

## Lab 1: Setup Liberty



Contents

**LAB 1**

<b>SETUP THE LIBERTY ENVIRONMENT .....</b>	<b>4</b>
1.1 THE LAB ENVIRONMENT .....	5
<b>1.1.1 LOGIN TO THE "LIBERTY vPOT ... DESKTOP" VM AND GET STARTED.....</b>	<b>6</b>
1.2 EXTRACT THE EXTENDED LIBERTY vPOT IMAGE .....	9
1.3 INSTALL WEBSphere LIBERTY .....	10
1.4 INFORMATION ONLY – OPEN LIBERTY .....	14
1.4.1 INFORMATION ONLY – OPEN LIBERTY DOWNLOADS, DOCKER, AND DEVOPS.....	15
1.5 CREATE TEST SERVER.....	16
1.5.1 SHORT RECAP OF WHAT YOU JUST DID .....	19
1.6 TEST THE WEBSphere DEVELOPER TOOLS (WDT).....	20
1.7 CREATE LIBERTY SERVER IN WDT .....	22



## Lab 1 Setup the Liberty environment

In the initial setup of Liberty, you will install Liberty using the **Archive** method. Then, you will create your first Liberty server, start it, and deploy a simple Java EE application to the server to test the Liberty server runtime.

You will use an Ubuntu Linux VM that has been prepared for the Liberty labs and includes the following:

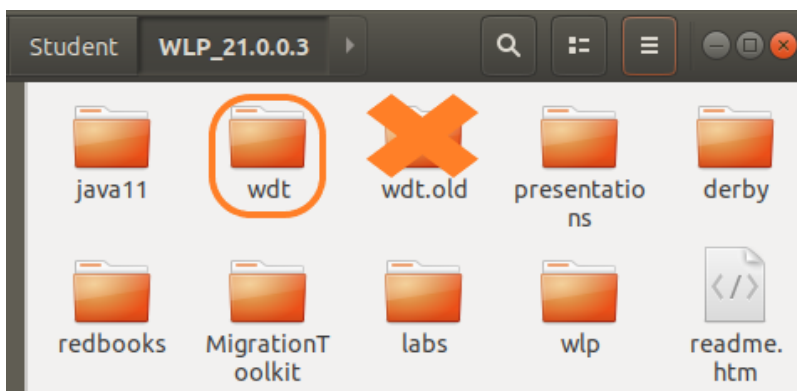
- The **Liberty** package has been downloaded to **/home/ibmdemo/Downloads** directory
- The **IBM JDK** has been downloaded to **/home/ibmdemo/Downloads** directory
- The required Operating System libraries have been installed.

### Information:



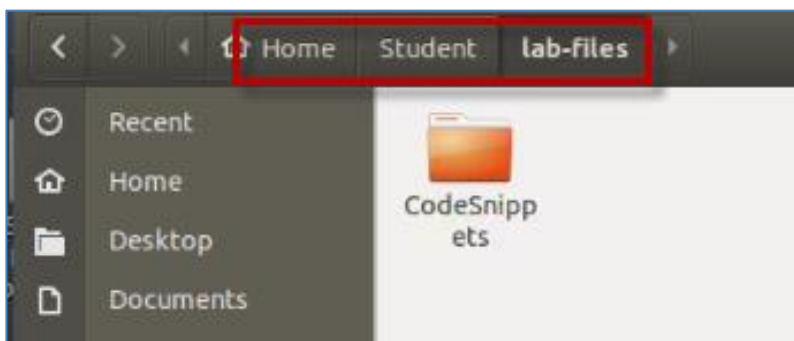
The Liberty image contains WebSphere Developer Tools for Eclipse.

The VMWare image contains the “wdt” folder that contains the WebSphere Developer Tools that is used for the lab.



**TIP:** To reduce typing or copy & past of commands, you can find the related code snippets or commands in the VMWare image in the directory:

**/home/ibmdemo/Student/lab-files/CodeSnippets/Bootcamp\_Lab1\_setup\_CodeSnippets.txt**



## 1.1 The lab environment

One (1) Linux VM has been provided for this lab.



The “**Liberty vPOT ... Desktop**” VM has the following software available:

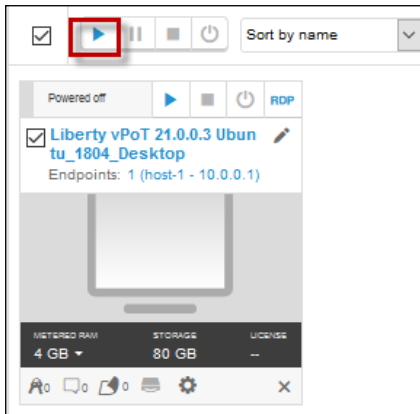
- Application Project with Liberty
- Maven 3.6.0
- The login credentials for the **Liberty vPOT ... Desktop** VM are:

User ID: **ibmdemo**

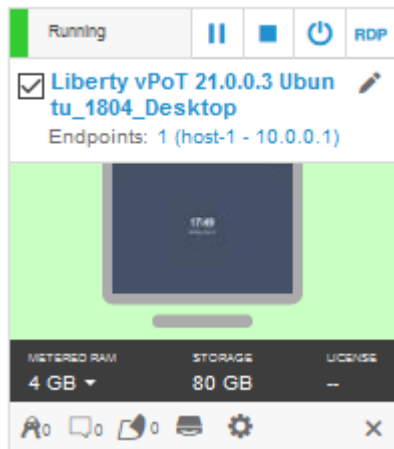
Password: **passw0rd** (That is a numeric zero in passw0rd)

## 1.1.1 Login to the "Liberty vPOT ... Desktop" VM and Get Started

\_\_1. If the VM is **not** already started, start it by clicking the **Play** button.



\_\_2. After the VM is started, click the “**Liberty vPOT ... Desktop**” VM icon to access it.



\_\_3. **Login with ibmdemo ID.**

\_\_a. Click on the “ibmdemo” icon on the Ubuntu screen.



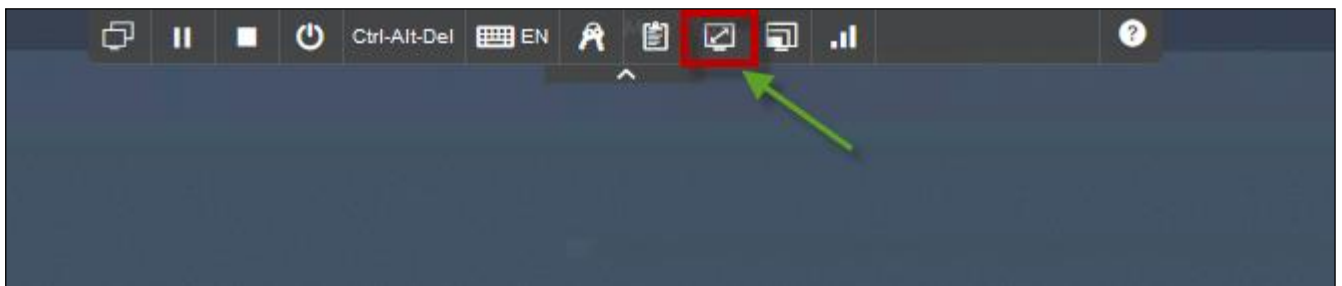
\_\_b. When prompted for the password for “ibmdemo” user, enter “passw0rd” as the password:

Password: **passw0rd** (lowercase with a zero instead of the o)



\_\_4. Resize the Skytap environment window for a larger viewing area while doing the lab.

From the Skytap menu bar, click on the “Fit to Size”  icon. This will enlarge the viewing area to fit the size of your browser window.

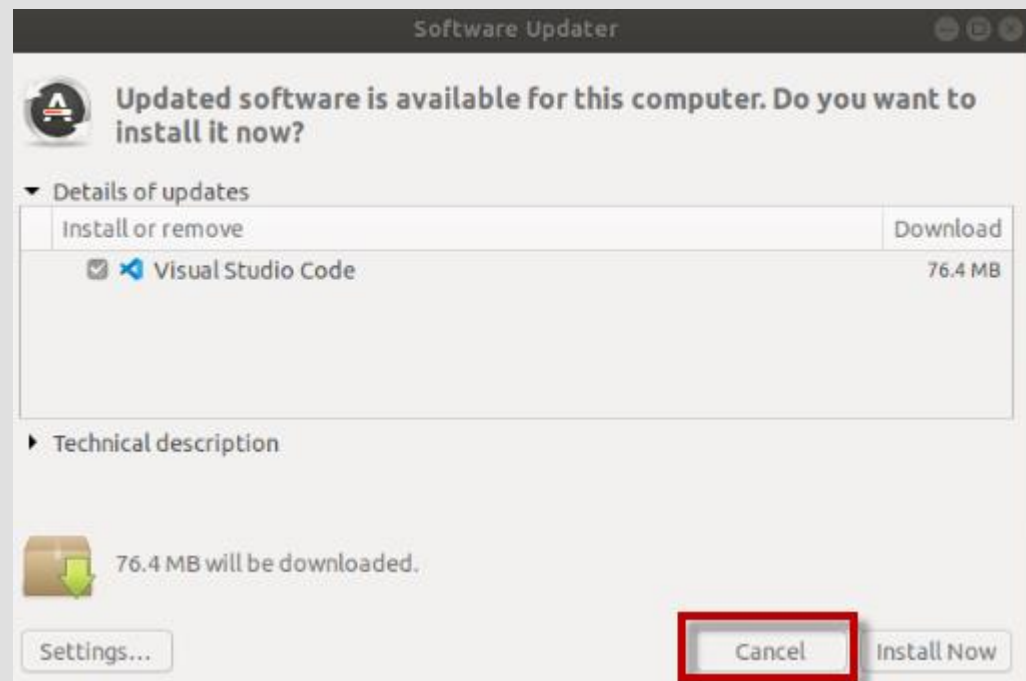


**Important:**

**Click CANCEL....** If, at any time during the lab, you get a pop-up asking to install updated software onto the Ubuntu VM.

The one we experience is an update available for VS Code.

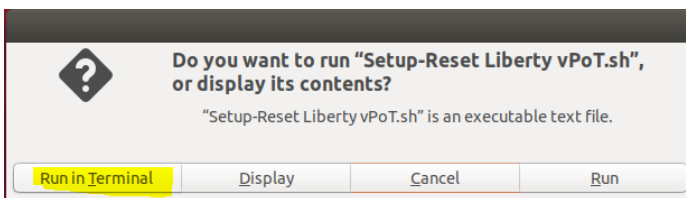
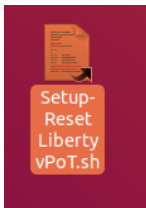
**CLICK CANCEL!**





## 1.2 Extract the extended Liberty vPoT image

- \_\_1. Locate the script on the desktop and double-click on it to **run in terminal**



- \_\_2. Wait until the image has been extracted. Once completed, press **Enter** to continue:

```
Terminal
File Edit View Search Terminal Help
*****
* The Ubuntu image has been prepared for the Liberty vPoT hands-on labs.
* Also the Liberty vPoT package has been downloaded to /home/ibmdemo/Downloads.
* For the Liberty vPoT labs, {LAB_HOME} will refer to: /home/ibmdemo/Student/WLP_21.0.0.3
*****
Press Enter to continue
```

- \_\_3. The Liberty vPOT contents are extracted to **/home/ibmdemo/Student/WLP\_21.0.0.3**, and is referred as **{LAB\_HOME}** throughout the labs.

### Information:



If you need to reset the image at some point of time, make sure that all programs are stopped and run the script again.

This will clean-up the directory **/home/ibmdemo/Student/WLP\_\*** and extract the image again.

## 1.3 Install WebSphere Liberty

For your convenience, Liberty has already been installed into the directory **/home/ibmdemo/Student/WLP\_21.0.0.3/wlp**.

However, in this section, you will install a new instance of Liberty to learn how easy and fast it is to install Liberty, using the **archive installation** method.

Liberty can be installed via IBM Installation Manager, but the more convenient way is to use an Archive Install.

The related packages can be downloaded from Passport Advantage, but the most recent versions are only available on the IBM Support pages. There are separate packages available for the Liberty edition and for the IBM Java SDK. Liberty for Developers downloads and documentation is located here:

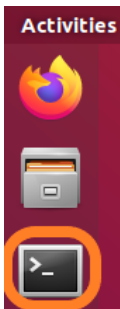
<https://www.ibm.com/support/pages/node/6250961#asset/>

In this section we will use these two packages, which are **Java JDK v8** and **Liberty ND** edition.

- IBM Java SDK for WebSphere: ibm-java-sdk-8.0-6.26-linux-x86\_64.tgz
- IBM WebSphere Liberty ND Archive Install: wlp-nd-all-21.0.0.3.jar

To install Liberty using the Archive method, perform the following steps:

\_\_1. Open a Terminal window by clicking on the appropriate icon:



\_\_2. Navigate to the “Student” directory:

```
cd /home/ibmdemo/Student
```

\_\_3. Create a temporary directory and navigate to that directory

```
mkdir temp  
cd temp
```

- \_\_4. Extract the IBM Java SDK to the ~/Student/temp/directory. This will extract the JDK to the directory **ibm-java-x86\_64-80**

```
tar -zxvf '/home/ibmdemo/Downloads/ibm-java-sdk-8.0-6.26-linux-x86_64.tgz'
```

- \_\_5. Extract the WebSphere Liberty ND package to the ~/Student/temp/directory.

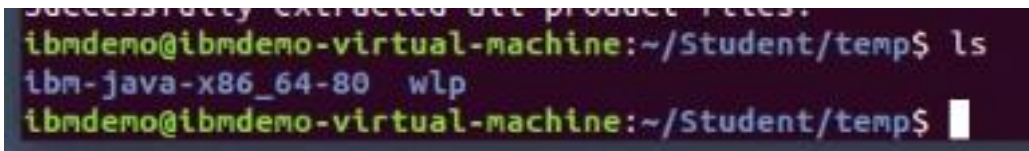
```
ibm-java-x86_64-80/bin/java -jar '/home/ibmdemo/Downloads/wlp-nd-all-21.0.0.3.jar' --acceptLicense .
```

The Liberty Archive is a Java “jar” file. To extract the archive, use the java -jar command.

- The **--acceptLicense** option is included to automatically accept the license, without having additional prompts.
- The **dot** at the end of the command means to extract the archive to the current directory.
- The Liberty archive is extracted, and Liberty is now installed in the **wlp** subdirectory.

- \_\_6. List the two directories that were created via the extraction of the Java JDK and Liberty ND

```
ls
```

A terminal window screenshot showing the command 'ls' being executed in the directory ~/Student/temp. The output shows two subdirectories: 'ibm-java-x86\_64-80' and 'wlp'. The prompt is 'ibmdemo@ibmdemo-virtual-machine:~/Student/temp\$'.

- Liberty is now installed the **wlp** directory.
- IBK JDK is now installed in the **ibm-java-x86\_64-80** directory

- \_\_7. Set the **JAVA\_HOME** path to tell Liberty to use the Java SDK that you just extracted

```
export JAVA_HOME=~/Student/temp/ibm-java-x86_64-80/jre/
```

- \_\_8. Display the product information for Liberty
- \_\_a. Display the Liberty product edition and version

```
wlp/bin/productInfo version
```

The output should indicate the Liberty Edition is **ND**, and the product version should match the Liberty Archive that is used in the lab.

Liberty fixpacks are delivered on a four-week continuous delivery interval.

- The Version indicates the YEAR and MONTH of the Liberty fixpack installed.
- 21.0.0.3 is the 3<sup>rd</sup> fixpack in 2021

```
ibmdemo@ibmdemo-virtual-machine:~/Student/temp$ wlp/bin/productInfo version
Product name: WebSphere Application Server
Product version: 21.0.0.3
Product edition: ND
```

- \_\_b. Display the list of installed features

```
wlp/bin/productInfo featureInfo
```

The output should look similar to the illustration below, which includes all of the Liberty ND features. The screen shot is only a partial list of installed features of Liberty ND.

```
ibmdemo@ibmdemo-virtual-machine:~/Student/temp$ wlp/bin/productInfo featureInfo
acmeCA-2.0 [2.0.0]
adminCenter-1.0 [1.0.0]
apiDiscovery-1.0 [1.0.0]
appClientSupport-1.0 [1.0.0]
appSecurity-1.0 [1.1.0]
appSecurity-2.0 [1.0.0]
appSecurity-3.0 [1.0.0]
appSecurityClient-1.0 [1.0.0]
audit-1.0 [1.0.0]
batch-1.0 [1.0.0]
batchManagement-1.0 [1.0.0]
beanValidation-1.0 [1.0.0]
beanValidation-1.1 [1.0.0]
beanValidation-2.0 [1.0.0]
bells-1.0 [1.0.0]
bluemixUtility-1.0 [1.0.0]
blueprint-1.0 [1.0.0]
cdi-1.0 [1.0.0]
```

- \_\_c. **You have just installed Liberty in the temp directory using the archive install method.**

It's that simple to install Liberty. Creating and working with a new Liberty server is equally simple. You will do that in the next sections of the lab.

There are more than 200 features installed via the Liberty ND package. By default, you would start with a much more light-weight package. If you are interested to see which features belong to which Liberty Edition, please look here:

<https://www.ibm.com/docs/en/was-liberty/nd?topic=management-liberty-features>



**Information:**

As mentioned before, Liberty has already been installed into the directory **/home/ibmdemo/Student/WLP\_21.0.0.3\*/wlp**.

We will **NOT** use the Liberty installed in the **temp directory** in the next sections of the lab. You can ignore the temp directory.

- \_\_9. **Close** the Terminal window.

## 1.4 INFORMATION ONLY – Open Liberty

How does Open Liberty fit into the Liberty family of editions?

Open Liberty is the lightweight open source server runtime that can be ideal for building Java microservices and cloud-native apps.



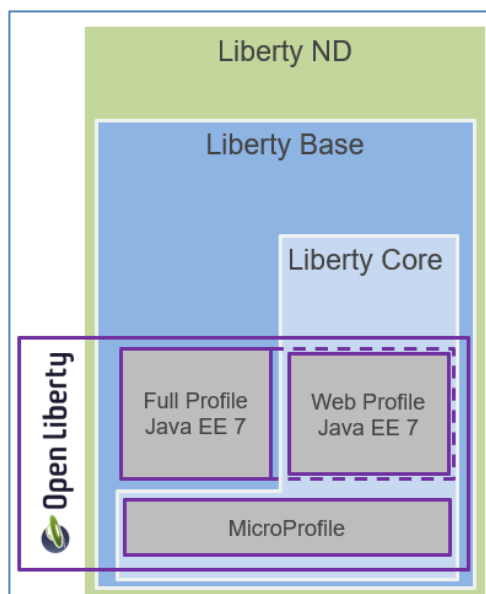
Open Liberty provides a proven open source foundation for the WebSphere Liberty portfolio. All of the editions of Liberty are built on top of the same code base from Open Liberty.

- Liberty Core offers supported Web Profile
- Liberty Base offers supported Full Profile
- Liberty ND offers supported Full Profile and clustering / workload management capabilities

All 3 commercial editions provide additional capabilities:

- ✓ Programming model extensions
- ✓ Production quality of service extensions
- ✓ Security extensions

**And IBM's full support**



## 1.4.1 INFORMATION ONLY – Open Liberty Downloads, Docker, and Devops

How do you get Open Liberty?

Open Liberty can be obtained via zip file downloads, or through Maven, Gradle, and Docker.

### Get started with a simple download

- Visit <http://openliberty.io> to find the latest Open Liberty builds
- Incorporate Open Liberty into your DevOps process



Visit <https://openliberty.io/downloads/> to find the latest Open Liberty releases and builds.

Open Liberty Zip File Download packages are available here:

[https://openliberty.io/downloads/#runtime\\_releases](https://openliberty.io/downloads/#runtime_releases)

Below are examples of getting Open Liberty using Maven, Gradle, and Docker.

Maven

Gradle

Docker

```
<dependency>
  <groupId>io.openliberty</groupId>
  <artifactId>openliberty-runtime</artifactId>
  <version>[21.0.0.5,)</version>
  <type>zip</type>
</dependency>
```

Maven

Gradle

Docker

```
dependencies {
    libertyRuntime group: 'io.openliberty', name: 'openliberty-runtime', version: '[21.0.0.5,)'
}
```

Maven

Gradle

Docker

```
docker pull open-liberty
```

## 1.5 Create test server

- \_\_1. Open a Terminal window by clicking on the appropriate icon:



- \_\_2. Navigate to the Liberty runtime installation directory {LAB\_HOME}/wlp/bin

```
cd /home/ibmdemo/Student/WLP_21.0.0.3/wlp
```

- \_\_3. Run the following command to **create** a new server named “myServer”

```
bin/server create myServer
```

```
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$ bin/server create myServer
Server myServer created.
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$
```

The server command supports actions for starting, stopping, creating, packaging, and dumping a Liberty server.

The server create command creates a new Liberty server with the name specified.

Additional detail on the server command can be found here: <https://www.ibm.com/docs/en/was-liberty/base?topic=line-server-command-options>

- \_\_4. The new server is created in the following directory:

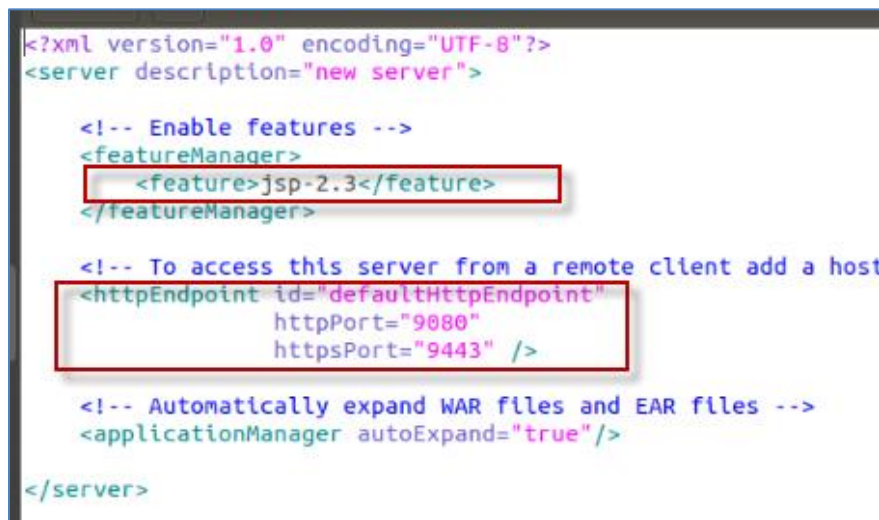
```
/home/ibmdemo/Student/WLP_21.0.0.3/wlp/usr/servers/myServer
```

```
ibmdemo@ibmdemo-virtual-machine: ~/Student/WLP_21.0.0.3/wlp/usr/servers/myServer
File Edit View Search Terminal Help
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp/usr/servers/myServer$ ls
apps  dropins  server.env  server.xml  workarea
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp/usr/servers/myServer$
```



- \_\_5. The **server.xml** file is the complete default server configuration. Open an editor to view the server configuration file. Ensure you are in the correct directory before opening the server.xml file.

```
cd /home/ibmdemo/Student/WLP_21.0.0.3/wlp  
gedit usr/servers/myServer/server.xml
```



The server.xml defines a minimal configuration needed to start a Liberty server.

In this example, it includes only the **JSP feature**, and defines the **HTTP** and **HTTPS** endpoints that the server is listening for incoming HTTP(S) requests.

- \_\_6. **Close** the gedit editor.
- \_\_7. Start the server instance using the **server start** command:

```
bin/server start myServer
```

```
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$ bin/server start myServer  
Starting server myServer.  
Server myServer started with process ID 3878.  
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$
```

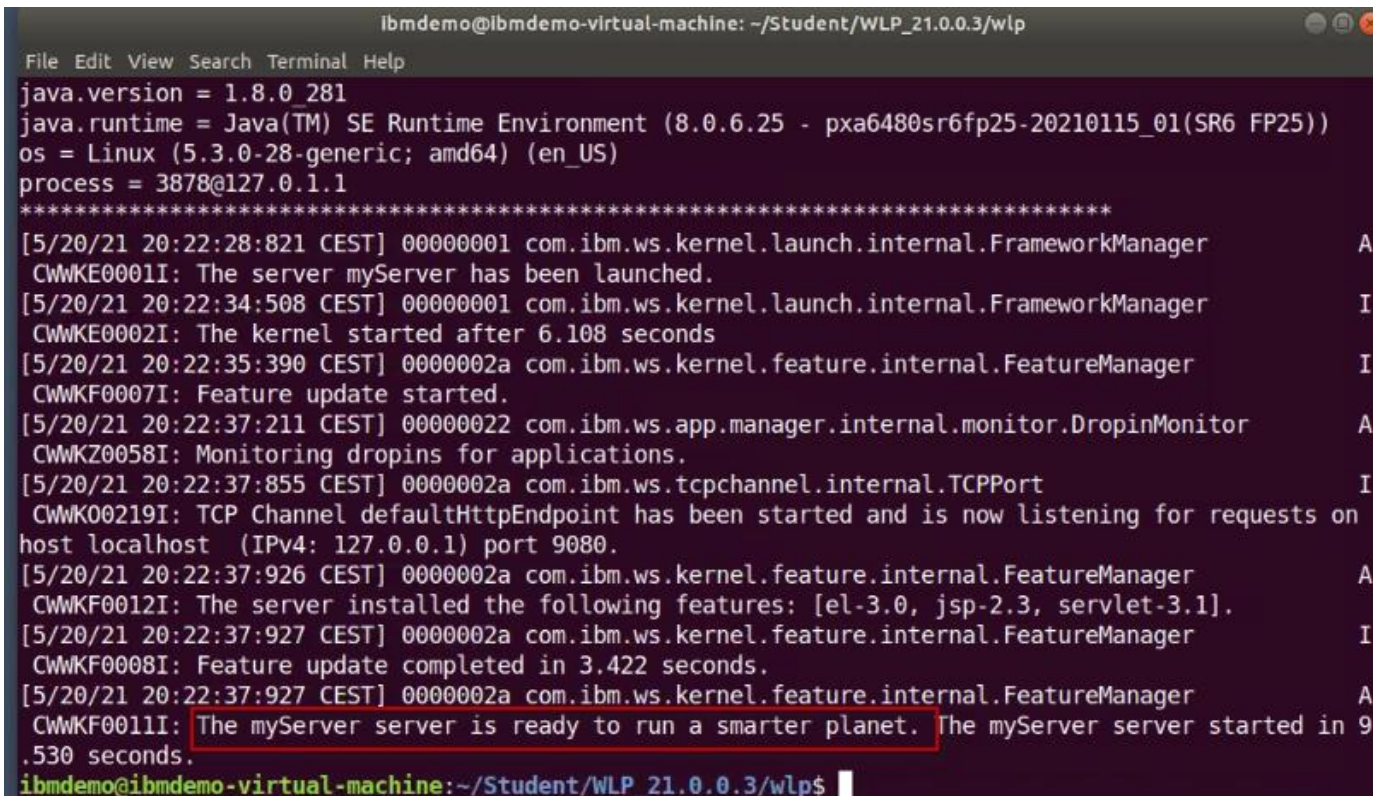
This runs the server in the background and the output is written to files in the {LAB\_HOME}/wlp/usr/servers/myServer/logs directory.

Alternatively, to start the server in the foreground (so the console messages are seen in the command window) you can use the command "**bin/server run myServer**"

- \_\_8. View the Liberty server's **messages.log** file to see the server startup messages

```
cat usr/servers/myServer/logs/messages.log
```

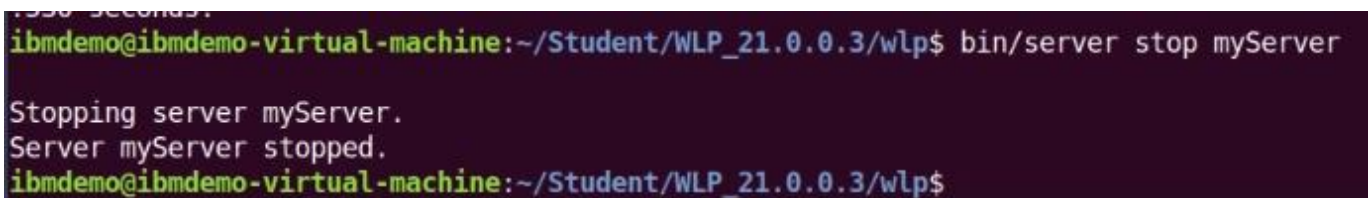
The server is started when the message “**The myServer server is ready to run a smarter planet**” is displayed in the messages.log file.



```
ibmdemo@ibmdemo-virtual-machine: ~/Student/WLP_21.0.0.3/wlp
File Edit View Search Terminal Help
java.version = 1.8.0_281
java.runtime = Java(TM) SE Runtime Environment (8.0.6.25 - pxa6480sr6fp25-20210115_01(SR6 FP25))
os = Linux (5.3.0-28-generic; amd64) (en_US)
process = 3878@127.0.1.1
*****
[5/20/21 20:22:28:821 CEST] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager
CWwKE0001I: The server myServer has been launched.
[5/20/21 20:22:34:508 CEST] 00000001 com.ibm.ws.kernel.launch.internal.FrameworkManager
CWwKE0002I: The kernel started after 6.108 seconds
[5/20/21 20:22:35:390 CEST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
CWwKF0007I: Feature update started.
[5/20/21 20:22:37:211 CEST] 00000022 com.ibm.ws.app.manager.internal.monitor.DropinMonitor
CWwKZ0058I: Monitoring dropins for applications.
[5/20/21 20:22:37:855 CEST] 0000002a com.ibm.ws.tcpchannel.internal.TCPPort
CWwK00219I: TCP Channel defaultHttpEndpoint has been started and is now listening for requests on
host localhost (IPv4: 127.0.0.1) port 9080.
[5/20/21 20:22:37:926 CEST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
CWwKF0012I: The server installed the following features: [el-3.0, jsp-2.3, servlet-3.1].
[5/20/21 20:22:37:927 CEST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
CWwKF0008I: Feature update completed in 3.422 seconds.
[5/20/21 20:22:37:927 CEST] 0000002a com.ibm.ws.kernel.feature.internal.FeatureManager
CWwKF0011I: The myServer server is ready to run a smarter planet. The myServer server started in 9
.530 seconds.
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$
```

- \_\_9. Stop the server with the **server stop** command:

```
bin/server stop myServer
```



```
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$ bin/server stop myServer
Stopping server myServer.
Server myServer stopped.
ibmdemo@ibmdemo-virtual-machine:~/Student/WLP_21.0.0.3/wlp$
```

\_\_\_10. Having verified the Liberty installation, you can delete the test server “**myServer**” simply by removing its directory.

\_\_\_a. Delete the server by deleting the {LAB\_HOME}/wlp/usr/servers/myServer directory.

```
rm -rf usr/servers/myServer
```

You now have a Liberty runtime environment that is ready to create servers to run applications.

### 1.5.1 Short recap of what you just did

In the previous step, you deleted a Liberty server configuration. You did NOT delete the Liberty installation and its binaries.

This is an important concept of Liberty. The Liberty Runtime binaries are separate from the Liberty Server configurations.

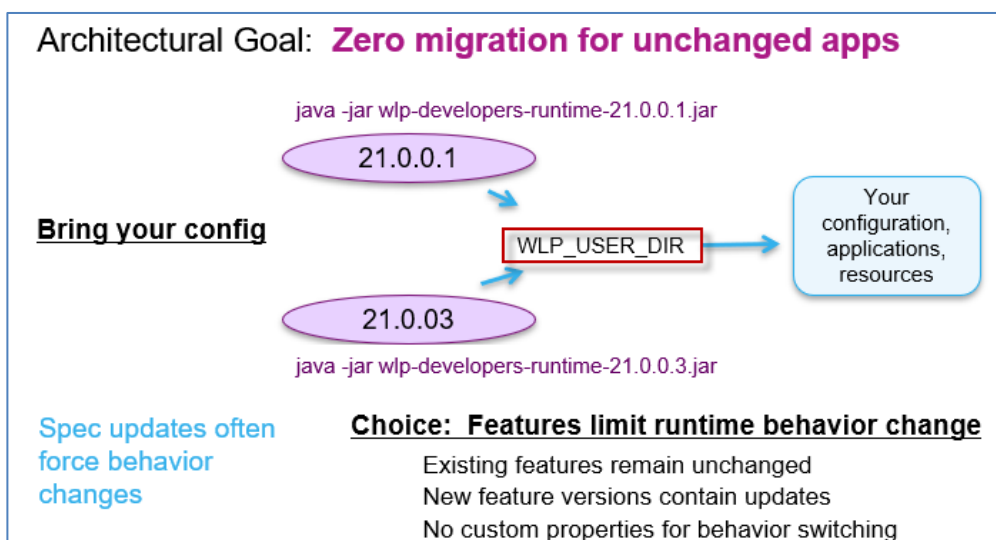
- Liberty was installed in ~/Student/WLP\_21.0.0.3/wlp
- By default, the Liberty Server configurations are in ~/Student/WLP\_21.0.0.3/wlp/usr/

The Server configuration directory can be located anywhere, including outside of the default location of /wlp/usr.

The Server configuration location can be over-ridden using the Liberty built-in variable “**`${wlp.user.dir}`**”.

In fact, this concept is instrumental in Liberty providing zero migration for unchanged apps while keeping current with Liberty Fixpacks, as illustrated below.

1. Just install a new Liberty Fixpack (Archive install)
2. Update the WLP\_USER\_DIR variable to point to your existing server configurations location.



## 1.6 Test the WebSphere Developer Tools (WDT)

You can manage Liberty from the command line and edit the server configuration files using your favorite editor.

However, the WebSphere Developer Tools (WDT) provide a great configuration editor, server controls and application publishing, as well as many other time-saving utilities. In this section, you will explore using WDT with Liberty.

Normally, you would first download and install Eclipse, followed by the installation of WDT Eclipse plugin.

For this lab, we have bundled everything into a single zip file. The directory {LAB\_HOME}/wdt contains a prebuilt and expanded WDT.



### Information:

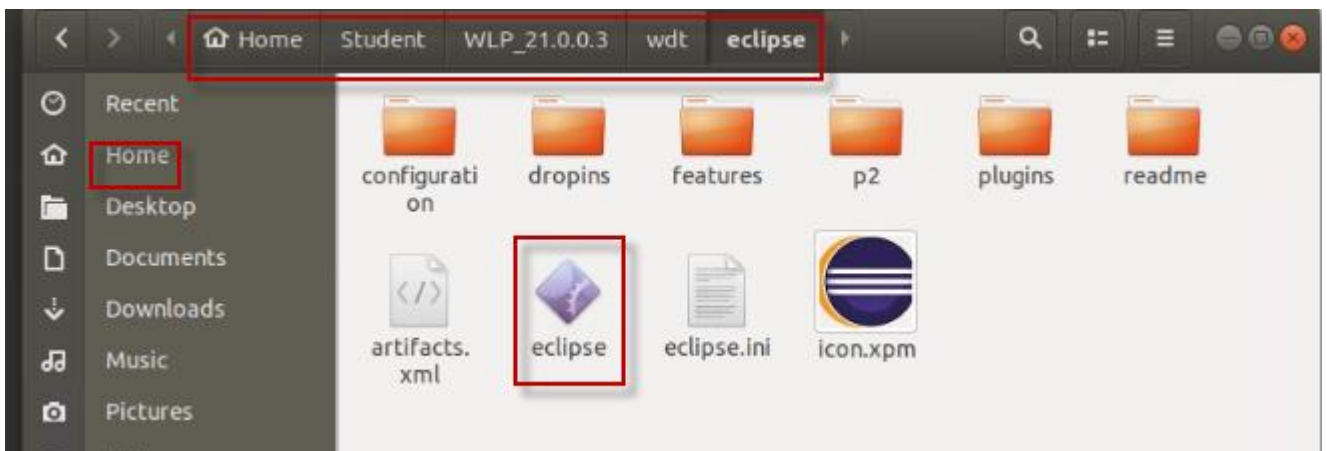
Upon first startup, it may take Eclipse up to a minute to start as it initializes.

#### \_\_1. Launch Eclipse

##### \_\_a. Use the **File Explorer** to navigate to the directory:

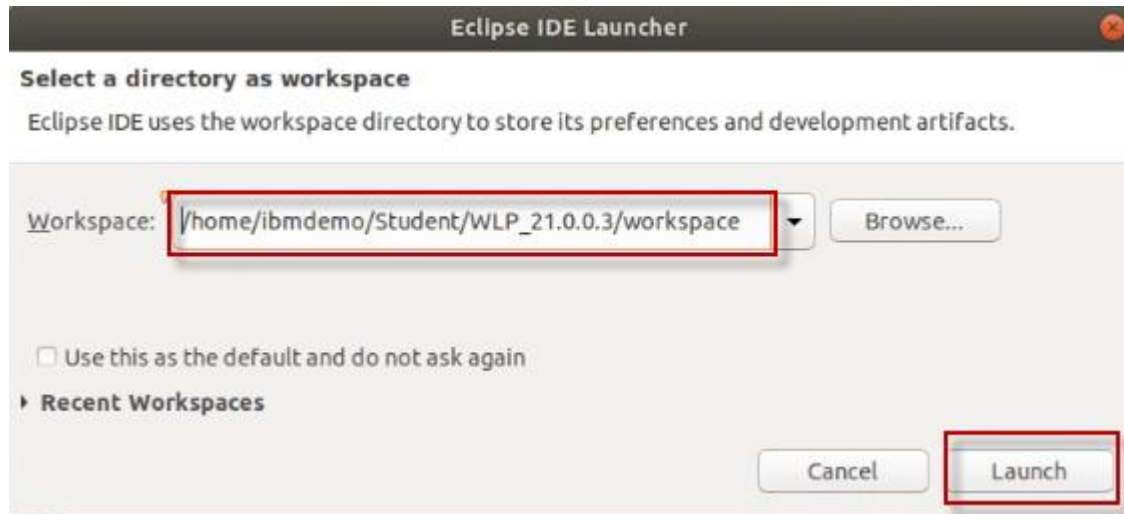
Home > Student > WLP\_21.0.0.3 > wdt > eclipse

##### \_\_b. Double-click on the **eclipse** executable to start Eclipse.

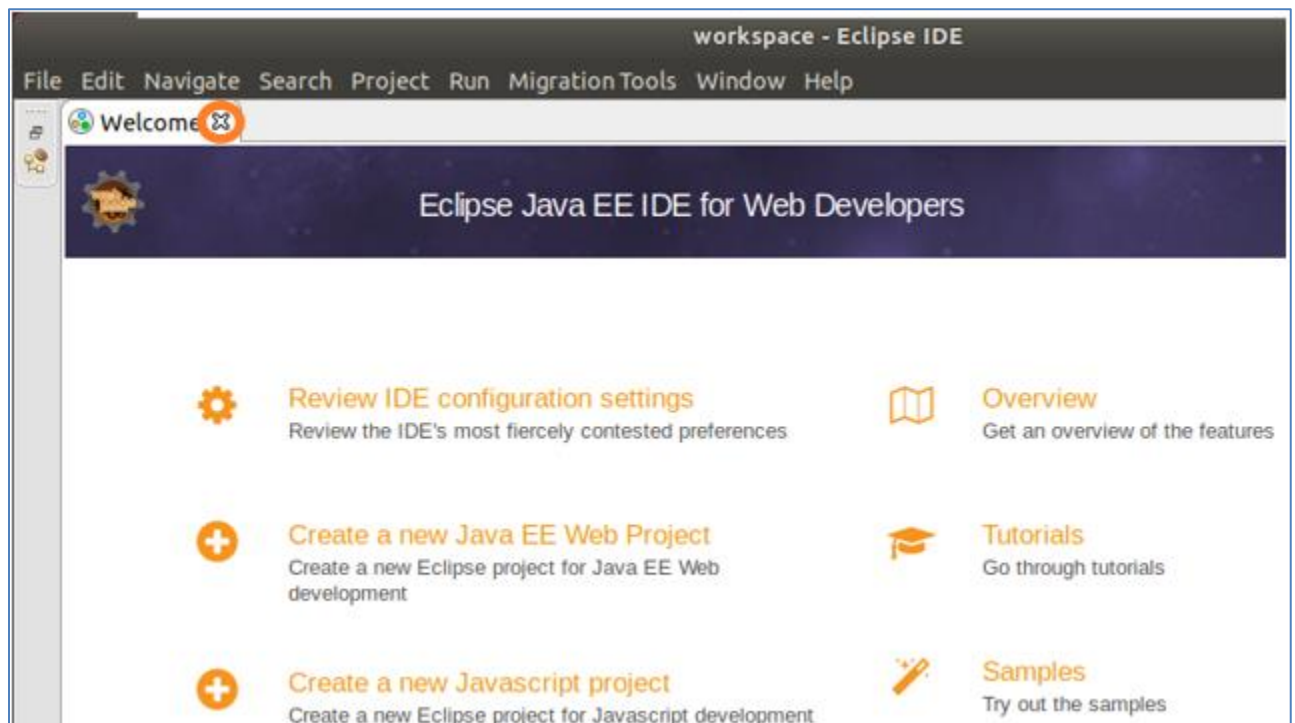


- \_\_c. When the Eclipse launcher prompts you to select a workspace, enter the following directory. Then click the **Launch** button.

```
/home/ibmdemo/Student/WLP_21.0.0.3/workspace
```



- \_\_d. Close the **welcome page** by clicking on the 'X' icon.





## 1.7 Create Liberty Server in WDT

Previously in the lab, you used the command line to create and start a Liberty server.

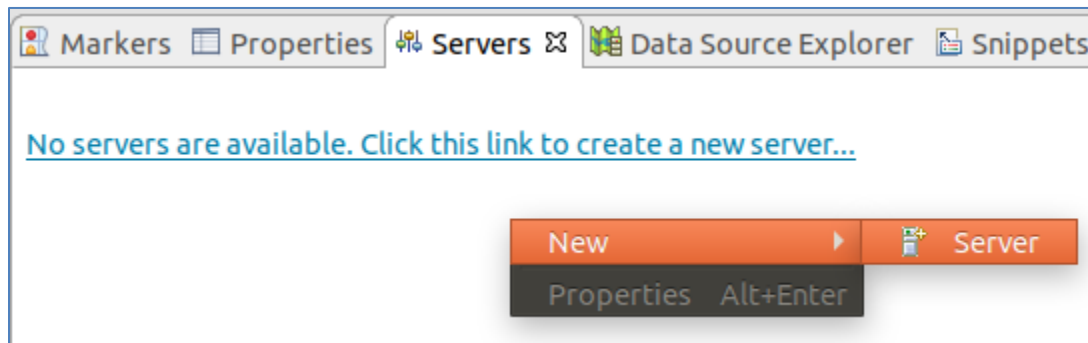
Developers often work in an integrated development environments such as Eclipse, VS Code, or IntelliJ, to name a few, to enhance their productivity.

IBM WebSphere Developer Tools for Eclipse enhances developer productivity by providing a lightweight set of tools that you can use to develop, assemble, and deploy Java EE, OSGi and mobile applications to WebSphere Application Server traditional and Liberty.

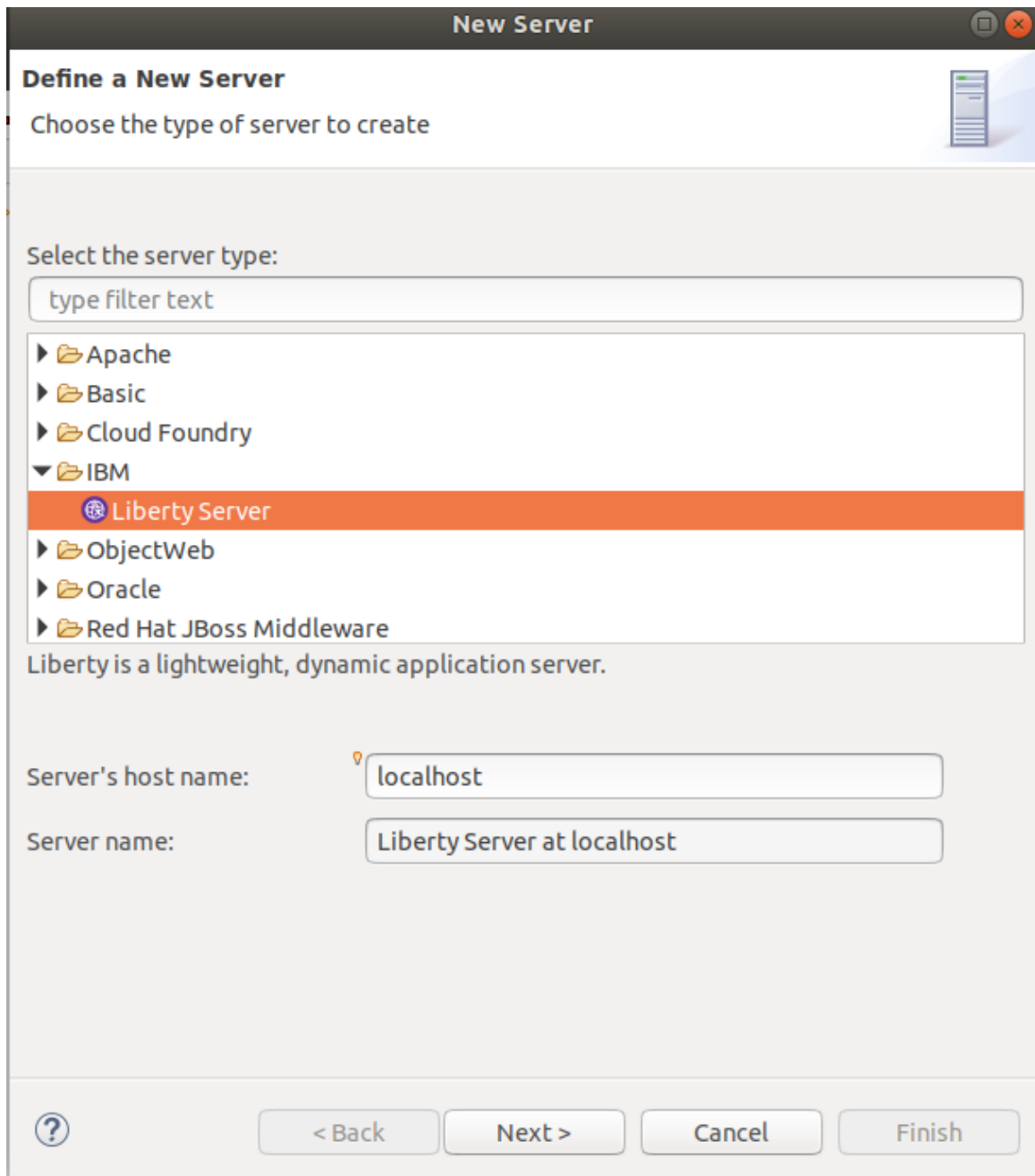
In this section, you will use the WebSphere Developer Tools with the Eclipse IDE to work with Liberty.

First, create a Liberty server using the integrated tools.

- \_\_1. At the bottom of the Eclipse workbench, open the **Servers** view by clicking the **Servers** tab.
- \_\_2. Right-click within the windows of the **Servers view** and select **New > Server**



- \_\_\_3. Under the **server type** list, expand **IBM** and select the **Liberty Server** type.



- \_\_\_4. Use the default eclipse server name as supplied (**localhost**).

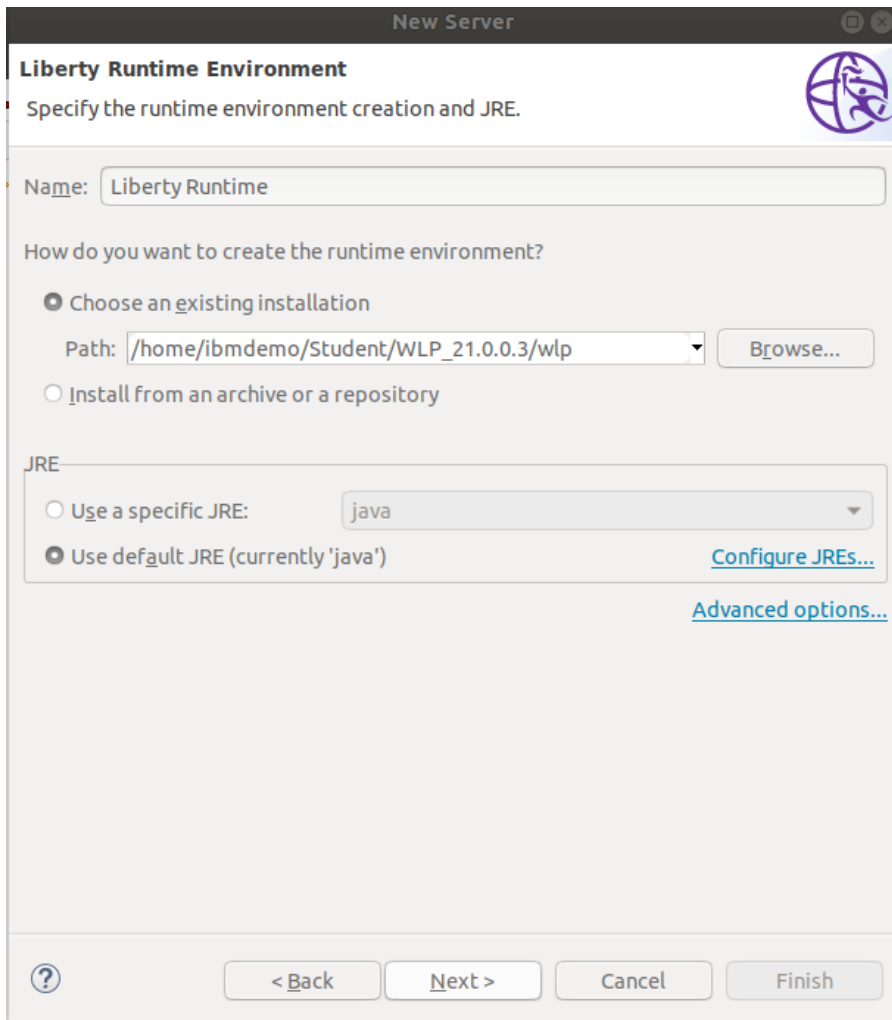
- \_\_\_a. Click **Next**.

This creates the liberty server object in eclipse, and the **Liberty Runtime Environment** page is displayed.

\_\_\_5. Now eclipse needs to associate the '**localhost**' server with a server configuration in a Liberty runtime (the runtime that you installed previously in the lab).

\_\_\_a. In the **Path** field under the "**Choose an existing installation**" section, type or browse for the directory where you installed the Liberty runtime environment shown below:

```
/home/ibmdemo/Student/WLP_21.0.0.3/wlp
```

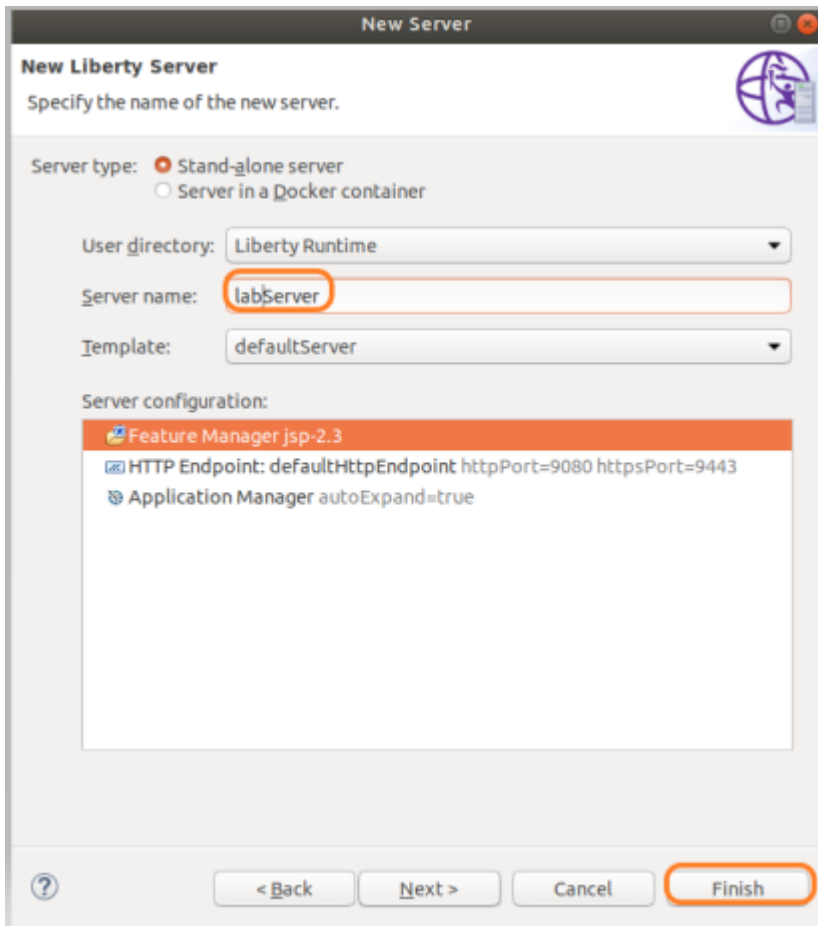


The screenshot shows the 'New Server' dialog box in the Eclipse IDE. The title bar says 'New Server'. The main heading is 'Liberty Runtime Environment' with a sub-instruction 'Specify the runtime environment creation and JRE.' and the Eclipse logo. The 'Name' field contains 'Liberty Runtime'. Under 'How do you want to create the runtime environment?', the 'Choose an existing installation' radio button is selected. The 'Path' field contains '/home/ibmdemo/Student/WLP\_21.0.0.3/wlp' and has a 'Browse...' button next to it. The 'Install from an archive or a repository' radio button is unselected. In the 'JRE' section, the 'Use default JRE (currently 'java')' radio button is selected, and the 'java' text is shown in a dropdown menu. There are links for 'Configure JREs...' and 'Advanced options...'. At the bottom, there are buttons for '< Back', 'Next >', 'Cancel', and 'Finish', along with a help icon (?) on the left.

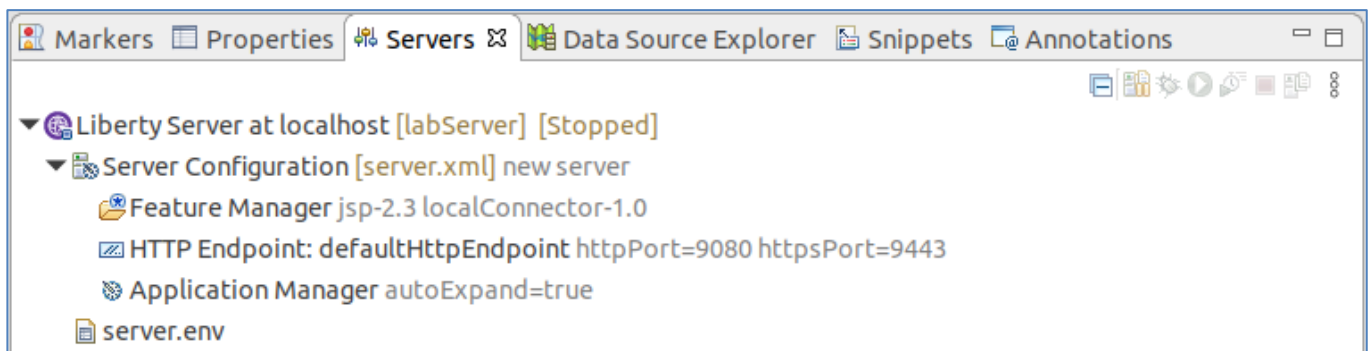
\_\_\_b. Click **Next**. to continue



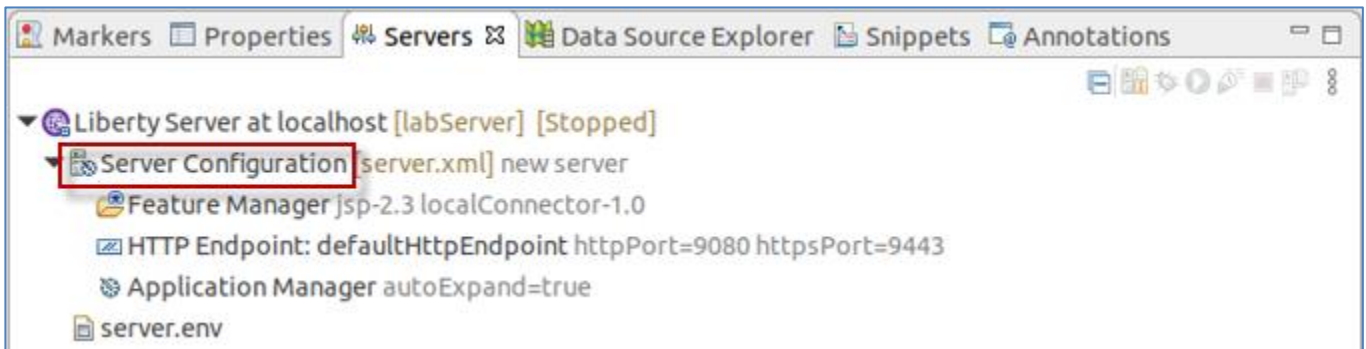
- \_\_\_6. To create the server configuration in the runtime, replace in the **Server name** field with **labServer**. Then click **Finish**.



- \_\_\_7. The new server will appear in the Servers view. You can expand the server to show a quick view of the configuration.

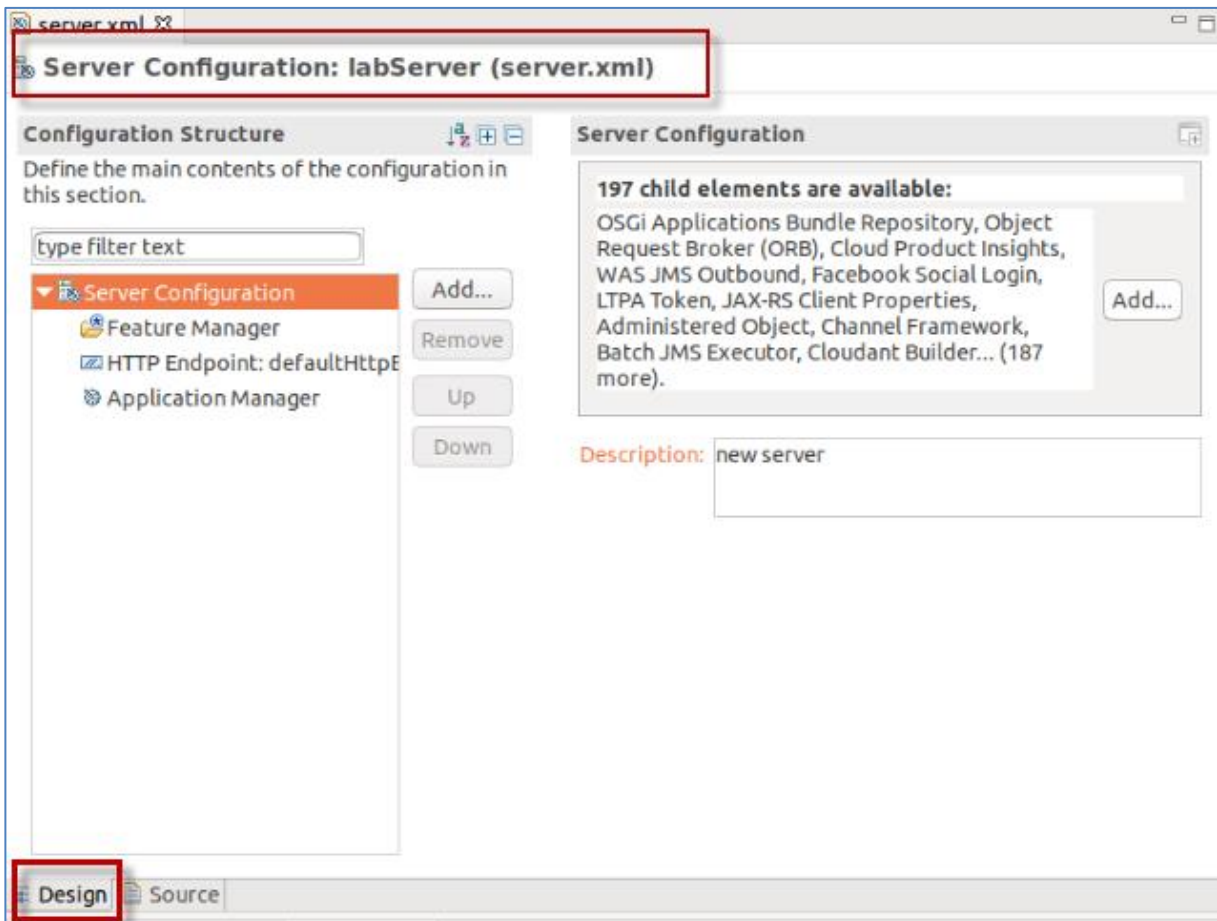


\_\_8. Open the server configuration editor by double-clicking on **Server Configuration (server.xml)**:

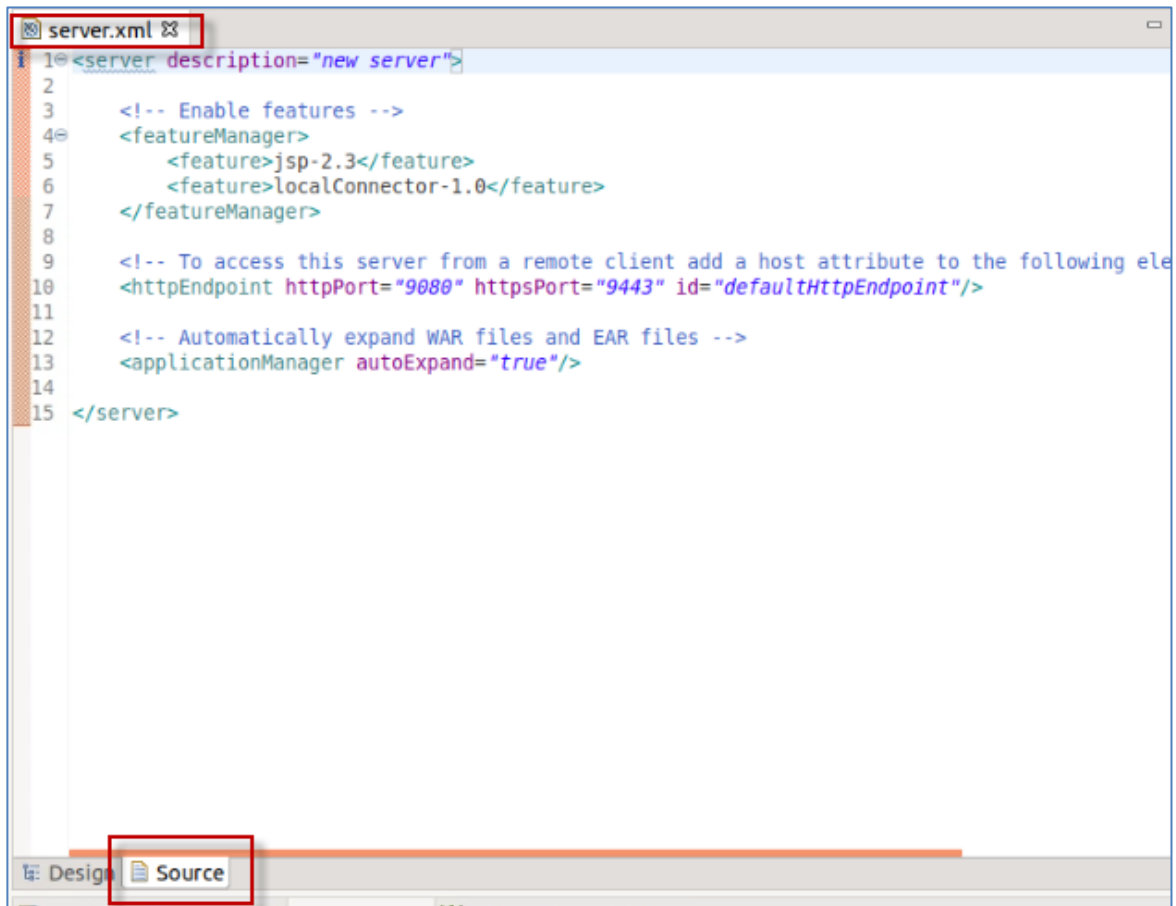


\_\_9. The Server configuration view is displayed. The default view is **“Design”** mode, which provides an intuitive server configuration editor UI.

\_\_a. Click on the **“Design”** view tab if it is not already selected.



\_\_b. You can click on the Source" view tab to edit the server.xml source directly.



## Cleanup

- \_\_10. **Close** the **server.xml** editor
- \_\_11. **Exit** Eclipse by selecting using the **File > Exit** from the Eclipse main menu.
- \_\_12. Close any open **Terminal** windows

**=== END OF LAB ===**