

Guided Exercise: Configuring Servers

In this lab work, you will configure servers in a managed domain using the JBoss EAP Command Line Interface (CLI) and deploy applications in the managed domain.

Resources	
Files	/home/student/JB248/labs/domain /home/student/JB248/labs/host
app url	ND

Results

You should be able to set up and start a managed domain with two server groups and four servers, and deploy applications to them.

before you start

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

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

1. In this guided exercise, you will be simulating running three host computers on workstations.

In order not to alter the existing EAP 7 installation in /opt/jboss-eap-7.0, the installation script has created a domain controller in the machine1 folder in the /home/student/JB248/labs/domain directory and two folders in the /home/student/ JB248/labs/host directory named machine2 and machine3, which simulate the host controllers.

You also set up a pre-configured managed domain with the two hosts and the domain controller. There are two predefined server groups in the managed domain. You must start the managed domain and define four new servers using the JBoss EAP 7 CLI.

2. Start the domain controller (machine1) and the two slave hosts (machine2 and machine3).

2.1. The domain controller should be started first, followed by the slaves, since which will provide the configuration of the host controllers.

Start the domain controller using a new terminal window:

```
[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
```

2.2. Start the host controller on machine2.

Run the following command from your `/opt/jboss-eap-7.0/bin` folder in a new terminal window on your workstation to start machine2 using the `host-slave.xml` configuration file:

```
[student@workstation domain]$ 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
```

23. Observe the terminal window of the domain controller. must see entry

Log showing slave host2 connecting:

```
[Host Controller] 11:42:16,348 INFO [org.jboss.as.domain.controller] (Host Controller Service Threads -
36) WFLYHC0019: Registered remote slave host "host2", JBoss JBoss EAP 7.0.0.GA (WildFly
2.1.2.Final-redhat-1)
```

2.4. Start machine3 to join the managed domain.

Run the following command from your `/opt/jboss-eap-7.0/bin` folder in a new terminal window on your workstation to start machine3 using the `host-slave.xml` configuration file:

```
[student@workstation domain]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./domain.sh \
-Djboss.domain.base.dir=/home/student/JB248/labs/host/machine3/domain/ \ --host-config=host-
slave.xml \
-Djboss.domain.master.address=172.25.250.254
```

2.5. Observe the terminal window of the domain controller. must see entry

Log showing slave host3 connecting:

```
[Host Controller] 11:42:16,348 INFO [org.jboss.as.domain.controller] (Host Controller Service Threads -
36) WFLYHC0019: Registered remote slave host "host3", JBoss JBoss EAP 7.0.0.GA (WildFly
2.1.2.Final-redhat-1)
```

You have now started a managed domain with the domain controller running on host machine1 and two slaves, machine2 and machine3, managed by the domain controller. In the next step, you will access the EAP 7 CLI and configure the servers.

3. Access the JBoss EAP CLI and configure the servers.

3.1. In a new terminal window on workstation, start the EAP CLI and connect to the domain controller:

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

Chapter 5. Configuring servers in a managed domain

- 3.2. Before creating the servers, you need to verify that the user groups have been created. Verify that the Group1 and Group2 server groups have been created and are running. In the EAP CLI, run the following commands:**

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

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

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

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

- 3.3. Define a new server named server-one on Host host2 with a port offset value of 100, and assign it to the Group1 server group.**

```
[domain@workstation:9990 /] /
host=host2/server-config=server-one:add\ (auto-
start=true,group=Group1,socket-binding-port-offset=100) {

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

- 3.4. Define another server named server-two on Host host2 with a value of port offset of 200, and assign it to the Group2 server group.**

```
[domain@workstation:9990 /] /
host=host2/server-config=server-two:add\ (auto-
start=true,group=Group2,socket-binding-port-offset=200) {

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

- 3.5. Define a third server named server-three on Host host3 with a port offset value of 300, and assign it to the Group1 server group.**

```
[domain@workstation:9990 /] /host=host3/server-config=server-three:add\ (auto-
start=true,group=Group1,socket-binding-port-offset=300)
```

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

3.6. Define a fourth server named server-four on Host host3 with a port offset value of 400, and assign it to the Group2 server group.

```
[domain@workstation:9990 /] /host=host3/server-config=server-four:add\ (auto-
start=true,group=Group2,socket-binding-port-offset=400) {

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



use

The auto-start property is set to true to ensure that the server starts automatically when the host controller it is configured on is started or restarted. If you do not want to start the server automatically, set the property to false.

4. Start the servers.

4.1. Although you defined and created the servers, they are not running yet. Start the servers in the managed domain.

```
[domain@workstation:9990 /] /host=host2/server-config=server-one:start {

  "outcome" => "success",
  "result" => "STARTING"
}
```

Monitor the machine2 console window and verify that server-one starts:

```
[Host Controller] 14:02:31,089 INFO [org.jboss.as.host.controller] (Host Controller Service Threads
- 44) WFLYHC0023: Starting server server-one 14:02:31,101 INFO
[org.jboss.as.process.Server:server-one.status]
(ProcessController-threads - 4) WFLYPC0018: Starting process 'Server:server one'
...
```

Notice that because you defined server-one with a port offset value of 100, the ports are consequently bound:

```
[Server:server-one] 14:02:34,649 INFO [org.wildfly.extension.undertow] (MSC service thread 1-2)
WFLYUT0012: Started server default-server.
```

Chapter 5. Configuring servers in a managed domain

```
[Server:server-one] 14:02:34,651 INFO [org.wildfly.extension.undertow] (MSC service thread 1-2)
WFLYUT0018: Host default-host starting [Server:server-one] 14:02:34,794 INFO
[org.wildfly.extension.undertow] (MSC service thread 1-4) WFLYUT0006: Undertow HTTP listener
default listening on 172.25.254.250:8180
```

4.2. Start server-two, server-three, and server-four, and verify that the HTTP listener is bound to ports 8280, 8380, and 8480, respectively.

```
[domain@workstation:9990 /] /host=host2/server-config=server-two:start {

    "outcome" => "success",
    "result" => "STARTING"

} [domain@workstation:9990 /] /host=host3/server-config=server-three:start {

    "outcome" => "success",
    "result" => "STARTING"

} [domain@workstation:9990 /] /host=host3/server-config=server-four:start {

    "outcome" => "success",
    "result" => "STARTING"

}
```



use

Instead of starting servers one by one, you can start all servers in a server group with one command:

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

    "outcome" => "success",
    ...

}
```

5. Shut down the Servers, Server Groups, and Host Controllers.

5.1. You can manage the life cycle (start, stop, restart, suspend, etc.) of servers, pools, and host controllers directly from the JBoss EAP CLI. This offers flexibility in terms of domain administration from a single access point, as well as increased security, since you do not need to provide SSH login access to host computers in the managed domain.

5.2. Stop server-one using the JBoss EAP CLI:

```
[domain@workstation:9990 /] /host=host2/server-config=server-one:stop {

    "outcome" => "success",
    "result" => "STOPPING"

}
```

Verify that the application example is NOT accessible at `http://172.25.250.254:8180/example`. The example application should still be accessible at `http://172.25.250.254:8380/example`, since server-three is still running.

5.3. Stop server-three using the JBoss EAP CLI:

```
[domain@workstation:9990 /] /host=host3/server-config=server-three:stop {

    "outcome" => "success",
    "result" => "STOPPING"

}
```

Verify that the application example is now NOT accessible at the URL `http://172.25.250.254:8380/example`.

5.4. Stop all servers in Group2 using the JBoss EAP CLI:

```
[domain@workstation:9990 /] /server-group=Group2:stop-servers {

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

}
```

Verify that the version application is NOT accessible at both `http://172.25.250.254:8280/version` and `http://172.25.250.254:8480/version`, since you shut down the entire server pool.

5.5. Stop both host controllers (host2 and host3) on machine2 and machine3 using the JBoss EAP CLI. Shut down the machine2 host controller first:

```
[domain@workstation:9990 /] /host=host2:shutdown {

    "outcome" => "success",
    "result" => undefined

}
```

Monitor the machine2 console window and verify that the host controller has shut down.

5.6. Stop the host controller on machine3 using the JBoss EAP CLI:

```
[domain@workstation:9990 /] /host=host3:shutdown {

    "outcome" => "success",
    "result" => undefined

}
```

Monitor the machine3 console window and verify that the host controller has shut down.

Chapter 5. Configuring servers in a managed domain

6. Perform cleaning.

6.1. After you have shut down all host controllers, shut down the domain controller. Press **Ctrl+C** to stop the domain controller. (Alternatively, you can shut down the domain controller using the JBoss EAP CLI, as demonstrated in the previous step.)

6.2. This concludes the guided exercise.