

# **Oracle Banking APIs**

**Installation Guide- Non-Linux  
Platforms  
Release 19.2.0.0.0**

**Part No. F25153-01**

**December 2019**

**ORACLE®**

Installation Guide- Non-Linux Platforms

December 2019

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

[www.oracle.com/financialservices/](http://www.oracle.com/financialservices/)

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

# Table of Contents

<b>1. Preface</b> .....	<b>1</b>
1.1 Intended Audience .....	1
1.2 Documentation Accessibility .....	1
1.3 Access to OFSS Support .....	1
1.4 Structure .....	1
1.5 Related Information Sources.....	1
<b>2. Manual OBAPI installation</b> .....	<b>2</b>
2.1 Create OBAPI Tablespace (file obapi_create_tablespace.sql).....	2
2.2 Create Audit tablespace (file obapi_audit_create_tablespace.sql).....	2
2.3 Create user (file obapi_create_user.sql).....	2
2.4 Create role (file obapi_create_role.sql).....	3
2.5 Grants Execution (file clip_user_grants.sql) .....	3
2.6 Files execution in sequences on above schema (ex. OBAPI_\${POST_FIX}).....	3
2.7 OBPM Database Installation (OBPM Favor).....	4
2.8 Tablespace Creation (file obpm_create_tablespace.sql).....	4
2.9 CREATE BIGFILE TABLESPACE TBS_\${EHMS_SCHEMA_NAME } .....	4
2.10 User Creation (file obpm_create_user.sql) .....	4
2.11 Create role (file obpm_create_role.sql).....	5
2.12 2.12 CREATE ROLE ROLE_\${ EHMS_SCHEMA_NAME } NOT IDENTIFIED; .....	5
2.13 2.13 Grants Execitions .....	5
2.14 2.14 Scripts Execution .....	5
2.15 2.15 Policy Seeding .....	6

<b>3. WEBLOGIC Setup and Configuration.....</b>	<b>7</b>
3.1 Setting Domain JTA Transaction timeout .....	7
3.2 Creating DIGX data source .....	8
3.3 Creating NONXA data source .....	11
3.4 Creating BATCH data source.....	13
3.5 Creating SYSCONFIG data source.....	16
3.6 Creating B1A1 data source .....	19
3.7 Create JMS server and JMS Module .....	22
3.8 Creating WLS_JMS_AUDIT_PS FileStore .....	33
3.9 Creating AuditJMSServer JMS Server.....	33
3.10 Creating WLS_JMS_REPORT_PS FileStore .....	33
3.11 Creating ReportsJMSServer JMS Server .....	44
3.12 Creating jpa-cache JMS Server .....	45
3.13 Creating WLS_JPA_PS FileStore .....	45
3.14 Creating ExtSystemReceiver JMS Server -- WLS_JMS_EXTSYSRECEIVER_PS FileStore....	46
3.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS_JMS_EXTSYSENDER_PS.....	48
3.16 Creating UBSForeignServer JMS Server .....	49
3.17 Creating OBPMForeignServer JMS Server .....	50
<b>4. Deploying Applications .....</b>	<b>51</b>
<b>5. Configured jps-config.xml.....</b>	<b>52</b>
<b>6. Configured Providers.....</b>	<b>53</b>

# 1. Preface

## 1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

## 1.2 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>.

## 1.3 Access to OFSS Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## 1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

## 1.5 Related Information Sources

For more information on Oracle Banking APIs Release 19.2.0.0.0, refer to the following documents:

- User Manual Oracle Banking APIs Installation Guide

## 2. Manual OBAPI installation

OBAPI Database Installation with OBPM FLAVOR

Create required OBAPI tablespace and user in below sequence.

### 2.1 Create OBAPI Tablespace (file obapi\_create\_tablespace.sql)

Execute the file available @ \${OBAPI\_INSTALLER}/installables/db/OBAPI/obapi\_create\_tablespace.sql

Update the datafile path and tablespace name and execute the file

Example: -

Refer installer.properties file variable POST\_FIX and replace in the below command.

```
CREATE BIGFILE TABLESPACE OBAPI_${POST_FIX} DATAFILE
'${DATAFILE_PATH}/OBAPI_${POST_FIX}.dbf'
    SIZE 500M
    AUTOEXTEND ON NEXT 100M
    LOGGING
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;
```

### 2.2 Create Audit tablespace (file obapi\_audit\_create\_tablespace.sql)

Execute the file available @ \${OBAPI\_INSTALLER}/installables/db/OBAPI/obapi\_audit\_create\_tablespace.sql

Example :-

Refer installer.properties file variable POST\_FIX and replace in the below command

```
CREATE BIGFILE TABLESPACE OBAPI_AUDIT_${POST_FIX}
    DATAFILE '${DATAFILE_PATH}/OBAPI_AUDIT_${POST_FIX}.dbf'
    SIZE 500M
    AUTOEXTEND ON NEXT 100M
    LOGGING
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;
```

### 2.3 Create user (file obapi\_create\_user.sql)

Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBAPI/obapi\_create\_user.sql

Example: -

Refer installer.properties file variable POST\_FIX and replace in the below command

```
create user OBAPI_${POST_FIX} identified by welcome1;
```

```
alter user OBAPI_${ POST_FIX} default tablespace OBAPI_${ POST_FIX};
alter user OBAPI_${ POST_FIX} temporary tablespace temp;
alter user OBAPI_${ POST_FIX} quota unlimited on OBAPI_${ POST_FIX} ;
alter user OBAPI_${ POST_FIX} quota unlimited on OBAPI_AUDIT_${ POST_FIX} ;
```

## 2.4 Create role (file obapi\_create\_role.sql)

Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBAPI/obapi\_create\_role.sql

Example:-

Refer installer.properties file variable POST\_FIX and replace in the below command

```
CREATE ROLE OBAPI_ROLE_${POST_FIX} NOT IDENTIFIED;
grant CONNECT, CREATE SESSION, CREATE TABLE, CREATE SEQUENCE,CREATE
TRIGGER, CREATE DATABASE LINK,CREATE VIEW, CREATE PROCEDURE, CREATE
SYNONYM, CREATE TYPE,CREATE JOB to OBAPI_ROLE_${POST_FIX};
grant OBAPI_ROLE_${POST_FIX} to OBAPI_${POST_FIX};
```

## 2.5 Grants Execution (file clip\_user\_grants.sql)

Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBAPI/clip\_user\_grants.sql

Example:-

Refer installer.properties file variable POST\_FIX and replace in the below command

```
grant create any sequence to OBAPI_${POST_FIX};
grant execute on DBMS_LOCK to OBAPI_${POST_FIX};
```

## 2.6 Files execution in sequences on above schema (ex. OBAPI\_\${POST\_FIX})

- clip\_master\_script.sql
- clip\_constraints.sql
- clip\_seeds\_executable.sql
- clip\_master\_generic\_rest\_script.sql

\*\*\*\*\* SUCCESSFULLY installed OBAPI database \*\*\*\*\*

## 2.7 OBPM Database Installation (OBPM Favor)

Create required OBAPI tablespace and user in below sequence

## 2.8 Tablespace Creation (file obpm\_create\_tablespace.sql)

Execute the file available @ \${OBAPI\_INSTALLER}  
/installables/db/OBPM/obpm\_create\_tablespace.sql

Example:-

Refer installer.properties file variable EHMS\_SCHEMA\_NAME and replace in the below command

## 2.9 CREATE BIGFILE TABLESPACE TBS\_\${EHMS\_SCHEMA\_NAME}

```
DATAFILE '${DATAFILE_PATH}/TBS_${EHMS_SCHEMA_NAME}.dbf'
SIZE 500M
AUTOEXTEND ON NEXT 100M
LOGGING
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
```

## 2.10 User Creation (file obpm\_create\_user.sql)

Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/obpm\_create\_user.sql

Example:-

Refer installer.properties file variable EHMS\_SCHEMA\_NAME and replace in the below command

```
create user ${ EHMS_SCHEMA_NAME } identified by welcome1;
alter user ${ EHMS_SCHEMA_NAME } default tablespace TBS_${ EHMS_SCHEMA_NAME };
alter user ${ EHMS_SCHEMA_NAME } temporary tablespace temp;
alter user ${ EHMS_SCHEMA_NAME } quota unlimited on TBS_${ EHMS_SCHEMA_NAME };
```



## 2.11 Create role (file obpm\_create\_role.sql)

Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/obpm\_create\_role.sql

Example:-

Refer installer.properties file variable EHMS\_SCHEMA\_NAME and replace in the below command

## 2.12 2.12 CREATE ROLE ROLE\_\${ EHMS\_SCHEMA\_NAME } NOT IDENTIFIED;

```
grant CONNECT, CREATE SESSION, CREATE TABLE, CREATE SEQUENCE,CREATE
TRIGGER, CREATE DATABASE LINK,CREATE VIEW, CREATE PROCEDURE, CREATE
SYNONYM, CREATE TYPE,CREATE JOB to ROLE_${ EHMS_SCHEMA_NAME };
```

```
grant ROLE_${ EHMS_SCHEMA_NAME } to ${ EHMS_SCHEMA_NAME } ;
```

## 2.13 2.13 Grants Execitions

Replace \$\$schema with \${ EHMS\_SCHEMA\_NAME } in the below files

- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/FCUBS\_GR\_PRIV.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/FCOBPM\_GR\_PRIV.sql

## 2.14 2.14 Scripts Execution

- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/table-scripts.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/ubs\_object\_scripts.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/obpm\_object\_scripts.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/execute-seeds.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/obpm-seeds.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/DIGX\_FW\_CONFIG\_ALL\_O.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/DIGX\_FW\_ABOUT\_OBPM.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/DIGX\_FW\_CONFIG\_VAR\_B.sql
- Execute the file available @ \${OBAPI\_INSTALLER} /installables/db/OBPM/DIGX\_FW\_CONFIG\_UBS\_ALL\_O.sql

## 2.15 2.15 Policy Seeding

TEMP\_PATH=Temporary Path

```
cp ${OBAPI_INSTALLER}/installables/policies/Entitlement_log4j.properties to
TEMP_PATH/db/Entitlement_log4j.properties
```

```
cp ${OBAPI_INSTALLER}/installables/policies/Task_log4j.properties to
TEMP_PATH/db/Task_log4j.properties
```

```
cp ${OBAPI_INSTALLER}/installables/policies/Dashboard_seed_log4j.properties to
TEMP_PATH/db/Dashboard_seed_log4j.properties
```

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}/installables/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_DATAB
ASE_SID'
```

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

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

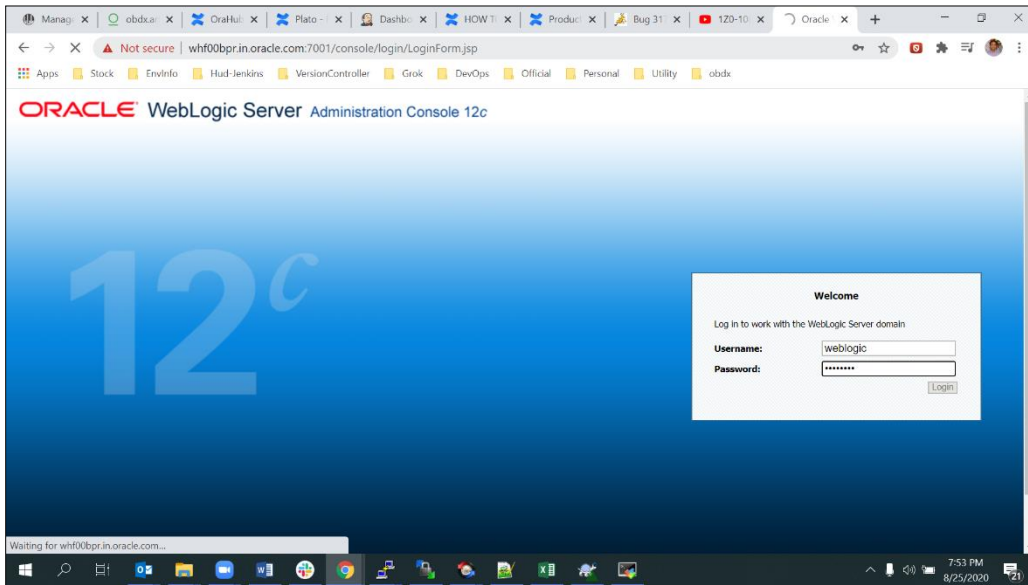
Home

### 3. WEBLOGIC Setup and Configuration

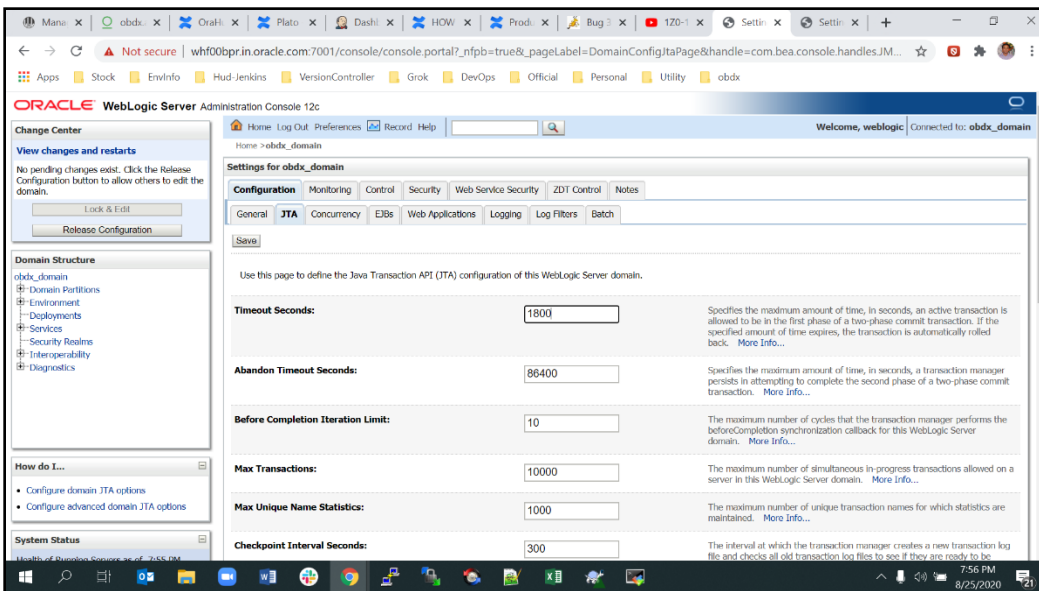
Assuming that rcu , weblogic domain created , managed server, cluster created , node manager configured.

#### 3.1 Setting Domain JTA Transaction timeout

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



2. click on DOMAIN\_NAME → JTA → set Timeout Seconds to 1800 → click on save → Activate changes



## 3.2 Creating DIGX data source

1. Navigate to Data Source → click on new → Provide details and click on finish.

The screenshot shows the 'Create a New JDBC Data Source' wizard in the Oracle WebLogic Administration Console. The wizard is titled 'Create a New JDBC Data Source' and has buttons for 'Back', 'Next', 'Finish', and 'Cancel'. The 'JDBC Data Source Properties' section contains the following fields:

- Name:** DIGX
- Scope:** Global
- JNDI Name:** DIGX
- Database Type:** Oracle

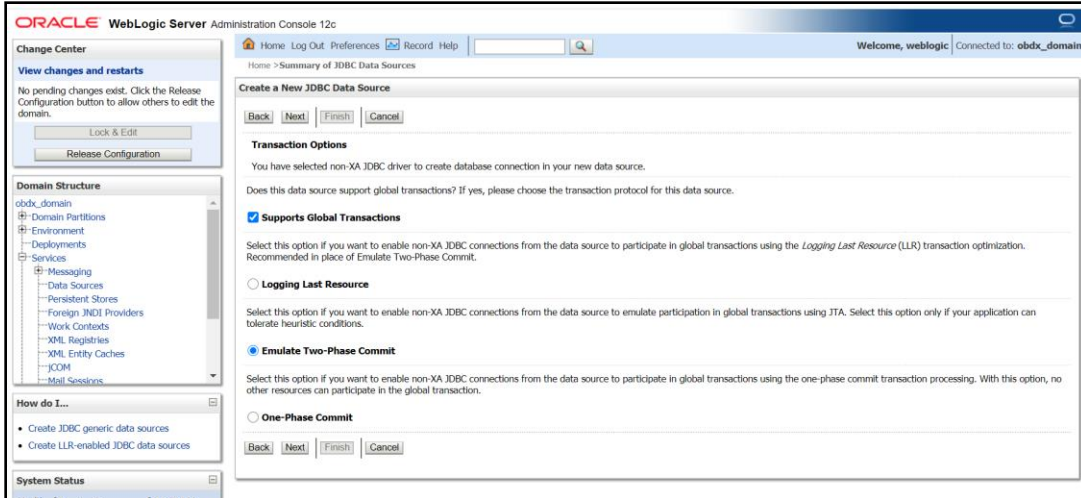
2. **Name:** - DIGX

**JNDI Name:** - DIGX

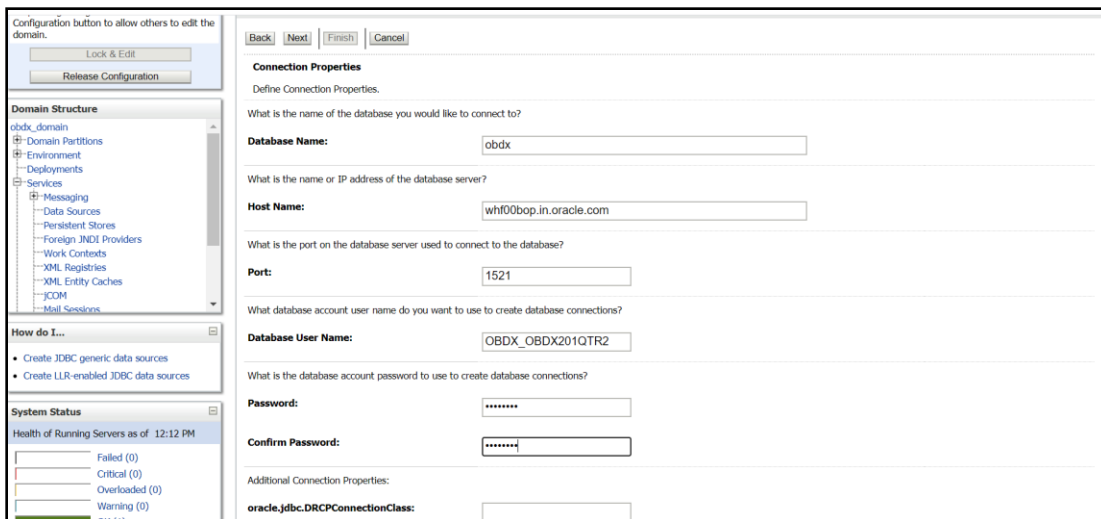
The screenshot shows the 'Create a New JDBC Data Source' wizard in the Oracle WebLogic Administration Console. The wizard is titled 'Create a New JDBC Data Source' and has buttons for 'Back', 'Next', 'Finish', and 'Cancel'. The 'JDBC Data Source Properties' section contains the following fields:

- Database Type:** Oracle
- Database Driver:** Oracle's Driver (Thin) for Instance connections; Versions: Any

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



#### 4. Select Emulate Two-Phase Commit



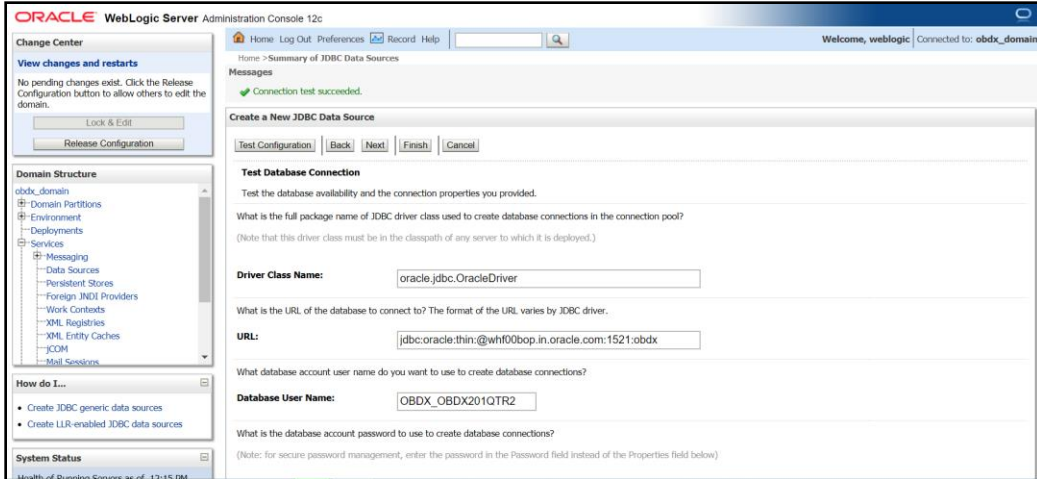
#### 5. Provide

**Database Name:** - Database SID

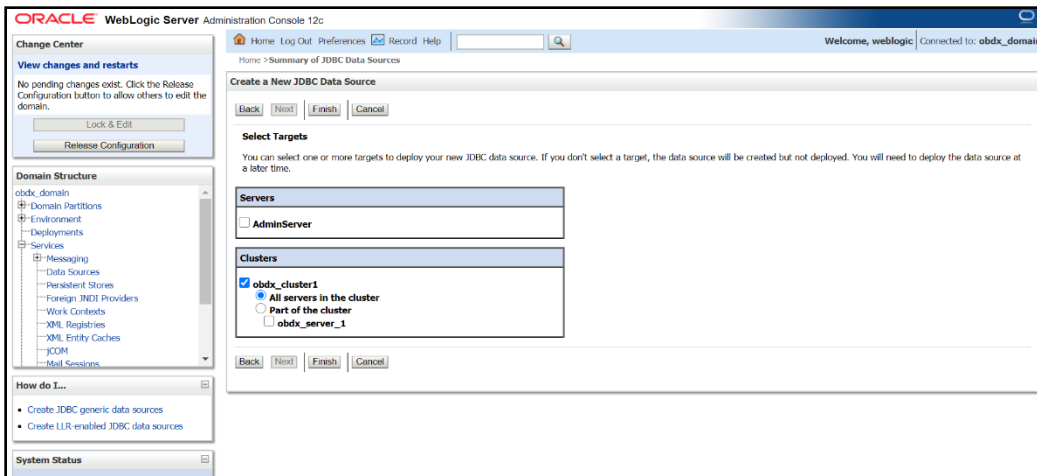
**Host Name:** - Database hostname

**Port:** - Database port Number

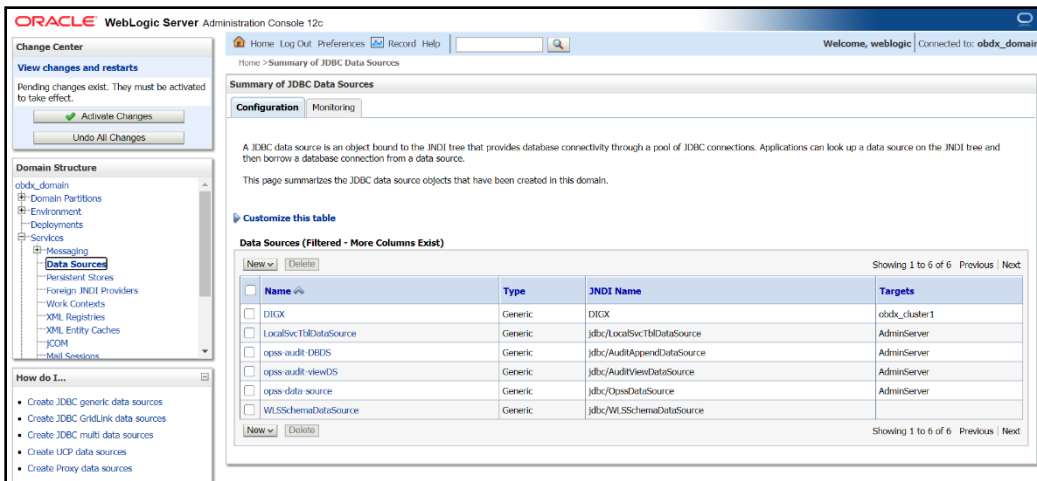
**Database user Name:** - OBAPI\_\${POST\_FIX}



## 6. Test Configuration

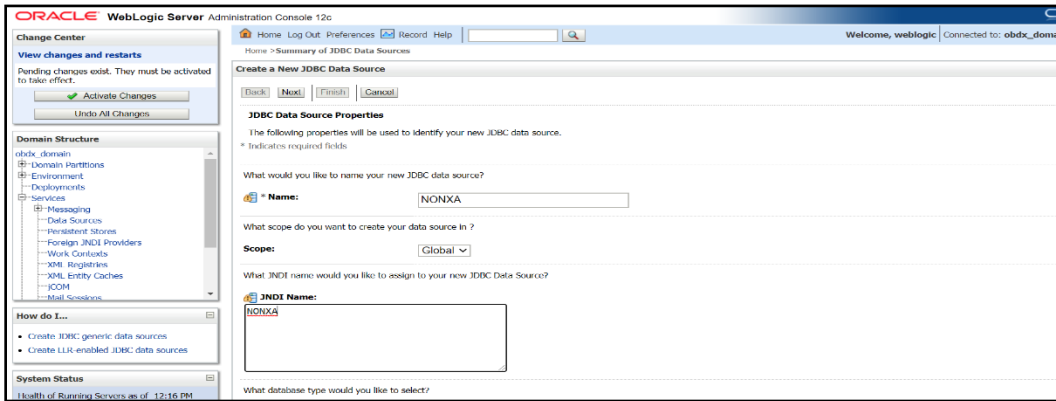


## 7. Target to cluster



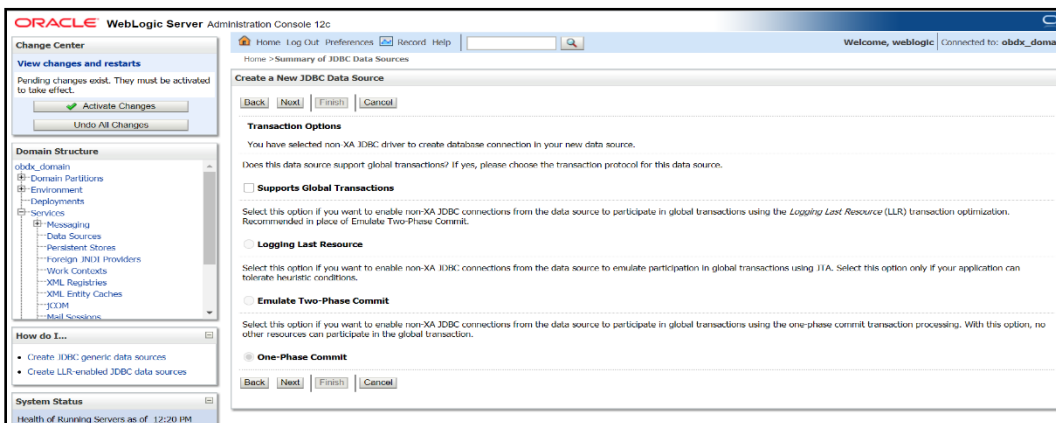
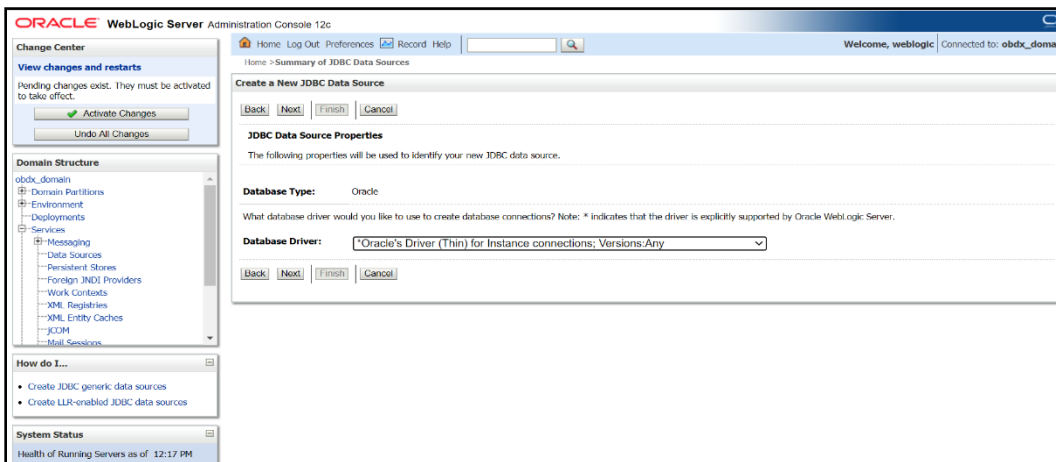
### 3.3 Creating NONXA data source

1. Navigate to Data Source → click on new → Provide details and click on finish



2. Name :- NONXA

JNDI Name :- NONXA



3. Click Next

#### 4. Provide

**Database Name:** - Database SID

**Host Name:** - Database hostname

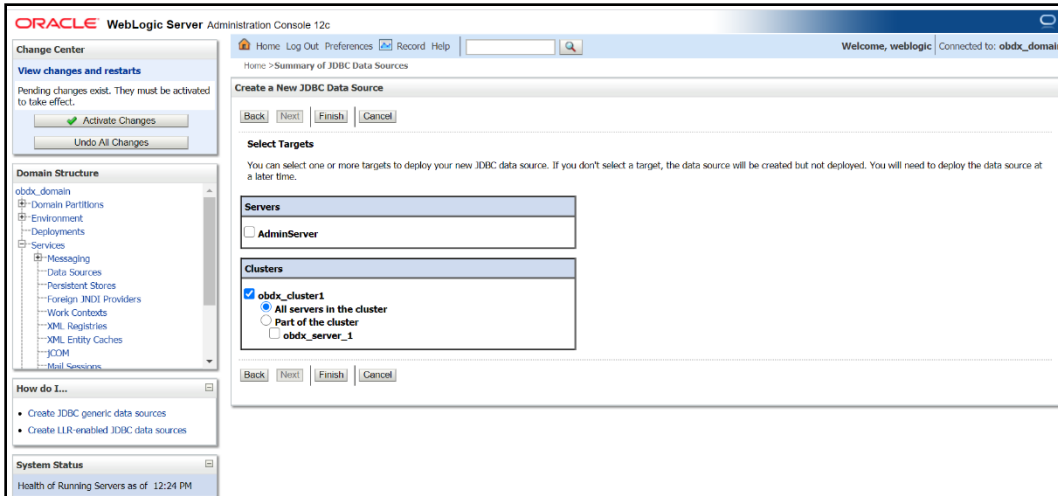
**Port:** - Database port Number

**Database user Name:** - OBAPI\_\${POST\_FIX}

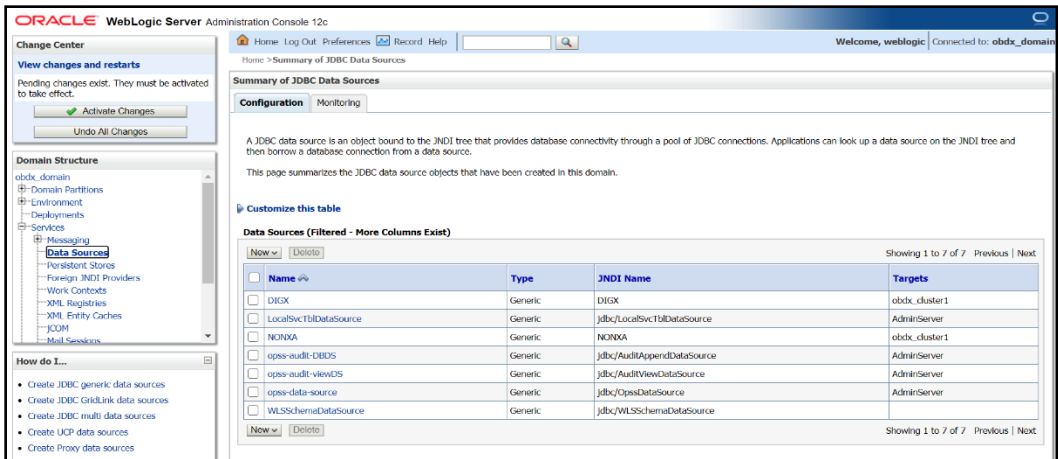
**Password:-** Database user password

#### 5. Test Configuration

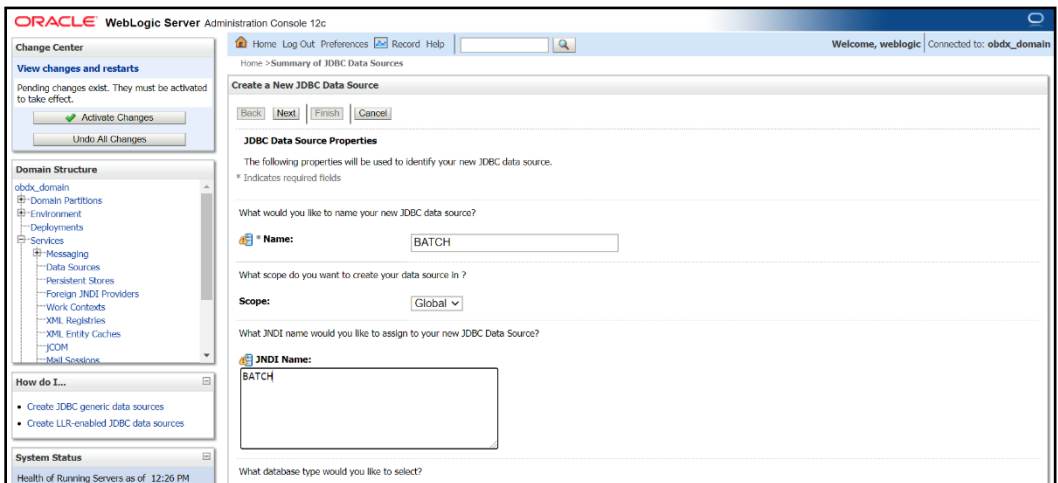




6. Select target as cluster --> Finish

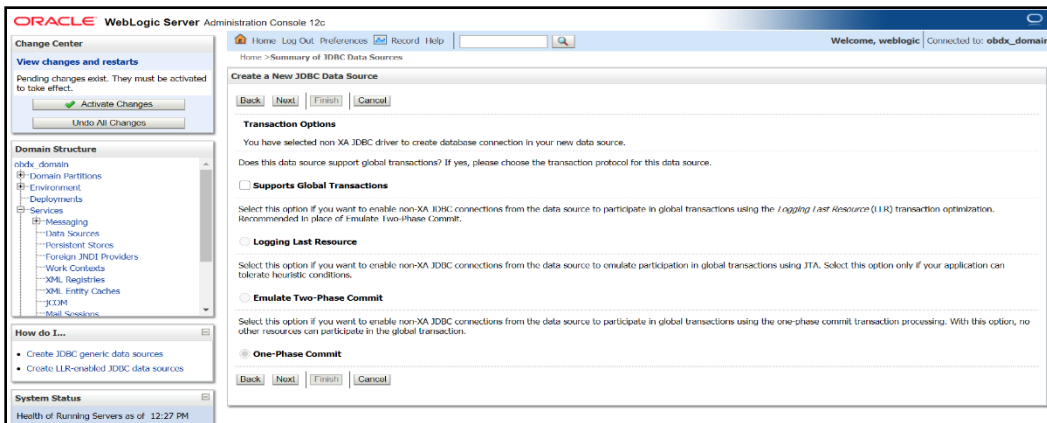
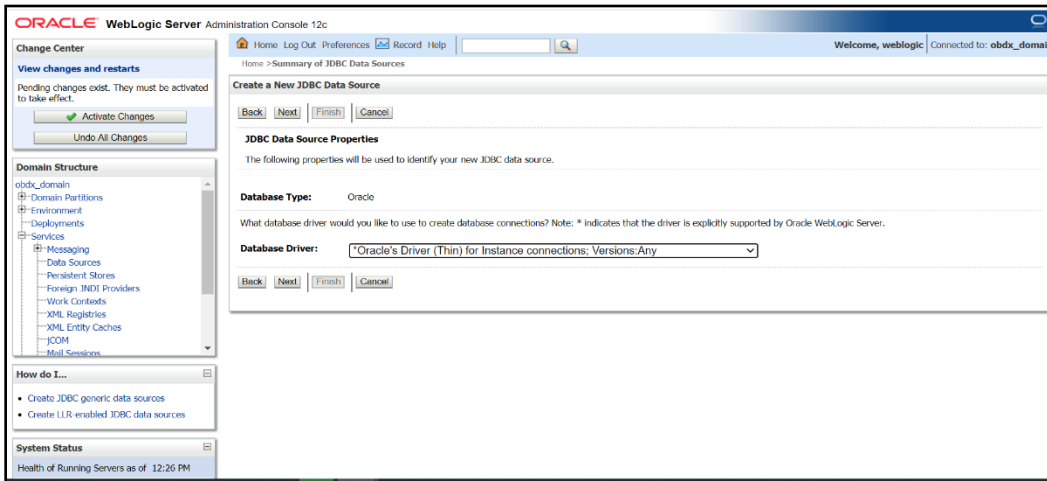


### 3.4 Creating BATCH data source

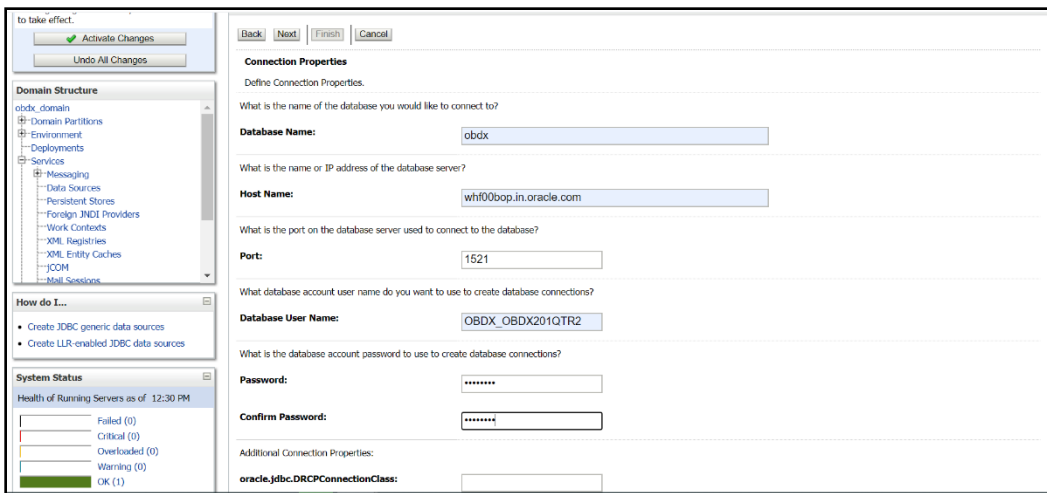


1. Name :- BATCH

JNDI Name :- BATCH



2. Click Next



### 3. Provide

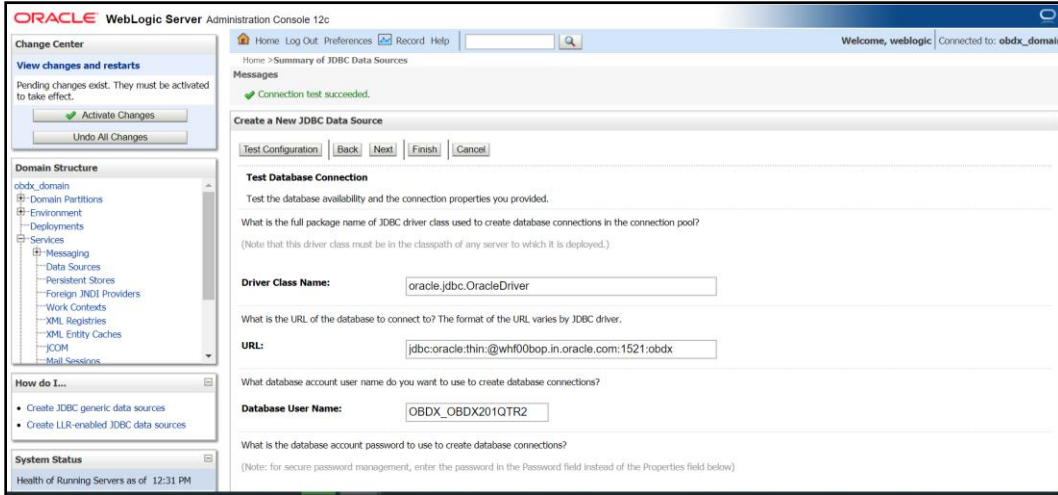
**Database Name:** - Database SID

**Host Name:** - Database hostname

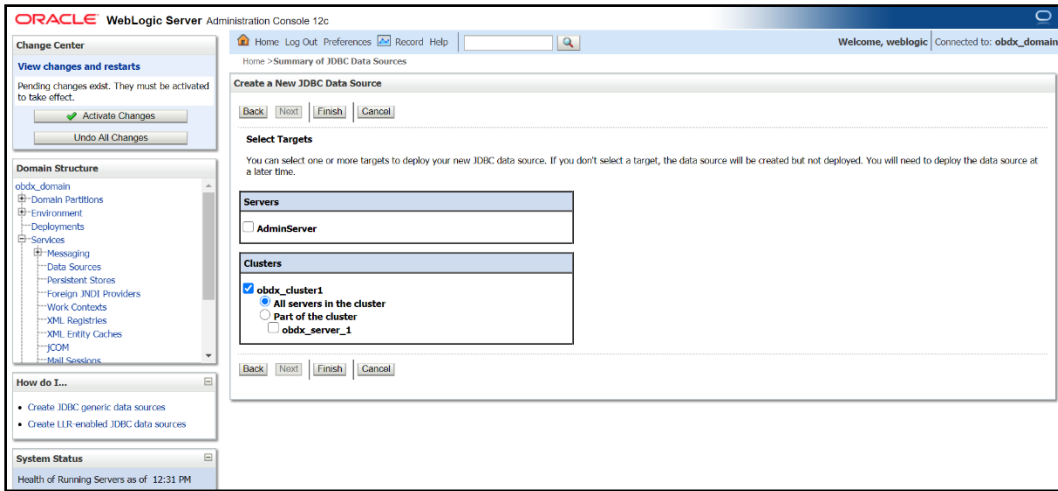
**Port:** - Database port Number

**Database user Name:** - OBAPI\_\${POST\_FIX}

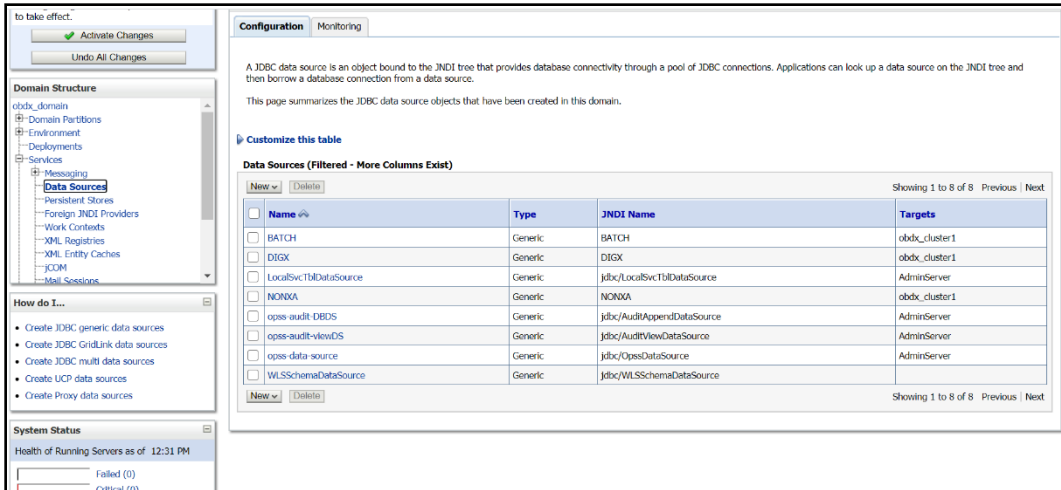
**Password:-** Database user password



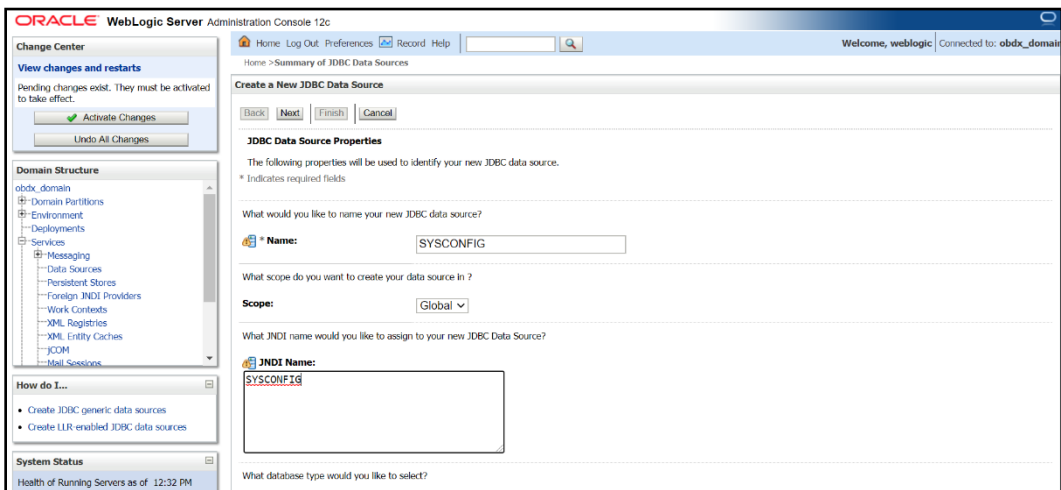
### 4. Test Configuration



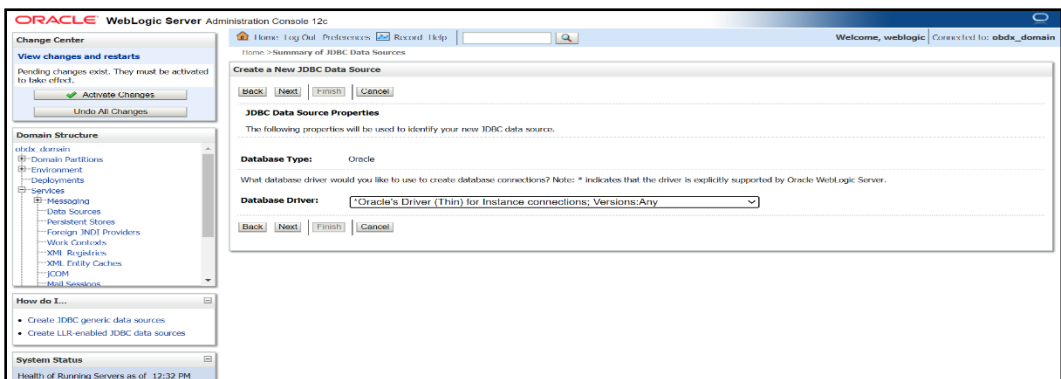
### 5. Target Cluster and click on Finish

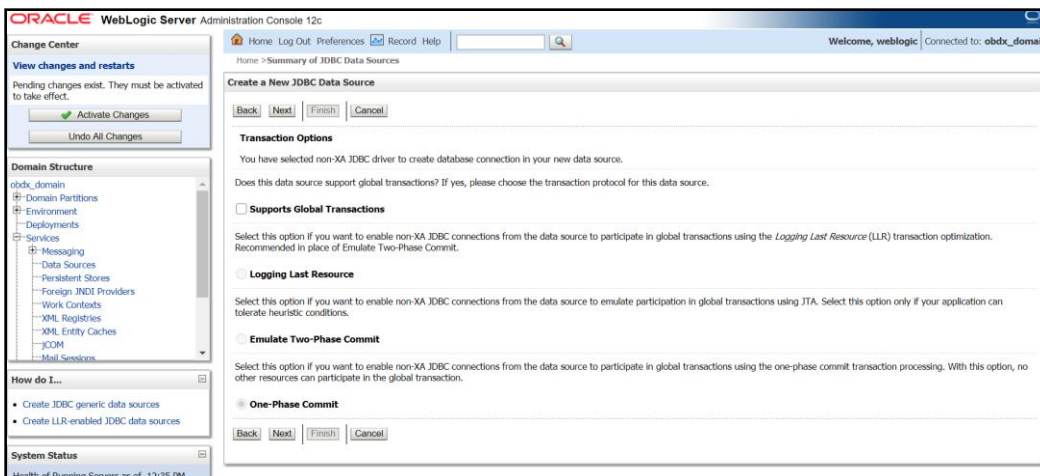


### 3.5 Creating SYSCONFIG data source

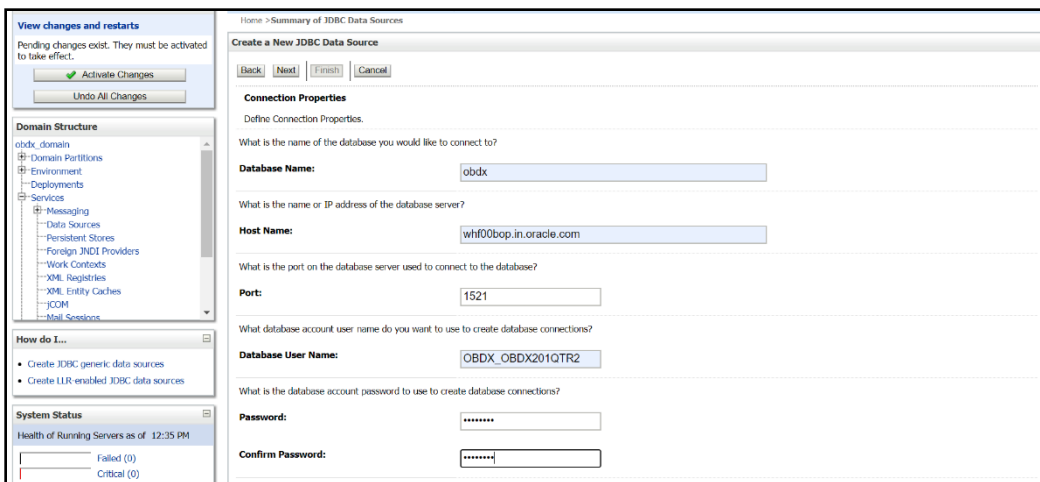


- 1. **Name :- SYSCONFIG**
- JNDI Name :- SYSCONFIG**



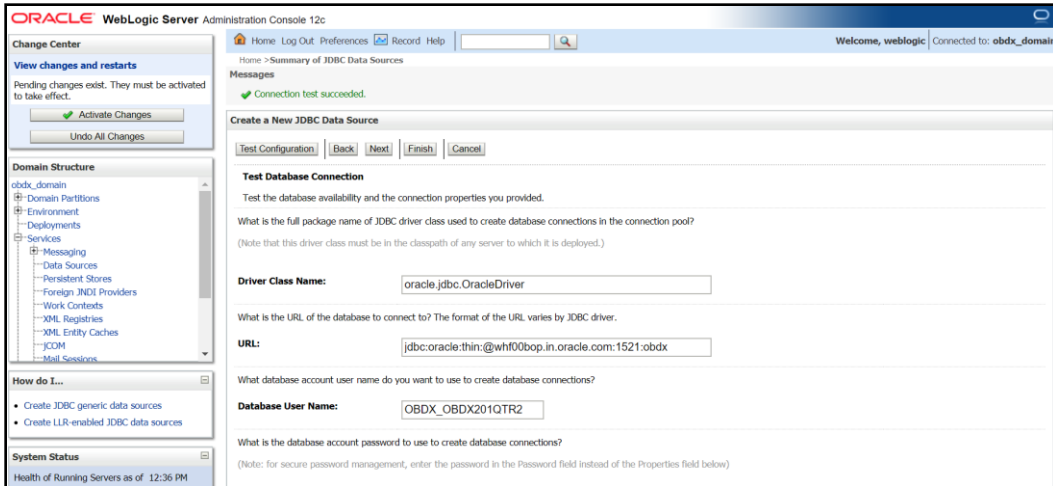


2. Click on Next

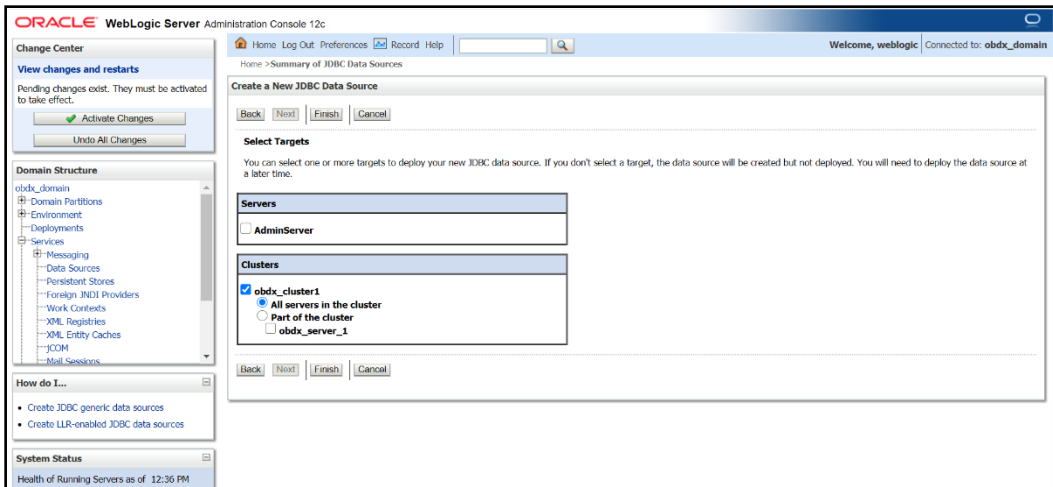


3. Provide

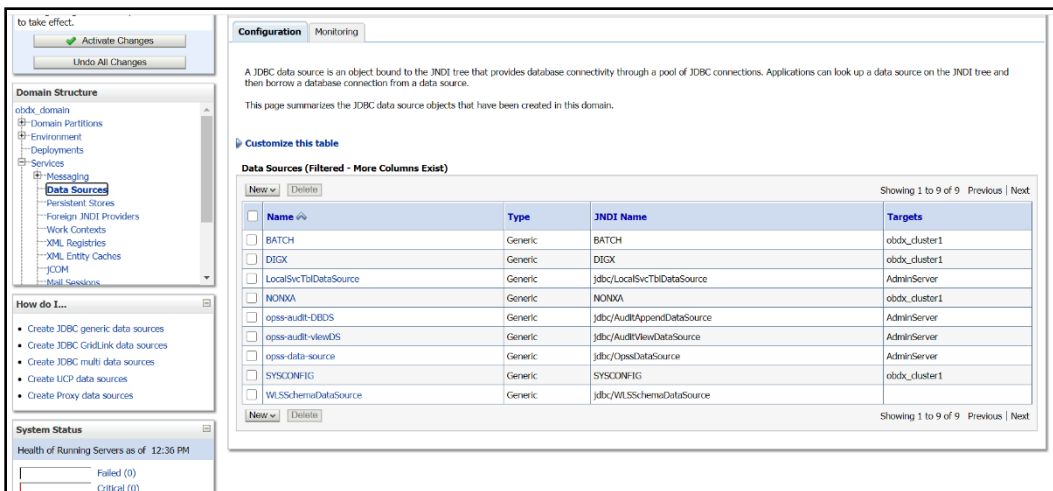
- Database Name:** - Database SID
- Host Name:** - Database hostname
- Port:** - Database port Number
- Database user Name:** - OBAPI\_\${POST\_FIX}
- Password:-** Database user password



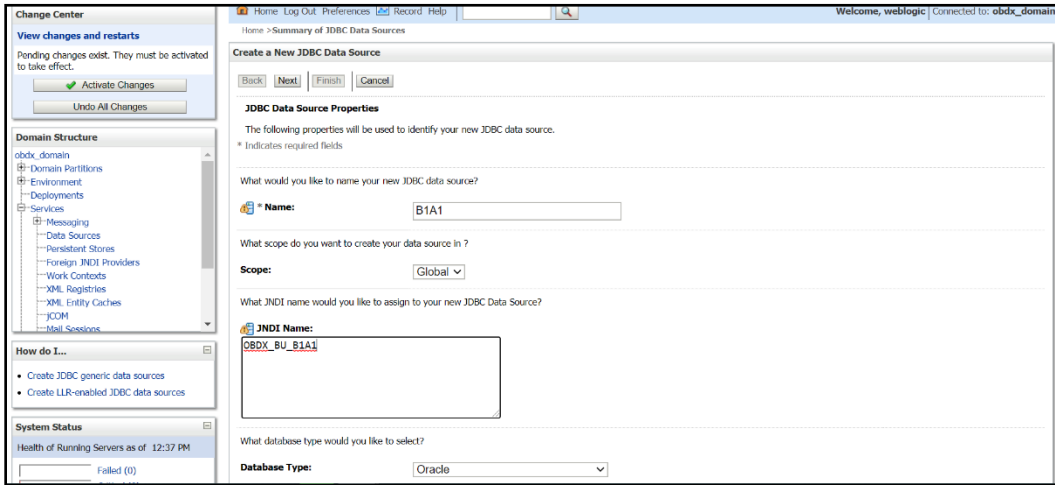
#### 4. Test Configuration



#### 5. Select target as cluster and click on Finish

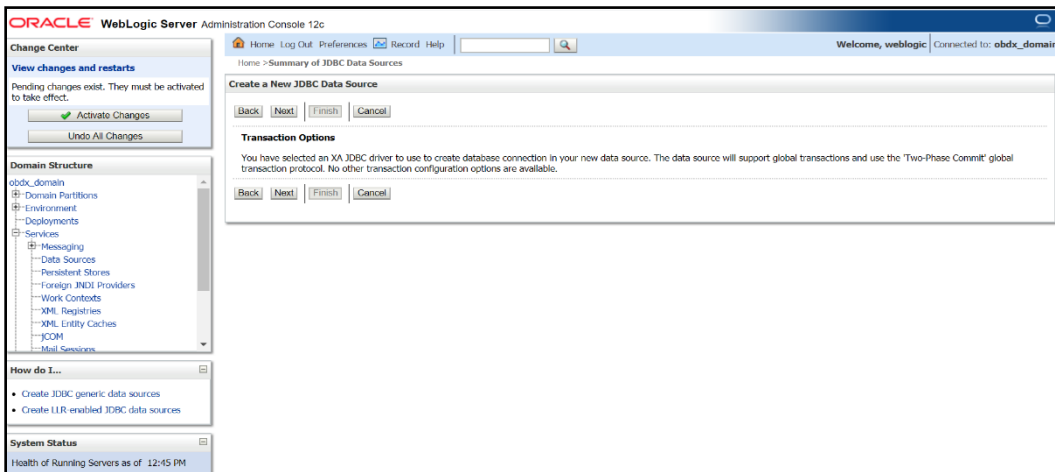
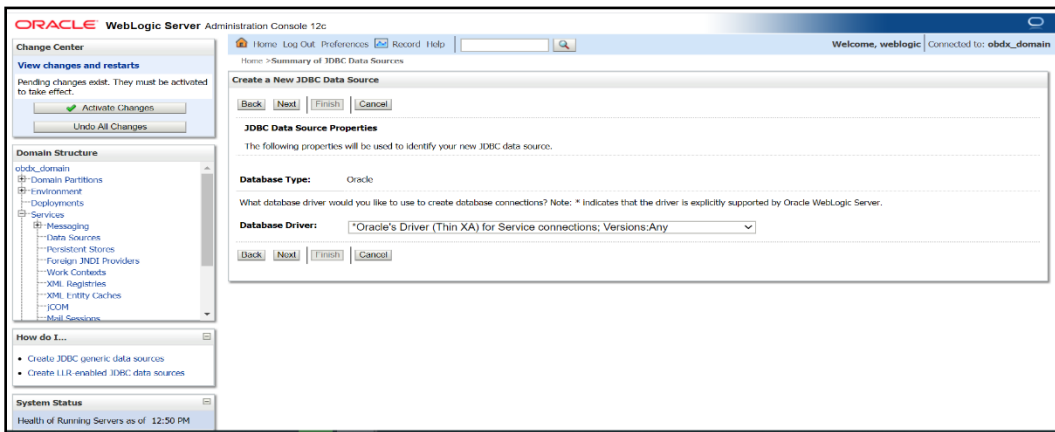


### 3.6 Creating B1A1 data source



1. Name:- B1A1

JNDI Name :- OBDX\_BU\_B1A1



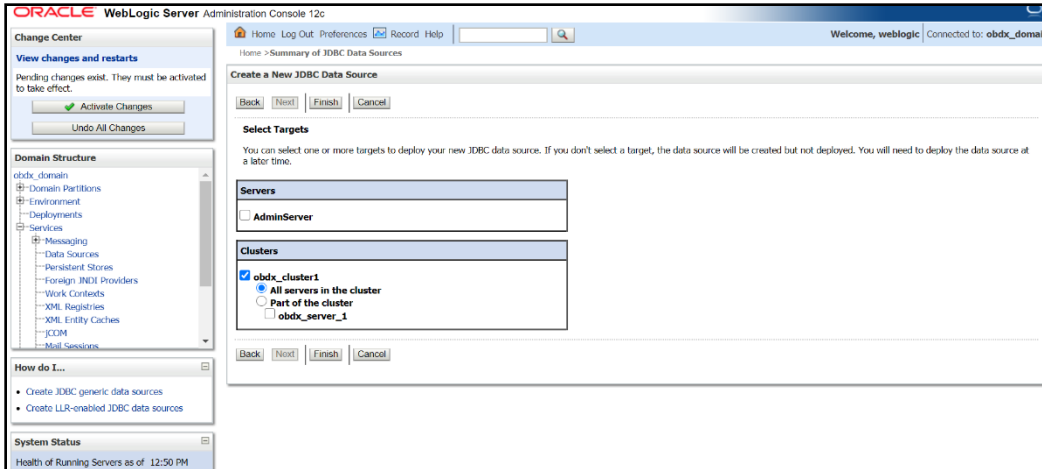
2. Click on Next

3. Provide

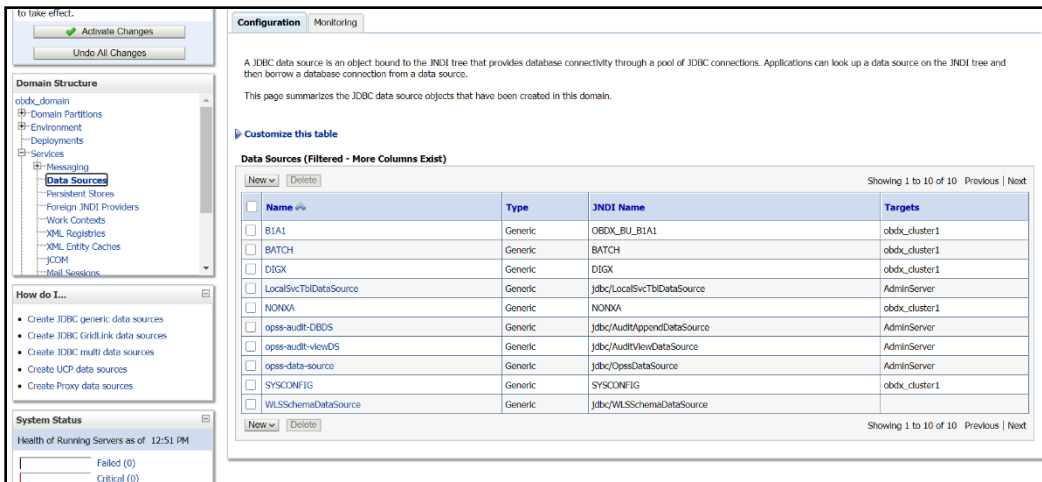
- Database Name:** - Database SID (\$EHMS\_DATABASE\_SID)
- Host Name:** - Database hostname (\$EHMS\_DATABASE\_HOSTNAME)
- Port:** - Database port Number (\$EHMS\_DATABASE\_PORT)
- Database user Name:** - \${ EHMS\_SCHEMA\_NAME }
- Password:** - Database user \${ EHMS\_SCHEMA\_NAME } password

4. Test Configuration



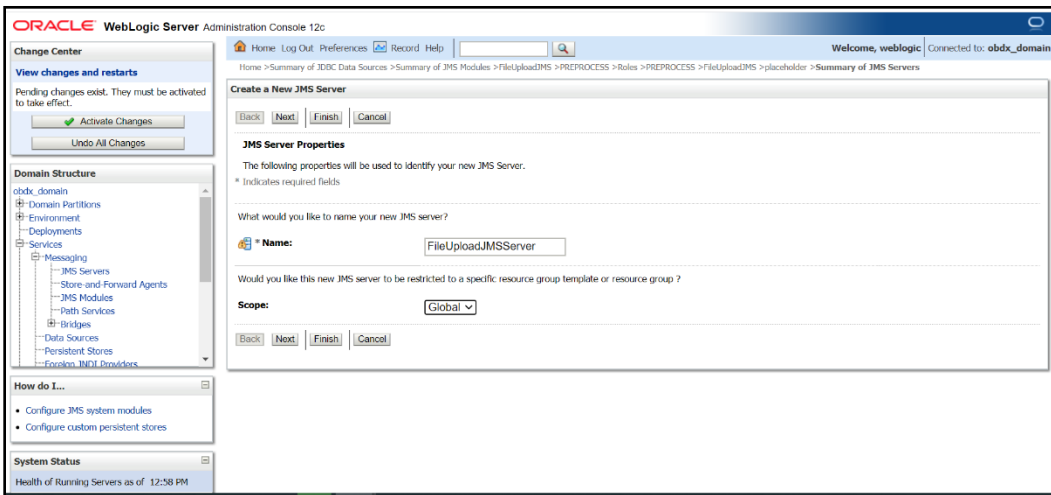
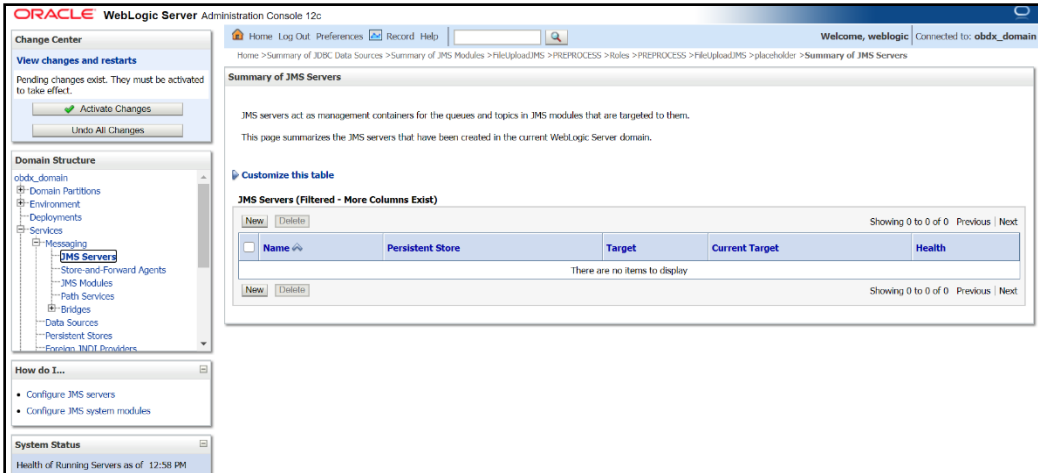


5. Set target as cluster and click on Finish

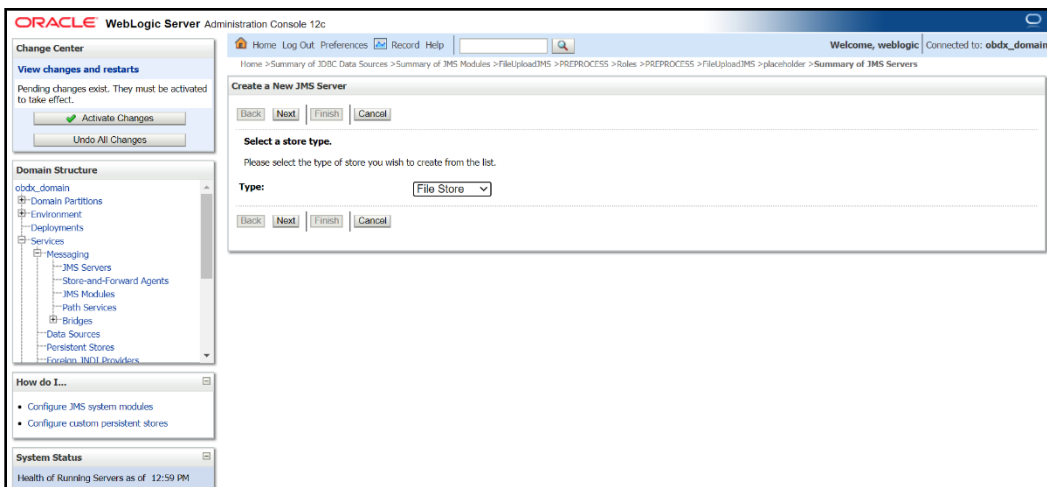
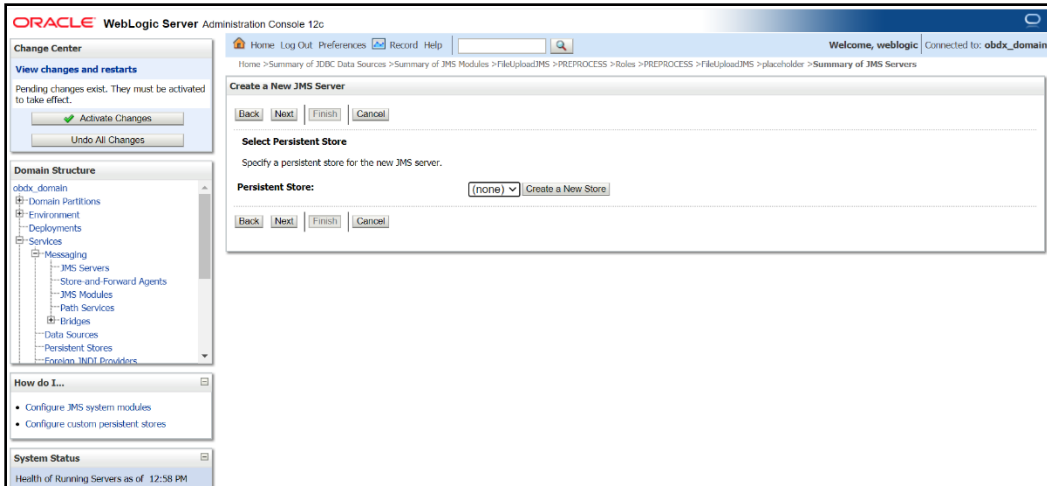


### 3.7 Create JMS server and JMS Module

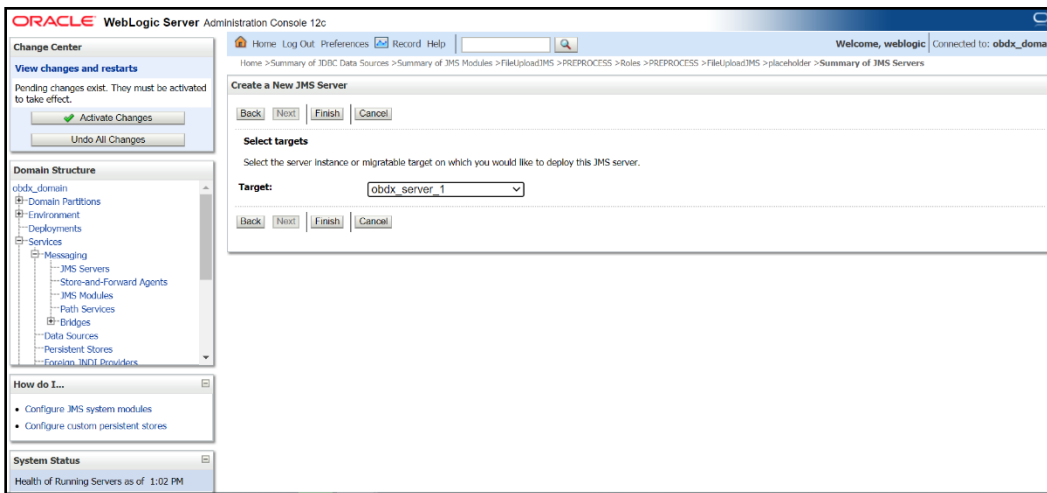
- Creating FileUploadJMS JSM Module
- Creating WLS\_JMS\_FILEUPLOAD\_PS FileStore
- Creating FileUploadJMSServer JMS Server



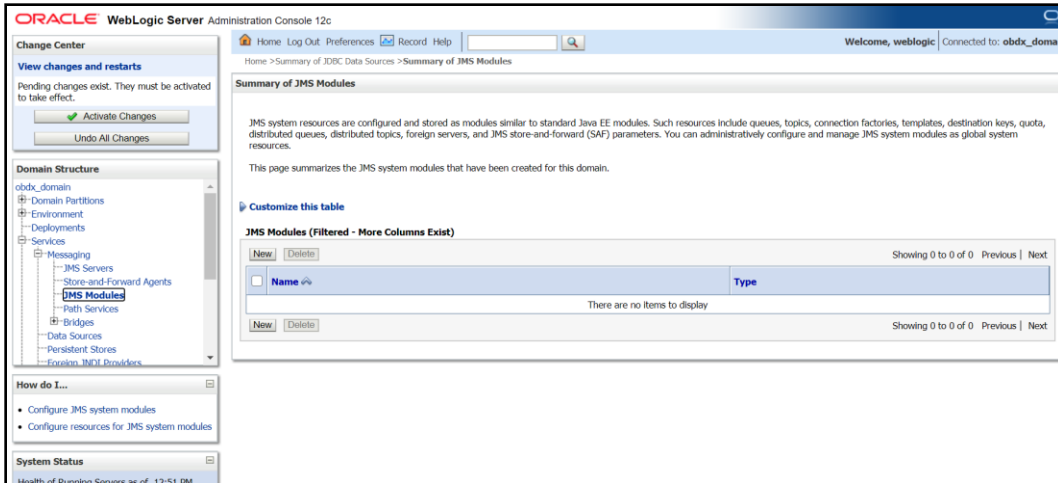
1. Click on JMS Servers → Name – FileUploadJMSServer -- > Click on Next



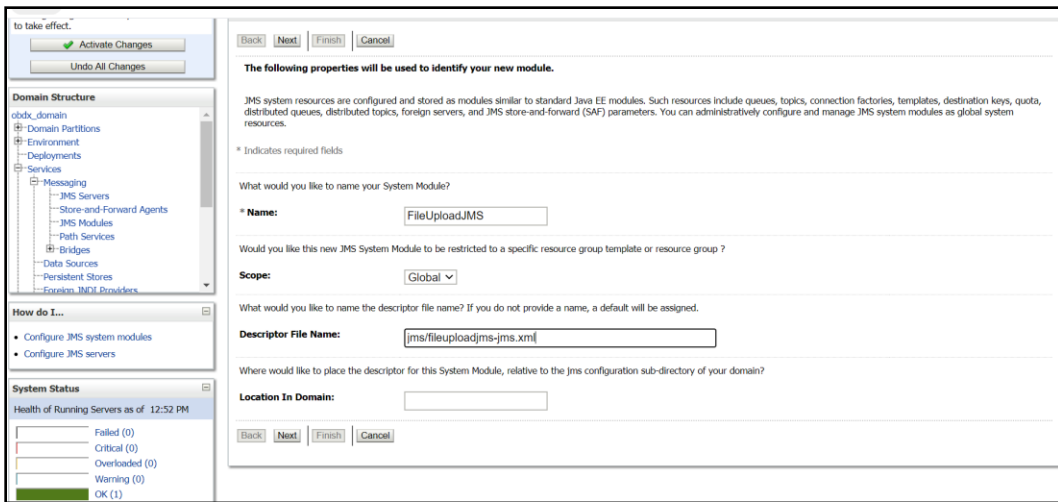
2. Select Type as File Store and click on Next



3. Select target as managed server and click on Finish



4. Left hand side click on JMS Module -- click on New

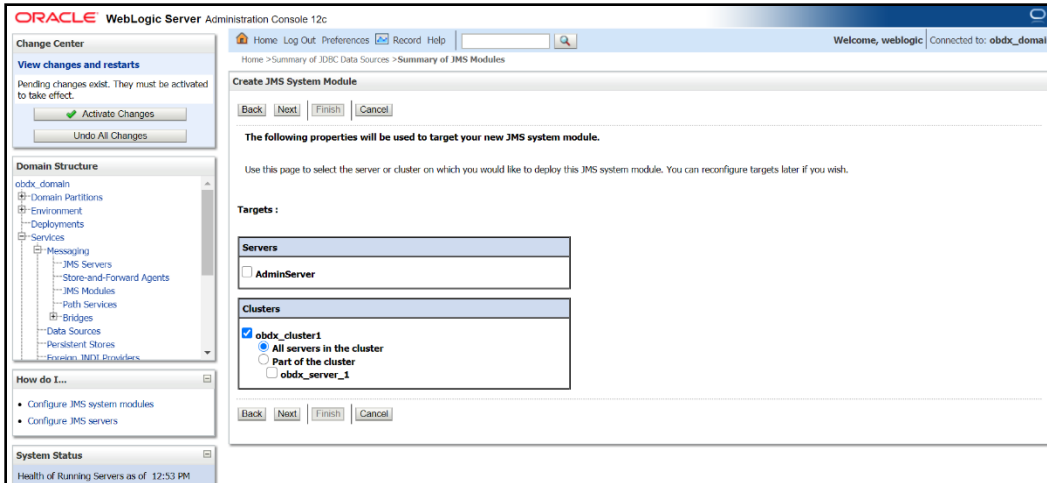


5. Name :- FileUploadJMS

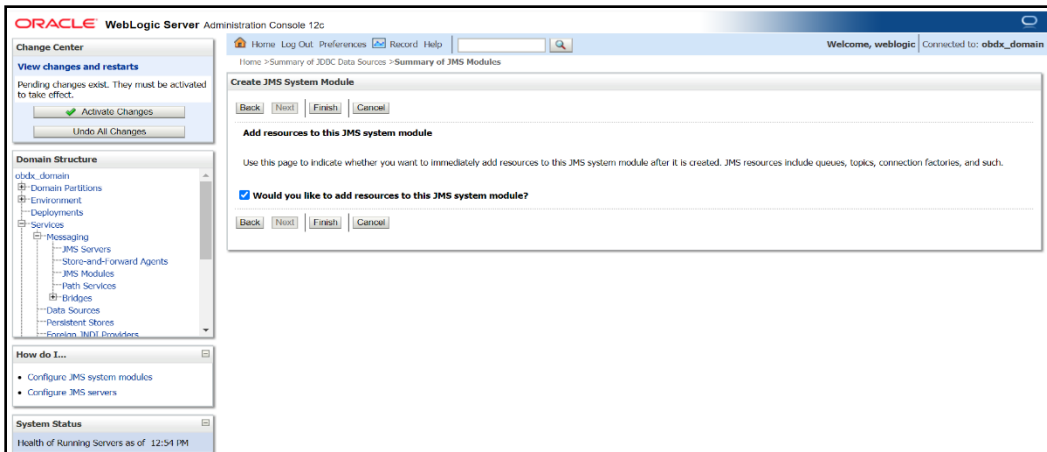
Scope:- Global

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

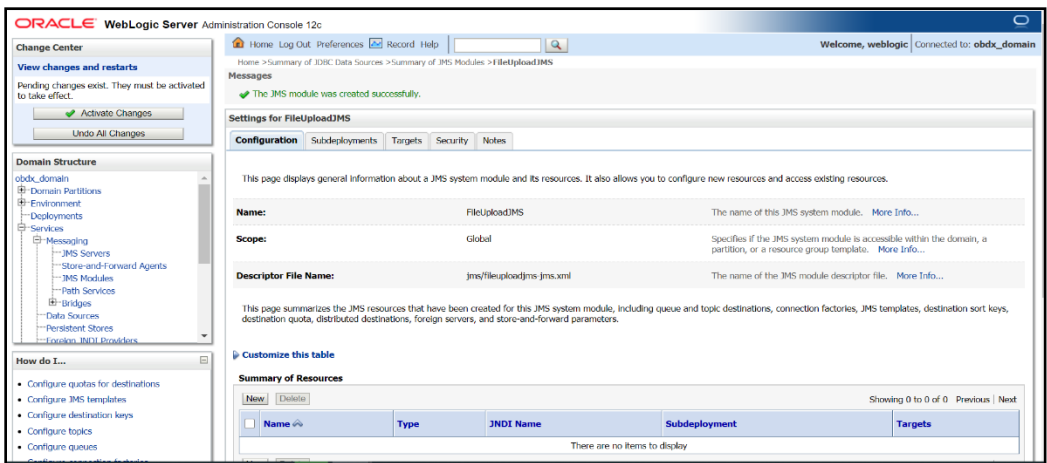
6. Click on Next



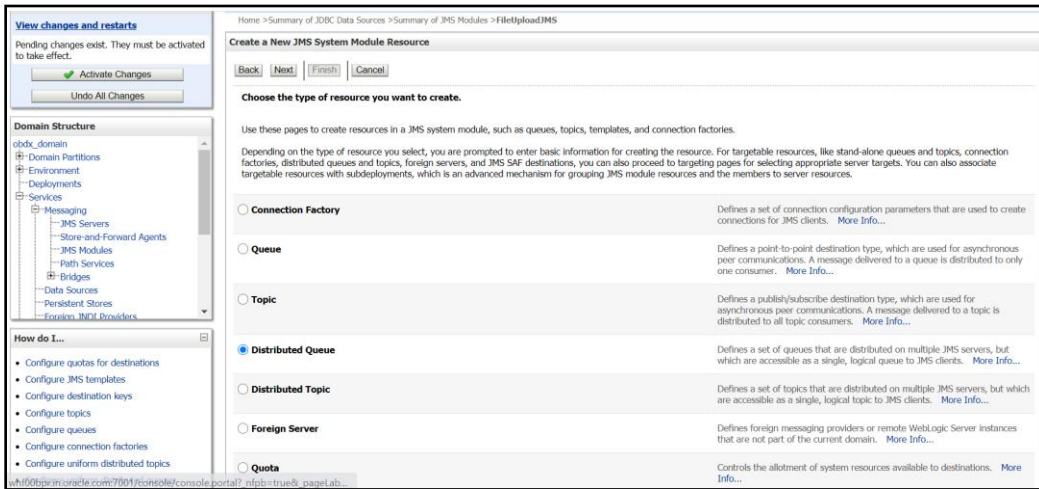
7. Set target as cluster → click on Next



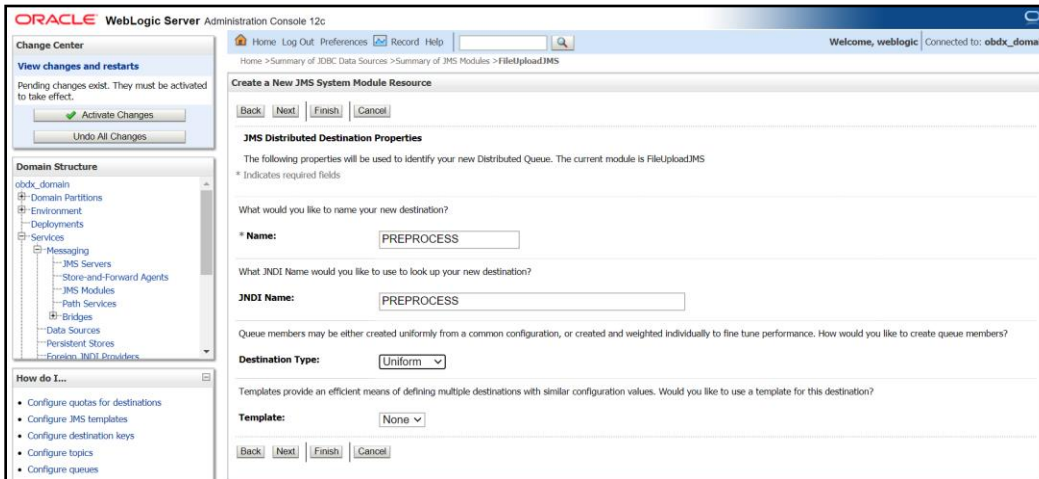
8. Select Would you like to add resources to this JMS system module and click on finish



9. Select new



10. Select Distributed Queue and click next



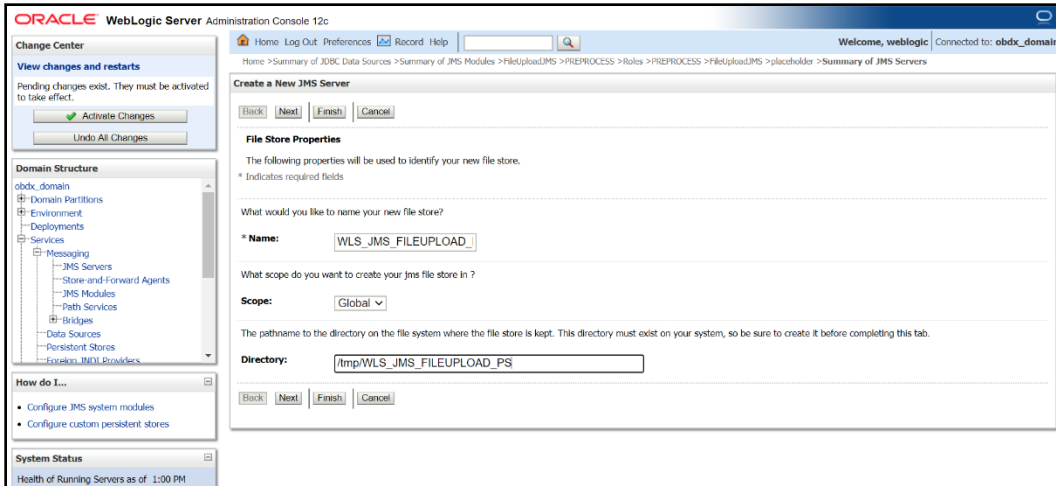
11. Provide

**Name:** - PREPROCESS

**JNDI Name:** - PREPROCESS

**Destination Type:** - Uniform

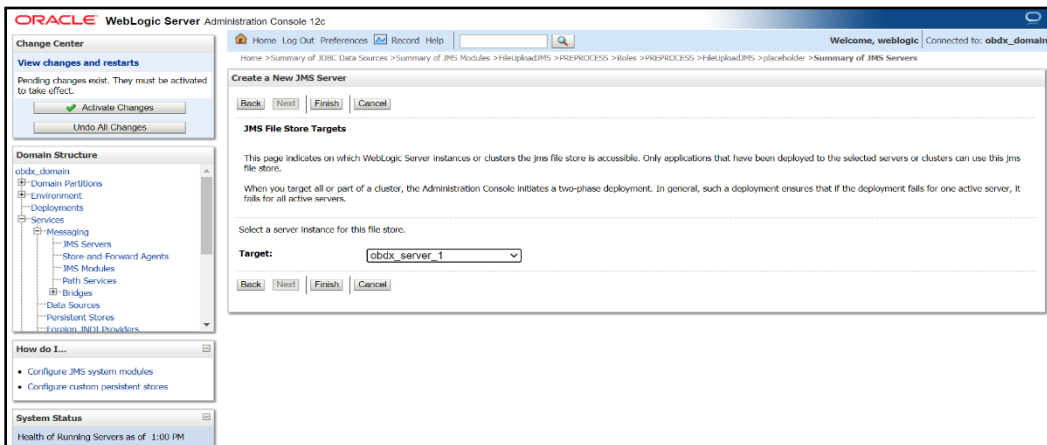
**Template:** - None



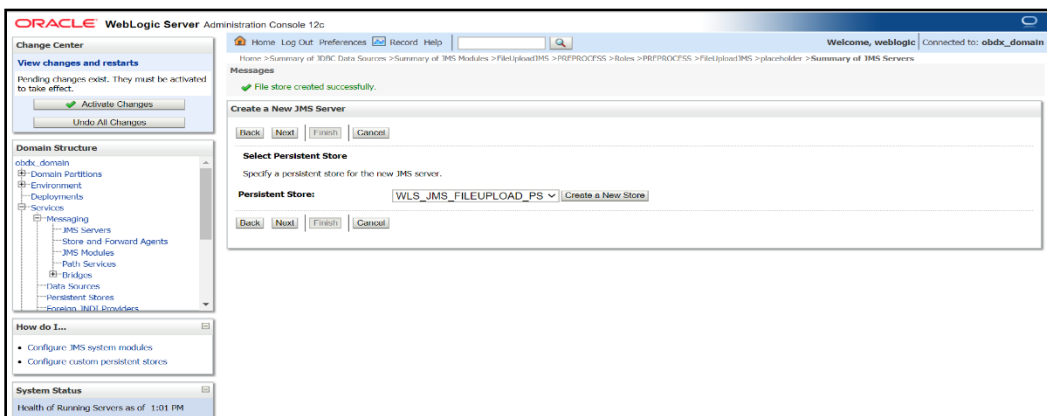
**12. Name :- WLS\_JMS\_FILEUPLOAD\_PS**

**Scope :- Global**

**Directory :- /tmp/WLS\_JMS\_FILEUPLOAD\_PS**

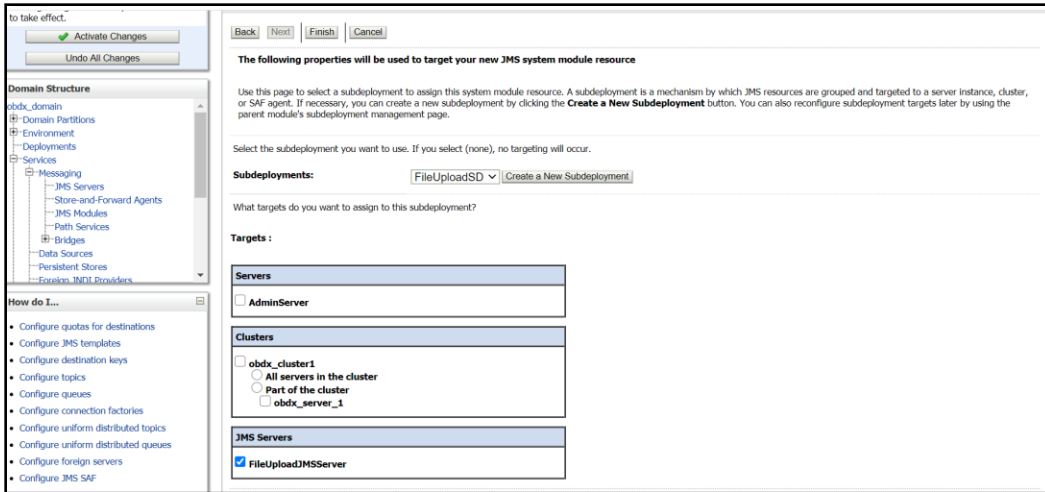


**13. Select target as managed server**

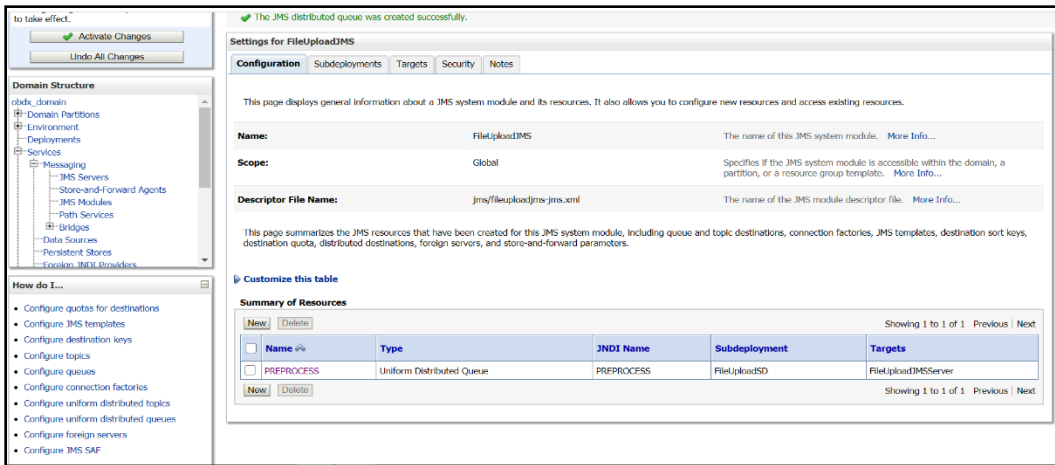


**14. Select WLS\_JMS\_FILEUPLOAD\_PS and click on Next**

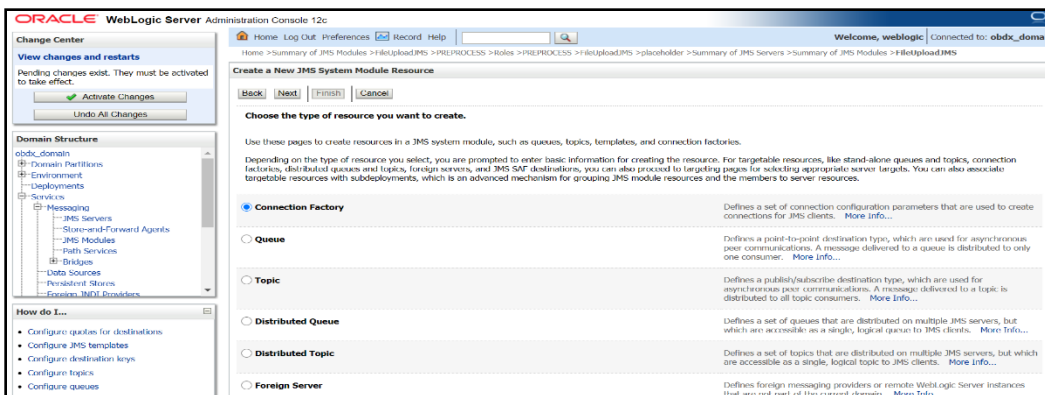
**15. Select Create a New Subdeployment and create FileUploadSD**



16. Select FileUploadJMSServer and click on Finish

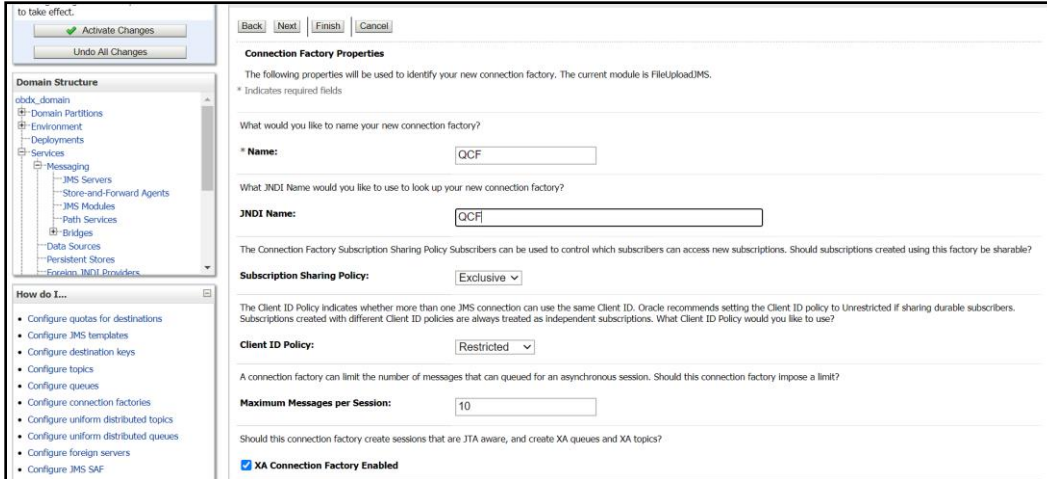


17. Similarly Go into FileuploadJMS module and click on Next



18. Select Connection factory -> Click Next





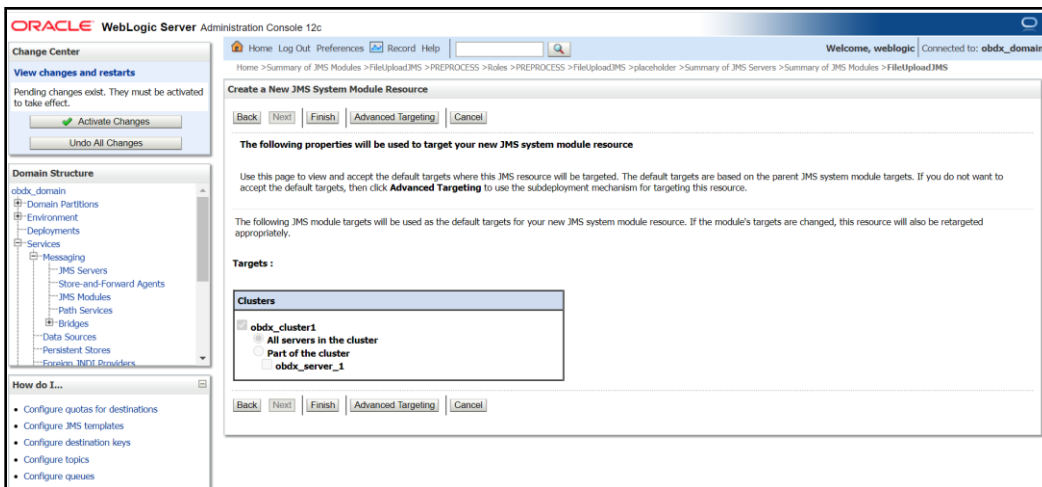
### 19. Provide

**Name :- OCF**

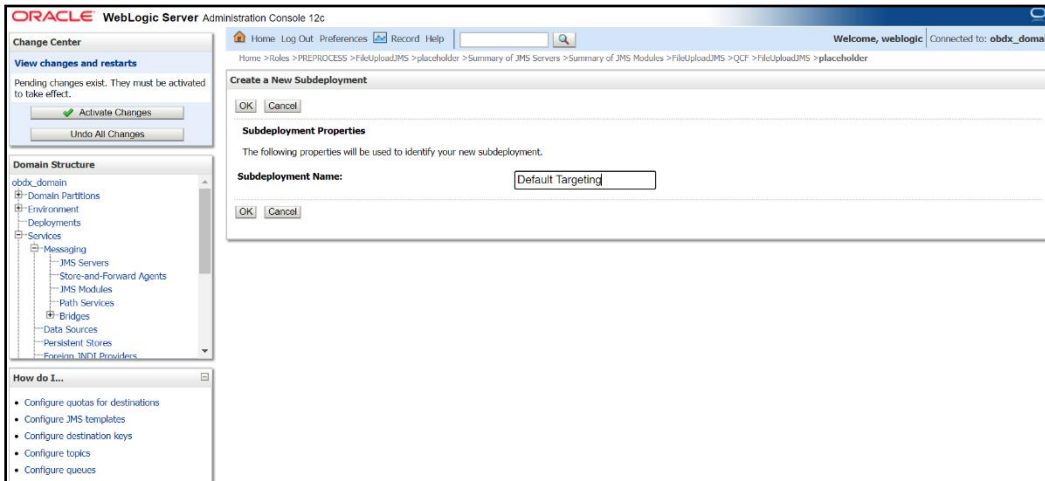
**JNDI Name :- OCF**

**Subscription Sharing Policy :- Exclusive**

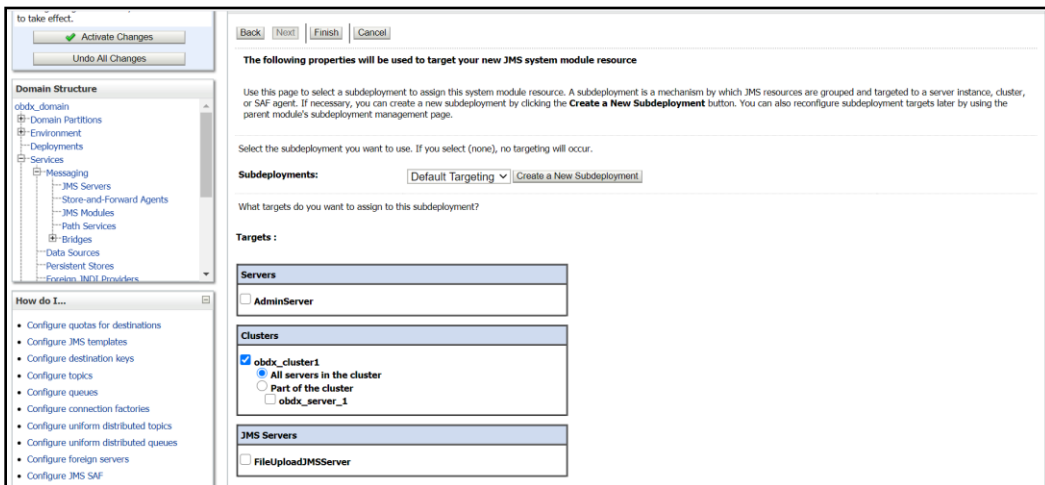
**Client ID Policy :- Restricted**



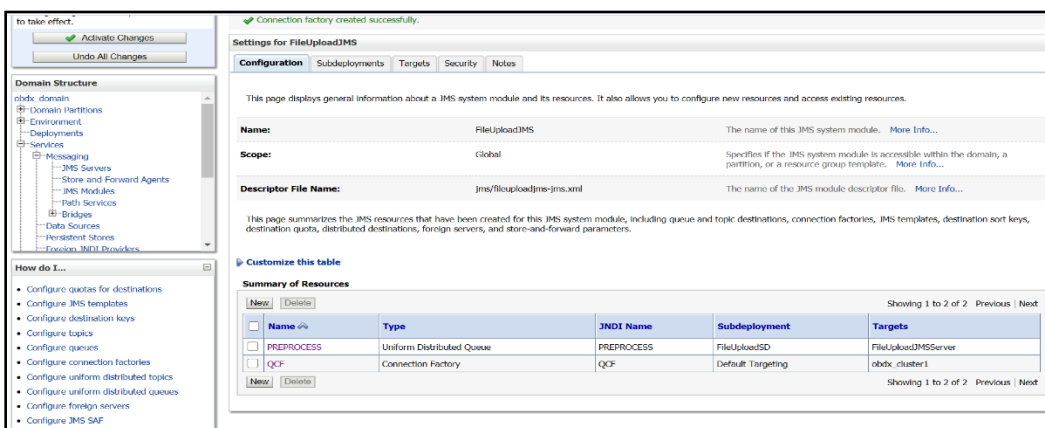
## 20. Click on Advanced targeting



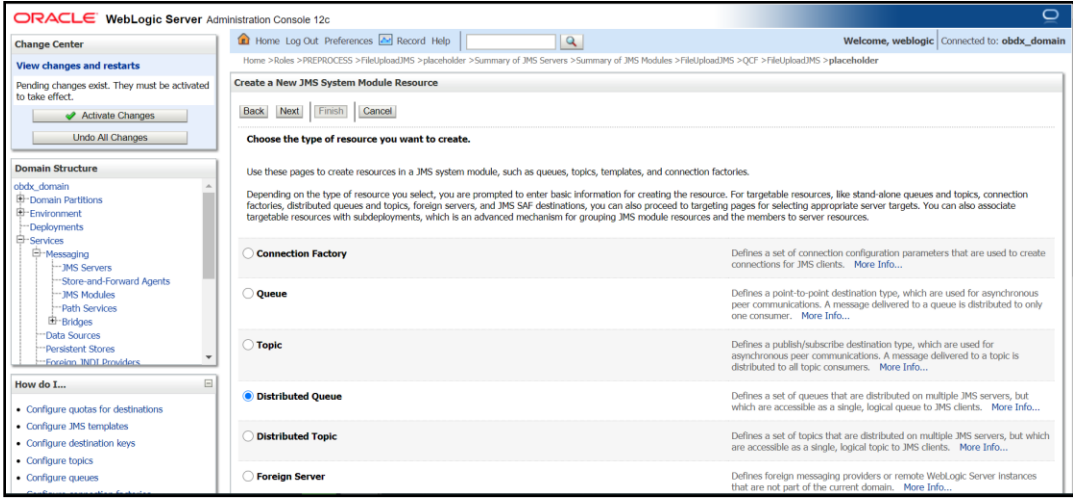
## 21. Provide Subdeployment Name as Default Targeting



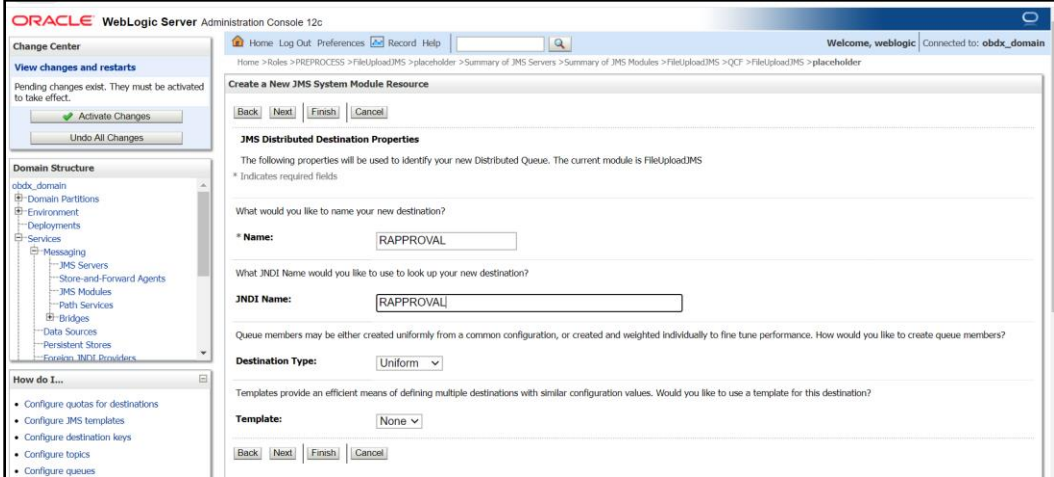
## 22. Select cluster and click on Finish



23. Go to FileUpload JMS click on New

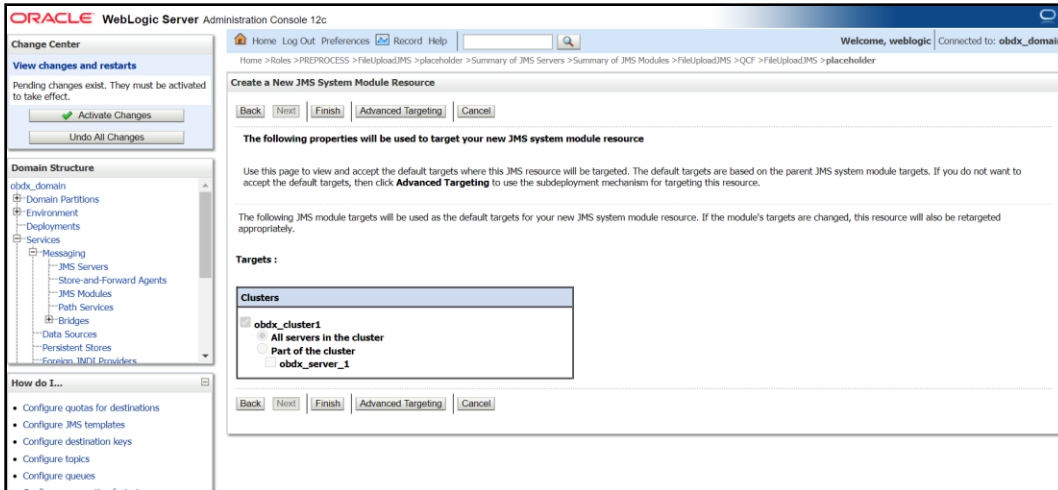


24. Select Distributed Queue

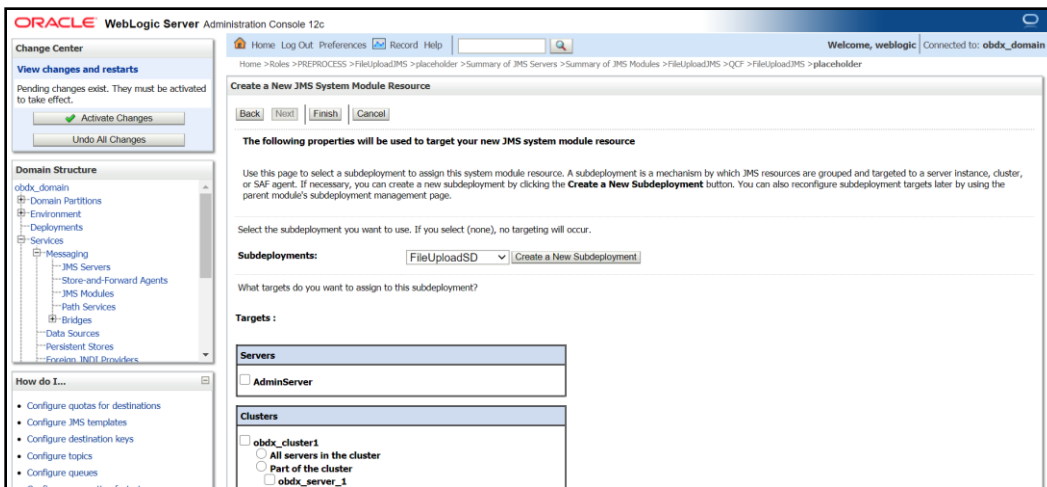


25. Provide

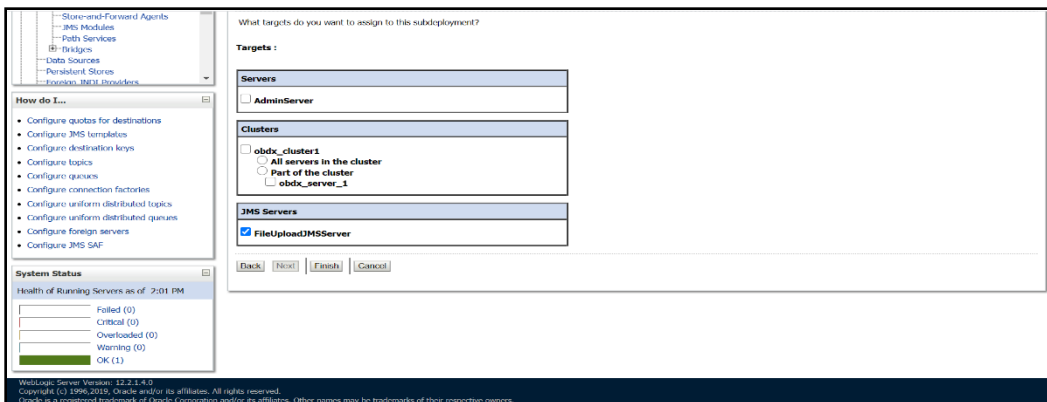
- Name :- RAPPROVAL**
- JNDI Name :- RAPPROVAL**
- Destination Type:- Uniform**
- Template :- None**



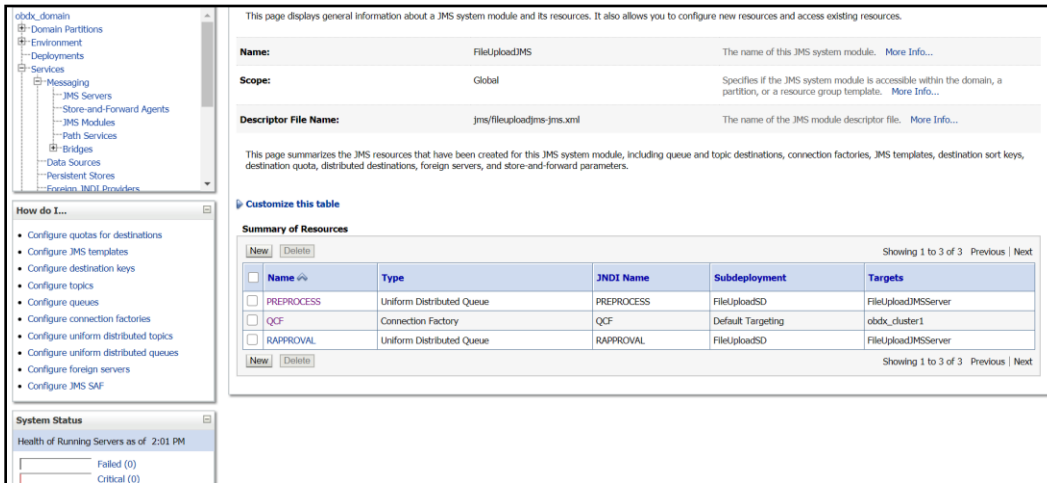
## 26. Select Advance targeting



## 27. Select Subdeployment :- FileUploadSD



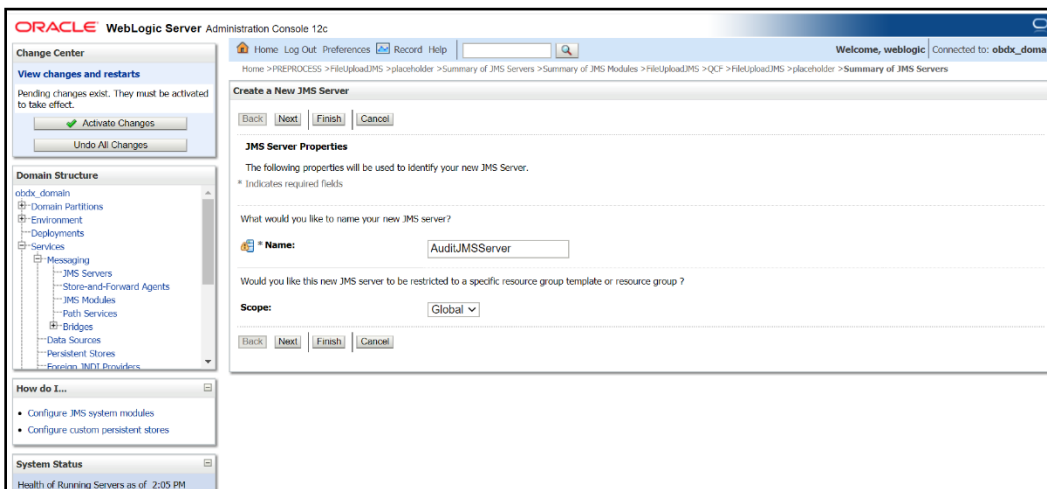
28. Select FileUploadJMSServer and click on Finish



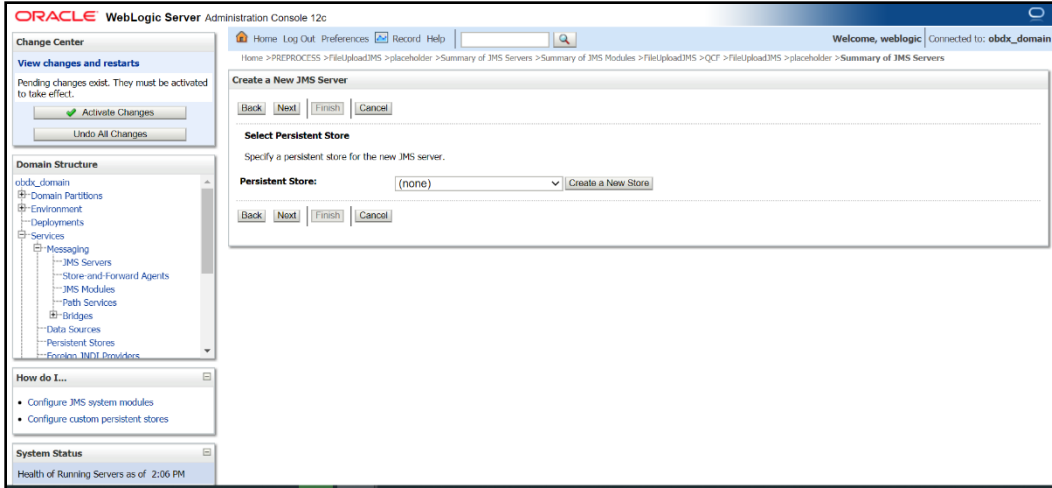
3.8 Creating WLS\_JMS\_AUDIT\_PS FileStore

3.9 Creating AuditJMSServer JMS Server

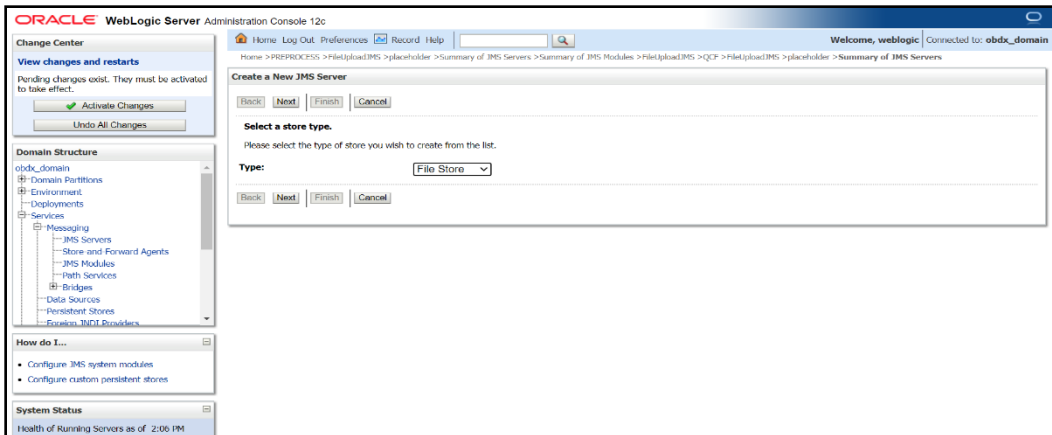
3.10 Creating WLS\_JMS\_REPORT\_PS FileStore



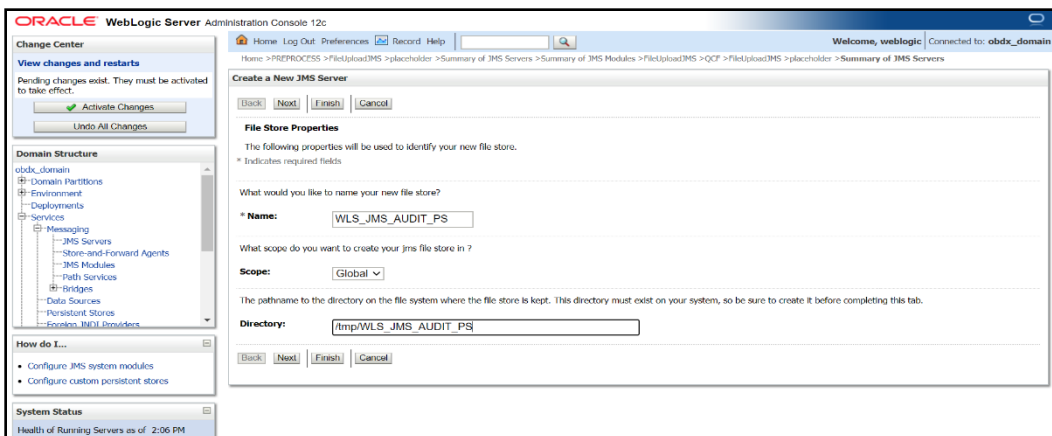
1. Click on JMS server and click on New
2. Provide Name as AuditJMSServer , Scope as Global



3. Click on Create a New Store



4. Select File Store

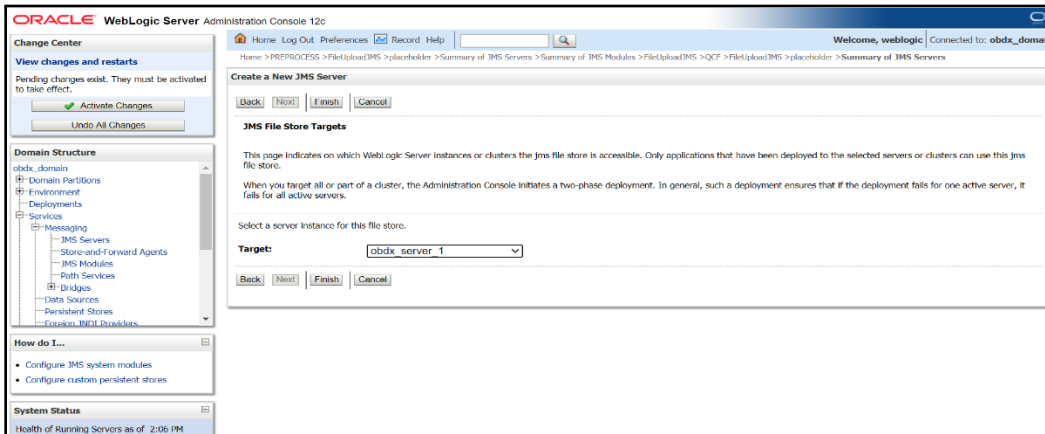


5. Provide

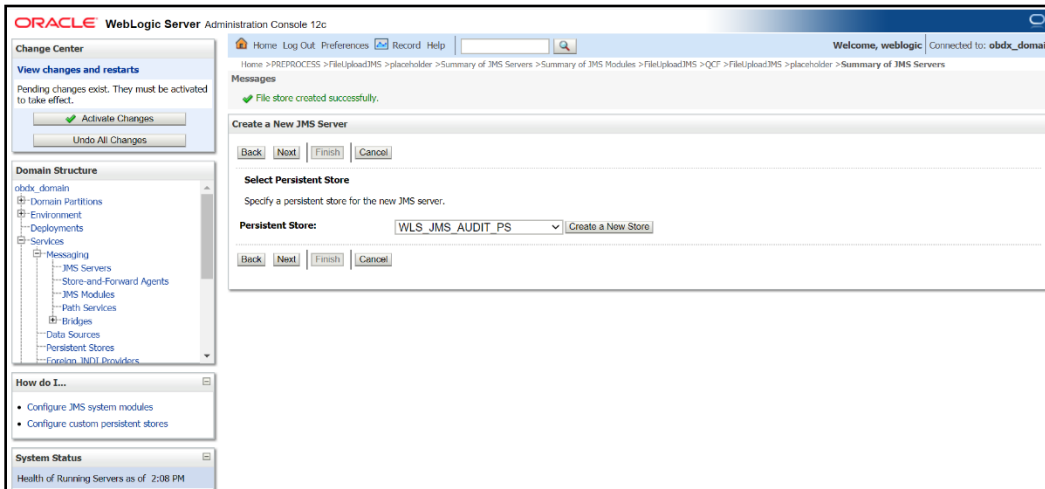
**Name :-** WLS\_JMS\_AUDIT\_PS

**Scope :-** Global

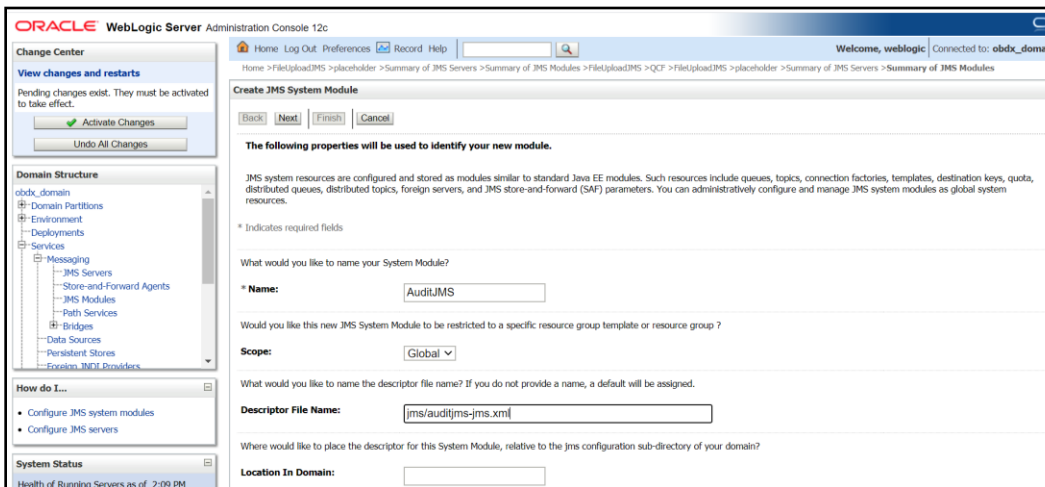
Directory :- /tmp/WLS\_JMS\_AUDIT\_PS



6. Select Target as managed server and click on Finish



7. Select the new store created WLS\_JMS\_AUDIT\_PS and click on Next

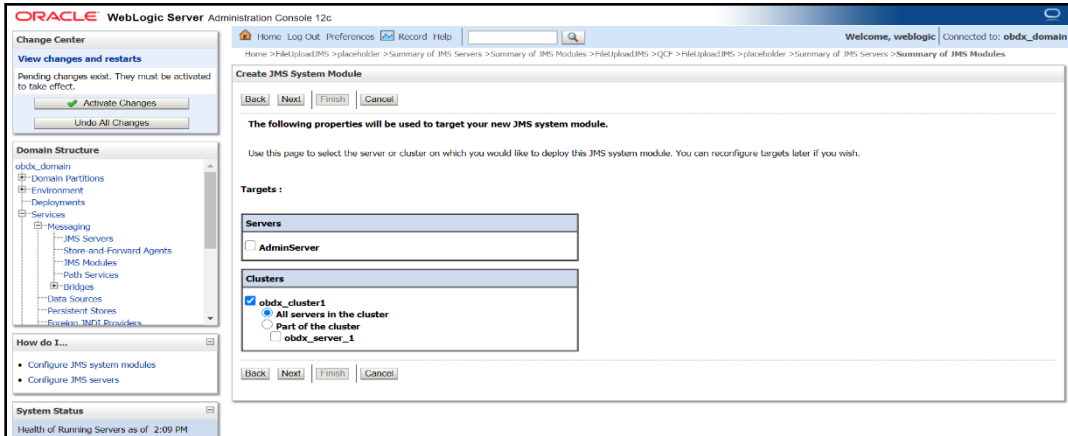


## 8. Provide

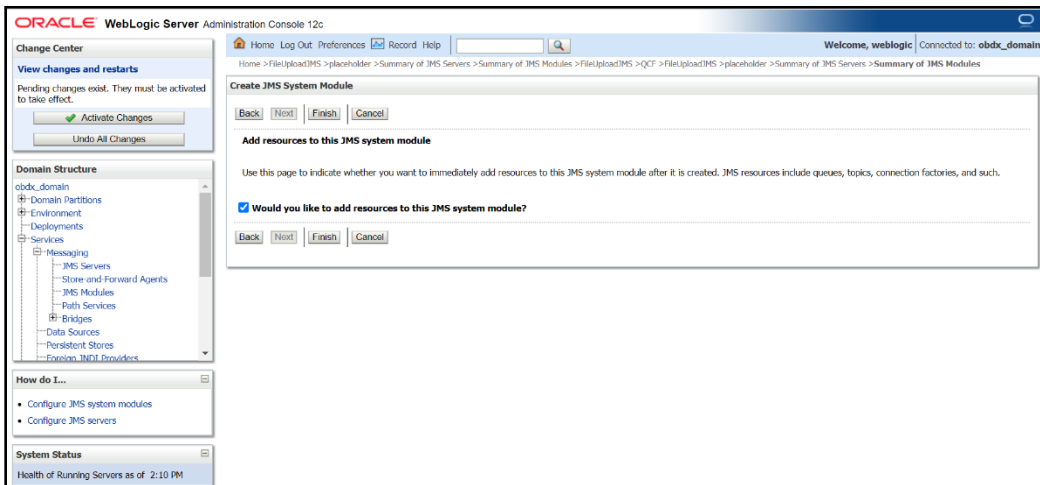
**Name :-** AuditJMS

**Scope :-** Global

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

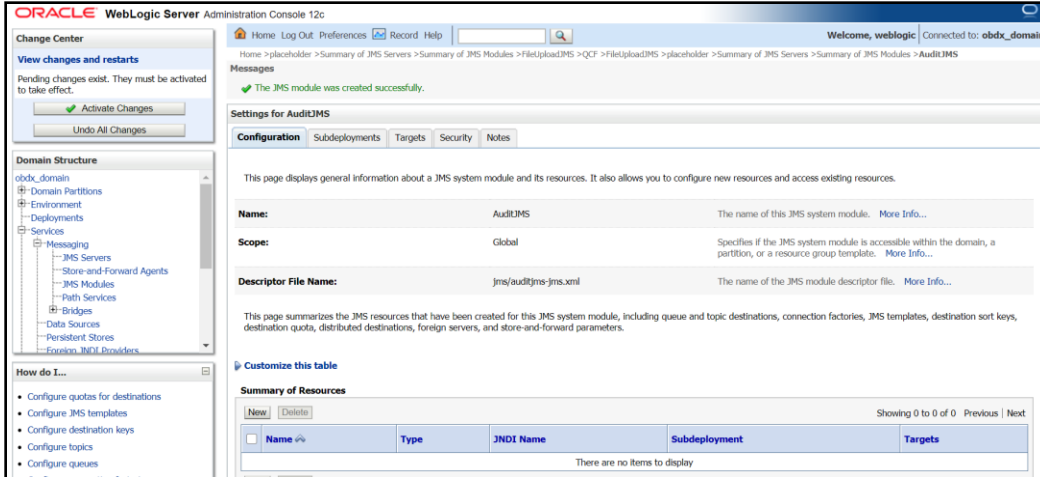


## 9. Select Cluster as a target

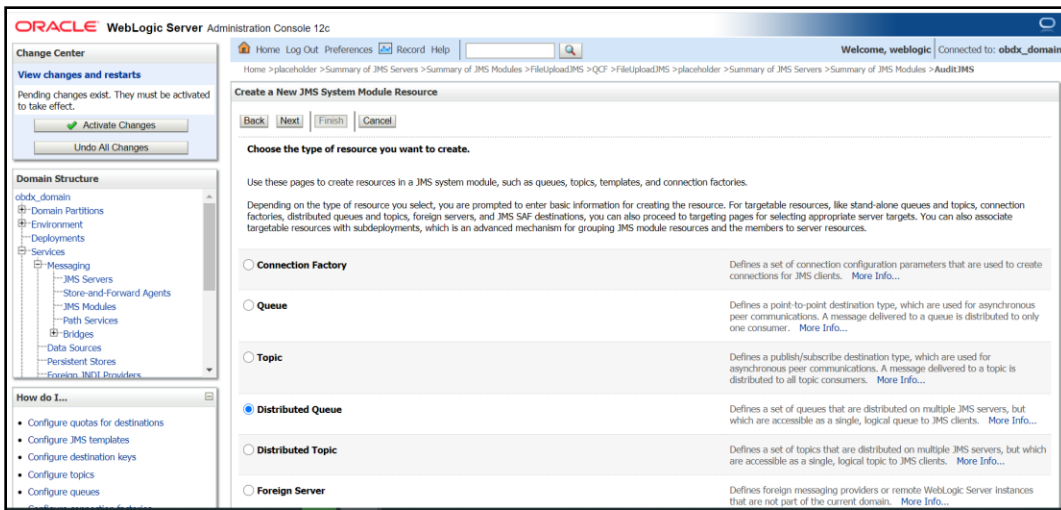


## 10. Select would you like to add resource to this JMS system module?

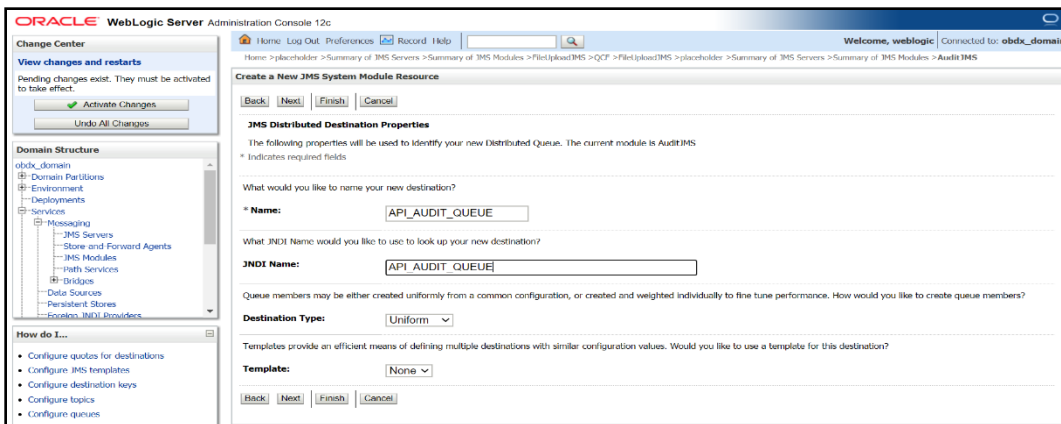




11. Click on new



12. Select Distributed Queue

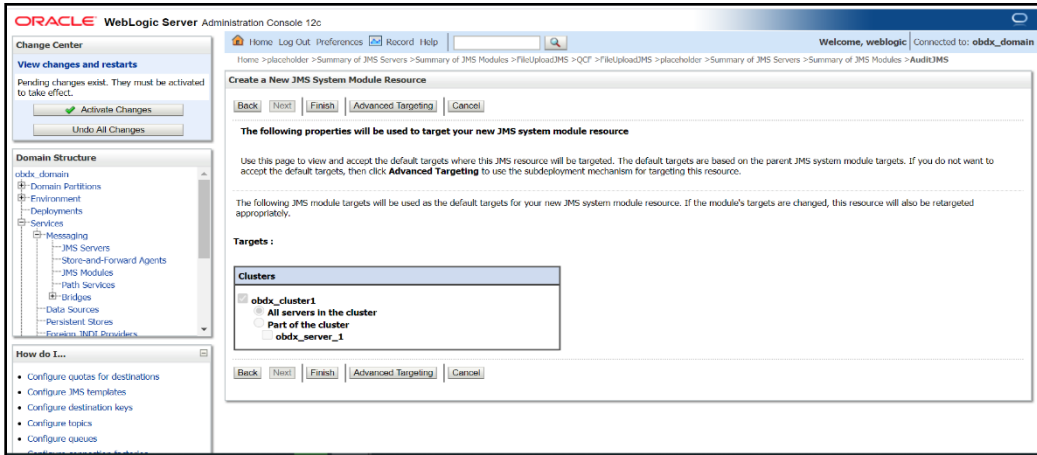


13. Name:- API\_AUDIT\_QUEUE

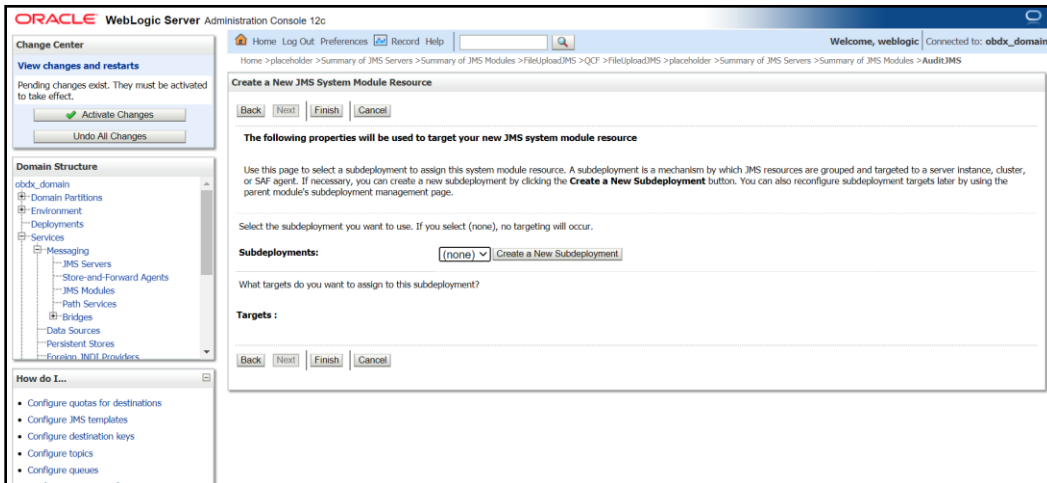
JNDI Name:- API\_AUDIT\_QUEUE

Destination Type :- Uniform

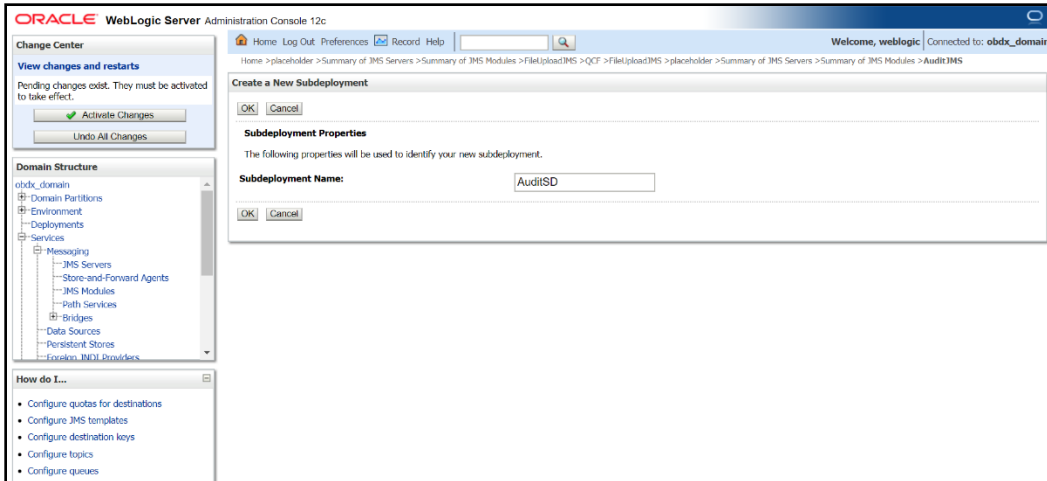
Template:- None



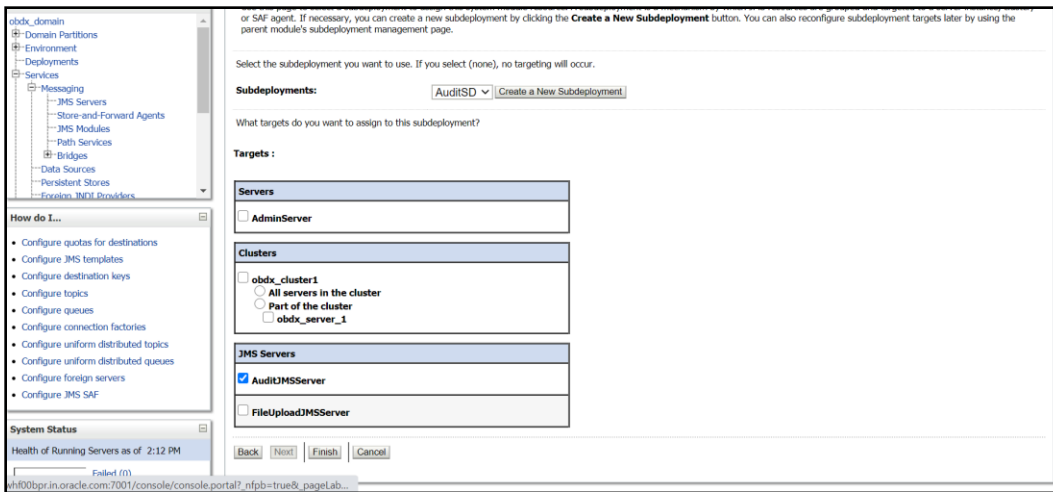
14. Select Advance targeting



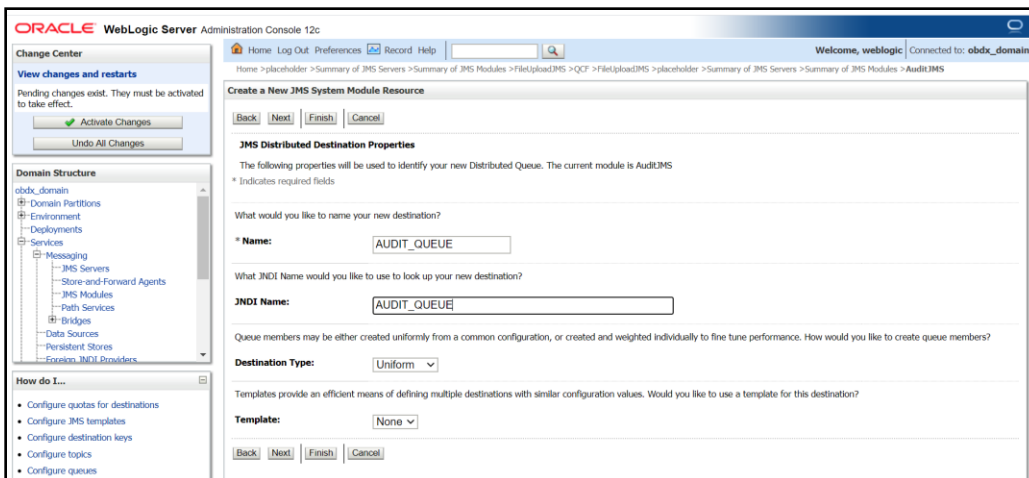
15. Click on Create a New Subdeployment

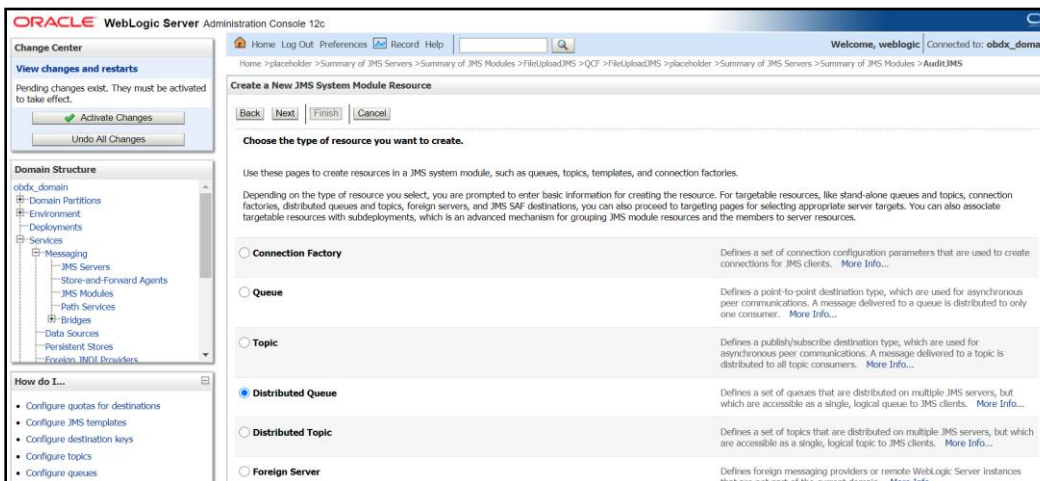
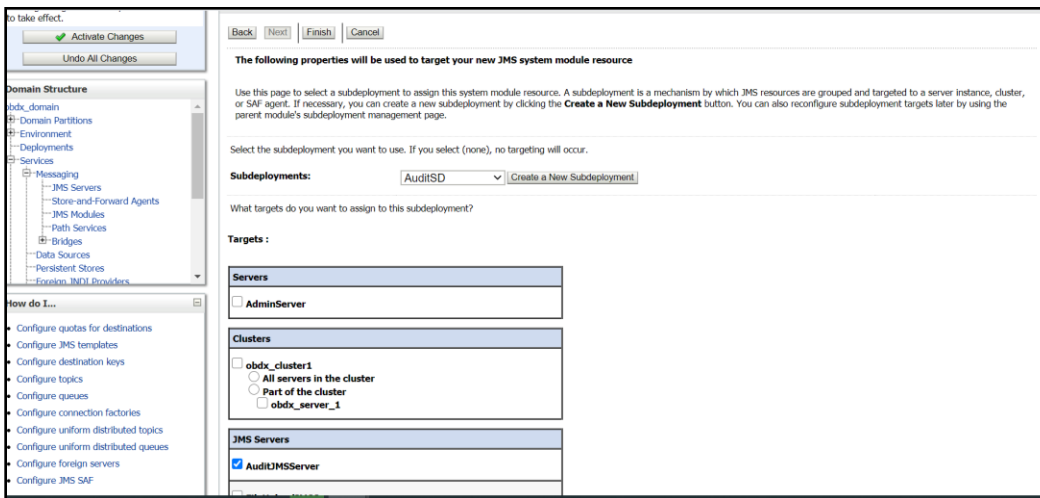
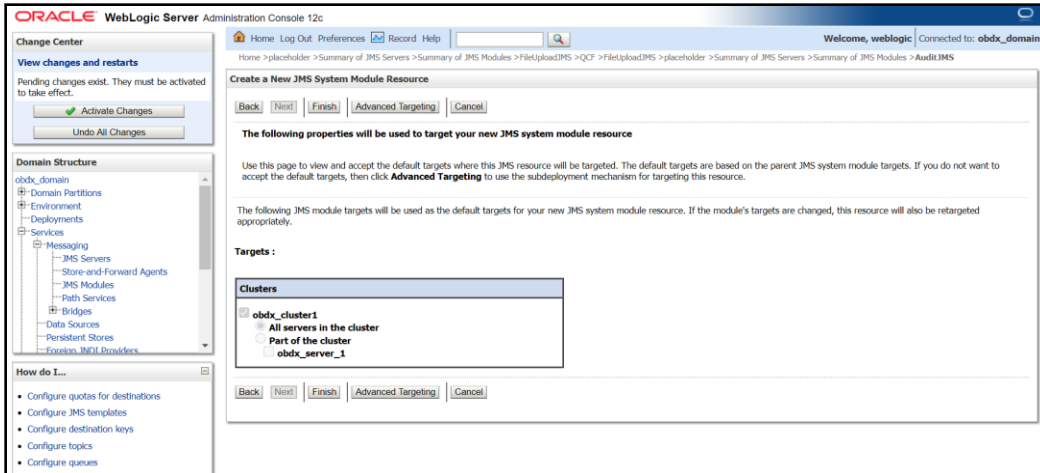


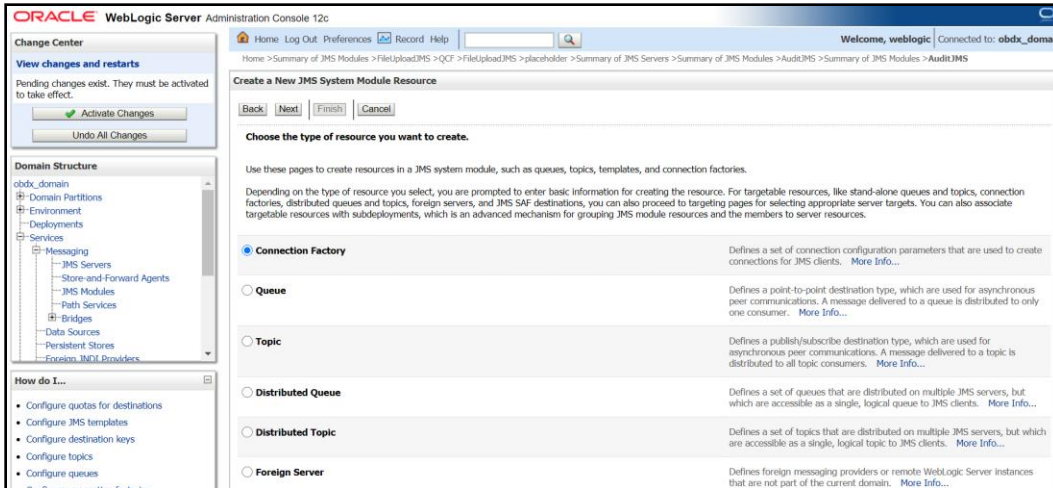
### 16. Provide Subdeployment Name as AuditSD



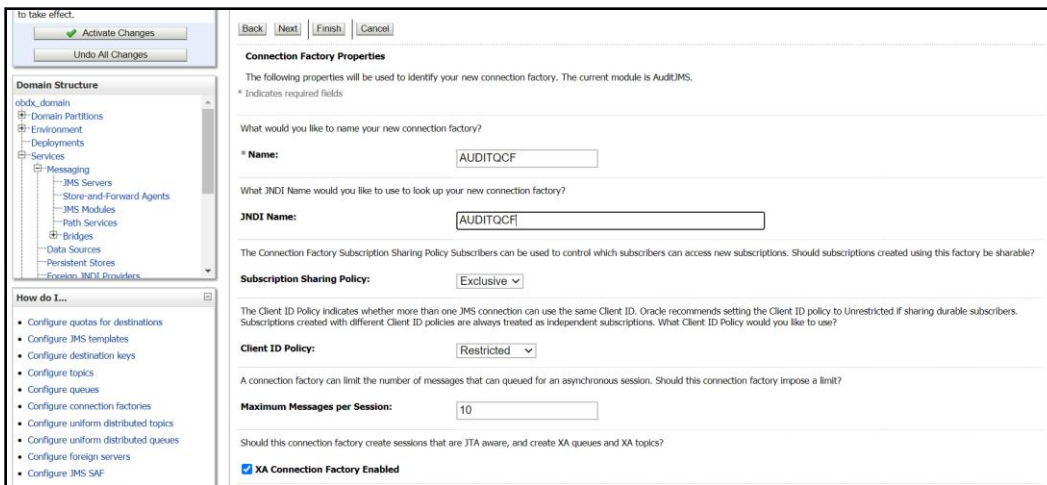
### 17. Select Target as AuditJMSServer







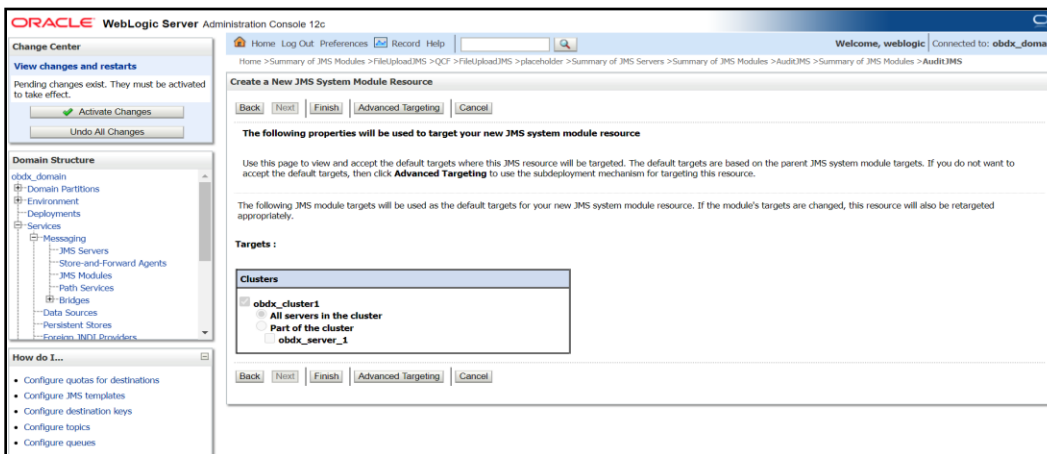
18. Click on connection Factory



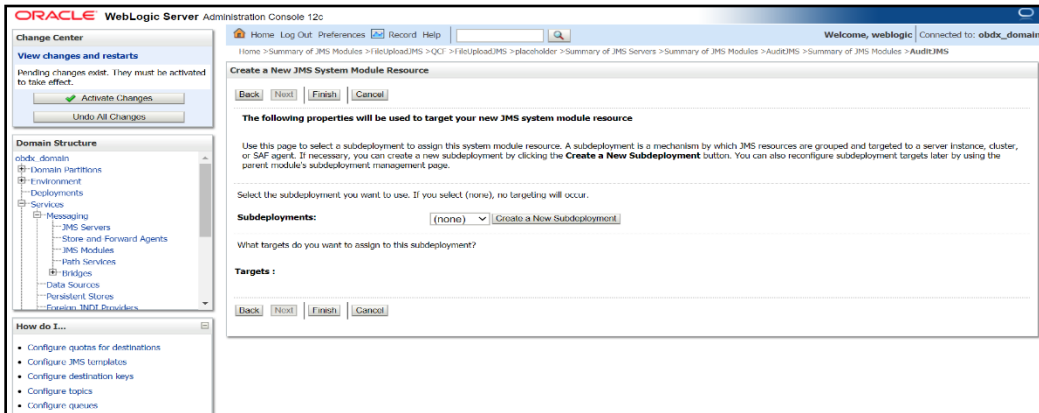
19. Provide

Name :- AUDITQCF

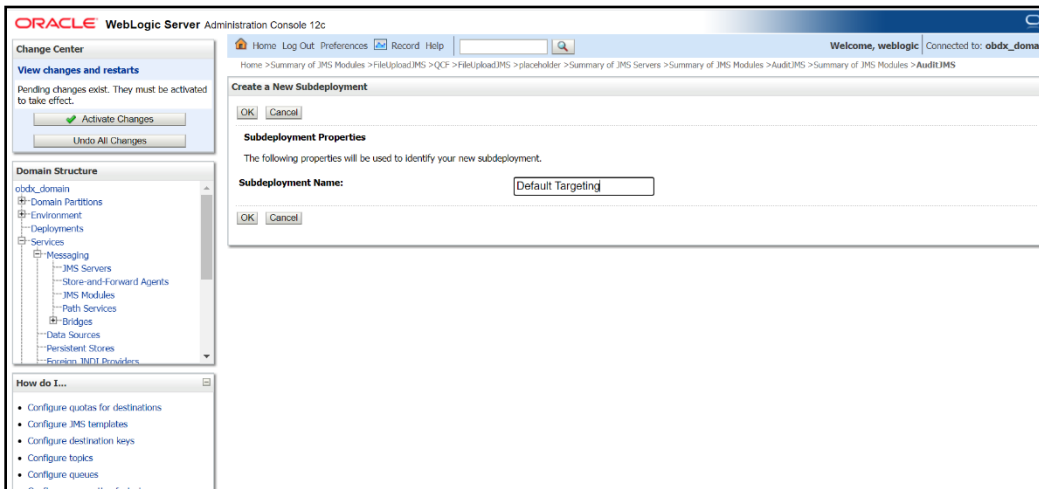
JNDI Name :- AUDITQCF



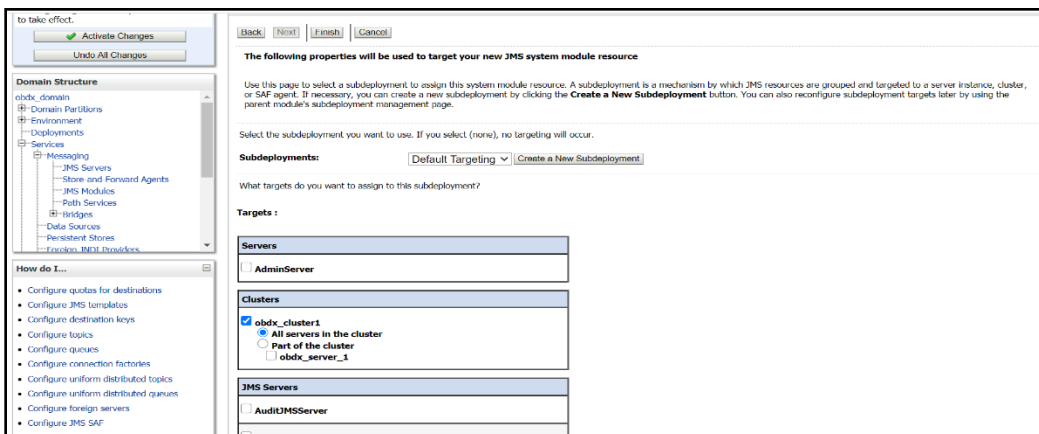
## 20. Click on Advanced Targeting



## 21. Click on Create a New Subdeployment



## 22. Give Subdeployment Name as Default Targeting



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

to take effect.

Activate Changes

Undo All Changes

Domain Structure

- obdx\_domain
- Domain Partitions
- Environment
- Deployments
- Services
  - Messaging
    - JMS Servers
      - Store and Forward Agents
      - JMS Modules
      - Path Services
    - Bridges
    - Data Sources
    - Persistent Stores
    - Foreign JNDI Providers

How do I...

- Configure quotas for destinations
- Configure JMS templates
- Configure destination keys
- Configure topics
- Configure queues
- Configure connection factories
- Configure uniform distributed topics
- Configure uniform distributed queues
- Configure foreign servers
- Configure JMS SAF

Settings for AuditJMS

Configuration Subdeployments Targets Security Notes

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

**Name:** AuditJMS The name of this JMS system module. [More Info...](#)

**Scope:** Global Specifies if the JMS system module is accessible within the domain, a partition, or a resource group template. [More Info...](#)

**Descriptor File Name:** Jms/auditjms-jms.xml The name of the JMS module descriptor file. [More Info...](#)

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-end-forward parameters.

Customize this table

Summary of Resources

New Delete Showing 1 to 3 of 3 Previous Next

<input type="checkbox"/>	Name ↕	Type	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	API_AUDIT_QUEUE	Uniform Distributed Queue	API_AUDIT_QUEUE	AuditSD	AuditJMSServer
<input type="checkbox"/>	AUDITQCF	Connection Factory	AUDITQCF	Default Targeting	obdx_cluster1
<input type="checkbox"/>	AUDIT_QUEUE	Uniform Distributed Queue	AUDIT_QUEUE	AuditSD	AuditJMSServer

New Delete Showing 1 to 3 of 3 Previous Next

### 3.11 Creating ReportsJMSServer JMS Server

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

ORACLE WebLogic Server Administration Console 12c

Home > FileUploadJMS > QCF > FileUploadJMS > placeholder > Summary of JMS Servers > Summary of JMS Modules > AuditJMS > Summary of JMS Modules > AuditJMS > Summary of JMS Servers

Messages

✔ JMS server created successfully

**Summary of JMS Servers**

JMS servers act as management containers for the queues and topics in JMS modules that are targeted to them. This page summarizes the JMS servers that have been created in the current WebLogic Server domain.

Customize this table

JMS Servers (Filtered - More Columns Exist)

Name	Persistent Store	Target	Current Target	Health
<input type="checkbox"/> AuditJMSServer	WLS_JMS_AUDIT_PS	obdx_server_1	obdx_server_1	
<input type="checkbox"/> FileUploadJMSServer	WLS_JMS_FILEUPLOAD_PS	obdx_server_1	obdx_server_1	
<input type="checkbox"/> ReportsJMSServer	WLS_JMS_REPORT_PS	obdx_server_1	obdx_server_1	

ORACLE WebLogic Server Administration Console 12c

Home > Summary of JMS Servers > Summary of JMS Modules > Summary of JMS Servers > Summary of JMS Bridge Destinations > Summary of Store-and-Forward Agents > Summary of JMS Modules > Summary of Persistent Stores > Summary of JMS Modules > ReportsJMSModule > Summary of JMS Modules

Messages

✔ JMS server created successfully

**Summary of JMS Modules**

JMS system resources are configured and stored as modules similar to standard Java EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources. This page summarizes the JMS system modules that have been created for this domain.

Customize this table

JMS Modules (Filtered - More Columns Exist)

Name	Type
<input type="checkbox"/> AuditJMS	JMSSystemResource
<input type="checkbox"/> FileUploadJMS	JMSSystemResource
<input type="checkbox"/> ReportsJMSModule	JMSSystemResource

to take effect.

✔ Connection factory created successfully.

**Settings for ReportsJMSModule**

Configuration Subdeployments Targets Security Notes

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

**Name:** ReportsJMSModule The name of this JMS system module. More Info...

**Scope:** Global Specifies if the JMS system module is accessible within the domain, a partition, or a resource group template. More Info...

**Descriptor File Name:** Jms/reportjmsmodule-jms.xml The name of the JMS module descriptor file. More Info...

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

Customize this table

**Summary of Resources**

Name	Type	JNDI Name	Subdeployment	Targets
<input type="checkbox"/> REPORTADHOC	Uniform Distributed Queue	REPORTADHOC	ReportsSubdeployment	ReportsJMSServer
<input type="checkbox"/> REPORTSCHEDULED	Uniform Distributed Queue	REPORTSCHEDULED	ReportsSubdeployment	ReportsJMSServer
<input type="checkbox"/> ReportsQCF	Connection Factory	ReportsQCF	Default Targeting	obdx_cluster1



- Under ReportsJMSModule create UniformDistributed Queue and connection factory as show above in the screen shot.

REPORTADHOC – Uniform Distributed Queue

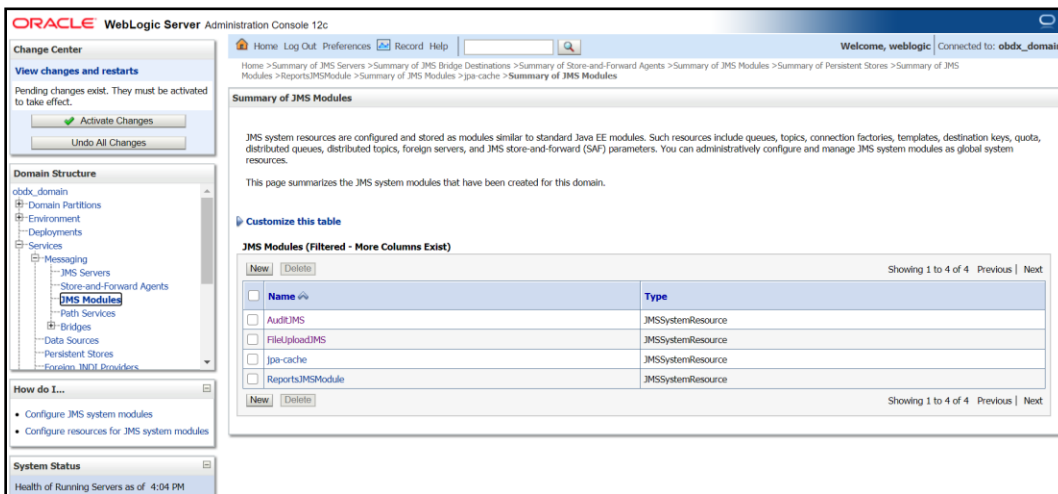
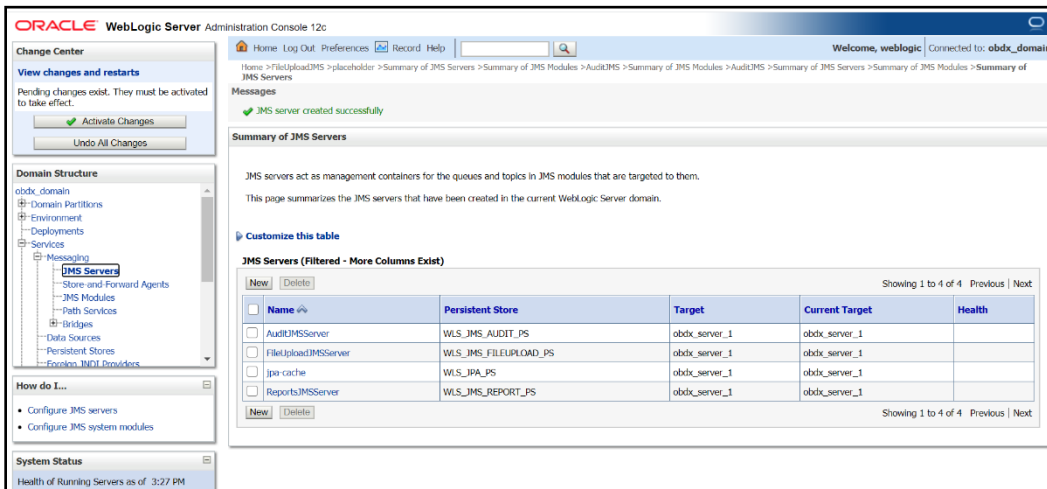
REPORTSCHEDULED -- Uniform Distributed Queue

ReportsQCF – Connection Factory

### 3.12 Creating jpa-cache JMS Server

### 3.13 Creating WLS\_JPA\_PS FileStore

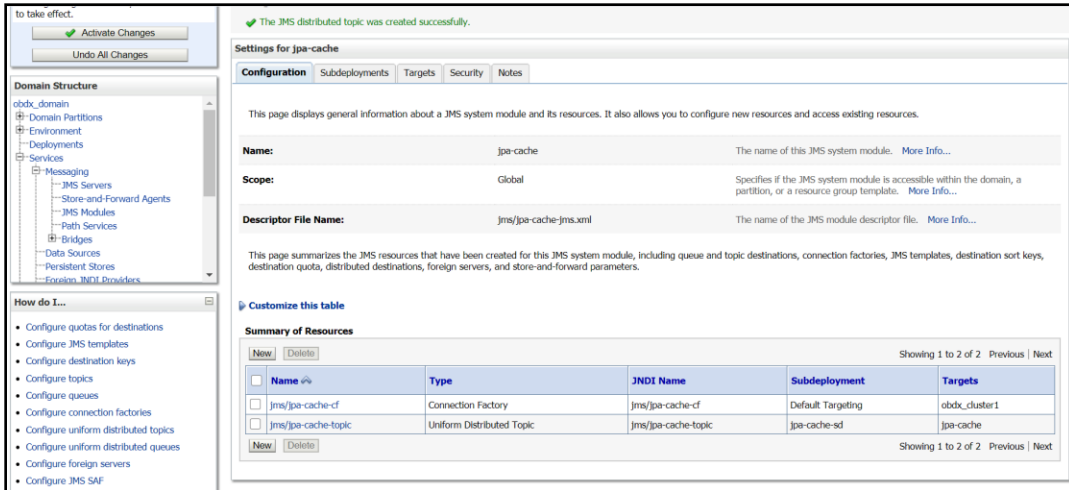
Create jpa-cache JMS server and jpa-cache JMS Module as show in below screen shot



- Under jpa-cache JMS Module create connection Factory and Uniform Distributed topic as shown in below screen shot

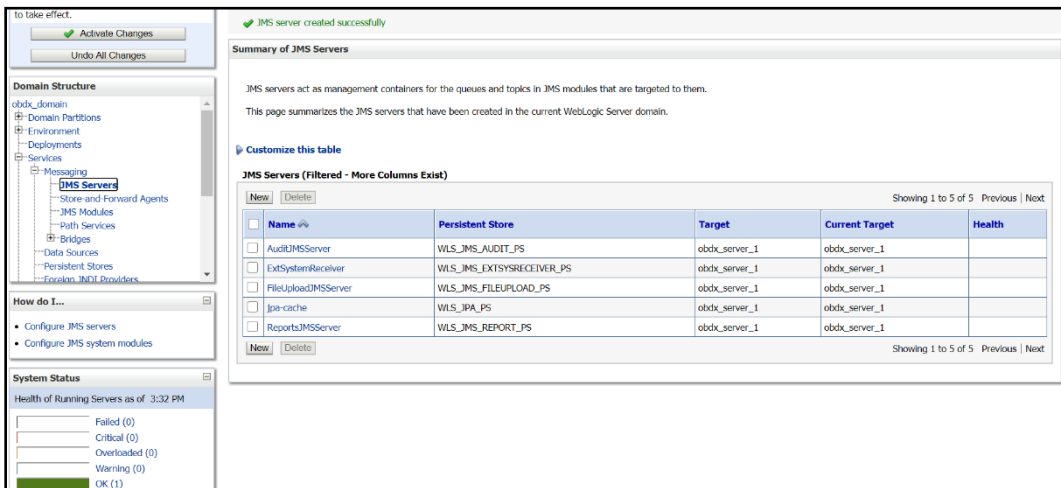
Jms/jpa-cache-cf --- Connection Factory

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



### 3.14 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

to take effect.

**Domain Structure**

- obdx\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
    - Messaging
      - JMS Servers
      - Store-and-Forward Agents
      - JMS Modules**
      - Path Services
      - Bridges
    - Data Sources
    - Persistent Stores
    - Foreign JNDI Providers

**How do I...**

- Configure JMS system modules
- Configure resources for JMS system modules

**System Status**

Health of Running Servers as of: 4:16 PM

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

**Summary of JMS Modules**

JMS system resources are configured and stored as modules similar to standard Java EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quota, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

**Customize this table**

**JMS Modules (Filtered - More Columns Exist)**

Showing 1 to 5 of 5 Previous | Next

<input type="checkbox"/>	Name ↕	Type
<input type="checkbox"/>	AuxiliaryJMS	JMSSystemResource
<input type="checkbox"/>	ExtSystemReceiver	JMSSystemResource
<input type="checkbox"/>	FileUploadJMS	JMSSystemResource
<input type="checkbox"/>	java-cache	JMSSystemResource
<input type="checkbox"/>	ReportsJMSModule	JMSSystemResource

Showing 1 to 5 of 5 Previous | Next

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

to take effect.

**Domain Structure**

- obdx\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
    - Messaging
      - JMS Servers
      - Store-and-Forward Agents
      - JMS Modules
      - Path Services
      - Bridges
      - Data Sources
      - Persistent Stores
      - Foreign JNDI Providers

**How do I...**

- Configure quotas for destinations
- Configure JMS templates
- Configure destination keys
- Configure topics
- Configure queues
- Configure connection factories
- Configure uniform distributed topics
- Configure uniform distributed queues
- Configure foreign servers
- Configure JMS SAF

**Settings for ExtSystemReceiver**

The JMS distributed queue was created successfully.

**Configuration** Subdeployments Targets Security Notes

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

**Name:** ExtSystemReceiver The name of this JMS system module. [More Info...](#)

**Scope:** Global Specifies if the JMS system module is accessible within the domain, a partition, or a resource group template. [More Info...](#)

**Descriptor File Name:** jms/extsystemreceiver-jms.xml The name of the JMS module descriptor file. [More Info...](#)

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quota, distributed destinations, foreign servers, and store-and-forward parameters.

**Customize this table**

**Summary of Resources**

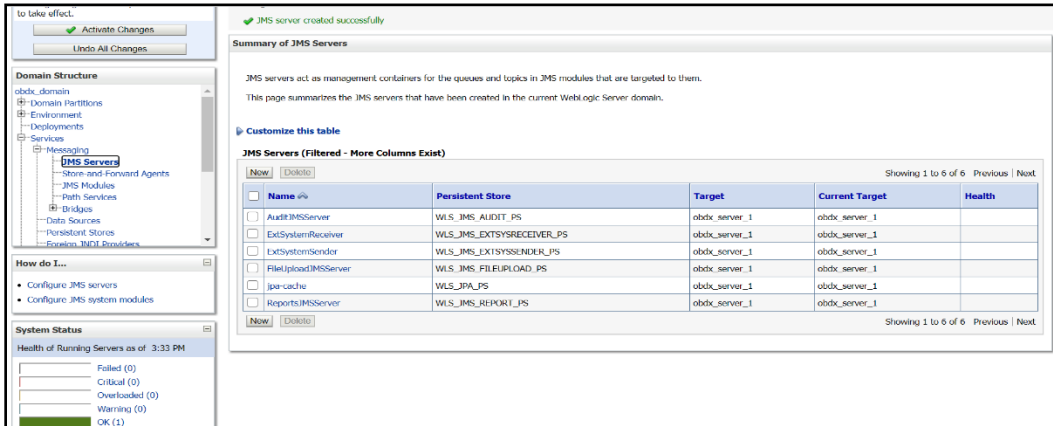
Showing 1 to 2 of 2 Previous | Next

<input type="checkbox"/>	Name ↕	Type	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	ExtSystemReceiverQCF	Connection Factory	ExtSystemReceiverQCF	Default Targeting	obdx_cluster1
<input type="checkbox"/>	ExtSystemReceiverQueue	Uniform Distributed Queue	ExtSystemReceiverQueue	ExtSystemReceiverSub	ExtSystemReceiver

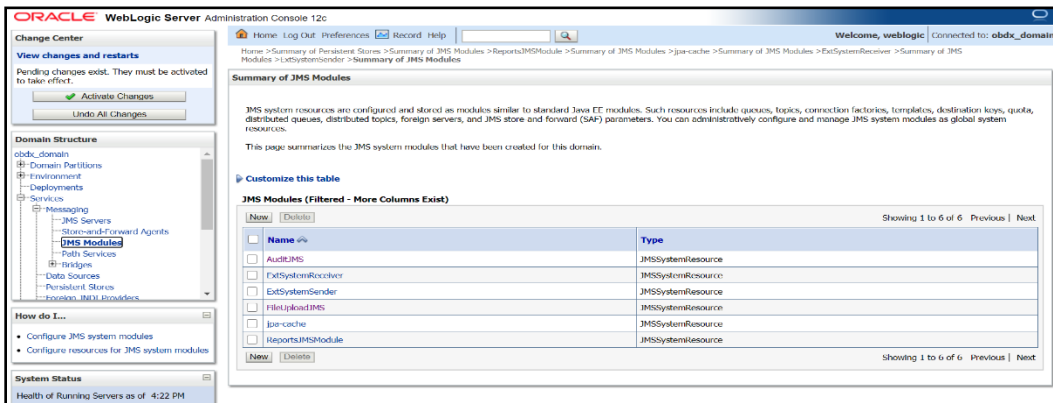
Showing 1 to 2 of 2 Previous | Next

### 3.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS\_JMS\_EXTSYSSENDER\_PS

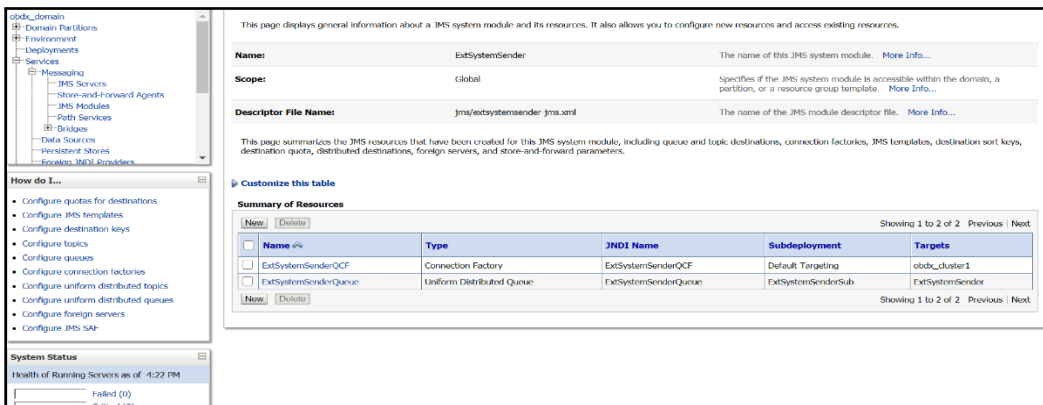
As show below create JMS Server ExtSystemSender



#### 1. Create ExtSystemSender JMS Module



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



### 3.16 Creating UBSSystemModule JMS Server

#### 1. In JMSModule create UBSSystemModule

to take effect.

**Domain Structure**

- obdx\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
    - Messaging
      - JMS Servers
      - Store-and-Forward Agents
      - JMS Modules**
      - Path Services
    - Bridges
    - Data Sources
    - Persistent Stores
    - Foreign JNDI Providers

**How do I...**

- Configure JMS system modules
- Configure resources for JMS system modules

**System Status**

Health of Running Servers as of 4:24 PM

- Failed (0)
- Critical (0)
- Overloaded (0)
- Warning (0)
- OK (1)

**Summary of JMS Modules**

JMS system resources are configured and stored as modules similar to standard Java EE modules. Such resources include queues, topics, connection factories, templates, destination keys, quotas, distributed queues, distributed topics, foreign servers, and JMS store-and-forward (SAF) parameters. You can administratively configure and manage JMS system modules as global system resources.

This page summarizes the JMS system modules that have been created for this domain.

**Customize this table**

**JMS Modules (Filtered - More Columns Exist)**

Showing 1 to 7 of 7 Previous | Next

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	AuditJMS	JMSSystemResource
<input type="checkbox"/>	ExtSystemReceiver	JMSSystemResource
<input type="checkbox"/>	ExtSystemSender	JMSSystemResource
<input type="checkbox"/>	FileUploadJMS	JMSSystemResource
<input type="checkbox"/>	Jpa-cache	JMSSystemResource
<input type="checkbox"/>	ReportsJMSModule	JMSSystemResource
<input type="checkbox"/>	UBSSystemModule	JMSSystemResource

Showing 1 to 7 of 7 Previous | Next

#### 2. Under UBSSystemModule create UBSSystemForeignServer – Foreign Server as shown below

to take effect.

**Domain Structure**

- obdx\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
    - Messaging
      - JMS Servers
      - Store-and-Forward Agents
      - JMS Modules
      - Path Services
    - Bridges
    - Data Sources
    - Persistent Stores
    - Foreign JNDI Providers

**How do I...**

- Configure quotas for destinations
- Configure JMS templates
- Configure destination keys
- Configure topics
- Configure queues
- Configure connection factories
- Configure uniform distributed topics
- Configure uniform distributed queues
- Configure foreign servers
- Configure JMS SAF

**Settings for UBSSystemModule**

The foreign server was created successfully.

**Configuration** Subdeployments Targets Security Notes

This page displays general information about a JMS system module and its resources. It also allows you to configure new resources and access existing resources.

**Name:** UBSSystemModule The name of this JMS system module. [More Info...](#)

**Scope:** Global Specifies if the JMS system module is accessible within the domain, a partition, or a resource group template. [More Info...](#)

**Descriptor File Name:** jms/ubssystemmodule-jms.xml The name of the JMS module descriptor file. [More Info...](#)

This page summarizes the JMS resources that have been created for this JMS system module, including queue and topic destinations, connection factories, JMS templates, destination sort keys, destination quotas, distributed destinations, foreign servers, and store-and-forward parameters.

**Customize this table**

**Summary of Resources**

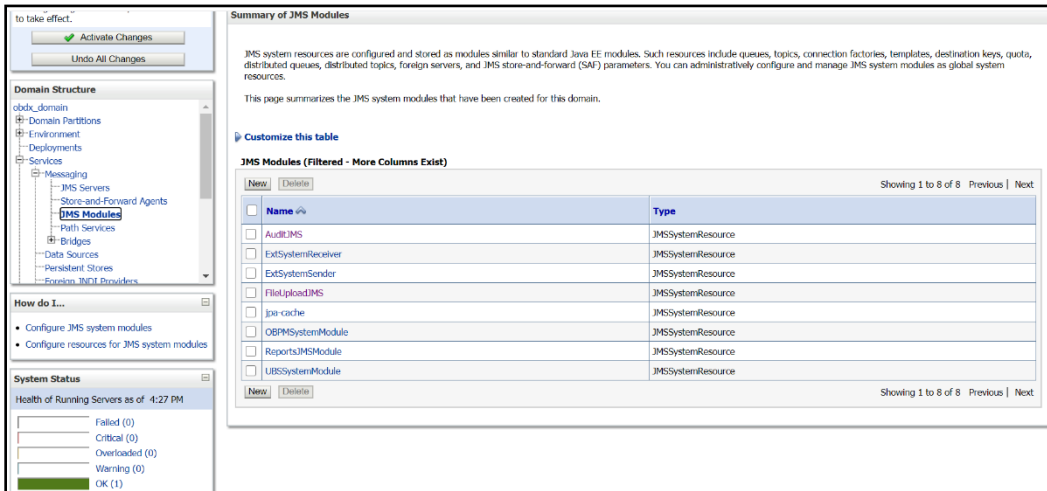
Showing 1 to 1 of 1 Previous | Next

<input type="checkbox"/>	Name	Type	JNDI Name	Subdeployment	Targets
<input type="checkbox"/>	UBSSystemForeignServer	Foreign Server	N/A	UBSSubdeployment	obdx_cluster1

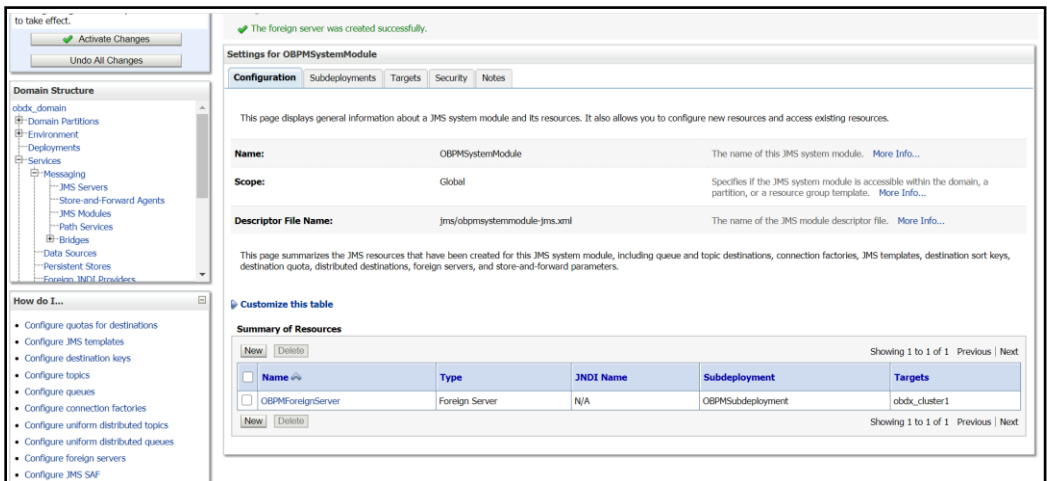
Showing 1 to 1 of 1 Previous | Next

### 3.17 Creating OBPMForeignServer JMS Server

1. In JMSModule create OBPMSystemModule



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



[Home](#)

## 4. Deploying Applications

### Deployment of Lib and Apps

`${MW_HOME}/wlsserver/common/deployable-libraries/jax-rs-2.0.war` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.framework.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.extsystem.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.core.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.thirdparty.app.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.rest.idm.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/BatchResourceAdapter.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/AuditMDBEAR.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/com.ofss.digx.app.connector.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.mdb.report.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.timer.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.oauth.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/obapi/deploy/obapi.app.oauth.rest.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/cz/obapi.cz.app.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/cz/obapi.cz.extsystem.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/cz/obapi.cz.thirdparty.app.domain.ear` (Target - obapi\_cluster, AdminServer)

`${OBAPI_INSTALLER}/installables/app/components/ubs/deploy/obapi.app.soap.ear` (Target - obapi\_cluster)

`${OBAPI_INSTALLER}/installables/app/components/thp/deploy/ExtxfaceSimulatorMDB.ear` (Target - obapi\_cluster)

## 5. Configured jps-config.xml

Update the jps-config.xml

Edit \$DOMAIN\_HOME/config/fmwconfig/jps-config.xml file and add following entries.

- 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.GenericIdentityStoreProvider">
      <description>Custom IdStore Provider</description>
      </serviceProvider>
      
```
- 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>
      
```
- find <jpsContext name="default"> tag in the file, add below serviceInstanceRef between <jpsContext name="default"></jpsContext>.
 

```

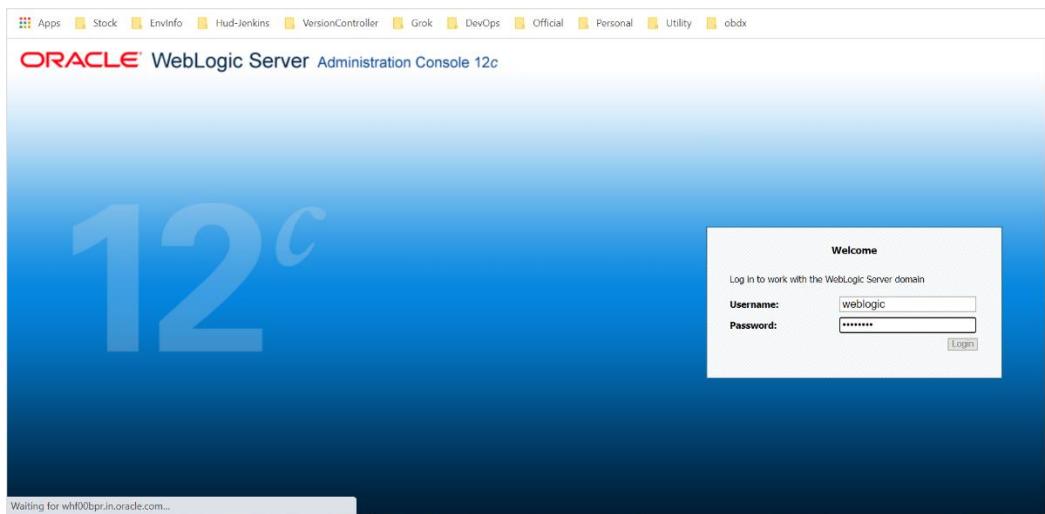
      <serviceInstanceRef ref="idstore.custom"/>
      
```

[Home](#)



## 6. Configured Providers

### 1. Logging into weblogic domain with admin credentials (ex. Weblogic)

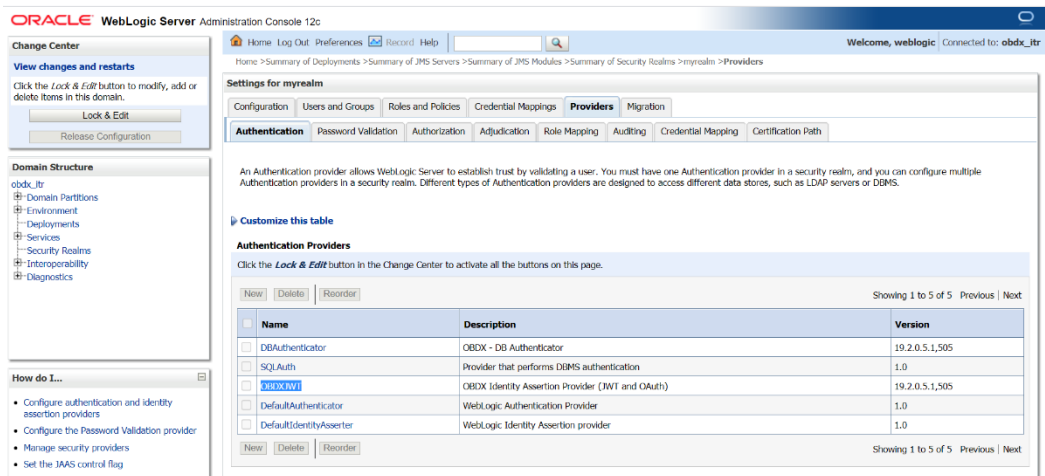


### 2. Navigate to Security Realms → myrealm → Providers → Create below providers.

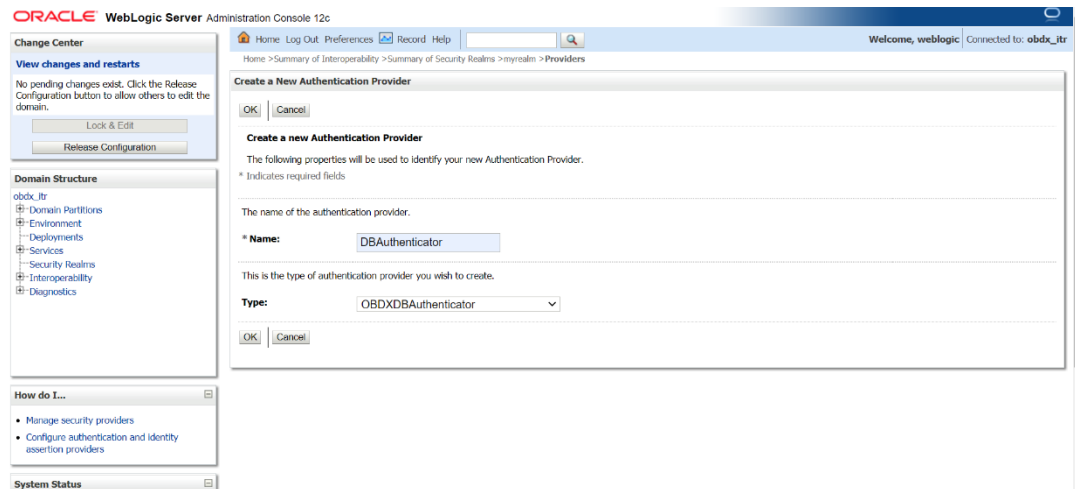
[DBAuthenticator](#)

[SQLAuth](#)

[OBAPIJWT](#)



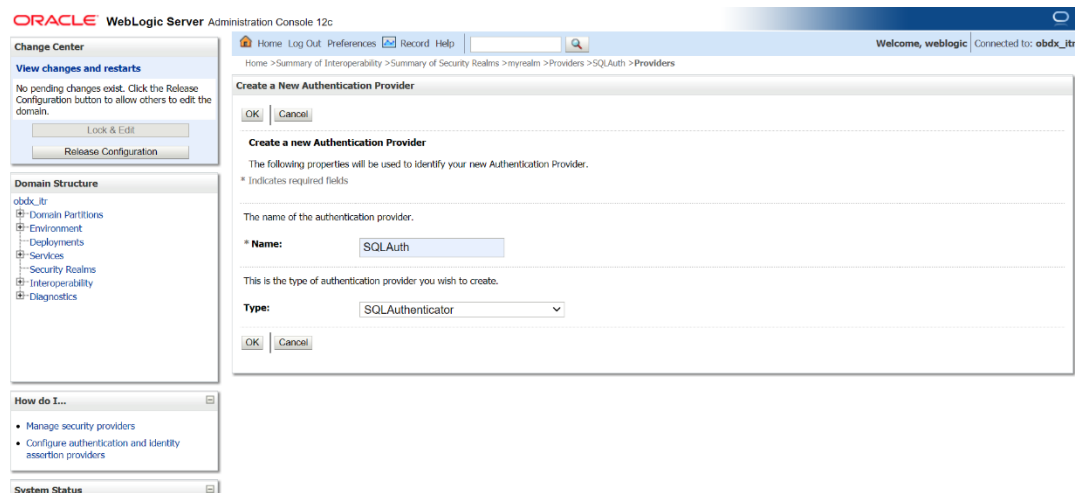
3. Click on New → Provide the field as show below



4. Click on OK.

5. Next create SQLAuth

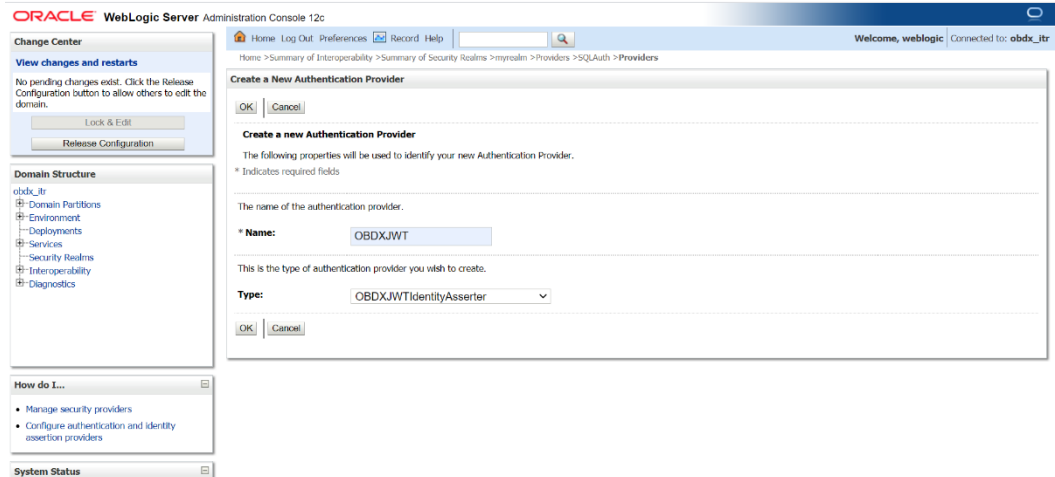
6. Click on New → and Provide the details as show below



7. Then click OK.

8. Create [OBAPIJWT](#) Providers.

9. Click on new → provide the details as show in the screen shot.



10. Once all the changes are done do complete domain restart.