Server configuration

Goals

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

- **Configure a server on a host controller.**
- **Describe the different attributes of a server configuration that are found configured in the host.xml file of the host controller.**
- **Describe the different options available for creating and managing servers in a managed domain.**
- **Describe how to shut down servers, server groups, and the entire managed domain of the EAP CLI.**

Definition of servers

A server is a logical EAP runtime instance that runs in a managed domain. A server must belong to a server group, even if the group only contains one server. Individual servers are managed by their respective host controllers, which communicate with the domain controller and ensure that all servers within a managed domain have the same settings.

Each server belongs to a unique ***server group*** **and runs within a *host.* A host can run multiple servers simultaneously using the port offset concept provided by the EAP to ensure that no socket or network port confrontations occur at runtime.**

The servers are configured in the host.xml file on individual host controllers, which run on a physical server (or a virtual machine).

server configuration

Servers in a managed domain are defined in the <servers> section of host.xml on each host controller. The <server> child element within the <servers> parent element can be used to define a server. Here is a sample <server> definition:

```
<servers>
    <server name="serverA" group="main-server-group" >
        <socket-bindings socket-binding-group="standard-sockets" /> </server>

    <server name="serverB" group="other-server-group" auto-start="true" >
        <jvm name="default" >
            ...
        </jvm>
        <socket-bindings socket-binding-group="standard-sockets" port-offset="350" > </server>
</servers>
```

1. The name attribute is required and is the name of the server. It must be unique within the host.
2. The group attribute is required and must refer to a valid server group name previously defined in domain.xml.
3. The socket-binding-group attribute references a valid socket-binding group value defined in domain.xml.
4. The auto-start attribute defaults to true and causes the server to start automatically when the host controller starts or restarts. This can be set to false if you want to disable automatic starting of the server on the host, in which case you need to start the server manually, if a host controller starts or reboots.
5. The <jvm> element configures the JVM settings for the server. Values defined here in the <server> element will override jvm values from the host level or from the server pool level. JVM memory allocation in EAP will be covered later in this course.
6. The port-offset value is used to avoid network port clashes when multiple servers are defined on a host. In this case, serverB will run with an offset value of 350 from the base value defined for this profile. For example, on this host, the Undertow web server component will run on port 8080 for serverA, since it does not have a defined port offset, and for serverB, the web server will run on port 8430 (for example, 8080 +350).

use

A host can have any number of servers. To avoid port conflicts, use the port-offset attribute of <socket-bindings>.

use

Notice that the <server> element does not contain any data about deployed applications. This is because all implementations for a server are configured at the server group level in the domain.xml domain configuration file.

Server administration using the EAP administration console

Servers in a managed domain can be created and managed in the Hosts section, on the Runtime tab of the administration console.

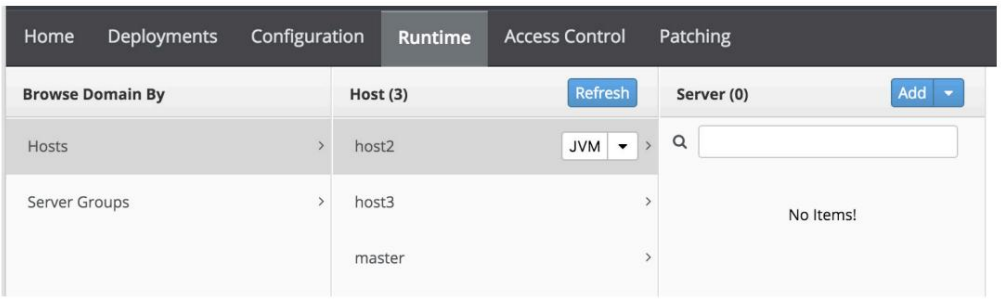

Figure 5.22: Runtime view

To add a new server, you need to specify three key attributes; a unique name for the server, a valid ***server group*** **from the list of server groups defined in the domain.xml configuration file, and a *port offset*** value **in case there are multiple servers running on this server host.**

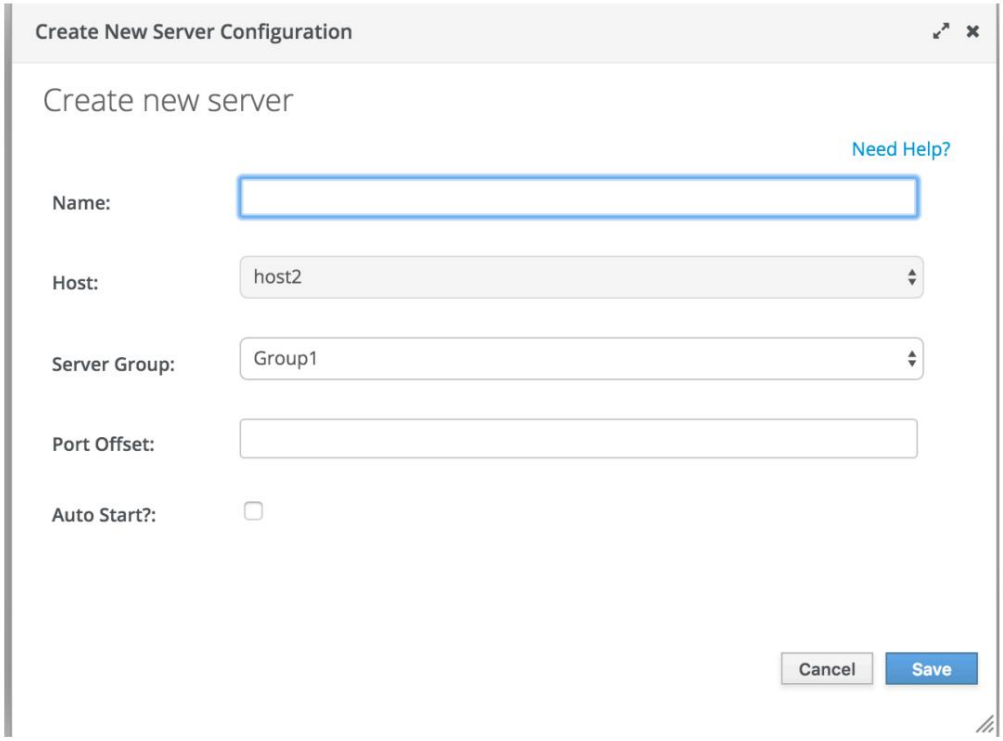

Figure 5.23: Create a new server

To remove a server from a host, make sure the server is stopped.

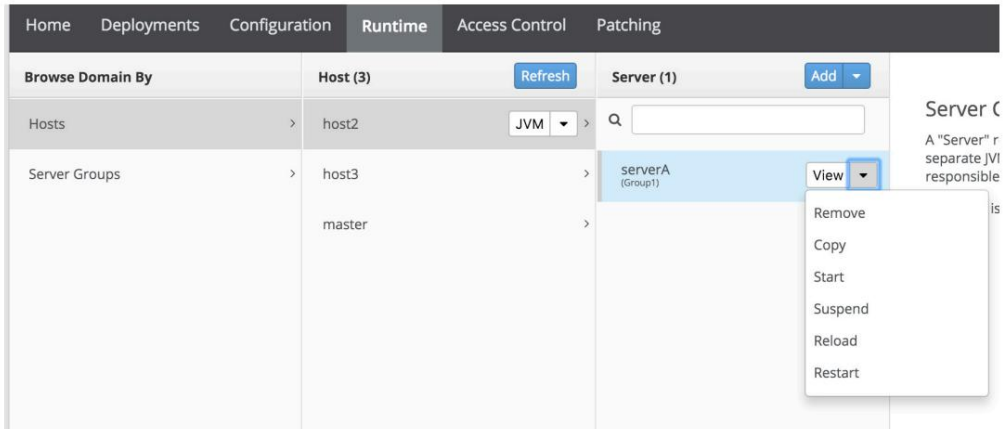

Figure 5.24: Server Options - Delete

Individual servers can be started, stopped, restarted, reloaded, and suspended by clicking the View dropdown menu next to the server in the Hosts section on the Runtime tab of the management console.

Configuring and managing servers using the JBoss EAP CLI Servers can also be created and managed using the JBoss EAP CLI. The advantage of this approach is that the steps to create and manage the servers can be scripted as part of an automated workflow, and it is faster to create multiple servers in batch mode. This approach also has the option to atomically allocate or revert multiple steps in batch mode, ensuring the integrity of the host.xml configuration file in the event of an error during CLI command execution.

Servers in a managed domain are configured in the JBoss EAP CLI host/ namespace. There are commands to create, view, start, stop, restart, reload, suspend, and remove servers under this namespace. To add a new server, you can run the following command:

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

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

To start a single server, run the following command:

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

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

To stop a single server, run the following command:

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

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

All servers in a server group can be run by using the following command:

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

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

Similarly, to stop all servers in a server group, run the following command:

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

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

use

The blocking=true property ensures that control returns to the CLI request only after all servers in the server group have been successfully started or stopped. The default behavior is for control to return immediately on CLI request, and servers start or stop asynchronously in the background.

To remove a server, you must first ensure that the server has been stopped. To remove the server, run the following command:

```
[domain@workstation:9990 /] /host=host2/server-config=serverA:remove() {

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

In EAP 7 it is possible to suspend a server using the suspend operation. No server in suspended mode will accept new requests from any client until the server is back up. A suspended server can begin serving requests immediately by executing the resume operation.

Added proper shutdown support. This is different from a regular shutdown, as the server is suspended before it is shut down. To invoke this behavior, a timeout parameter must be specified for the shutdown, stop, or stop-servers operations. The server waits until all requests have timed out before shutting down.

You can also gracefully shut down a host controller from the EAP CLI. To shut down a host controller, first ensure that all servers running on this host are stopped, and then run the following command:

```
[domain@workstation:9990 /] /host=host2:shutdown(timeout=1000) {

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

Similarly, it is possible to gracefully shut down the domain controller from the EAP CLI. To shut down the domain controller (assuming it is named master), run the following command:

```
[domain@workstation:9990 /] /host=master:shutdown(timeout=1000) { {"outcome"

=> "success"}
[domain@localhost:9990 /]
The controller has closed the connection.
[disconnected /]
```

```
}
```