Oracle® FLEXCUBE Investor Servicing

Weblogic Configuration

14.7.7.0.0
G33234-01
April 2025

Oracle FLEXCUBE Investor Servicing Weblogic Configuration, 14.7.7.0.0

G33234-01

Contents

Preface

Oracle FLEXCUBE Investor Servicing is a comprehensive mutual funds automation software from Oracle® Financial Servicing Software Ltd.©.

You can use the system to achieve optimum automation of all your mutual fund investor servicing processes, as it provides guidelines for specific tasks, descriptions of various features and processes, and general information.

This topic contains the following sub-topics:

Purpose

This manual is designed to help acquaint you with the installation of **Oracle FLEXCUBE Investor Servicing** application.

Audience

This manual is intended for the following User/User Roles:

Table 1 Users and Roles

Users	Roles
Implementation team	Implementation of Oracle FLEXCUBE Investor Servicing
Presales team	Install Oracle FLEXCUBE Investor Servicing for demo purpose
Bank personnel	Who installs Oracle FLEXCUBE Investor Servicing

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used are as follows:

Table 2 Acronyms and Abbreviations

Abbreviation	Description
FCIS	Oracle FLEXCUBE Investor Servicing
OEM	Oracle Enterprise Manager

Table 2 (Cont.) Acronyms and Abbreviations

Abbreviation	Description
EMS	Electronic Messaging Service
EJB	Enterprise Java Bean
MDB	Message Driven Beans

1

Configure SSL on Oracle Weblogic

This topic explains the configurations for SSL on Oracle Weblogic Application Server.

This topic contains the following sub-topics.

1.1 Choose the Identity and Trust Stores

This topic explains how to choose the identity and trust stores.

Oracle Financial Services Software recommends that the choice of Identity and Trust stores be made upfront. 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 Software 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 are located in the `JAVA_HOME/jre/lib/security` directory. It is highly recommended to change the default Java standard trust store password, 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, refer to the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server.

1.2 Set up SSL on Oracle Weblogic

This topic explains the steps to set up the SSL on Oracle Weblogic.

You need to perform the following steps to set up SSL on the Oracle Weblogic Application server:

1. Obtain an identity (private key and digital certificates) and trust (certificates of trusted certificate authorities) for the 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 keystores for the Oracle Weblogic Application server in the Administration console.
4. Set SSL attributes for the private key alias and password in the Oracle Weblogic Administration console.

1.3 Certificates and Keypairs

This topic explains the Certificates and Keypairs used for validating the authenticity of the server.

Certificates are used for validating the authenticity of the server. Certificates contain the name of the owner, certificate usage, duration of validity, resource location, or distinguished name (DN), which includes the common name (CN - website address or e-mail address depending on the usage) and the certificate ID of the person who certified (signs) this 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 key tool 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, the 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 the Oracle Weblogic server to configure SSL.

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

2

Choose the Identity and Trust Stores

This topic explains how to choose the identity and trust stores.

Oracle Financial Services Software recommends that the choice of Identity and Trust stores be made upfront. 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 Software 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 are located in the `JAVA_HOME/jre/lib/security` directory. It is highly recommended to change the default Java standard trust store password, 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, refer to the Oracle Weblogic Server documentation on Securing Oracle Weblogic Server.

3

Obtain the Identity Store

This topic explains the creation of Identity Stores.

This topic contains the following sub-topics.

- Create Identity Store with Self-Signed Certificates
  This topic explains the steps to create Identity Store with Self-Signed Certificates.
- Create Identity Store with Trusted Certificates Issued by CA
  This topic explains to create identity store with trusted certificates issued by CA.

3.1 Create Identity Store with Self-Signed Certificates

This topic explains the steps to create Identity Store with Self-Signed Certificates.

Create 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.

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

Creation of Self-signed Certificate

Browse to the bin folder of JRE from the command prompt and type the following 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 the Oracle Weblogic Server.
3. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE 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 identify the SSL Certificate you are creating, for example, by department or by the 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.

> **Note:**
>
> 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 the Oracle Weblogic Server.

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 Create Identity Store with Trusted Certificates Issued by CA

This topic explains to create identity store with trusted certificates issued by CA.

Create Public and Private Key Pair

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

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

> **Note:**
>
> The placeholders should be replaced with suitable values when running the command.

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 the Oracle Weblogic Server.
3. **First and Last Name (CN):** Enter the domain name of the machine used to access FLEXCUBE 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 identify the SSL Certificate you are creating, for example, by department or by the 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.

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>
```

Generate 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.

> **Note:**
>
> 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.

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 -
keystoreD:\keystores\FCUBSKeyStore.jks

Enter keystore password:[Enter the password used to access the keystore]
Enter key password for <cvrhp0729>
(RETURN if same as keystore password):[Enter the password used to access the
key in the keystore]
```

Obtain 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.

Import 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. 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 is chosen (in the earlier step). 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.

Import 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 -
fileD:\keystores\VerisignIntermediateCA.cer -trustcacerts -
keystoreD:\keystoreworkarea\FCUBSKeyStore.jks

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


Certificate was added to keystore.
```

Import 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.

```
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 the identity store, depending on factors including the trustworthiness of the CA, the necessity of transporting the trust store across the machine, among others.

4

Configure Identity and Trust Stores for Weblogic

This topic explains how to configure identity and trust stores for Weblogic.

- Enable SSL on Oracle Weblogic Server
  This topic provides the systematic instructions to enable SSL on Oracle Weblogic Server.
- Configure Identity and Trust Stores
  This topic provides the systematic instructions to configure identity and trust stores.

4.1 Enable SSL on Oracle Weblogic Server

This topic provides the systematic instructions to enable SSL on Oracle Weblogic Server.

To configure SSL on the Oracle Weblogic server, log in into the **Administration Console** and follow the steps given below:

1. Click the **Lock & Edit** button under **Change Center**.
2. Expand the **Servers** node.
3. Select the name of the server for which you want to enable SSL (example - exampleserver).
4. Navigate to **Configuration** and select the **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 Configure Identity and Trust Stores

This topic provides the systematic instructions to configure identity and trust stores.

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

1. Click the **Lock & Edit** button under **Change Center**.
2. Expand the **Servers** node.
3. Select the name of the server for which you want to configure the keystores (example - exampleserver).
4. Navigate to **Configuration** and select the **Keystores** tab.
5. In the **Keystores** field, select the method for storing and managing private keys/digital certificate pairs and trusted CA certificates.

   This choice should match the one made in the topic *Choose the Identity and Trust Stores*.
6. In the **Identity** section, provide the following details:
   a. **Custom Identity Keystore File Name:** Fully qualified path to the Identity keystore.

b. **Custom Identity Keystore Type:** Set this attribute to JKS, the type of the keystore. If it is left blank, it defaults to **Java KeyStore (JKS)**.

c. **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 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:

   a. **Custom Trust Keystore**: The fully qualified path to the trust keystore.

   b. **Custom Trust Keystore Type**: Set this attribute to JKS, the type of the keystore. If it is left blank, it defaults to **Java KeyStore (JKS)**.

   c. **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 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.

   > **Note:**
   >
   > When identity and trust stores are of the JKS format, the passphrases are not required.

5

Set SSL Attributes for Managed Servers

This topic explains how to set SSL attributes for managed servers.

- Set SSL Attributes for Private Key Alias and Password
  This topic provides the systematic instructions to set SSL attributes for private key alias and password.

5.1 Set SSL Attributes for Private Key Alias and Password

This topic provides the systematic instructions to set SSL attributes for private key alias and password.

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

1. Click the **Lock & Edit** button under **Change Center**.
2. Expand the **Servers** node.
3. Select the name of the server for which you want to configure the keystores (example - exampleserver).
4. Navigate to **Configuration** and select the **SSL** tab.
5. Select **Keystores** from **Identity and Trust Locations**.
6. Under **Identity** section, specify the following details:
   a. **Private Key Alias:** Set this attribute to the alias name defined for the key pair when creating the key pair in the Identity keystore.
   b. **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. Click **Activate Changes** button under **Change Center**.
9. Go to the **controls** tab, check the appropriate server, and click **Restart SSL**. Confirm when it prompts.

6
Test Configuration

This topic explains to test the configuration

Once the Oracle Weblogic has been configured for SSL, deploy the application in the usual manner. The application can be tested in SSL mode after deployment.

To launch the application in SSL mode, enter the URL in the following format: `https://(Machine Name):(SSL_Listener_port_no)/(Context_root)`

It is recommended that the **Oracle FLEXCUBE Investor Servicing** web application be accessed via the HTTPS channel instead of the HTTP channel.

7

Create Resources on Weblogic

This topic explains the steps to be executed to deploy the FCIS and Gateway applications in the Application Server.

This topic contains the following sub-topics:

- Resource Administration
  This topic deals with the process of Resource Administration on Oracle Weblogic.
- Create JMS Server
  This topic explains the systematic instructions to create the JMS server in the Weblogic application server.
- Create JMS Modules
  This topic explains the systematic instructions to create the JMS Modules in the Weblogic application server.
- Create Subdeployment
  This topic explains the systematic instructions to create the subdeployment in the Weblogic application server.
- Create JMS Queue
  This topic explains the systematic instructions to create the JMS Queue in the Weblogic application server.
- Create JMS Connection Factory
  This topic explains the systematic instructions to create the JMS Connection Factory in the Weblogic application server.

7.1 Resource Administration

This topic deals with the process of Resource Administration on Oracle Weblogic.

All the resources mention in the topic *Resources To be Created* are need to be created before deployment. One example for each category is explained in the following sub-topics.

- Create Data Source
  This topic explains the methods to create data sources.
- XA Enabled Data Source
  This topic explains the systematic instructions to create the XA enabled data source in the Weblogic Application server.
- Non-XA Enabled Data Source
  This topic explains the systematic instructions to create the Non-XA enabled data source in the Weblogic application server.
- Scheduler Data Source configuration
  This topic gives an overview to configure Scheduler Data Source.

7.1.1 Create Data Source

This topic explains the methods to create data sources.

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

Prerequisites

To create the data source, the OCI needs to be enabled.

For this, download Oracle Instant Client and install it. The details are given below:

Table 7-1 Oracle Instant Client

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, the user needs to provide the directory path where Instant Client is installed.

The user needs to do the data source configuration with the 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 **setDomainEnv.bat**. This must be the directory path 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 **setDomainEnv.sh**. This must be the directory path 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 **setDomainEnv.sh**.
  - If the target database is Autonomous Database then configure the TNS_ADMIN in the DB client of the Application server with the Autonomous Database Wallet given by the Database Administrator.

7.1.2 XA Enabled Data Source

This topic explains the systematic instructions to create the XA enabled data source in the Weblogic Application server.

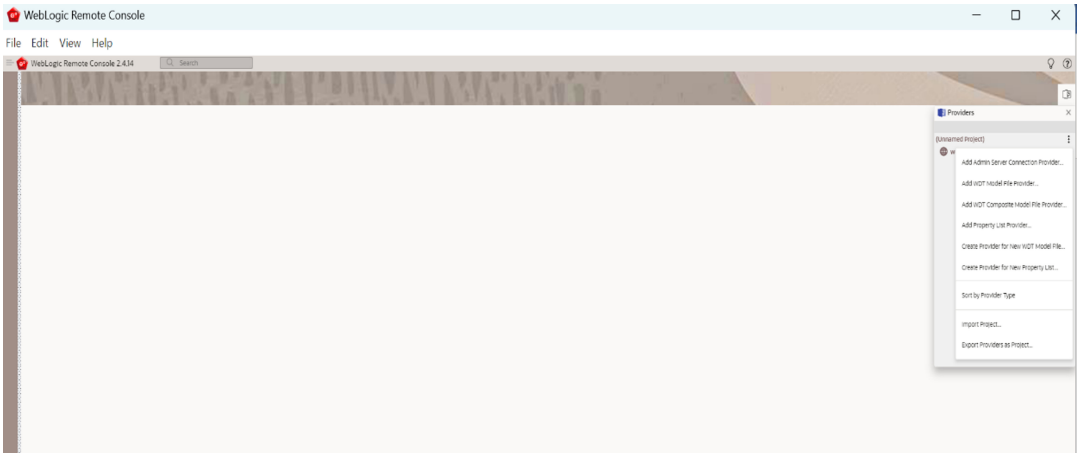
Follow the steps given below to create the XA enabled data source for Gateway Application (MDB):

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

Figure 7-1 Weblogic Remote Console

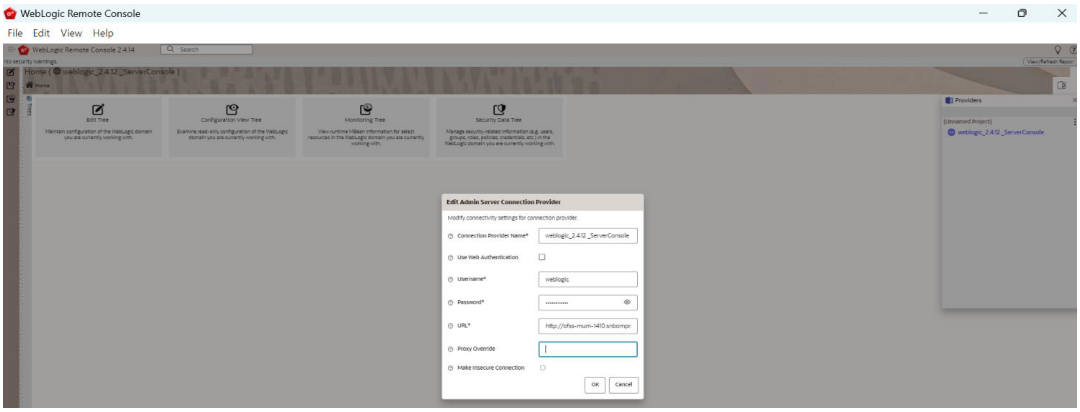


2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

   The **Edit Admin Server Connection Provider** popup window is displayed.

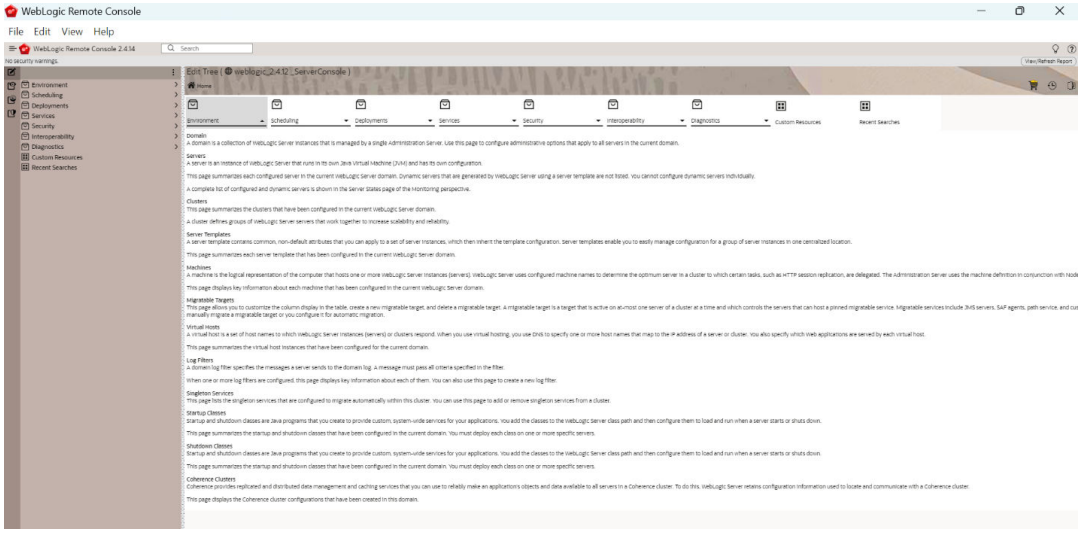
Figure 7-2 Edit Admin Server Connection Provider



3. Click **Edit Tree** icon after logging into the WebLogic Console.

   The following screen is displayed.

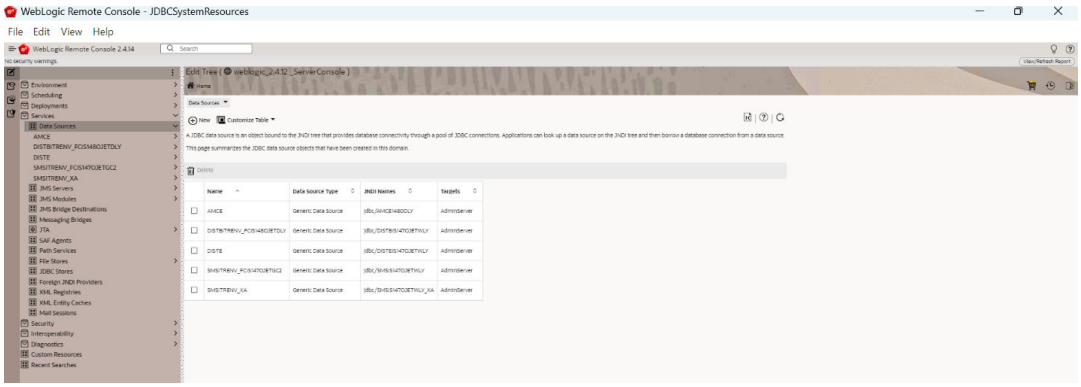
Figure 7-3 Weblogic Remote Console_Edit Tree



4. Go to **Services** and then select **Data Sources**.

   The **Data Sources** screen is displayed.

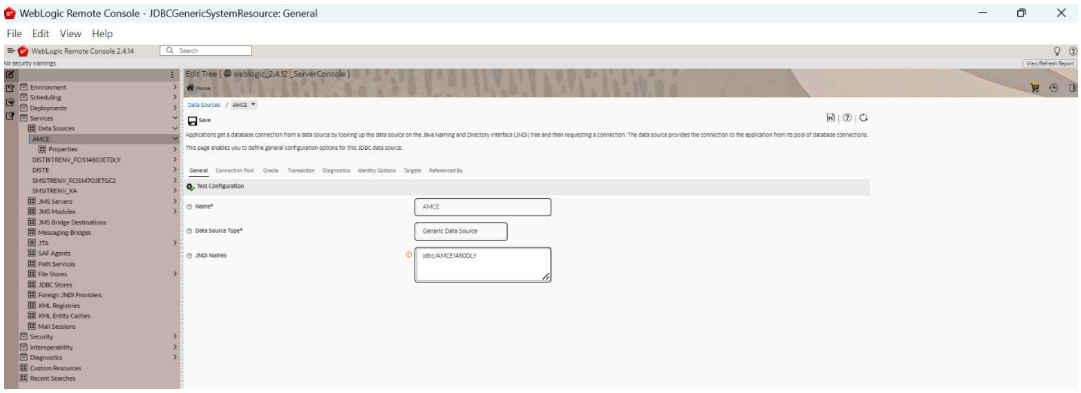
Figure 7-4 Weblogic Remote Console_Services_Data Sources



5. Double-click on the data source created to edit, make the necessary changes, and save them.

   The following screen is displayed.

Figure 7-5 Modify Data Sources



6. Click **New** to create a new data source.

   The following screen is displayed.

Figure 7-6 Create a New Data Source

7. On the **Create a New JDBC System Resource** screen, specify the fields.

   For more information on fields, refer to the field description table.

Table 7-2 Create a New JDBC System Resource

Field	Description
JDBC Datasource Name	Name of the data source. SMSIS147OJETWLY
JNDI Name	JNDI name which will be used for lookup. jdbc/SMSIS147OJETWLY_XA
Database Type	Specify the database type as Oracle from the drop-down list. Oracle
Data Source Type	Generic Data Source
Database Driver	*Oracle's Driver (Thin XA) for Application Continuity Versions: Any
Database Name	Service Name
Host Name	Host Name

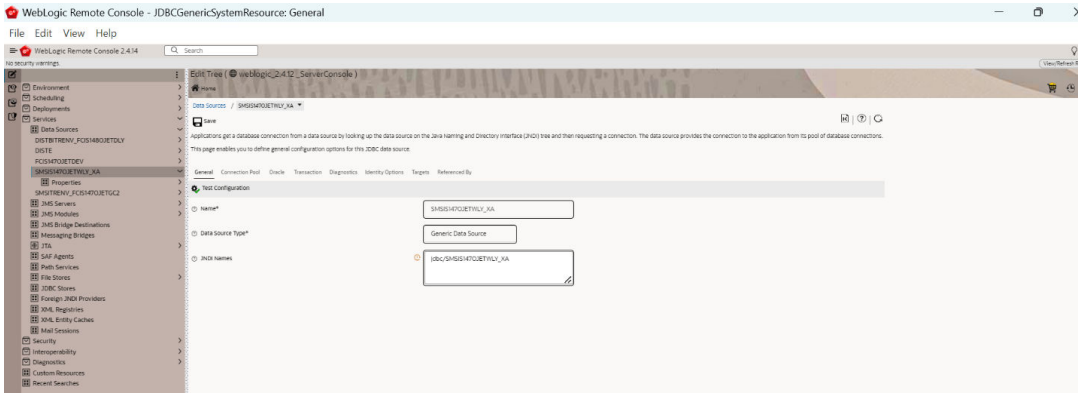
Table 7-2 (Cont.) Create a New JDBC System Resource

Field	Description
Port	Port Number
Database Username	Data Base user id
Password	Data Base Password

8. Click **Create**.

   The following screen is displayed.

Figure 7-7 Click Create

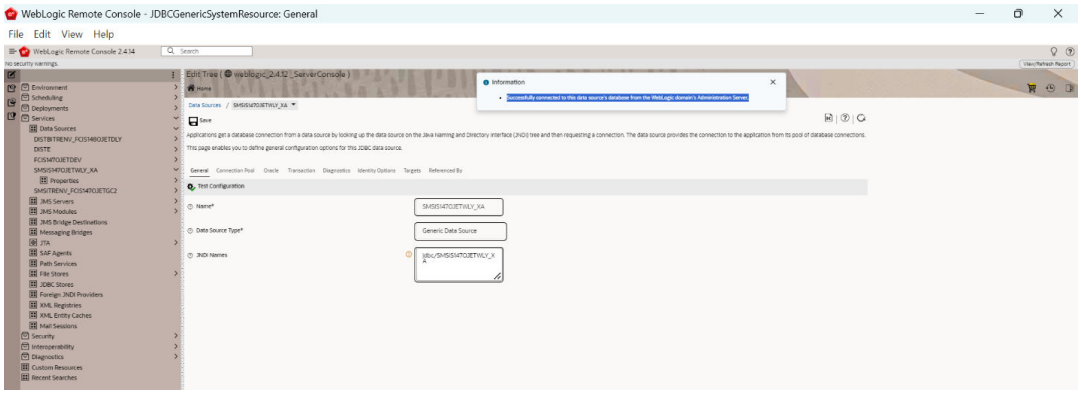


9. Click **Test Configuration** to test the Data source.

   The Applications get a database connection from a data source by looking up the data source on the **Java Naming and Directory Interface (JNDI)** tree and then requesting a connection. If the connection is established successfully, the message `Successfully connected to this data source's database from the WebLogic domain's Administration Server` is displayed.

   The following screen is displayed.

Figure 7-8 Test Configuration Information Message

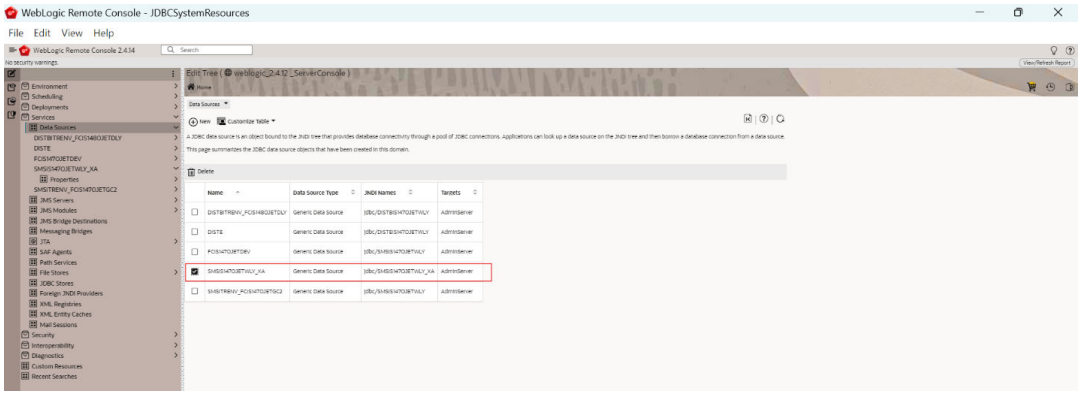


10. You can view the data source created under **Data Sources** in the **Services**.

    SMSIS147OJETWLY_XA datasource has been created.

    The following screen is displayed.

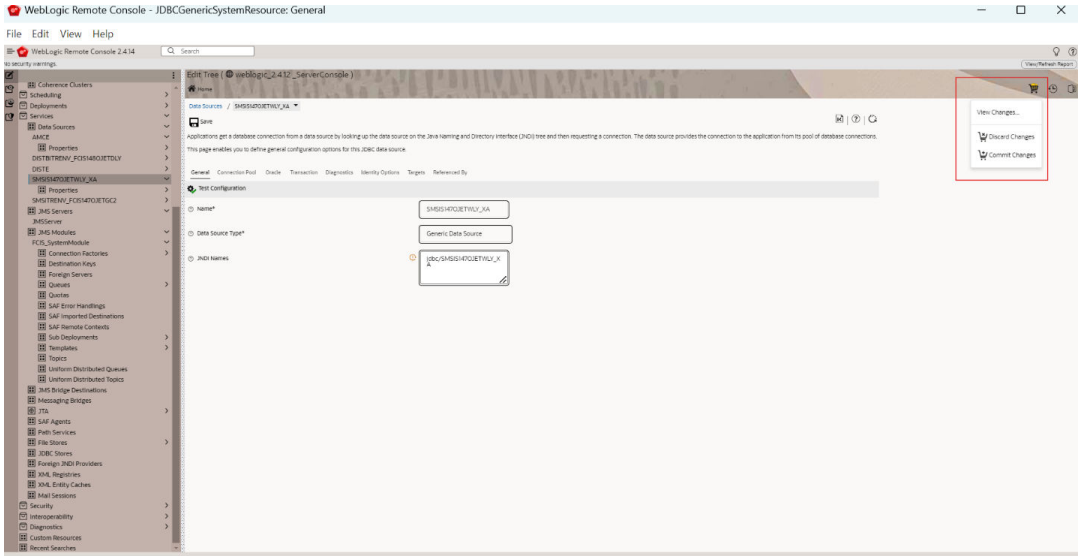
Figure 7-9 View Data Sources



11. Click the **View or Commit** icon to view or commit the changes.

    The following screen is displayed.

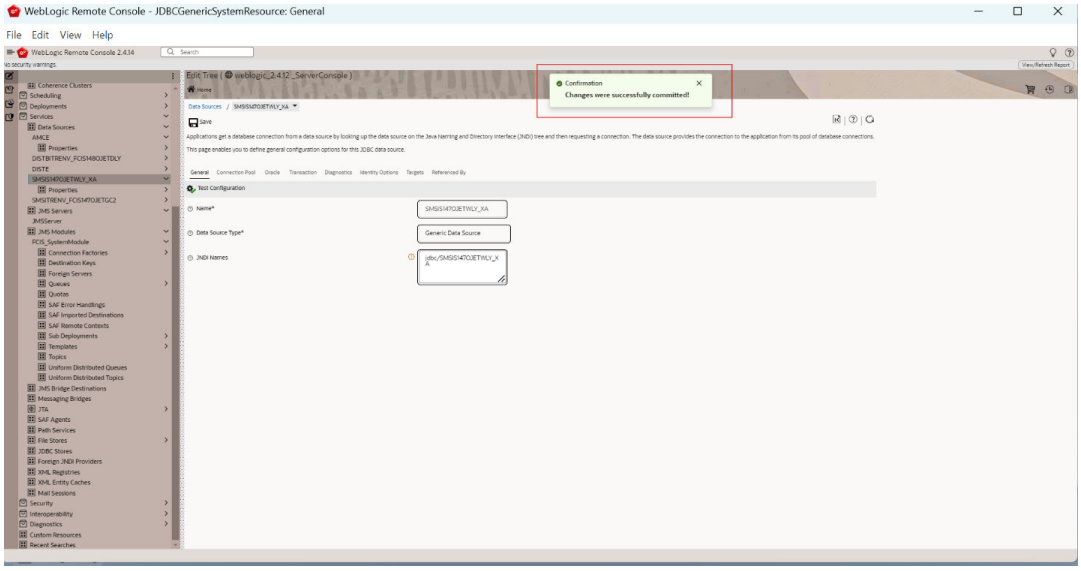
Figure 7-10 View or Commit Changes



12. Select **Commit Changes** option to apply the changes.

    The following screen is displayed.

Figure 7-11 Changes Applied Information Message



7.1.3 Non-XA Enabled Data Source

This topic explains the systematic instructions to create the Non-XA enabled data source in the Weblogic application server.

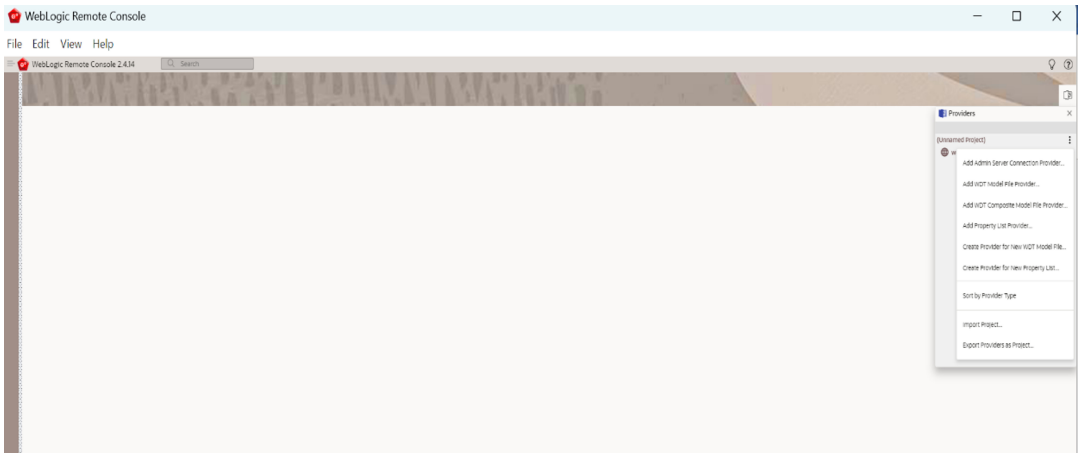
Follow the steps given below to create the XA enabled data source for Gateway Application (MDB):

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

Figure 7-12 WebLogic Remote Console

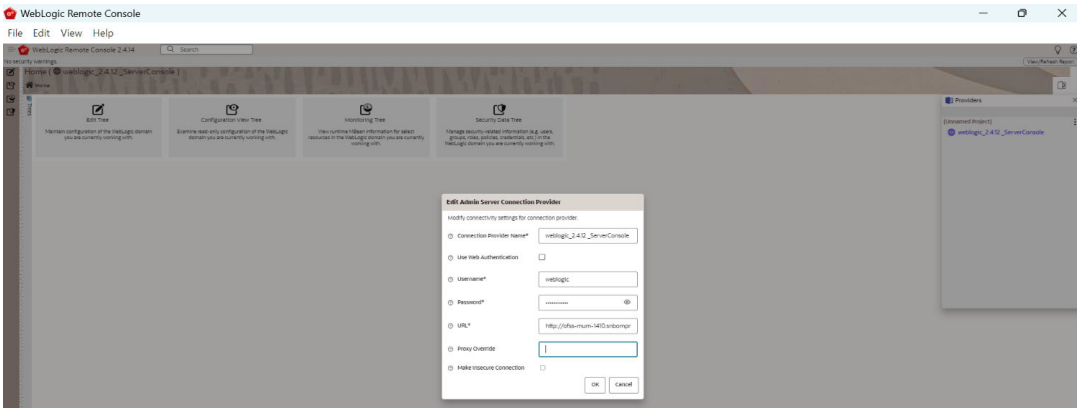


2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

The **Edit Admin Server Connection Provider** popup window is displayed.

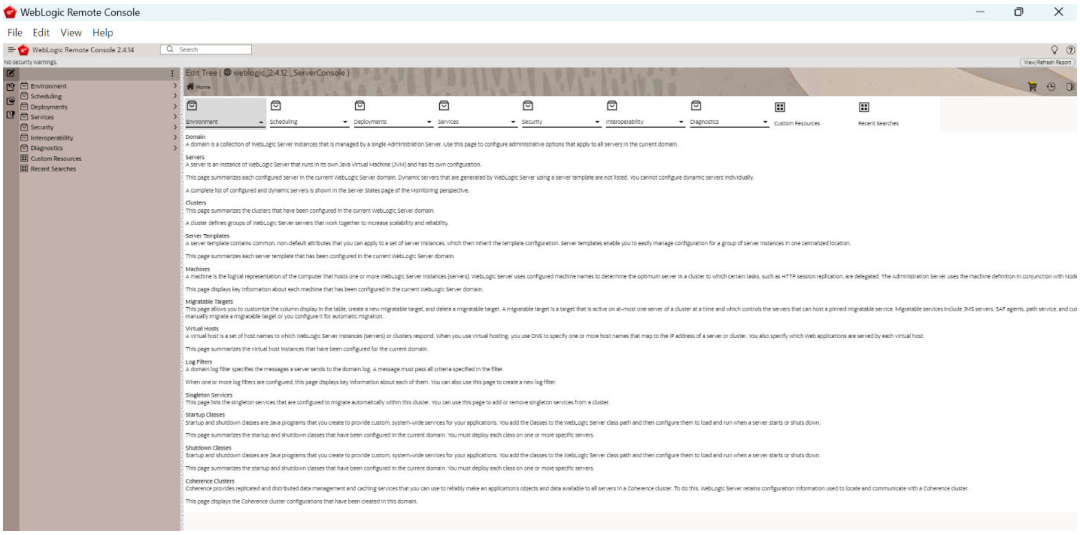
Figure 7-13 Edit Admin Server Connection Provider



3. Click **Edit Tree** icon after logging into the WebLogic Console.

   The following screen is displayed.

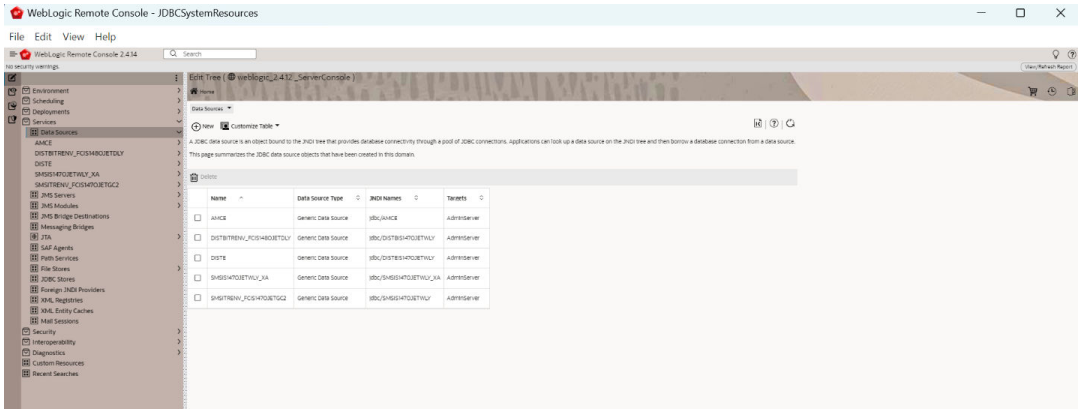
Figure 7-14 Weblogic Remote Console_Edit Tree



4. Go to **Services** and then select **Data Sources**.

   The **Data Sources** screen is displayed.

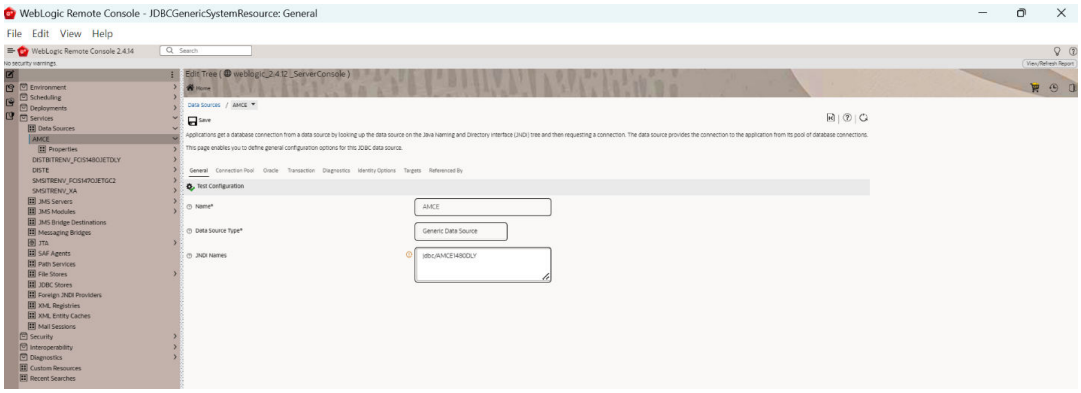
Figure 7-15 Weblogic Remote Console_Services_Data Sources



5. Double-click on the data source created to edit, make the necessary changes, and save them.

   The following screen is displayed.

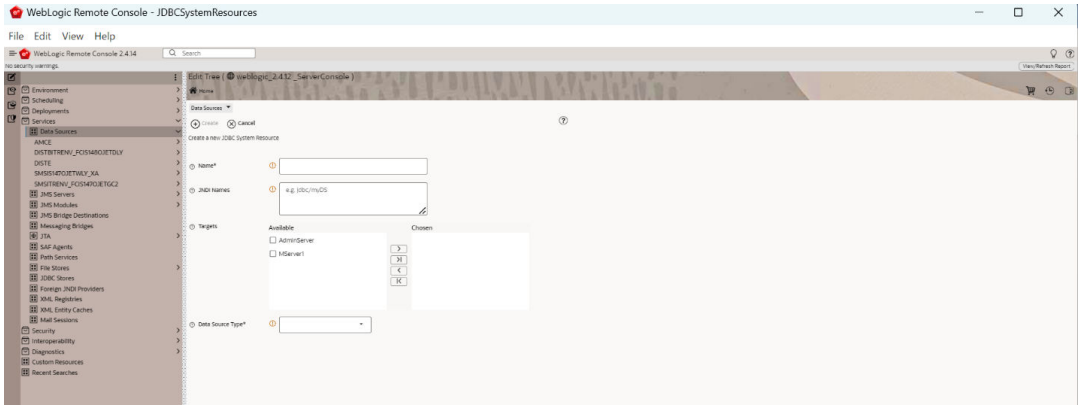
Figure 7-16 Modify Data Sources



6. Click **New** to create a new data source.

   The following screen is displayed.

Figure 7-17 Create a New Data Source



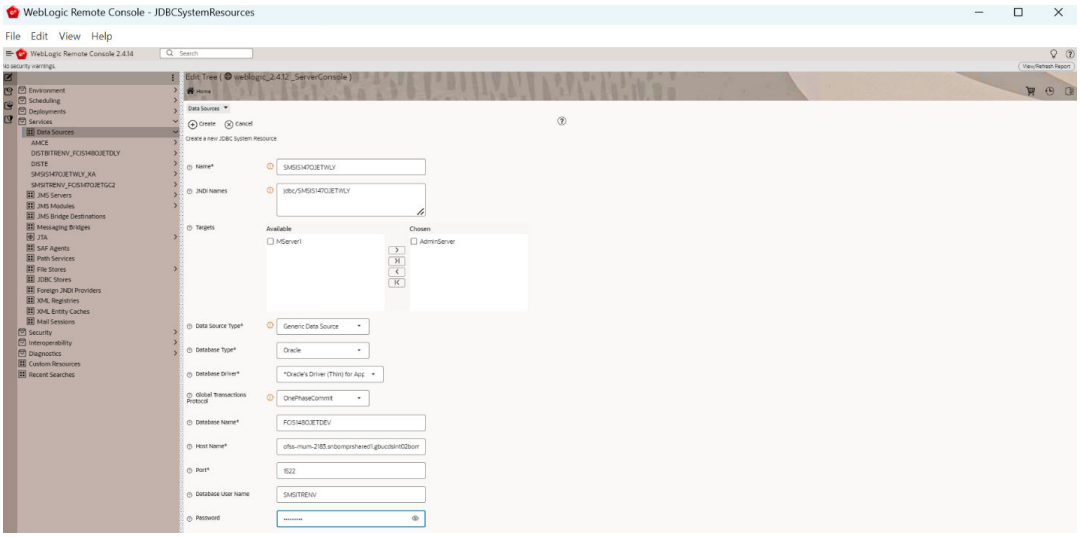
7. On the **Create a New JDBC System Resource** screen, specify the fields.

   For more information on fields, refer to the field description table.

Table 7-3 Create a New JDBC System Resource

Field	Description
JDBC Datasource Name	Name of the data source. SMSIS147OJETWLY
JNDI Name	JNDI name which will be used for lookup. jdbc/SMSIS147OJETWLY_XA
Database Type	Specify the database type as Oracle from the drop-down list. Oracle
Data Source Type	Generic Data Source
Database Driver	*Oracle's Driver (Thin) for Application Continuity Versions: Any
Global Transactions Protocol	OnePhaseCommit
Database Name	DB Service Name
Host Name	DB Host Name
Port	DB Port Number
Database Username	Data Base user name
Password	Data Base Password

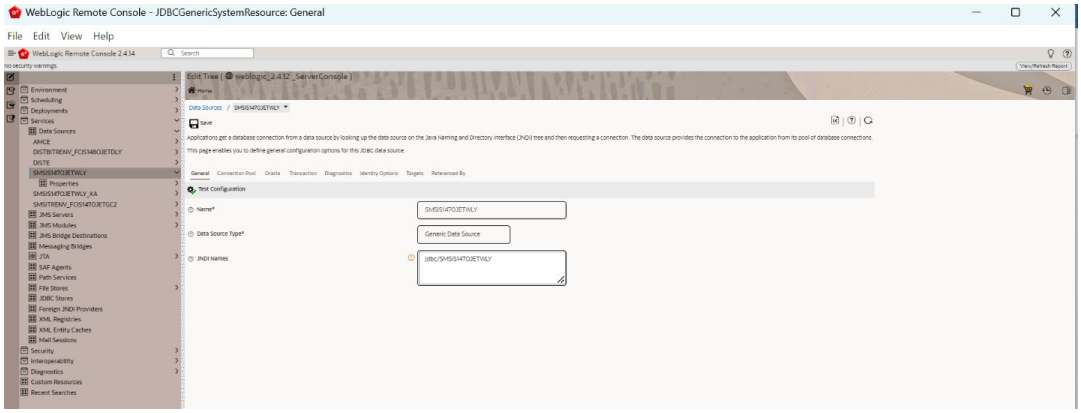
Figure 7-18 Specify Details



8. Click **Create**.

   The following screen is displayed.

Figure 7-19 Click Create

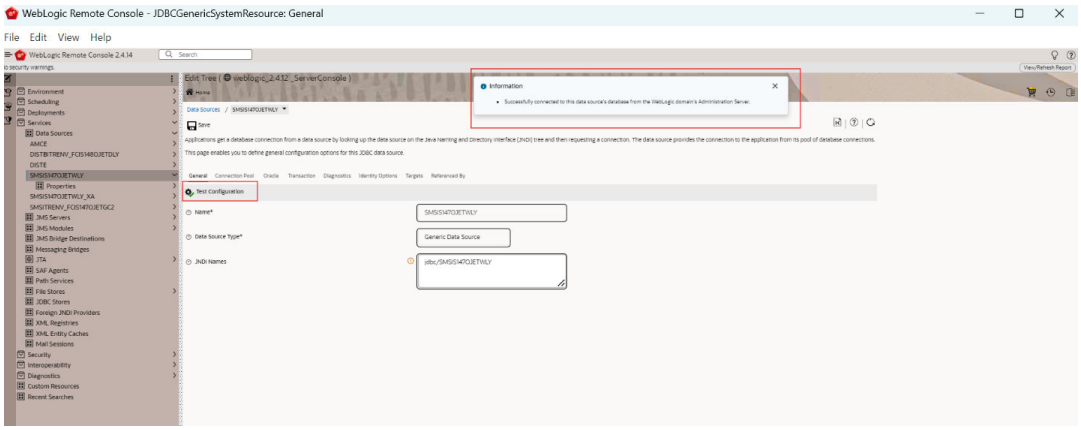


9. Click **Test Configuration** to test the Data source.

   The Applications get a database connection from a data source by looking up the data source on the **Java Naming and Directory Interface (JNDI)** tree and then requesting a connection. If the connection is established successfully, the message `Successfully connected to this data source's database from the WebLogic domain's Administration Server` is displayed.

   The following screen is displayed.

Figure 7-20 Test Configuration Information Message

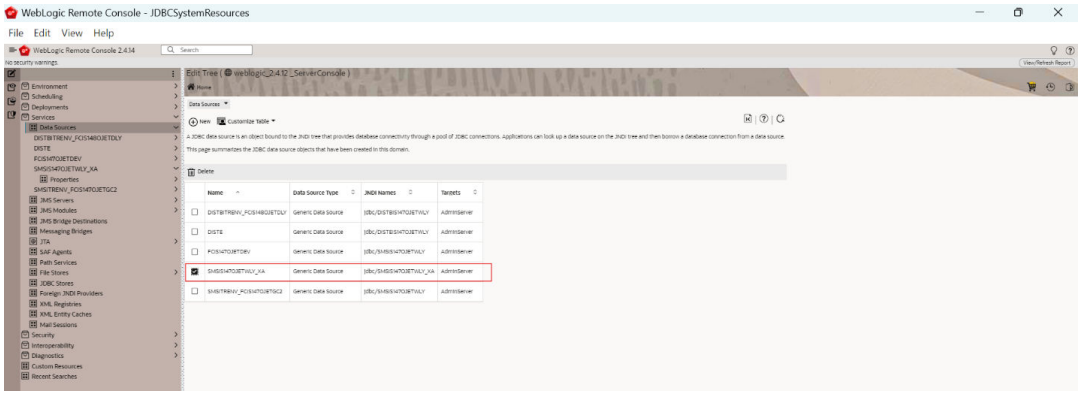


10. You can view the data source created under **Data Sources** in the **Services**.

    SMSIS147OJETWLY_XA datasource has been created.

    The following screen is displayed.

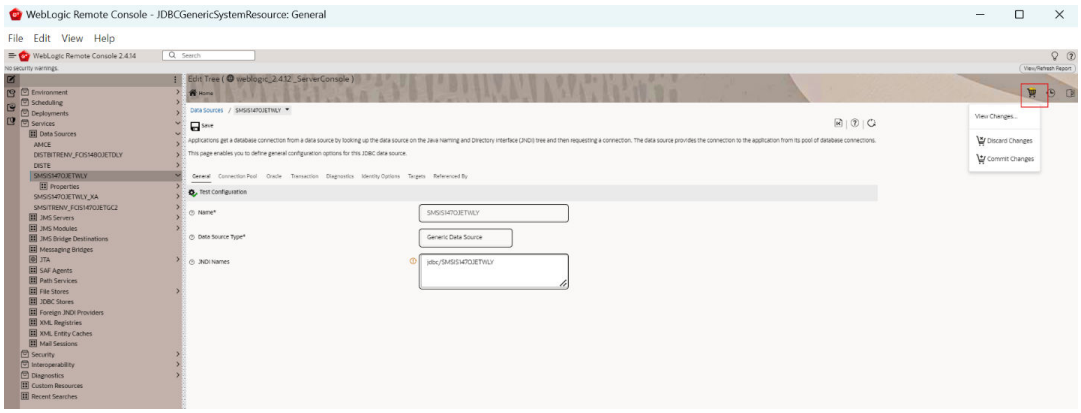
Figure 7-21 View Data Sources



11. Click the **View or Commit** icon to view or commit the changes.

    The following screen is displayed.

Figure 7-22 View or Commit Changes

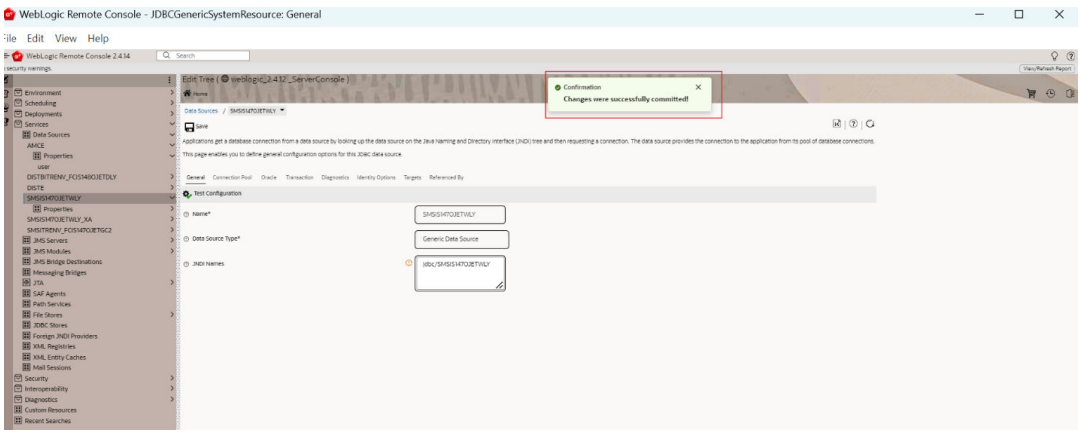


12. Select **Commit Changes** option to apply the changes.

    SMSIS147OJETWLY datasource is created

    The following screen is displayed.

Figure 7-23 Changes Applied Information Message



Note:

- 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`.

7.1.4 Scheduler Data Source configuration

This topic gives an overview to configure Scheduler Data Source.

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 7-1 FCIS And Scheduler Data Source configuration

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

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

Refer the table for the equivalent XA data sources for Scheduler.

Table 7-4 FCIS And Scheduler Data Source configuration

LOB schemas for FCIS	XA Data Sources for Scheduler	Jndi Name for Scheduler
jdbc/BR1204R1	BR1204R1_XA	jdbc/BR1204R1_XA
jdbc/EN1204R1	EN1204R1_XA	jdbc/EN1204R1_XA
jdbc/AMC1204R1	AMC1204R1_XA	jdbc/AMC1204R1_XA

7.2 Create JMS Server

This topic explains the systematic instructions to create the JMS server in the Weblogic application server.

To create the JMS server, follow the steps given below:

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

The **WebLogic Remote Console** screen is displayed.

2. Click **Providers** and select **Add Admin server Connection Provider**.

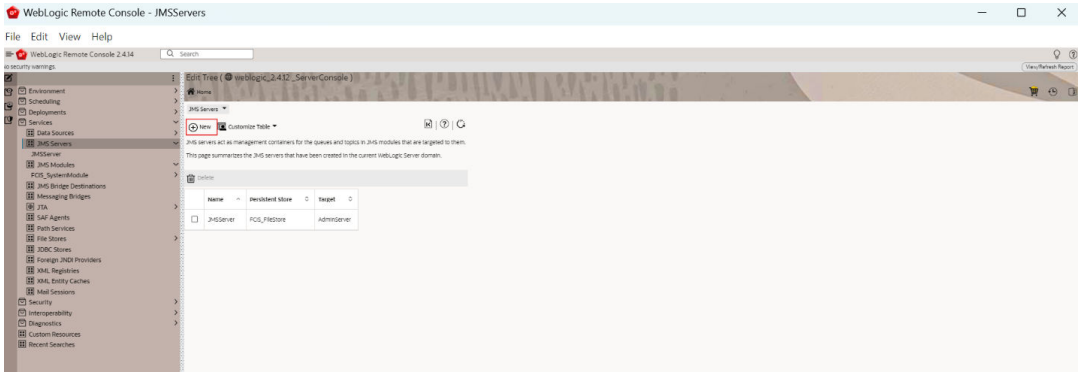
   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

3. Click **Edit Tree** icon after logging into the WebLogic Console.

4. Go to **Services** and then select **JMS Servers**.

   The **JMS Servers** screen is displayed.

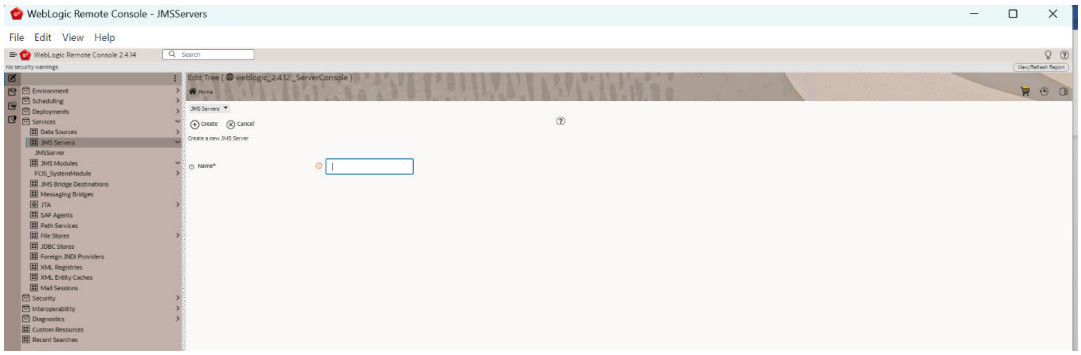
Figure 7-24 JMS Servers



5. Click **New** to create a new JMS Server.

   The following screen is displayed.

Figure 7-25 Create a New JMS Server



6. On the **JMS Servers** screen, specify the fields.

   For more information on fields, refer to the field description table.

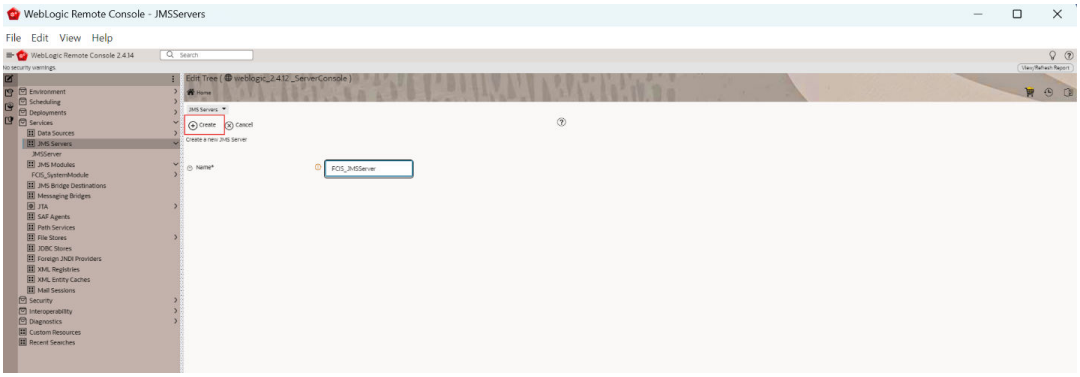
Table 7-5 Create a New JMS Server

Field	Description
JMS Server Name	FCIS_JMSServer

7. Click **Create**.

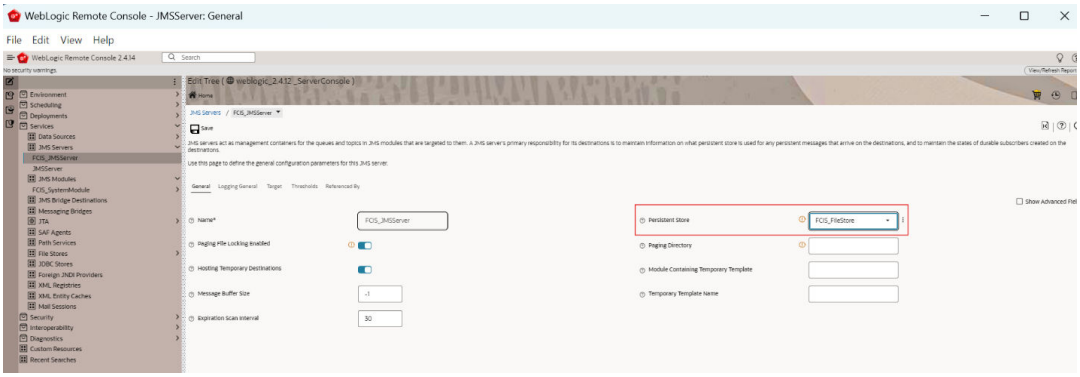
   The following screen is displayed.

Figure 7-26 Click Create



8. Specify the **Persistent Store** as **FCIS_FileStore**.

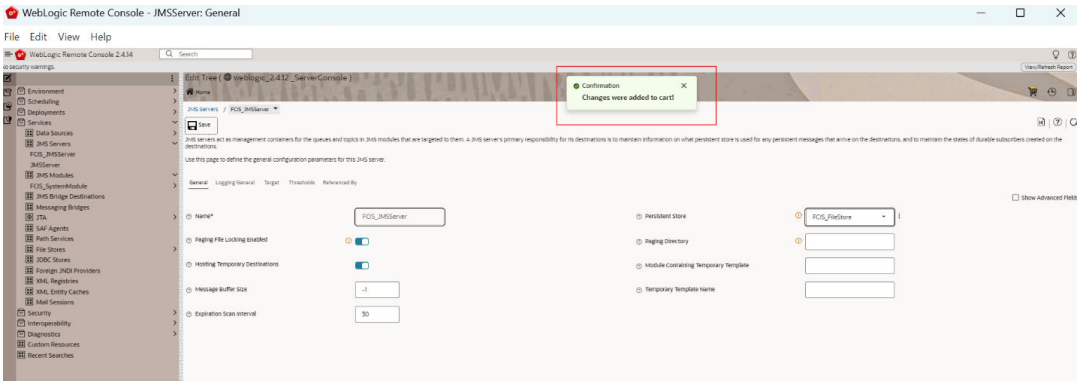
Figure 7-27 JMS Servers_Persistent Store



9. Click **Save**.

   The following screen is displayed.

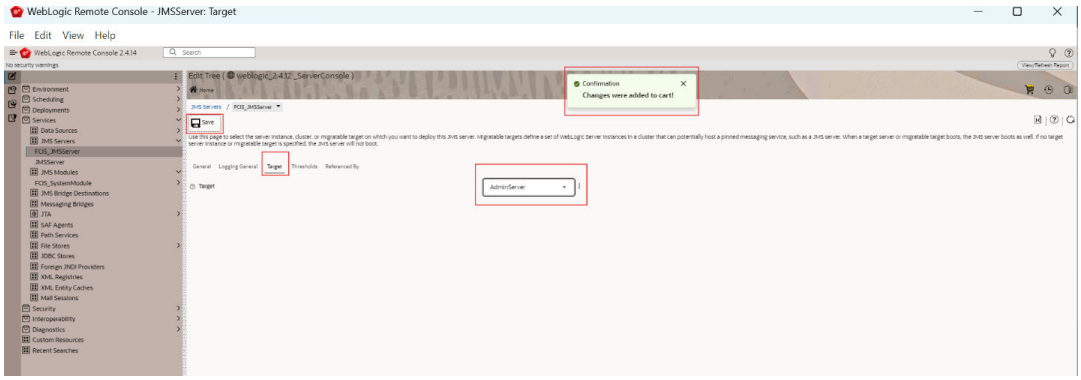
Figure 7-28 Save



10. Click the **Target Tab** and select the target, and save the changes.

    The following screen is displayed.

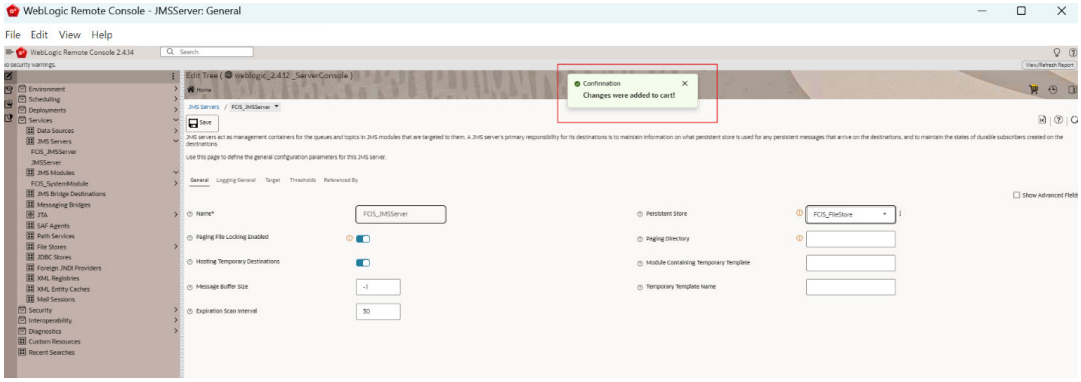
Figure 7-29 Target Tab



11. Click the **Cart** icon to view, commit or discard the changes.

    The `Changes were successfully committed` is displayed on committing the changes.

Figure 7-30 Changes successfully committed



7.3 Create JMS Modules

This topic explains the systematic instructions to create the JMS Modules in the Weblogic application server.

To create the JMS Modules, follow the steps given below:

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

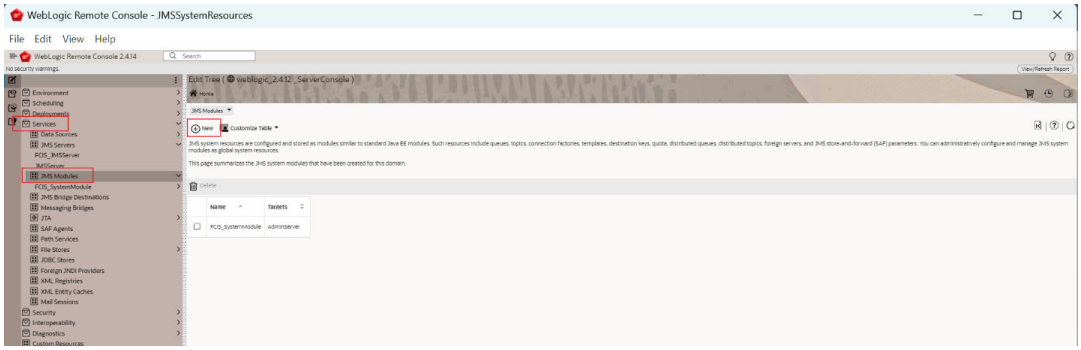
   The **Edit Admin Server Connection Provider** popup window is displayed.

3. Click **Edit Tree** icon after logging into the WebLogic Console.

4. Go to **Services** and then select **JMS Modules**.

   The **JMS Modules** screen is displayed.

Figure 7-31 JMS Modules



5. Click **New** to create a new JMS Module.
6. On the **JMS Modules** screen, specify the fields.

   For more information on fields, refer to the field description table.

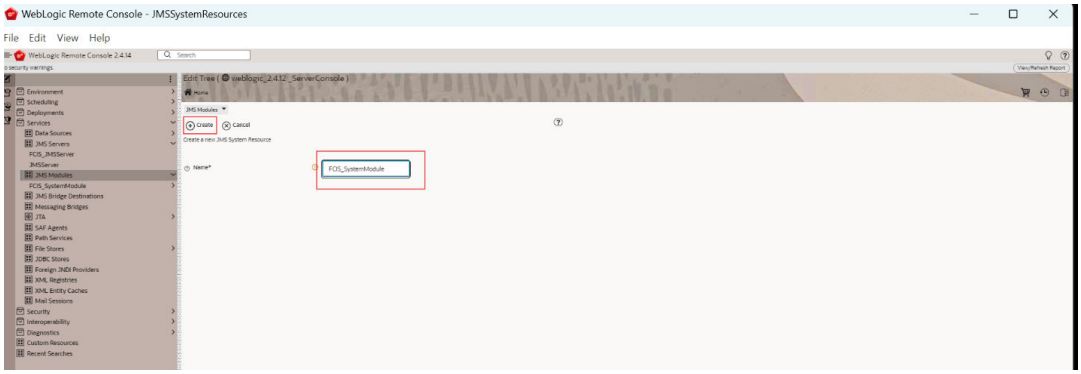
Table 7-6 Create a New JMS Module

Field	Description
Name	JMS Modules Name

7. Click **Create**.

   The following screen is displayed.

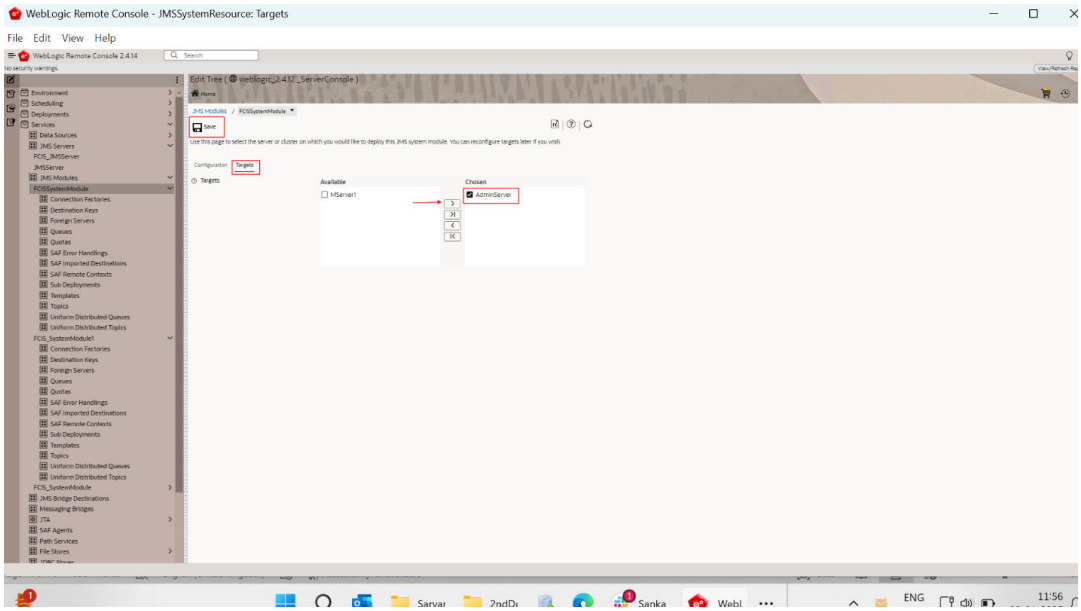
Figure 7-32 Create a New JMS Module



8. Click the **Targets Tab**.

   The following screen is displayed.

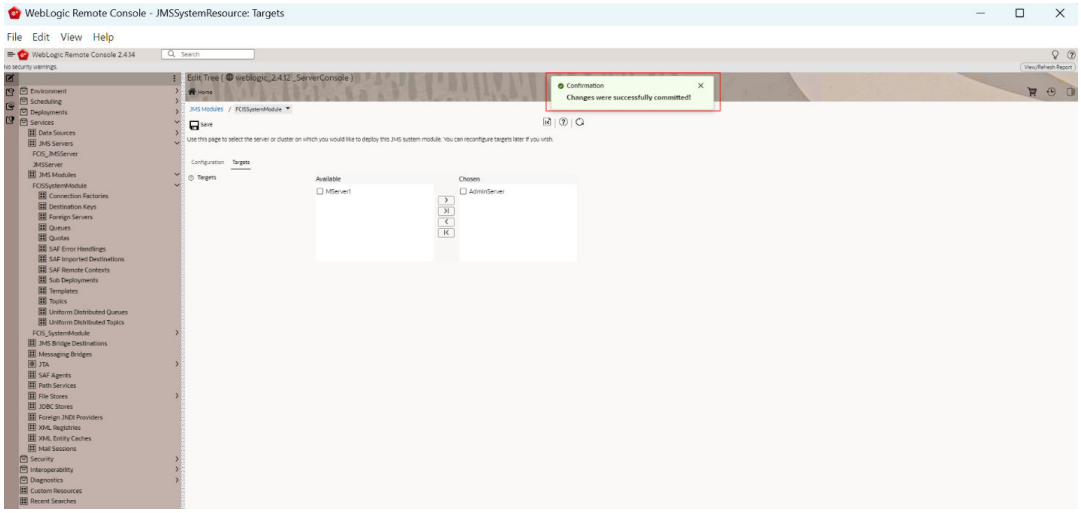
Figure 7-33 JMS Modules_Targets Tab



9. Select the **AdminServer** option and click **Save**.
10. Click the **Cart** icon to view, commit or discard the changes and select **Commit Changes** to apply the changes.

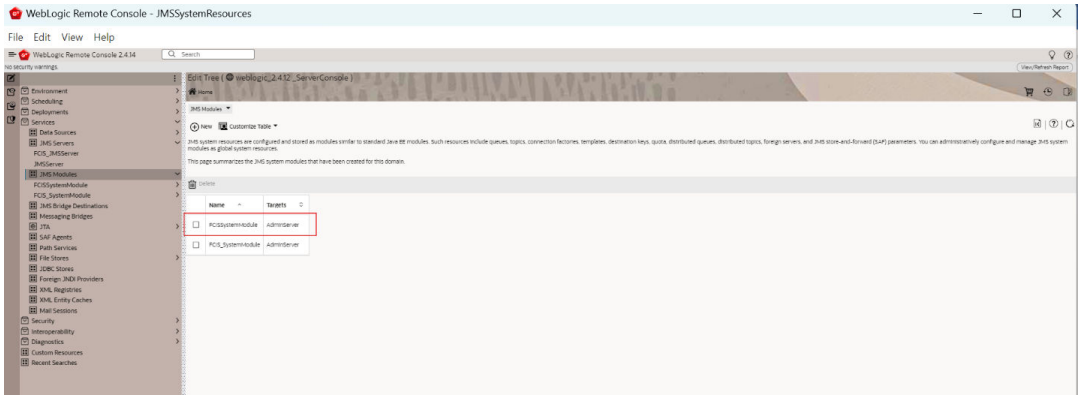
    The `Changes were successfully committed` is displayed on committing the changes.

Figure 7-34 Changes successfully committed



11. You can view the new module created in the **JMS Modules** screen.

Figure 7-35 New Module Created



7.4 Create Subdeployment

This topic explains the systematic instructions to create the subdeployment in the Weblogic application server.

Follow the steps given below to create the subdeployments:

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

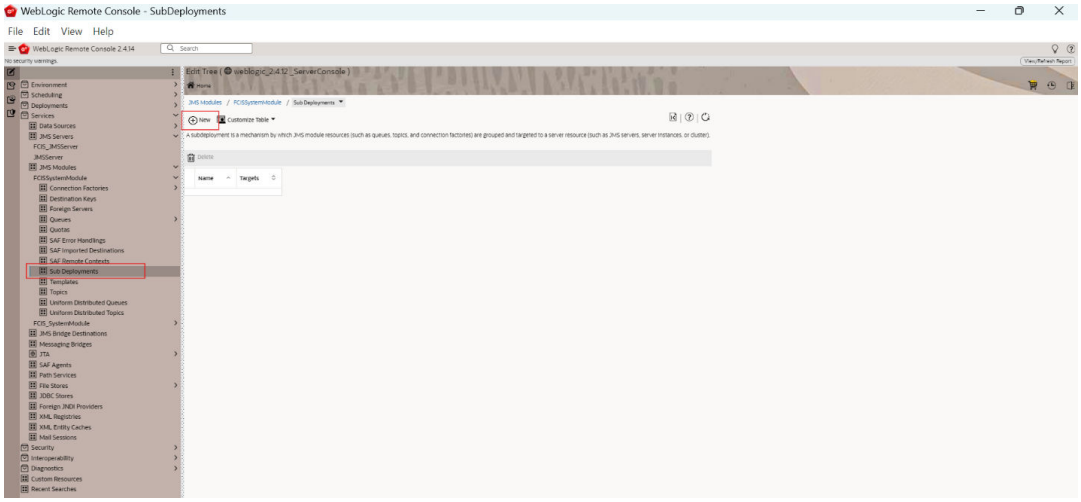
   The **Edit Admin Server Connection Provider** popup window is displayed.

3. Click **Edit Tree** icon after logging into the WebLogic Console.

4. Go to **Services** and then select **Sub Deployments** within **JMS Modules**.

   The **Sub Deployments** screen is displayed.

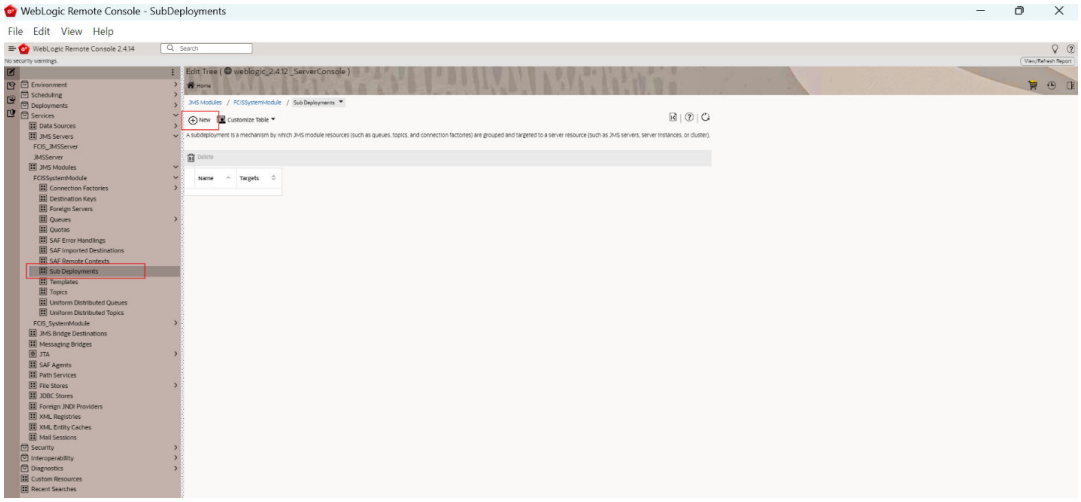
Figure 7-36 Sub Deployments



5. Select the JMS module created and click the **Sub Deployments**.

   The following screen is displayed.

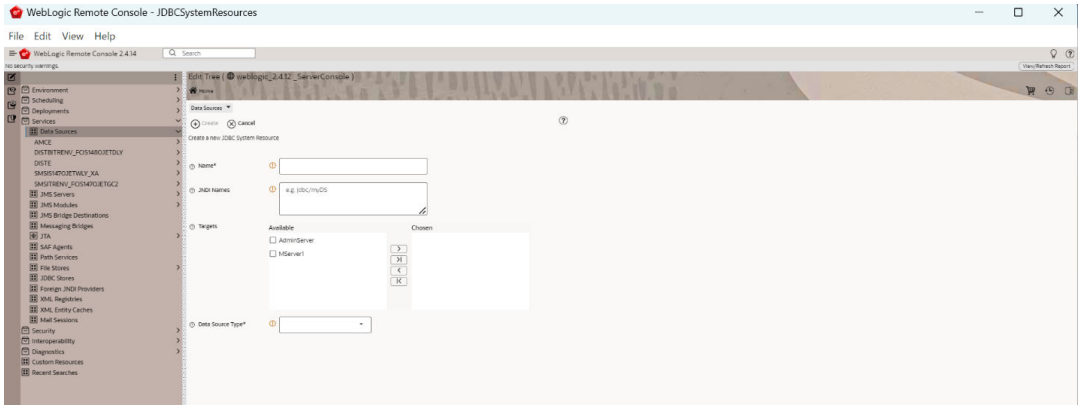
Figure 7-37 Create a New Sub Deployment



6. Click **New** to create a new Sub Deployment.

   The following screen is displayed.

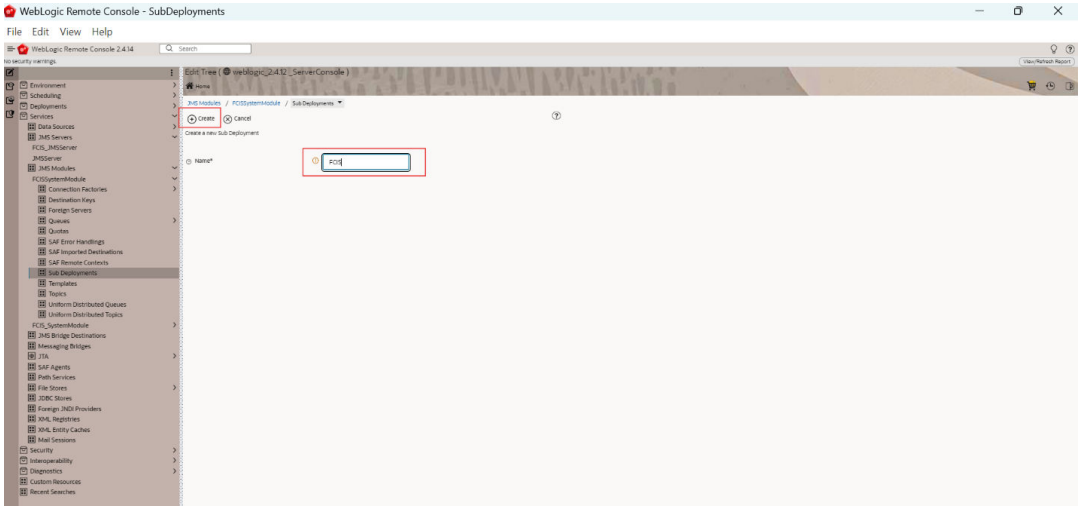
Figure 7-38 Create a New Sub Deployment



7. Specify the **Subdeployment Name** as **FCIS**.
8. Click **Create**.

   The following screen is displayed.

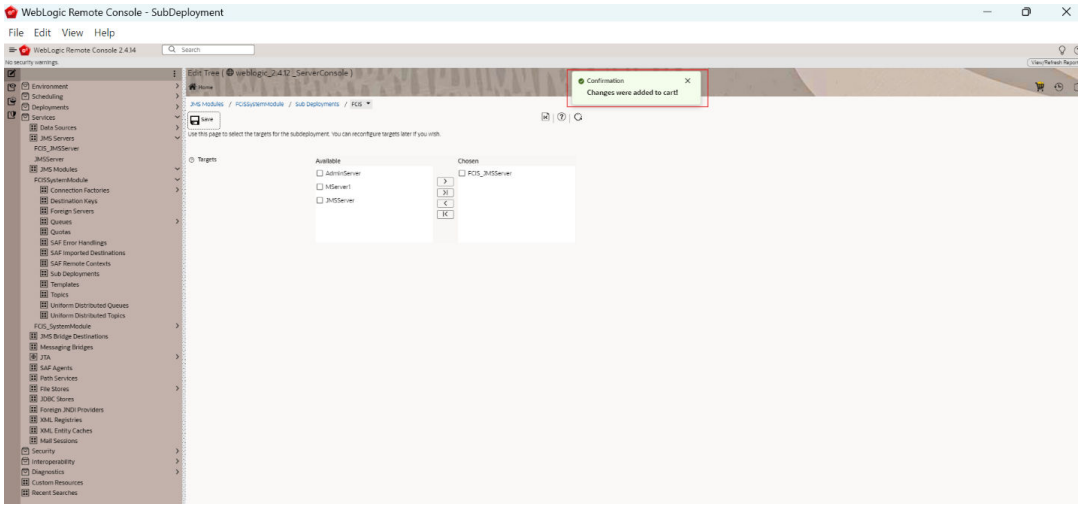
Figure 7-39 Sub Deployments_Create



9. Select the **JMS Server** created.
10. Click **Save**.

    The following screen is displayed.

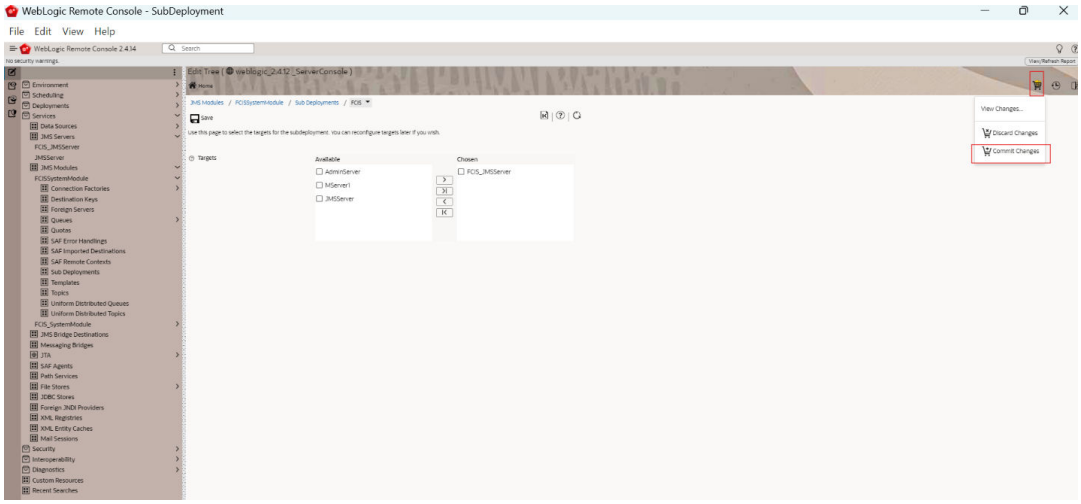
Figure 7-40 Save a new Sub Deployment



11. Click the **Cart** icon to view, commit or discard the changes and select **Commit Changes** to apply the changes.

    The `Changes were successfully committed` is displayed on committing the changes.

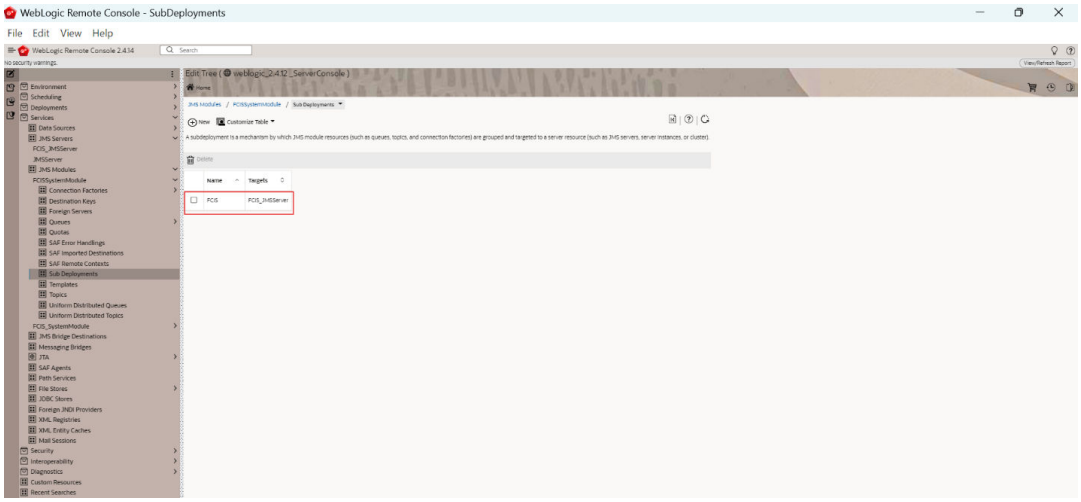
Figure 7-41 Changes successfully committed



12. You can view the new module created in the **JMS Modules** screen.

    The new Subdeployment created is displayed.

Figure 7-42 New Subdeployment created



7.5 Create JMS Queue

This topic explains the systematic instructions to create the JMS Queue in the Weblogic application server.

Follow the steps given below to create the JMS Queue:

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

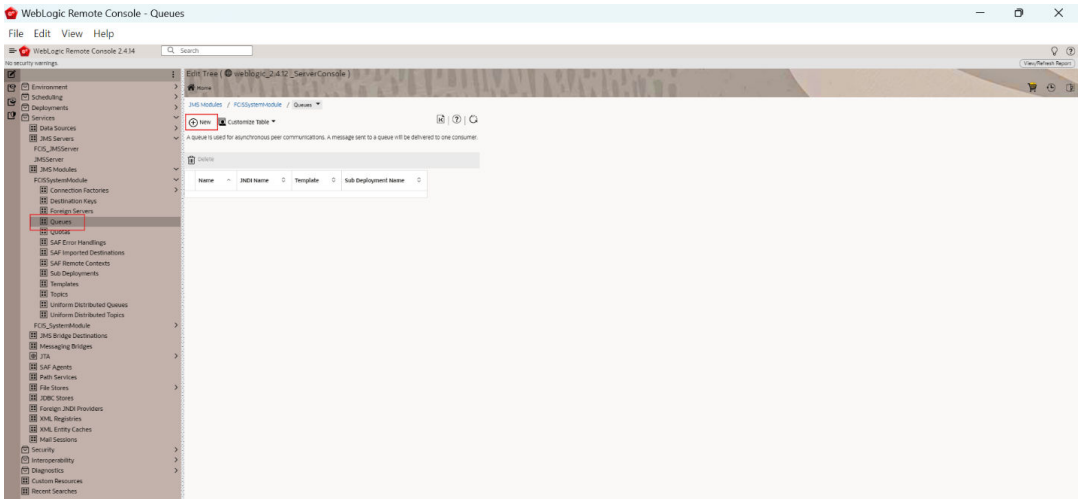
   The **Edit Admin Server Connection Provider** popup window is displayed.

3. Click **Edit Tree** icon after logging into the WebLogic Console.

4. Go to **Services** and then select **JMS Module** created earlier.

   The **Queues** option is displayed.

Figure 7-43 JMS Queues



5. Click **New** to create a new Queue.
6. On the **Create a new Queue** screen, specify the fields.

   For more information on fields, refer to the field description table.

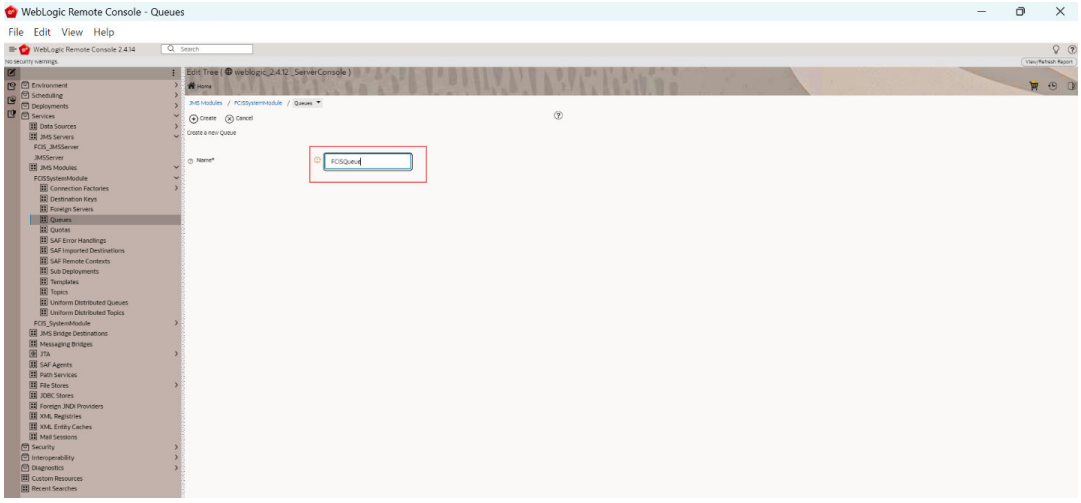
Table 7-7 Create a New Queue

Field	Description
Name	Queue Name

7. Click **Create**.

   The following screen is displayed.

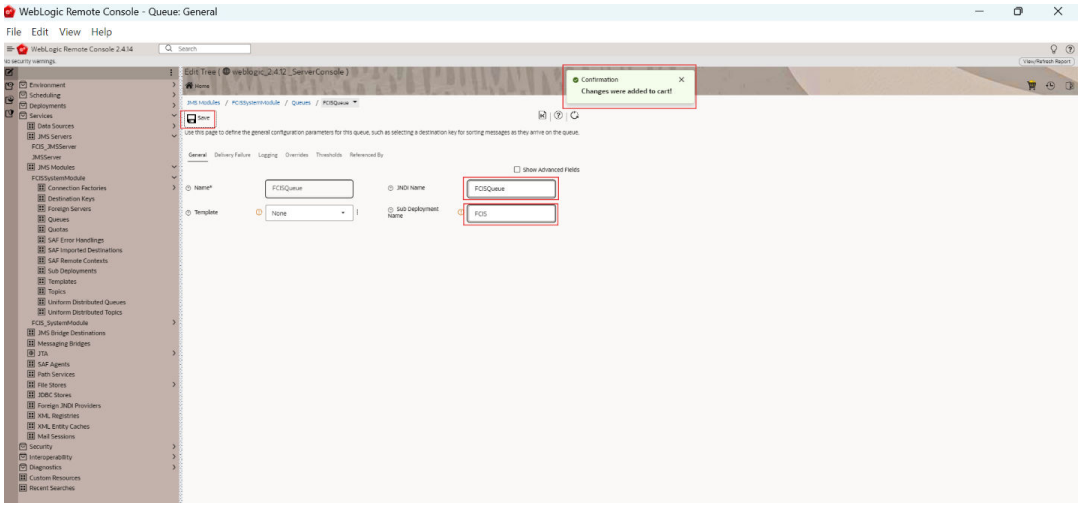
Figure 7-44 Create a New Queue



8. You need to set the configuration for FCIS_SystemModule. Click **Save**.

   The following screen is displayed.

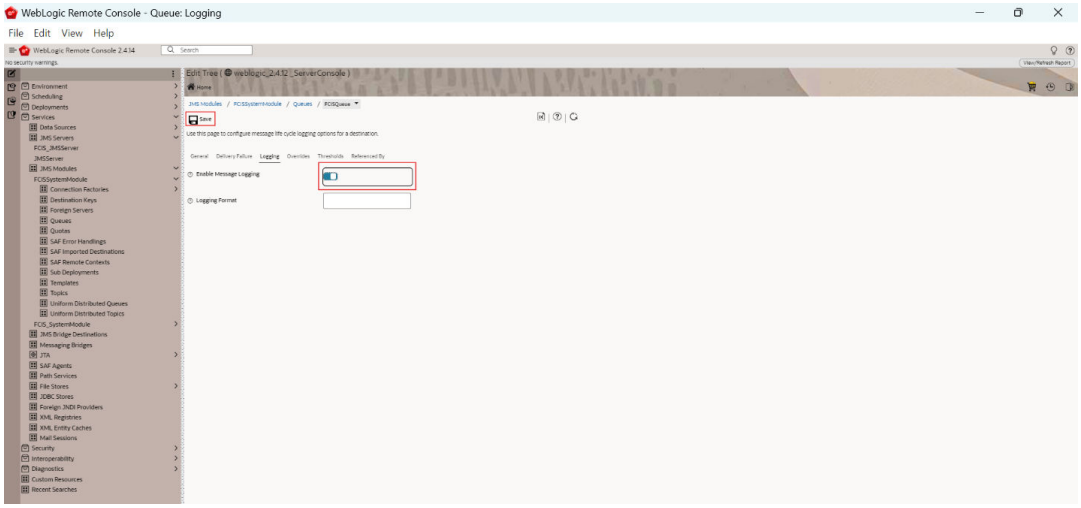
Figure 7-45 Save New Queue



9. Click the **Logging** Tab. Select the **Enable Message Logging**.

   The following screen is displayed.

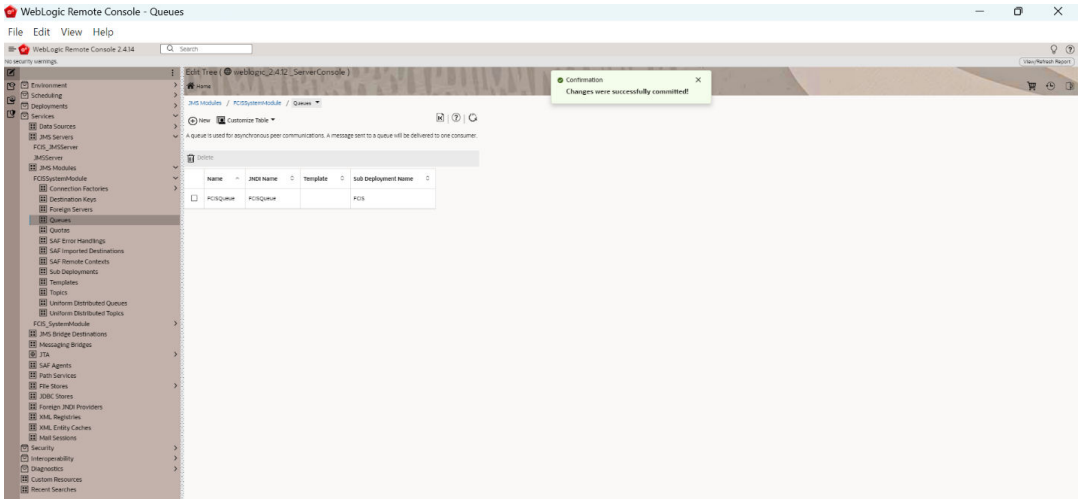
Figure 7-46 Queues_Logging Tab



10. Click the **Cart** icon to view, commit or discard the changes and select **Commit Changes** to apply the changes.

    The `Changes were successfully committed` is displayed on committing the changes.

Figure 7-47 Commit Changes



11. You can create more queues by repeating the same steps to create the other queues.

    The **JMS Queues** created are displayed.

Figure 7-48 More Queues

	Name	JNDI Name	Template	Sub Deployment Name
☐	EMS_INQUEUE	EMS_INQUEUE		FCIS
☐	EMS_OUTQUEUE	EMS_OUTQUEUE		FCIS
☐	EMS_QUEUE_BACK	EMS_QUEUE_BACK		FCIS
☐	EMS_QUEUE_DLQ	EMS_QUEUE_DLQ		FCIS
☐	NOTIFY_DEST_QUEUE	NOTIFY_DEST_QUEUE		FCIS

7.6 Create JMS Connection Factory

This topic explains the systematic instructions to create the JMS Connection Factory in the Weblogic application server.

You need to create the connection factory after creating the queues. To create the JMS Connection Factory, follow the steps given below:

1. Start the Administration Console of WebLogic Application server.

   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Click **Providers** and select **Add Admin server Connection Provider**.

   The user must enter the required URL, username, and password to establish a connection to the Admin Console.

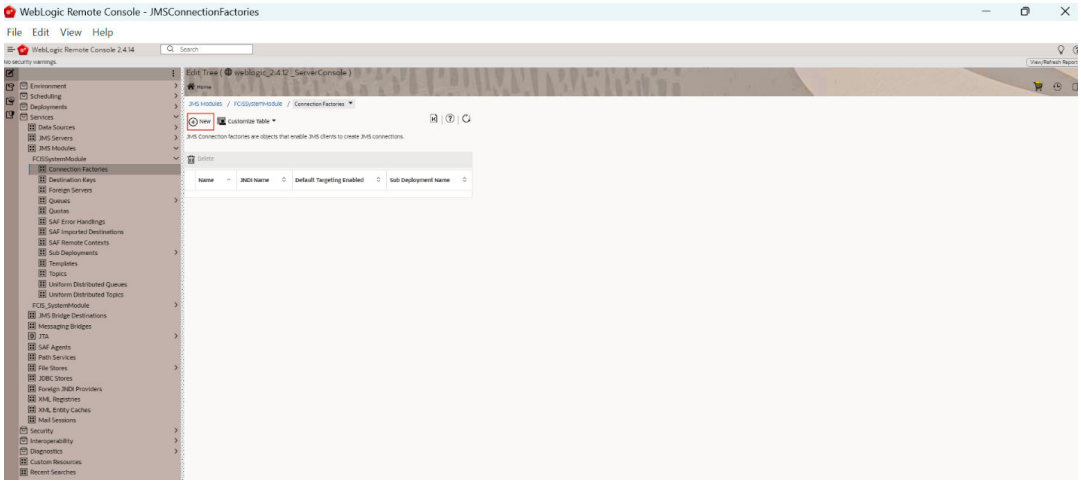
   The **Edit Admin Server Connection Provider** popup window is displayed.

3. Click **Edit Tree** icon after logging into the WebLogic Console.

4. Go to **Services** and then select **JMS Module** created earlier.

   The **Connection Factory** option is displayed.

Figure 7-49 Connection Factory



5. Click **New** to create a new Connection Factory.
6. On the **Create a new Connection Factory** screen, specify the fields.

   For more information on fields, refer to the field description table.

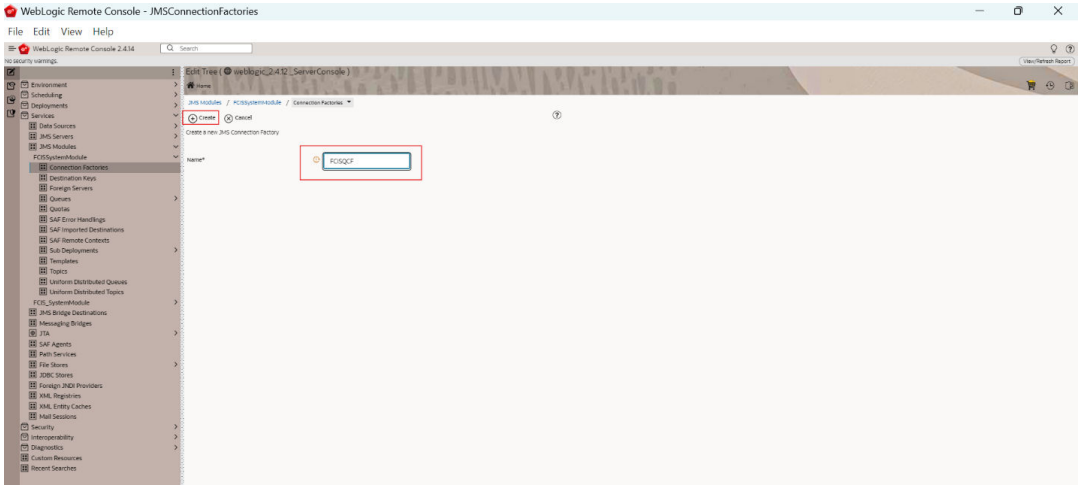
Table 7-8 Create a new Connection Factory

Field	Description
Name	Connection Factory Name

7. Click **Create**.

   The following screen is displayed.

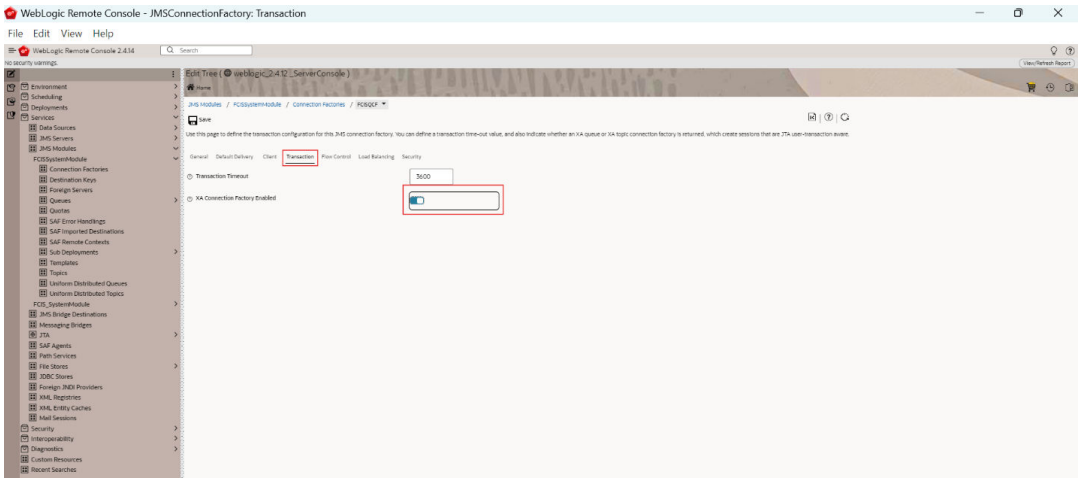
Figure 7-50 Create a new Connection Factory



8. Specify the JNDI Name as **FCISQCF**. Click **Save**.
9. Click the **Transaction** Tab. Select the **XA Connection Factory Enabled**.

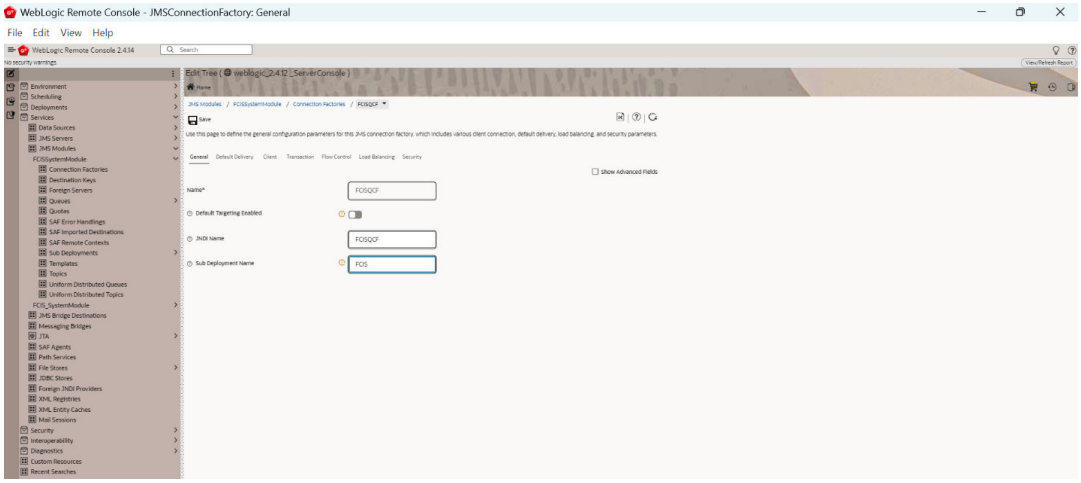
   The following screen is displayed.

Figure 7-51 Connection Factories_Transaction Tab



10. On the **General** tab, select the **Sub Deployment Name** as **FCIS**.

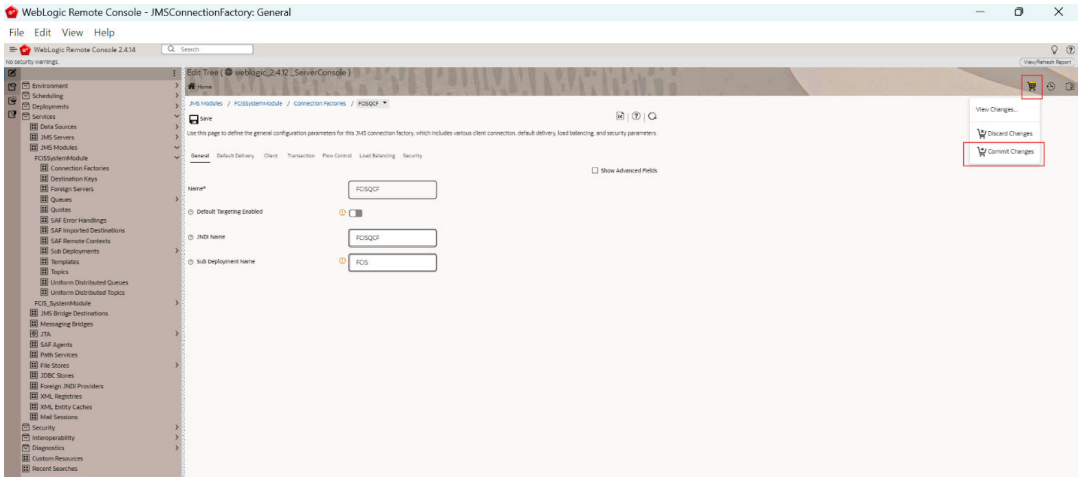
Figure 7-52 Connection Factories_General Tab



11. Click the **Cart** icon to view, commit or discard the changes and select **Commit Changes** to apply the changes.

    The `Changes were successfully committed` is displayed on committing the changes.

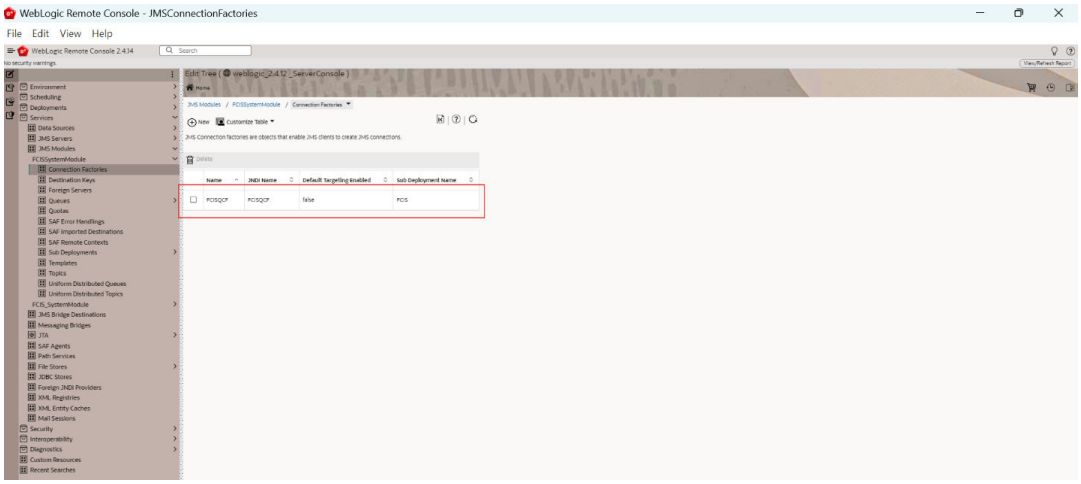
Figure 7-53 Commit Changes



12. You can view the new Connection Factory created in the **Connection Factory** screen.

    The new Connection Factory created is displayed.

Figure 7-54 New Connection Factory created



8
Configure Weblogic Server

This topic explains the steps for configuring Oracle WebLogic Application server for Oracle FLEXCUBE Investor Servicing.

To configure the Oracle WebLogic Application server, follow the steps given below:

1. Start the Administration Console of WebLogic Application server.

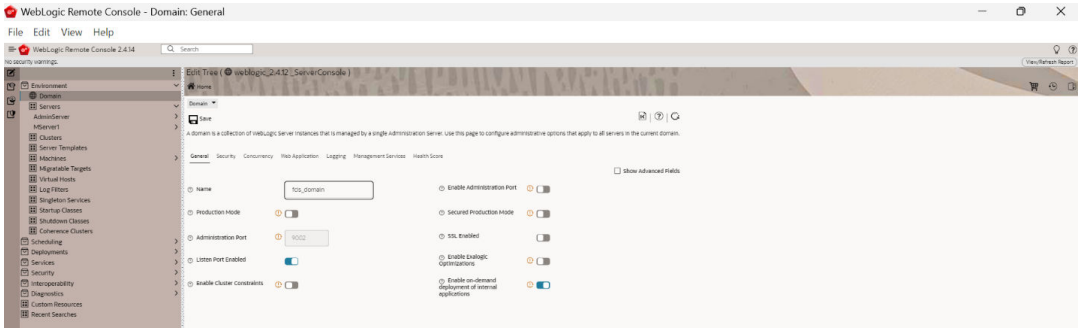
   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Select the **Domain** from the domain structure.

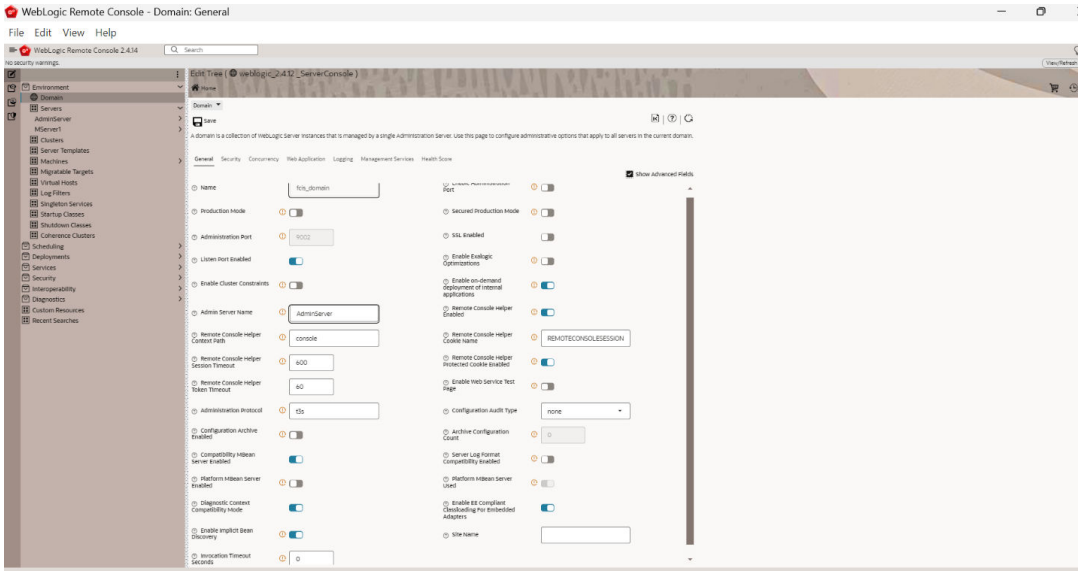
   Example: fcis_domain

Figure 8-1 Domain



3. Under **General Configuration** tab, select the options as shown below.

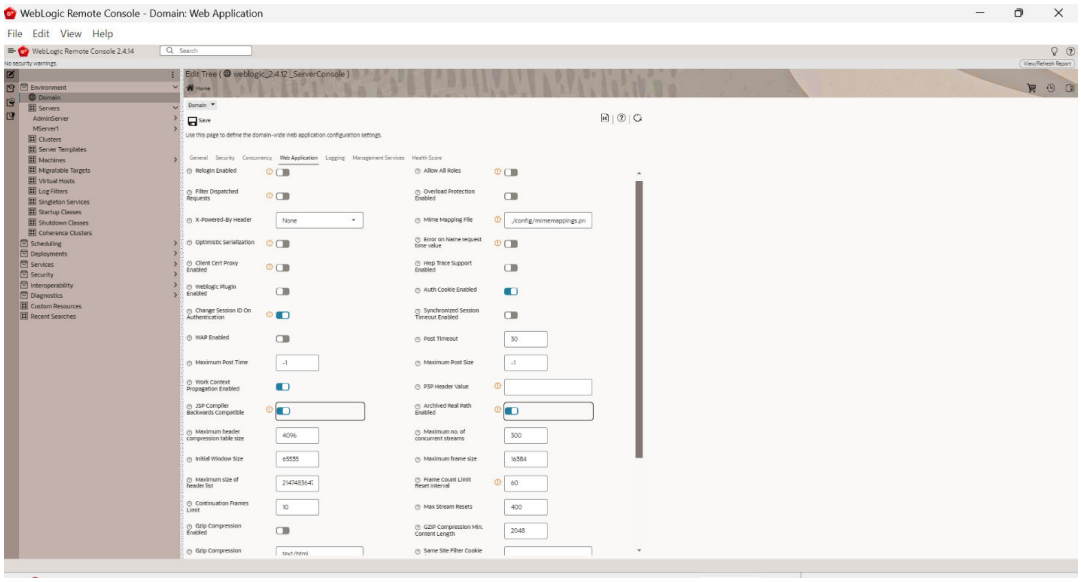
Figure 8-2 Domain_General



4. On the **Web Applications** tab, select the options **JSP Compiler Backwards Compatible**, **Archived Real Path Enabled**, and other options as shown in the figure.

   The following screen is displayed.

Figure 8-3 Domain_Web Applications Tab



5. Click **Save**.
6. Click the **Cart** icon to view, commit, or discard the changes.

   The following screen is displayed.

7. Select **Commit Changes** option to apply the changes.

   The `Changes were successfully committed` is displayed on committing the changes.

9

Setup/Configure Mail Session in WebLogic

This topic explains the steps to set up/configure mail sessions in Weblogic.

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

This topic contains the following sub-topics:

- Create JavaMail Session
  This topic explains the steps to create JavaMail Session.
- Configuration of the TLS/SSL Trust Store for Weblogic Server
  This topic explains the configuration of the TLS/SSL Trust Store for Weblogic Server.

9.1 Create JavaMail Session

This topic explains the steps to create JavaMail Session.

1. Start the Administration Console of WebLogic Application server.

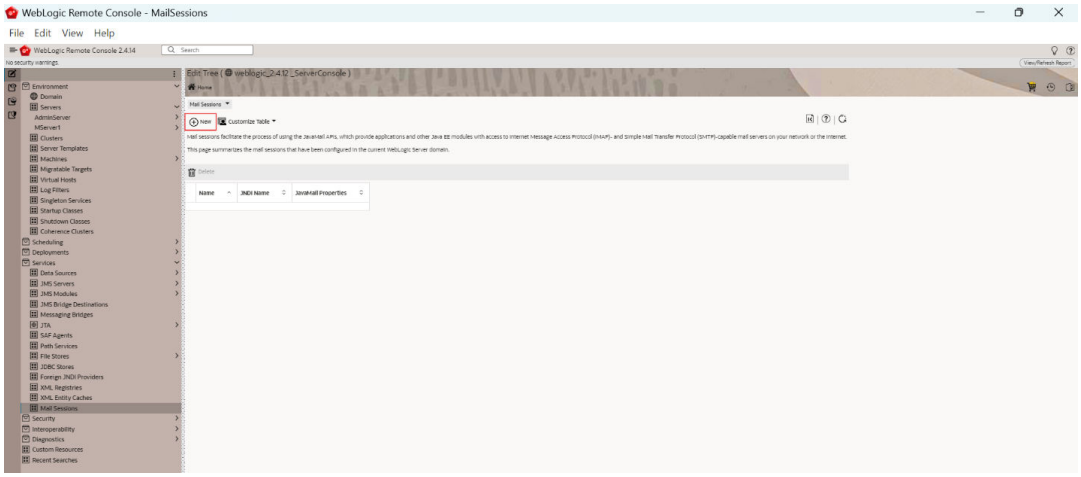
   You can start the Application by opening the Oracle `Weblogic Remote Console` application post installing the application from the link `https://github.com/oracle/weblogic-remote-console/releases`.

   The **WebLogic Remote Console** screen is displayed.

2. Go to **Services** and then select **Mail Sessions**.

   The **Mail Sessions** screen is displayed.

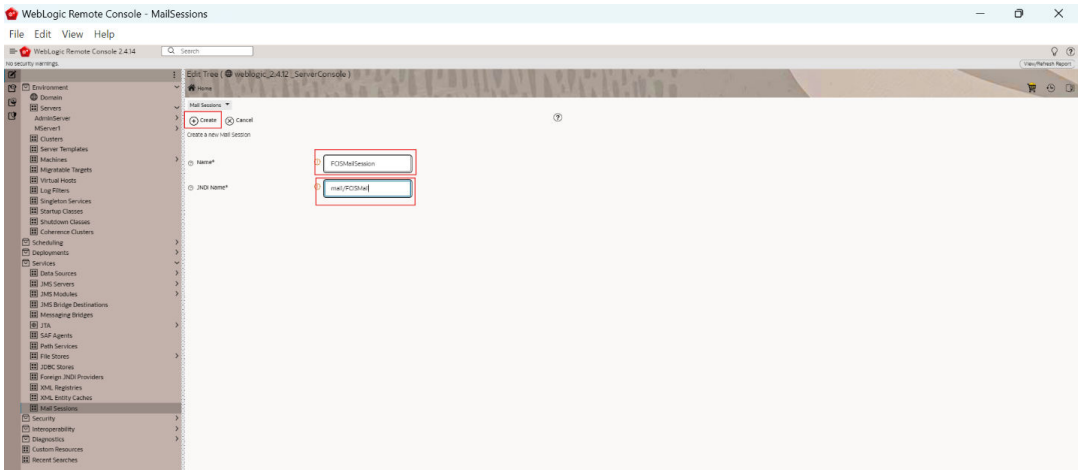
   Figure 9-1 Mail Sessions



3. Click **New** to create a new Mail Session.
4. Click **Create**.

   The following screen is displayed.

Figure 9-2 Create a New Mail Session



5. Specify the **Name** and **JNDI Name**.

   Examples:
   Name as FCISMailSession and JNDI Name as mail/FCISMail.

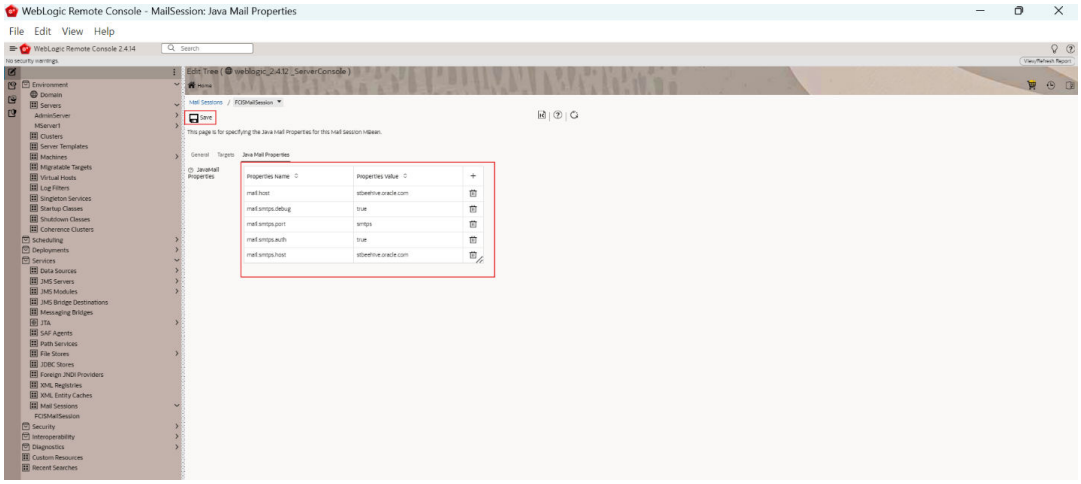
> **Note:**
>
> This JNDI name needs to be maintained in fcubs.properties file with encrypted format.

6. On the **Java Mail Properties** tab, specify the details and save.

   Java Mail Properties

```
mail.host=<HOST_MAIL_SERVER>

Eg: samplename

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
```

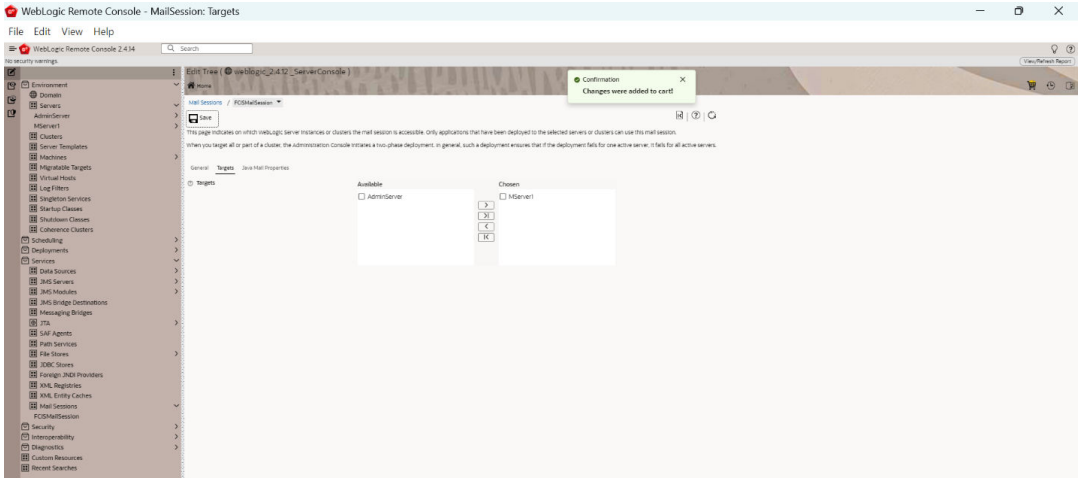
Figure 9-3 Save



7. Click the **Targets** tab.

   The following screen is displayed.

Figure 9-4 Targets Tab



8. Select the required servers and click **Save**.

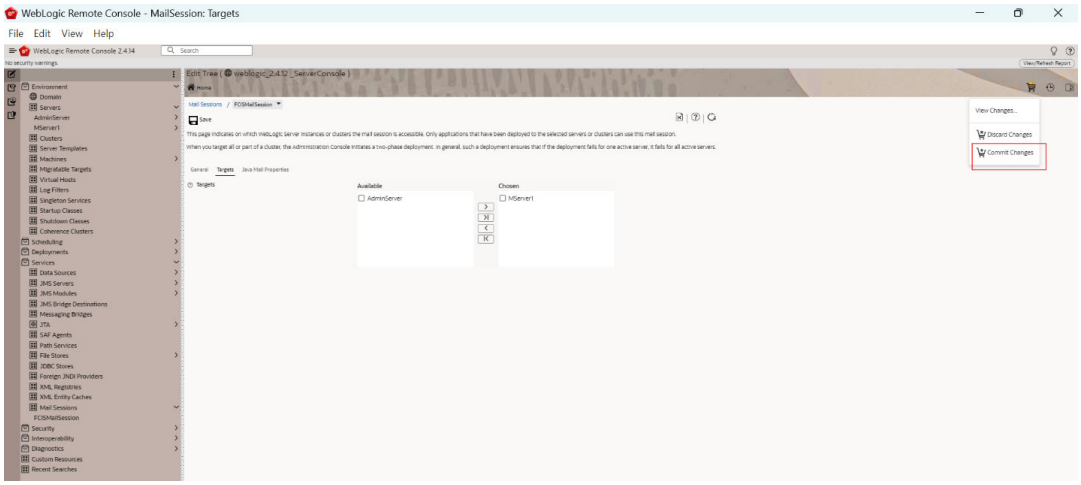
   `fcubs.properties` file needs to be updated with the encrypted values of:

   - SMTP_HOST
   - SMTP_USER
   - SMTP_PASSWORD
   - SMTP_JNDI

9. Click the **Cart** icon to view, commit, or discard the changes.

   The following screen is displayed.

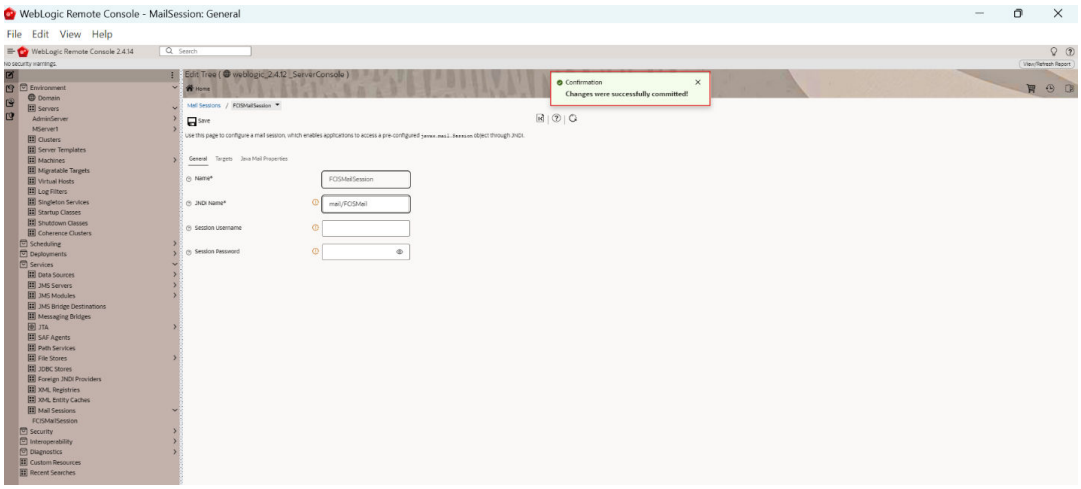
Figure 9-5 Committ Changes



10. Select **Commit Changes** option to apply the changes.

    The `Changes were successfully committed` is displayed on committing the changes.

Figure 9-6 Changes Committed Message



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

This topic explains the configuration of the TLS/SSL Trust Store for Weblogic Server.

As described in the previous topics, 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).