Oracle® Banking APIs

Installation Guide- Non-Linux Platforms

Patchset Release 22.2.3.0.0
F99655-02
June 2024

Oracle Banking APIs Installation Guide- Non-Linux Platforms, Patchset Release 22.2.3.0.0

F99655-02

Contents

Preface

Purpose v
Audience v
Documentation Accessibility v
Diversity and Inclusion v
Conventions v
Related Resources vi
Screenshot Disclaimer vi
Acronyms and Abbreviations vi

1 Manual OBAPI Installation

1.1 Policy Seeding 1-1

2 WEBLOGIC Setup and Configuration

2.1 Setting Domain JTA Transaction Timeout 2-1
2.2 Creating DIGX Data Source 2-2
2.3 Creating NONXA Data Source 2-5
2.4 Creating BATCH Data Source 2-8
2.5 Creating SYSCONFIG Data Source 2-11
2.6 Creating B1A1 Data Source 2-14
2.7 Create JMS Server and JMS Module 2-17
2.8 Creating WLS_JMS_AUDIT_PS FileStore 2-28
2.9 Creating AuditJMSServer JMS Server 2-28
2.10 Creating WLS_JMS_REPORT_PS FileStore 2-28
2.11 Creating ReportsJMSServer JMS Server 2-39
2.12 Creating jpa-cache JMS Server 2-40
2.13 Creating WLS_JPA_PS FileStore 2-40
2.14 Creating ExtSystemReceiver JMS Server - WLS_JMS_EXTSYSRECEIVER_PS FileStore 2-42
2.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS_JMS_EXTSYSSENDER_PS 2-43
2.16 Creating UBSForeignServer JMS Server 2-44

2.17 Creating OBPMForeignServer JMS Server 2-45

3 Deploying Applications

4 Configured jps-config.xml

5 List of Topics

Index

Preface

Purpose

This guide is designed to help acquaint you with the Oracle Banking APIs application. This guide provides answers to specific features and procedures that the user need to be aware of the module to function successfully.

Audience

This document is intended for the following audience:

- Customers
- Partners

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.

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.

Related Resources

For more information on any related features, refer to the following documents:

- Oracle Banking APIs Installation Manuals

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 in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
OBAPI	Oracle Banking APIs

1

Manual OBAPI Installation

OBAPI Database Installation with OBPM FLAVOR

Once obdx and ehms schema created in base installer, please proceed to below path for patchset scripts execution -

OBAPI_Installer/installables/OBAPI/<Installation type>/<version>/db/<version>/OBAPI/

Inside above path ddl, dml, and constraints folders are present inside which OBDX scripts will be present which needs to be executed manually.

If any place holder or variables that needs to be replaced manually before executing.

Similarly for other modules also you can find scripts those are to be executed in below path -

OBAPI_Installer/installables/OBAPI/<Installation type>/<version>/db/<version>/

Inside above path ddl, dml, and constraints folders are present inside which OBAPI scripts will be present which needs to be executed.

1.1 Policy Seeding

```
TEMP_PATH=Temparory Path
cp ${OBAPI INSTALLER}/installables/OBAPI/<Installation
        type>/<version>/policies/Entitlement_log4j.properties to
        TEMP_PATH/db/Entitlement_log4j.properties
cp ${OBAPI INSTALLER}/installables/OBAPI/<Installation type>/<version>
/policies /Task_log4j.properties to TEMP_PATH/db/Task_log4j.propertiescp $
{OBAPI INSTALLER}/installables/OBAPI/
<Installation type>/<version>/policies /Dashboard_seed_log4j.properties to
TEMP_PATH/db/Dashboard_seed_log4j.propertiesupdate
<logs_path> in the above file (TEMP_PATH) to desired location.
```

update <logs_path> in the above file (TEMP_PATH) to desired location.

Execute below command in sequence.

```
Were SCHEMA_NAME=OBAPI_${POST_FIX} and SCHEMA_PASS= Password of
        OBAPI_${POST_FIX} .


# $JAVA_HOME/bin/java -Djava.util.logging.config.file=
        TEMP_PATH/db/Task_log4j.properties -jar ${OBAPI INSTALLER}/OBAPI/
<Installation
        type>/<version>/policies/com.ofss.digx.utils.feed.data.task.jar /
installables/policies/Task.csv
        oracle.jdbc.OracleDriver SCHEMA_NAME SCHEMA_PASS
```

```
        'jdbc:oracle:thin:@OBAPI_DATABASE_HOSTNAME:OBAPI_DATABASE_PORT/
OBAPI_DATABASE_SID'


# $JAVA_HOME/bin/java -Djava.util.logging.config.file=
        TEMP_PATH/db/Dashboard_seed_log4j.properties -jar ${OBAPI INSTALLER}/
OBAPI/<Installation
        type>/<version>/policies/com.ofss.digx.utils.dashboard.jar ${OBAPI
        INSTALLER}/}/OBAPI/<Installation type>/<version>/policies/
dashboard_json/
        oracle.jdbc.OracleDriver SCHEMA_NAME SCHEMA_PASS
        'jdbc:oracle:thin:@OBAPI_DATABASE_HOSTNAME:OBAPI_DATABASE_PORT/
OBAPI_DATABASE_SID'


# $JAVA_HOME/bin/java -Djava.util.logging.config.file=
        TEMP_PATH/db/Entitlement_log4j.properties -jar ${OBAPI INSTALLER}/}/
OBAPI/<Installation
          type>/<version>/policies/
com.ofss.digx.utils.entitlement.feed.data.jar  ${OBAPI INSTALLER}/}/OBAPI/
<Installation
        type>/<version>/policies/Resources.csv ${OBAPI INSTALLER}/}/OBAPI/
<Installation
        type>/<version>/policies/Entitlement.csv ${OBAPI INSTALLER}/}/OBAPI/
<Installation
        type>/<version>/policies/Day0Policy.csv KERNEL
oracle.jdbc.OracleDriver SCHEMA_NAME SCHEMA_PASS
        'jdbc:oracle:thin:@OBAPI_DATABASE_HOSTNAME:OBAPI_DATABASE_PORT/
OBAPI_DATABASE_SID'
```

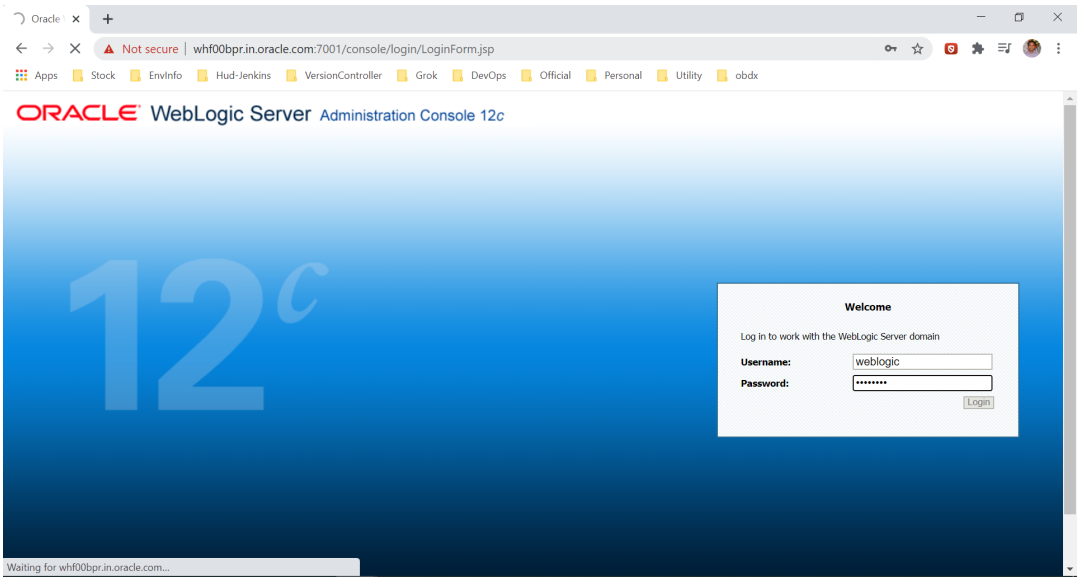
2

WEBLOGIC Setup and Configuration

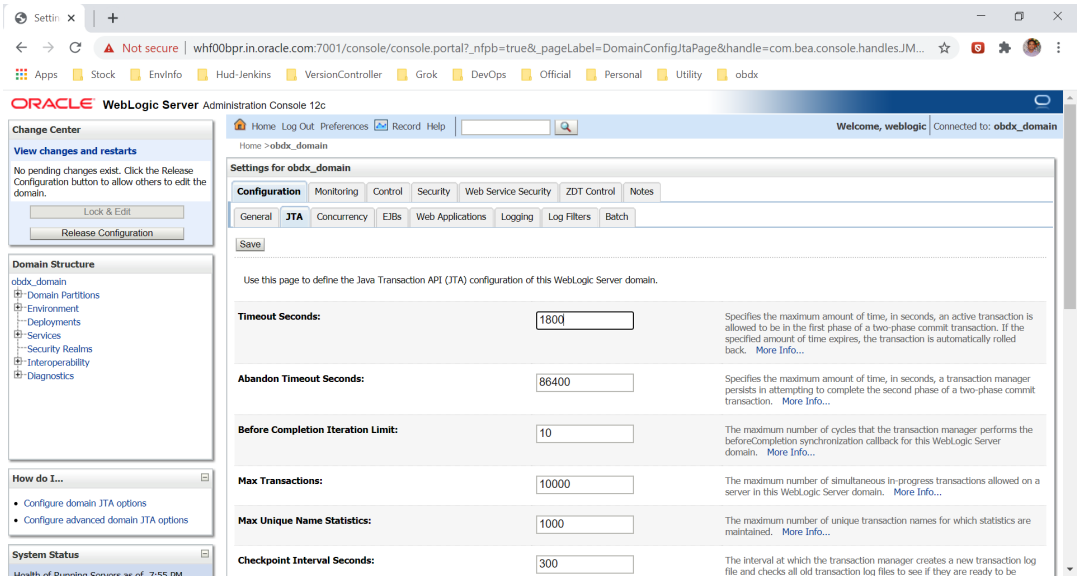
Once OBAPI and EHMS schema created, weblogic domain created, managed server, cluster and node manager configured, proceed with below steps.

2.1 Setting Domain JTA Transaction Timeout

1. Loging into weblogic domain with admin credentials (ex. weblogic).



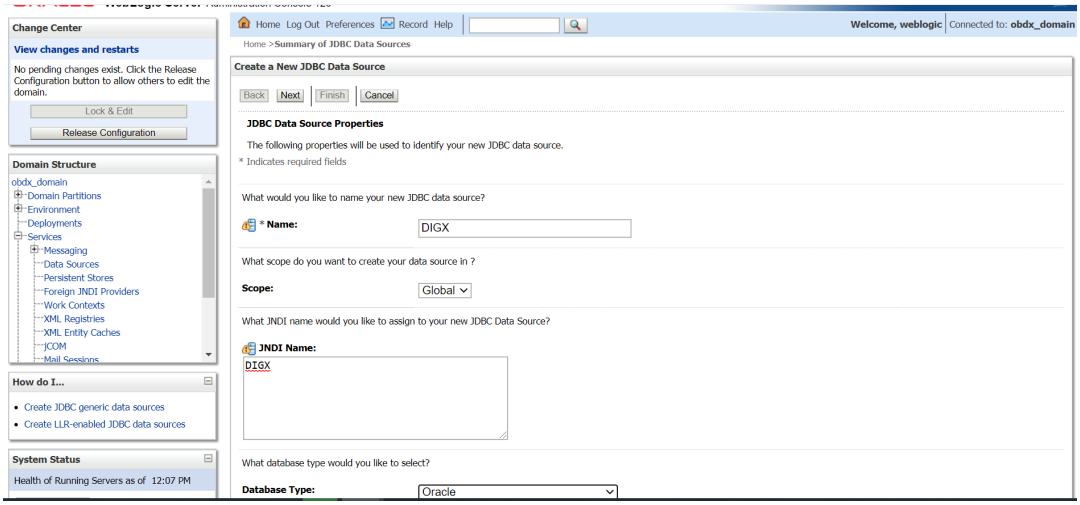
2. Click on `DOMAIN_NAME` → `JTA` → set Timeout Seconds to 1800 → click **Save** → Activate changes.



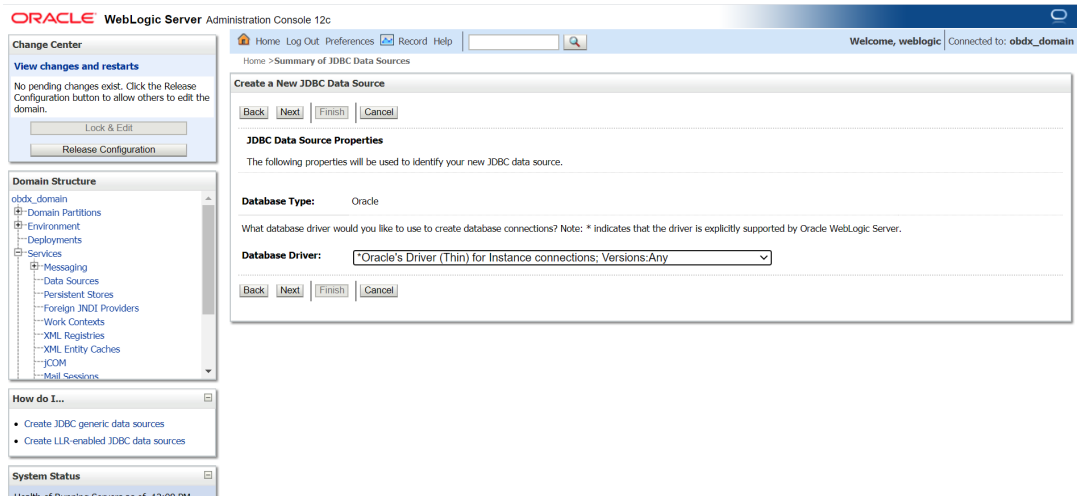
Verify once if below datasources are already created post 22.2.0.0.0 base installation and if present proceed to JMS Server and JMS Module creation, if not created proceed with below steps.

2.2 Creating DIGX Data Source

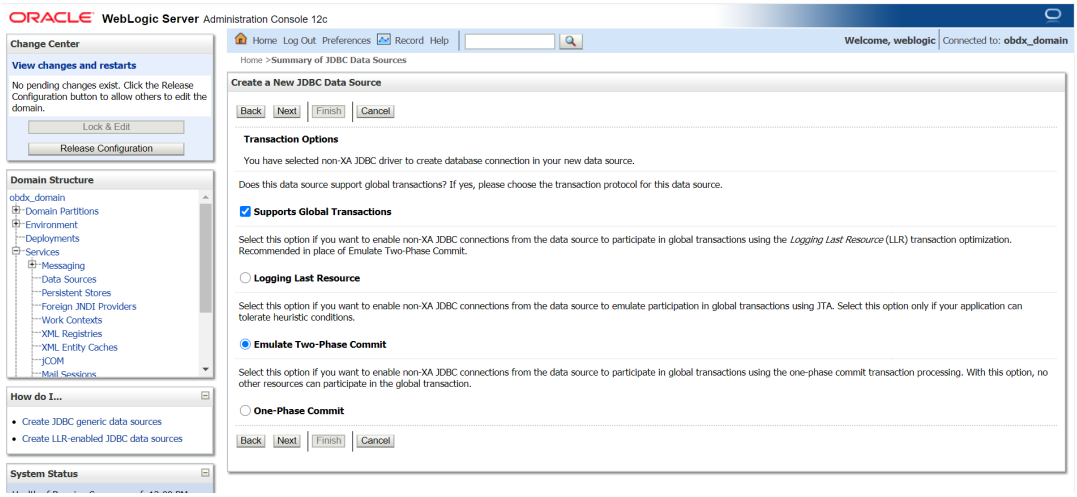
1. Navigate to Data Source → click **New** → Provide details and click **Finish**.



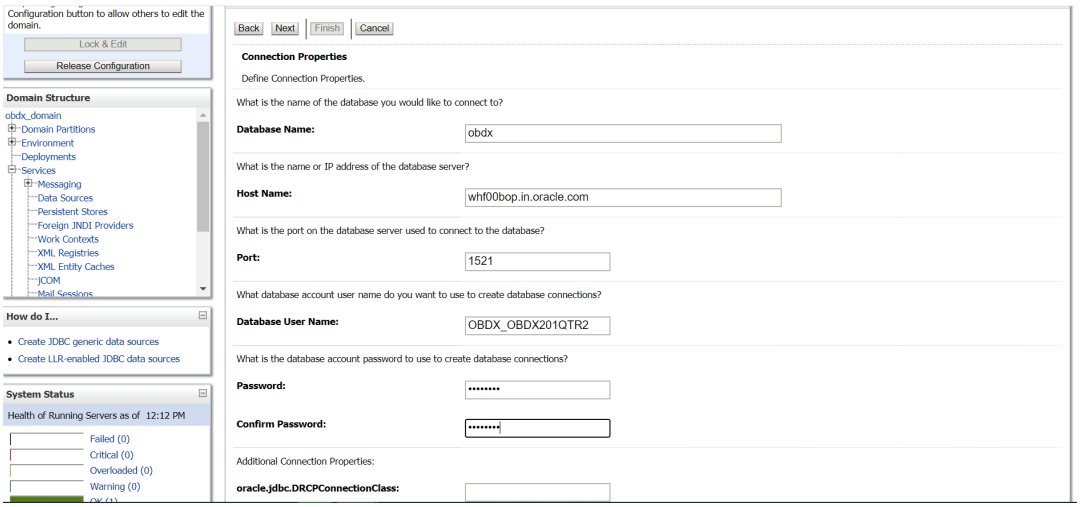
2. **Name:** `DIGX`
JNDI Name: `DIGX`



3. Select Oracle's Driver (Thin) for Instance connections;



4. Select Emulate Two-Phase Commit.

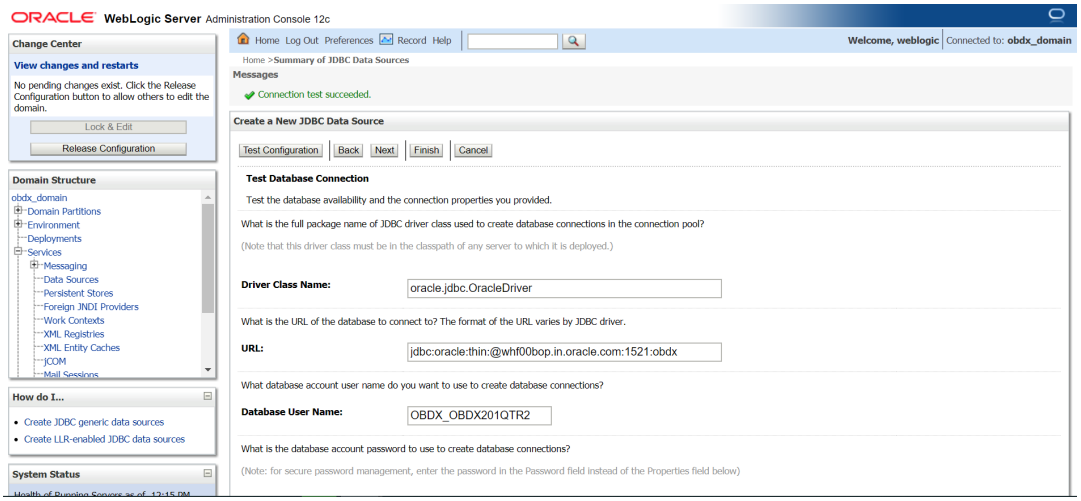


5. **Provide**
Database Name: Database SID

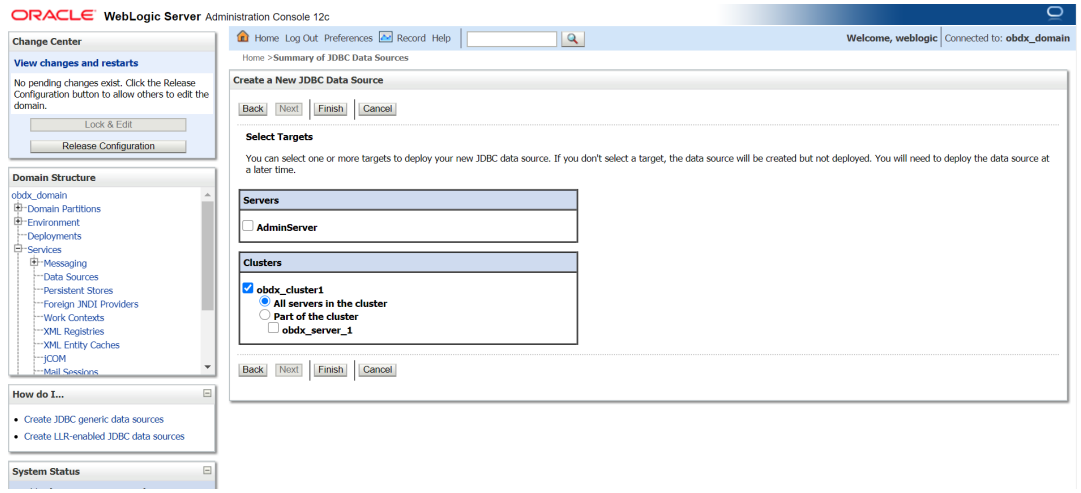
Host Name: Database hostname

Port: Database port Number

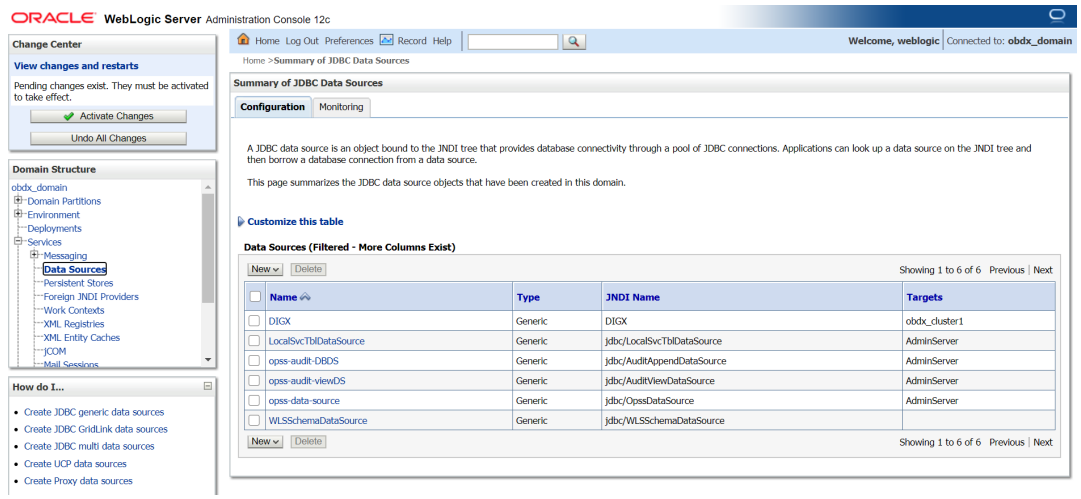
Database user Name: `OBAPI_${POST_FIX}`



6. Test Configuration.

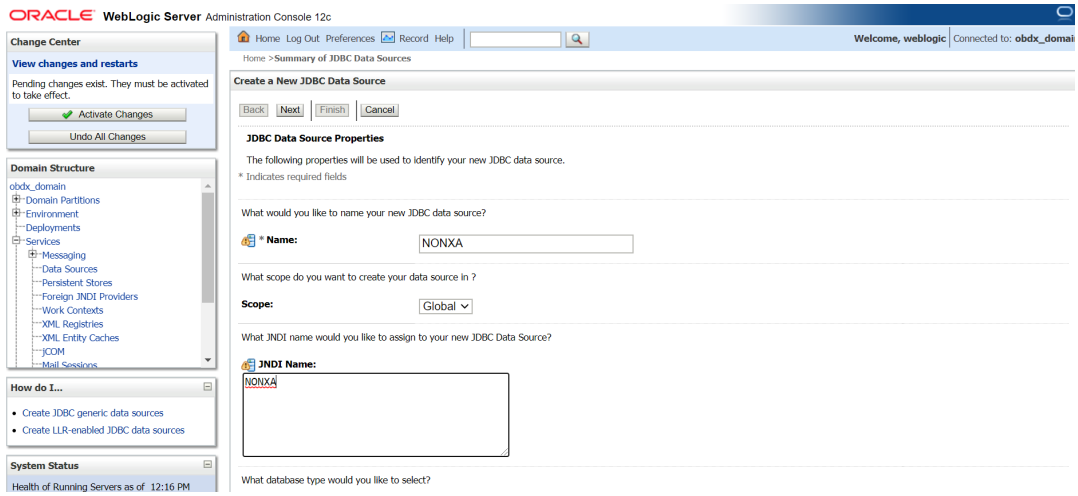


7. Target to cluster.

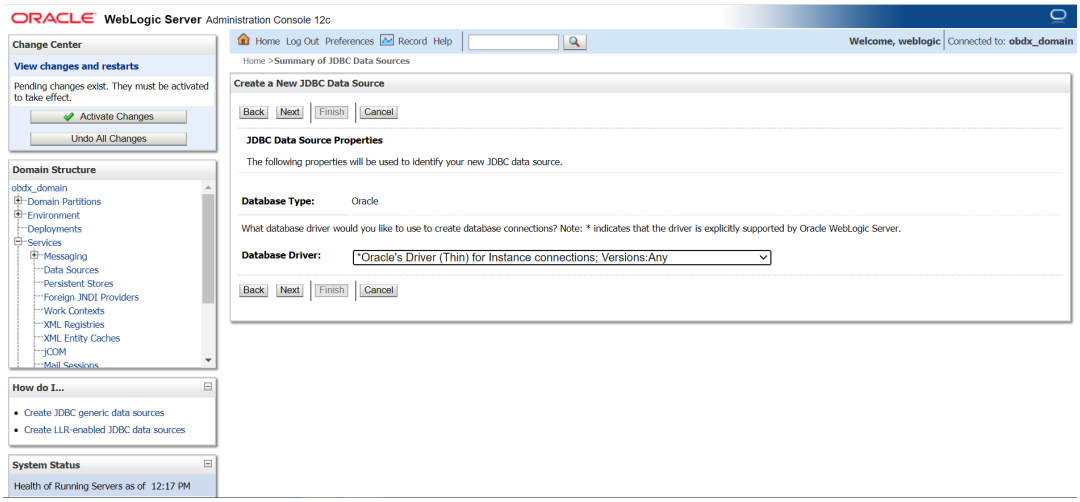


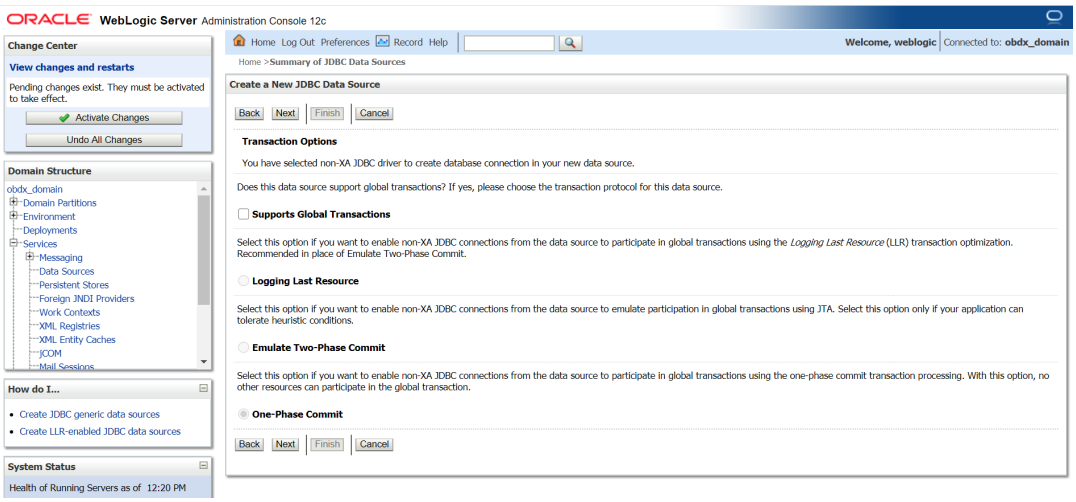
2.3 Creating NONXA Data Source

1. Navigate to Data Source → click **New** → Provide details and click **Finish**.

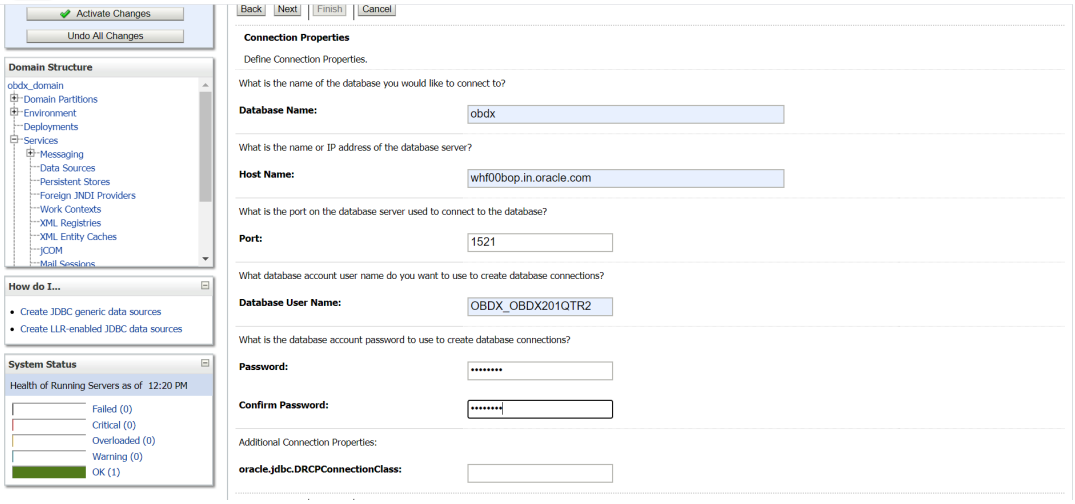


2. **Name :** `NONXA`
JNDI Name : `NONXA`





3. Click **Next**.



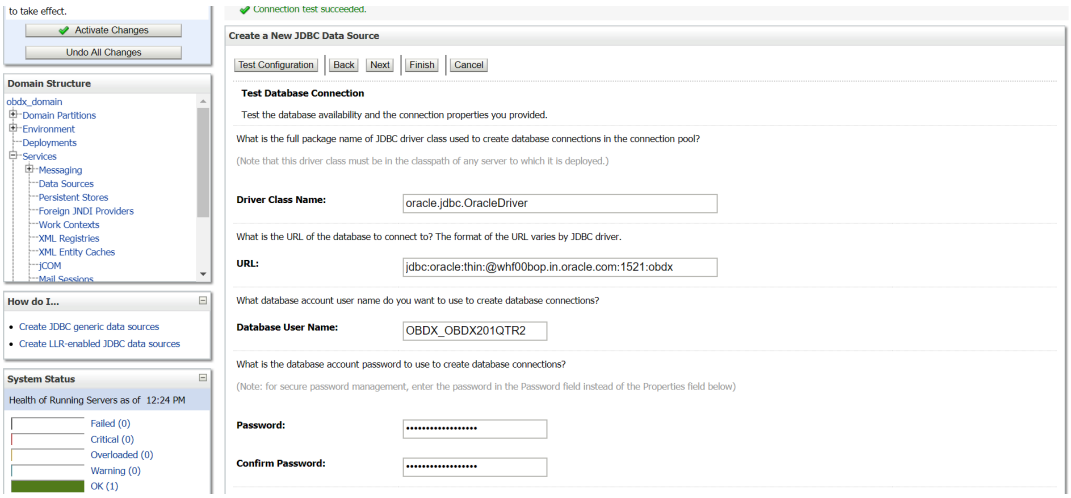
4. **Provide**
   Database Name: Database SID

   Host Name: Database hostname

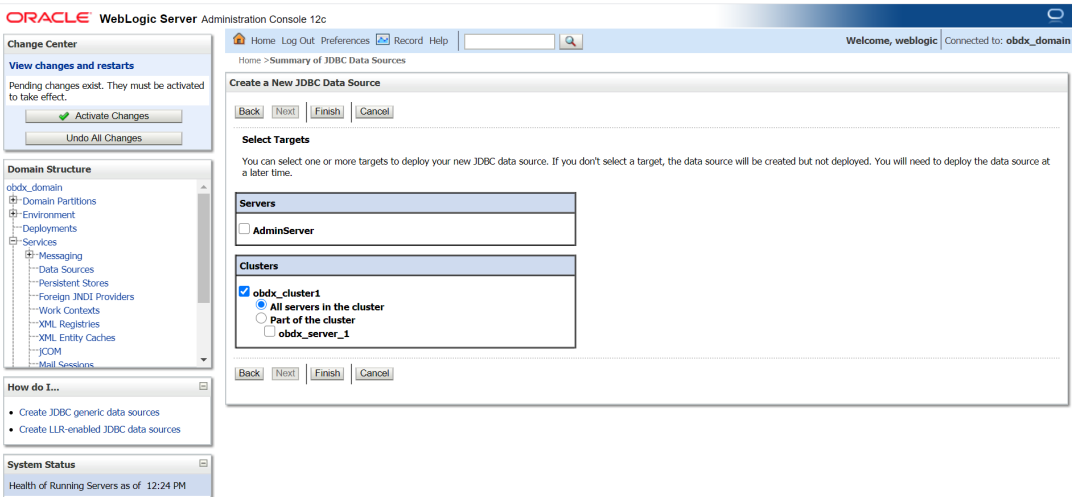
   Port: Database port Number

   Database user Name: `OBAPI_${POST_FIX}`

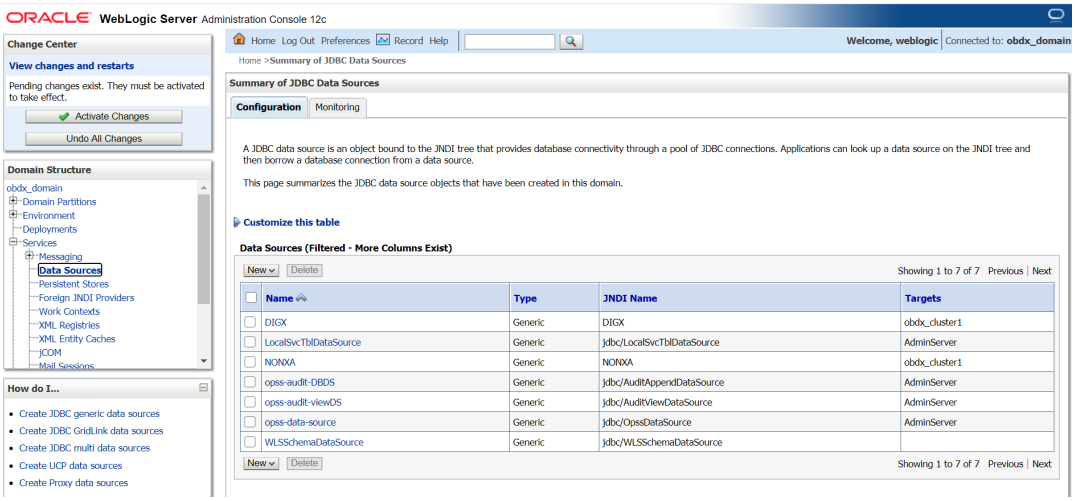
   Password: Database user password



5. Test Configuration.

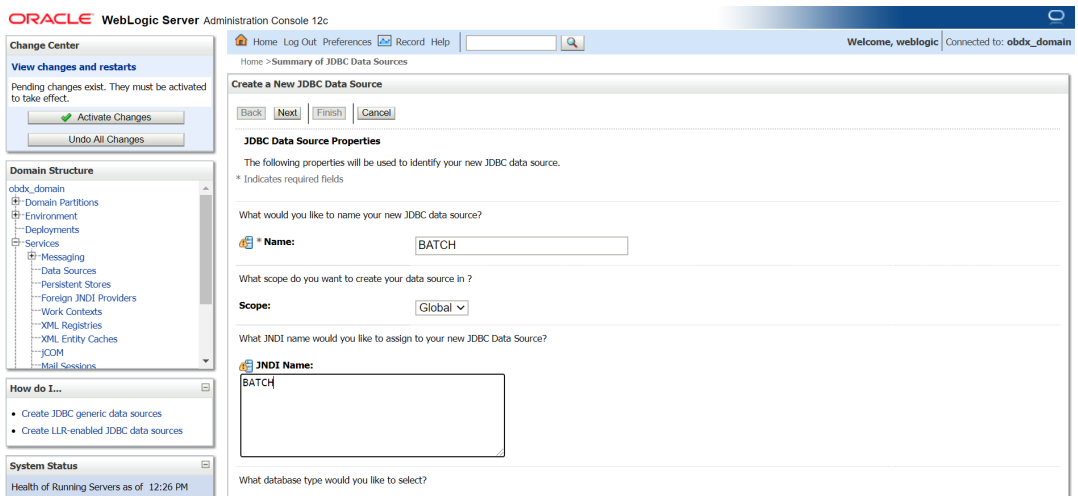


6. Select target as cluster → **Finish**.

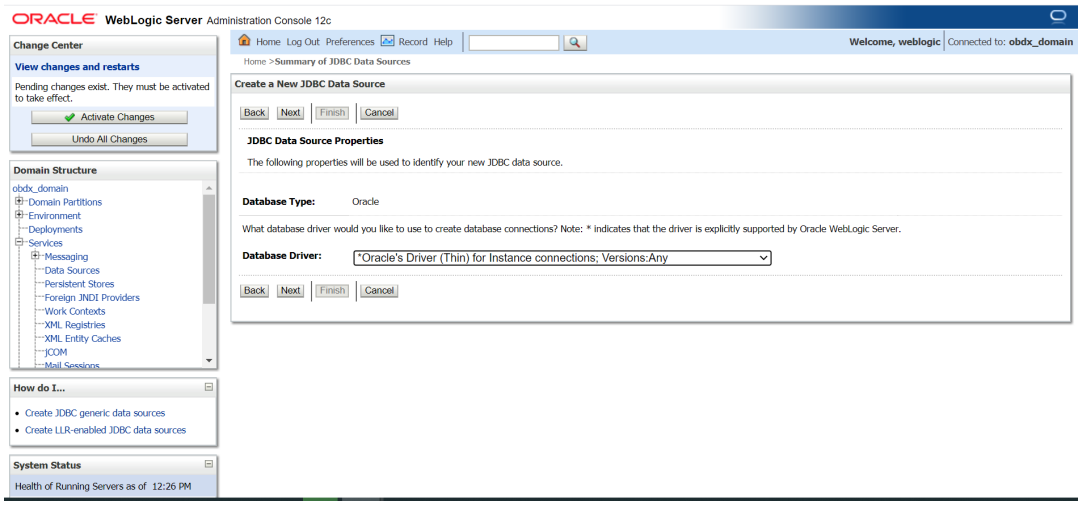


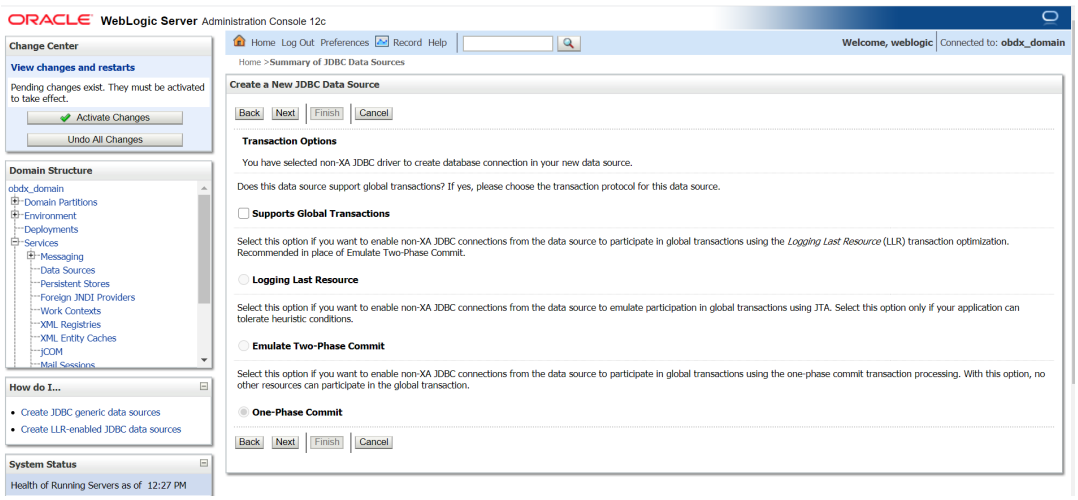
2.4 Creating BATCH Data Source

1.

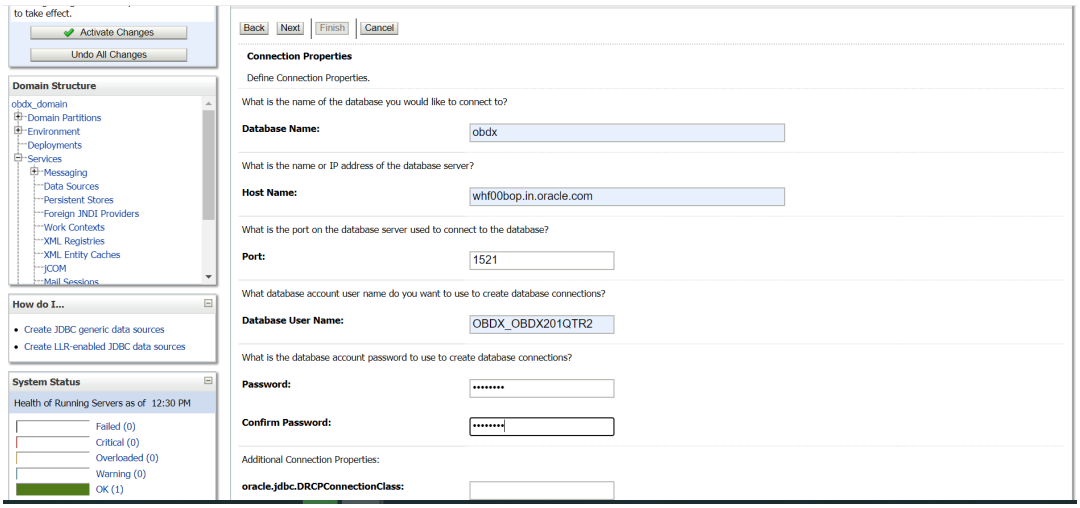


2. **Name :** BATCH
JNDI Name : BATCH





3. Click **Next**.



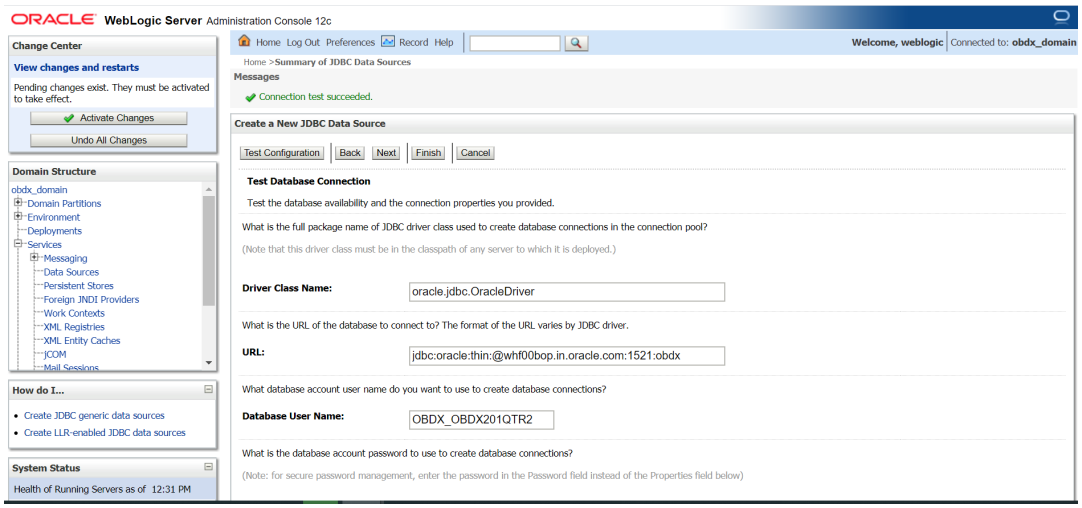
4. **Provide**
Database Name: Database SID

Host Name: Database hostname

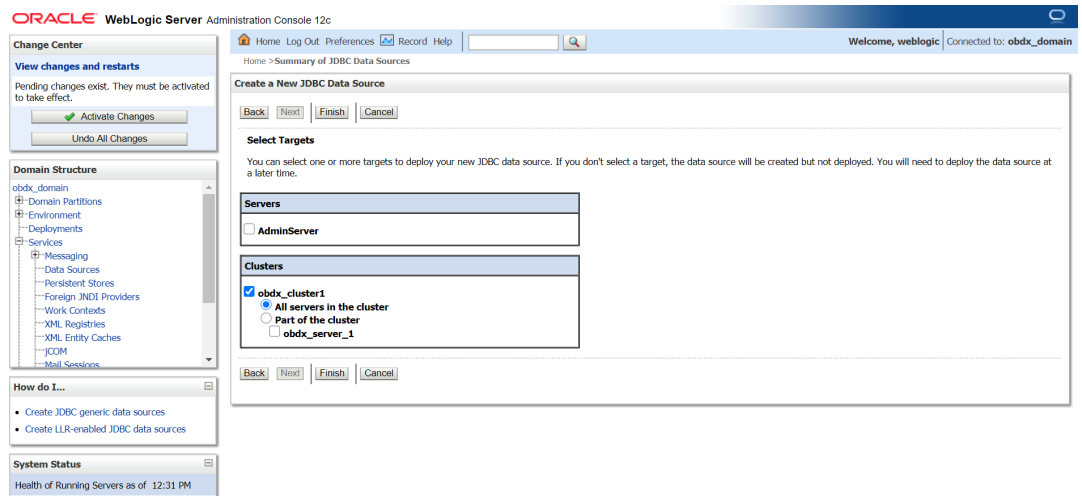
Port: Database port Number

Database user Name: `OBDX_${POST_FIX}`

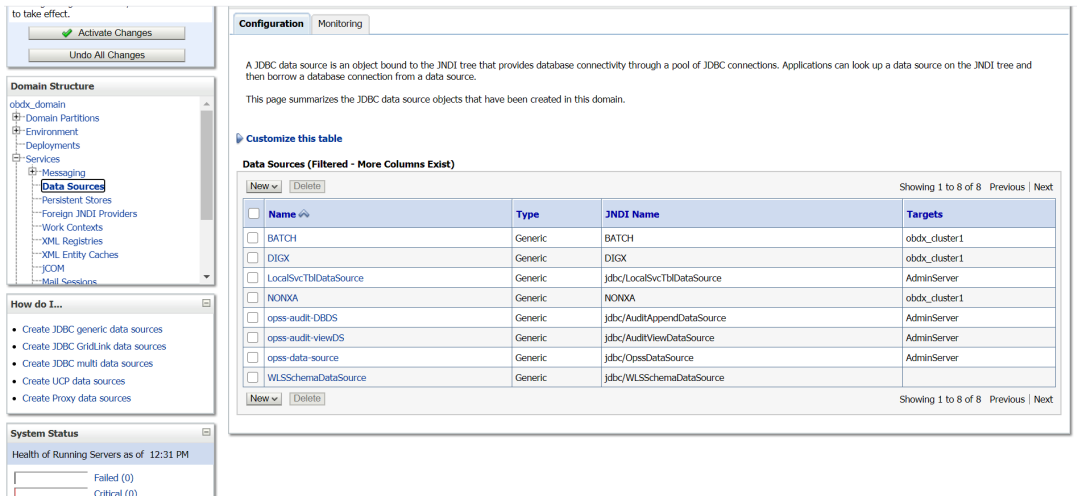
Password: Database user password



5. Test Configuration.

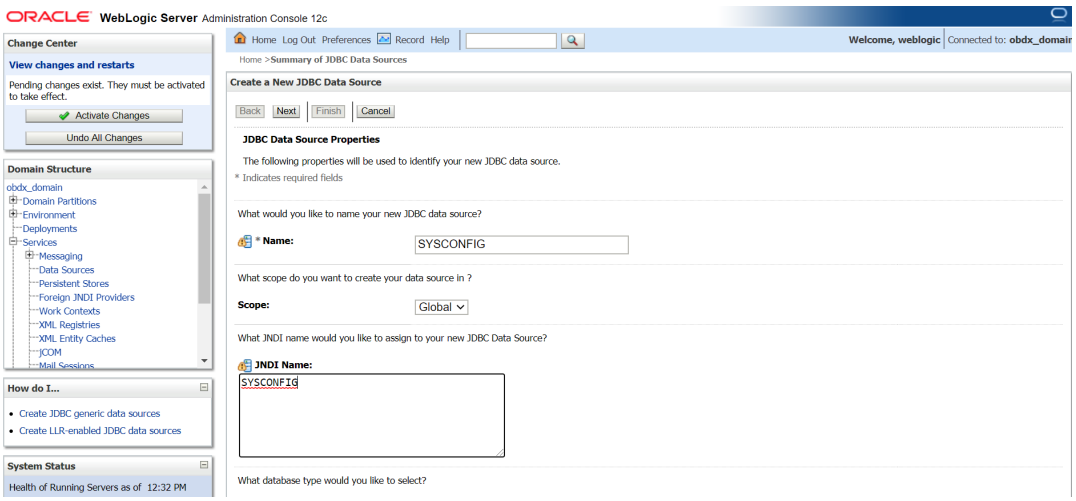


6. Target Cluster and click **Finish**.

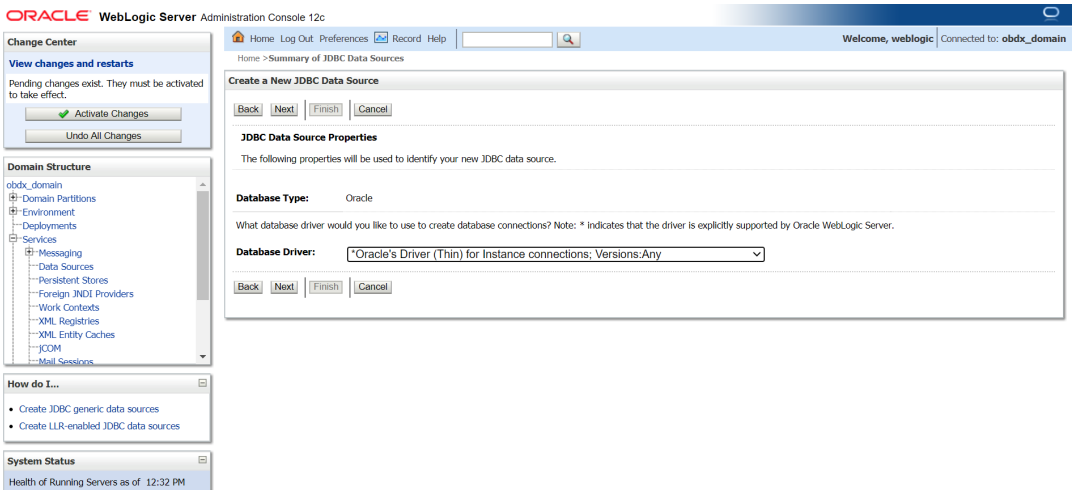


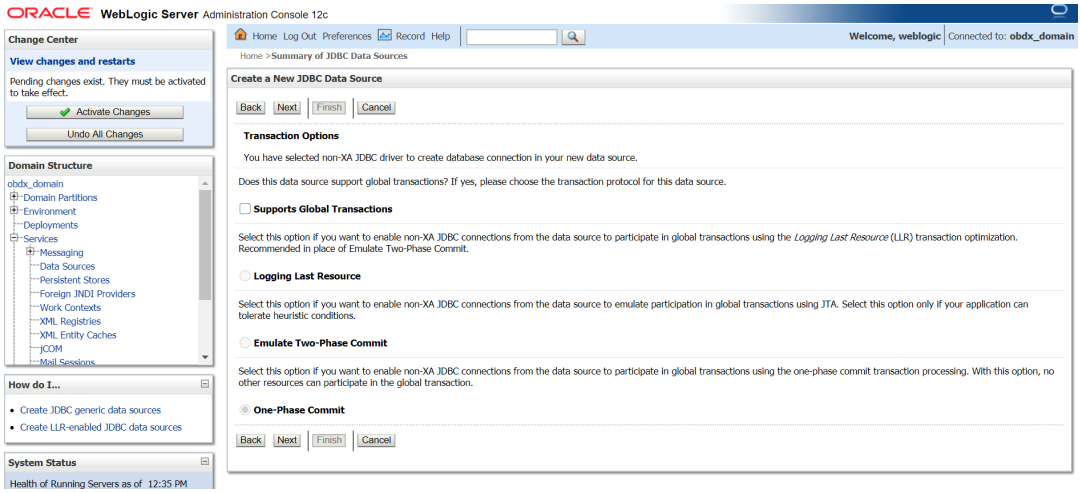
2.5 Creating SYSCONFIG Data Source

1.

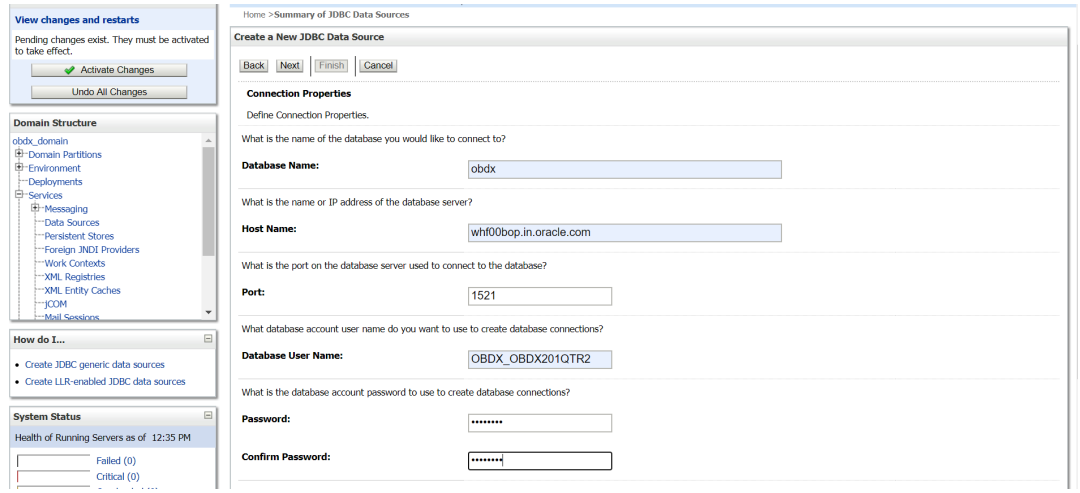


2. **Name :** SYSCONFIG
JNDI Name : SYSCONFIG





3. Click **Next**.



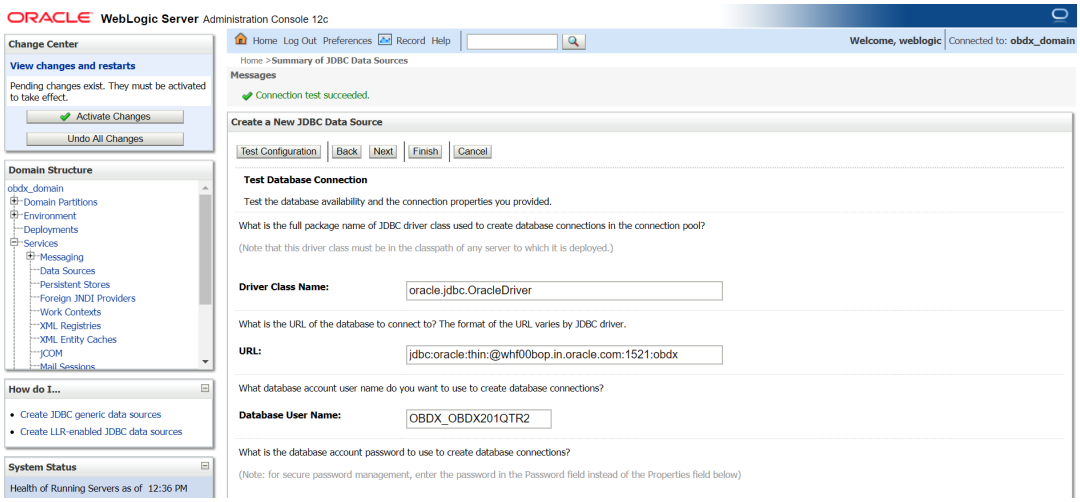
4. **Provide**
   Database Name: Database SID

   Host Name: Database hostname

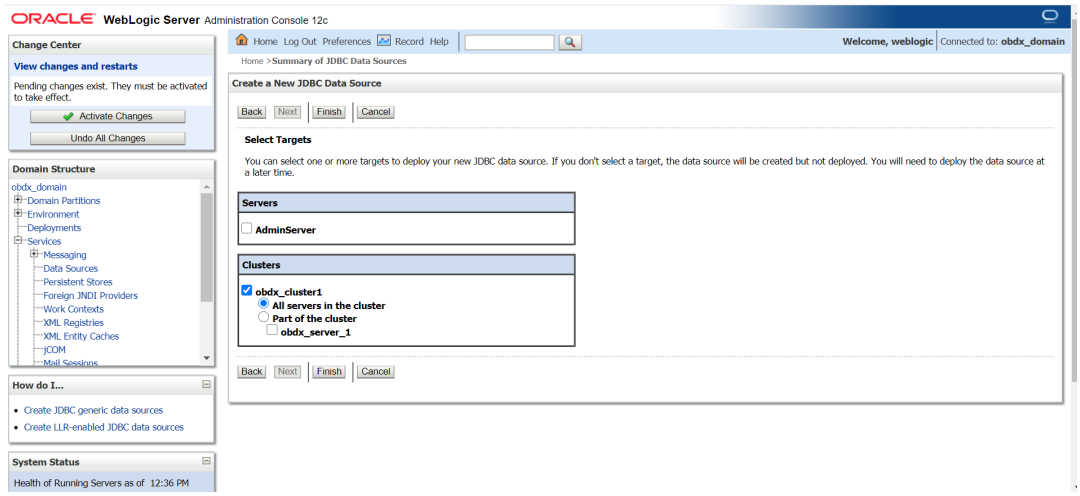
   Port: Database port Number

   Database user Name: `OBDX_${POST_FIX}`

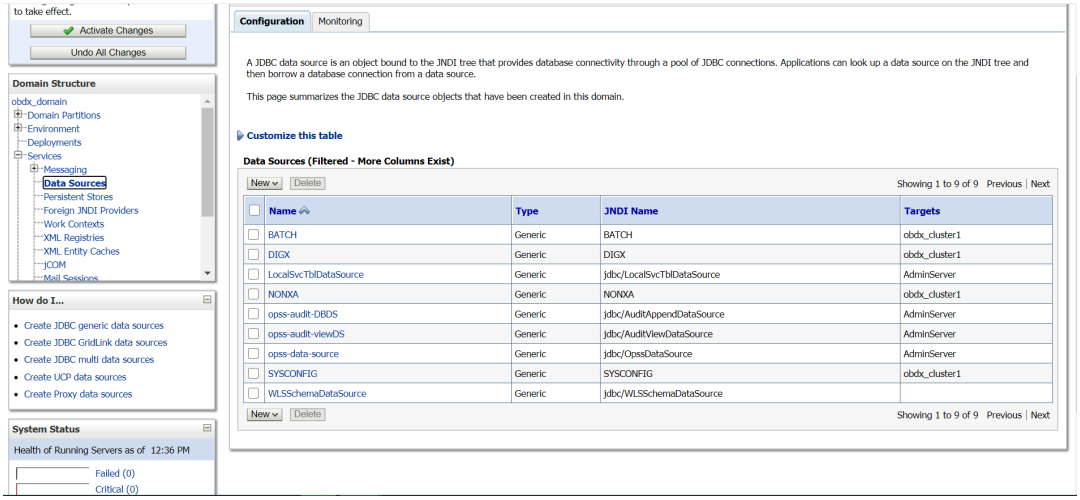
   Password: Database user password



5. Test Configuration.

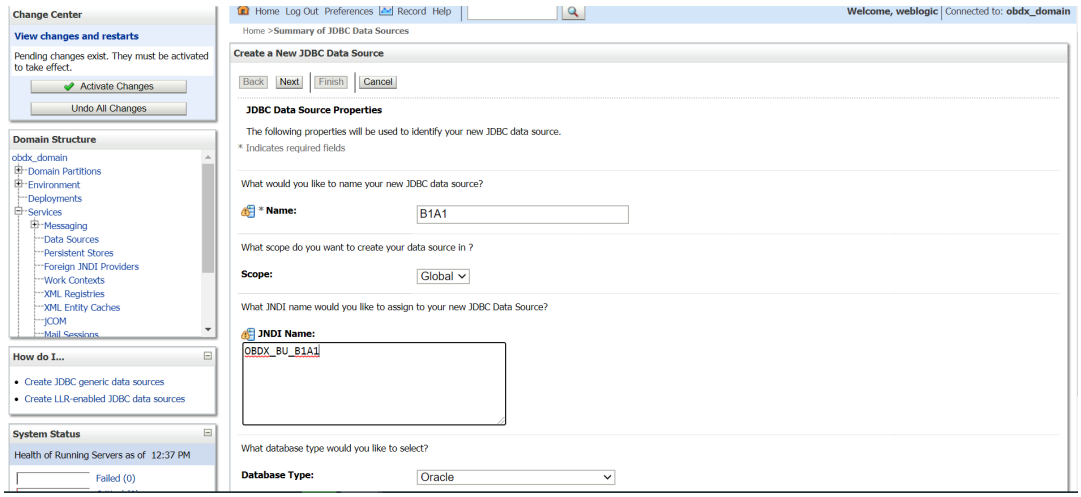


6. Select target as cluster and click **Finish**.

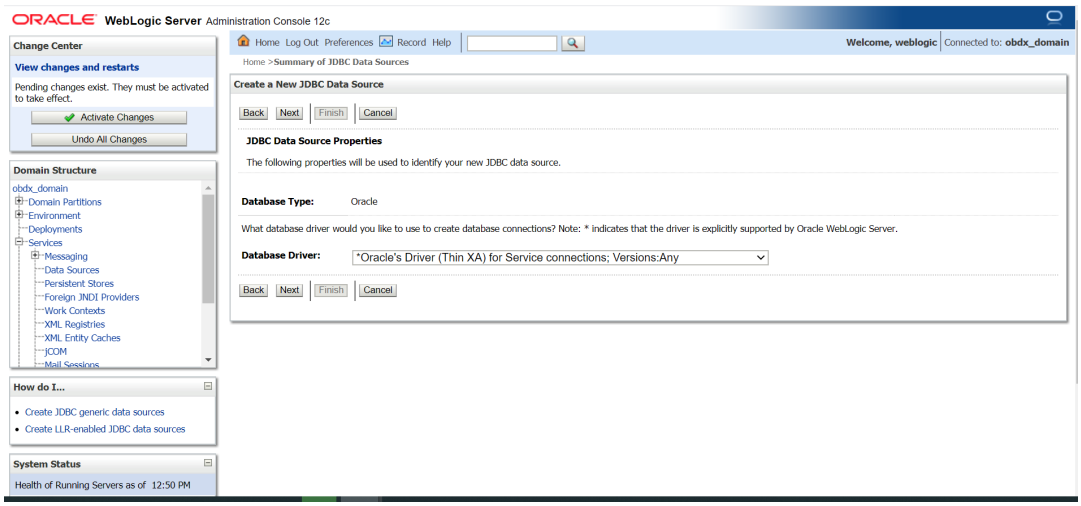


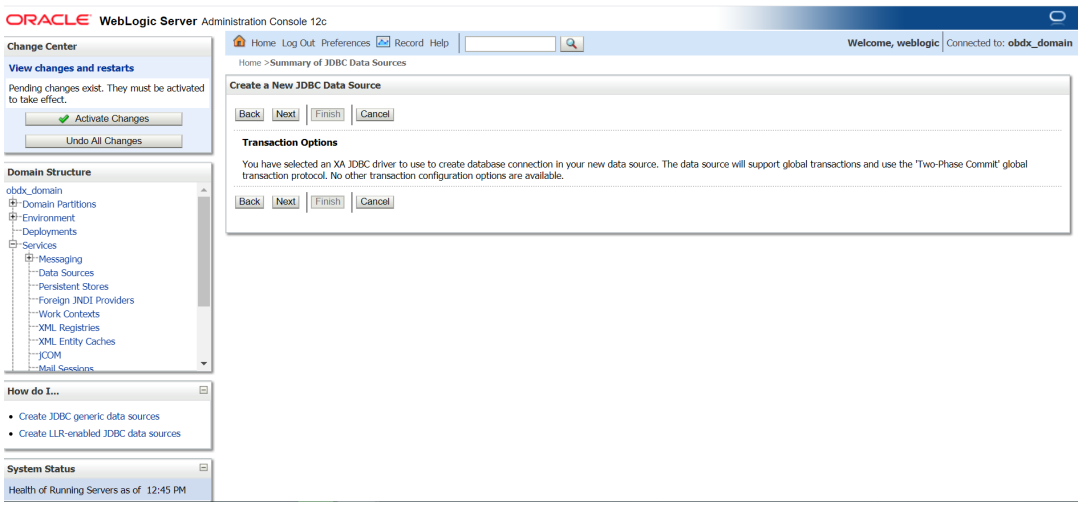
2.6 Creating B1A1 Data Source

1.

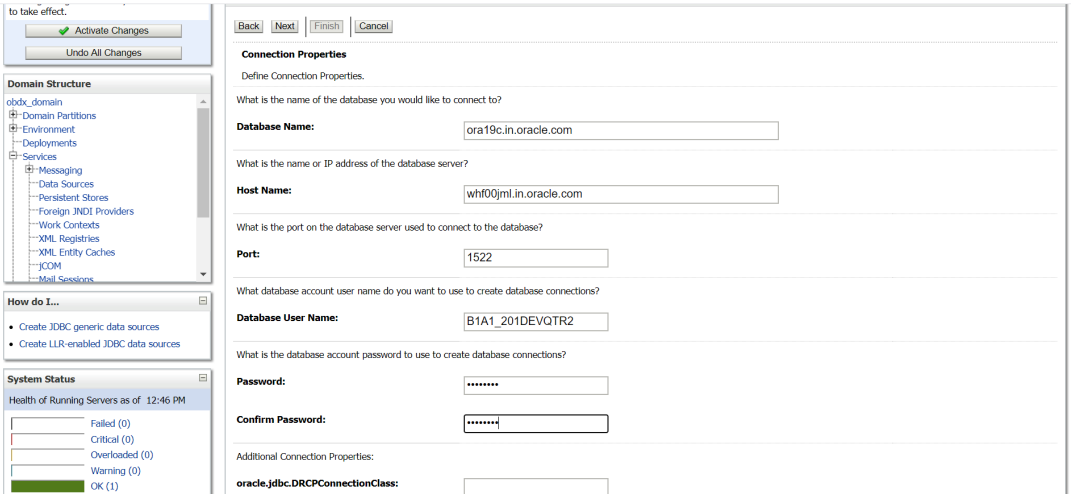


2. **Name:** `B1A1`
JNDI Name : `OBDX_BU_B1A1`





3. Click **Next**.



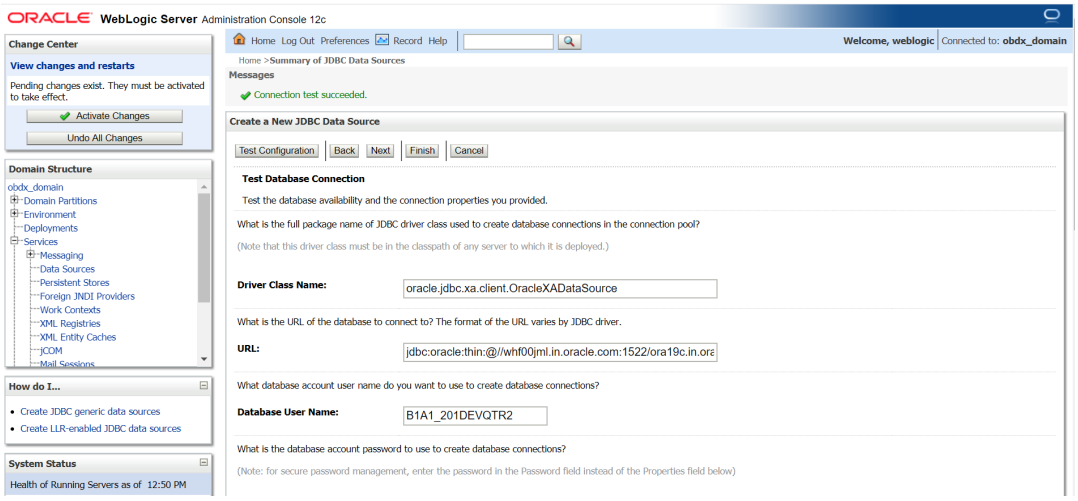
4. **Provide**
Database Name: Database SID ($EHMS_DATABASE_SID)

Host Name: Database hostname (`$EHMS_DATABASE_HOSTNAME`)

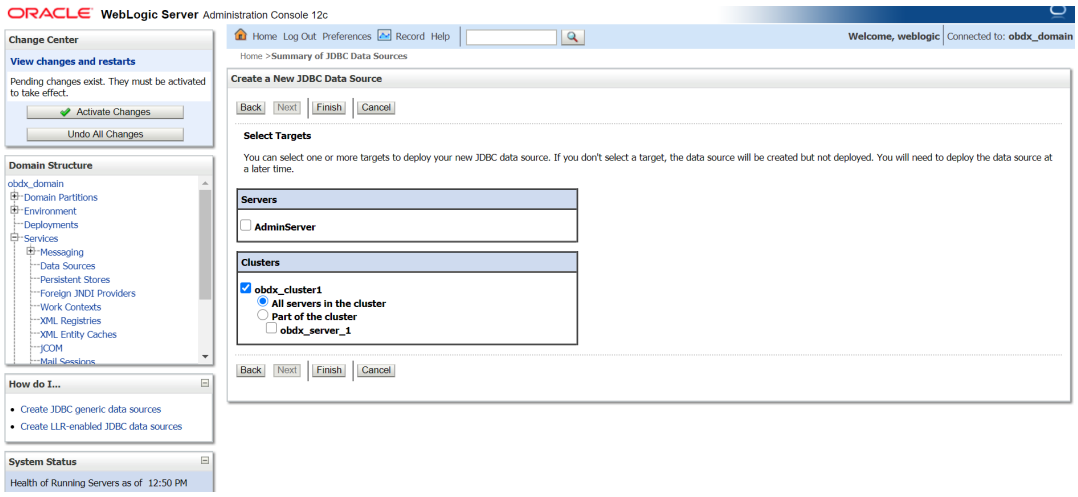
Port: Database port Number (`$EHMS_DATABASE_PORT`)

Database user Name: `${ EHMS_SCHEMA_NAME }`

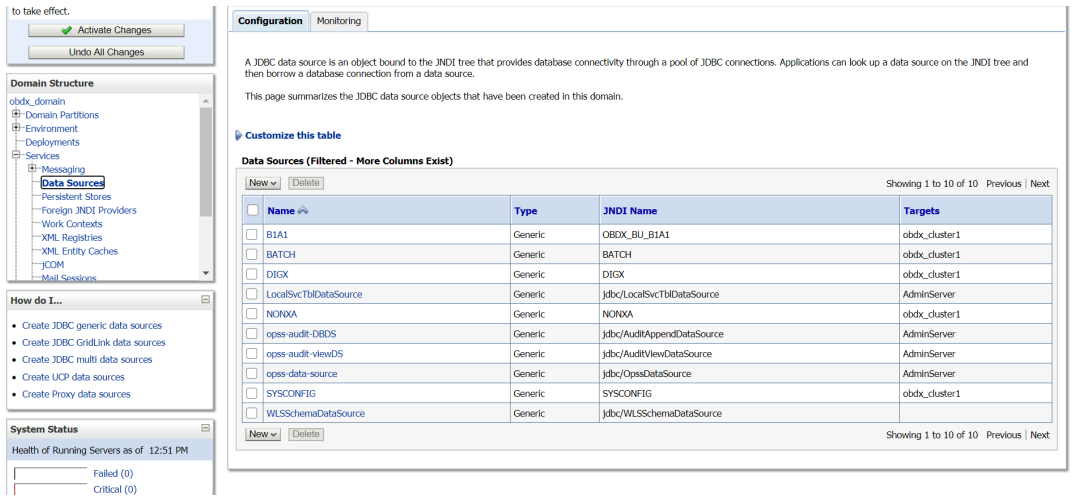
Password: Database user `${ EHMS_SCHEMA_NAME }` password



5. Test Configuration.



6. Set target as cluster and click **Finish**.

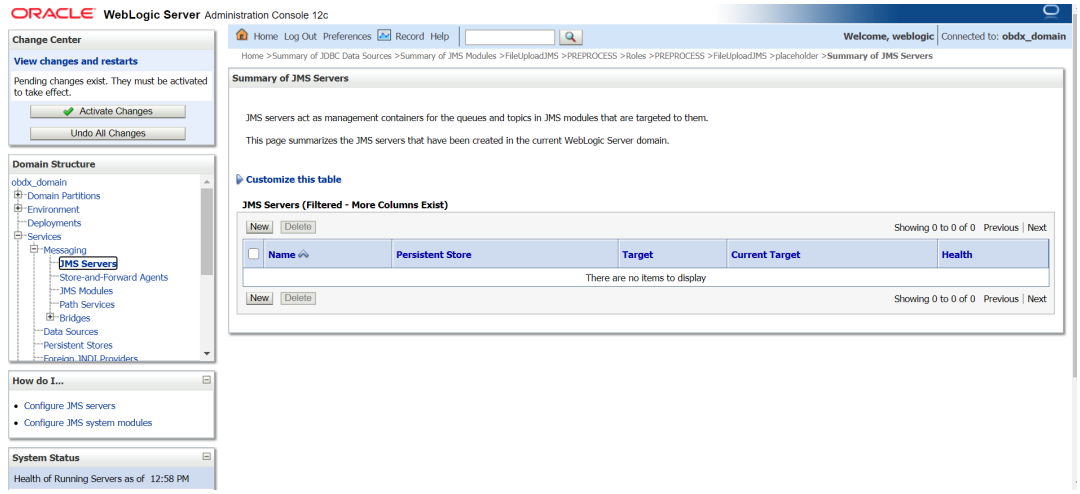


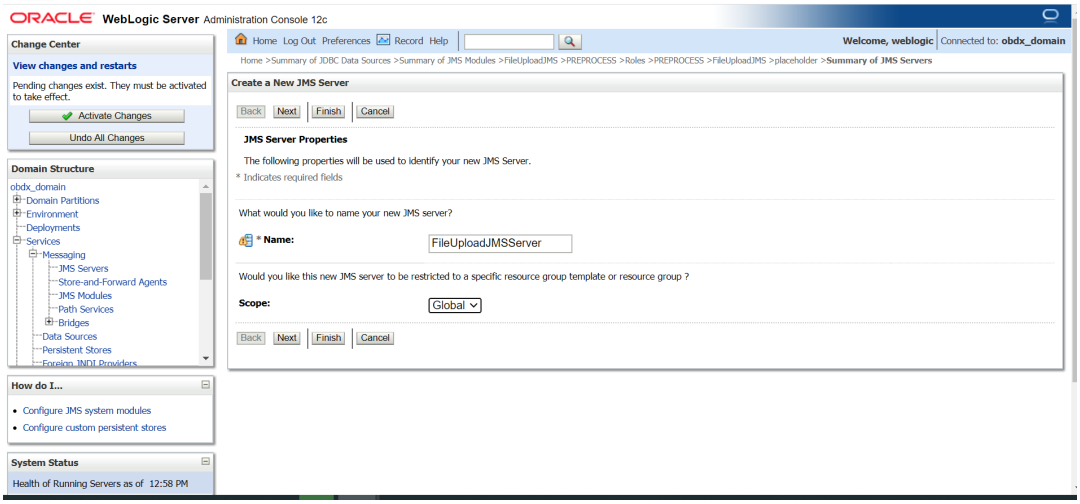
Before starting with below step please verify if below mentioned JMS Servers and Modules are present, if not please refer to jms.xml file present in path - OBAPI_Installer\installables\OBAPI\<Installation type>\<version>/ config/xml/jms

2.7 Create JMS Server and JMS Module

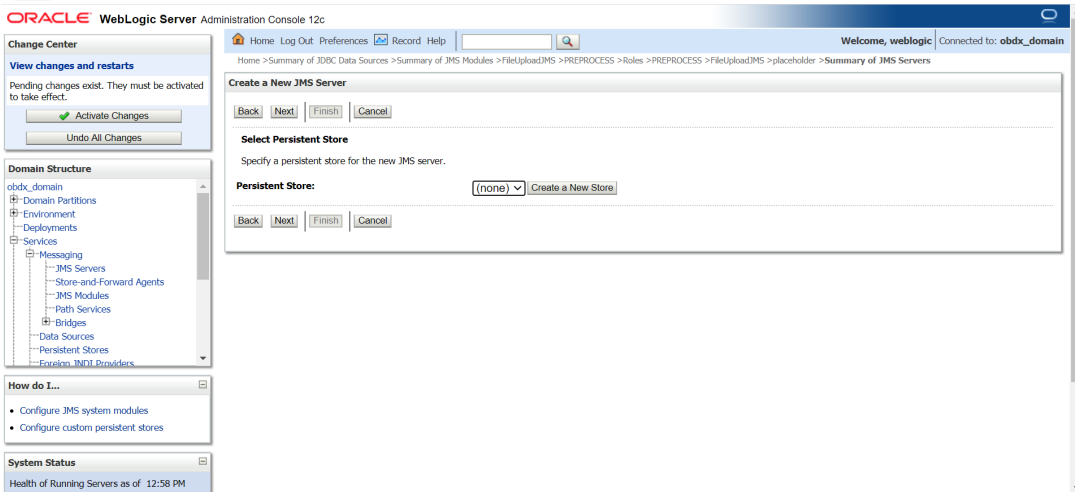
- Creating `FileUploadJMS JSM Module`
- Creating `WLS_JMS_FILEUPLOAD_PS FileStore`
- Creating `FileUploadJMSServer JMS Server`

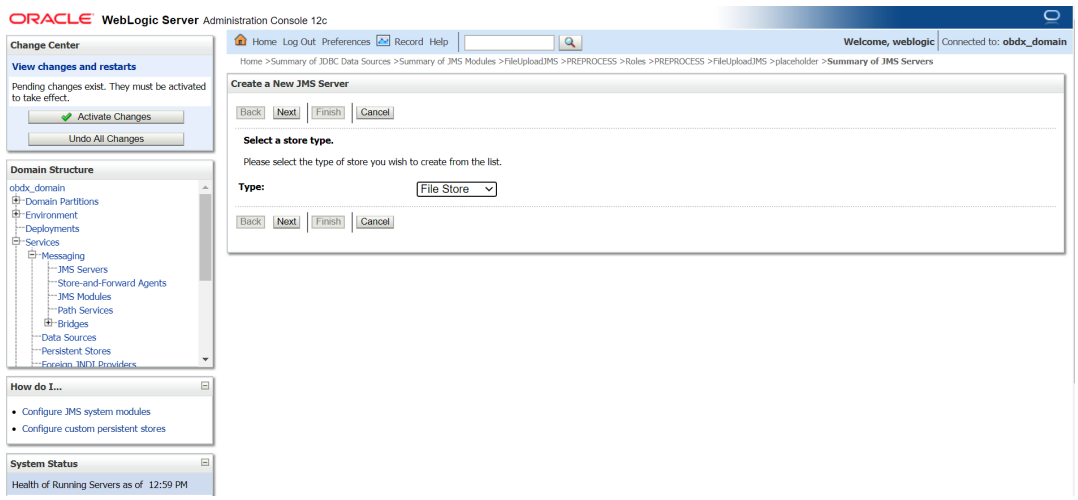
1.



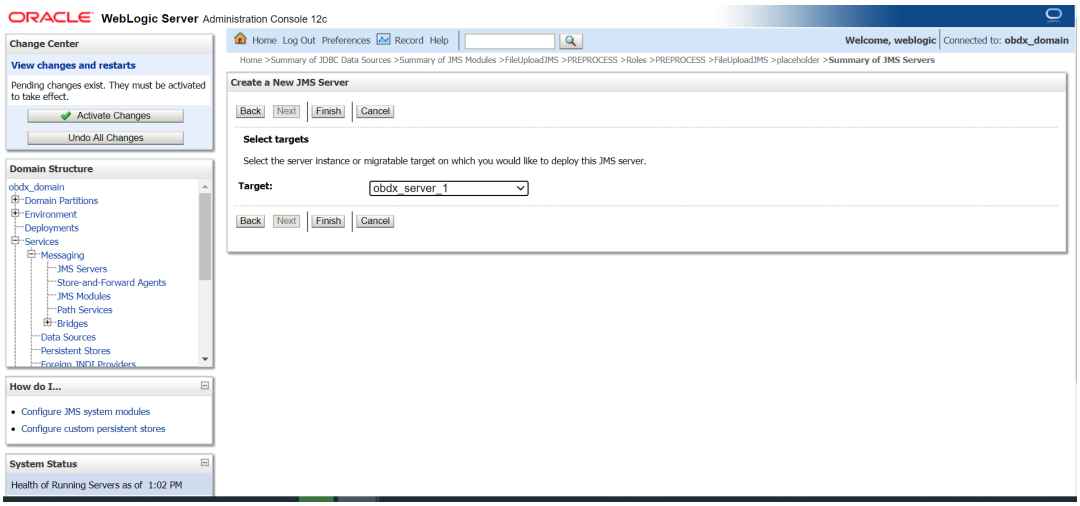


2. Click on JMS Servers → Name – FileUploadJMSServer → Click **Next**.

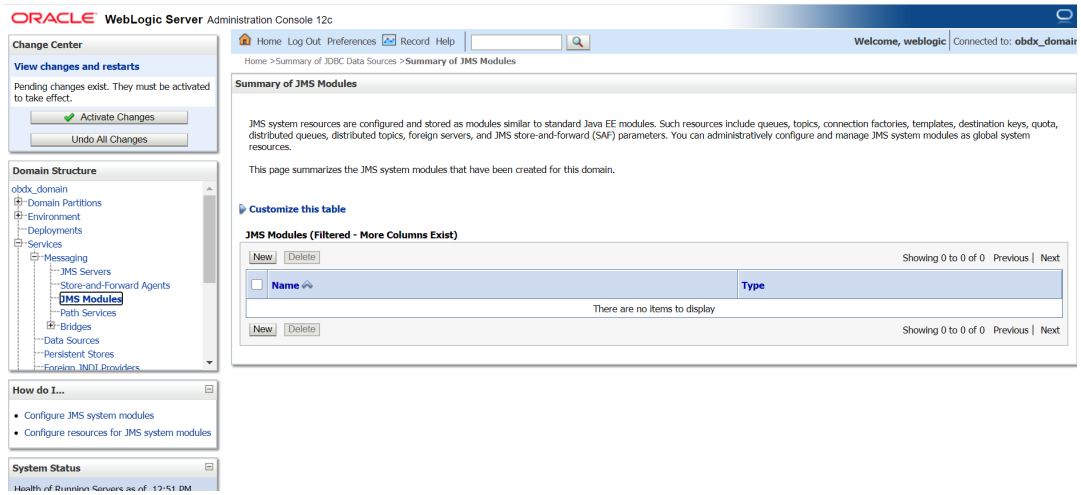




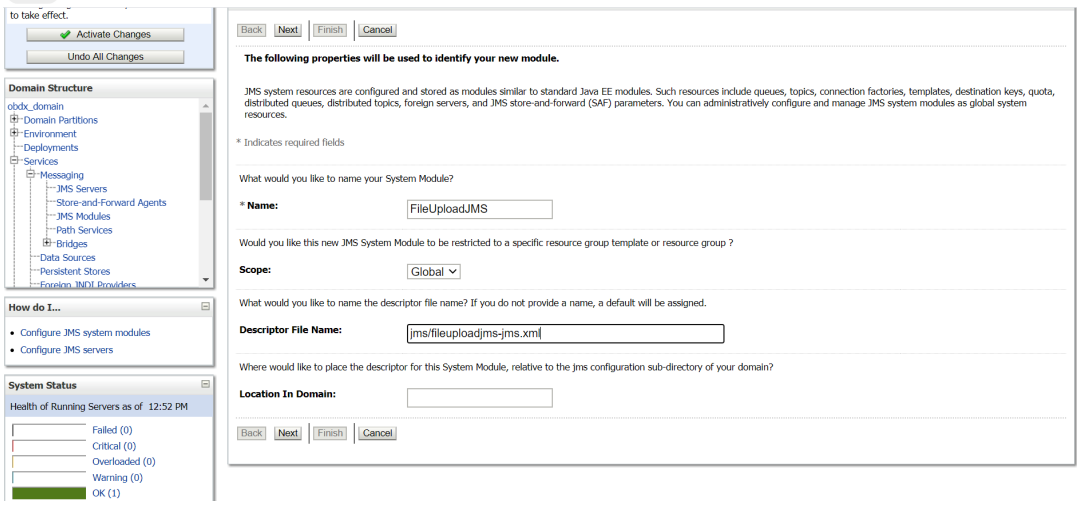
3. Select **Type** as **File Store** and click **Next**.



4. Select target as managed server and click **Finish**.



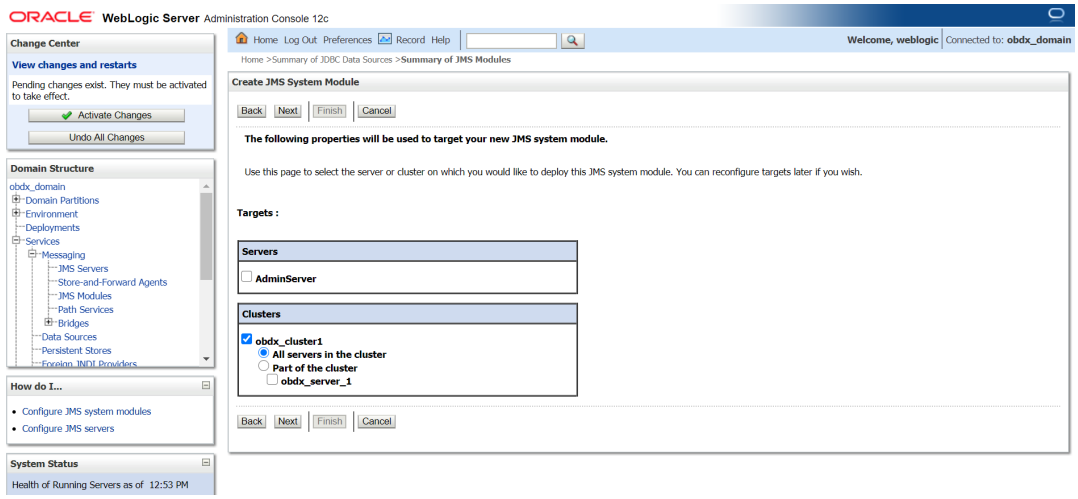
5. Left hand side click on JMS Module → click **New**.



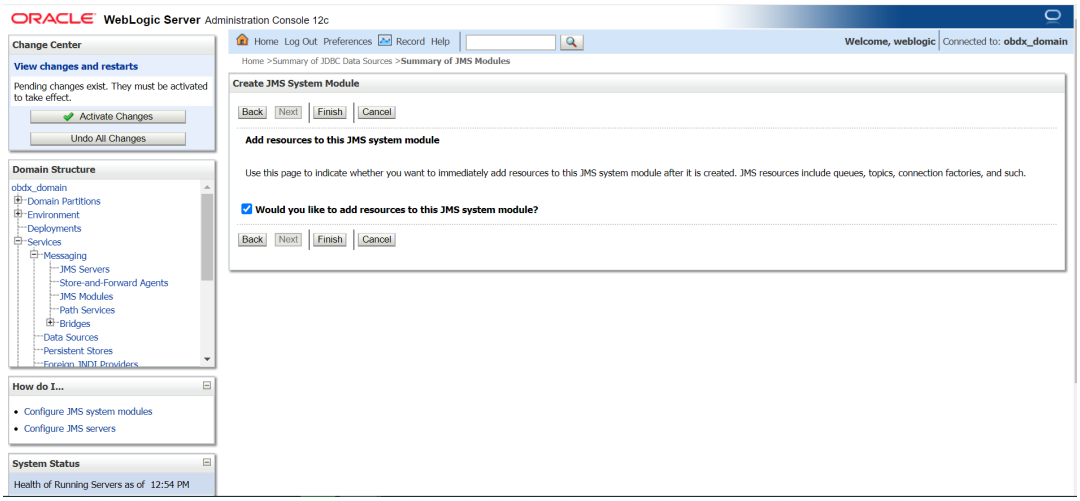
6. **Name :** `FileUploadJMS`
   Scope: Global

   Descriptor File Name: `jms/fileuploadjms-jms.xml`

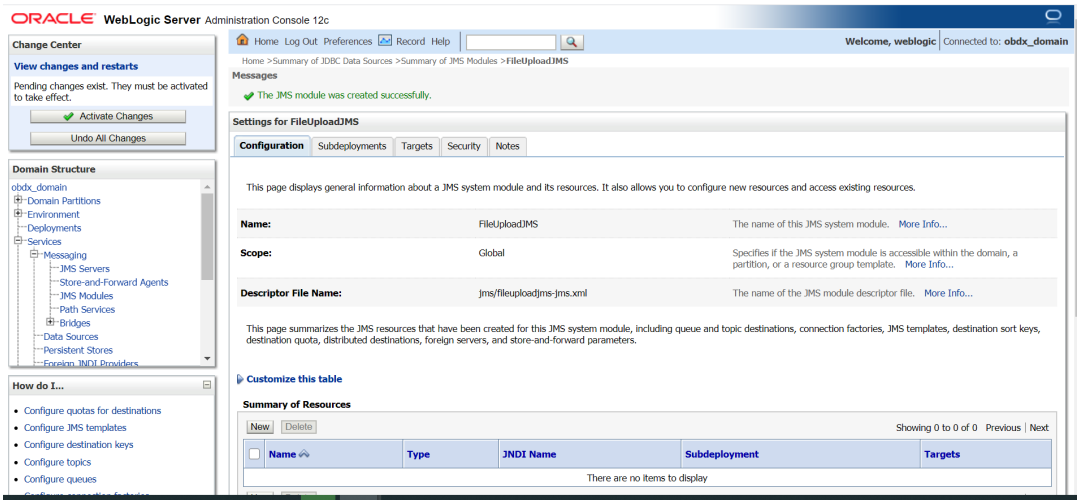
7. Click **Next**.



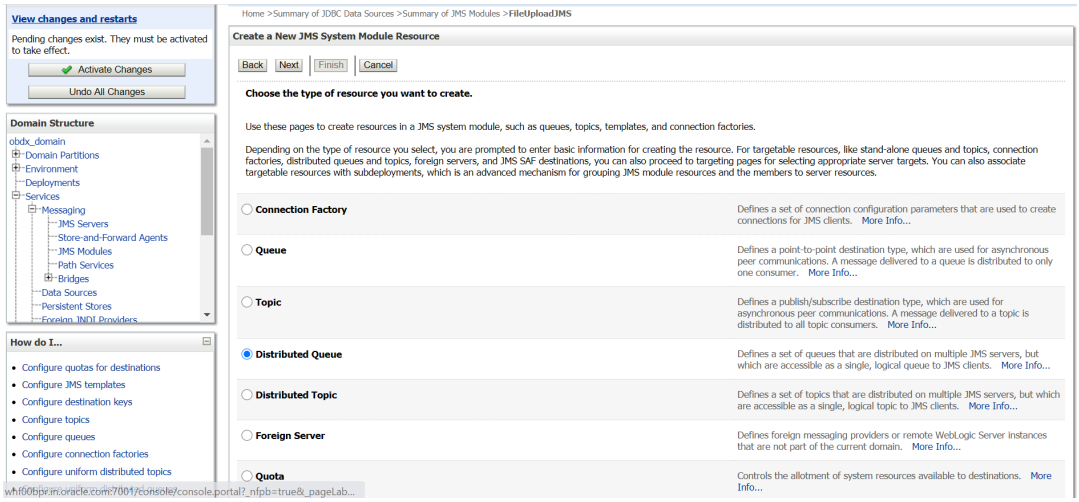
8. Set target as cluster → click **Next**.



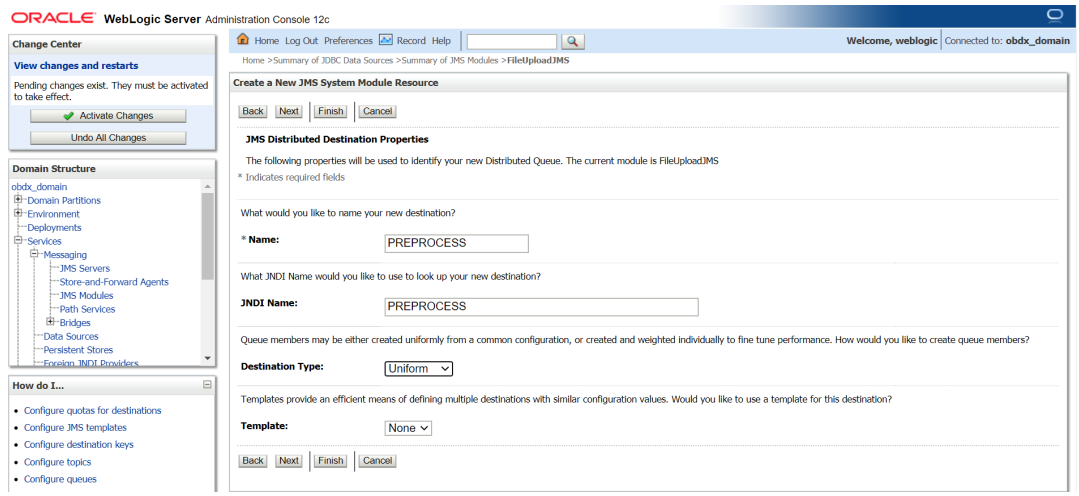
9. Select **Would you like to add resources to this JMS system** module and click **Finish**.



10. Select **New**.



11. Select **Distributed Queue** and click**Next**.

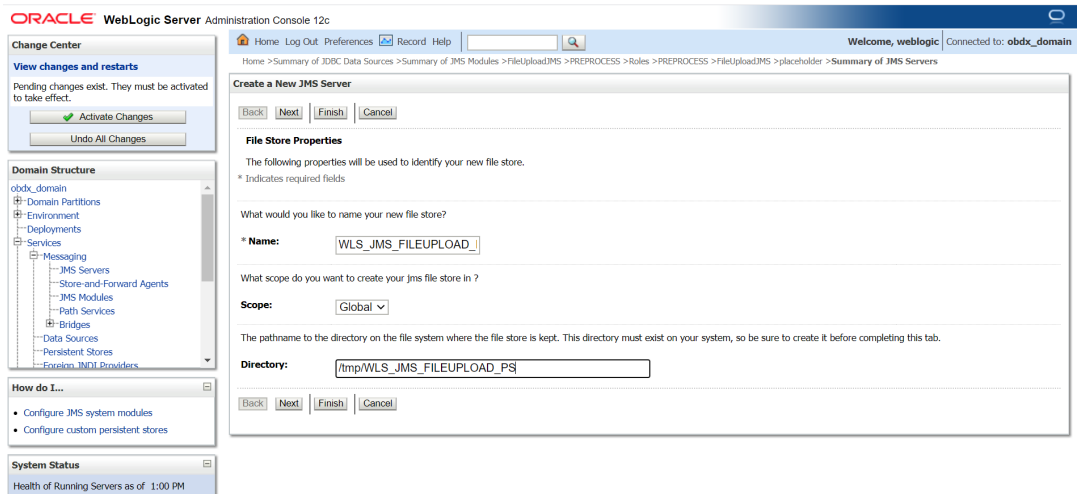


12. **Provide**
Name: `PREPROCESS`

JNDI Name: `PREPROCESS`

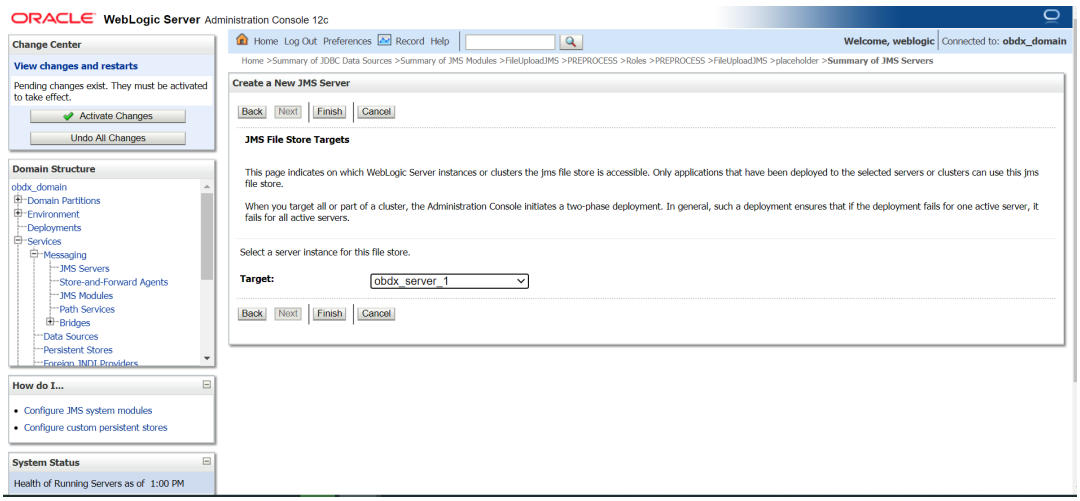
Destination Type: Uniform

Template: None

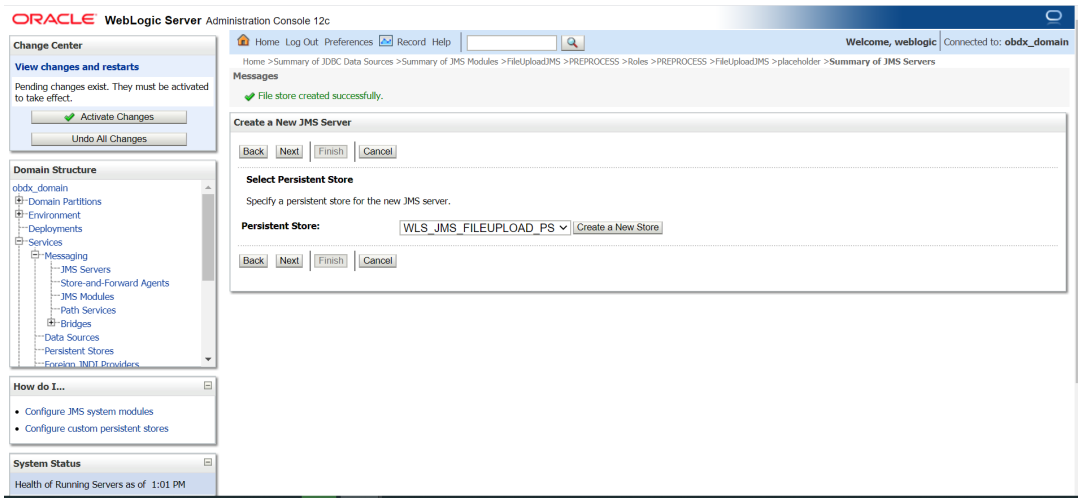


13. **Name :** `WLS_JMS_FILEUPLOAD_PS`
Scope : Global

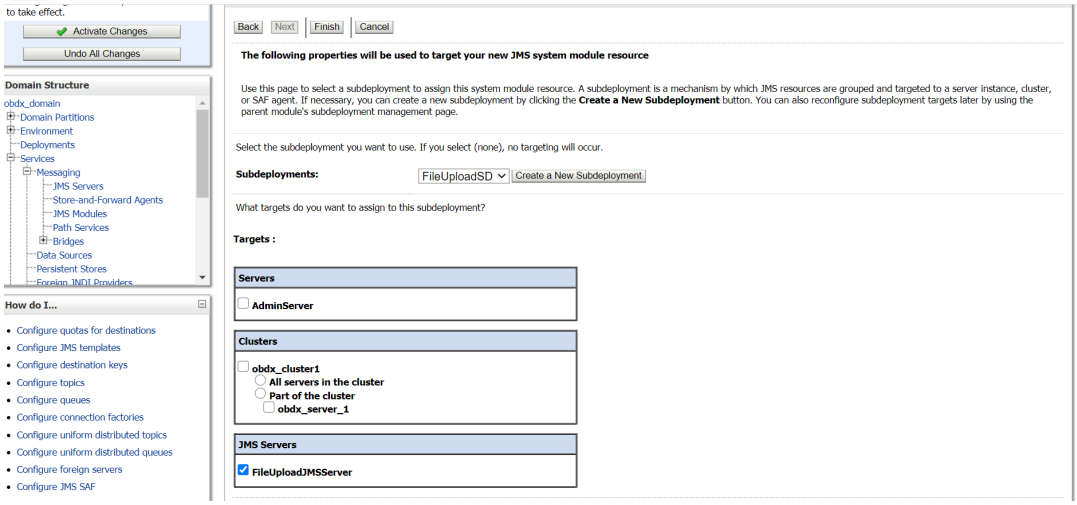
Directory : `/tmp/WLS_JMS_FILEUPLOAD_PS`



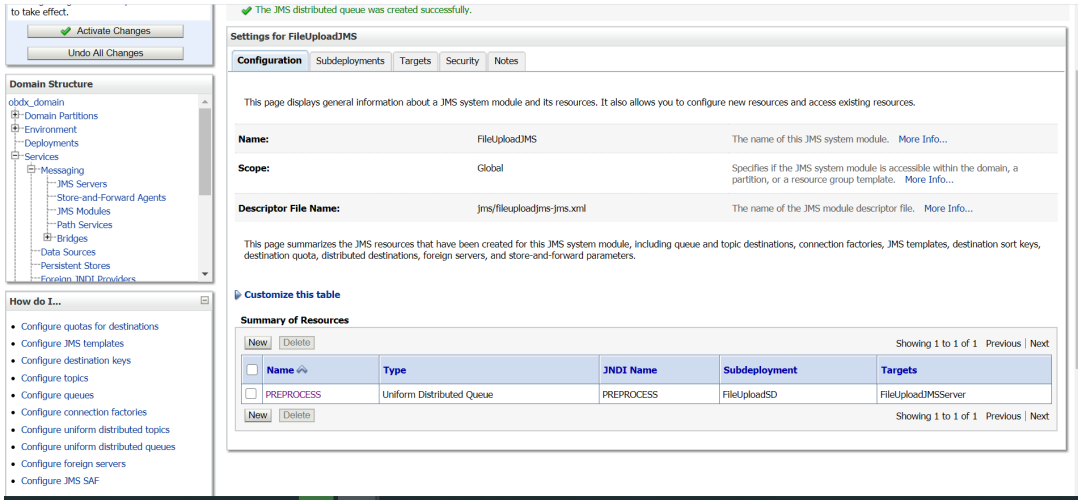
14. Select target as managed server.



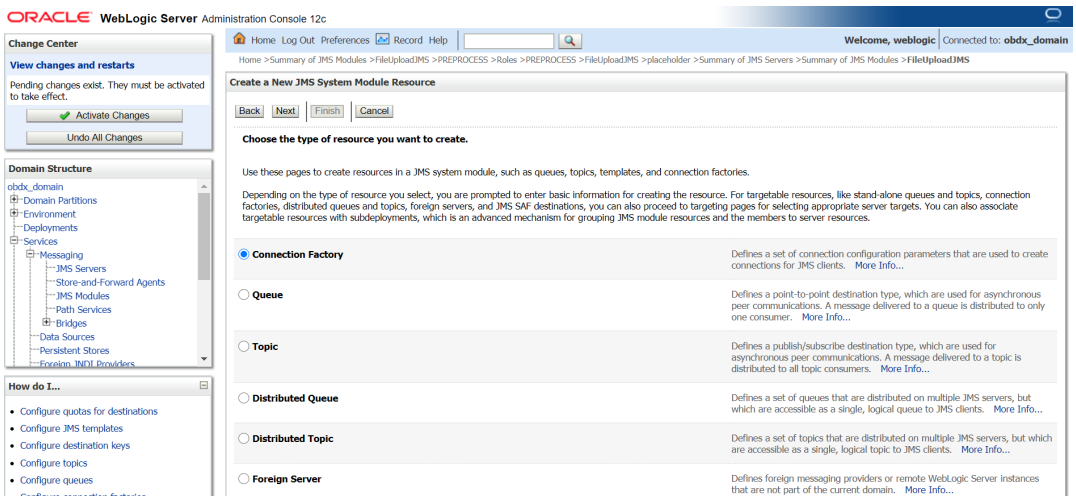
15. Select `WLS_JMS_FILEUPLOAD_PS` and click **Next**.
16. Select **Create a New Subdeploymeny** and create **FileUploadSD**.



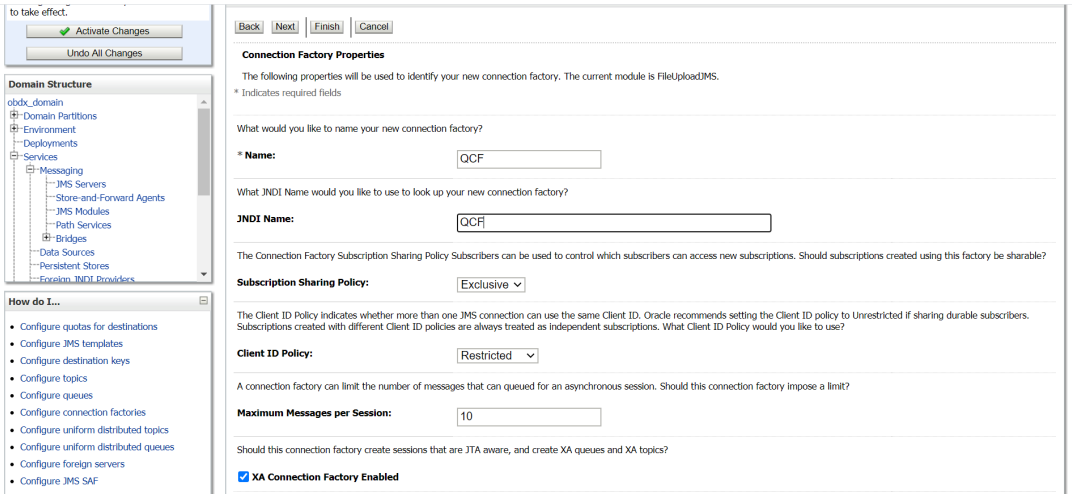
17. Select **FileUploadJMSServer** and click **Finish**.



18. Similarly Go into **FileuploadJMS** module and click **Next**.



19. Select Connection factory → Click **Next**.

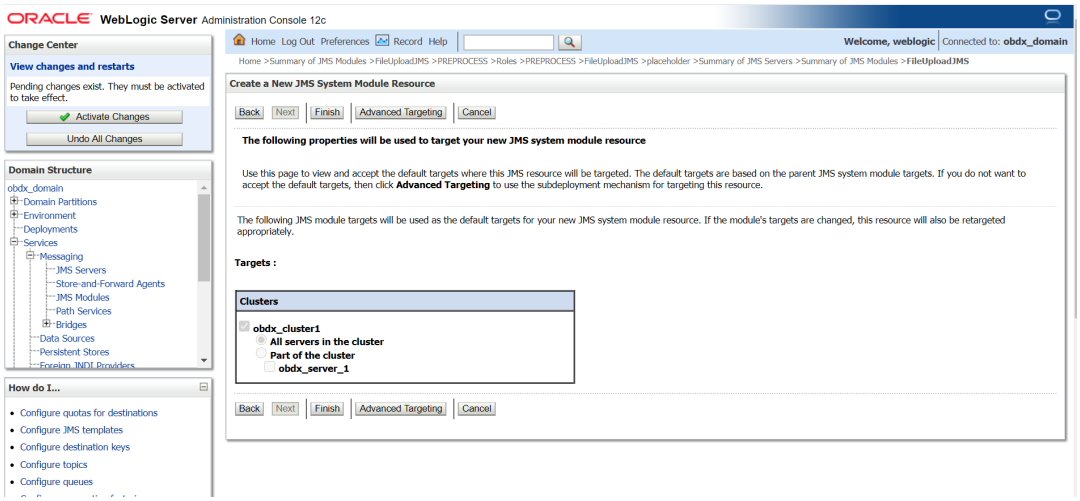


20. **Provide**
Name : `OCF`

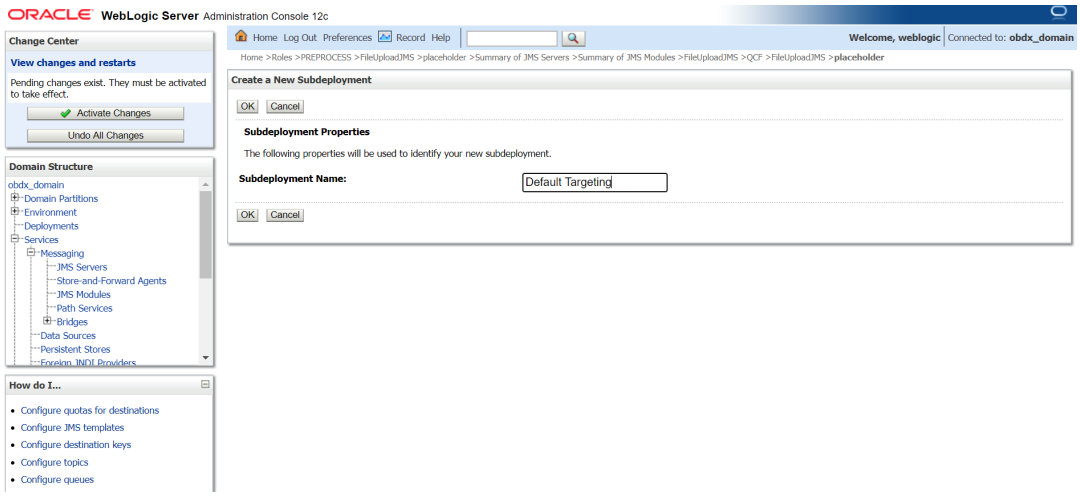
JNDI Name : `OCF`

Subscription Sharing Policy : Exclusive

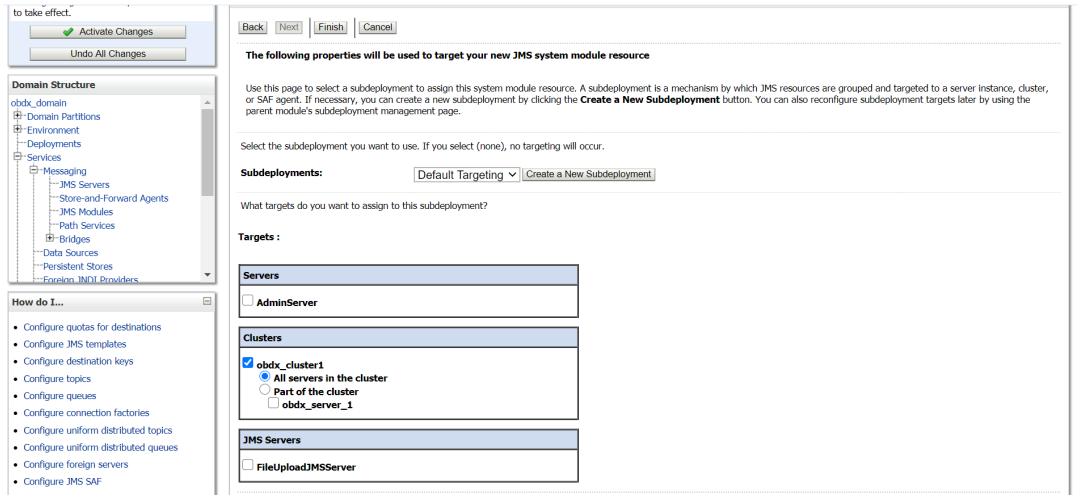
Client ID Policy : Restricted



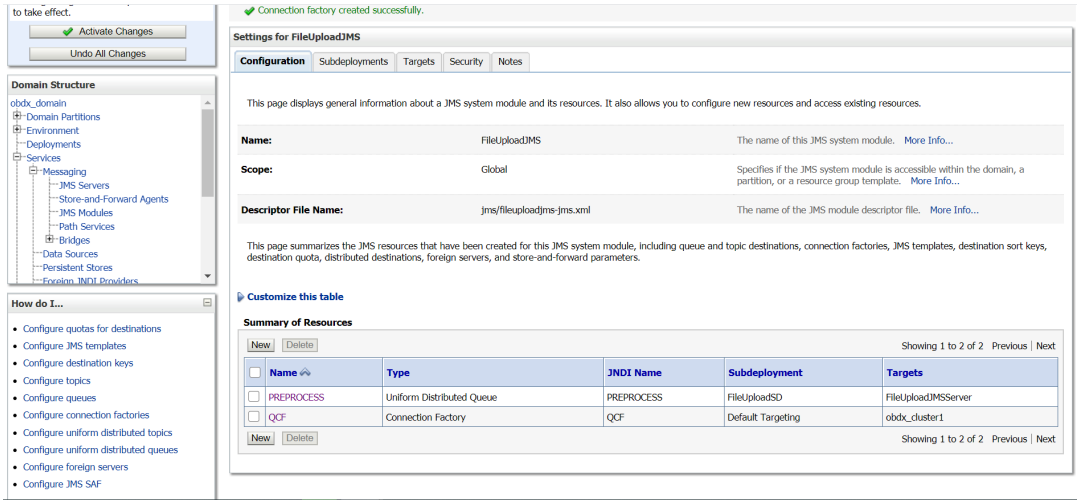
21. Click on Advanced targeting.



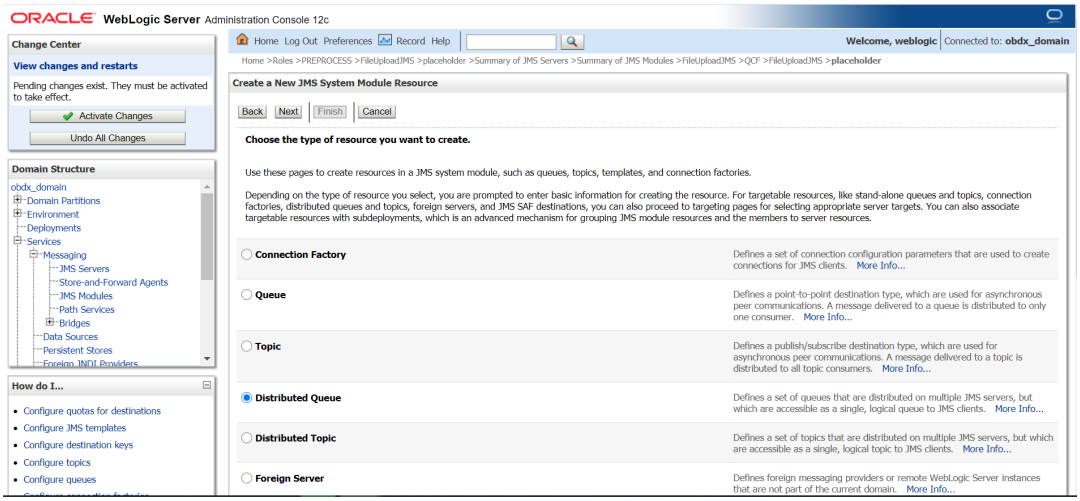
22. Provide Subdeployment Name as Default Targeting.



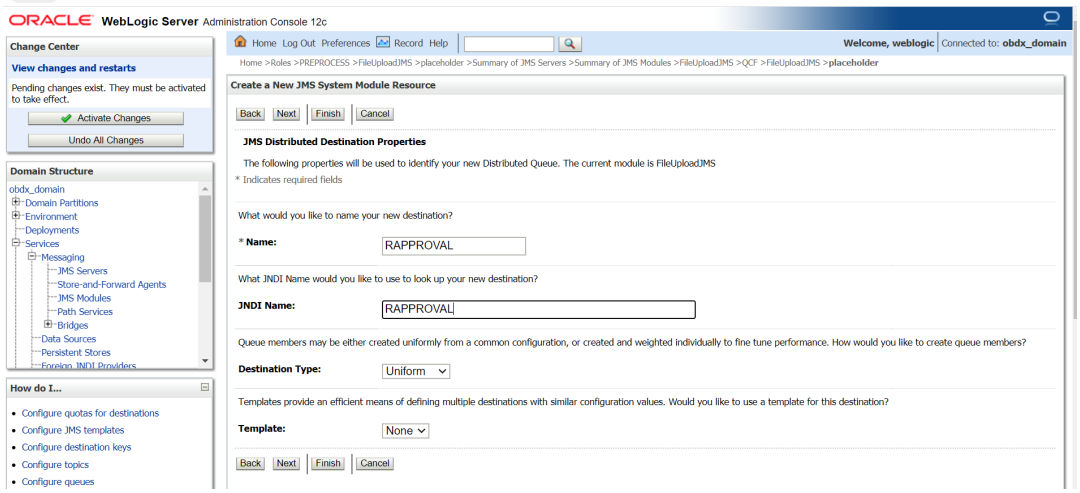
23. Select cluster and click **Finish**.



24. Go to **FileUpload JMS** and click **New**.



25. Select Distributed Queue.

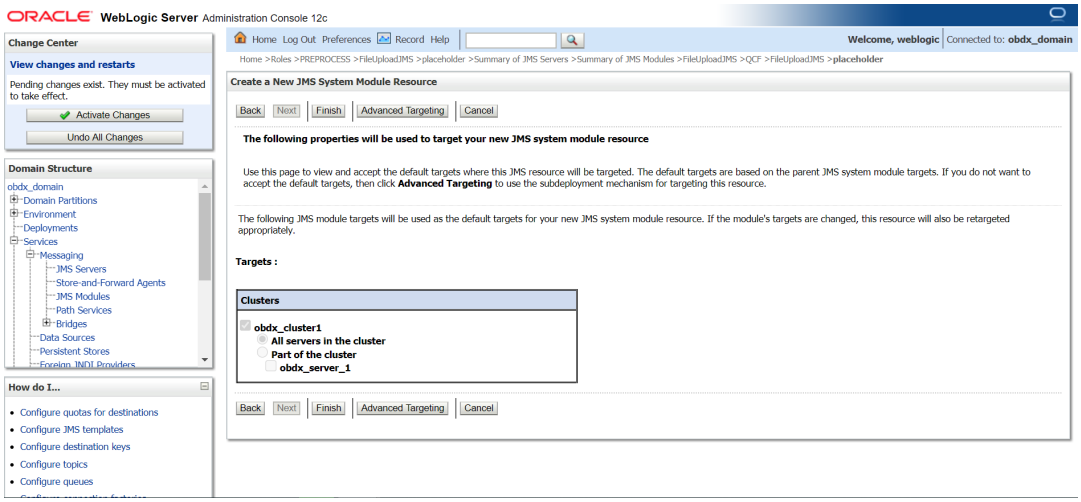


26. **Provide**
Name : `RAPPROVAL`

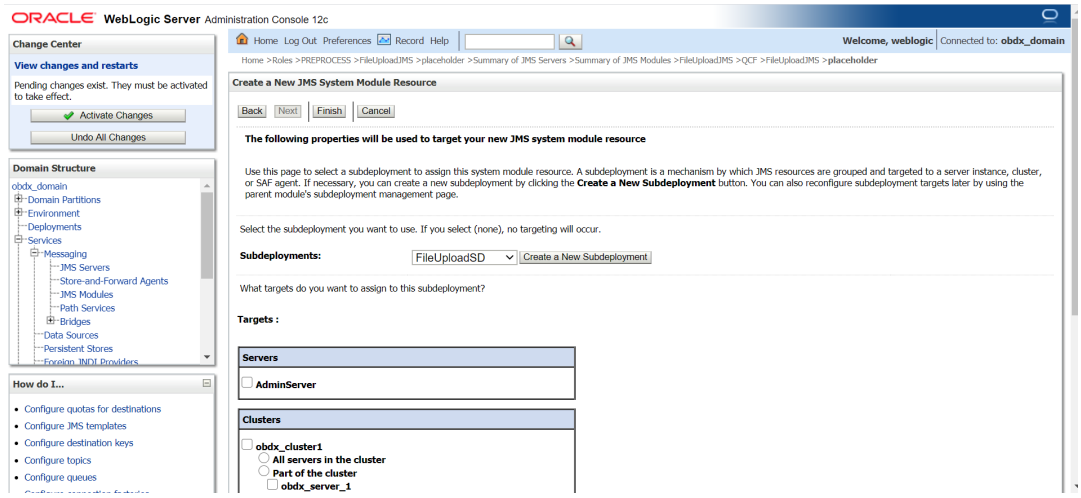
JNDI Name : `RAPPROVAL`

Destination Type: Uniform

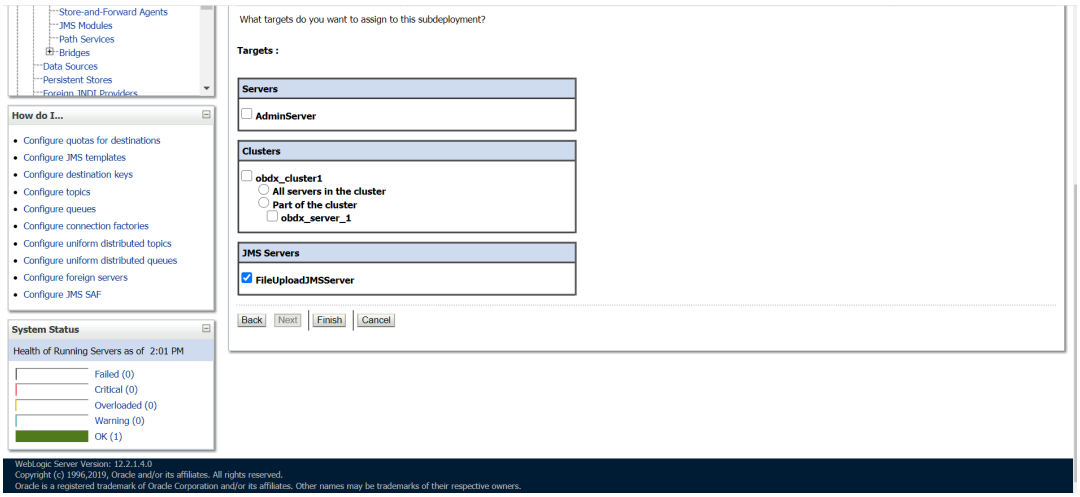
Template : None



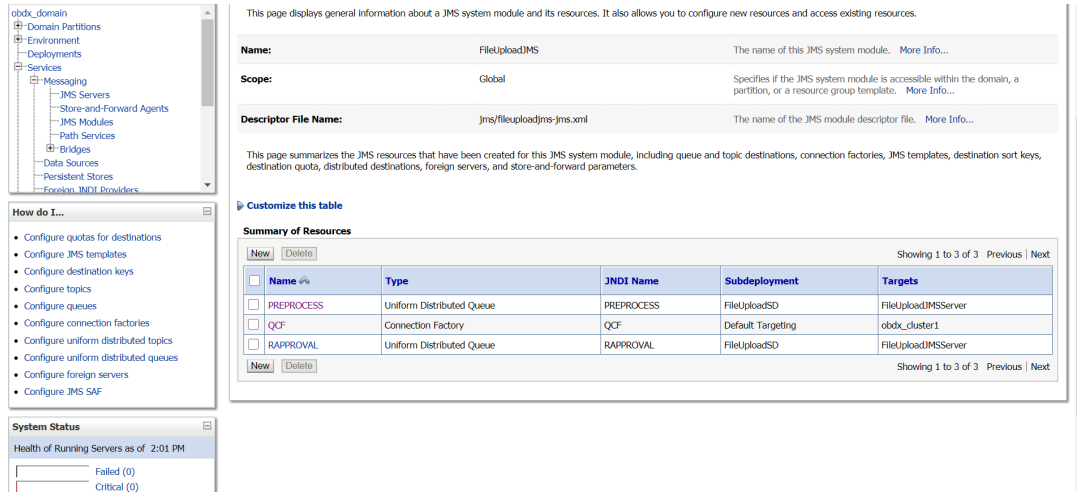
27. Select Advance targeting.



28. Select Subdeployment: FileUploadSD.



29. Select **FileUploadJMSServer** and click **Finish**.

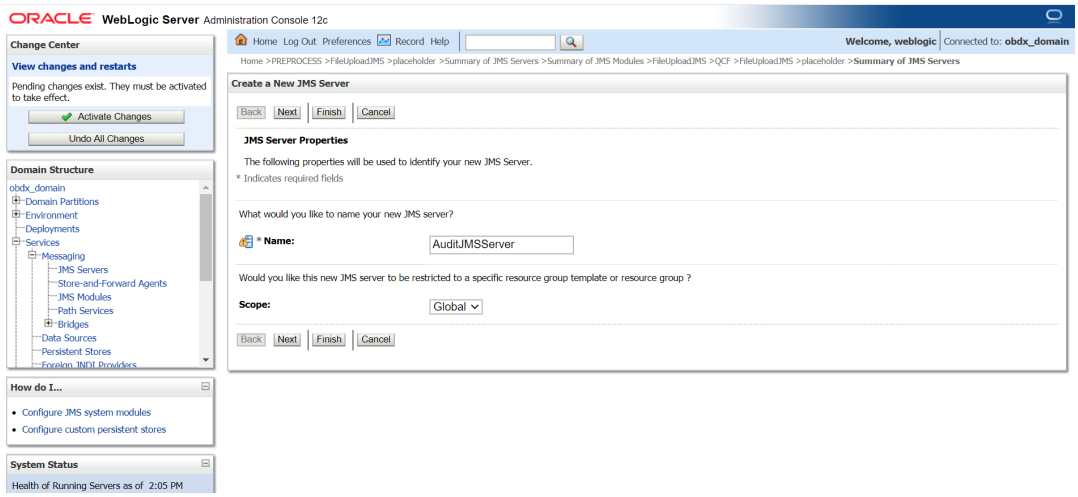


2.8 Creating WLS_JMS_AUDIT_PS FileStore

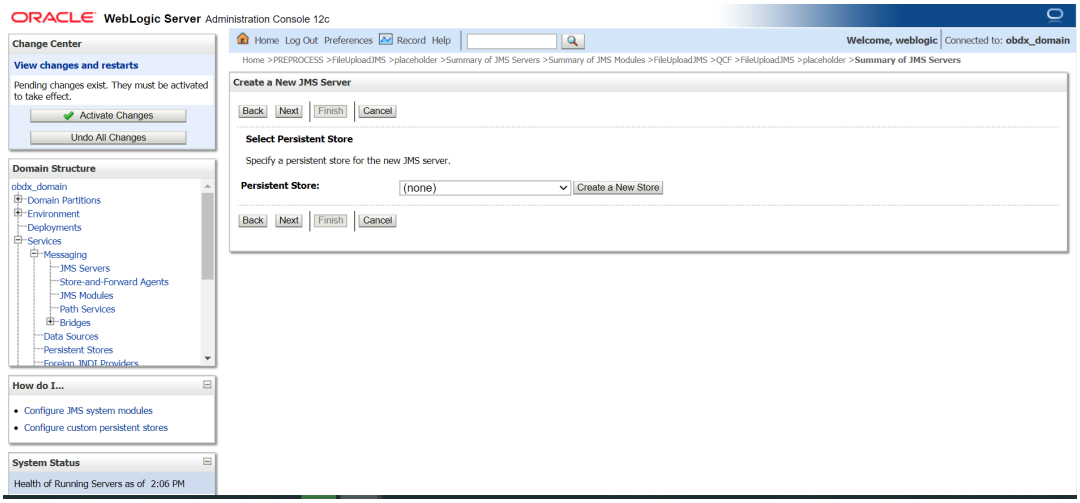
2.9 Creating AuditJMSServer JMS Server

2.10 Creating WLS_JMS_REPORT_PS FileStore

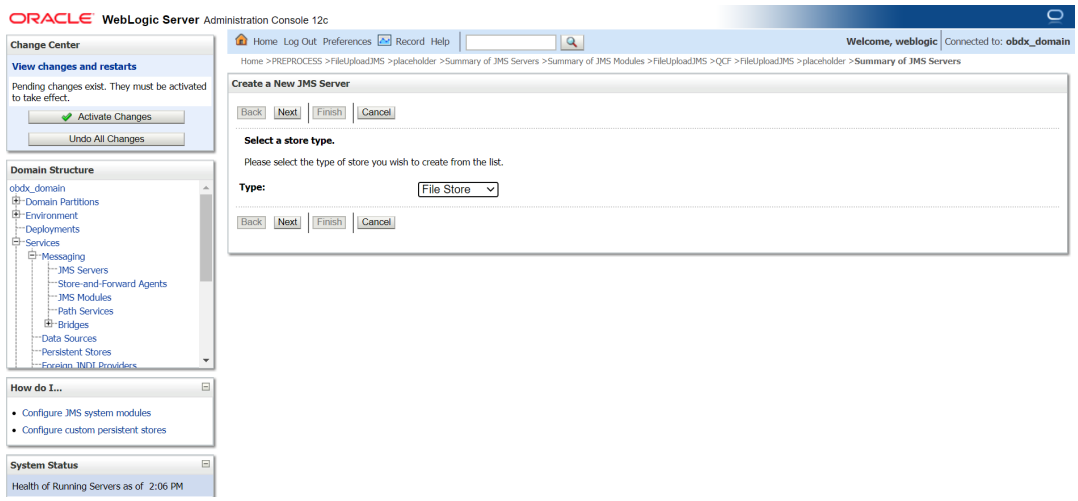
1.



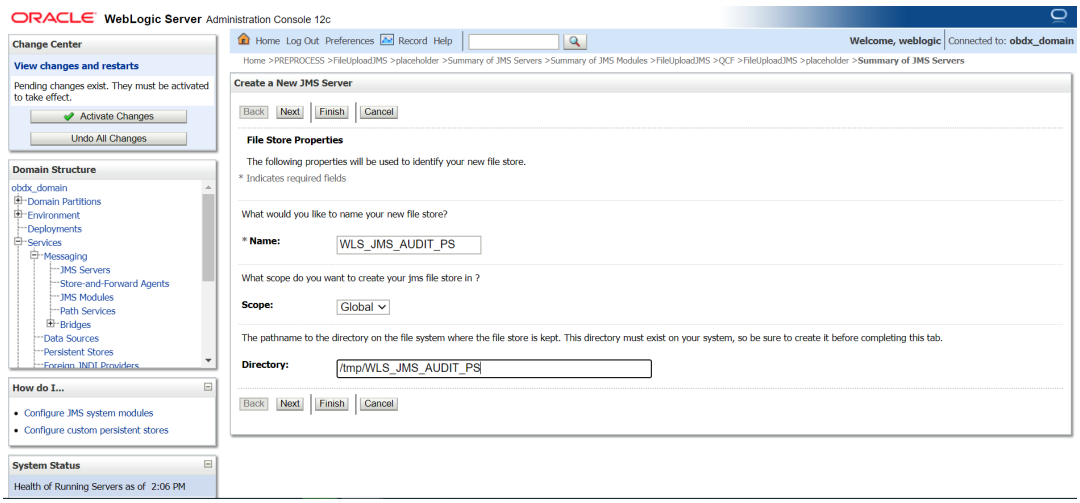
2. Click on JMS server and click **New** .
3. Provide Name as AuditJMSServer , Scope as **Global**.



4. Click on **Create a New Store**.



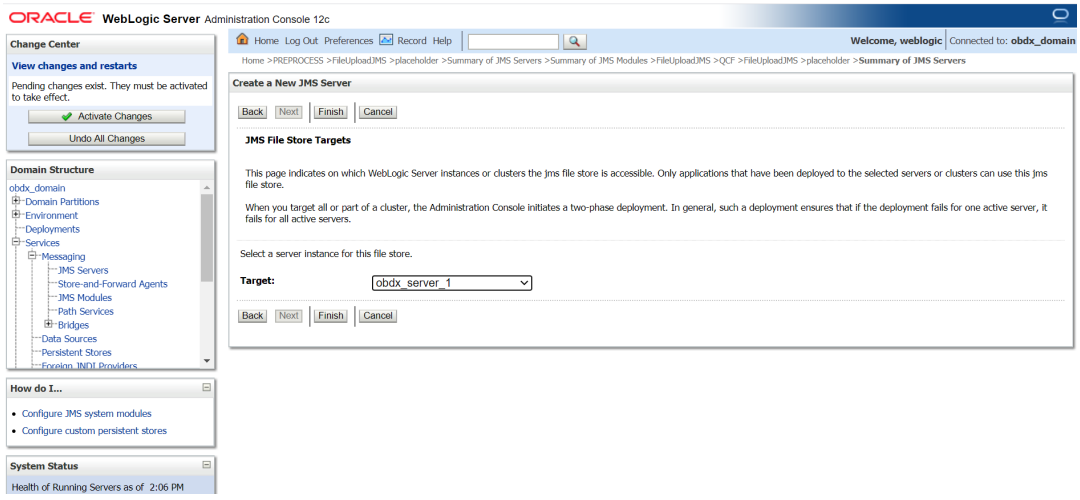
5. Select **File Store**.



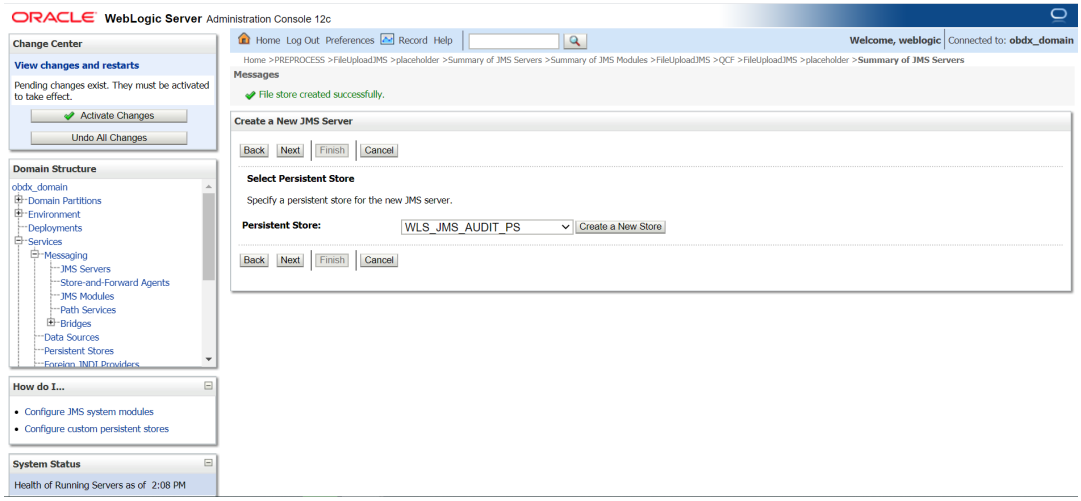
6. **Provide**
Name : `WLS_JMS_AUDIT_PS`.

Scope : Global

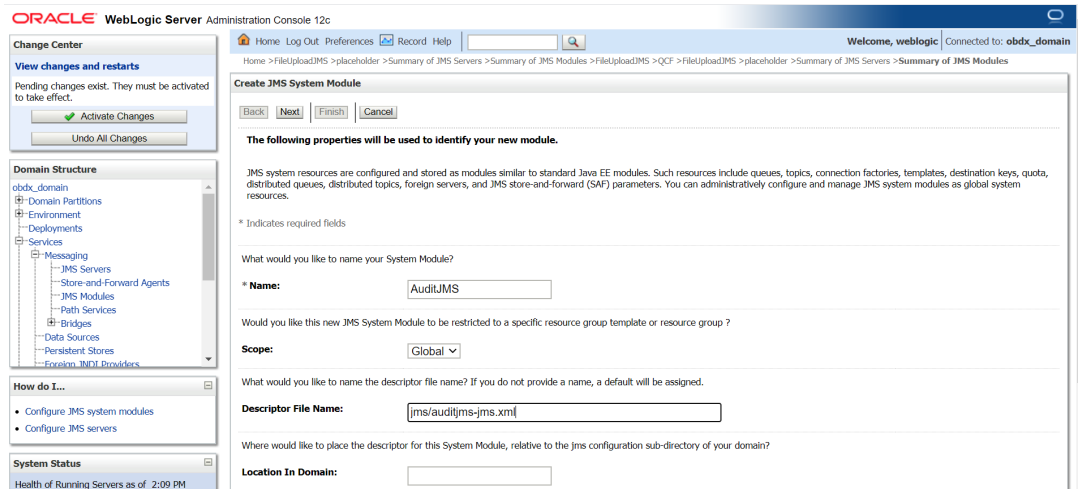
Directory : `/tmp/WLS_JMS_AUDIT_PS`.



7. Select Target as managed server and click **Finish**.



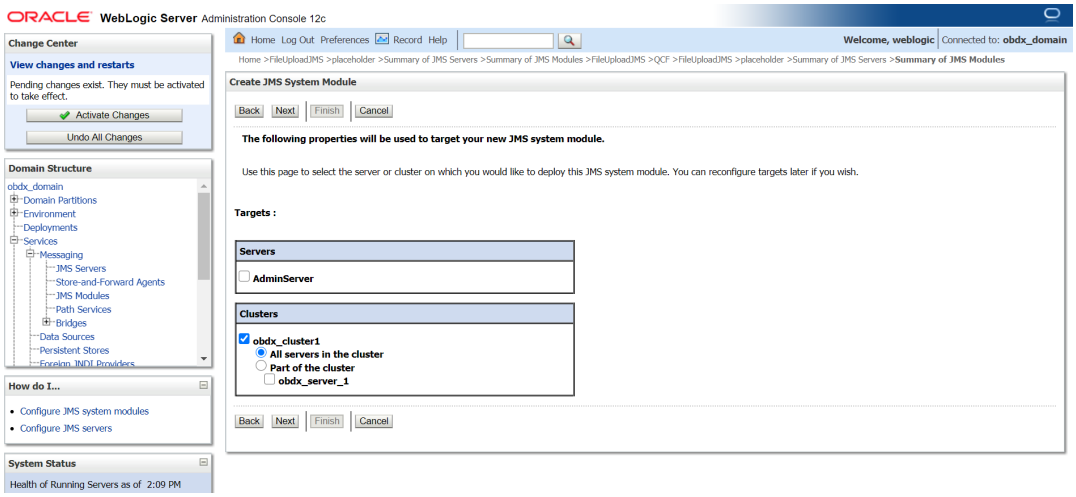
8. Select the new store created `WLS_JMS_AUDIT_PS` and click **Next**.



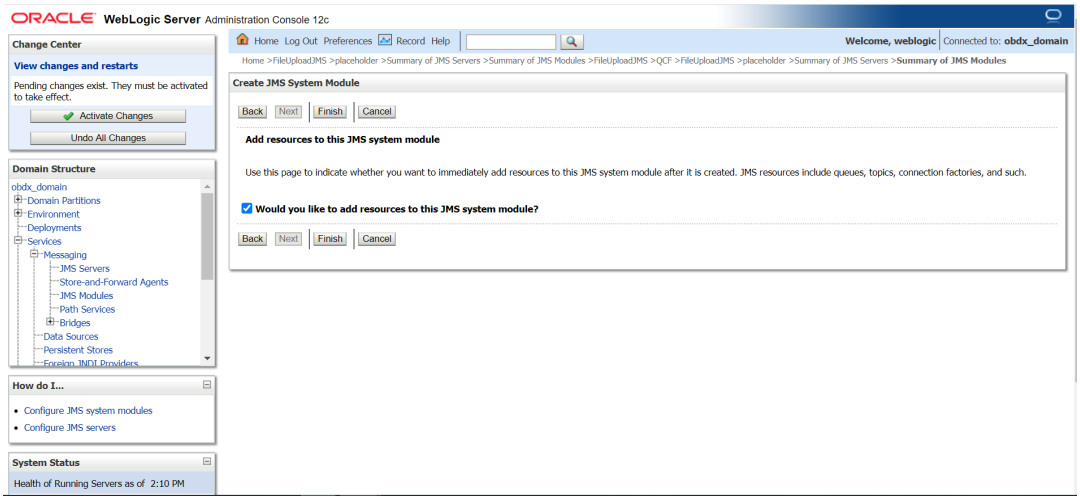
9. **Provide**
   Name : `AuditJMS`

   Scope : Global

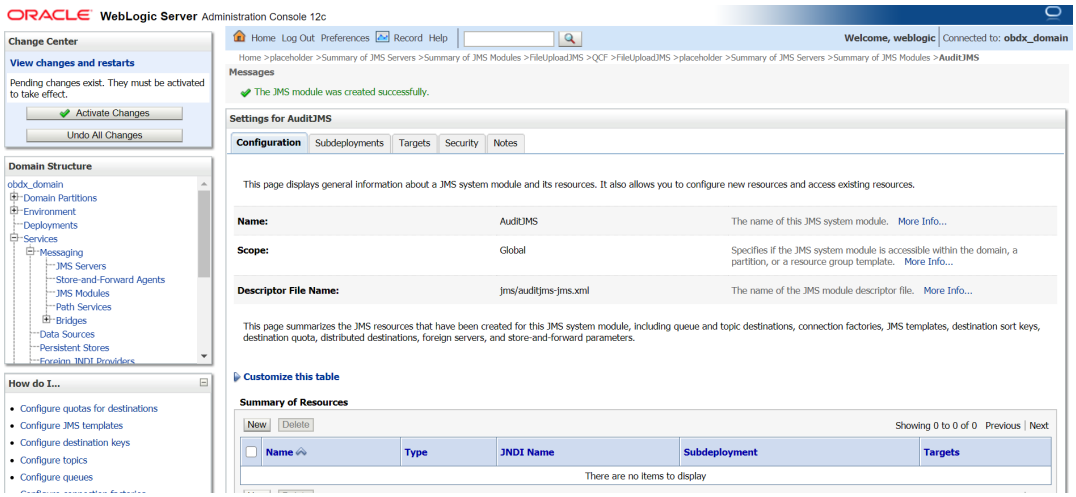
   Descriptor File Name: `jms/auditjms-jms.xml`



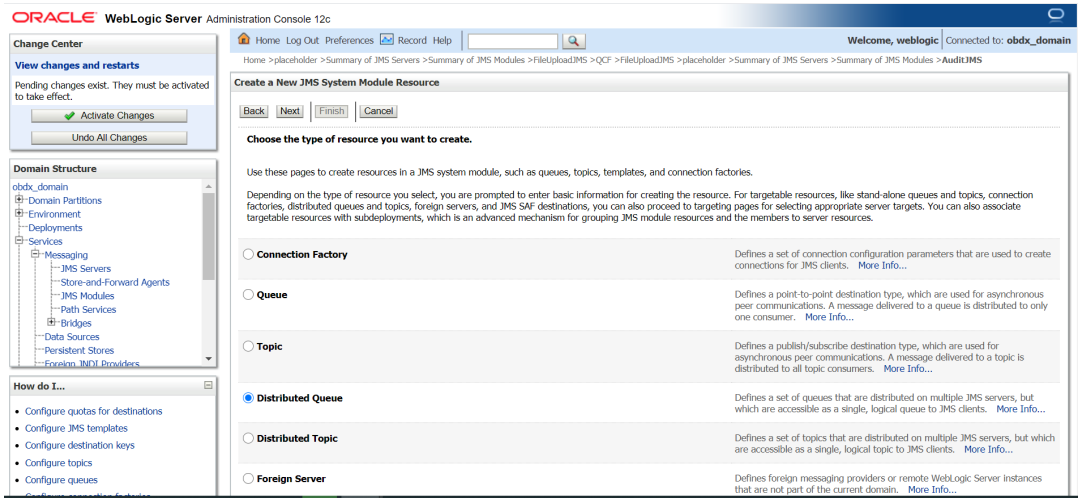
10. Select Cluster as a target.



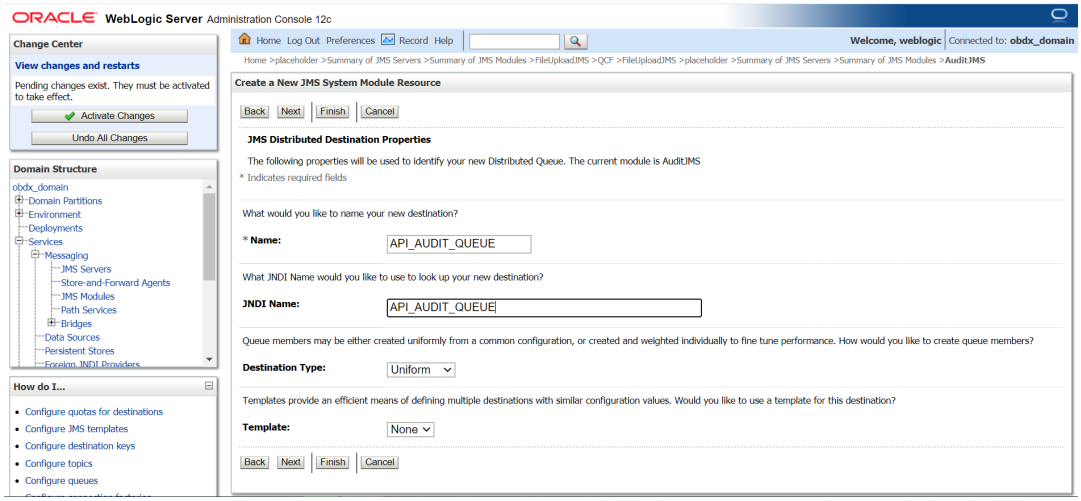
11. Select **would you like to add resource to this JMS system module?**.



12. Click **New**.



13. Select Distributed Queue.

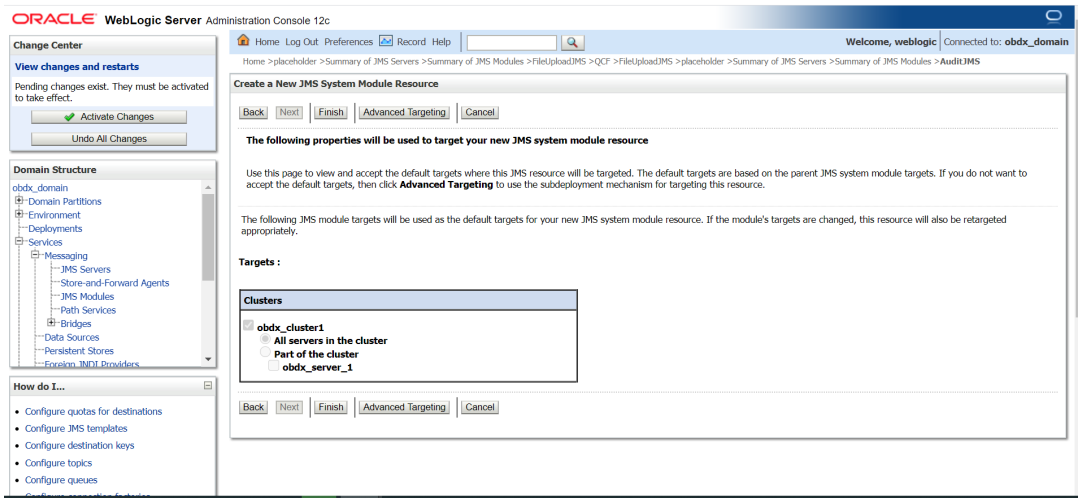


14. Provide:
Name: `API_AUDIT_QUEUE`

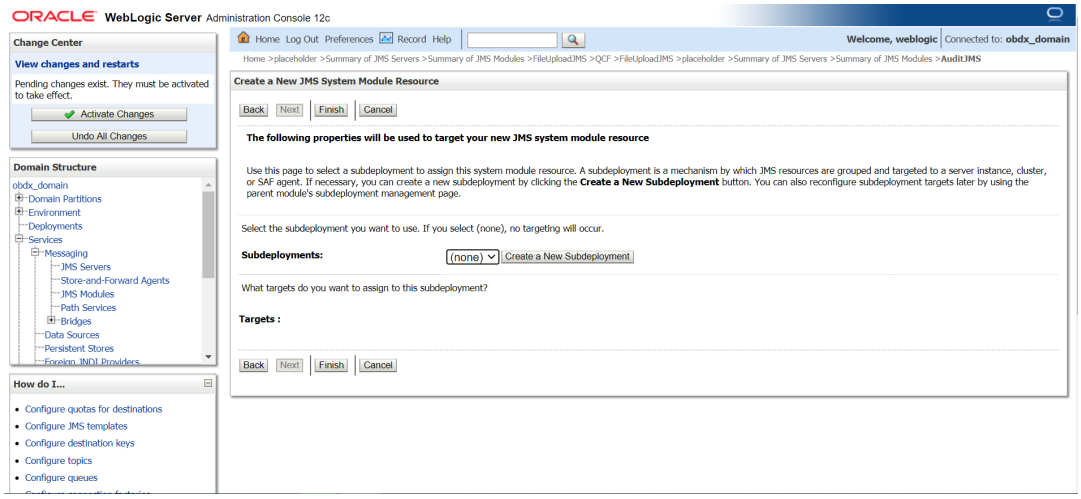
JNDI Name: `API_AUDIT_QUEUE`

Destination Type : Uniform

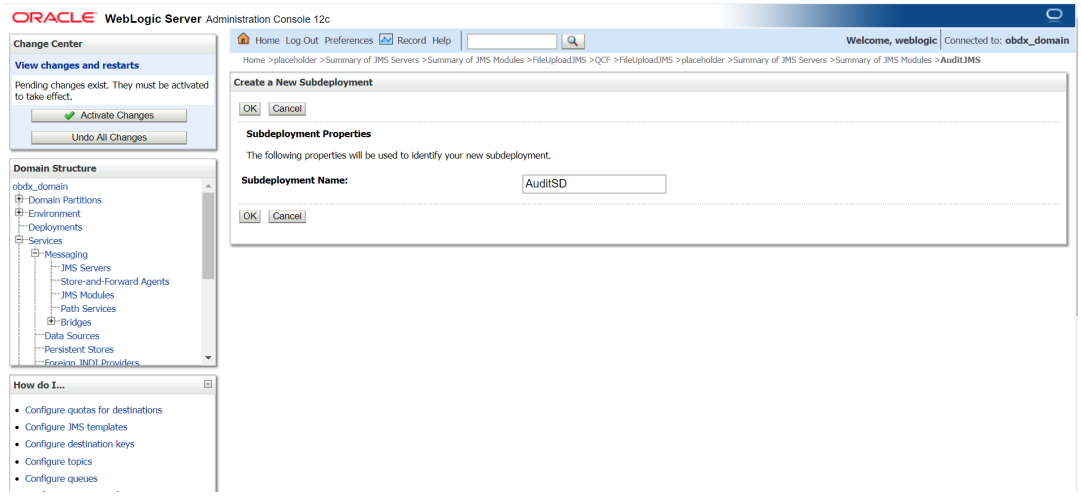
Template:- None



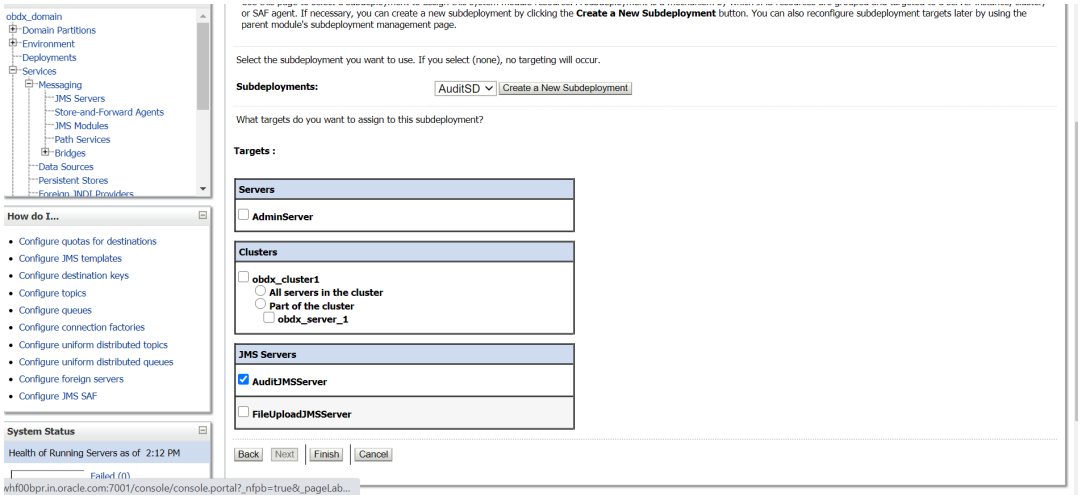
15. Select Advance targeting.



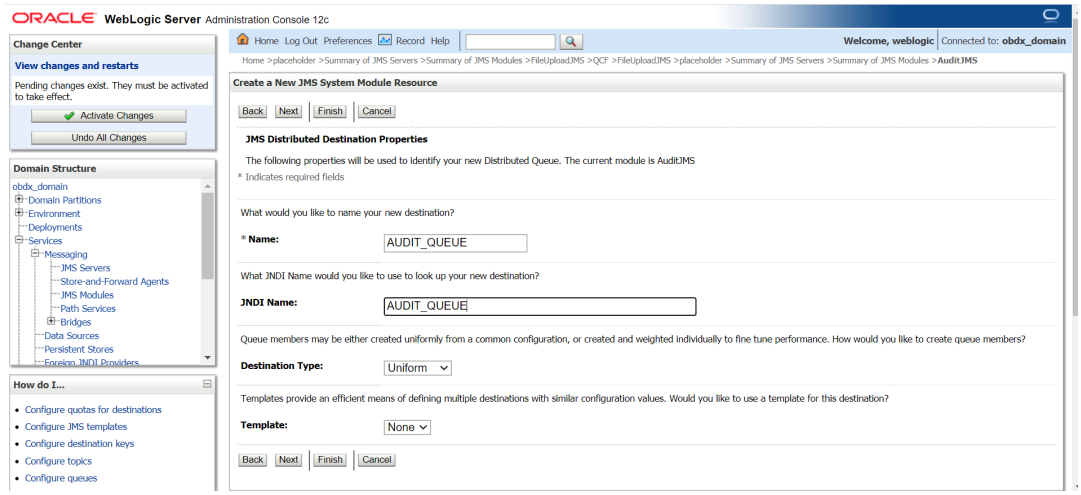
16. Click on Create a New Subdeployment.

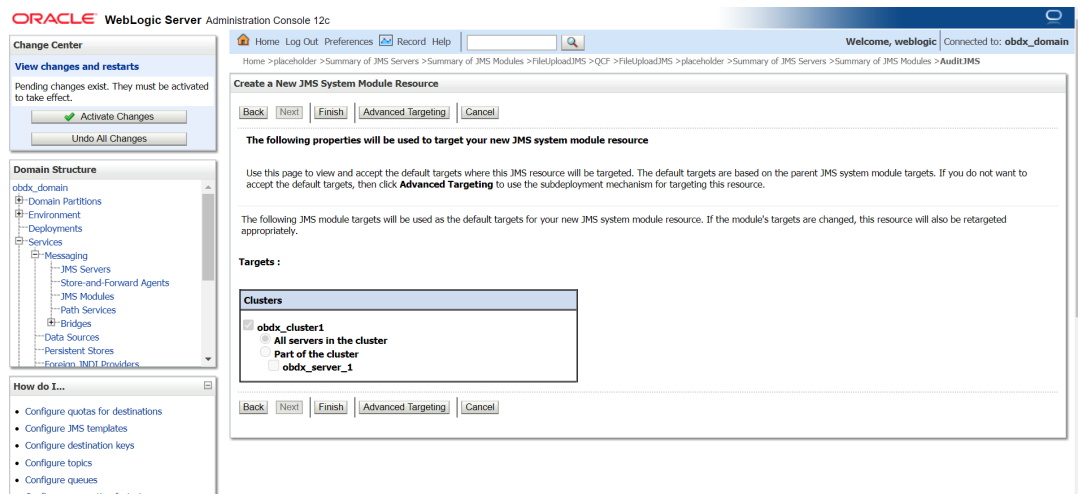


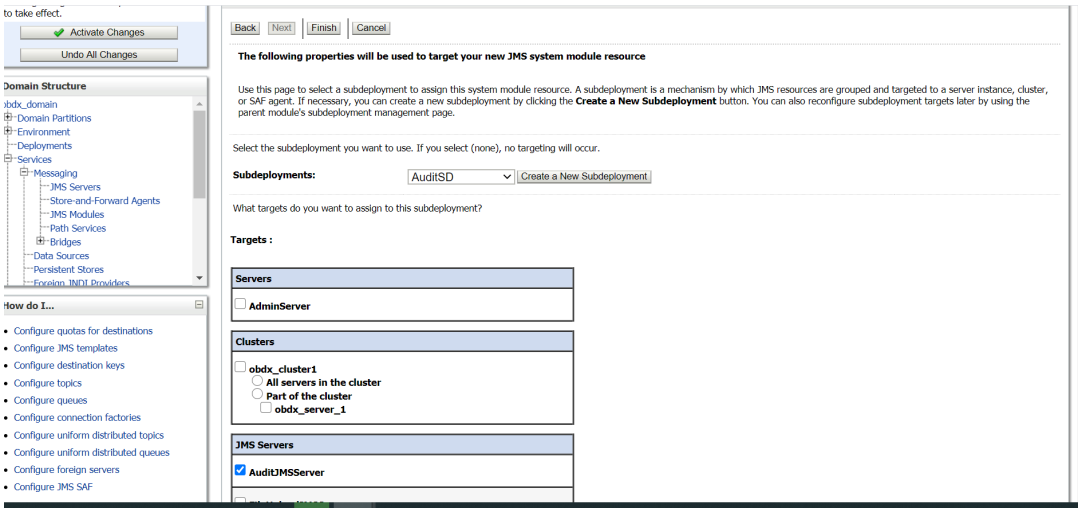
17. Provide Subdeployment Name as AuditSD.

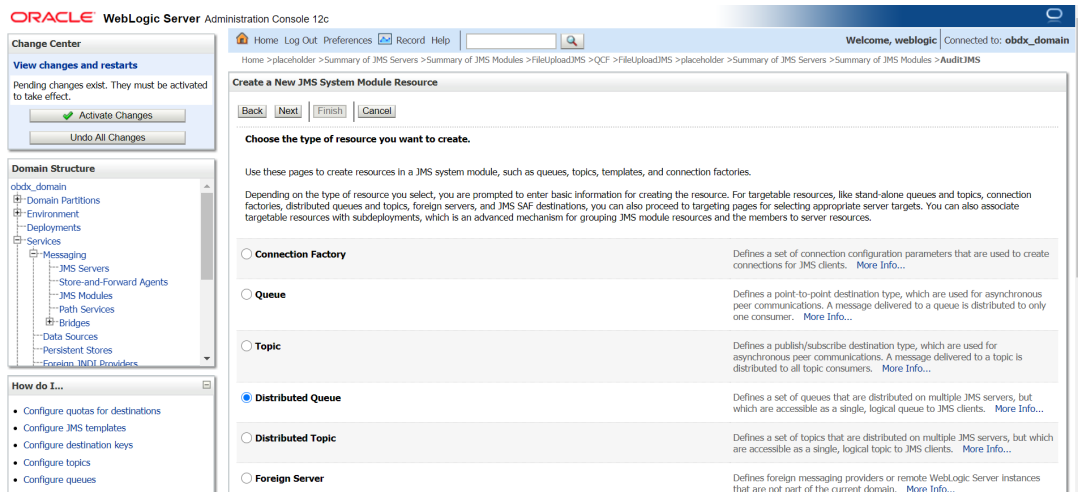


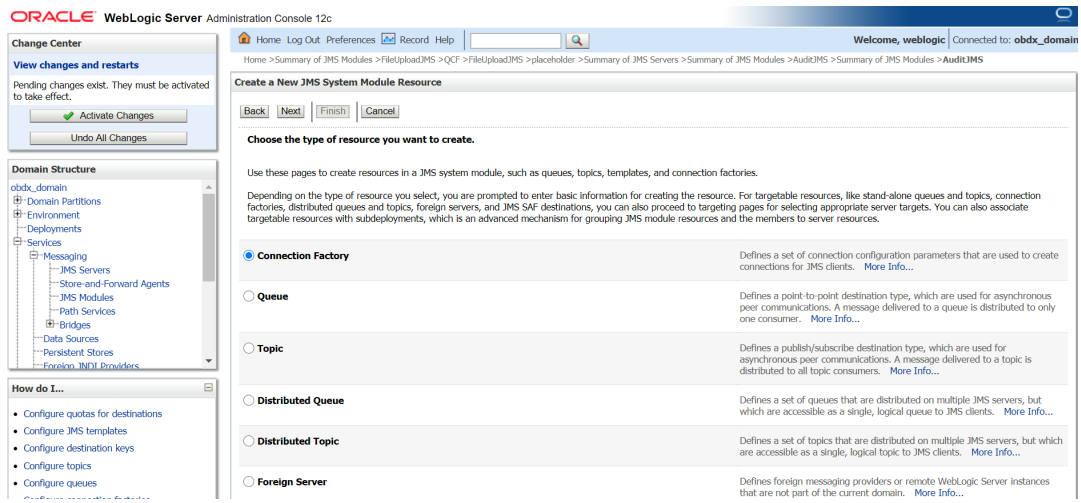
18. Select Target as AuditJMSServer.



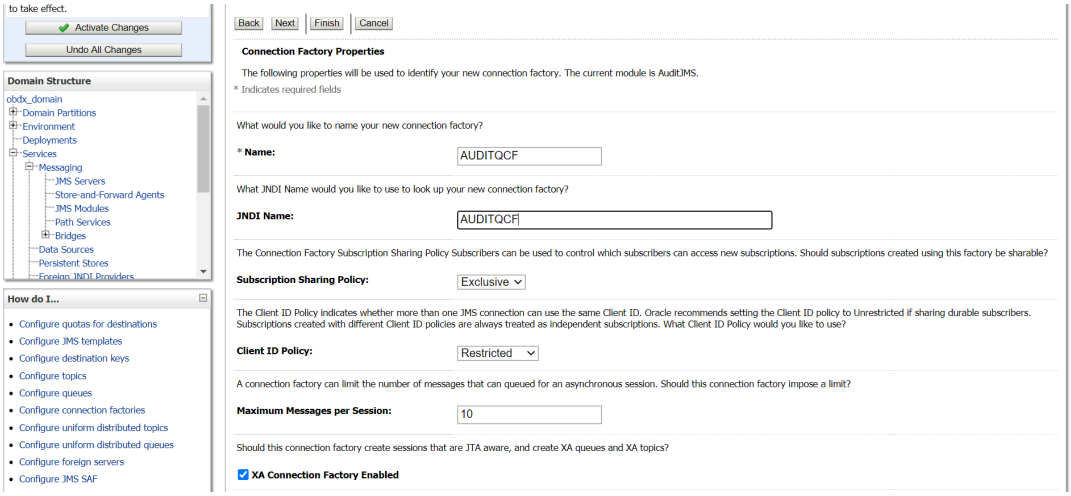






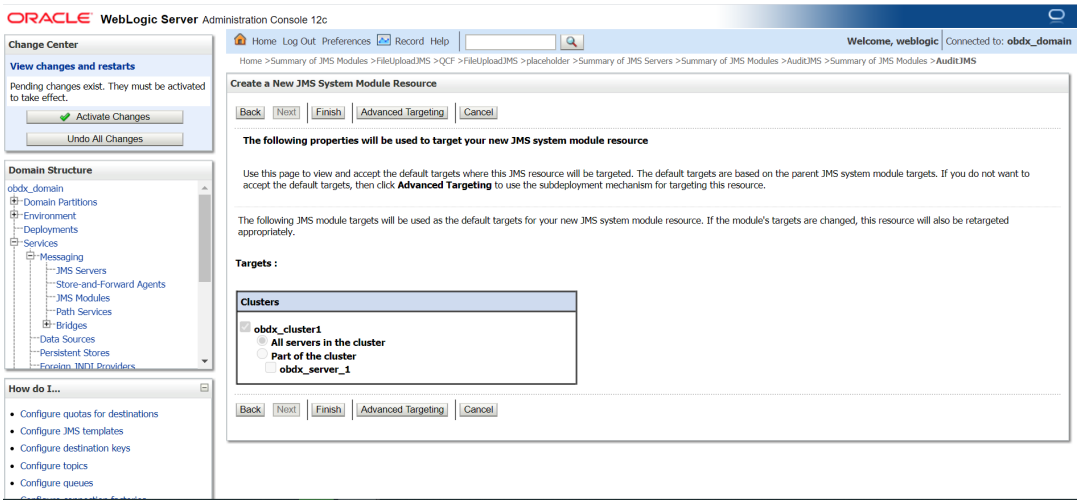


19. Click on connection Factory.

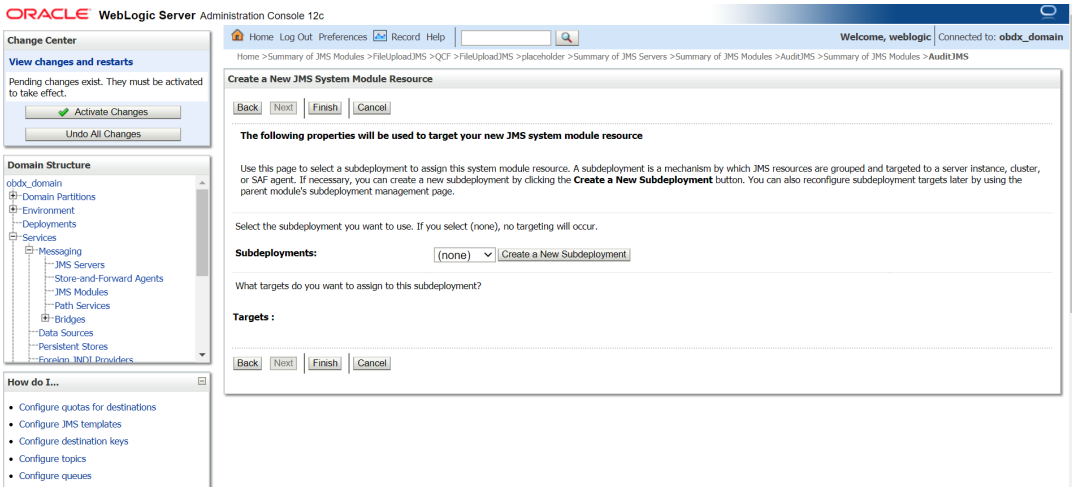


20. **Provide**
Name : `AUDITQCF`

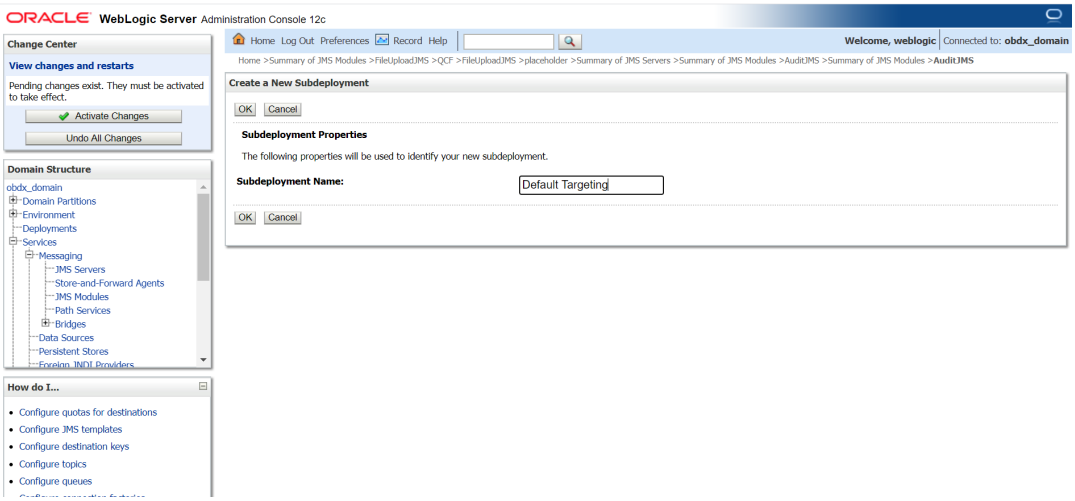
JNDI Name : `AUDITQCF`



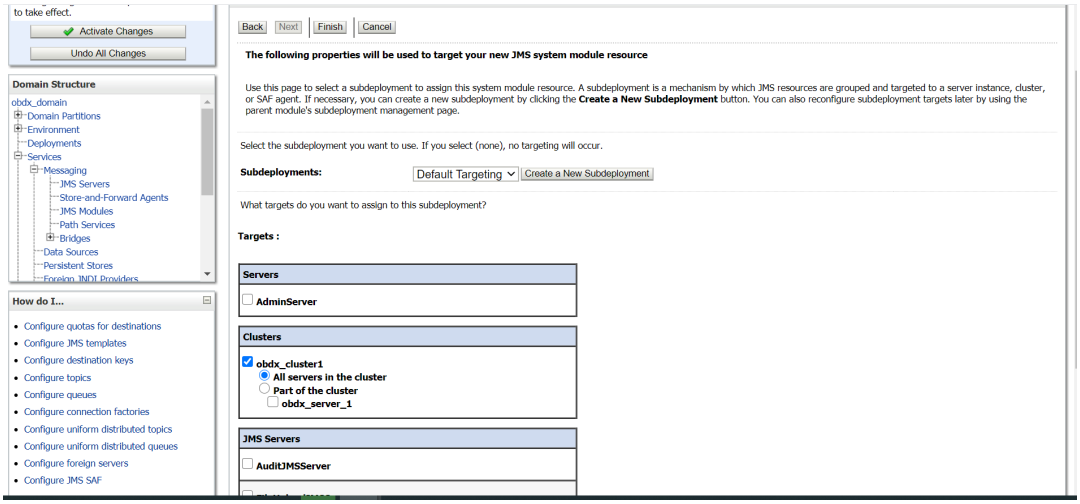
21. Click on Advanced Targeting.



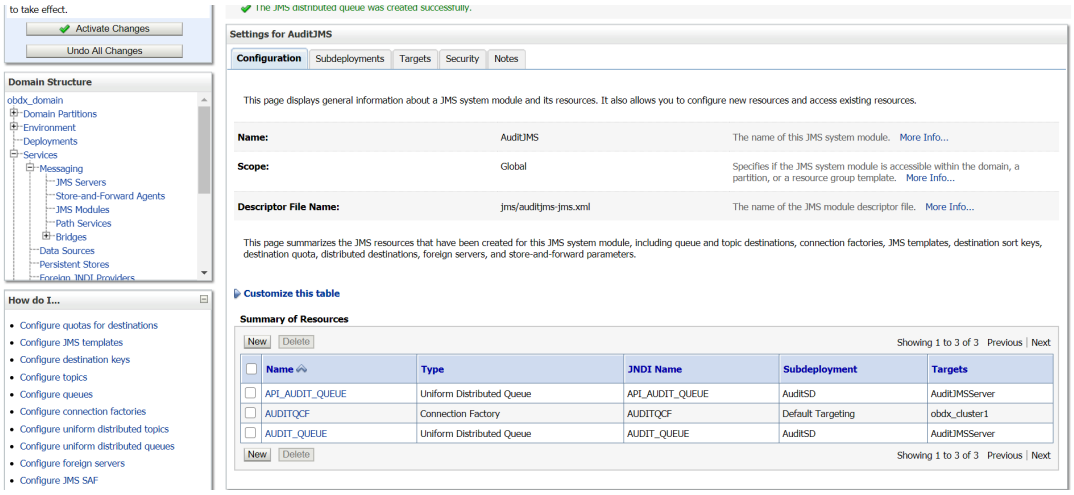
22. Click on Create a New Subdeployment.



23. Give Subdeployment Name as Default Targeting.

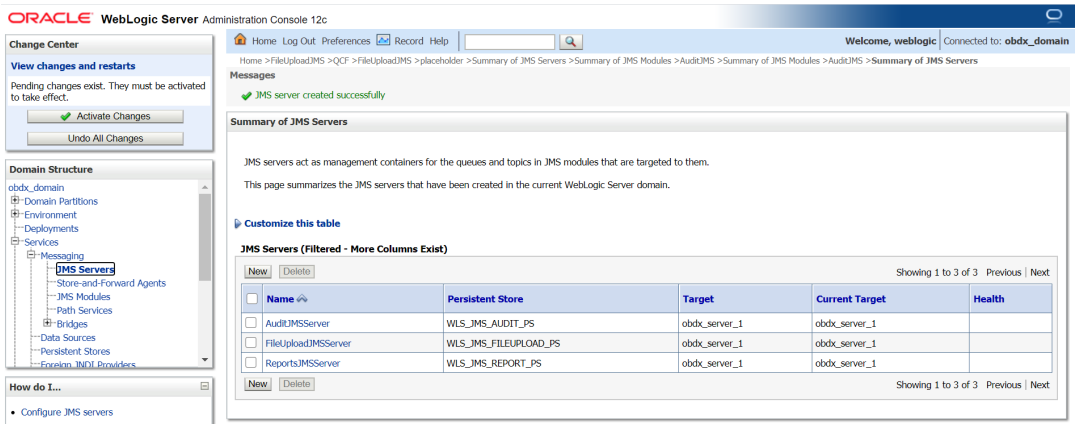


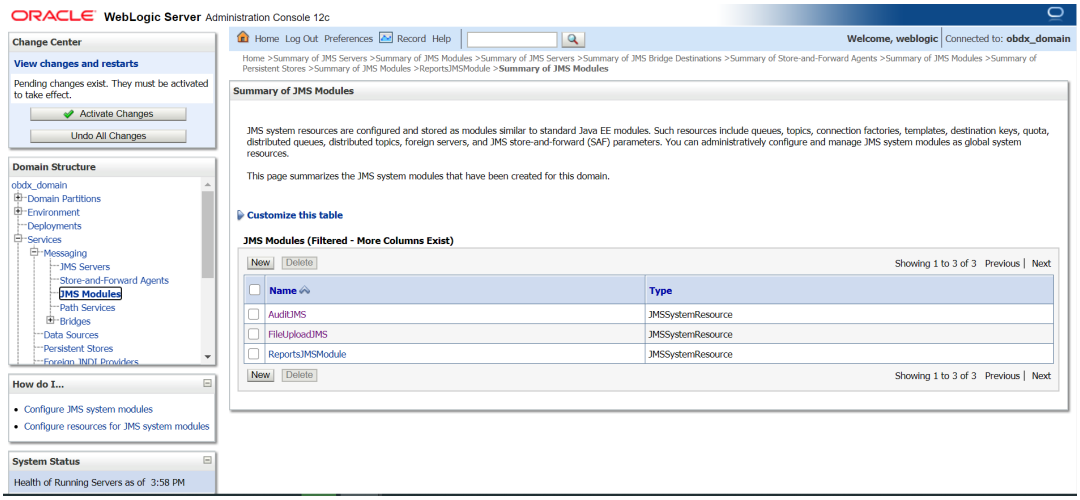
24. Under AuditJMS module Create Uniform Distrubuted Queue and connection Factory as show below in the screen shot.

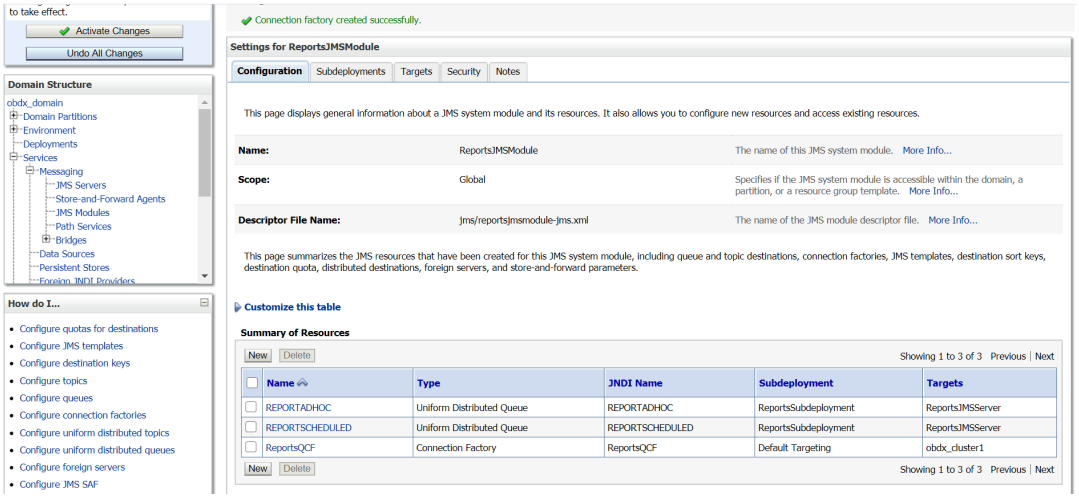


2.11 Creating ReportsJMSServer JMS Server

1. Similarly create ReportsJMSServer under JMS Server and ReportsJMSModule under JMS Module.







2. Under ReportsJMSModule create UniformDistributed Queue and connection factory as show above in the screen shot.
`REPORTADHOC` – Uniform Distributed Queue

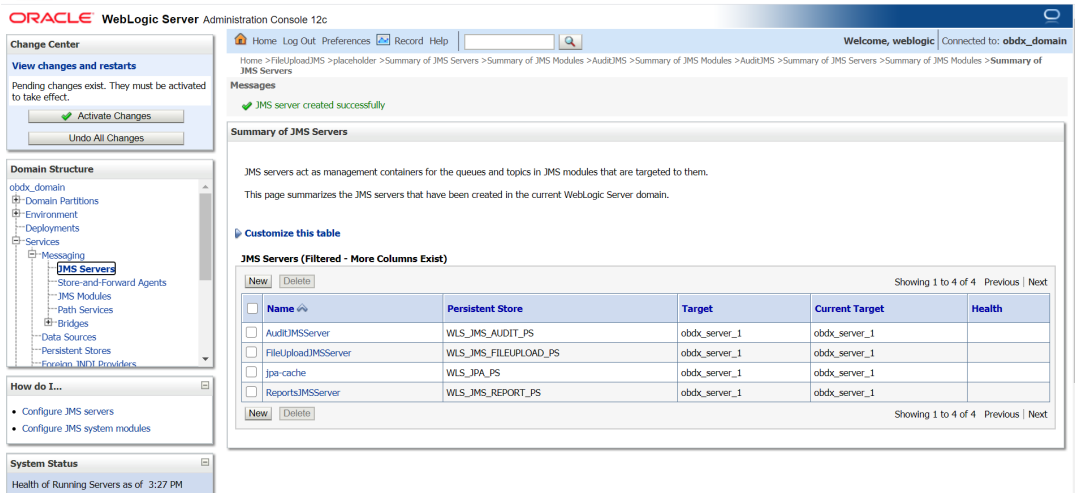
`REPORTSCHEDULED` - Uniform Distributed Queue

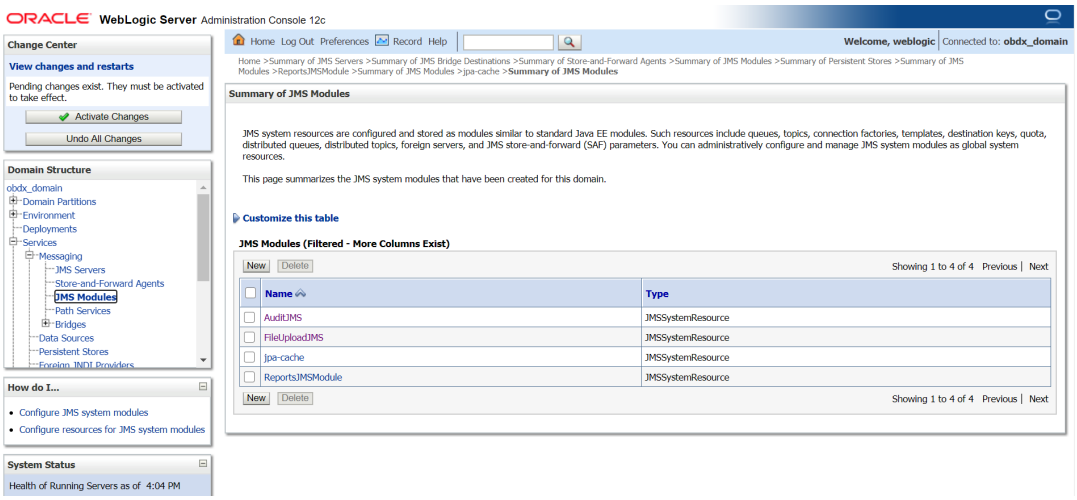
`ReportsQCF` – Connection Factory

2.12 Creating jpa-cache JMS Server

2.13 Creating WLS_JPA_PS FileStore

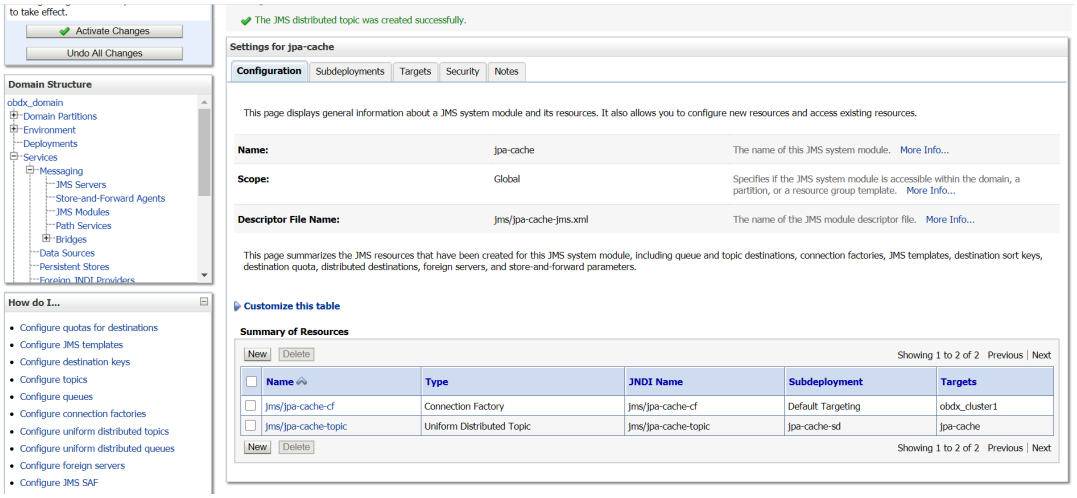
1. Create jpa-cache JMS server and jpa-cache JMS Module as show in below screen shot.





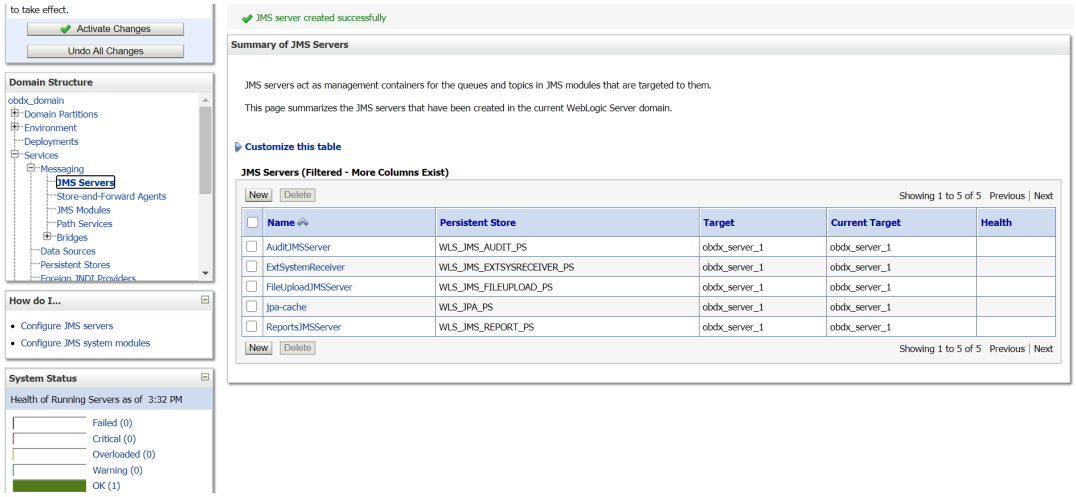
2. Under jpa-cache JMS Module create connection Factory and Uniform Distrbuted topic as shown in below screen shot.
Jms/jpa-cache-cf --- Connection Factory

Jms/jpa-cache-topic --- Uniform Distributed Topic

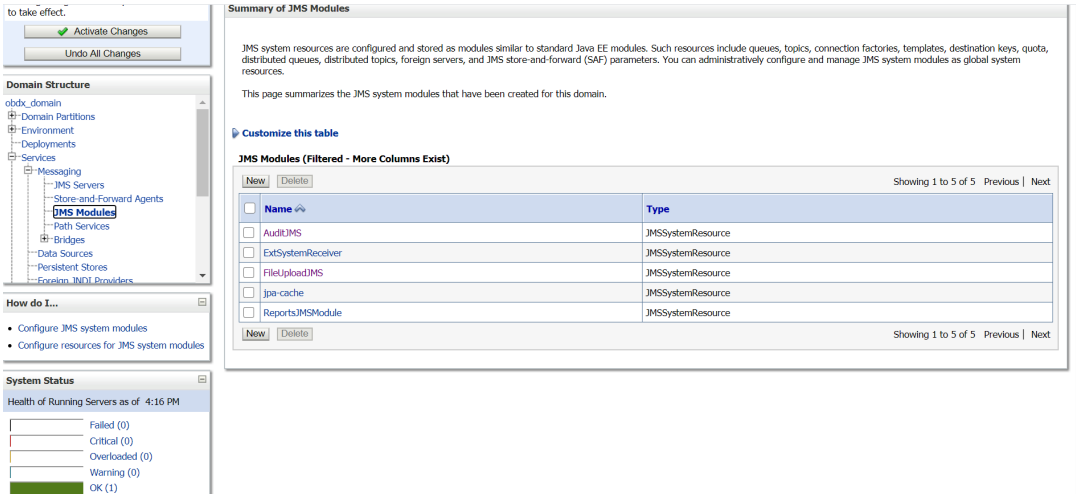


2.14 Creating ExtSystemReceiver JMS Server - WLS_JMS_EXTSYSRECEIVER_PS FileStore

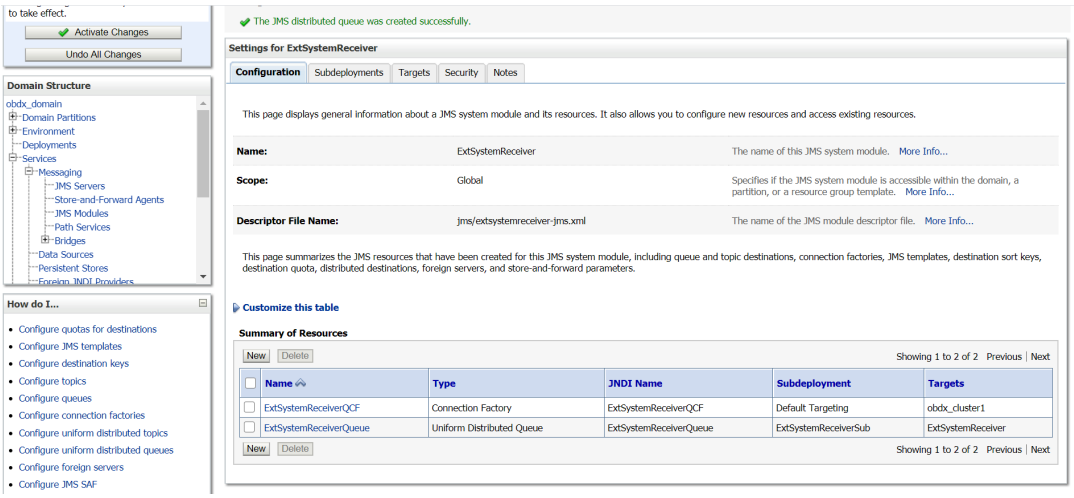
1. Create ExtSystemReceiver JMS Server Persistent store file store as WLS_JMS_EXTSYSRECEIVER_PS as show in below screen shot.



2. Create ExtSystemReceiver JMS Module as below.

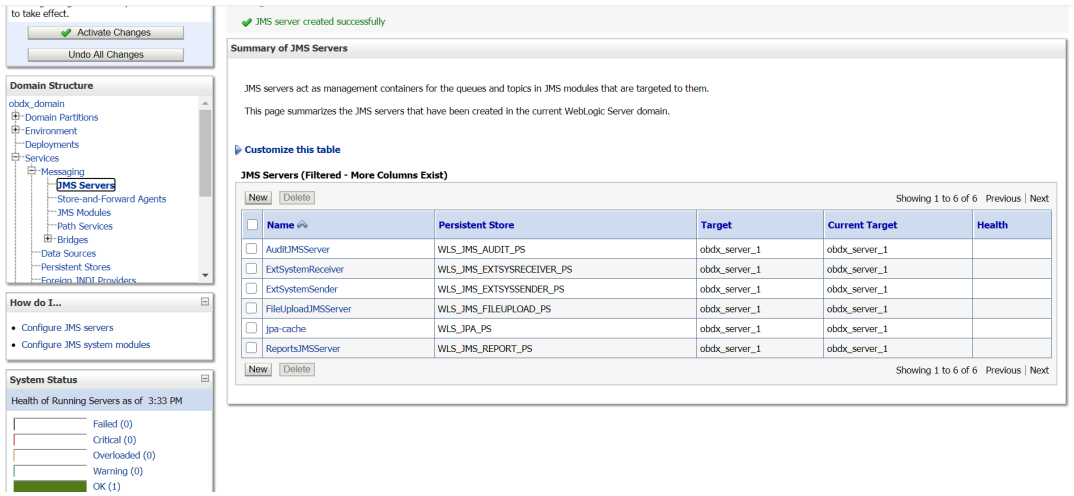


3. Create ExtSystemReceiverQCF – connection Factory and ExtSystemReceiverQueue – uniform Distributed Queue in ExtSystemReceiver JMS Module refer below screen shot.

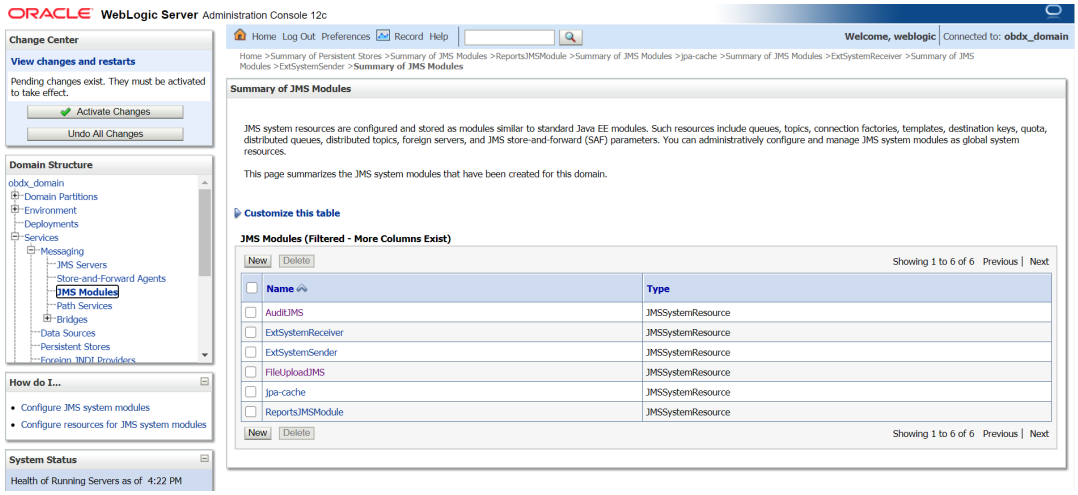


2.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS_JMS_EXTSYSSENDER_PS

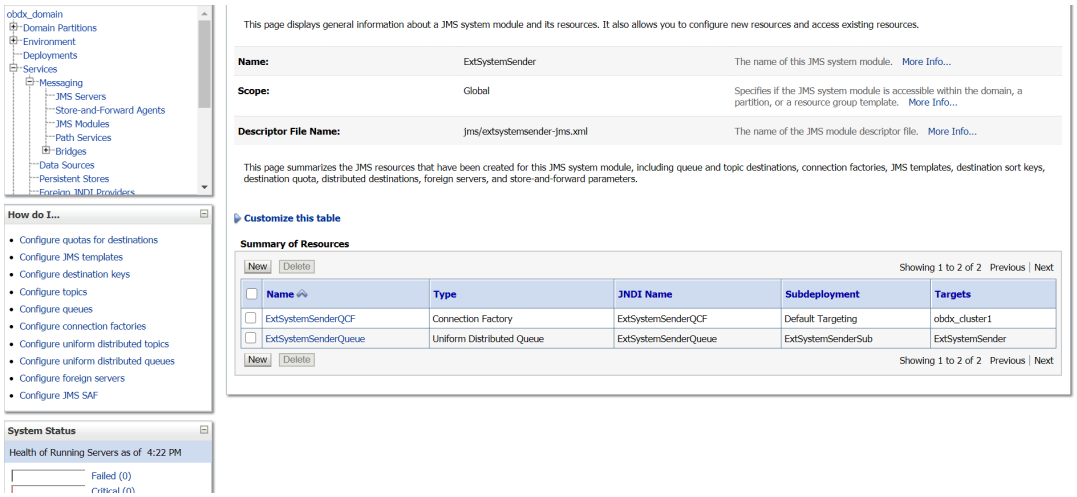
1. As show below create JMS Server ExtSystemSender.



2. Create ExtSystemSender JMS Module.

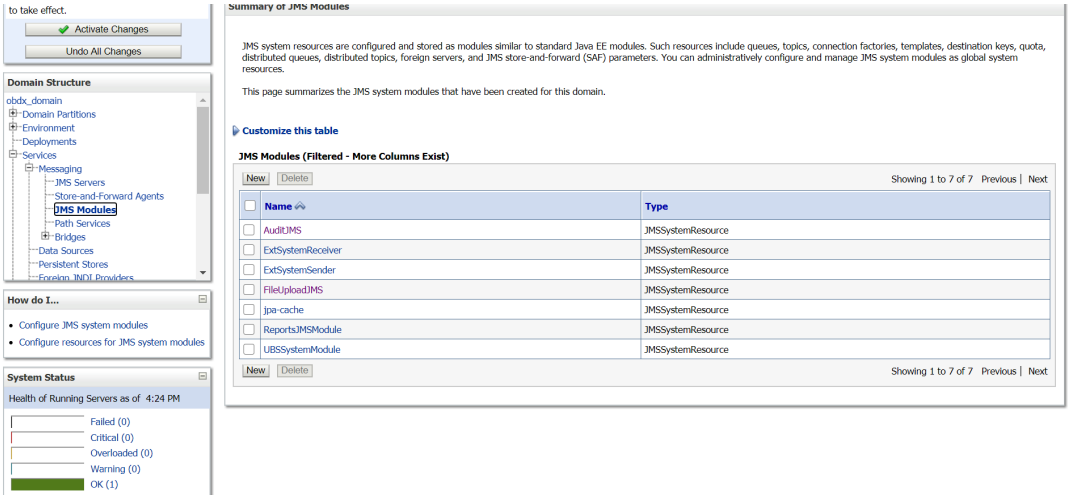


3. Under ExtSystemSender JMS Module create ExtSystemSenderQCF – connection Factory and ExtSystemSenderQueue – Uniform Distributed Queue as show below.

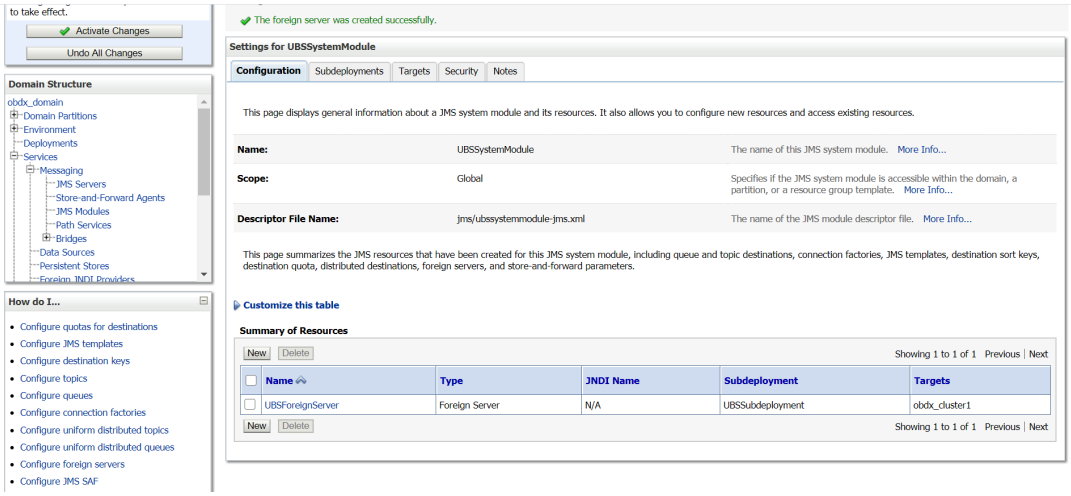


2.16 Creating UBSForeignServer JMS Server

1.

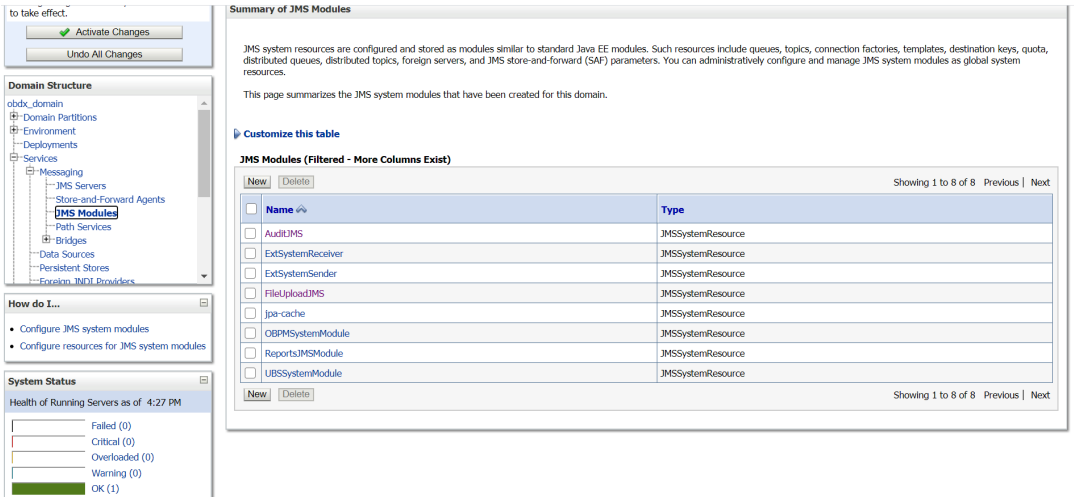


2. Under UBSSystemModule create UBSForeignServer – Foreign Server as shown below.

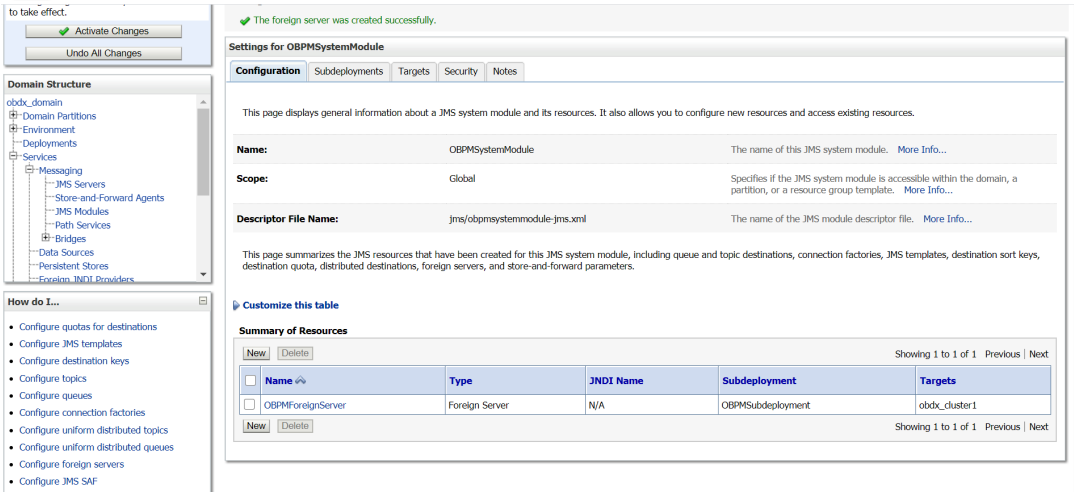


2.17 Creating OBPMForeignServer JMS Server

1. In JMSModule create OBPMSystemModule.



2. Under OBPMSystemModule create OBPMForeignServer – Foregin Server as show below in screen shot.



3
Deploying Applications

Deployment of Lib and Apps

```
Wars and Libs which are independent are present in path-
        OBDX_Installer\installables\OBDX\<Installation type>\<version>/
        app/components/common
Wars that are created on runtime will be available in
        path- OBDX_Installer/ OBDX_Installer/ExecInstances/<date>/app/wars.
Please refer below XML file for list of wars to be deployed.
<application name="digx-admin.war" displayedName="digx-admin"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="99"/>
<application name="digx-sms.war" displayedName="digx-sms"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="99"/>
<application name="digx-cms.war" displayedName="digx-cms"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-coherence.war" displayedName="digx-coherence"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="0"/>
<application name="digx-common.war" displayedName="digx-common"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="100"/>
<application name="digx-corporateloan.war" displayedName="digx-corporateloan"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-creditfacility.war" displayedName="digx-
creditfacility"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-edx.war" displayedName="digx-edx"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-extxfacesimulator.war" displayedName="digx-
extxfacesimulator"
        target="@wls_cluster_name@" location="@deploy_path@" type="common"
deployOrder="100"/>
<application name="digx-infra.war" displayedName="digx-infra"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="100"/>
<application name="digx-kafkanotification.war" displayedName="digx-
kafkanotification"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-liquiditymanagement.war" displayedName="digx-
liquiditymanagement"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
```

```
<application  name="digx-loanapplication.war" displayedName="digx-
loanapplication"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
        deployOrder="100"/>
<application name="digx-payments.war" displayedName="digx-payments"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-pfm.war" displayedName="digx-pfm"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-processmanagement.war" displayedName="digx-
processmanagement"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-retail.war" displayedName="digx-retail"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-scf.war" displayedName="digx-scf"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<application name="digx-scfcm.war" displayedName="digx-scfcm"
target="@wls_cluster_name@"
        location="@deploy_path@" type="" deployOrder="100"/>
<library name="digx-shared-libs.war" displayedName="digx-shared-libs"
        target="@wls_cluster_name@,AdminServer" location="@deploy_path@"
type="common" deployOrder="0"/>
<application name="digx-tradefinance.war" displayedName="digx-tradefinance"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-ukob.war" displayedName="digx-ukob"
target="@wls_cluster_name@"
        location="@deploy_path@" type="common" deployOrder="100"/>
<application name="digx-virtual-account.war" displayedName="digx-virtual-
account"
        target="@wls_cluster_name@" location="@deploy_path@" type=""
deployOrder="100"/>
<application name="digx-webauthn.war" displayedName="digx-webauthn"
target="@wls_cluster_name@"
        location="@installerhome@/installables/app/components/common" type=""
deployOrder="100"/>
<library name="digx-lzn-libs.war" displayedName="digx-lzn-libs"
target="@wls_cluster_name@"
        location="@installerhome@/installables/app/components/common" type=""
deployOrder="100"/>
```

4
Configured jps-config.xml

Update the jps-config.xml

Edit `$DOMAIN_HOME/config/fmwconfig/jps-config.xml` file and add following entries.

1. Find <serviceProviders> tag in the file, add below serviceProvider between <serviceProviders></serviceProviders>.

```
<serviceProvider type="IDENTITY_STORE" name="custom.provider"
class="oracle.security.jps.internal.idstore.generic.GenericIdentityStorePro
vider">
<description>Custom IdStore Provider</description></serviceProvider>
```

2. Find <serviceInstances> tag in the file, add below serviceInstances between <serviceInstances></serviceInstances>.

```
<serviceInstance name="idstore.custom" provider="custom.provider"
location="dumb">
<description>Custom Identity Store Service Instance</description>
<property name="idstore.type" value="CUSTOM"/>
<property name="ADF_IM_FACTORY_CLASS"
value="com.ofss.sms.dbAuthenticator.providers.db.DBIdentityStoreFactory"/>

<property name="DATASOURCE_NAME" value="DIGX"/>
</serviceInstance>
```

3. Find <jpsContext name="default"> tag in the file, add below serviceInstanceRef between <jpsContext name="default"></jpsContext>.

```
<serviceInstanceRef ref="idstore.custom"/>
```

5

List of Topics

This user manual is organized as follows:

Table 5-1 List of Topics

Topics	Description
Preface	This topic provides information on the introduction, intended audience, list of topics, and acronyms covered in this guide.
Manual OBAPI installation	This topic provides a step to install OBDX database manually.
WEBLOGIC Setup and Configuration	This topic provides information about the creations of weblogic domain , managed server, creations of cluster, configuration of node manager.
Deploying Applications	This topic describes deployment of Lib and Apps.
Configured jps-config.xml	This topic explains the configuration of jps-config.xml.

Index

C

Configured jps-config.xml, *4-1*
Create JMS Server and JMS Module, *2-17*
Creating B1A1 Data Source, *2-14*
Creating BATCH Data Source, *2-8*
Creating DIGX Data Source, *2-2*
Creating ExtSystemReceiver JMS Server - WLS_JMS_EXTSYSRECEIVER_PS FileStore, *2-42*
Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS_JMS_EXTSYSSENDER_PS, *2-43*
Creating NONXA Data Source, *2-5*
Creating OBPMForeignServer JMS Server, *2-45*
Creating ReportsJMSServer JMS Server, *2-39*
Creating SYSCONFIG Data Source, *2-11*
Creating UBSForeignServer JMS Server, *2-44*
Creating WLS_JMS_REPORT_PS FileStore, *2-28*
Creating WLS_JPA_PS FileStore, *2-40*

D

Deploying Applications, *3-1*

M

Manual OBAPI Installation, *1-1*

P

Policy Seeding, *1-1*

S

Setting Domain JTA Transaction Timeout, *2-1*

W

WEBLOGIC Setup and Configuration, *2-1*