**". Click this link:



Name	Runtime Name	Add to Groups	Remove
example.war	example.war	Add to Groups	Remove

5. We want **example.war** deployed to the main-server-group, so place a checkmark in the box next to main-server-group, then click the Save button:



Name	Runtime Name	Add to Groups	Update Content	Remove
example.war	example.war	Add to Groups	Update Content	Remove

1-1 of 1

6. Look in the terminal window of **host2**. You should see a log message stating that **example.war** was deployed onto **server-one**.

7. Look in the terminal window of **host3**. The application should not be deployed onto **host3**, because there are no Servers on **host3** in the **main-server-group**.

2. Verify the Deployment
   1. Point your browser to the example application on server-one, which is at `http://192.168.0.xx:8080/example`. You should see the "Welcome to EAP 6" page of the **example** application.

   2. Similarly, point your browser to **server-two**, which is `http://192.168.0.xx:8230/example/`. You should get a 404 error, because the **example** application is not deployed on **server-two**.

   3. Open the file **domain.xml** in the **machine1/domain/configuration** folder. Notice that **example.war** now appears 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.

3. Deploy another Application
   1. Go back to the **Manage Deployments** page of the Management Console of your Domain. Click the **Add Content** button.

   2. Click the **Choose File** button and upload the file ***LABS*/Lab03_03/version.war**. Click the **Next** button, then **Save** to add **version.war** to the **Content Repository**.

   3. Deploy **version.war** to the server group named **other-server-group**, which contains your **server-two** and **server-three** Server instances.

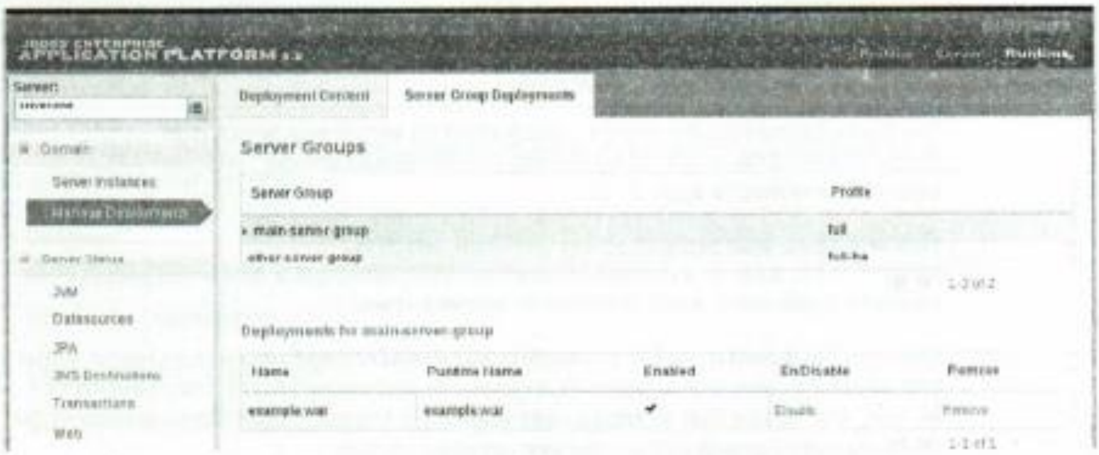
   4. Look in the terminal windows of both **host2** and **host3**. The **version.war** application should have been deployed onto **server-two** on **host2** and **server-three** on **host3**.

4. Verify the Deployment
   1. Point your browser to **server-three**, which is at `http://192.168.0.xx:9080/version/`. You should see the output of the Version App.

   2. Point your browser to **server-two**, which is at `http://192.168.0.xx:8230/version/`. The Version App should appear at this URL also.

   3. Point your browser to **`server-one`**, which is at `http://192.168.0.xx:8080/version/`. You should get a 404 error because **`server-one`** is not a member of the **`other-server-group`**.

5. Undeploy an Application
   1. Go to the **Manage Deployments** page of the Management Console for your Domain Controller.
   2. At the top of this page are two tabs: **Deployment Content** and **Server Group Deployments**. Click on the **Server Group Deployments** tab.
   3. Notice the deployed applications are listed by Server Group. Click on the **`main-server-group`** in the **Server Groups** list.



Server Groups

Server Group	Profile
▸ main-server-group	full
other-server-group	full-ha

1-2 of 2

Deployments for main-server-group

Name	Runtime Name	Enabled	En/Disable	Remove
example.war	example.war	✓	Disable	Remove

1-1 of 1

   4. Next to **`example.war`** in the list of **Deployments for main-server-group**, click the **Disable** link to disable the application. Click the **Confirm** button in the popup dialog window.
   5. View the **`example.war`** application on **`server-one`** in your browser and verify that you now get a 404 error.
   6. Click the **Remove** link now to remove **`example.war`** from the **`main-server-group`**.
   7. Following the same steps, disable and remove **`version.war`** from the **`other-server-group`**.
   8. The applications are now removed from your EAP Servers. To verify, try loading them in your web browser. You should get 404 errors.

6. Remove a Deployment

   In the upcoming labs, you are going to define different servers and server groups, and therefore you will not be needing the **`example.war`** and **`version.war`** applications deployed.

   1. Click on the **Deployment Content** tab of the Manage Deployments page.

2. Next to **example.war** in the **Content Repository** list, click the **Remove** link. Then click the **Confirm** button when the confirmation window pops up.

3. Similarly, remove **version.war** from the **Content Repository**.

4. STOP! In this Unit, you have deployed three Host Controllers in a Domain, with one of the Hosts acting as the Domain (or master) Controller. You have also seen how to deploy an application onto a Server Group. In the next Unit, we will discuss in detail how to define Server Groups and Servers.

Lab 04_01: Defining Server Groups

Performance Checklist

Lab Overview: In this exercise, you will use the Management Console to define new Server Groups.

Lab Resources/Configuration:

Lab Files Location:	n/a
Application URL:	`http://192.168.0.xx:9990/console/App.html#server-groups`

Success Criteria: After completing this exercise, you will have two new Server Groups: **`dev-group`** and **`production-group`**.

Lab Outline:

1. Review the Existing Socket Binding Group Definitions
2. Define a Server Group
3. Define a Second Server Group
4. Verify the Changes

Before you begin...

You should have your Domain Controller on **`machine1`** running, as well as your **`machine2`** and **`machine3`** Host Controllers.

□ 1. Review the Existing Socket Binding Group Definitions

□ 1.1. Go to the **Profiles** page of the Management Console of your Domain Controller.

□ 1.2. Click on the **Socket Binding** link in the **General Configuration** section.

□ 1.3. How many Socket Binding Groups are defined in your Domain? ____________

□ 1.4. What is the **`http`** port number for the **`standard-sockets`** group?

____

□ 1.5. What is the multicast port being used by the **`jgroups-mping`** service in the **`ha-sockets`** group? ____________

□ 1.6. To modify a port number, you select the binding from the list and click the **Edit** button. To demonstrate, click on the **`https`** socket binding for the **`ha-sockets`** group. Click the **Edit** button and change the port to **9443**. Click the **Save** button to commit your changes. You should see **9443** next to **`https`** now in the list of **Available Socket Bindings**.

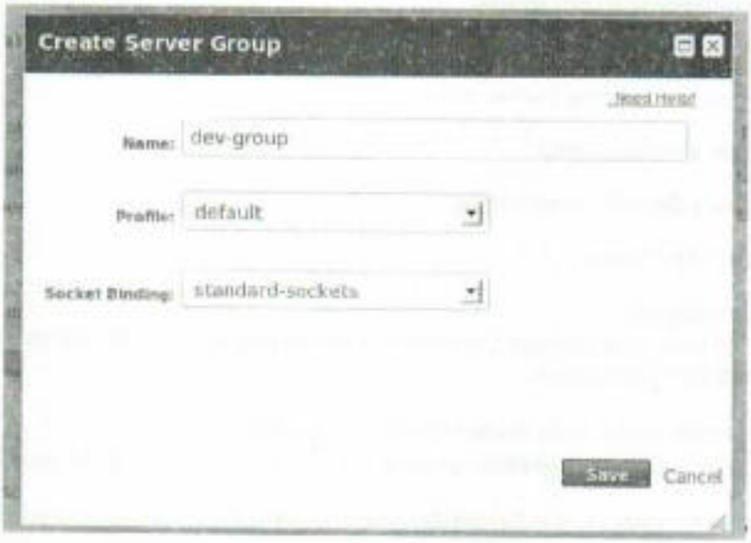
> Socket Binding Groups are defined at the Domain level in **domain.xml**. You need to know which Socket Binding Groups are defined in **domain.xml** because a Server Group definition needs to include a reference to one of your Socket Binding Groups. You will see how to do this in the next step.

□ 2. Define a Server Group

□ 2.1. On the **Server** page, click on the **Server Groups** link in the **Server** section. Notice there are two pre-defined Server Groups: **main-server-group** and **other-server-group**.

□ 2.2. Click the **Add** button.

□ 2.3. Enter **dev-group** for the **Name**, select **default** for the **Profile**, and select **standard-sockets** for the **Socket Binding**.



Create Server Group

Need Help?

Name: dev-group

Profile: default

Socket Binding: standard-sockets

□ 2.4. Click the **Save** button. You should now see **dev-group** in the list of **Available Group Configurations**.

□ 3. Define a Second Server Group

□ 3.1. Repeat the same steps to define a new Server Group named **production-group** that uses the **ha** profile and the **ha-sockets** socket binding group.

□ 4. Verify the Changes

□ 4.1. Using a text editor, open the file **domain.xml** in your **machine1/domain/configuration** folder.

□ 4.2. At the end of this file, verify that the definitions of **dev-group** and **production-group** appear in the **<server-groups>** section

□ 4.3. Also verify that **dev-group** uses the **standard-sockets** binding group, and **production-group** uses the **ha-sockets** binding group.

Note

You have defined two Server Groups, **dev-group** and **production-group**, to represent a development environment and a production environment, respectively. In the next lab, you will configure a single Server on **host2** in the **dev-group**, and you will also configure two Servers for the **production-group** - one Server on **host2** and another Server on **host3**. Continue on to the next Lab!

Lab 04_02: Defining Servers

Performance Checklist

Lab Overview:

In this exercise, you will define three new Servers.

Lab Resources/Configuration:

Lab Files Location:	LABS/Lab03_3
Application URL:	`http://192.168.0.xx:9990/console/App.html#server-instances`

Success Criteria: After completing this exercise, you will have three Servers spread across two Host Controllers.

Outcome: Three new Server instances.

Lab Outline:

1. Delete a Server
2. Delete a Server Group
3. Define a New Development Server in the **dev-server-group**
4. Define Two Servers in the **production-group**
5. Start the New Servers
6. Test the Server Configurations

□ 1. Delete a Server

The first step you will perform is to delete the existing servers on **host2** and **host3**.

□ 1.1. Before you can delete a Server definition, the Server has to be stopped. On the **Runtime** page of the Management Console, click on **server-one**, then click the **Stop** button to stop this server.

□ 1.2. Verify **server-one** has stopped by checking the output in the terminal window of **host2**.

□ 1.3. Click on the **Server** page of the Management Console and select **Server Configurations** from the list on the left navigation tree.

□ 1.4. Select **server-one** from the list of Servers and click the **Remove** button. The list of Servers should now only contain **server-two** for **host2**.

□ 1.5. Following the same steps, stop and delete **server-two** from **host2**.

□ 1.6. Similarly, stop and delete **server-three** from **host3**.

□ 2. Delete a Server Group

□ 2.1. On the **Server** page of the Management Console, click on the **Server Groups** link.

□ 2.2. You no longer have any Servers in the **main-server-group**, so you are going to delete it now. Click on **main-server-group** in the list of **Available Group Configurations** and click the **Remove** button. Click **Confirm** and the **main-server-group** should no longer appear in the list.

□ 2.3. Similarly, delete the Server Group named **other-server-group**.

□ 2.4. You should only have two Server Groups defined in your Domain: **dev-group** and **production-group**.

Note

Your environment now consists of three Host Controllers but no Server instances. You will define new Servers next.

□ 3. Define a New Development Server in the **dev-group**

□ 3.1. On the **Server Configurations** page, select **host2** from the **Host** drop-down list in the upper-left corner.

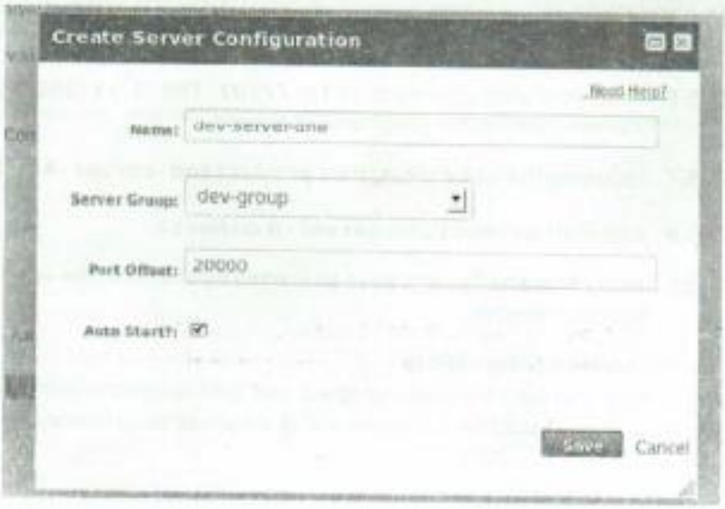
□ 3.2. The list of **Available Server Configurations** should be empty. Click the **Add** button to add a new one.

□ 3.3. Enter **dev-server-one** for the name of the Server.

□ 3.4. Select **dev-group** for the Server Group.

□ 3.5. Enter **20000** for the **port-offset**.

□ 3.6. Check the box for **Auto Start**. Your new Server should look like:



Create Server Configuration

Name: dev-server-one

Server Group: dev-group

Port Offset: 20000

Auto Start?: ☑

□ 3.7. Click the **Save** button. The Server will be created and you should see it in the **Available Server Configurations** list of **host2**.

Insight

Although you selected **Auto Start** for **dev-server-one**, it is not started yet. The **Auto Start** option is for denoting if the Server should start automatically when the Host Controller starts. When a new Server is defined, it needs to be started manually the first time, or you need to restart the Host Controller. You will start **dev-server-one** manually in just a moment.

□ 4. Define Two Servers in the **production-group**

□ 4.1. Performing the same steps that you used in defining **dev-server-one**, define a new Server on **host2** named **production-server-A** that is a member of **production-group**. Use a **port-offset** of **30000** and configure it to automatically start.

□ 4.2. Define a new Server on **host3** named **production-server-B** that is a member of the **production-group**. Use a **port-offset** of **40000** and configure it to automatically start.

□ 5. Start the New Servers

□ 5.1. Go to the **Runtime** page of the Management Console.

□ 5.2. Select **host2** from the drop-down list in the upper-left corner of the page.

□ 5.3. Click on **dev-server-one** in the list of Servers, then click the **Start** button.

□ 5.4. Click **Confirm** when the confirmation dialog pops up, and **dev-server-one** should now have a checkmark in the **Active** column.

□ 5.5. Look in the terminal window of **host2**. There should be log entries showing **dev-server-one** starting up.

□ 5.6. Point your web browser to `http://192.168.0.xx:28080/`. The default Welcome screen for EAP6 should appear.

□ 5.7. Following the same steps, start **production-server-A** on **host2**.

□ 5.8. Also start **production-server-B** on **host3**.

□ 5.9. Verify that the Servers have all started by checking the output in the respective terminal window.

□ 6. Test the Server Configurations

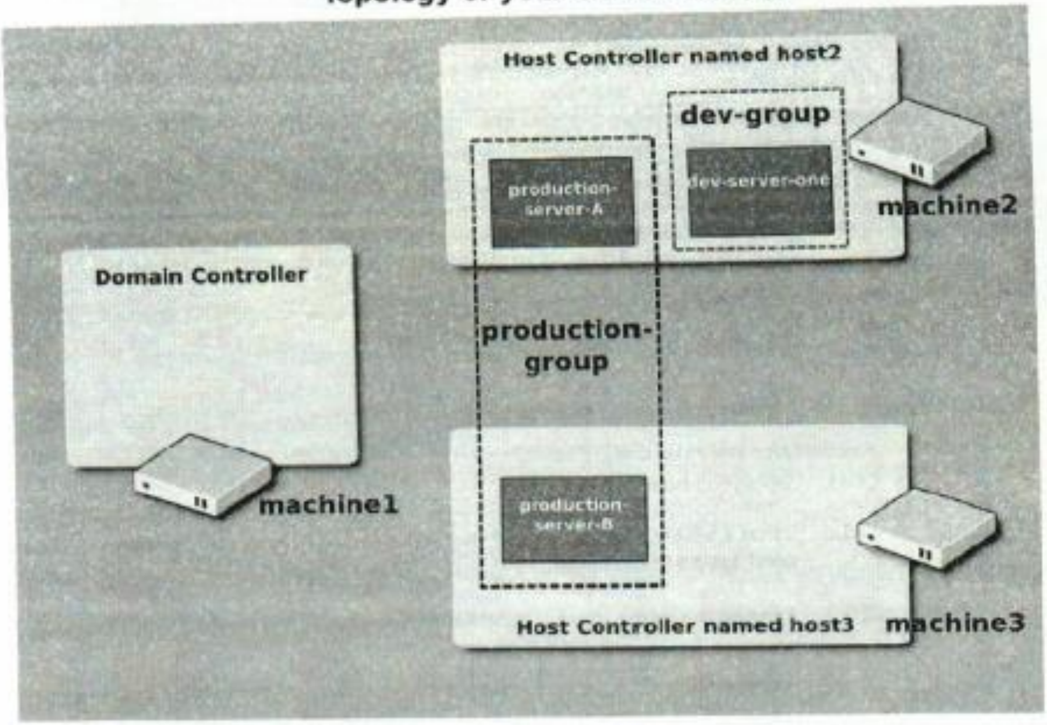
These tasks may be a bit challenging without detailed instructions, but performing the following steps should be a good review of what has been learned up to this point.

□ 6.1. Deploy **example.war** (found in the ***LABS/Lab03_03*** folder) onto the **dev-group** Server Group and verify it is deployed properly by viewing the application in your web browser.

□ 6.2. Deploy **version.war** (found in the ***LABS/Lab03_03*** folder) onto the **production-group** server group and verify it is deployed properly.

□ 6.3. The following diagram shows the Host view of your current setup:

Topology of your Current Setup



□ 7. STOP!

Remove the example.war and version.war deployments from the content repository.

Lab 04_03: Join a Remote Domain

Performance Checklist

Lab Overview: In this lab, you will have your Host Controller join a Domain running on the instructor's machine.

Lab Resources/Configuration:

Lab Files Location:	n/a
Application URL:	n/a

Success Criteria: Your Host Controller will be connected to the Domain Controller on another machine in the classroom.

Lab Outline:

1. Reconfigure the Host Controller

2. Start the Host Controller

3. Verify the Connection

□ 1. Reconfigure the Host Controller

□ 1.1. Stop your **host2** instance.

□ 1.2. Using a text editor, open the **host-slave.xml** file in your **machine2/domain/configuration** folder.

□ 1.3. Append the name of your machine to the end of the name of your host. For example if your machine is **station7**:

```
<host name="host2_station7" xmlns="urn:jboss:domain:1.3">
```

Insight

Every Host Controller in a Domain requires a unique name.

Important

The instructor needs to delete the **security-realm** attribute from the **<native-interface>** definition in **host-master.xml**, otherwise the Host Controllers will not be able to connect because they are not providing any credentials. We will discuss authenticating Host Controllers with the Domain Controller in Unit 13, *An Introduction to Clustering*.

□ 1.4. Save your changes to **host-slave.xml** and close the text editor.

□ 2. Start the Host Controller

□ 2.1. Start your **host2** again, except this time specify the IP address of the master you are going to attempt to connect to by defining the **jboss.domain.master.address** property:

```
./domain.sh -Djboss.domain.master.address=192.168.0.254
            --host-config=host-slave.xml
            -Djboss.domain.base.dir=/home/student/JB248/opt/machine2/
domain/
```

□ 3. Verify the Connection

□ 3.1. Watch the log events in the terminal window and verify that your **host2** successfully connects to the other Domain Controller in the classroom. (It is easier to verify the connection by looking at the log events on the Domain Controller.)

□ 3.2. Point your browser to the Management Console of the Domain Controller at `http://192.168.0.254:9990`. On the **Runtime** page, your Host should appear in the **Server** drop-down list in the upper-left corner.

□ 3.3. You should also see your Servers in the list of **Server Instances**.

□ 4. STOP!

At this point, you have shown that you can connect to a remote domain controller. Revert your host name for machine2 back to host2, reconnect it to your local domain controller, and ensure that all three EAP machines are up and running.