FCUBS JMS Configuration Using Websphere Default Messaging Provider

Oracle FLEXCUBE Universal Banking

Release 14.1.0.0.0

[May] [2018]

Table of Contents

1. Purpose

The purpose of this document is to explain the steps required for JMS Configuration in cluster mode using WEBSPHERE DEFAULT MESSAGING PROVIDER for Websphere 8.5.5

2. Introduction

The default messaging provider is installed and runs as part of WebSphere Application Server, and needs no further administration. WebSphere administrative console is used to configure JMS resources for applications and can manage messages and subscriptions associated with JMS destinations.

The default messaging provider is the Java™ Message Service (JMS) API implementation for messaging (connection factories, JMS destinations, and so on). The concrete destinations (queues and topic spaces) behind the default messaging provider interface are implemented in a service integration bus.

The default messaging provider is based on service integration technologies., this document deals with

- Service Bus Creation

  A service integration bus consists of one or more bus members. A bus member can be an application server or a cluster. Each bus member will have one (or possibly more in the case of clusters) messaging engine that manages connections to the bus and messages.

- JMS connection factories and service integration
  A JMS connection factory is used to create connections to JMS resources on a service integration bus.

- JMS queue resources and service integration
  Creation of JMS queue resources provided by the default messaging provider for JMS point-to-point messaging and supported by a service integration bus.

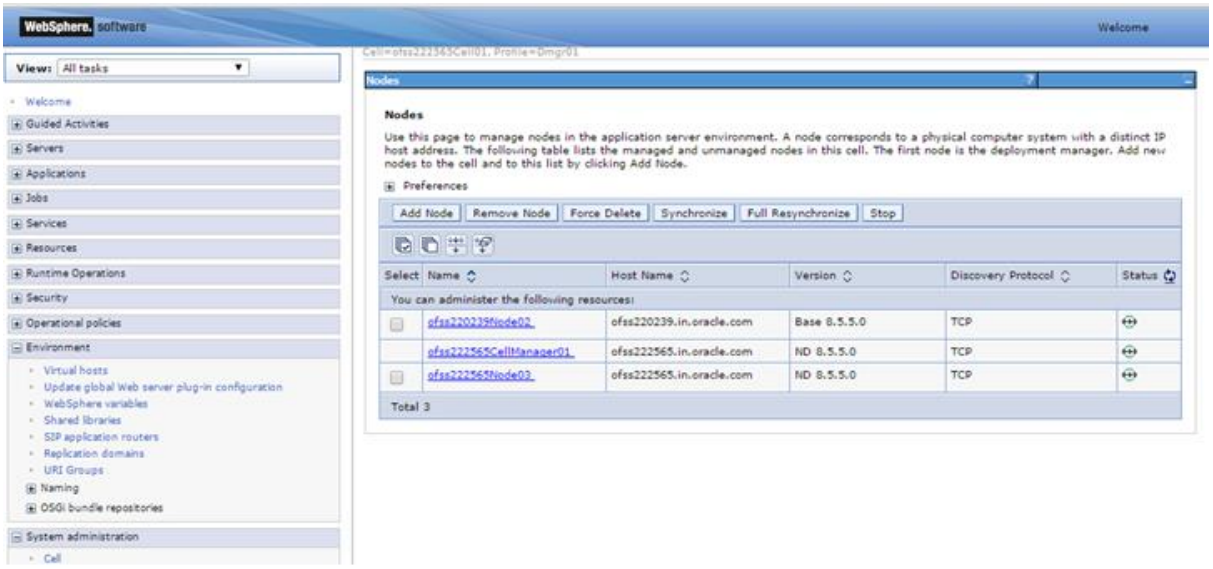
- Application access to JMS resources
  Describes the application access to Java Message Service (JMS) resources provided by the default messaging provider.

3. Pre-Requisites

The document assumes that the below are created before proceeding JMS creation.

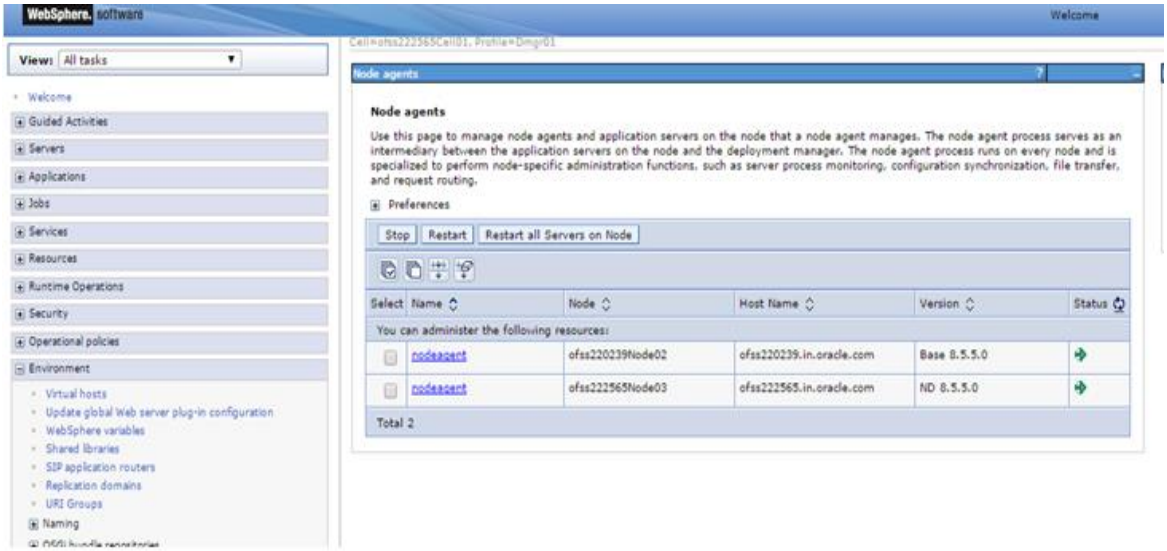
3.1 Nodes

2 nodes are created

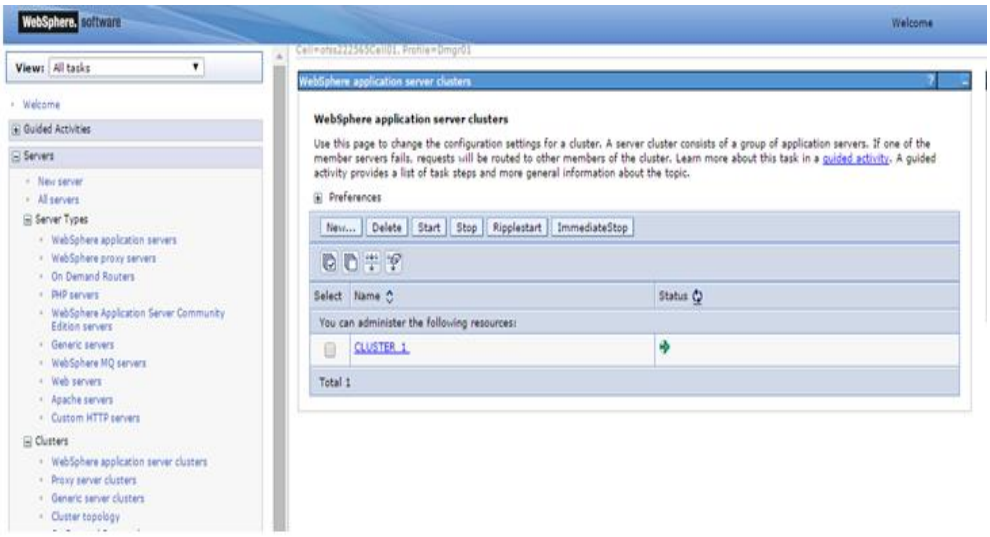


3.2 Node Agents

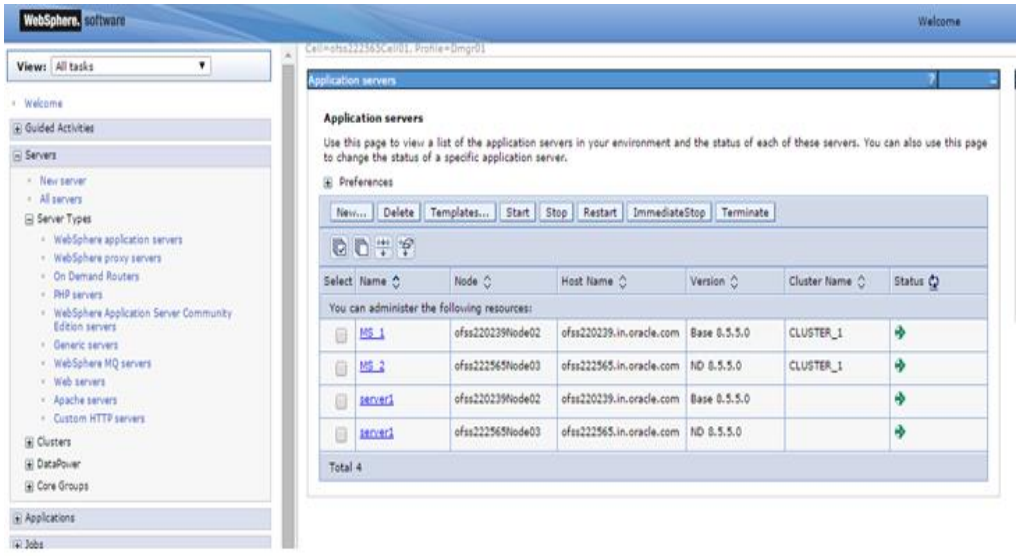
Both the Node Agents are started.



3.3 Cluster

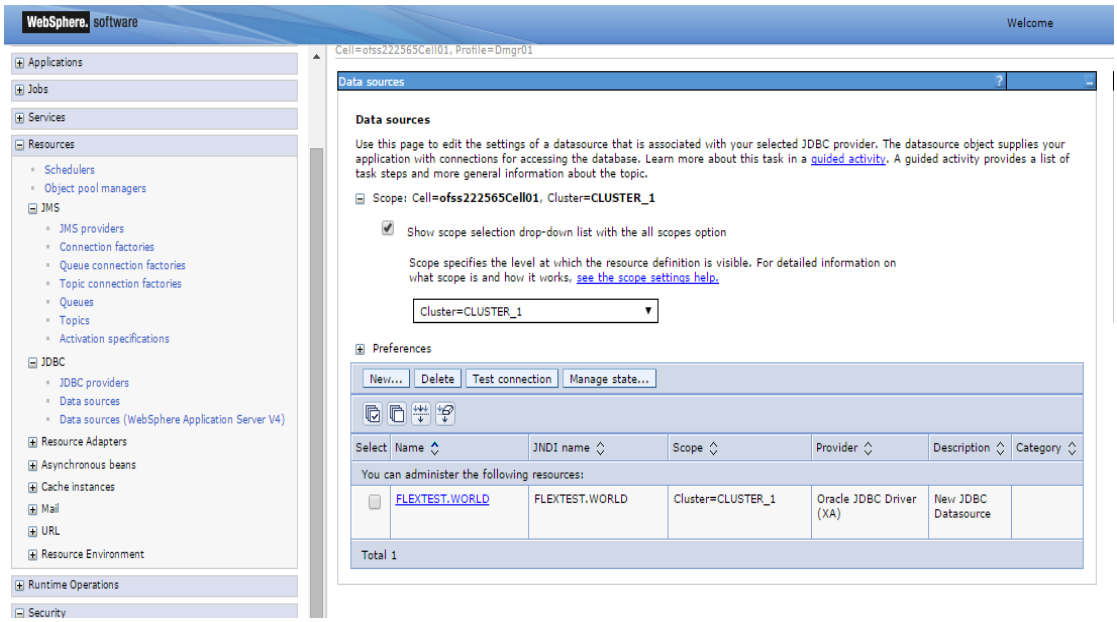


3.4 Managed Servers



3.5 DataSource

Ensure that DataSource required for the MDB ear is created with Target as Cluster_1



3.6 Shared Folder

Shared folders for File Store Creation are required and this folder should be accessible across both the servers (eg, NFS mount). For fail over of messaging engines to another, all servers in cluster require a separate folder. If there are 4 Managed Servers in the clusters then 4 separate folders are required.

Eg,

/scratch/MessageStore/JMS_1/

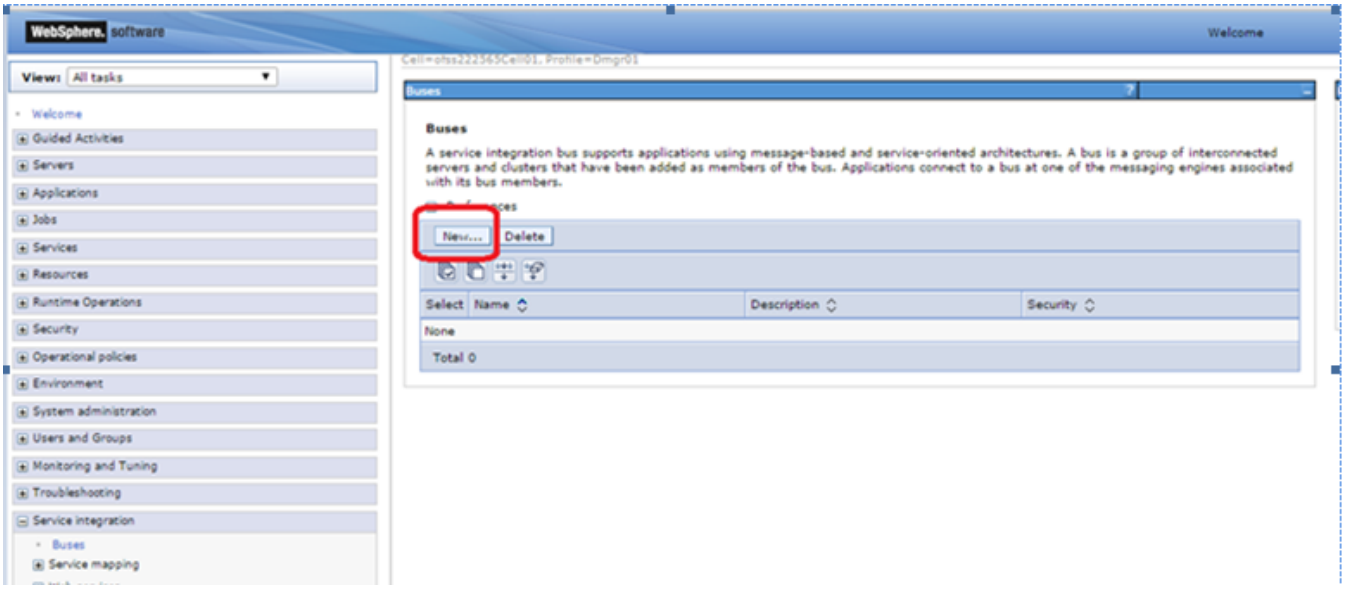
/scratch/MessageStore/JMS_2/

/scratch/ MessageStore /JMS_3/

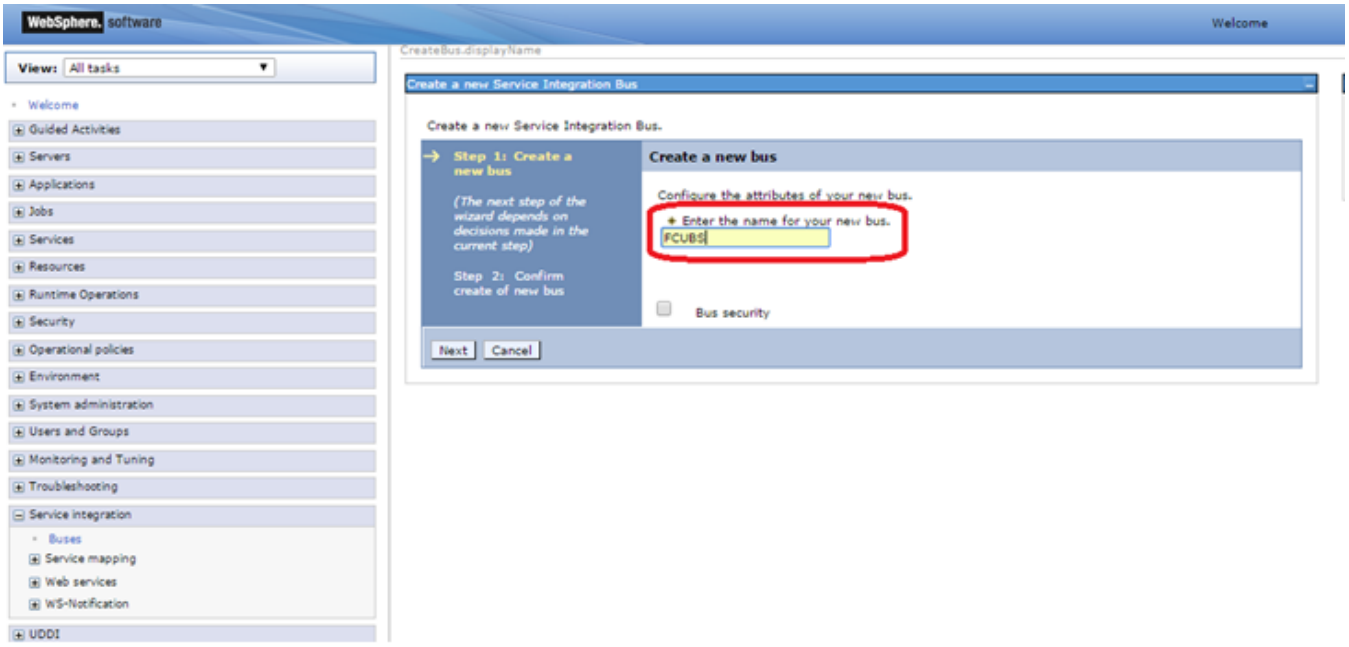
4. JMS Configuration

4.1 Service Integration Bus Creation

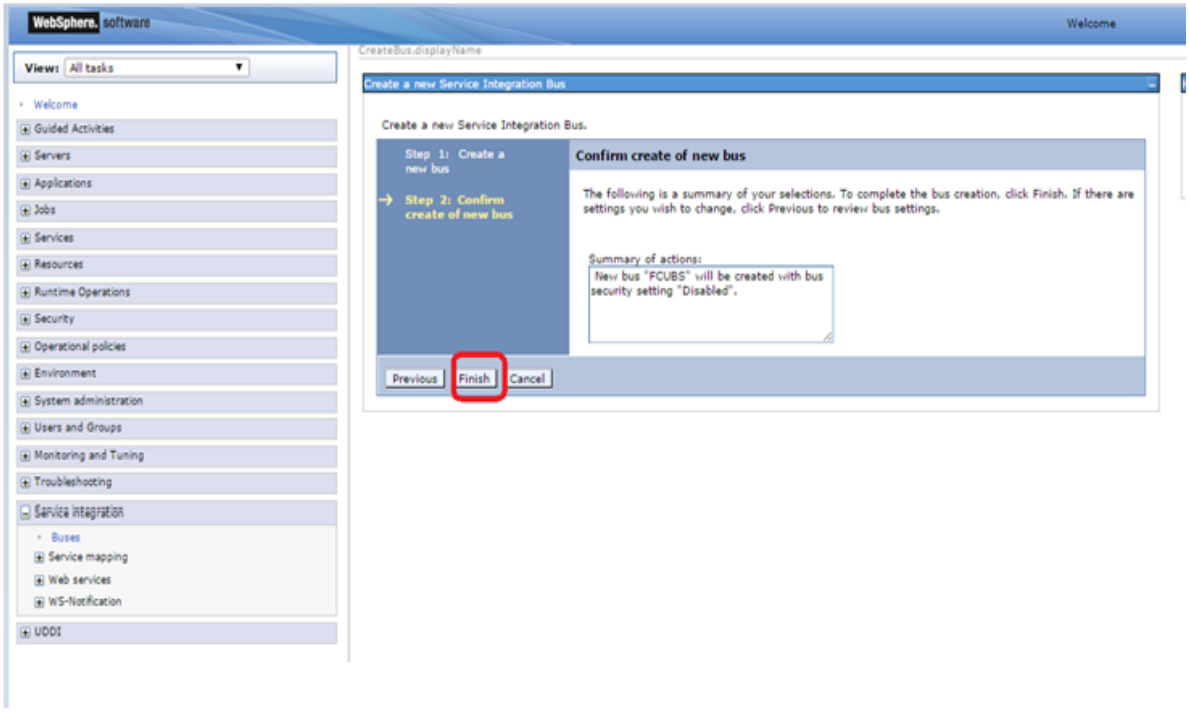
1) Navigate to Service Integration > Buses > Click on New



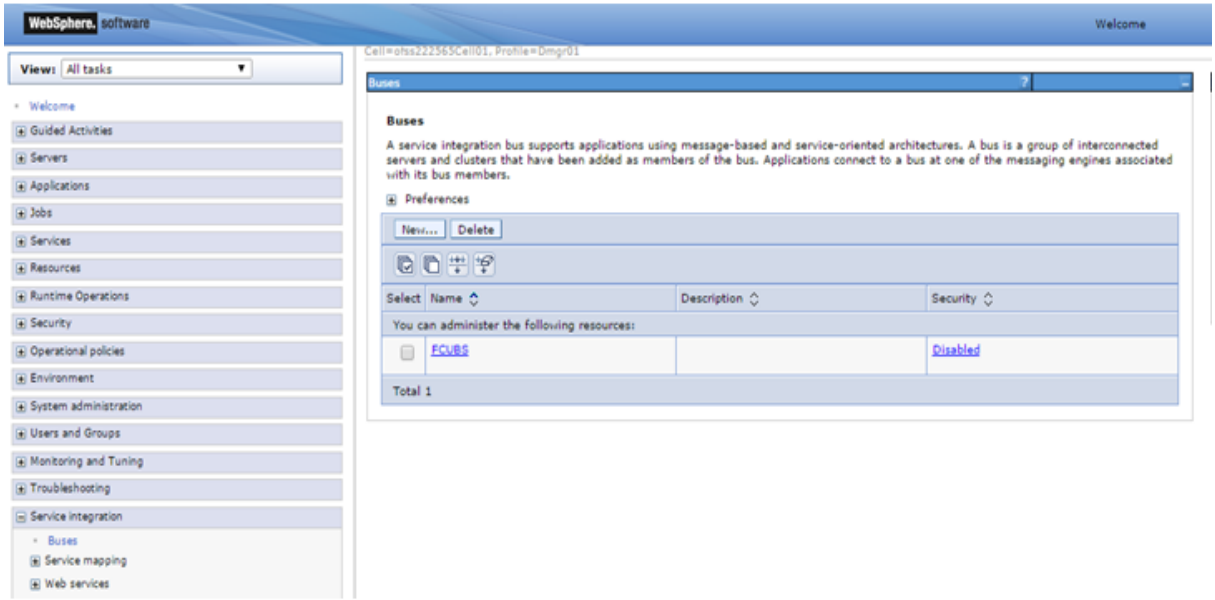
2) Enter Name for the new Bus, Uncheck “Bus Security” if security is not enabled during FCUBS property file build and click on Next



3) Click on Finish

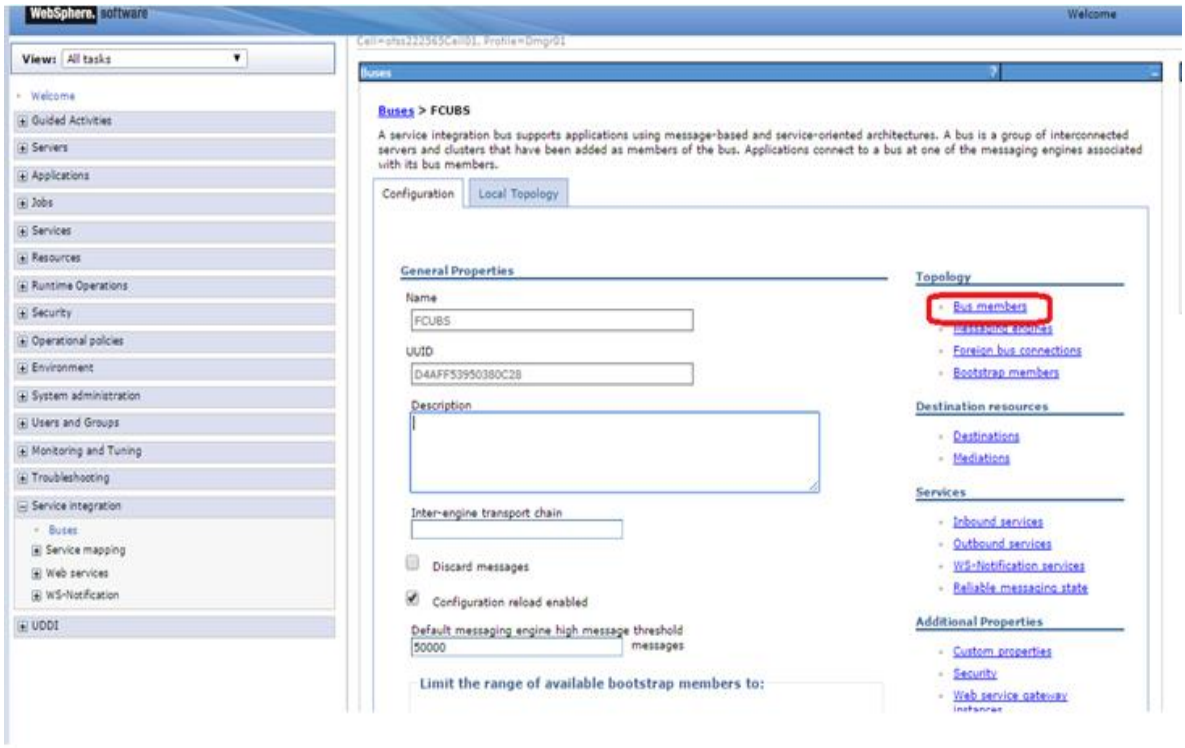


4) New Bus FCUBS is created

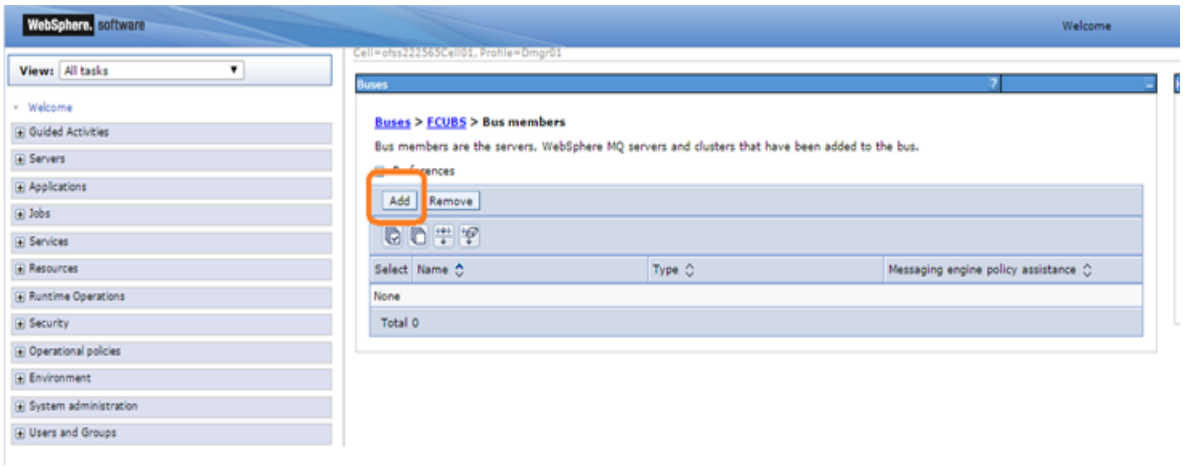


4.2 Bus Member (File Store Creation)

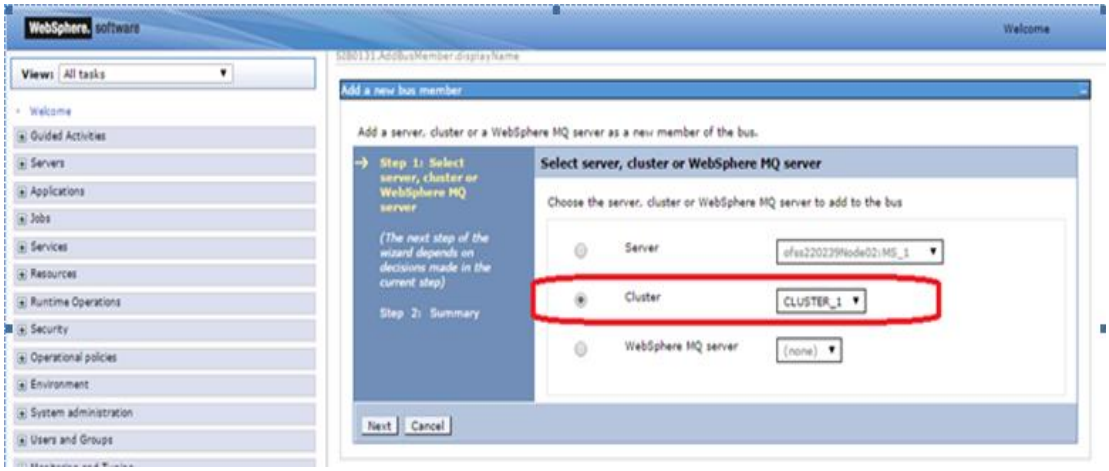
1) Navigate to Service Integration > Buses > Click on FCUBS(new bus Created) > Click on Bus Member under Topology



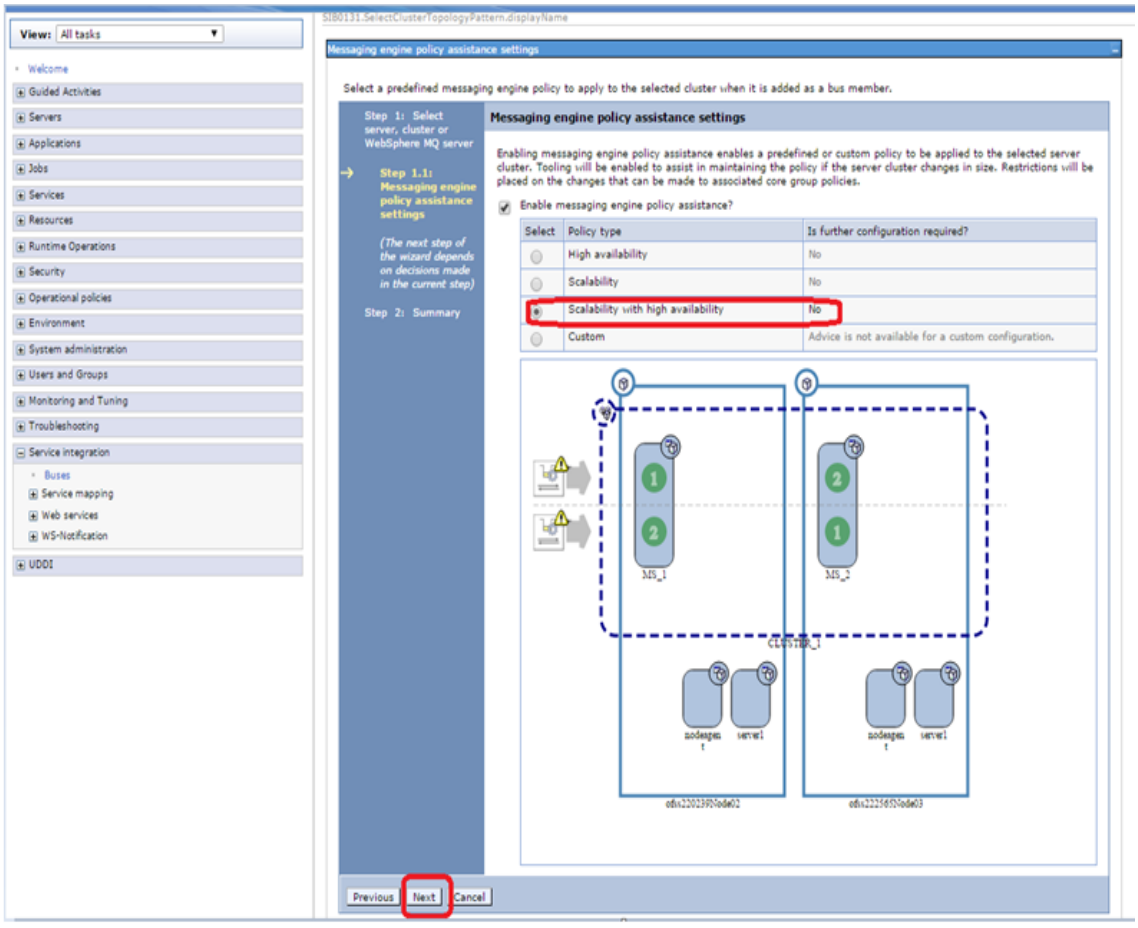
2) Click On Add



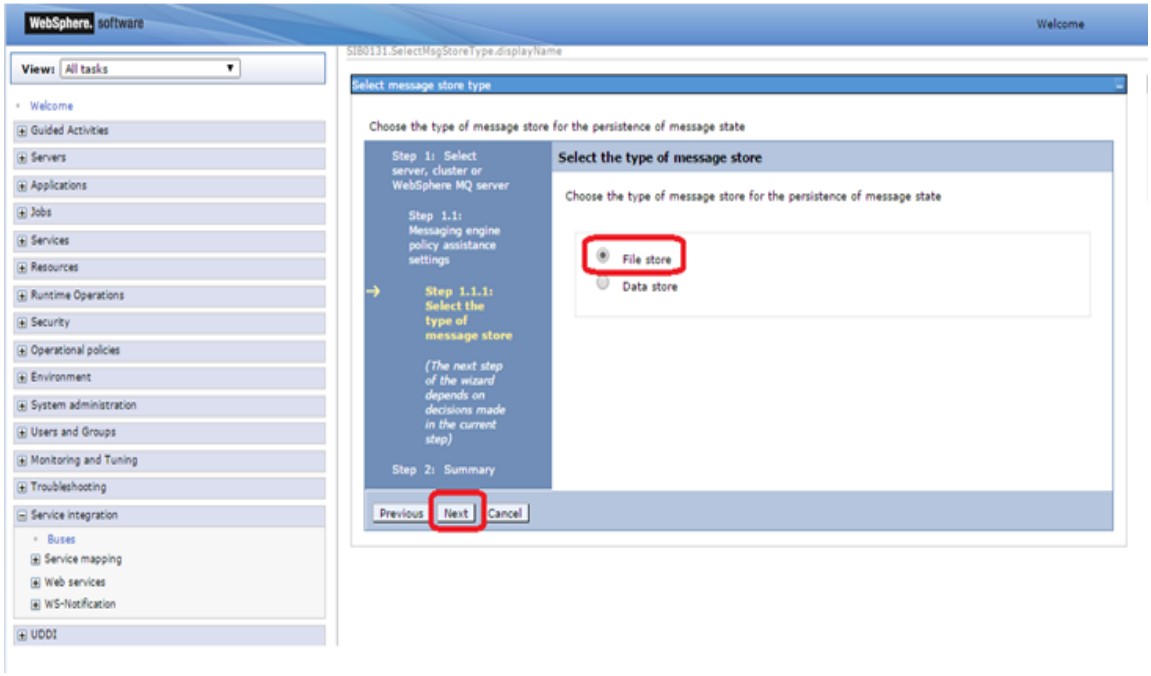
3) Select Cluster and Click on Next



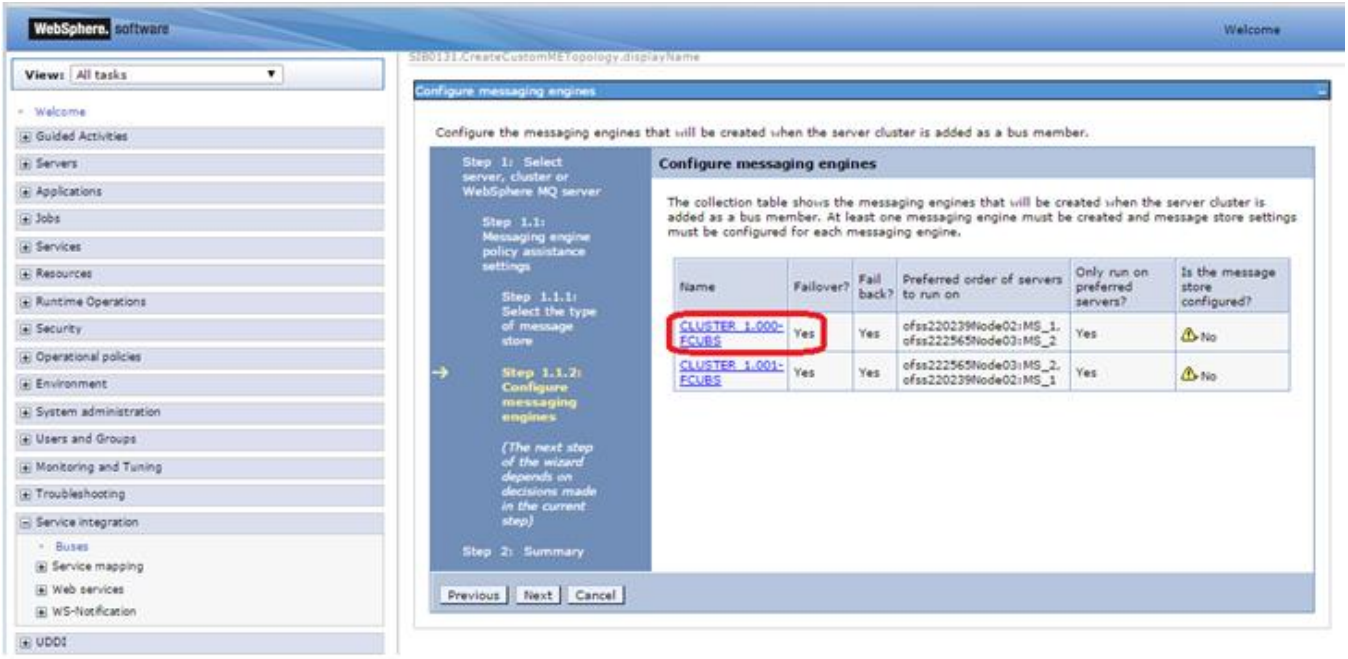
4) Select Scalablity and High Availablity Policy Type and Click on Next.



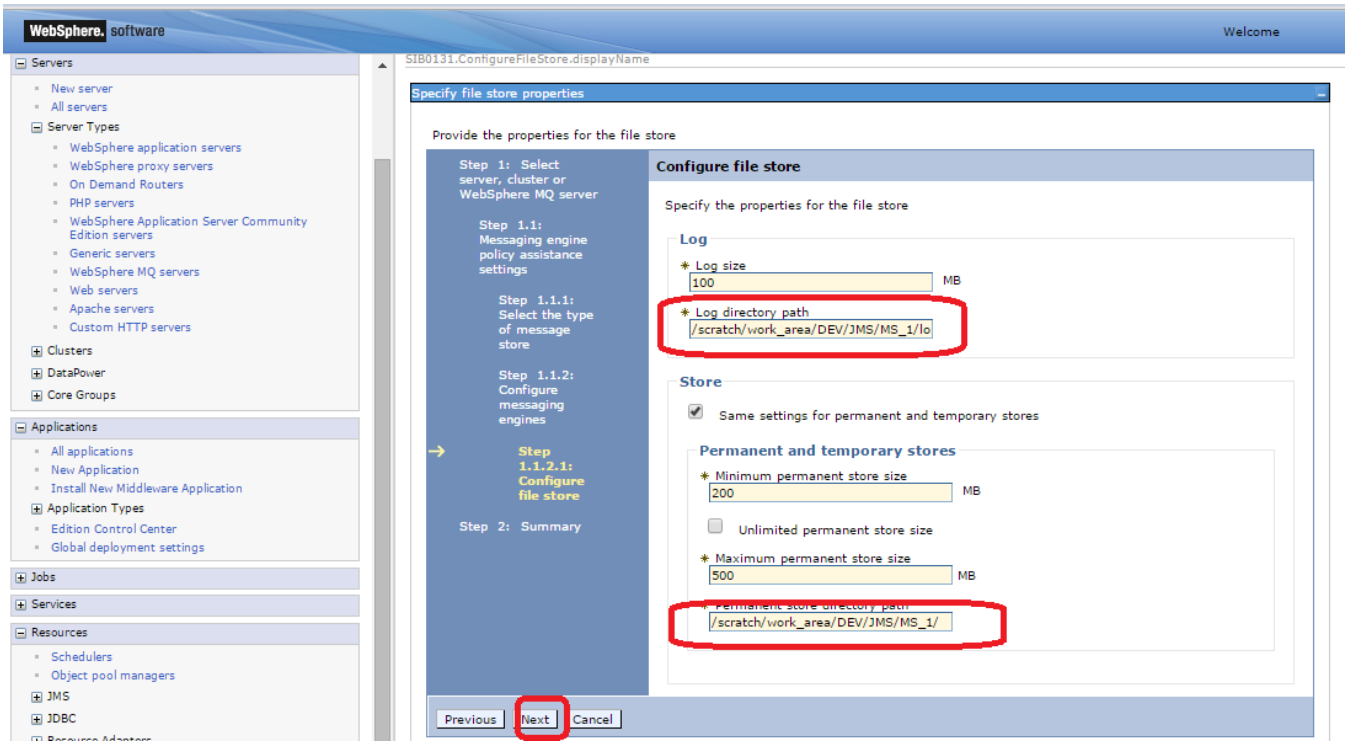
5) Select File Store and Click on Next



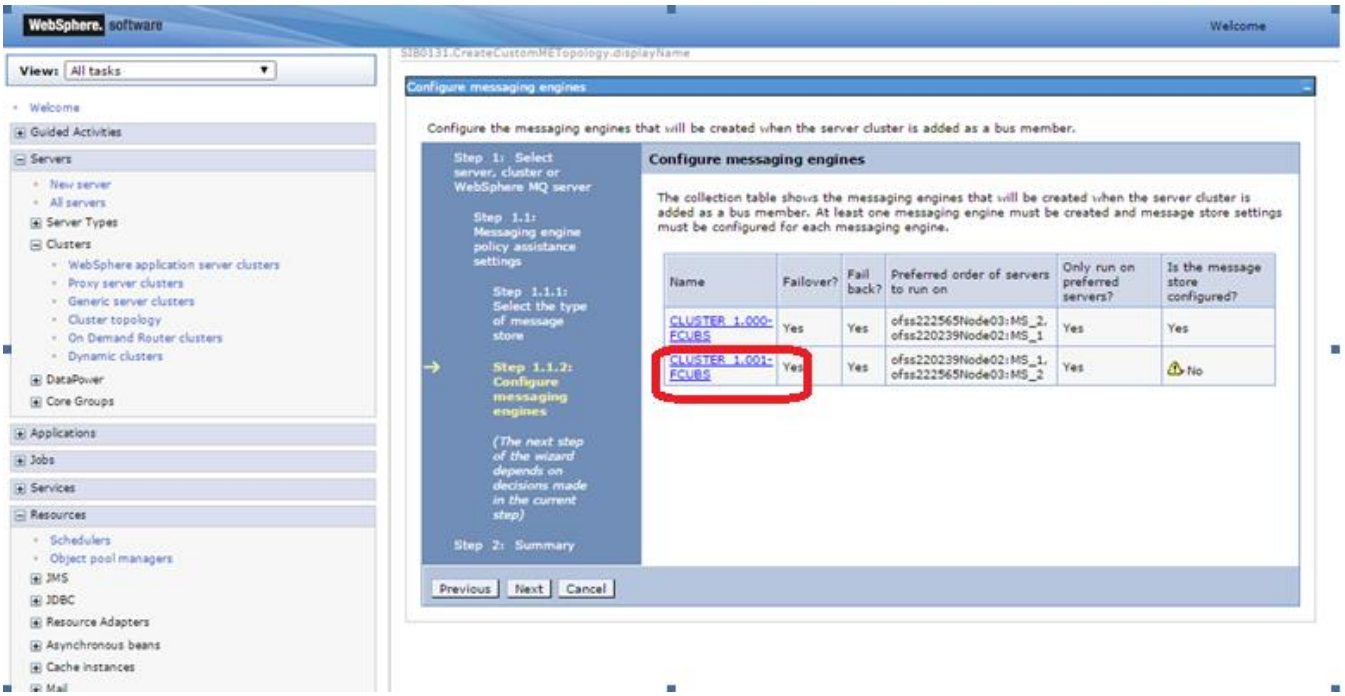
6) Select the Node 1 Message Engine



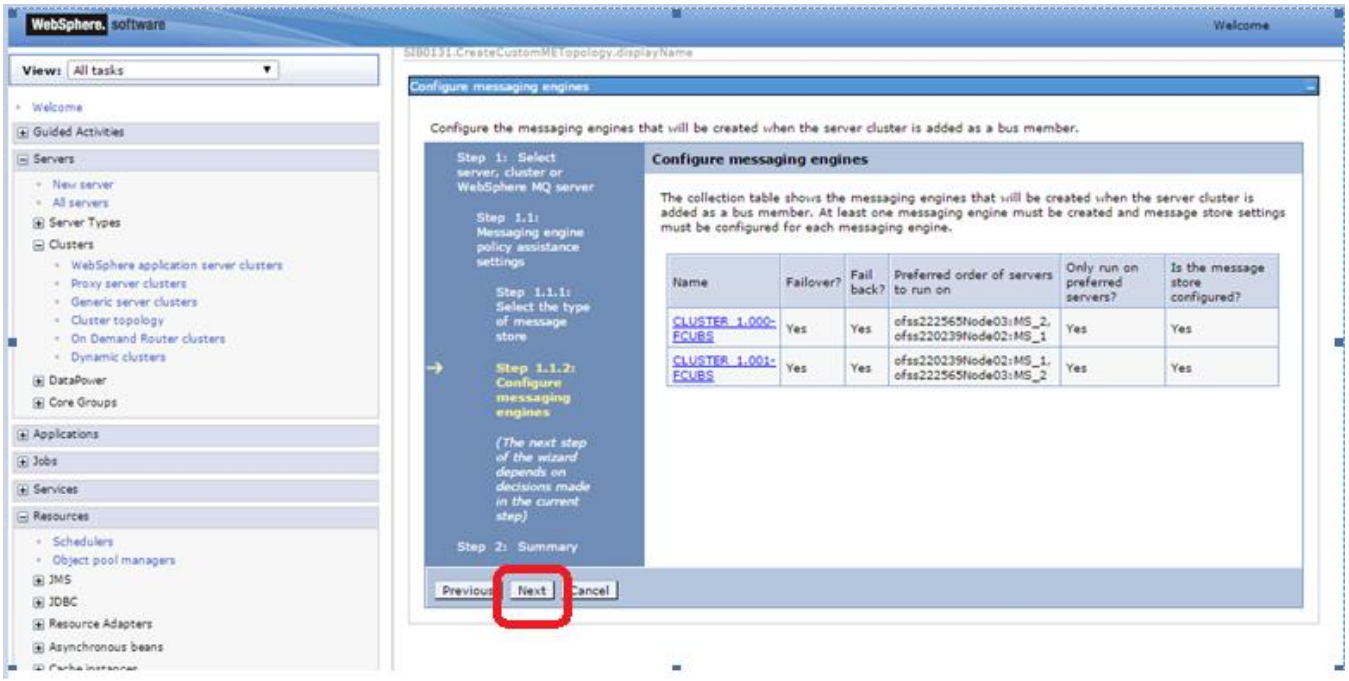
7) Enter the Log Directory Path and Permanent store directory path(shared path across the nodes) and Click on Next



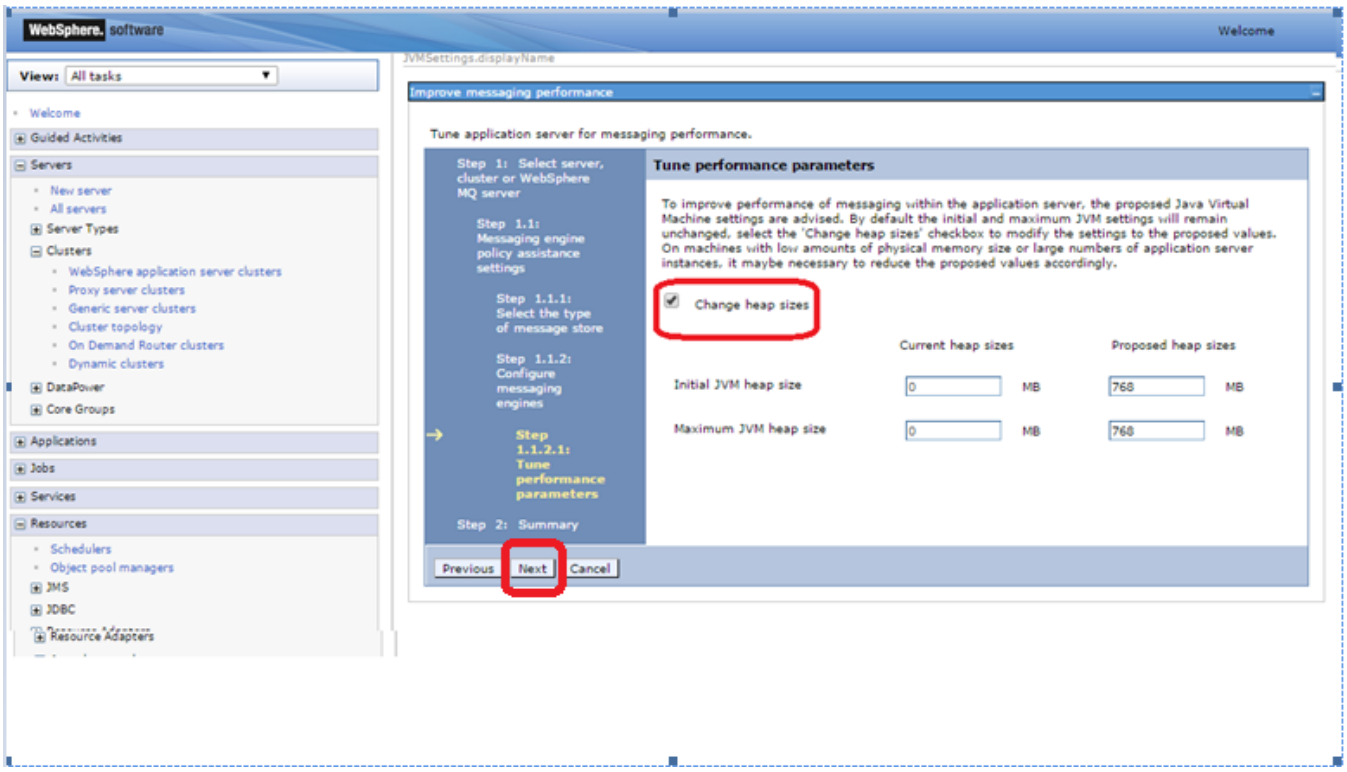
8) Click on other message engine and set the FileStore



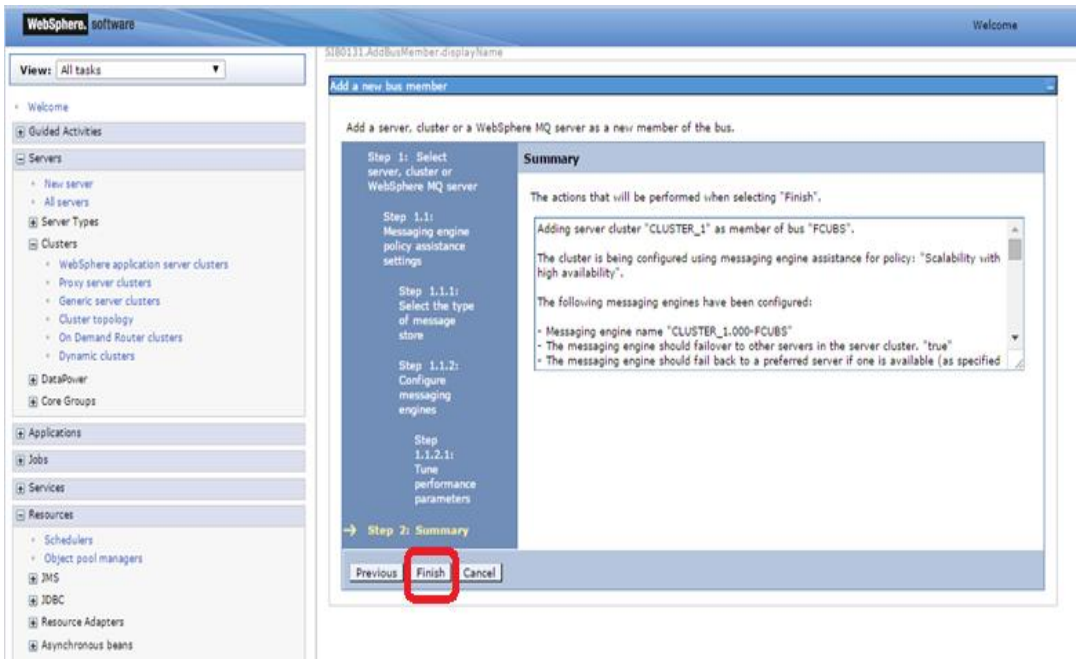
9) Click on Next after Setting FileStore for all messaging engines



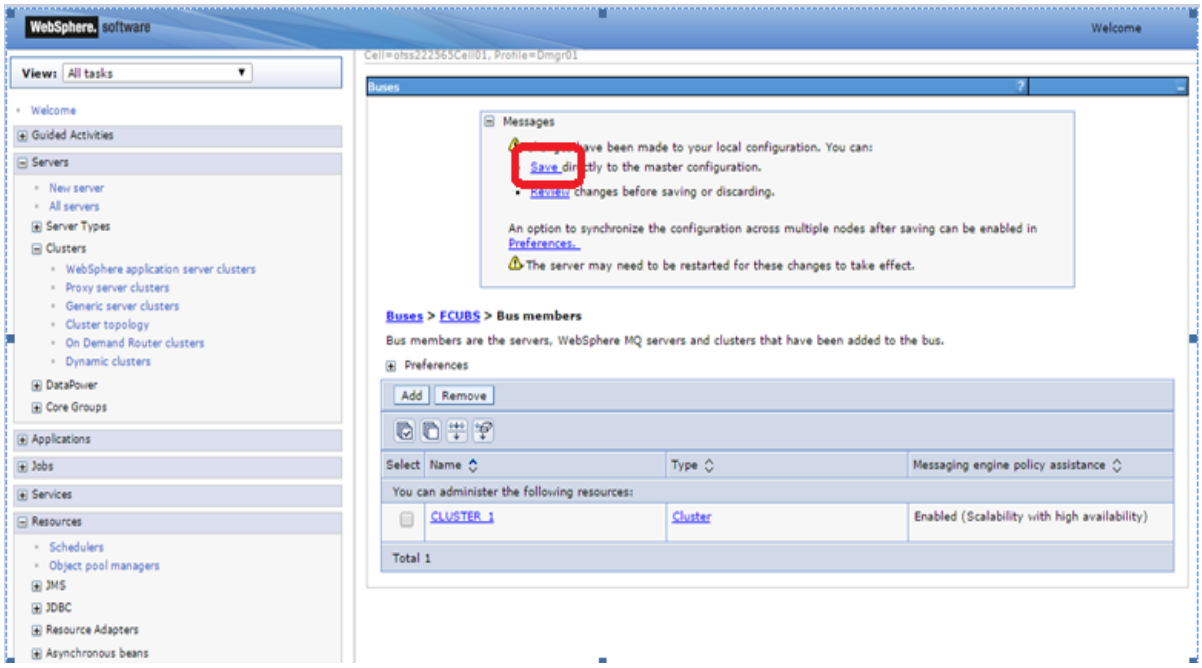
10) Select Change Heap Sizes and Click on Next



11) Click on Finish

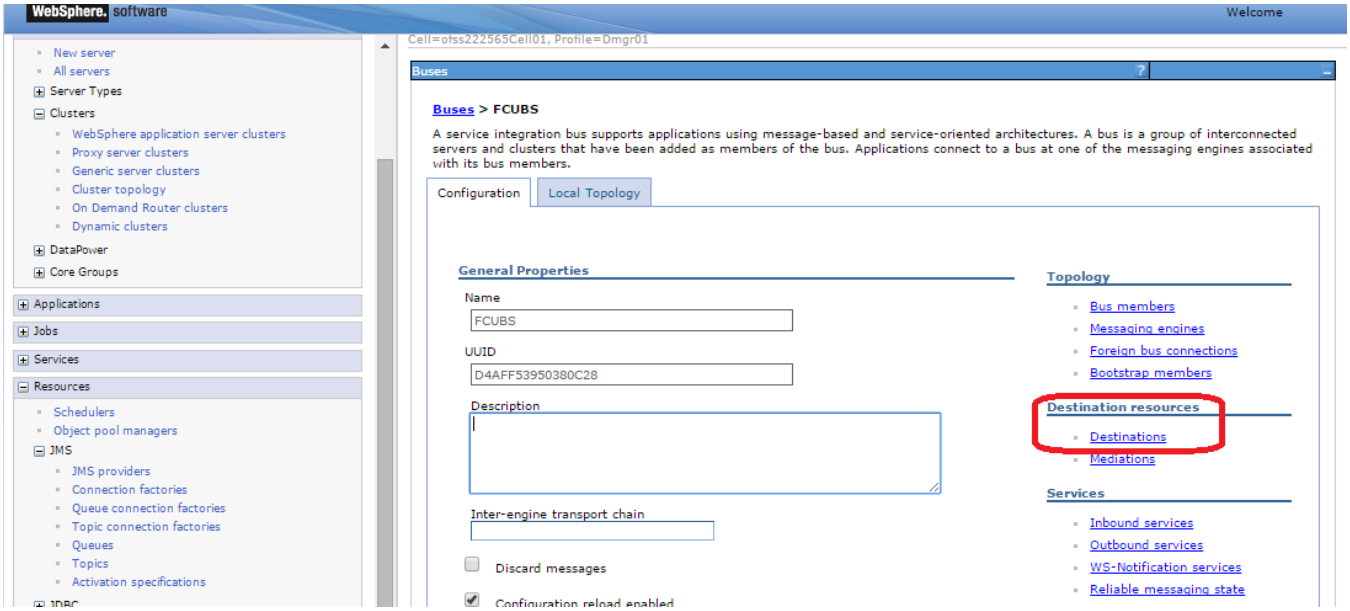


12) Click on Save

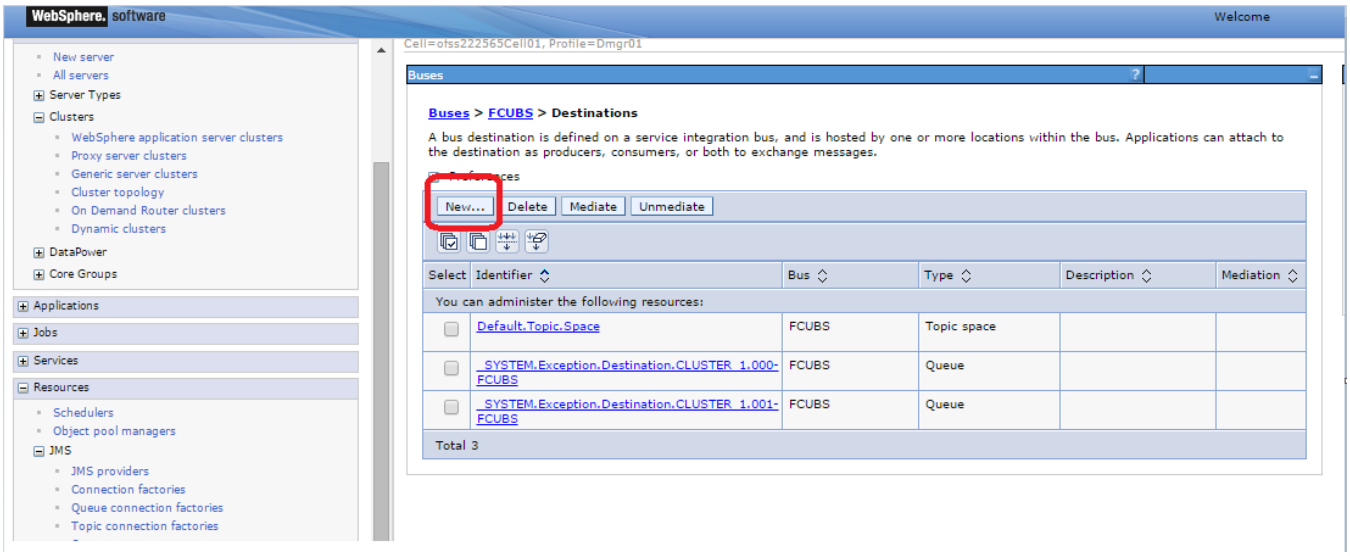


4.3 Destination Queue Creation

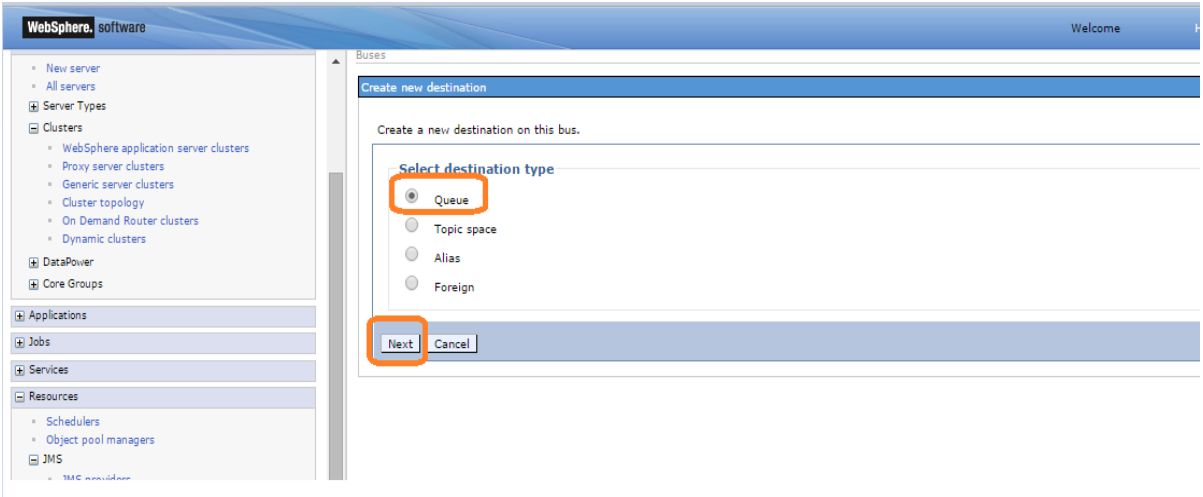
1) Navigate to Service Integration > Buses > Click on FCUBS(new bus Created) > Click on Destination under Destination Resources



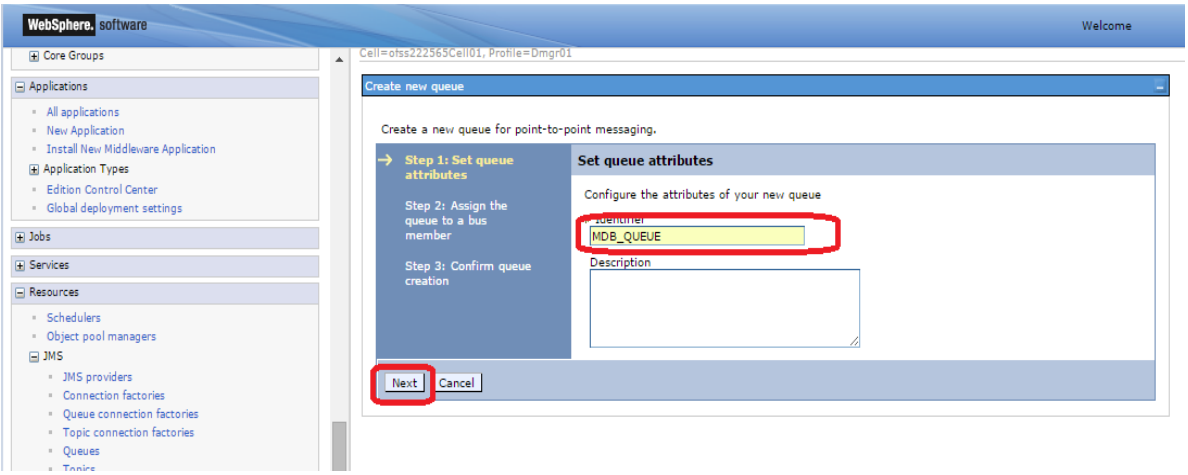
2) Click on New



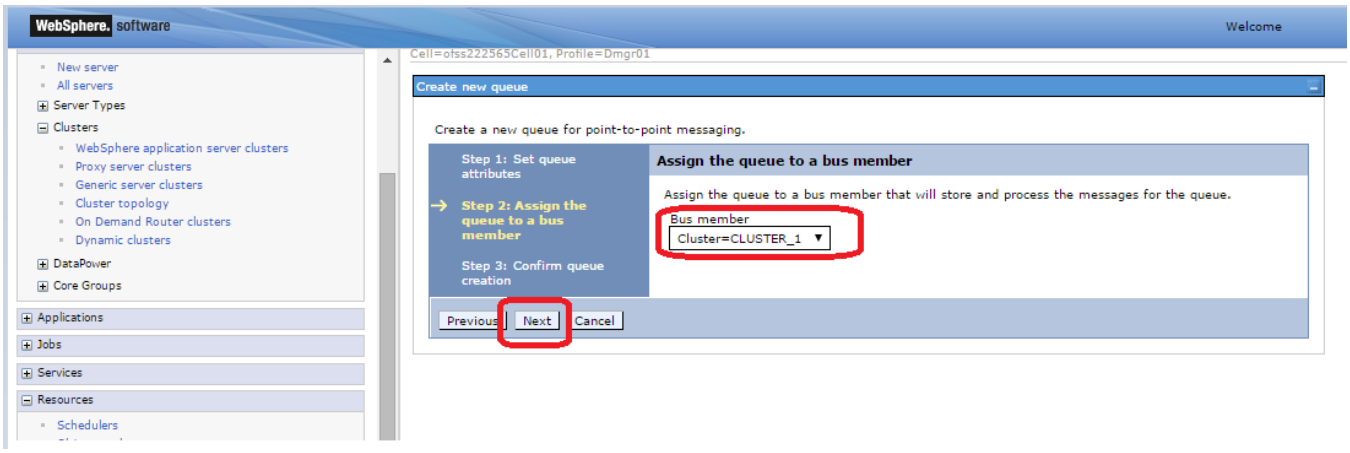
3) Select Queue and Click on Next



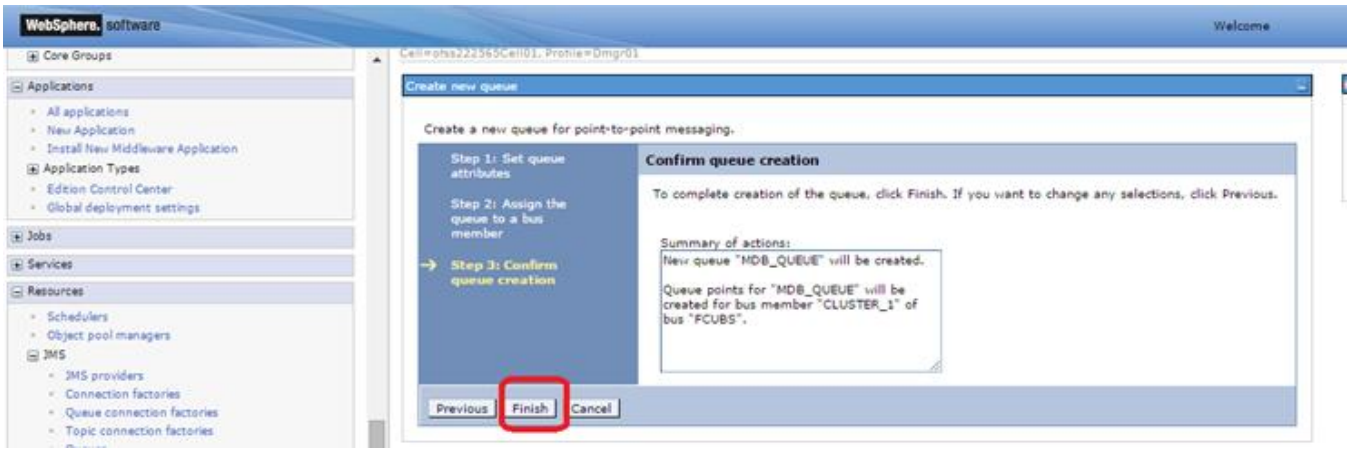
4) Enter Identifier as MDB_QUEUE and Click on Next



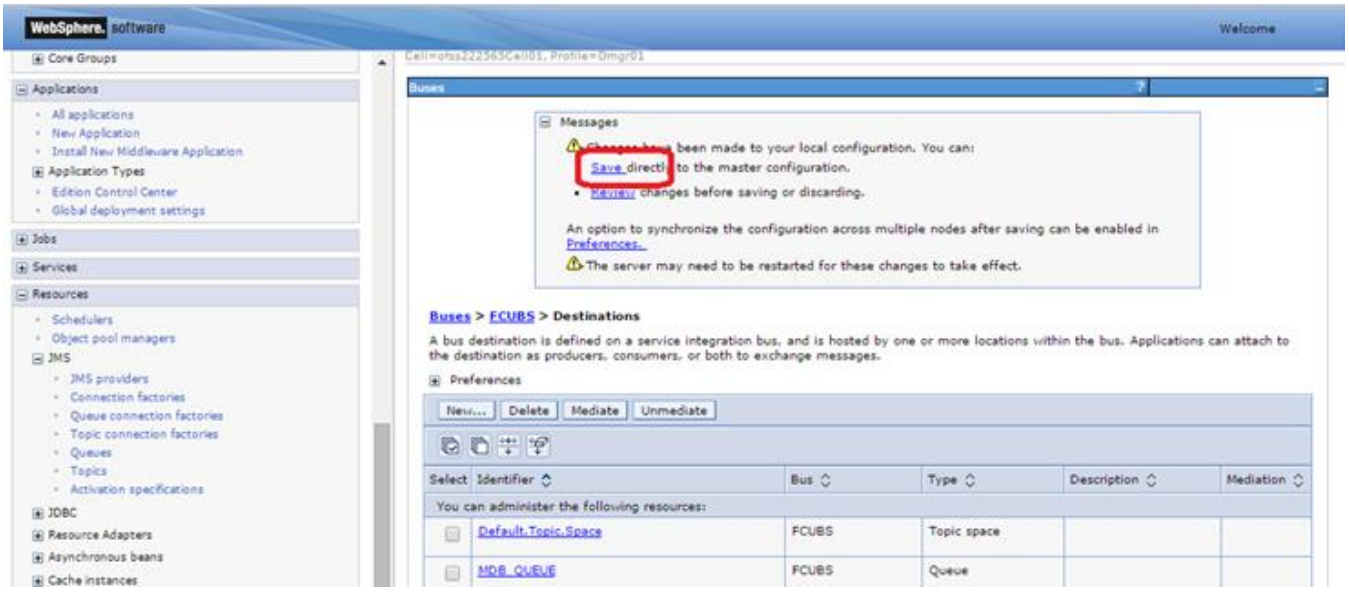
5) Select Bus Member as Cluster and Click on Next



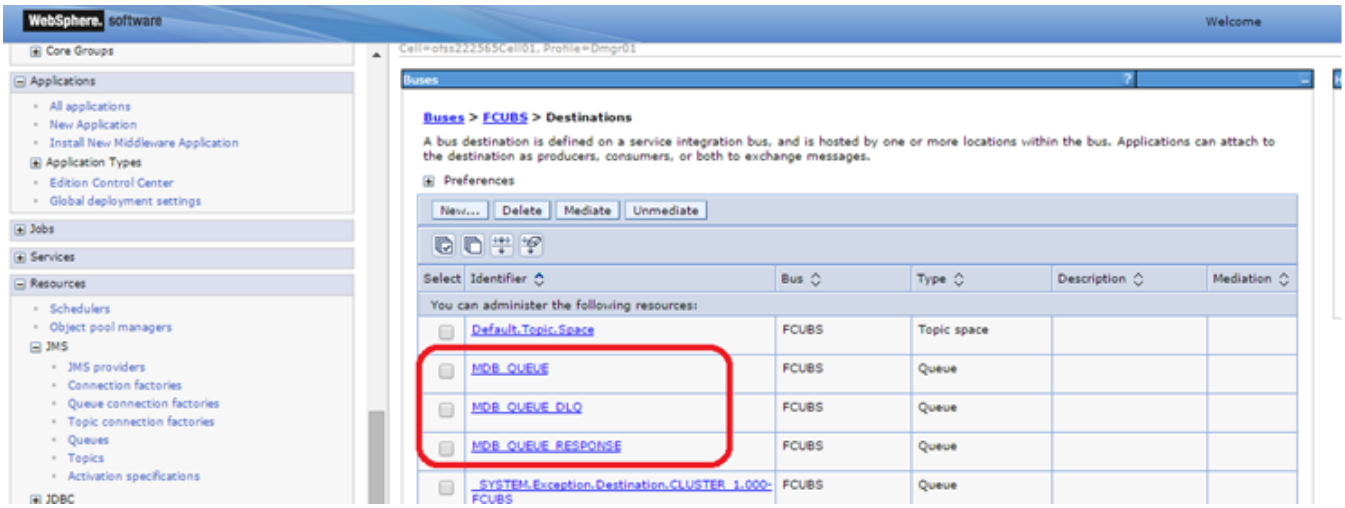
6) Click on Finish



7) Click on Save



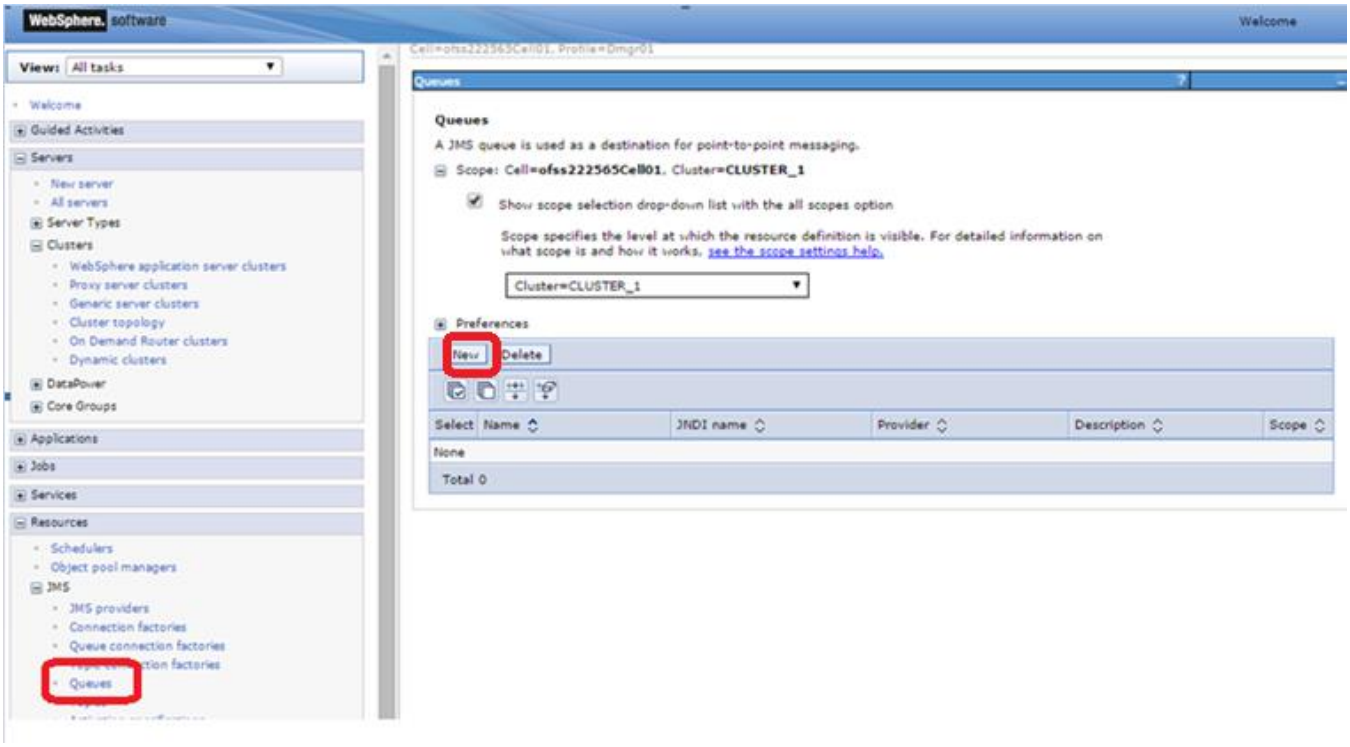
8) Similarly create Destinations for all the other Queue's required



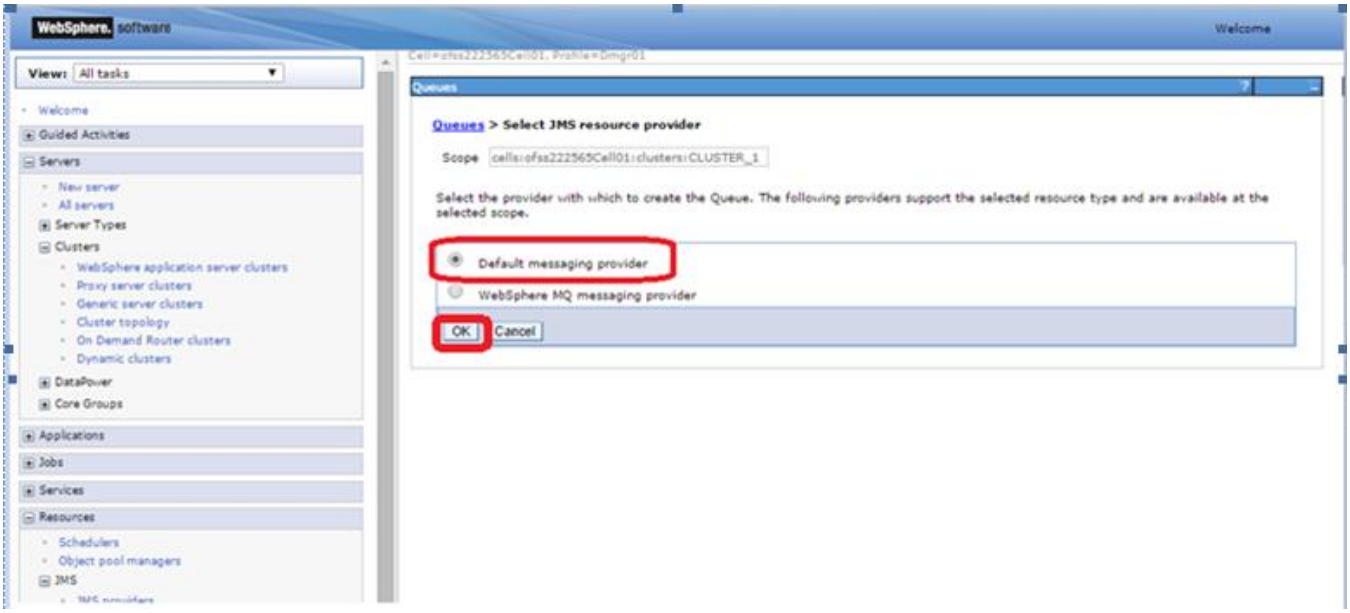
5. Resource Creation

5.1 Queue Creation

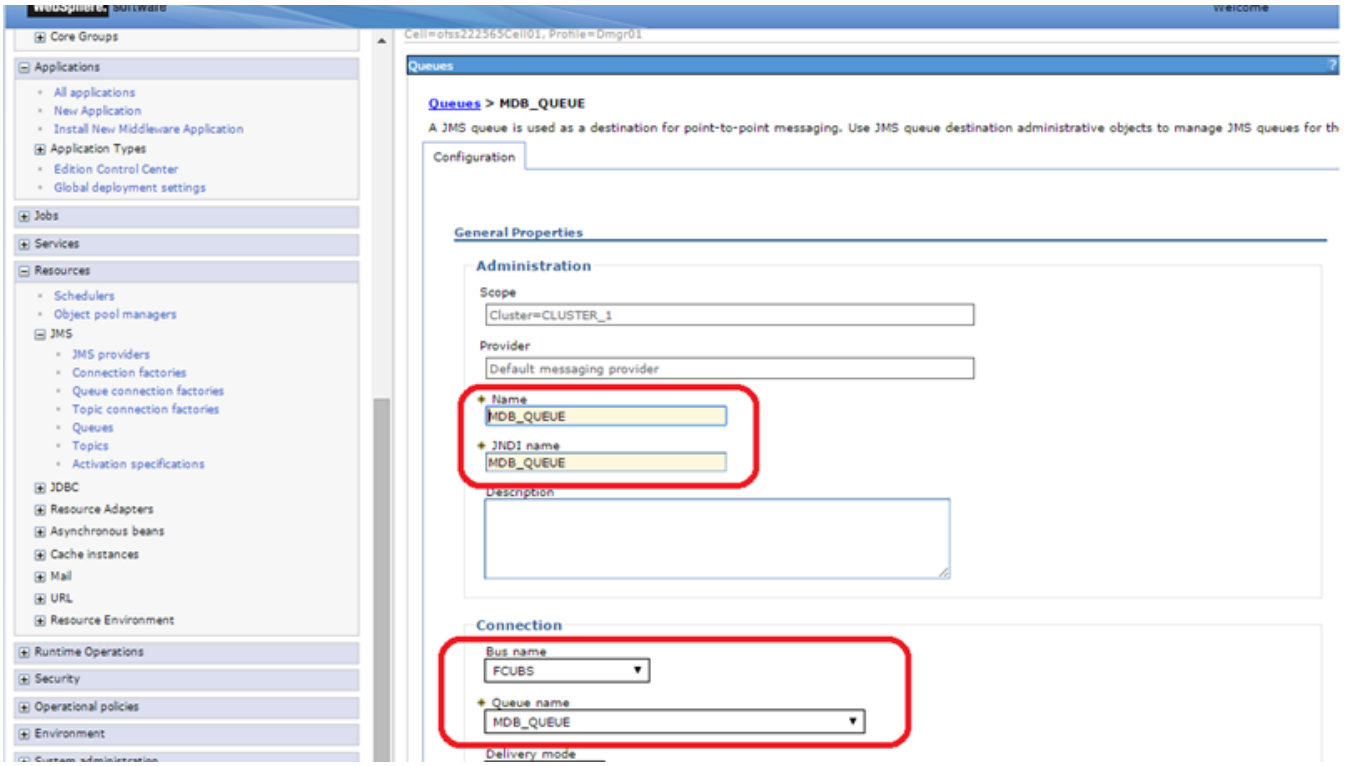
1) Navigate to Resources > JMS > Queues > Select Scope as Cluster and Click on New



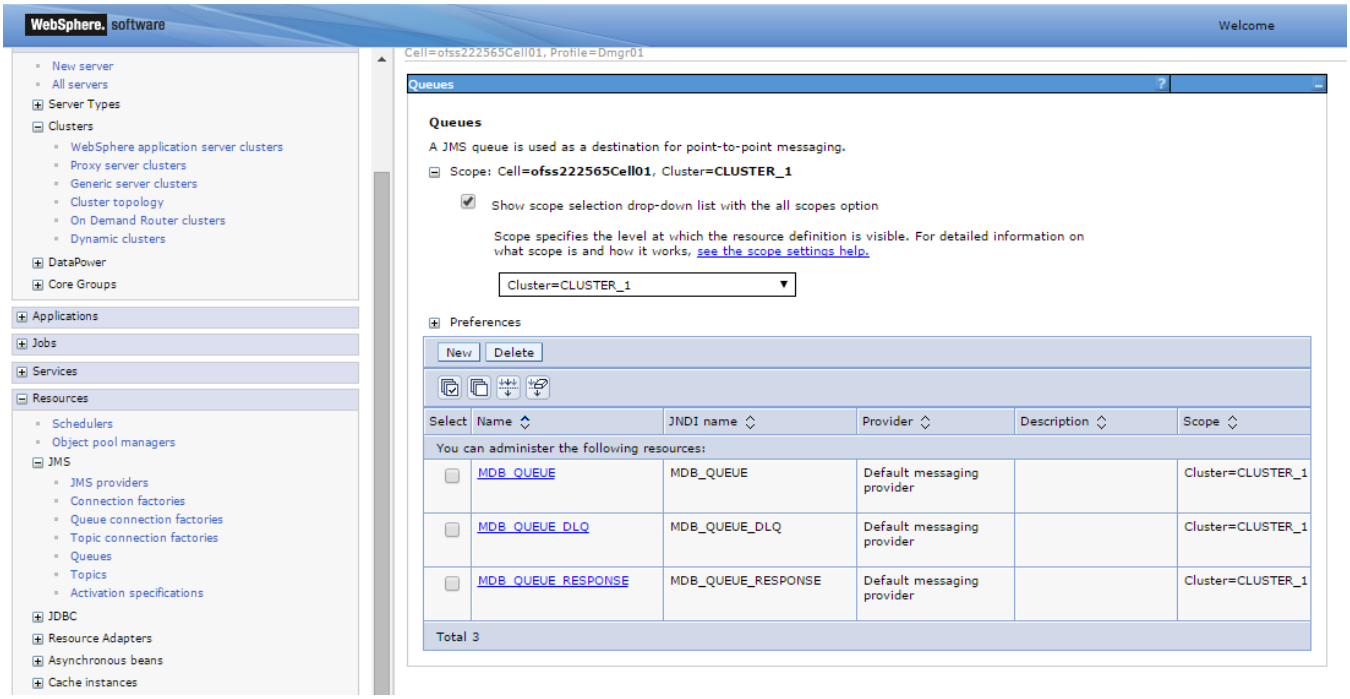
2) Select “Default messaging provider” and Click on OK



3) Enter The Name, JNDI Name. Select Bus and Queue Name accordingly and Click on OK

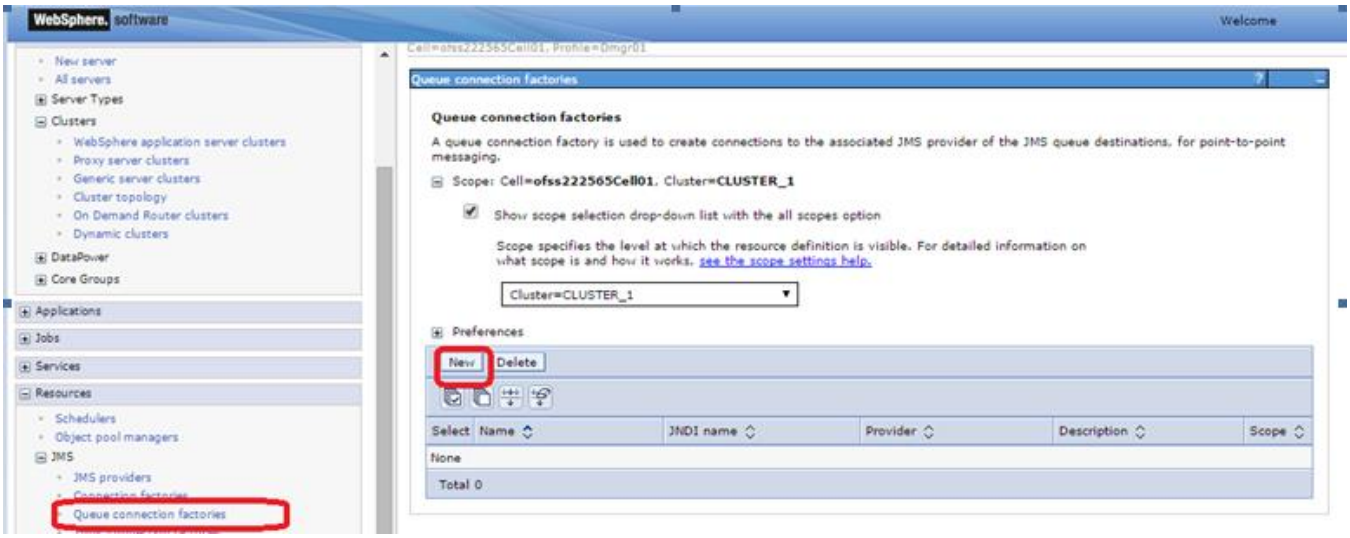


4) Similarly create other Queue's required

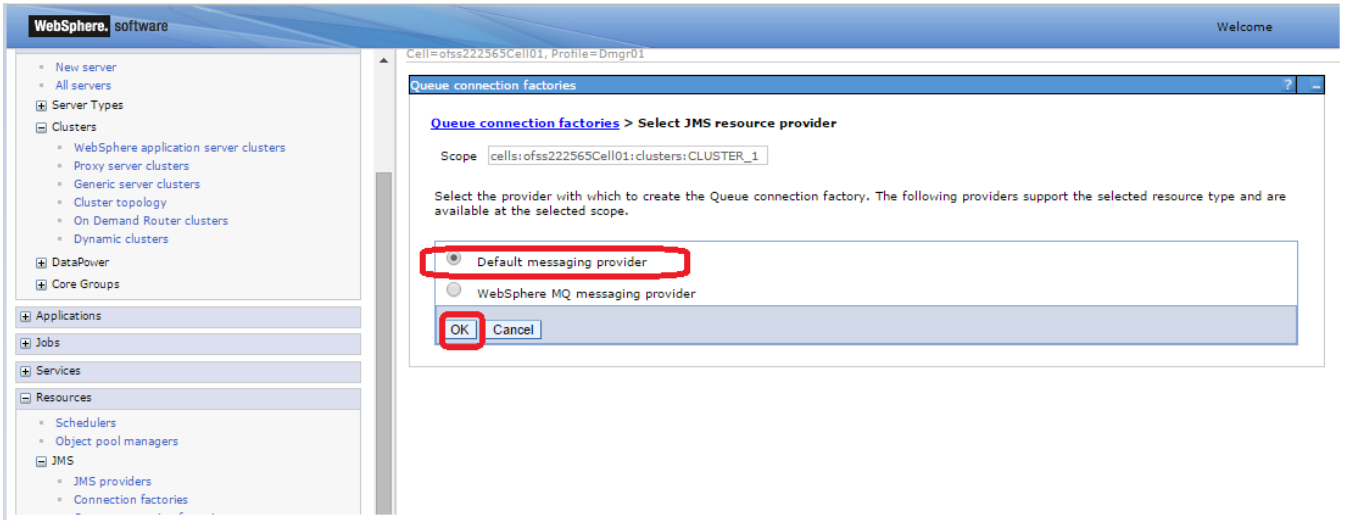


5.2 Connection Factory Creation

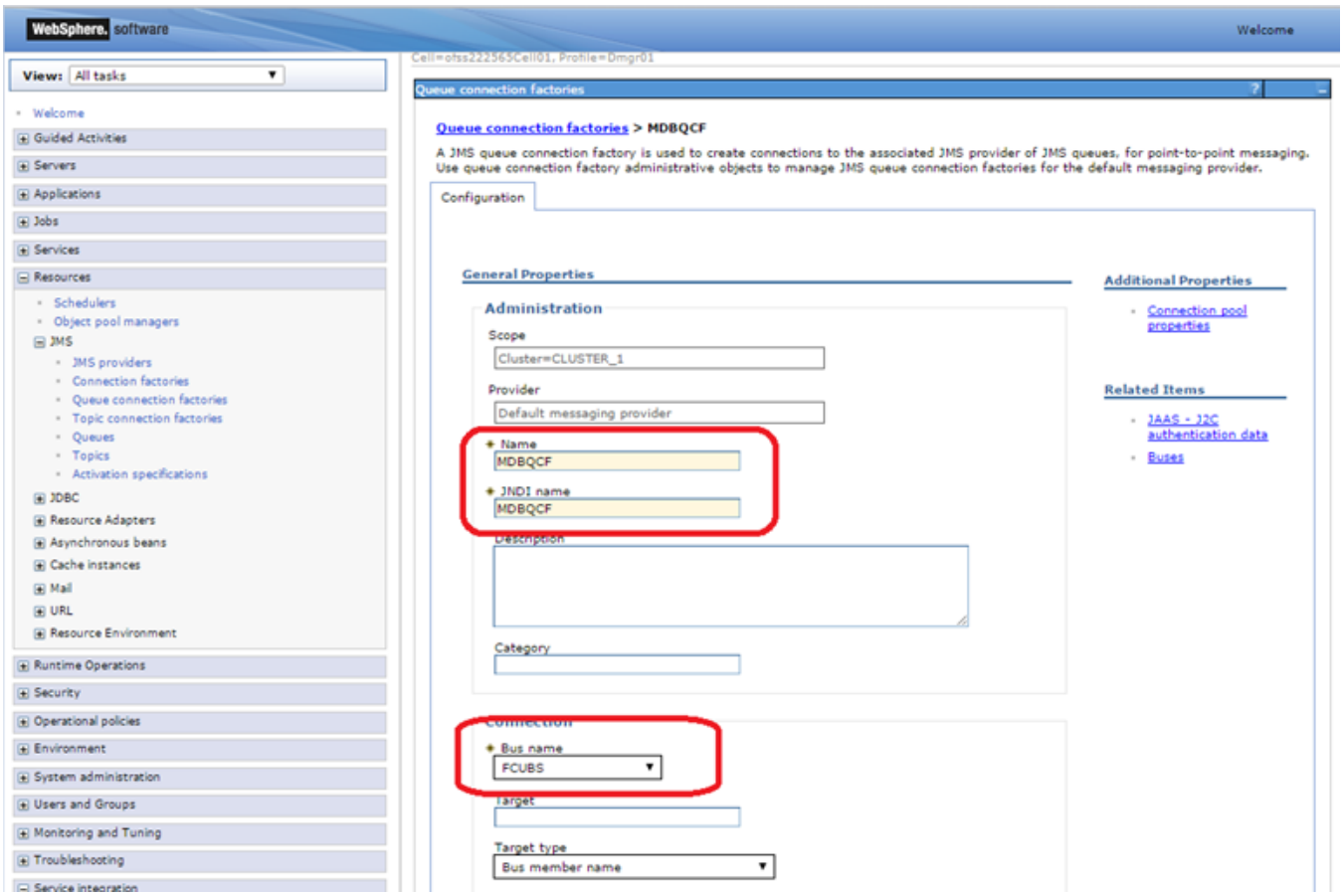
1) Navigate to Resources > JMS > Queue Connection Factory > Select Scope as Cluster and Click on New



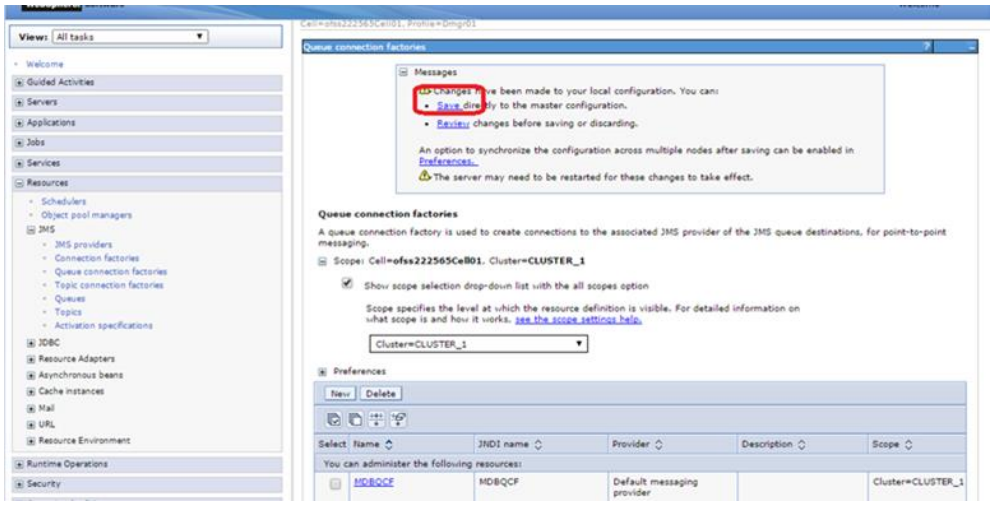
2) Select “Default messaging provider” and Click on OK



Enter Name, JNDI Name, Select Bus Name and Click on OK

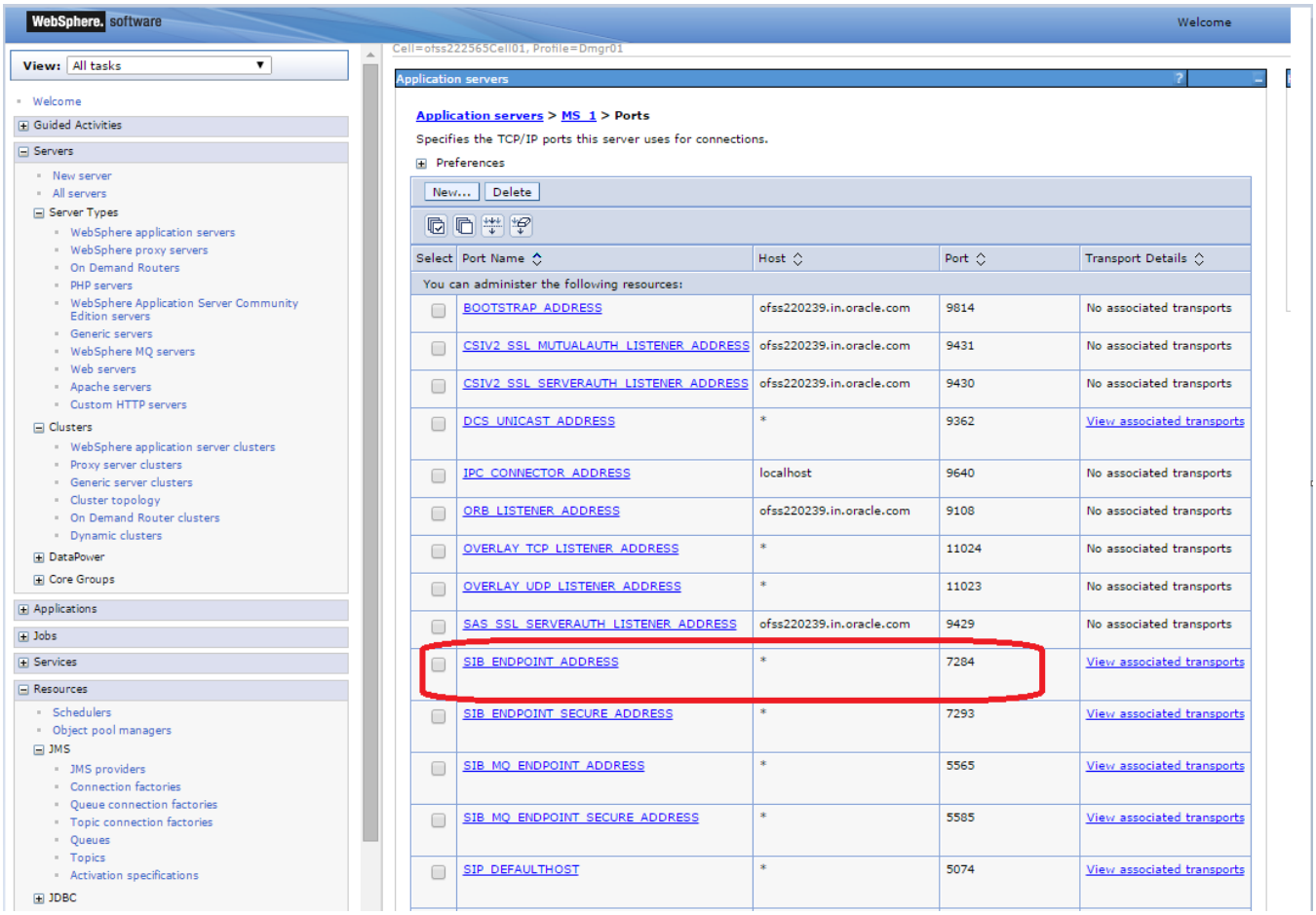


3) Click on Save



5.2.1 Managed Servers SIB Ports

1) Navigate to Servers > Websphere Application Servers > SERVER_NAME > Click on Ports under Communications > Note down the port of SIB_ENDPOINT_ADDRESS



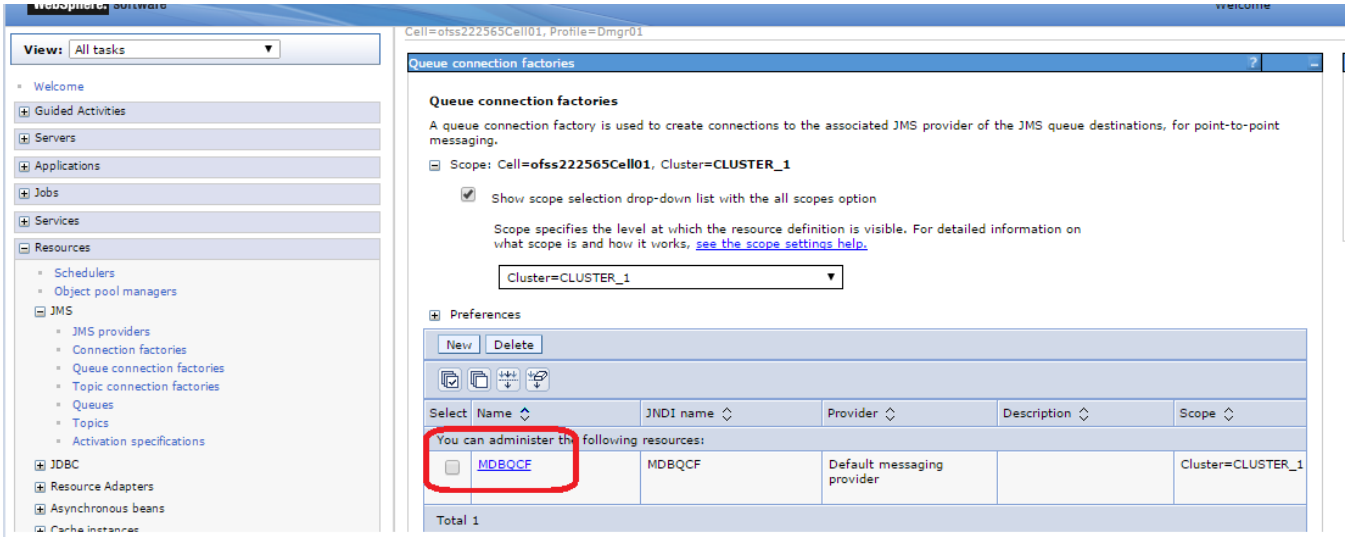
2) Similarly navigate to all other managed servers in the cluster and note down the port of SIB_ENDPOINT_ADDRESS

3) Prepare the “Provider Endpoint” String as below

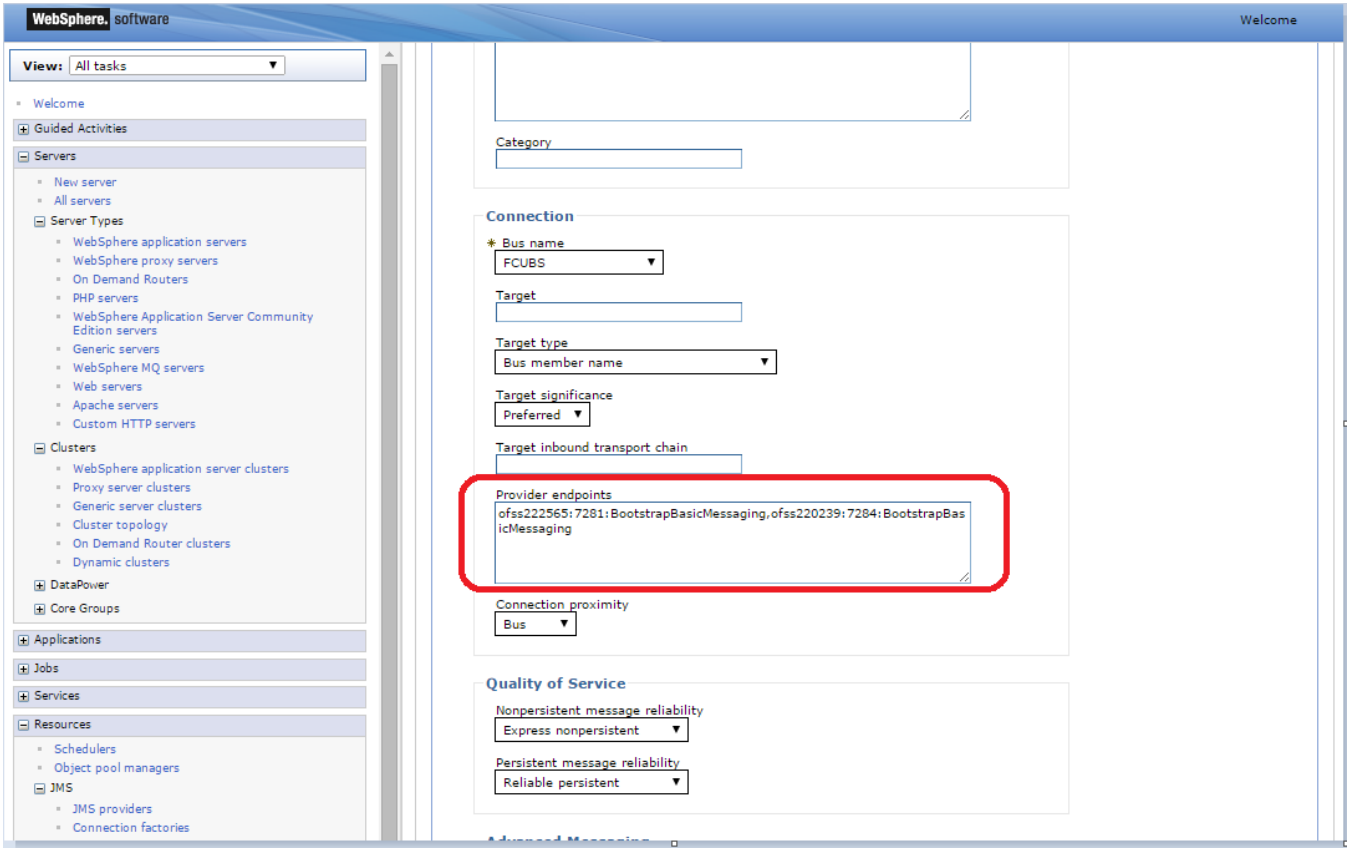
<hostname1/IP Address1>:<PORT ofSIB_ENDPOINT_ADDRESS>:BootstrapBasicMessaging

In this case the Provider Endpoint String would be ofss222565:7281:BootstrapBasicMessaging,ofss220239:7284:BootstrapBasicMessaging

Navigate to Resources > JMS > Queue Connection Factory > Click on newly created connection factory

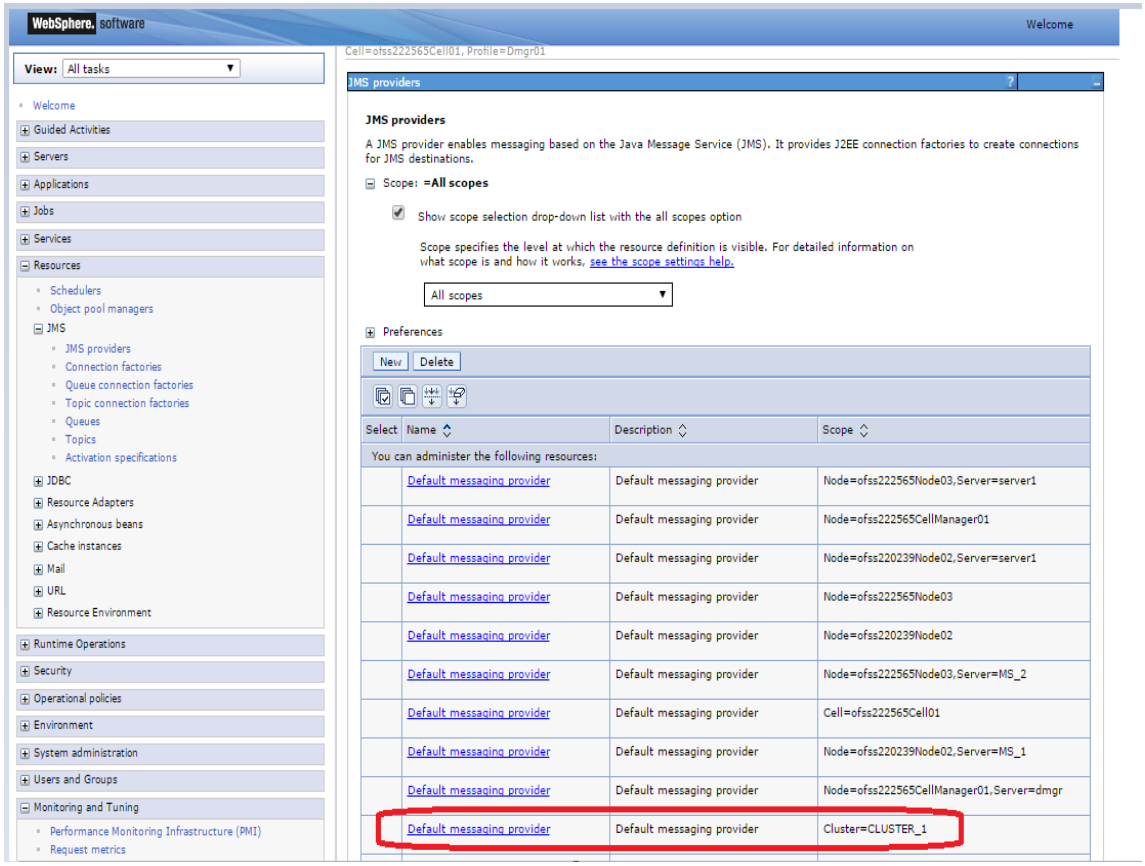


4) Update the Provider endpoints as prepared above and Click on OK

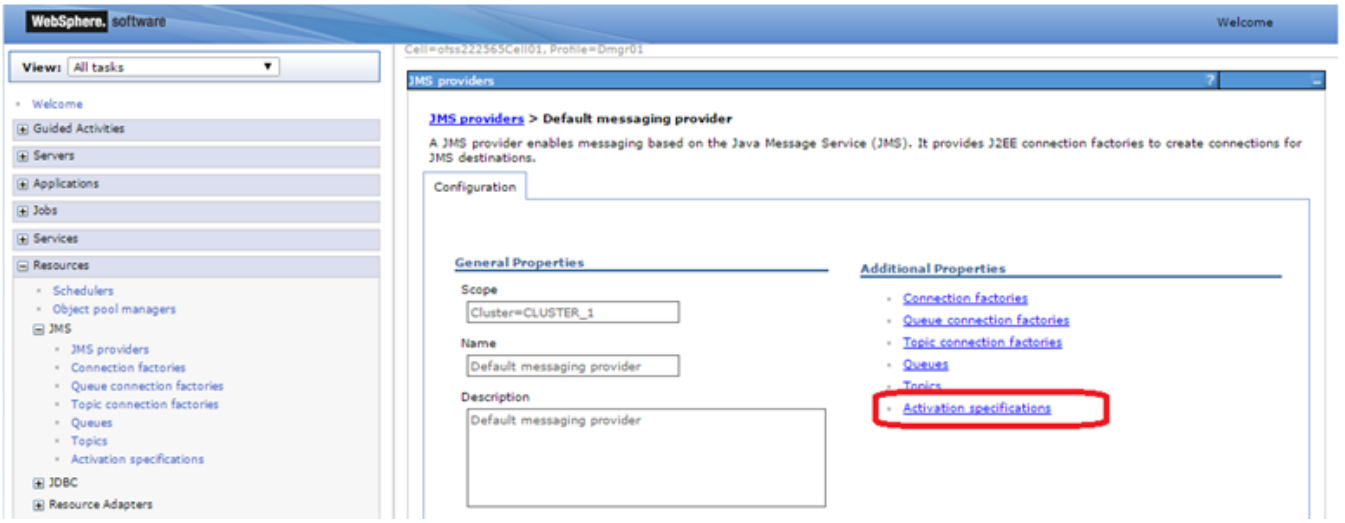


5.3 JMS Activation Specifications for Cluster

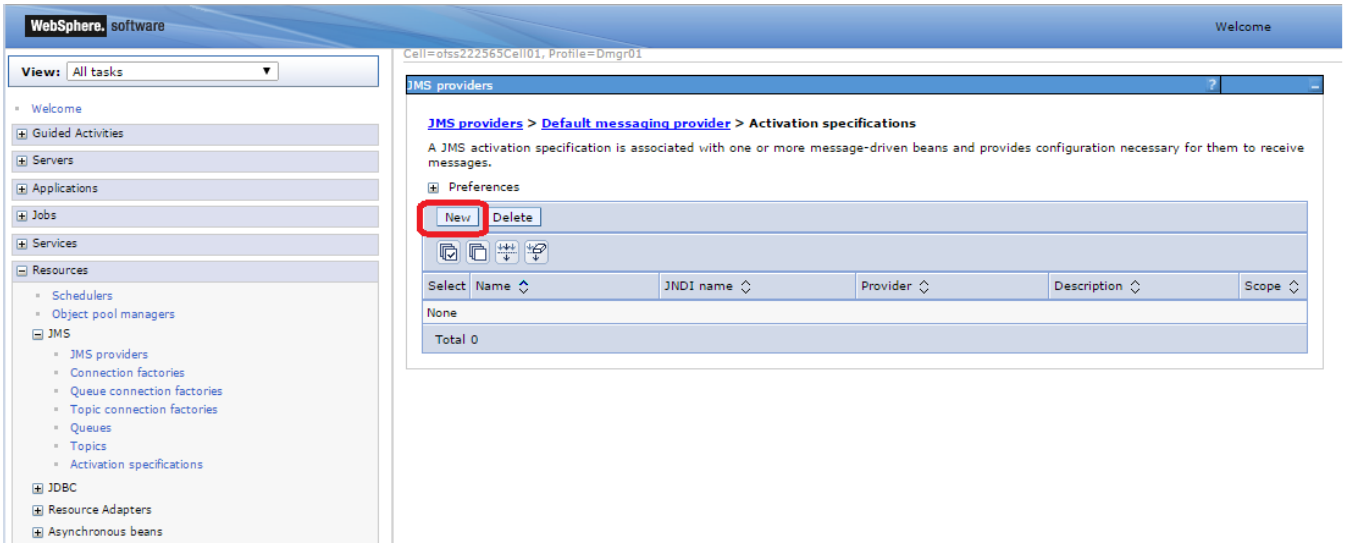
1) Navigate to Resources > JMS > JMS Providers > Click Default messaging provider for the cluster created



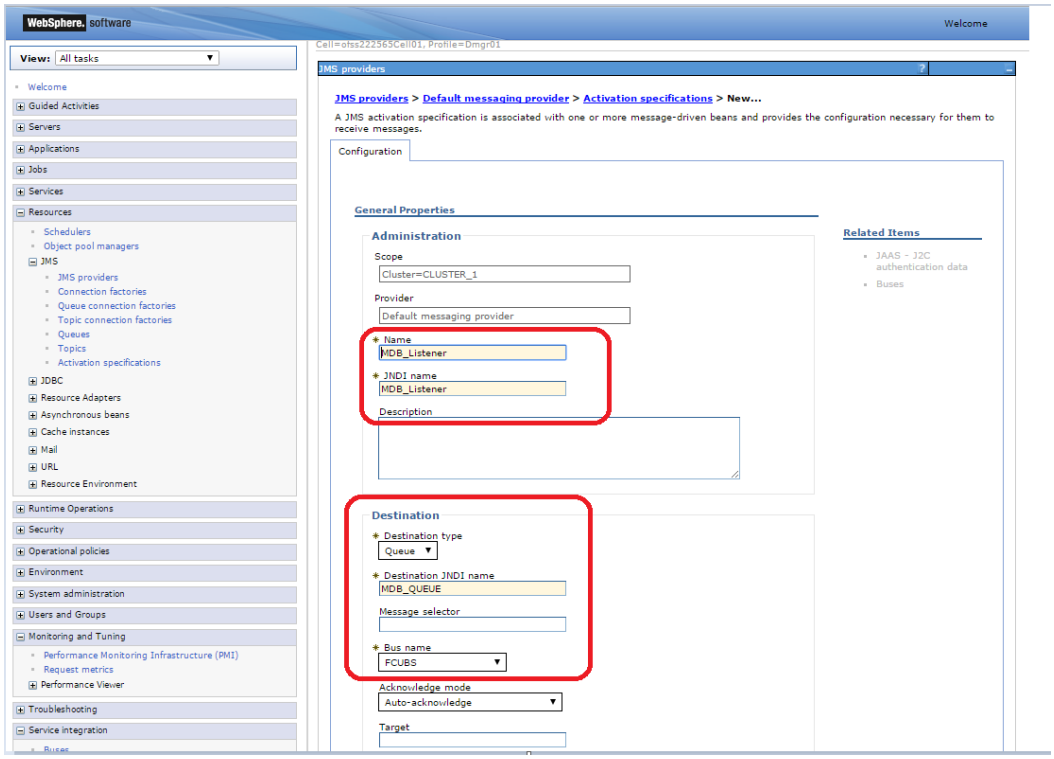
2) Under Additional Properties, click Activation specifications



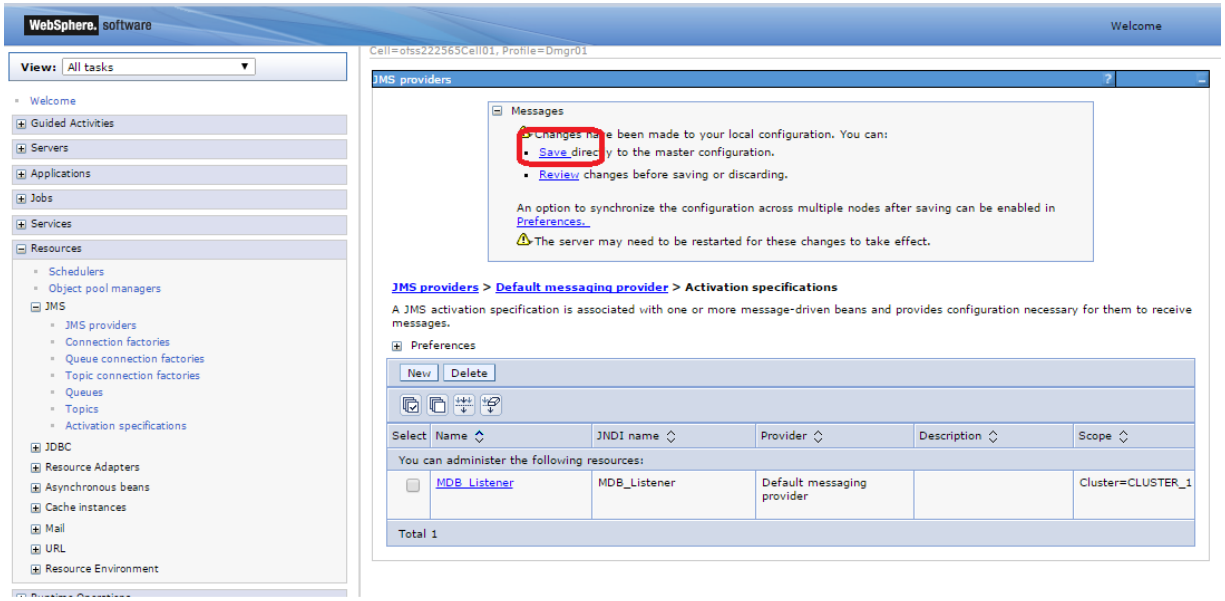
3) Click on New



4) Enter Name, JNDI Name, Select Destination Type as Queue and Enter Queue Name, Select Bus and Click on OK

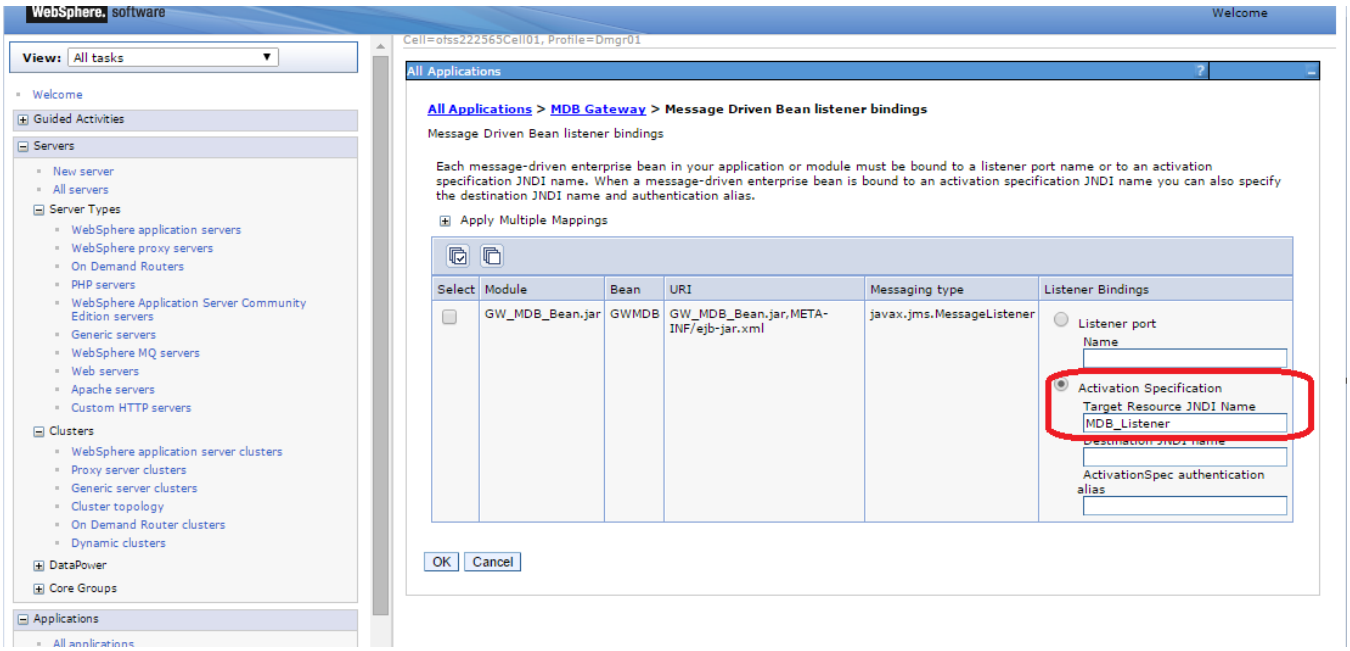


5) Click on Save



1) Deploy the EAR with Target as Cluster_1. Except below step rest is usual way of deploying the EAR.

2) During deployment give the Activation Specification Created above in the Activation Specification.



6.1 Restart Servers

Restart the Admin and Managed Servers.

7. Frequently Asked Questions

7.1 How to Test the Deployment

1) Send a sample message from the any third party application by connecting to

   iiop://<hostname or ip>:<BOOTSTRAP_ADDRESS>

   eg: iiop://ofss222565:9811

2) Verify at backend or in the MDB log if the message is processed successfully.

   Or

1) Use the below java program to send a sample message.
2) Set Java Home
3) Set $WAS_HOME/runtimes/com.ibm.ws.ejb.thinclient_8.5.0.jar, $WAS_HOME/runtimes/com.ibm.ws.sib.client.thin.jms_8.5.0.jar and javaee.jar in the CLASSPATH.
4) Change the URL, USER, PASSWORD, messageText in the Java Program and Compile.
5) Run the program and verify at backend or in MDB log.

```
import java.util.Hashtable;

import javax.jms.JMSException;

import javax.jms.Queue;

import javax.jms.QueueConnection;

import javax.jms.QueueConnectionFactory;

import javax.jms.QueueSender;

import javax.jms.QueueSession;

import javax.jms.Session;

import javax.naming.Context;

import javax.naming.InitialContext;

import javax.naming.NamingException;
```

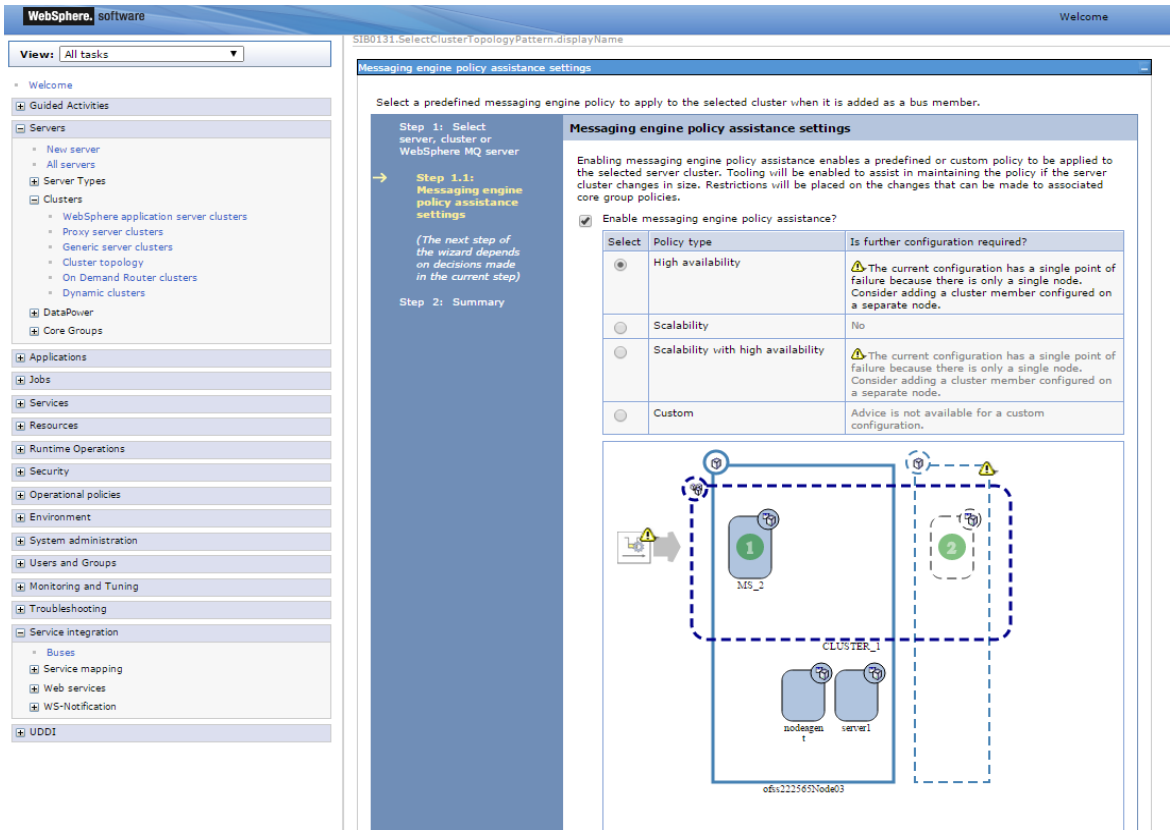
```
import javax.jms.TextMessage;


public class JMSQueueTest {
    public JMSQueueTest() {
        super();    }
    private Context ctx;
    private InitialContext initialContext;
    private QueueConnectionFactory queueCF;
    private QueueConnection queueConn;
    private QueueSession queueSession;
    private Queue queue;
    private QueueSender queueSender;
    private final static String JNDI_FACTORY =
"com.ibm.websphere.naming.WsnInitialContextFactory";
    private final static String JMS_FACTORY = "MDBQCF";
    private final static String QUEUE = "MDB_QUEUE";
    private final static String URL = "iiop://ofss222565:9811";
    private TextMessage txtMessage;
    private static String USER = "wasadmin";
    private static String PASSWORD = "wasadmin123";
    private static String messageText = "Hello!";
    private InitialContext getInitialContext(String url) throws Exception {
        Hashtable envHash = new Hashtable();
        envHash.put(Context.INITIAL_CONTEXT_FACTORY, JNDI_FACTORY);
        envHash.put(Context.PROVIDER_URL, url);
        envHash.put(Context.SECURITY_PRINCIPAL, USER);
        envHash.put(Context.SECURITY_CREDENTIALS, PASSWORD);
```

```
    try {
        return new InitialContext(envHash);
    } catch (NamingException e) {
        e.printStackTrace();        }
    return new InitialContext(envHash);    }
private void init(Context ctx, String queueName) {
    try {
        ctx = getInitialContext(URL);
        queueCF = (QueueConnectionFactory)ctx.lookup(JMS_FACTORY);
        queueConn = queueCF.createQueueConnection();
        queueSession = queueConn.createQueueSession(false,Session.SESSION_TRANSACTED);
        queue = (Queue)ctx.lookup(queueName);
        queueSender = queueSession.createSender(queue);
        txtMessage = queueSession.createTextMessage();
        queueConn.start();
    } catch (Exception e) {
        e.printStackTrace();        }    }
private void close() throws JMSException {
    queueSender.close();
    queueSession.close();
    queueConn.close();    }
private void sendMessage(String message) throws JMSException {
    txtMessage.setText(messageText);
    queueSender.send(txtMessage);    }
public static void main(String[] args) throws Exception {
    JMSQueueTest jmsq = new JMSQueueTest();
    InitialContext ico = jmsq.getInitialContext(URL);
```

```
        try {
            jmsq.init(ico, QUEUE);
            jmsq.sendMessage(messageText);
        } catch (JMSException jmse) {
            jmse.printStackTrace();
        } finally {
            jmsq.close();       }
            }
}
```

7.2 Warning during Bus Member Creation

During Bus member creation Warning is shown in “Is Further configuration Required?”



Examine the resulting diagram and the messages for the selected messaging engine policy type. Act on the messages as follows:

1) To add a server or a node, go back and change the cluster topology before you continue with the current procedure.

2) To add or remove messaging engines, under Additional Properties, click Messaging engines and use the options on the resulting pane.

3) To correct messaging engine policies, under Additional Properties, click Messaging engine policy maintenance and use the options on the resulting pane.

When the "Is further configuration required" column for the selected messaging engine policy type displays No, the configuration is complete.

7.3 Message Engines Not Gettting Started

Message engine fail to start and gives SIB Service Bus Unavailable error.

1) Ensure that shared folders are empty

2) Restart the Managed Servers

3) Check the Status of message engines

7.4 Cannot Establish Connection Error

When a message is received on the Queue it throws below error

Caused by: com.ibm.websphere.sib.exception.SIResourceException: CWSIC1001E: A client attempted to connect with a remote messaging engine but the connection cannot be completed. Ensure the messaging engine is started: exception com.ibm.ws.sib.jfapchannel.JFapConnectFailedException: CWSIJ0063E: A network connection to host name localhost/127.0.0.1, port 7,276 cannot be established.

1) Ensure that Provider EndPoint contains the SIB_ENDPOINT_ADDRESS of all the servers comma separated

2) Eg: <hostname1>:<port1>:BootstrapBasicMessaging, <hostname2>:<port 2>:BootstrapBasicMessaging,

3) Restart the servers after making changes

7.5 How to setup for Scheduler/Notifications

The above document can be used for setting up JMS for scheduler/notifications but additional queues and connection factory needs to be created. Also the FCUBS application needs to be deployed.

7.6 What other modules uses JMS Queue's

JMS is used by following modules, relevant queues and factories needs to be created additionally

- EMS for swift messages
- GI for upload
- ELCM
- BIP

8. References

1) GATEWAY_Applications_WAS.doc

2) Resource_Creation_WAS.doc

3) FCUBS_Application_WAS.doc

4) http://129.33.205.81/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.iseries.doc/ae/welc6topmanaging.html

5) http://publib.boulder.ibm.com/infocenter/wsdoc400/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.iseries.doc/tasks/tjn9999_.html

FCUBS_JMS_Websphere_Configuration
[May] [2018]
Version 14.1.0.0.0

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
https://www.oracle.com/industries/financial-services/index.html