

CHAPTER 10: APPLICATION DEPLOYMENT

Theory

WebSphere Application Server supports three different types of applications to be deployed that are:

- *Enterprise Applications*, that can be an EAR (Enterprise Archive), a WAR (Web Application) file or a JAR (Java Archive) file. When an EJB module or Web module is installed, WebSphere Application Server will automatically pack it as an EAR file. It is similar case for WAR files, meaning, deployment process will convert WAR file in to EAR automatically.
- *Business Level Application*, is a WebSphere Application Server configuration artifact that provides a complete definition of an application from business view. It doesn't contain any application file but presents a configuration that contains units which represents application files.
- *Asset*, is an application binary file such as EAR files, EJB modules, web modules, shared libraries and etc. that are stored in a repository managed by WebSphere Application Server.

You can deploy the applications from local file system that means the file system of the system that deployment manager runs or from a remote file system which means your system that is connected to administrative console.



Task: Start Default Application

Step: Make sure that default application is running. It can be started by checking checkbox and click submit action.

Connected (unencrypted) to wk-caas-17176e45d6ae415691f1a0c72e3cd15e-93d1528abf27f0066a88ef:1 ()

new: All tasks

Welcome

Guided Activities

Servers

- New server
- All servers
- Server Types
 - WebSphere application servers
 - Liberty profile servers
 - WebSphere proxy servers
 - On Demand Routers
 - PHP servers
 - WebSphere Application Server Community Edition servers
 - Generic servers
 - WebSphere MQ servers
 - Web servers
 - Apache Tomcat servers
 - BEA WebLogic servers
 - JBoss servers
 - External WebSphere Application Servers
 - Apache servers
 - Custom HTTP servers

Clusters

DataPower

Core Groups

Applications

- All applications
- New Application
- Install New Middleware Application

All Applications

This page lists all the applications supported and managed by WebSphere.

Preferences

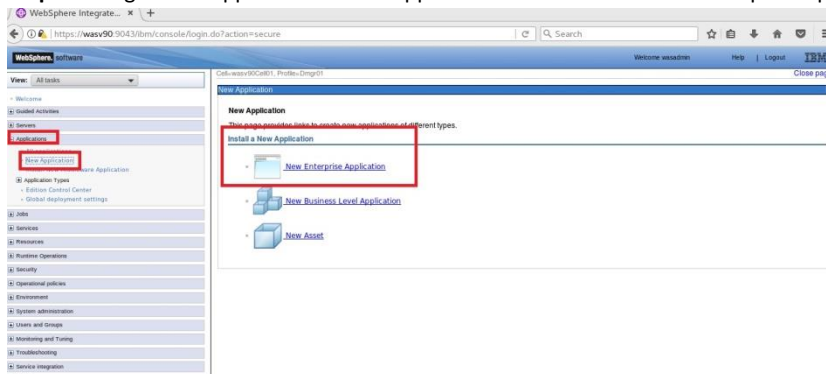
Add... Remove Submit Action

Select	Name	Edition	Edition State	Type	Status	Action
<input checked="" type="checkbox"/>	DefaultApplication	Base edition	Active	Java 2 Platform, Enterprise Edition		Start

Total 1

Task: Deploy a sample application

Step 1: Navigate to “Applications>New Application” and click on “New Enterprise Application”.



Step 2: Run the following commands in the terminal:

```
$ cd /headless/Desktop/websphere
```

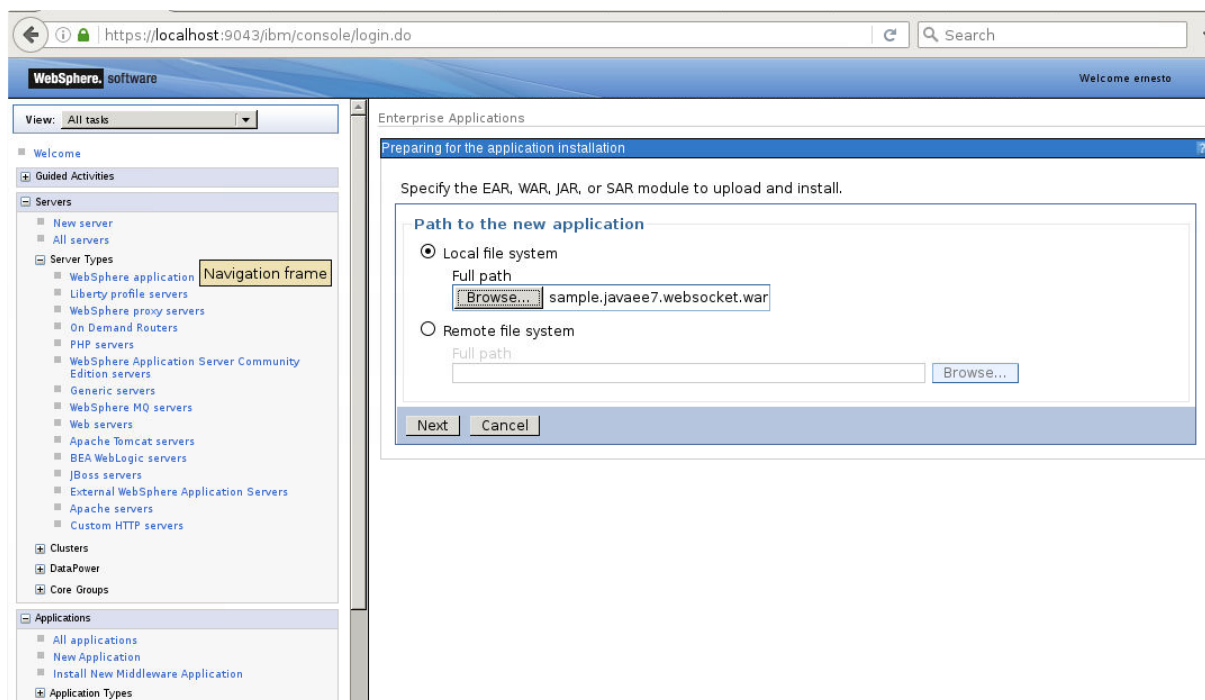
```
$ jar -xvf sample.javaee7.websocket.war
```

```
$ cd ./sample.javaee7.websocket/src/main/webapp
```

```
$ jar -cvf MyWebApp.war *
```

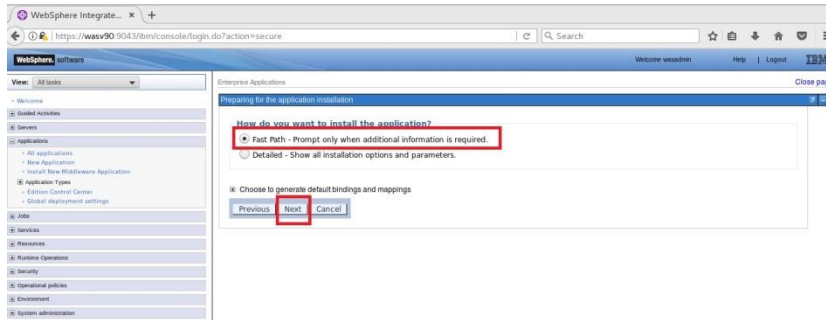
Locate the “MyWebApp.war” war file on the local file system and click “Next”.

/headless/Desktop/websphere

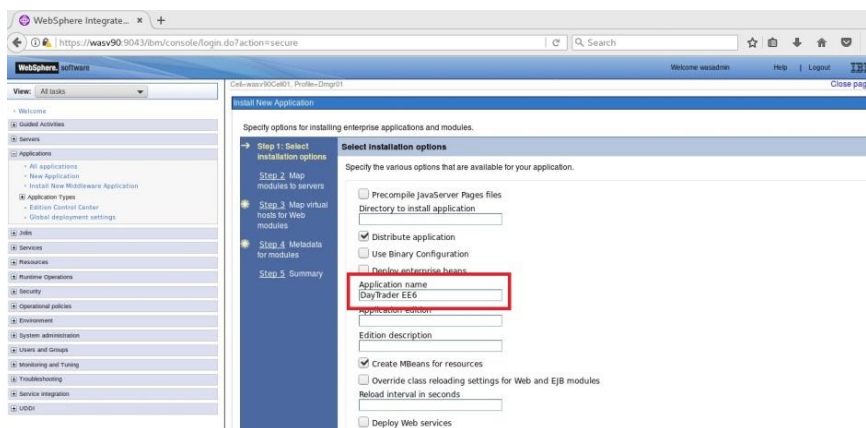




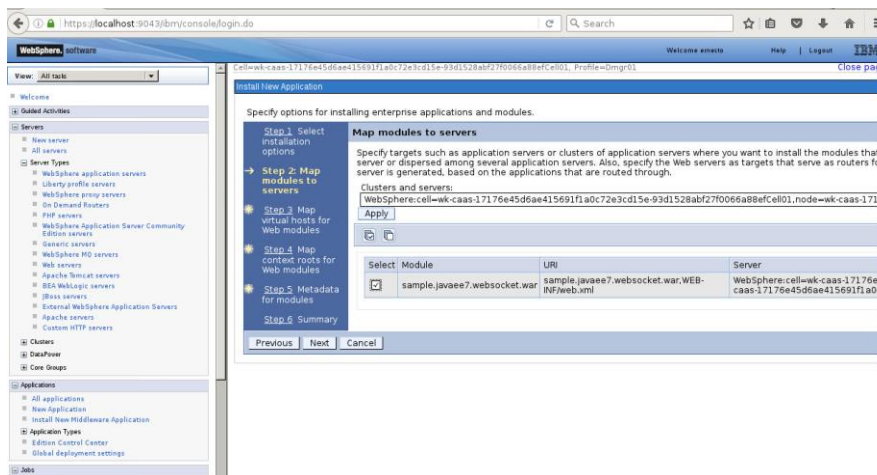
Step 3: Select “Fast Path” and click “Next”.



Step 4: Change application name to “SampleApplication” and click “Next”.



Step 5: Map module of the application and click “Apply”, then click “Next”.





Step 6: Enter context root **“/sample”** and Click **“Next”**.

The screenshot shows the 'Install New Application' wizard in the WebSphere Integrated Solutions Console. The 'Map context roots for Web modules' step is active, showing a table for mapping web modules to context roots.

Web module	URI	Context Root
sample.javaee7.websocket.war	sample.javaee7.websocket.war, WEB-INF/web.xml	/sample

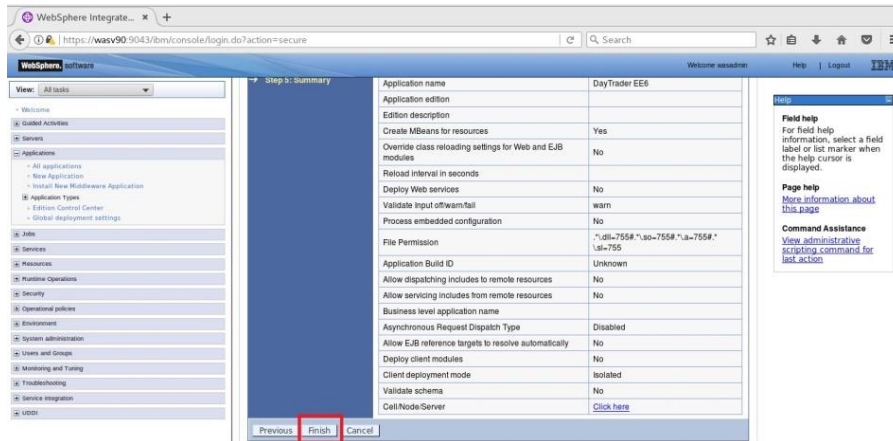
The left sidebar shows the navigation tree with 'Step 4: Map context roots for Web modules' selected. The bottom of the wizard has 'Previous', 'Next', and 'Cancel' buttons.

The screenshot shows the 'Install New Application' wizard in the WebSphere Integrated Solutions Console. The 'Metadata for modules' step is active, showing a table for mapping web modules to context roots and a checkbox for the 'metadata-complete' attribute.

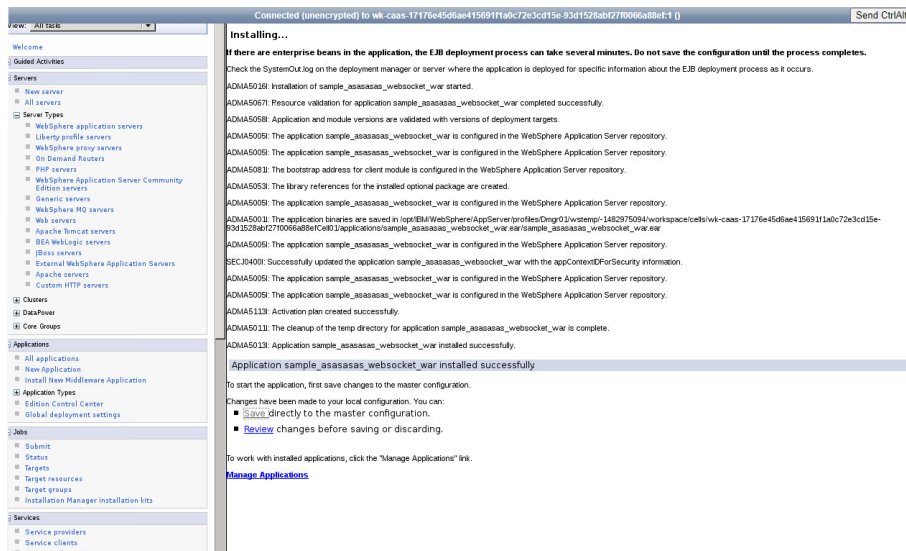
Module	URI	metadata-complete attribute
sample.javaee7.websocket.war	sample.javaee7.websocket.war, WEB-INF/web.xml	<input checked="" type="checkbox"/>

The left sidebar shows the navigation tree with 'Step 5: Metadata for modules' selected. The bottom of the wizard has 'Previous', 'Next', and 'Cancel' buttons.

Step 7: Click “Finish” to start deployment.



Step 8: You should see the success message, then click “Save” to write changes to the master configuration.

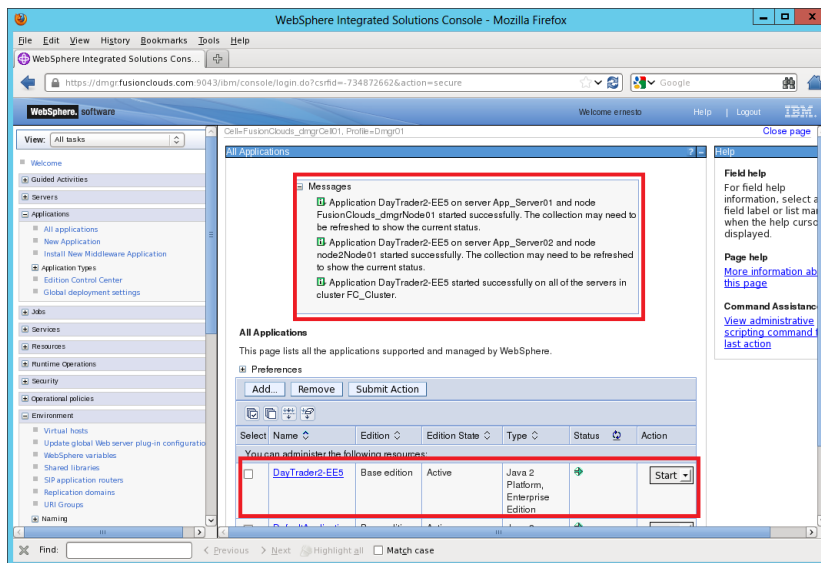




Step 9: Select the application and action as “Start” and click “Submit Action”.

Note: Wait for couple of minutes before starting application so that changes are synced.

You should see the success message as follows. Reload status to get updated status.



Task is complete!

Task: Access New Application

<http://localhost:9080/sample>

**this assumes you mapped to the URI of sample.

<http://<update>.eastus.azurecontainer.io:9080/sample.javaee7.websocket/>

Access Default applications

Open browser and enter “http://localhost:9080/snoop” to access application.

JSP can be accessed at “http://localhost:9080/HitCount.jsp”

