

WebLogic Configuration
Oracle FLEXCUBE Investor Servicing
Release 14.7.0.0.0
[November] [2022]



Table of Contents

1.	CONFIGURING SSL ON ORACLE WEBLOGIC	1-1
1.1	INTRODUCTION	1-1
1.2	SETTING UP SSL ON ORACLE WEBLOGIC	1-1
1.3	CERTIFICATES AND KEYPAIRS	1-1
2.	CHOOSING THE IDENTITY AND TRUST STORES.....	2-1
2.1	INTRODUCTION	2-1
3.	OBTAINING THE IDENTITY STORE	3-1
3.1	CREATING IDENTITY STORE WITH SELF-SIGNED CERTIFICATES	3-1
3.1.1	<i>Creation of Self-signed Certificate</i>	<i>3-1</i>
3.2	CREATING IDENTITY STORE WITH TRUSTED CERTIFICATES ISSUED BY CA.....	3-3
3.2.1	<i>Creation of Public and Private Key Pair.....</i>	<i>3-3</i>
3.2.2	<i>Generating CSR.....</i>	<i>3-5</i>
3.2.3	<i>Obtaining Trusted Certificate from CA</i>	<i>3-5</i>
3.2.4	<i>Importing Certificate into Identity Store.....</i>	<i>3-5</i>
4.	CONFIGURING IDENTITY AND TRUST STORES FOR WEBLOGIC	4-1
4.1	ENABLING SSL ON ORACLE WEBLOGIC SERVER.....	4-1
4.2	CONFIGURING IDENTITY AND TRUST STORES	4-1
5.	SETTING SSL ATTRIBUTES FOR MANAGED SERVERS	5-1
5.1	SETTING SSL ATTRIBUTES FOR PRIVATE KEY ALIAS AND PASSWORD.....	5-1
6.	TESTING CONFIGURATION.....	6-1
6.1	TESTING CONFIGURATION	6-1
7.	CREATING RESOURCES ON WEBLOGIC	7-1
7.1	INTRODUCTION	7-1
7.2	RESOURCE ADMINISTRATION	7-1
7.2.1	<i>Creating Data Source</i>	<i>7-1</i>
7.2.2	<i>JMS Server Creation.....</i>	<i>7-19</i>
7.2.3	<i>JMS Modules Creation</i>	<i>7-25</i>
7.2.4	<i>Sub Deployment Creation.....</i>	<i>7-29</i>
7.2.5	<i>JMS Queue Creation.....</i>	<i>7-33</i>
7.2.6	<i>JMS Connection Factory Creation</i>	<i>7-38</i>
7.3	CONFIGURING WEBLOGIC FOR ORACLE FLEXCUBE.....	7-43
7.4	SETUP/CONFIGURE MAIL SESSION IN WEBLOGIC	7-47
7.4.1	<i>Creating JavaMail Session</i>	<i>7-47</i>
7.4.2	<i>Configuration of the TLS/SSL Trust Store for Weblogic Server</i>	<i>7-51</i>

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 2048 -sigalg  
SHA256withRSA -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:

3. **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.
4. **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.
5. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE INSTALLER SERVICING, for instance, `www.example.com`
6. **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.
7. **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.

8. **Name of your City or Locality:** The city in which your organization is physically located, for example Mumbai.
9. **Name of your State or Province:** The state/province in which your organization is physically located, for example Maharashtra.
10. **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 2048 bits, the signature algorithm as SHA256withRSA, 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:

```
keytool -genkey -keystore FCUBSKeyStore.jks -alias cvrhp0729 -dname
"CN=10.10.10.10,OU=OFSS" -keyalg "RSA" -sigalg "SHA256withRSA" -
keysize 2048 -validity 1000
```

```
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
```

```
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, 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 2048 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:

7. **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.
8. **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.
9. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE INSTALLER SERVICING, for instance, www.example.com
10. **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.

11. **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.
12. **Name of your City or Locality:** The city in which your organization is physically located, for example Mumbai.
13. **Name of your State or Province:** The state/province in which your organization is physically located, for example Maharashtra.
14. **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:

```
keytool -genkey -keystore FCUBSKeyStore.jks -alias cvrhp0729 -dname
"CN=10.10.10.10,OU=OFSS" -keyalg "RSA" -sigalg "SHA256withRSA" -
keysize 2048 -validity 1000
```

```
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
```

```
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, 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

```
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 CA 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

```
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,

4. **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.
5. **cert_file** is the location of the file containing the PKCS#7 formatted reply from the CA, containing the signed certificate.
6. **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

```
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 recommended that the Oracle FLEXCUBE INSTALLER SERVICING 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 FCIS 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.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 for Gateway Application (MDB)

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.

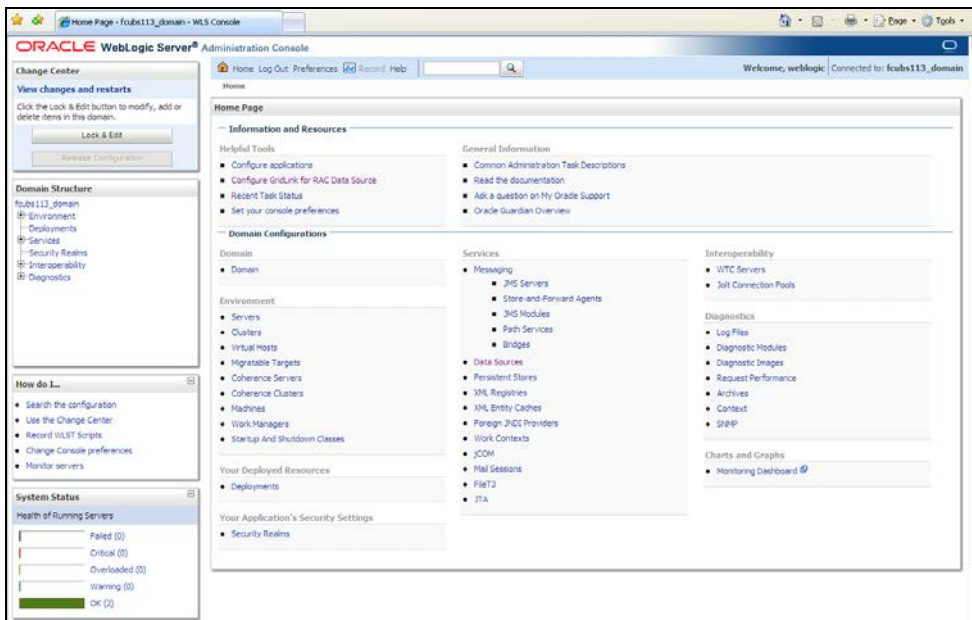
<http://10.10.10.10:1001/console>

Eg: <http://10.10.10.10:1001/console>



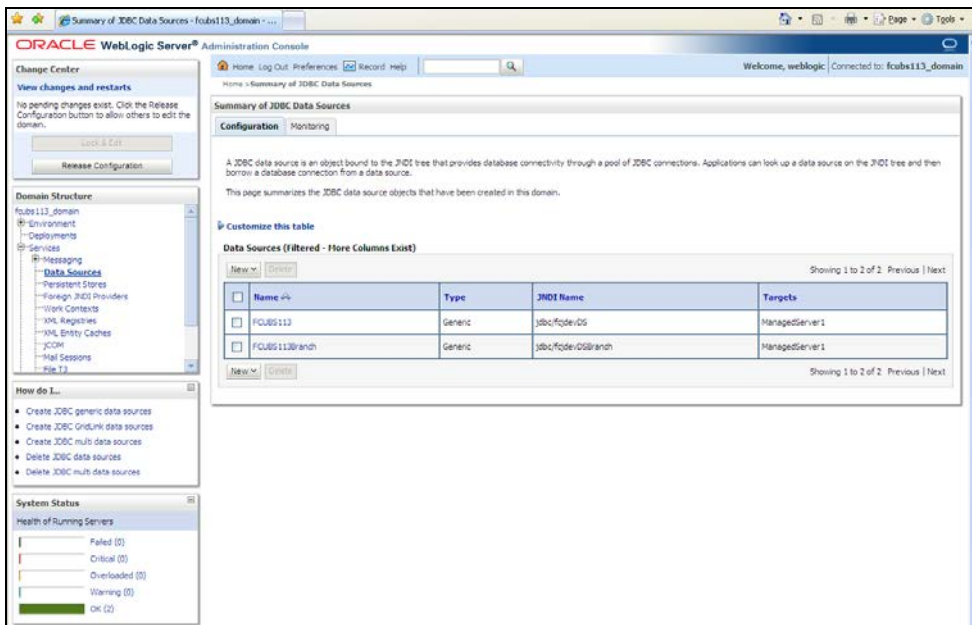
1. Specify the Weblogic administrator user name and password. Click 'Log In'.

Navigate to Oracle Weblogic home page.

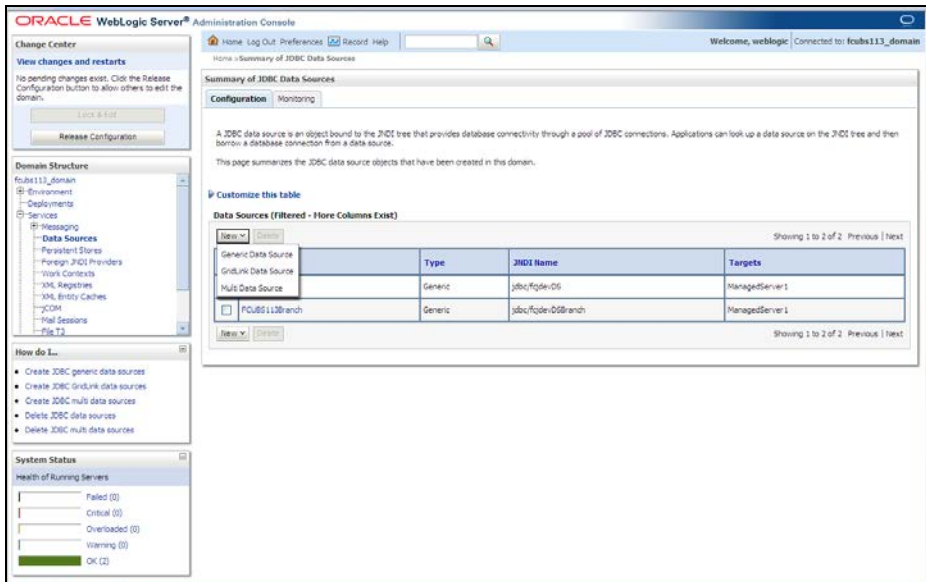


2. Click 'LOCK & EDIT'.

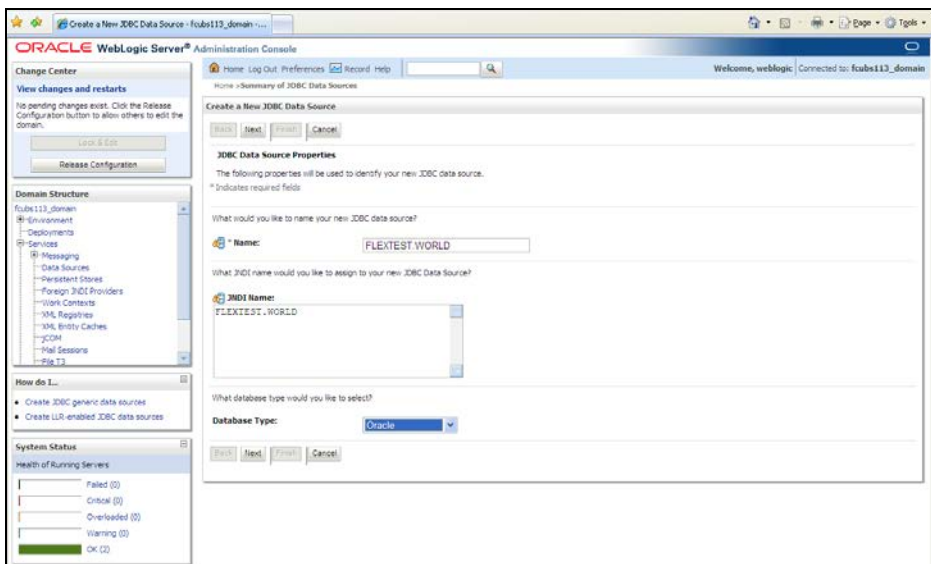
Following screen is displayed:



3. Expand 'Services' and then 'Data Sources' under it. Click 'Lock & Edit' button.



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

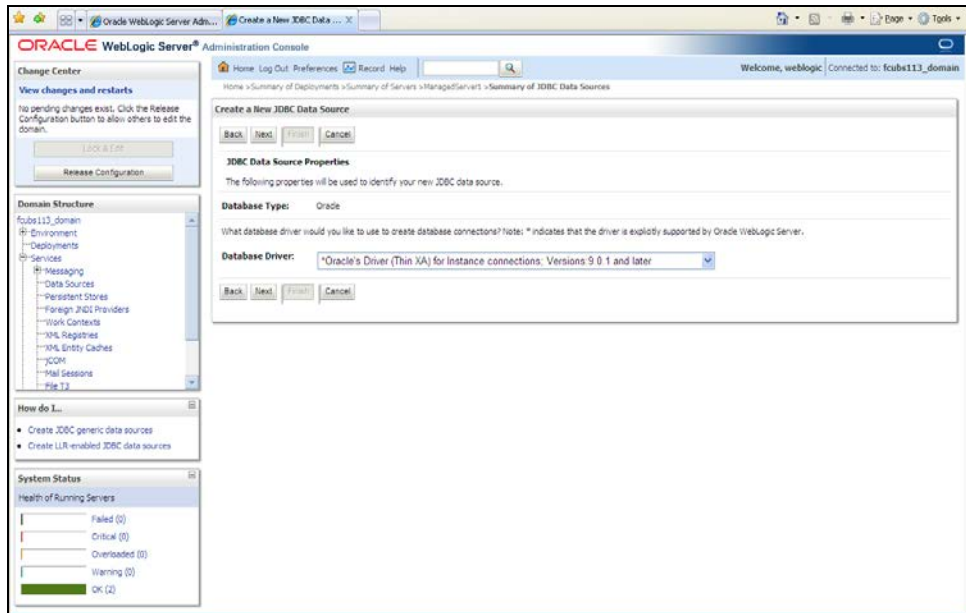


- Specify the following details:

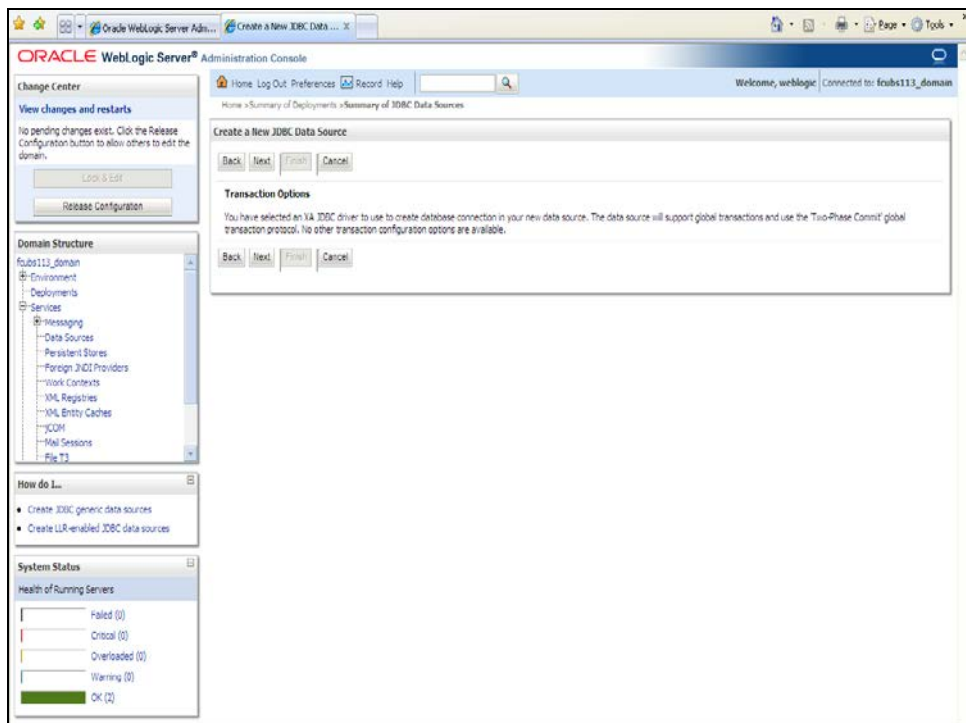
JDBC Datasource Name	FLEXTEST.WORLD
JNDI Name	FLEXTEST.WORLD
Database Type	Oracle

- Click 'Next'.

The following screen is displayed:



7. Select the XA database driver as shown in the figure. Click 'Next'.



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

The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create a New JDBC Data Source - fcuhs113_domain'. The left sidebar contains a 'Domain Structure' tree with 'fcuhs113_domain' selected, and a 'System Status' section showing 'Health of Running Servers' with 'OK (2)'. The main content area is titled 'Create a New JDBC Data Source' and has a 'Next' button highlighted. The 'Connection Properties' section is active, showing fields for 'Database Name' (KERDEV2), 'Host Name' (10.10.10.10), 'Port' (1010), 'Database User Name' (FCPB1121), and 'Password' (masked with asterisks). The 'Confirm Password' field is also masked with asterisks. The 'Next' button is highlighted.

9. This screen defines the connection properties. Set the details as shown below:

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'.

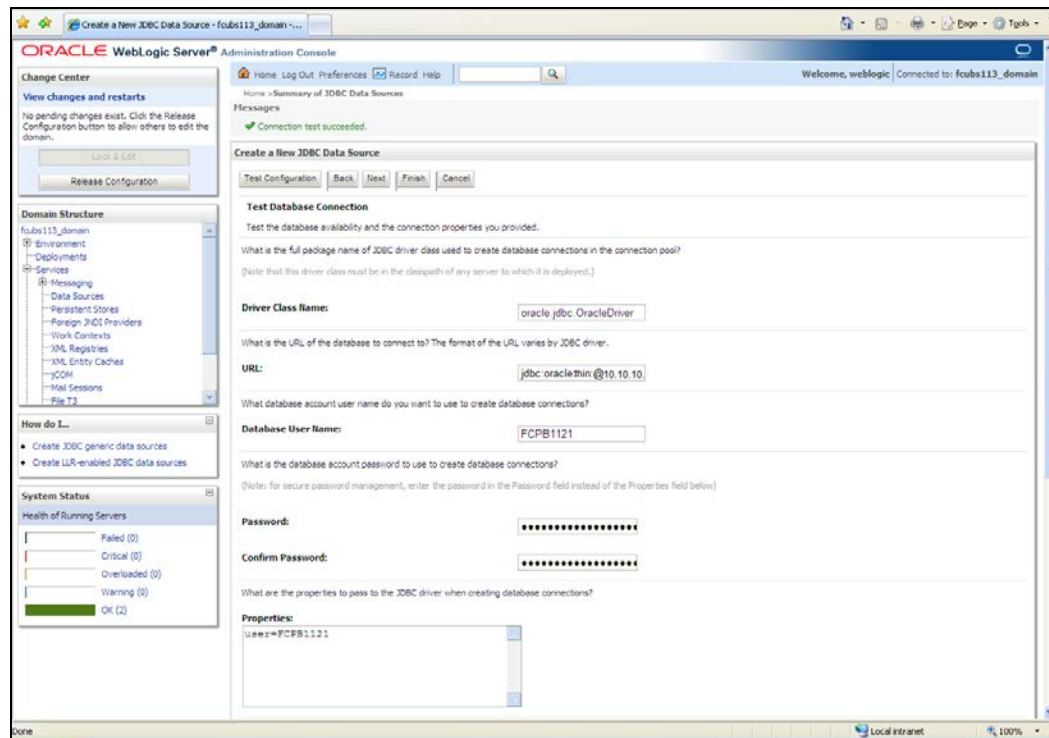
The following screen is displayed.

The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create a New JDBC Data Source - DefaultDomain'. The left sidebar contains a 'Domain Structure' tree with 'DefaultDomain' selected, and a 'System Status' section showing 'Health of Running Servers' with 'OK (1)'. The main content area is titled 'Test Database Connection' and has a 'Test Configuration' button highlighted. The 'Test Database Connection' section is active, showing fields for 'Driver Class Name' (oracle.jdbc.OracleDriver), 'URL' (jdbc:oracle:thin:@10.10.10.10), 'Database User Name' (FCPB1121), and 'Password' (masked with asterisks). The 'Confirm Password' field is also masked with asterisks. The 'Properties' section is active, showing a text area for 'user=FCPB1121'. The 'Test Configuration' button is highlighted.

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

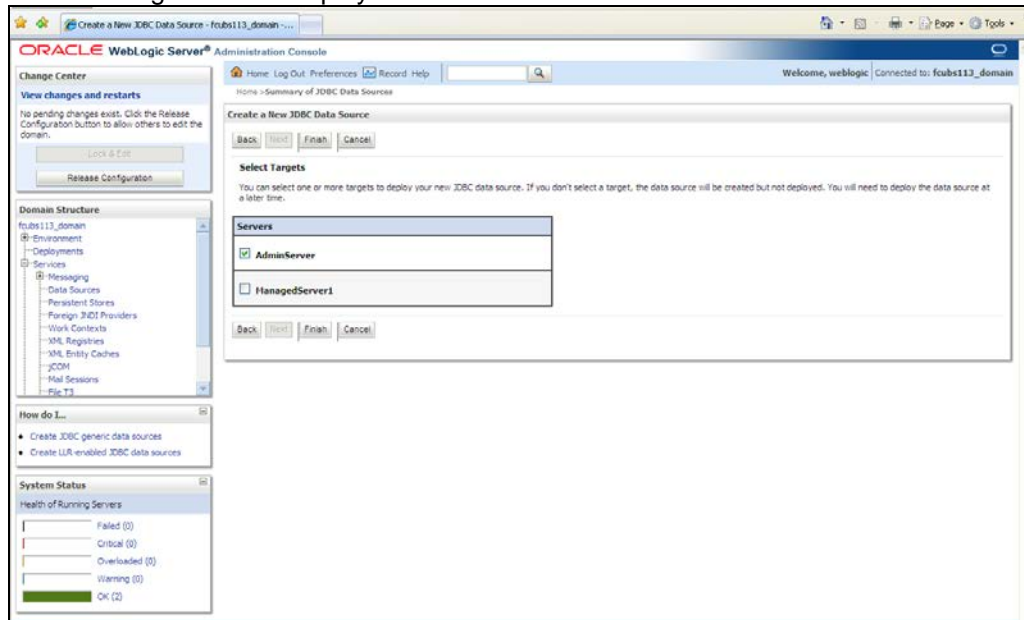
13. Specify the URL.
14. jdbc:oracle:thin:@10.10.10.10:1001:<INSTANCE_NAME>Specify the Database Username (Eg: FCPB1121) and password.
15. Confirm the password.
16. Click 'Test Configuration' tab.

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

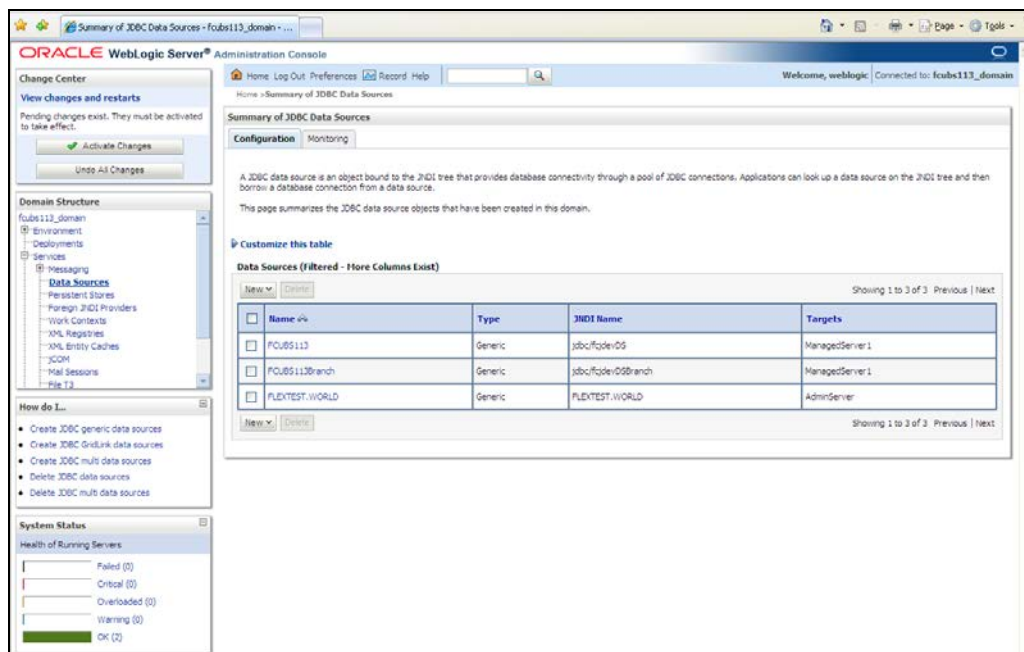


17. Click 'Next'.

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.

The screenshot shows the Oracle WebLogic Server Administration Console. The left pane displays the Domain Structure with 'Data Sources' selected. The main pane shows the 'Summary of JDBC Data Sources' page. A message at the top states: 'All changes have been activated. No restarts are necessary.' Below this, there is a table of Data Sources. The table has columns: Name, Type, JNDI Name, and Targets. The table lists three data sources: FOCUS113, FOCUS113Branch, and FLEXTTEST.WORLD. The FLEXTTEST.WORLD data source is highlighted.

Name	Type	JNDI Name	Targets
FOCUS113	Generic	jdbc/foctenOS	ManagedServer1
FOCUS113Branch	Generic	jdbc/foctenOSBranch	ManagedServer1
FLEXTTEST.WORLD	Generic	FLEXTTEST.WORLD	AdminServer

20. 'FLEXTTEST.WORLD' datasource has been created.

7.2.1.3 Non-XA Enabled Data Source For FCIS Application

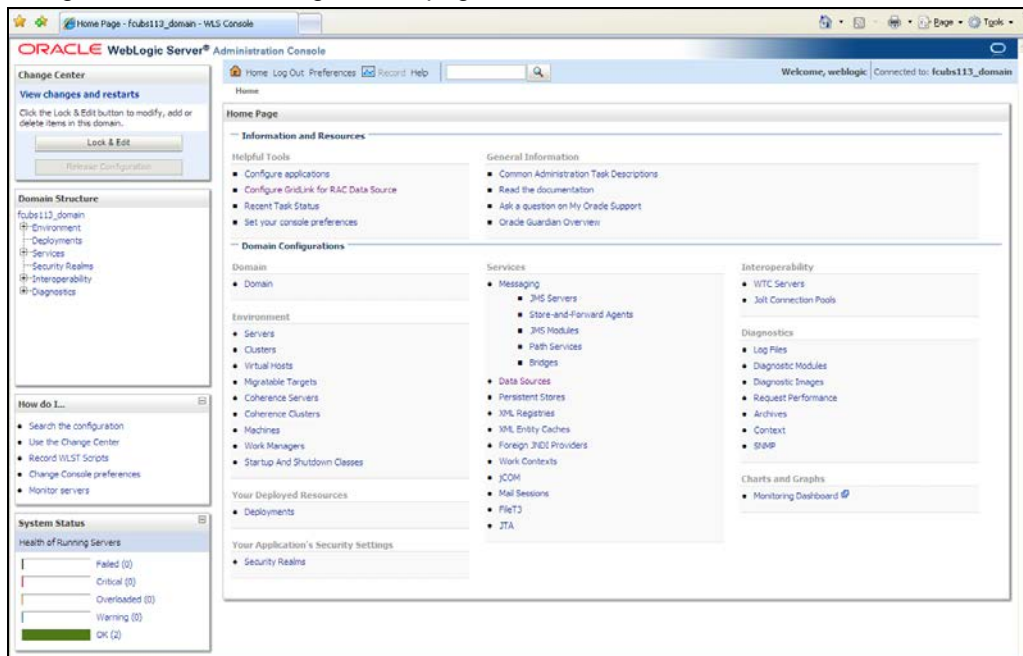
1. Follow the steps given below: Start the Administrative Console of Weblogic application server. You can start this by entering Oracle Weblogic Admin Console URL in the address bar in an internet browser.

http:10.10.10.10:1001/console Eg: <http://10.10.10.10:1001/console>

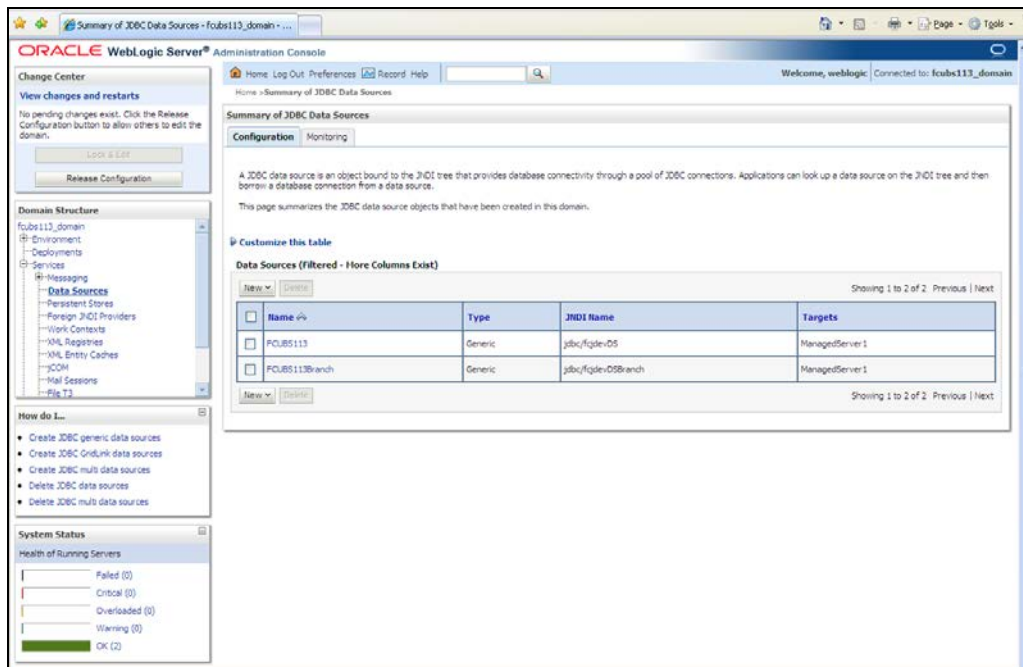


2. Specify the Weblogic administrator user name and password. Click 'Log In'.

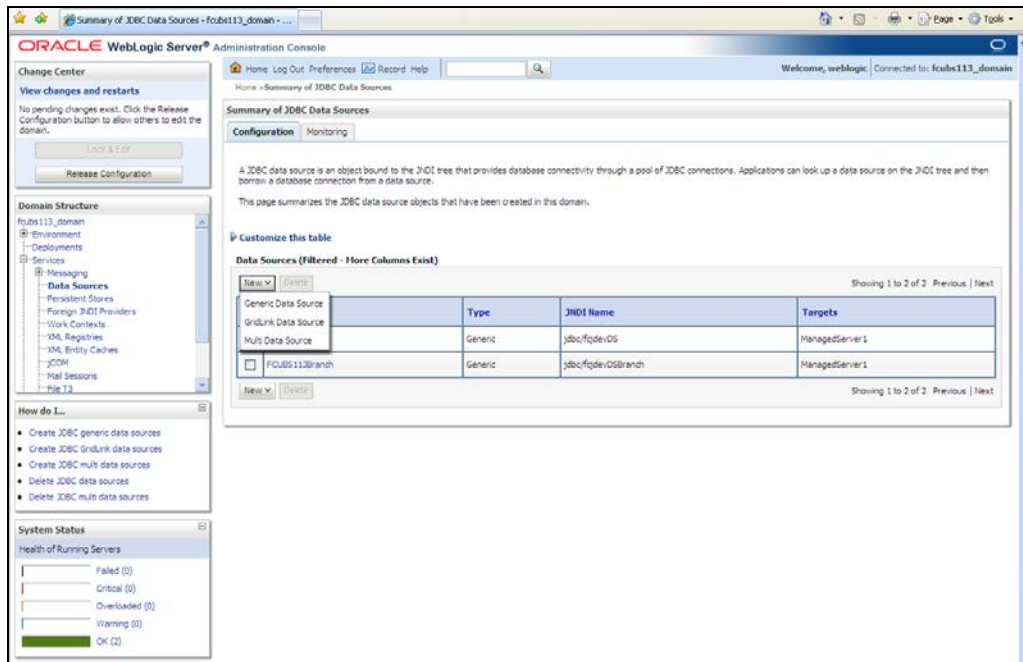
3. Navigate to Oracle WebLogic home page.



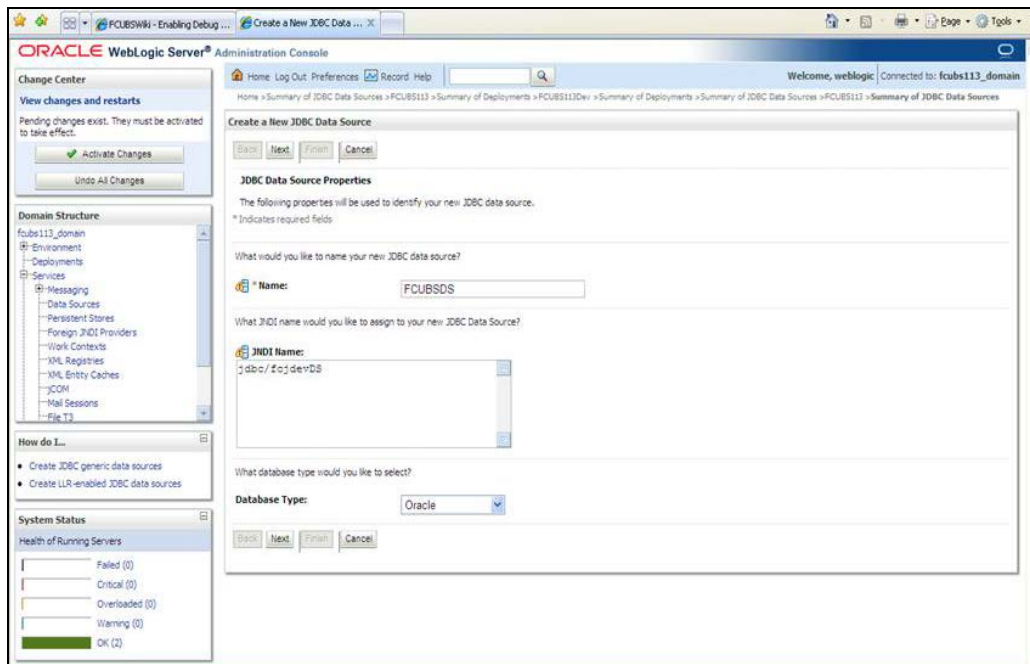
The following screen is displayed:



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



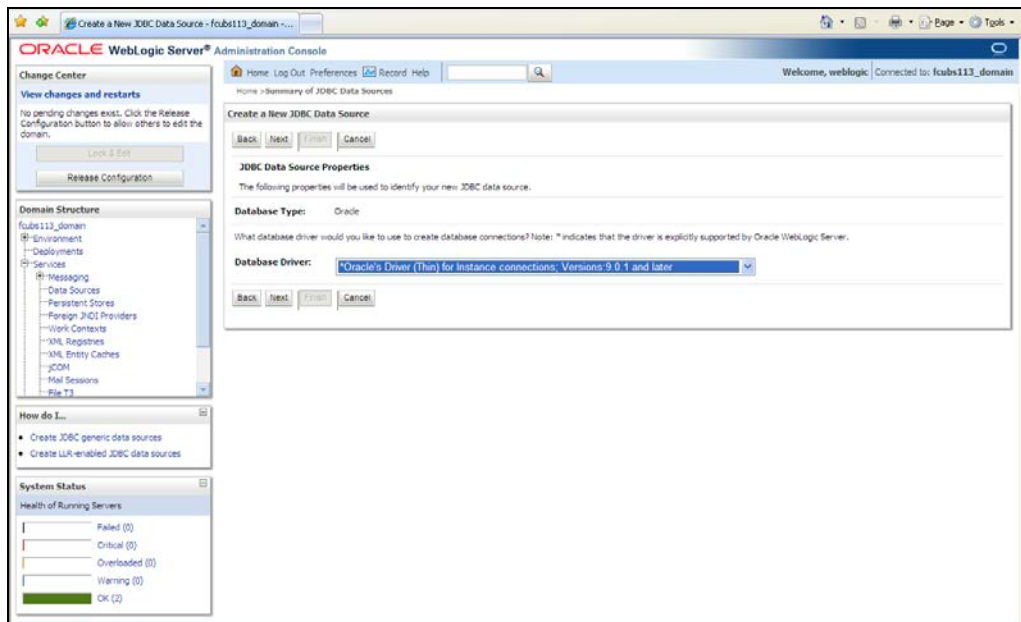
5. To create a new data source, click 'New' and select 'Generic Data Source'.



6. Specify the following details:

JDBC Datasource Name	FCISDS
JNDI Name	jdbc/fcjdevDS
Database Type	Oracle

7. Click 'Next'.



8. Select the database driver as shown in the figure. Click 'Next'.

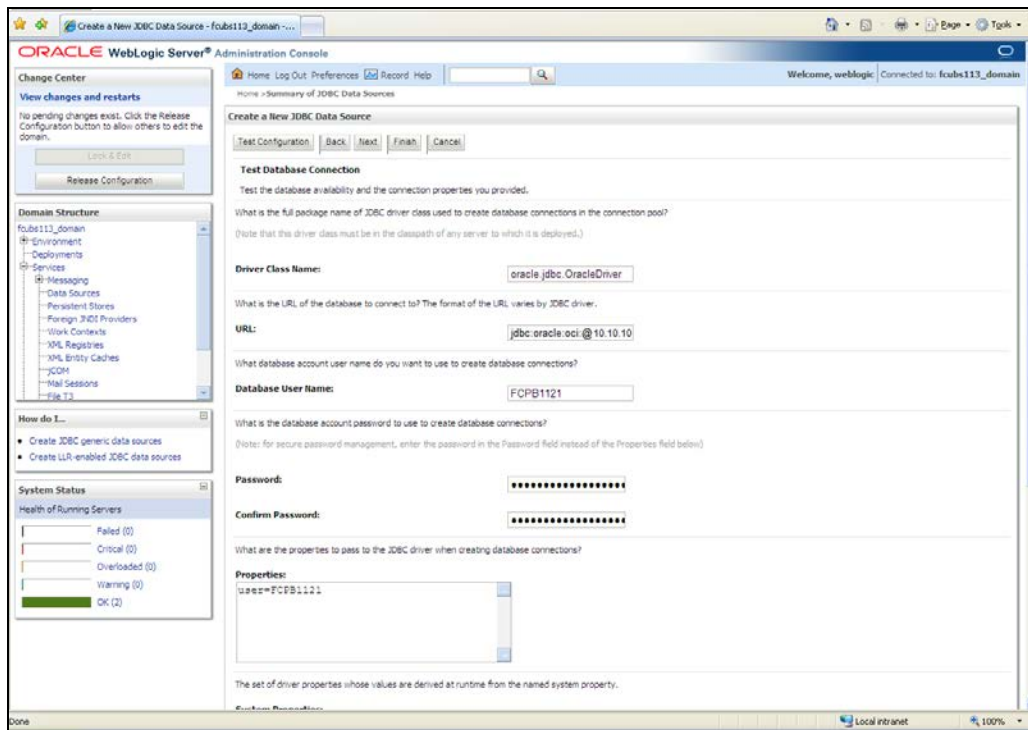
Following screen is displayed:

The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create a New JDBC Data Source - fcuhs113_domain...'. The left sidebar contains a 'Domain Structure' tree with 'fcuhs113_domain' expanded, showing 'Environment', 'Deployments', 'Services', 'Data Sources', 'Persistent Stores', 'Foreign J2EI Providers', 'Work Contexts', 'XSL Registries', 'XSL Entry Caches', 'JCDM', 'Mail Sessions', and 'File FS'. Below this is a 'How do I...' section with links to 'Create JDBC generic data sources' and 'Create LJR-enabled JDBC data sources'. The 'System Status' section shows 'Health of Running Servers' with a green bar indicating 'OK (2)'. The main content area is titled 'Create a New JDBC Data Source' and has a 'Next' button. The 'Transaction Options' section is active, showing a message: 'You have selected non-XA JDBC driver to create database connection in your new data source. Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.' There are three radio buttons: 'Supports Global Transactions' (unchecked), 'Logging Last Resource' (selected), and 'Emulate Two-Phase Commit' (unchecked). The 'Logging Last Resource' option has a description: 'Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the Logging Last Resource (LJR) transaction optimization. Recommended in place of Emulate Two-Phase Commit.' The 'Emulate Two-Phase Commit' option has a description: 'Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application can tolerate heuristic conditions.' At the bottom of the section are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

9. Select Logging Last Resource then uncheck 'Support Global Transactions'. Click 'Next'. The following screen is displayed:

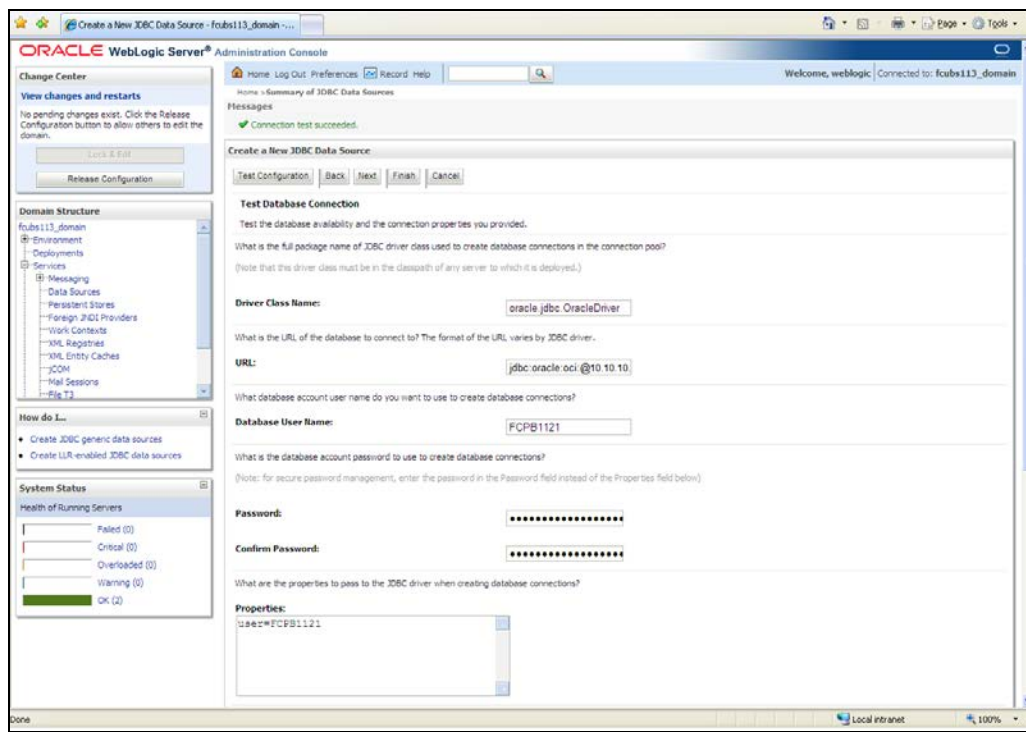
The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create a New JDBC Data Source - fcuhs113_domain...'. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Create a New JDBC Data Source' and has a 'Next' button. The 'Connection Properties' section is active, showing a message: 'Define Connection Properties. What is the name of the database you would like to connect to?'. There are several input fields: 'Database Name' (value: KERDEV2), 'Host Name' (value: 10.10.10.10), 'Port' (value: 1010), 'Database User Name' (value: FCPB1121), 'Password' (masked with asterisks), and 'Confirm Password' (masked with asterisks). At the bottom of the section are 'Back', 'Next', 'Finish', and 'Cancel' buttons.

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.

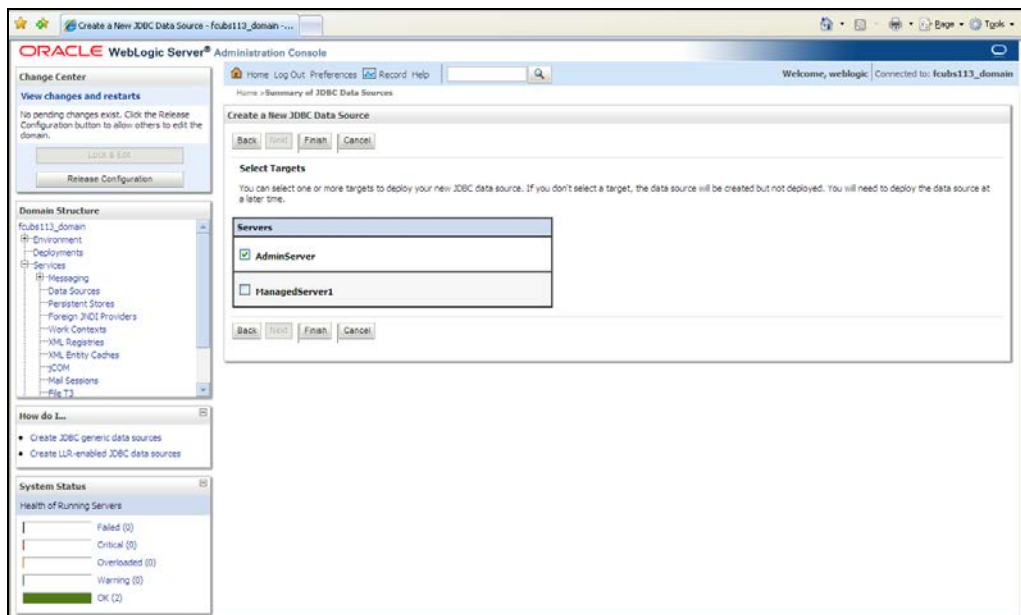


13. Specify the Driver Class Name (Eg: oracle.jdbc.OracleDriver)
14. Specify the URL.
Default URL: jdbc:oracle:thin:@10.10.10.10:1001:<INSTANCE_NAME>.
Change the default URL to: jdbc:oracle:oci:@10.10.10.10:1010:<INSTANCE_NAME>
15. Specify the Database Username (Eg: testdb) and password.
16. Confirm the password.
17. Click 'Test Configuration' tab.

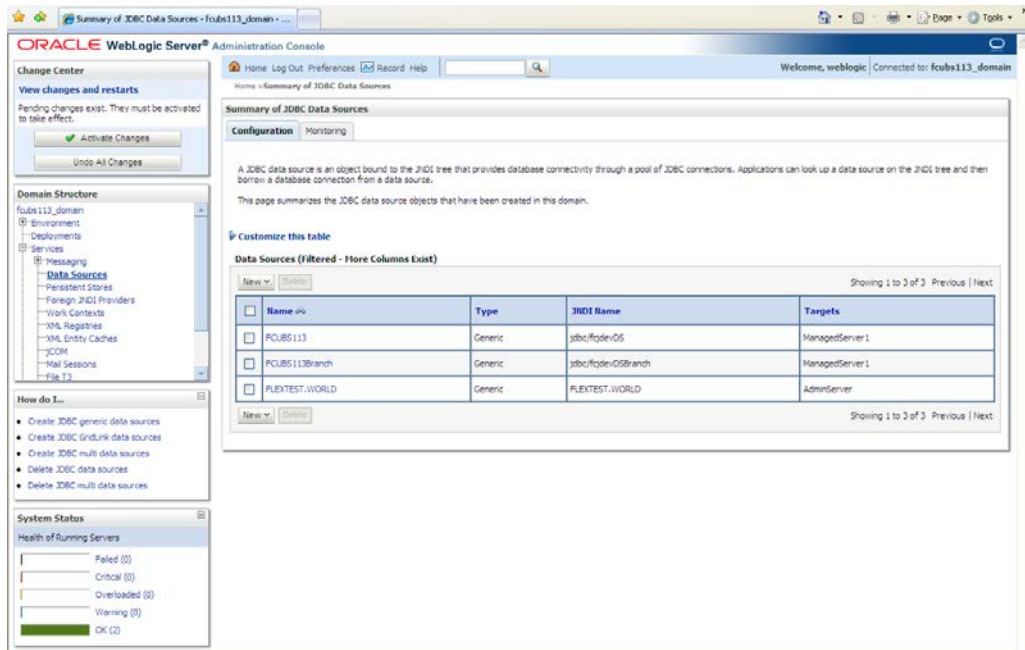
18. If the connection is established successfully, the message 'Connection test succeeded' is displayed.



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

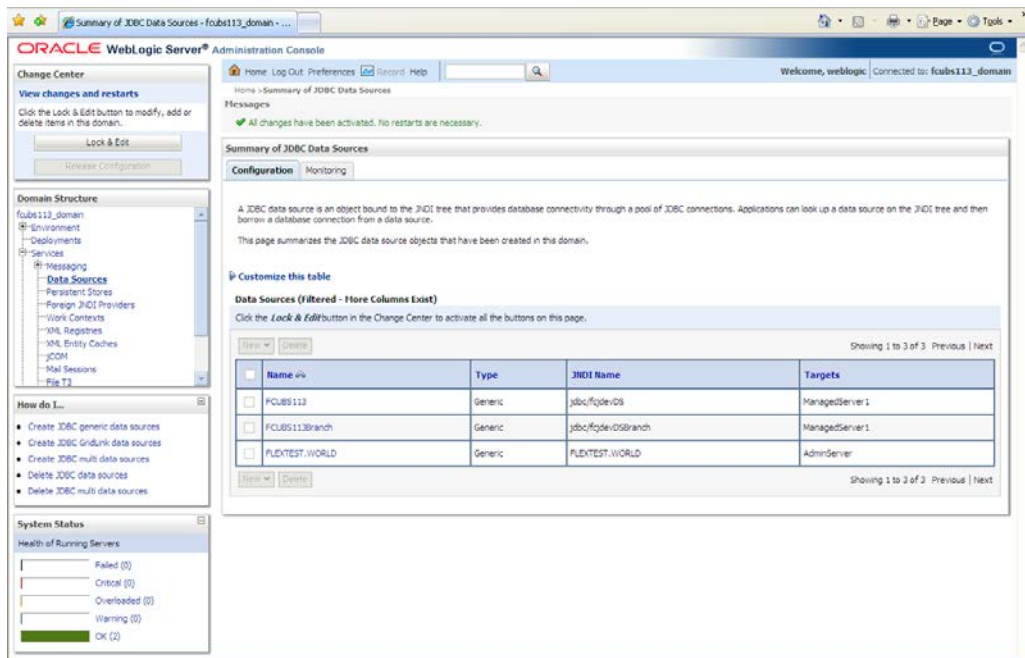


20. Check the boxes against the required servers. Click 'Finish'. The following screen is displayed:



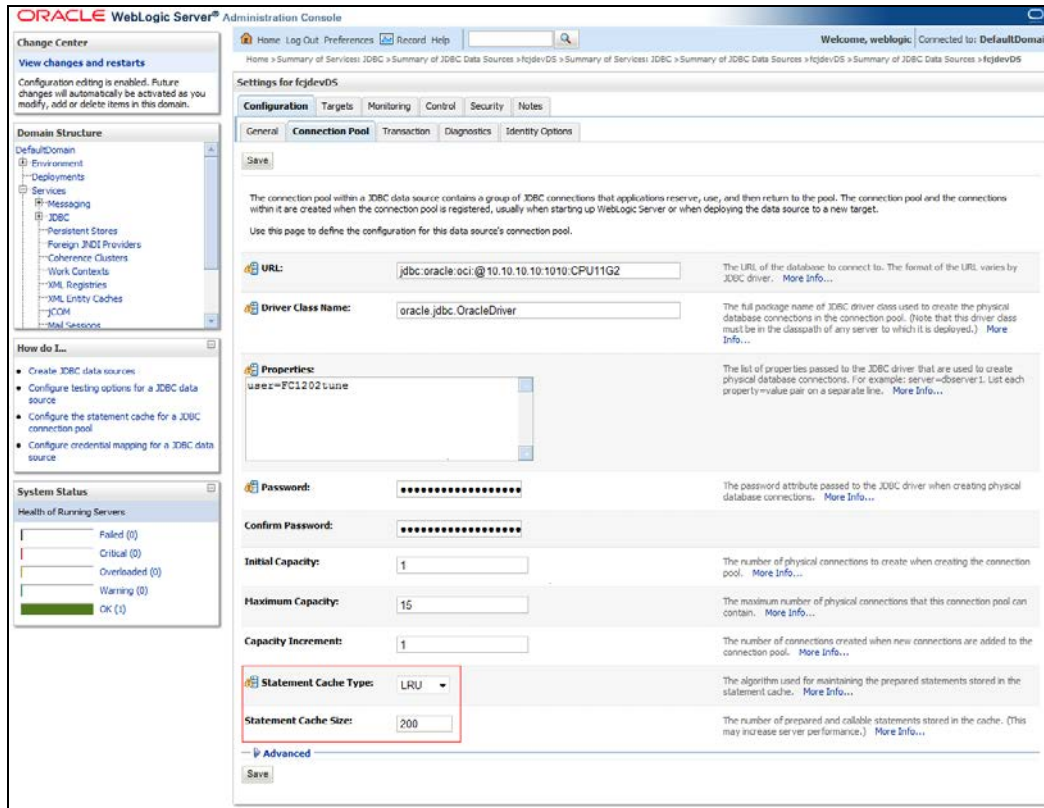
21. Click 'Activate Changes' button. Click 'Activate Changes' button on the left pane.

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



22. 'FCISDS' 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'.



Note the following

- You need to create another data source for Oracle FCIS with the JNDI name '<Non-XA FCIS HOST JNDI name>_ASYNC'. For example, if the Oracle FCIS HOST Non XA data source JNDI name is 'jdbc/fcjdevDS', then you need to create another data source for FCIS 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 FCIS BRANCH JNDI name>_ASYNC'.

7.2.1.4 Scheduler Data Source configuration

For all the LOB and SMS schema created for FCIS, Equivalent XA data sources are required for Scheduler with Jndi name as "jndi name of LOB/SMS schema"+ "_XA" (Standard naming convention)

Example

If there are three LOB schema's for FCIS with below jndi names

- ✓ jdbc/BR1204R1
- ✓ jdbc/EN1204R1
- ✓ jdbc/AMC1204R1

Equivalent Data source setup for scheduler will be

Data source name	Jndi Name
BR1204R1_XA	jdbc/BR1204R1_XA
EN1204R1_XA	jdbc/EN1204R1_XA
AMC1204R1_XA	jdbc/AMC1204R1_XA

7.2.2 JMS Server Creation

Follow the steps given below:

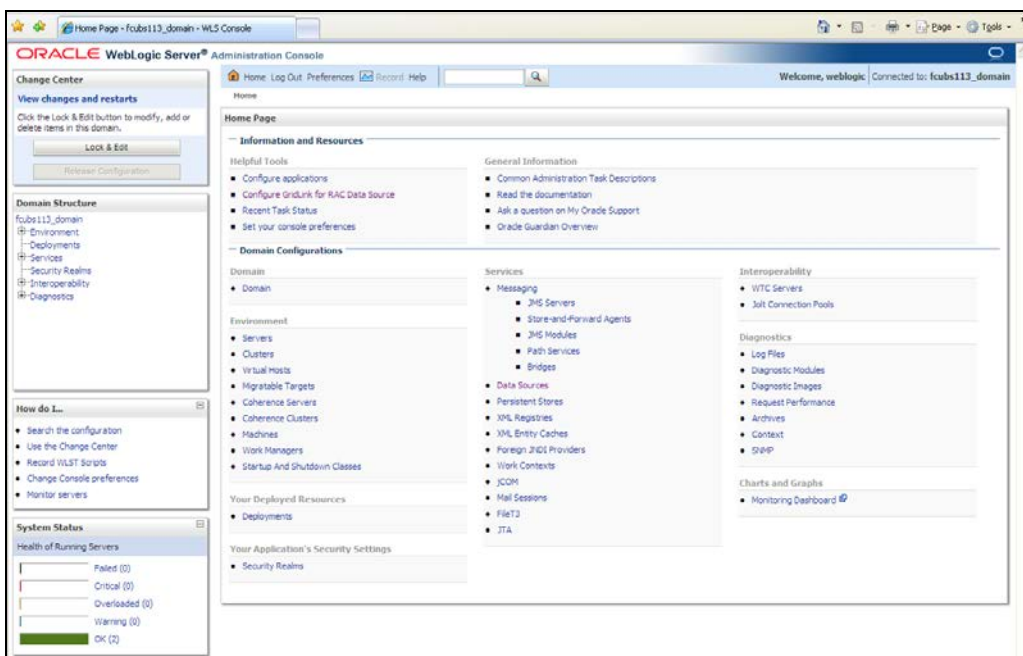
1. Start the Administrative Console of Weblogic application server. You can start this by entering Oracle Weblogic Admin Console URL in the address bar in an internet browser.

<http://10.10.10.10:1001/console> Eg: http://10.10.10.10:1001/console

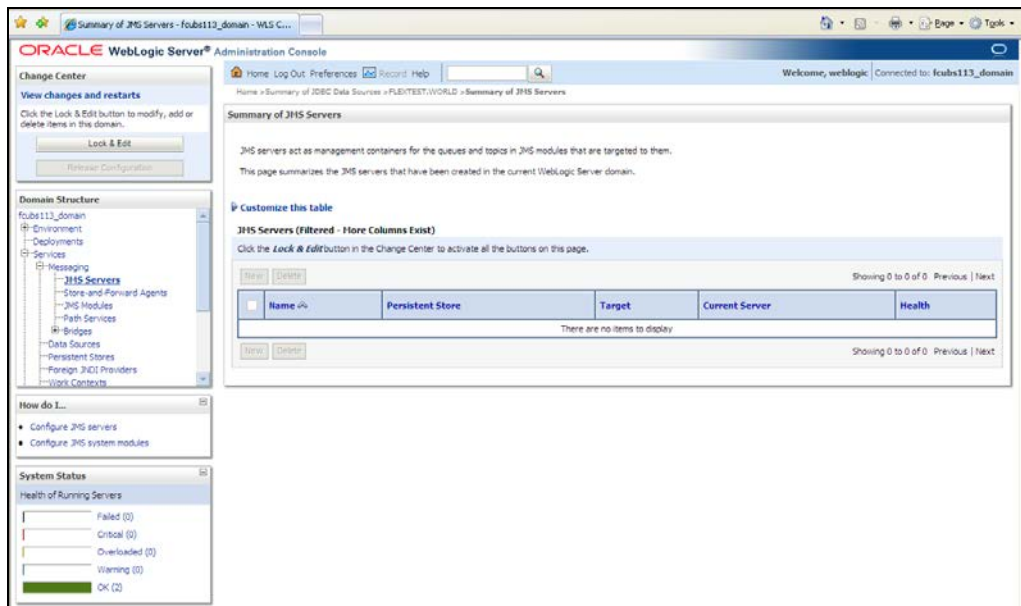
Following screen is displayed:



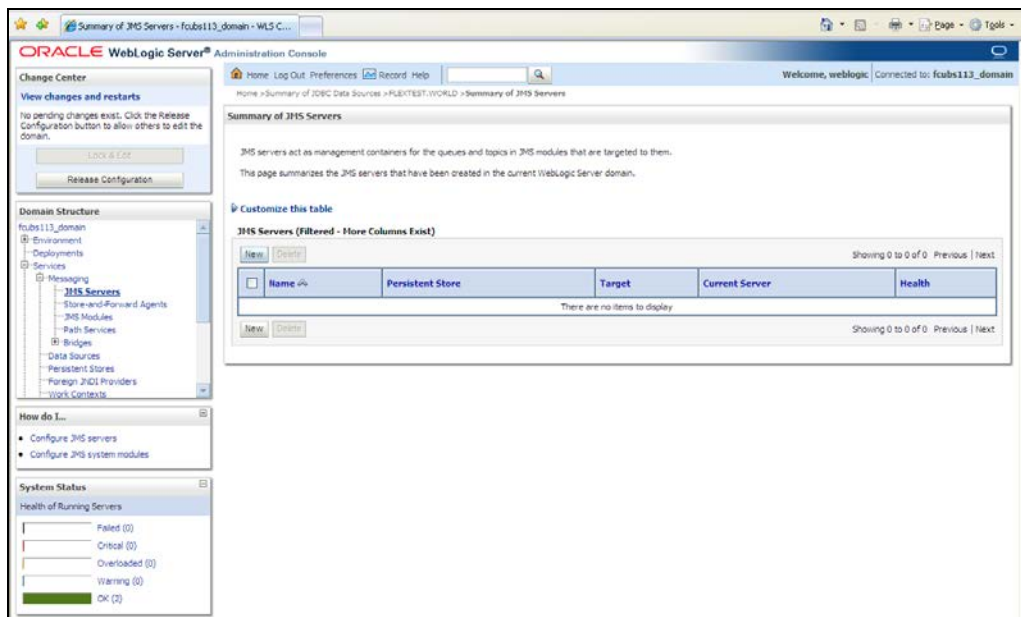
2. Specify the Weblogic administrator user name and password. Click 'Log In'.
3. Navigate to Oracle Weblogic home page.



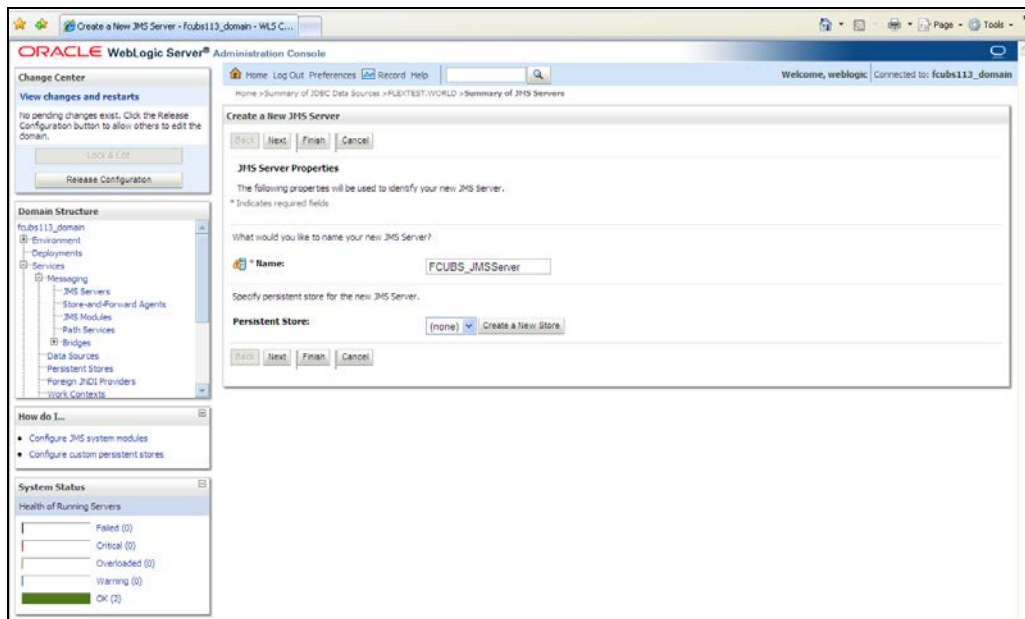
4. Following screen is displayed:



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



6. Click 'New'.

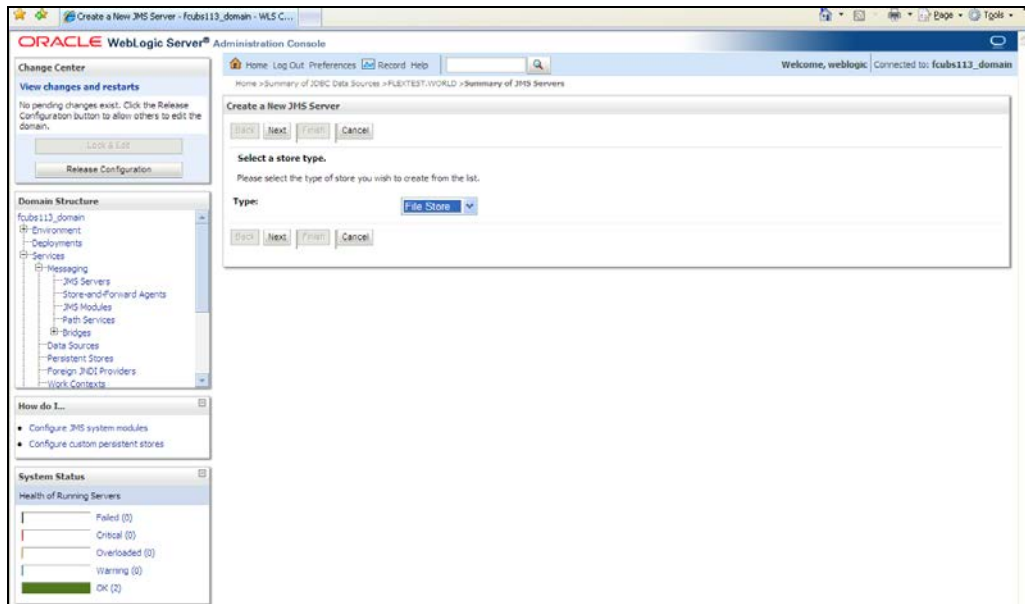


7. Specify the following details:

JMS Server Name

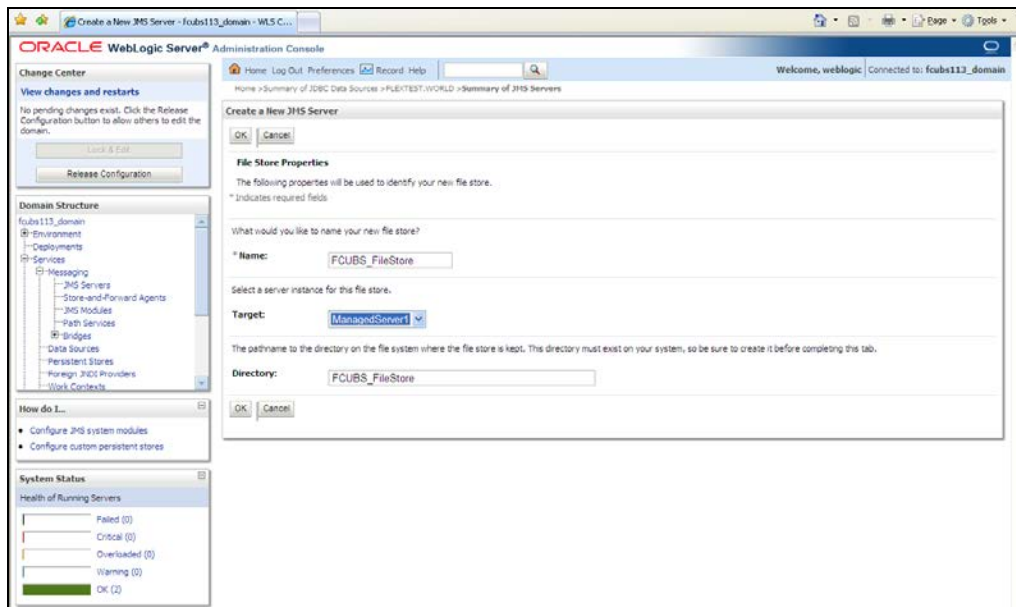
FCIS_JMSServer

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



9. Select 'File Store' as the type and click 'Next'.

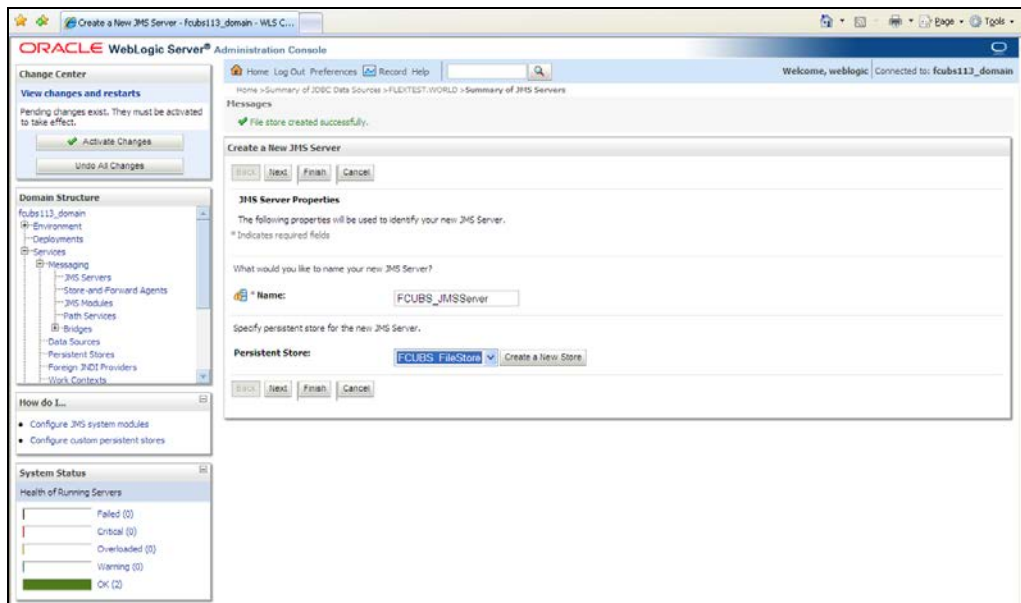
Following screen is displayed:



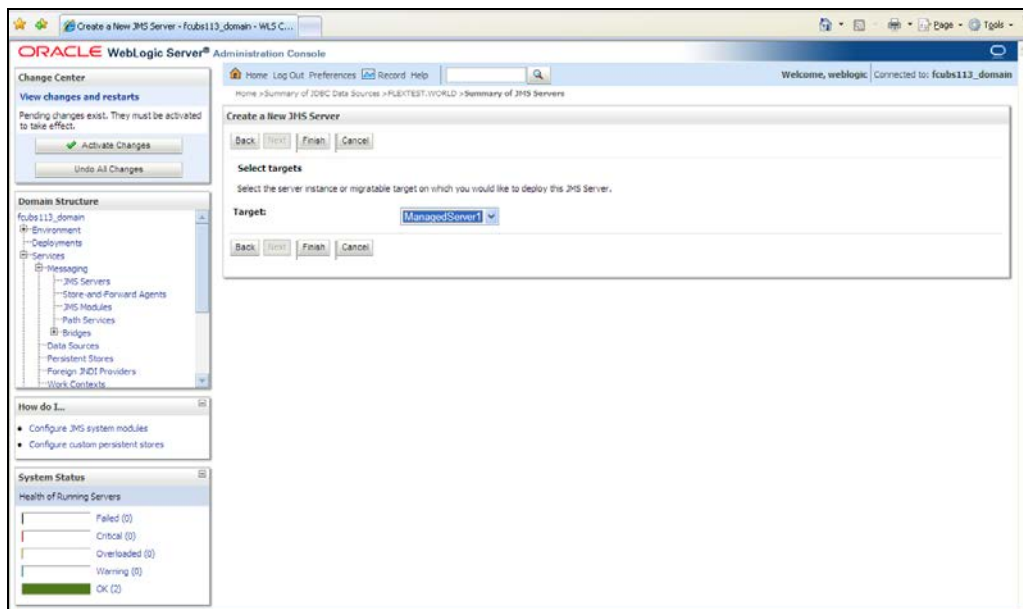
10. To identify the new File Store, specify the following properties:

- Specify the file store name as FCIS_FileStore.
- Select a server. For this file store, you may select ManagedServer1 (created by the user).
- Specify the Directory as FCIS_FileStore.
- Click 'OK'.

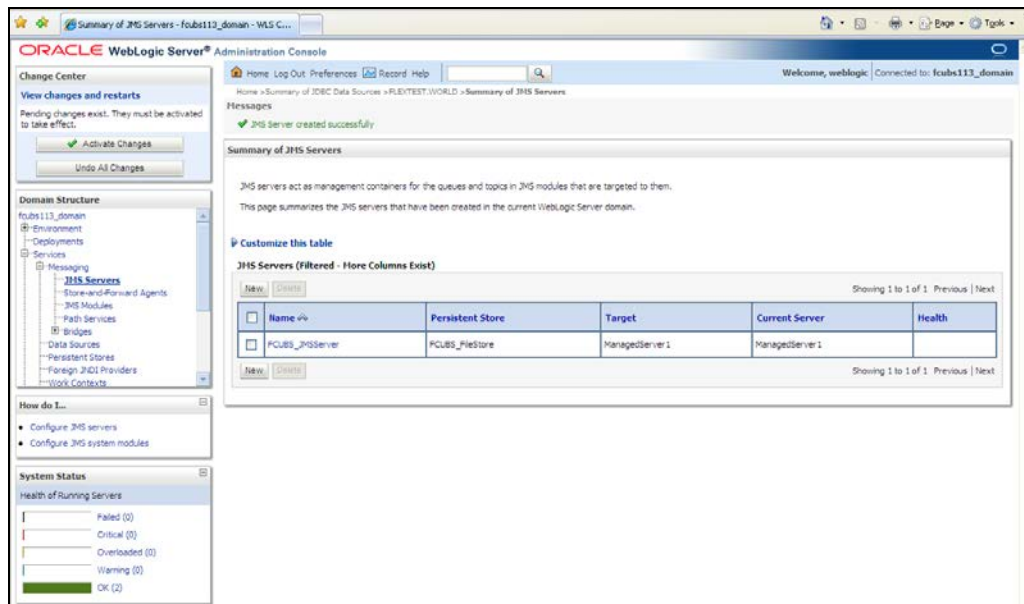
The following screen is displayed with message 'File store created successfully'.



11. Click 'Next'.



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



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

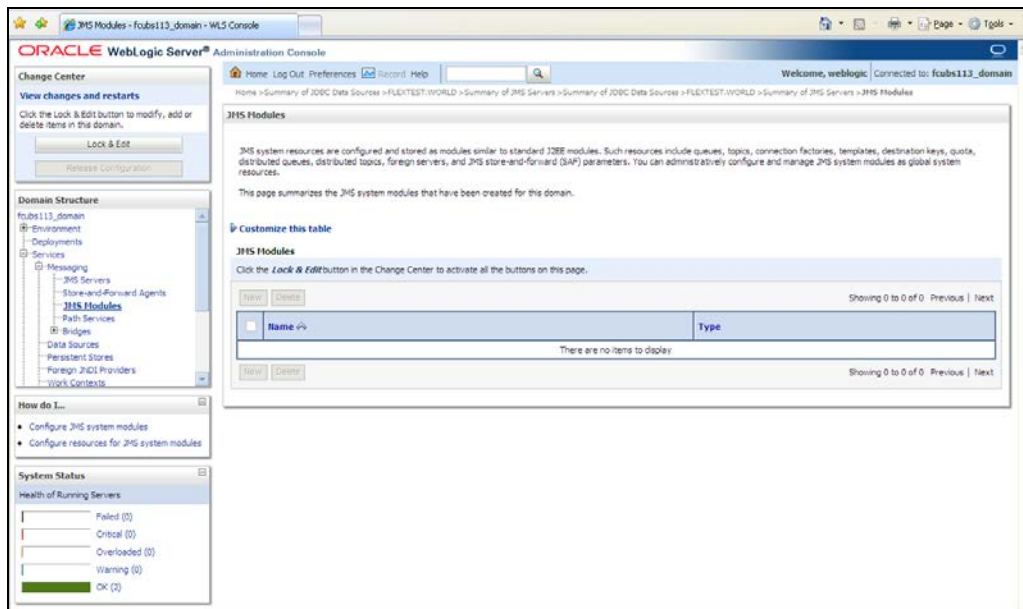
14. Click 'Activate Changes' under Change Center. The message 'All changes have been activated. No restarts are necessary' is displayed.

7.2.3 JMS Modules Creation

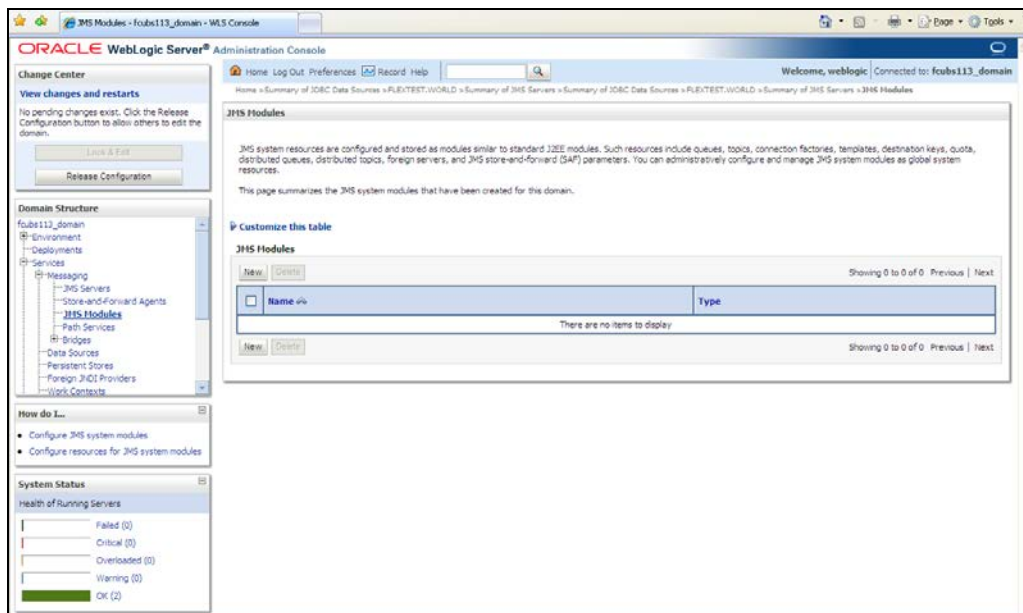
Follow the steps given below:

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.

The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create JMS System Module'. It contains a 'Name' field with the value 'FCUBS_SystemModule' and a 'Descriptor File Name' field with the value 'FCUBS_SystemModule'. The 'Location In Domain' field is empty. The left sidebar shows the domain structure and system status.

Enter the System Module Name as FCIS_SystemModule.

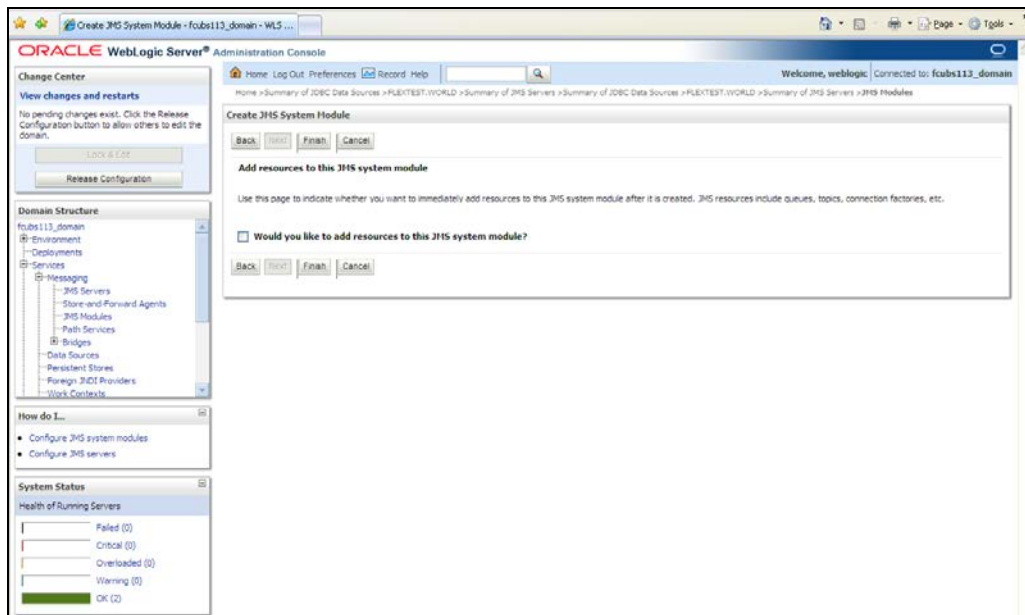
Enter the Description File Name as FCIS_SystemModule.

4. Click 'Next'.

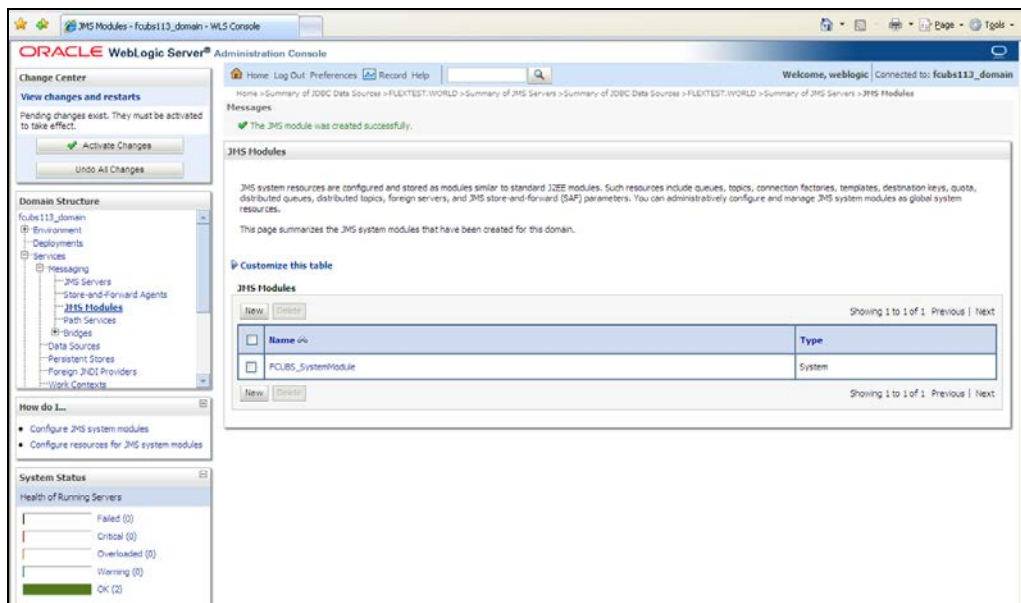
The following screen is displayed.

The screenshot shows the Oracle WebLogic Server Administration Console. The main window is titled 'Create JMS System Module'. It contains a 'Targets' section with a table showing 'AdminServer' and 'ManagedServer1'. The 'ManagedServer1' checkbox is checked. The left sidebar shows the domain structure and system status.

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

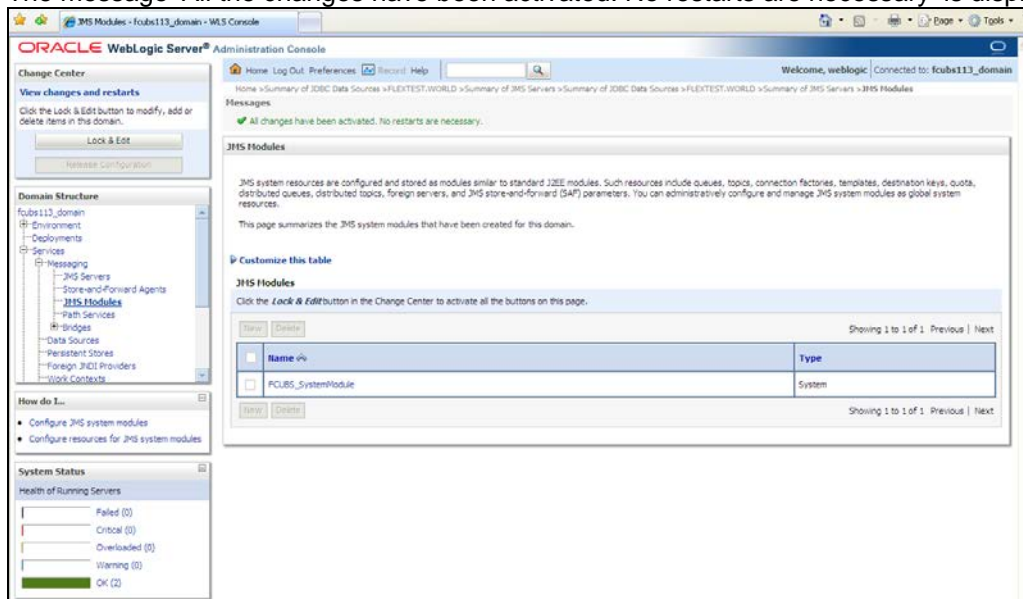


6. Click 'Finish' button. The following screen is displayed.



7. Click 'Activate Changes' button on the left pane.

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

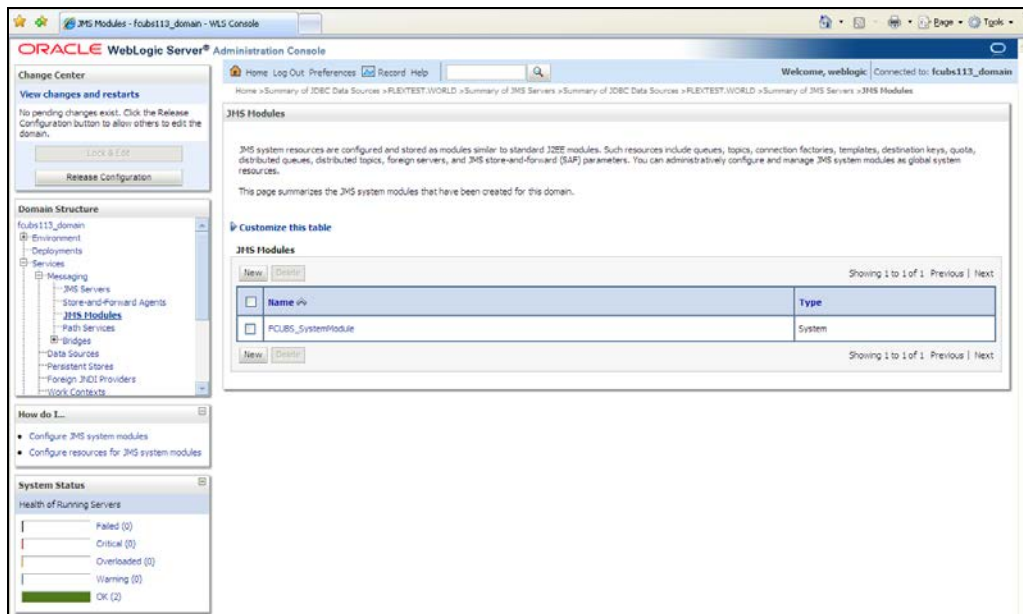


7.2.4 Sub Deployment Creation

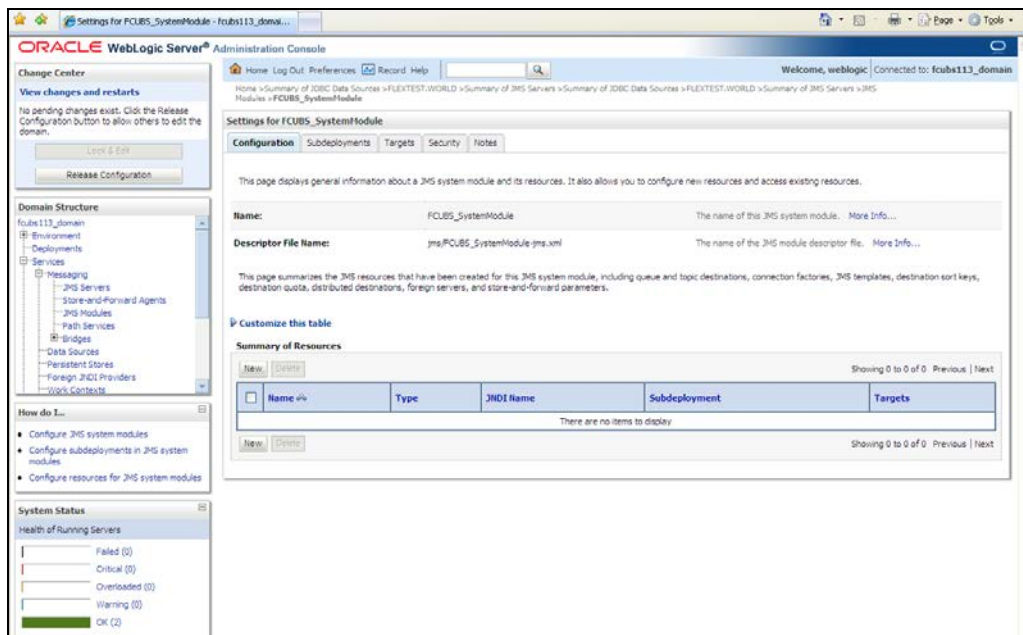
Follow the steps given below:

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

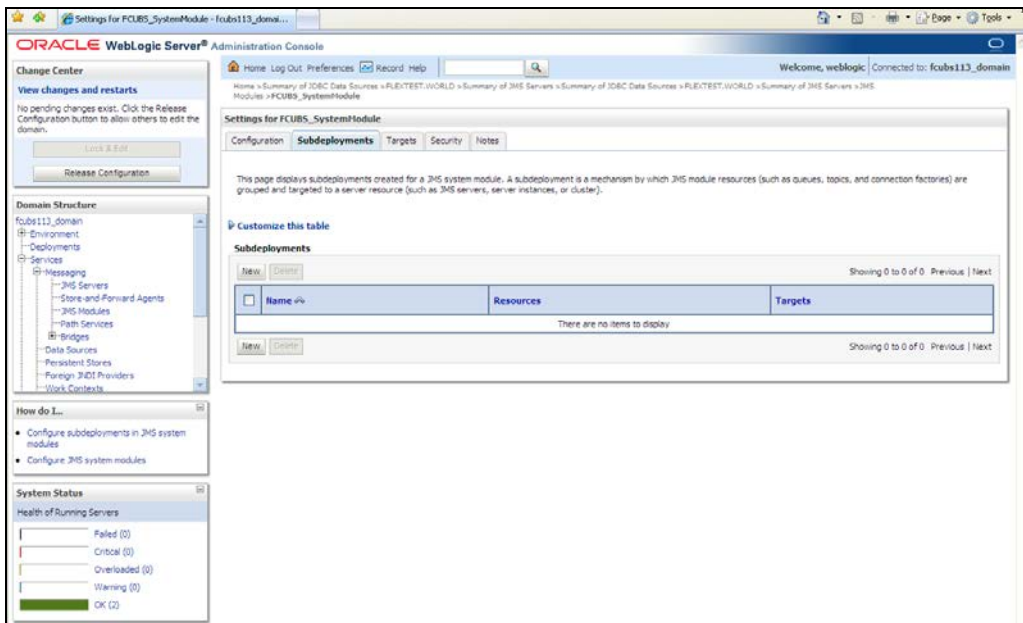
The following screen is displayed:



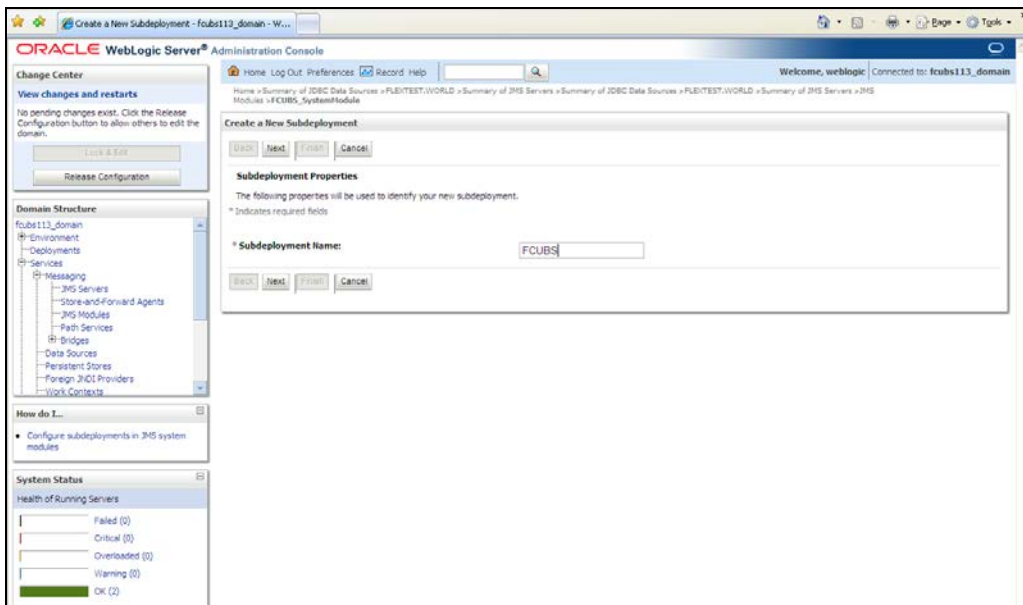
2. Click 'Lock & Edit' button.
3. Select the JMS module created earlier.



4. Click 'Subdeployments' tab.

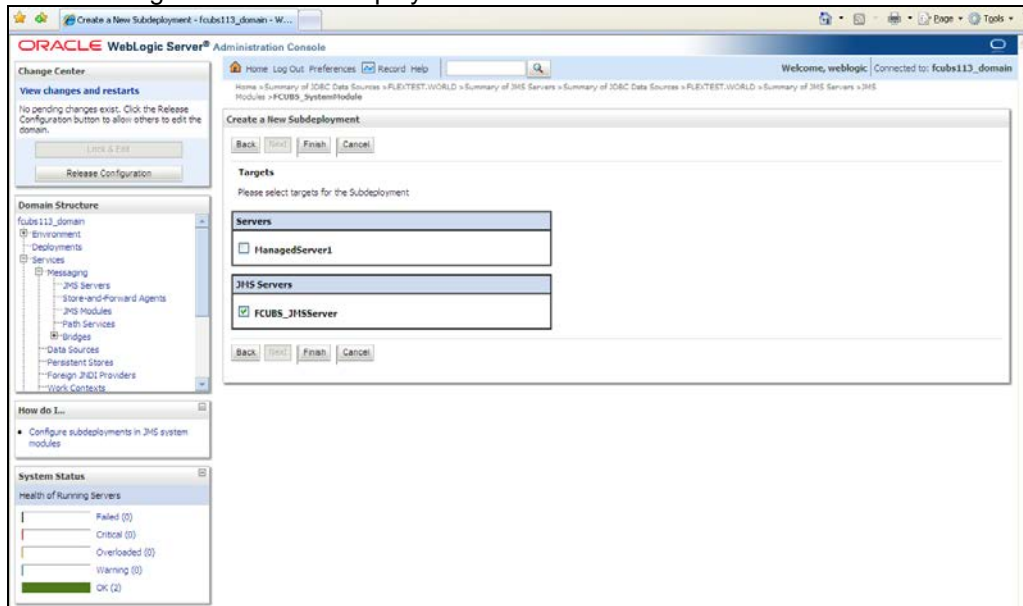


5. Click 'New'. The following screen is displayed.

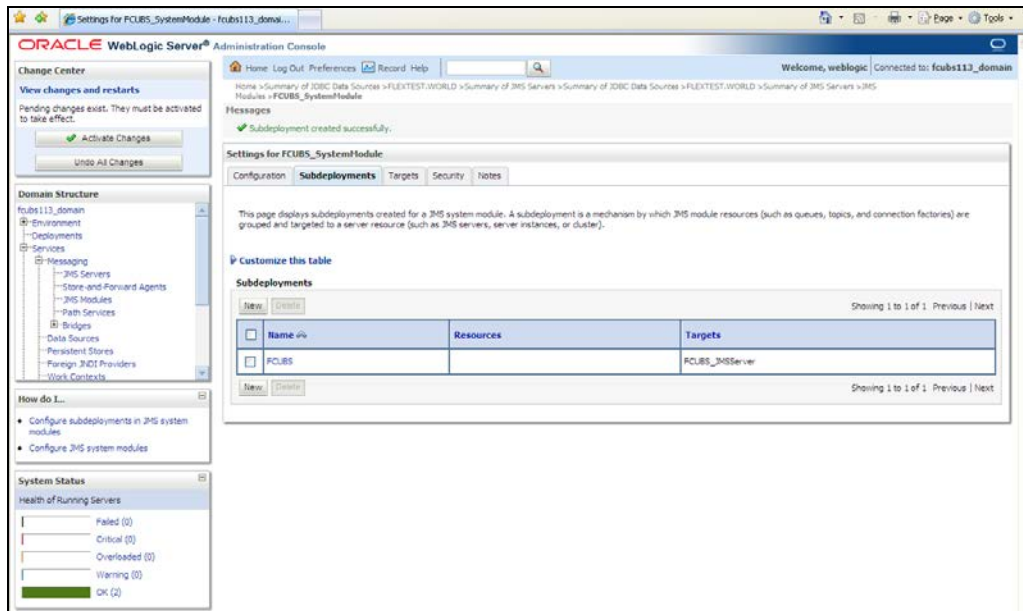


- Specify the Subdeployment Name as 'FCIS'. Then click 'Next'.

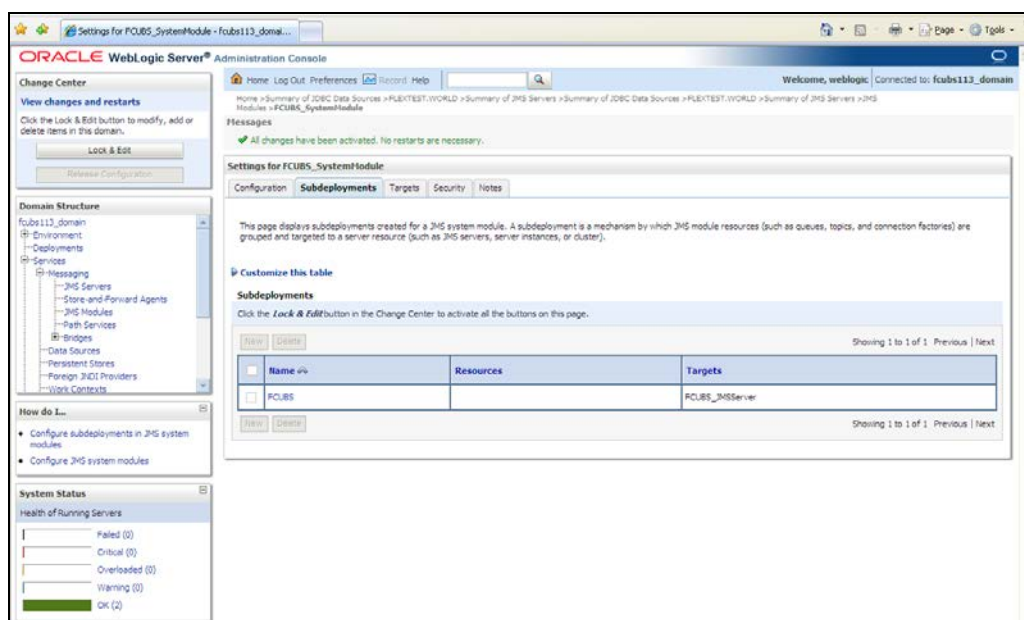
The following screen will be displayed.



- Select the JMS Server (as created by the user).
- Click 'Finish' button.
- Following screen is displayed.

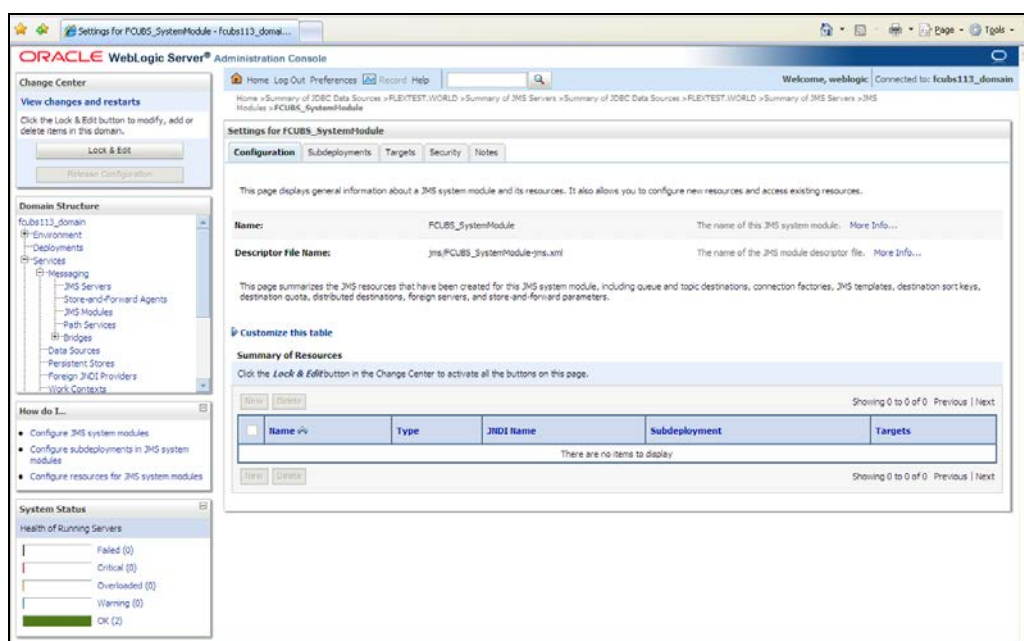


10. Click 'Activate Changes'. Following screen is displayed.



7.2.5 JMS Queue Creation

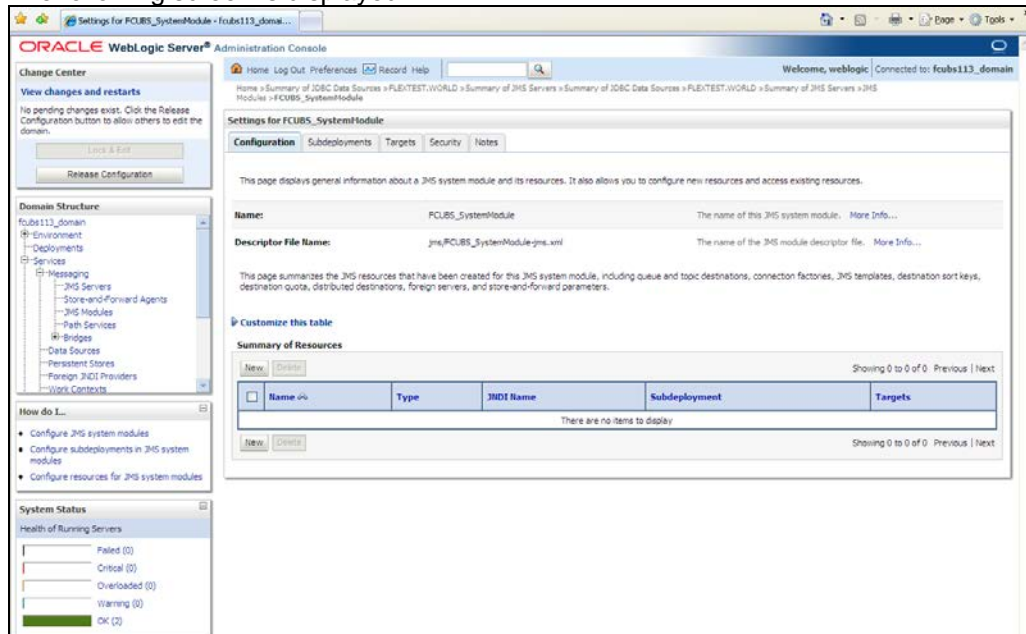
1. Select the JMS Module created earlier.



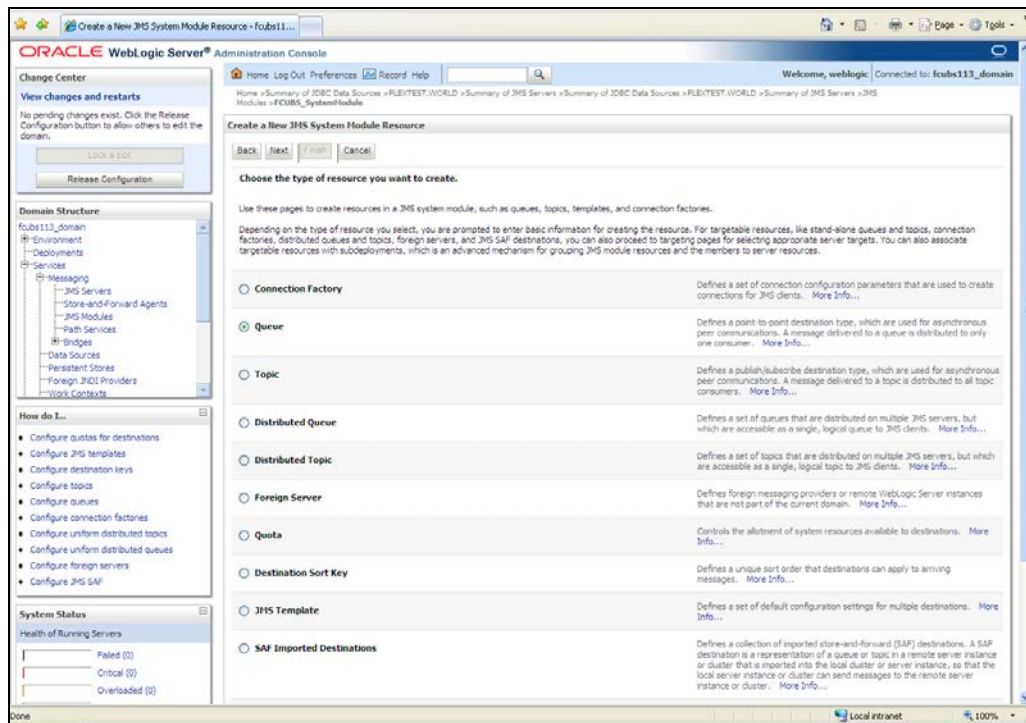
2. You need to set the configuration for FCIS_SystemModule is to be set.

3. Click 'Configuration'. Then click 'Lock & Edit'.

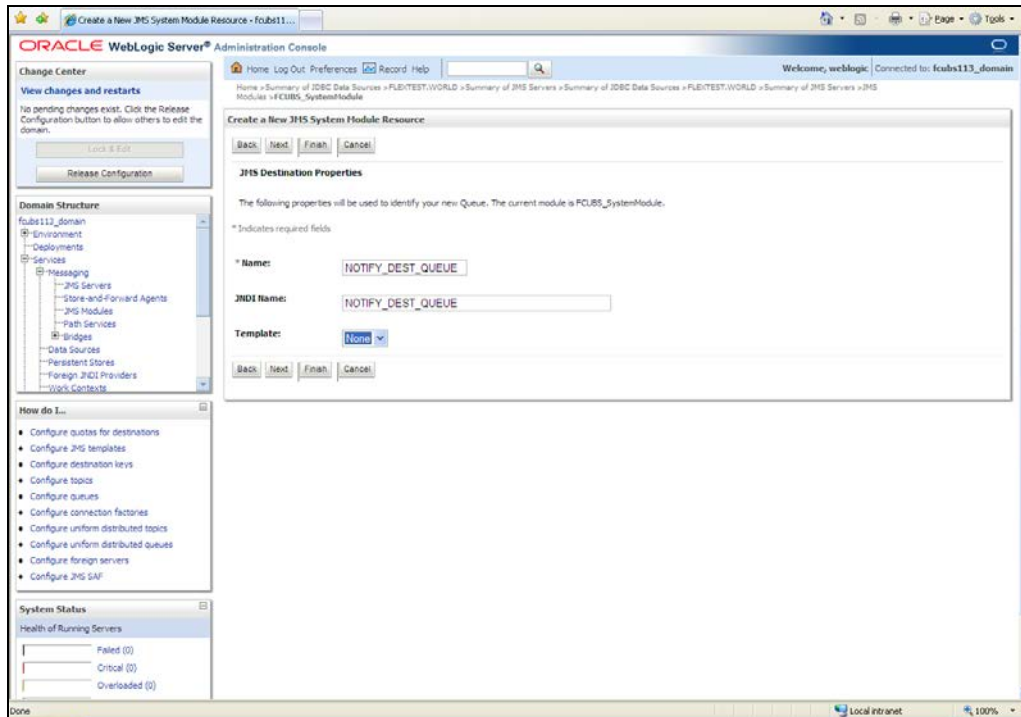
The following screen is displayed.



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



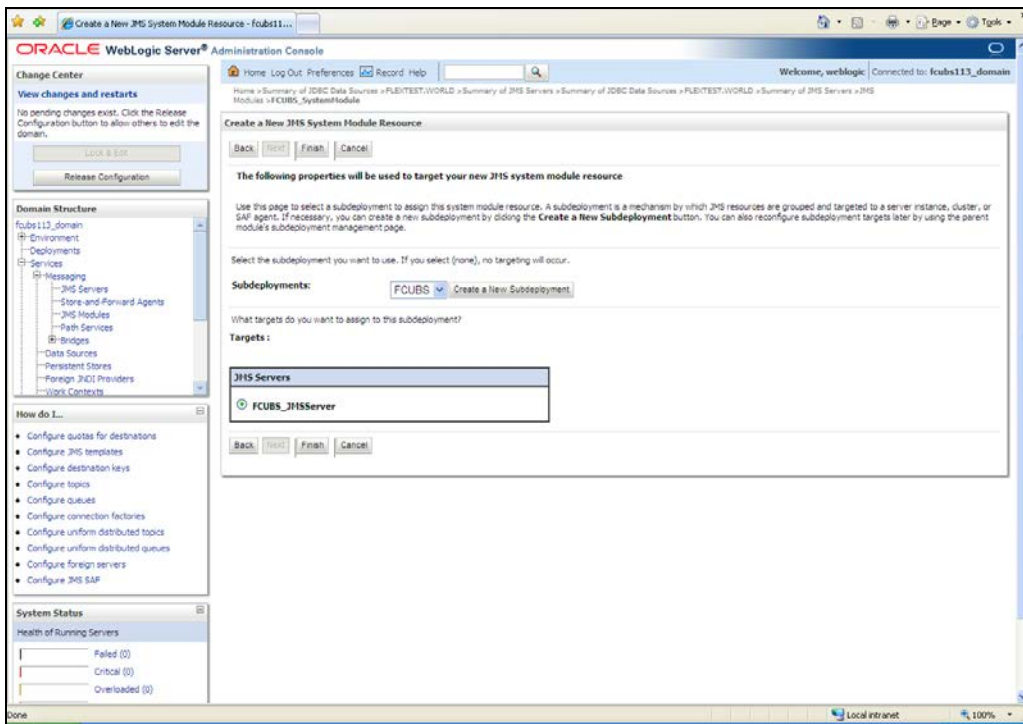
5. Select the 'Queue' option. Then click 'Next'.



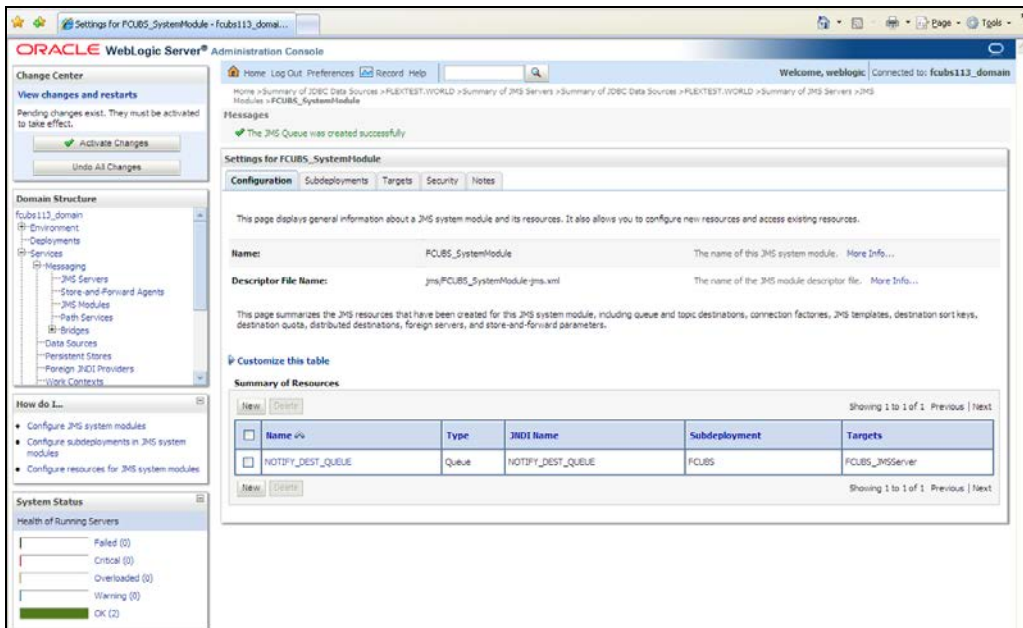
For creating new JMS System Module Resources, follow the steps given below:

- 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.



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



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

The screenshot shows the Oracle WebLogic Server Administration Console. The left sidebar contains the 'Change Center' and 'Domain Structure' panels. The main area displays the 'Settings for FCUBS_SystemModule' configuration page. The 'Configuration' tab is active, showing details for the JMS system module. Below the configuration details, there is a 'Summary of Resources' section with a table listing the resources.

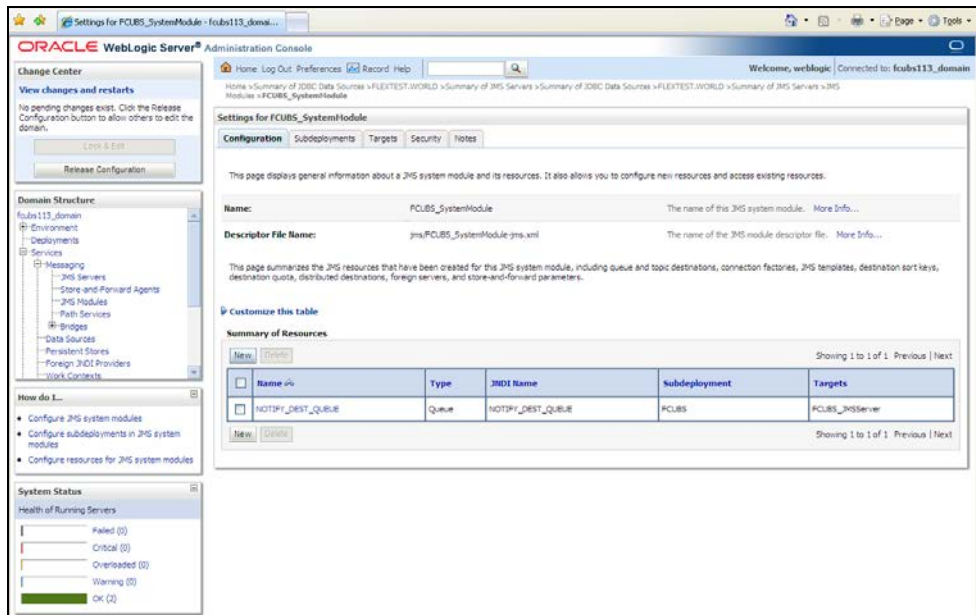
Name	Type	JNDI Name	Subdeployment	Targets
NOTIFY_DEST_QUEUE	Queue	NOTIFY_DEST_QUEUE	FCUBS	FCUBS_JMSServer

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

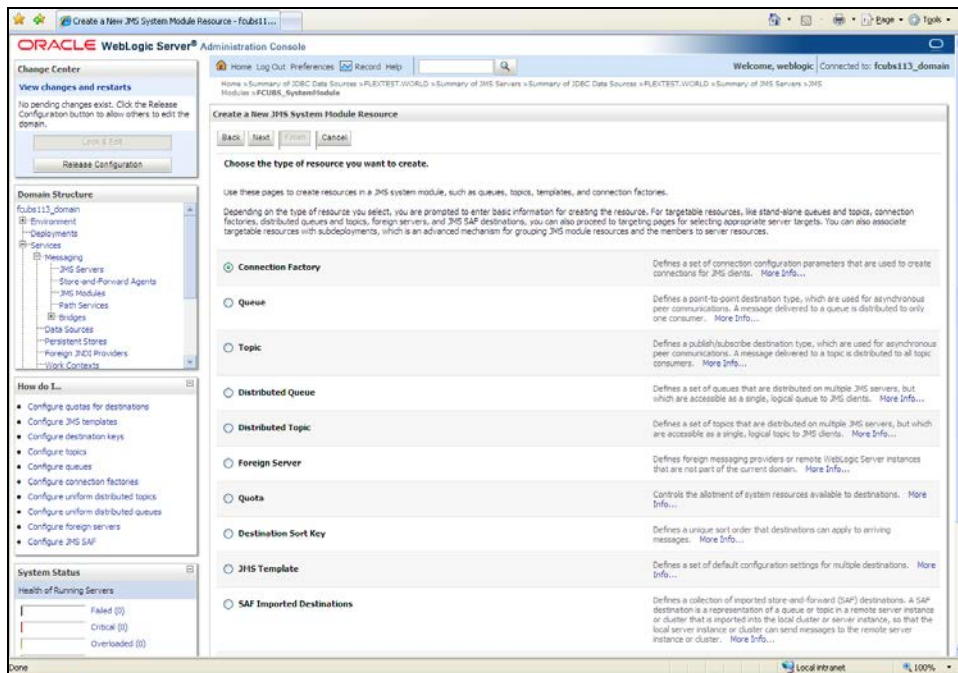
7.2.6 JMS Connection Factory Creation

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

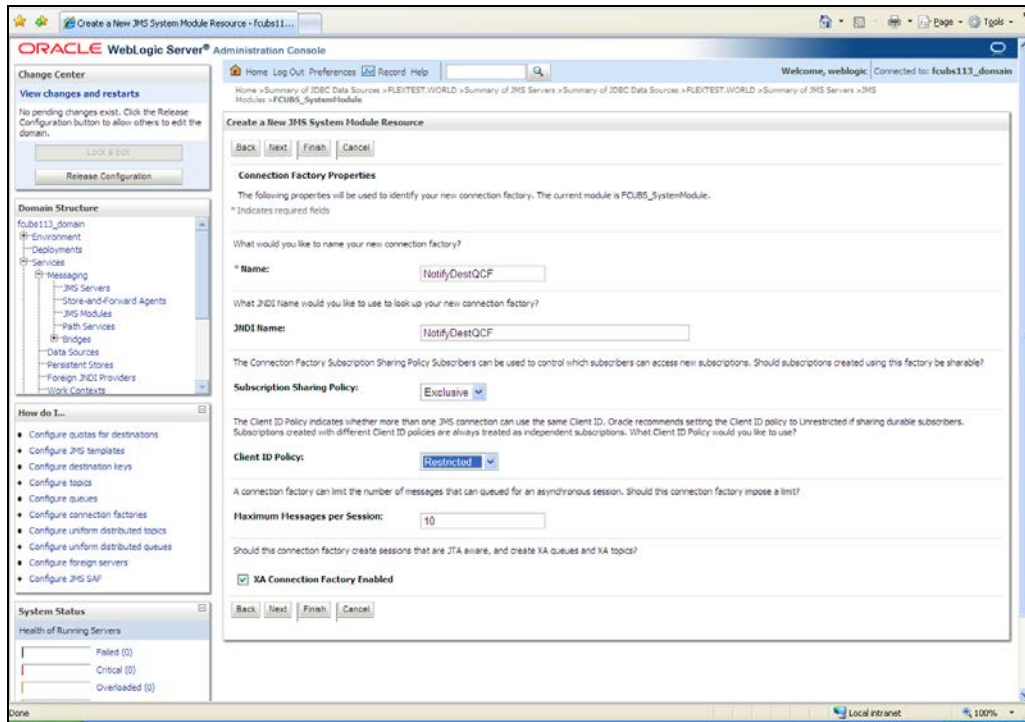
1. Click 'New'.



The following screen is displayed:

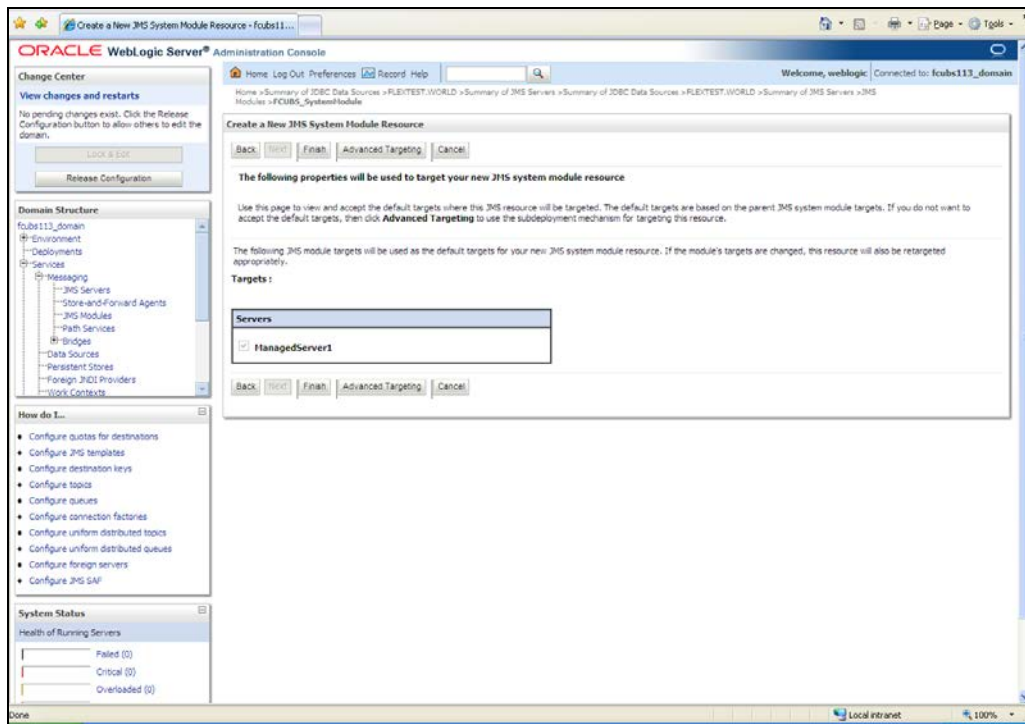


2. Select 'Connection Factory'. Click 'New'.

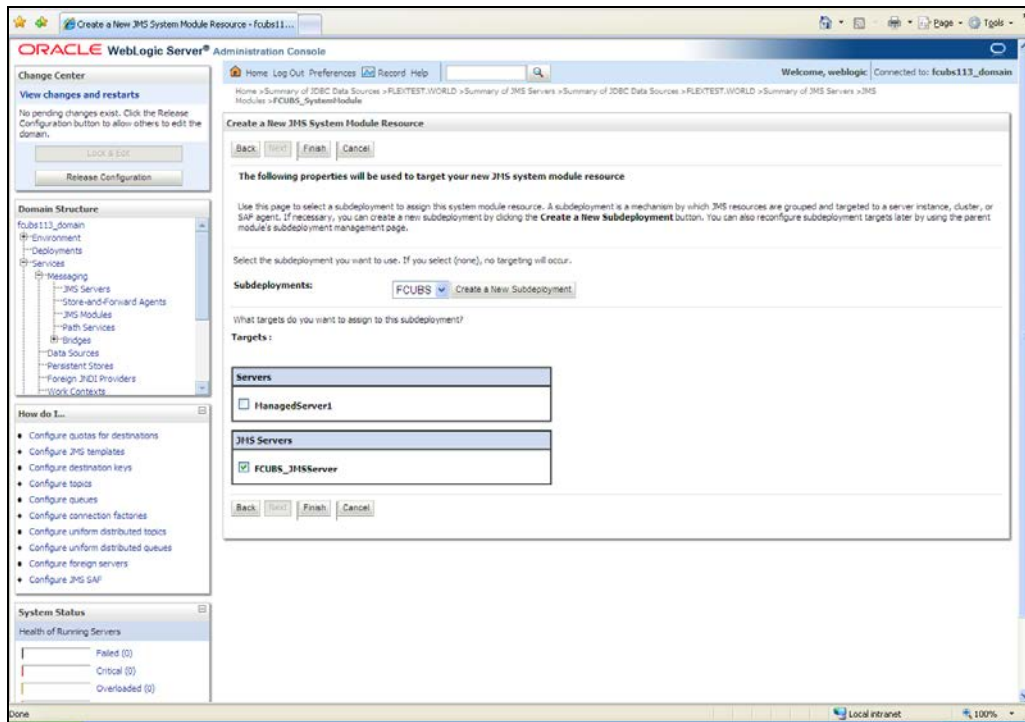


3. Enter the Name of the Connection Factory as 'NotifyDestQCF'.
4. Enter the JNDI Name as 'NotifyDestQCF'.
5. Check the box 'XA Connection Factory Enabled'.
6. Click 'Next'.

The following screen is displayed:



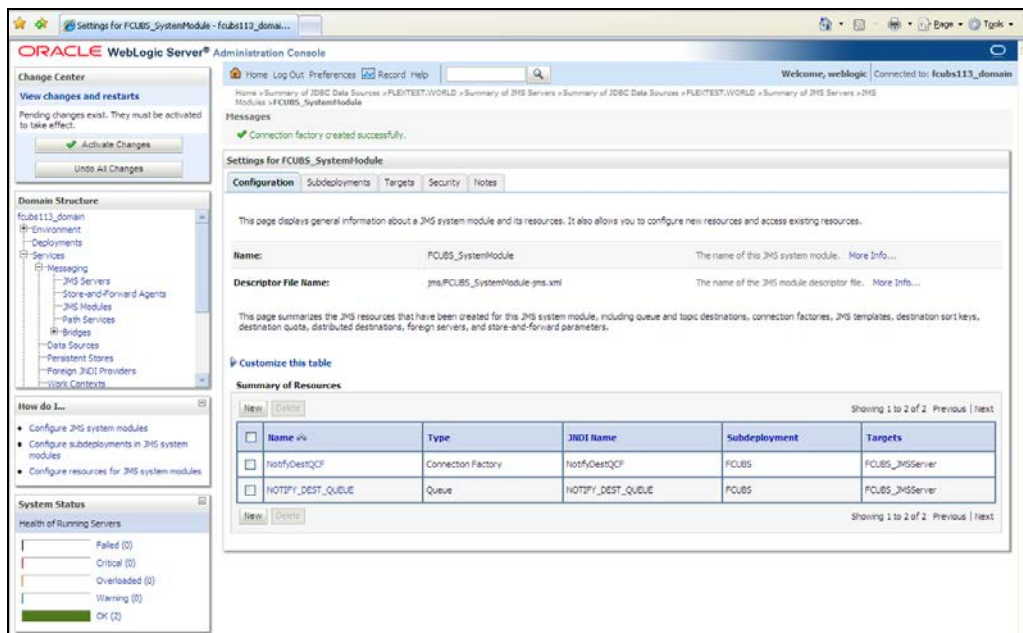
7. Click 'Advanced Targeting'. The following screen is displayed.



8. Select the 'Subdeployments' as FCIS.

9. Under JMS Servers, check the box against 'Managed Server'.

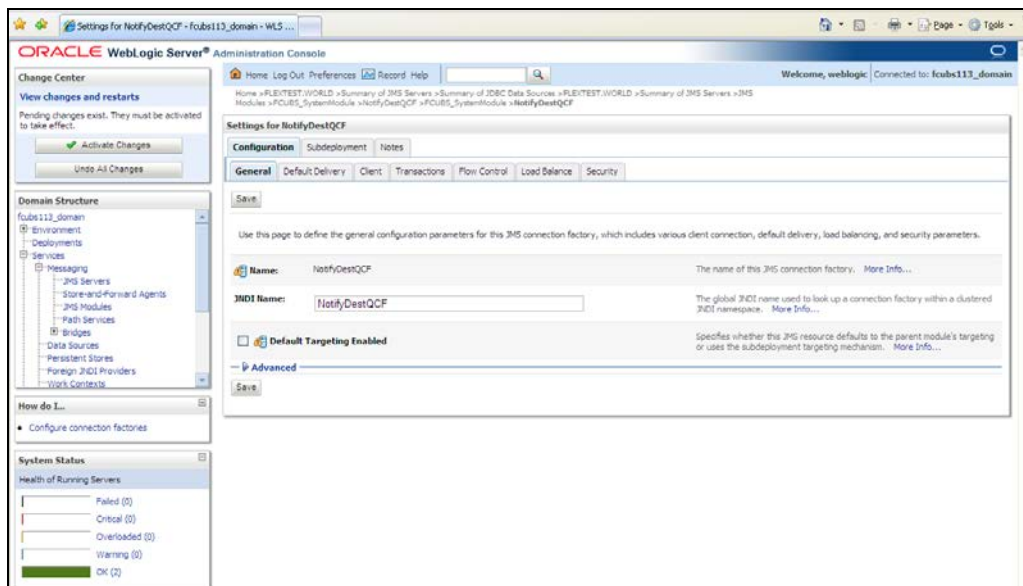
10. Click 'Finish'. The following screen is displayed:



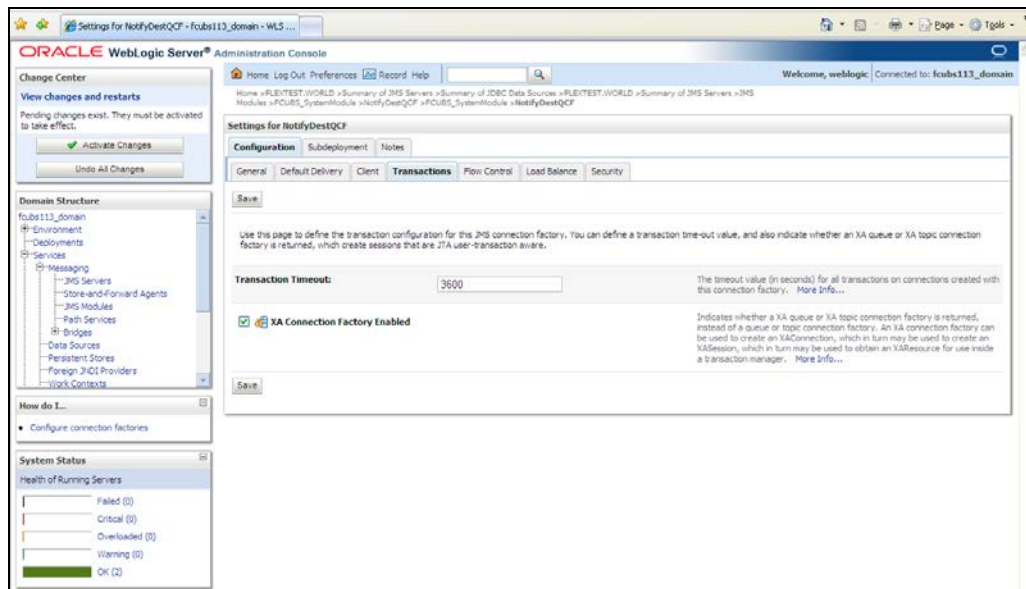
11. The message 'Connection Factory created successfully' is displayed.

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

The following screen will be displayed.

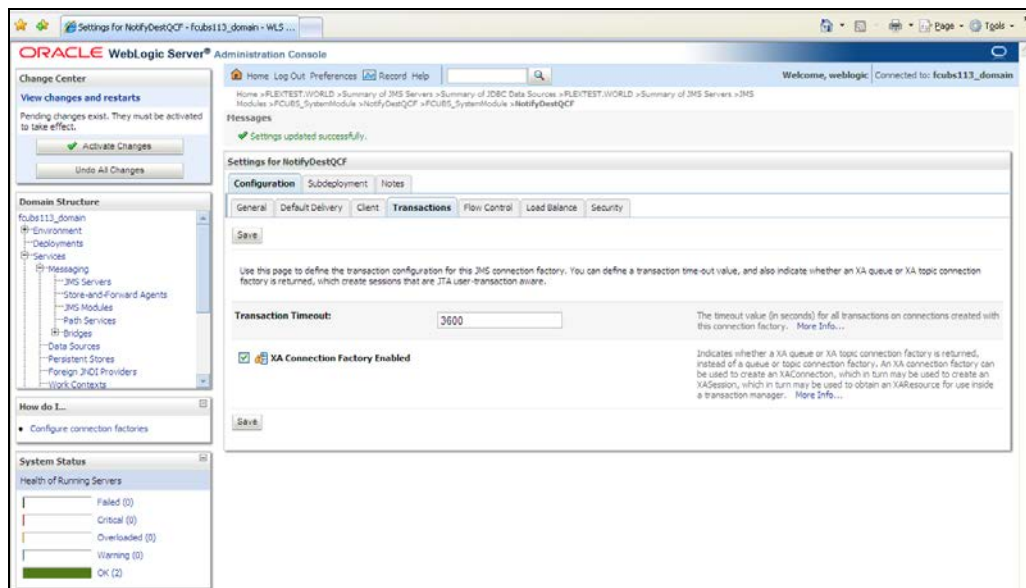


13. Click 'Transactions' Tab. The following screen is displayed.



14. Check the box 'XA Connection Factory Enabled'.

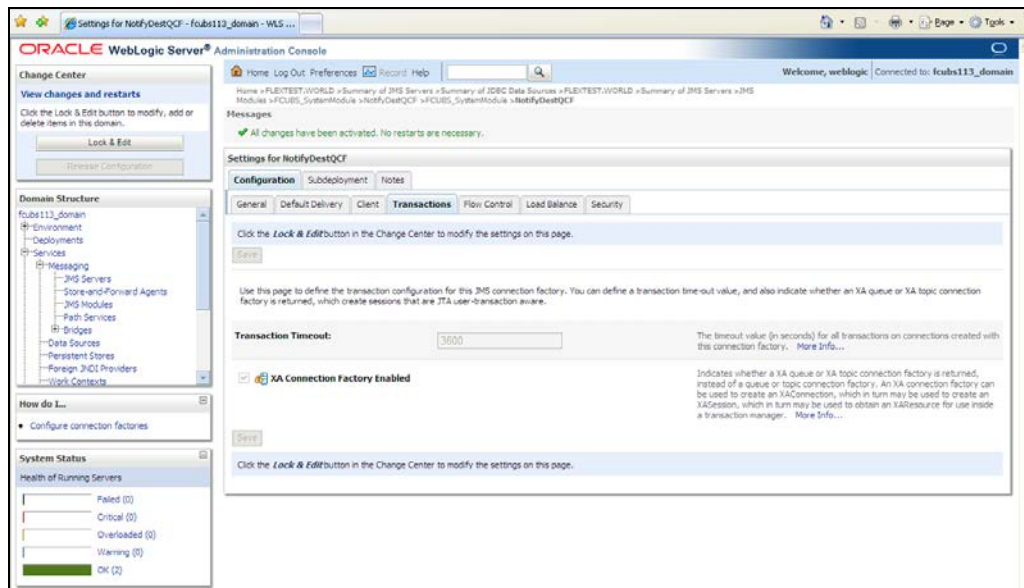
15. Click 'Save'. The following screen is displayed.



16. The message 'Settings updated successfully' is displayed.

17. Click 'Activate Changes' button under 'Change Center'.

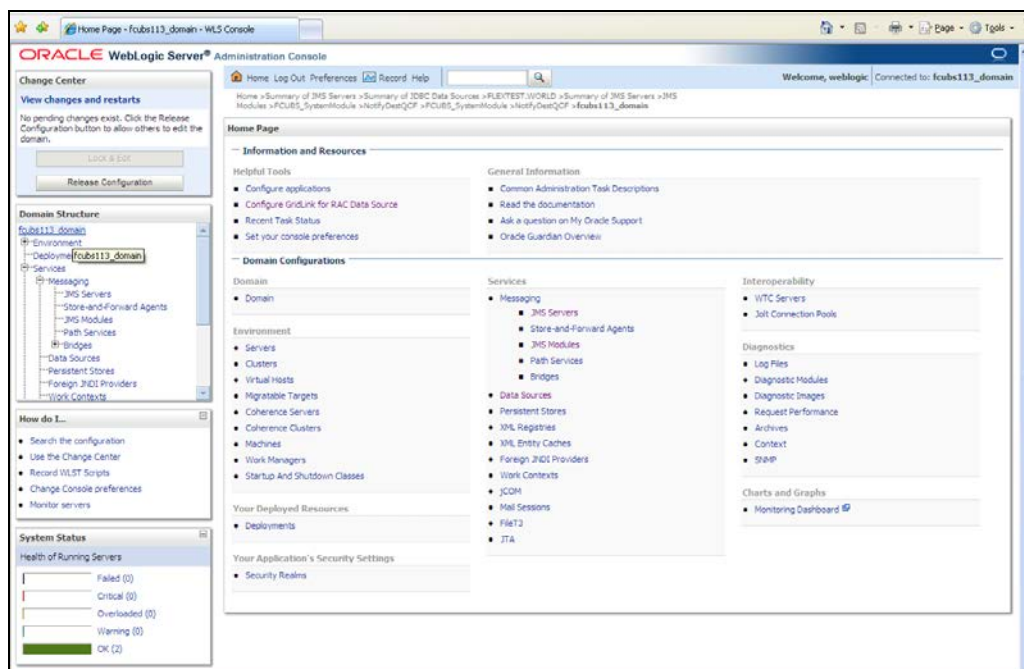
The message 'All the changes have been activated. No restarts are necessary' is displayed.



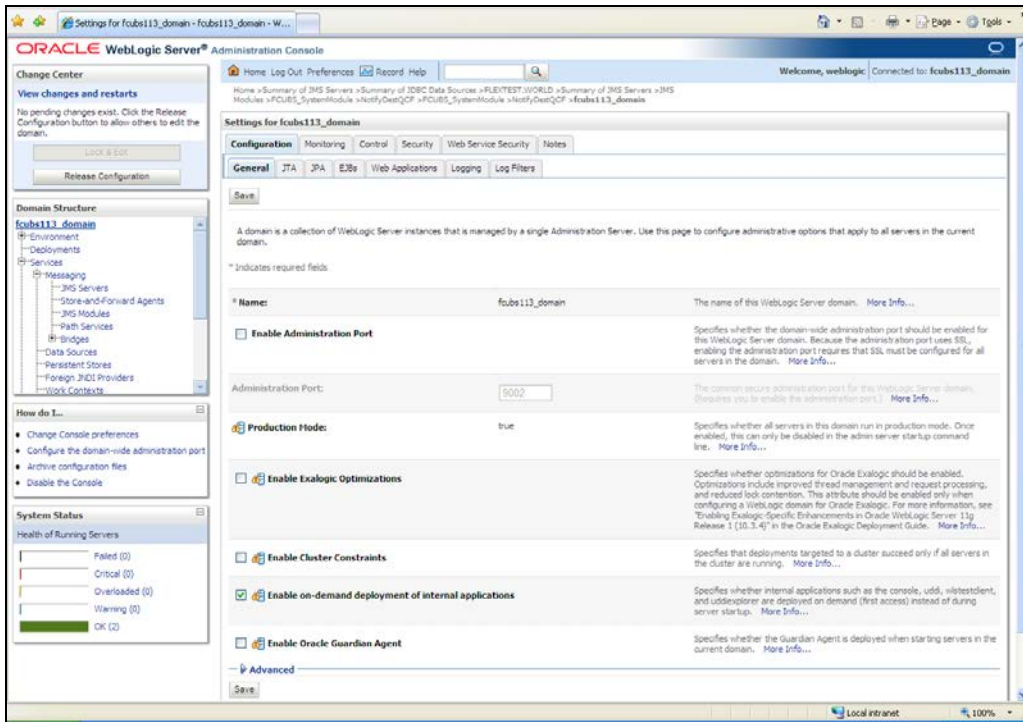
7.3 Configuring Weblogic for Oracle FLEXCUBE

This section explains the steps for configuring Oracle WebLogic application server for Oracle FLEXCUBE Investor Servicing. Follow the steps given below:

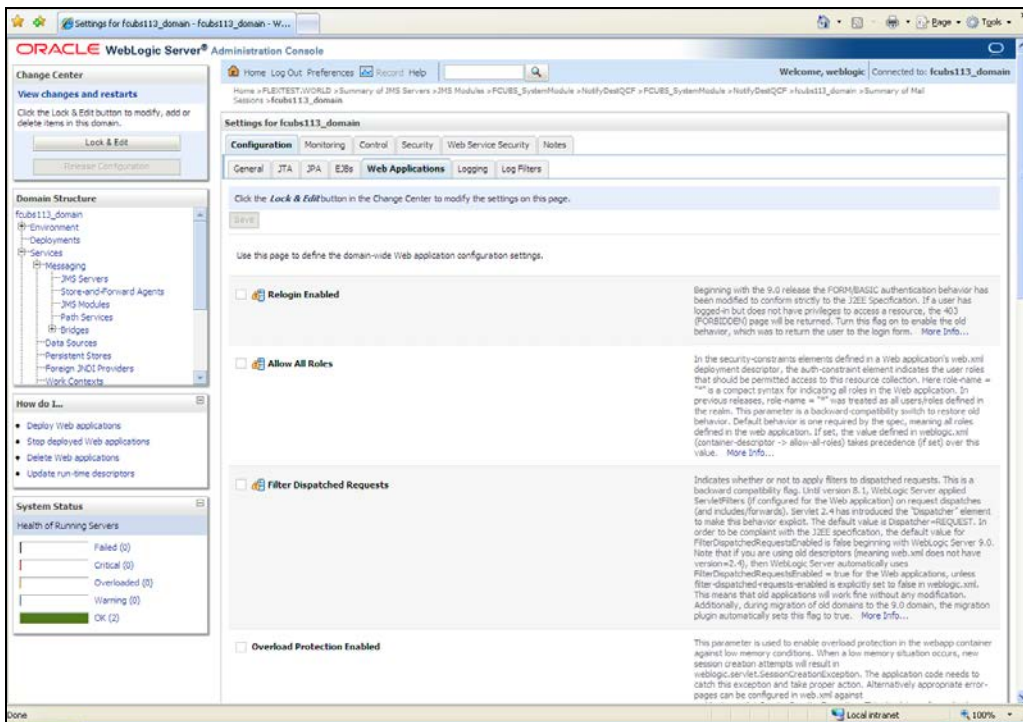
1. Select the domain from the domain structure as shown below. (Eg: fcis_domain).



The following screen is displayed:



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



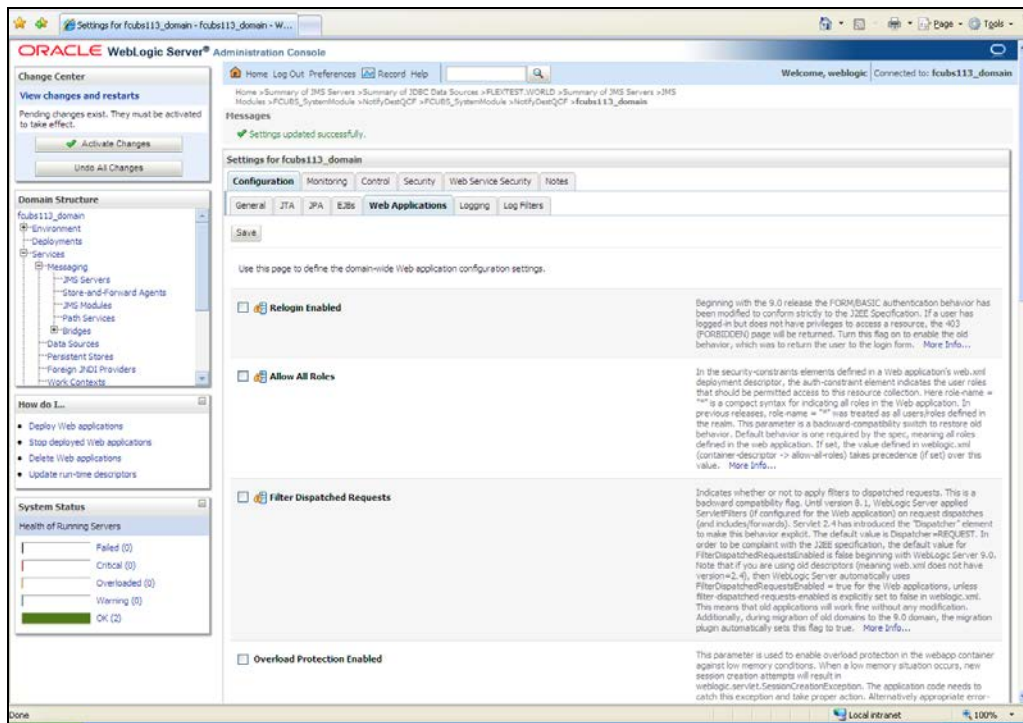
3. Scroll down and ensure that the details are as shown in the figure. The remaining portion of the screen is given below:

Settings for fouls112_domain - fouls112_domain - W...

<input type="checkbox"/> HTTP Trace Support Enabled	Returns the value of <code>HttpTraceSupportEnabled</code> . More Info...
<input type="checkbox"/> WebLogic Plugin Enabled	Specifies whether or not the proprietary WL Proxy-Client-IP header should be honored. (This is needed only when WebLogic plugins are configured.) More Info...
<input checked="" type="checkbox"/> Auth Cookie Enabled	Whether authcookie feature is enabled or not. More Info...
<input checked="" type="checkbox"/> Change Session ID On Authentication	Global property to determine if we need to generate a new SessionID after authentication. When this property set to "false", the previous sessionID will be retained even after authentication. More Info...
<input type="checkbox"/> WAP Enabled	Indicates whether the session ID should include J2M information. (Checking this box may be necessary when using URL rewriting with WAP devices that limit the size of the URL to 128 characters, and may also affect the use of replicated sessions in a cluster.) When this box is selected, the default size of the URL will be set at 52 characters, and it will not contain any special characters. More Info...
Post Timeout: <input type="text" value="30"/>	The amount of time this server waits between receiving chunks of data in an HTTP POST data before it times out. (This is used to prevent denial-of-service attacks that attempt to overload the server with POST data.) More Info...
Maximum Post Time: <input type="text" value="-1"/>	Max Post Time (in seconds) for reading HTTP POST data in a servlet request. MaxPostTime < 0 means unlimited. More Info...
Maximum Post Size: <input type="text" value="-1"/>	The maximum post size this server allows for reading HTTP POST data in a servlet request. A value less than 0 indicates an unlimited size. More Info...
<input checked="" type="checkbox"/> Work Context Propagation Enabled	Indicates whether or not <code>WorkContextPropagation</code> is enabled. By default it is turned on. There is a little overhead involved in propagating <code>WorkContexts</code> . Therefore, if you don't care about <code>WorkContext</code> propagation, turn this value off in production environments. More Info...
P3P Header Value: <input type="text"/>	Returns the P3P Header value that will be sent with all responses for http requests (if non null). The value of this header points to the location of the policy reference file for the Web site. More Info...
<input checked="" type="checkbox"/> JSP Compiler Backwards Compatible	Global property to determine the behavior of the JSP compiler. When this property set to "true", the JSP compiler throws a translation error for JSPs that do not conform to the JSP 2.0 specification. This property exists for backward compatibility. More Info...
<input checked="" type="checkbox"/> Archived Real Path Enabled	Global property to determine the behavior of <code>getRealPath()</code> for archived web applications. When this property set to "true", <code>getRealPath()</code> will return the canonical path of the resource files. More Info...

4. Check the options 'JSP Compiler Backwards Compatible' and 'Archived Real Path Enabled'.
5. Click 'Save'.

6. The following screen is displayed:



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

8. Click the button 'Active Changes' and then restart the admin server.

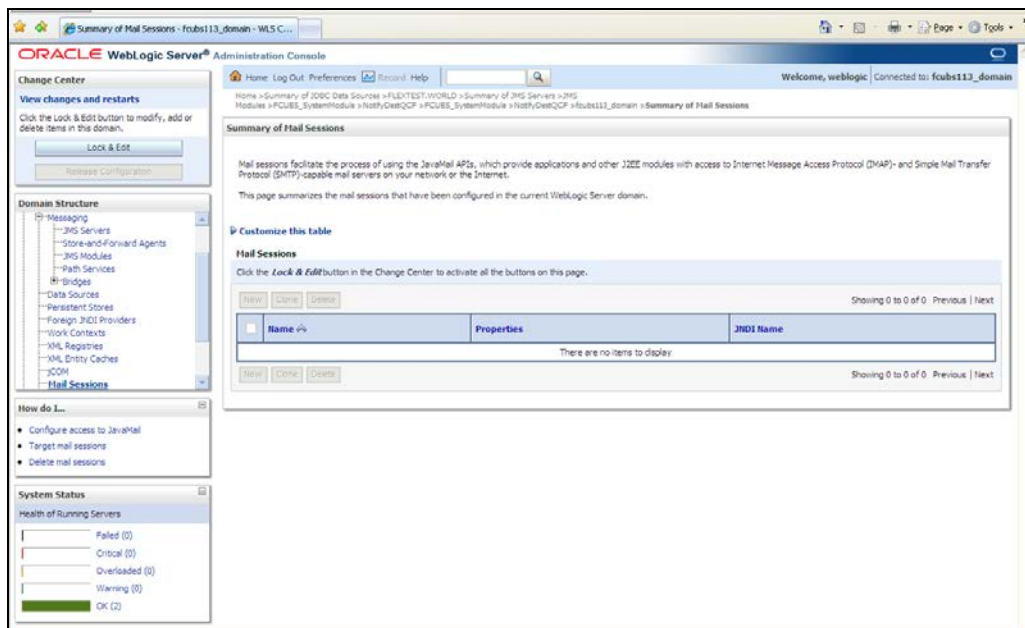
7.4 Setup/Configure Mail Session in Weblogic

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

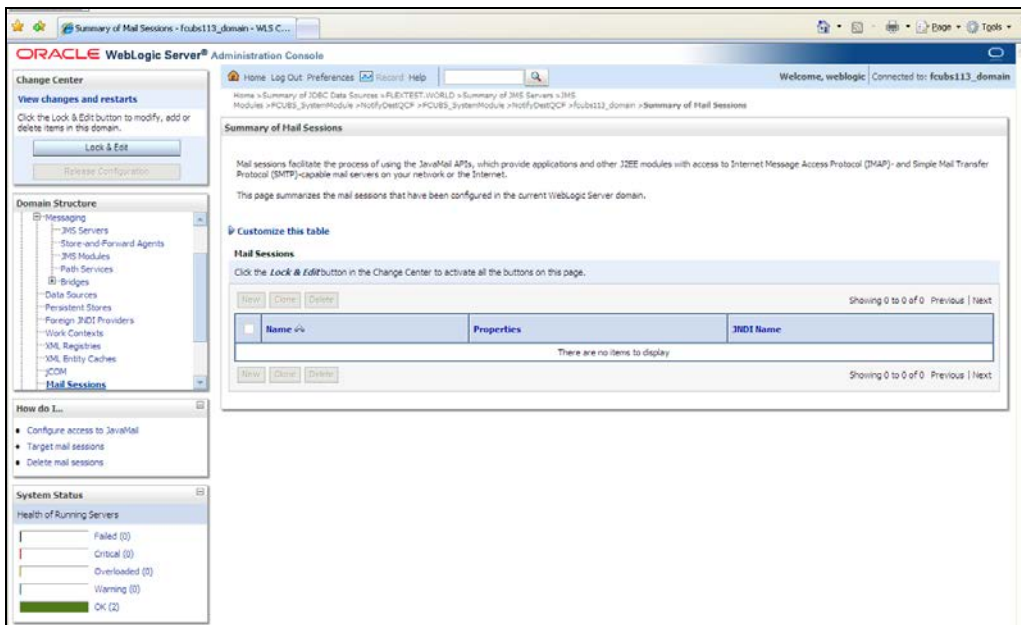
7.4.1 Creating JavaMail Session

To configure mail session, follow the steps below.

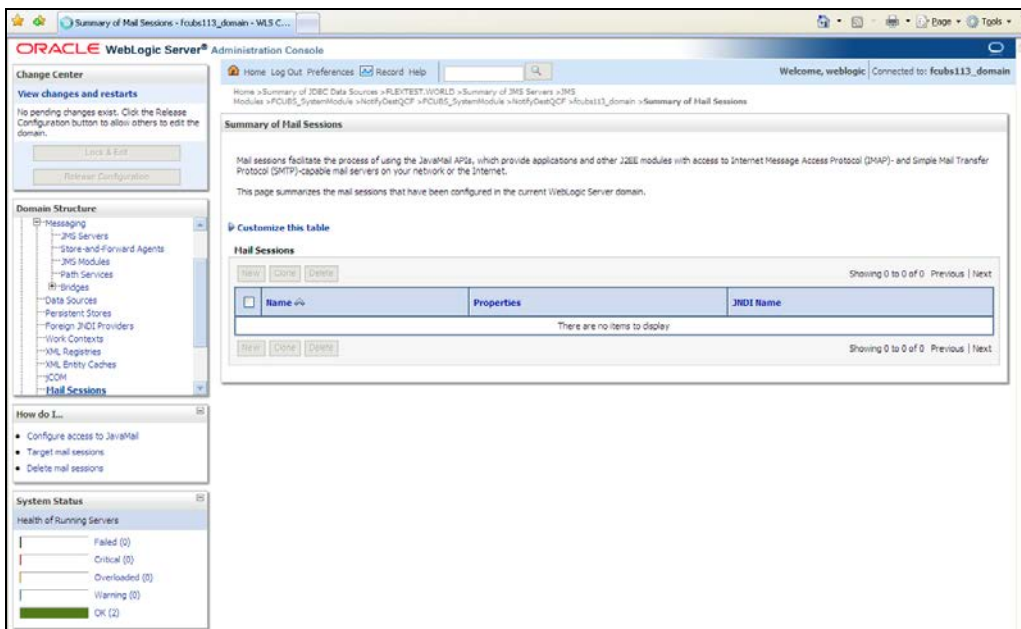
9. Expand 'Services' on the left pane of the application server. Click 'Mail Sessions'.



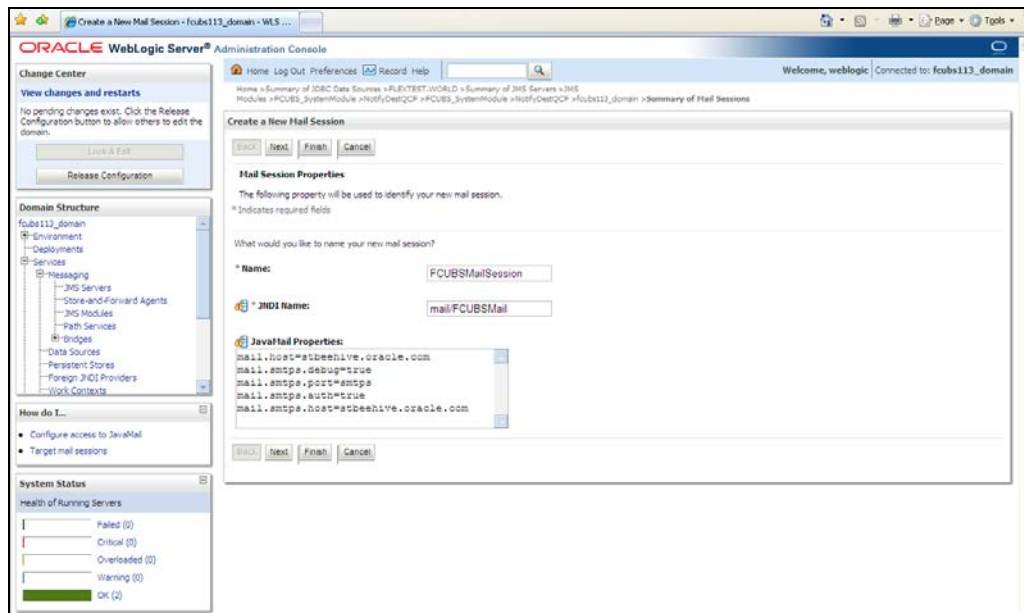
10. Click 'Lock & Edit' for creating a new session.



11. Following screen is displayed.



12. Click 'New' for creating a new session.



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

Name

FCISMailSession

JNDI Name

mail/FCISMail



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

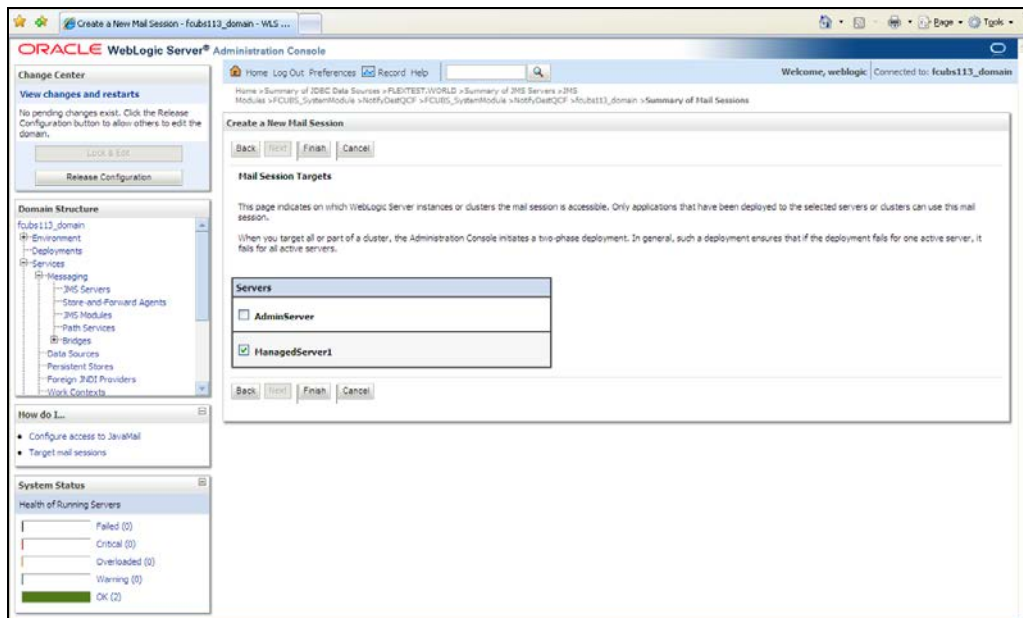
mail.smtps.auth=true

mail.smtps.host==<HOST_SMTPS_MAIL_SERVER>

Eg: samplename.mail.com

14. Click 'Next'.

The following screen is displayed.



15. Check the box against the required servers and click 'Finish' to complete the configuration.

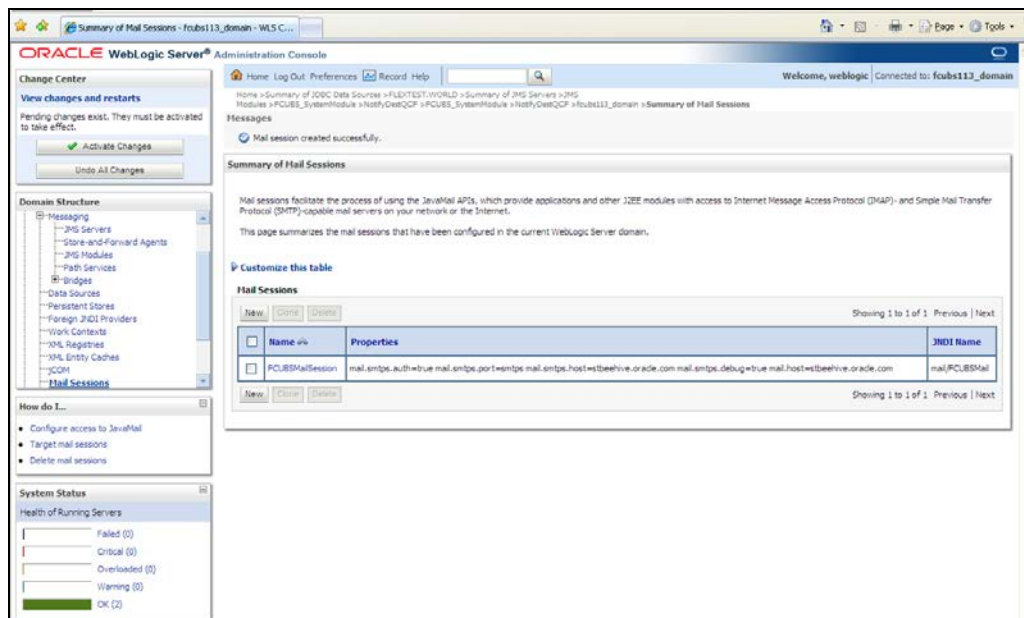


'fcubs.properties' file needs to be updated with the encrypted values of

- SMTP_HOST
- SMTP_USER
- SMTP_PASSWORD
- SMTP_JNDI

This can be achieved using the Oracle FLEXCUBE Investor Servicing Installer.

16. 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 FLEXCUBE INSTALLER SERVICING 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 FLEXCUBE INSTALLER SERVICING 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 FLEXCUBE INSTALLER SERVICING 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).



Weblogic Configuration
[November] [2022]
Version 14.7.0.0.0

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/

Copyright © [2007], [2022], Oracle and/or its affiliates.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

