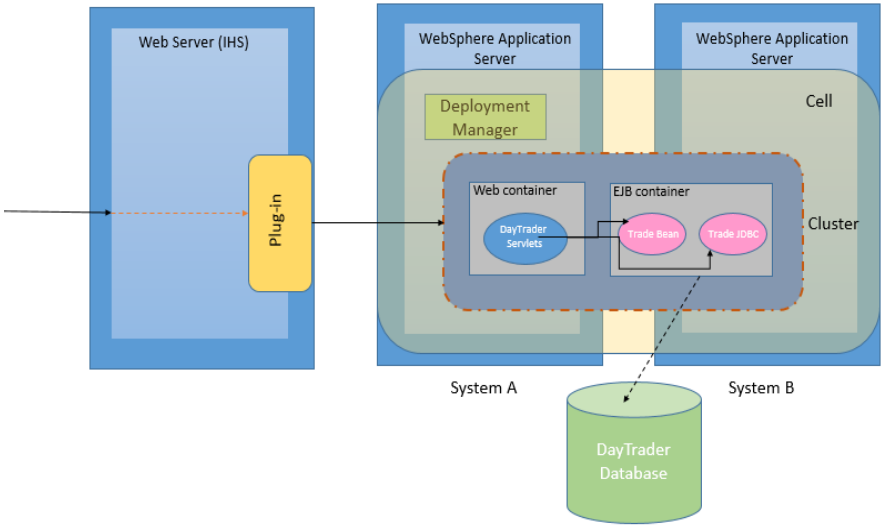
CHAPTER 10: APPLICATION DEPLOYMENT

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.



Before deploying an application, it is better to setup JDBC and JMS settings that will be used by the application. During installation of an application, you can set many options specific to your deployment.

One of the most important steps during the installation is mapping the application modules. You need to map the application to appropriate scope and also to your web server. This will ensure that with proper plug-in configuration, your server will be able to route the requests to your application.

AIM

Day Trader is a benchmark application that simulates an online stock trading system. It was originally developed by IBM and later donated to the Apache Geronimo community. In this lab exercise, we will install Day Trader application and to achieve this goal, we need to follow 2 steps:

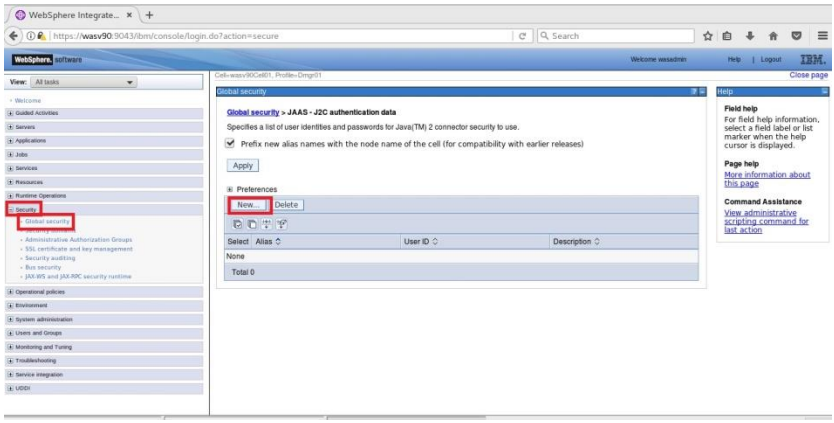
Task 1: Setup JMS and JDBC settings
Task 2: Deploy a sample application

Lab Exercise 10: APPLICATION DEPLOYMENT

1. **Setup JMS and JDBC settings**
2. **Deploy a sample application**

Task 1: Setup JMS and JDBC Settings

Step 1: Navigate to “Security>Global security>JAAS-J2C authentication data” and click “New”.



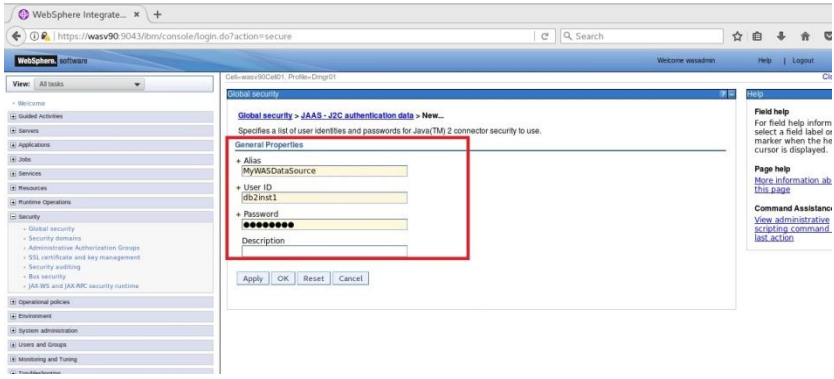
Step 2: Add authentication alias for the database connection and administrative user.

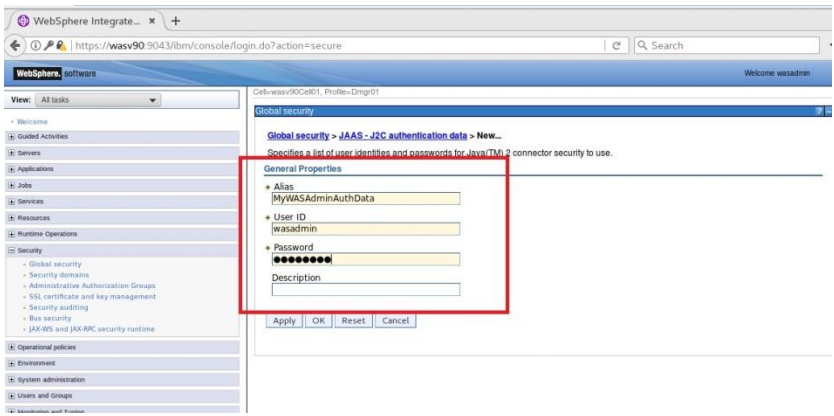
Alias: MyWASDataSourceAuthData,

User ID: db2inst1 (Database user to run queries.)

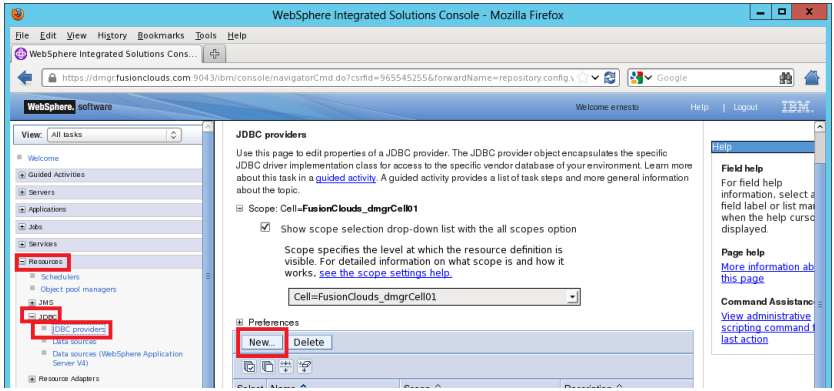
Alias: MyWASAdminAuthData

User ID: ernesto (Administrative user of WebSphere Application Server)

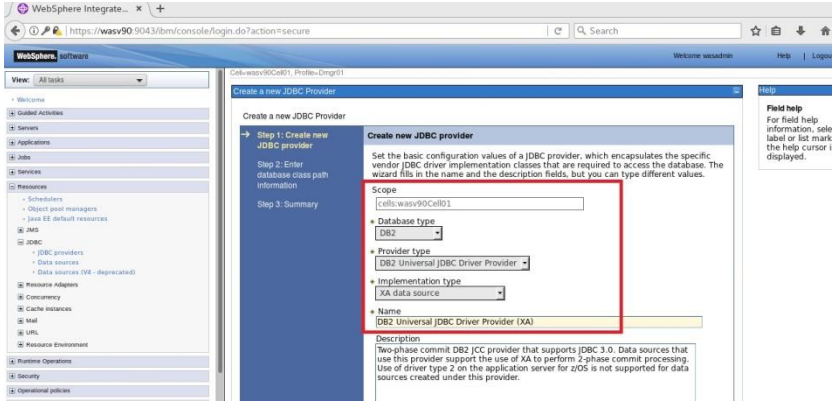




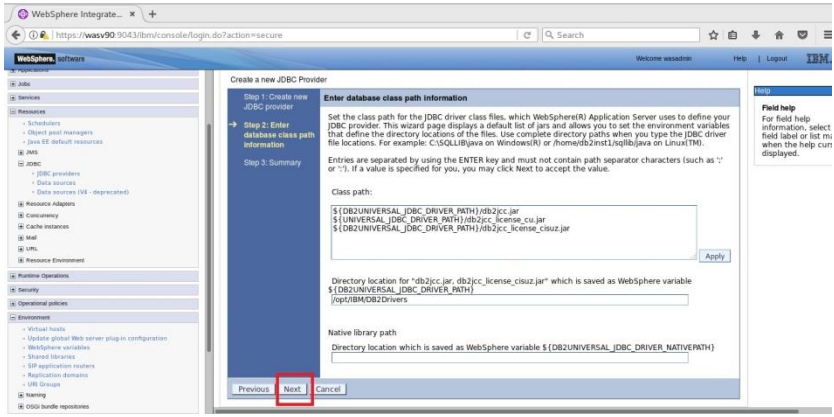
Step 3: Navigate to “Resources>JDBC>JDBC providers” and for the cell scope click “New”.



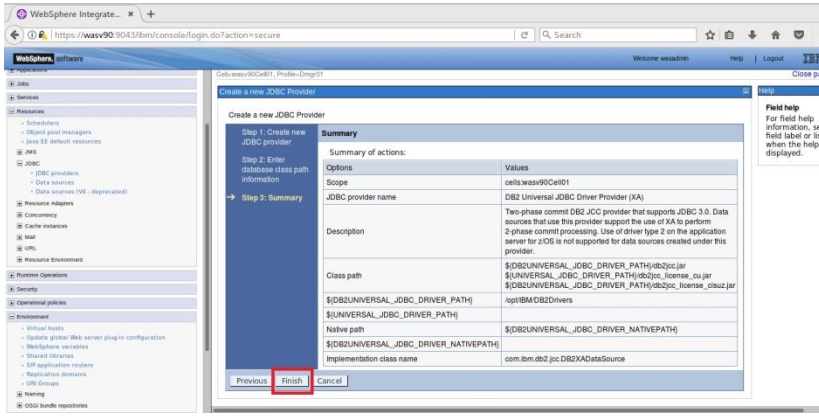
Step 4: Enter the database details. You can use the image below for DB2 configuration.



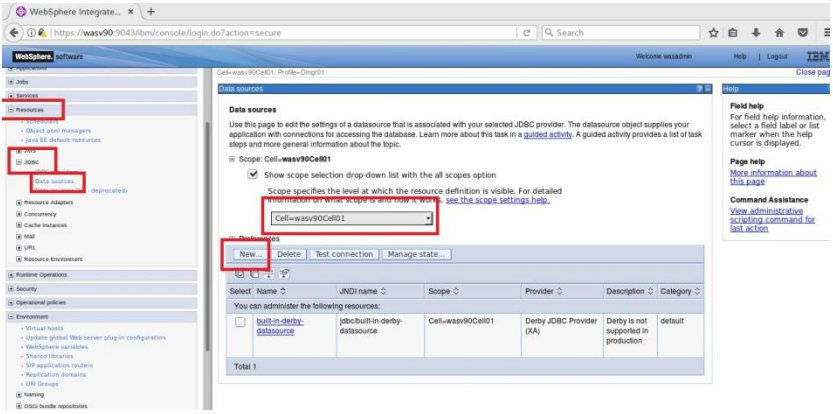
Step 5: Enter the path of the database drivers and then click “Next”.



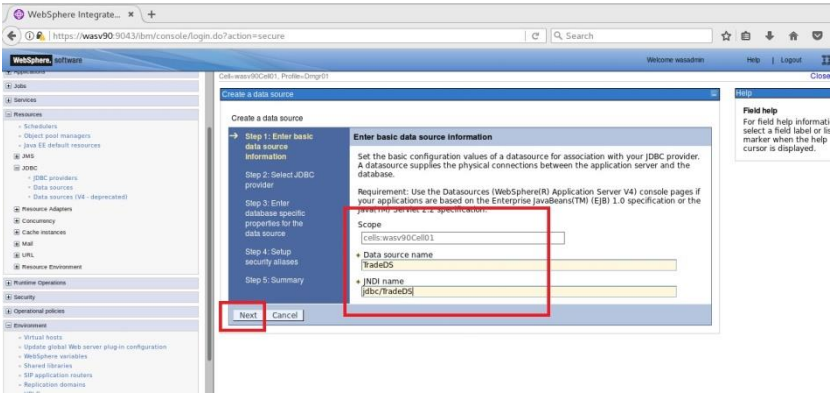
Step 6: Click “Finish” to complete.



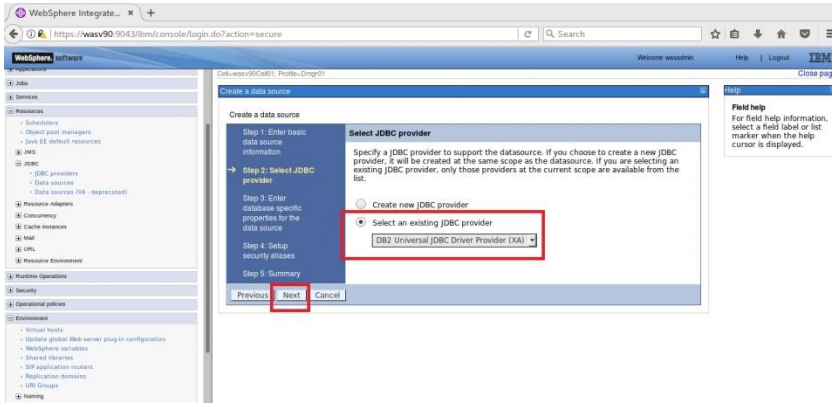
Step 7: Navigate to “Resources>JDBC>Data sources” and for the cell scope click “New”.



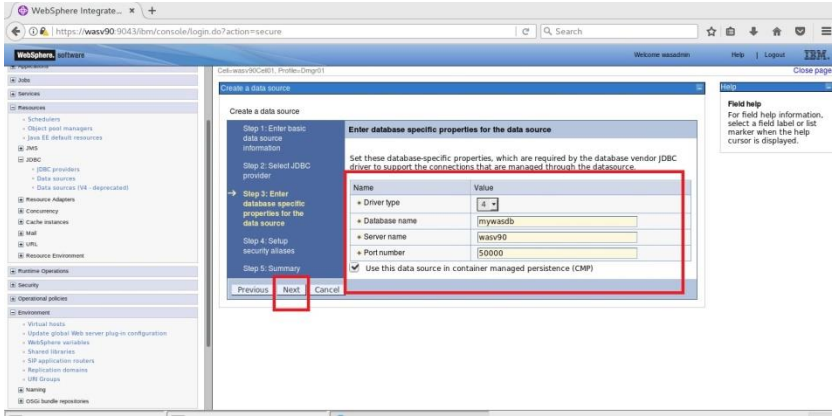
Step 8: Use “TradeDS” as a name and “jdbc/TradeDS” ad JNDI name and then click “Next”.



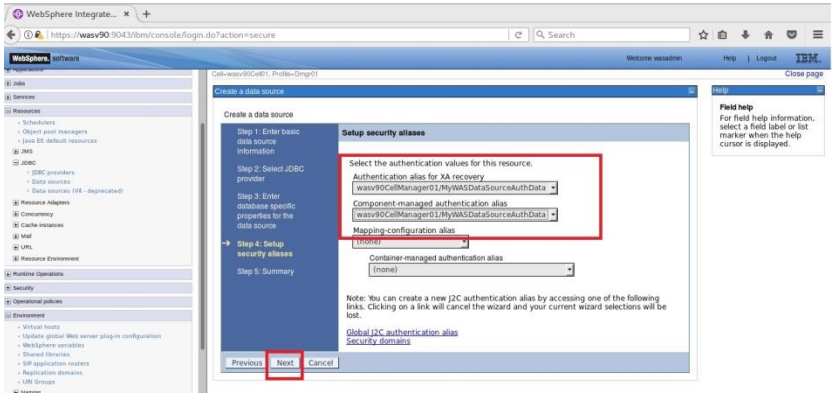
Step 9: Select the newly created JDBC provider and click “Next”.



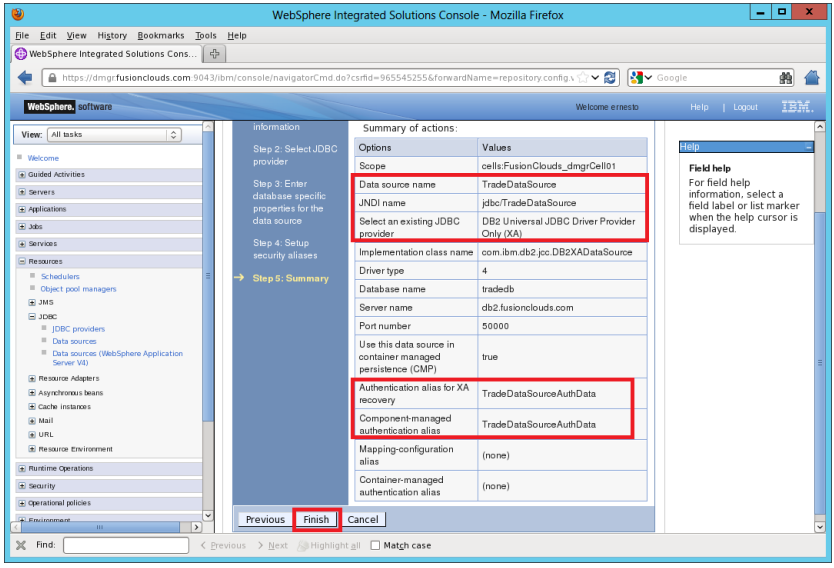
Step 10: Enter database specific properties. (database name is “mywasdb”).



Step 11: Select “MyWASDataSourceAuthData” for authentication alias setting as below.



Step 12: Click “Finish” to complete.



Step 13: Repeat the same steps to add following data sources.

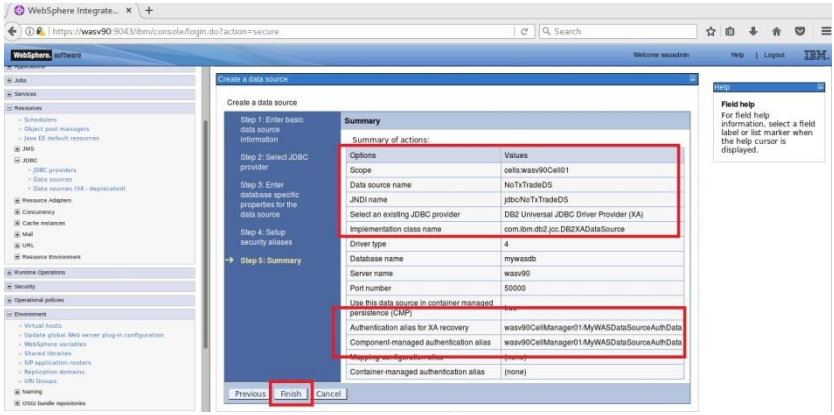
Data source Name:NoTxTradeDS

JNDI name: jdbc/ NoTxTradeDS

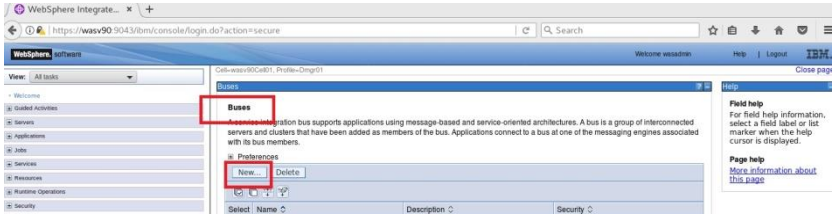
Data source Name:METradeDS

JNDI name: jdbc/ METradeDS

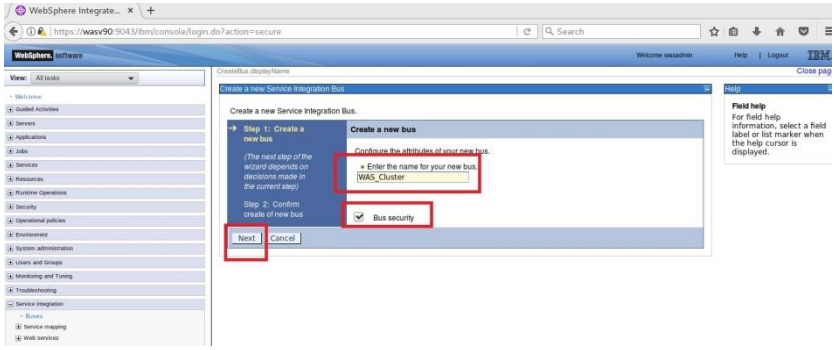

UNDER
JMS & JDBC
Deploy App.


WebSphere Integrate...
https://wasv90:9043/ibm/console/login.do?action=secure
Create a data source
Create a data source
Step 1: Enter basic data source information
Step 2: Select JDBC provider
Step 3: Enter database specific properties for the data source
Step 4: Setup security aliases
Step 5: Summary
Summary
Summary of actions:
Options
Values
Scope
cells:wasv90Cell01
Data source name
NoTxTradeDS
JNDI name
jdbc/NoTxTradeDS
Select an existing JDBC provider
DB2 Universal JDBC Driver Provider (XA)
Implementation class name
com.ibm.db2.jcc.DB2XADataSource
Driver type
4
Database name
mywasdb
Server name
wasv90
Port number
50000
Use this data source in container managed persistence (CMP)
Authentication alias for XA recovery
wasv90CellManager01/MyWASDataSourceAuthData
Component-managed authentication alias
wasv90CellManager01/MyWASDataSourceAuthData
Container-managed authentication alias
(none)
Previous
Finish
Cancel
Help
Field help
For field help information, select a field label or list marker when the help cursor is displayed.

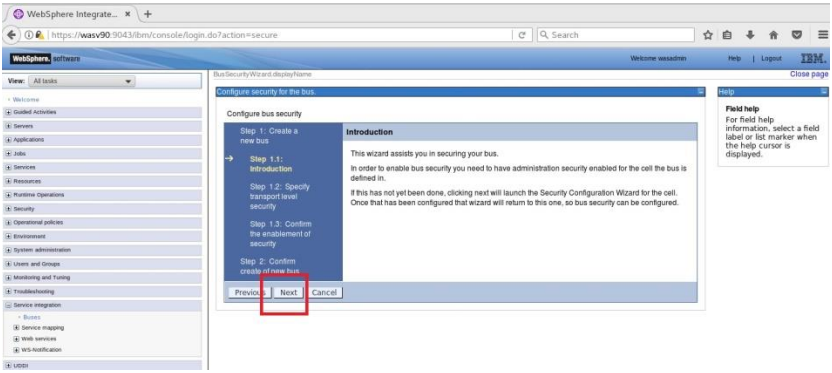
Step 14: Navigate to “Service integration>Buses” and click “New”.



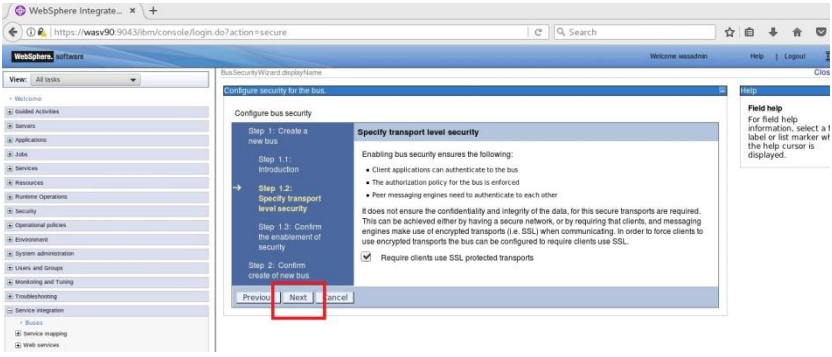
Step 15: Enter name for the bus “WAS_Cluster” and mark “Bus security” then click “Next”.



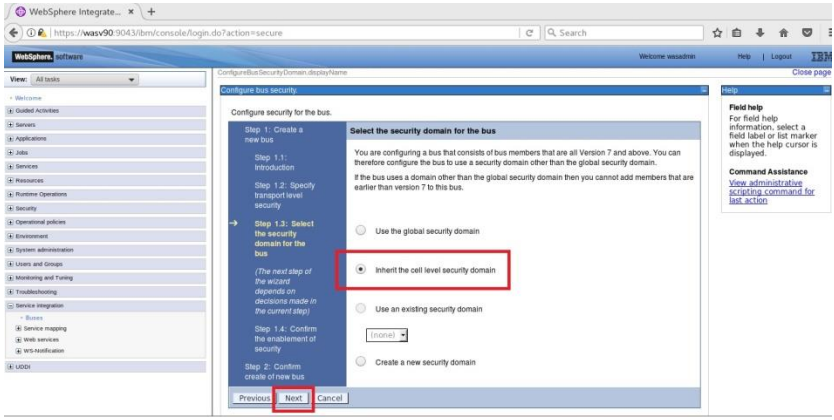
Step 16: Click “Next” to continue”.



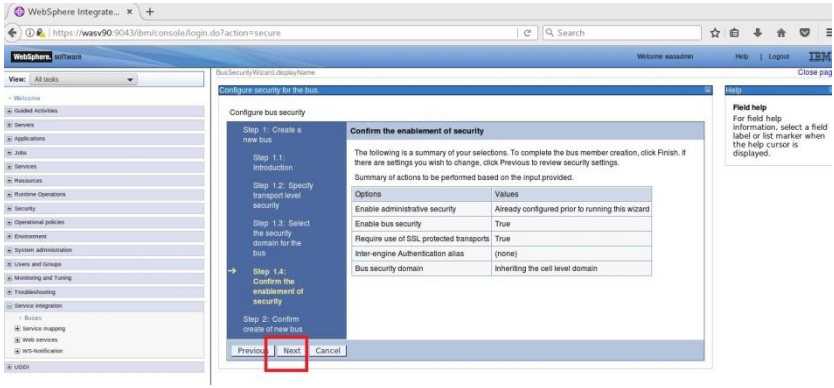
Step 17: Click “Next” to continue.



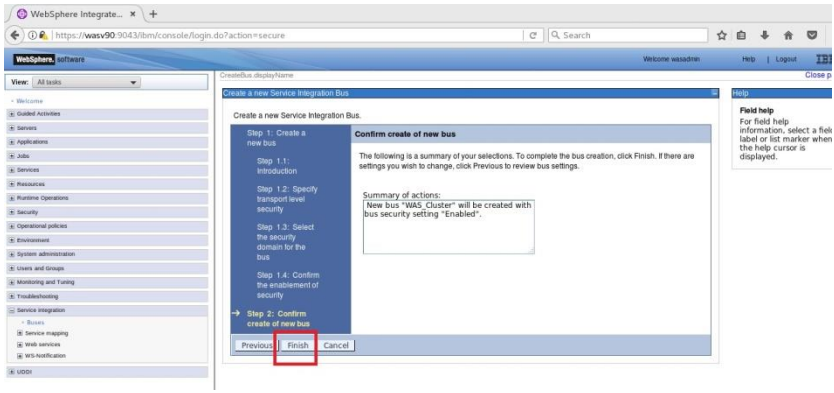
Step 18: Select “Inherit the cell level security domain” and click “Next”.



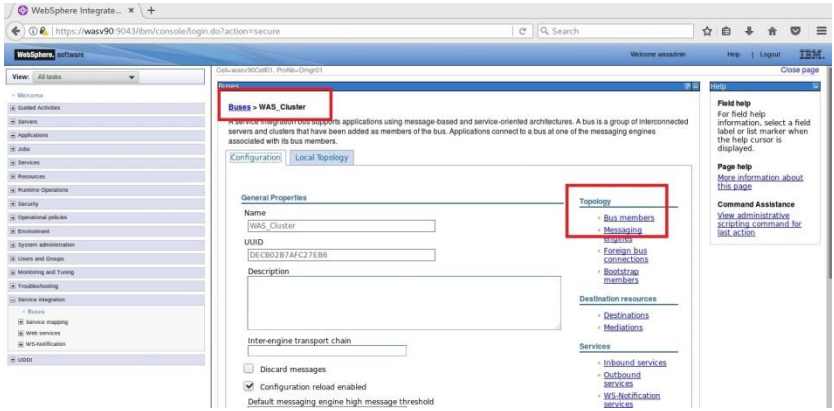
Step 19: Click “Next” to continue.



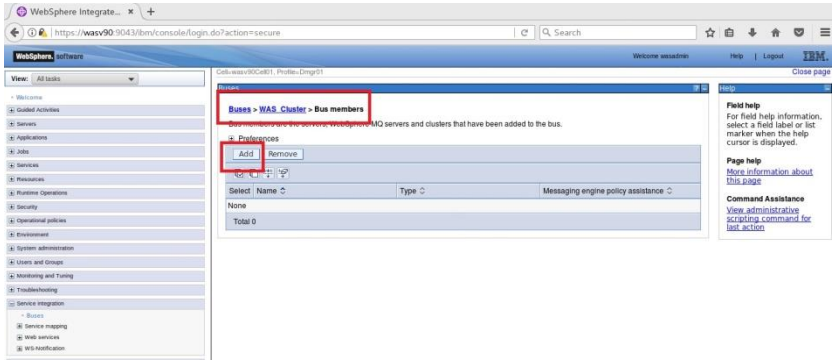
Step 20: Click “Finish” to confirm create of the bus.



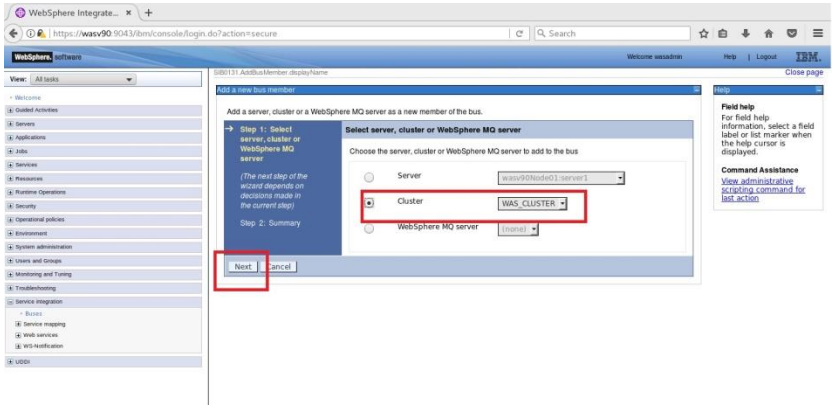
Step 21: Navigate to “Service integration>Buses>WAS_Cluster” and click “Bus members” under “Topology”.



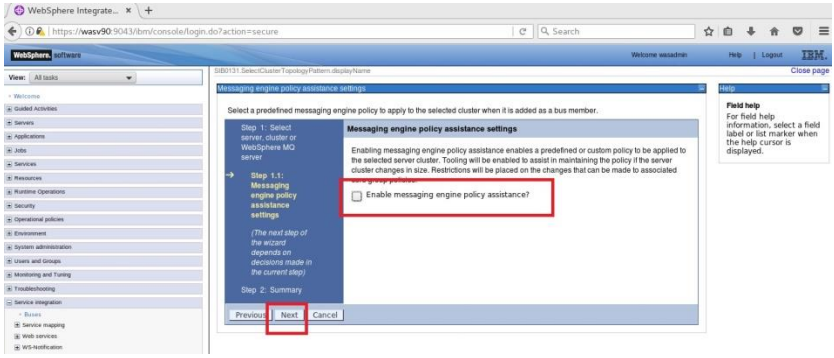
Step 22: Click “Add”.



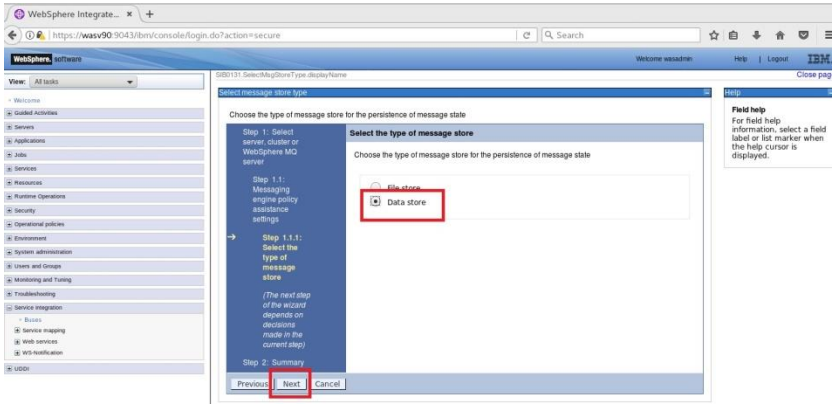
Step 23: Select “Cluster” and “WAS_CLUSTER” then click “Next”.



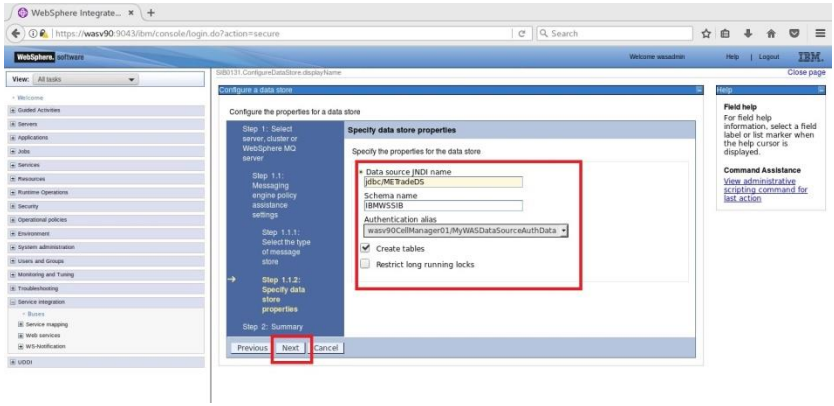
Step 24: Uncheck “Enable messaging engine policy assistance” and click “Next”.



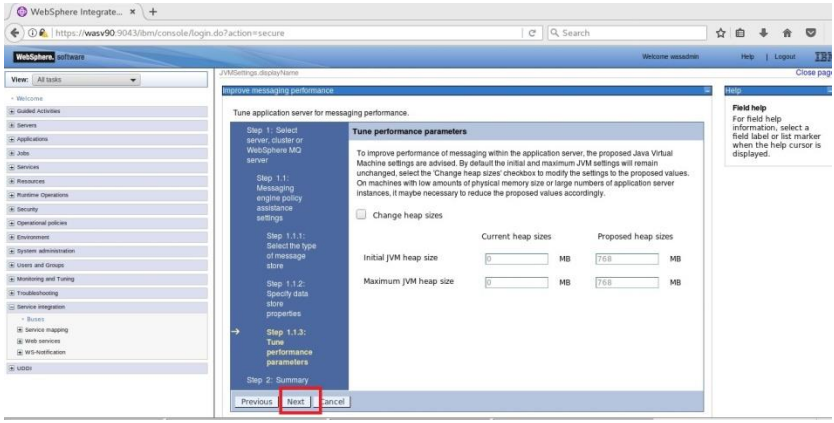
Step 25: Select “Data store” and click “Next”.



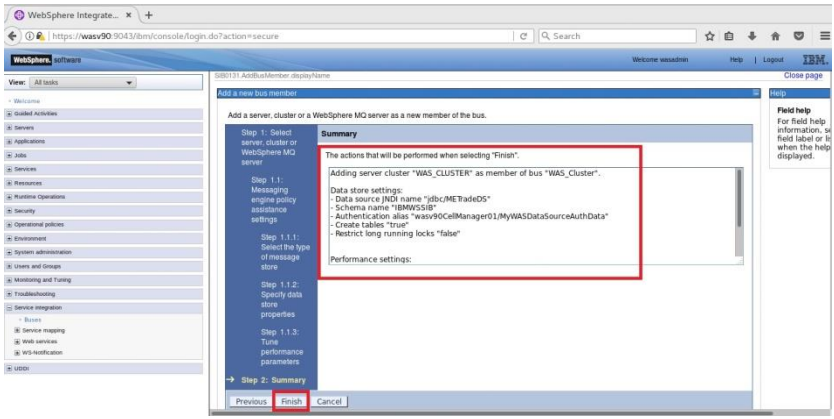
Step 26: Configure JNDI name as “jdbc/MEDataSource” and select the authentication alias “TradeDataSourceAuthData” then click “Next”.



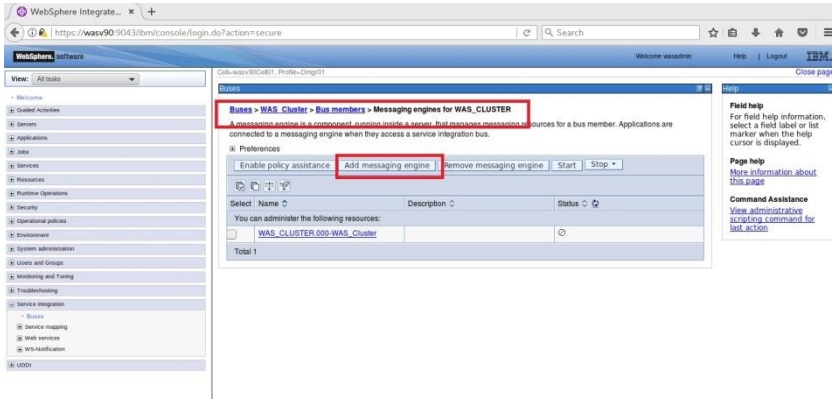
Step 27: Select “Next” to continue.



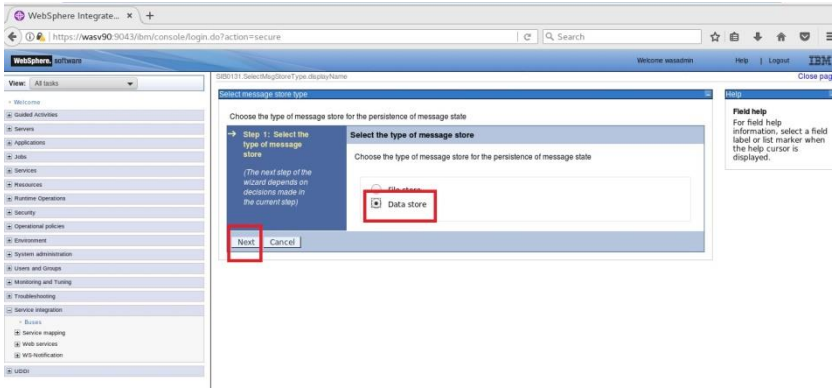
Step 28: Click “Finish” to complete.



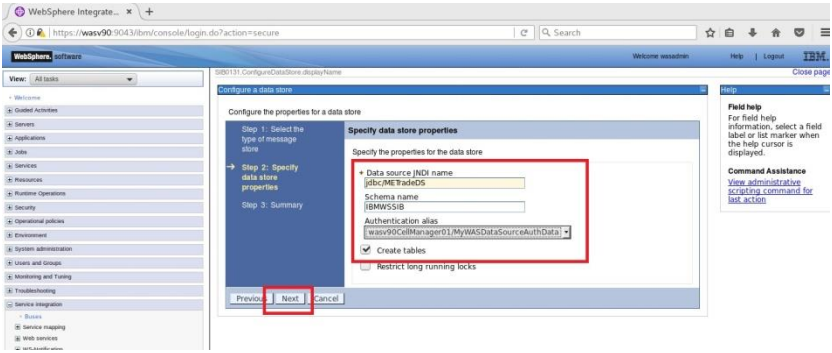
Step 29: Navigate to “System integration>Buses>WAS_Cluster>Bus members>Messaging engines” and click “Add messaging engine”.



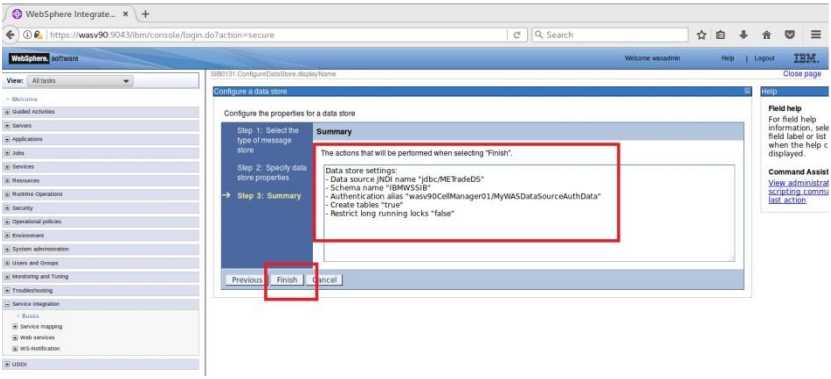
Step 30: Select “Data store” and click “Next”.



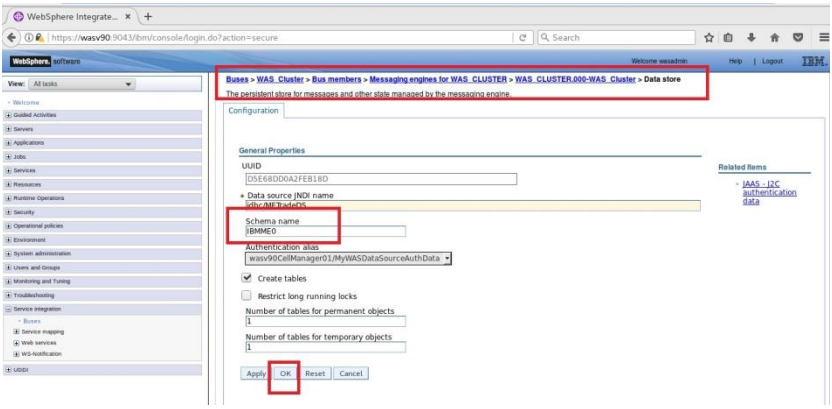
Step 31: Use “jdbc/MEDataSource” as JNDI name and select “TradeDataSourceAuthData” for authentication alias and click “Next”.



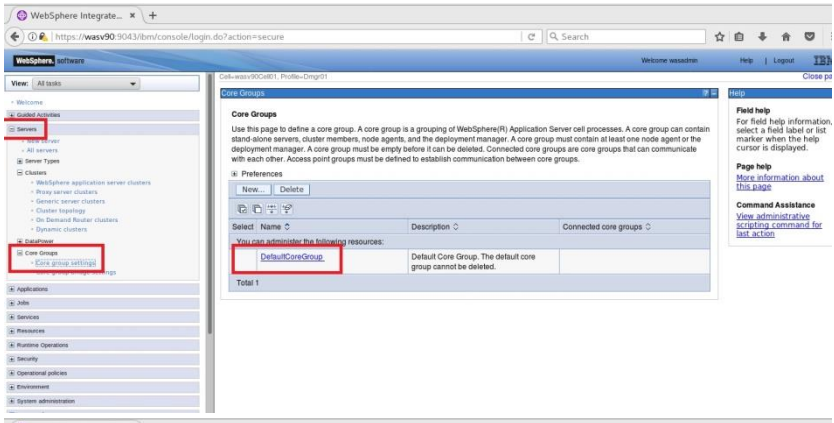
Step 32: Click “Finish” to complete.

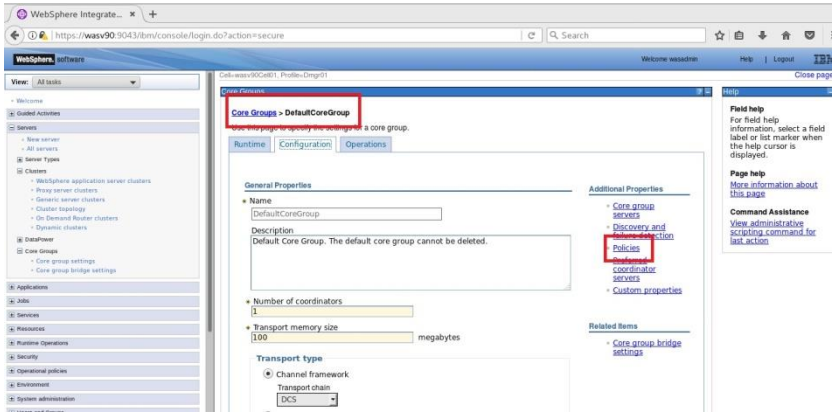


Step 33: Navigate to “Service integration>Buses>WAS_Cluster>Bus members>Messaging engines for WAS_CLUSTER>WAS_CLUSTER.000-WAS_Cluster”, under additional properties Messaging Store opens up the Data store and change the schema as “IBMME0” then click “OK”.

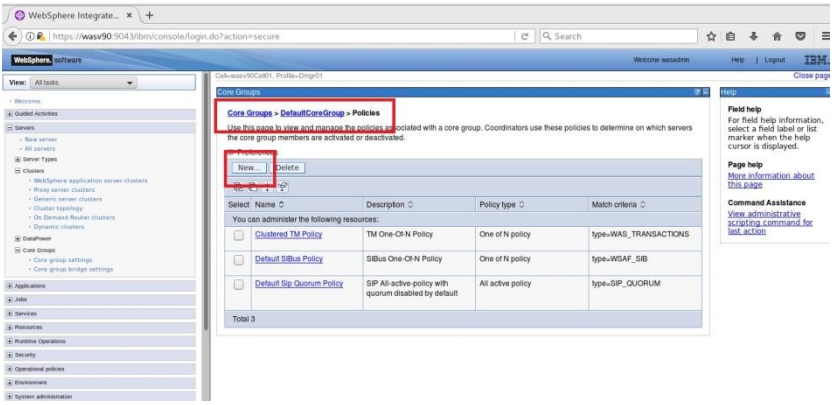


Step 34: Navigate to “Servers>Core Groups>Core group settings” and click on “DefaultCoreGroup” and then click on “Policies” under “Additional properties”.

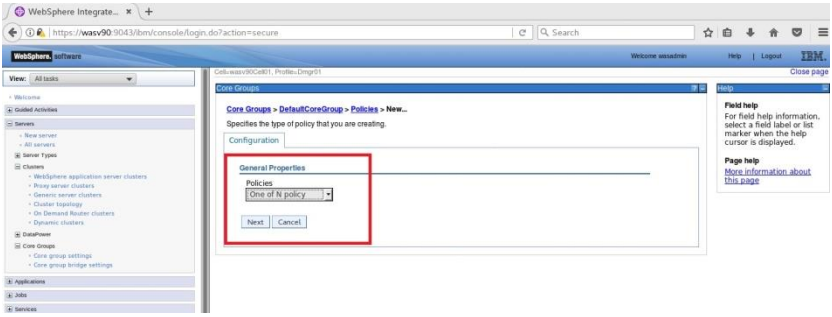




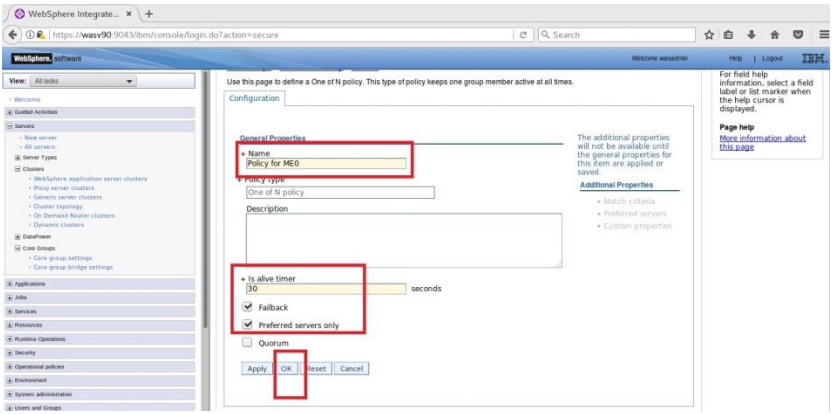
Step 35: Click on “New”.



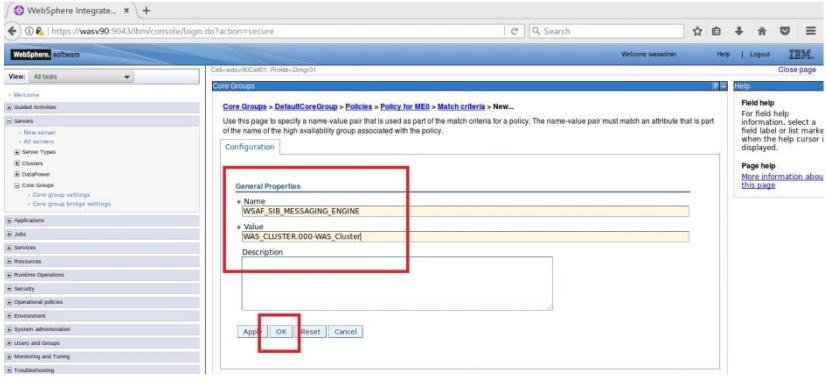
Step 36: Select “One of N policy” and click “Next”.



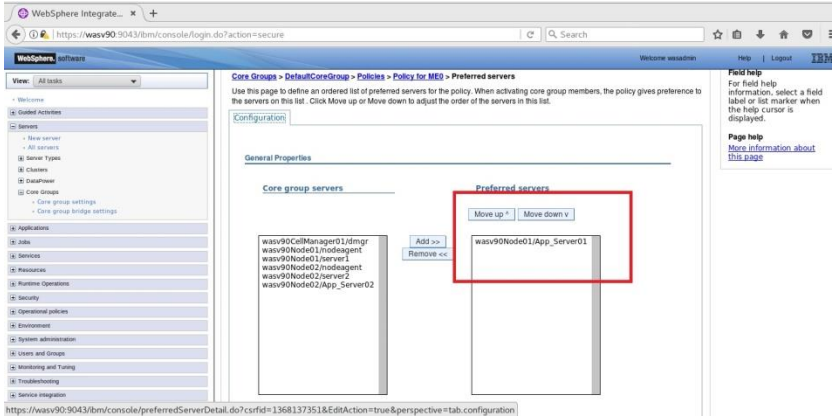
Step 37: Enter “Policy for ME0” as name, set “Is alive timer” to “30” and mark “Failback” and “Preferred servers only” then click “OK”.



Step 38: Navigate to “Core Groups>DefaultCoreGroup>Policies>Policy for ME0>Match criteria>” and add “WSAF_SIB_MESSAGING_ENGINE” as name and “WAS_CLUSTER.000-WAS_Cluster> as value, then click “OK”.

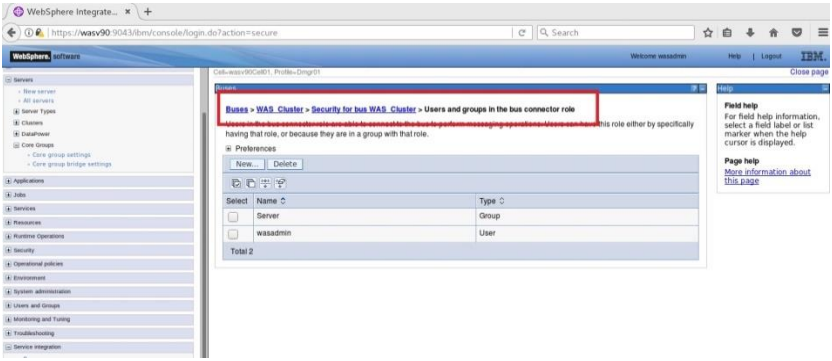


Step 39: Navigate to “Core Groups>DefaultCoreGroup>Policies>Policy for ME0>Preferred servers” and add the first member of the cluster, then “OK”.

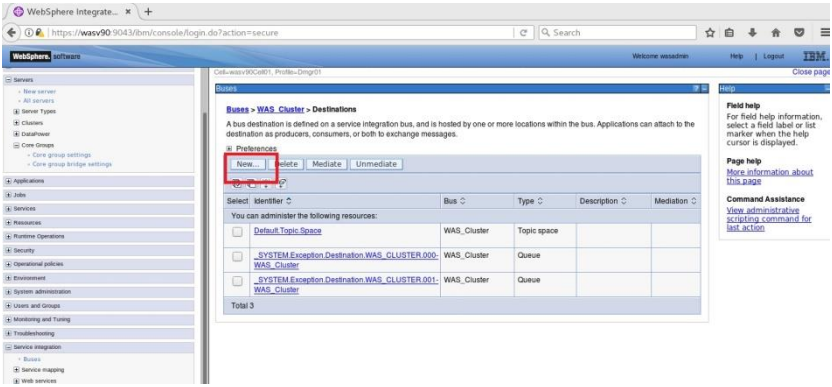


Step 40: Repeat the same steps from 35 to 39 for the messaging engine “ME1” and for the second member of the cluster.

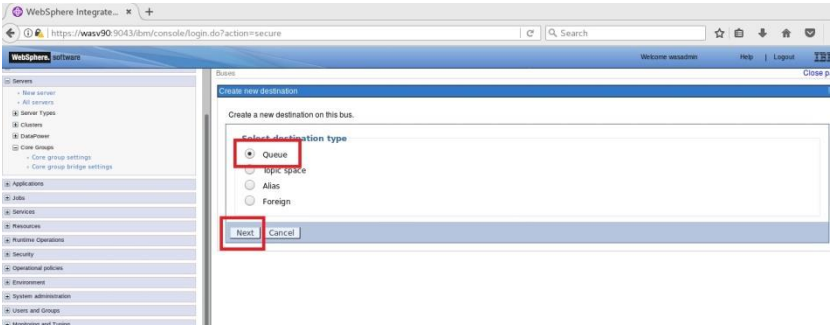
Step 41: Navigate to “System integration>Buses>WAS_Cluster>Security for bus WAS_Cluster>Users and groups in the bus connector role”and add administrative user “Ernesto” as name and “User” as type.



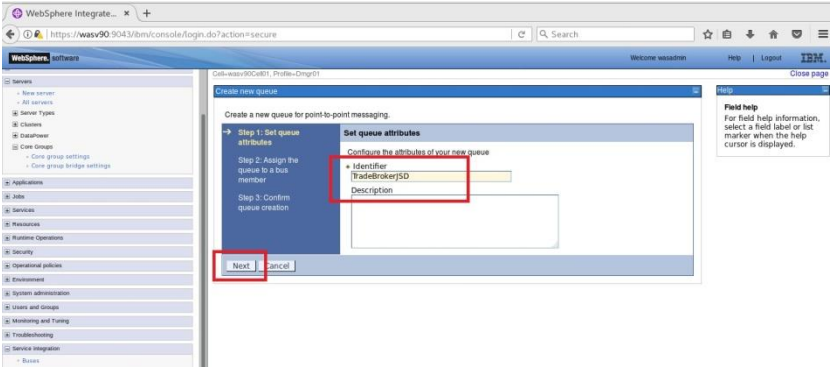
Step 42: Navigate to “Buses>WAS_Cluster>Destinations” and click “New”.



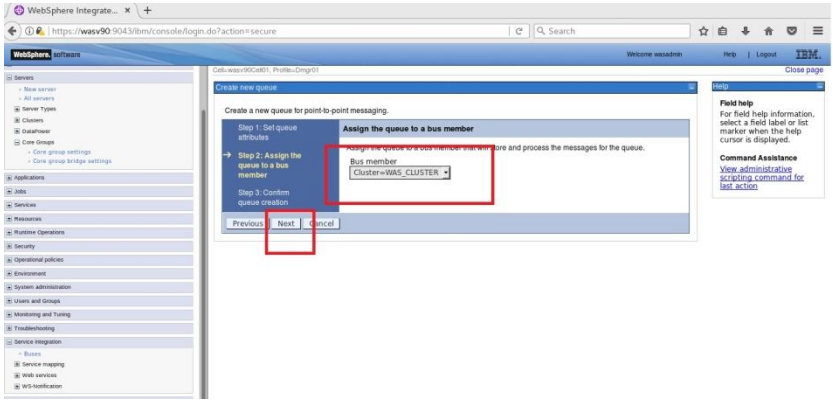
Step 43: Select “Queue” and then click “Next”.



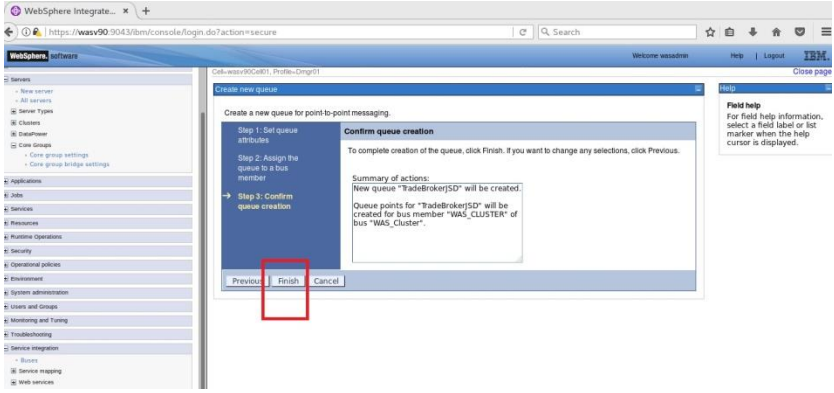
Step 44: Use “TradeBrokerJSD” as identifier then click “Next”.



Step 45: Select “Cluster=WAS_CLUSTER” as bus member and click “Next”.



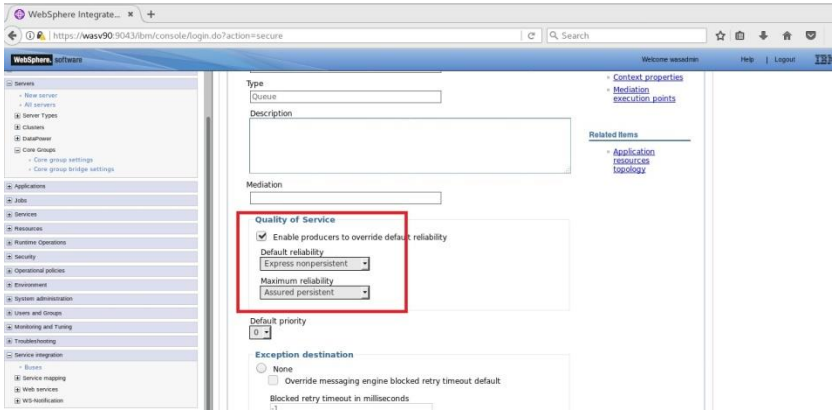
Step 46: Click “Finish”.



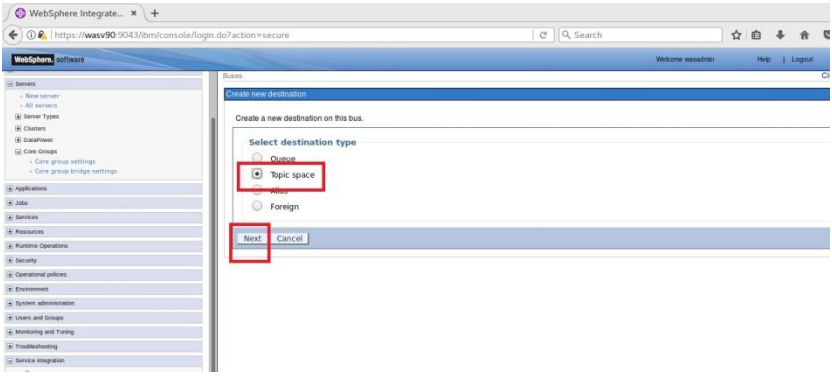
Step 47: Navigate to “Buses>WAS_Cluster>Destinations>TradeBrokerJSD” and change “Quality of Service” settings as follows.

Default reliability: Express nonpersistent

Maximum reliability: Assured persistent.

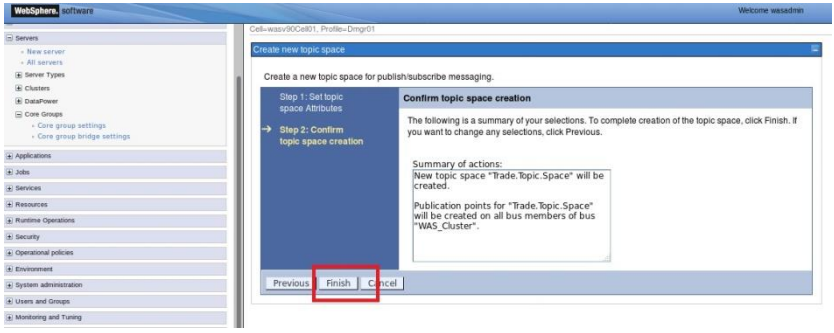


Step 48: Navigate to “Buses>WAS_Cluster>Destinations” and click “New

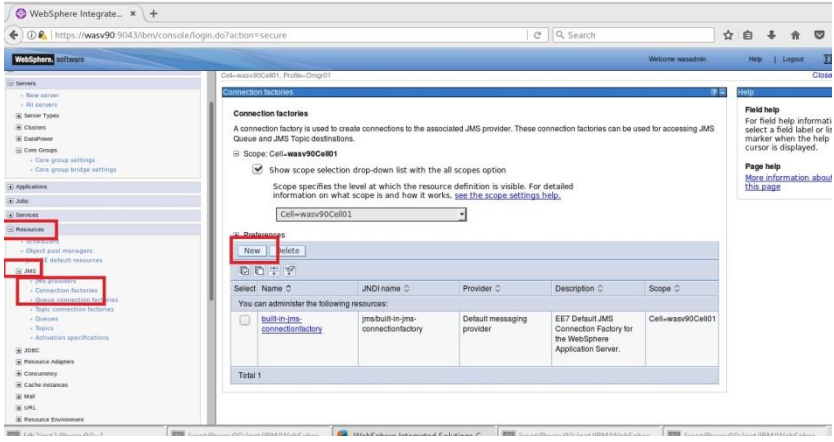


Step 49: Use “Trade.Topic.Space” as identifier and click “Next”.

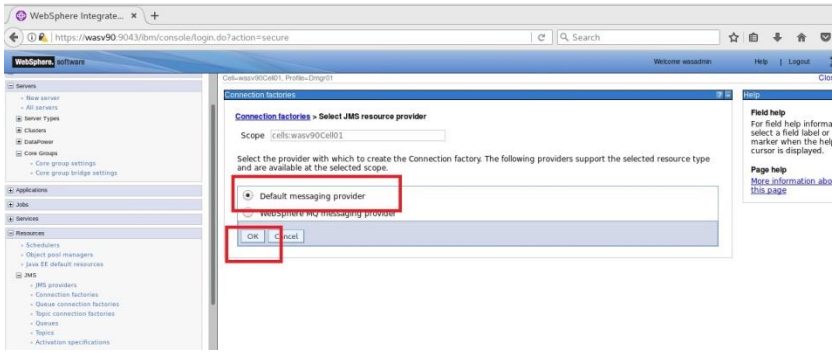
Step 50: Click “Finish” to complete.



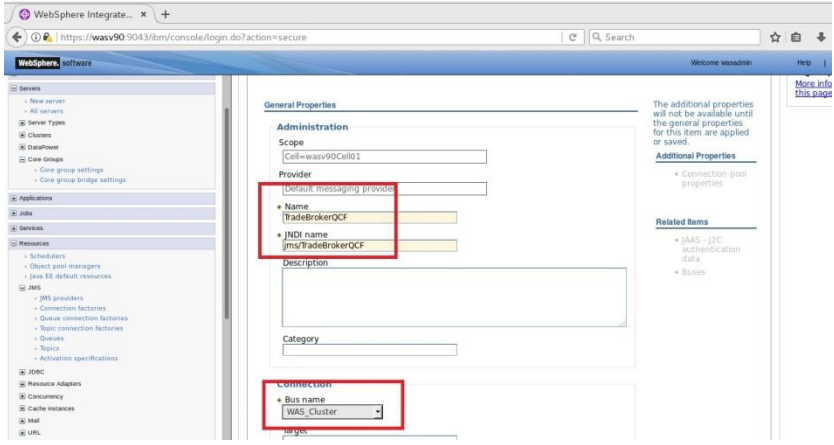
Step 51: Navigate to “Resources>JMS>Queue connection factories” and click “New”.



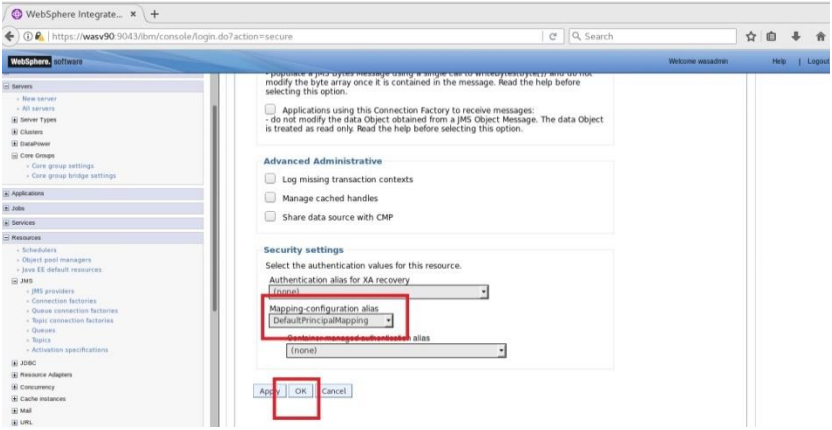
Step 52: Select “Default messaging provider” and click “OK”.



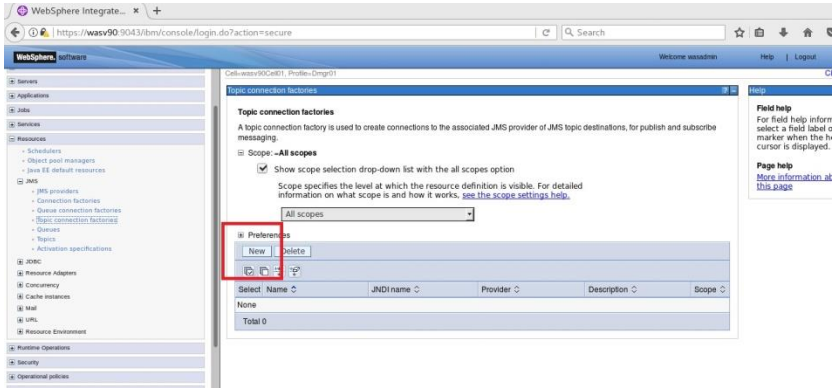
Step 53: Use “TradeBrokerQCF” as name, “jms/TradeBrokerQCF” as JNDI name, select “WAS_Cluster” as bus member.



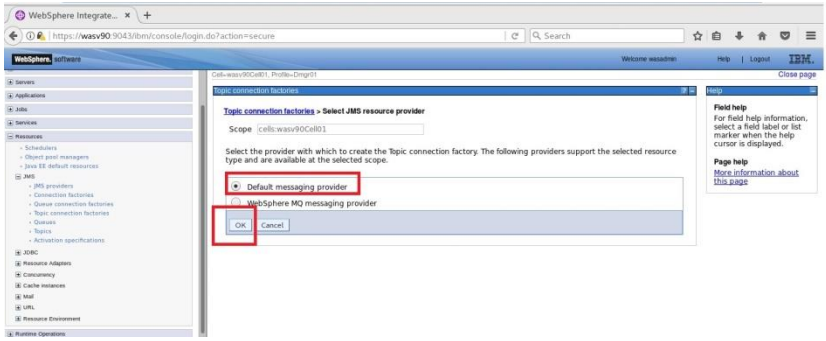
Select “DefaultPrincipalMapping” as “Mapping-configuration alias” and click “OK”.



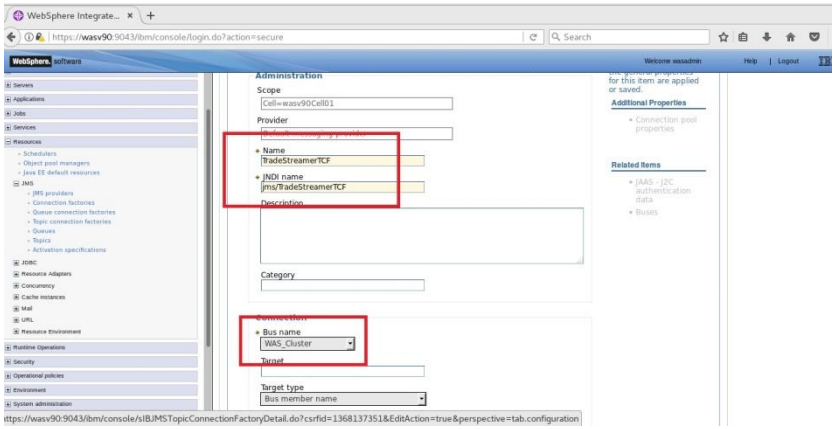
Step 54: Navigate to “Resources>JMS>Topic connection factories” and click “New”.



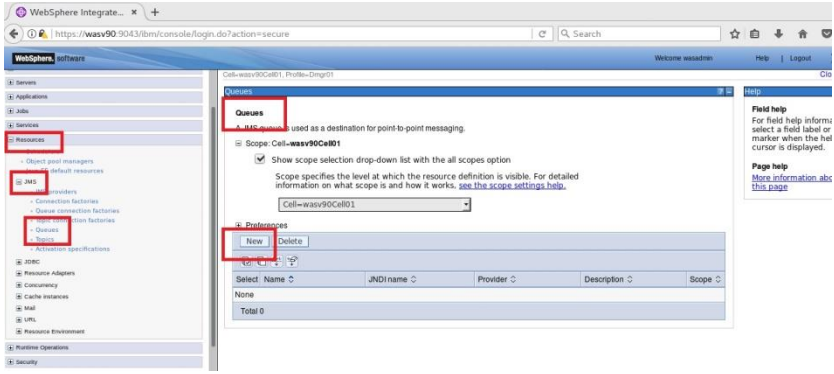
Step 55: Select “Default messaging provider” and click “OK”.



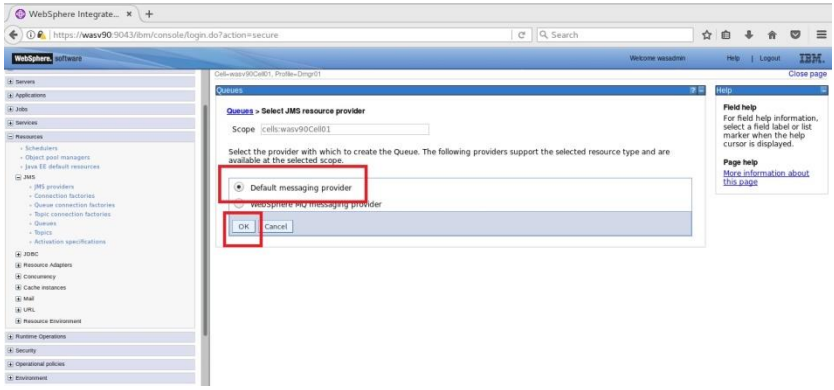
Step 56: Use “TradeStreamerTCF” as name, “jms/TradeStreamerTCF” as JDNI name and select “WAS_Cluster” as bus name and click “OK”.



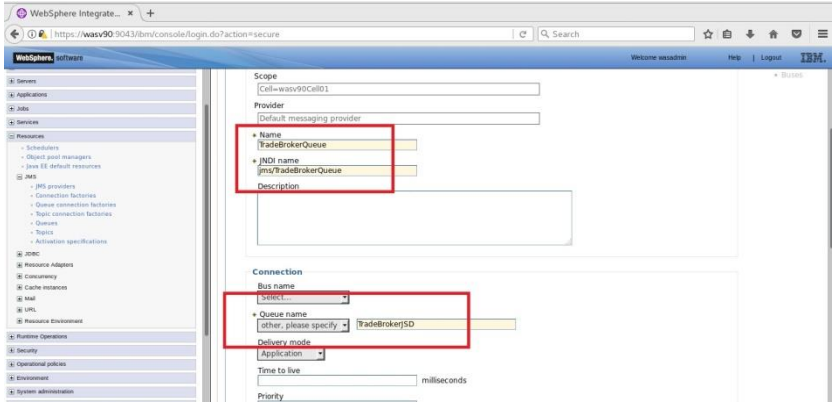
Step 57: Navigate to “Resources>JMS>Queues” and click “New”.



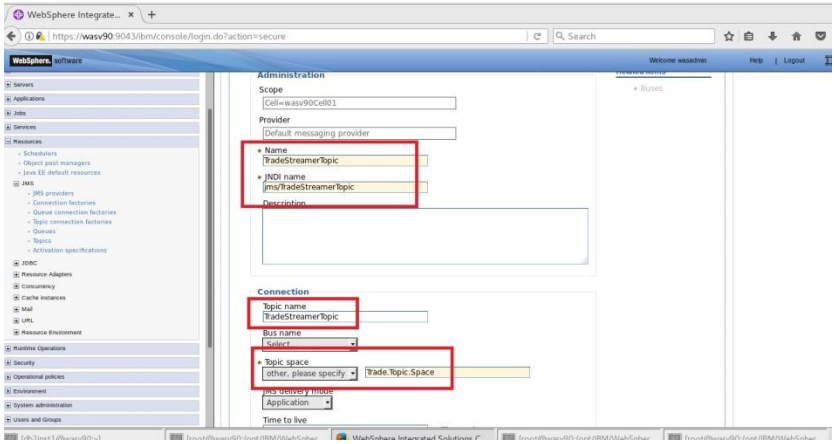
Step 58: Select “Default messaging provider” and click “OK”.



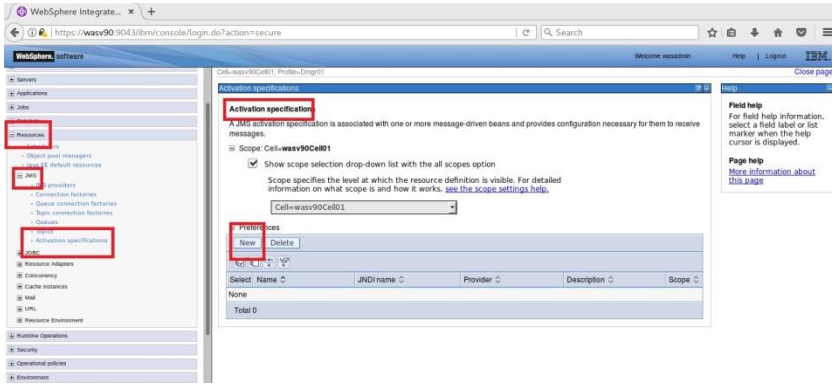
Step 59: Use “TradeBrokerQueue” as name “jms/TradeBrokerQueue” as JNDI name, select “other, please specify” and enter “TradeBrokerJSD” as queue name and click “OK”.



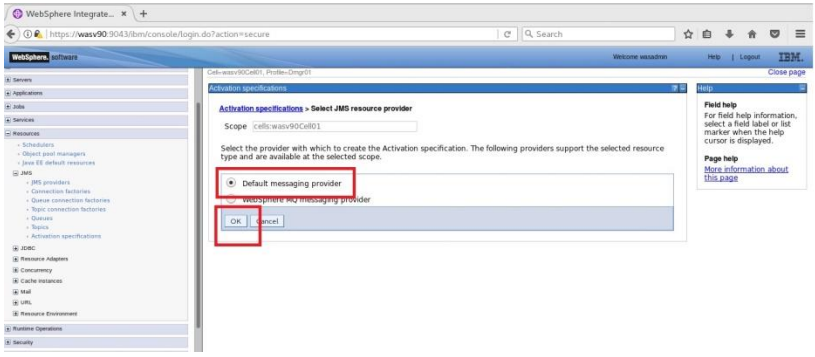
Step 60: Navigate to “Resources>JMS>Topics”, click “New”. Use “TradeStreamerTopic” as name, “jms/TradeStreamerTopic”, “TradeStreamerTopic” as topic name, and specify “Trade.Topic.Space” as topic space and click “OK”.



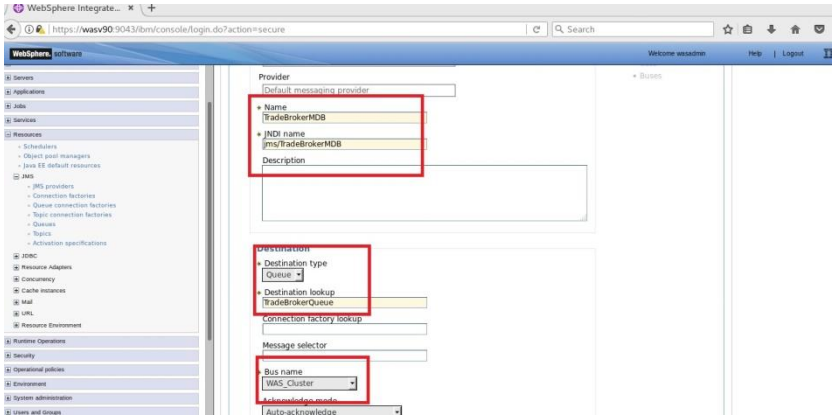
Step 61: Navigate to “Resources>JMS>Activation specification” and click “New”.



Step 62: Select “Default messaging provider” and click “OK”.

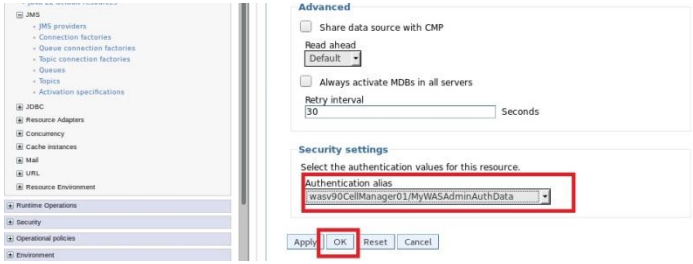


Step 63: Use “TradeBrokerMDB”, “eis/TradeBrokerMDB” as JNDI name, select “Queue” as destination and “jms/TradeBrokerQueue” as destination JNDI name and “WAS_Cluster” as bus name.

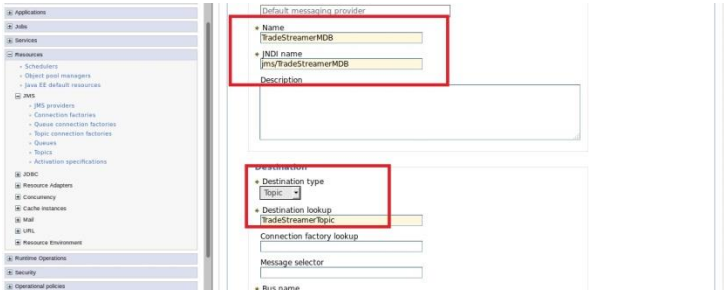


Step 64: Use “Subscription Durability” as follows.

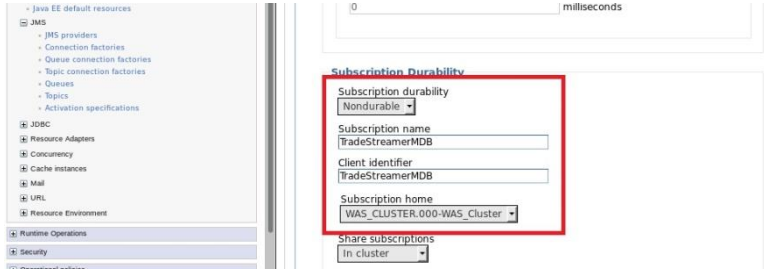
Step 65: Use “MyWASAdminAuthData” as authentication alias and click “OK”.

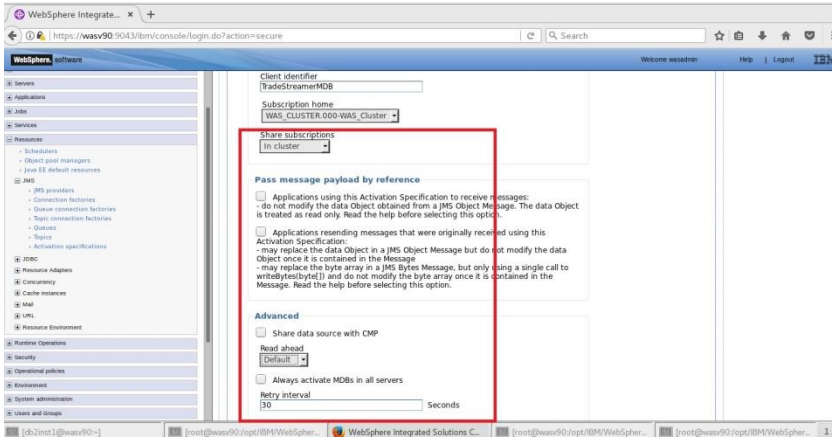


Step 66: Navigate to “Resources>JMS>Activation specification”, click “New”, Select “Default messaging provider” then click “OK”. Use “TradeStreamerMDB” as name, “eis/TradeStreamerMDB” as JNDI name, select “Topic” as destination and “jms/TradeStreamerTopic” as destination JNDI name, select “WAS_Cluster” as bus name.



Step 67: Configure “Subscription Durability”, “Advanced” and “Security settings” like in the images below and click “OK”.

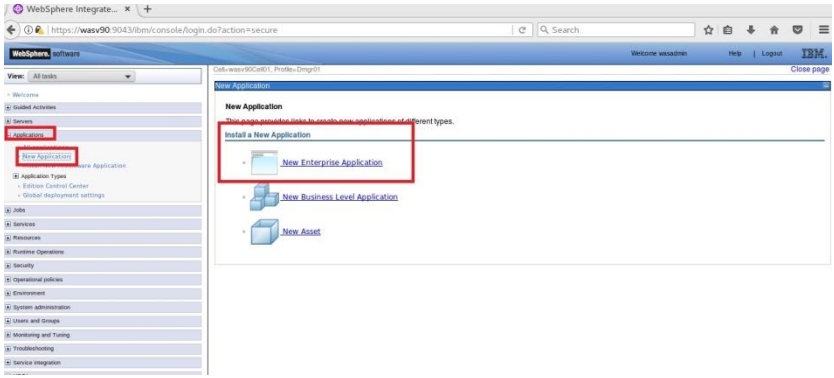




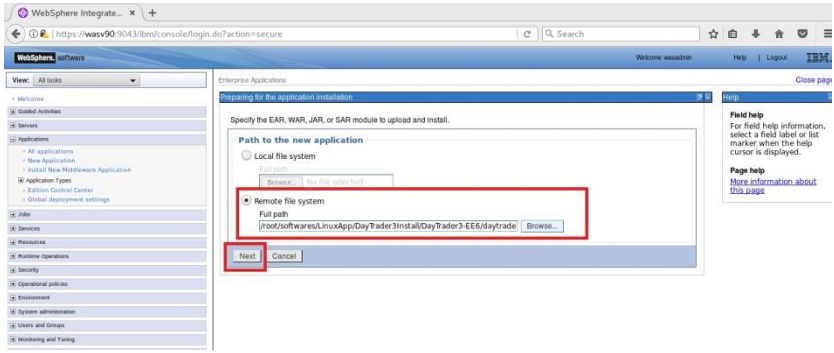
Task 1 is complete!

Task 2: Deploy a sample application

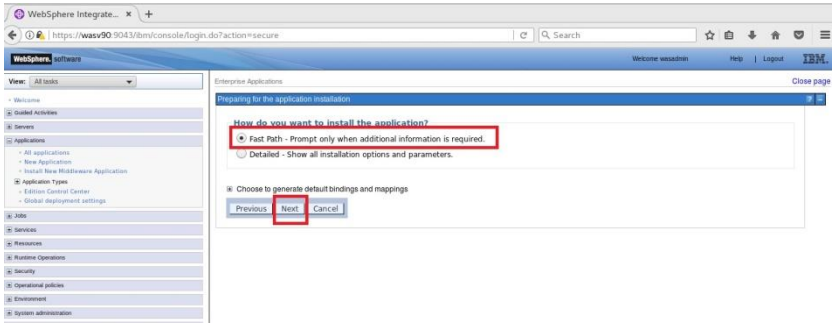
Step 1: Navigate to “Applications>New Application” and click on “New Enterprise Application”.



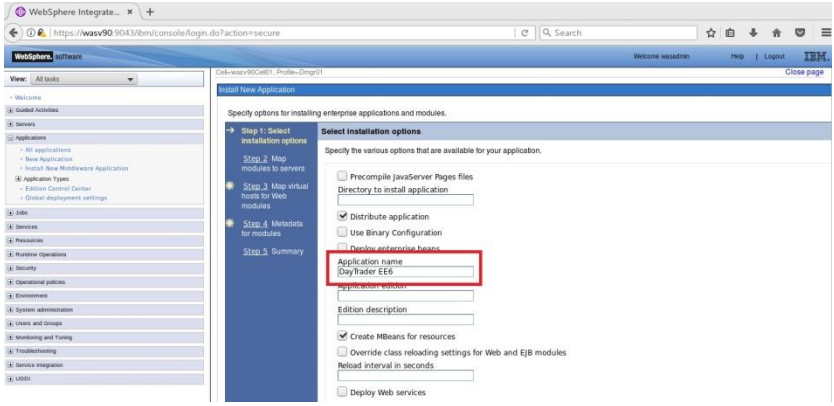
Step 2: Locate the ear file on the local file system and click “Next”.



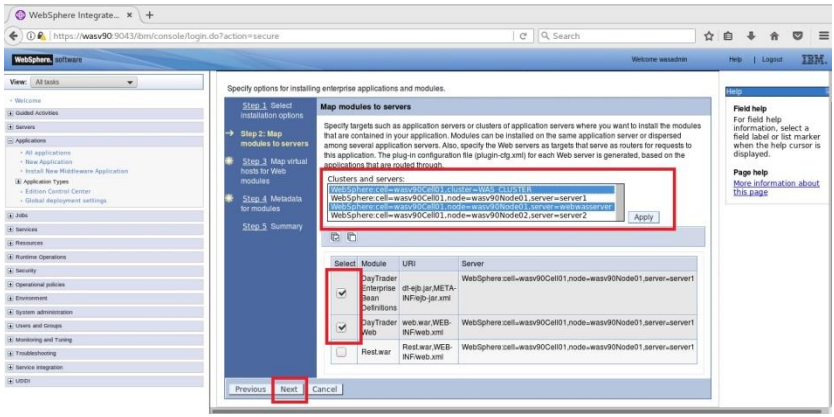
Step 3: Select “Fast Path” and click “Next”.



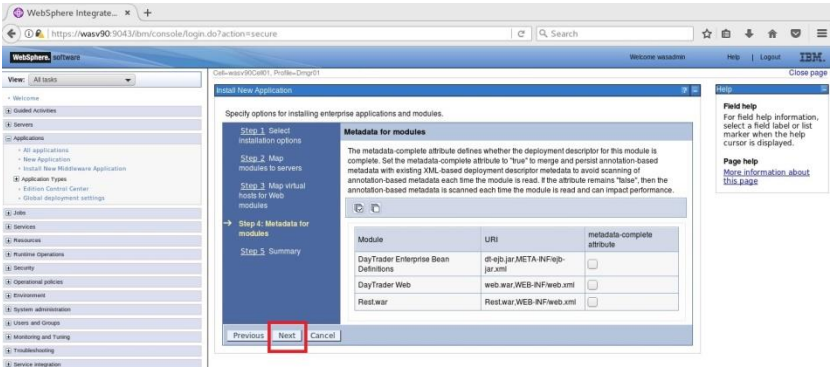
Step 4: Change application name to “DayTrader2-EE6” and click “Next”.



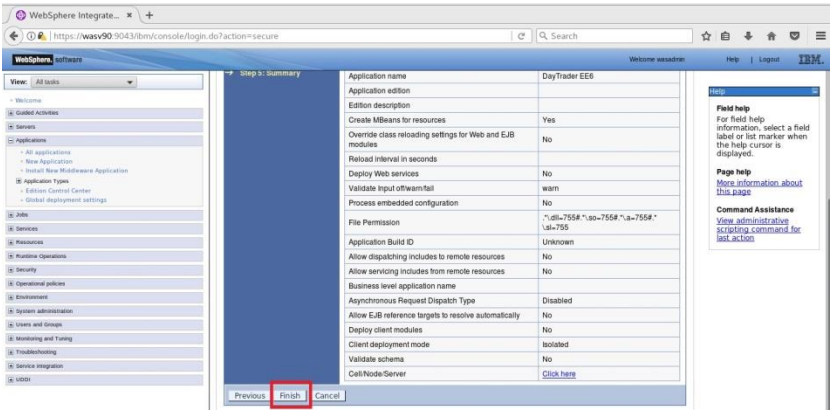
Step 5: Select cluster and the web server to map both of the modules of the application and click “Apply”, then click “Next”.



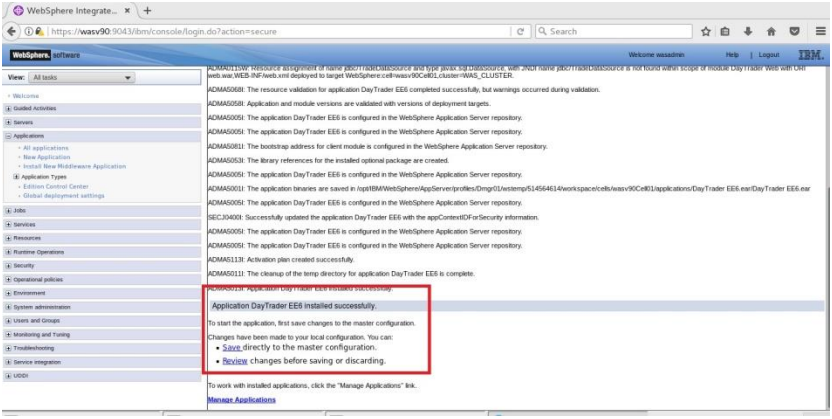
Step 6: Click “Next”.



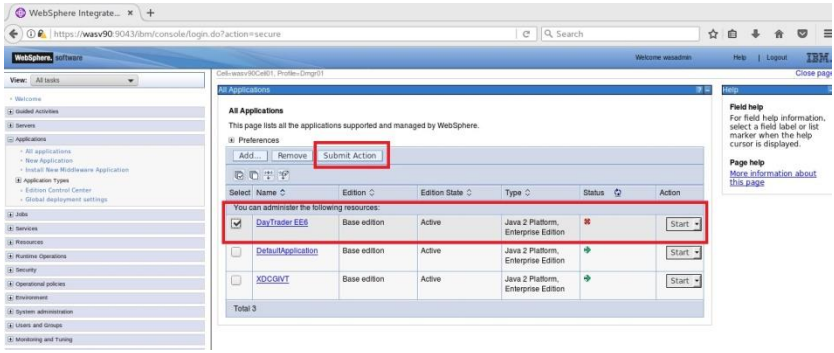
Step 7: Click “Finish” to start deployment.



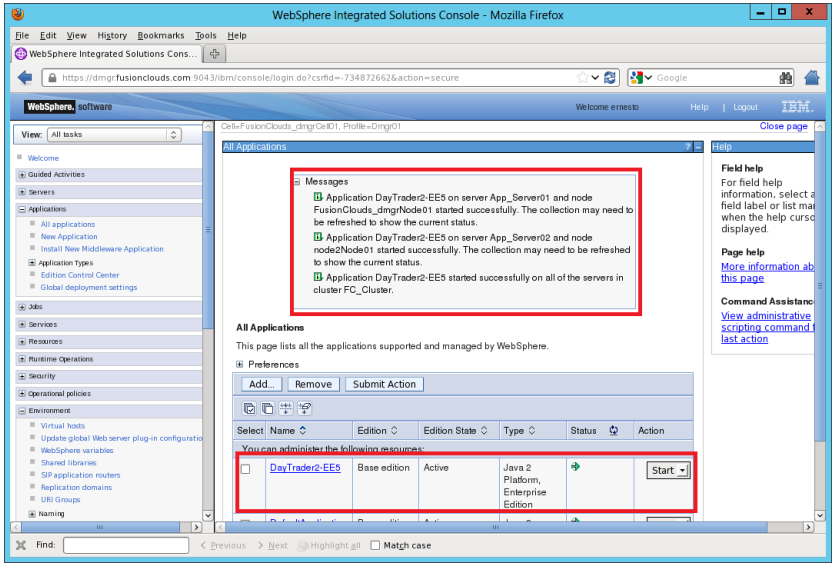
Step 8: You should see the success message, then click “Save” to write changes to the master configuration.



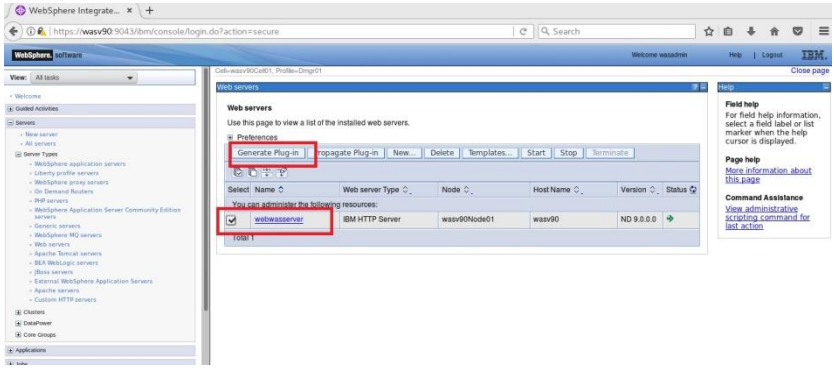
Step 9: Select the application and action as “Start” and click “Submit Action”.



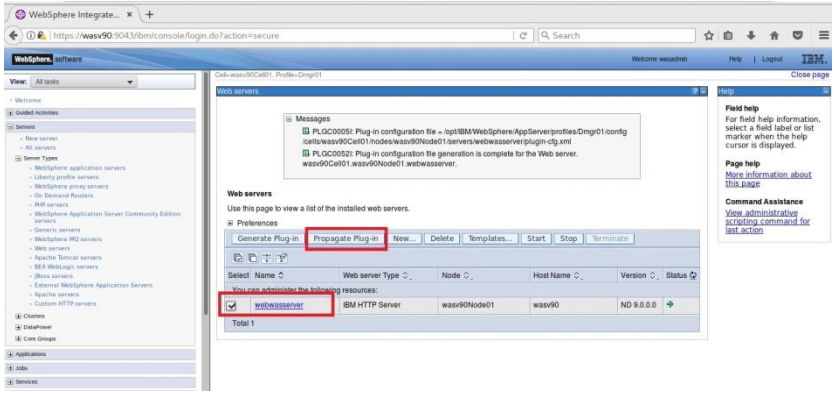
You should see the success message as follows.



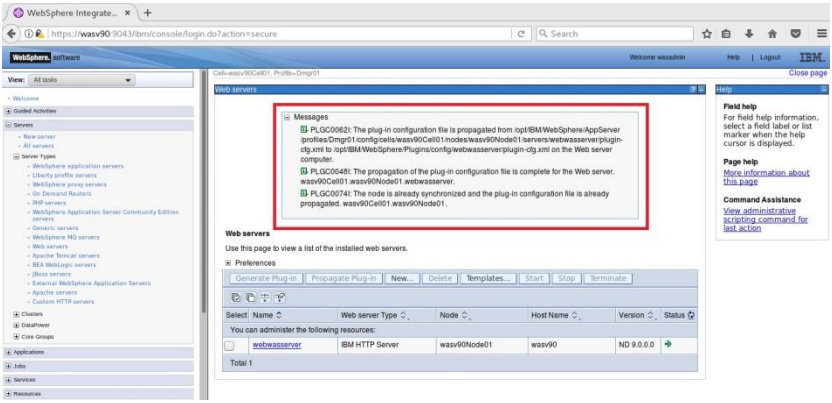
Step 10: Navigate to “Servers>Server Types>Web servers” and select the web server then click “Generate Plug-in”.



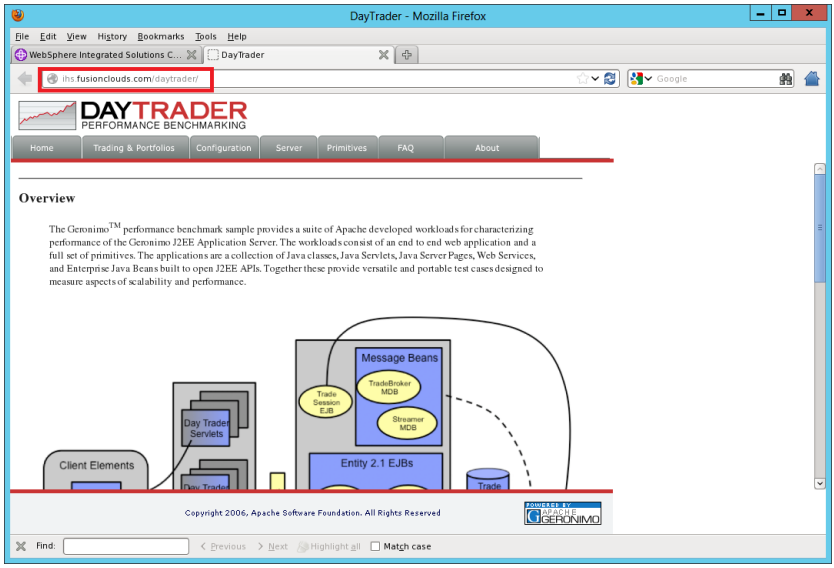
Step 11: Click “Propagate Plug-in” to push new configuration to the web server.



Step 11: Check the message and make sure that new plug-in configuration is pushed to the web server.



Step 11: Navigate to your web server to check the application. (https://ihs.fusionclouds.com/daytrader).



Task 2 is complete!

SUMMARY

WebSphere Application Server supports three different types of applications to be deployed that are enterprise applications, business level applications and assets. You can deploy those applications from your system or you can send the files to the system that hosts deployment manager. Before installing the application, you need to setup the environment according to the needs of application like JMS and JDBC settings. In order to access the application through web server, you have to make sure that you mapped the application modules to the web server, generate and propagate the plug-in configuration to the web server.

REFERENCES

- http://publib.boulder.ibm.com/infocenter/iadthelp/v6r0/index.jsp?topic=/com.ibm.etools.j2c.ims.doc/tasks/timsj2cwasinstear.html
- http://pic.dhe.ibm.com/infocenter/clmhelp/v3r0/index.jsp?topic=%2Fcom.ibm.jazz.install.doc%2Ftopics%2Fc_deploying_was.html
- https://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2011-06-08+10%3A34%3A22.216702R&S_TACT=&S_CMP=

INDEX