Payments Weblogic Configuration
Oracle Banking Payments

Release 14.4.0.0.0

[May] [2020]

ORACLE
FINANCIAL SERVICES

ORACLE

Table of Contents

1. Configuring SSL on Oracle Weblogic

1.1 Introduction

This chapter details out the configurations for SSL on Oracle Weblogic application server.

1.2 Setting up SSL on Oracle Weblogic

To setup SSL on Oracle Weblogic application server, you need to perform the following tasks:

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for Oracle Weblogic application server.
2. Store the identity and trust. Private keys and trust CA certificates are stored in keystores.
3. Configure the identity and trust the keystores for Oracle Weblogic application server in the administration console.
4. Set SSL attributes for the private key alias and password in Oracle Weblogic administration console.

1.3 Certificates and Keypairs

Certificates are used for validating the authenticity of the server. Certificates contains the name of the owner, certificate usage, duration of validity, resource location or distinguished name (DN), which includes the common name (CN - web site address or e-mail address depending of the usage) and the certificate ID of the person who certified (signs) these information. It also contains the public key and a hash to ensure that the certificate has not been tampered with. A certificate is insecure until it is signed. Signed certificates cannot be modified.

A certificate can be self signed or obtained from a reputable certificate authority such as Verisign, Inc., Entrust.net, Thawte, GeoTrust or InstantSSL.

SSL uses a pair of cryptographic keys - a **public key** and a **private key**. These keys are similar in nature and can be used alternatively. What one key encrypts can be decrypted by the other key of the pair. The private key is kept secret, while the public key is distributed using the certificate.

A **keytool** stores the keys and certificates in a **keystore**. The default keystore implementation implements it as a file. It protects private keys with a password. The different entities (key pairs and the certificates) are distinguished by a unique ‘alias’. Through its keystore, Oracle Weblogic server can authenticate itself to other parties.

In Java, a keystore is a ‘java.security.KeyStore’ instance that you can create and manipulate using the **keytool** utility provided with the Java Runtime.

There are two keystores to be managed by Oracle Weblogic server to configure SSL.

- Identity Keystore: Contains the key pairs and the Digital certificate. This can also contain certificates of intermediate CAs.
- Trust Keystore: Contains the trusted CA certificates.

2. Choosing the Identity and Trust Stores

2.1 Introduction

Oracle Financial Services Software recommends that the choice of Identity and Trust stores be made up front. Oracle Weblogic server supports the following combinations of Identity and Trust stores:

- Custom Identity and Command Line Trust
- Custom Identity and Custom Trust
- Custom Identity and Java Standard Trust
- Demo Identity and Demo Trust

Oracle Financial Services does not recommend choosing Demo Identity and Demo Trust for production environments.

It is recommended to separate the identity and trust stores, since each Weblogic server tends to have its own identity, but might have the same set of trust CA certificates. Trust stores are usually copied across Oracle Weblogic servers, to standardize trust rules; it is acceptable to copy trust stores since they contain public keys and certificates of CAs. Unlike trust stores, identity stores contain private keys of the Oracle Weblogic server, and hence should be protected against unauthorized access.

Command Line Trust, if chosen requires the trust store to be specified as a command line argument in the Weblogic Server startup script. No additional configuration of the trust store is required in the Weblogic Server Administration Console.

Java Standard Trust would rely on the cacerts files provided by the Java Runtime. This file contains the list of trust CA certificates that ship with the Java Runtime, and is located in the 'JAVA_HOME/jre/lib/security' directory. It is highly recommended to change the default Java standard trust store password from 'changeit' (without quotes), and the default access permission of the file. Certificates of most commercial CAs are already present in the Java Standard Trust store. Therefore, it is recommended to use the Java Standard Trust store whenever possible. The rest of the document will assume the use of Java Standard Trust, since most CA certificates are already present in it.

One can also create custom trust stores containing the list of certificates of trusted CAs.

For further details on identity and trust stores, please refer the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server.

3. Obtaining the Identity Store

3.1 Creating Identity Store with Self-Signed Certificates

Self-signed certificates are acceptable for use in a testing or development environment. Oracle Financial Services does not recommend the use of self-signed certificates in a production environment.

In order to create a self-signed certificate, the genkeypair option provided by the keytool utility of Sun Java 6 needs to be utilized.

3.1.1 Creation of Self-signed Certificate

Browse to the bin folder of JRE from the command prompt and type the following command.


The items highlighted in blue are placeholders, and should be replaced with suitable values when running the command.

```
keytool -genkeypair -alias alias -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair created. This alias is required later when configuring the SSL attributes for the managed servers in Oracle Weblogic Server.
2. ***keystore*** is used to specify the location of the JKS file. If no JKS file is present in the path provided, one will be created.

The command will prompt for the following attributes of the certificate and keystore:

1. **Keystore Password**: Specify a password that will be used to access the keystore. This password needs to be specified later, when configuring the identity store in Oracle Weblogic Server.
2. **Key Password**: Specify a password that will be used to access the private key stored in the keystore. This password needs to be specified later, when configuring the SSL attributes of the managed server(s) in Oracle Weblogic Server.
3. **First and Last Name (CN)**: Enter the domain name of the machine used to access Banking payments, for instance, www.example.com
4. **Name of your Organizational Unit**: The name of the department or unit making the request, for example, BPD. Use this field to further identify the SSL Certificate you are creating, for example, by department or by physical server.
5. **Name of your Organization:** The name of the organization making the certificate request, for example, Oracle Financial Services. It is recommended to use the company or organization's formal name, and this name entered here must match the name found in official records.

6. **Name of your City or Locality**: The city in which your organization is physically located, for example Mumbai.
7. **Name of your State or Province**: The state/province in which your organization is physically located, for example Maharashtra.
8. **Two-Letter Country Code for this Unit**: The country in which your organization is physically located, for example US, UK, IN etc.

 The key generation algorithm has been specified as RSA, the key size as 1024 bits, the signature algorithm as SHA1withRSA, and the validity days as 365. These can be changed to suitable values if the need arises. For further details, please refer to the documentation of the keytool utility in the JDK utilized by Oracle Weblogic Server.

Example

Listed below is the result of a sample execution of the command:

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
genkeypair -alias selfcert -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore D:\keystores\FCUBSKeyStore.jks

Enter keystore password:<Enter a password to protect the keystore>

Re-enter new password:<Confirm the password keyed above>

What is your first and last name?

  [Unknown]:  cvrhp0729.i-flex.com

What is the name of your organizational unit?

  [Unknown]:  BPD

What is the name of your organization?

  [Unknown]:  Oracle Financial Services

What is the name of your City or Locality?

  [Unknown]:  Mumbai

What is the name of your State or Province?

  [Unknown]:  Maharashtra

What is the two-letter country code for this unit?

  [Unknown]:  IN

Is CN=cvrhp0729.i-flex.com, OU=BPD, O=Oracle Financial Services,
L=Mumbai, ST=Maharashtra, C=IN correct?

  [no]:  yes


Enter key password for <selfcert>

        (RETURN if same as keystore password):<Enter a password to
protect the key>

Re-enter new password:<Confirm the password keyed above>
```

3.2 Creating Identity Store with Trusted Certificates Issued by CA

3.2.1 Creation of Public and Private Key Pair

Browse to the bin folder of JRE from the command prompt and type the following command.


The items highlighted in blue are placeholders, and should be replaced with suitable values when running the command.

```
keytool -genkeypair -alias alias -keyalg keyalg -keysize keysize -
sigalg sigalg -validity valDays -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair created. This alias is required later when configuring the SSL attributes for the managed servers in Oracle Weblogic Server.
2. ***keyalg*** is the key algorithm used to generate the public and private key pair. The RSA key algorithm is recommended.
3. ***keysize*** is the size of the public and private key pairs generated. A key size of 1024 or more is recommended. Please consult with your CA on the key size support for different types of certificates.
4. ***sigalg*** is the algorithm used to generate the signature. This algorithm should be compatible with the key algorithm and should be one of the values specified in the Java Cryptography API Specification and Reference.
5. ***valdays*** is the number of days for which the certificate is to be considered valid. Please consult with your CA on this period.
6. ***keystore*** is used to specify the location of the JKS file. If no JKS file is present in the path provided, one will be created.

The command will prompt for the following attributes of the certificate and keystore:

1. **Keystore Password**: Specify a password that will be used to access the keystore. This password needs to be specified later, when configuring the identity store in Oracle Weblogic Server.
2. **Key Password**: Specify a password that will be used to access the private key stored in the keystore. This password needs to be specified later, when configuring the SSL attributes of the managed server(s) in Oracle Weblogic Server.
3. **First and Last Name (CN)**: Enter the domain name of the machine used to access Banking UBS, for instance, www.example.com
4. **Name of your Organizational Unit**: The name of the department or unit making the request, for example, BPD. Use this field to further identify the SSL Certificate you are creating, for example, by department or by physical server.

5. **Name of your Organization:** The name of the organization making the certificate request, for example, Oracle Financial Services. It is recommended to use the company or organization's formal name, and this name entered here must match the name found in official records.
6. **Name of your City or Locality**: The city in which your organization is physically located, for example Mumbai.
7. **Name of your State or Province**: The state/province in which your organization is physically located, for example Maharashtra.
8. **Two-letter Country Code for this Unit**: The country in which your organization is physically located, for example US, UK, IN etc.

Example

Listed below is the result of a sample execution of the command:

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
genkeypair -alias cvrhp0729 -keyalg RSA -keysize 1024 -sigalg
SHA1withRSA -validity 365 -keystore D:\keystores\FCUBSKeyStore.jks

Enter keystore password:<Enter a password to protect the keystore>

Re-enter new password:<Confirm the password keyed above>

What is your first and last name?

  [Unknown]:  cvrhp0729.i-flex.com

What is the name of your organizational unit?

  [Unknown]:  BPD

What is the name of your organization?

  [Unknown]:  Oracle Financial Services

What is the name of your City or Locality?

  [Unknown]:  Mumbai

What is the name of your State or Province?

  [Unknown]:  Maharashtra

What is the two-letter country code for this unit?

  [Unknown]:  IN

Is CN=cvrhp0729.i-flex.com, OU=BPD, O=Oracle Financial Services,
L=Mumbai, ST=Maharashtra, C=IN correct?

  [no]:  yes


Enter key password for <cvrhp0729>

        (RETURN if same as keystore password):<Enter a password to
protect the key>

Re-enter new password:<Confirm the password keyed above>
```

3.2.2 Generating CSR

To purchase an SSL certificate, one needs to generate a Certificate Signing Request (CSR) for the server where the certificate will be installed.

A CSR is generated from the server and is the server's unique "fingerprint". The CSR includes the server's public key, which enables server authentication and secure communication.

If the keystore file or the password is lost and a new one is generated, the SSL certificate and the private key will no longer match. A new SSL Certificate will have to be requested.

The CSR is created by running the following command in the bin directory of the JRE:

```
keytool -certreq -alias alias -file certreq_file -keystore keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. The private key associated with the alias will be utilized to create the CSR. Specify the alias of the key pair created in the previous step.
2. ***certreq_file*** is the file in which the CSR will be stored.
3. ***keystore*** is the location of the keystore containing the public and private key pair.

Example

Listed below is the result of a sample execution of the command

```
D:\Oracle\Weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -certreq -
alias cvrhp0729 -file D:\keystores\certreq.csr -keystore
D:\keystores\FCUBSKeyStore.jks

Enter keystore password:[Enter the password used to access the
keystore]

Enter key password for <cvrhp0729>[Enter the password used to access
the key in the keystore]
```

3.2.3 Obtaining Trusted Certificate from CA

The processes of obtaining a trusted certificate vary from one C A to another. The CA might perform additional offline verification. Consult the CA issuing the certificate for details on the process to be followed for submission of the CSR and for obtaining the certificate.

3.2.4 Importing Certificate into Identity Store

Store the certificate obtained from the CA in the previous step, in a file, preferably in PEM format. Other formats like the p7b file format would require conversion to the PEM format. Details on performing the conversion are not listed here. Please refer to the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server, for details on converting a Microsoft p7b file to the PEM format.

The command to be executed for importing a certificate into the identity store depend on whether the trust store chosen (in the earlier step; see section 2 of this document). It is highly recommended to verify the trust path when importing a certificate into the identity store. The commands provided below assume the use of the Java Standard Trust store.

Importing the Intermediate CA certificate

Most Certificate Authorities do not use the root CA certificates to issue identity certificates for use by customers. Instead, Intermediate CAs issue identity certificates in response to the submitted CSRs.

If the Intermediate CA certificate is absent in the Java Standard Trust store, the trust path for the certificate will be incomplete for the certificate, resulting in warnings issued by Weblogic Server during runtime. To avoid this, the intermediate CA certificate should be imported into the identity keystore. Although the intermediate CA certificate can be imported into the Java Standard Trust store, this is not recommended unless the intermediate CA can be trusted.

The following command should be executed to import the intermediate CA certificate into the keystore.

```
keytool -importcert -alias alias -file cert_file -trustcacerts -keystore
keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. Specify the alias of the key pair used to create the CSR in the earlier step.
2. ***cert_file*** is the location of the file containing the intermediate CA certificate in a PKCS#7 format (PEM or DER file).
3. ***keystore*** is the location of the keystore containing the public and private key pair.


The trustcacerts flag is used to consider other certificates (higher intermediaries and the root CA) in the chain of trust. If no chain of trust is established during verification, the certificate will be displayed and one would be prompted to verify it. It is recommended that due diligence be observed, when the prompt is displayed to verify a certificate when a chain of trust is absent.

Listed below is a sample execution of the command

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
importcert -alias verisigntrialintermediateca -file
D:\keystores\VerisignIntermediateCA.cer -trustcacerts -keystore
D:\keystoreworkarea\FCUBSKeyStore.jks

Enter keystore password:<Enter the password used to access the
keystore>

Certificate was added to keystore
```

Importing the Identity certificate

The following command should be executed to import the identity certificate into the keystore.

```
keytool -importcert -alias alias -file cert_file -trustcacerts -keystore
keystore
```

In the above command,

1. ***alias*** is used to identify the public and private key pair. Specify the alias of the key pair used to create the CSR in the earlier step.

2. ***cert_file*** is the location of the file containing the PKCS#7 formatted reply from the CA, containing the signed certificate.

3. ***keystore*** is the location of the keystore containing the public and private key pair.

The trustcacerts flag is used to consider other certificates (intermediate CAs and the root CA) in the chain of trust. If no chain of trust is established during verification, the certificate will be displayed and one would be prompted to verify it. It is recommended that due diligence be observed, when the prompt is displayed to verify a certificate when a chain of trust is absent.

Listed below is a sample execution of the command

```
D:\Oracle\weblogic11g\jrockit_160_05_R27.6.2-20\bin>keytool -
importcert -alias cvrhp0729 -file D:\keystores\cvrhp0729.cer -
trustcacerts -keystore D:\keystoreworkarea\FCUBSKeyStore.jks

Enter keystore password:<Enter the password used to access the
keystore>

Enter key password for <cvrhp0729>:<Enter the password used to access
the private key>

Certificate reply was installed in keystore
```

The previous set of commands assumed the presence of the appropriate root CA certificate (in the chain of trust) in the Java Standard Trust store, i.e. in the cacerts file. If the CA issuing the identity certificate (for the Weblogic Server) does not have the root CA certificate in the Java Standard Trust store, one can opt to import the root CA certificate into cacerts, or into the identity store, depending on factors including trustworthiness of the CA, necessity of transporting the trust store across machine, among others.

4. Configuring Identity and Trust Stores for Weblogic

4.1 Enabling SSL on Oracle Weblogic Server

To configure SSL on Oracle Weblogic server, login in to the Admin Console and follow the steps given below:

1. Under 'Change Center', click the button 'Lock & Edit'.
2. Expand 'Servers' node.
3. Select the name of the server for which you want to enable SSL (example - exampleserver).
4. Go to 'Configuration' and select General' tab.
5. Select the option 'SSL Listen Port Enabled' and specify the SSL listen port.
6. Against 'Listen Address', specify the hostname of the machine in which the application server is installed.

4.2 Configuring Identity and Trust Stores

To configure the Identity and Trust stores in Oracle Weblogic Server, log in to the Admin Console of Weblogic Server.

1. Under 'Change Center', click the button 'Lock & Edit'.
2. Expand 'Servers' node.
3. Select the name of the server for which you want to configure the keystores (example - exampleserver).
4. Go to 'Configuration' and select 'Keystores' tab.
5. In the filed 'Keystores', select the method for storing and managing private keys/digital certificate pairs and trusted CA certificates. This choice should match the one made in Section 2 of this document (Choosing the Identity and Trust Stores).
6. In the 'Identity' section, provide the following details:
   - **Custom Identity Keystore File Name**: Fully qualified path to the Identity keystore.
   - **Custom Identity Keystore Type**: Set this attribute to JKS, the type of the keystore. If left blank, it is defaulted to JKS (Java KeyStore).
   - **Custom Identity Keystore PassPhrase**: The password you enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. Oracle Weblogic server only reads from the keystore. So whether or not you define this property depends on the requirements of the keystore.
7. In the 'Trust' section, provide the following details:

   If you choose **Java Standard Trust**, specify the password used to access the trust store.

If you choose **Custom Trust**, the following attributes have to be provided:

- **Custom Trust Keystore**: The fully qualified path to the trust keystore.
- **Custom Trust Keystore Type**: Set this attribute to JKS, the type of the keystore. If left blank, it defaults to JKS (Java KeyStore).
- **Custom Trust Keystore Passphrase**: The password you enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. Oracle Weblogic Server only reads from the keystore. So, whether or not you define this property depends on the requirements of the keystore.

When identity and trust stores are of the JKS format, the passphrases are not required.

5. Setting SSL attributes for Managed Servers

5.1 Setting SSL Attributes for Private Key Alias and Password

To configure the private key alias and password, log in to the Oracle Weblogic Server Admin Console.

1. Under '**Change Center**', click the button 'Lock & Edit'.
2. Expand '**Servers**' node.
3. Select the name of the server for which you want to configure keystores (example - exampleserver).
4. Go to '**Configuration**' and select '**SSL**' tab.
5. Select 'Keystores' from '**Identity and Trust Locations**'.
6. Under 'Identity' section, specify the following details:
   - **Private Key Alias**: set this attribute to the alias name defined for the key pair when creating the key pair in the Identity keystore.
   - **Private Key Passphrase**: The password defined for the key pair (alias_password), at the time of its creation. . Confirm the password.
7. Click '**Save**'.
8. Under '**Change Center**', click '**Activate changes**'.
9. Go to **controls** tab, check the appropriate server and click '**Restart SSL**'. Confirm when it prompts.

6. Testing Configuration

6.1 Testing Configuration

Once the Oracle Weblogic has been configured for SSL, deploy the application in the usual manner. After deployment, you can test the application in SSL mode. To launch the application in SSL mode you need to enter the URL in the following format:

https://(Machine Name):(SSL_Listener_port_no)/(Context_root)

It is essential that the Oracle Banking payments web application be accessed via the HTTPS channel, instead of the HTTP channel.

7. Creating Resources on Weblogic

7.1 Introduction

This document explains the steps to be executed to deploy the FC payments application and gateway application in application server.

7.2 Resource Administration

This section deals with the process of resource administration on Oracle Weblogic.

All the resources mention in “Resources To be Created” document are need to be created before deployment. One example for each category is explained in the following subsections.

7.2.1 Creating Data Source

The method for creating data sources is explained under the following headings.

7.2.1.1 Prerequisites

You need to create the data source with OCI enabled. For this, download Oracle Instant Client and install it. The details are given below.

Package	Download Location	Remarks
Oracle Instant Client Package	http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html	Install Oracle Instant Client in a local directory. While configuring Weblogic for Windows or Unix/Linux box, you need to provide the directory path where Instant Client is installed.

You need to do the data source configuration with OCI driver enabled. The configurations are given below.

- Oracle Weblogic on Windows Box:
  - Set *{ORACLE_HOME}* in the environment variable.
  - Update the Environment Variable Path as *{ORACLE_HOME}/Instance Client*. This is required to load all the *.dll* files.
  - Ensure that the *ojdbc*.jar* file in *{WL_HOME}/server/lib/ojdbc*.jar* is the same as the file *{ORACLE_HOME}/jdbc/lib/ojdbc*.jar*. This is required for ensuring compatibility.
  - Update PATH in *StartWebLogic.bat* or in *setDomainEnv.bat.* This must be the path of directory where Oracle Instant Client is installed.
- Oracle Weblogic on Unix/Linux Box:

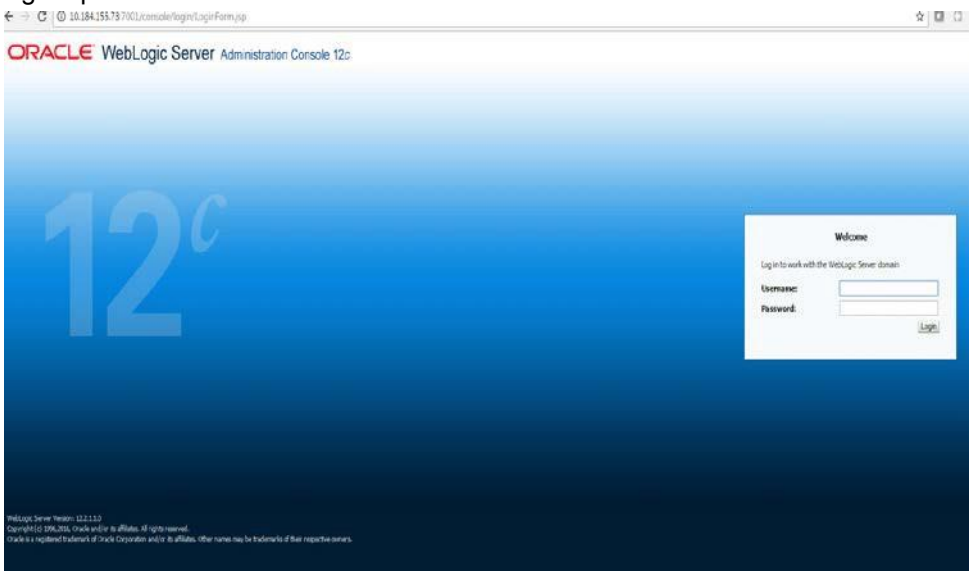
- Set *{ORACLE_HOME}* in the environment variable.
- Update the environment variable *LD_LIBRARY_PATH* as *{ORACLE_HOME}/lib*. This is to load all the *.so* files.
- Ensure that the *ojdbc*.jar* file in *{WL_HOME}/server/lib/ojdbc*.jar* is the same as the file *{ORACLE_HOME}/jdbc/lib/ojdbc*.jar*. This is to ensure compatibility.
- Update *LD_LIBRARY_PATH* in *StartWeblogic.sh* or in *setDomainEnv.sh.* This must be the path of directory where Oracle Instant Client is installed.
- If you are still not able to load the *.so* files, then you need to update the *EXTRA_JAVA_PROPERTIES* by setting *Djava.library.path* as *{ORACLE_HOME}/lib* in *StartWebLogic.sh* or in *setDomainEnv.sh.*

7.2.1.2 XA Enabled Data Source

Follow the steps given below:

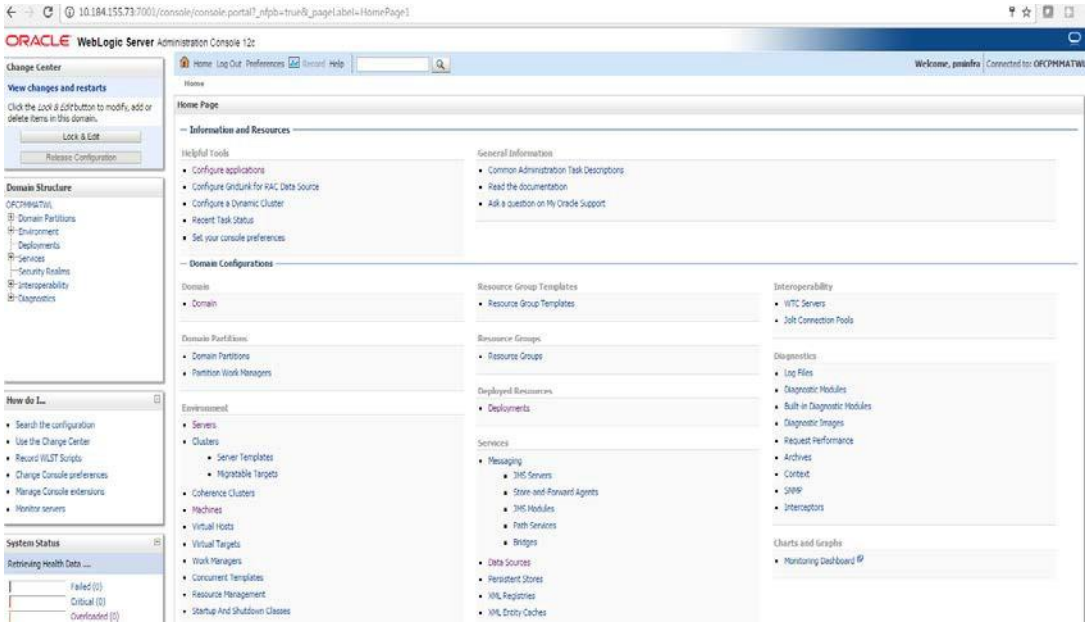
1. Start the Administrative Console of Weblogic application server. You can start this by entering Oracle Weblogic Admin Console URL in the address bar in an internet browser.

   http://10.10.10.10:1001/console
   Eg: http://10.10.10.10:1001/console



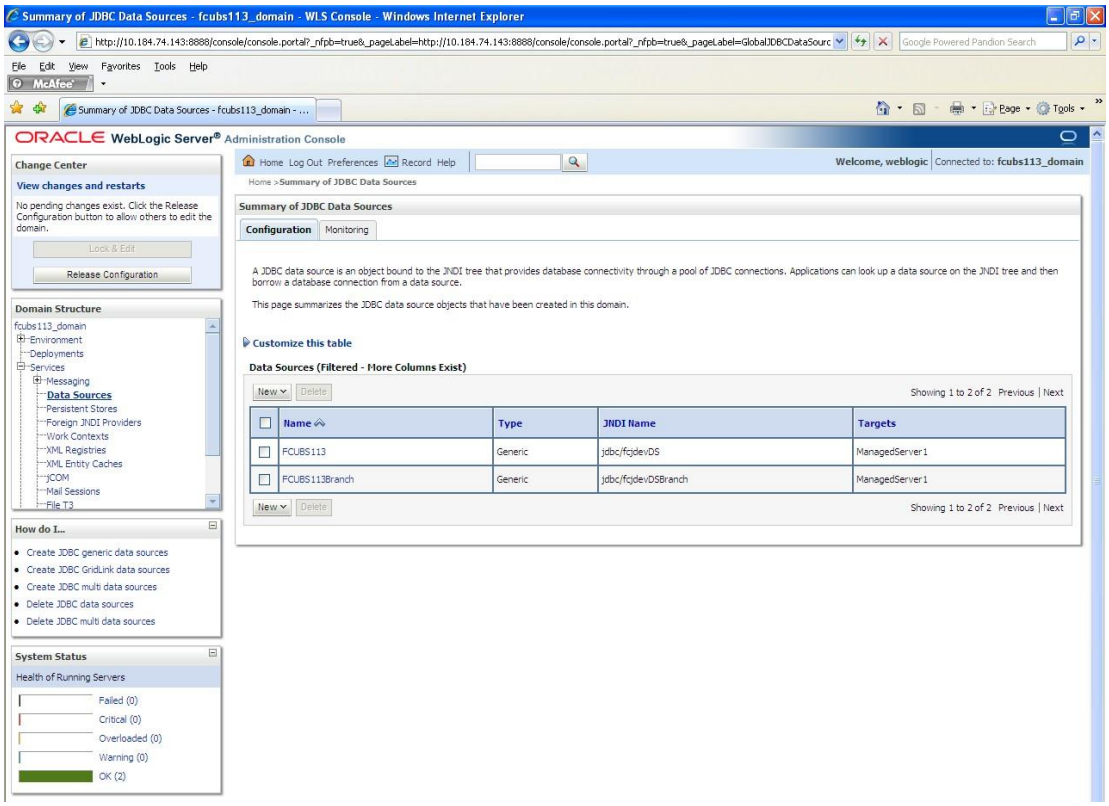
2. Specify the Weblogic administrator user name and password. Click 'Log In'.

3. Navigate to Oracle Weblogic home page.

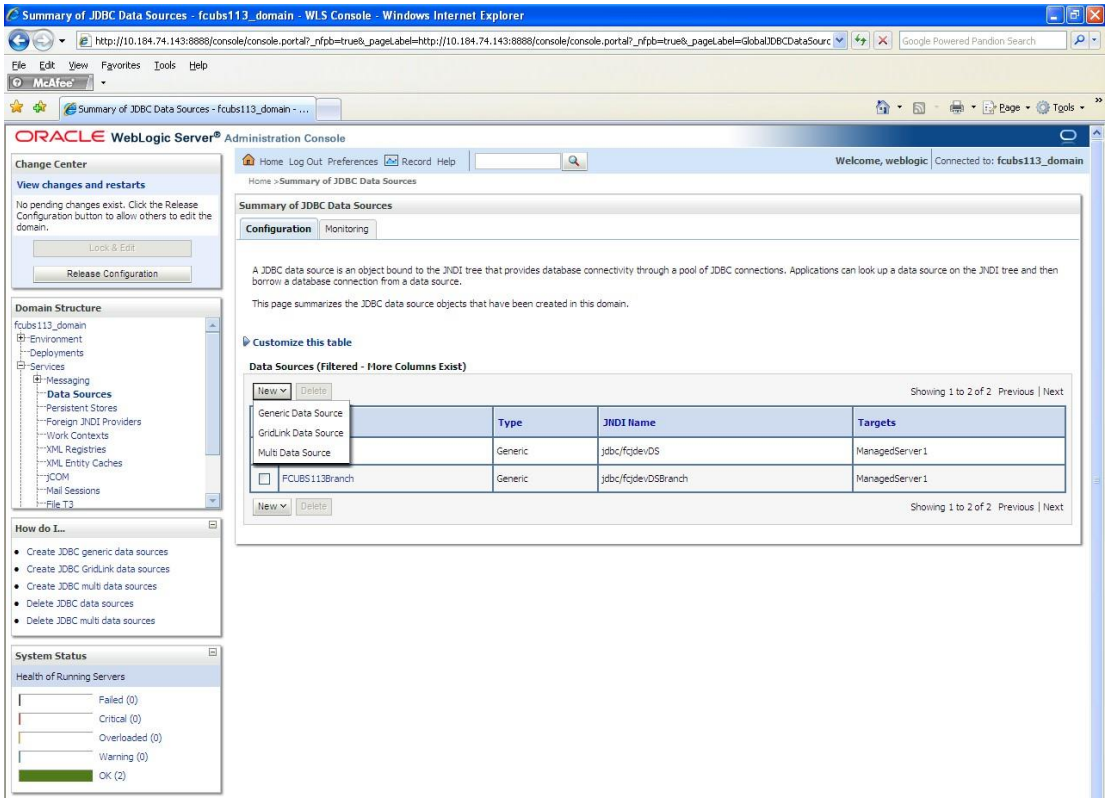


4. Click 'LOCK & EDIT'.

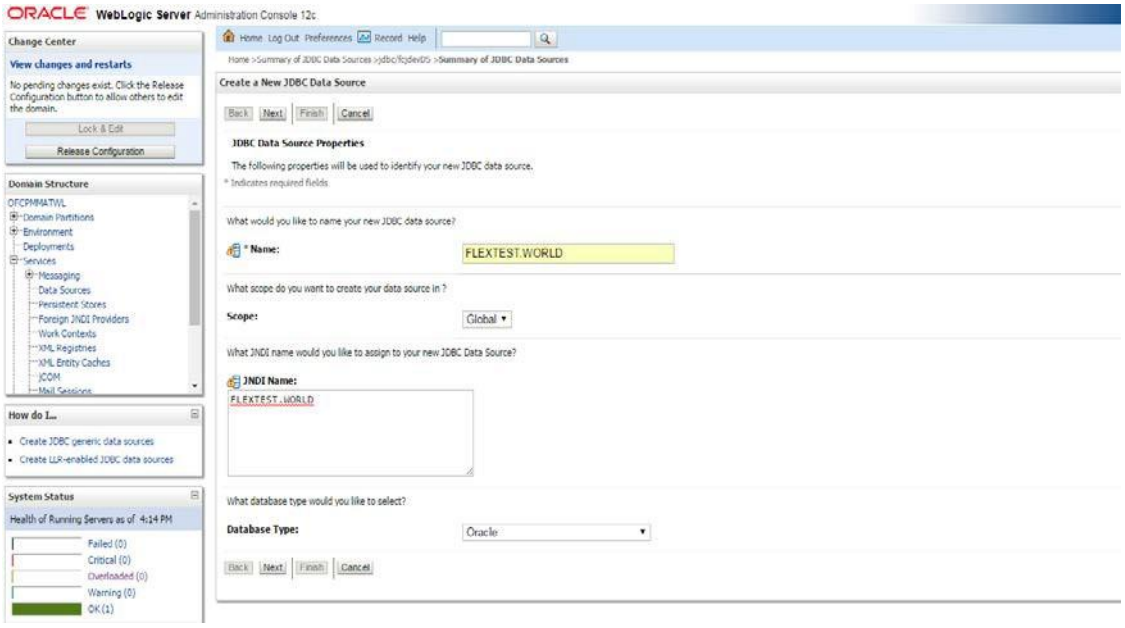
Following screen is displayed:



5. Expand ‘Services’ and then ‘Data Sources’ under it. Click ‘Lock & Edit’ button.



6. To create a new data source, click ‘New’ and select ‘Generic Data Source’. The following screen is displayed.

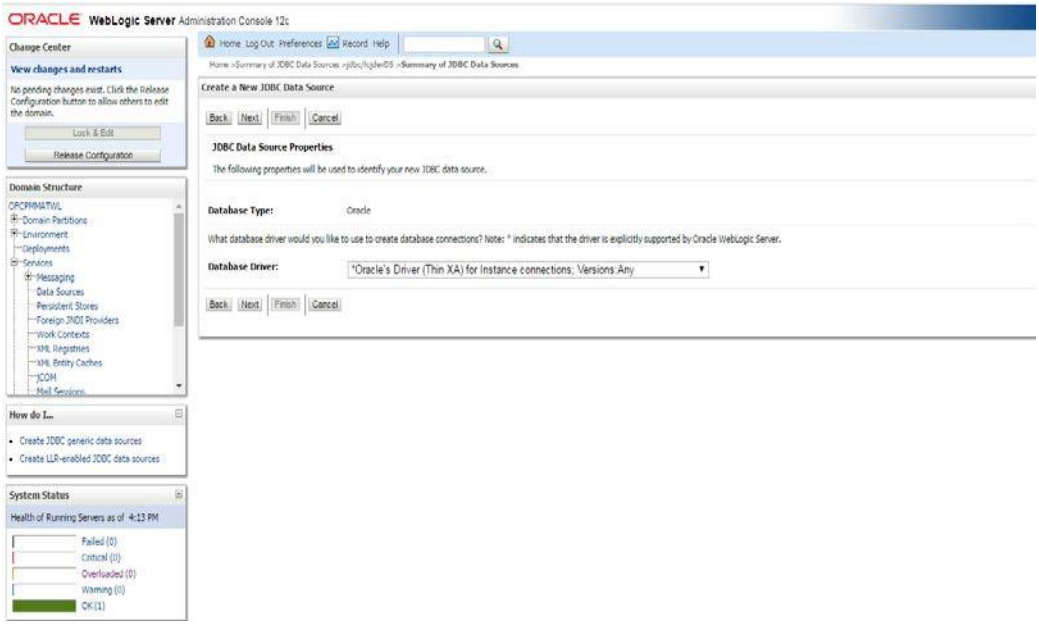


7. Specify the following details:

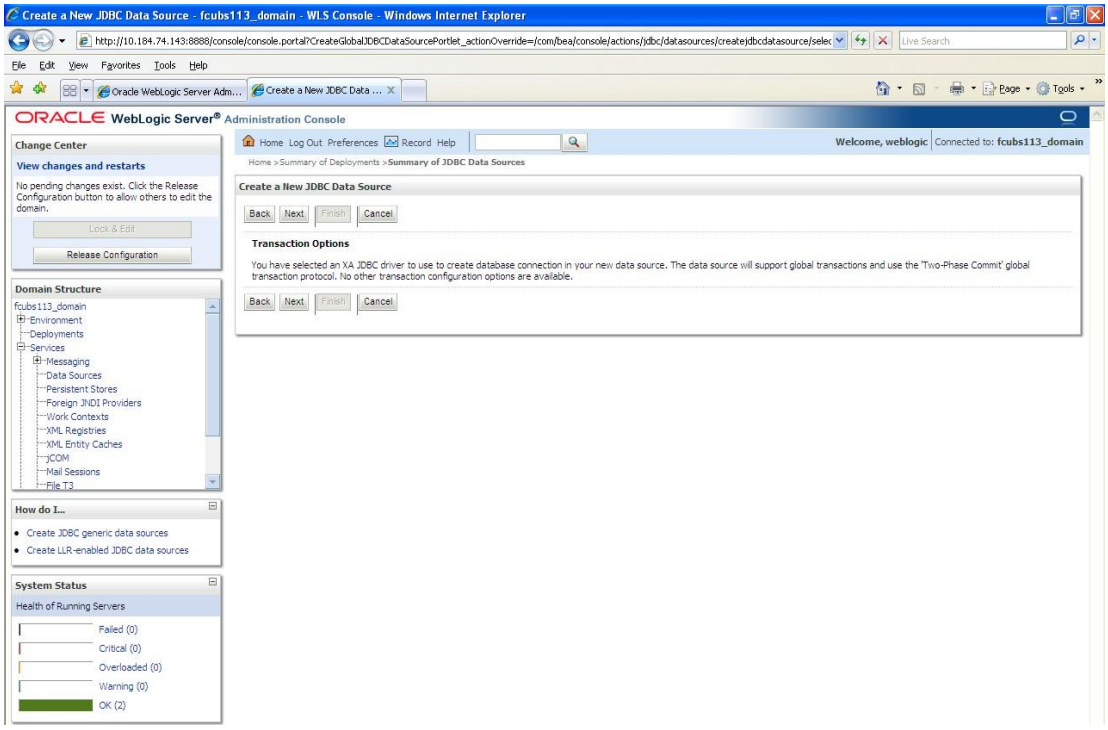
JDBC Datasource Name	Name of the data source
JNDI Name	JNDI name which will be used for lookup
Database Type	Type of the database which is Oracle

8. Click ‘Next’.

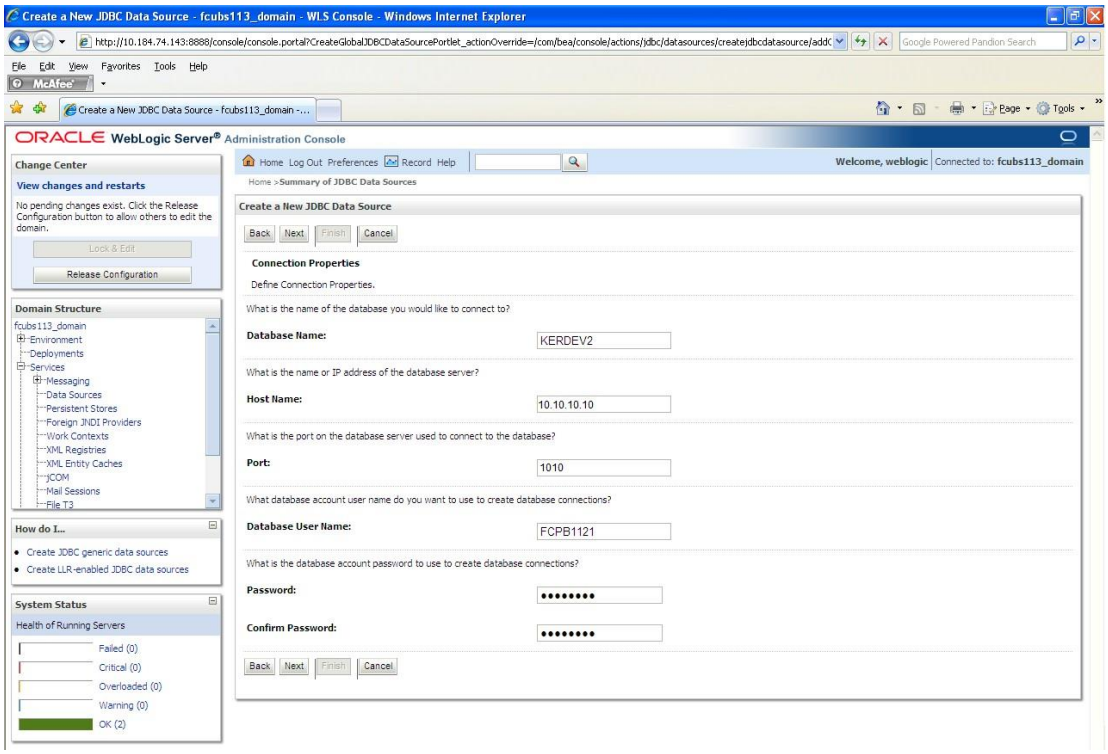
   The following screen is displayed:



9. Select the database driver as shown in the figure. Click ‘Next’.

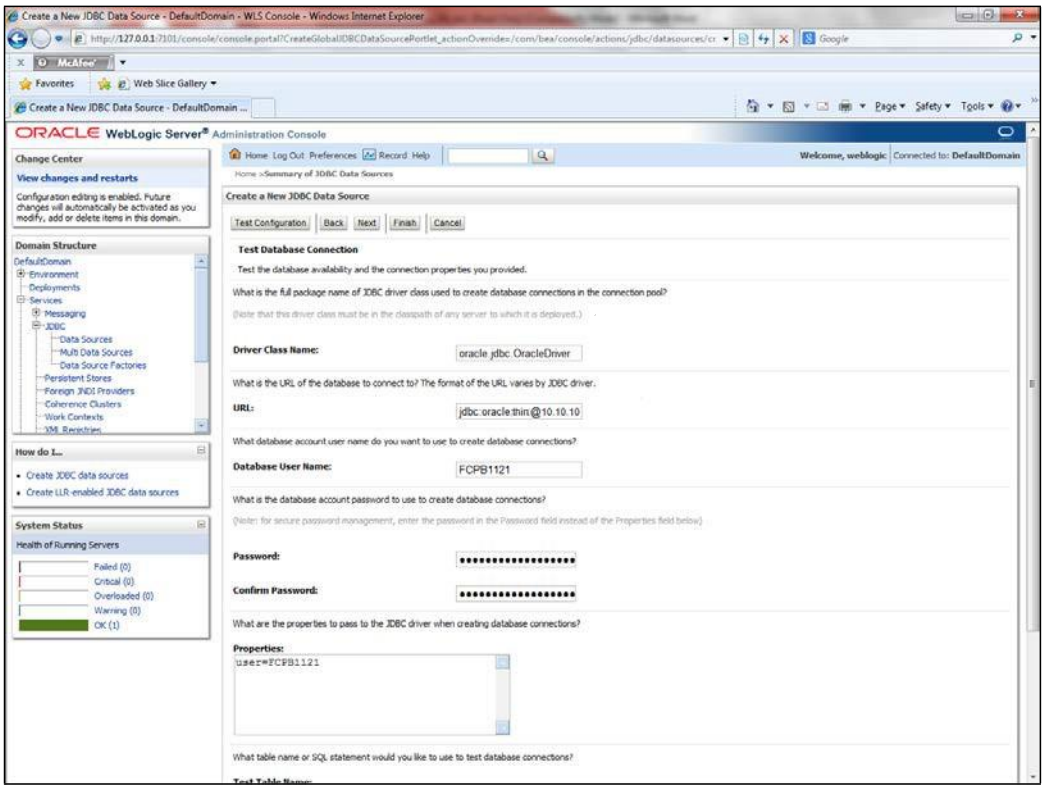


10. Specify the Database Name, Host Name, Port of the database server to connect, Database User Name and Password. Confirm the password.



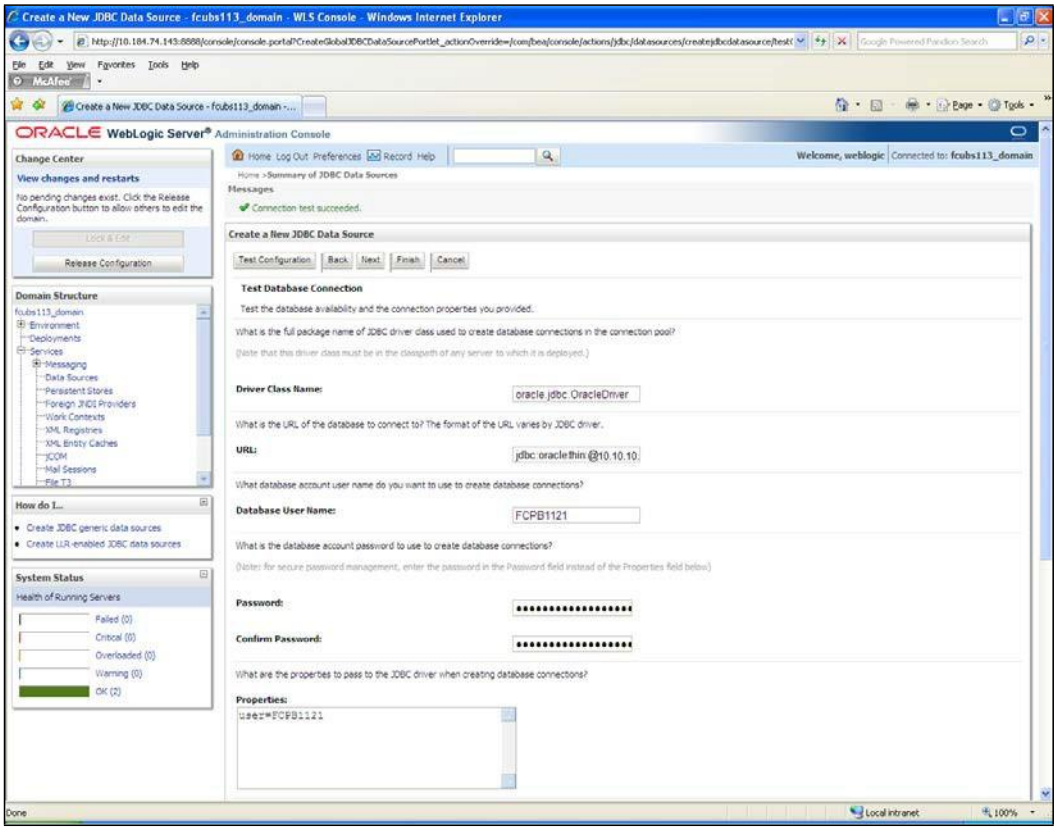
11. Click ‘Next’.

    The following screen is displayed.



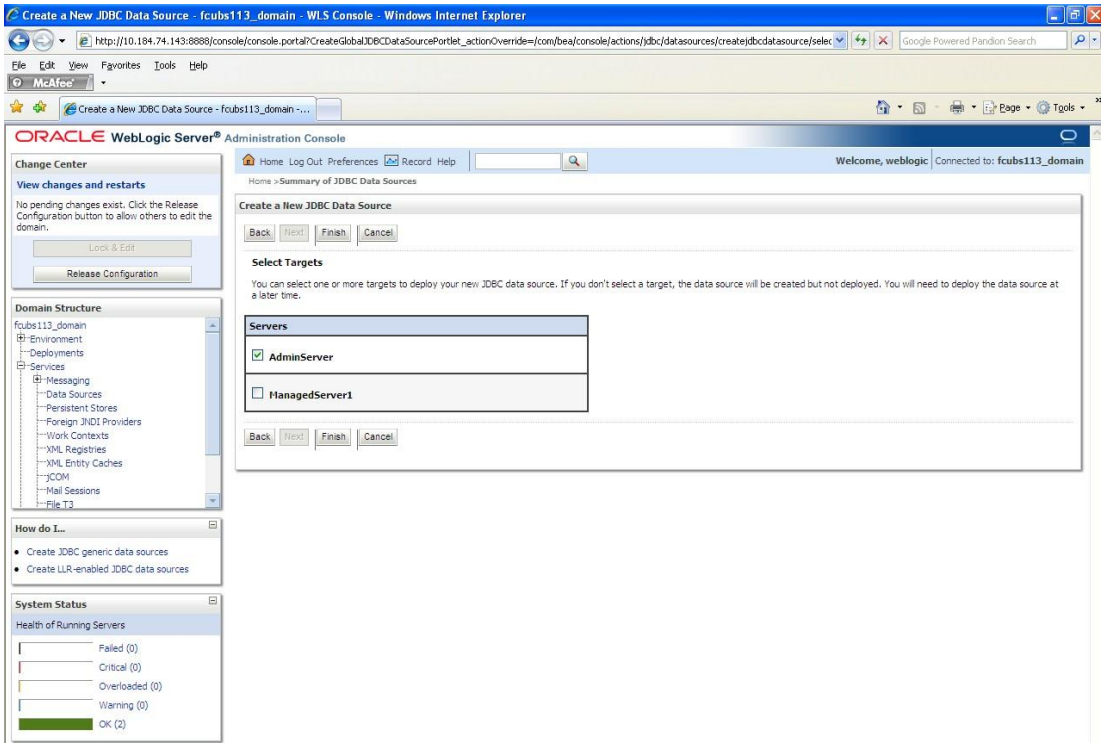
12. Specify the Driver Class Name (Eg: oracle.jdbc.OracleDriver)
13. Specify the URL.
14. jdbc:oracle:thin:@10.10.10.10:1001:<INSTANCE_NAME>Specify the Database Username (Eg: FCPB1121) and password.
15. Confirm the password.
16. Click ‘Test Configuration’ tab.

If the connection is established successfully, the message ‘Connection test succeeded’ is displayed.

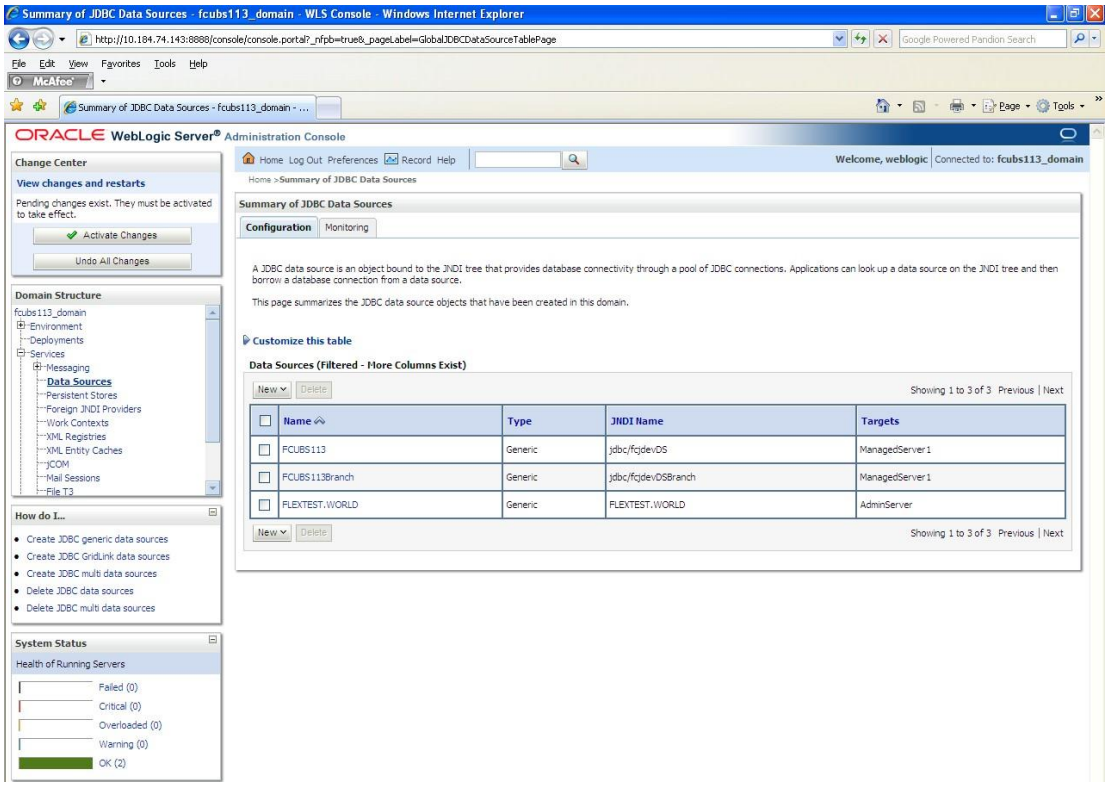


17. Click ‘Next’.

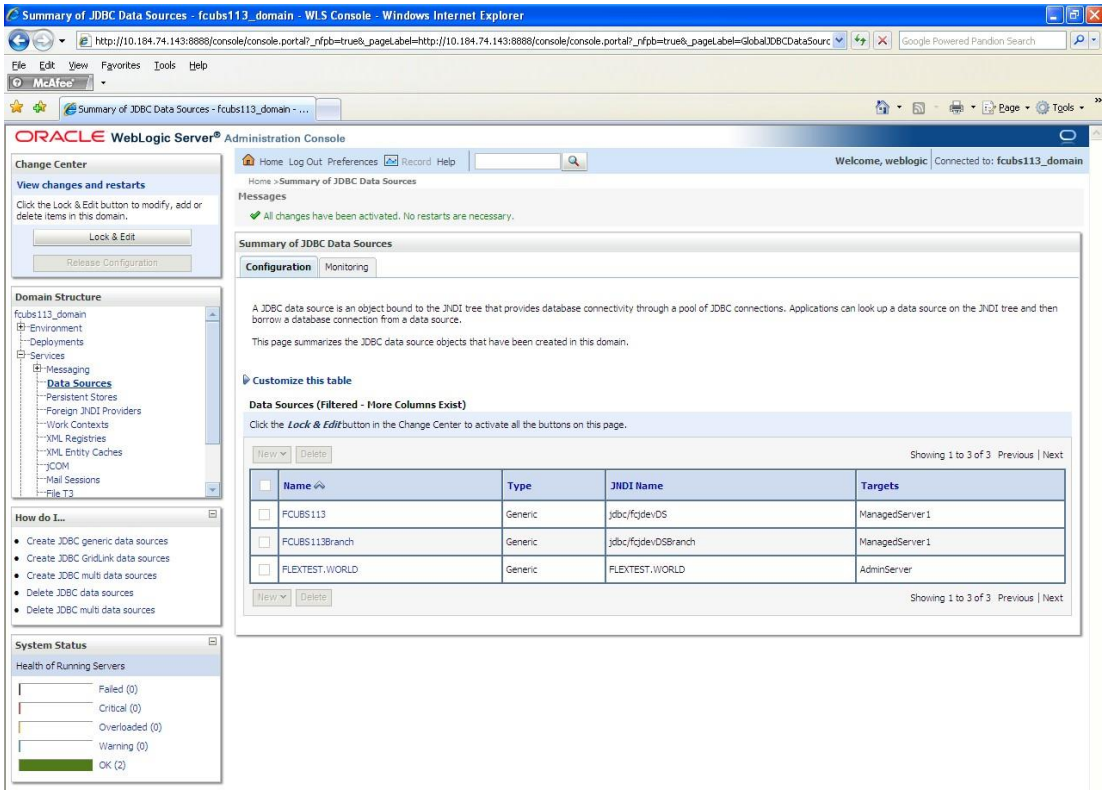
The following screen is displayed:



18. Check the boxes against the required servers. Click ‘Finish’. The following screen is displayed:



19. Click ‘Activate Changes’ button. Click ‘Activate Changes’ button on the left pane. The message ‘All the changes have been activated. No restarts are necessary’ is displayed.

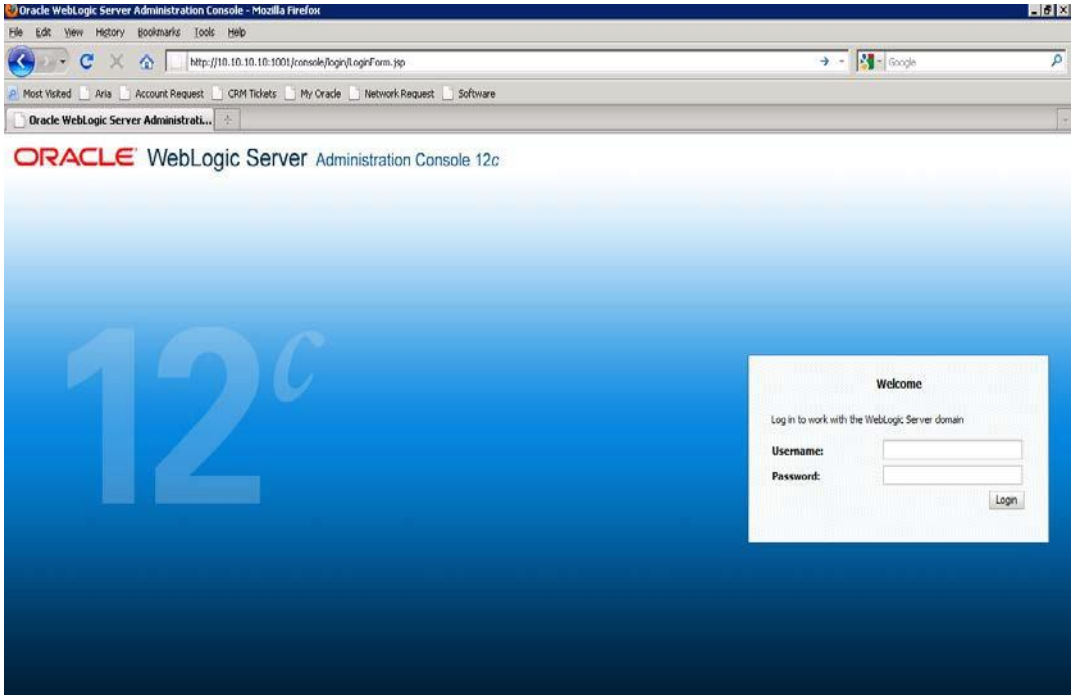


20. The datasource has been created.

21. Refer to “Resources_To_ Be_Created.doc” for the list of XA datasources to be created.

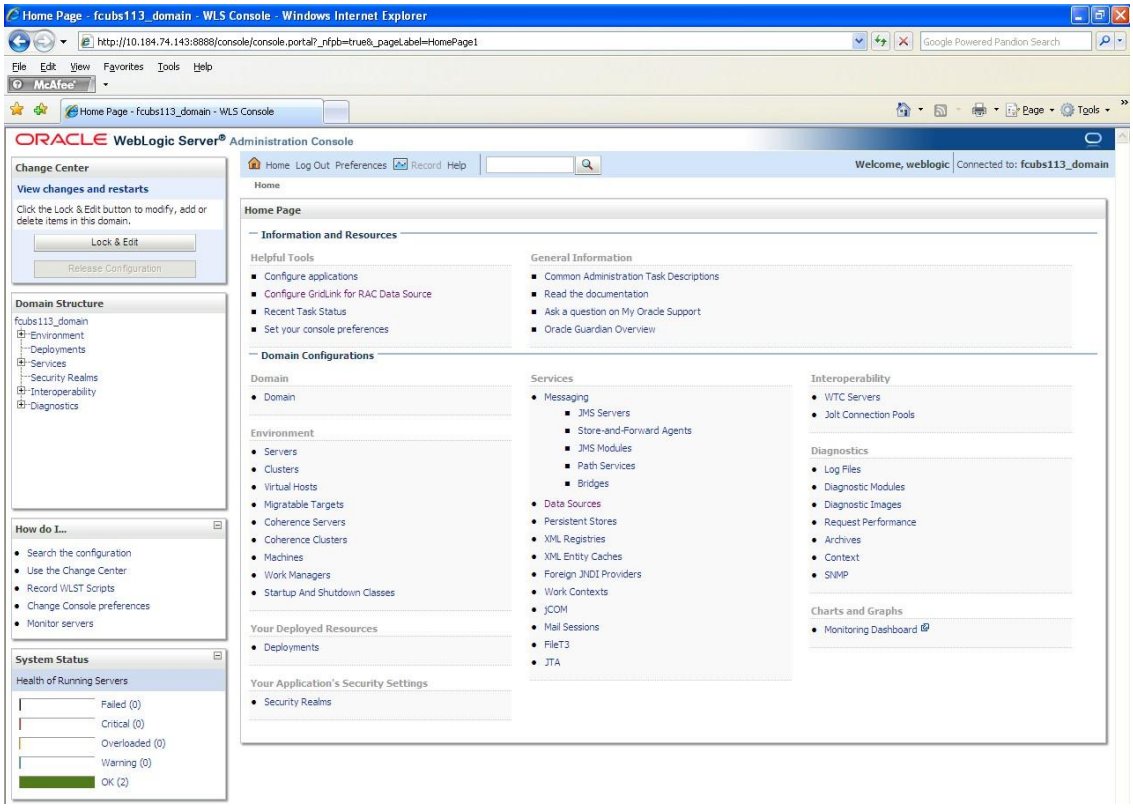
7.2.1.3 Non-XA Enabled Data Source

1. Follow the steps given below:Start the Administrative Console of Weblogic application server. You can start this by entering Oracle Weblogic Admin Console URL in the address bar in an internet browser.
   http:10.10.10.10:1001/console Eg: http://10.10.10.10:1001/console

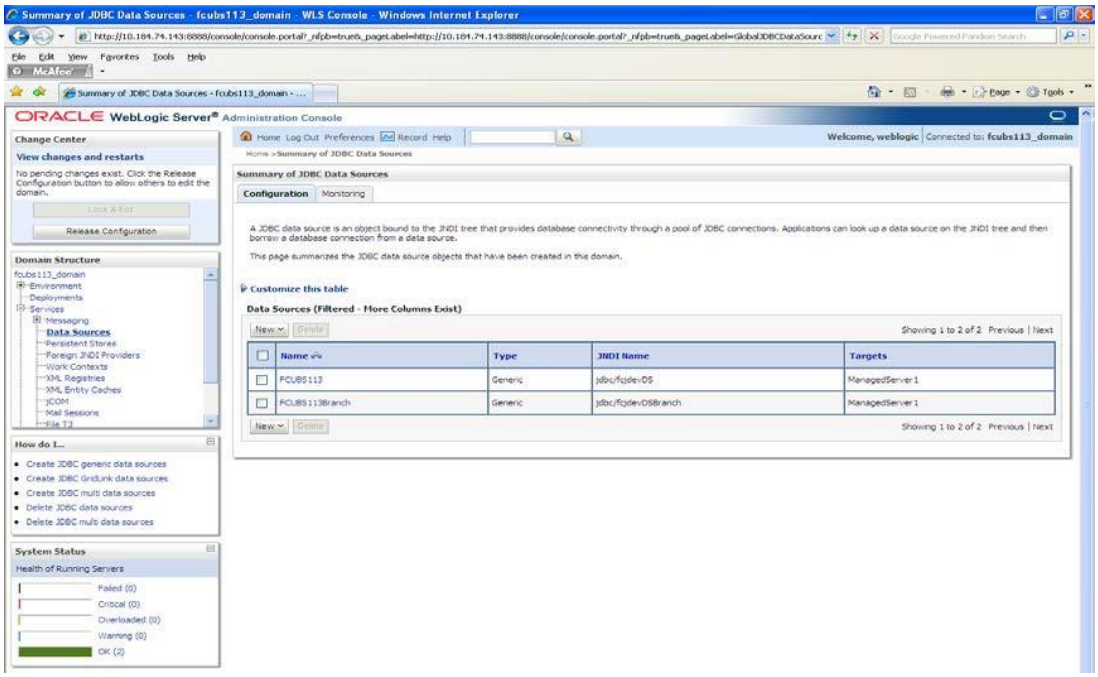


2. Specify the Weblogic administrator user name and password. Click ‘Log In’.

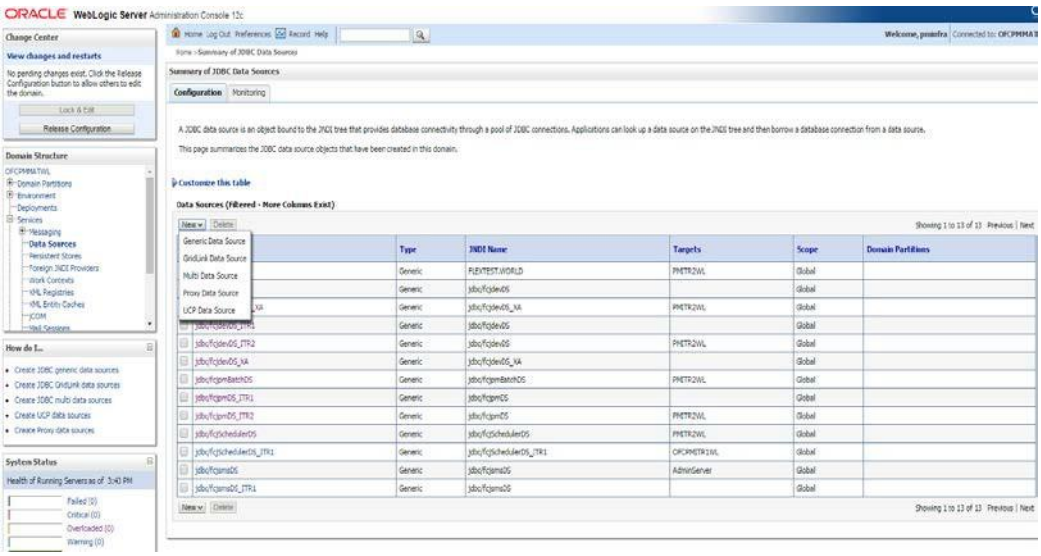
3. Navigate to Oracle Weblogic home page.



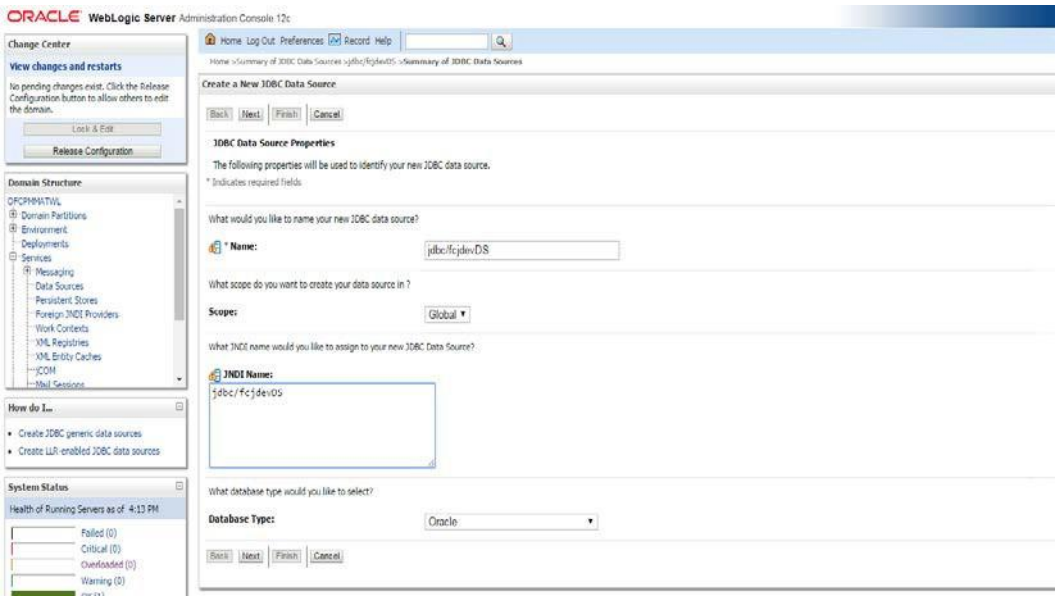
The following screen is displayed:



4. Expand ‘Services’ and then ‘Data Sources’ under it. Click ‘Lock & Edit’ button.



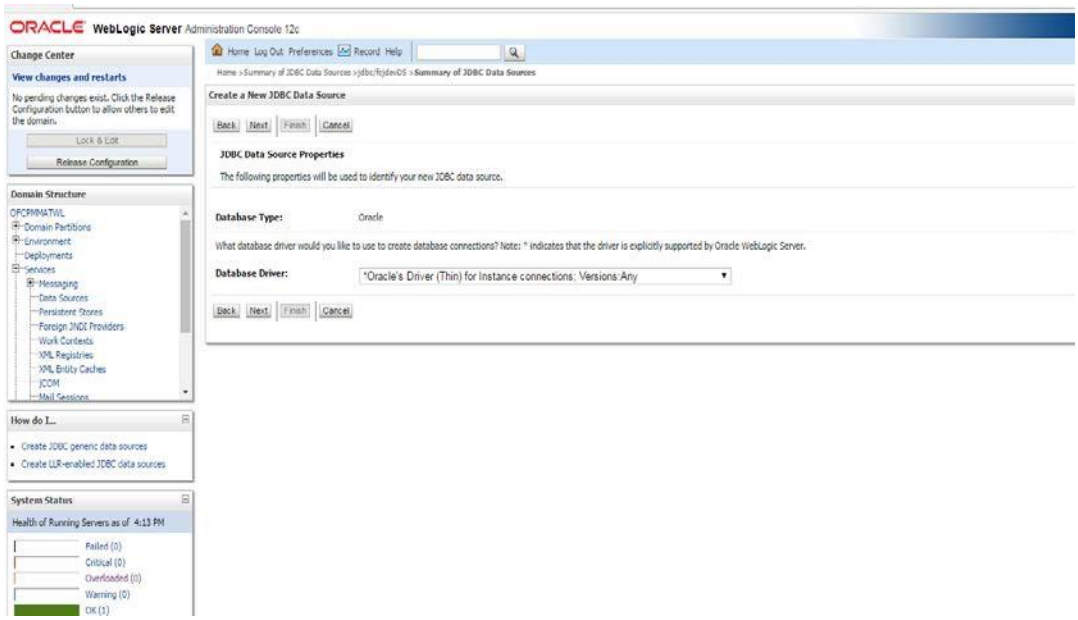
5. To create a new data source, click ‘New’ and select ‘Generic Data Source’.



6. Specify the following details:

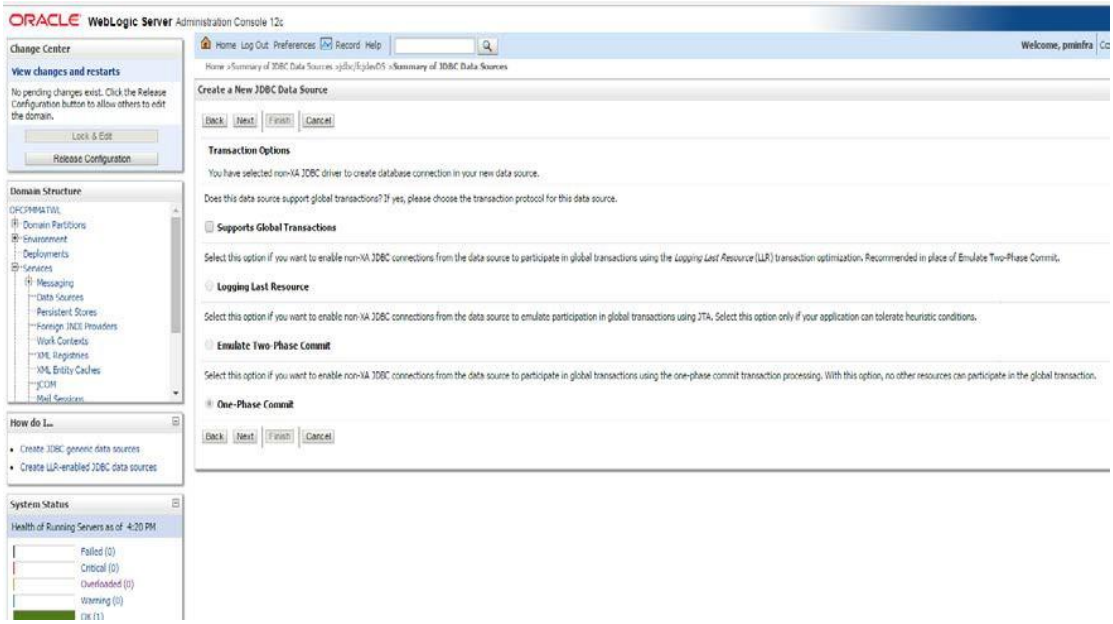
JDBC Datasource Name	Name of the Datasource
JNDI Name	JNDI for lookup
Database Type	Oracle

7. Click ‘Next’.

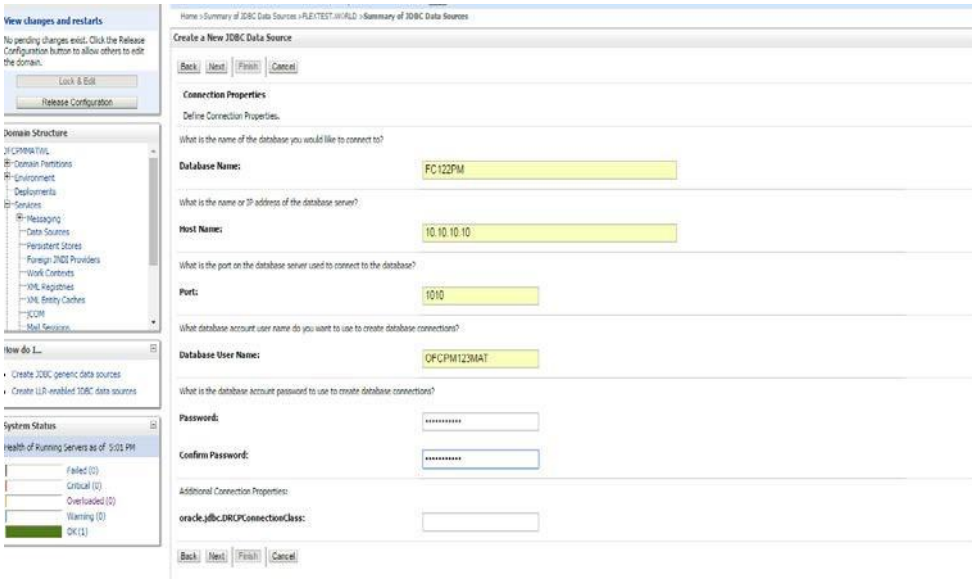


8. Select the database driver as shown in the figure. For Payments Online datasource, check óSupport Global Transactionsôand select óLogging Last Resourceô

   Following screen is displayed:



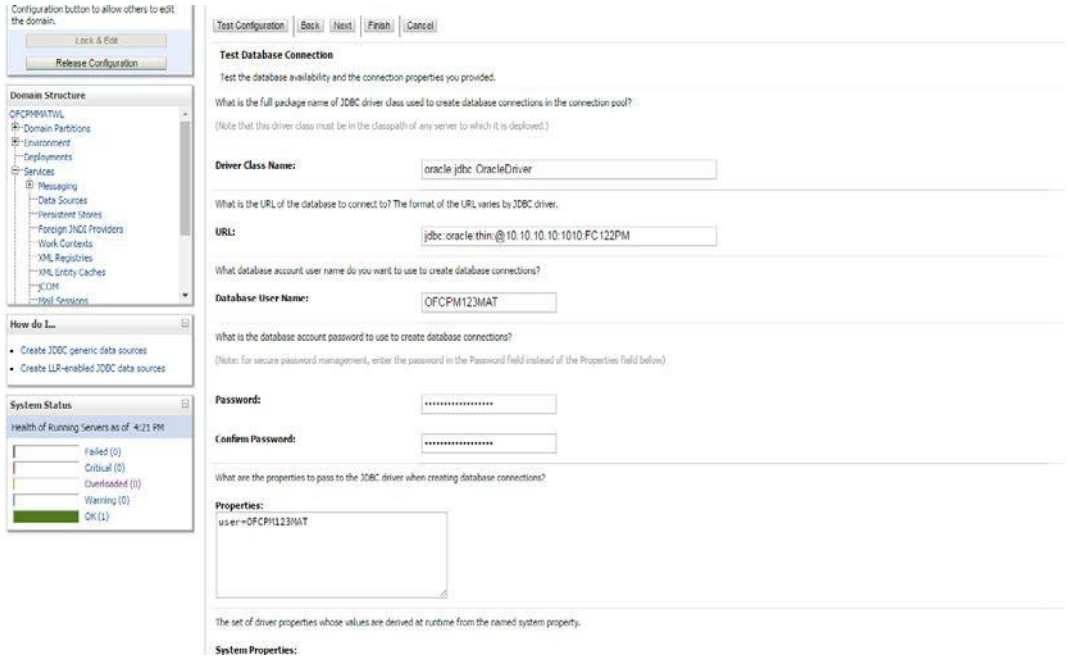
9. For other datasources,click ‘Next’. The following screen is displayed:



10. This screen defines the connection properties. Set the details as given below:

11. Specify the Database Name, Host Name, Port of the database server to connect, Database User Name and Password. Confirm the password.

12. Click ‘Next’. The following screen is displayed.

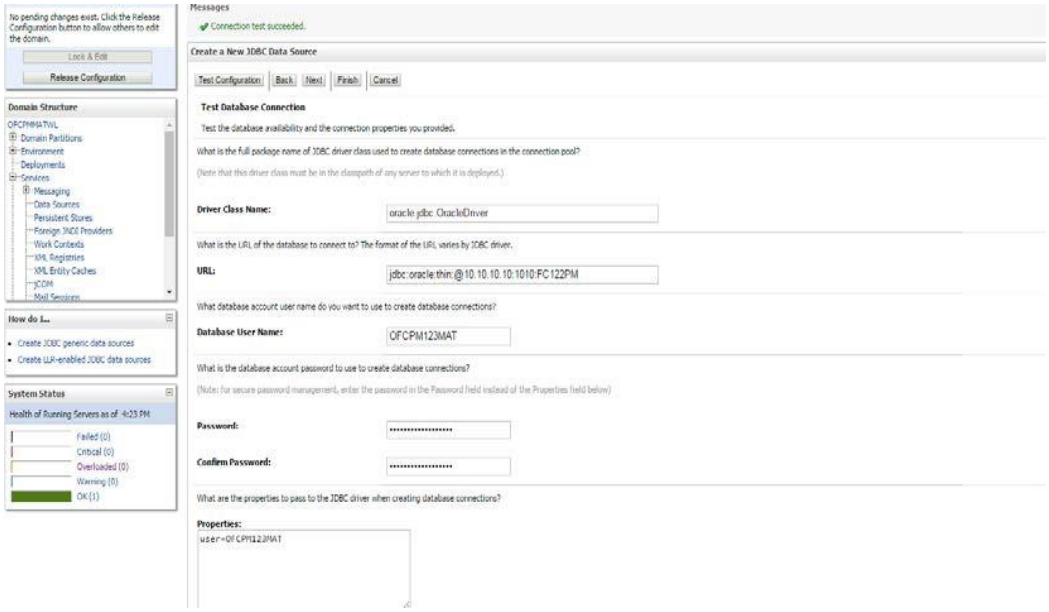


13. Specify the Driver Class Name (Eg: oracle.jdbc.OracleDriver)

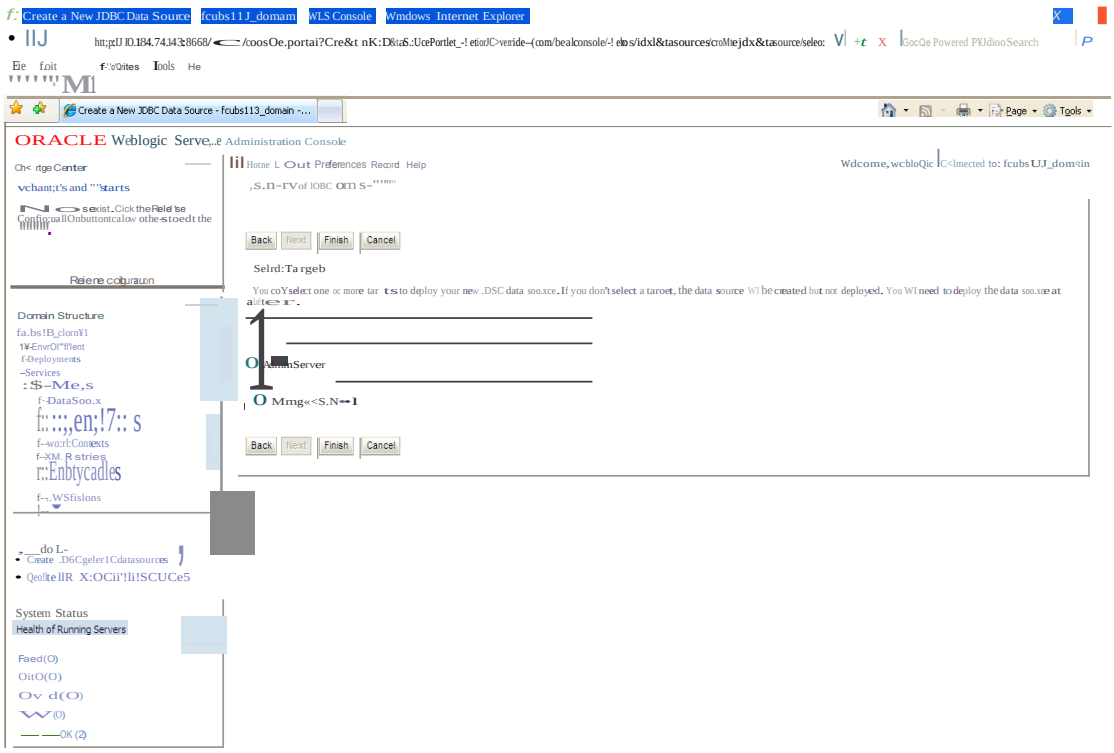
14. Specify the URL.

Default URL: jdbc:oracle:thin:@10.10.10.10:1001:<INSTANCE_NAME>.
Change the default URL to: jdbc:oracle:oci:@10.10.10.10:1010:<INSTANCE_NAME>

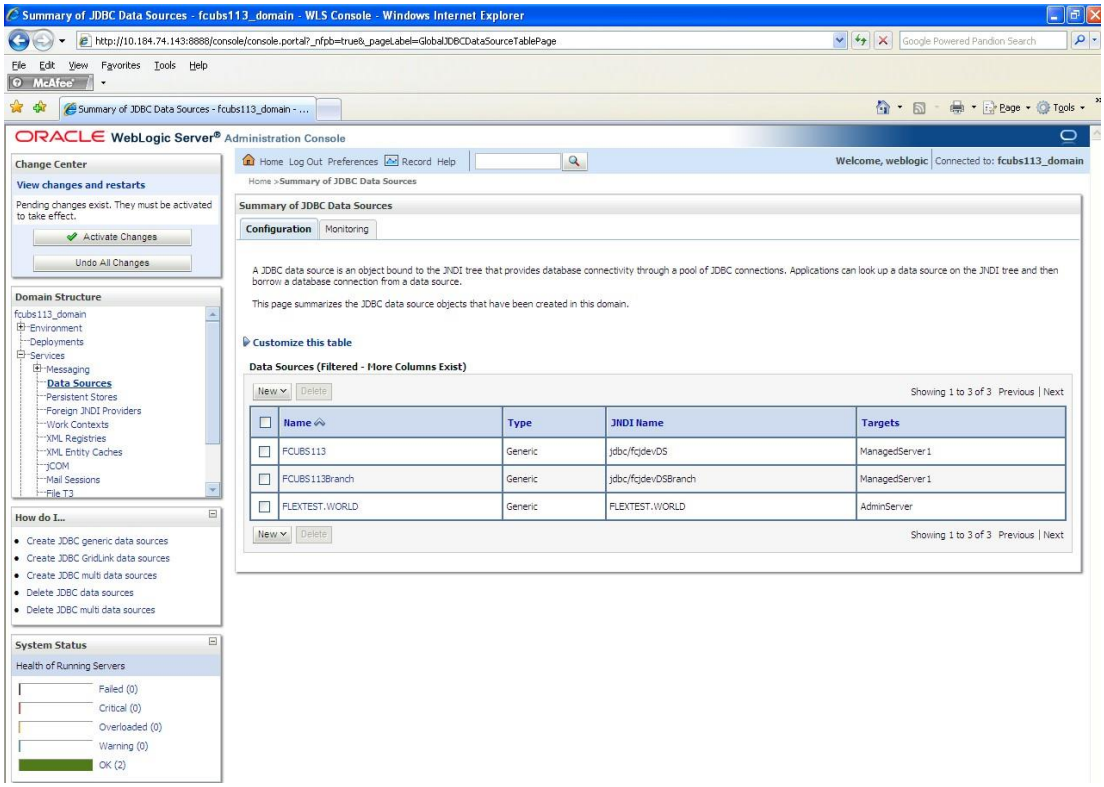
15. Specify the Database Username (Eg: testdb) and password.
16. Confirm the password.
17. Click 'Test Configuration' tab.
18. If the connection is established successfully, the message 'Connection test succeeded' is displayed.



19. Click 'Next'. The following screen is displayed:

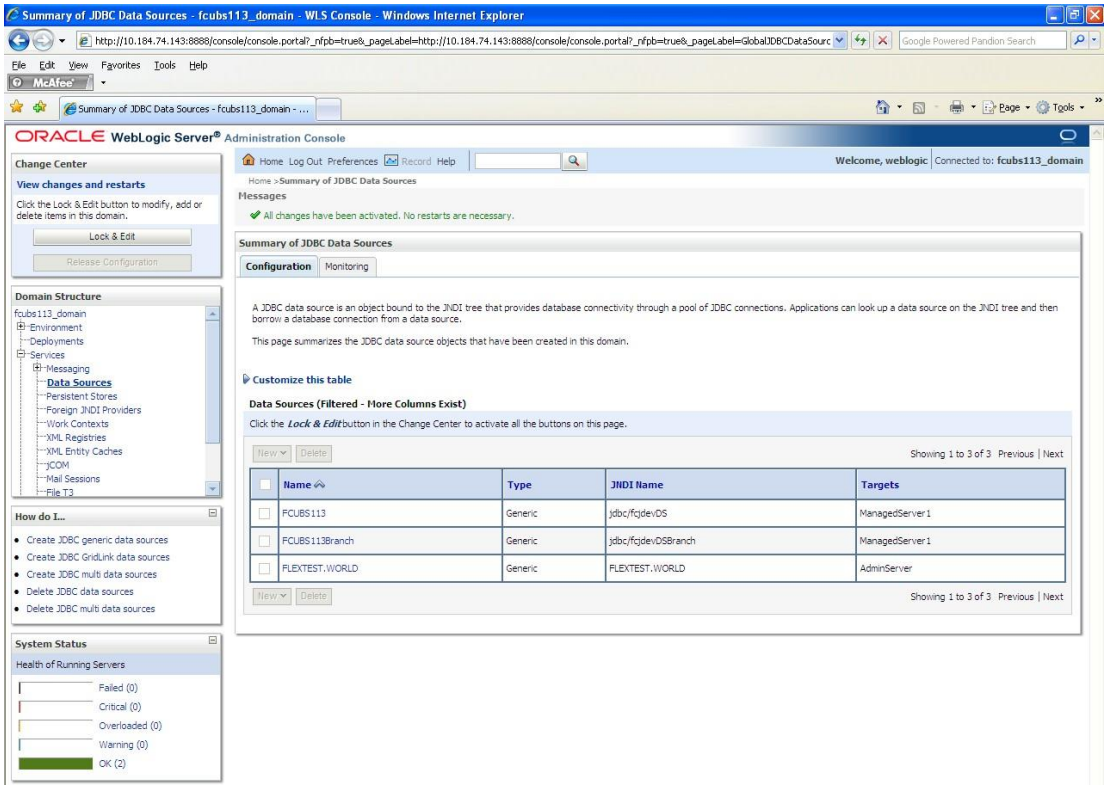


20. Check the boxes against the required servers. Click ‘Finish’. The following screen is displayed:



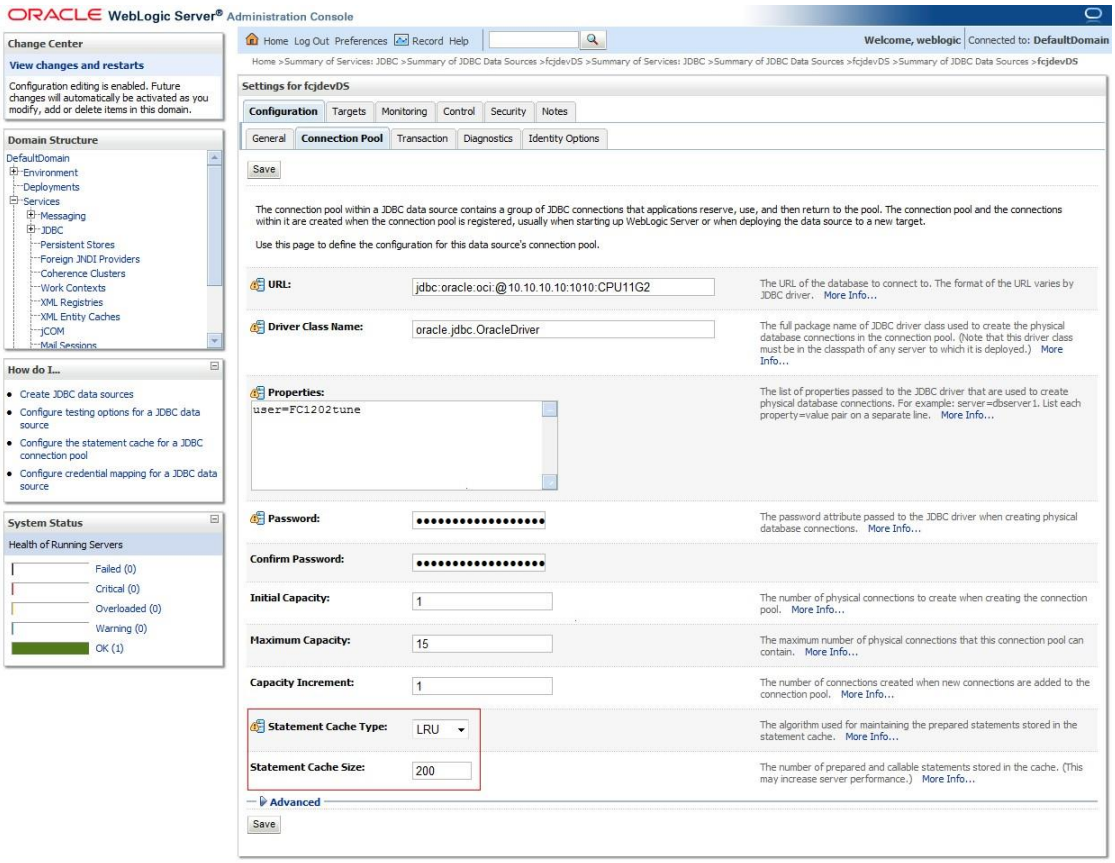
21. Click ‘Activate Changes’ button. Click ‘Activate Changes’ button on the left pane.

The message ‘All the changes have been activated. No restarts are necessary’ is displayed.



22. ‘FCUBSDS’ datasource is created.

23. Click the datasource, and then click on the Connection Pool tab.



24. Select the statement cache type as ‘LRU’.
25. Specify the statement cache size as ‘200’.
26. Click ‘Save’.
27. Refer to “Resources_To_ Be_Created.doc” for the list of Non-XA datasources to be created.

Note the following

- You need to create another data source for Oracle FCpayments with the JNDI name ‘<Non-XA FCUBS HOST JNDI name>_ASYNC’ for batch process. For example, if the Oracle FCUBS HOST Non XA data source JNDI name is ‘jdbc/fcjdevDS’, then you need to create another data source for FCUBS with the JNDI name ‘jdbc/fcjdevDS_ASYNC’.
- While creating a branch using the ‘Branch Parameters Maintenance’ (STDBRANC) screen, if you have created a data source for the branch, then you need to create a corresponding ASYNC data source with the JNDI name ‘<Non-XA FCpayments BRANCH JNDI name>_ASYNC’.

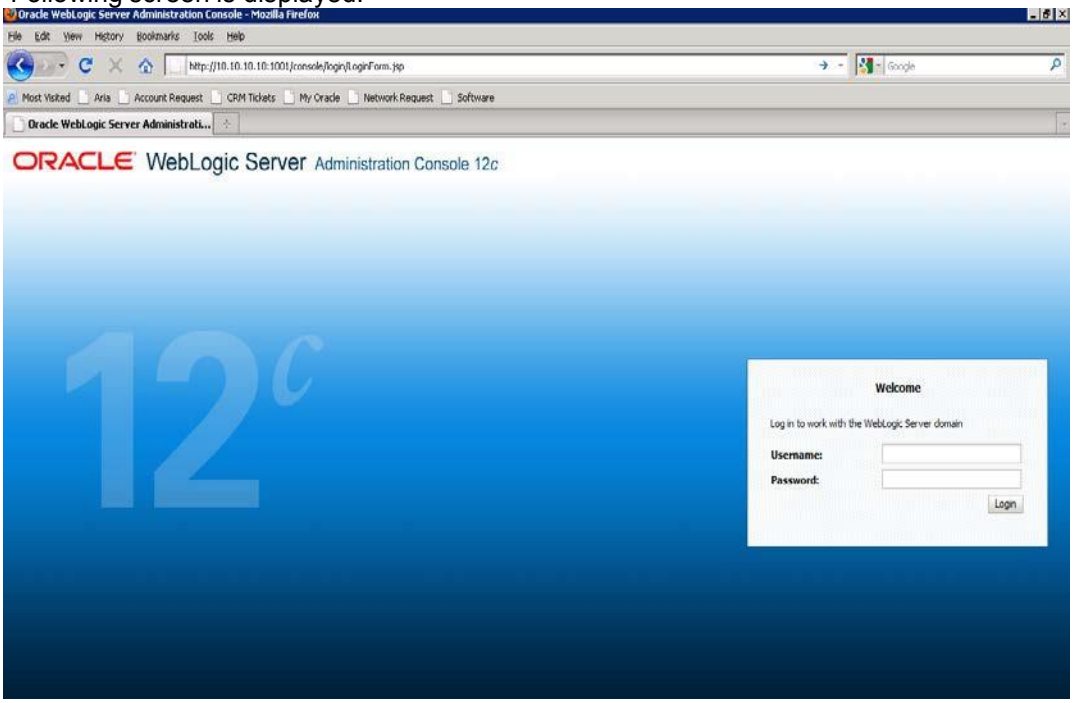
7.2.2 JMS Server Creation

Follow the steps given below:

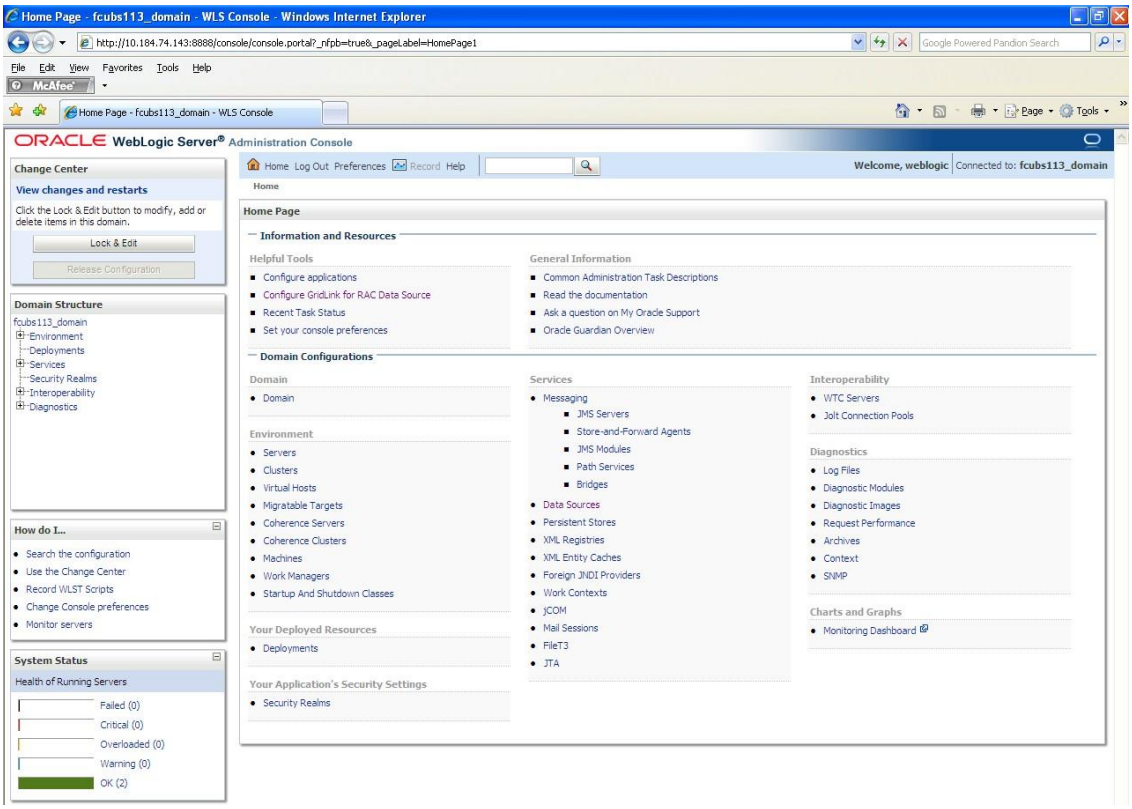
1. Start the Administrative Console of Weblogic application server. You can start this by entering Oracle Weblogic Admin Console URL in the address bar in an internet browser.

http://10.10.10.10:1001/console Eg: http://10.10.10.10:1001/console

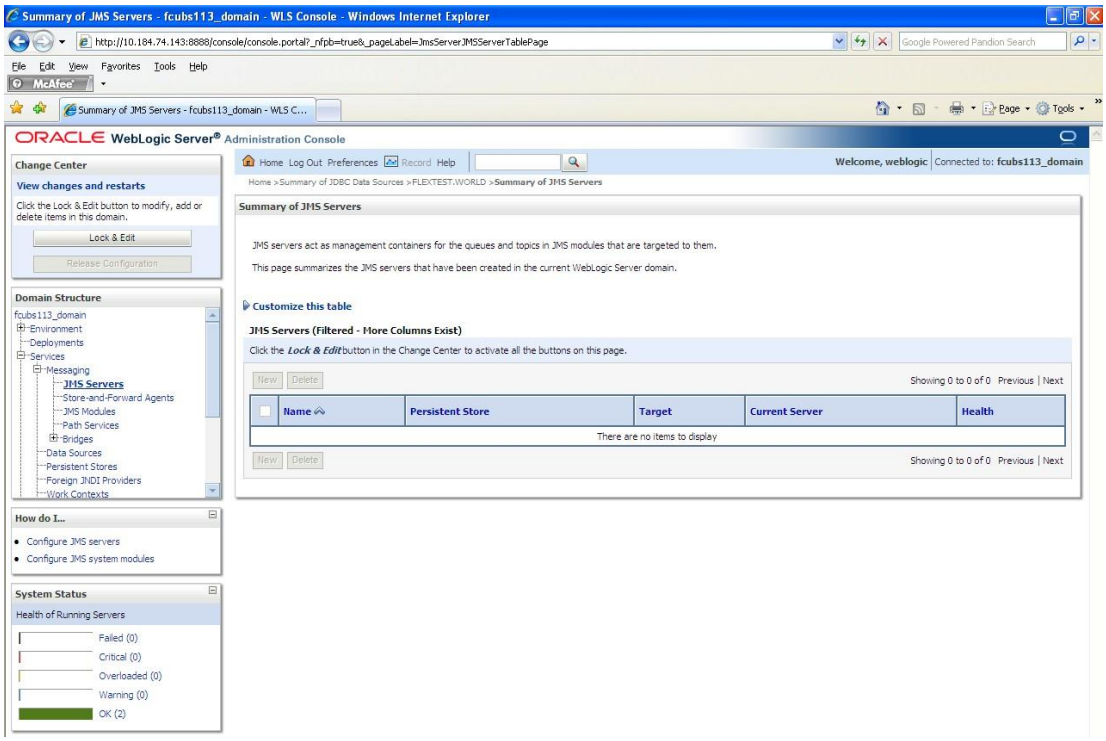
Following screen is displayed:



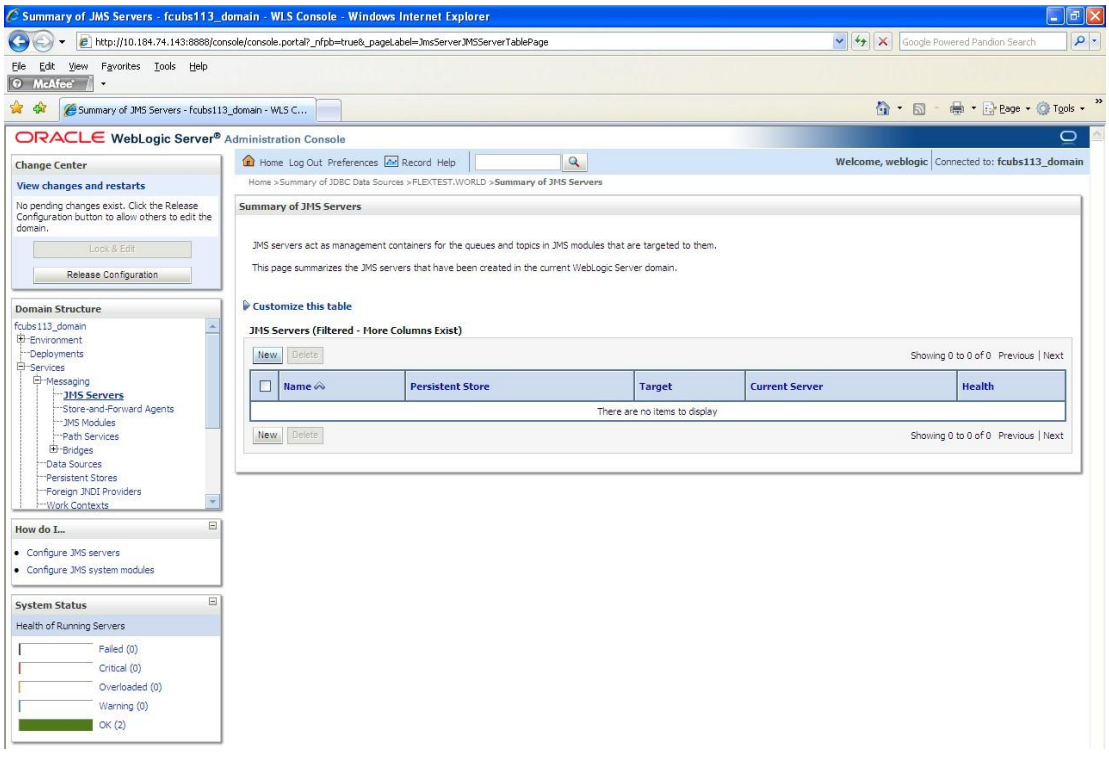
2. Specify the Weblogic administrator user name and password. Click ‘Log In’.
3. Navigate to Oracle Weblogic home page.



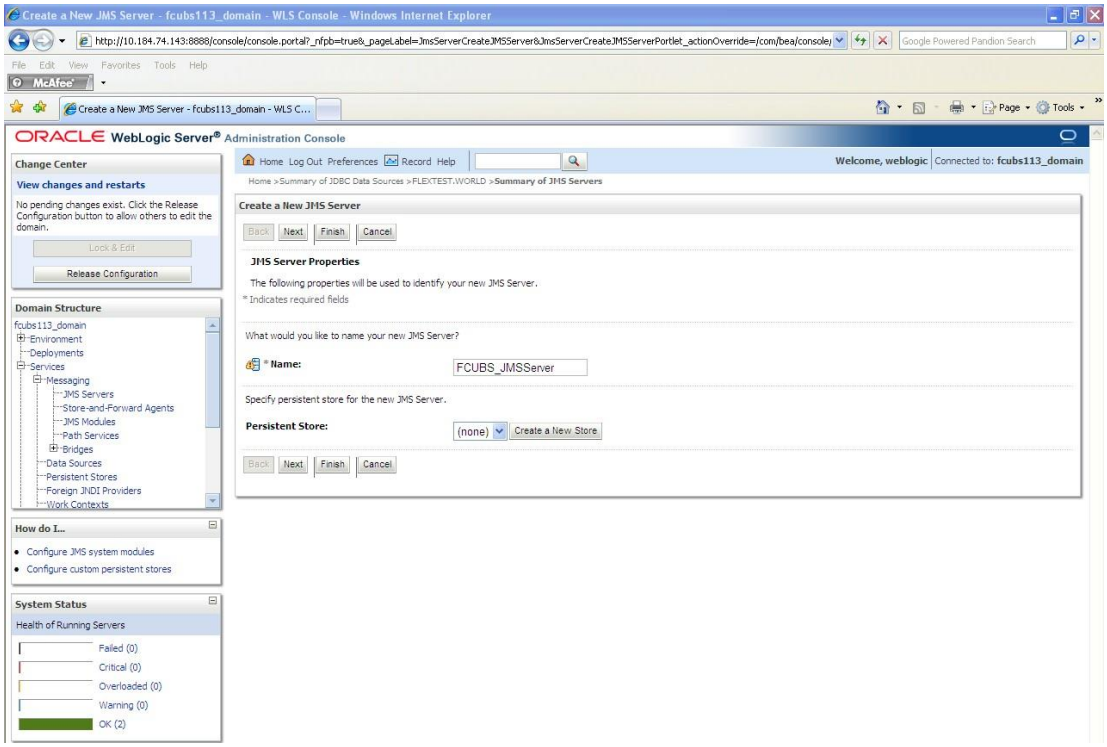
4. Following screen is displayed:



5. Expand ‘Services’ and then ‘Messaging’ and ‘JMS Server ‘under it. Click ‘Lock & Edit’ button.



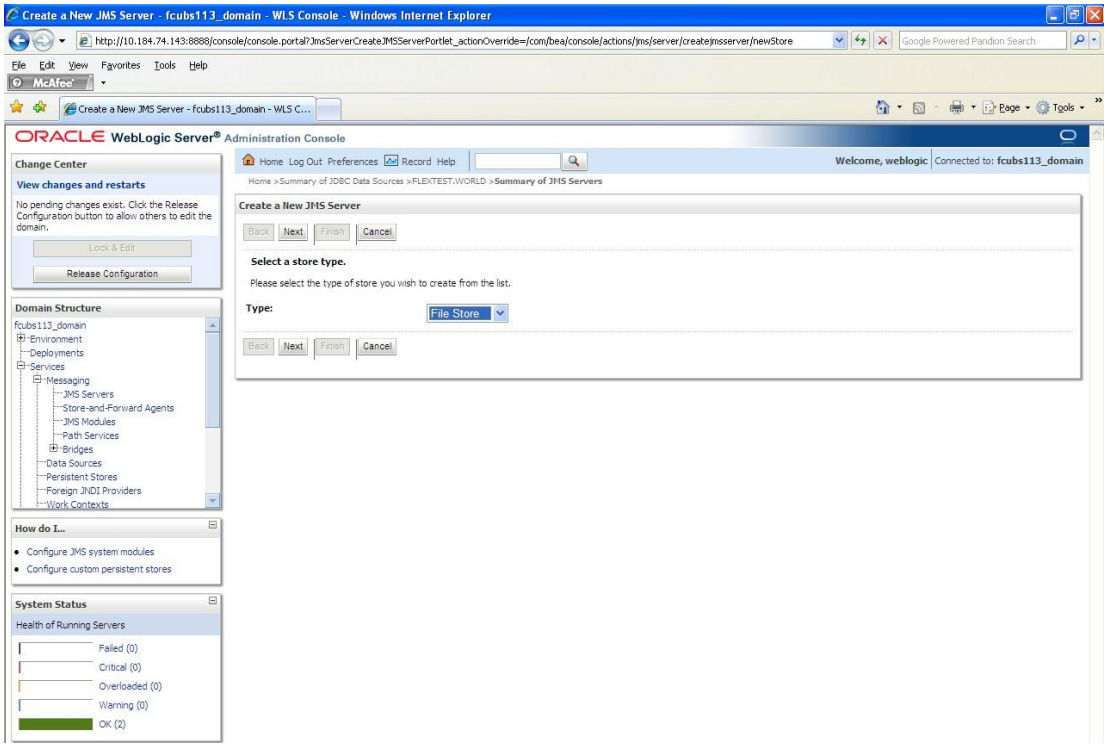
6. Click ‘New’.



7. Specify the following details:

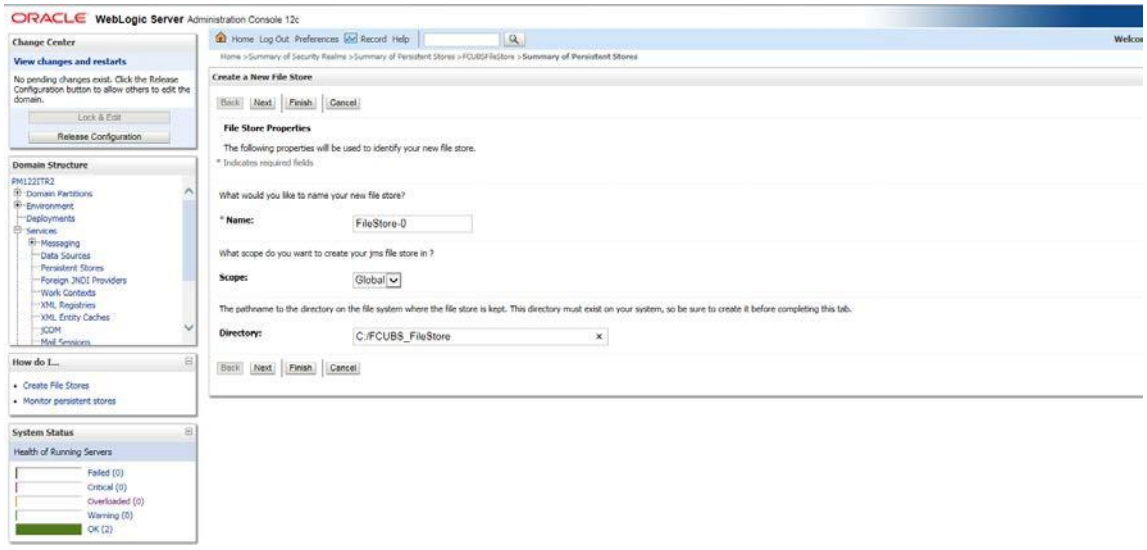
JMS Server Name	Specify the name of JMS Server.

8. Click ‘Create a new Store’ button. The following screen is displayed.



9. Select ‘File Store’ as the type and click ‘Next’.

   Following screen is displayed:

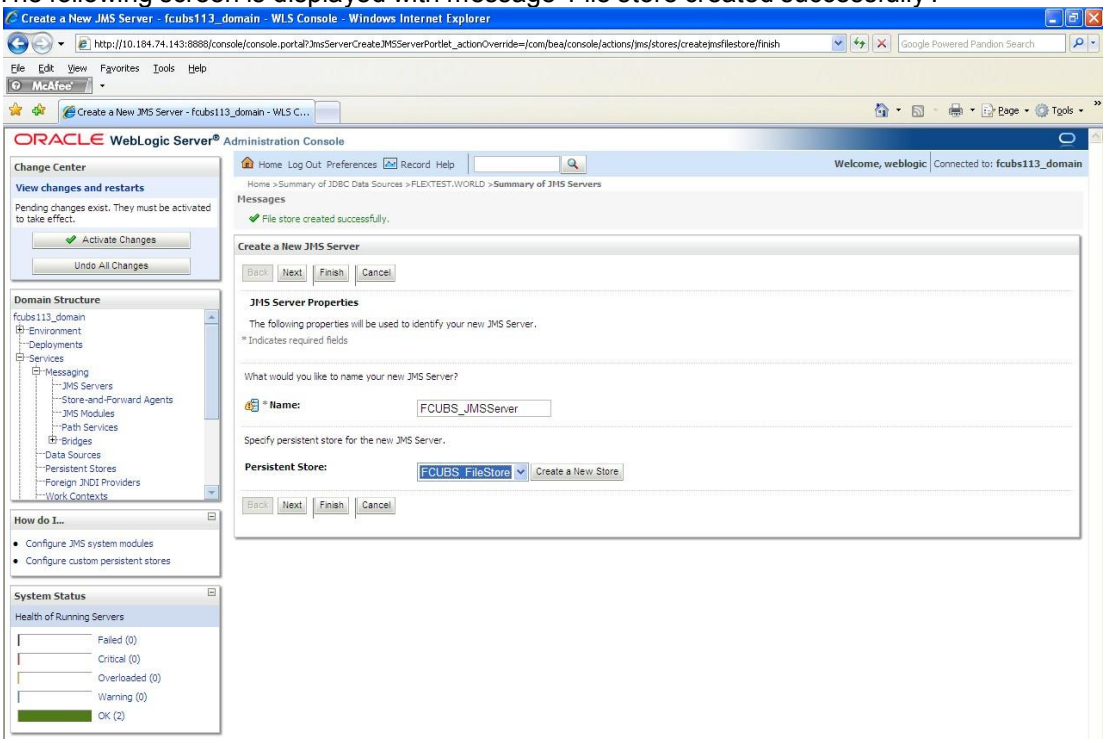


10. To identify the new File Store, specify the following properties:

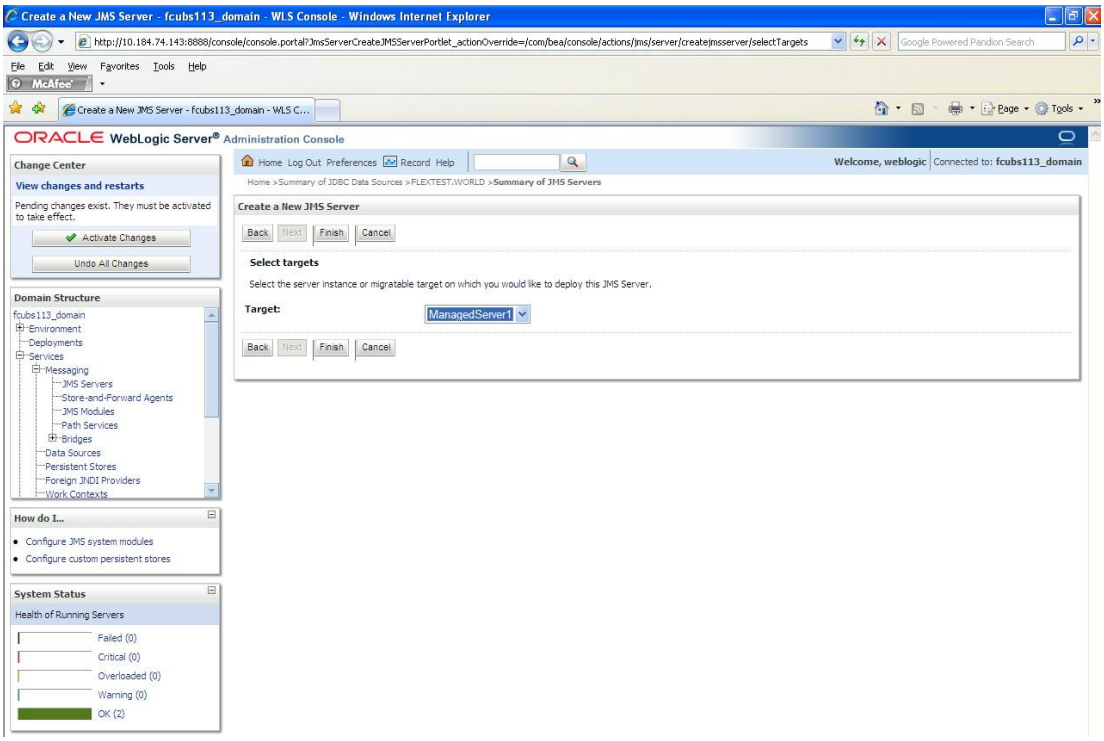
    - Specify the file store name as FCpayments_FileStore.
    - Select a server. For this file store, you may select ManagedServer1 (created by the user).
    - Specify the Filestore Directory path as C:/FCpayments_FileStore.

- Click ‘OK’.

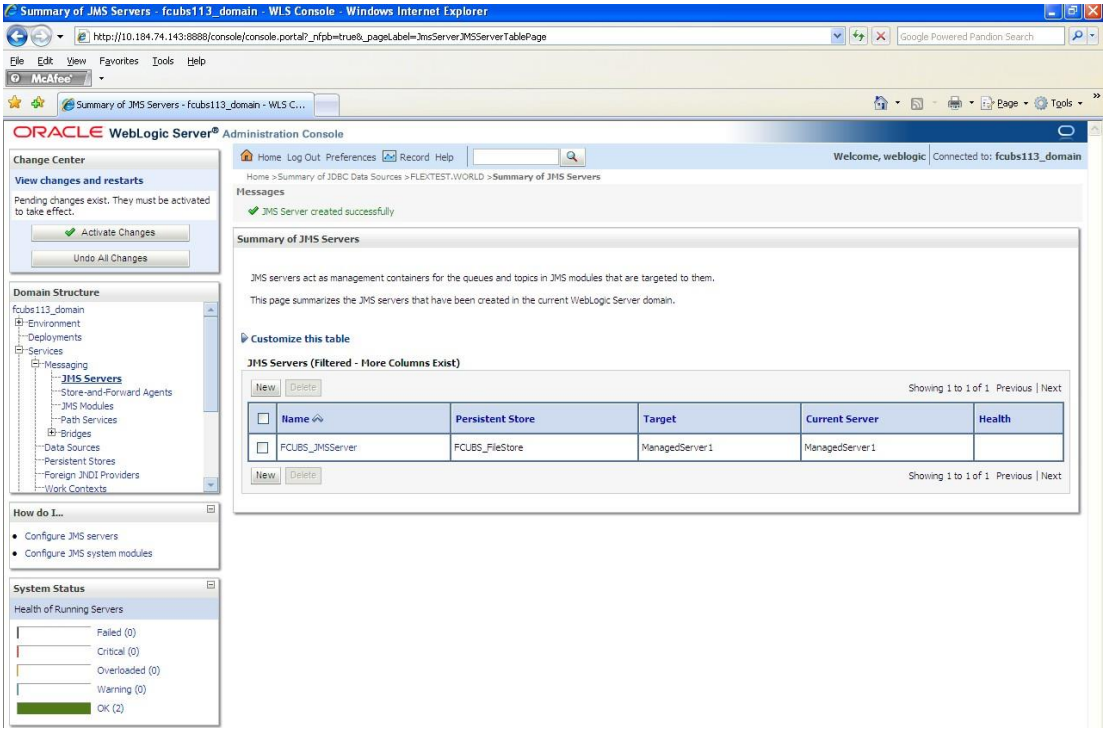
The following screen is displayed with message ‘File store created successfully’.



11. Click ‘Next’.



12. Select the target managed server. Click 'Finish'.



13. The message 'JMS Server created successfully' is displayed.

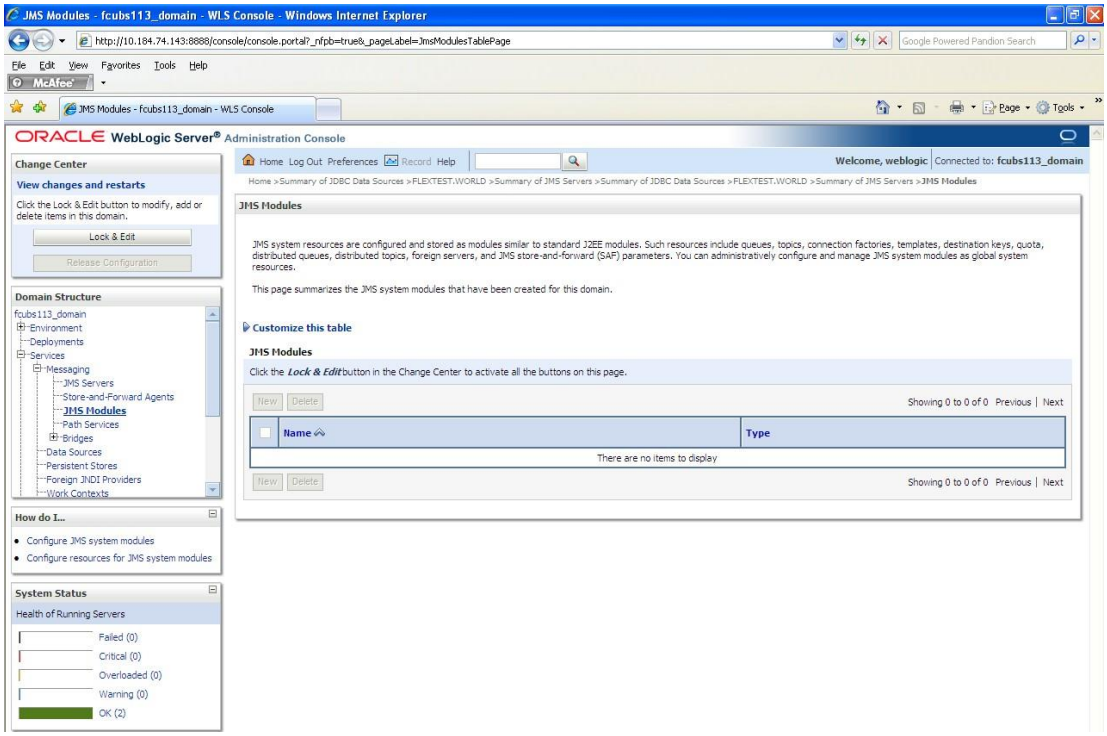
14. Click ‘Activate Changes’ under Change Center. The message ‘All changes have been activated. No restarts are necessary’ is displayed.

7.2.3 JMS Modules Creation

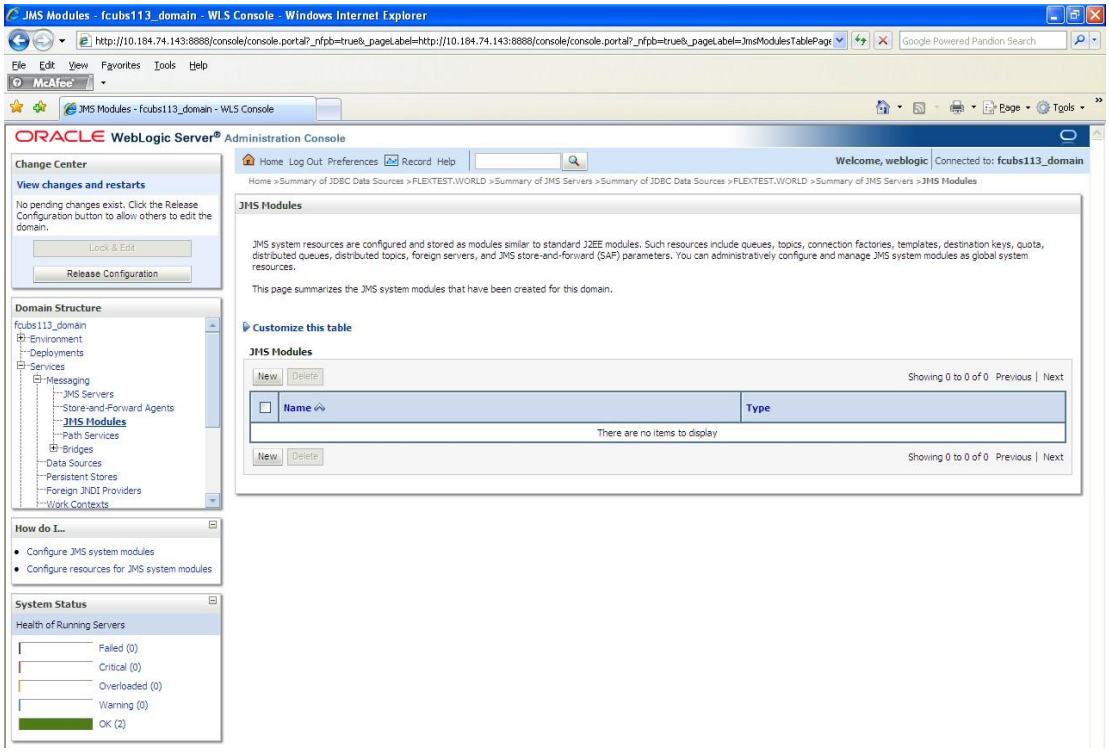
Follow the steps given below:

1. Navigate to the WEBLOGIC Home Page. Click ‘JMS Modules’ on domain structure by expanding ‘Messaging’.

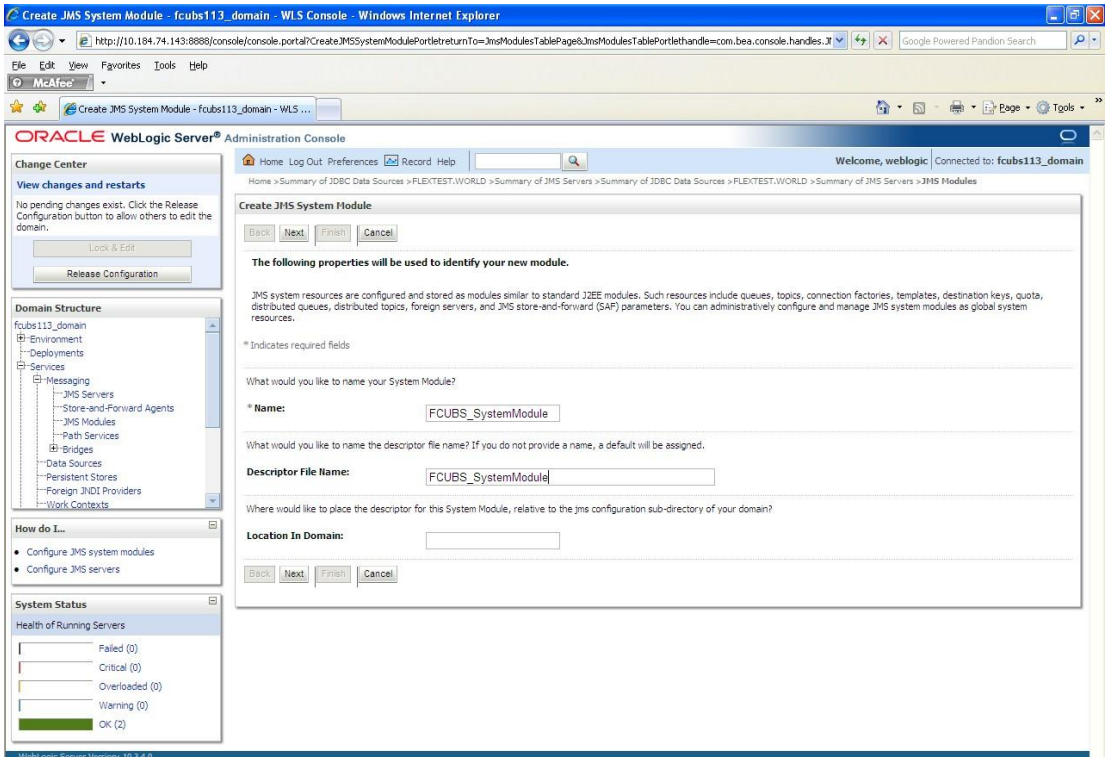
   The following screen is displayed:



2. For creating New JMS System Modules, click ‘Lock & Edit’ button.



3. Click ‘New’ button. The following screen is displayed.

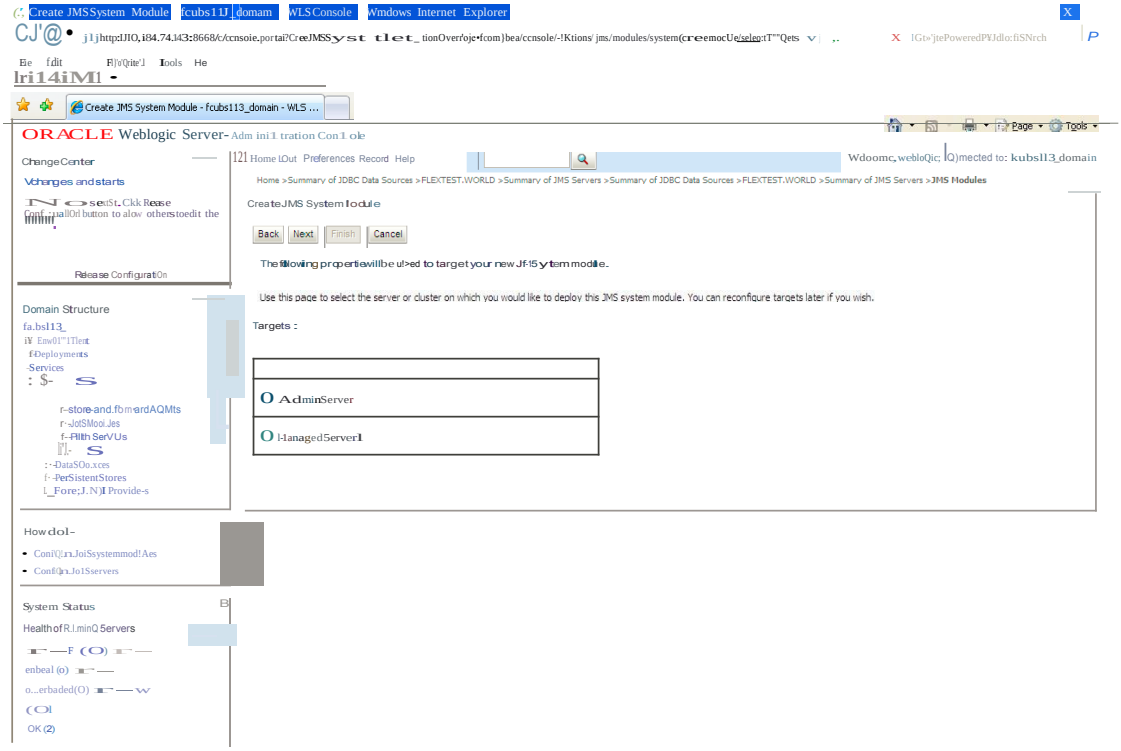


Enter the System Module Name as FCUBS_SystemModule.

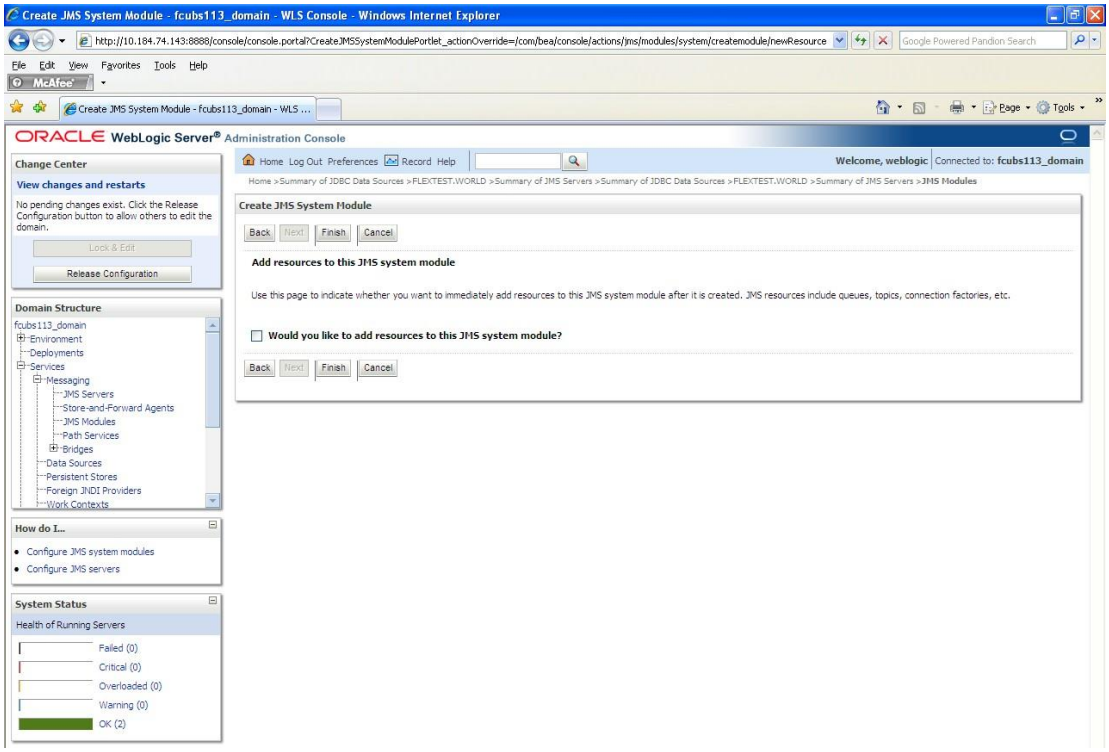
Enter the Description File Name as FCUBS_SystemModule.

4. Click 'Next'.

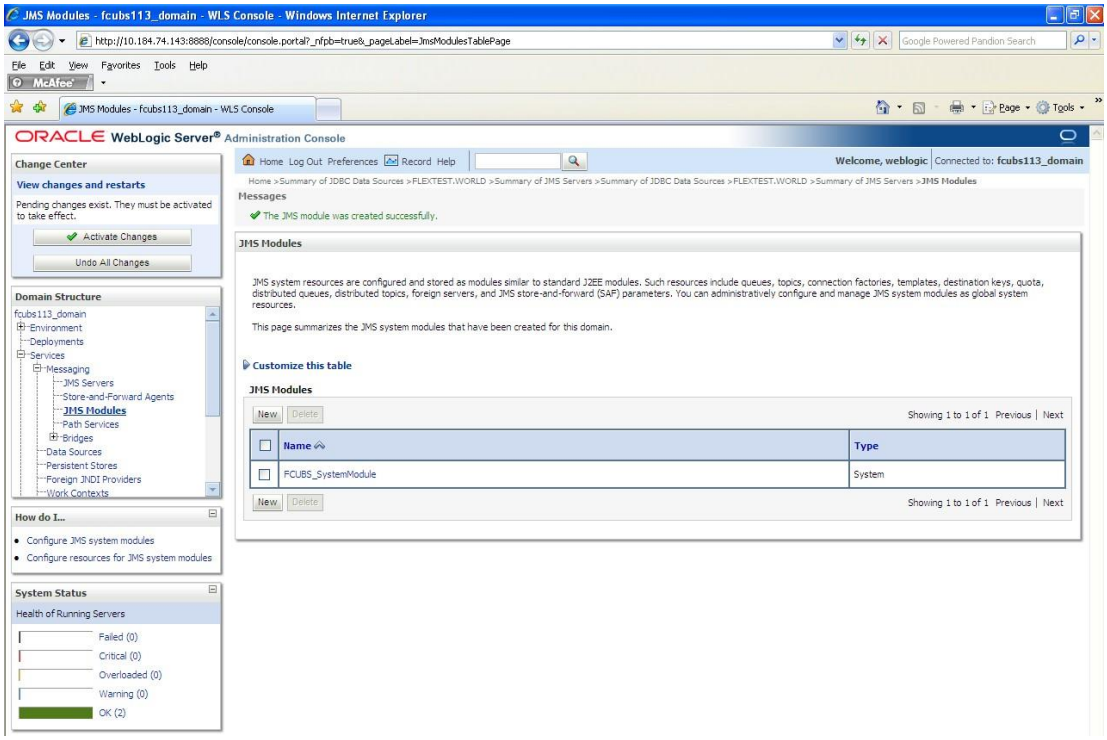
The following screen is displayed.



5. Check the box against the server created. Click ‘Next’. The following screen is displayed.

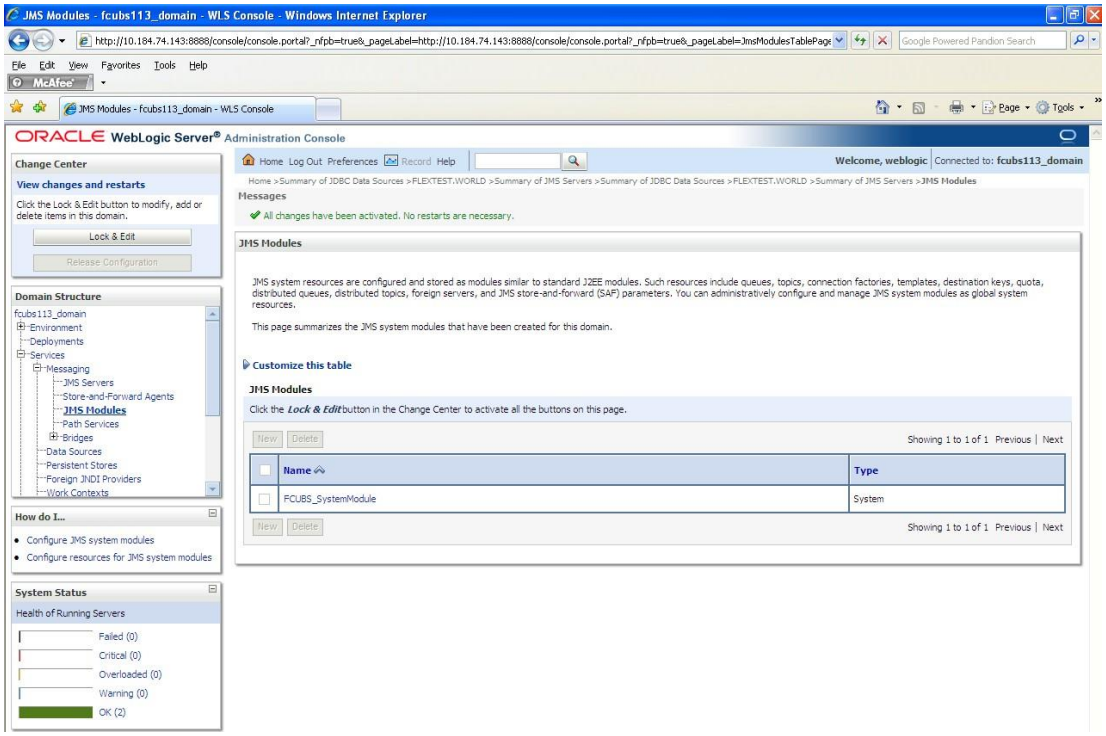


6. Click ‘Finish’ button. The following screen is displayed.



7. Click ‘Activate Changes’ button on the left pane.

The message ‘All the changes have been activated. No restarts are necessary’ is displayed.

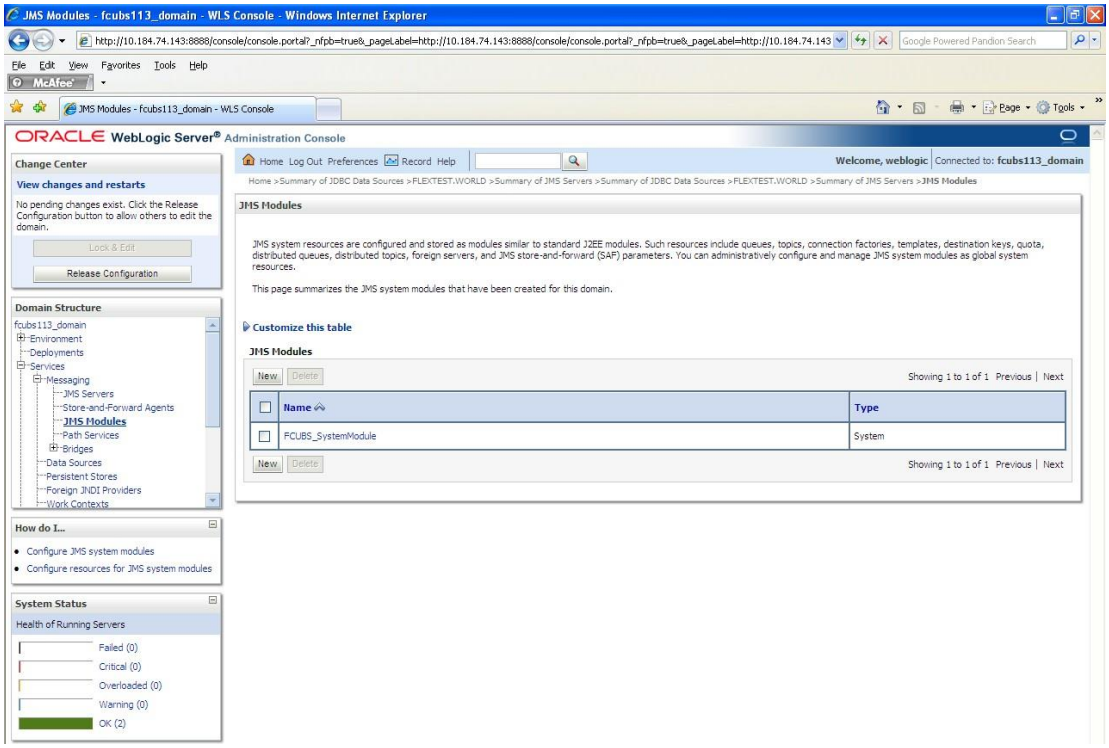


7.2.4 Subdeployment Creation

Follow the steps given below:

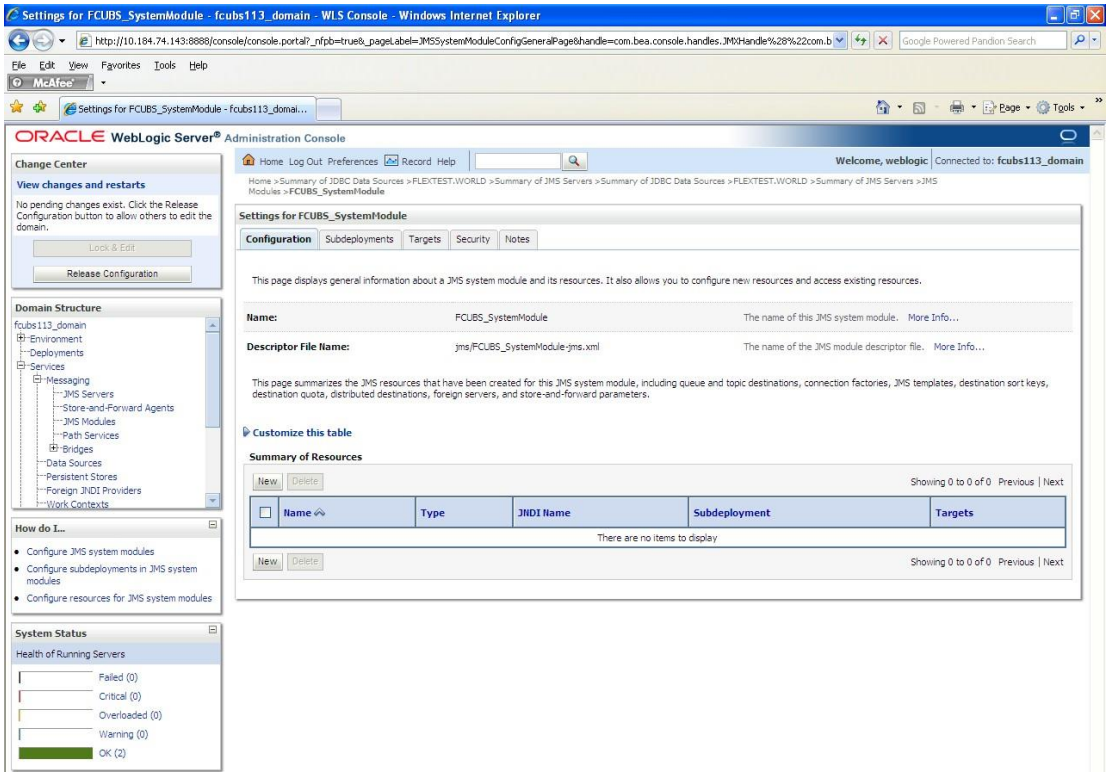
1. Navigate to the WEBLOGIC Home Page. Click ‘JMS Modules’ on domain structure by expanding ‘Messaging’.

The following screen is displayed:

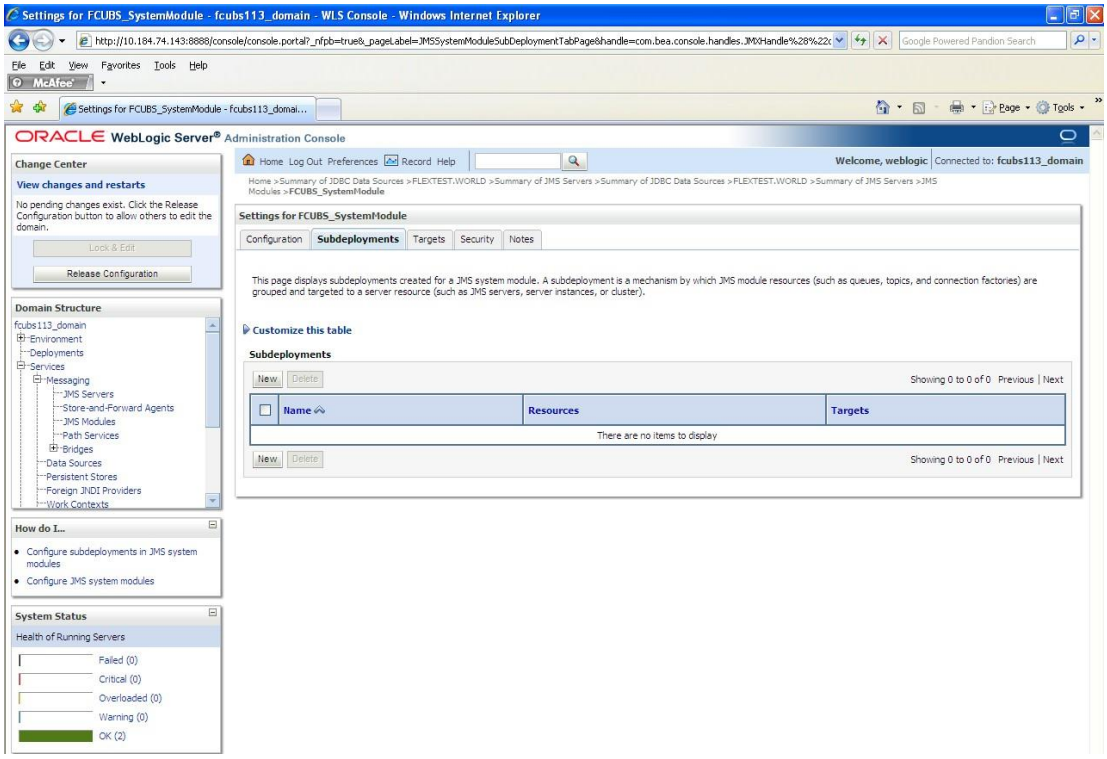


2. Click ‘Lock & Edit’ button.

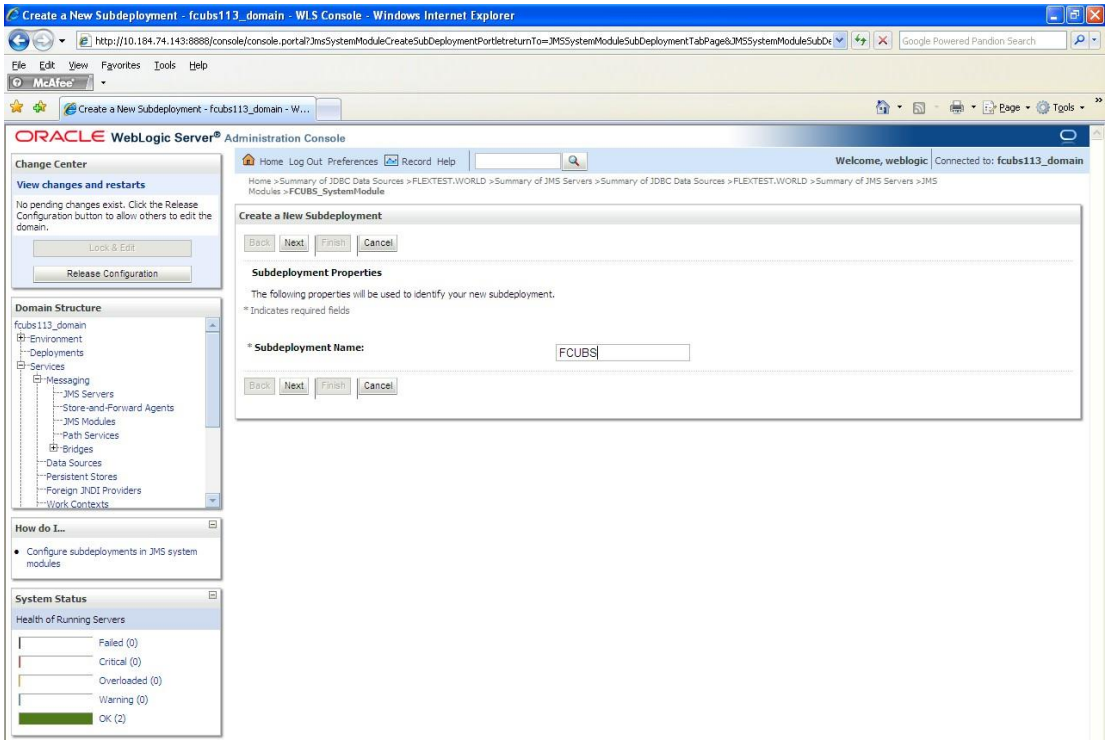
3. Select the JMS module created earlier.



4. Click ‘Subdeployments’ tab.

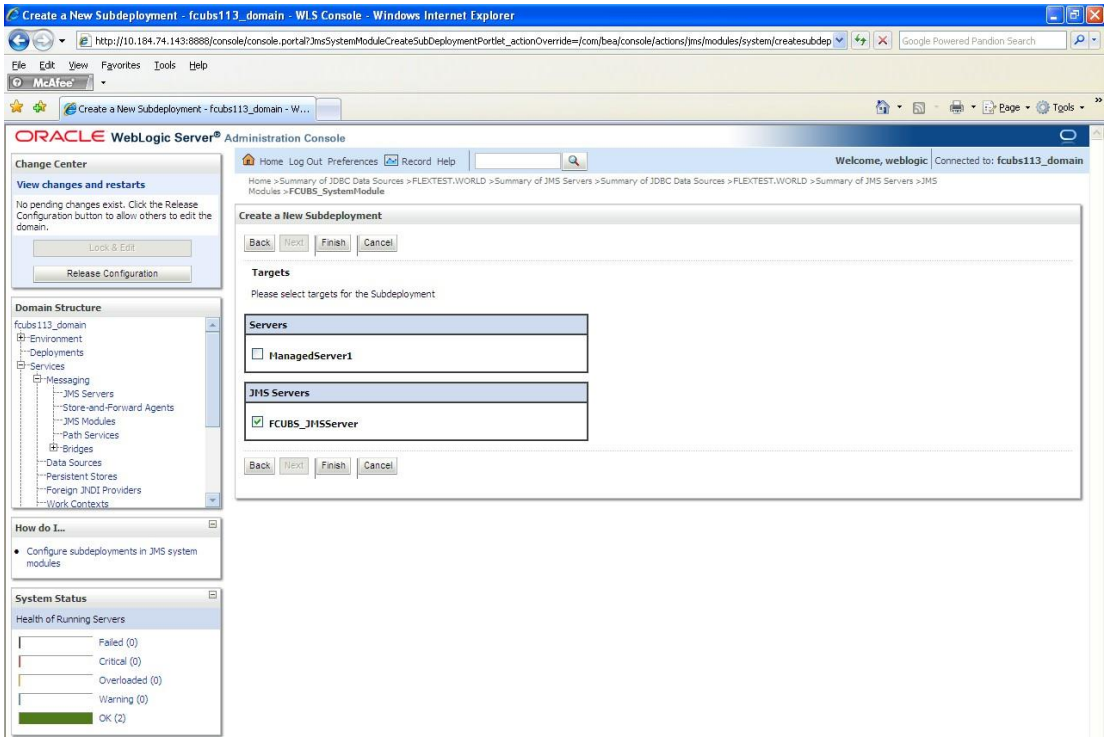


5. Click ‘New’. The following screen is displayed.



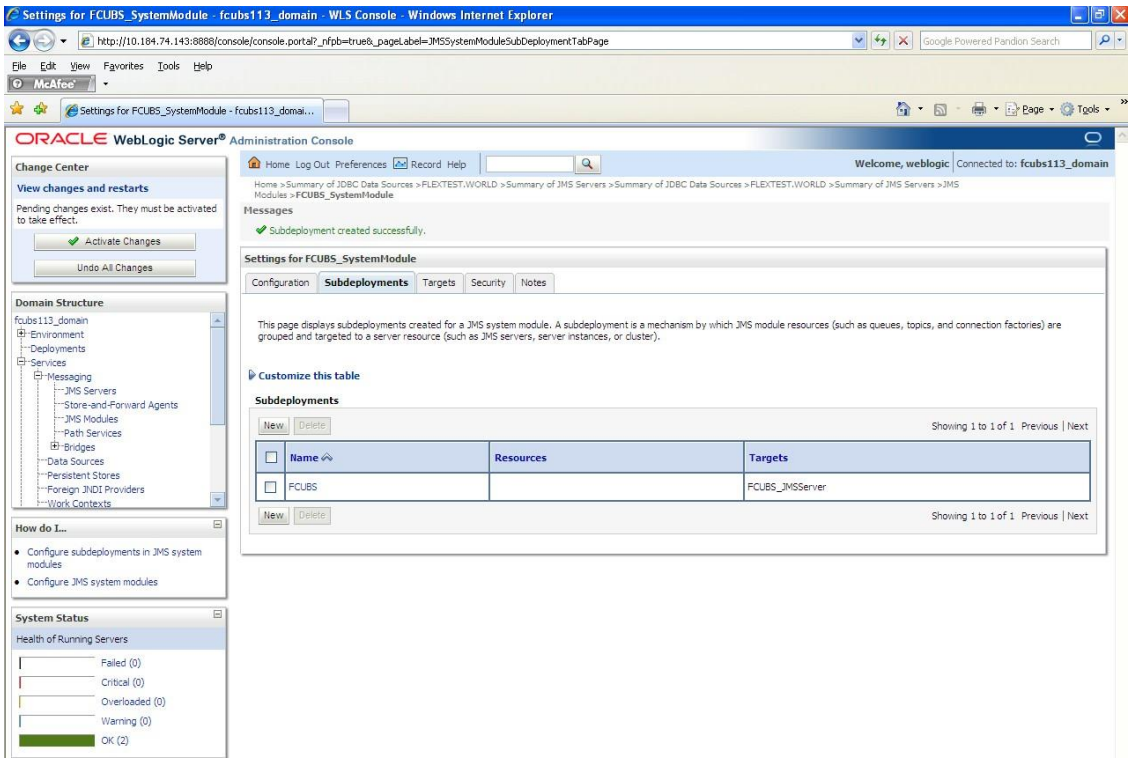
6. Specify the Subdeployment Name as ‘FCUBS’. Then click ‘Next’.

The following screen will be displayed.

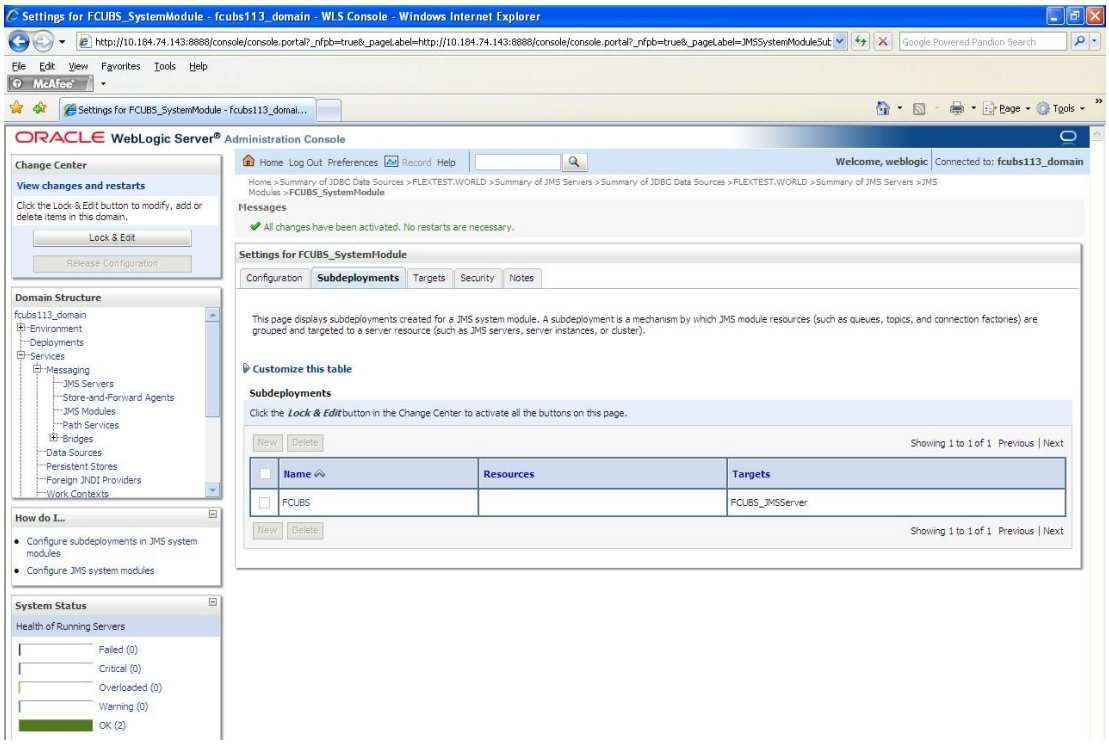


7. Select the JMS Server (as created by the user).
8. Click ‘Finish’ button.

9. Following screen is displayed.

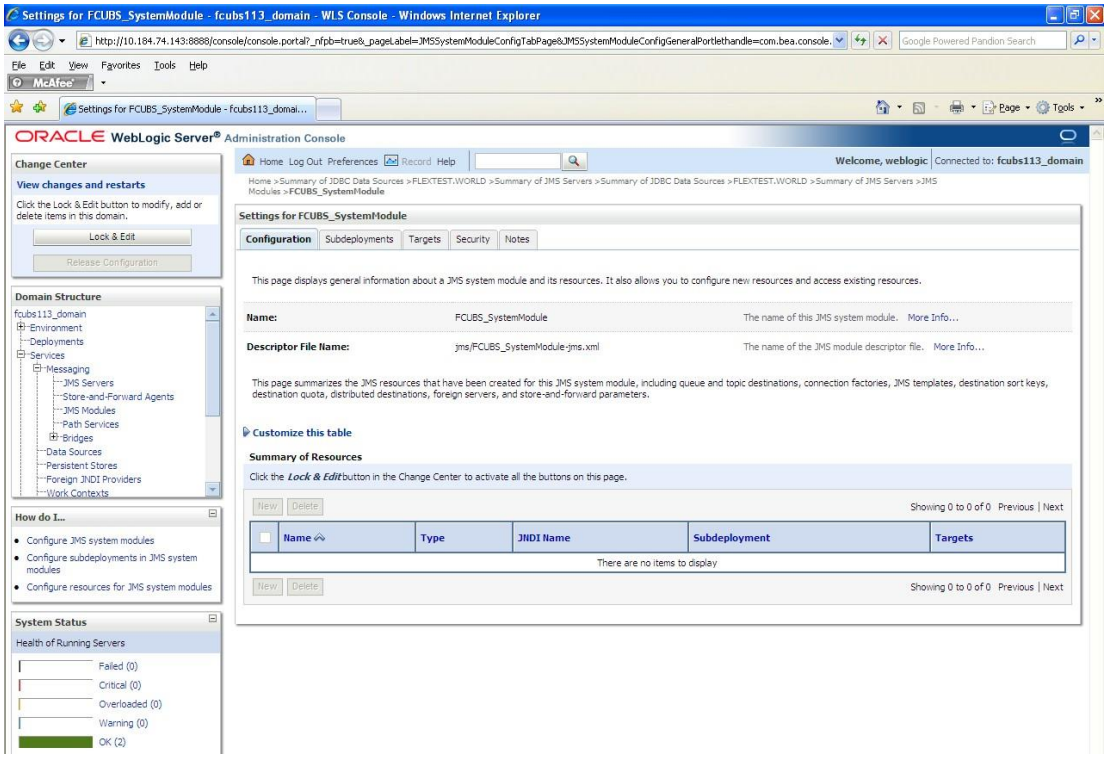


10. Click ‘Activate Changes’. Following screen is displayed.



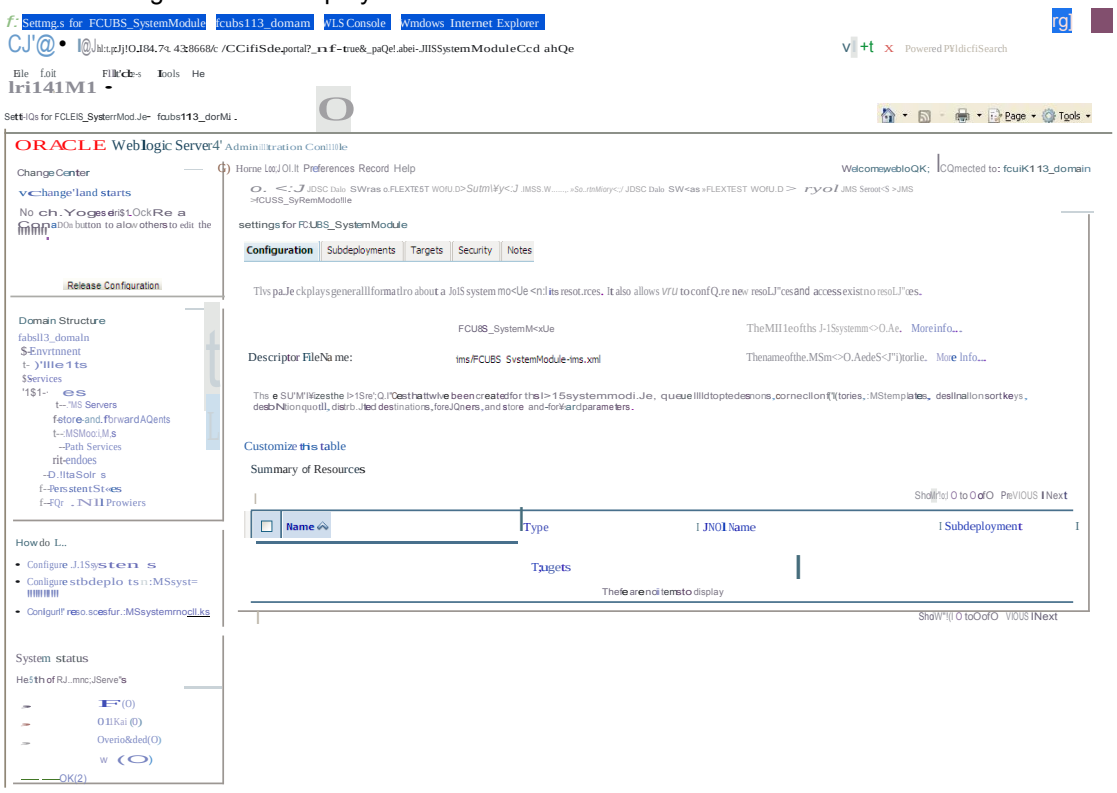
7.2.5 JMS Queue Creation

1. Select the JMS Module created earlier.



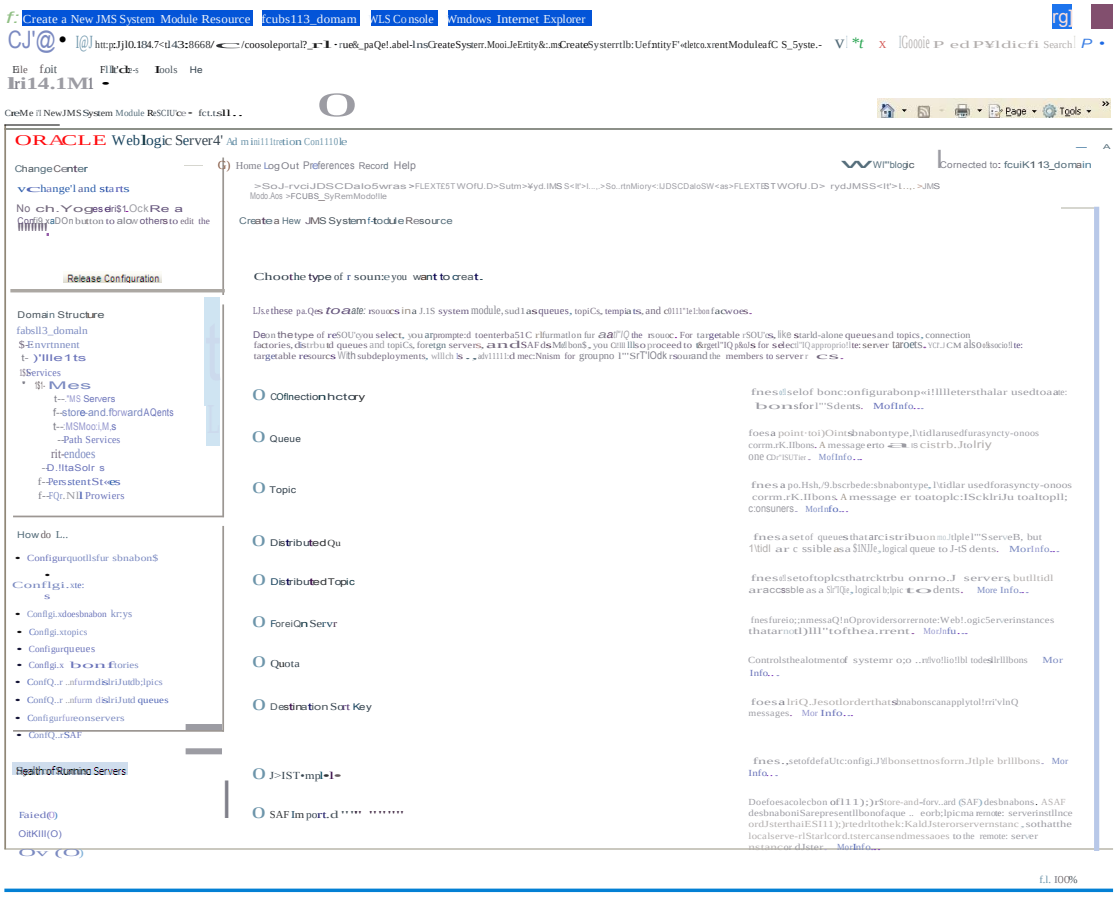
2. You need to set the configuration for FCUBS_SystemModule is to be set.
3. Click ‘Configuration’. Then click ‘Lock & Edit’.

The Following screen is displayed.

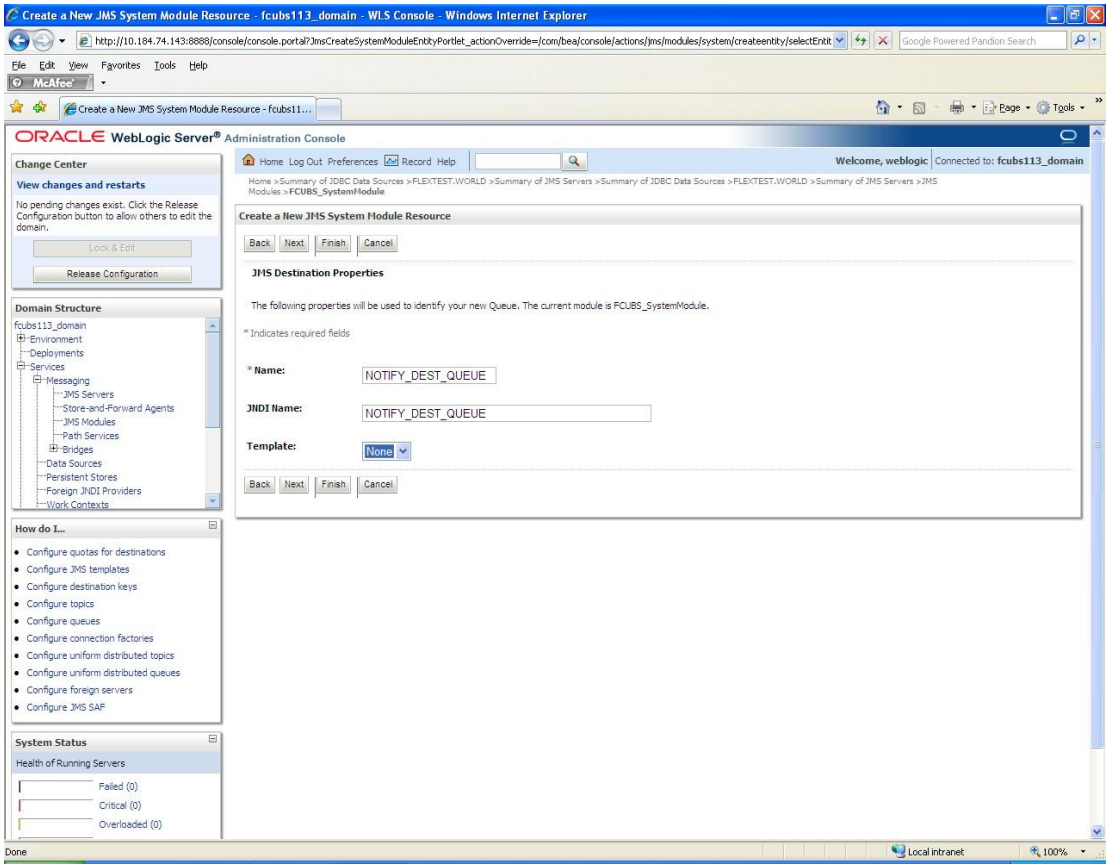


ORACLE

4. Click 'New'. The following screen is displayed.



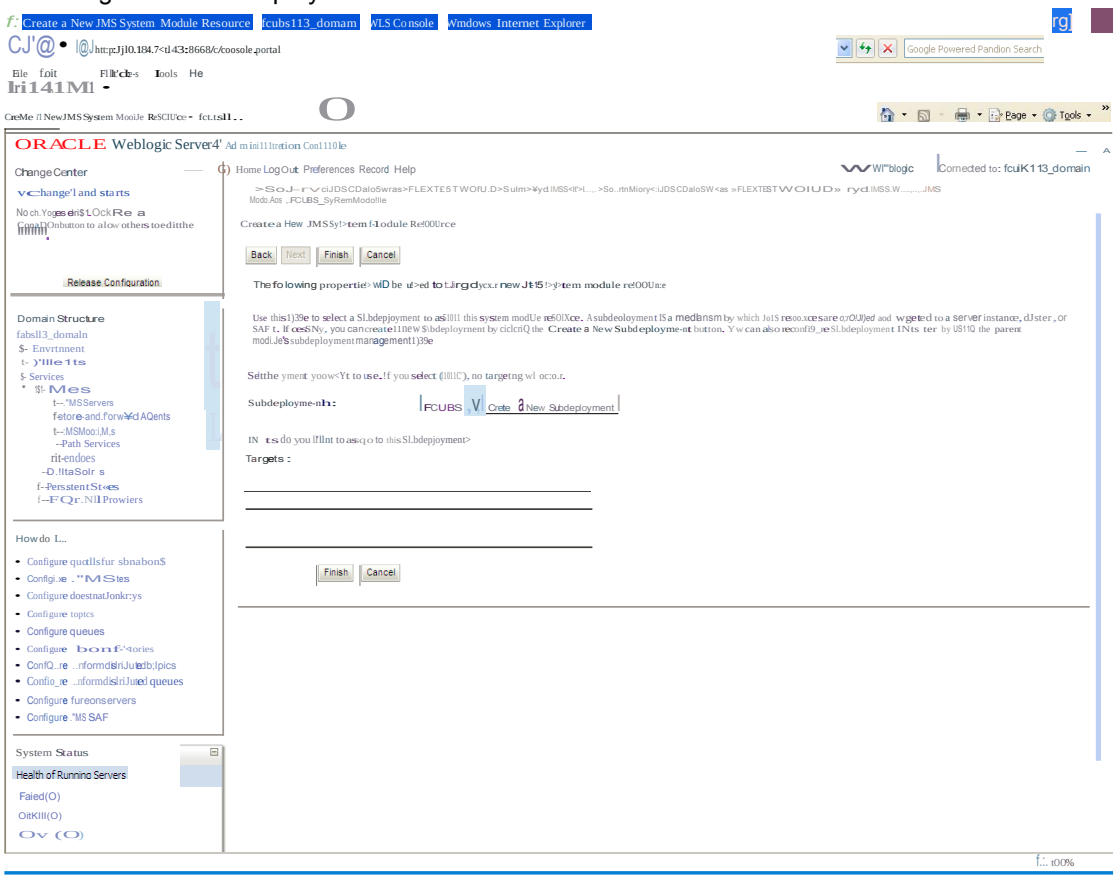
5. Select the ‘Queue’ option. Then click ‘Next’.



For creating new JMS System Module Resources, follow the steps given below:

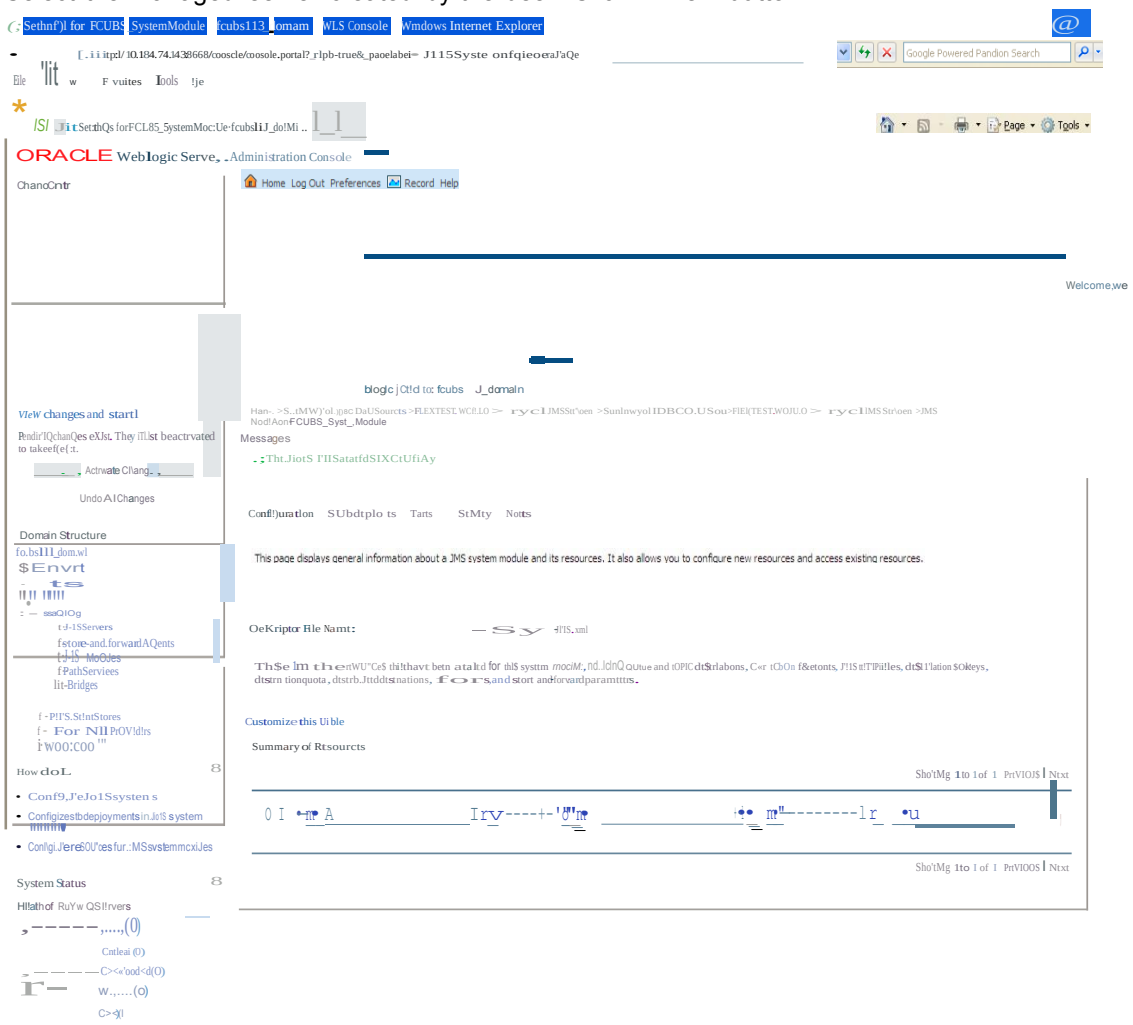
- Enter the Name of the Queue as ‘NOTIFY_DEST_QUEUE’.
- Enter the JNDI Name as ‘NOTIFY_DEST_QUEUE’.
- Select the Template as ‘None’.
- Click ‘Next’.

Following screen is displayed.



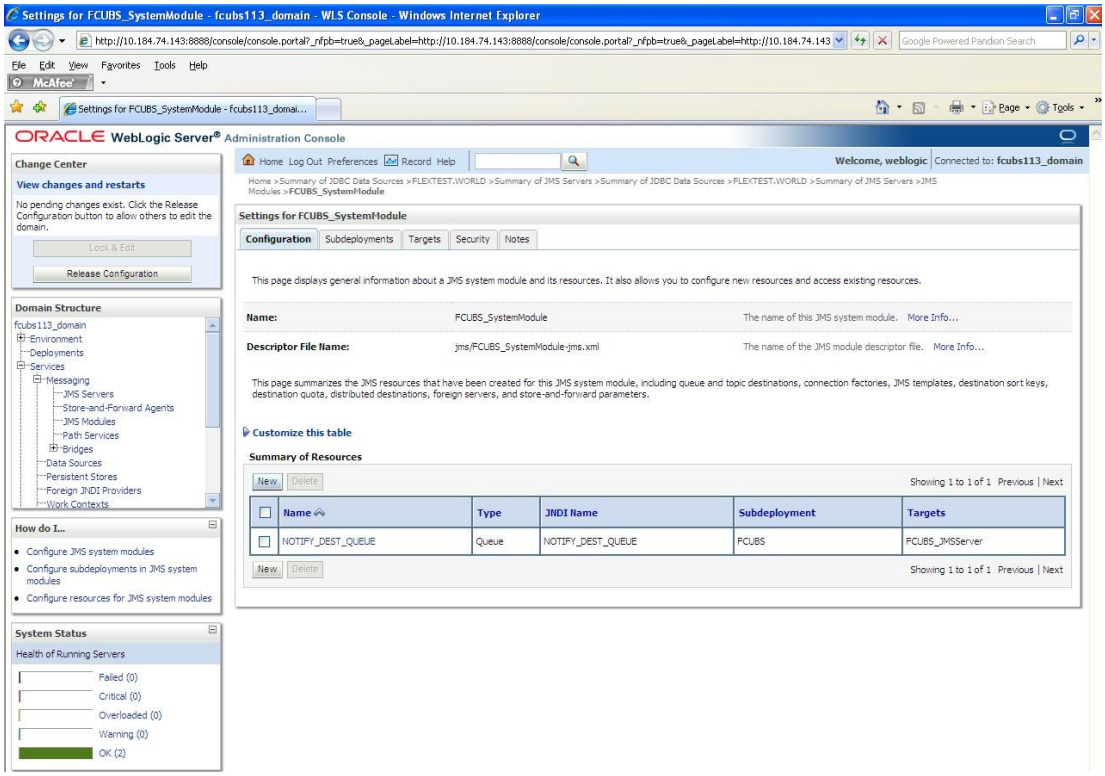
ORACLE

6. Select the managed server created by the user. Click 'Finish' button.



ORACLE

7. The JMS Queue has been created successfully. Click ‘Activate Changes’ under ‘Change Center’.

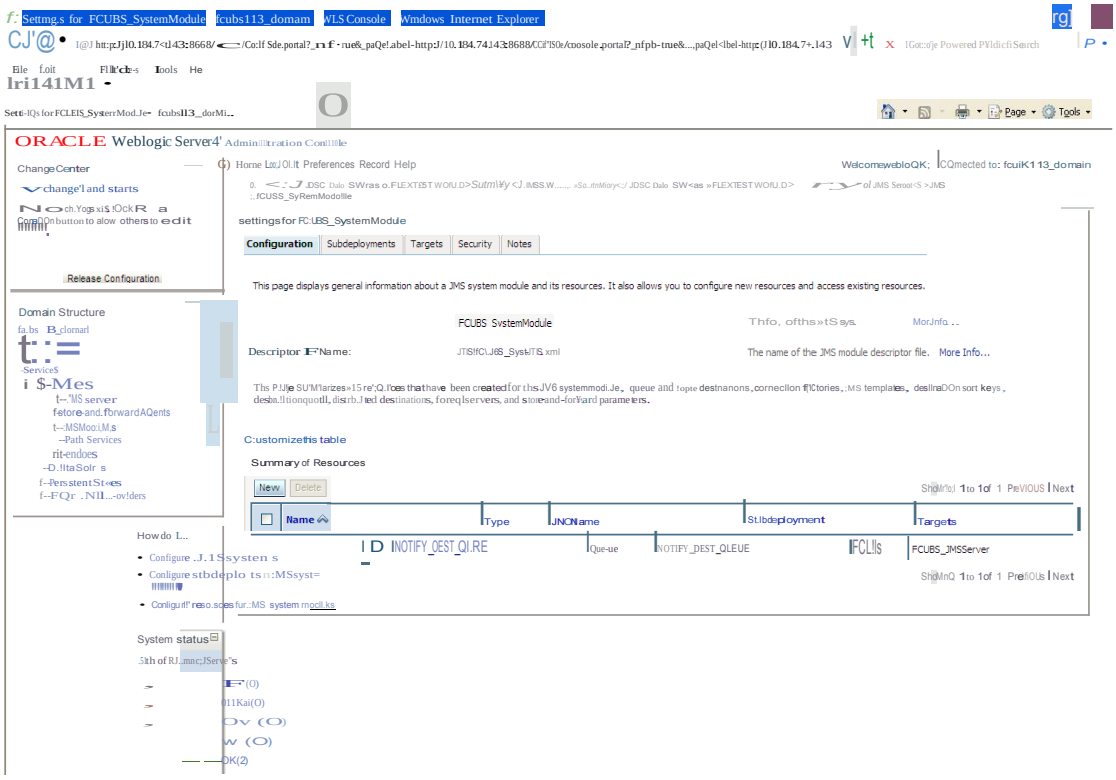


8. Click ‘New ‘to create more Queues. You may follow the same steps to create other queues.

7.2.6 JMS Connection Factory Creation

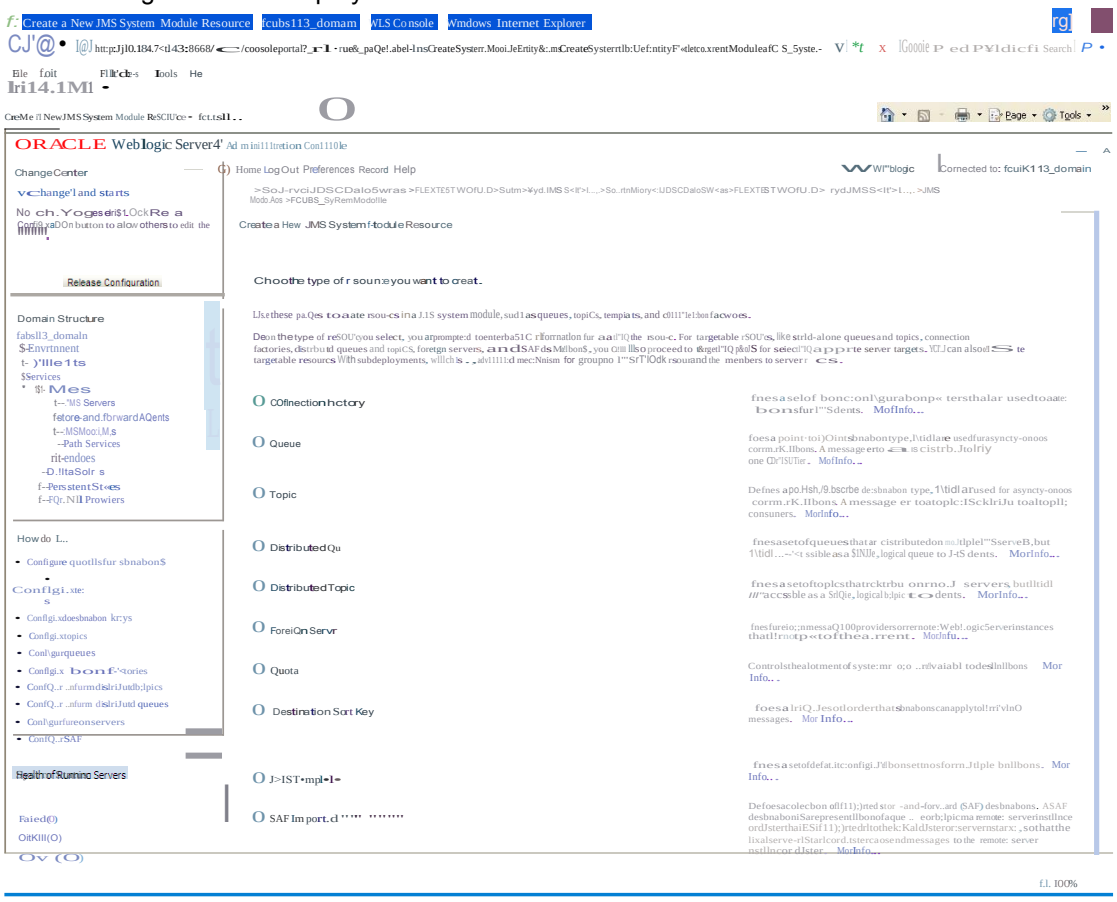
After creating the queues, you need to create the connection factory. To perform this, follow the steps given below:

1. Click 'New'.

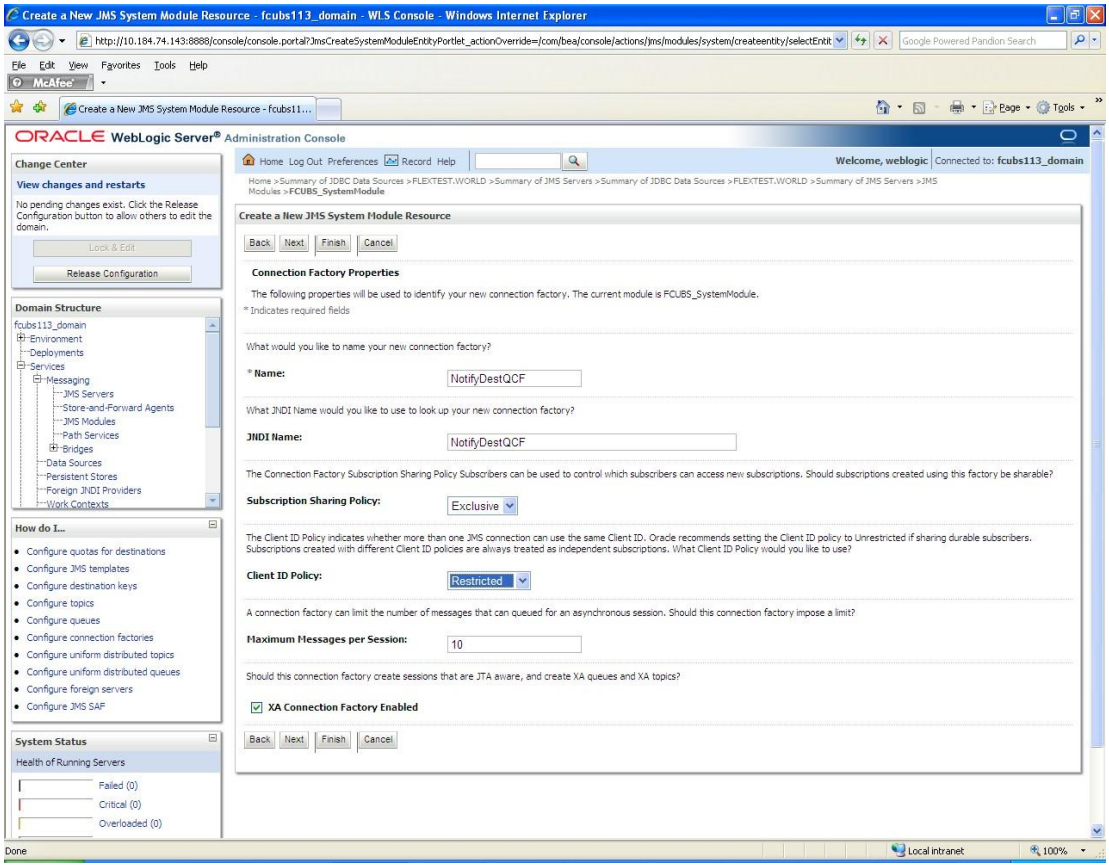


ORACLE

The following screen is displayed:

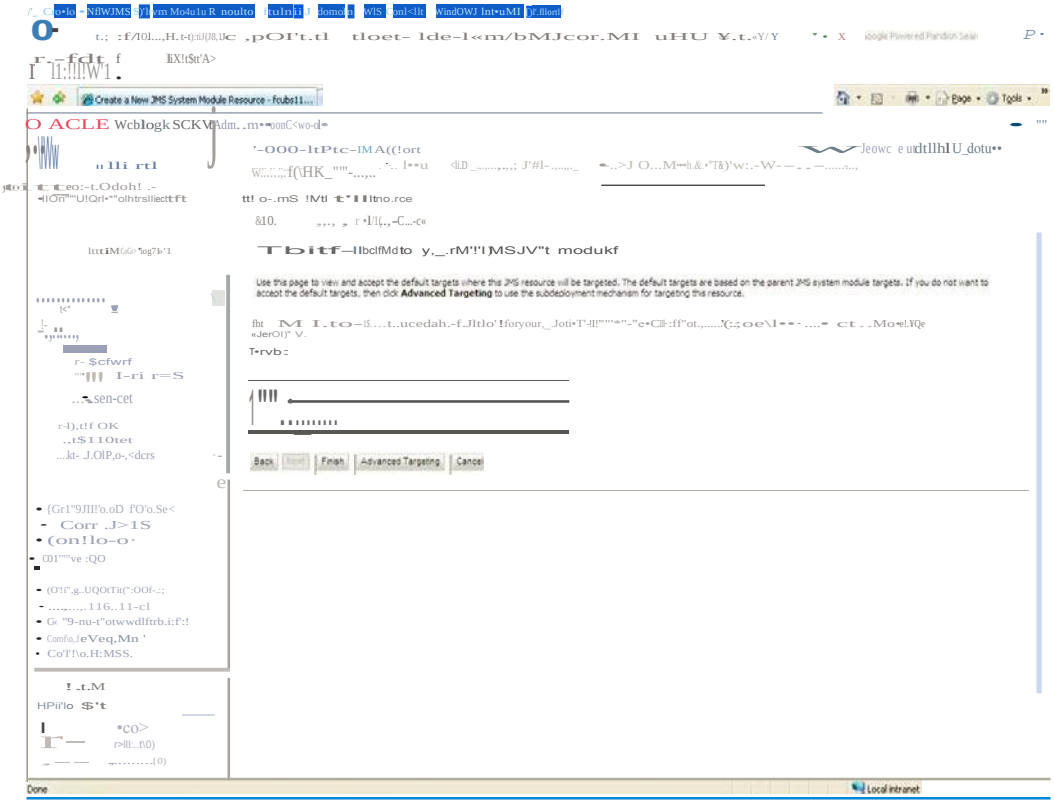


2. Select ‘Connection Factory’. Click ‘Next’.



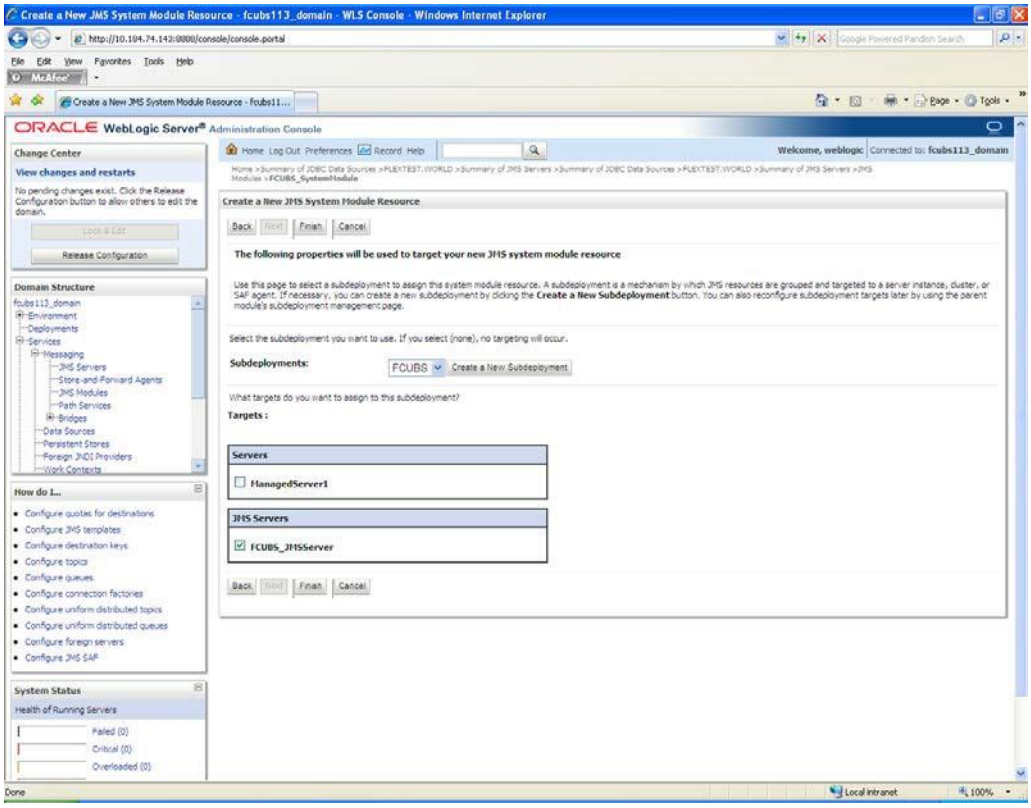
3. Enter the Name of the Connection Factory as ‘NotifyDestQCF’.
4. Enter the JNDI Name as ‘NotifyDestQCF’.
5. Check the box ‘XA Connection Factory Enabled’.
6. Click ‘Next’.

The following screen is displayed:



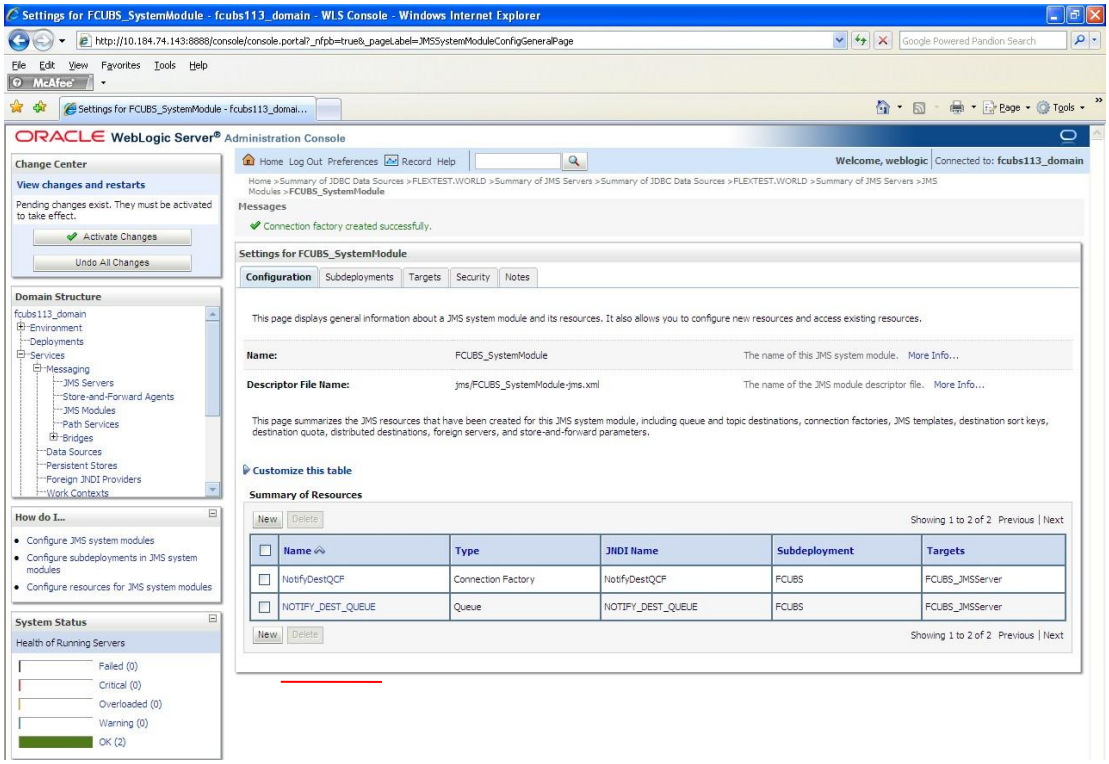
ORACLE

7. Click ‘Advanced Targeting’. The following screen is displayed.



8. Select the ‘Subdeployments’ as FCUBS.
9. Under JMS Servers, check the box against ‘Managed Server’.

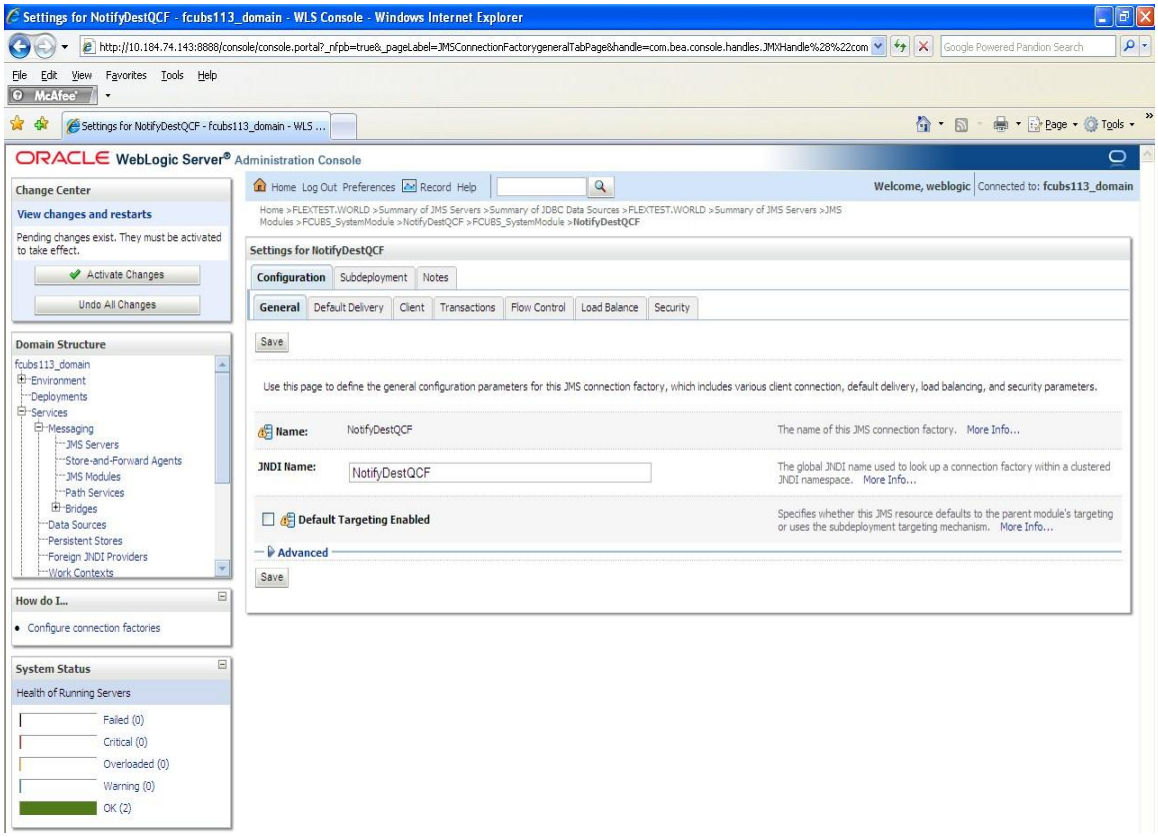
10. Click ‘Finish’. The following screen is displayed:



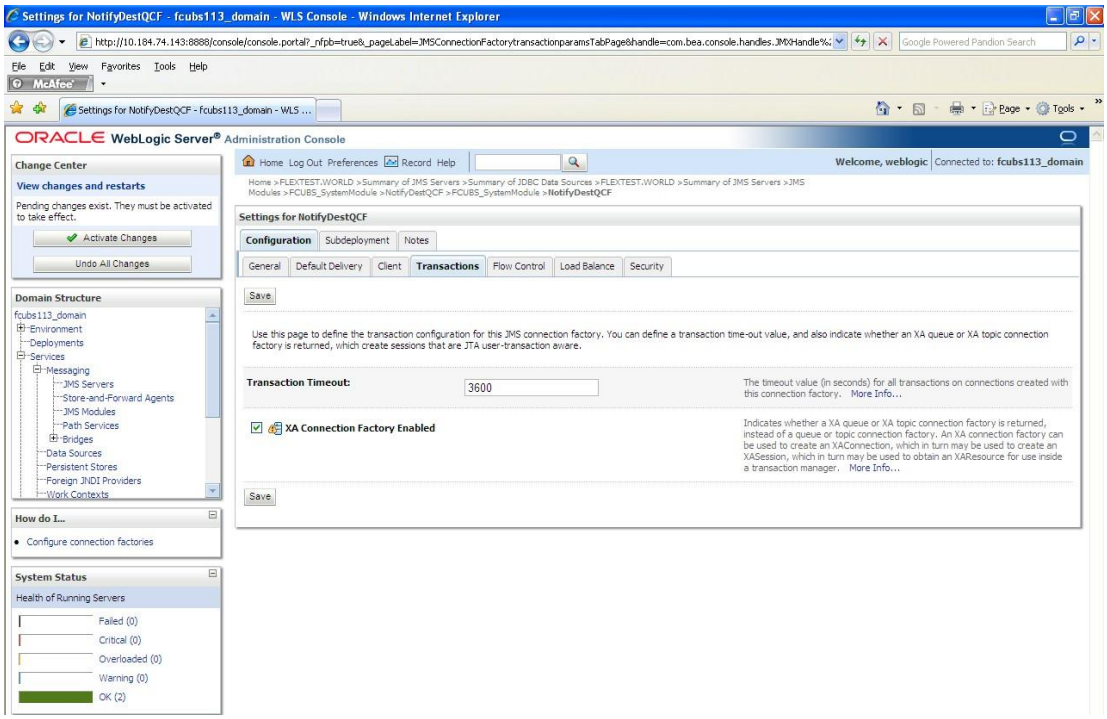
11. The message ‘Connection Factory created successfully’ is displayed.

12. Click on the Connection Factory ‘NotifyDestQCF’ to have XA Connection Factory enabled.

The following screen will be displayed.

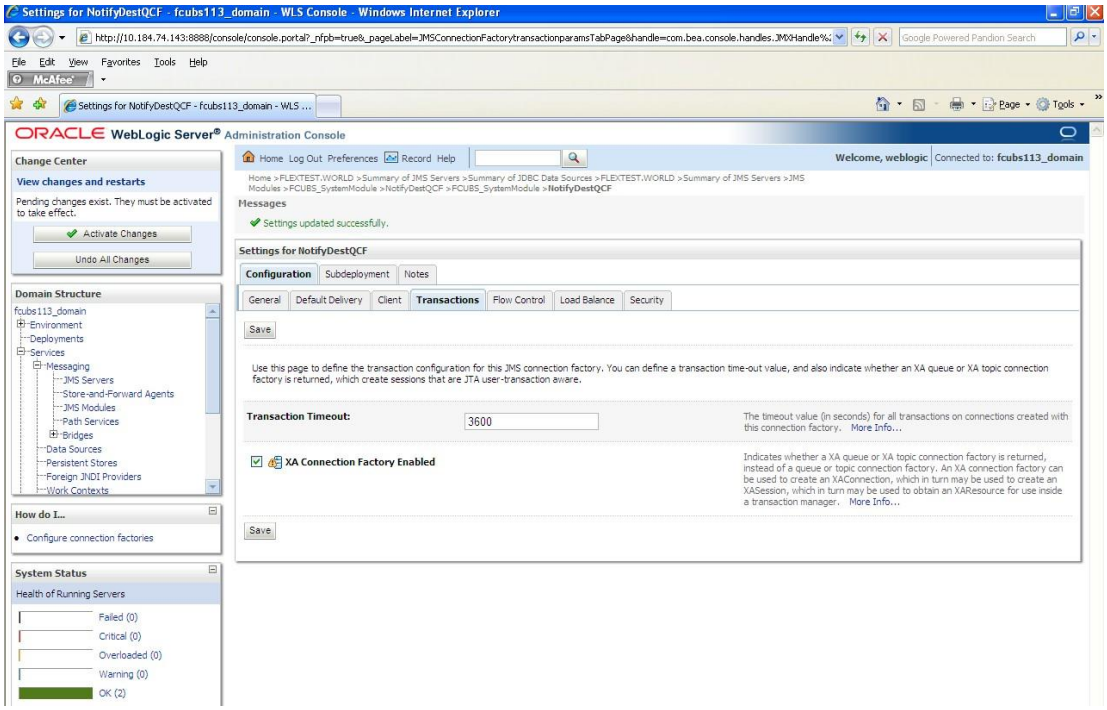


13. Click ‘Transactions’ Tab. The following screen is displayed.



14. Check the box ‘XA Connection Factory Enabled’.

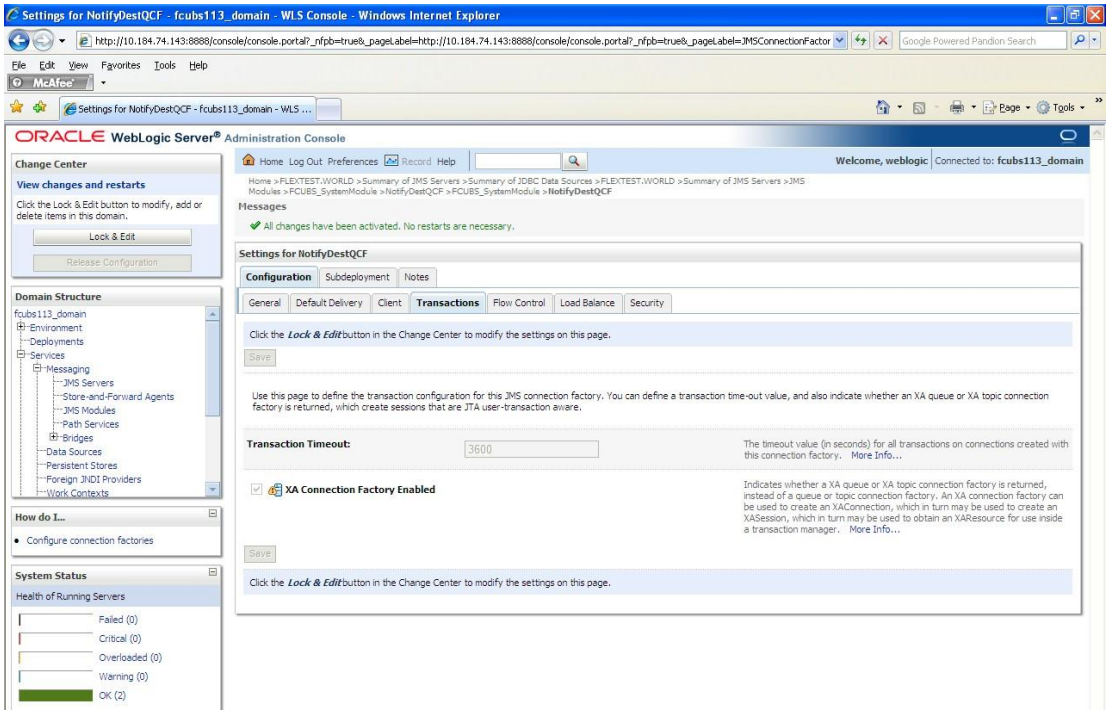
15. Click ‘Save’. The following screen is displayed.



16. The message ‘Settings updated successfully’ is displayed.

17. Click ‘Activate Changes’ button under ‘Change Center’.

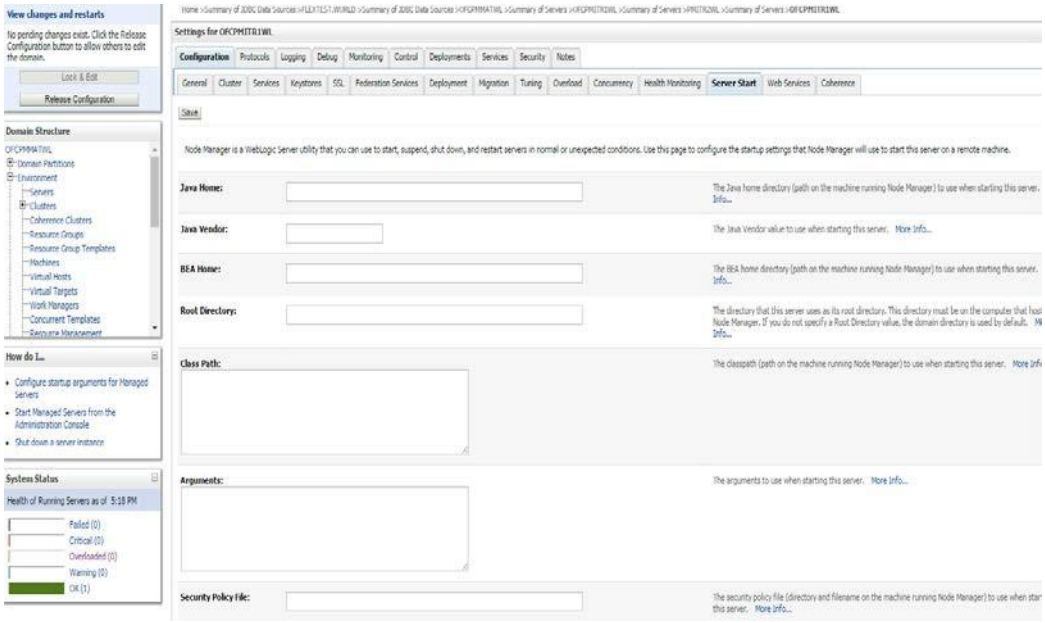
    The message ‘All the changes have been activated. No restarts are necessary’ is displayed.



7.3 Configuring Weblogic for Oracle Banking payments

This section explains the steps for configuring Oracle WebLogic application server for Oracle Banking payments. Follow the steps given below:

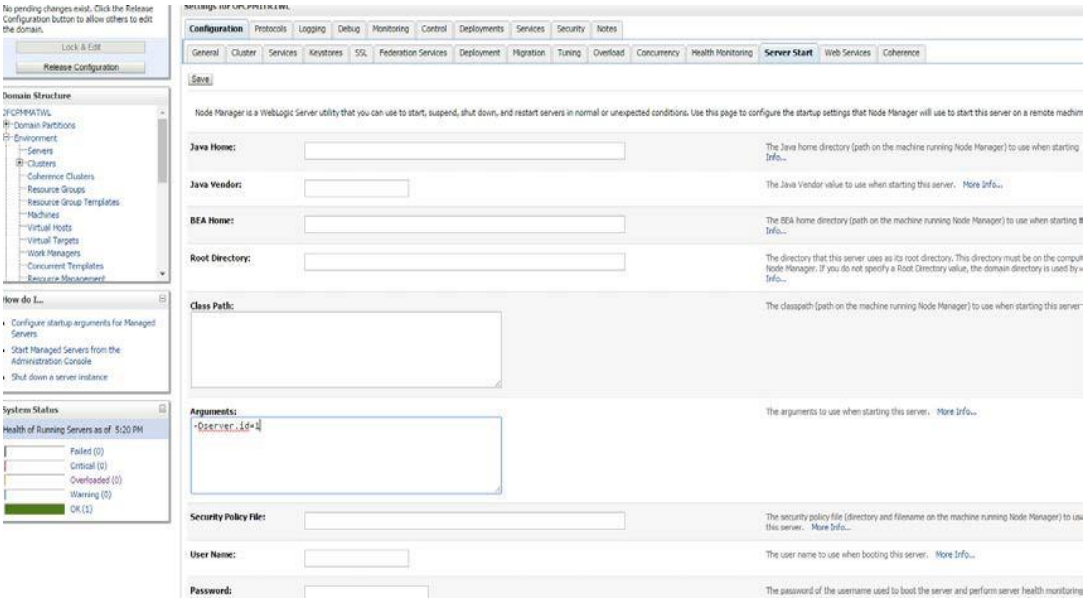
1. Select the servers from domain structure shown below.



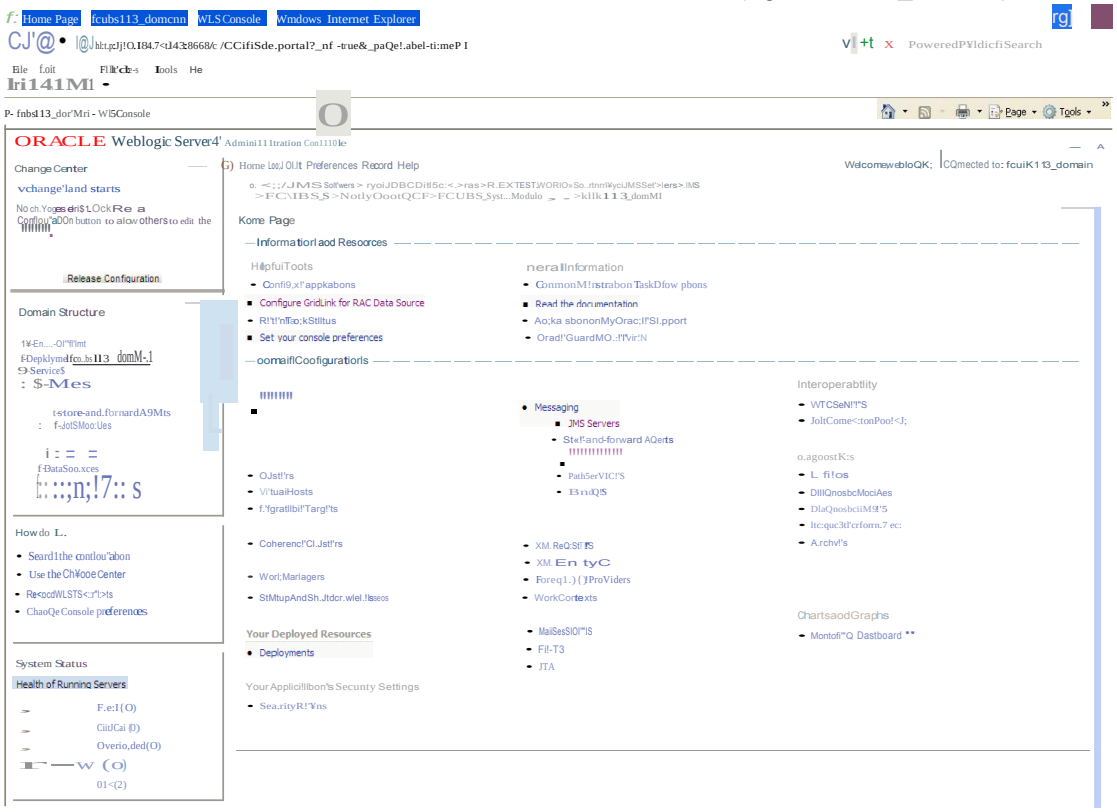
2. Under 'Server Start' tab > Arguments provide '-Dserver.id=1' – in case of Manage server.

   The following screen is displayed.

This attribute is used for Reference Number generation in payments module.

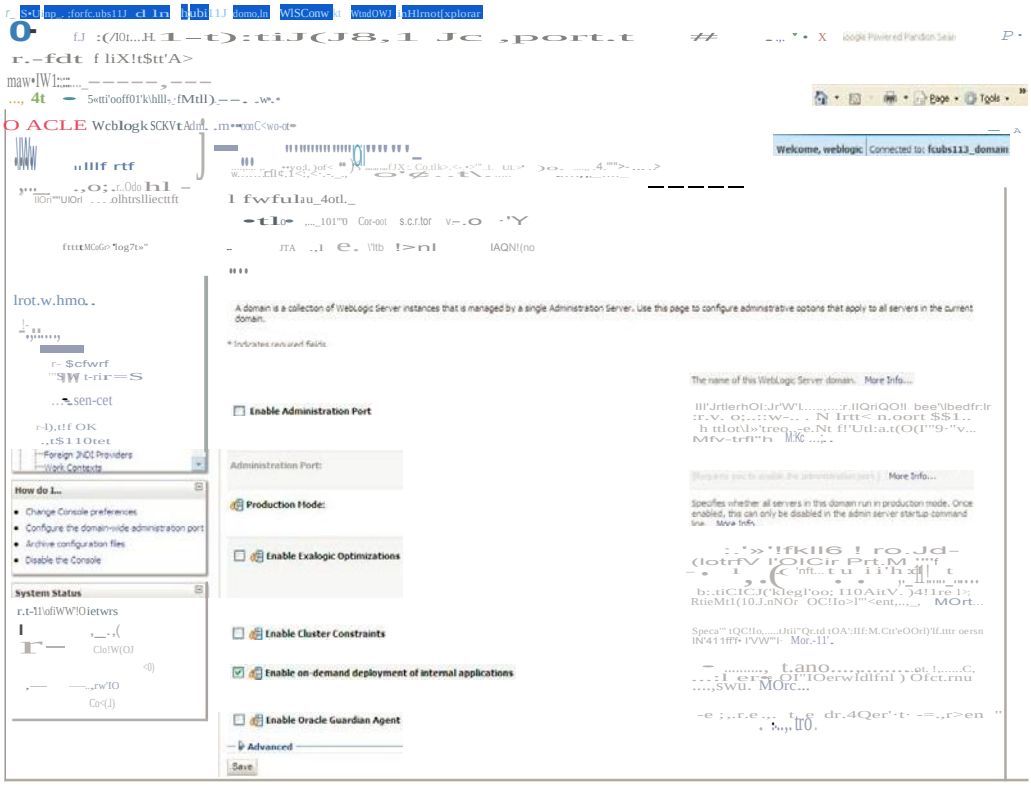


3. Select the domain from the domain structure as shown below. (Eg: fcubs113_domain).



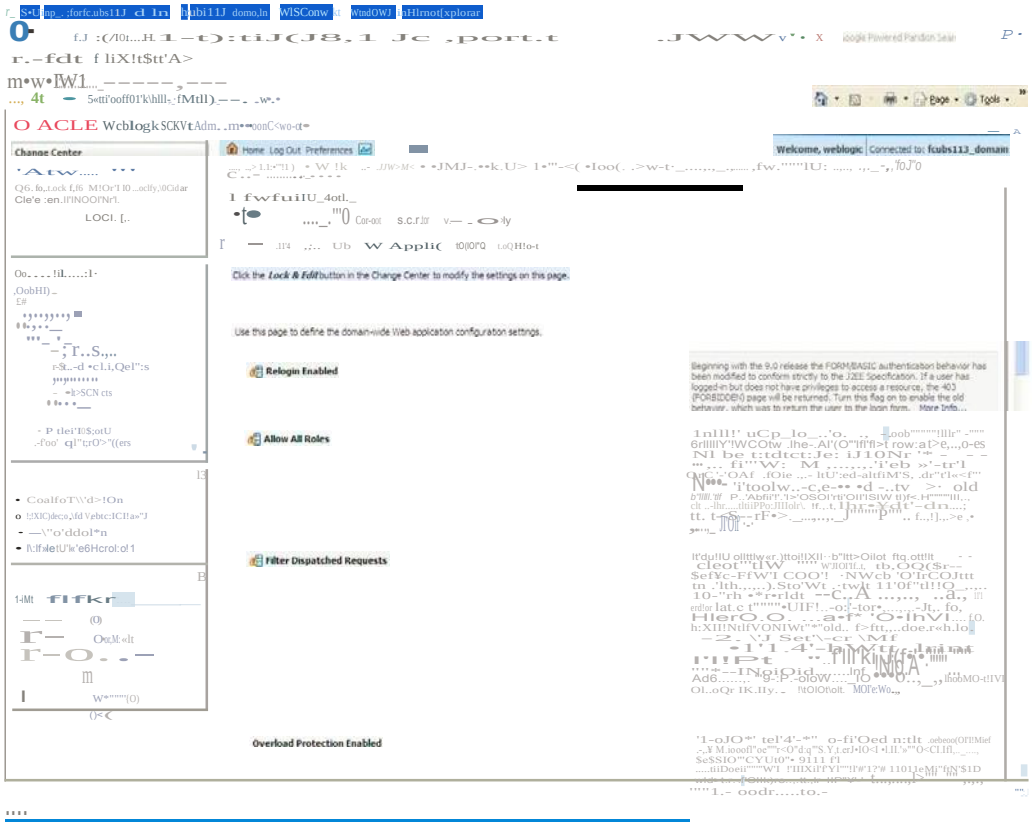
ORACLE

The following screen is displayed:

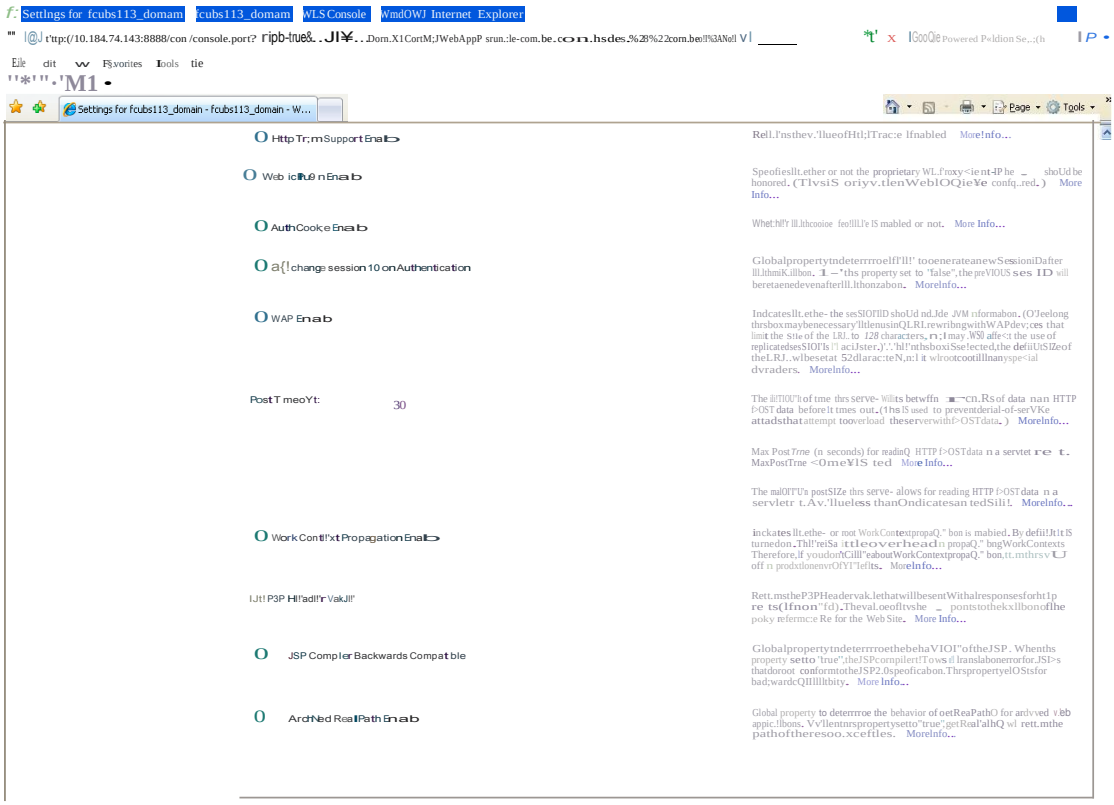


ORACLE

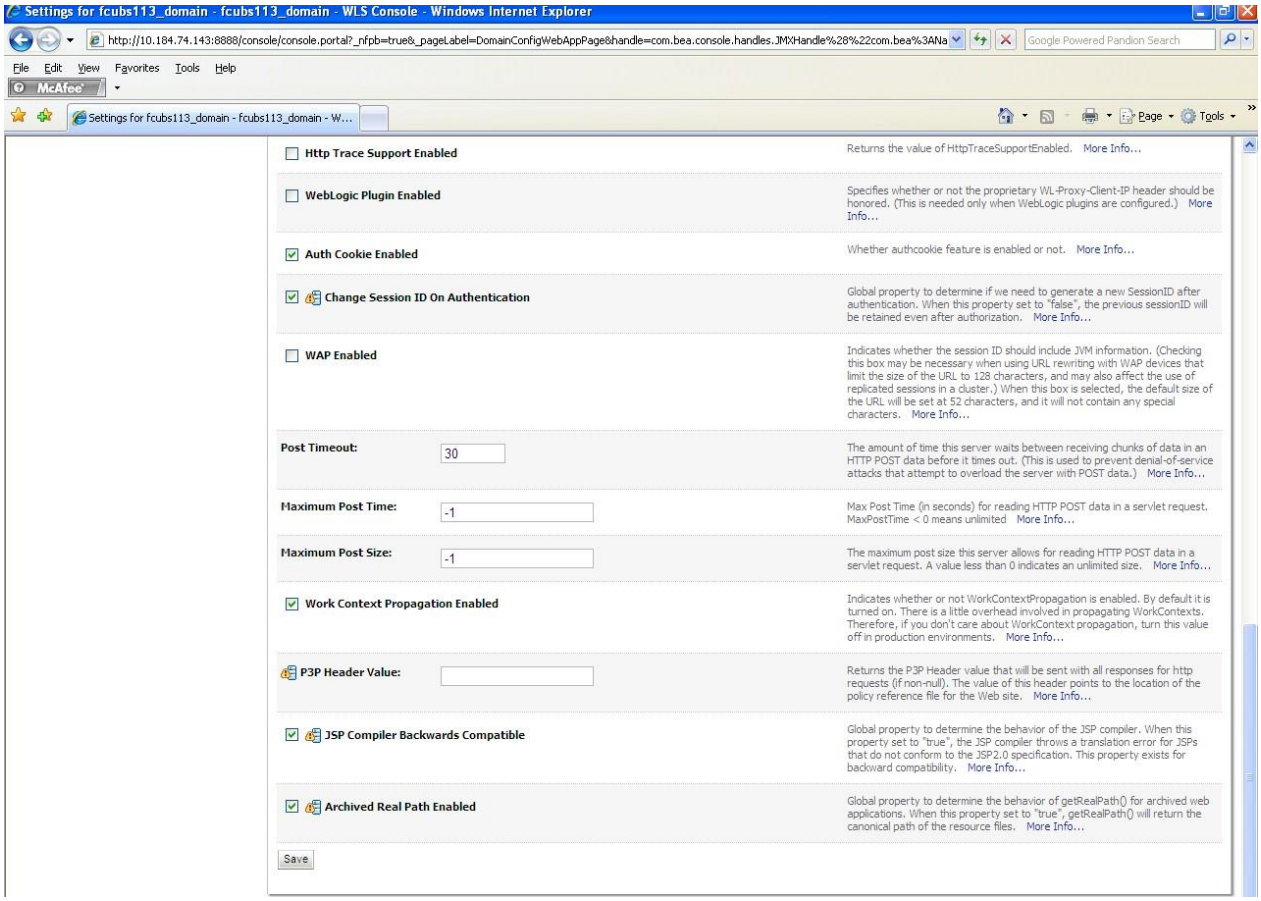
4. Under 'configuration' tab ,Select 'Web Applications'. The following screen is displayed.



5. Scroll down and ensure that the details are as shown in the figure. The remaining portion of the screen is given below:

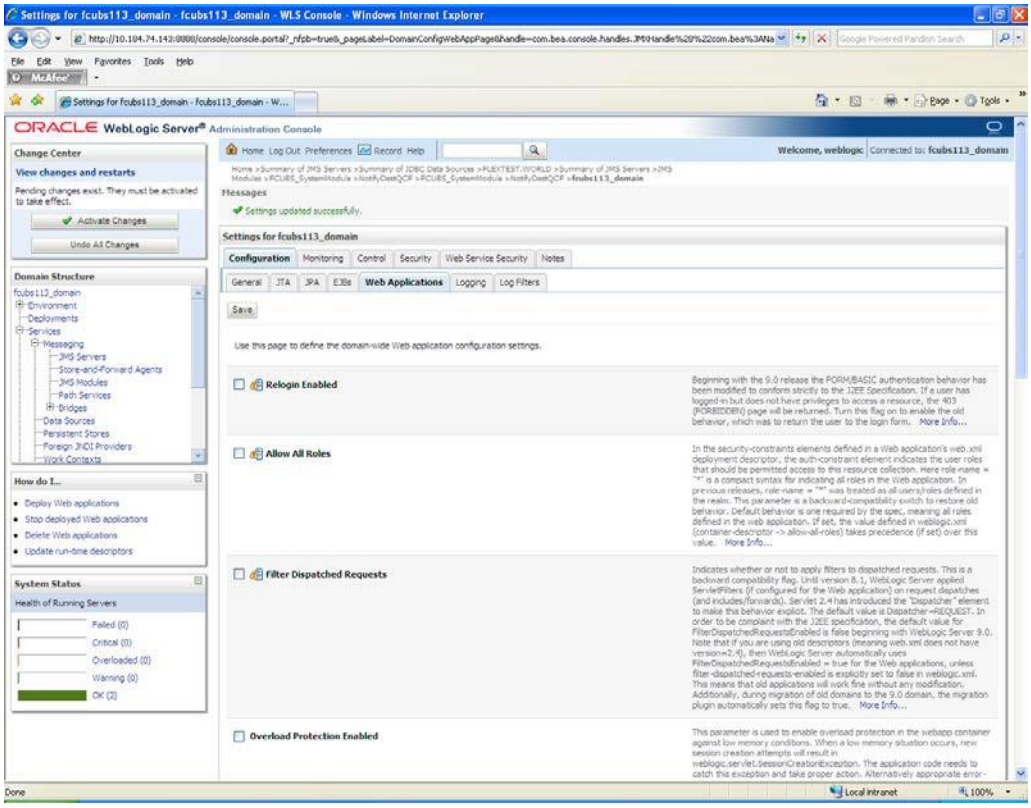

Settlngs for fcubs113_domam
fcubs113_domam
WLS Console
WmdOWJ Internet Explorer
t'ttp:(/10.184.74.143:8888/con /console.port? ripb-true&..JI¥...Dorn.X1CortM;JWebAppP srun.:le-com.be.con.hsdes.%2f%22corn.be!!%3ANo!l V
Eile dit w Fsxvorites Iools tie
''*''.'M1
Settings for fcubs113_domain - fcubs113_domain - W...
Page
Tools
O Http Tr;m Support Enab
Rell.l'nsthev.'IlueofHtl;lTrac:e lfnabled More!nfo...
O Web ic!ue nEnab
Speofiesllt.ether or not the proprietary WL.Proxy<ie nt-IP he _ shoUd be honored. (TlvsiS oriyv.tlenWeblOQie¥e confq..red.) More Info...
O AuthCooke Enab
Whet:hl!'r lll.lthcooioe feo!llLl'e IS mabled or not. More Info...
O a{!chang: session 10 onAuthentication
Globalpropertytndeterrroelf'Il!' tooenerateanewSessioniDafter lll.lthmiK.illbon. 1 -- "ths property set to "false", the preVIOUS ses ID will beretaenedevenafterlll.lthonzabon. MoreInfo...
O WAP Enab
Indcatesllt.ethe- the sesSIOI'lID shoUd nd.Jde JVM nformabon. (O'Jeelong thrsboxmaybenecessary'lltlenusinQLRI.rewritbngwithWAPdev;ces that limit the Stle of the LRJ..to 128 characters, n;I may.W$0 affe<:t the use of replicatedsesSIOI'Is''l aciJster.)'.'.'hl!'nthsboxiSse!ected,the defiiUtSIZeof theLRJ..wlbesetat 52dlarac:teN,n:l it wlrootcootilllnanyspe<ial dvraders. MoreInfo...
PostT meoYt: 30
The ilr'TIOU"h of tme thrs serve- Wiltts betwffn ■¬cn.Rs of data nan HTTP f>OST data before It tmes out.(1hs IS used to preventderial-of-serVKe attadsthat attempt tooverload theserverwithf>OSTdata.) MoreInfo...
Max Post Trne (n seconds) for readinQ HTTP f>OSTdata n a servtet re t. MaxPostTrne <Ome¥lS ted More Info...
The malOfl"U'n postSIZe thrs serve- alows for reading HTTP f>OSTdata n a servletr t.Av.'llueless thanOndicatesan tedSili!. MoreInfo...
O Work Contl!'xt Propagation Enab
inckates llt.ethe- or root WorkContextpropaQ." bon is mabied. By defiit.Jt:t IS turnedon.Thl!'reiSa tttleoverheadn propaQ." bngWorkContexts Therefore,if youdon'tCilll"eaboutWorkContextpropaQ." bon,tt.mthrsv UJ off n prodxtlonenvrOfYI"lefits. MoreInfo...
I.Jt! P3P Hl!'adl!'r Vak.ll!'
Rett.mstheP3PHeadervak.lethatwillbesentWithalresponsesforht1p re ts(Ifnon"fd).Theval.oeoflvsbe _ pontstothekxllbonofIhe poky reformc:e Re for the Web Site. More Info...
O JSP Compl er Backwards Compat ble
Globalpropertytndeterrrroethebeha VIOI"oftheJSP. Whenths property setto 'true",theJSPcornpilert'Towsth lranslaboneroforJSI>s thatdoroot conformtotheJSP2.0speoficabon.ThrspropertyelOStsfor bad;wardcQIIIIltbity. More Info...
0 ArdNed RealPath Enab
Global property to deterrroe the behavior of oetReaPathO for ardvved v.eb appic.!lbons. VVllentnrspropertysetto"true",getRealalhQ wl rett.mthe pathoftheresoo.xceftles. MoreInfo...

ORACLE



6. Check the options ‘JSP Compiler Backwards Compatible’ and ‘Archived Real Path Enabled’.
7. Click ‘Save’.

8. The following screen is displayed:



9. Ensure that the message 'Settings are updated successfully' is displayed.

10. Click the button 'Active Changes'.

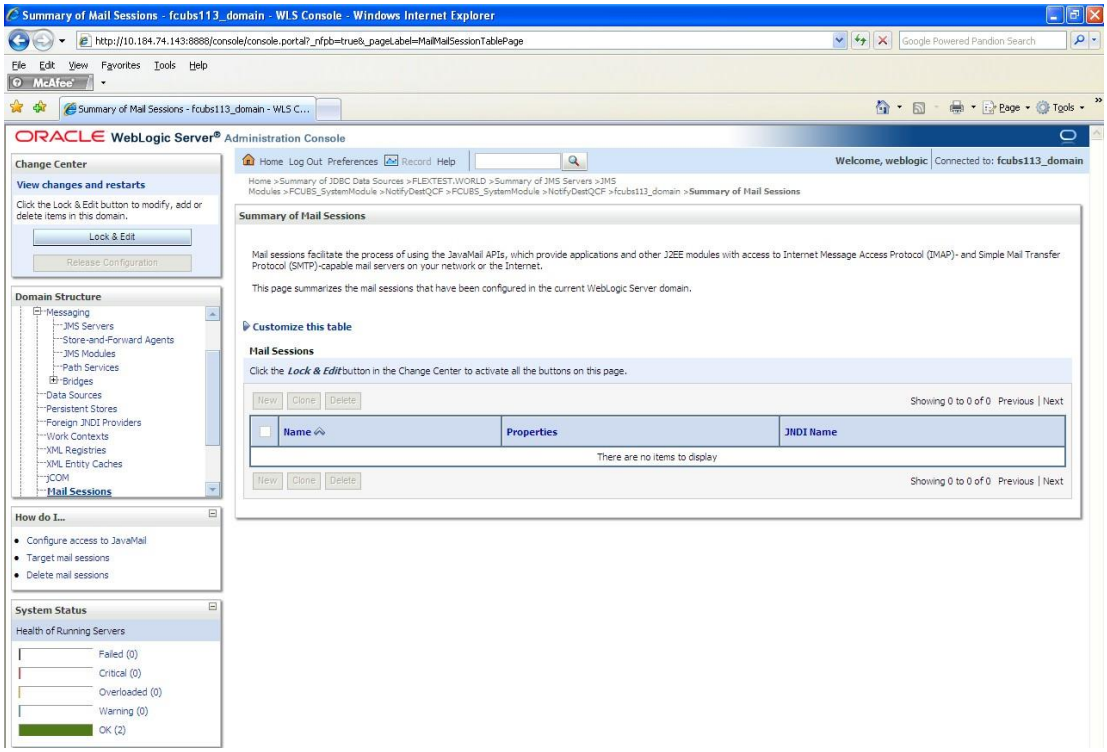
7.4 Setup/Configure Mail Session in Weblogic

This section describes the set of configurations changes required in Oracle Weblogic Server when Oracle Banking payments is configured to generate and send passwords to users via e-mail.

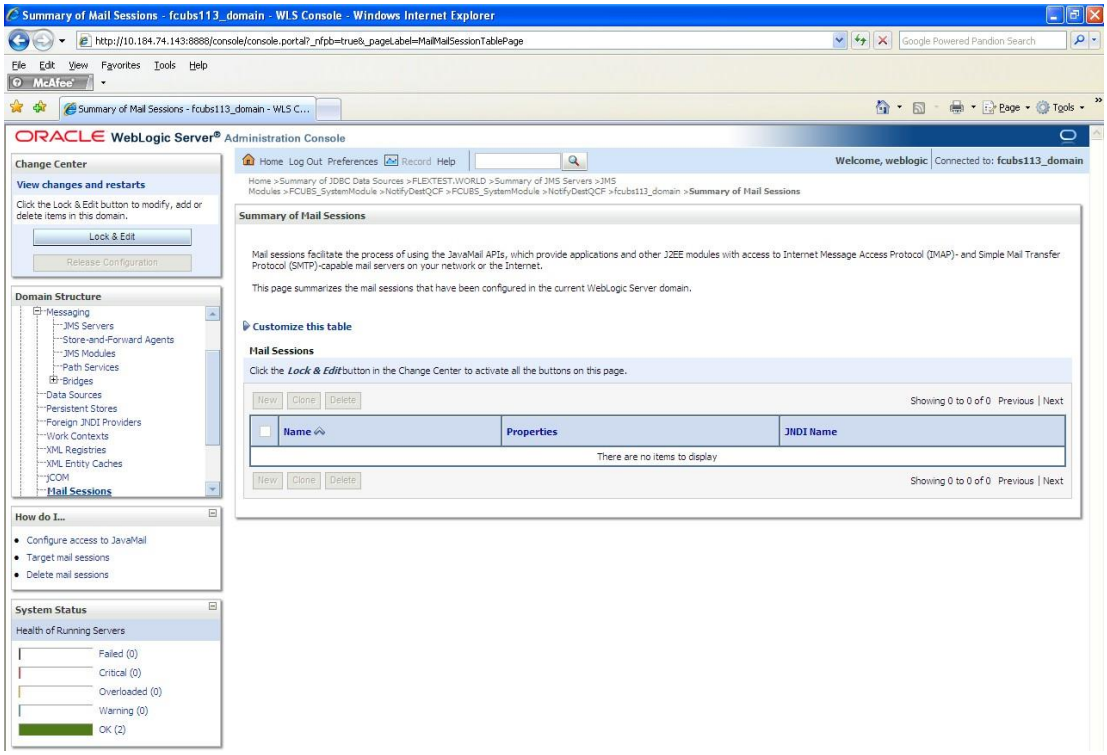
7.4.1 Creating JavaMail Session

To configure mail session, follow the steps below.

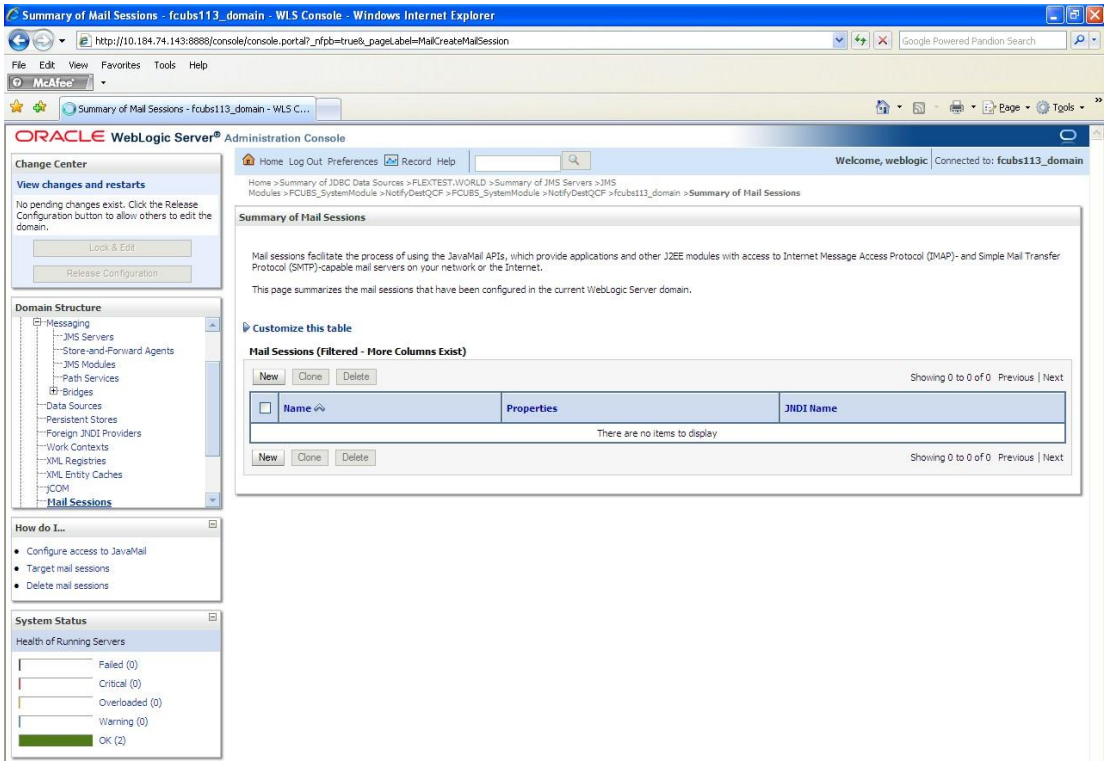
1. Expand ‘Services’ on the left pane of the application server. Click ‘Mail Sessions’.



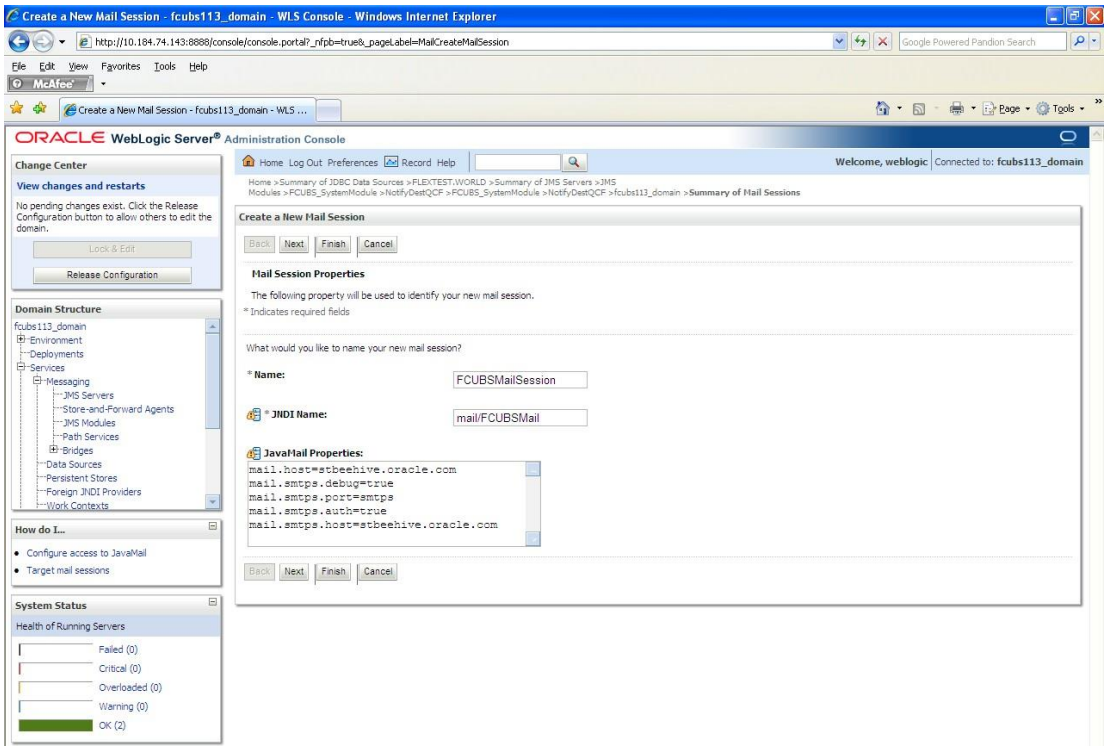
2. Click ‘Lock & Edit’.



3. Following screen is displayed; Click ‘New’ for creating a new session.



4. Following screen is displayed.



5. Specify the required details to create a session. Sample details are given below:

Name

FCUBSMailSession

JNDI Name

mail/FCUBSMail

 This JNDI name needs to be maintained in fcubs.properties file with encrypted format.

Java Mail Properties

mail.host=<HOST_MAIL_SERVER>

Eg: samplename.mail.com

mail.smtps.port=<SMTPS_SERVER_PORT> Eg: 1010

mail.transport.protocol=<MAIL_TRANSFER_PROTOCOL>

Eg: smtps

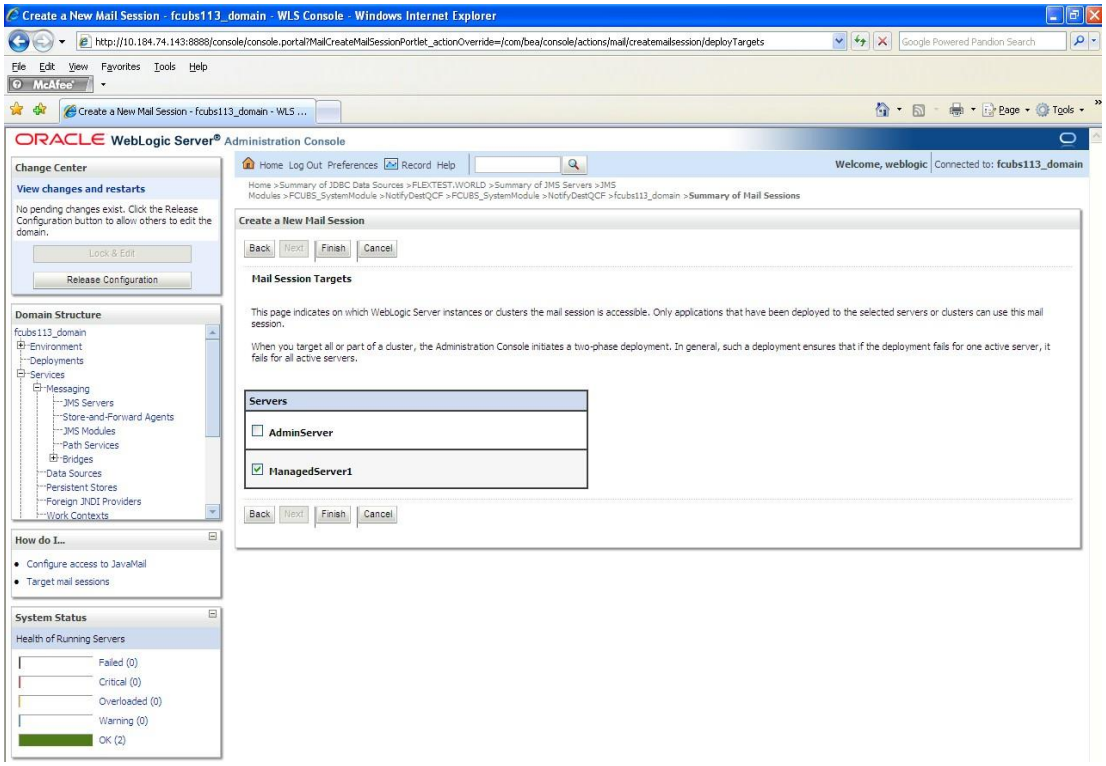
mail.smtps.auth=true

mail.smtps.host==<HOST_SMTPS_MAIL_SERVER>

Eg: samplename.mail.com

6. Click ‘Next’.

The following screen is displayed.



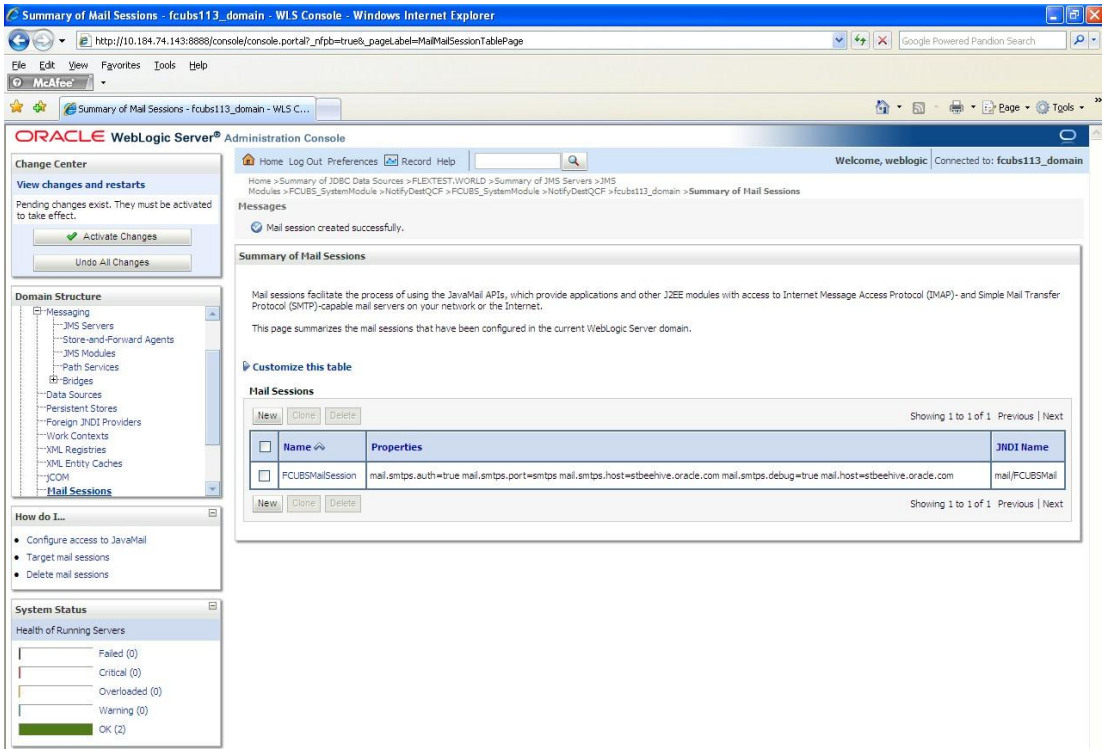
7. Check the box against the required servers and click ‘Finish’ to complete the configuration.

‘fcubs.properties’ file needs to be updated with the encrypted values of

- SMTP_HOST
- SMTP_USER
- SMTP_PASSWORD
- SMTP_JNDI

This can be achieved using the Oracle Banking UBS Installer.

8. Click ‘Active Changes’ button to activate the current mail session settings.



7.4.2 Configuration of the TLS/SSL Trust Store for Weblogic Server

As described in the previous section, Oracle Banking payments uses SMTPS to send outgoing mails. SMTPS uses SSL to ensure transport-level security of the mail messages and hence, the certificate of the mail server needs to be imported into the trust store(s) of the Managed Servers where Oracle Banking payments is deployed.

The certificate of the mail server needs to be specifically imported into the trust store configured for the Managed Server(s), as configured in the Oracle Banking payments Installation guide titled ‘SSL Configuration On Weblogic’ (SSL_Configuration).

For further details on importing the certificate of the mail server into the trust store, refer to the documentation for the Sun Java keytool utility (Key and Certificate Management tool).

Oracle Banking Payments Weblogic Configuration
[May] [2020]
Version 14.4.0.0.0

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
www.oracle.com/financialservices/