

Installation Guide- Non-Linux Platforms  
Oracle Banking APIs  
Patchset Release 21.1.2.0.0

Part No. F40802-01

August 2021

**ORACLE®**

Installation Guide- Non-Linux Platforms

August 2021

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 © 2006, 2021, 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>1-1</b> |
| 1.1 Intended Audience.....   | 1-1        |
| 1.2 Documentation Accessibility.....   | 1-1        |
| 1.3 Access to Oracle Support.....  | 1-1        |
| 1.4 Structure .....  | 1-1        |
| 1.5 Related Information Sources.....   | 1-1        |
| <b>2. Manual OBAPI installation .....</b>                                      | <b>2-1</b> |
| 2.1 Create OBAPI Tablespace (file obapi_create_tablespace.sql) .....           | 2-1        |
| 2.2 Create Audit tablespace (file obapi_audit_create_tablespace.sql) .....     | 2-1        |
| 2.3 Create user (file obapi_create_user.sql) .....                             | 2-2        |
| 2.4 Create role (file obapi_create_role.sql) .....                             | 2-2        |
| 2.5 Grants Execution (file clip_user_grants.sql) .....                         | 2-3        |
| 2.6 Files execution in sequences on above schema (ex. OBAPI_\${POST_FIX})..... | 2-3        |
| 2.7 OBPM Database Installation (OBPM Favor) .....                              | 2-3        |
| 2.8 Tablespace Creation (file obpm_create_tablespace.sql) .....                | 2-4        |
| 2.9 CREATE BIGFILE TABLESPACE TBS_\${EHMS_SCHEMA_NAME }.....                   | 2-4        |
| 2.10 User Creation (file obpm_create_user.sql).....                            | 2-4        |
| 2.11 Create role (file obpm_create_role.sql) .....                             | 2-5        |
| 2.12 CREATE ROLE ROLE_\${ EHMS_SCHEMA_NAME } NOT IDENTIFIED;.....              | 2-5        |
| 2.13 Grants Execitions.....  | 2-5        |
| 2.14 Scripts Execution .....   | 2-5        |
| 2.15 Policy Seeding .....  | 2-6        |
| <b>3. WEBLOGIC Setup and Configuration .....</b>                               | <b>3-1</b> |

|           |  |            |
|-----------|--|------------|
| 3.1       | Setting Domain JTA Transaction timeout .....   | 3-1        |
| 3.2       | Creating DIGX data source .....  | 3-2        |
| 3.3       | Creating NONXA data source .....   | 3-5        |
| 3.4       | Creating BATCH data source .....   | 3-7        |
| 3.5       | Creating SYSCONFIG data source .....   | 3-11       |
| 3.6       | Creating B1A1 data source .....  | 3-14       |
| 3.7       | Create JMS server and JMS Module.....  | 3-17       |
| 3.8       | Creating WLS_JMS_AUDIT_PS FileStore .....  | 3-29       |
| 3.9       | Creating AuditJMSServer JMS Server .....   | 3-29       |
| 3.10      | Creating WLS_JMS_REPORT_PS FileStore.....  | 3-29       |
| 3.11      | Creating ReportsJMSServer JMS Server.....  | 3-40       |
| 3.12      | Creating jpa-cache JMS Server.....   | 3-42       |
| 3.13      | Creating WLS_JPA_PS FileStore .....  | 3-42       |
| 3.14      | Creating ExtSystemReceiver JMS Server -- WLS_JMS_EXTSYSRECEIVER_PS FileStore .                     | 3-44       |
| 3.15      | Creating ExtSystemSender JMS Server Persistent Store FileStore as<br>WLS_JMS_EXTSYSSENDER_PS ..... | 3-46       |
| 3.16      | Creating UBSForeignServer JMS Server .....   | 3-47       |
| 3.17      | Creating OBPMForeignServer JMS Server.....   | 3-48       |
| <b>4.</b> | <b>Deploying Applications .....</b>  | <b>4-1</b> |
| <b>5.</b> | <b>Configured jps-config.xml .....</b>   | <b>5-1</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 Oracle 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 Patchset Release 21.1.2.0.0, refer to the following documents:

- Oracle Banking APIs Installation Manuals

---

## 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 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 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 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 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'
```

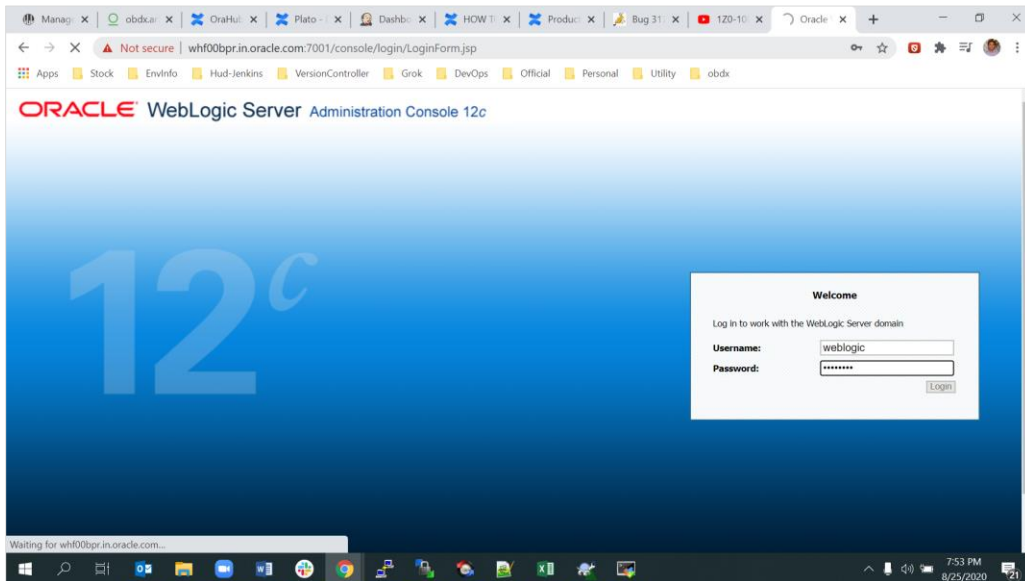
[Home](#)

# 3. WEBLOGIC Setup and Configuration

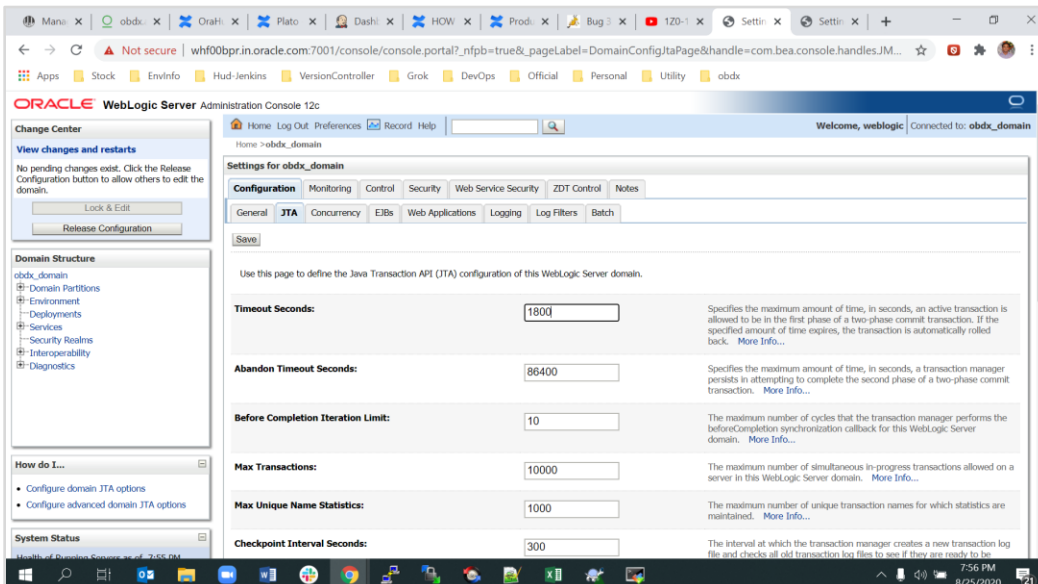
Assuming that rcu , weblogic domain created , managed server, cluster created , node manager configured.

## 3.1 Setting Domain JTA Transaction timeout

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

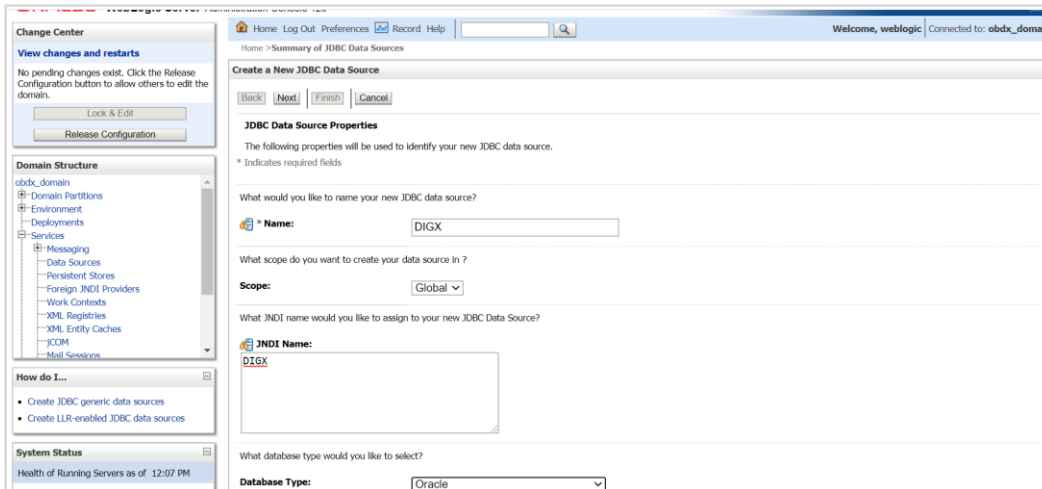


2. click on DOMAIN\_NAME → JTA → set Timeout Seconds to 1800 → click on save → Activate changes



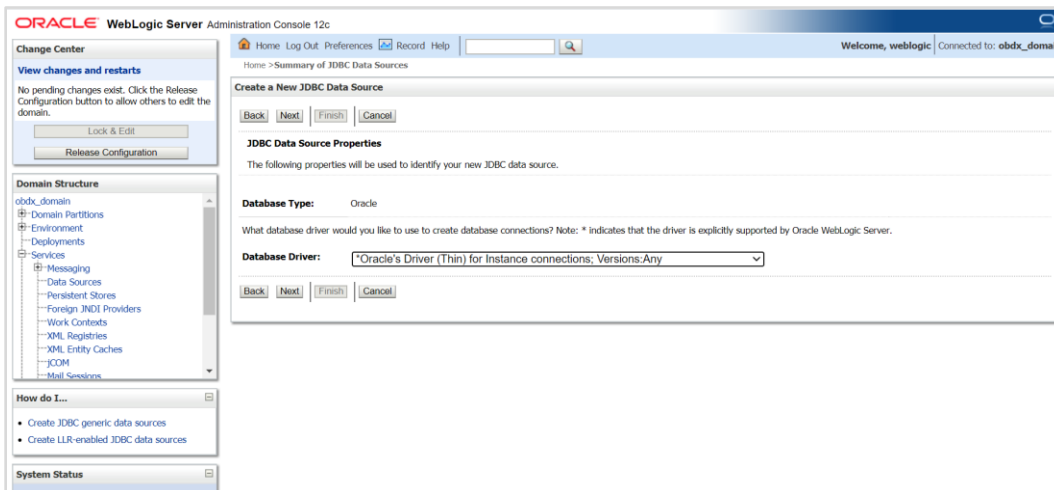
### 3.2 Creating DIGX data source

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

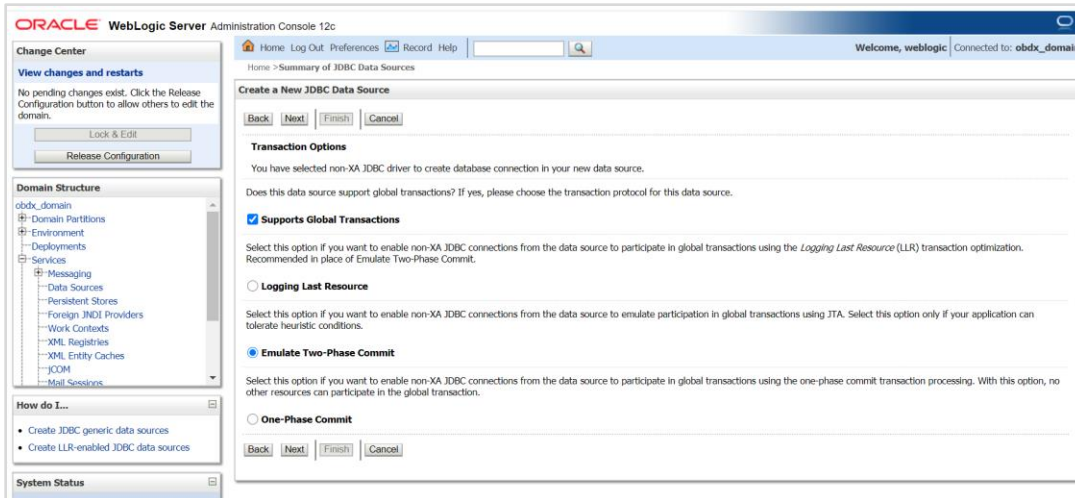


2. Name: - DIGX

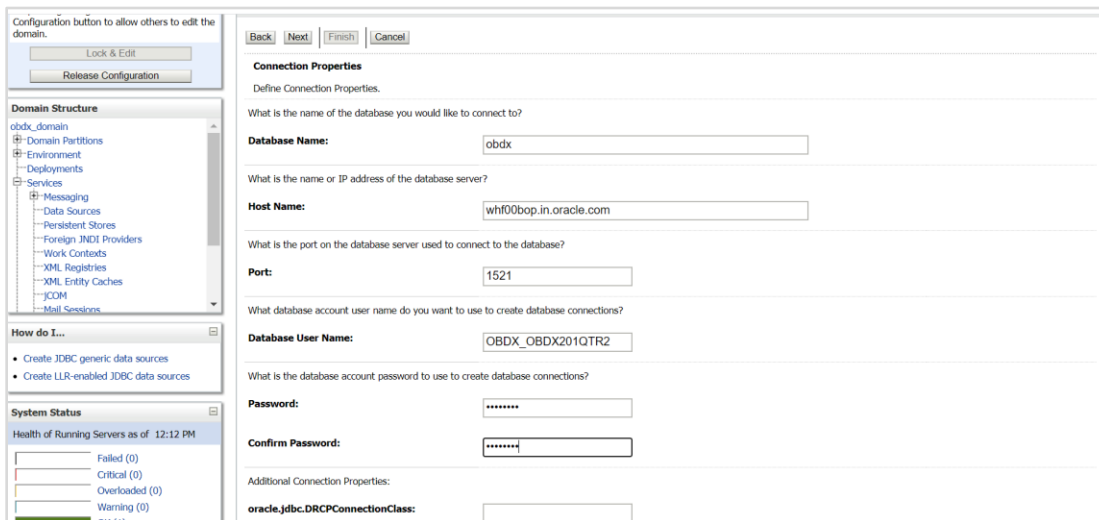
JNDI Name: - DIGX



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

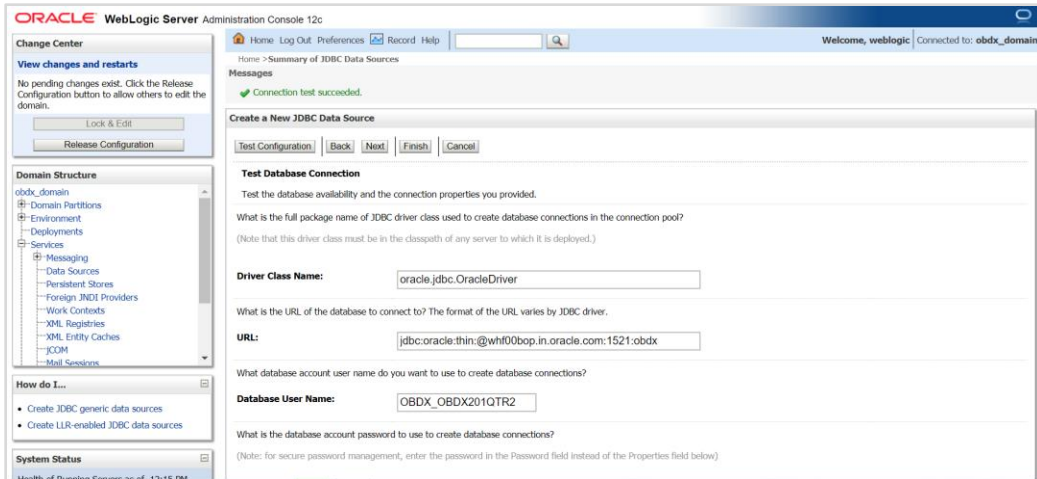


4. Select Emulate Two-Phase Commit

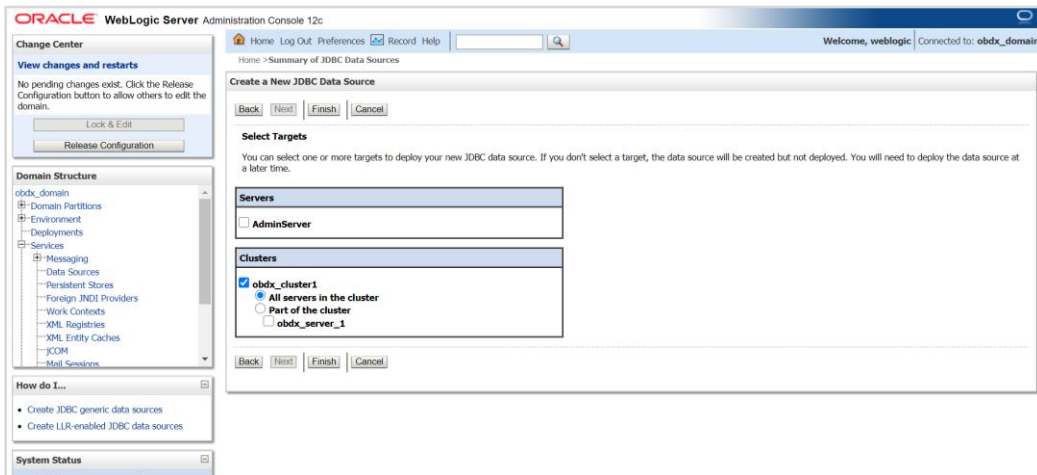


5. Provide

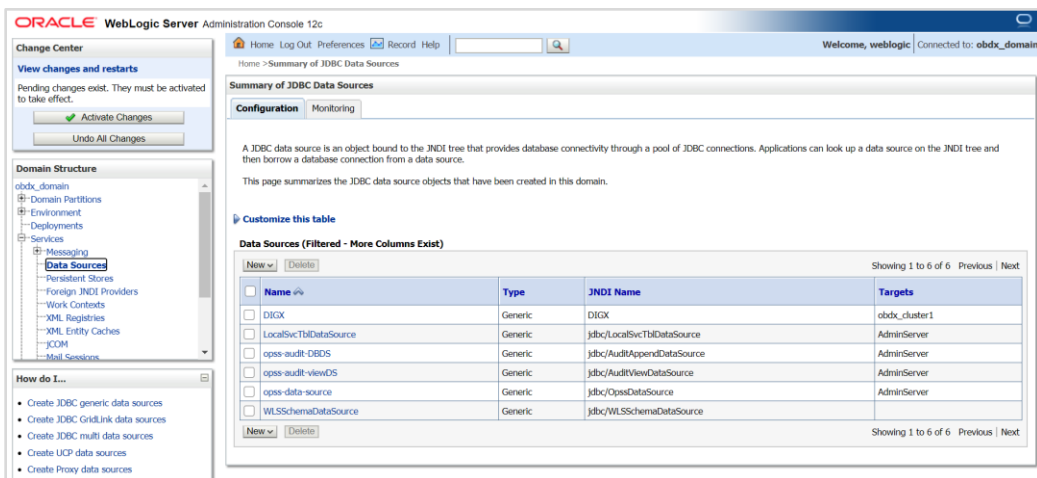
- Database Name:** - Database SID
- Host Name:** - Database hostname
- Port:** - Database port Number
- Database user Name:** - OBAPI\_\${POST\_FIX}



## 6. Test Configuration

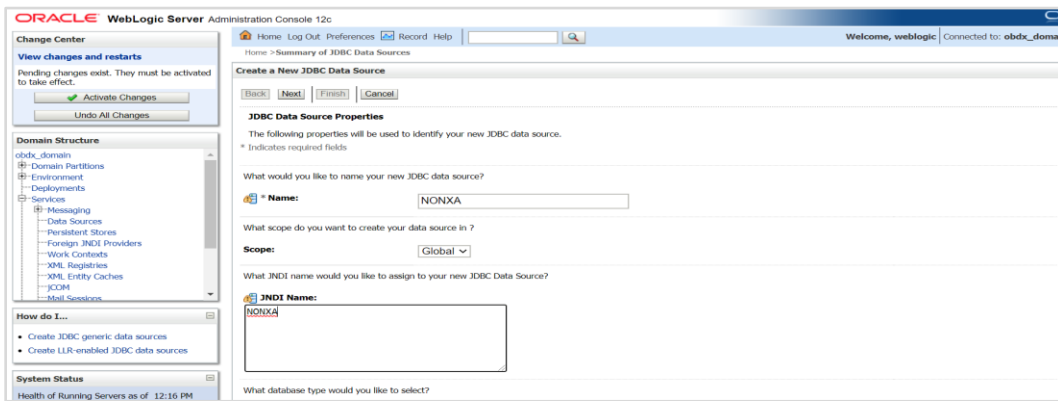


## 7. Target to cluster



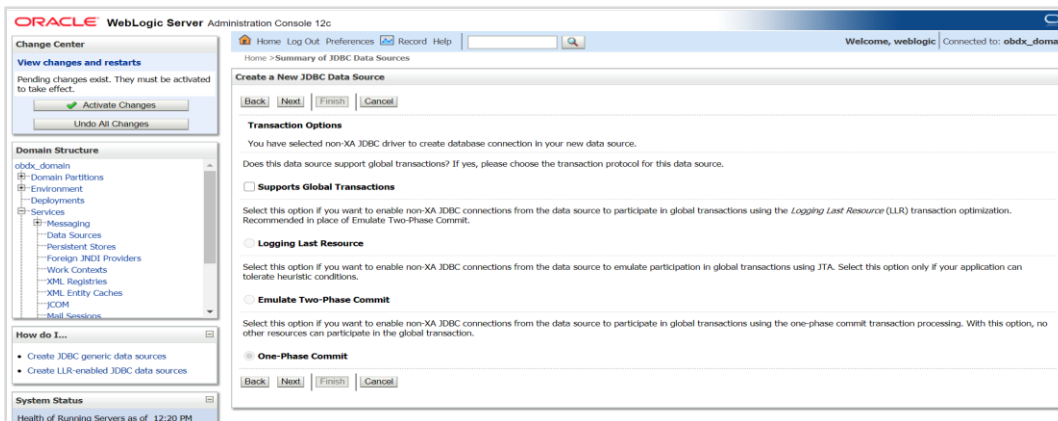
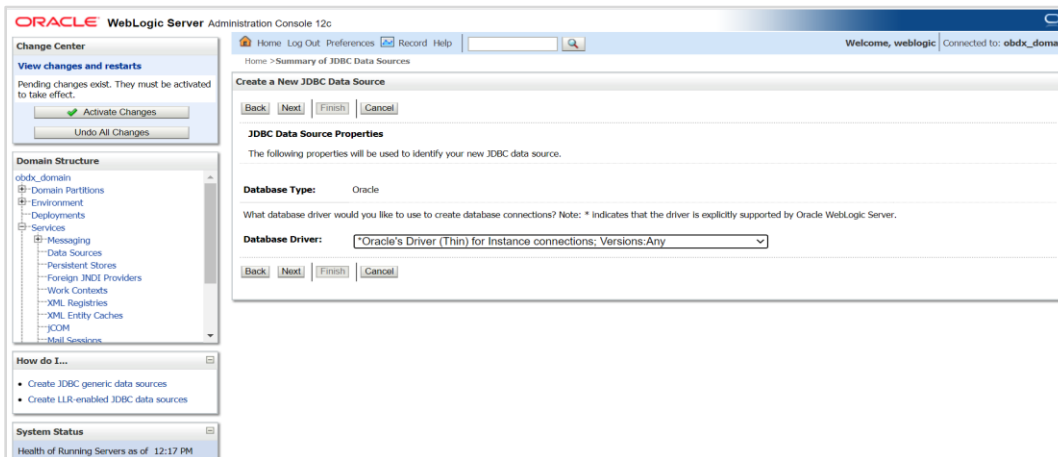
### 3.3 Creating NONXA data source

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



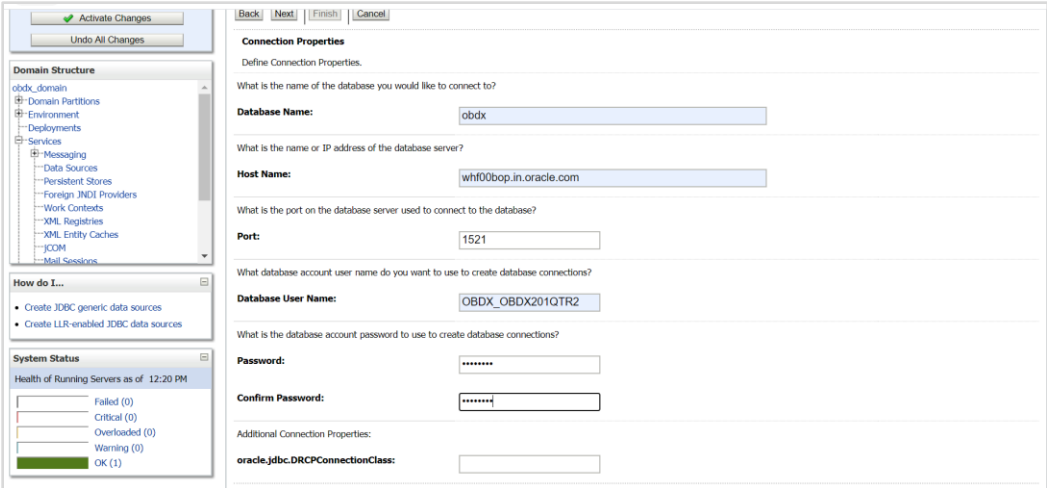
2. Name :- NONXA

JNDI Name :- NONXA



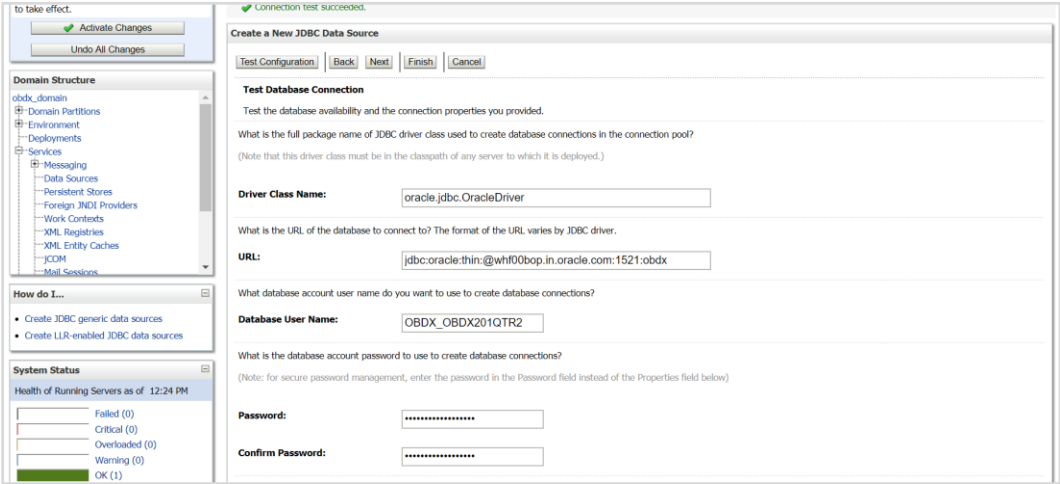
3. Click Next



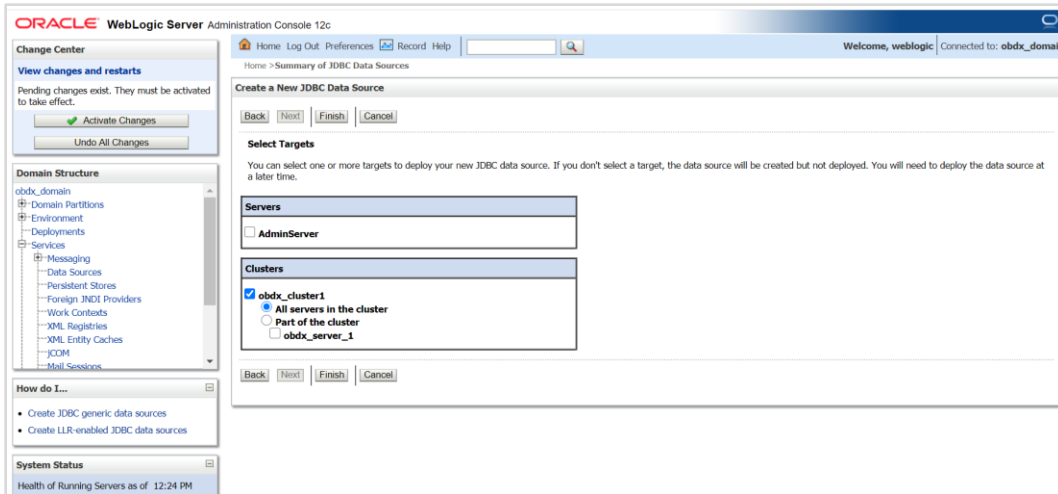


4. Provide

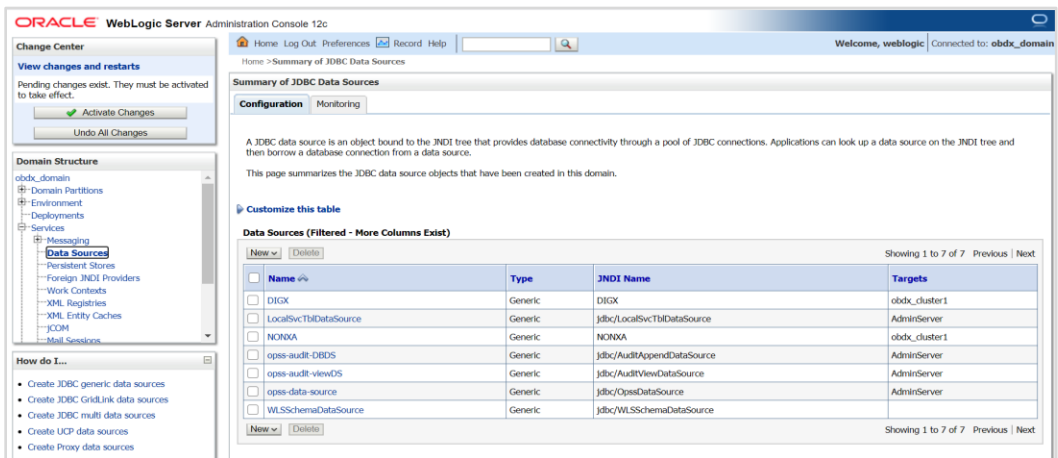
- Database Name:** - Database SID
- Host Name:** - Database hostname
- Port:** - Database port Number
- Database user Name:** - OBAPI\_\${POST\_FIX}
- Password:-** Database user password



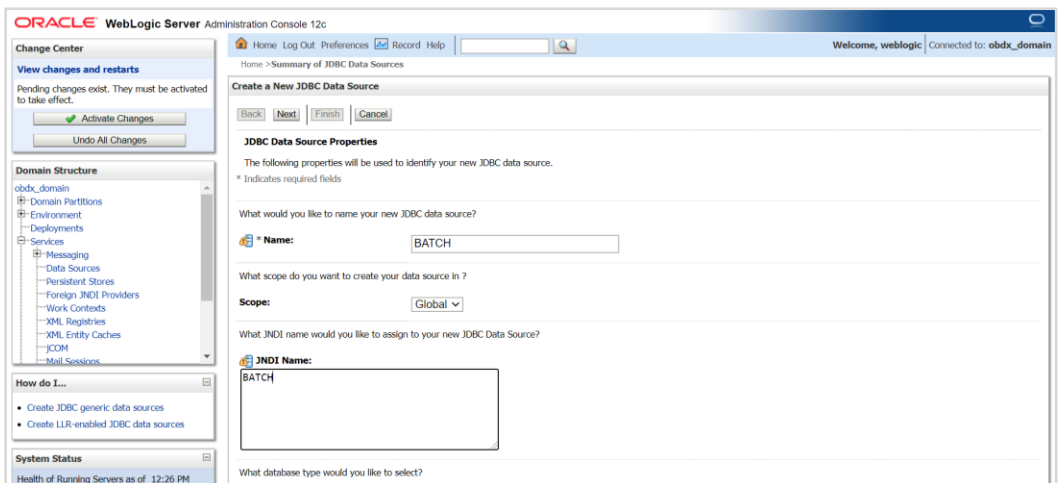
5. Test Configuration



6. Select target as cluster --> Finish

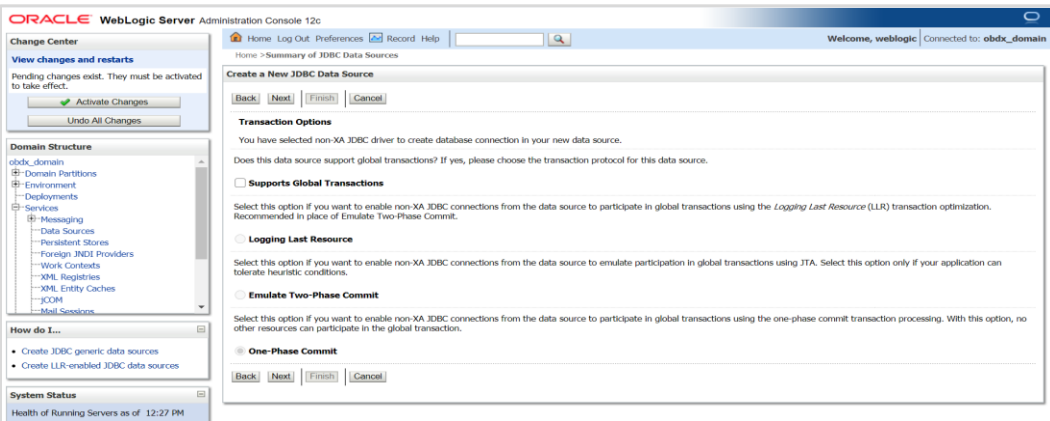
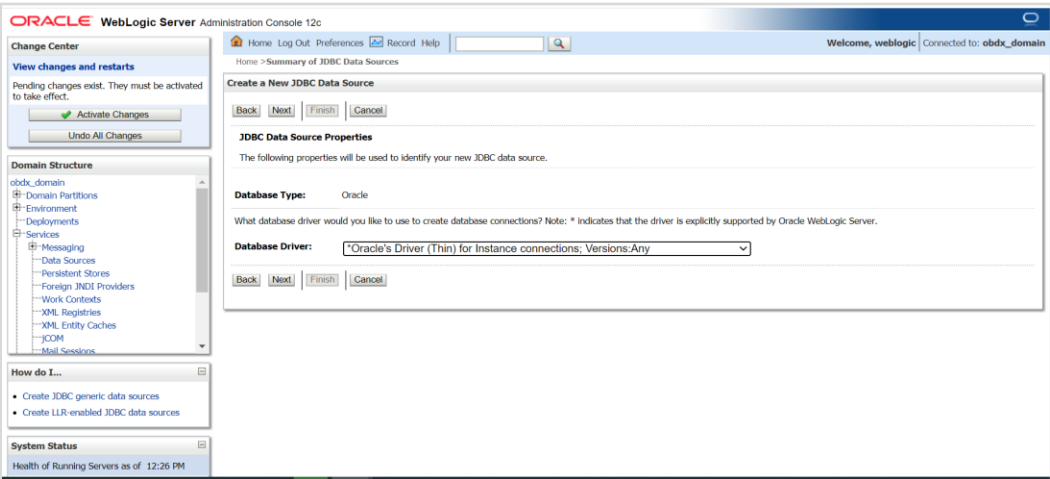


### 3.4 Creating BATCH data source

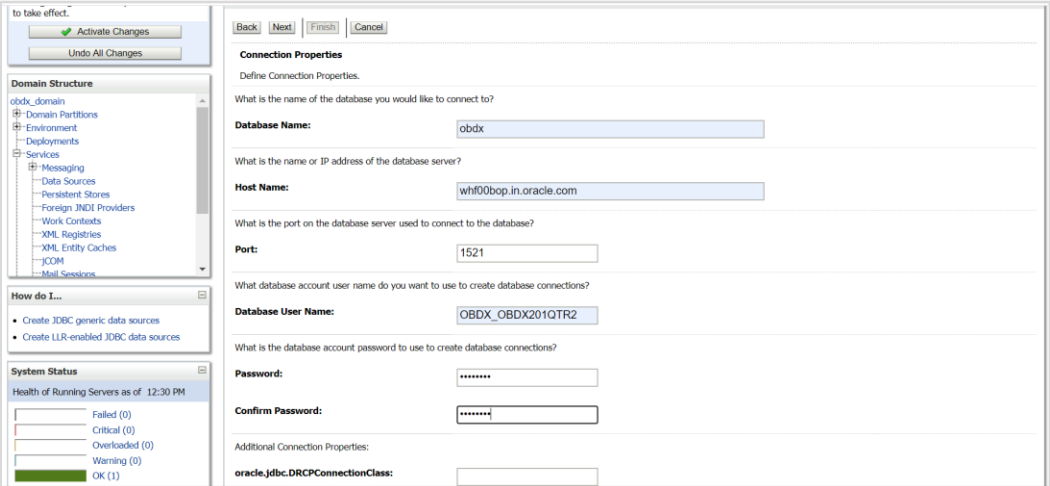


7. Name :- BATCH

JNDI Name :- BATCH



8. Click Next



## 9. Provide

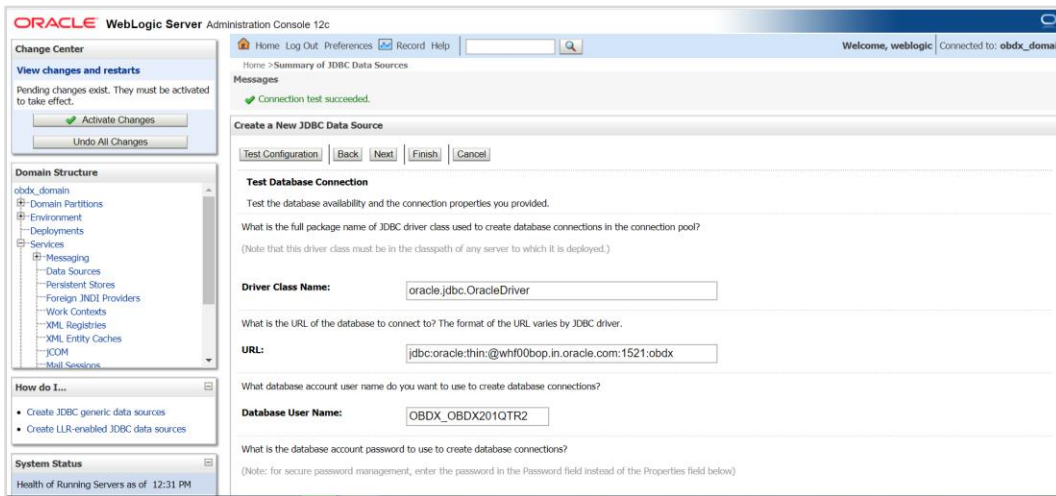
**Database Name:** - Database SID

**Host Name:** - Database hostname

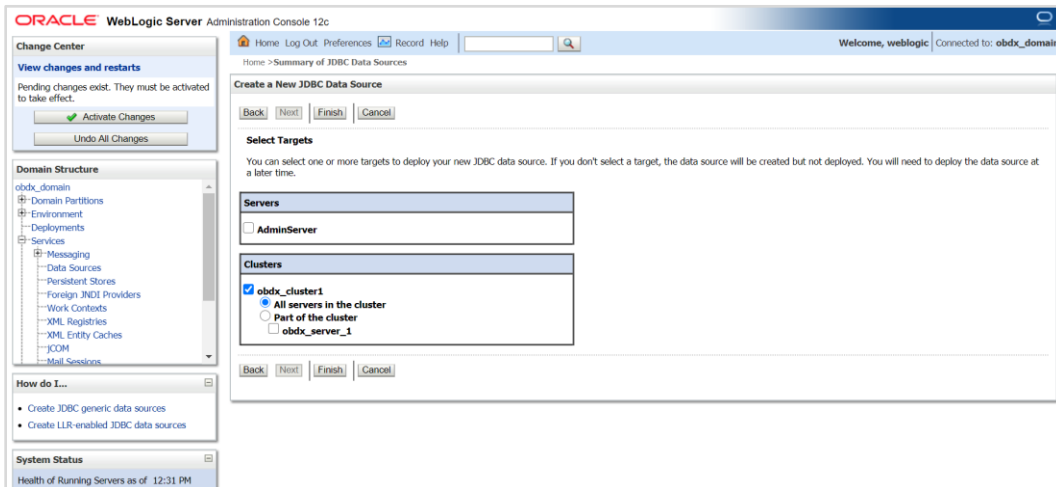
**Port:** - Database port Number

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

**Password:-** Database user password



## 10. Test Configuration



## 11. Target Cluster and click on Finish

to take effect.

**Domain Structure**

- obdx\_domain
  - Domain Partitions
  - Environment
    - Deployments
    - Services
      - Data Sources**
      - Persistent Stores
      - Foreign JNDI Providers
      - Work Contexts
      - XML Registries
      - XML Entity Caches
      - JCOM
      - Mail Sessions

**How do I...**

- Create JDBC generic data sources
- Create JDBC GridLink data sources
- Create JDBC multi data sources
- Create UCP data sources
- Create Proxy data sources

**System Status**

Health of Running Servers as of 12:31 PM

Failed (0)

Critical (0)

---

**Configuration** | Monitoring

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC connections. Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source.

This page summarizes the JDBC data source objects that have been created in this domain.

**Customize this table**

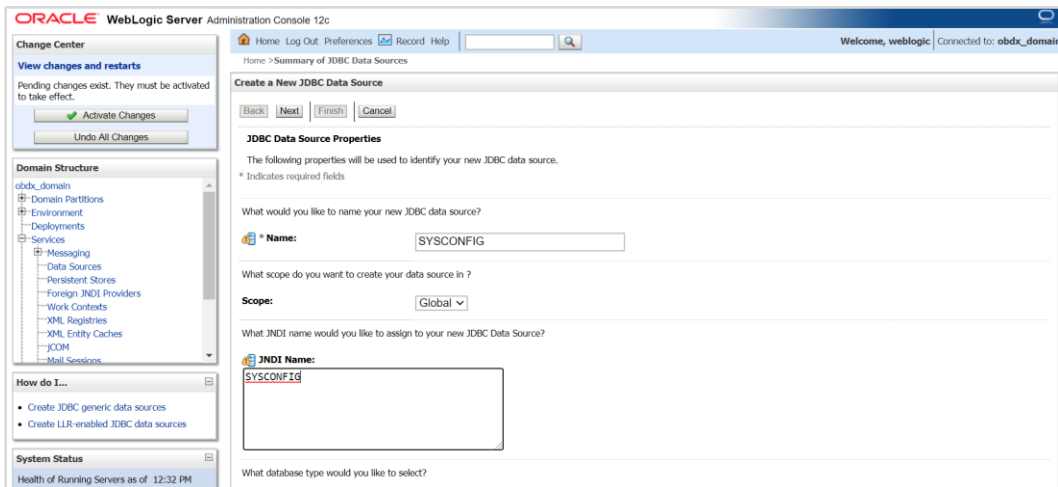
**Data Sources (Filtered - More Columns Exist)**

Showing 1 to 8 of 8 Previous Next

| <input type="checkbox"/> | Name ↕                | Type    | JNDI Name                  | Targets       |
|--------------------------|-----------------------|---------|----------------------------|---------------|
| <input type="checkbox"/> | BATCH                 | Generic | BATCH                      | obdx_cluster1 |
| <input type="checkbox"/> | DIGX                  | Generic | DIGX                       | obdx_cluster1 |
| <input type="checkbox"/> | LocalSvcTblDataSource | Generic | jdbc/LocalSvcTblDataSource | AdminServer   |
| <input type="checkbox"/> | NONXA                 | Generic | NONXA                      | obdx_cluster1 |
| <input type="checkbox"/> | opss-audit-DBDS       | Generic | jdbc/AuditAppendDataSource | AdminServer   |
| <input type="checkbox"/> | opss-audit-viewDS     | Generic | jdbc/AuditViewDataSource   | AdminServer   |
| <input type="checkbox"/> | opss-data-source      | Generic | jdbc/OpssDataSource        | AdminServer   |
| <input type="checkbox"/> | WLSschemaDataSource   | Generic | jdbc/WLSschemaDataSource   |               |

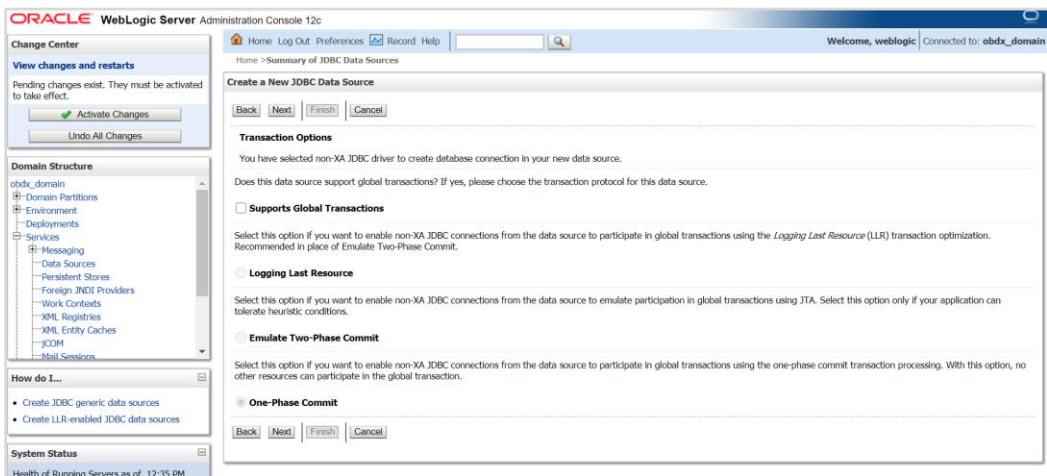
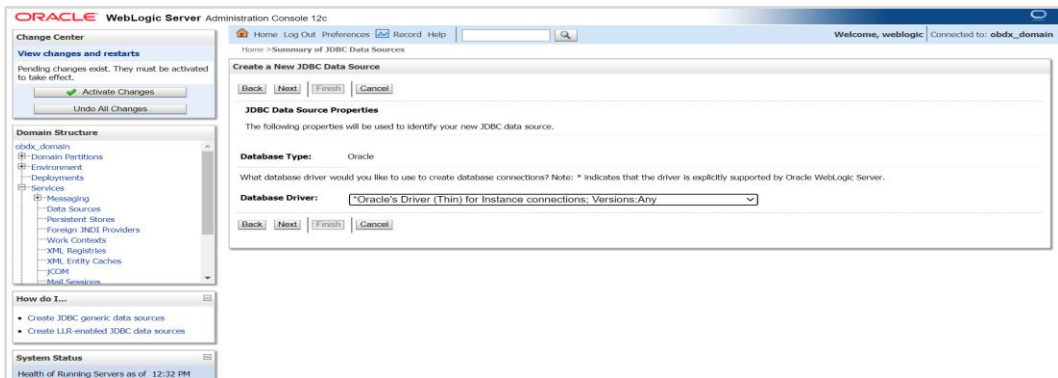
Showing 1 to 8 of 8 Previous Next

## 3.5 Creating SYSCONFIG data source

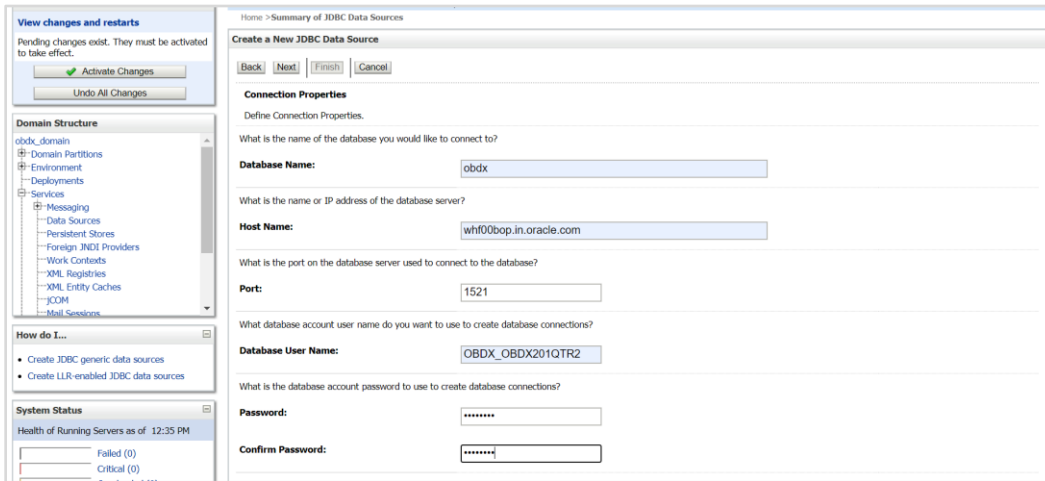


12. Name :- SYSCONFIG

JNDI Name :- SYSCONFIG



13. Click on Next



14. Provide

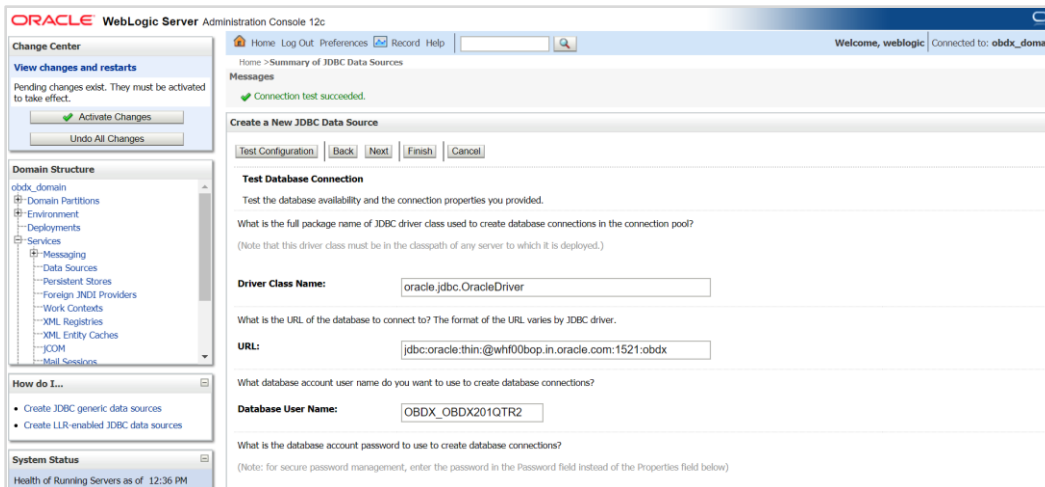
**Database Name:** - Database SID

**Host Name:** - Database hostname

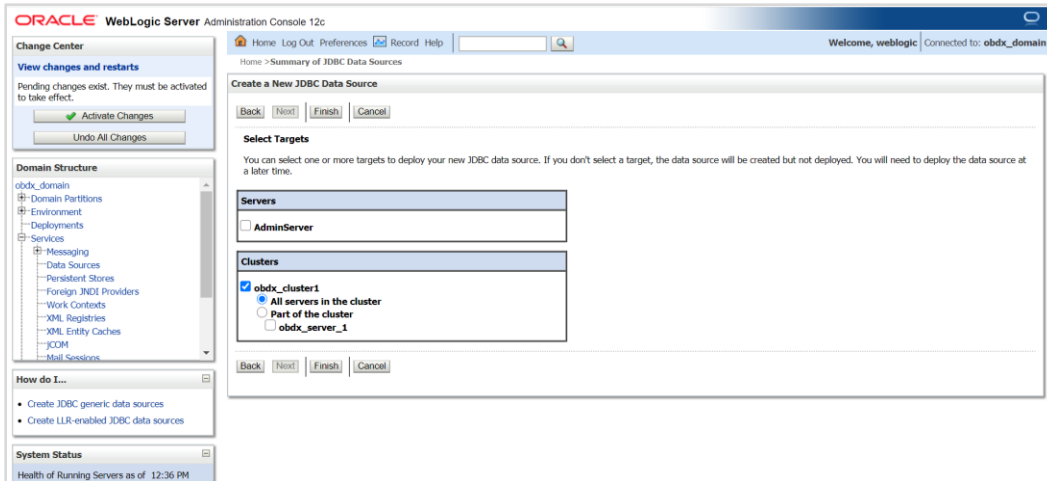
**Port:** - Database port Number

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

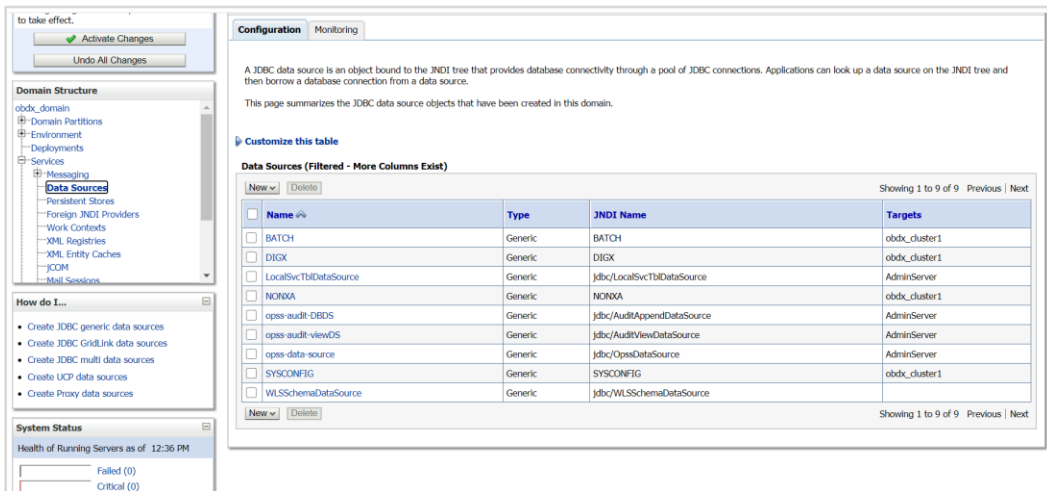
**Password:-** Database user password



15. Test Configuration

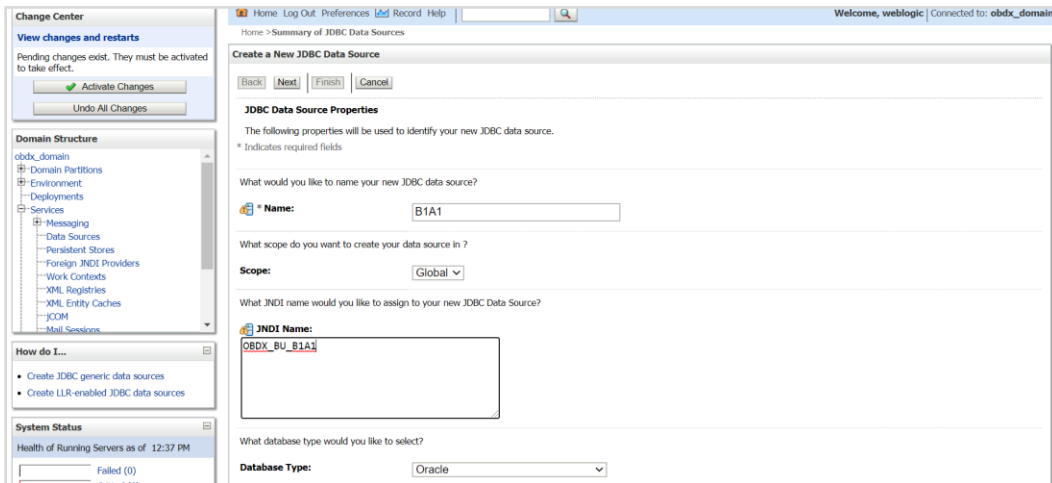


16. Select target as cluster and click on Finish



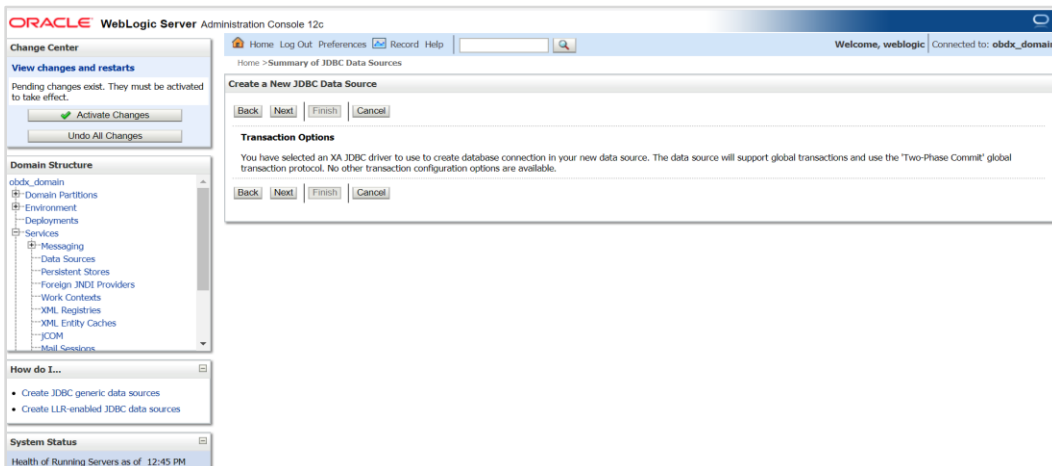
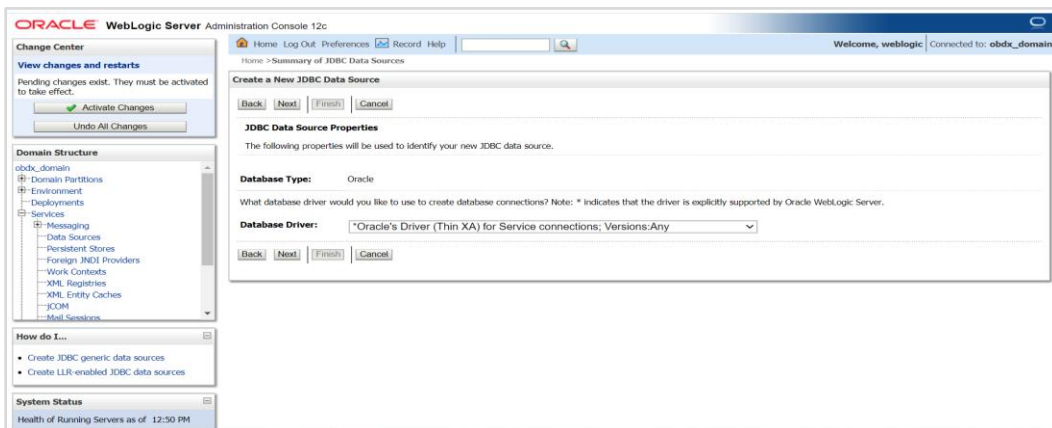


### 3.6 Creating B1A1 data source



17. Name:- B1A1

JNDI Name :- OBDX\_BU\_B1A1



18. Click on Next

19. Provide

**Database Name:** - Database SID (\$EHMS\_DATABASE\_SID)

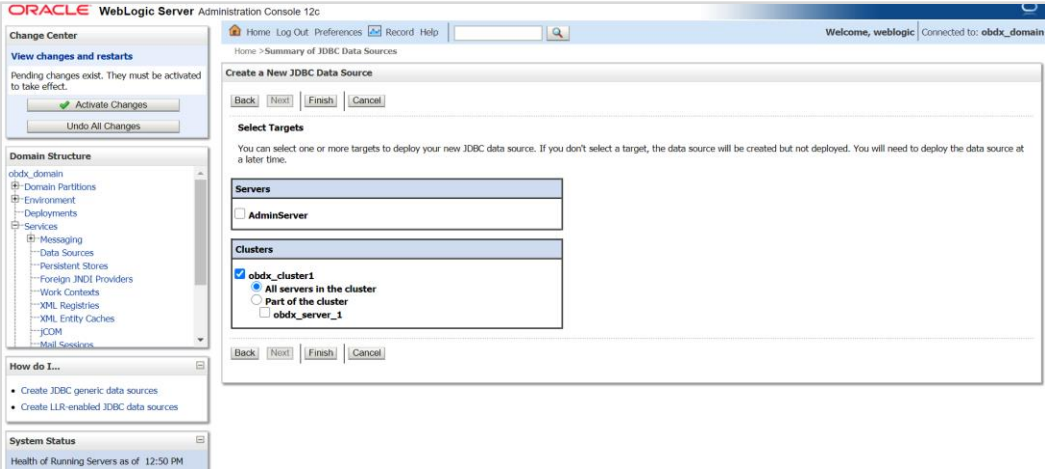
**Host Name:** - Database hostname (\$EHMS\_DATABASE\_HOSTNAME)

**Port:** - Database port Number (\$EHMS\_DATABASE\_PORT)

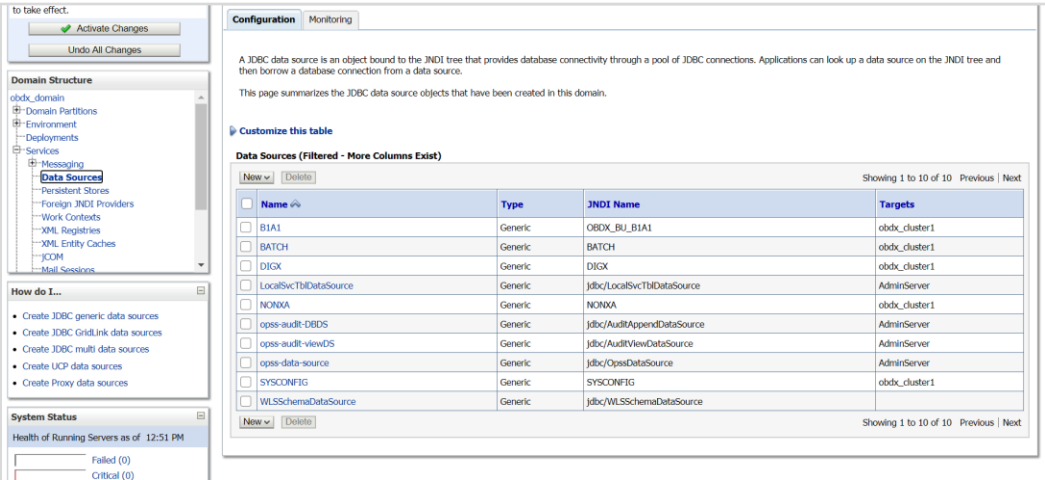
**Database user Name:** - \${ EHMS\_SCHEMA\_NAME }

**Password:** - Database user \${ EHMS\_SCHEMA\_NAME } password

20. Test Configuration

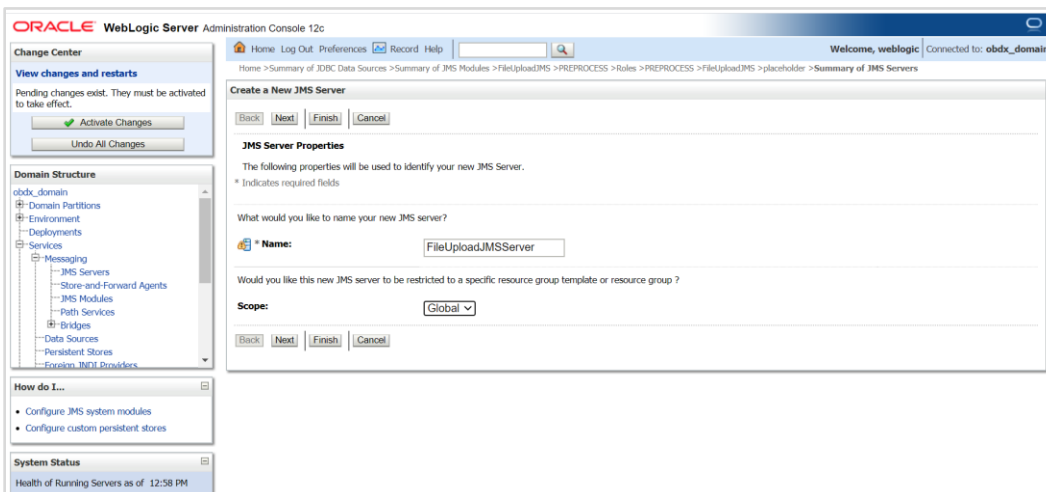
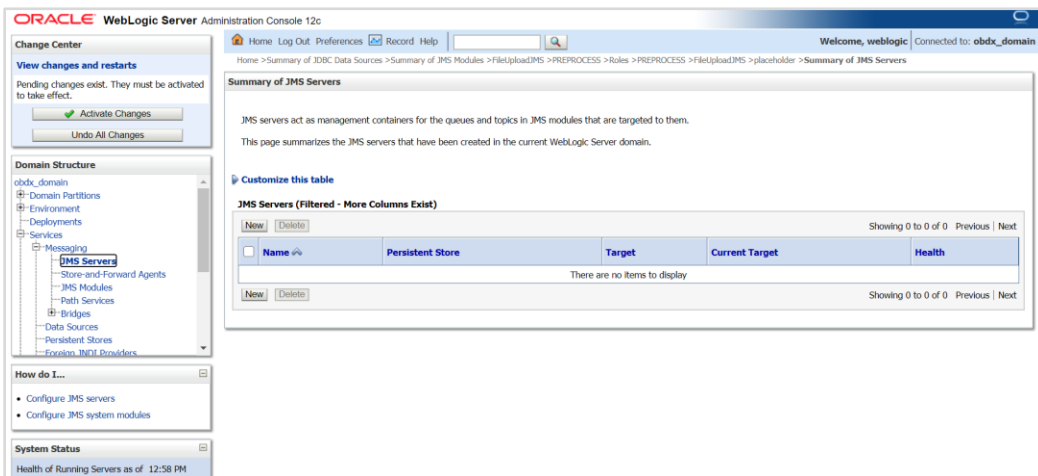


21. Set target as cluster and click on Finish

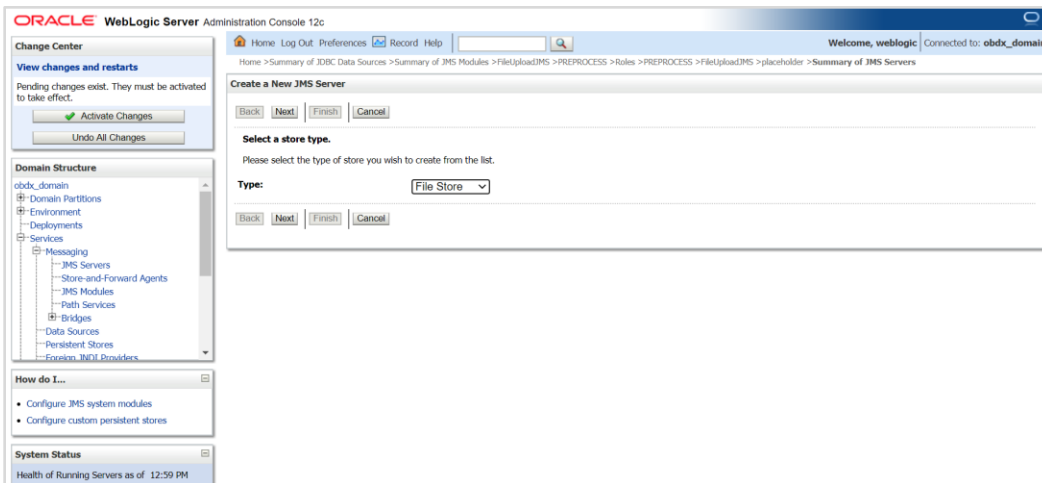
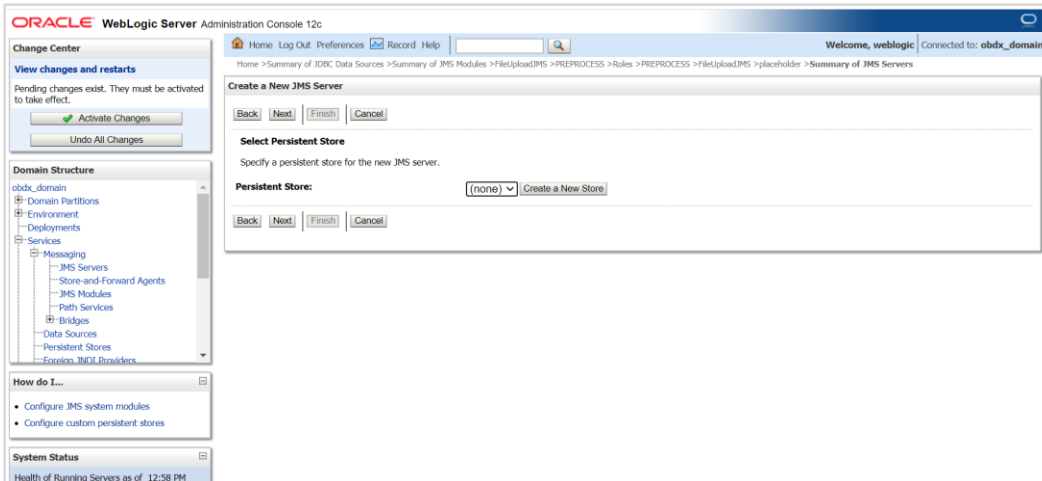


### 3.7 Create JMS server and JMS Module

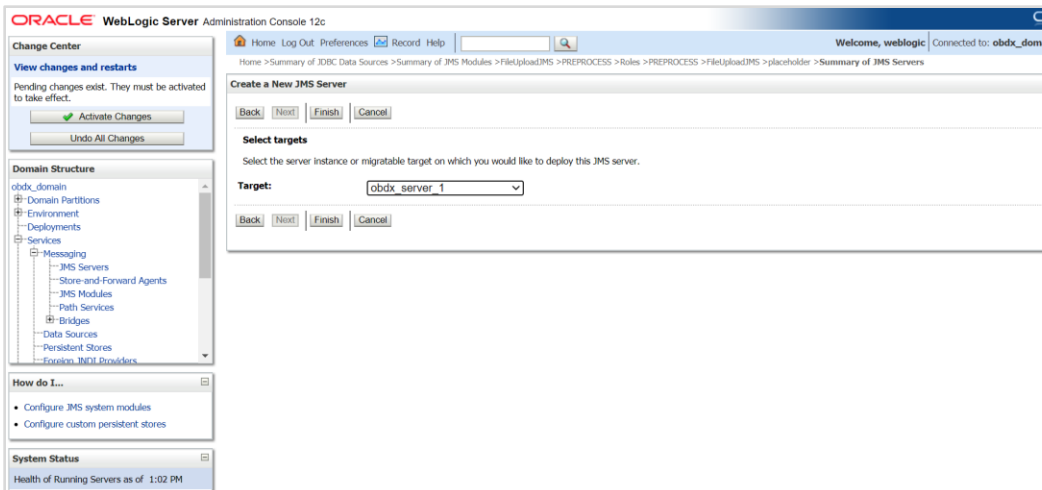
- Creating FileUploadJMS JSM Module
- Creating WLS\_JMS\_FILEUPLOAD\_PS FileStore
- Creating FileUploadJMSServer JMS Server



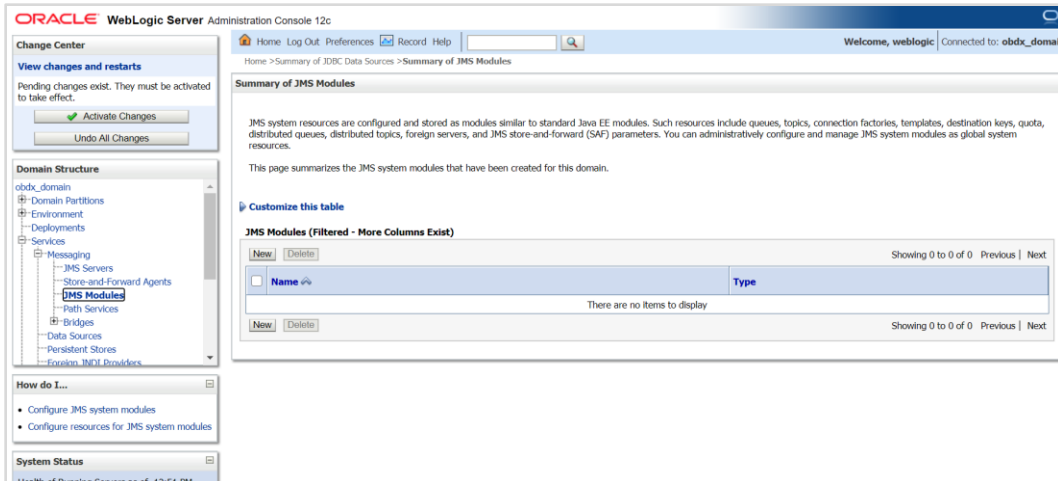
22. Click on JMS Servers → Name – FileUploadJMSServer -- > Click on Next



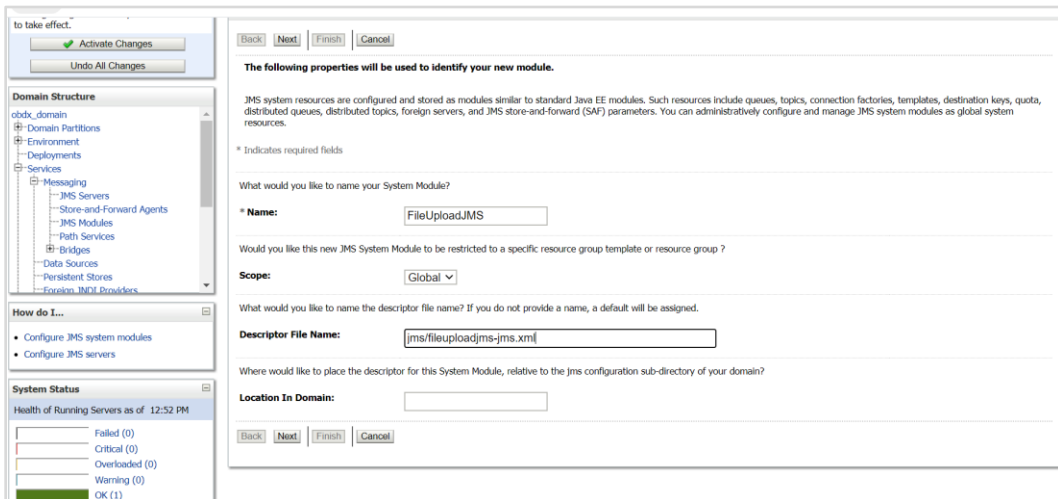
23. Select Type as File Store and click on Next



24. Select target as managed server and click on Finish



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

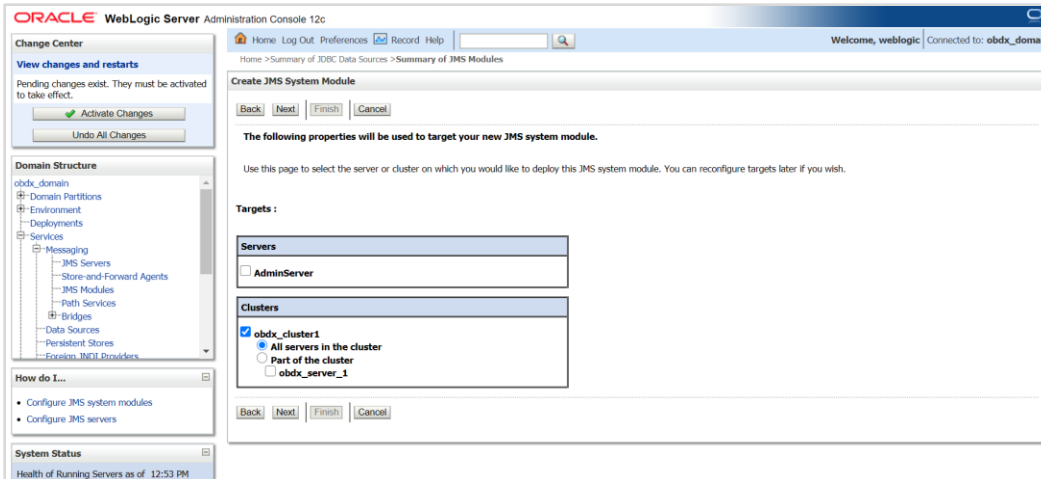


26. Name :- FileUploadJMS

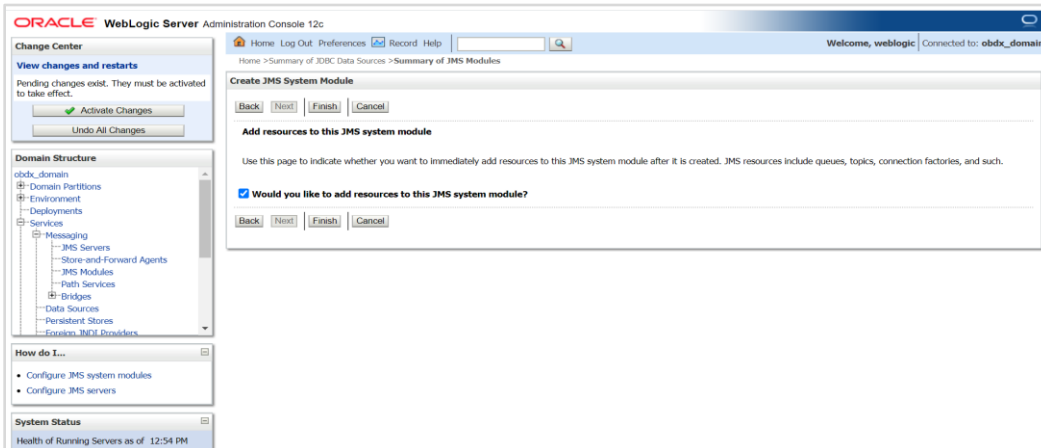
Scope:- Global

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

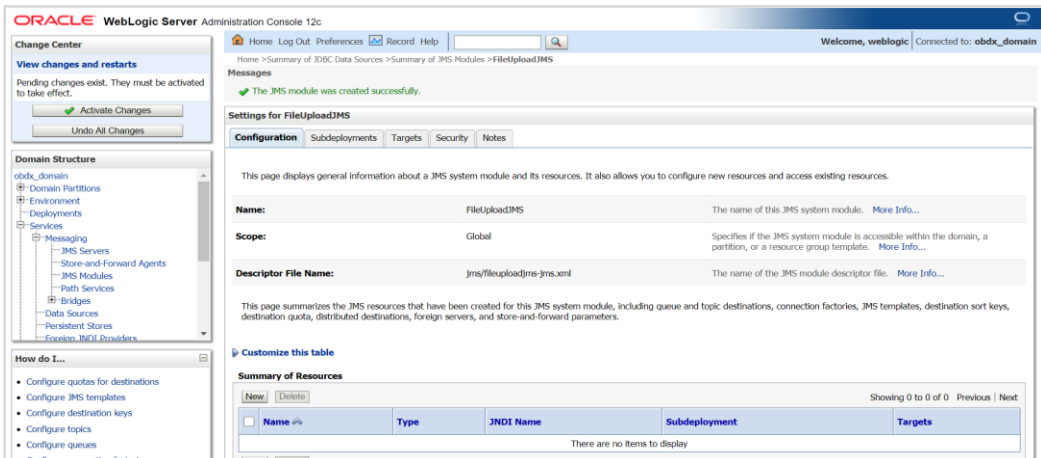
27. Click on Next



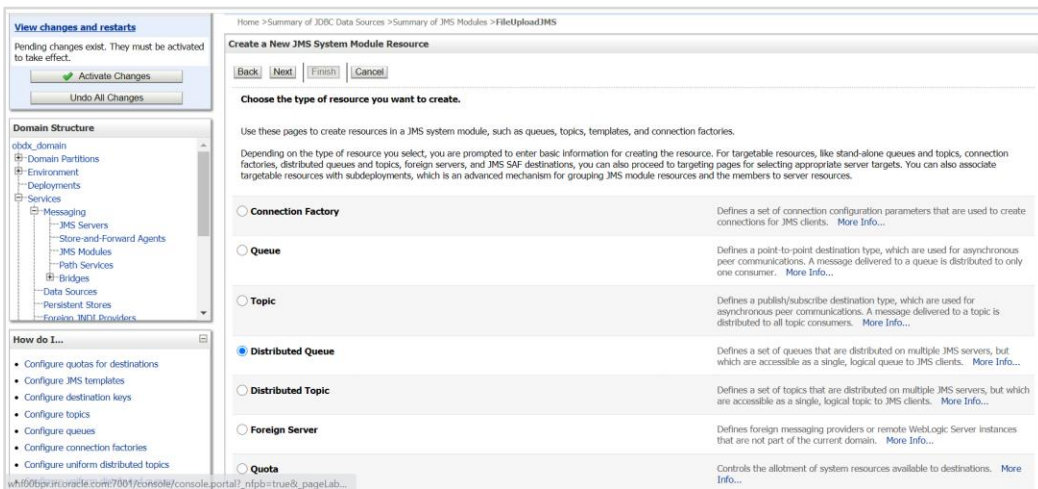
28. Set target as cluster → click on Next



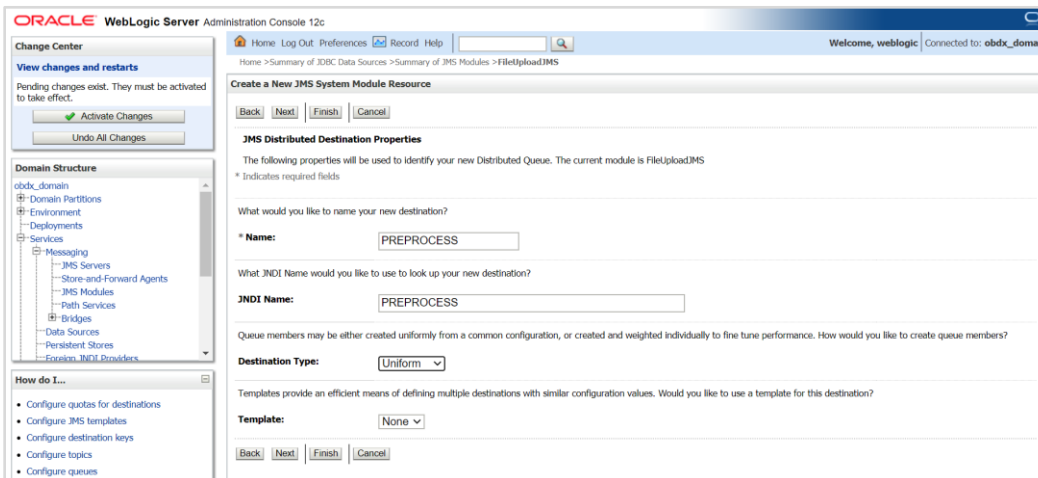
29. Select Would you like to add resources to this JMS system module and click on finish



30. Select new



31. Select Distributed Queue and click next



32. Provide

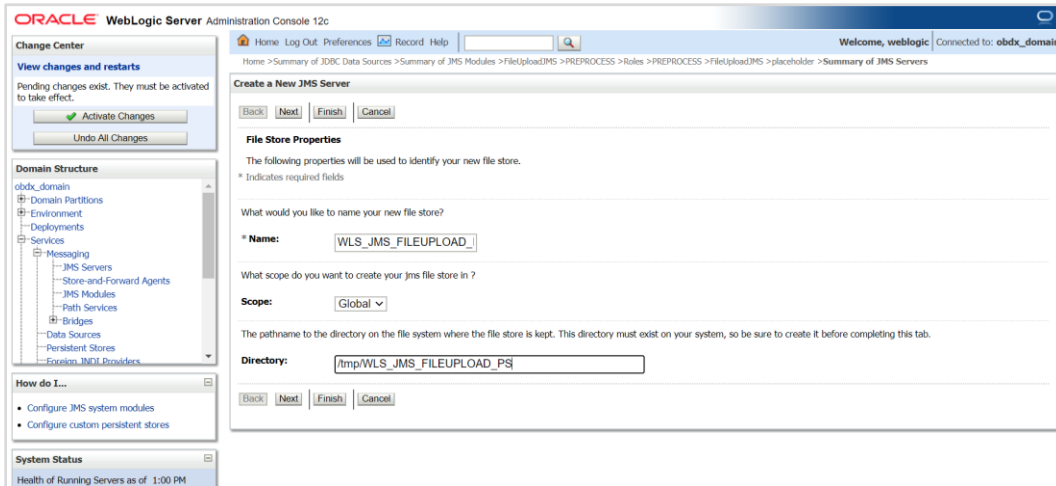
**Name:** - PREPROCESS

**JNDI Name:** - PREPROCESS

**Destination Type:** - Uniform

**Template:** - None

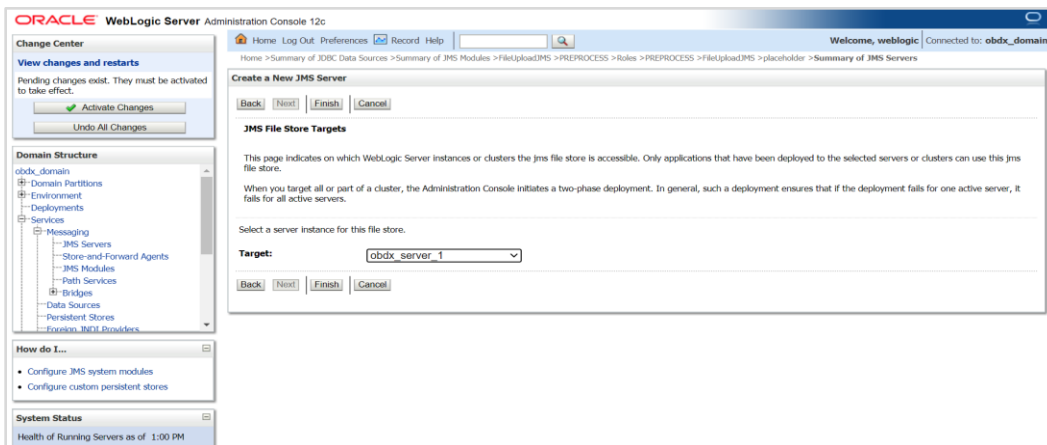




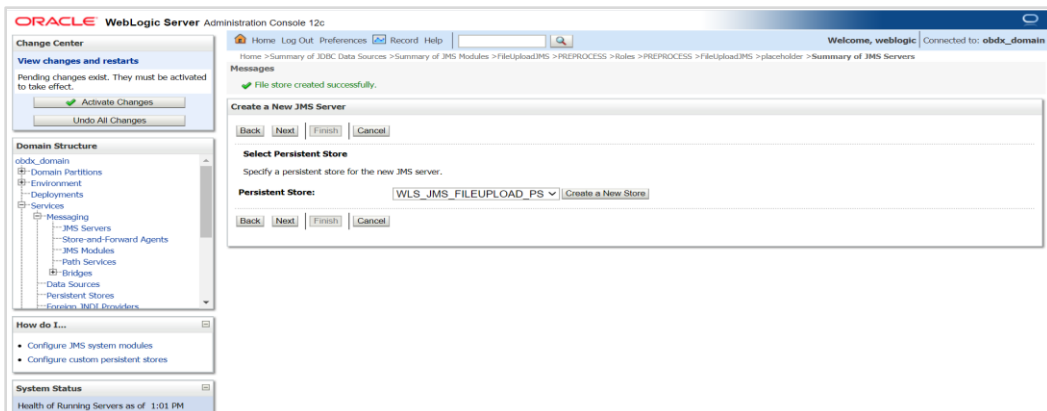
33. Name :- WLS\_JMS\_FILEUPLOAD\_PS

Scope :- Global

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

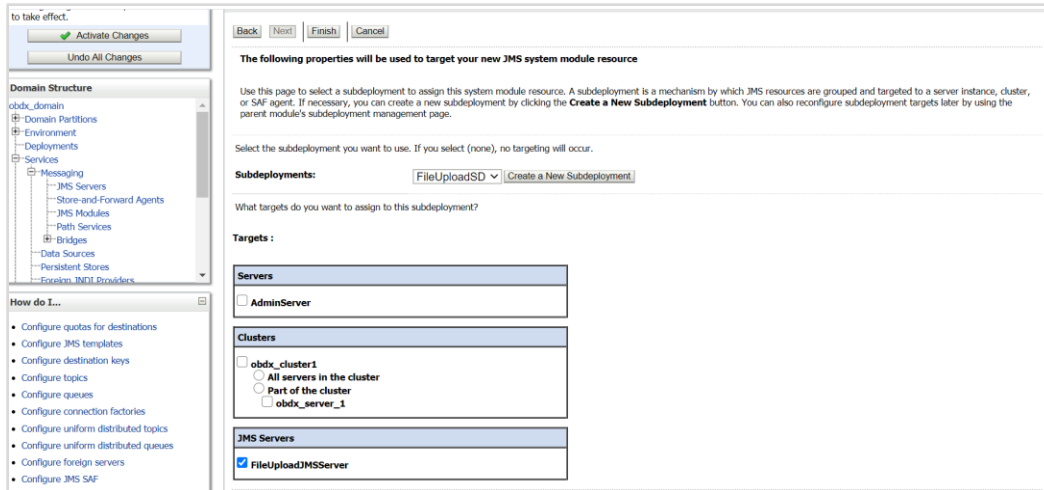


34. Select target as managed server

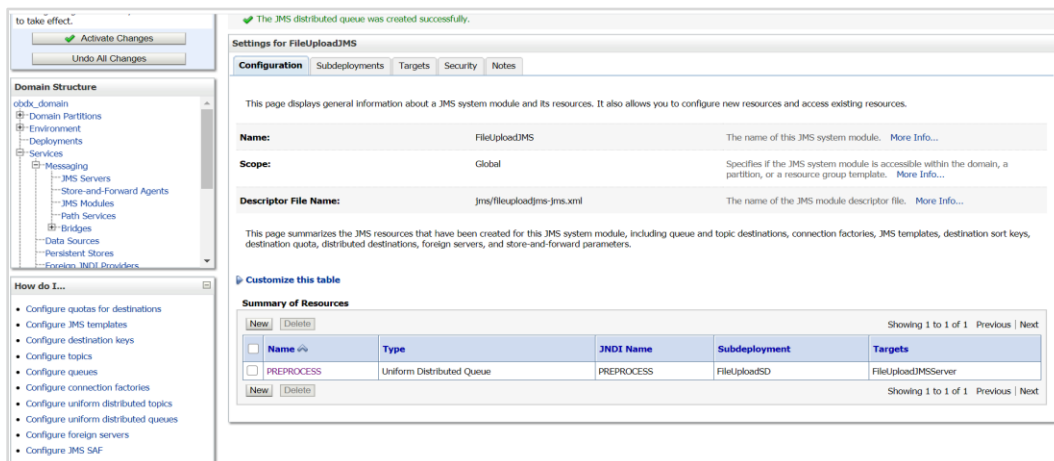


35. Select WLS\_JMS\_FILEUPLOAD\_PS and click on Next

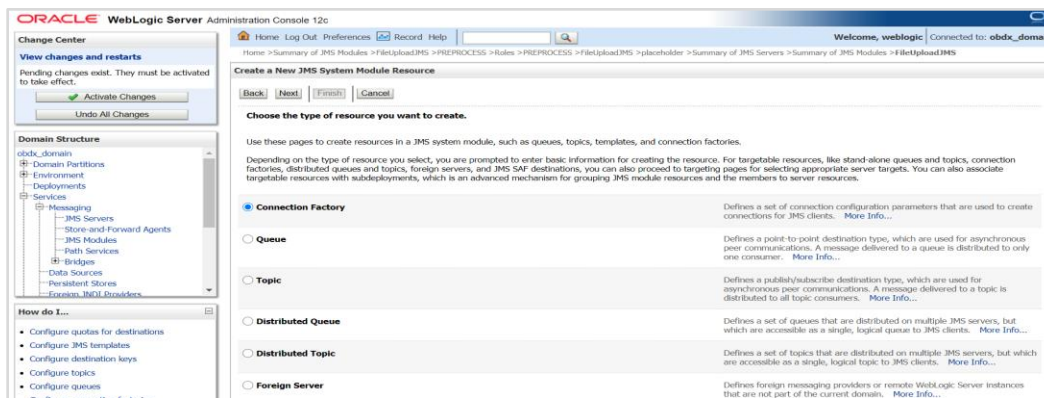
36. Select Create a New Subdeployment and create FileUploadSD



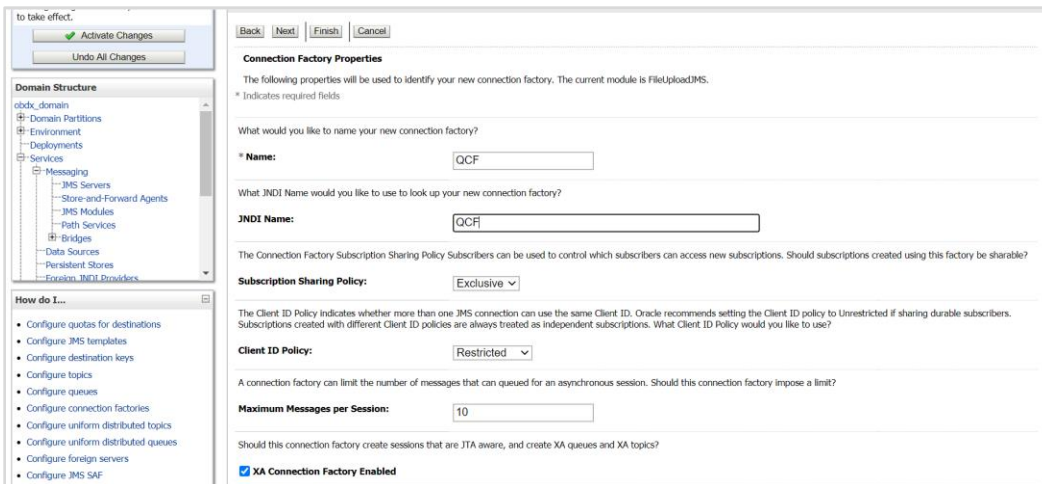
37. Select FileUploadJMSServer and click on Finish



38. Similarly Go into FileuploadJMS module and click on Next



39. Select Connection factory → Click Next



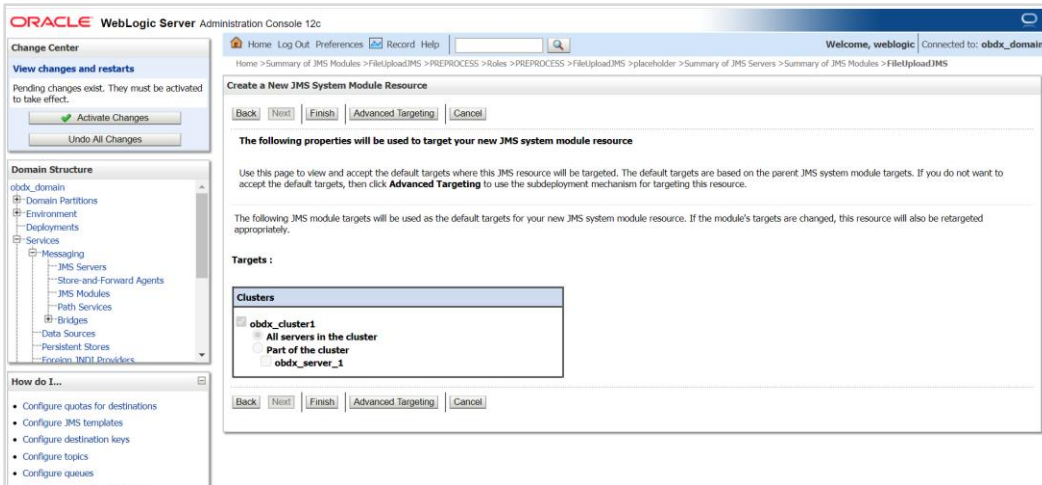
#### 40. Provide

Name :- OCF

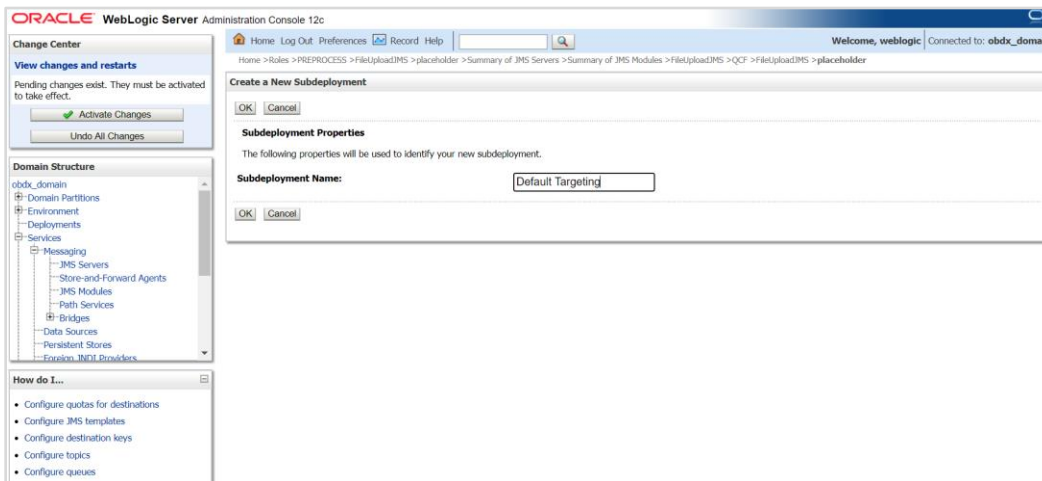
JNDI Name :- OCF

Subscription Sharing Policy :- Exclusive

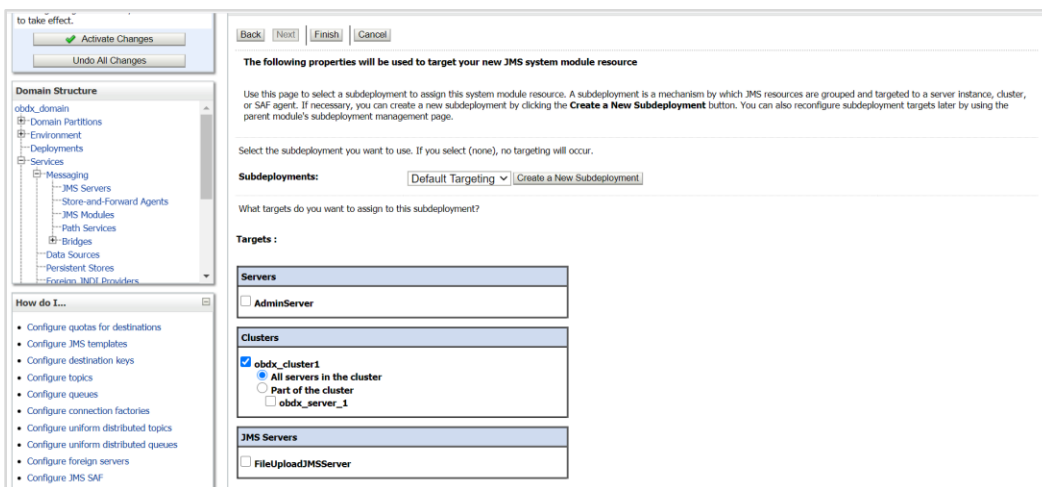
Client ID Policy :- Restricted



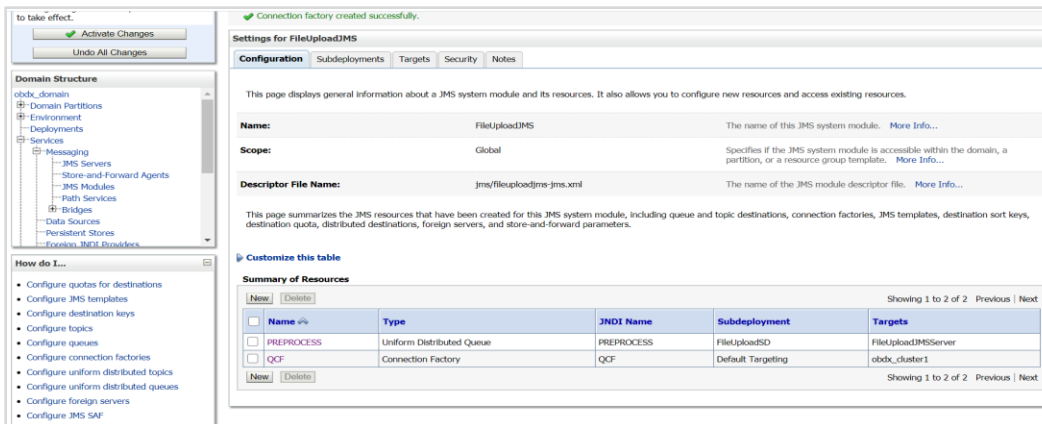
41. Click on Advanced targeting



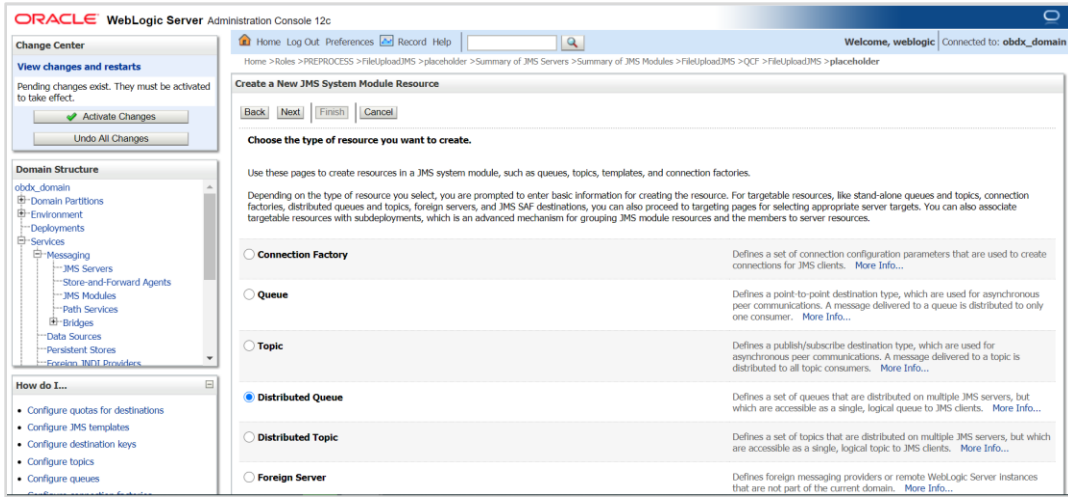
42. Provide Subdeployment Name as Default Targeting



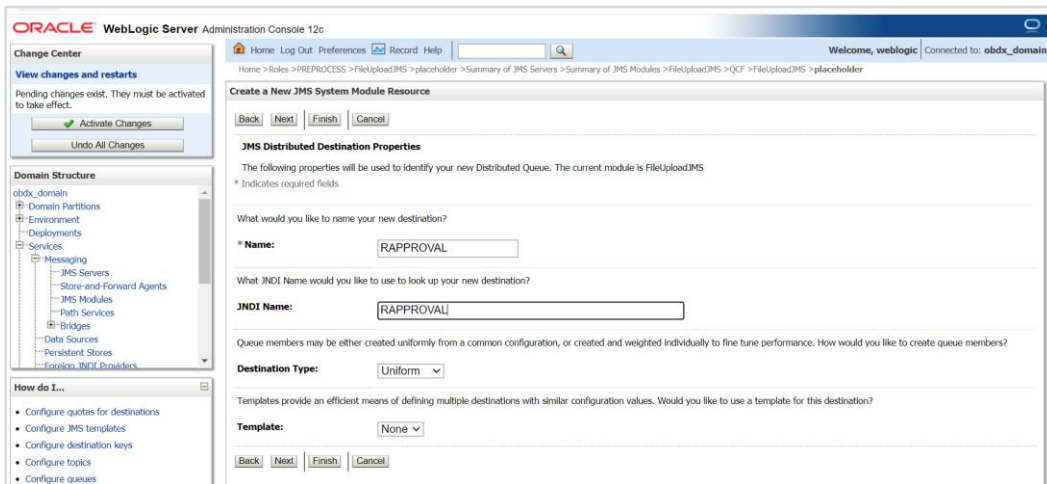
43. Select cluster and click on Finish



44. Go to FileUpload JMS click on New



45. Select Distributed Queue



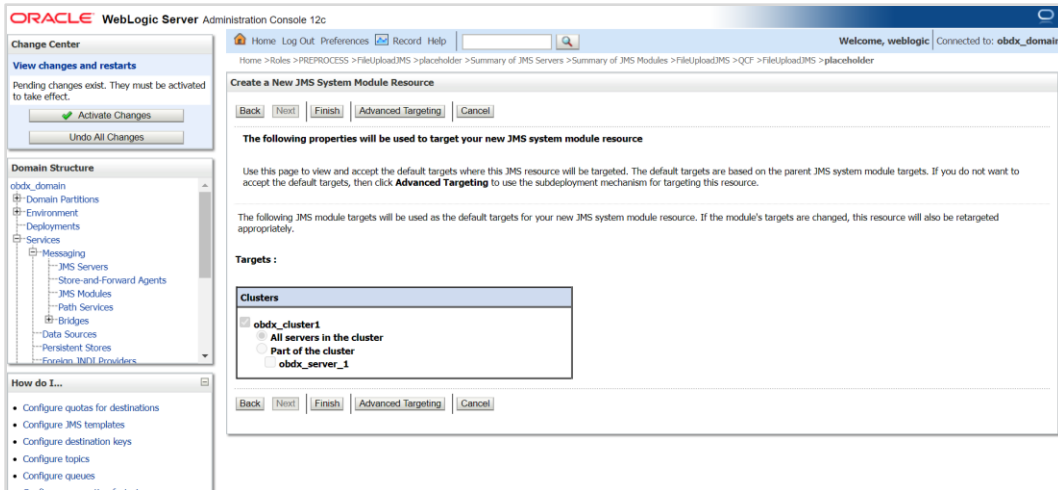
46. Provide

**Name :-** RAPPROVAL

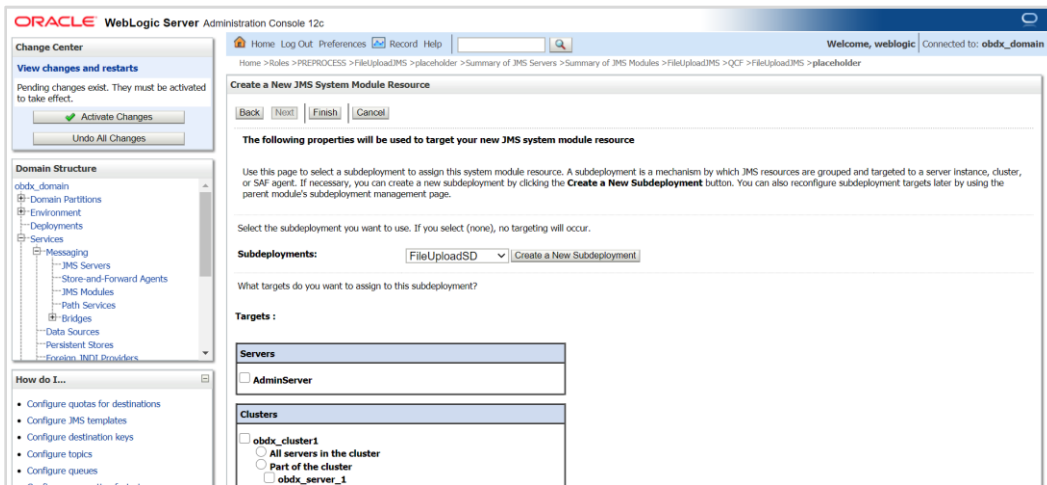
**JNDI Name :-** RAPPROVAL

**Destination Type:-** Uniform

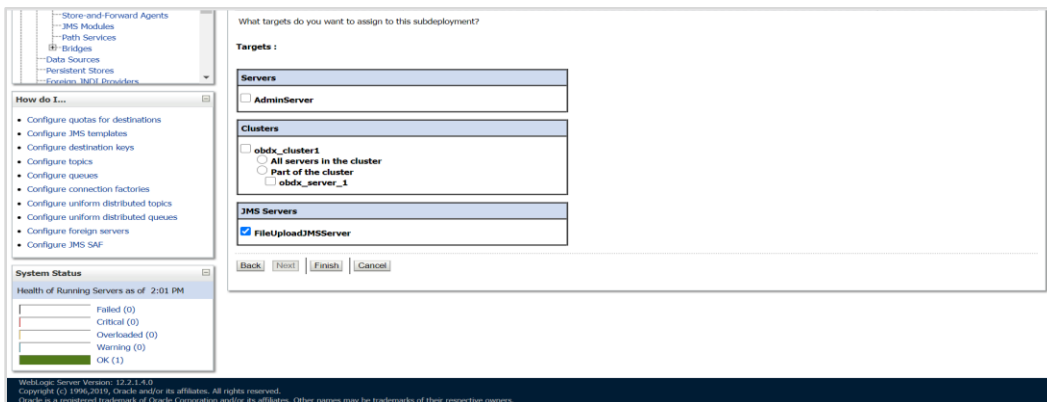
**Template :-** None



47. Select Advance targeting



48. Select Subdeployment :- FileUploadSD



49. Select FileUploadJMSServer and click on Finish

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:** FileUploadJMS 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/fileuploadjms-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**

[New](#) [Delete](#) Showing 1 to 3 of 3 [Previous](#) | [Next](#)

| <input type="checkbox"/> | Name ↕     | Type                      | JNDI Name  | Subdeployment     | Targets             |
|--------------------------|------------|---------------------------|------------|-------------------|---------------------|
| <input type="checkbox"/> | PREPROCESS | Uniform Distributed Queue | PREPROCESS | FileUploadSD      | FileUploadJMSServer |
| <input type="checkbox"/> | QCF        | Connection Factory        | QCF        | Default Targeting | obdc_cluster1       |
| <input type="checkbox"/> | RAPPROVAL  | Uniform Distributed Queue | RAPPROVAL  | FileUploadSD      | FileUploadJMSServer |

[New](#) [Delete](#) Showing 1 to 3 of 3 [Previous](#) | [Next](#)

**System Status**

Health of Running Servers as of 2:01 PM

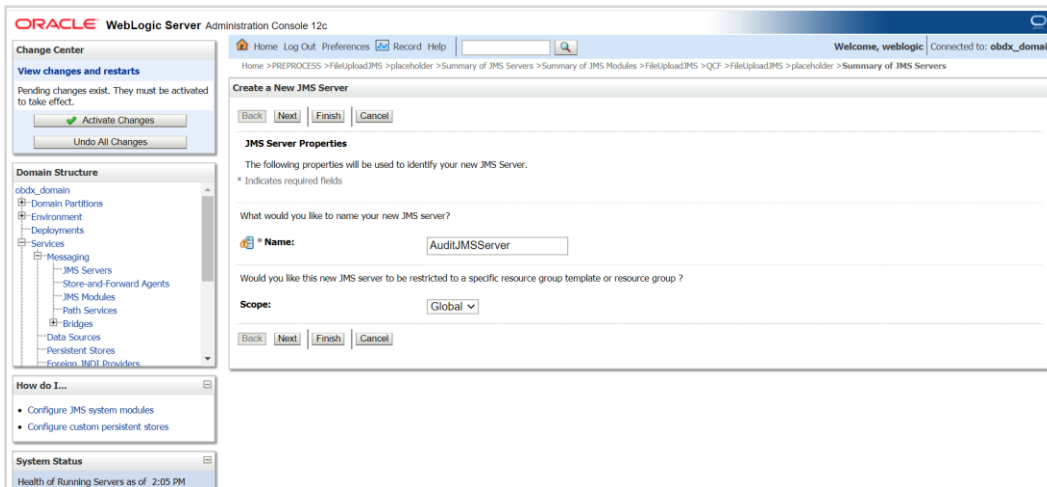
Failed (0)

Critical (0)

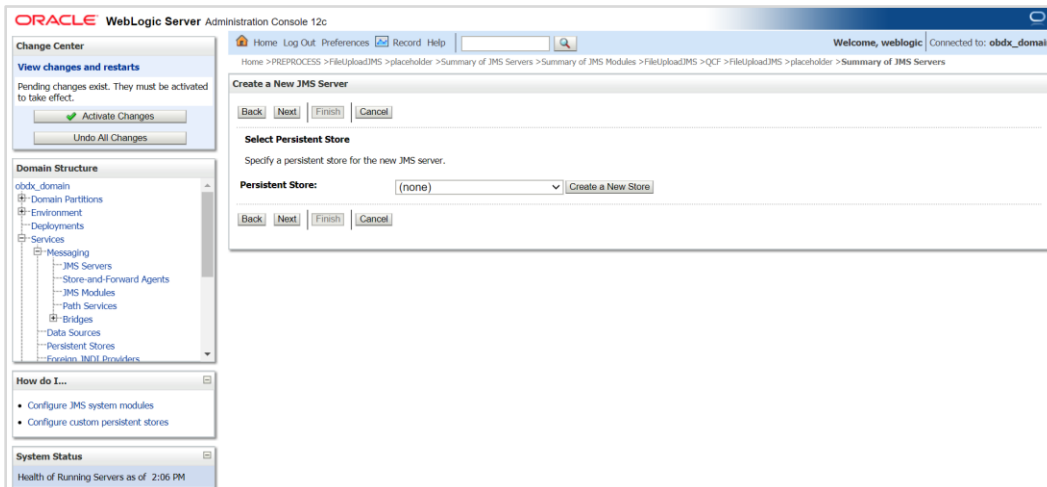
### 3.8 Creating WLS JMS AUDIT PS FileStore

### 3.9 Creating AuditJMSServer JMS Server

### 3.10 Creating WLS JMS REPORT PS FileStore

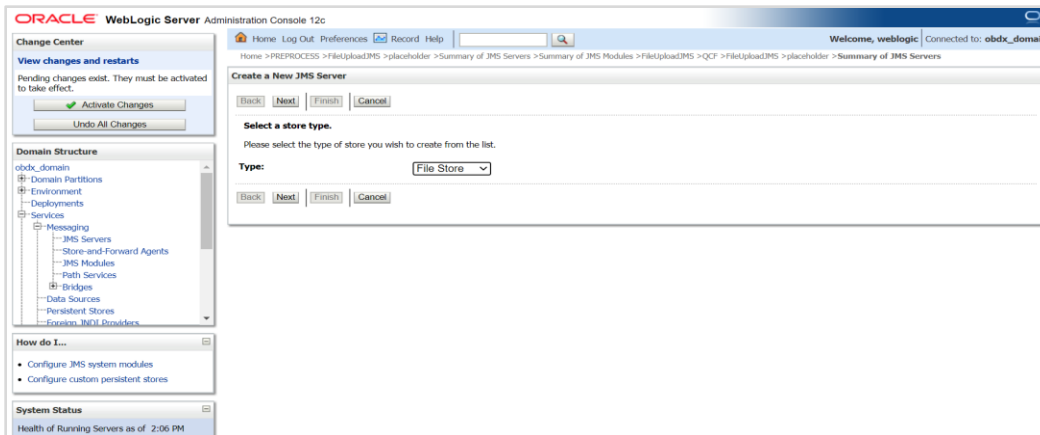


1. Click on JMS server and click on New
2. Provide Name as AuditJMSServer , Scope as Global

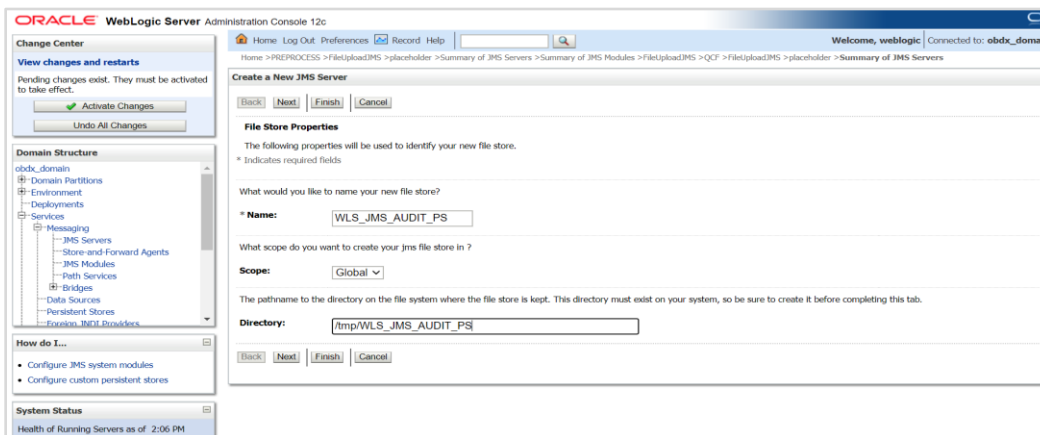




3. Click on Create a New Store



4. Select File Store

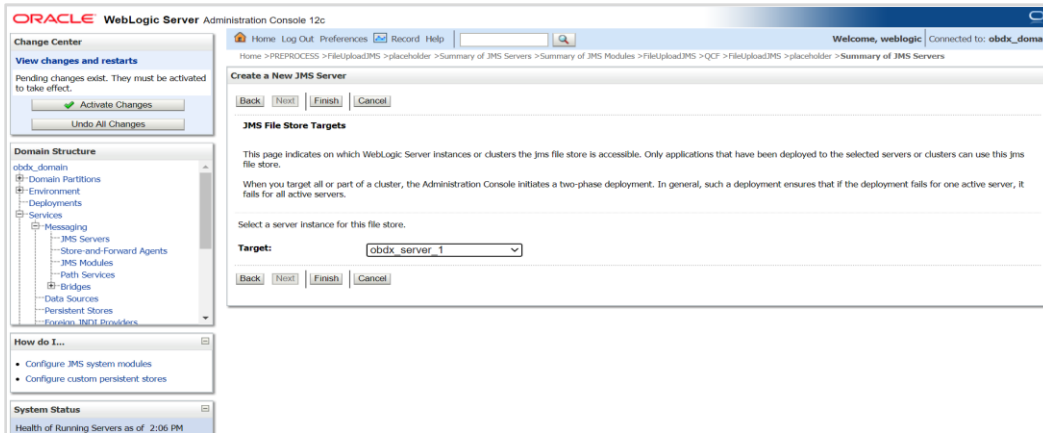


5. Provide

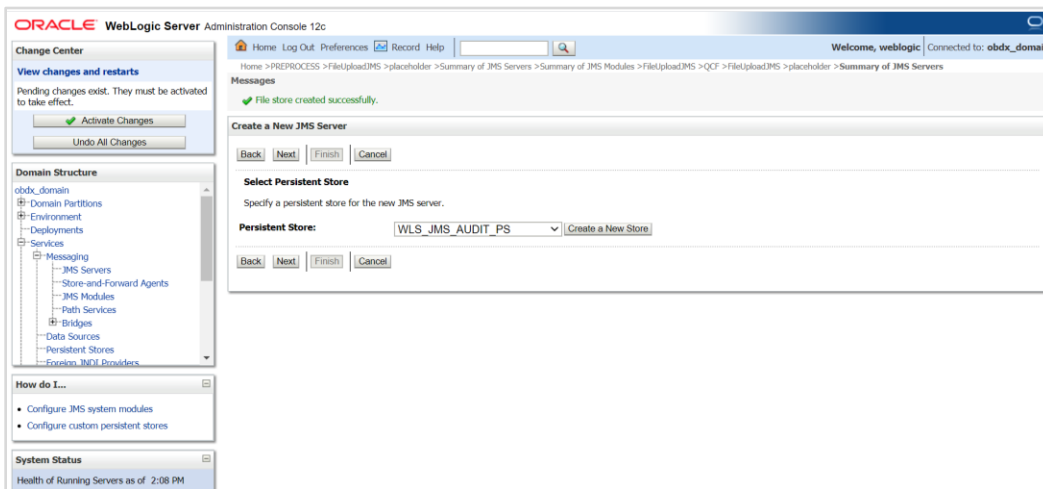
**Name :-** WLS\_JMS\_AUDIT\_PS

**Scope :-** Global

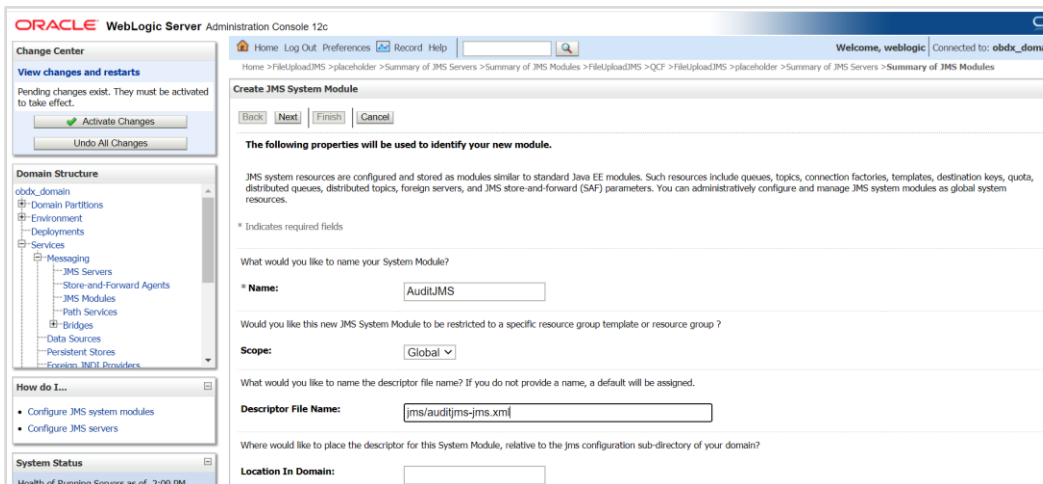
**Directory :-** /tmp/WLS\_JMS\_AUDIT\_PS



6. Select Target as managed server and click on Finish



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

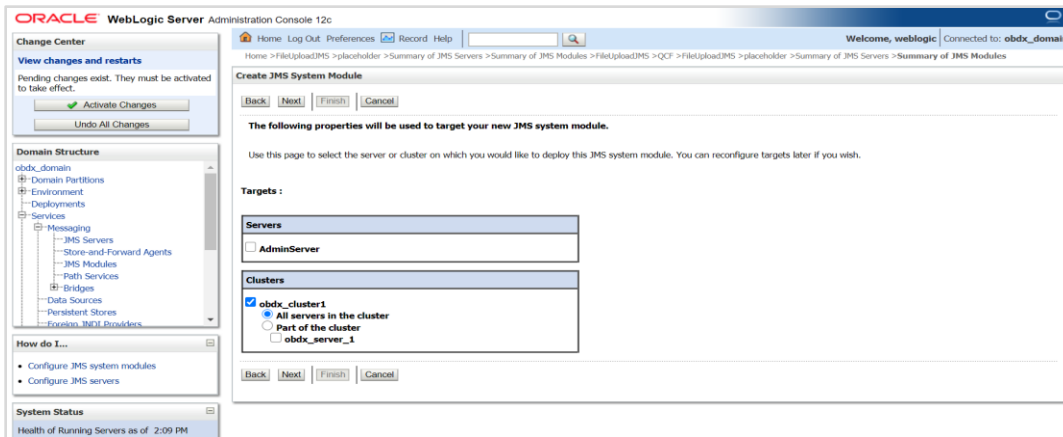


8. Provide

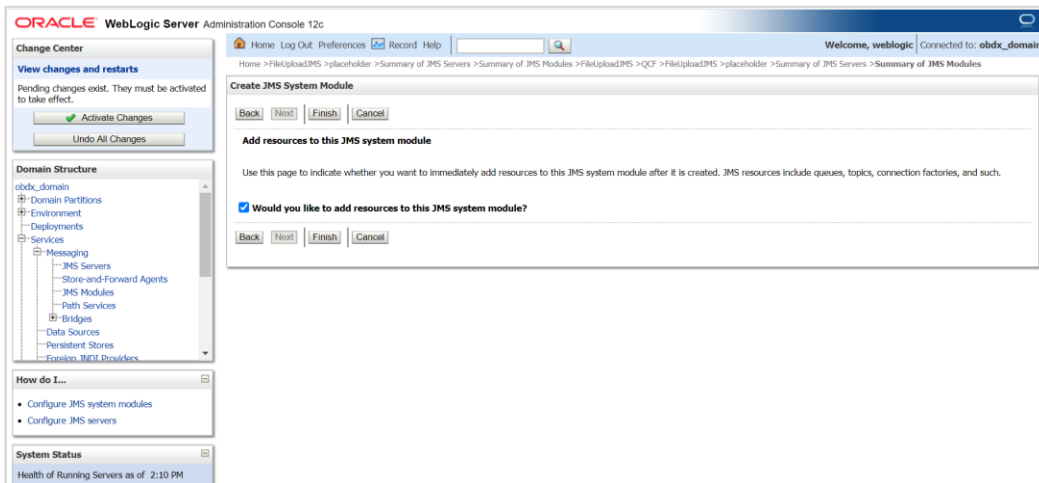
Name :- AuditJMS

Scope :- Global

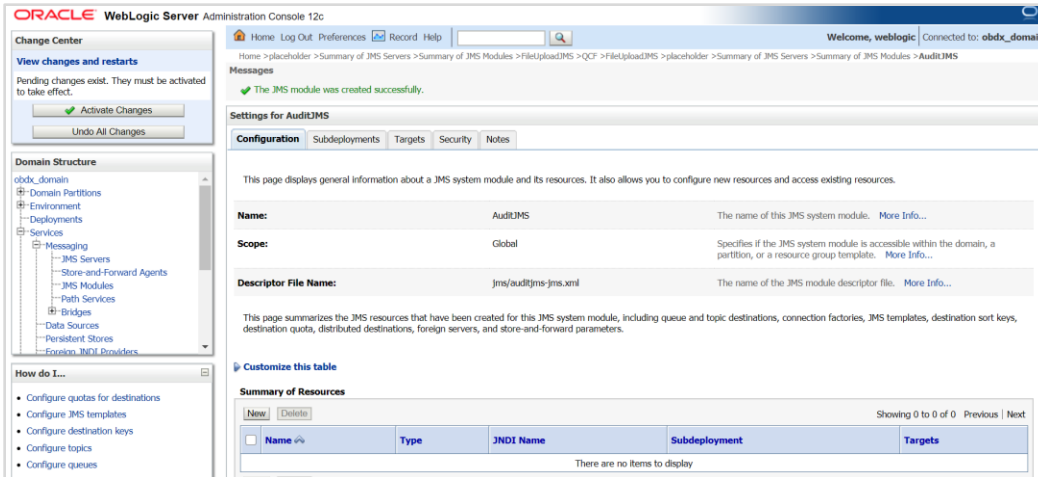
Descriptor File Name:- jms/auditjms-jms.xml



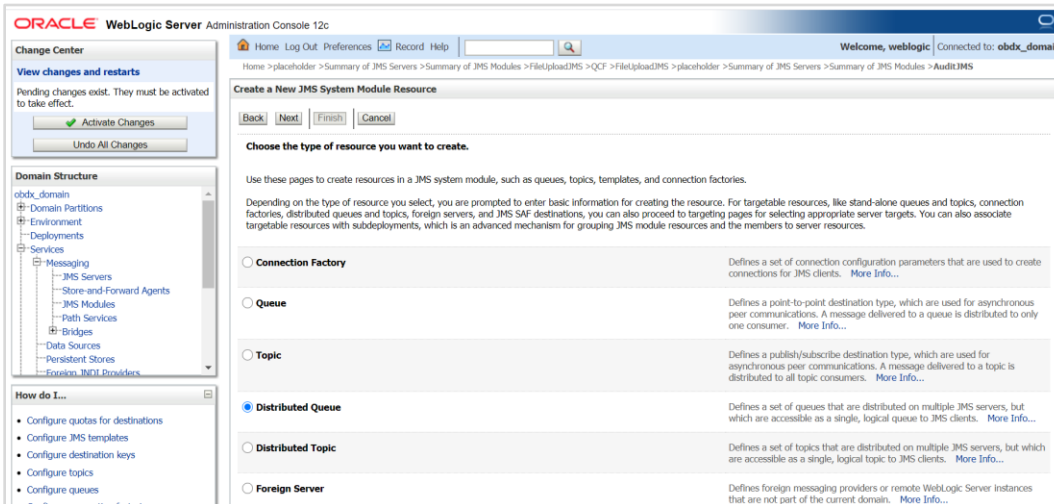
9. Select Cluster as a target



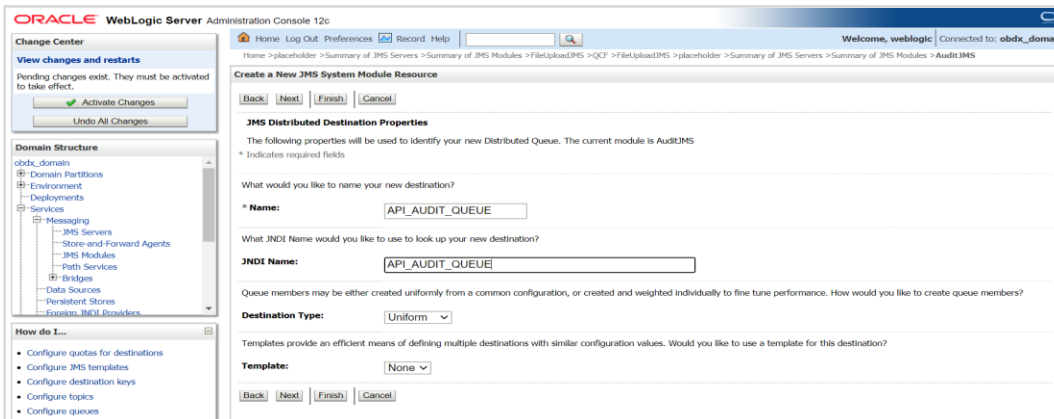
10. Select would you like to add resource to this JMS system module?



11. Click on new



12. Select Distributed Queue

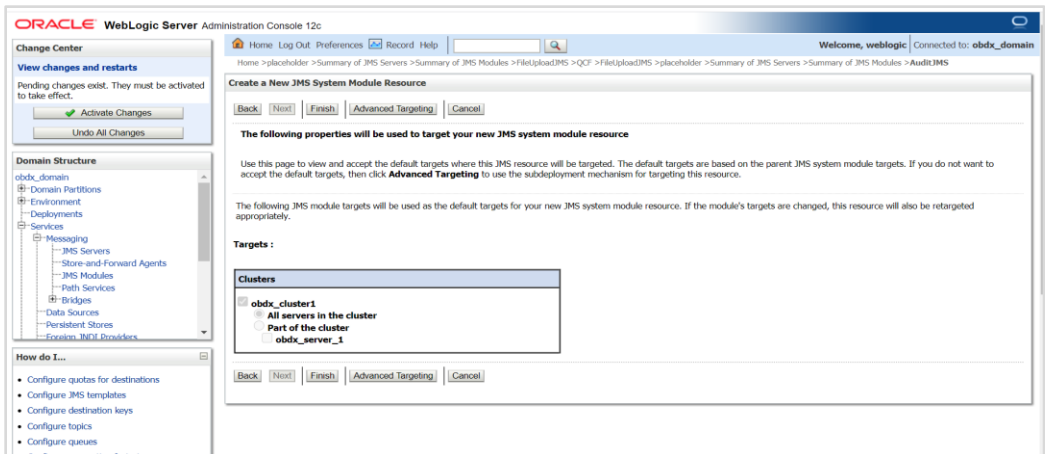


13. Name:- API\_AUDIT\_QUEUE

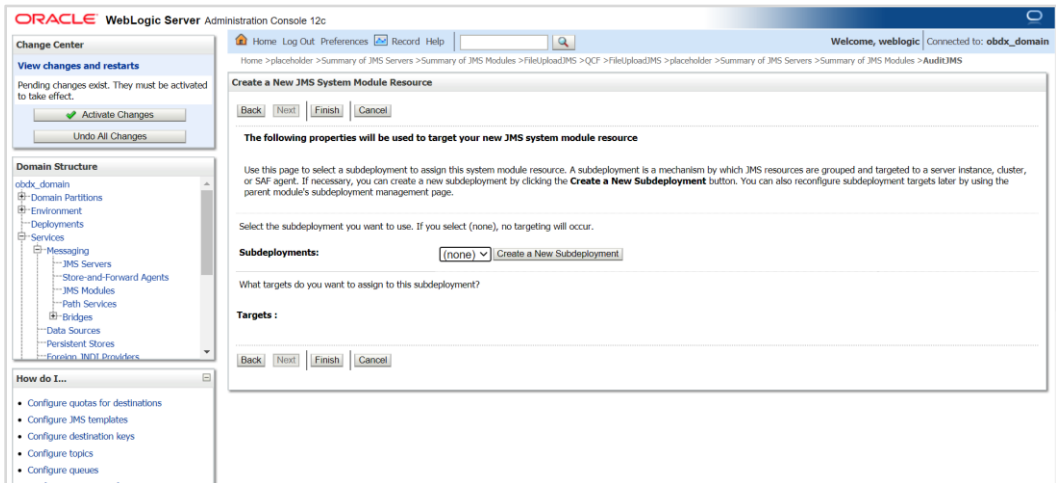
JNDI Name:- API\_AUDIT\_QUEUE

Destination Type :- Uniform

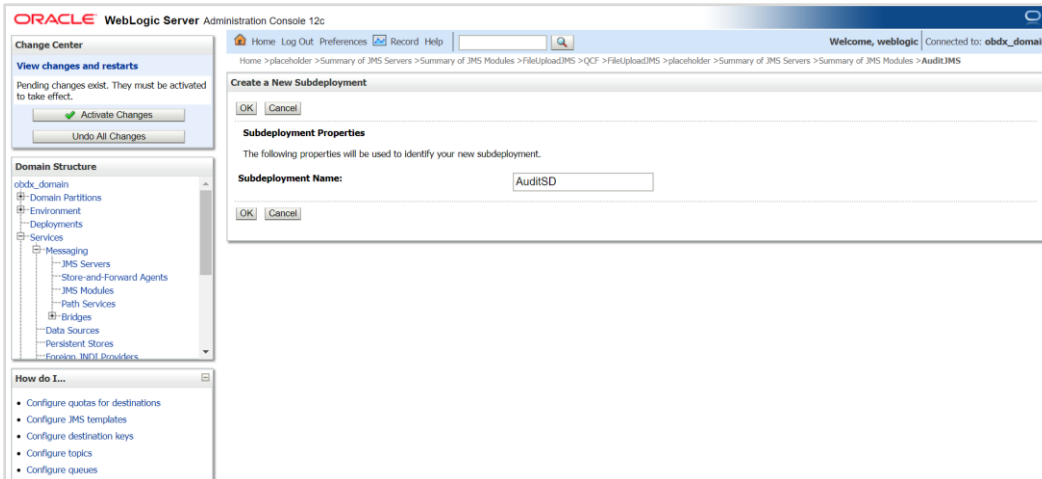
Template:- None



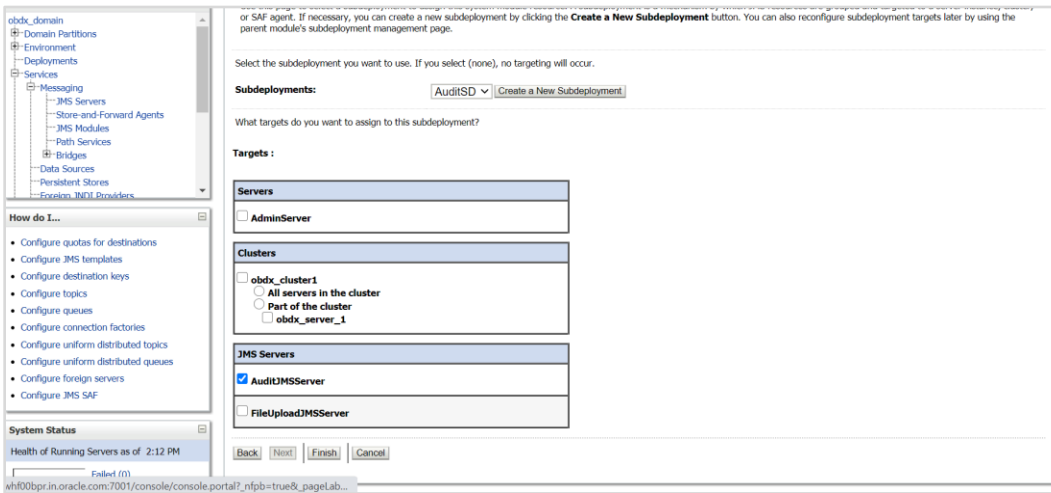
14. Select Advance targeting



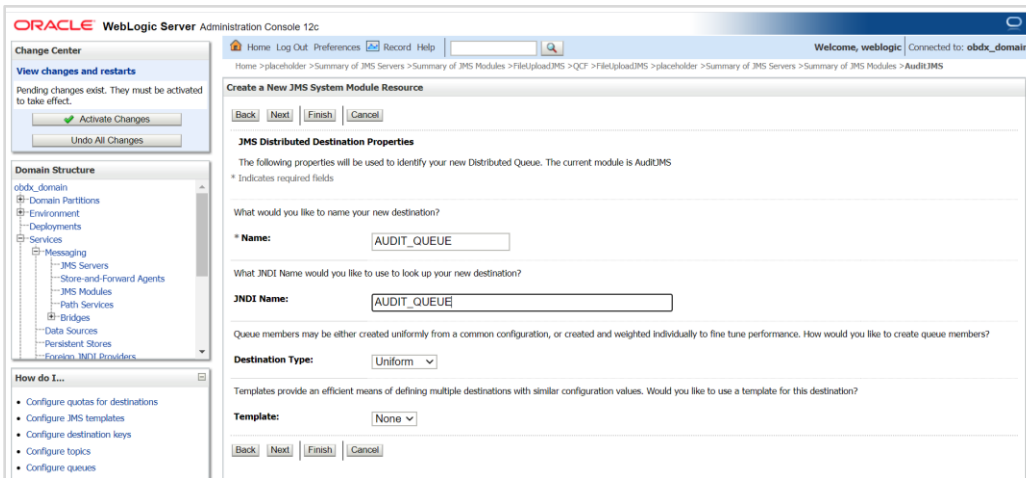
15. Click on Create a New Subdeployment

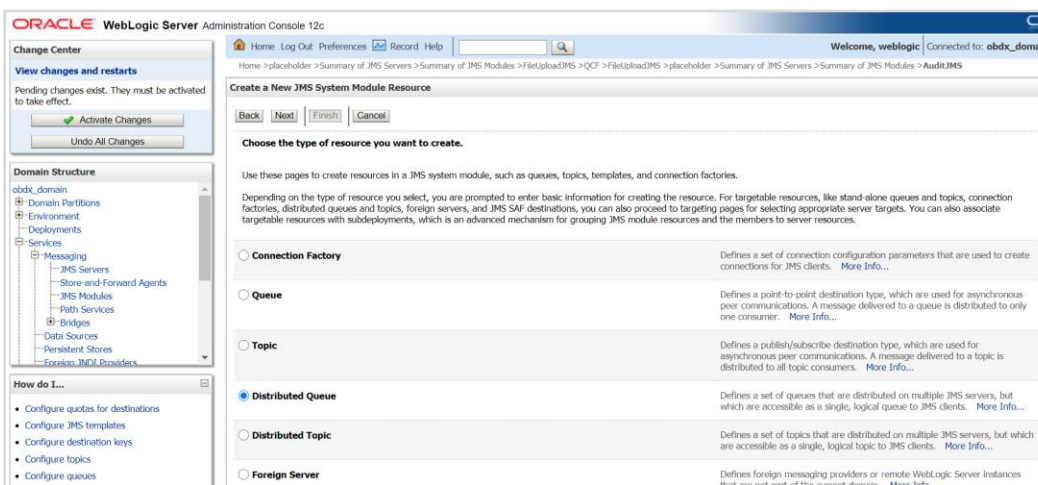
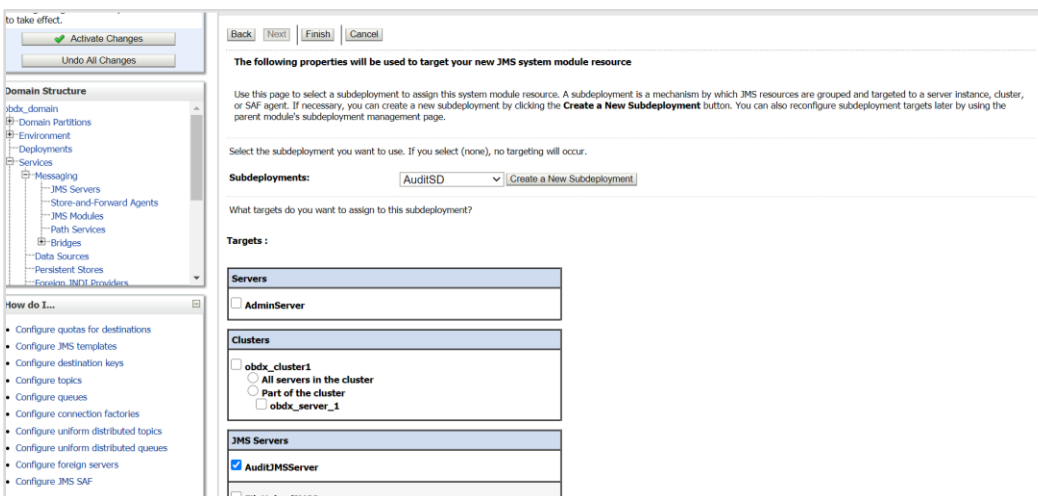
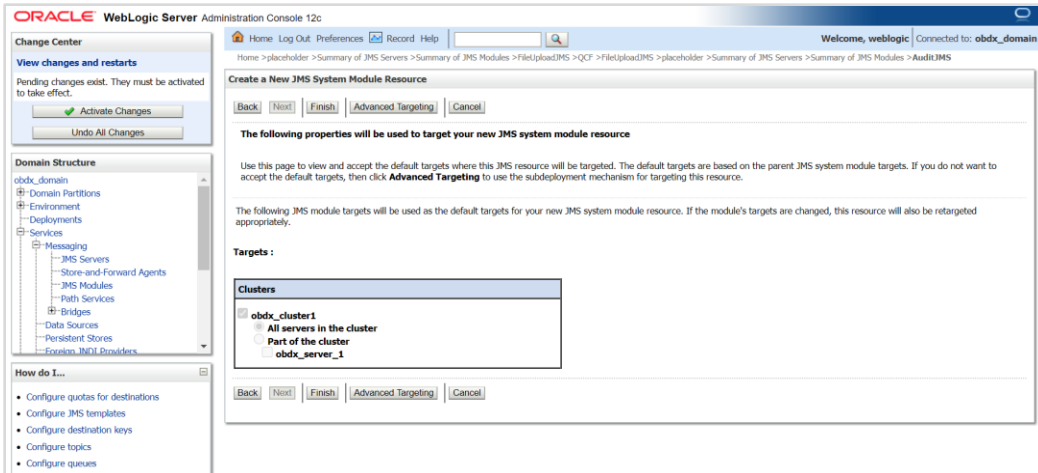


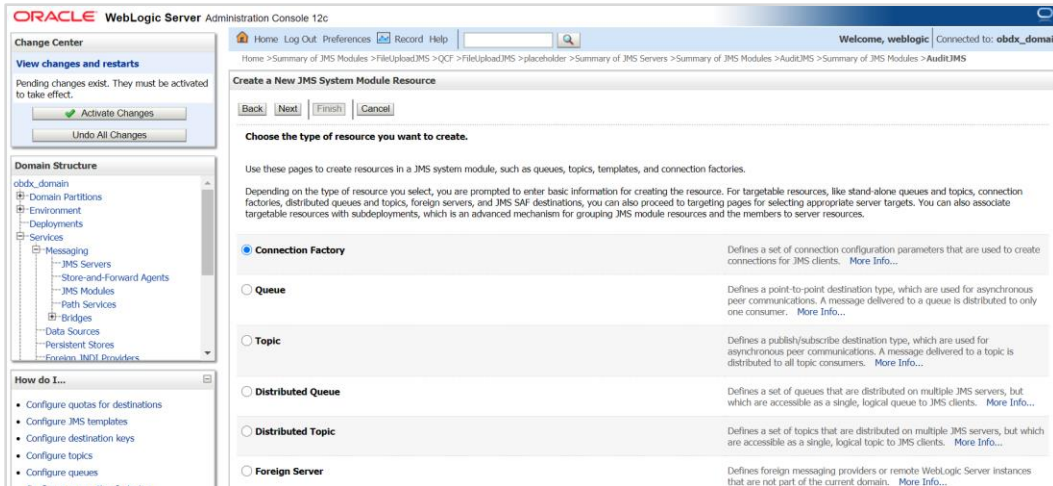
16. Provide Subdeployment Name as AuditSD



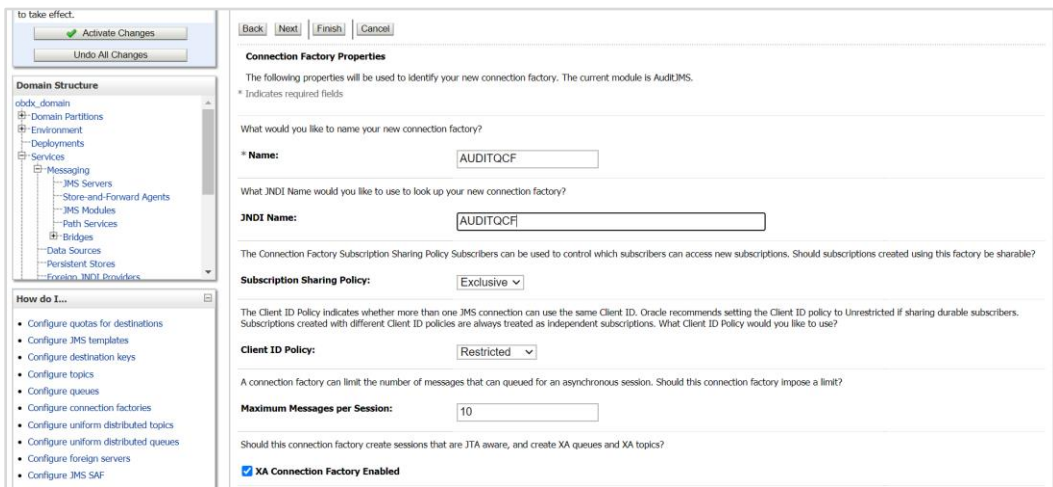
17. Select Target as AuditJMServer







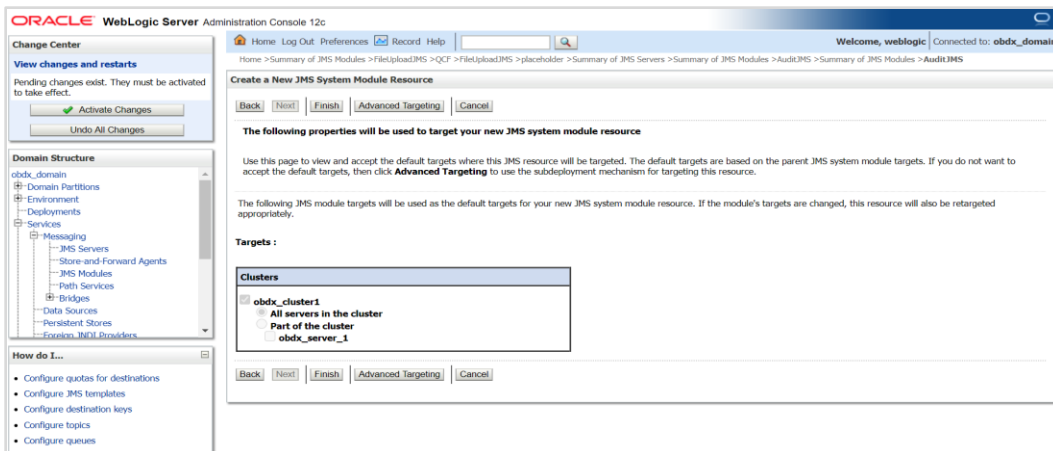
18. Click on connection Factory



19. Provide

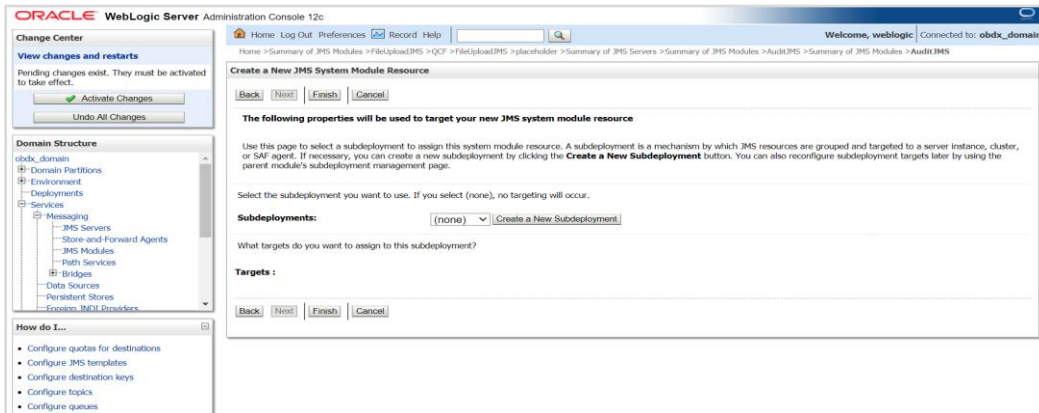
Name :- AUDITQCF

JNDI Name :- AUDITQCF

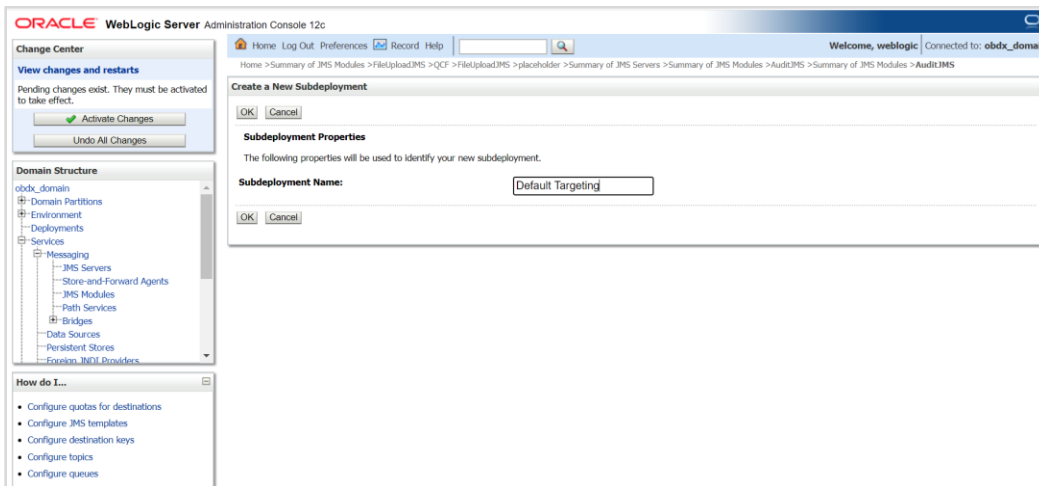




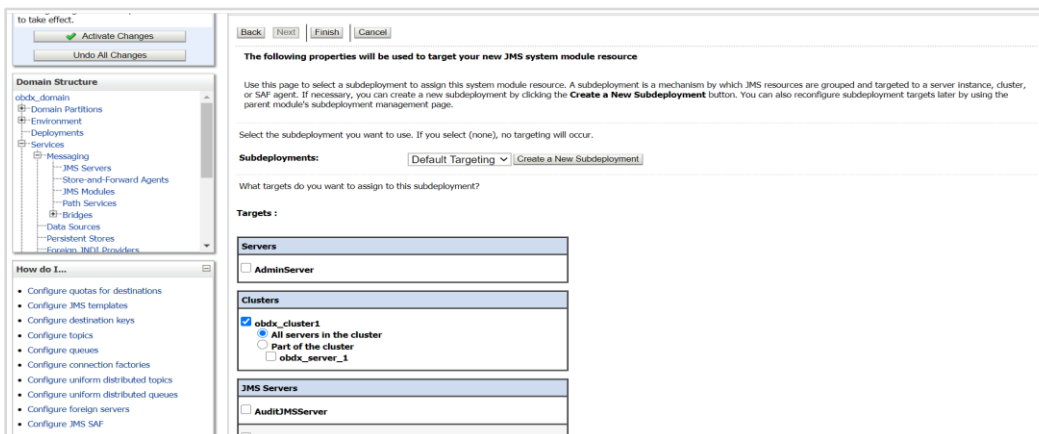
## 20. Click on Advanced Targeting



## 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 SAs

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 quotas, distributed destinations, foreign servers, and store-and-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

**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.

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

| Name                | Persistent Store      | Target        | Current Target | Health |
|---------------------|-----------------------|---------------|----------------|--------|
| AuditJMSServer      | WLS_JMS_AUDIT_PS      | obdx_server_1 | obdx_server_1  |        |
| FileUploadJMSServer | WLS_JMS_FILEUPLOAD_PS | obdx_server_1 | obdx_server_1  |        |
| ReportsJMSServer    | WLS_JMS_REPORT_PS     | obdx_server_1 | obdx_server_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.

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

| Name             | Type              |
|------------------|-------------------|
| AuditJMS         | JMSSystemResource |
| FileUploadJMS    | JMSSystemResource |
| ReportsJMSModule | JMSSystemResource |

**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/reportsjmsmodule-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.

**Summary of Resources**

| Name            | Type                      | JNDI Name       | Subdeployment        | Targets          |
|-----------------|---------------------------|-----------------|----------------------|------------------|
| REPORTADHOC     | Uniform Distributed Queue | REPORTADHOC     | ReportsSubdeployment | ReportsJMSServer |
| REPORTSCHEDULED | Uniform Distributed Queue | REPORTSCHEDULED | ReportsSubdeployment | ReportsJMSServer |
| ReportsQCF      | Connection Factory        | ReportsQCF      | Default Targeting    | obdx_cluster1    |

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

REPORTADHOC – Uniform Distributed Queue

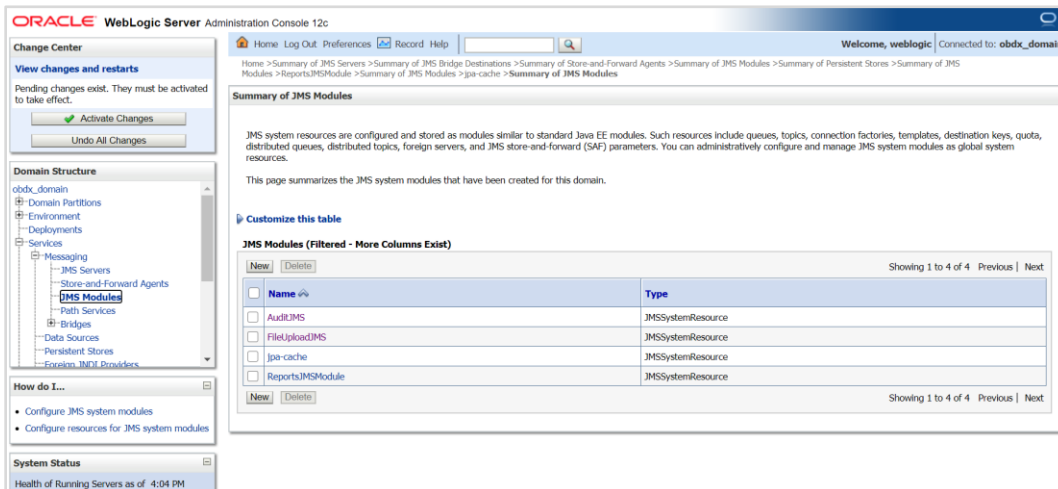
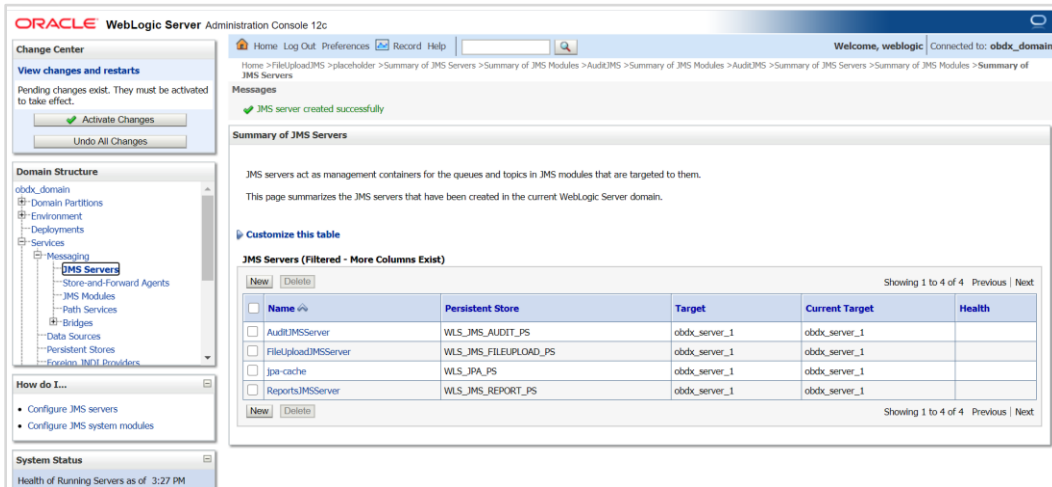
REPORTSCHEDULED -- Uniform Distributed Queue

ReportsQCF – Connection Factory

### 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



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

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

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

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

✔ The JMS distributed topic was created successfully.

**Settings for jpa-cache**

**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:** jpa-cache      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/jpa-cache-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**

[New](#) | [Delete](#)      Showing 1 to 2 of 2    [Previous](#) | [Next](#)

| <input type="checkbox"/> | Name ↕                | Type                      | JNDI Name             | Subdeployment     | Targets       |
|--------------------------|-----------------------|---------------------------|-----------------------|-------------------|---------------|
| <input type="checkbox"/> | .jms/jpa-cache-cf     | Connection Factory        | ./jms/jpa-cache-cf    | Default Targeting | obdx_cluster1 |
| <input type="checkbox"/> | ./jms/jpa-cache-topic | Uniform Distributed Topic | ./jms/jpa-cache-topic | jpa-cache-sd      | jpa-cache     |

[New](#) | [Delete](#)      Showing 1 to 2 of 2    [Previous](#) | [Next](#)

### 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.

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 servers
- Configure JMS system modules

**System Status**

Health of Running Servers as of 3:32 PM

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

**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 |
|-------------------|---------------------------|---------------|----------------|--------|
| AuditJMS          | WLS_JMS_AUDIT_PS          | obdx_server_1 | obdx_server_1  |        |
| ExtSystemReceiver | WLS_JMS_EXTSYSRECEIVER_PS | obdx_server_1 | obdx_server_1  |        |
| FileUploadJMS     | WLS_JMS_FILEUPLOAD_PS     | obdx_server_1 | obdx_server_1  |        |
| jpa-cache         | WLS_JPA_PS                | obdx_server_1 | obdx_server_1  |        |
| ReportsJMS        | WLS_JMS_REPORT_PS         | obdx_server_1 | obdx_server_1  |        |

Showing 1 to 5 of 5 Previous Next

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)

| Name              | Type              |
|-------------------|-------------------|
| AuditJMS          | JMSSystemResource |
| ExtSystemReceiver | JMSSystemResource |
| FileUploadJMS     | JMSSystemResource |
| jpa-cache         | JMSSystemResource |
| 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.

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
    - External 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

✓ The JMS distributed queue was created successfully.

Settings for ExtSystemReceiver

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

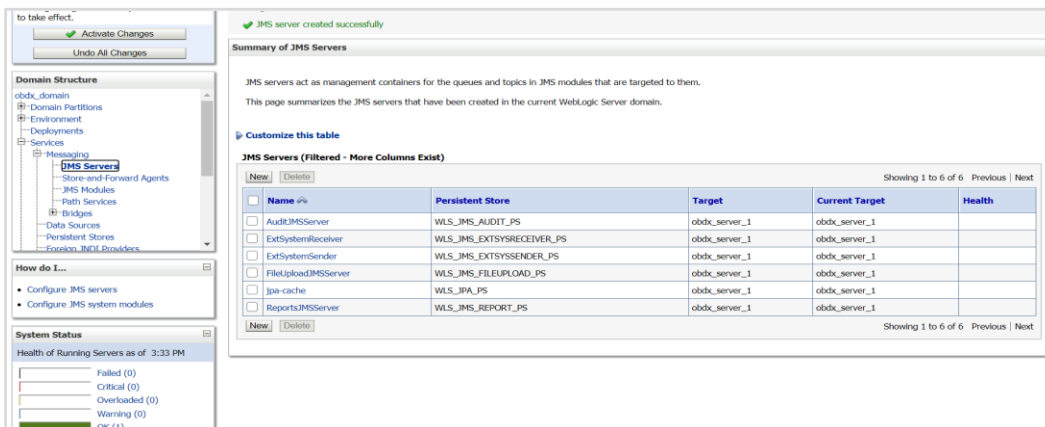
Summary of Resources

| Name                   | Type                      | JNDI Name              | Subdeployment        | Targets           |
|------------------------|---------------------------|------------------------|----------------------|-------------------|
| ExtSystemReceiverQCF   | Connection Factory        | ExtSystemReceiverQCF   | Default Targeting    | obdx_cluster1     |
| ExtSystemReceiverQueue | Uniform Distributed Queue | ExtSystemReceiverQueue | ExtSystemReceiverSub | ExtSystemReceiver |

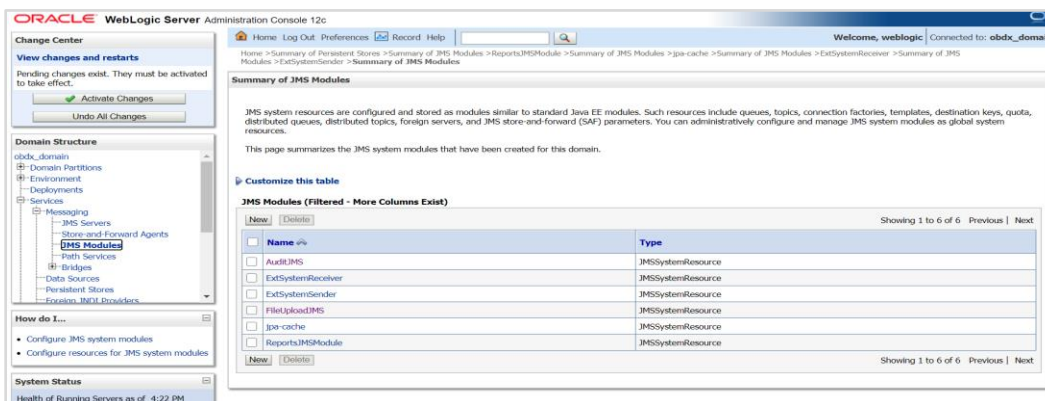


### 3.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS JMS EXTSYSSENDER PS

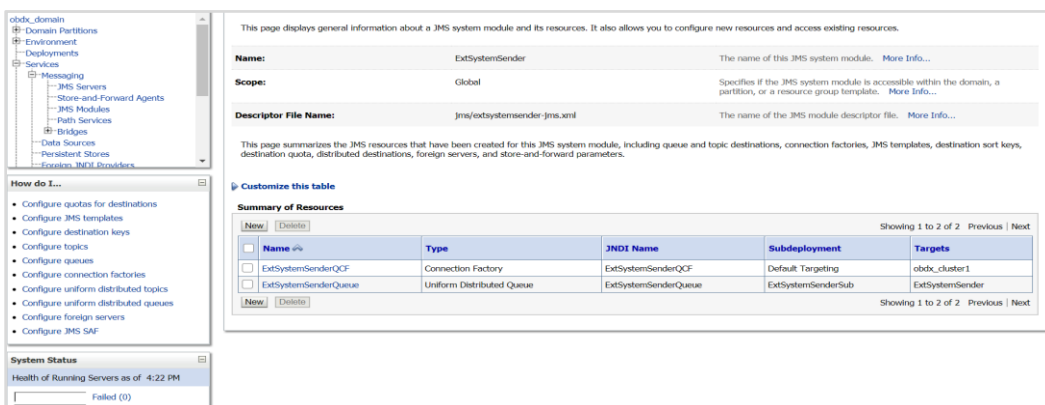
As show below create JMS Server ExtSystemSender



#### 1. Create ExtSystemSender JMS Module

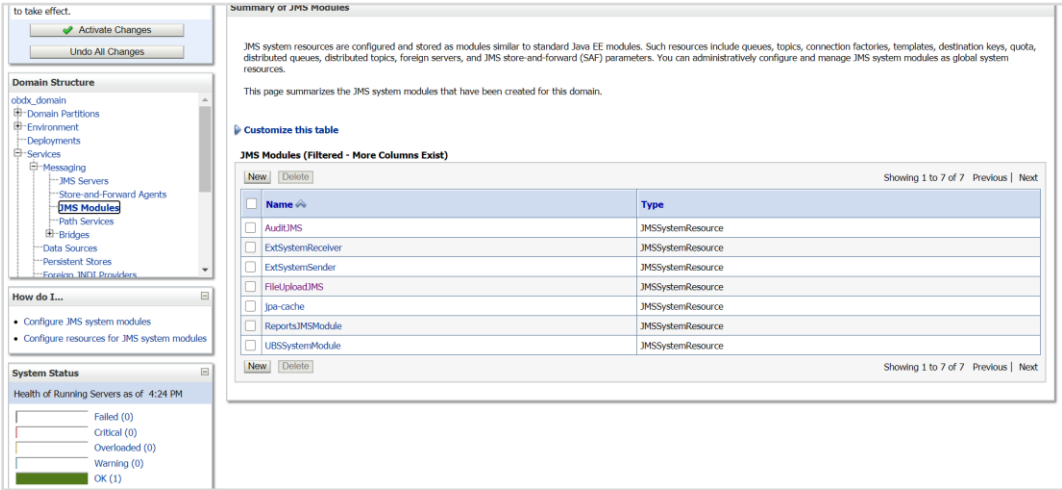


#### 2. Under ExtSystemSender JMS Module create ExtSystemSenderQCF – connection Factory and ExtSystemSenderQueue – Uniform Distributed Queue as show below

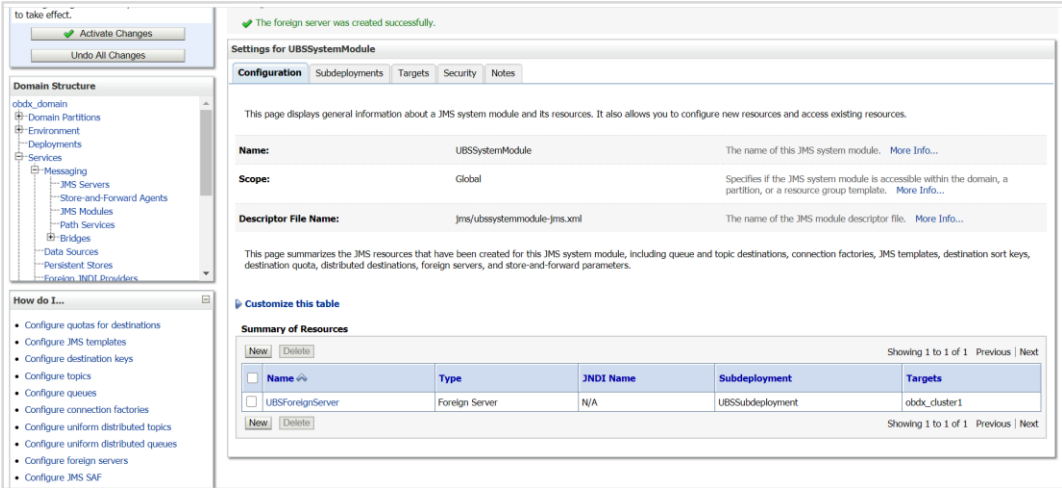


### 3.16 Creating UBSSystemModule JMS Server

1. In JMSModule create UBSSystemModule

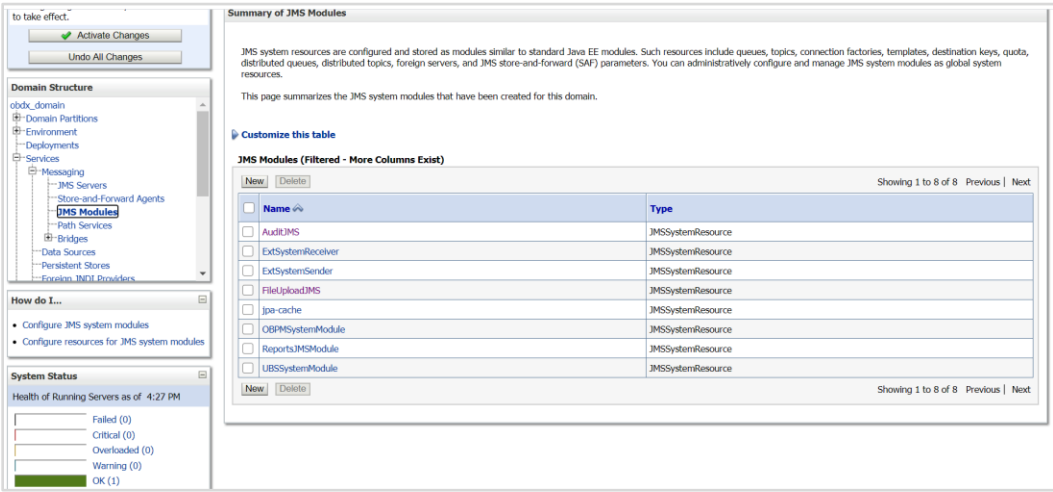


2. Under UBSSystemModule create UBSSystemForeignServer – Foreign Server as shown below

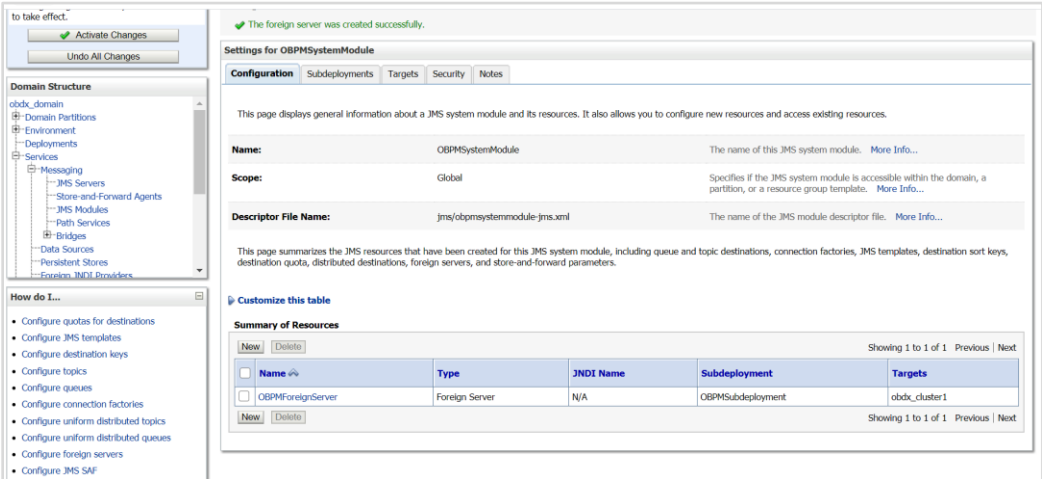


### 3.17 Creating OBPMForeignServer JMS Server

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/Extxfac SimulatorMDB.ear`  
(Target - obapi\_cluster)

[Home](#)

---

## 5. 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.GenericIdentityStoreProvider">

<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"/>
```

[Home](#)