**Creating a static cluster of Liberty servers**

Contents

Table of Contents

[Lab: Creating a static cluster of Liberty servers 3](#_Toc469389210)

[1.1 Prerequisites 4](#_Toc469389211)

[1.2 Install and set up IHS and WebSphere Plug-in 5](#_Toc469389212)

[1.3 Create Liberty servers to act as an application cluster 6](#_Toc469389213)

[1.4 Create a Liberty Collective 7](#_Toc469389214)

[1.5 Create a cluster containing the application servers 10](#_Toc469389215)

[1.6 Create a plug-in configuration file for the 'colorApp' cluster 11](#_Toc469389216)

[1.7 Testing the application cluster through the web server 12](#_Toc469389217)

[1.8 (Optional) View and control the cluster using the Admin Center 14](#_Toc469389218)

[1.9 Cleanup 16](#_Toc469389219)

[Notices 17](#_Toc469389220)

[Appendix A. Trademarks and copyrights 19](#_Toc469389221)

# Lab: Creating a static cluster of Liberty servers

In the Dynamic Routing lab, requests are shared across a cluster of Liberty servers that host the same application. The servers themselves are part of a Liberty collective, so they are 'managed' servers, but they are not organized into a *managed* *cluster*; the collective controller views them as individual servers. In this lab we will take the management of the servers one step further and organise them into a managed cluster, so that the collective controller can manage them as a group. One aspect of cluster management is the ability to generate a configuration file for the WebSphere web server Plug-in that includes all members of the cluster.

There are three ways to provide cluster configuration for the web server plug-in:

1. Create the plug-in configuration for each application server, use a utility to merge these configurations into a single file, then copy it to the web server installation. This method can be used for Liberty servers from any edition of WebSphere, and they don't have to be managed in a Liberty collective. This is the method used in the Merge Plugin lab.

2. Manage the application servers in a Liberty collective, and use the dynamicRouting feature in the collective controller to provide the routing information for each application server to the web server plug-in. For this method the web server plug-in configuration file (plugin-cfg.xml) only needs to contain routing information about the collective controller process; the plug-in then contacts the controller to obtain information about all the servers in the collective, and will direct HTTP requests to all of the applications in all the application servers. This method requires a WebSphere Network Deployment (ND) license for the collective controller host, but the application servers running the applications can be from any WebSphere edition (Liberty Core, WAS Base or ND). This is the method used in the Dynamic Routing lab.

3. Organize the application servers into managed clusters within a Liberty collective (using the clusterMember feature in each server), and use a utility to generate a merged plug-in configuration for all the servers in the cluster. This method requires a WebSphere Network Deployment (ND) license for all the Liberty servers. This is the method used in this lab.

The steps for this lab are as follows:

1. Install the IHS and WebSphere plug-in (you may have already done this for a previous lab).
2. Unzip the three provided server configurations, including two simple applications that display application server information.
3. Create a collective controller and join the application servers to the collective.
4. Create a cluster containing the three servers.
5. Generate the web server plugin configuration file for the cluster and place it in the plug-in configuration directory.
6. Send application requests through the web server and observe which servers the requests are directed to.
7. Optionally, run the Admin Center (Liberty's admin web UI) to view/control the cluster in the cluster from your browser.

Please refer to the following table for file and resource location references on different operating systems. This lab runs only on the Windows and Linux platforms as the WebSphere plugin is not provided for Mac OSX.

|  |  |  |
| --- | --- | --- |
| Location Ref. | OS | Absolute Path |
| *{LAB\_HOME}* | Windows | C:\wlp\_pot |
| Linux | ~/wlp\_pot |
| Mac OSX | not applicable |

## Prerequisites

The following preparation must be completed prior to beginning this lab:

1. Complete the Setup lab to install the JRE and Liberty runtime. You do not need to install the WDT developer tools (steps 0.5 and 0.6 in the Setup lab).
2. Note that this lab runs the IHS web server on port 9180. To check whether this port is available on your system, you can use the netstat command as follows:

Linux: **netstat -a | grep 9180**

Windows: **netstat -na | find "9180"**

If port 9180 is already in use on your host, select a different (available) port for this lab. Remember to use your chosen port number instead of port 9180 for the rest of this lab.

1. If you have completed the Merge Plugin or Dynamic Routing labs, remove the liberty servers you created for that lab by deleting the following directories:

**{LAB\_HOME}\wlp\usr\servers\blue**

**{LAB\_HOME}\wlp\usr\servers\green**

**{LAB\_HOME}\wlp\usr\servers\red**

**{LAB\_HOME}\wlp\usr\servers\controller**

## Install and set up IHS and WebSphere Plug-in

In the following steps we will add a separate IHS server. If you have already installed IHS and the plug-in for another lab you do not need to repeat this section.

This lab includes a copy of IHS and the WebSphere plug-in, which can be found under *{LAB\_HOME}*.

1. Unzip the {LAB\_HOME}\IHS-<version>-<platform>.zip for your system into the ${LAB\_HOME} directory. This will create a {LAB\_HOME}\IHS directory where IHS is now installed. This contains a copy of IBM HTTP Server, which is the Apache web server that is provided with WebSphere. The zip also has the WebSphere web server plug-in pre-installed in IHS.
2. Run the IHS postinstall script to perform some host-specific setup of IHS. Note this script creates the directory that we will use later for the plug-in log files: {LAB\_HOME}\IHS\plugin\logs\webserver1.

Run the following command:

Linux: $**{LAB\_HOME}/IHS/postinstall.sh**

Windows: **%LAB\_HOME%\IHS\postinstall.bat**

1. Windows only: install IHS as a service using this command:

**{LAB\_HOME}\IHS\bin\httpd.exe -k install -n "IBM HTTP Server V9.0"**

The output should be similar to this:

****

1. Change the default HTTP port number for IHS by editing the main configuration file, *{LAB\_HOME}\IHS\conf\httpd.conf* as follows

change the line

Listen 0.0.0.0:80

to use port 9180 (or an alternative port, if 9180 is not available on your system):

Listen 0.0.0.0:9180

1. Change the directory to *{LAB\_HOME}*\IHS\bin

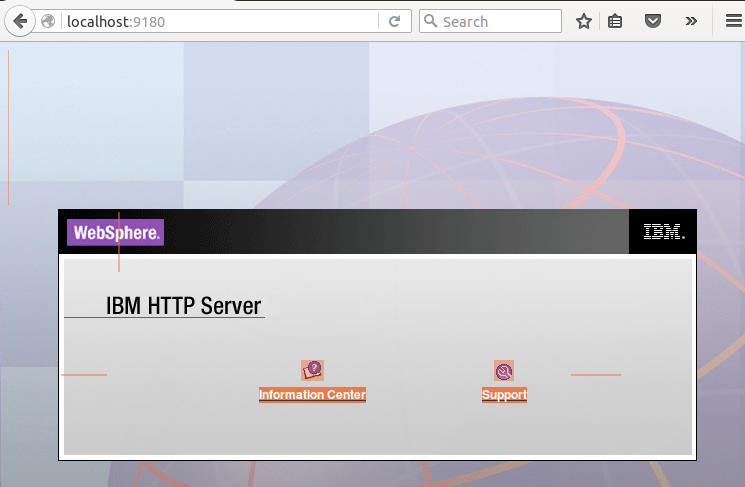
**cd *{LAB\_HOME}*\IHS\bin**

and start IHS with this command

Linux: **./apachectl -k start -f conf/httpd.conf**

Windows: **httpd -k start -f {LAB\_HOME}\IHS\conf\httpd.conf**

1. Check IHS is running: point your browser to <http://localhost:9180>. It should show the screen below which proves that IHS server is up and running. This screen comes from *{LAB\_HOME}*\IHS\htdocs\index.html. (This htdocs directory is the “document root” for the Apache Httpd server.)



1. Stop IHS with the following command:

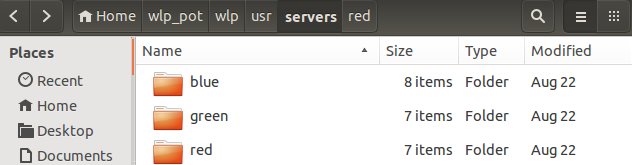
Linux: **./apachectl -k stop**

Windows: **httpd -k stop**

## Create Liberty servers to act as an application cluster

For this lab we will set up a static cluster of three servers. All three servers will contain the same two applications, and have the same configuration except for the HTTP port values (which need to be unique so they can all run on the same host).

1. Unzip the archives 'blue.zip', 'green.zip' and 'red.zip' into the **{LAB\_HOME}\wlp\usr\servers** directory, to create three servers called 'blue', 'green' and 'red'.



**Note:** These servers are configured to use unique port numbers as shown below. If these ports are not available on your system, edit the server.xml files to use available port numbers and use those numbers where needed for the remainder of this lab.

**blue: 9080, 8443**

**green: 9081, 8444**

**red: 9082, 8445**

1. Start the servers using these commands in different command windows (if you prefer to start the servers in the background from the same command window, use the 'server start' command instead of 'server run'):

**{LAB\_HOME}\wlp\bin\server run blue**

**{LAB\_HOME}\wlp\bin\server run green**

**{LAB\_HOME}\wlp\bin\server run red**

1. Test that the application is running by entering these in your browser: <http://localhost:9080/WhereAmI>

[http://localhost:9081/WhereAmI](http://localhost:9080/WhereAmI)

[http://localhost:9082/WhereAmI](http://localhost:9080/WhereAmI)

You will see that the servers are running the same aplication, which uses the server name to control the color of the text that is shown in the browser (to make it easy to tell, at a glance, which server the application is running in).

1. Stop the servers using these commands:

**{LAB\_HOME}\wlp\bin\server stop blue**

**{LAB\_HOME}\wlp\bin\server stop green**

**{LAB\_HOME}\wlp\bin\server stop red**

## Create a Liberty Collective

Cluster management requires that the application servers are members of a collective. In this section we will create a collective controller, which is a Liberty server process that manages a collective. Then we will join the application servers into the collective as members.

1. Run this command to create a new liberty server called 'controller':

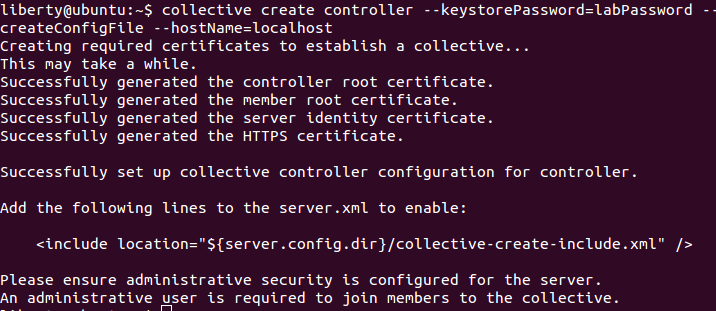
**{LAB\_HOME}\wlp\bin\server create controller**



1. Run this command to configure the new server as a Liberty collective controller:

**{LAB\_HOME}\wlp\bin\collective create controller --keystorePassword=labPassword --createConfigFile --hostName=localhost**

You should see output similar to this:



1. As indicated in the output above, edit the server.xml file for the controller server **{LAB\_HOME}\wlp\usr\servers\controller\server.xml**

a) add this line to include the collective controller configuration that was generated by the **collective create** command above:

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

b) change the HTTP port numbers so they don't clash with the ports being used by the member servers, for example:

<httpEndpoint id="defaultHttpEndpoint"

httpPort="**9083**"

httpsPort="**9446**" />

1. Edit the file **{LAB\_HOME}/wlp/usr/servers/controller/collective-create-include.xml** and add the **userName** and **userPassword** values for the quickStartSecurity element:

<quickStartSecurity userName="**adminid**" userPassword="**adminpw**" />

1. Start the controller using this command

**{LAB\_HOME}\wlp\bin\server start controller**

There should not be any errors output, and this message should appear (along with several others) in the {LAB\_HOME}\wlp\usr\servers\controller\logs\messages.log file:

CWWKX9003I: CollectiveRegistration MBean is available.

You may find it convenient to 'tail' the controller's messages.log file

1. Execute these commands to join the application servers to the collective:

**cd {LAB\_HOME}\wlp\usr\servers\blue**

**{LAB\_HOME}\wlp\bin\collective join blue --host=localhost --port=9446 --user=adminid --password=adminpw --keystorePassword=labPassword --createConfigFile --hostname=localhost**

Enter **y** in response to this question:

Do you want to accept the above certificate chain? (y/n)

**cd {LAB\_HOME}\wlp\usr\servers\ green**

**{LAB\_HOME}\wlp\bin\collective join green --host=localhost --port=9446 --user=adminid --password=adminpw --keystorePassword=labPassword --createConfigFile --hostname=localhost**

Enter **y** in response to this question:

Do you want to accept the above certificate chain? (y/n)

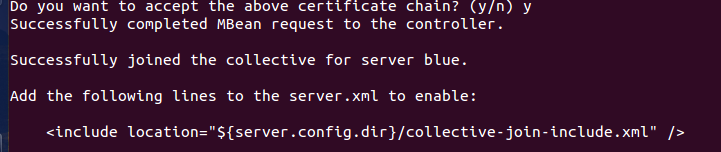
**cd {LAB\_HOME}\wlp\usr\servers\ red**

**{LAB\_HOME}\wlp\bin\collective join red --host=localhost --port=9446 --user=adminid --password=adminpw --keystorePassword=labPassword --createConfigFile --hostname=localhost**

Enter **y** in response to this question:

Do you want to accept the above certificate chain? (y/n)

In each case, the output will be quite long and will end something like this::



1. As indicated in the output above, edit the server.xml file for the application servers: **{LAB\_HOME}\wlp\usr\servers\blue\server.xml** and **{LAB\_HOME}\wlp\usr\servers\green\server.xml** and **{LAB\_HOME}\wlp\usr\servers\green\server.xml** toadd this line to both files, to include the collective member configuration that was generated by the **collective join** command above:

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

1. Start the servers with these commands (use 'server run' if you prefer to run them in the foreground):

**{LAB\_HOME}\wlp\bin\server start blue**

**{LAB\_HOME}\wlp\bin\server start green**

**{LAB\_HOME}\wlp\bin\server start red**

In the messages.log file of the controller you should see messages like this for red, blue and green servers:

CWWKX9076I: The collective member green on host localhost with the user directory %2Fhome%2Fliberty%2Fwlp\_pot%2Fwlp%2Fusr connected to the collective controller.

## Create a cluster containing the application servers

Creating a Liberty cluster is very easy. Simply add the clusterMember-1.0 feature to each application server, and specify the cluster name you want to use in each application server.

If you want to divide servers into different clusters, do that by specifying different cluster names in the server configurations. If you add the clusterMember-1.0 feature and don't specify a cluster name, the server will be joined to the 'defaultCluster'. A server can only belong to one cluster, but it can be moved to another simply by changing the cluster name in its configuration.

In this lab, we will include the configuration for the application servers in a separate configuration file, as it is the same configuration for all the servers.

1. Copy the file named **cluster-config.xml** into the **{LAB\_HOME}\wlp\usr\shared\config** directory.
2. Edit the **cluster-config.xml** file to replace **{LAB\_HOME}** with the real value for your system in this line:

<pluginConfiguration pluginInstallRoot="{LAB\_HOME}/IHS/plugin"/>

1. Edit the server.xml for the each application server to include that file by adding this line:

<include location="${shared.config.dir}/cluster-config.xml" />

to these files: **{LAB\_HOME}\wlp\usr\servers\blue\server.xml** and **{LAB\_HOME}\wlp\usr\servers\green\server.xml** and **{LAB\_HOME}\wlp\usr\servers\red\server.xml**

Note that the cluster creation happens without needing to restart any of the servers. The messages.log file for the collective controller should show a message indicating the creation of the cluster:

CWWKX9053I: The colorApp cluster has been created.

and the joining of each application server to the cluster:

CWWKX9051I: The blue server has been added to the colorApp cluster.

etc.

## Create a plug-in configuration file for the 'colorApp' cluster

The pluginUtility command was enhanced in Liberty 16.0.0.4 to generate plugin configuration files for named clusters. This is useful if you want to have different web servers handling traffic for different clusters within the same collective, for example.

For the pluginUtility to connect to the collective controller, it needs one of the JMX connector features to be active. Since this lab is running everything on the same host, we will use the localConnector-1.0 feature in the controller, so add that to the controller's server.xml:

<featureManager>

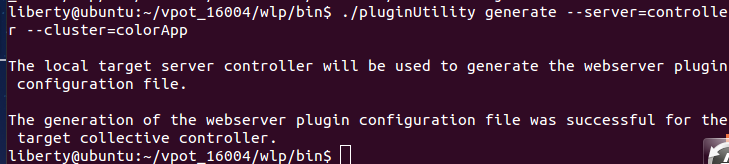
......

<feature>localConnector-1.0</feature>

</featureManager>

Run this command:

**{LAB\_HOME}\wlp\bin\pluginUtility generate --server=controller --cluster=colorApp**

****

In the messages.log file for the controller, there will be a message showing that the configuration file was written to the **logs/state** directory of the collective controller::

Merged plugin config file written to {LAB\_HOME}/wlp/usr/servers/controller/logs/state/colorApp-plugin-cfg.xml

1. Examine the generated **colorApp-plugin-cfg.xml from your current directory**.
2. Copy **.\colorApp-plugin-cfg.xml** to **{LAB\_HOME}\IHS\plugin\config\webserver1**

and rename the file to plugin-cfg.xml.

Note: If you prefer to keep the original name of the generated file, you will need to update the {LAB\_HOME}\IHS\conf\httpd.conf file; look for this section at the bottom and modify the file name:

<IfFile plugin/config/webserver1/plugin-cfg.xml>

LoadModule was\_ap22\_module plugin/bin/mod\_was\_ap22\_http.so

WebSpherePluginConfig /home/liberty/WLP\_16.0.0.4/IHS/plugin/config/webserver1/plugin-cfg.xml

</IfFile>

## Testing the application cluster through the web server

1. Start the IHS server:

Linux: **{LAB\_HOME}/IHS/bin/apachectl -k start -f conf/httpd.conf**

Windows: **{LAB\_HOME}\IHS\bin\httpd -k start -f {LAB\_HOME}\IHS\conf\httpd.conf**

A file called http\_plugin.log should appear in the **{LAB\_HOME}\IHS\plugin\logs\webserver1** directory, indicating that the plug-in configuration file was found and the plugin has started.

NOTE: In the log {WLP\_HOME}/IHS/plugin/logs/webserver1/http\_plugin.log there are plenty of errors as the files plugin-key.sth and plugin-key.kdb are not available. These do not prevent the lab to work but it would be good to warn that these might occur. Something like:

"You might see security errors but you can ignore them"

[08/Dec/2016:08:24:48.05069] 0000d3c3 0b570700 - ERROR: ws\_os: check\_file: Error accessing /home/admin/WLP\_16.0.0.4/IHS/plugin/config/webserver1/plugin-key.sth. A component of path does not name an existing file or path is an empty string.  Check the filename and correct the plugin-cfg.xml if necessary.

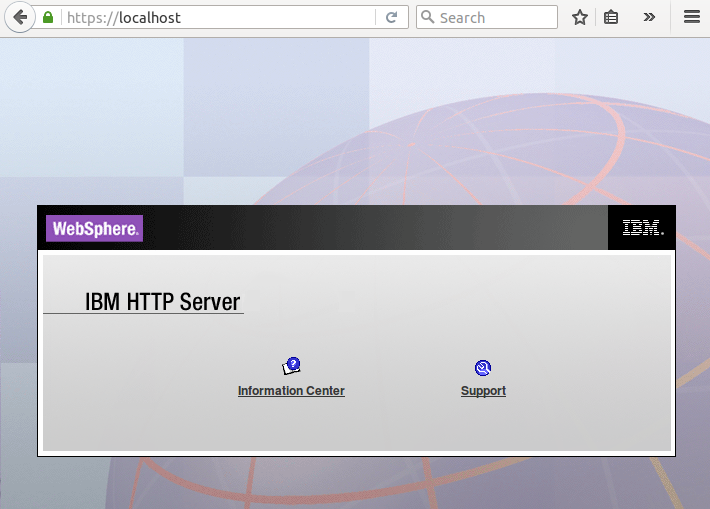
[08/Dec/2016:08:24:48.05079] 0000d3c3 0b570700 - ERROR: ws\_os: check\_file: Error accessing /home/admin/WLP\_16.0.0.4/IHS/plugin/config/webserver1/plugin-key.kdb. A component of path does not name an existing file or path is an empty string.  Check the filename and correct the plugin-cfg.xml if necessary.

[08/Dec/2016:08:24:48.05226] 0000d3c3 0b570700 - ERROR: lib\_security: logSSLError: str\_security (gsk error 408):  GSK\_ERROR\_BAD\_KEYFILE\_PASSWORD

Enter this in your browser:

<http://localhost:9180>

You should see the IHS welcome screen:



1. Access the first application through the web server and plugin using this URL:

[http://localhost:9180/WhereAmI](http://localhost/WhereAmI)

2. Refresh the browser several times and see the plug-in performing the round robin load balancing between the servers in the cluster.

3. Stop the blue server and see that the plug-in responds immediately by sending all requests to the green and red servers:

4. Restart the blue and see that the plug-in quickly starts routing requests to the restarted server

5. Now access the other application, which uses an HTTP session:

<http://localhost:9180/WhereAmIWithSession>

6. Refresh the browser a few times, and see that all the requests go back to the same server. This is because the plug-in is applying *session affinity* for the application to that server. If you stop that server, and refresh the browser again, you will see that the request *fails over* to a different server, and affinity is established with the new server.

This concludes the Static cluster lab. You have seen how to join application servers into a collective and configure a static cluster, and to generate a web server plug-in configuration file for the cluster. This can be useful in topologies where each cluster need to be fronted by its own web server.

## (Optional) View and control the cluster using the Admin Center

1. Add the Admin Center feature to the collective controller server, by adding the following to the controller's server.xml file (or to the included collective-create-include.xml file):

<featureManager>

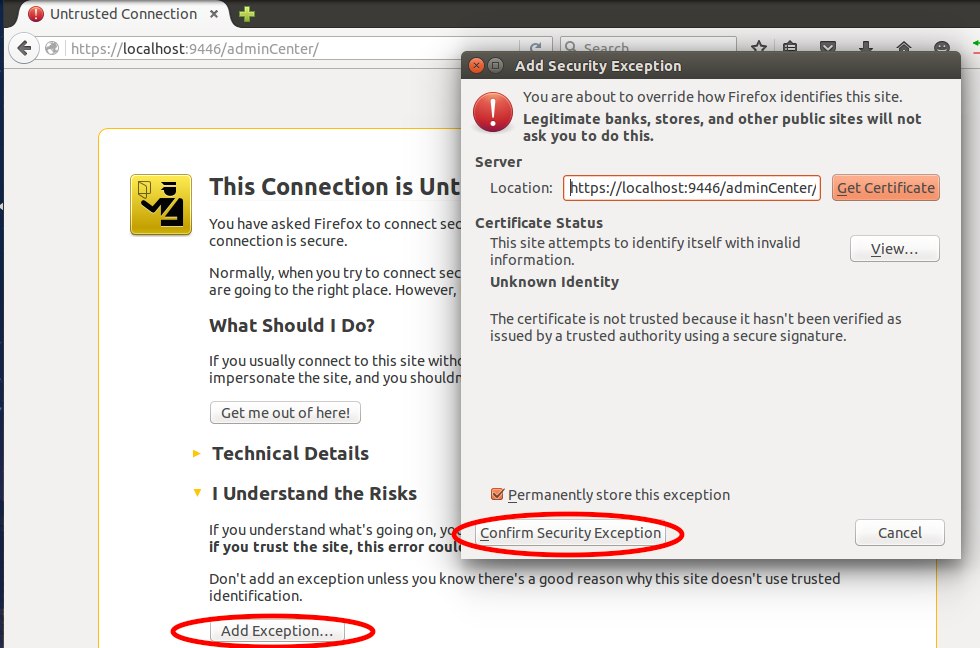
<feature>adminCenter-1.0</feature>

</featureManager>

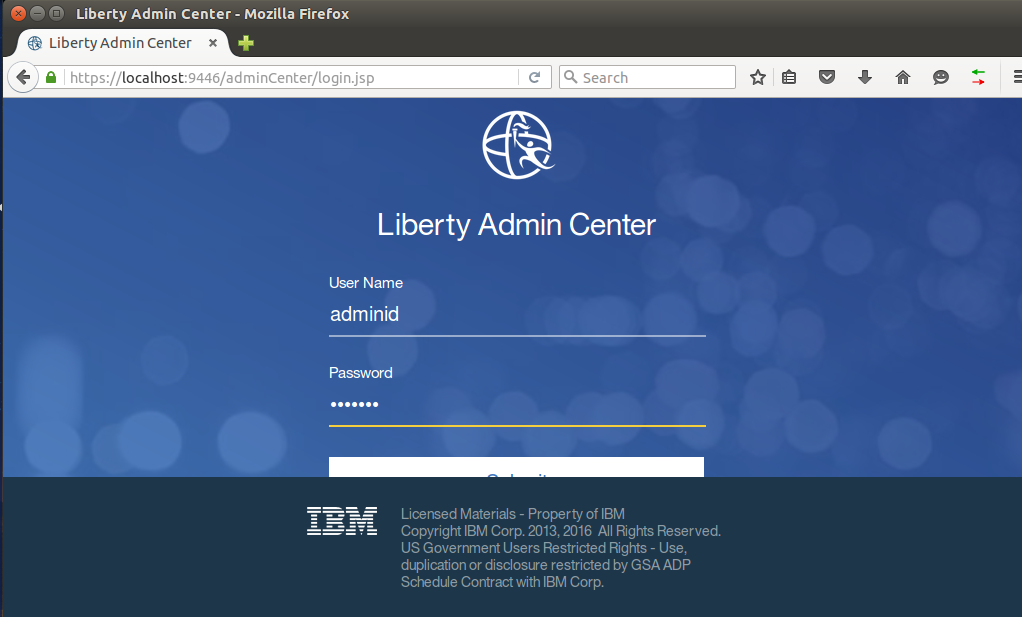
1. Access the Admin Center using this URL:

<http://localhost:9083/adminCenter/>

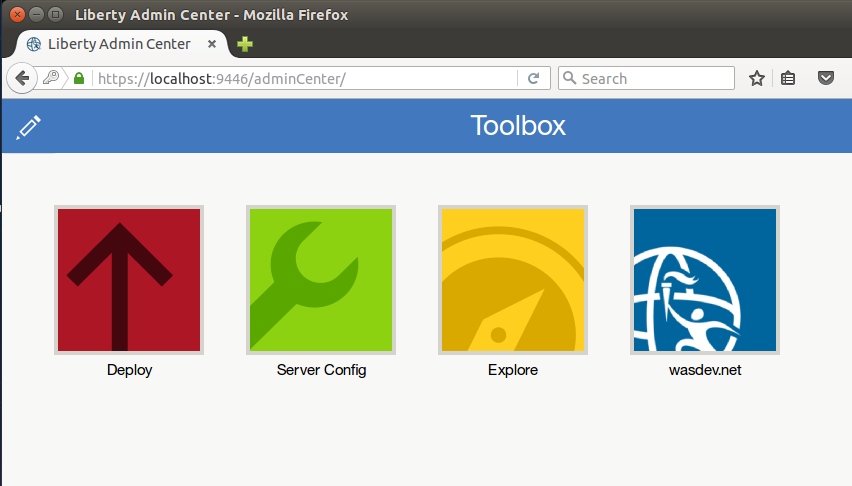
1. If asked, accept the security certificate

****

1. Log into the Admin Center with the username and password specified in the 'quickStartSecurity' element in the included collective-create-include.xml file (adminid/adminpw):



1. Once in the Admin Center, select the Explore tool, and use it to view, stop and start your cluster.



## Cleanup

1. Stop the liberty servers **blue, green, red** and **controller** using the following command for each server name: **{LAB\_HOME}/wlp/bin/server stop <server\_name>**

2. Stop the web server using the following command:

Linux: **./apachectl -k stop**

Windows: **httpd -k stop**

3. Delete any parts of the lab that you don't want to keep.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental. All references to fictitious companies or individuals are used for illustration purposes only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

1. Trademarks and copyrights

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IBM | AIX | CICS | ClearCase | ClearQuest | Cloudscape |  |
| Cube Views | DB2 | developerWorks | DRDA | IMS | IMS/ESA |  |
| Informix | Lotus | Lotus Workflow | MQSeries | OmniFind |  |  |
| Rational | Redbooks | Red Brick | RequisitePro | System i |  |  |
| *System z* | *Tivoli* | *WebSphere* | *Workplace* | *System p* |  |  |

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.



© Copyright IBM Corporation 2017.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

