Impact2014

April 27 - May 1 Las Vegas, NV

Session 1377

Build and Manage A WebSphere Liberty Application Cluster including the new Admin Center

Lab Instructions

Authors:
Michael C Thompson, WebSphere Developer, mcthomps@us.ibm.com
Chris Vignola, WebSphere Architecture, cvignola@us.ibm.com

Key Reference Notes for Lab 1377

Passwords and resources

Password information	
VMWare	User: root Password: web1sphere
Naming conventions	
Liberty installation	/opt/wlp
Lab materials	/opt/lab-materials
Lab Instructions (soft-copy)	/opt/lab-materials/ Build_Liberty_Cluster_Lab_Instructions.pdf

TABLE OF CONTENTS

1 Objective

In this hands-on lab, you build a real Liberty application cluster and manage it using the Liberty Admin Center, the web-based administrative interface. Both application clusters and the Admin Center are new capabilities in the IBM WebSphere Application Server V8.5.5 release. Learn what a Liberty collective is, how clusters are defined, and how to operate and manage clusters through both the command line and the Admin Center and its tools.

In the lab, attendees set up a collective, create a cluster, deploy and verify applications on the cluster, and perform basic operational tasks on the cluster. After completing this lab, participants are fully equipped to set up and operate their own production Liberty application clusters and manage them through the command line or the new Admin Center.

In this lab, you learn:

- The concepts and operations of a Liberty collective and clustering with the WebSphere Application Server Liberty profile
- Hands-on experience creating, configuring and performing operations on a collective and cluster
- Hands-on experience with the Jython scripting support
- Hands-on experience with the Admin Center, the new web-based administrative interface

2 Prerequisite Knowledge

- Basic Linux knowledge
- This lab uses gedit as the editor of choice in the command examples.
  You are free to use any editor you wish (the VM image has vi and emacs available).

When large amounts of text are displayed as part of an operation output or screen shot, the important portions are highlighted or otherwise indicated using red.

3 Step-by-Step Instructions

3.1 Introduction to Liberty

Approximate time: 5 - 10 minutes

These steps take you through the most basic operations supported by the Liberty profile. This entails starting the default server, deploying an application and dynamically changing the server's configuration.

The Liberty profile is designed to provide a world-class application infrastructure platform as well as a compelling developer experience.

3.1.1 Create and start the default server.

Open a terminal (right-click desktop and select Open in Terminal) and execute the following commands to create and start the default server:

```
[illegible]
```

```
[illegible]
```

3.1.2 Deploy the sample application "snoop" by adding the war file to dropins.

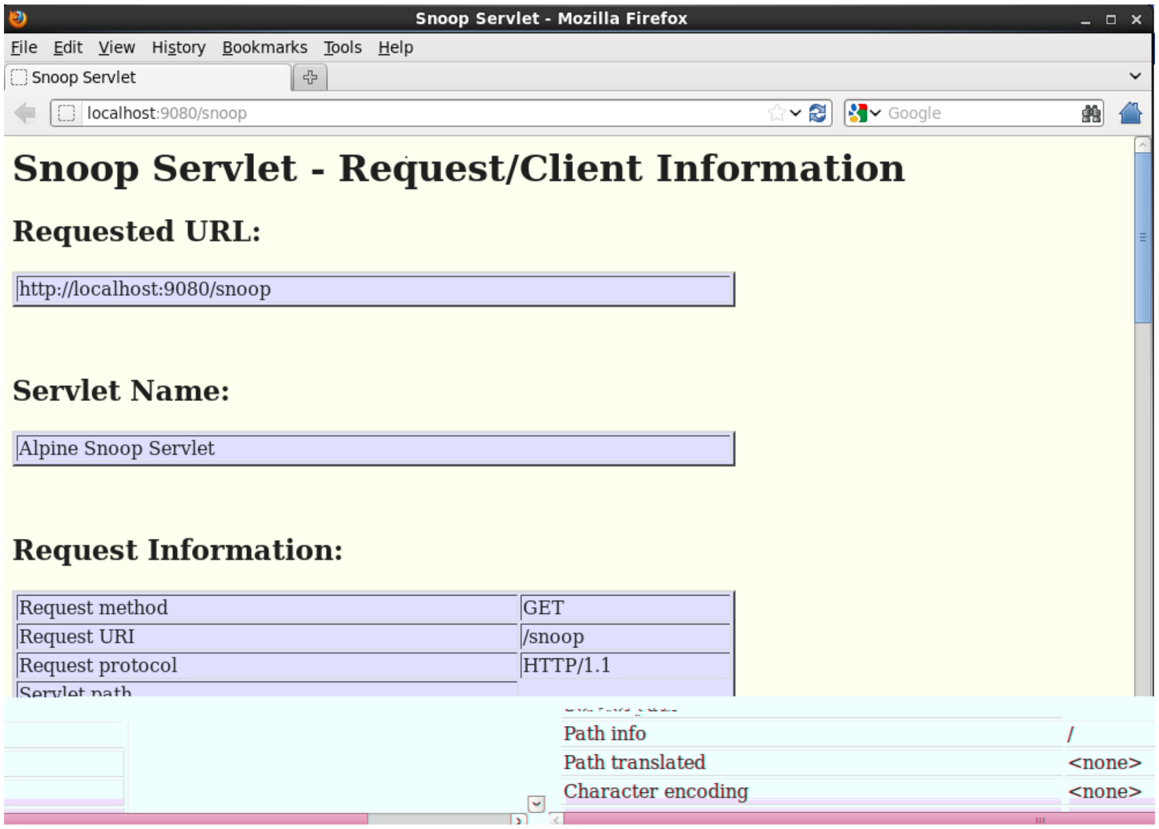
An application archive placed in the dropins directory will be automatically deployed and started. Liberty supports two ways to deploy applications: via the dropins directory demonstrated here, as well as through configuration. A configuration based deployment will be used later in this lab.

```
[illegible]
```

3.1.3 Access the sample application "snoop".

Start Firefox, go to the URL http://localhost:9080/snoop

You should see a page similar to the following screenshot:


Snoop Servlet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Snoop Servlet
localhost:9080/snoop
Google

Snoop Servlet - Request/Client Information

Requested URL:

http://localhost:9080/snoop

Servlet Name:

Alpine Snoop Servlet

Request Information:

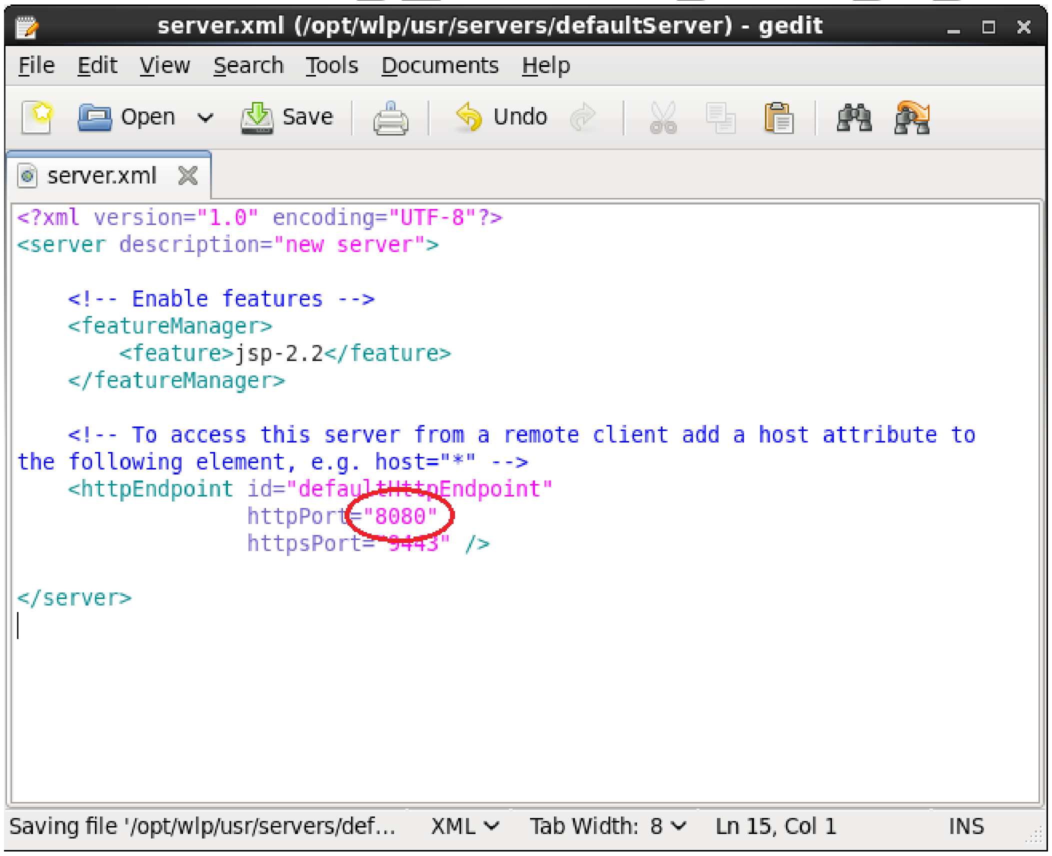
Request method	GET
Request URI	/snoop
Request protocol	HTTP/1.1
Servlet path	
Path info	/
Path translated	<none>
Character encoding	<none>

Close the browser.

3.1.4 Change the default HTTP port for the server.

The Liberty profile responds to configuration changes dynamically, no restart is required.

Change the HTTP port from 9080 to 8080.


server.xml (/opt/wlp/usr/servers/defaultServer) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="8080"
                  httpsPort="9443" />

</server>
```

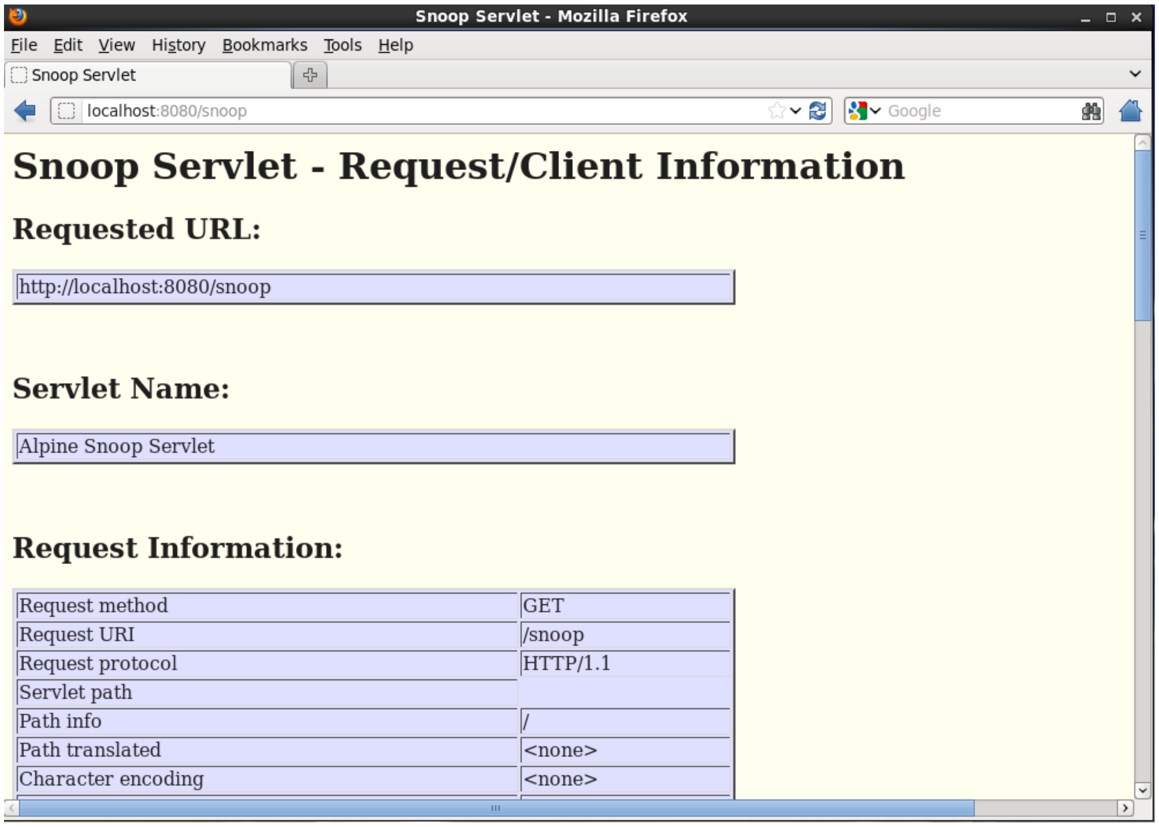
Saving file '/opt/wlp/usr/servers/def... XML Tab Width: 8 Ln 15, Col 1 INS

Save and close the file.

3.1.5 Use the new port to access the application.

Start Firefox, go to URL http://localhost:8080/snoop

You should see a page similar to the following screenshot:


Snoop Servlet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Snoop Servlet
localhost:8080/snoop
Google

Snoop Servlet - Request/Client Information

Requested URL:

http://localhost:8080/snoop

Servlet Name:

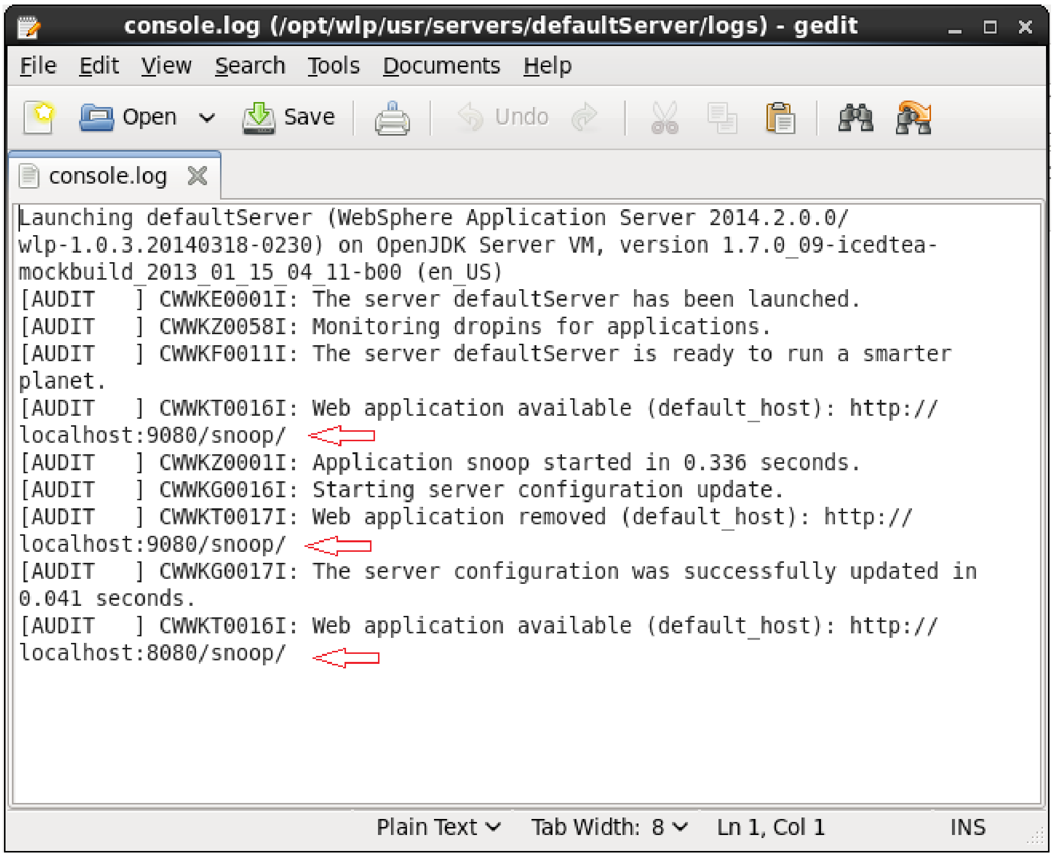
Alpine Snoop Servlet

Request Information:

Request method	GET
Request URI	/snoop
Request protocol	HTTP/1.1
Servlet path	
Path info	/
Path translated	<none>
Character encoding	<none>

Close the browser.

3.1.6 View the console.log messages for the web application.


console.log (/opt/wlp/usr/servers/defaultServer/logs) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
console.log
Plain Text Tab Width: 8 Ln 1, Col 1 INS

```
Launching defaultServer (WebSphere Application Server 2014.2.0.0/
wlp-1.0.3.20140318-0230) on OpenJDK Server VM, version 1.7.0_09-icedtea-
mockbuild_2013_01_15_04_11-b00 (en_US)
[AUDIT   ] CWWKE0001I: The server defaultServer has been launched.
[AUDIT   ] CWWKZ0058I: Monitoring dropins for applications.
[AUDIT   ] CWWKF0011I: The server defaultServer is ready to run a smarter
planet.
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://
localhost:9080/snoop/
[AUDIT   ] CWWKZ0001I: Application snoop started in 0.336 seconds.
[AUDIT   ] CWWKG0016I: Starting server configuration update.
[AUDIT   ] CWWKT0017I: Web application removed (default_host): http://
localhost:9080/snoop/
[AUDIT   ] CWWKG0017I: The server configuration was successfully updated in
0.041 seconds.
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://
localhost:8080/snoop/
```

Close the file.

3.1.7 Stop the defaultServer.

[illegible]

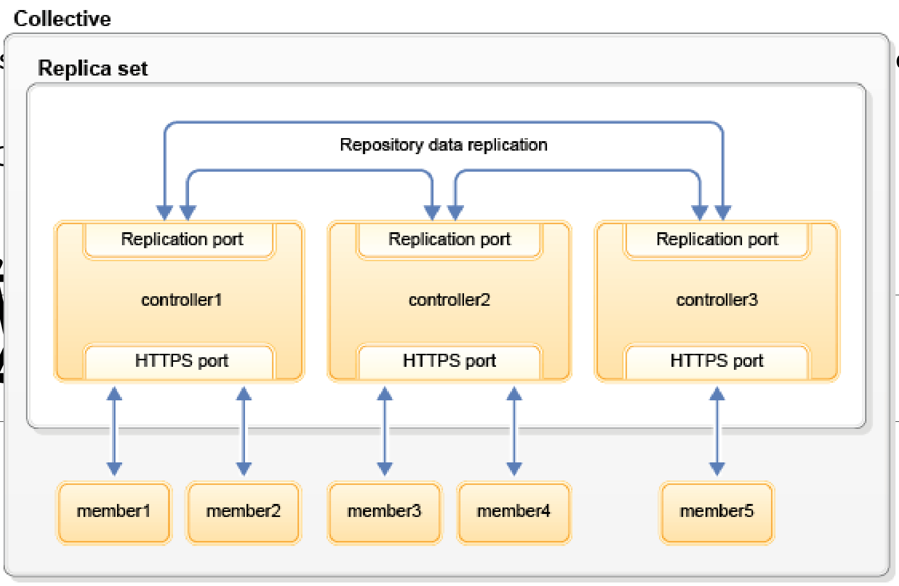
You now have experience with basic server operations, configuration and deploying an application!

3.2 Create a collective

Approximate time: 10 - 20 minutes

These steps take you through creating and configuring a basic collective. A collective is the set of Liberty servers in a single administrative domain. A collective consists of at least one "collective controller", a server with the collectiveController-1.0 feature enabled. Optionally, a collective may have many "collective members", servers with the collectiveMember-1.0 feature enabled. A collective may be configured to have many collective controllers, called a replica set. Configuration of the replica set is not covered in this lab, but documentation is available from wasdev.net.

The following illustration shows a sample collective topology with a replicate set of 3 controllers and 5 collective members.



In this s created.

3.2.1 C

3.2.2 Create the collective controller configuration.

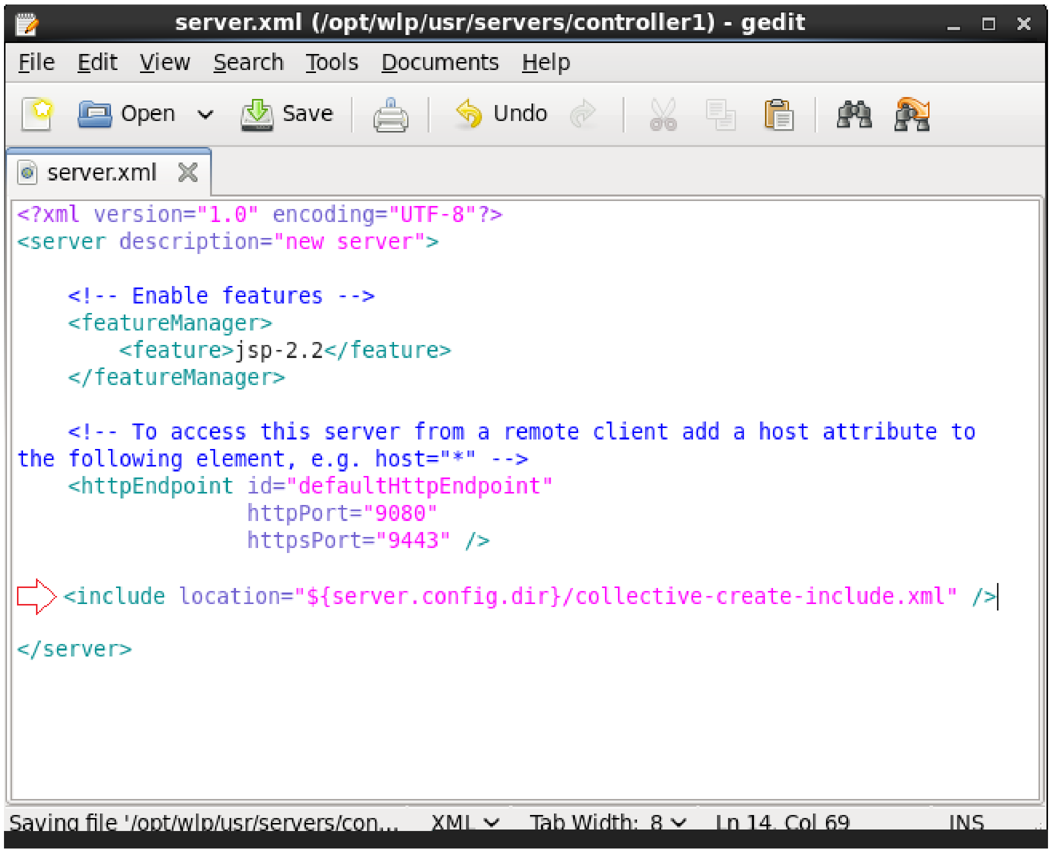
This will establish the administrative domain security configuration. The servers in the collective communicate with each other using signed SSL certificates. The 'collective create' command establishes the initial set of SSL keys.

The Liberty profile does not ship with any default passwords. As such, the the create command requires a keystore password. In this lab, all keystore passwords will be **'Impact2014'**. Each keystore password can be different, but to keep the [illegible] passwords will be the same.

```
[illegible]
```

3.2.3 Update the server.xml to include the generated XML file.

Add the include from the create operation.


server.xml (/opt/wlp/usr/servers/controller1) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9080"
                  httpsPort="9443" />

   <include location="${server.config.dir}/collective-create-include.xml" />

</server>
```

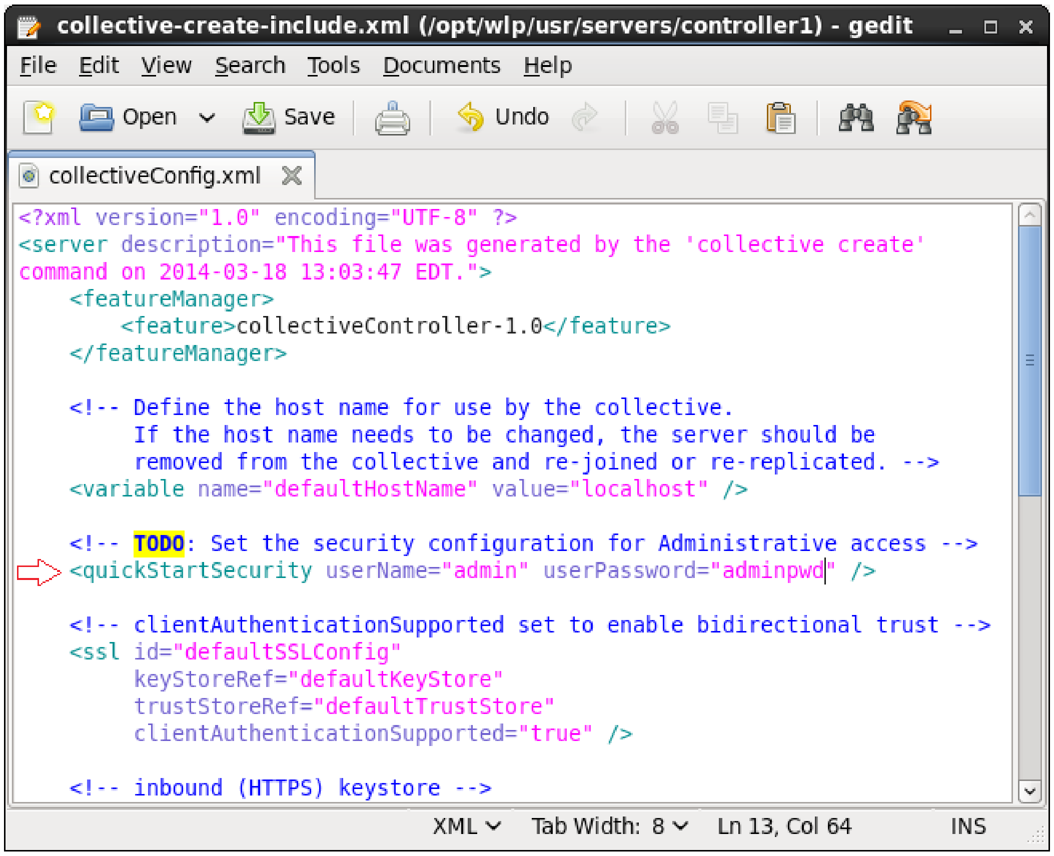
Saving file '/opt/wlp/usr/servers/con... XML Tab Width: 8 Ln 14, Col 69 INS

3.2.4 Update collective-create-include.xml.

The Liberty profile does not ship with any default users. Therefore, the administrator user name and password must be specified. For the purposes of this lab, use the user name **'admin'** and the password **'adminpwd'** to configure the [illegible] element, which will establish the administrator user.

Set the user and password to admin / adminpwd

- admin
- adminpwd


collective-create-include.xml (/opt/wlp/usr/servers/controller1) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
collectiveConfig.xml
XML Tab Width: 8 Ln 13, Col 64 INS

```
<?xml version="1.0" encoding="UTF-8" ?>
<server description="This file was generated by the 'collective create'
command on 2014-03-18 13:03:47 EDT.">
    <featureManager>
        <feature>collectiveController-1.0</feature>
    </featureManager>

    <!-- Define the host name for use by the collective.
         If the host name needs to be changed, the server should be
         removed from the collective and re-joined or re-replicated. -->
    <variable name="defaultHostName" value="localhost" />

    <!-- TODO: Set the security configuration for Administrative access -->
    <quickStartSecurity userName="admin" userPassword="adminpwd" />

    <!-- clientAuthenticationSupported set to enable bidirectional trust -->
    <ssl id="defaultSSLConfig"
         keyStoreRef="defaultKeyStore"
         trustStoreRef="defaultTrustStore"
         clientAuthenticationSupported="true" />

    <!-- inbound (HTTPS) keystore -->
```

3.2.5 Start the collective controller.

```
[illegible]
```

3.2.6 Verify the server started correctly and is ready to receive members.

Look for the following message: [illegible]

The CWWKX9003I message indicates that the collective controller is ready to receive connections from clients or members.

To quickly find the message, search for it using 'grep': [illegible]

```
[illegible]
```

3.2.7 Create a member to join the collective.

```
[illegible]
```

3.2.8 Join member1 to the collective.

This will exchange the SSL certificates necessary for the server to become a collective member. Liberty does not have a client-side SSL configuration, so you will need to accept the SSL certificate presented by the controller.

[illegible]

```
[illegible] y (hit Enter)
```

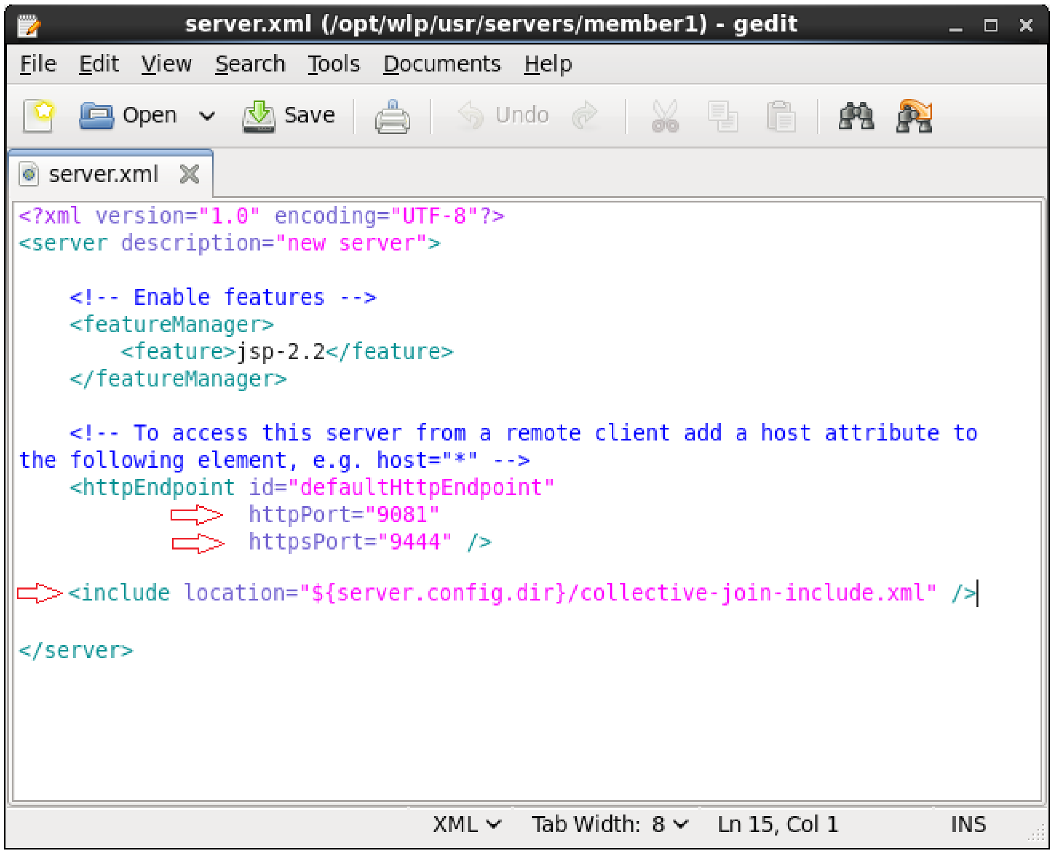
[illegible]

3.2.9 Update the server.xml for member1.

Because both the controller and member are running on the same system, the default ports for the member must be changed as the default ports are already in use by the controller. Update the HTTP and HTTPS ports to be 9081 and 9444.

Change the HTTP and HTTPS ports to 9081 and 9444.

Add the include from the join operation.


server.xml (/opt/wlp/usr/servers/member1) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
server.xml
XML
Tab Width: 8
Ln 15, Col 1
INS

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9081"
                  httpsPort="9444" />

    <include location="${server.config.dir}/collective-join-include.xml" />

</server>
```

Save and close the file.

3.2.10 Start the collective member

```
[illegible]
```

3.2.11 Verify the member started and is publishing information to the controller.

Look for the following messages:

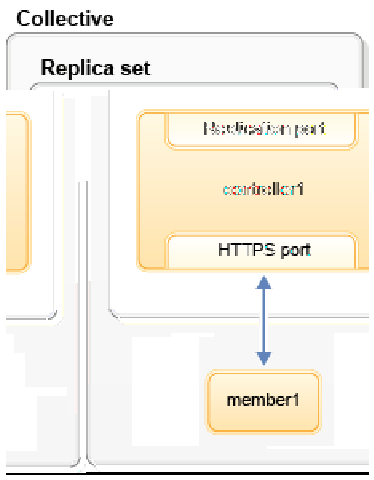
```
[illegible]
```

The CWWKX8112I, CWWKX8114I and CWWKX8116I messages indicate that the collective member is successfully communicating with the controller.

To quickly find the messages, search for them using grep

```
[illegible]
```

You now have a basic collective topology created. In this topology, member1 is a collective member and controller1 is a collective controller.



All collective members publish information about themselves to their collective controller. This published information is available for query directly from the controller without need of forwarding the request down to each collective member.

This published information is used by the controller to determine each member's operational state, and is used by the Admin Center to show information about the collective. The Admin Center is used in Section 3.4 Operations through the Admin Center.

3.3 Perform collective operations via scripting

Approximate time: 5 – 10 minutes

In this section, you use Jython scripting to perform MBean operations against the collective controller which allow you to start and stop registered collective members.

3.3.1 Run the Jython script provided to stop the collective member.

In order to run the script, set the CLASSPATH and JYTHONPATH environment variables to include the restConnector.jar and run the provided client jython script. The script is hard-coded for the host name, paths, user names and passwords used in these instructions. If you have modified any of these values, you will need to edit the script.

[illegible]

```
[illegible]
```

3.3.2 Verify the server stopped.

[illegible]

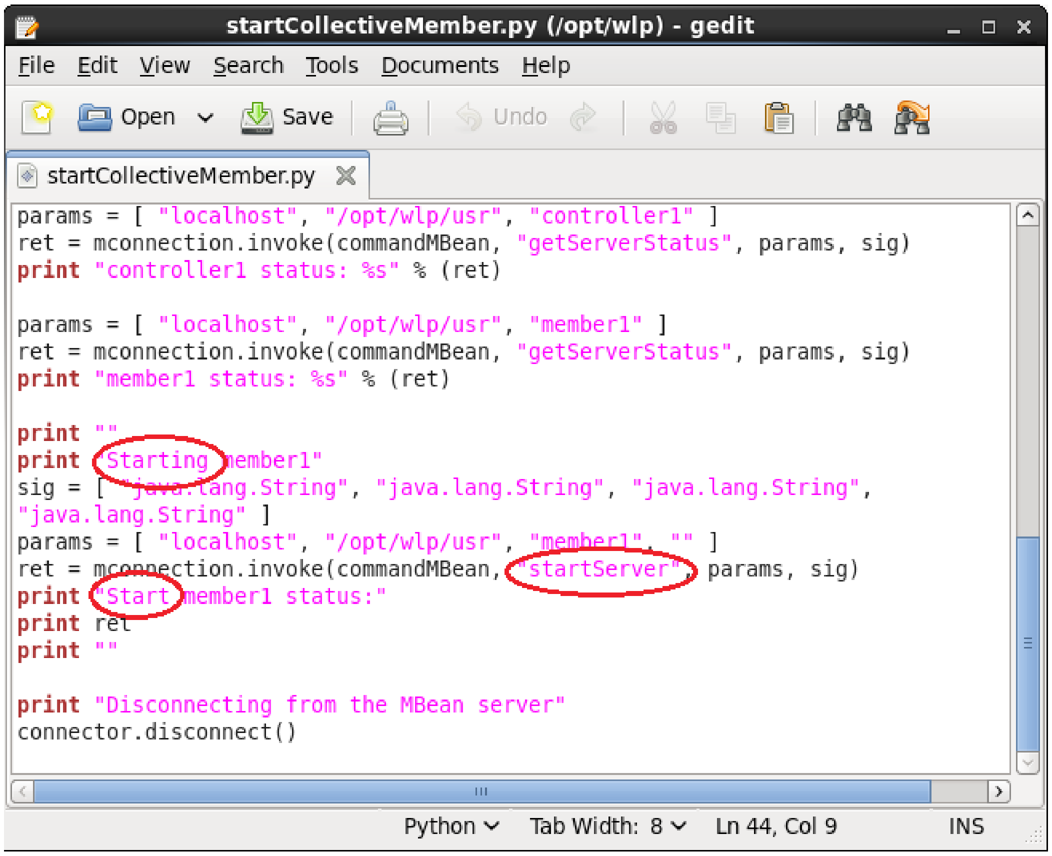
```
[illegible]
```

3.3.3 Start the collective member via scripting.

Copy the stopCollectiveMember.py script to startCollectiveMember.py.

Modify the startCollectiveMember.py script to invoke the startServer operation.

Change 'stop' to 'start':


startCollectiveMember.py (/opt/wlp) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
startCollectiveMember.py
Python Tab Width: 8 Ln 44, Col 9 INS

```
params = [ "localhost", "/opt/wlp/usr", "controller1" ]
ret = mconnection.invoke(commandMBean, "getServerStatus", params, sig)
print "controller1 status: %s" % (ret)

params = [ "localhost", "/opt/wlp/usr", "member1" ]
ret = mconnection.invoke(commandMBean, "getServerStatus", params, sig)
print "member1 status: %s" % (ret)

print ""
print "Starting member1"
sig = [ "java.lang.String", "java.lang.String", "java.lang.String",
"java.lang.String" ]
params = [ "localhost", "/opt/wlp/usr", "member1", "" ]
ret = mconnection.invoke(commandMBean, "startServer", params, sig)
print "Start member1 status:"
print ret
print ""

print "Disconnecting from the MBean server"
connector.disconnect()
```

Save and close the file.

Run the modified script.

[illegible]

```
[illegible]
```

3.3.4 Verify the server has been started

[illegible]

```
[illegible]
```

You now have basic Jython experience and have used the MBeans available on the collective controller to perform operations against the registered collective members.

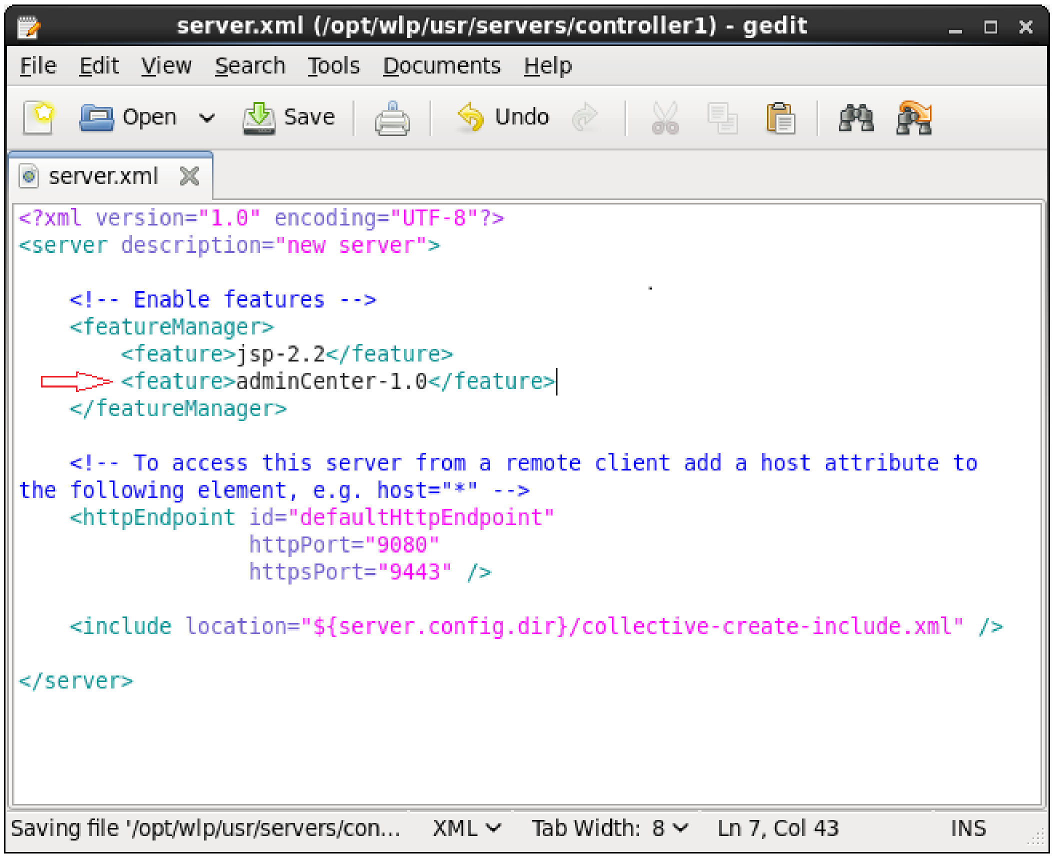
3.4 Operations through the Admin Center

Approximate time: 5 – 10 minutes

3.4.1 Enable the Admin Center.

The Admin Center is a feature, and is therefore not enabled by default. To enable the Admin Center, add the 'adminCenter-1.0' feature to the list of enabled features.

Add the adminCenter-1.0 feature.



```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>adminCenter-1.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9080"
                  httpsPort="9443" />

    <include location="${server.config.dir}/collective-create-include.xml" />

</server>
```

server.xml (/opt/wlp/usr/servers/controller1) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
server.xml
Saving file '/opt/wlp/usr/servers/con...
XML
Tab Width: 8
Ln 7, Col 43
INS

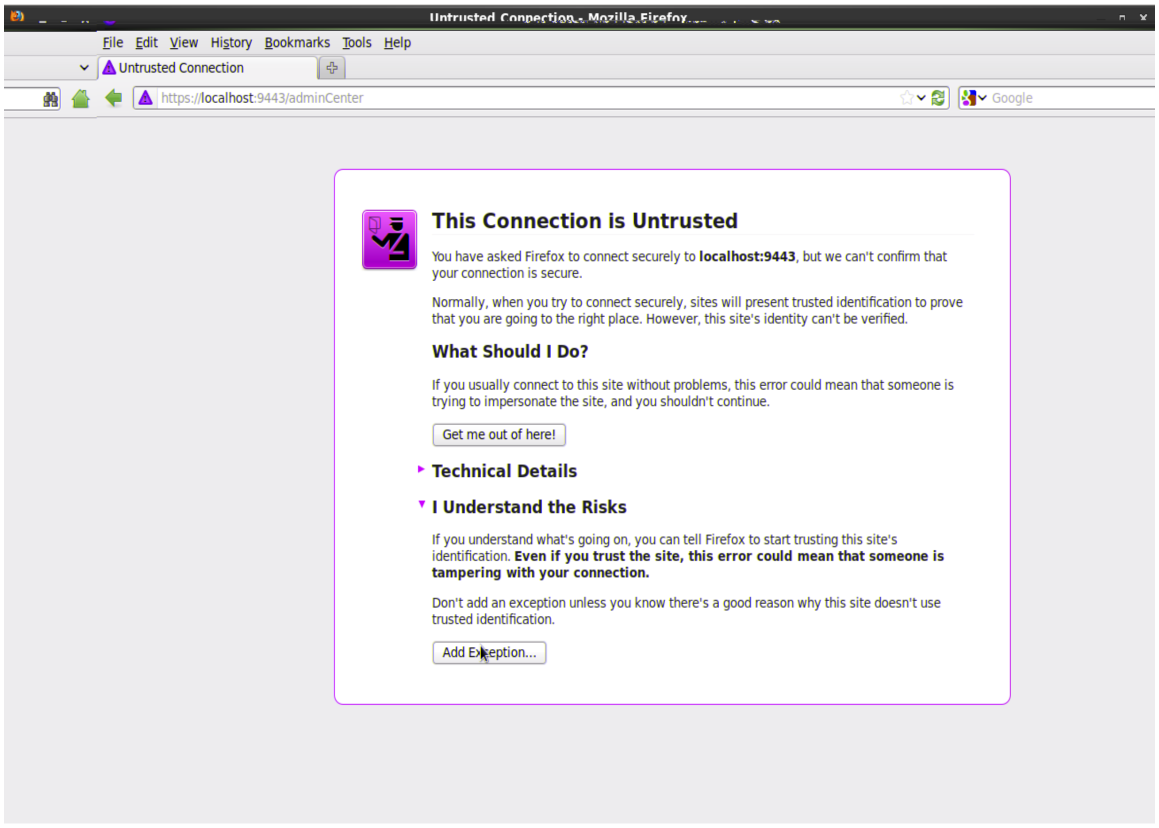
Save and close the file.

3.4.2 Log into the Admin Center

Start Firefox, go to URL http://localhost:9080/adminCenter

You will need to accept the server's SSL certificate.

First, click “Add Exception...”


Untrusted Connection - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Untrusted Connection
https://localhost:9443/adminCenter
Google

This Connection is Untrusted

You have asked Firefox to connect securely to **localhost:9443**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

Technical Details

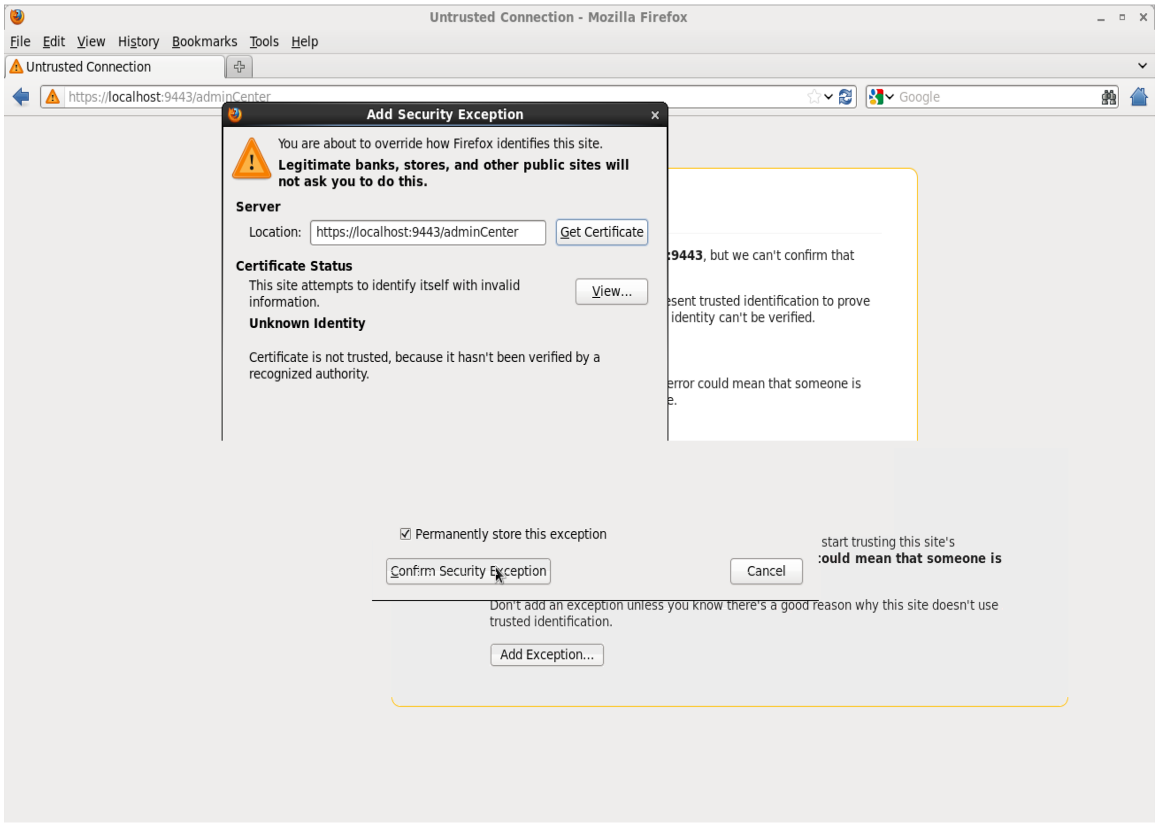
I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Add Exception...

Next, confirm the security exception by clicking "Confirm Security Exception".


Untrusted Connection - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Untrusted Connection
https://localhost:9443/adminCenter
Google
Add Security Exception
Get Certificate
View...
Permanently store this exception
Confirm Security Exception
Cancel
Add Exception...

You are about to override how Firefox identifies this site.

Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://localhost:9443/adminCenter

Certificate Status

This site attempts to identify itself with invalid information.

Unknown Identity

Certificate is not trusted, because it hasn't been verified by a recognized authority.

:9443, but we can't confirm that

esent trusted identification to prove identity can't be verified.

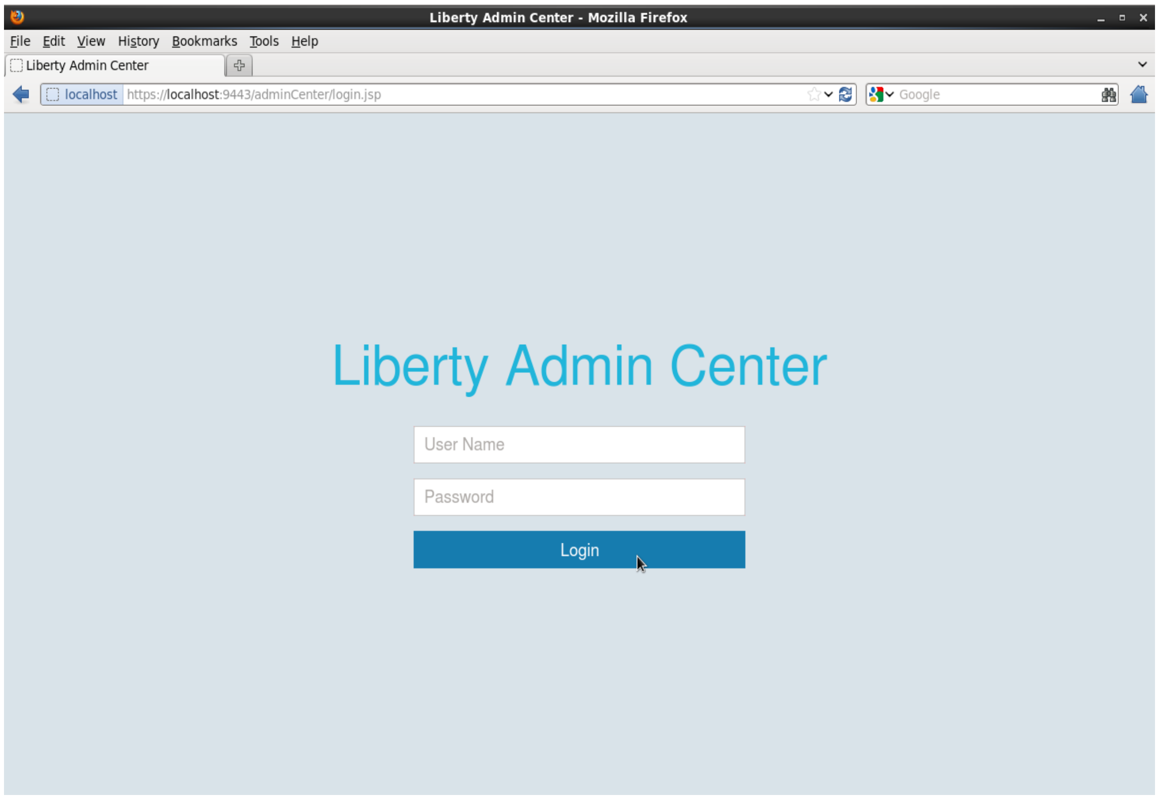
error could mean that someone is e.

start trusting this site's **could mean that someone is**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Now log in to the Admin Center with the administrator user name and password (**'admin'** / **'adminpwd'**).

Enter the user name and password and click “Login”.

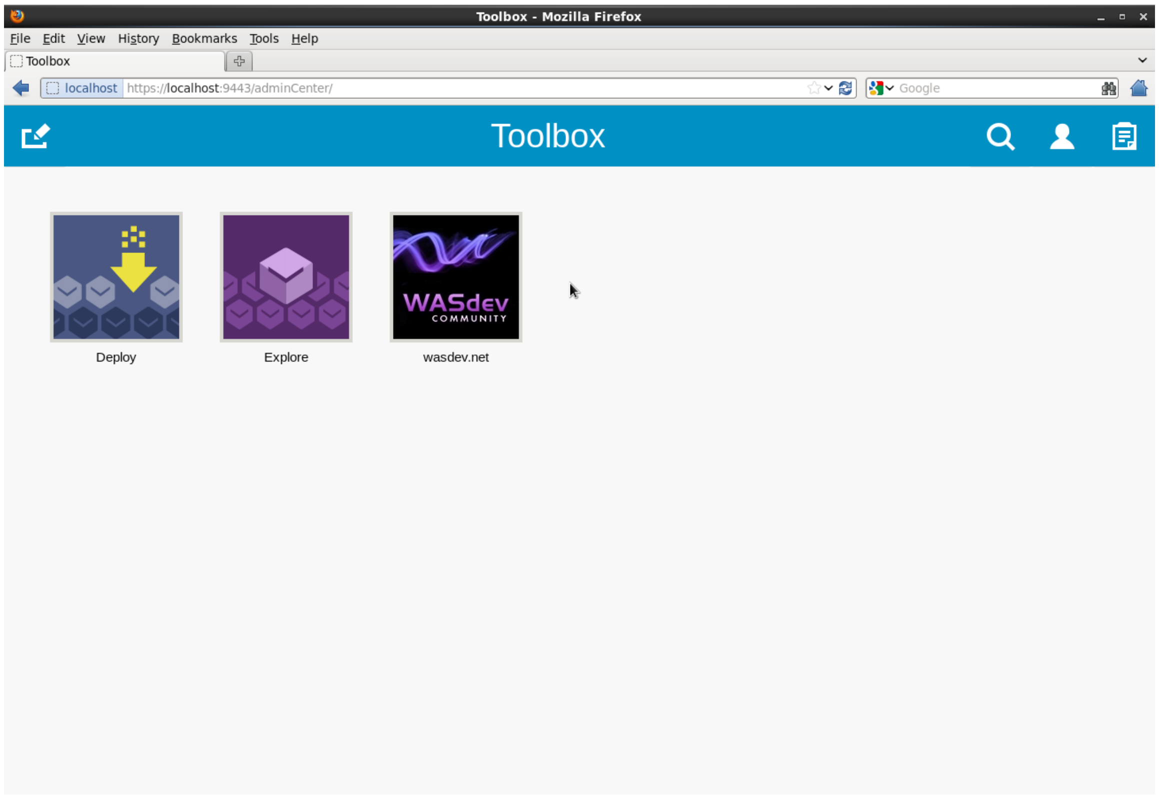

Liberty Admin Center - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Liberty Admin Center
https://localhost:9443/adminCenter/login.jsp
Google
Liberty Admin Center
User Name
Password
Login

3.4.3 The Toolbox

Each user of the Admin Center has a customized view called 'the Toolbox'. The Toolbox allows the user to choose the set of tools that they wish to use.

By default, the Toolbox is populated with the initial set of tools that are present in the catalog. The catalog is the set of all tools installed into the Liberty profile runtime. Users can also add links to commonly used pages by adding a bookmark.

The Toolbox view:

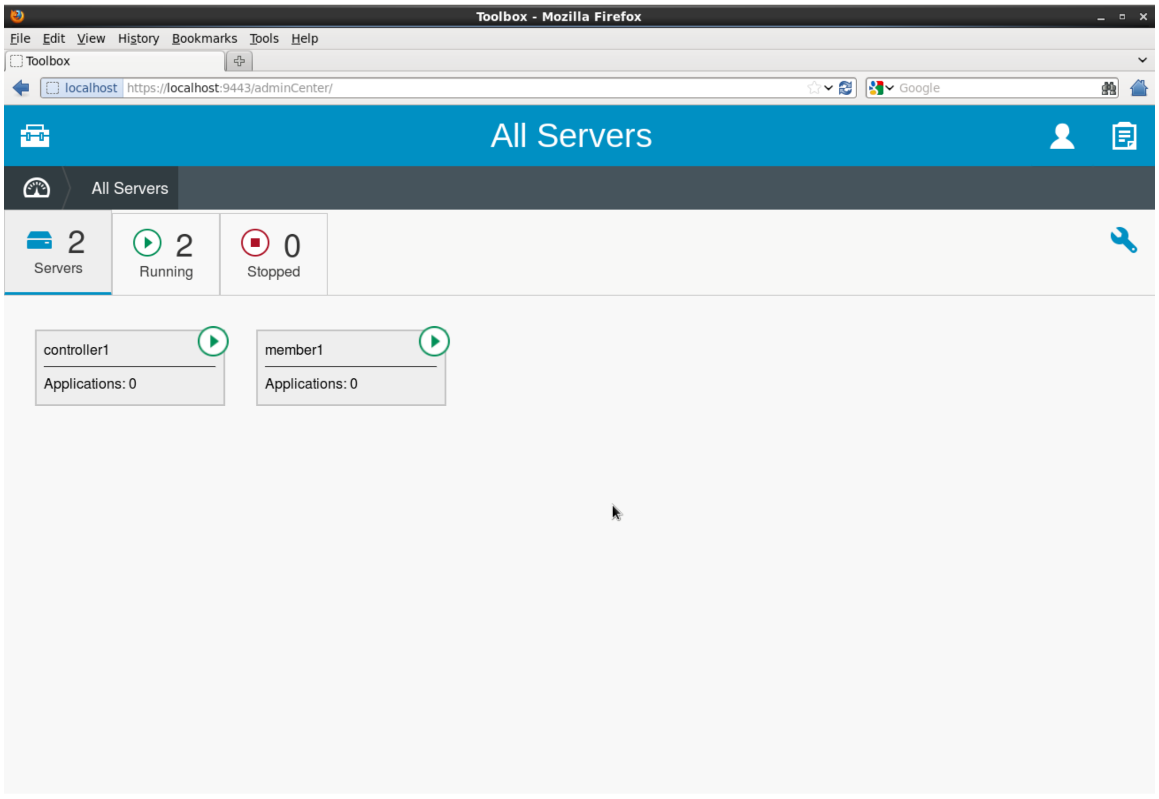

Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Toolbox
WASdev COMMUNITY
Deploy
Explore
wasdev.net

3.4.4 Select the Explore tool.

3.4.5 Click on the Servers dashboard element.

Each dashboard element displays the high-level information about elements in the topology. Clicking on the Servers category on the dashboard will display individual details about each server.

From this view, individual or groups of servers can be started or stopped, and details for a specific server can be seen by clicking on a server.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
https://localhost:9443/adminCenter/
Google
All Servers
All Servers
2 Servers
2 Running
0 Stopped

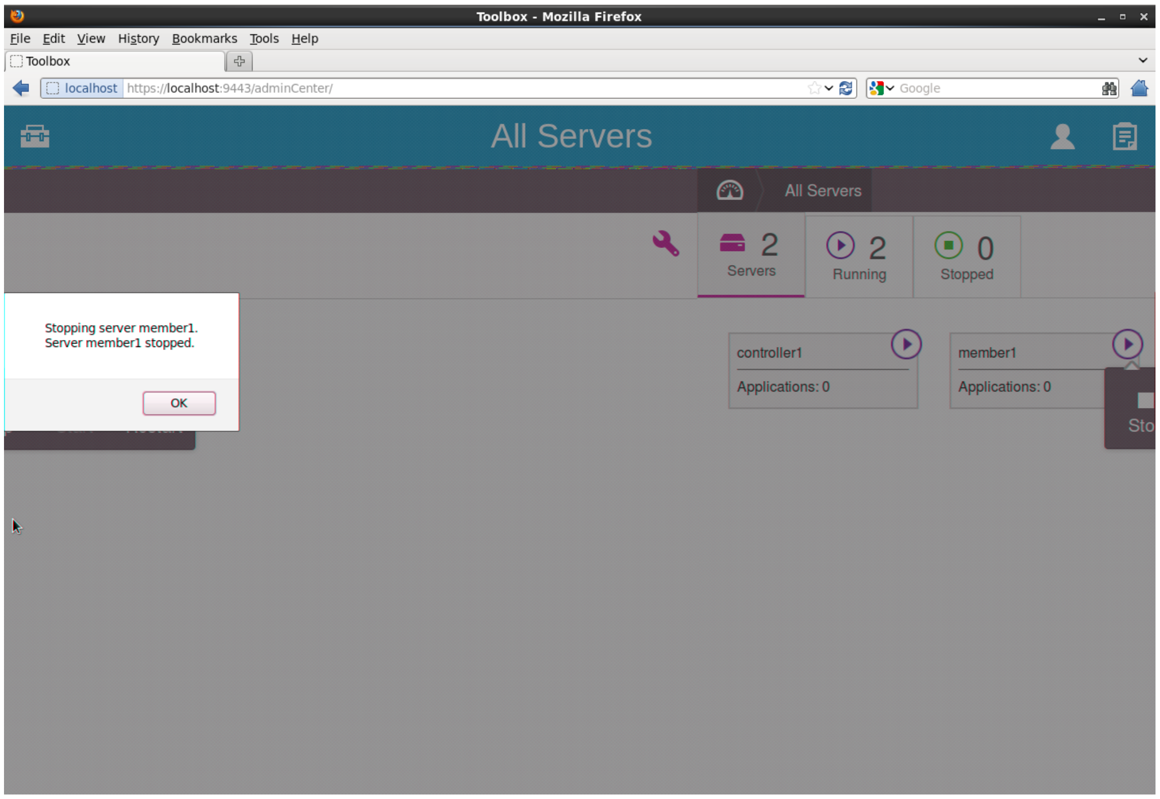
controller1
Applications: 0

member1
Applications: 0

3.4.6 Stop and restart member1 through the tool.

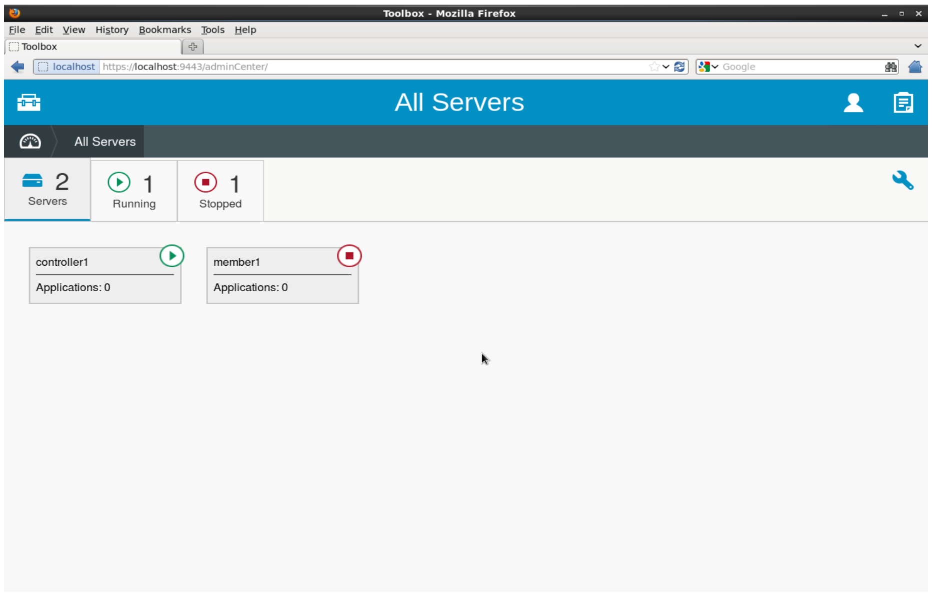
Operations can be performed on any server, cluster or application in the collective via the Admin Center. In this step, stop member1 by clicking on the green arrow icon in the upper right corner of the member1 card, and select stop.

Result of the stop operation:

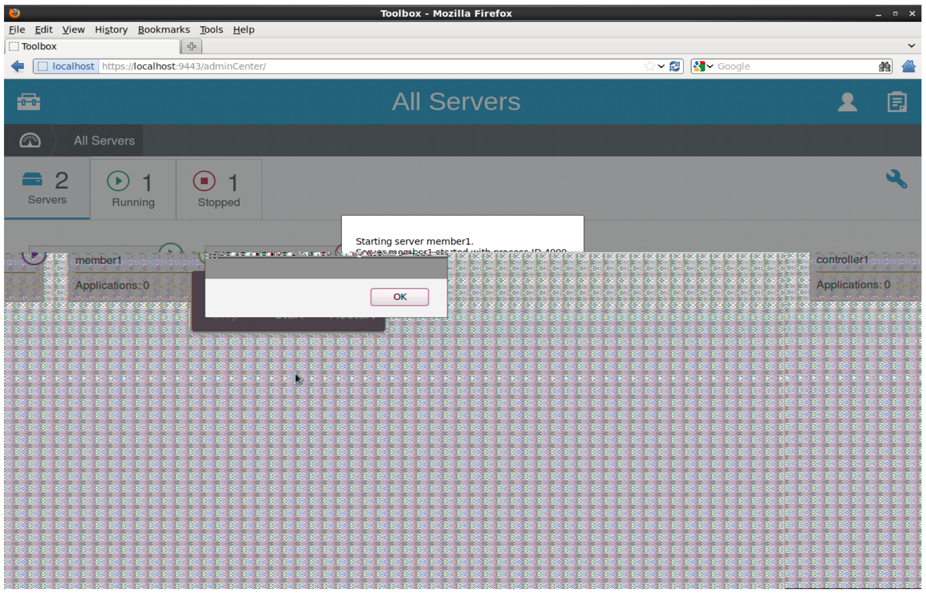


Stopping server member1.
Server member1 stopped.

The server can be restarted by repeating the steps and clicking on the red stopped icon and selecting the start operation.



Result of the start operation:



Starting server member1.
Server member1 started with process ID 4000.

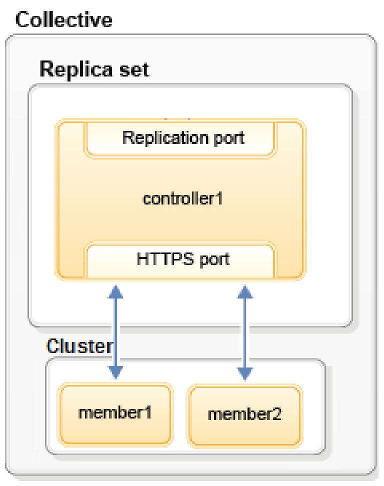
Be sure to leave member1 running at the end of this section.

Various information and multiple actions are available through the Admin Center. Not all aspects are examined in this lab.

3.5 Create a cluster

Approximate time: 15 - 30 minutes

In this section, you expand the collective by adding a new member, configure both members to join the default cluster, and finally use Jython scripting and the Admin Center to perform operations on the cluster.



3.5.1 Create a second member to join the collective.

[illegible]

3.5.2 Join member2 to the collective

[illegible]

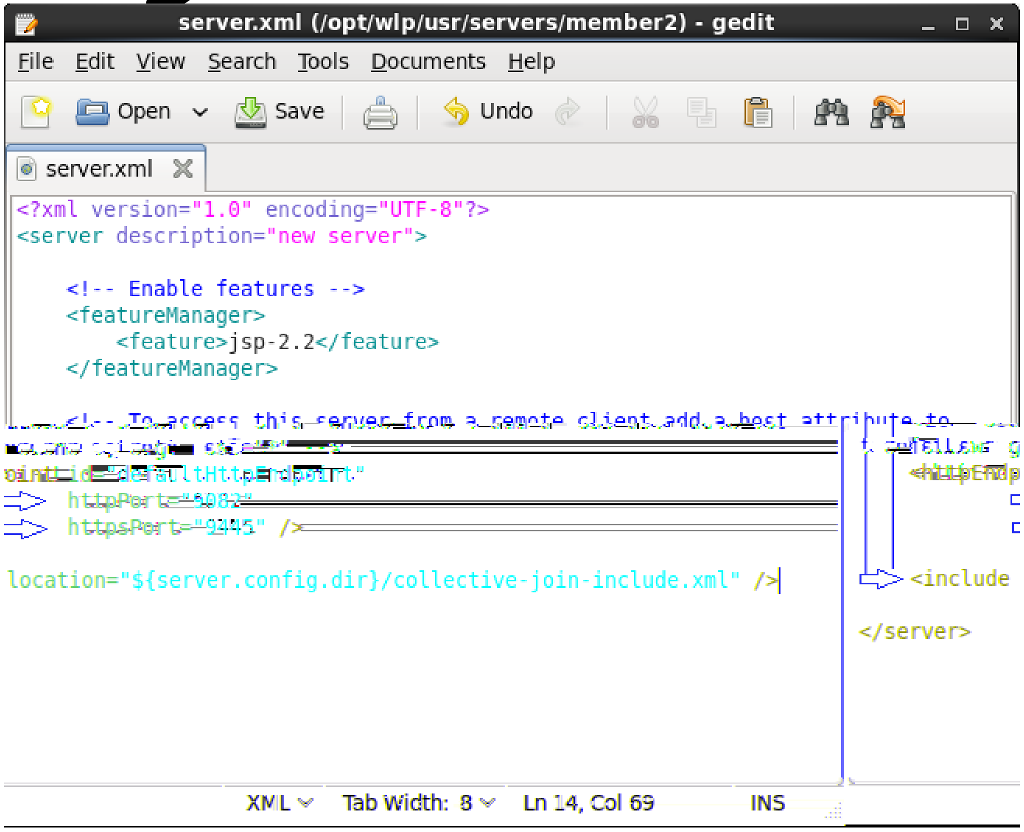
```
[illegible] y (hit Enter)
```

[illegible]

3.5.3 Update the server.xml for member2.

Change the HTTP and HTTPS ports to 9082 and 9445.

Add an include for the file generated by the join operation:


server.xml (/opt/wlp/usr/servers/member2) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
server.xml
XML Tab Width: 8 Ln 14, Col 69 INS

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9082"
                  httpsPort="9445" />

    <include location="${server.config.dir}/collective-join-include.xml" />

</server>
```

Save and close the file.

3.5.4 Start member2.

3.5.5 Verify the member started and is publishing information to the controller.

'grep' for messages CWWK[illegible] and CWWK[illegible]

```
[illegible]
```

3.5.6 Assign member1 and member2 to the default cluster.

The cluster configuration is dynamically updated and published to the collective controller.

Modify [illegible]

```
[illegible]
```

Add the following configuration to each server's server.xml:

```
[illegible]
```

You now have the default cluster created, with member1 and member2 belonging to the defaultCluster group. Multiple clusters can be defined with a single collective, but a server may only belong to one cluster group at a time.

3.5.7 Get status for "defaultCluster" via scripting.

The controller provides the `ClusterManagerMBean` which defines operations for obtaining information about defined clusters, starting and stopping all of the servers in a cluster, and for generating plugin-cfg.xml files.

The provided getClusterStatus.py script uses various `ClusterManagerMBean` operations to determine the defined clusters and the members which belong to the defaultCluster.

[illegible]

3.5.8 Stop cluster using the stopCluster.py sample script from wasdev.net

Multiple sample scripts are available for download on wasdev.net
The sample scripts are available in /opt/lab_materials/jython/sample_scripts

[illegible]

[illegible]

3.5.9 Verify the servers in the cluster group have stopped.

[illegible]

[illegible]

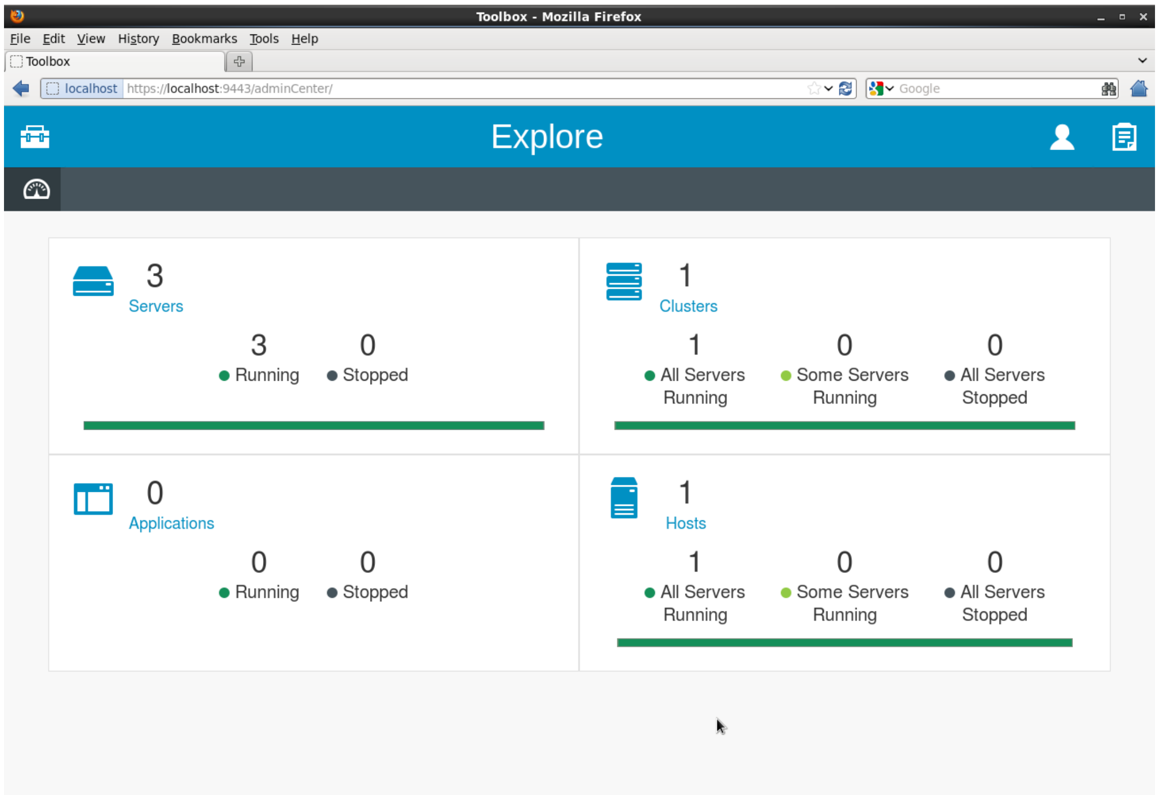
3.5.10 Start the cluster via the startCluster.py sample script.

```
[illegible]
```

The ClusterManager MBean is an example of how the collective controller acts as an operational repository. Data queries, such as the listClusterNames, listMembers and getStatus operations are performed entirely against the operational cache within the collective controller, while operations which require action on a target such as start and stop are performed against the respective collective member.

3.5.11 Launch the Explore tool from the Admin Center.

Now that a cluster is defined, the Explore dashboard will contain a card which shows information about the defined clusters.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Explore

3 Servers

3	0
● Running	● Stopped

1 Clusters

1	0	0
● All Servers Running	● Some Servers Running	● All Servers Stopped

0 Applications

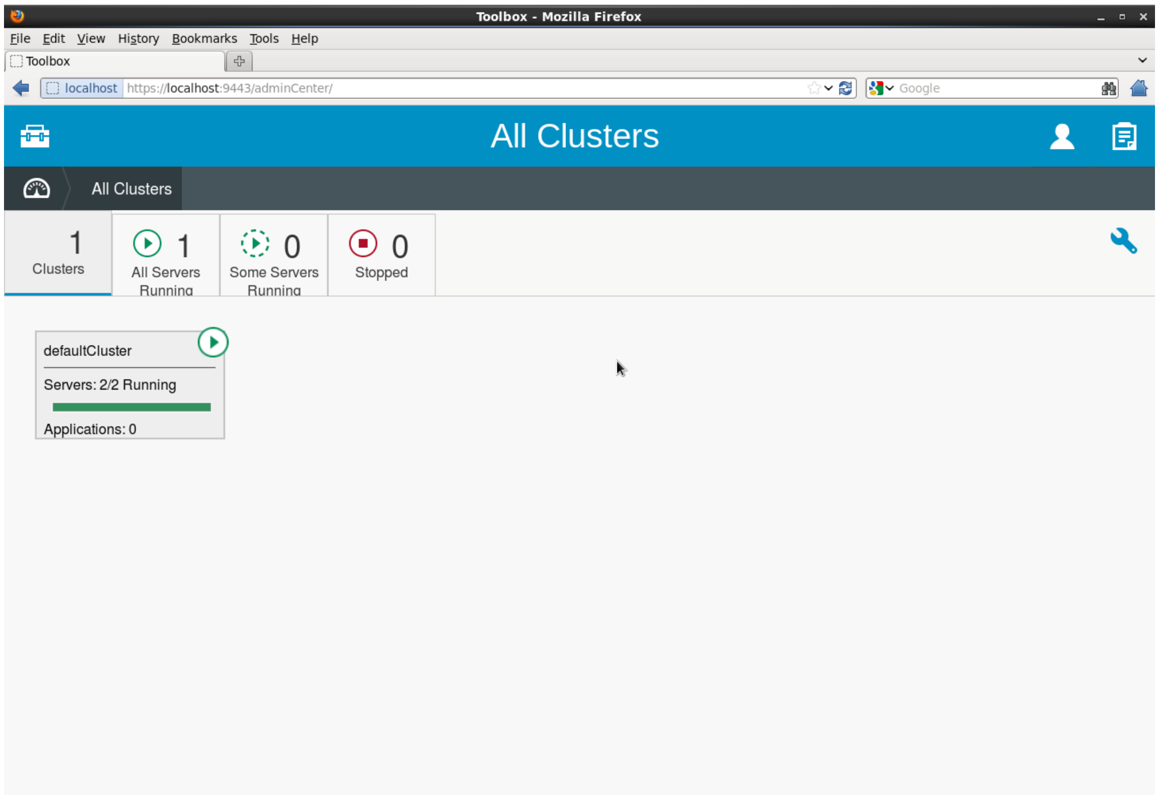
0	0
● Running	● Stopped

1 Hosts

1	0	0
● All Servers Running	● Some Servers Running	● All Servers Stopped

3.5.12 Click the Clusters category of the dashboard to see an overview of the defined clusters.

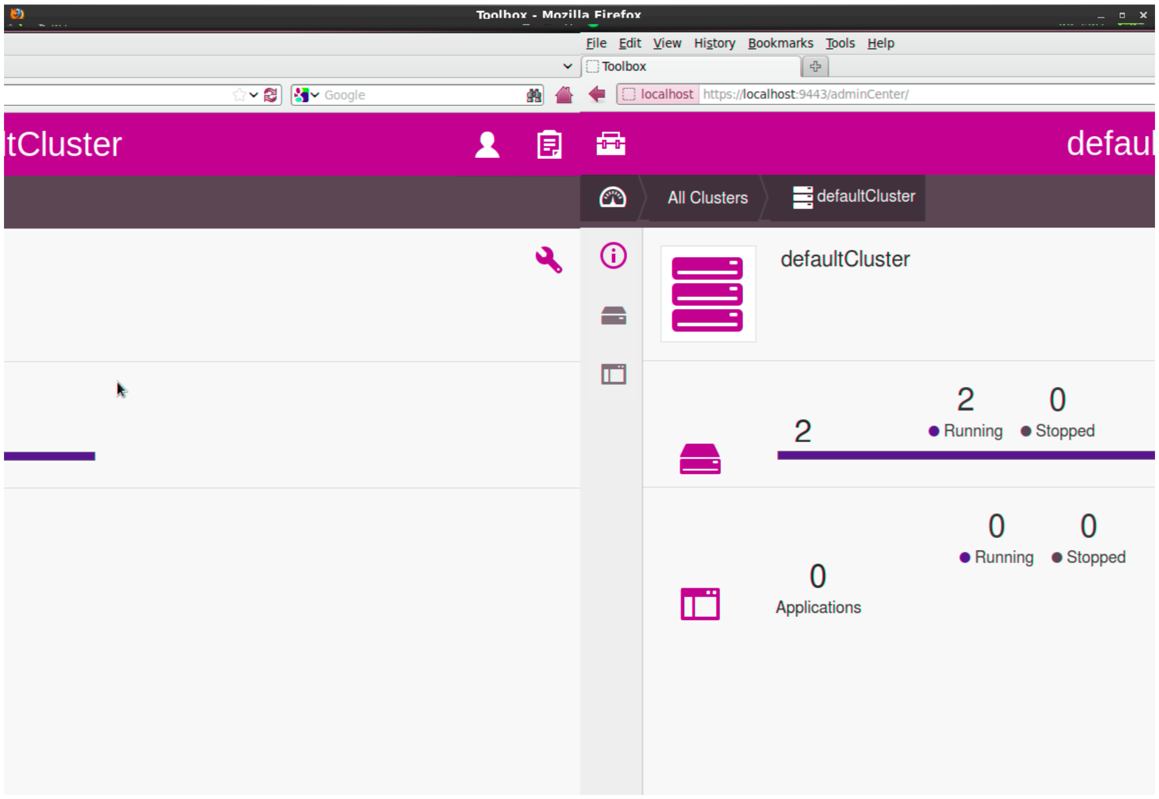
Click the defaultCluster card to see details about the cluster.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
All Clusters
All Clusters
1 Clusters
1 All Servers Running
0 Some Servers Running
0 Stopped

defaultCluster
Servers: 2/2 Running
Applications: 0

3.5.13 The defaultCluster details page shows the details of the cluster.

No applications are yet deployed to the cluster. Applications will be deployed in the next section.



3.6 Deploying applications to the cluster

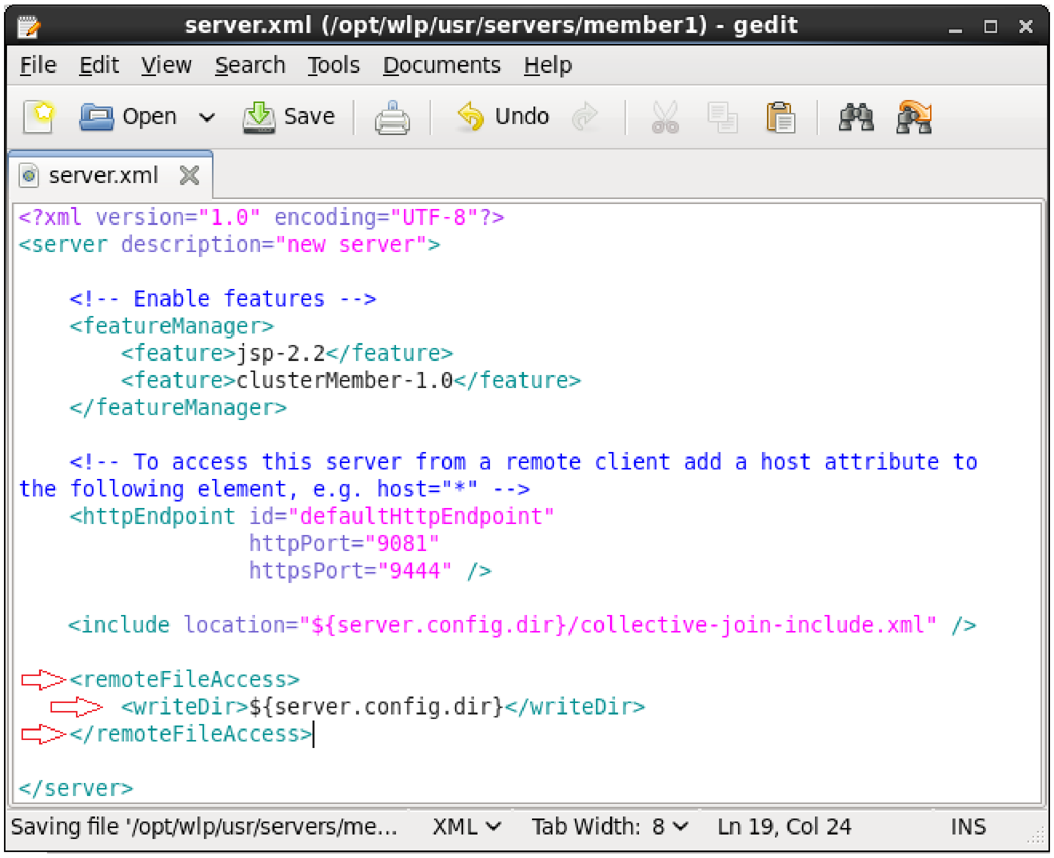
Approximate time: 20 – 40 minutes

In this section, you build upon the work done in previous sections to deploy applications and configuration to all members of a cluster. This section demonstrates the type of compound operations which are possible with the MBeans available in WebSphere Application Server Liberty Profile 8.5.5, as well as describes a best practice for cluster configuration.

3.6.1 Update the server.xml for each member to support file transfer.

By default, servers only support read-only file transfer from their configuration directory. Subsequent steps will use the file transfer service to deploy applications and configuration to the cluster.

Add the following lines to the server.xml. These lines will enable the file transfer service to write to the servers configuration directory.


server.xml (/opt/wlp/usr/servers/member1) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>clusterMember-1.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9081"
                  httpsPort="9444" />

    <include location="${server.config.dir}/collective-join-include.xml" />

    <remoteFileAccess>
        <writeDir>${server.config.dir}</writeDir>
    </remoteFileAccess>

</server>
```

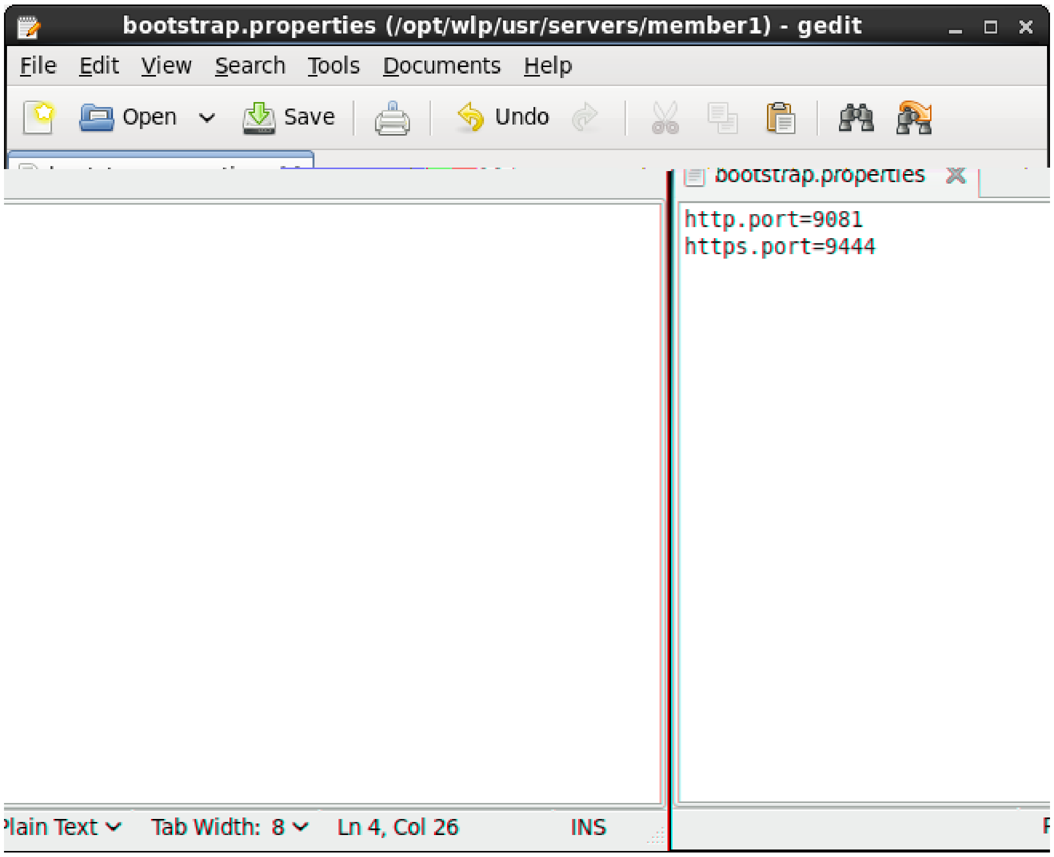
Saving file '/opt/wlp/usr/servers/me...   XML   Tab Width: 8   Ln 19, Col 24   INS

3.6.2 Create a bootstrap.properties file for each cluster member.

This file will contain the configuration which is unique to each server, such as port numbers. This will support using a common server.xml for all of the cluster members.

Set the contents of each bootstrap.properties as described below.

[illegible]


bootstrap.properties (/opt/wlp/usr/servers/member1) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
bootstrap.properties
Plain Text Tab Width: 8 Ln 4, Col 26 INS

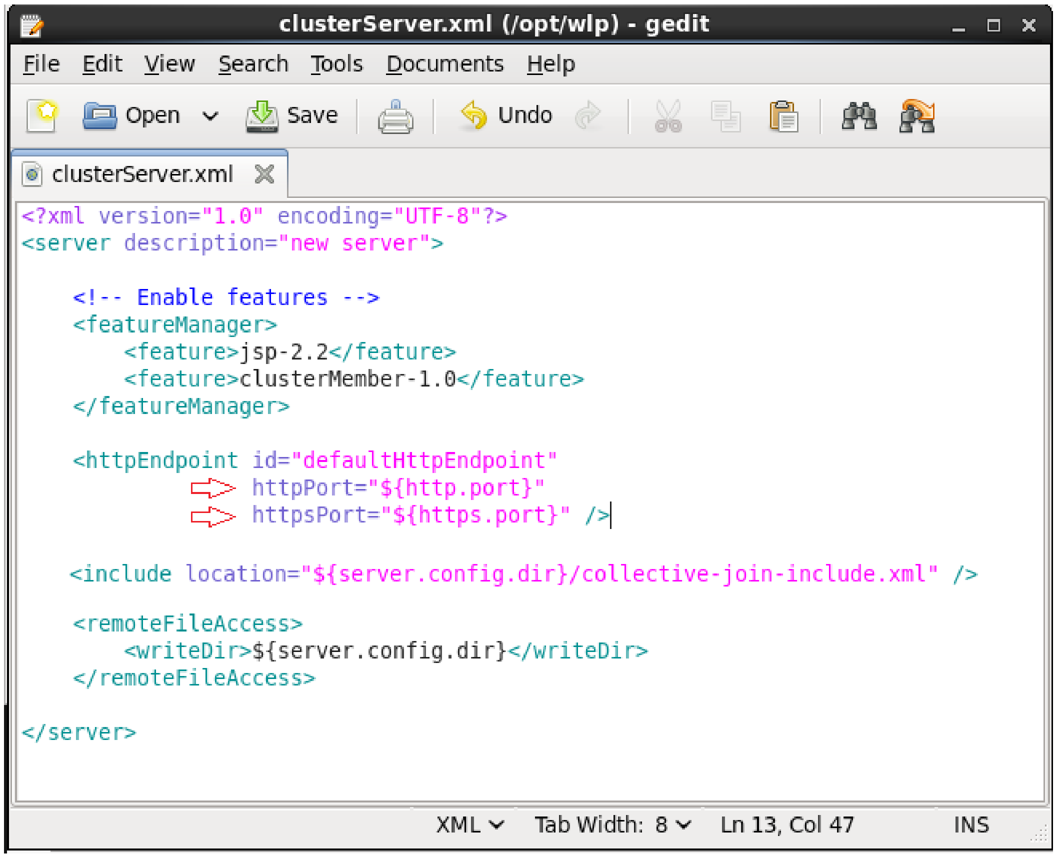
```
http.port=9081
https.port=9444
```

Save and close the files.

3.6.3 Copy the server.xml for member1 and create a new generic server.xml.

The generic server.xml will use the new values defined in the bootstrap.properties file. This copy will serve as the basis for the common cluster member configuration and will use variable substitution for the server specific configuration values.

Modify the following XML values to use the new values in bootstrap.properties:


clusterServer.xml (/opt/wlp) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
clusterServer.xml
XML Tab Width: 8 Ln 13, Col 47 INS

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>clusterMember-1.0</feature>
    </featureManager>

    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="${http.port}"
                  httpsPort="${https.port}" />

    <include location="${server.config.dir}/collective-join-include.xml" />

    <remoteFileAccess>
        <writeDir>${server.config.dir}</writeDir>
    </remoteFileAccess>

</server>
```

Save and close the file.

3.6.4 Push the application "snoop" to the cluster.

The deployAppToCluster.py script uses a combination of the ClusterManager and FileTransfer MBeans to push the application to all of the cluster members. This operation will push the application to ${server.config.dir}/apps/ for each cluster member.

This script does not alter the server.xml for the targets. The server.xml configuration will be updated in the next step.

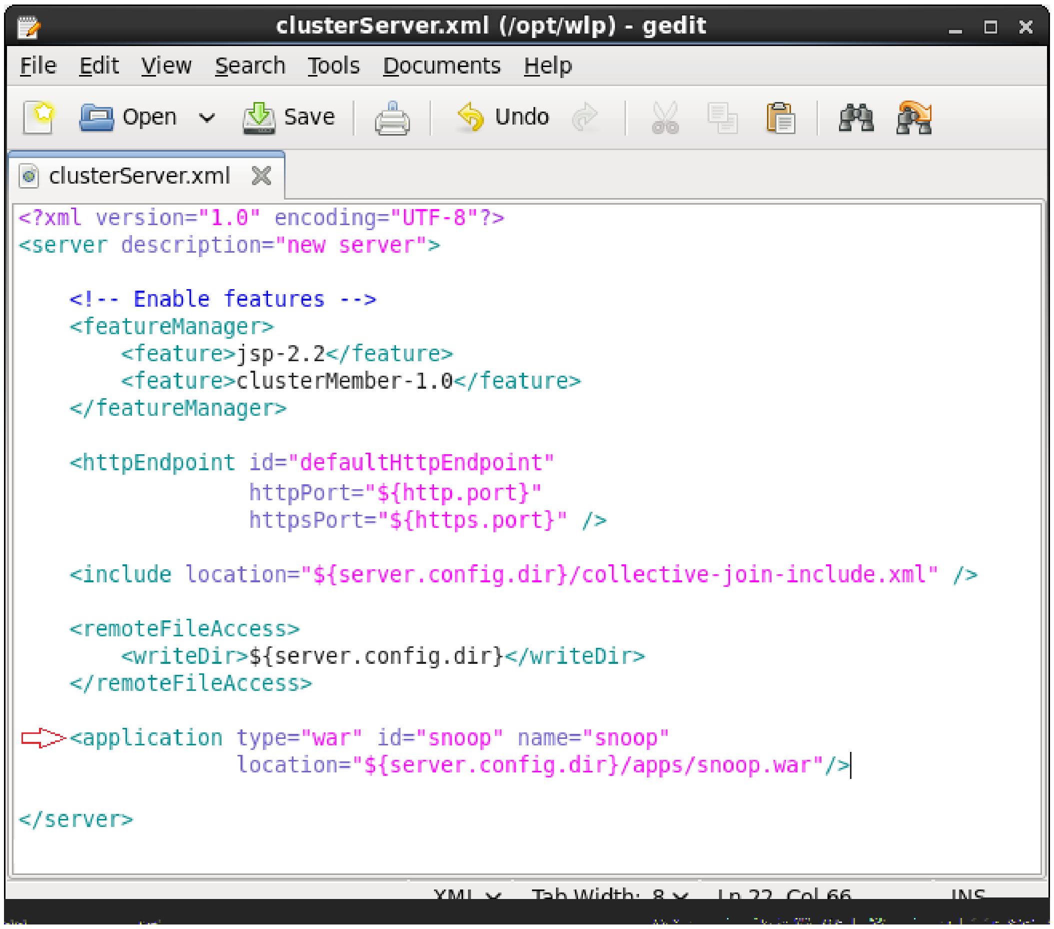
[illegible]

[illegible]

3.6.5 Update the common cluster configuration for the application "snoop".

Applications can be explicitly configured in the server.xml using the [illegible]

Add the following lines to the clusterServer.xml [illegible]


clusterServer.xml (/opt/wlp) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
clusterServer.xml
XML Tab Width: 8 Ln 22, Col 66 INS

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
        <feature>clusterMember-1.0</feature>
    </featureManager>

    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="${http.port}"
                  httpsPort="${https.port}" />

    <include location="${server.config.dir}/collective-join-include.xml" />

    <remoteFileAccess>
        <writeDir>${server.config.dir}</writeDir>
    </remoteFileAccess>

    <application type="war" id="snoop" name="snoop"
                location="${server.config.dir}/apps/snoop.war"/>

</server>
```

Save and close the file.

3.6.6 Restart the cluster members.

This step is required as the bootstrap.properties file is not dynamically loaded. The bootstrap.properties is processed only during JVM start up.

[illegible]

Command output omitted.
See previous steps 3.5.8 and 3.5.10 for the expected output.

3.6.7 Push the common cluster configuration to all of the members of the cluster.

This step will update the server.xml of all the cluster members. Each cluster member will dynamically update its runtime configuration after the file transfer completes.

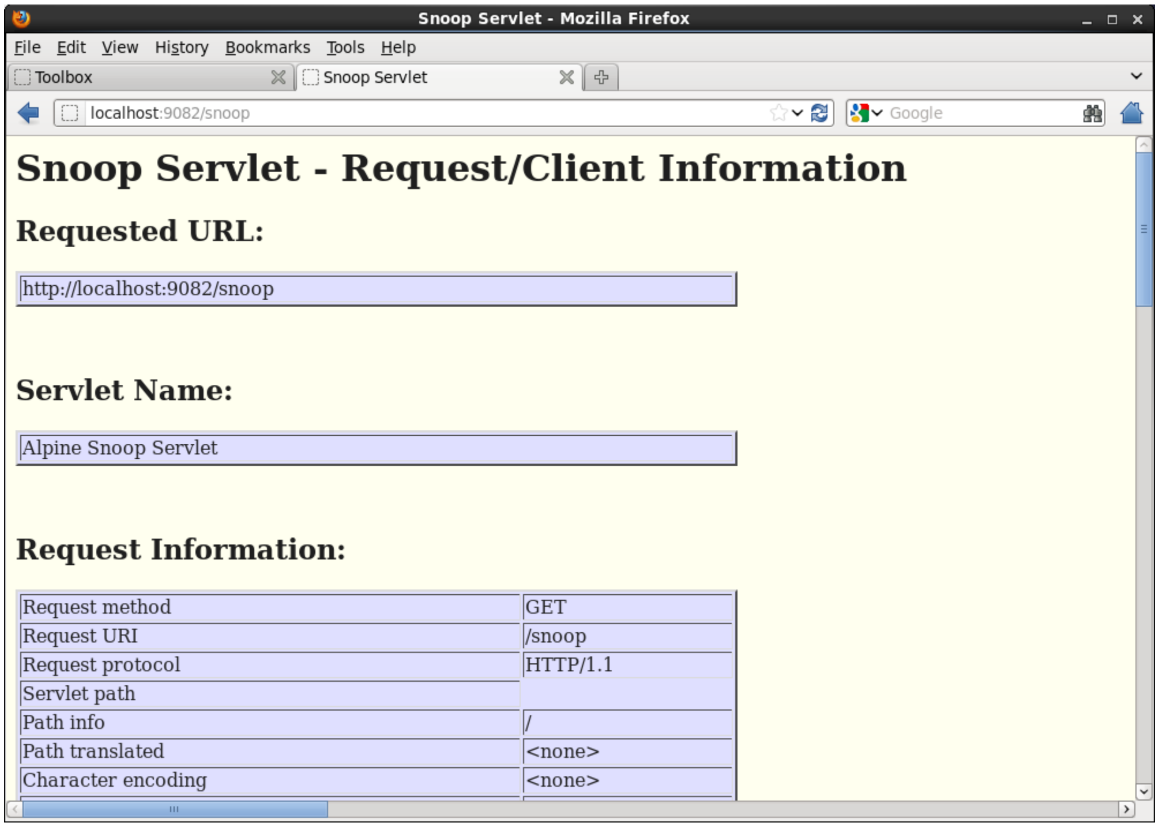
[illegible]

[illegible]

3.6.8 Access the application "snoop" running on member1 and member2.

The application is now available on both members of the cluster.
It is out of the scope of this lab, but IBM HTTP Server (IHS) can be used to perform load balancing across the cluster members.

Run firefox, go to URLs: http://localhost:9081/snoop
http://localhost:9082/snoop


Snoop Servlet - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
Snoop Servlet
localhost:9082/snoop
Google

Snoop Servlet - Request/Client Information

Requested URL:

http://localhost:9082/snoop

Servlet Name:

Alpine Snoop Servlet

Request Information:

Request method	GET
Request URI	/snoop
Request protocol	HTTP/1.1
Servlet path	
Path info	/
Path translated	<none>
Character encoding	<none>

Close the browser.

3.6.9 Install the application "ImpactWeb" to the cluster.

This operation combines application deploy and configuration file updates to demonstrate an alternate pattern for installing an application to a cluster.

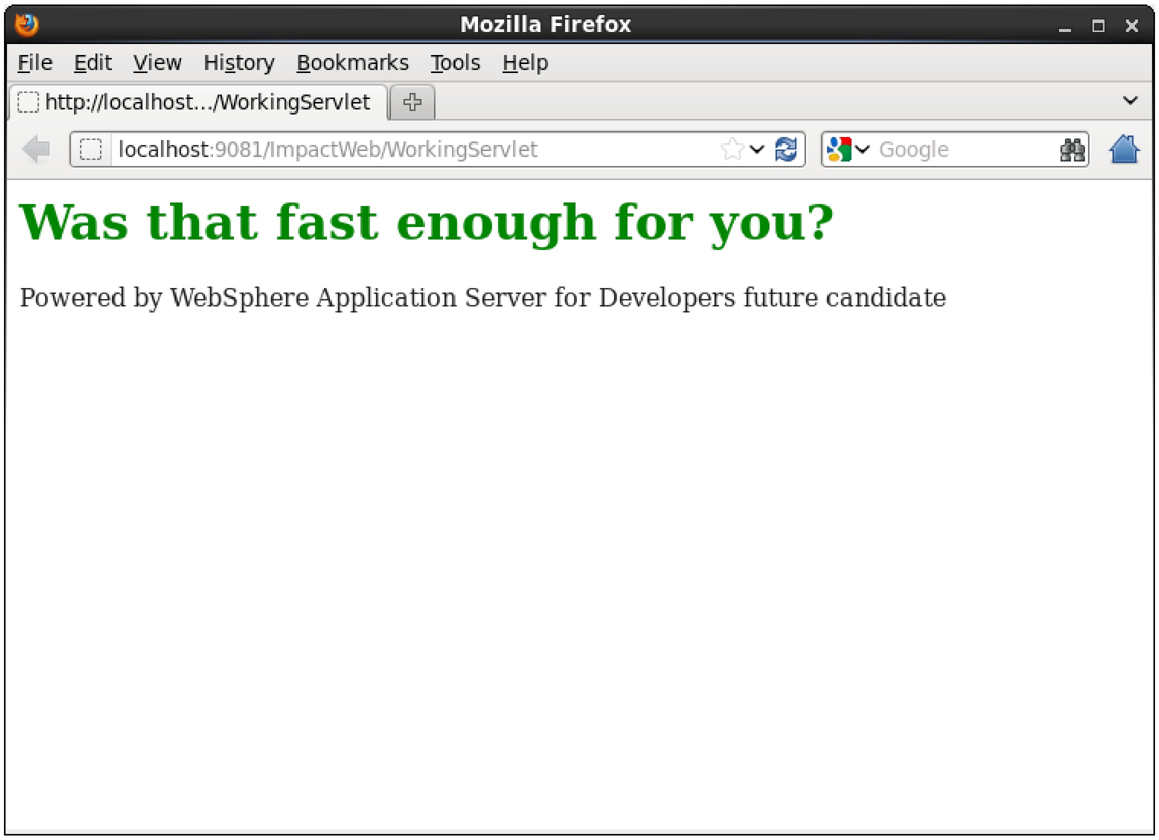
This deployment option is limited by the simple application configuration. For more complex application configurations, it is recommended to follow steps similar to 3.6.4 and 3.6.[illegible], as those steps provide for the most flexibility in the application [illegible]

[illegible]

```
[illegible]
```

3.6.10 Access the application "ImpactWeb" running on member1 and member2.

Run firefox, go to URLs: http://localhost:9081/ImpactWeb/WorkingServlet
http://localhost:9082/ImpactWeb/WorkingServlet


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost.../WorkingServlet
localhost:9081/ImpactWeb/WorkingServlet
Google

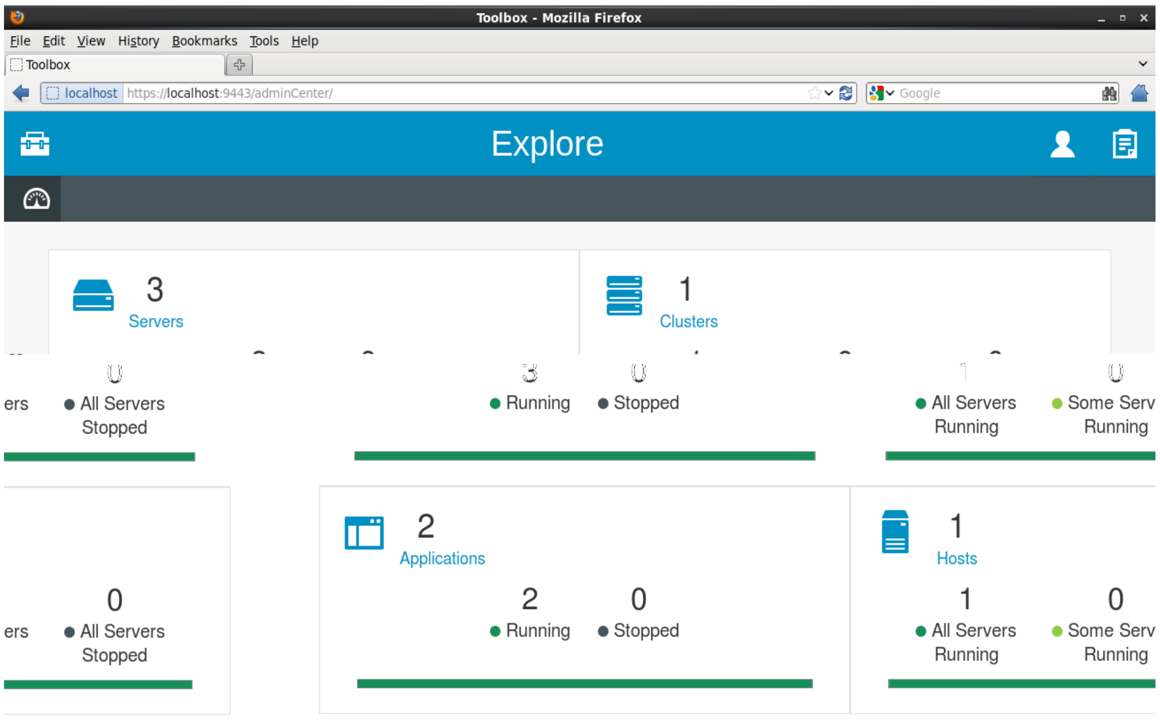
Was that fast enough for you?

Powered by WebSphere Application Server for Developers future candidate

Close the browser.

3.6.11 Launch Explore tool via the Admin Center.

The dashboard now has information about the defined servers, the cluster to which they belong, and the applications deployed.



Close the browser.

You now have an active cluster group with the "snoop" and "ImpactWeb" applications deployed. The cluster configuration is written in such a way that new cluster members can be easily added by creating new servers and setting their bootstrap.properties file accordingly. An alternate choice to support common configuration is to use include files supported by the server.xml. The server specific configuration can be stored in a separate include file, and the common configuration can be stored in the common server.xml.

3.7 Deploying server packages via Admin Center

Approximate time: 10 – 15 minutes

In this section, you use the Admin Center to deploy a server package and join the deployed server to the collective. A server package can be deployed to any host that is registered with the collective. The Deploy tool uses the collective file transfer operations. In order to transfer a file to a host, the host must be registered with the collective and the transfer paths must be specified.

3.7.1 Update the localhost configuration to define writable file transfer paths.

[illegible]

3.7.2 Create a server to be deployed.

[illegible]

3.7.3 Copy the sample application "snoop" to the server to be deployed.

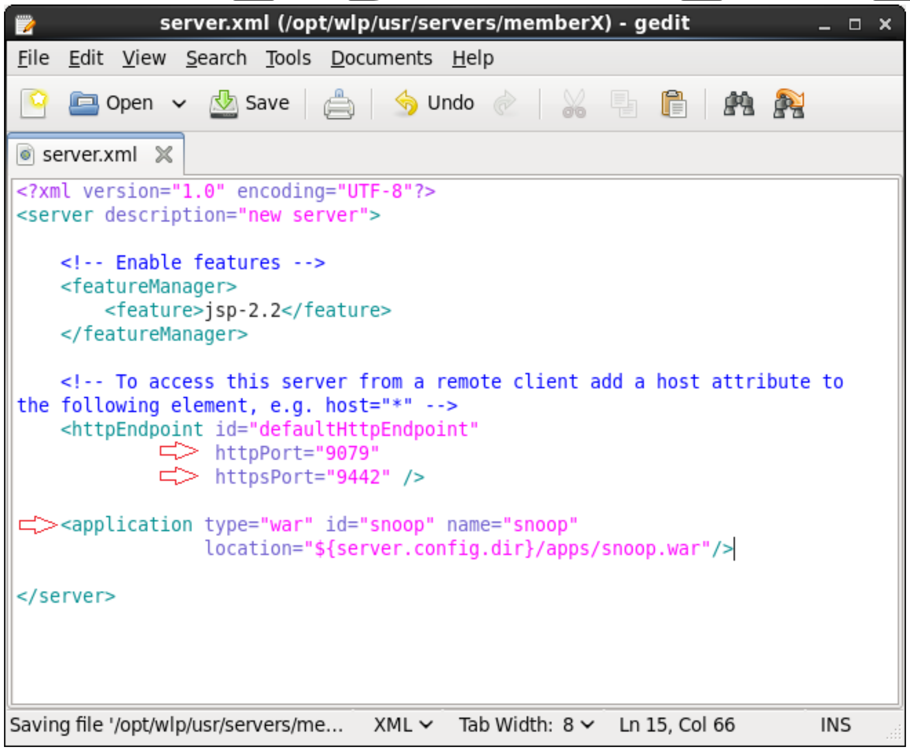
[illegible]

3.7.4 Configure the server to be deployed.

[illegible]

Change the HTTP and HTTPS ports to 9079 and 9442 and configure the app.

[illegible]


server.xml (/opt/wlp/usr/servers/memberX) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute to
the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9079"
                  httpsPort="9442" />

    <application type="war" id="snoop" name="snoop"
                 location="${server.config.dir}/apps/snoop.war"/>

</server>
```

Saving file '/opt/wlp/usr/servers/me... XML Tab Width: 8 Ln 15, Col 66 INS

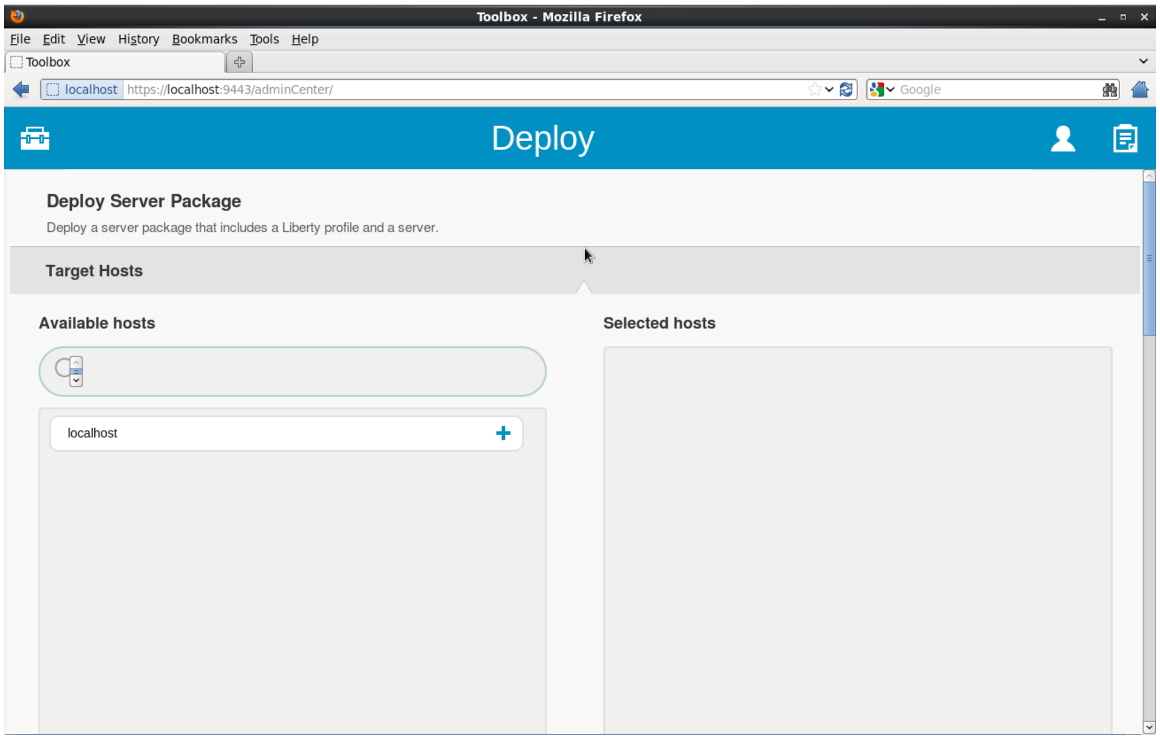
3.7.5 Package the server.

Packaging the server will create a self-contained copy of the entire Liberty installation which includes the server's configuration. This self-contained package will be deployed to a host in the next steps.

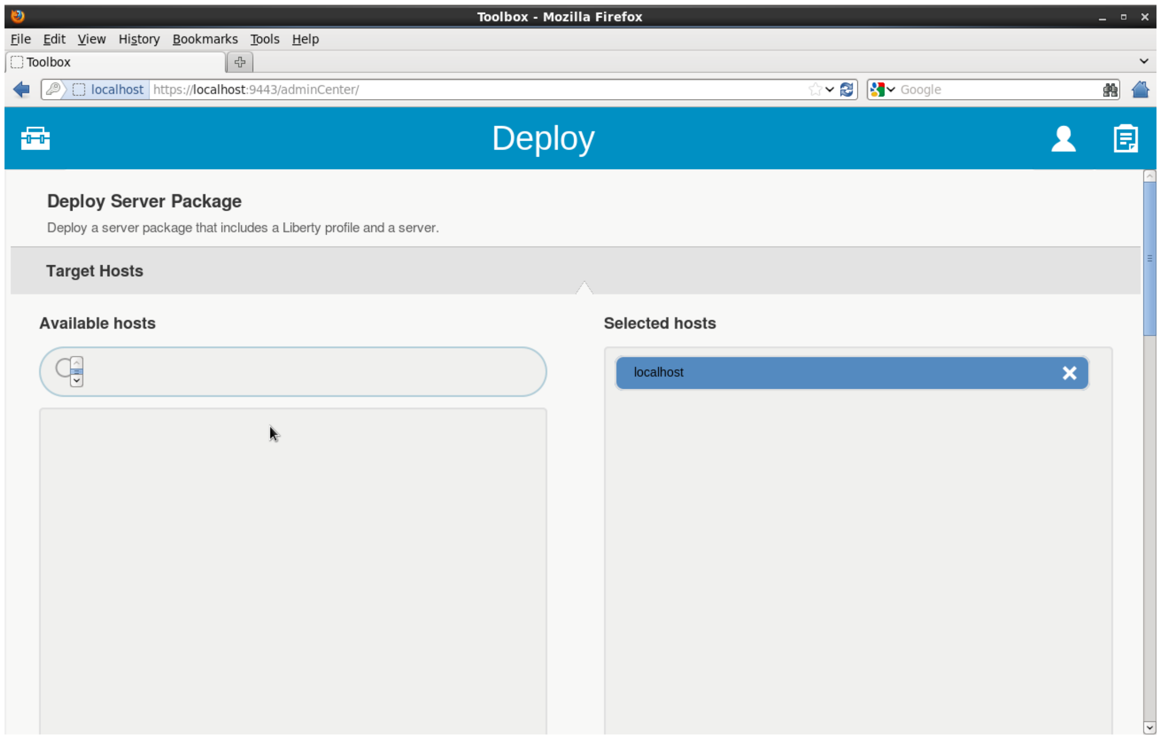
[illegible]

3.7.6 Launch the Deploy tool from the Admin Center.

In this lab, the only available host is localhost. In a real environment, additional hosts can be added as deployment targets.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy
Deploy Server Package
Deploy a server package that includes a Liberty profile and a server.
Target Hosts
Available hosts
Selected hosts
localhost

3.7.7 Select 'localhost' from the list of available hosts.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

Deploy Server Package

Deploy a server package that includes a Liberty profile and a server.

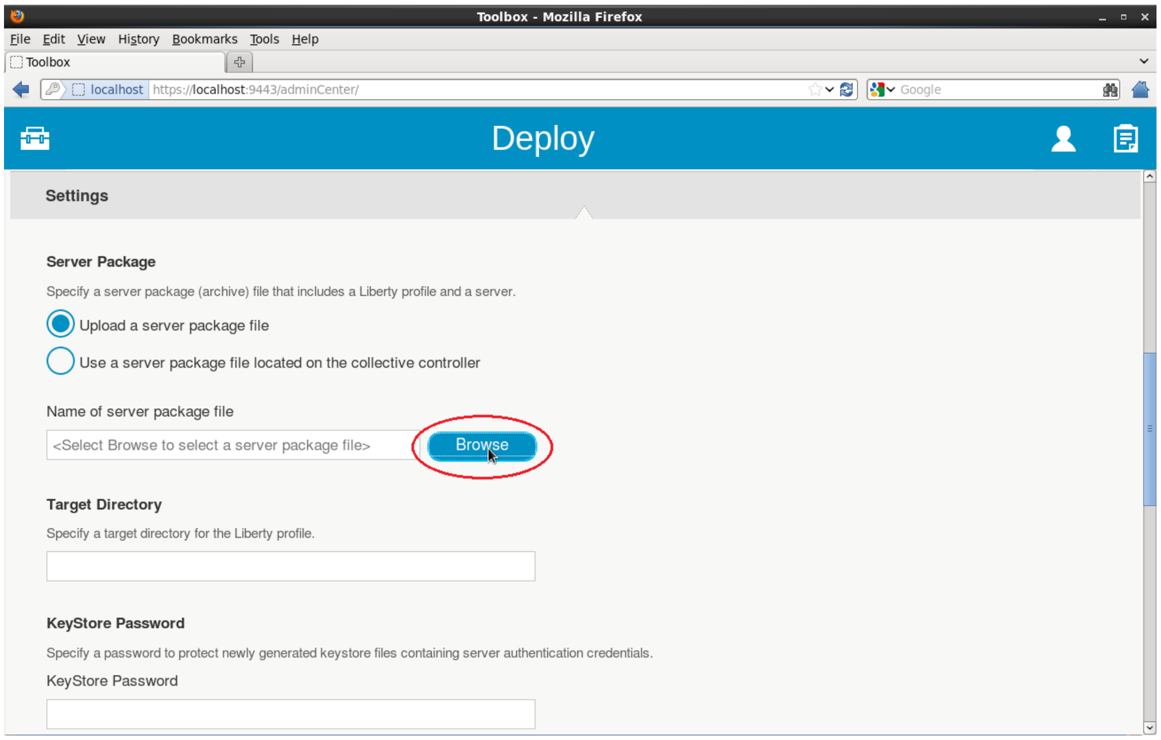
Target Hosts

Available hosts

Selected hosts

localhost

3.7.8 Scroll down and click 'Browse'.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

Settings

Server Package

Specify a server package (archive) file that includes a Liberty profile and a server.

- [✓ Upload a server package file]
- Use a server package file located on the collective controller

Name of server package file

<Select Browse to select a server package file> Browse

Target Directory

Specify a target directory for the Liberty profile.

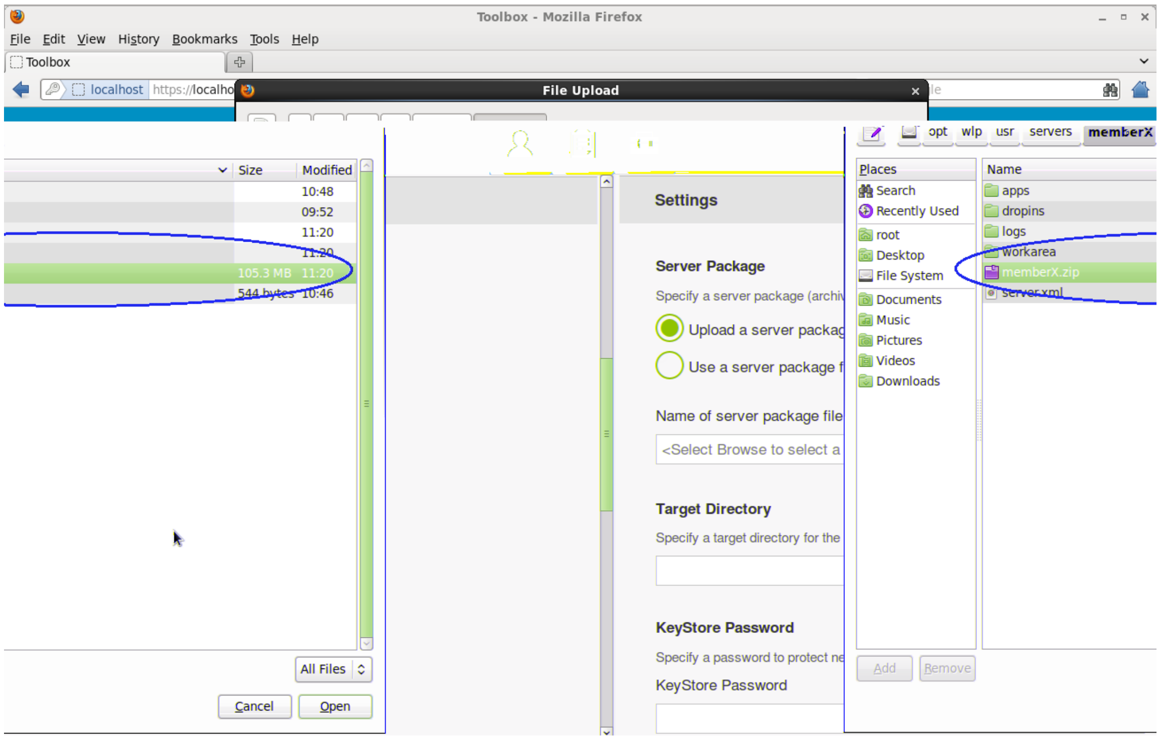
KeyStore Password

Specify a password to protect newly generated keystore files containing server authentication credentials.

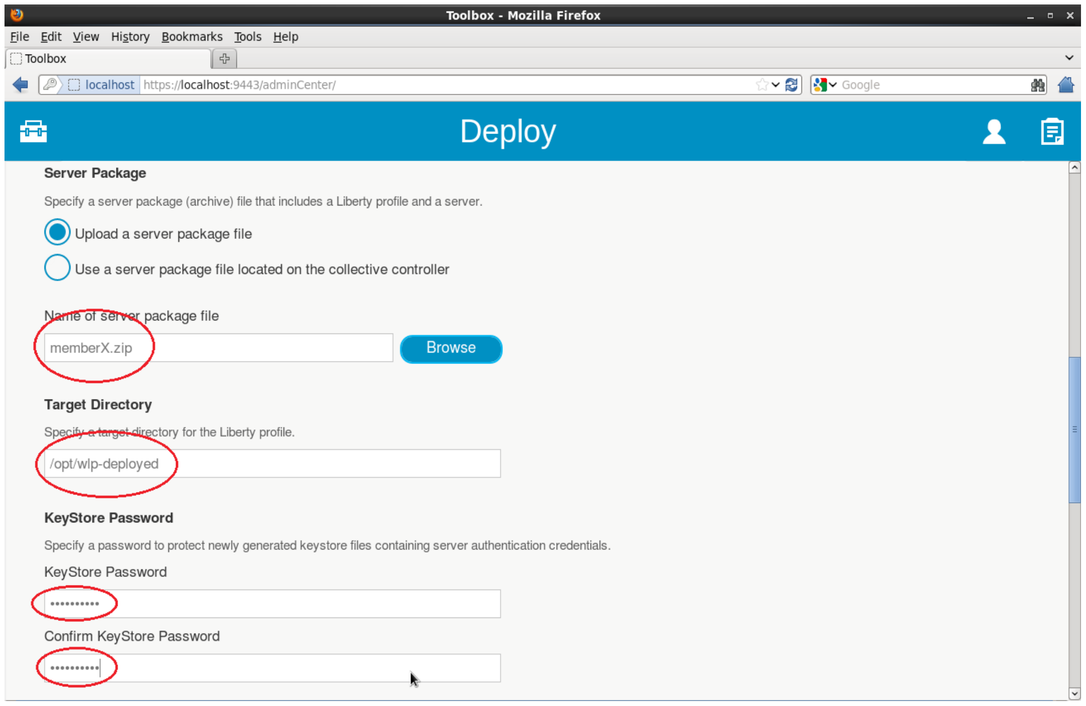
KeyStore Password

3.7.9 Upload memberX.zip.

Navigate to /opt/wlp/usr/servers/memberX via the File System button on the File Upload navigator's side bar.



3.7.10 Set the target installation directory to **/opt/wlp-deployed**
Set the keystore passwords to **Impact2014**


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

Server Package

Specify a server package (archive) file that includes a Liberty profile and a server.

[✓ Upload a server package file]

Use a server package file located on the collective controller

Name of server package file

memberX.zip

Browse

Target Directory

Specify a target directory for the Liberty profile.

/opt/wlp-deployed

KeyStore Password

Specify a password to protect newly generated keystore files containing server authentication credentials.

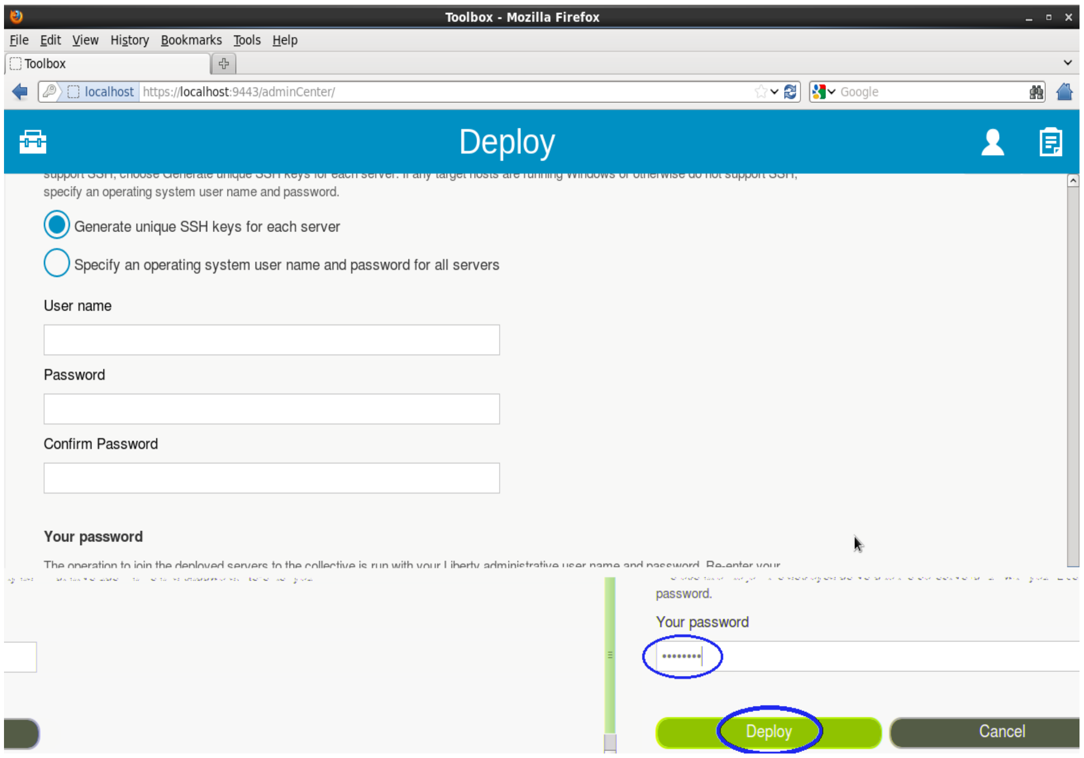
KeyStore Password

••••••••••

Confirm KeyStore Password

••••••••••

3.7.11 Enter the admin password '**adminpwd**' and click 'Deploy'


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

support SSH, choose Generate unique SSH keys for each server. If any target hosts are running Windows or otherwise do not support SSH, specify an operating system user name and password.

[✓ Generate unique SSH keys for each server]

Specify an operating system user name and password for all servers

User name

Password

Confirm Password

Your password

The operation to join the deployed servers to the collective is run with your Liberty administrative user name and password. Re-enter your password.

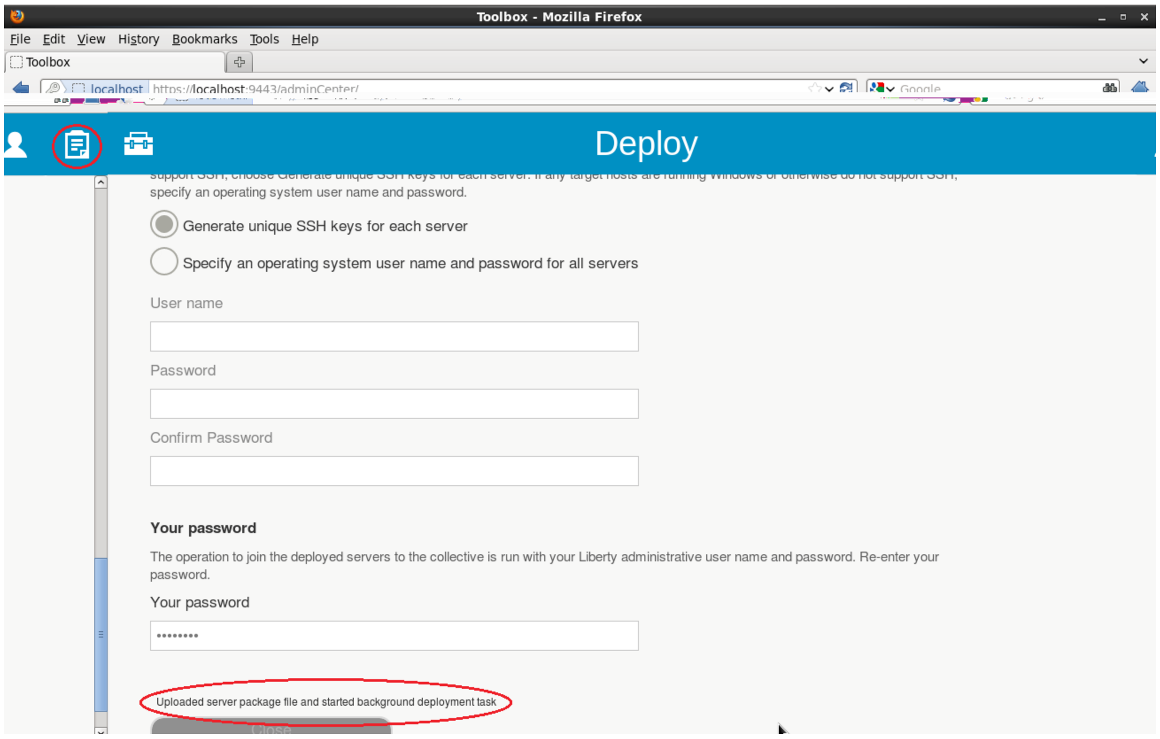
Your password

••••••••

Deploy

Cancel

3.7.12 Once the upload completes, select the background task button in the upper right-hand corner to view the background tasks.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

specify an operating system user name and password.

Generate unique SSH keys for each server

Specify an operating system user name and password for all servers

User name

Password

Confirm Password

Your password

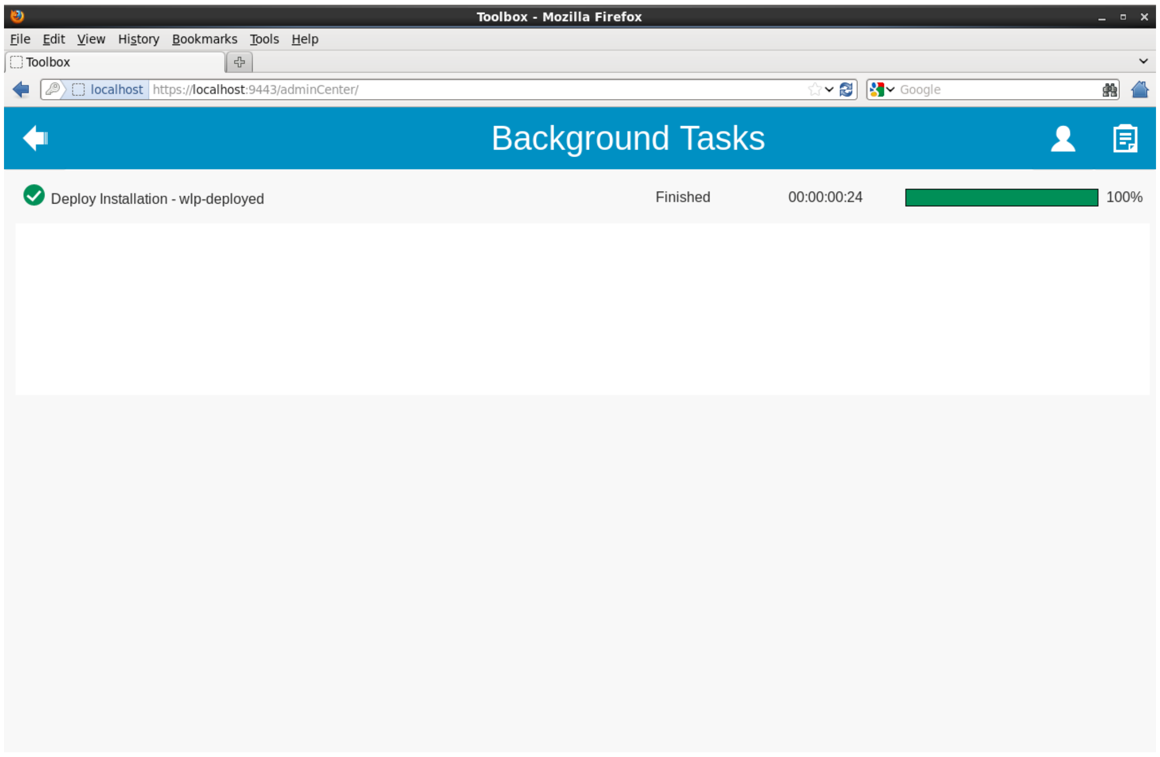
The operation to join the deployed servers to the collective is run with your Liberty administrative user name and password. Re-enter your password.

Your password

••••••••

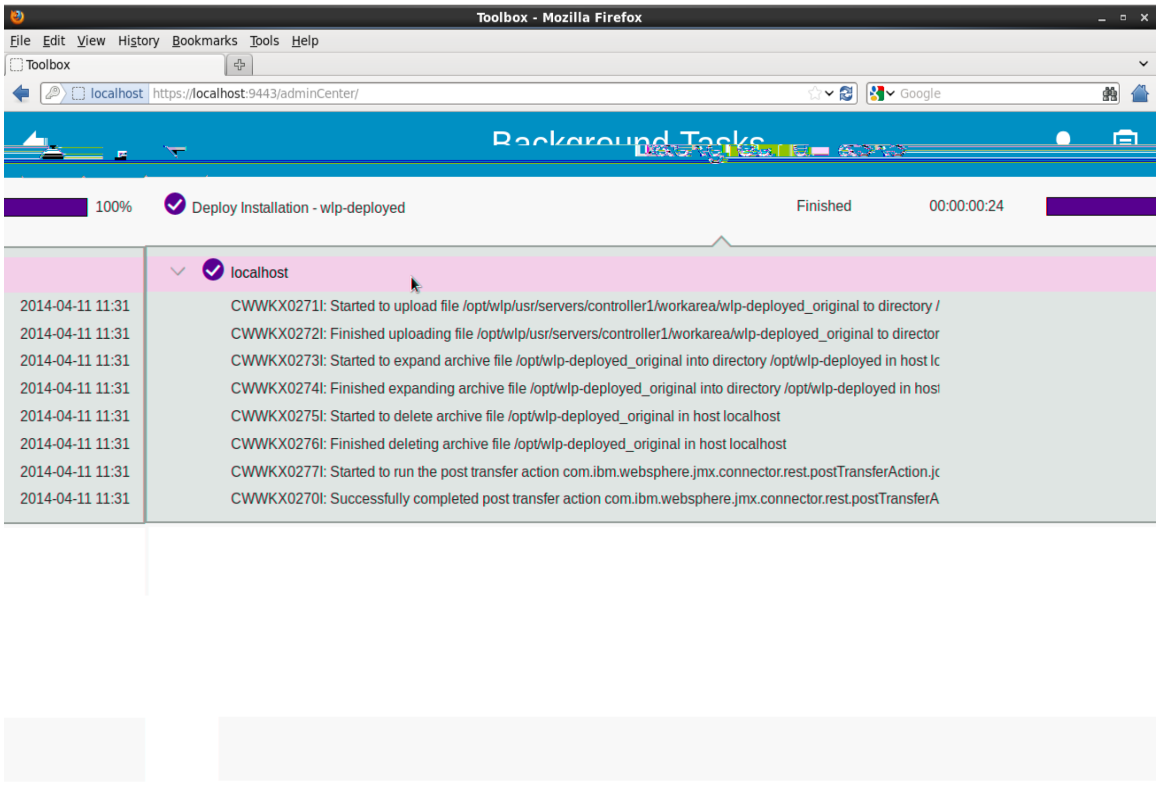
Uploaded server package file and started background deployment task

The Background Tasks page:


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Background Tasks

Deploy Installation - wlp-deployed Finished 00:00:00:24 100%

3.7.13 The background tasks can be expanded to see the details of the task.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Background Tasks

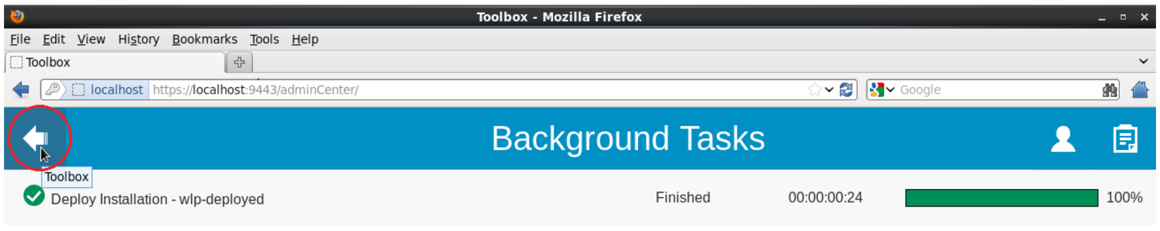
100% Deploy Installation - wlp-deployed Finished 00:00:00:24

localhost

```
2014-04-11 11:31  CWWKX0271I: Started to upload file /opt/wlp/usr/servers/controller1/workarea/wlp-deployed_original to directory /
2014-04-11 11:31  CWWKX0272I: Finished uploading file /opt/wlp/usr/servers/controller1/workarea/wlp-deployed_original to director
2014-04-11 11:31  CWWKX0273I: Started to expand archive file /opt/wlp-deployed_original into directory /opt/wlp-deployed in host lc
2014-04-11 11:31  CWWKX0274I: Finished expanding archive file /opt/wlp-deployed_original into directory /opt/wlp-deployed in host
2014-04-11 11:31  CWWKX0275I: Started to delete archive file /opt/wlp-deployed_original in host localhost
2014-04-11 11:31  CWWKX0276I: Finished deleting archive file /opt/wlp-deployed_original in host localhost
2014-04-11 11:31  CWWKX0277I: Started to run the post transfer action com.ibm.websphere.jmx.connector.rest.postTransferAction.jc
2014-04-11 11:31  CWWKX0270I: Successfully completed post transfer action com.ibm.websphere.jmx.connector.rest.postTransferA
```

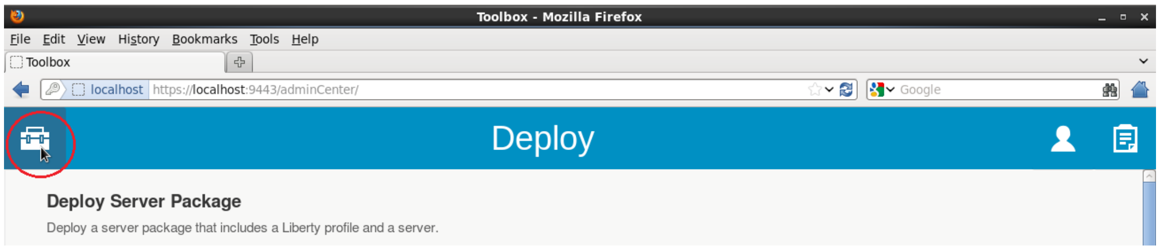
3.7.14 Return to the Explore tool.

Hit back button in the Background Tasks view.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Background Tasks
Toolbox

Deploy Installation - wlp-deployed Finished 00:00:00:24 100%

Hit the Toolbox button in the Deploy view.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
Deploy

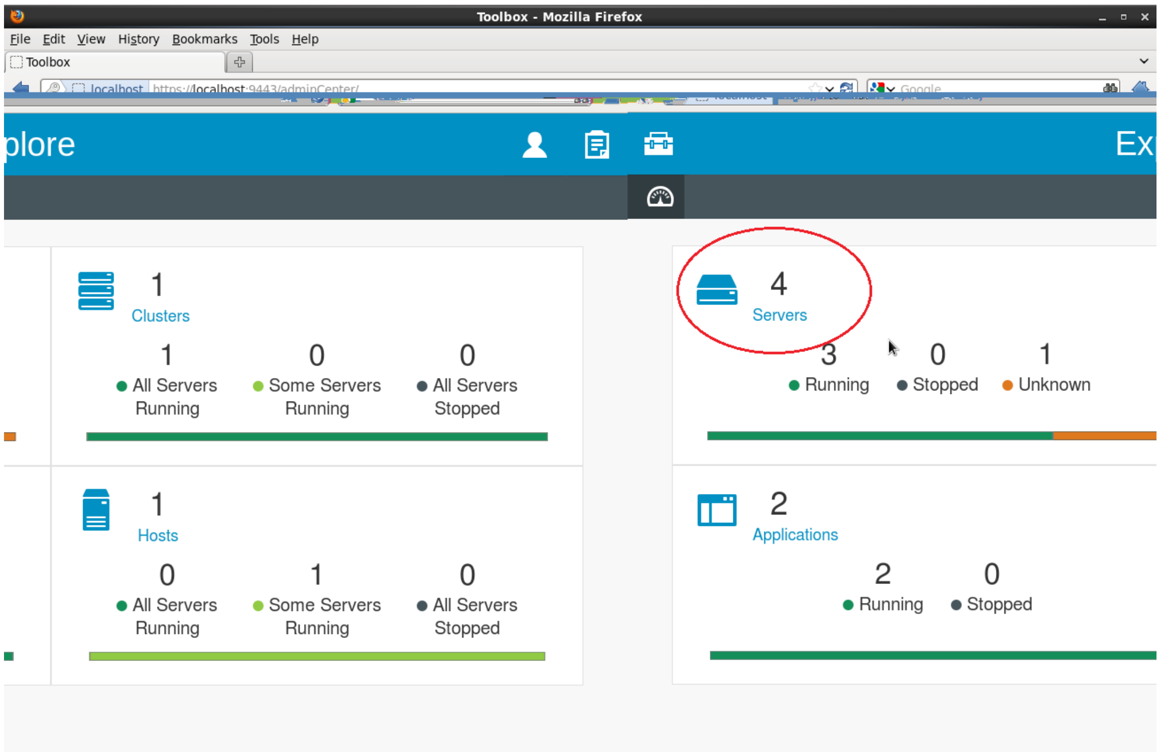
Deploy Server Package

Deploy a server package that includes a Liberty profile and a server.

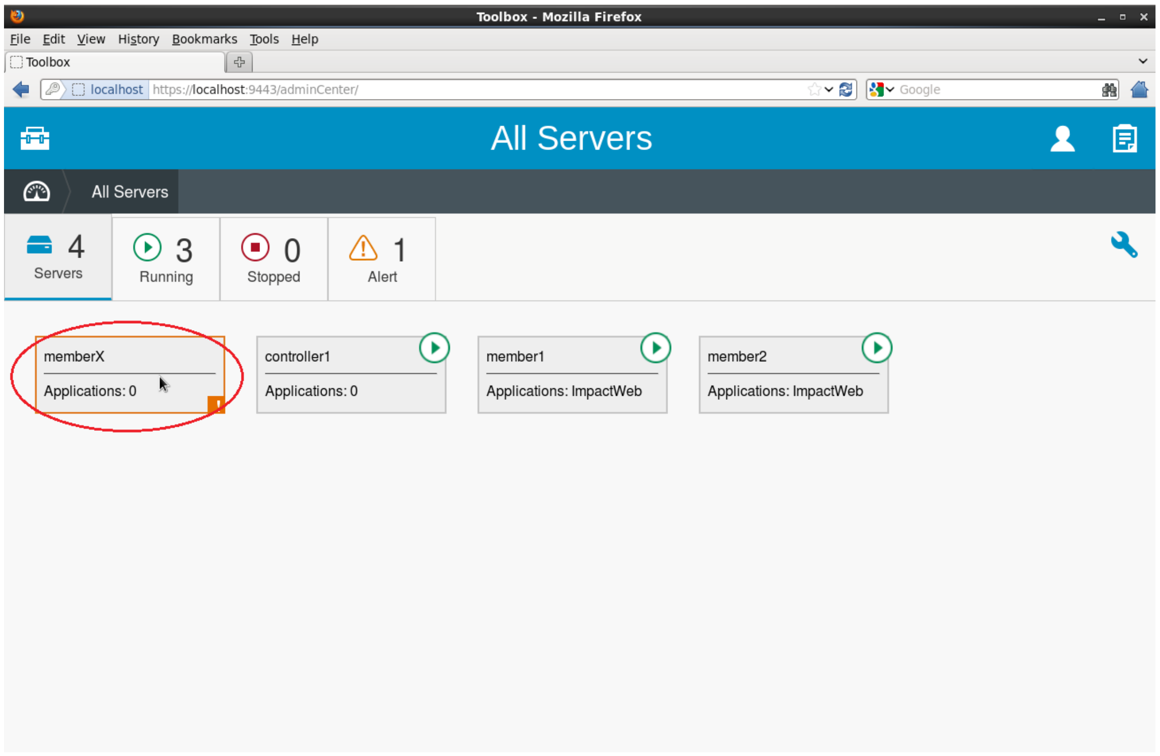
Launch the Explore tool.

3.7.15 The total number of servers has increased to 4 servers. Start the new server.

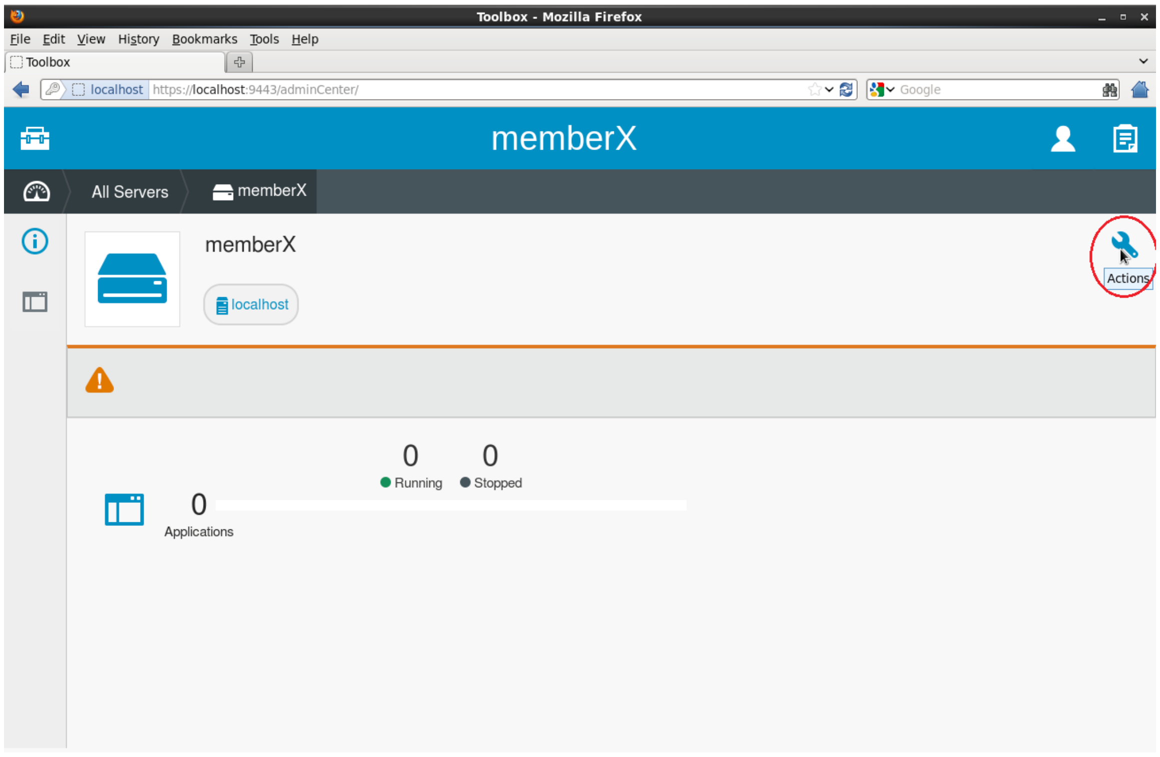
The newly deployed server will show up in 'Unknown' state because it has never been started. Click the Servers view.



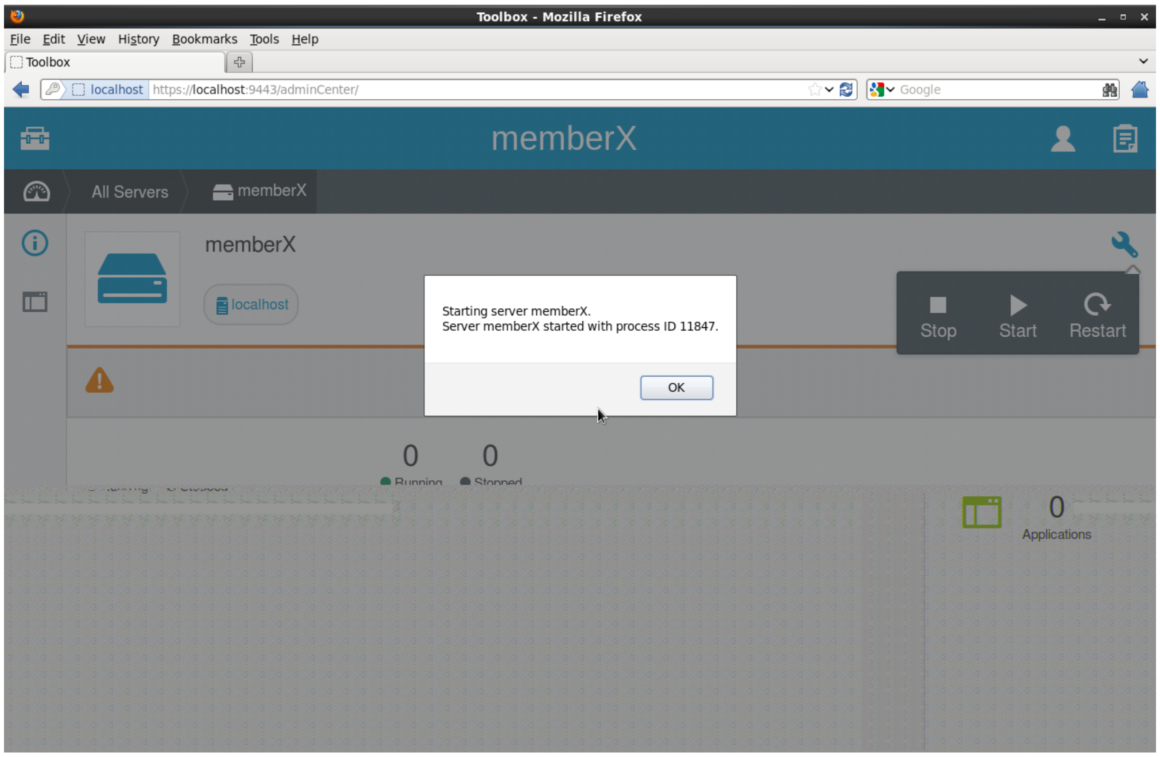
Click memberX.



Click the Actions button (wrench icon) and start the server.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
memberX
All Servers
memberX
memberX
localhost
Actions
0 Running
0 Stopped
0 Applications

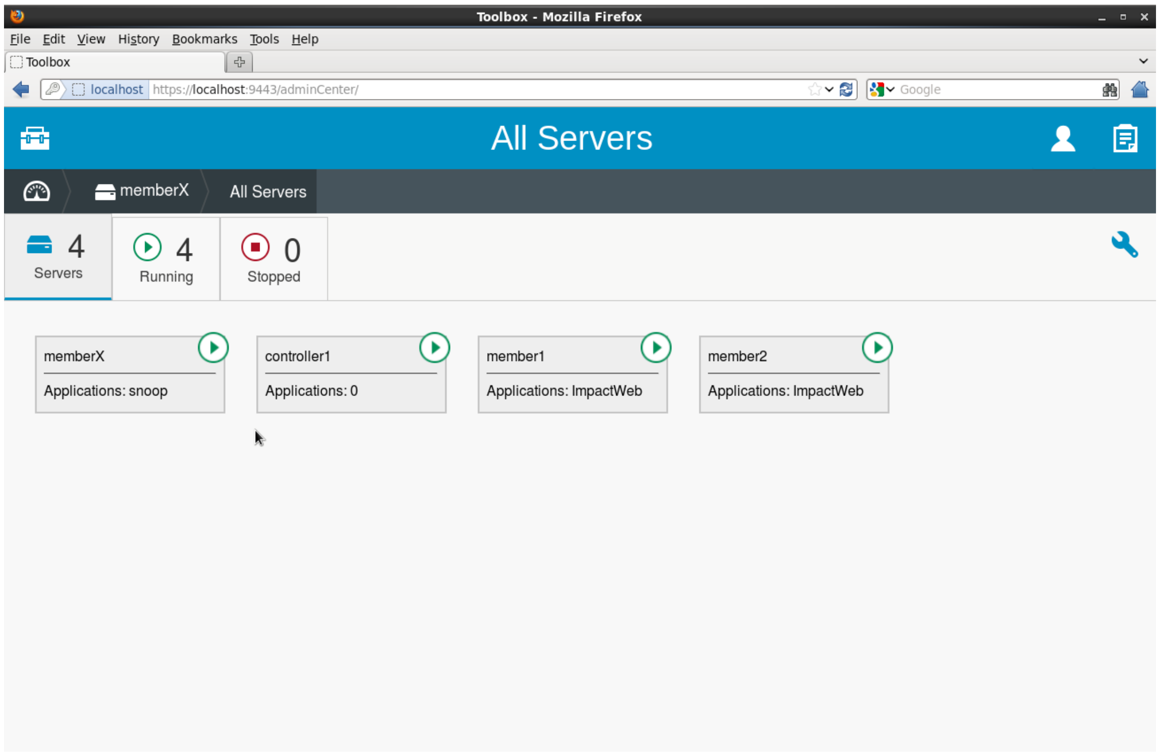
Result of the start operation:



Starting server memberX.
Server memberX started with process ID 11847.

3.7.16 The server memberX is now started.

Return to the 'All Servers' view by clicking on the breadcrumb.


Toolbox - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Toolbox
localhost https://localhost:9443/adminCenter/
Google
All Servers
memberX
All Servers
4 Servers
4 Running
0 Stopped

memberX
Applications: snoop

controller1
Applications: 0

member1
Applications: ImpactWeb

member2
Applications: ImpactWeb

Close the browser.

The deployed server is automatically joined to the collective when deployed via the Deploy tool and can be managed via the Admin Center as servers. Additional hosts can be registered to the collective via the [illegible] command. The Deploy tool highlights just one of the deployment options available through the MBeans provided by the collective controller. For more details on available collective controller MBeans, see the wasdev.net.

Thank you!
This lab is available from wasdev.net