Installation Guide- Non-Linux Platforms
Oracle Banking Digital Experience
Release 22.1.0.0.0

Part No. F56934-01

May 2022

ORACLE

Installation Guide- Non-Linux Platforms

May 2022

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/

Table of Contents

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 Digital Experience Patchset Release 21.1.5.0.0, refer to the following documents:

- Oracle Banking Digital Experience Installation Manuals

2. Manual OBDX installation

OBDX Database Installation with OBPM FLAVOR

Create required OBDX tablespace and user in below sequence.

2.1 Create OBDX Tablespace (file obdx_create_tablespace.sql)

Execute the file available @ ${OBDX INSTALLER}/installables/db/OBDX/ obdx_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 OBDX_${POST_FIX } DATAFILE '${DATAFILE_PATH}/OBDX_${ POST_FIX }.dbf'

SIZE 500M

AUTOEXTEND ON NEXT 100M

LOGGING

EXTENT MANAGEMENT LOCAL

SEGMENT SPACE MANAGEMENT AUTO;

2.2 Create Audit tablespace (file obdx_audit_create_tablespace.sql)

Execute the file available @ ${OBDX INSTALLER}/installables/db/OBDX/ obdx_audit_create_tablespace.sql

Example :-

Refer installer.properties file variable POST_FIX and replace in the below command

CREATE BIGFILE TABLESPACE OBDX_AUDIT_${ POST_FIX}

DATAFILE '${DATAFILE_PATH} /OBDX_AUDIT_${ POST_FIX}.dbf'

SIZE 500M

AUTOEXTEND ON NEXT 100M

LOGGING

EXTENT MANAGEMENT LOCAL

SEGMENT SPACE MANAGEMENT AUTO;

2.3 Create user (file obdx_create_user.sql)

Execute the file available @ ${OBDX INSTALLER} /installables/db/OBDX/obdx_create_user.sql

Example: -

Refer installer.properties file variable POST_FIX and replace in the below command

create user OBDX_${ POST_FIX} identified by welcome1;

alter user OBDX_${ POST_FIX} default tablespace OBDX_${ POST_FIX};

alter user OBDX_${ POST_FIX} temporary tablespace temp;

alter user OBDX_${ POST_FIX} quota unlimited on OBDX_${ POST_FIX} ;

alter user OBDX_${ POST_FIX} quota unlimited on OBDX_AUDIT_${ POST_FIX} ;

2.4 Create role (file obdx_create_role.sql)

Execute the file available @ ${OBDX INSTALLER} /installables/db/OBDX/obdx_create_role.sql

Example:-

Refer installer.properties file variable POST_FIX and replace in the below command

CREATE ROLE OBDX_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 OBDX_ROLE_${POST_FIX};

grant OBDX_ROLE_${POST_FIX} to OBDX_${POST_FIX};

2.5 Grants Execution (file clip_user_grants.sql)

Execute the file available @ ${OBDX INSTALLER} /installables/db/OBDX/clip_user_grants.sql

Example:-

Refer installer.properties file variable POST_FIX and replace in the below command

grant create any sequence to OBDX_${POST_FIX};

grant execute on DBMS_LOCK to OBDX_${POST_FIX};

2.6 Files execution in sequences on above schema (ex. OBDX_${POST_FIX})

- clip_master_script.sql
- clip_constraints.sql
- clip_seeds_executable.sql
- clip_master_generic_rest_script.sql

***** SUCCESSFULLY installed OBDX database *****

2.7 OBPM Database Installation (OBPM Favor)

Create required OBDX tablespace and user in below sequence

2.8 Tablespace Creation (file obpm_create_tablespace.sql)

Execute the file available @ ${OBDX 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 @ ${OBDX 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 @ ${OBDX 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 @ ${OBDX INSTALLER} /installables/db/OBPM/FCUBS_GR_PRIV.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/FCOBPM_GR_PRIV.sql

2.14 Scripts Execution

- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/table-scripts.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/ubs_object_scripts.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/obpm_object_scripts.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/execute-seeds.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/obpm-seeds.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/DIGX_FW_CONFIG_ALL_O.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/DIGX_FW_ABOUT_OBPM.sql

- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/DIGX_FW_CONFIG_VAR_B.sql
- Execute the file available @ ${OBDX INSTALLER} /installables/db/OBPM/DIGX_FW_CONFIG_UBS_ALL_O.sql

2.15 Policy Seeding

TEMP_PATH=Temparory Path

cp ${OBDX INSTALLER}/installables/policies/Entitlement_log4j.properties to TEMP_PATH/db/Entitlement_log4j.properties

cp ${OBDX INSTALLER}/installables/policies/Task_log4j.properties to TEMP_PATH/db/Task_log4j.properties

cp ${OBDX 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=OBDX_${POST_FIX} and SCHEMA_PASS= Password of OBDX_${POST_FIX} .

$JAVA_HOME/bin/java -Djava.util.logging.config.file= TEMP_PATH/db/Task_log4j.properties -jar ${OBDX 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:@OBDX_DATABASE_HOSTNAME:OBDX_DATABASE_PORT/OBDX_DATABASE_SID'

$JAVA_HOME/bin/java -Djava.util.logging.config.file= TEMP_PATH/db/Dashboard_seed_log4j.properties -jar ${OBDX INSTALLER}/installables/policies/com.ofss.digx.utils.dashboard.jar ${OBDX INSTALLER}/installables/policies/dashboard_json/ oracle.jdbc.OracleDriver SCHEMA_NAME SCHEMA_PASS 'jdbc:oracle:thin:@OBDX_DATABASE_HOSTNAME:OBDX_DATABASE_PORT/OBDX_DATABASE_SID'

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

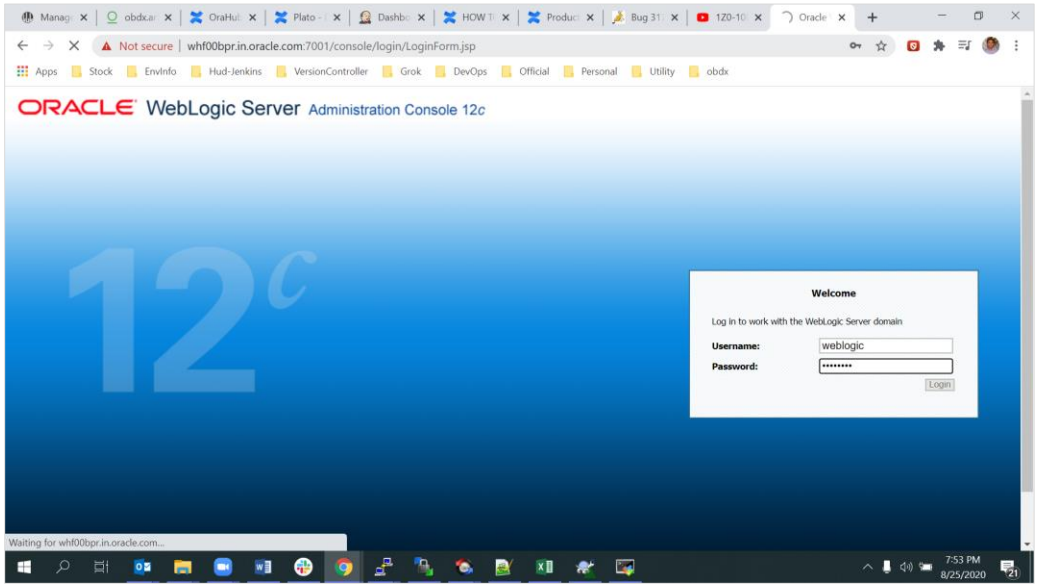
Home

3. WEBLOGIC Setup and Configuration

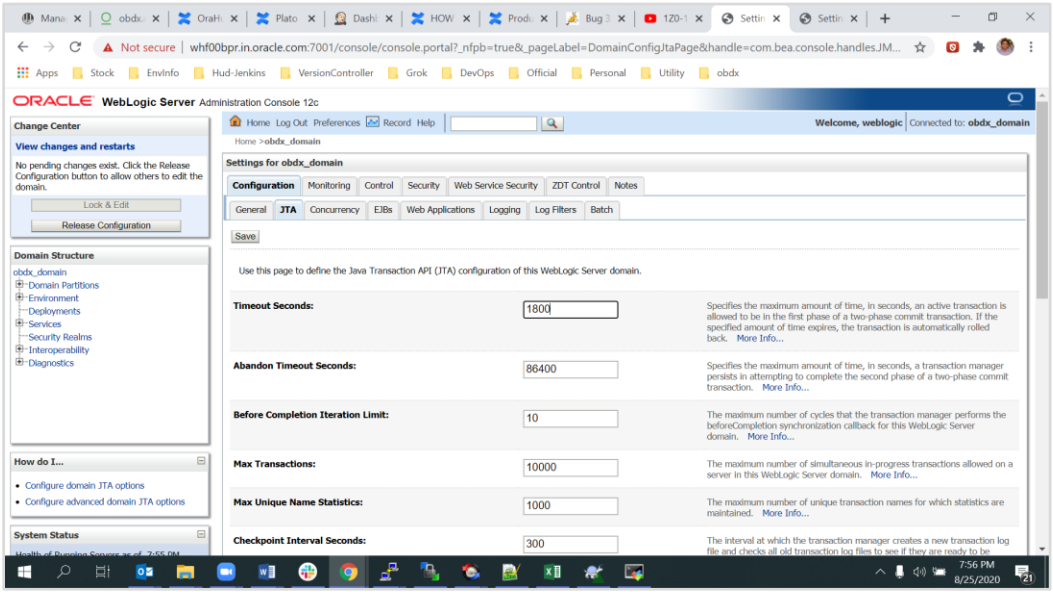
Assuming that rcu , weblogic domain created , managed server, cluster created , node manager configured.

3.1 Setting Domain JTA Transaction timeout

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

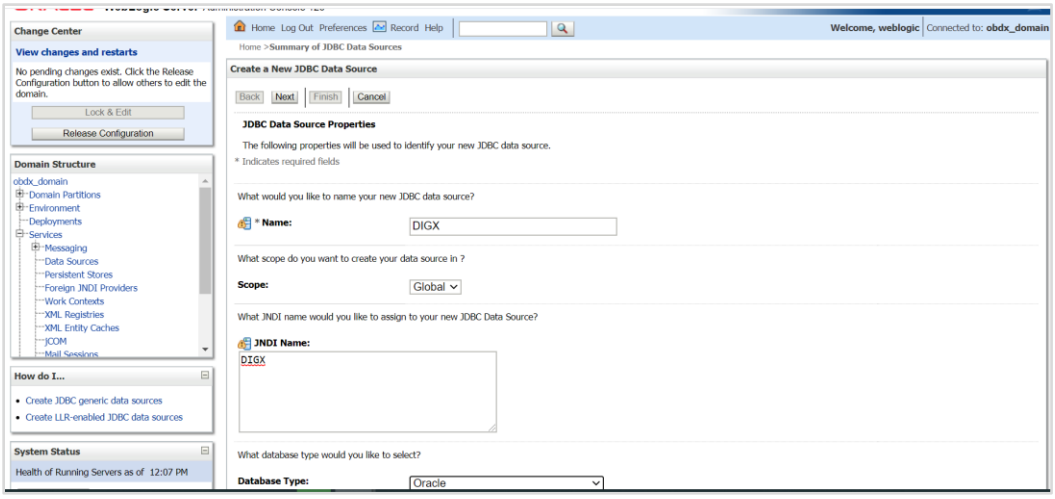


2. click on DOMAIN_NAME→ JTA→ set Timeout Seconds to 1800 → click on save → Activate changes



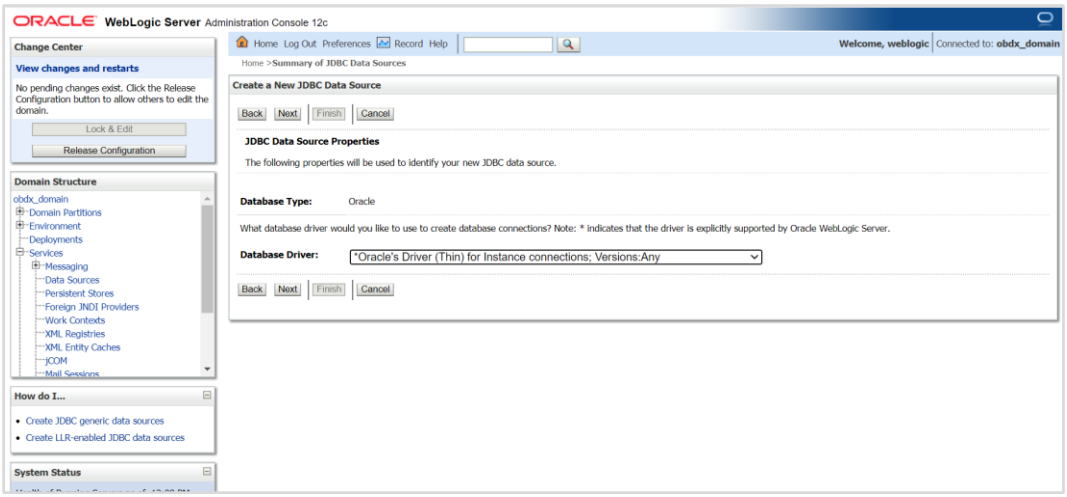
3.2 Creating DIGX data source

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

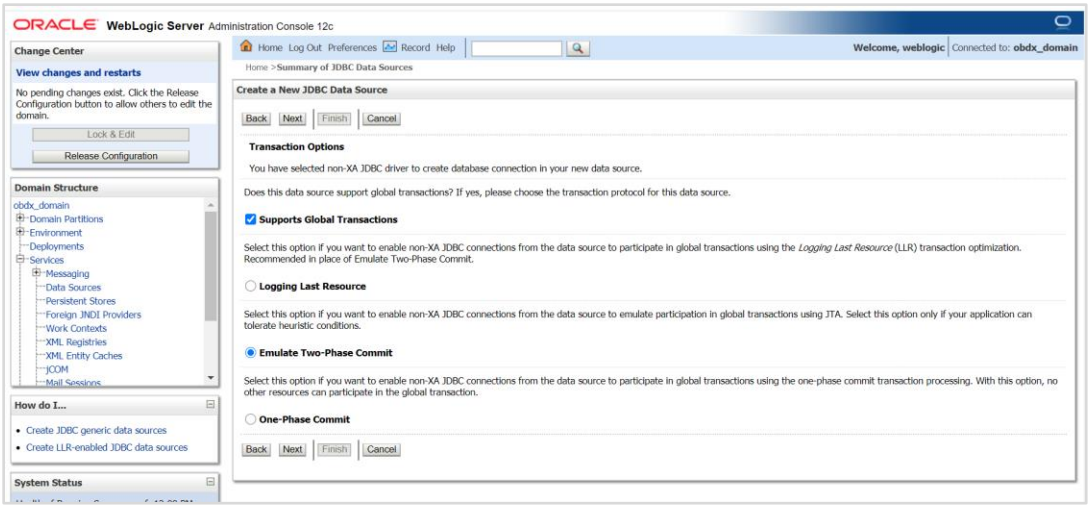


2. **Name: -** DIGX

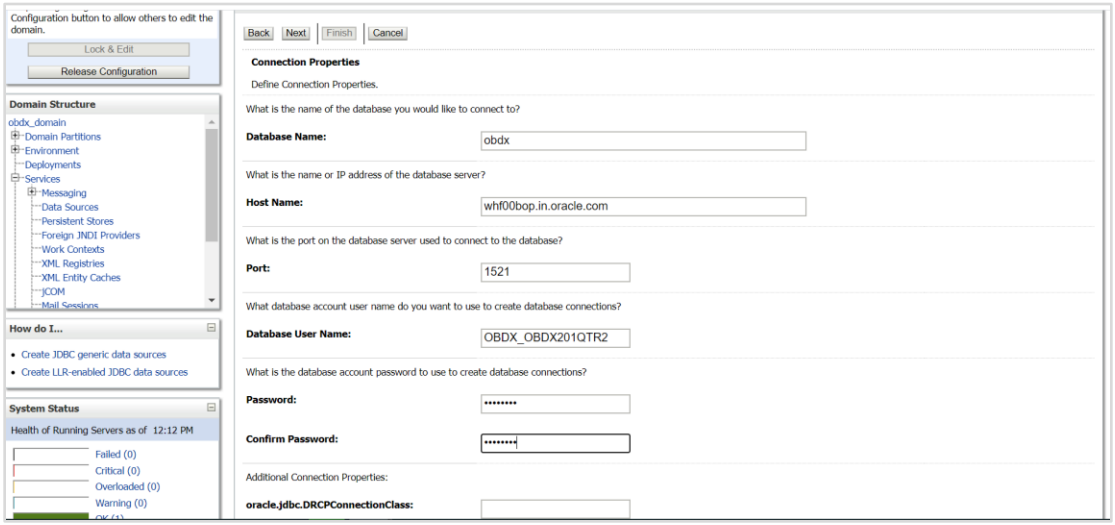
JNDI Name: - DIGX



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



4. Select Emulate Two-Phase Commit



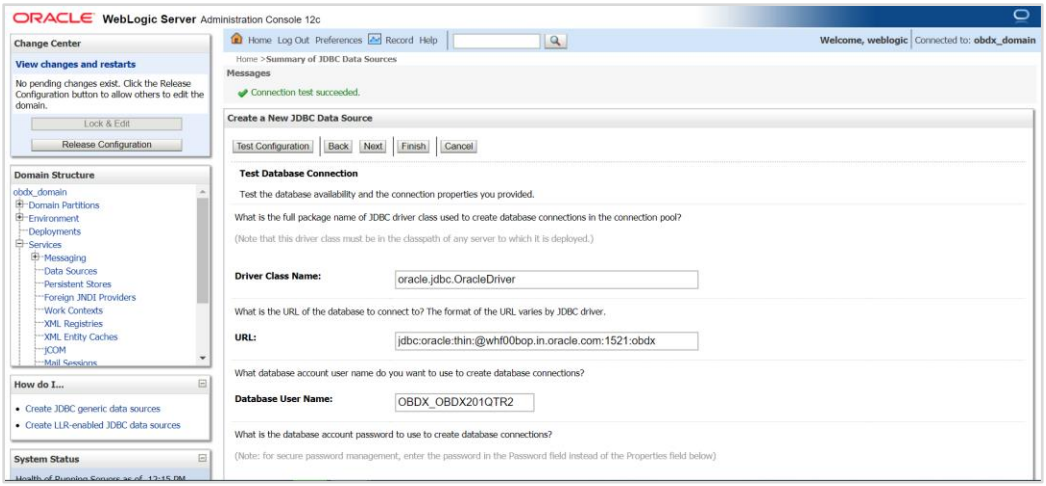
5. Provide

Database Name: - Database SID

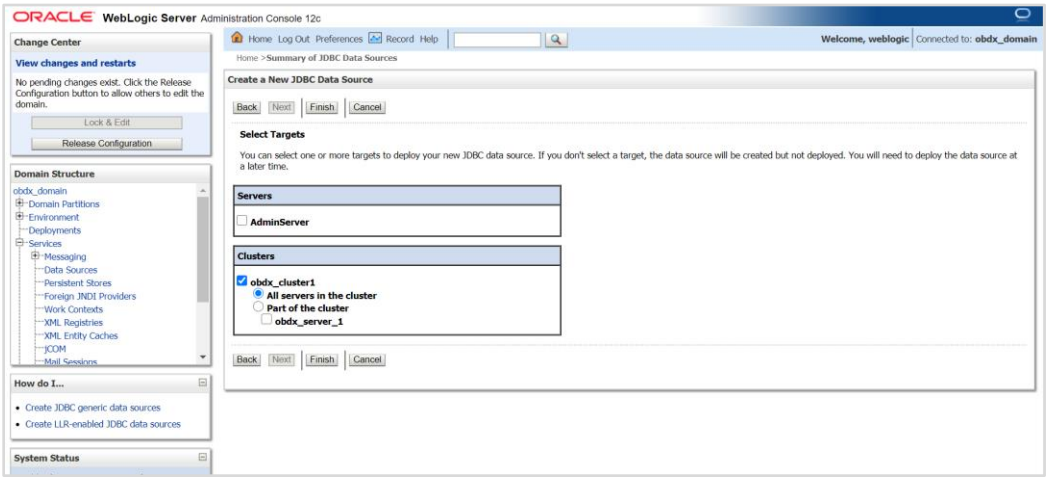
Host Name: - Database hostname

Port: - Database port Number

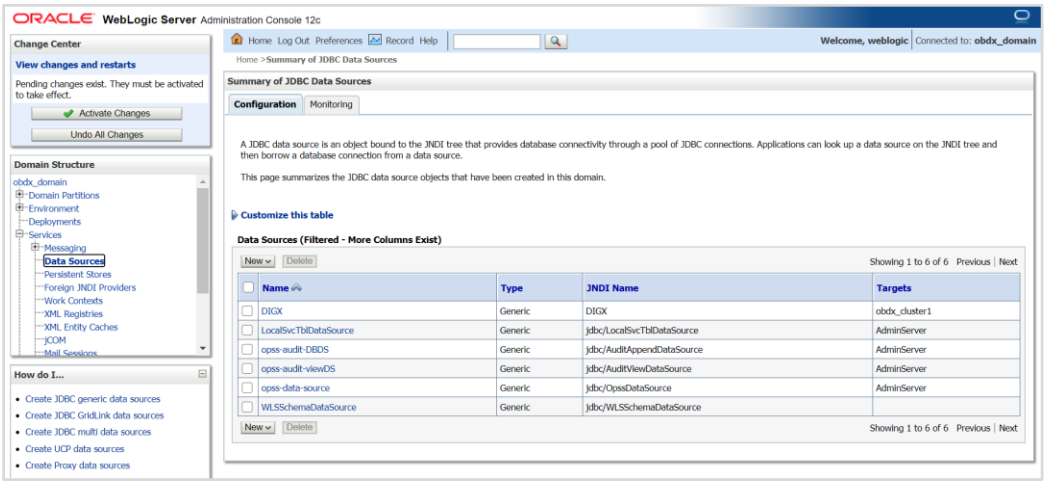
Database user Name: - OBDX_${POST_FIX}



6. Test Configuration

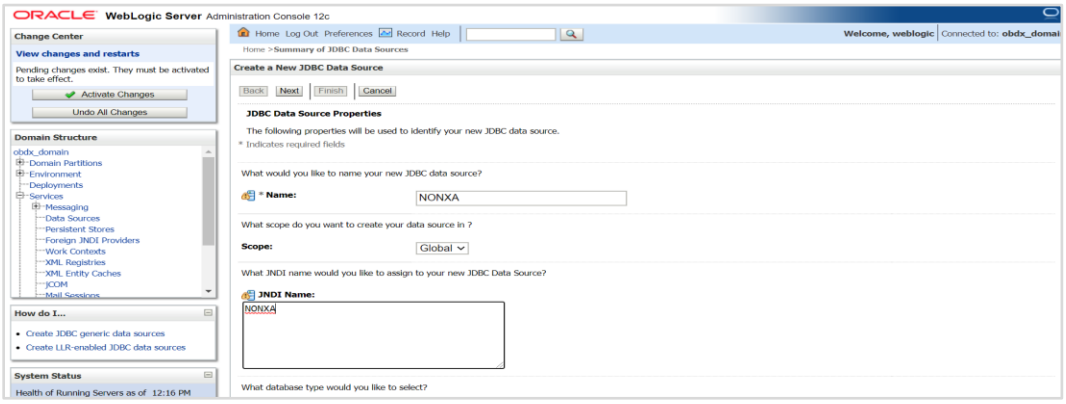


7. Target to cluster



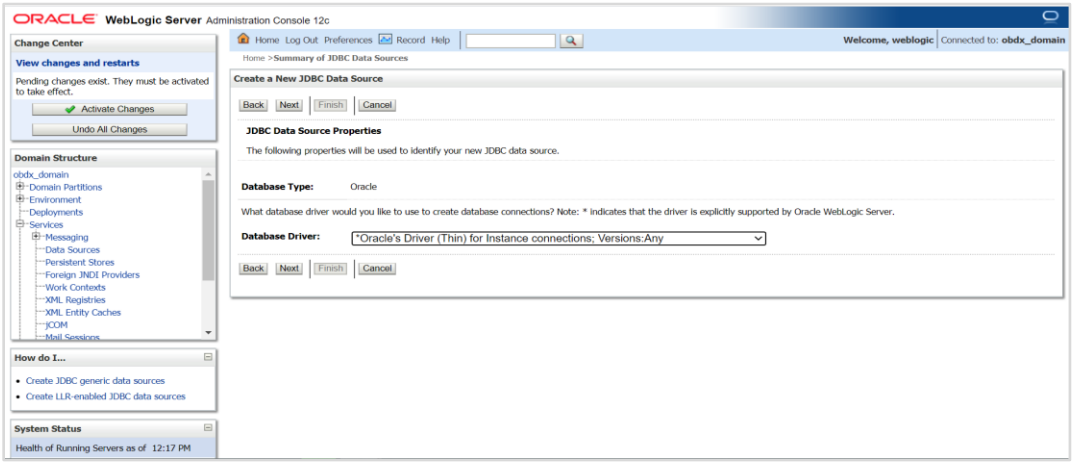
3.3 Creating NONXA data source

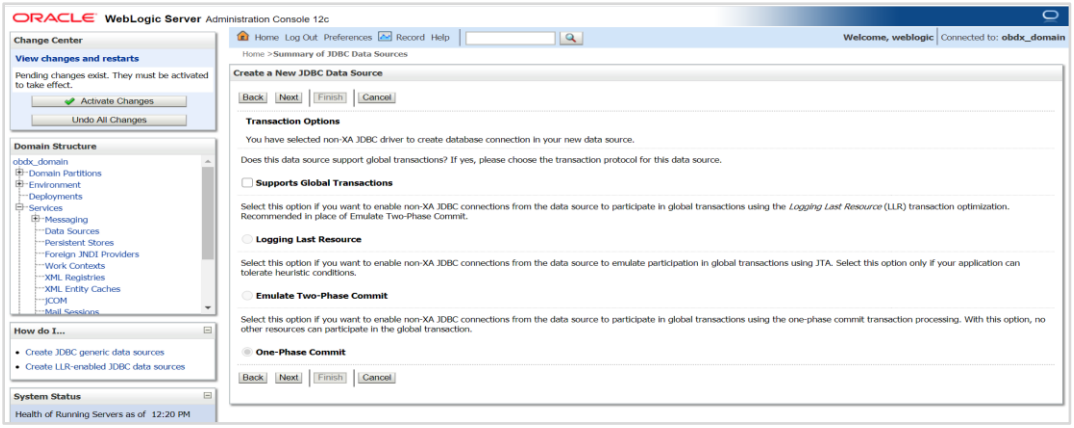
1. Navigate to Data Source → click on new → Provide details and click on finish



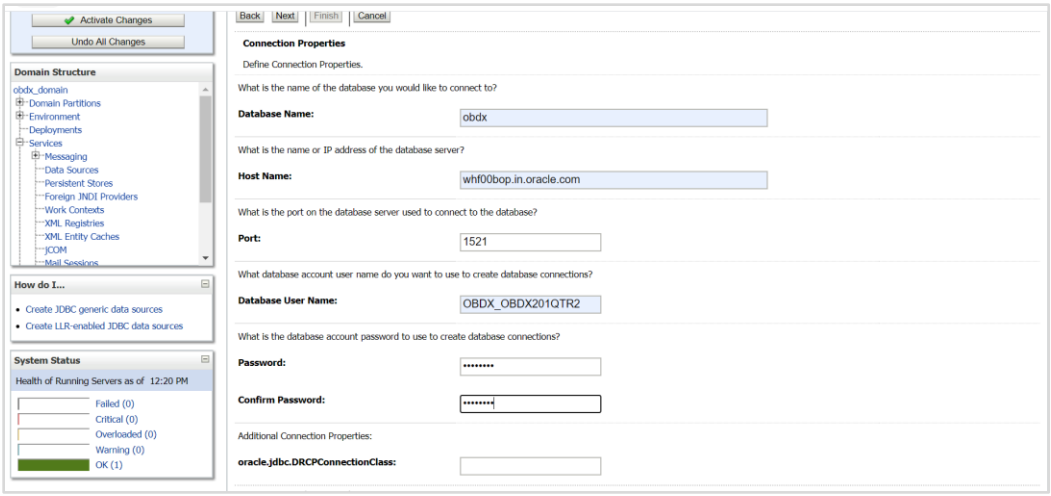
2. **Name :-** NONXA

JNDI Name :- NONXA





3. Click **Next**



4. Provide

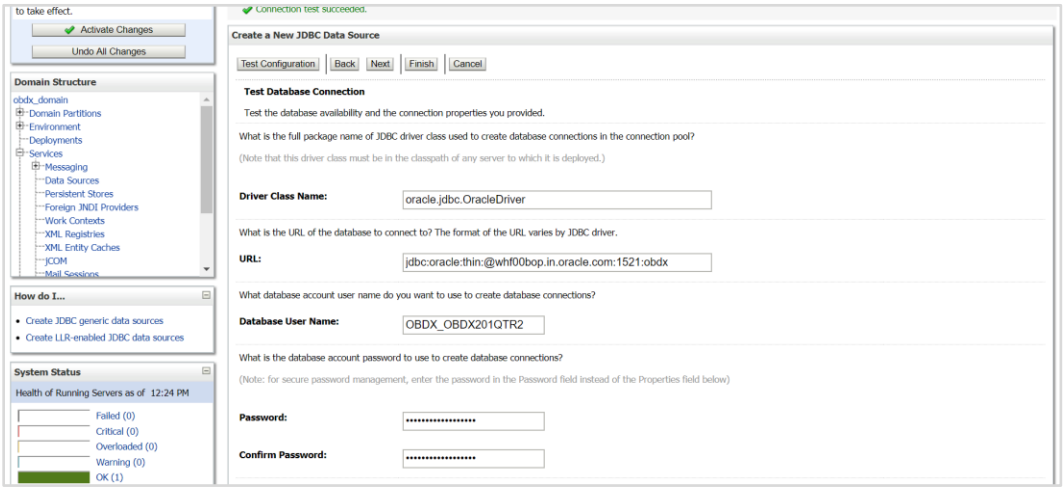
Database Name: - Database SID

Host Name: - Database hostname

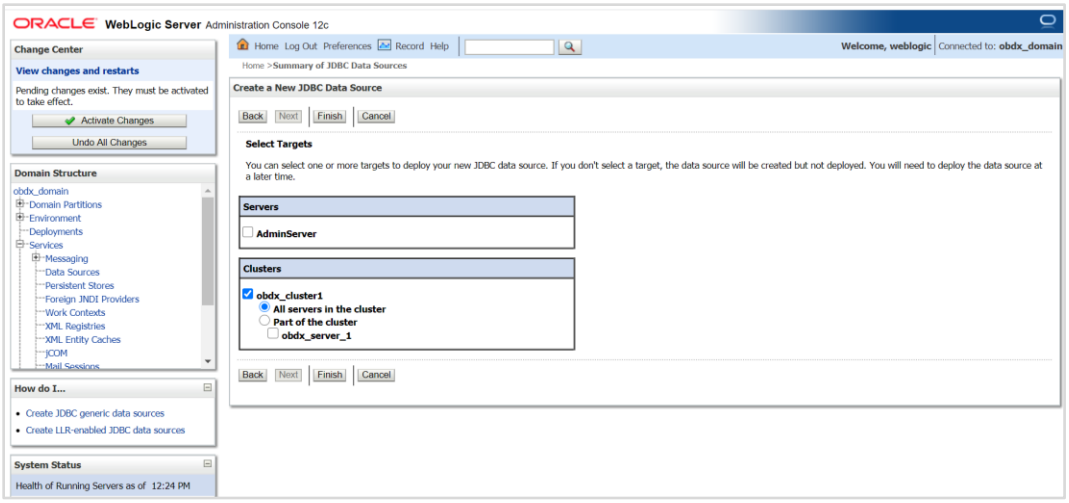
Port: - Database port Number

Database user Name: - OBDX_${POST_FIX}

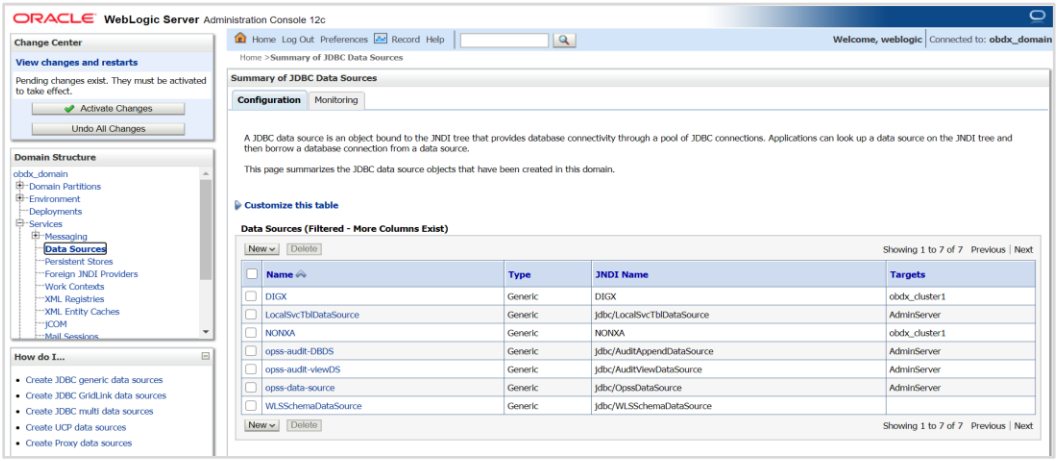
Password:- Database user password



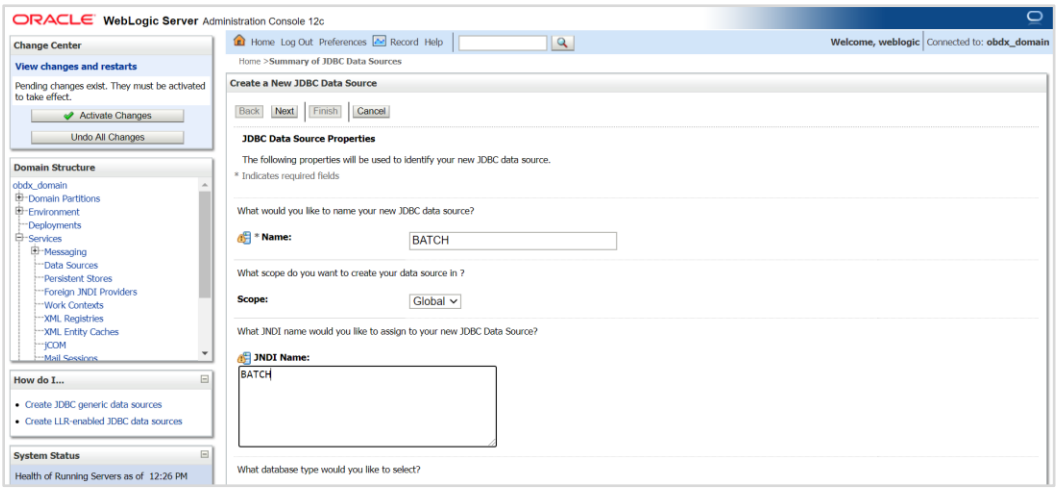
5. Test Configuration



6. Select target as cluster -- > Finish

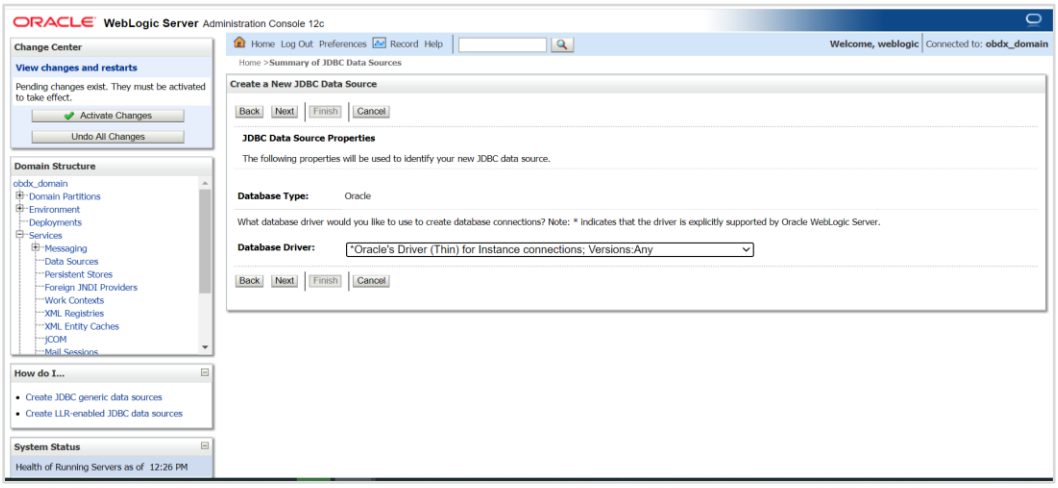


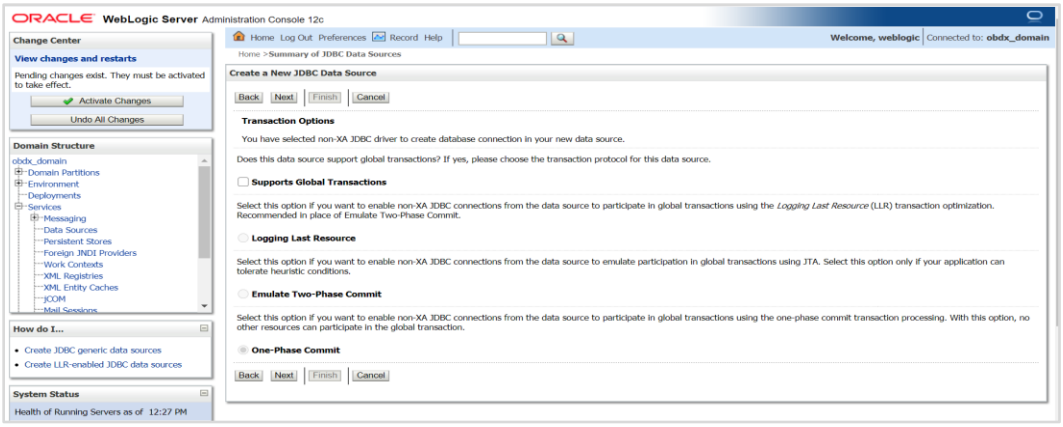
3.4 Creating BATCH data source



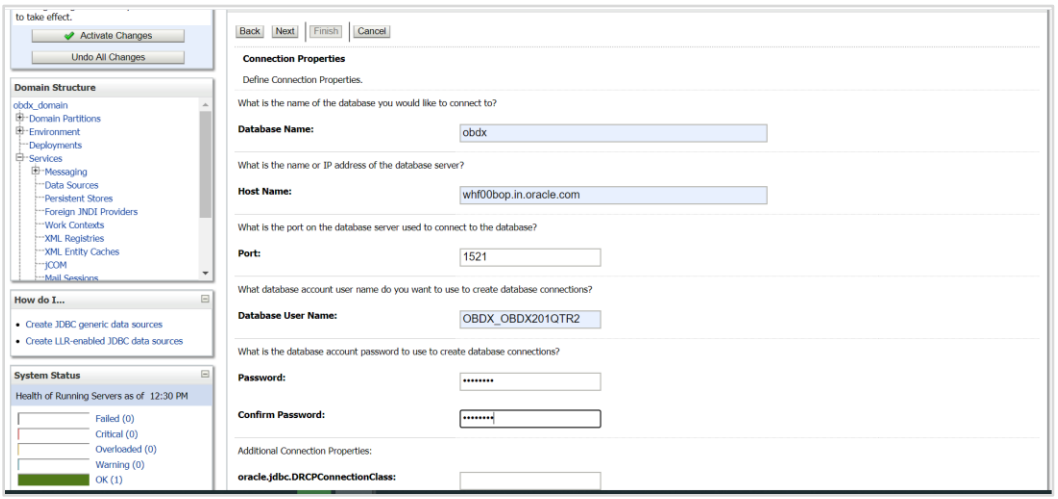
7. **Name :-** BATCH

JNDI Name :- BATCH





8. Click **Next**



9. Provide

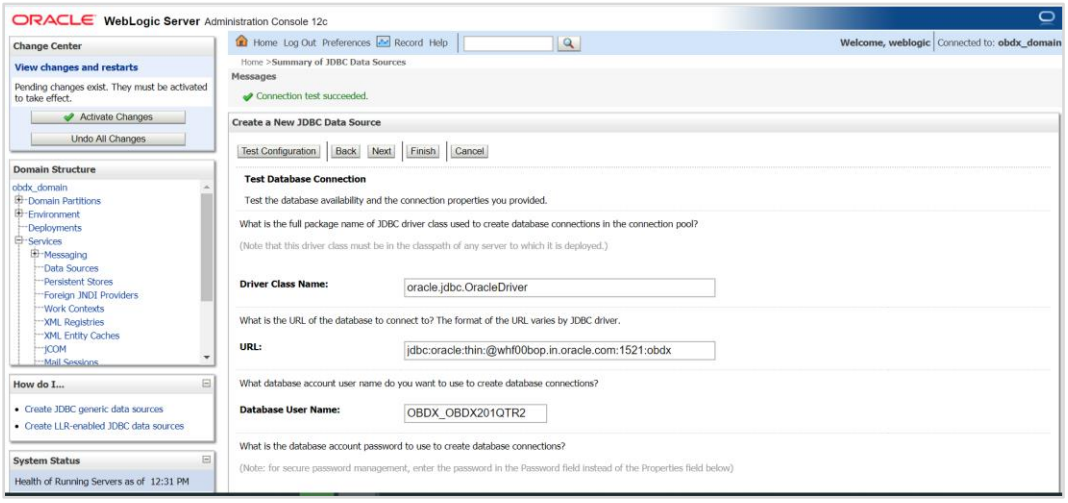
Database Name: - Database SID

Host Name: - Database hostname

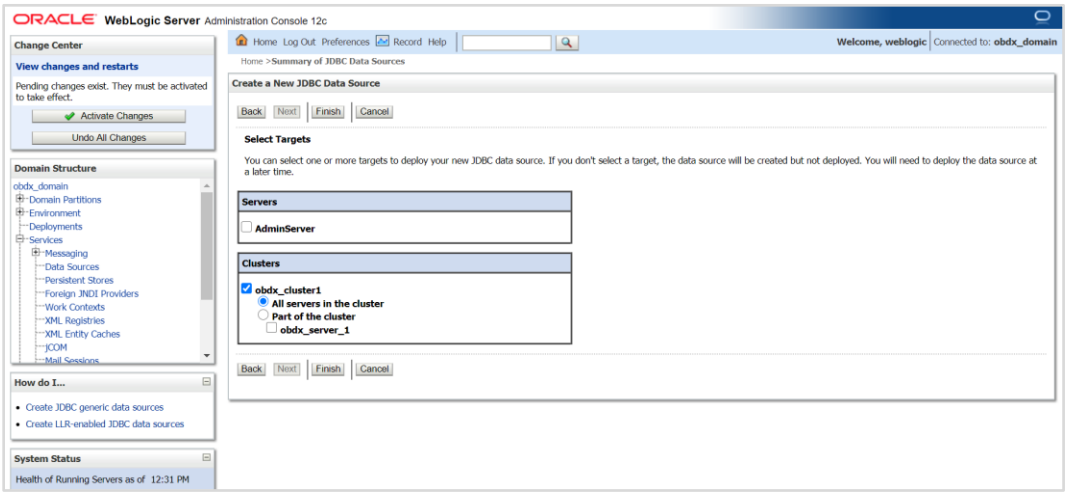
Port: - Database port Number

Database user Name: - OBDX_${POST_FIX}

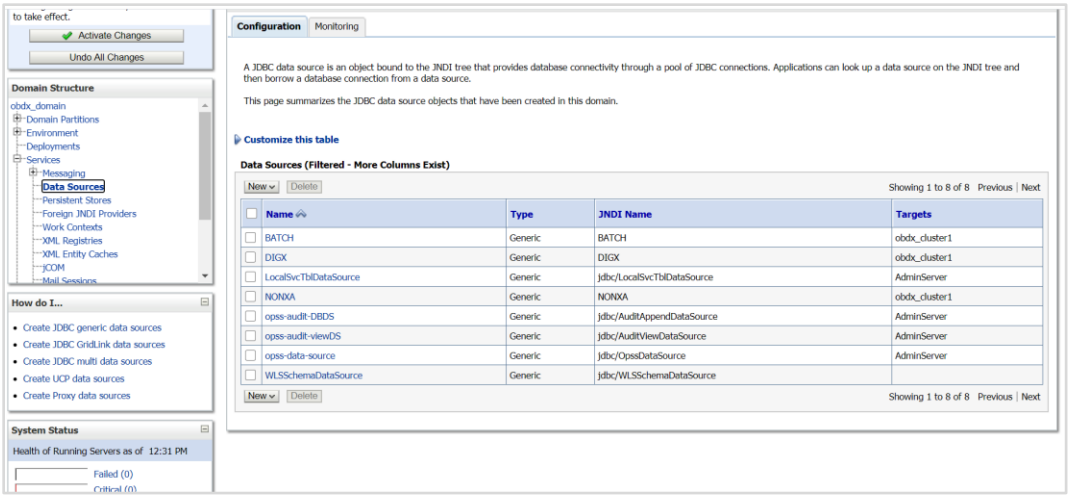
Password:- Database user password



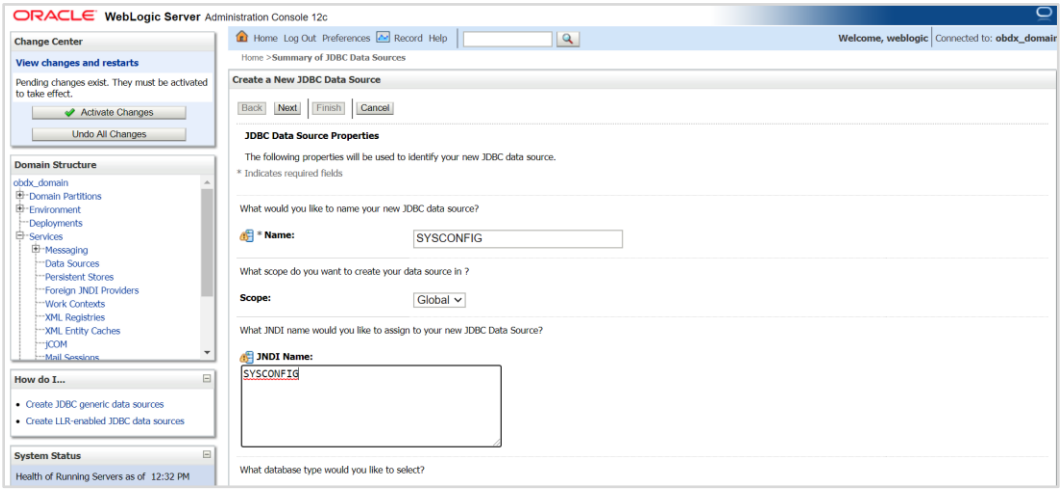
10. Test Configuration



11. Target Cluster and click on Finish

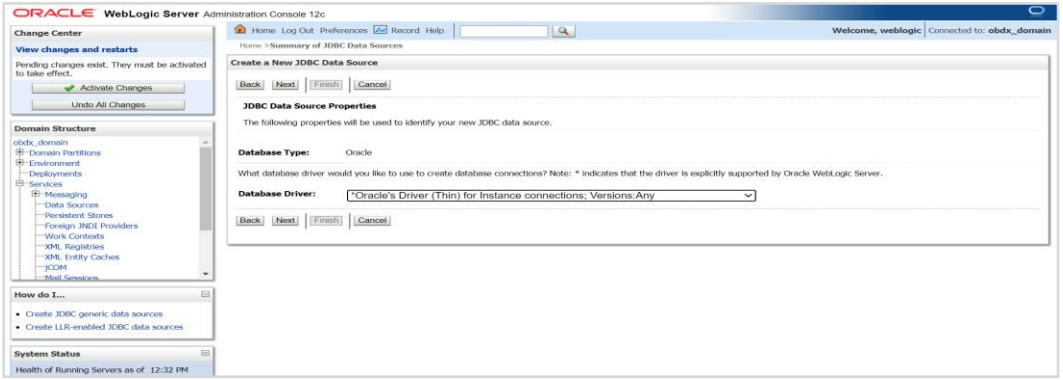

to take effect.
Activate Changes
Undo All Changes
Domain Structure
obdx_domain
Domain Partitions
Environment
Deployments
Services
Messaging
Data Sources
Persistent Stores
Foreign JNDI Providers
Work Contexts
XML Registries
XML Entity Caches
jCOM
Mail Sessions
How do I...
Create JDBC generic data sources
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)
New
Delete
Showing 1 to 8 of 8 Previous | Next
Name
Type
JNDI Name
Targets
BATCH Generic BATCH obdx_cluster1
DIGX Generic DIGX obdx_cluster1
LocalSvcTblDataSource Generic jdbc/LocalSvcTblDataSource AdminServer
NONXA Generic NONXA obdx_cluster1
opss-audit-DBDS Generic jdbc/AuditAppendDataSource AdminServer
opss-audit-viewDS Generic jdbc/AuditViewDataSource AdminServer
opss-data-source Generic jdbc/OpssDataSource AdminServer
WLSSchemaDataSource Generic jdbc/WLSSchemaDataSource
New
Delete
Showing 1 to 8 of 8 Previous | Next

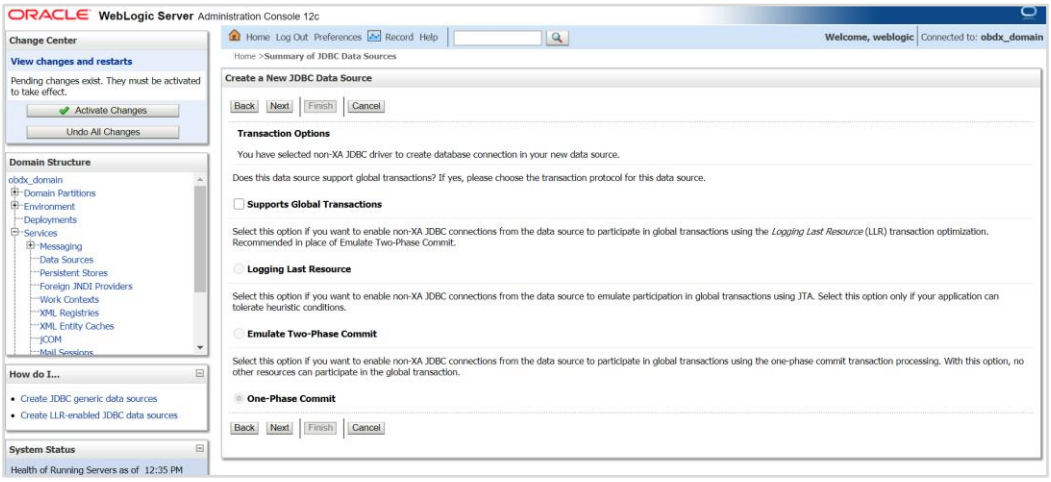
3.5 Creating SYSCONFIG data source



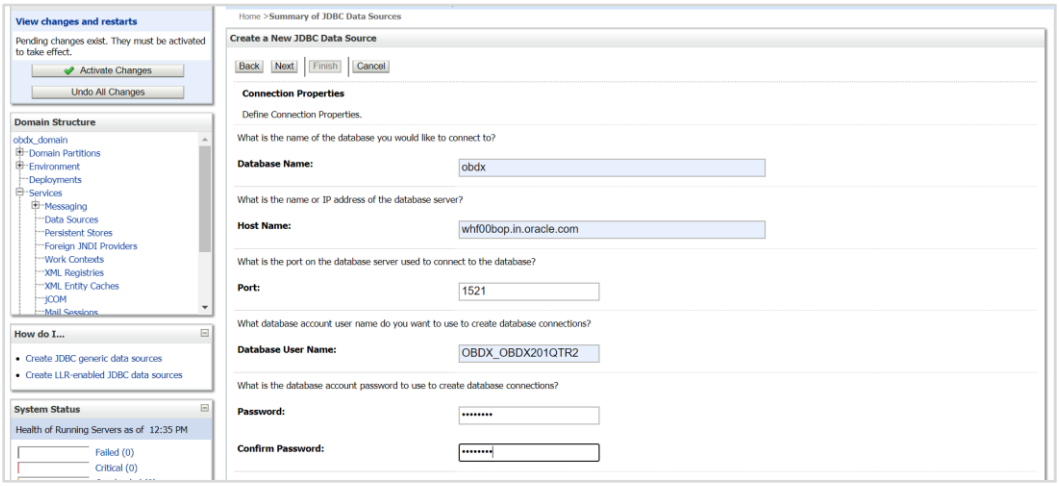
12. **Name :-** SYSCONFIG

JNDI Name :- SYSCONFIG





13. Click on Next



14. Provide

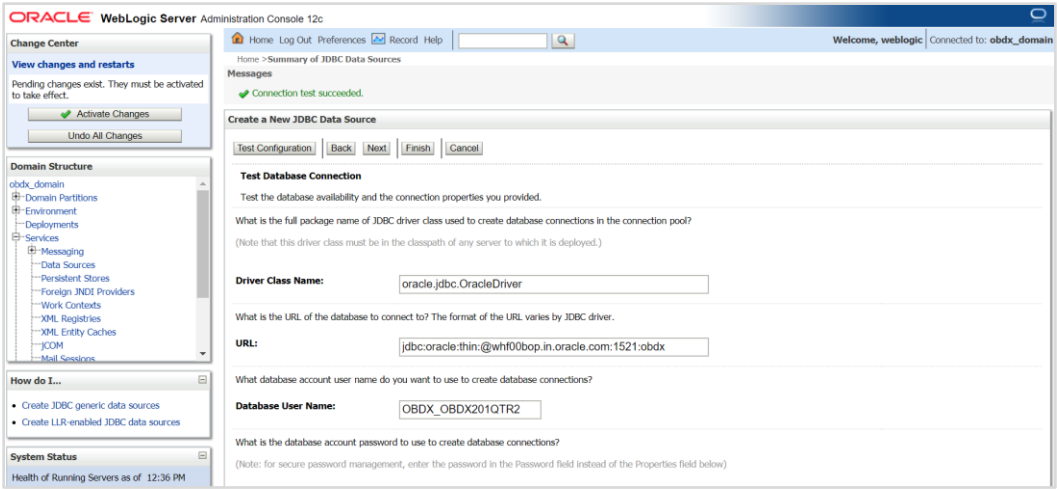
Database Name: - Database SID

Host Name: - Database hostname

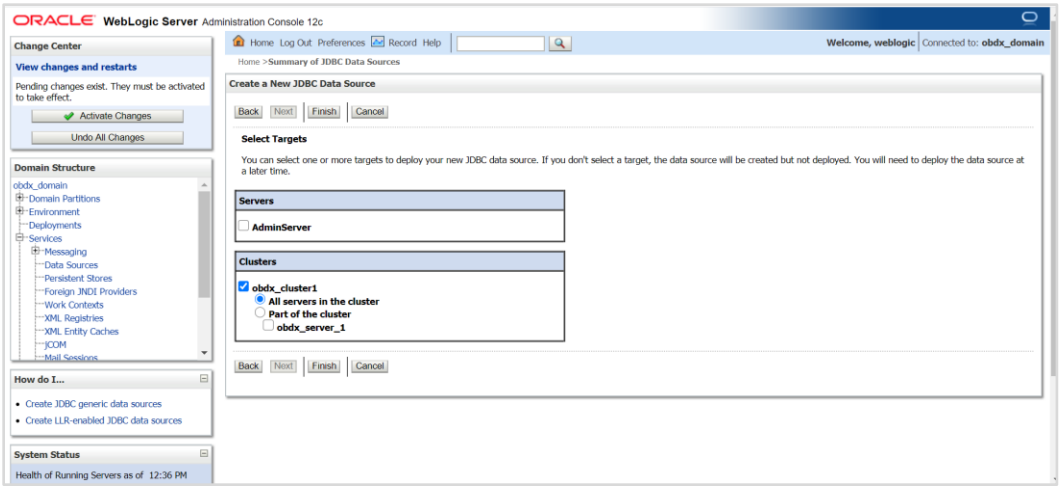
Port: - Database port Number

Database user Name: - OBDX_${POST_FIX}

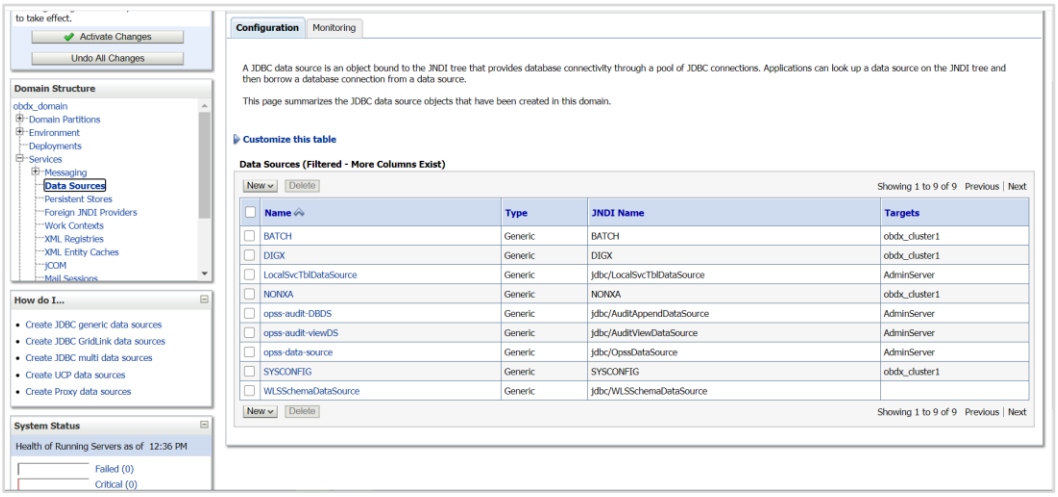
Password:- Database user password



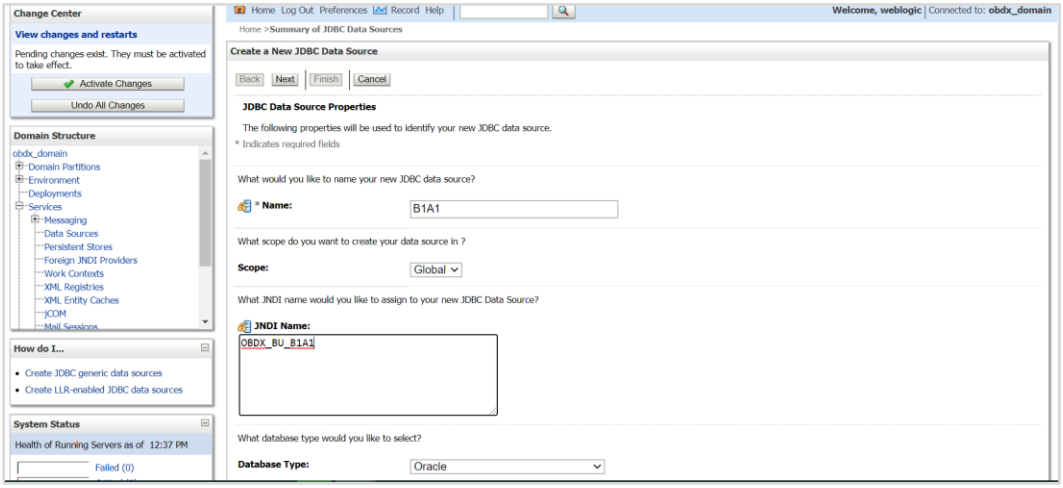
15. Test Configuration



16. Select target as cluster and click on Finish

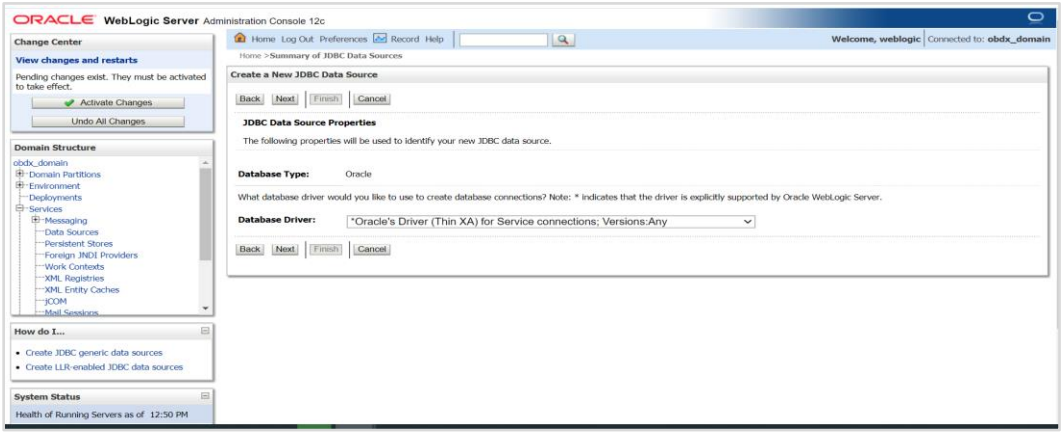


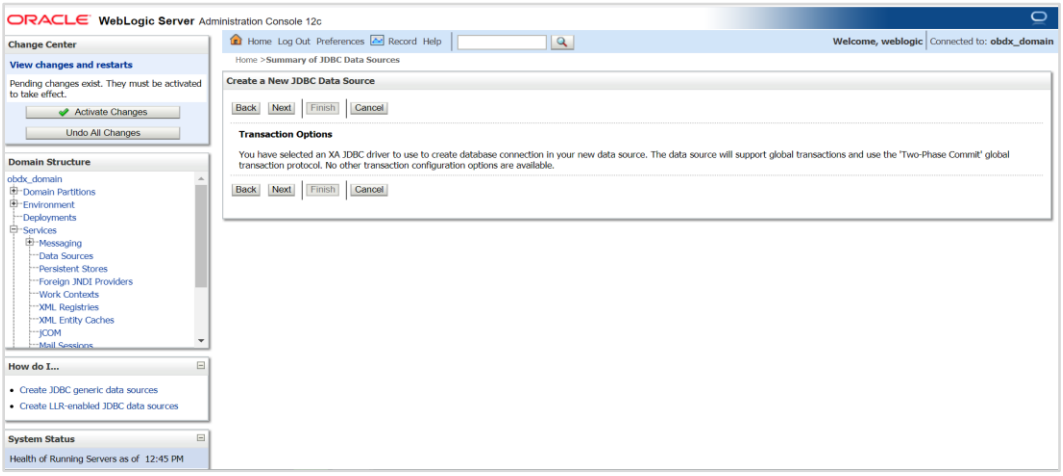
3.6 Creating B1A1 data source



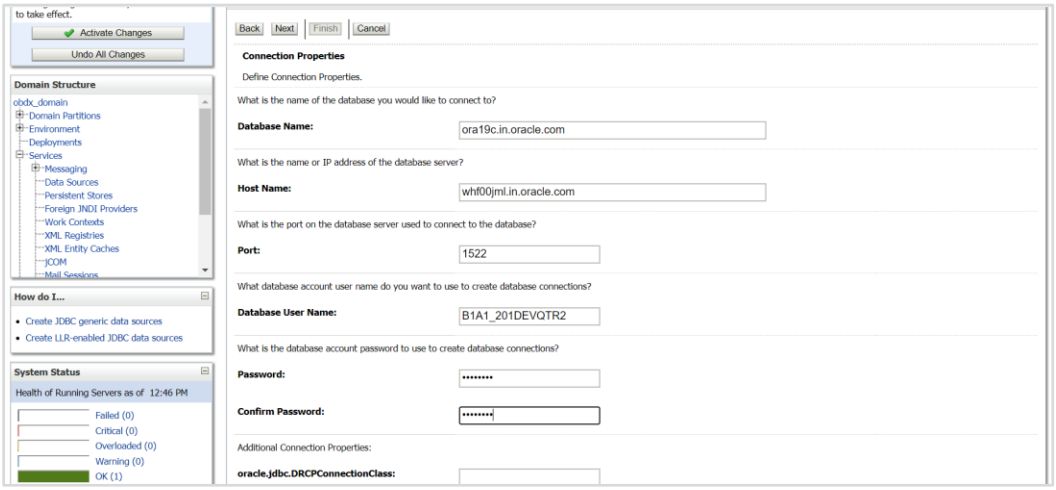
17. **Name:-** B1A1

JNDI Name :- OBDX_BU_B1A1





18. Click on Next



19. Provide

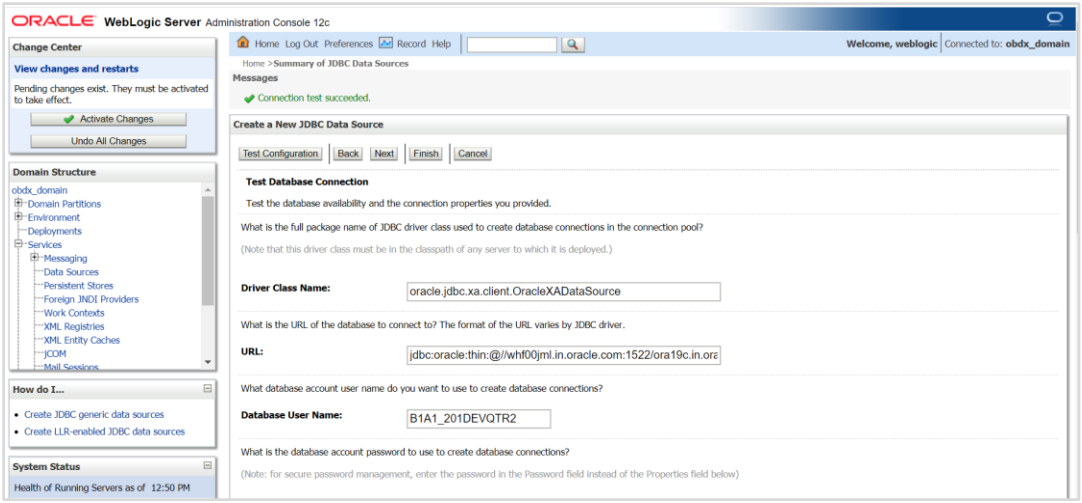
Database Name: - Database SID ($EHMS_DATABASE_SID)

Host Name: - Database hostname ($EHMS_DATABASE_HOSTNAME)

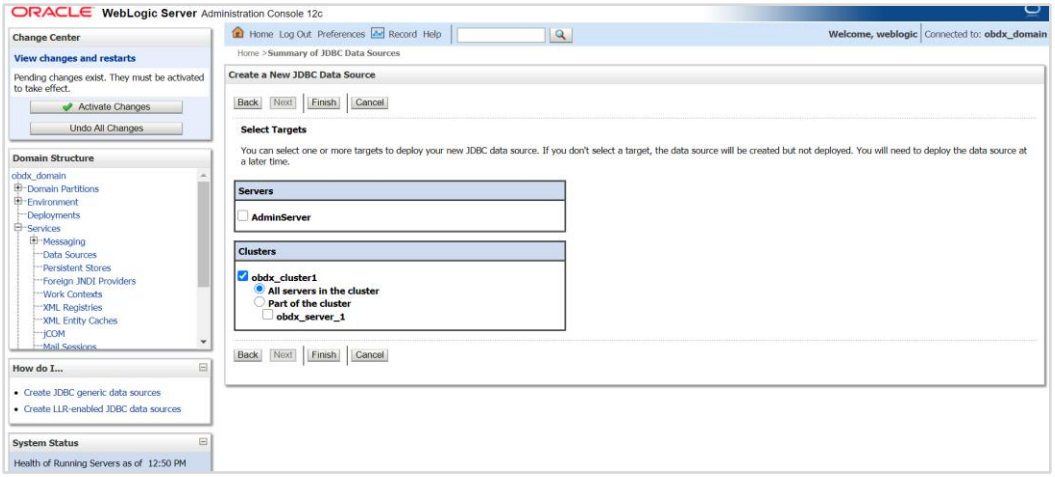
Port: - Database port Number ($EHMS_DATABASE_PORT)

Database user Name: - ${ EHMS_SCHEMA_NAME }

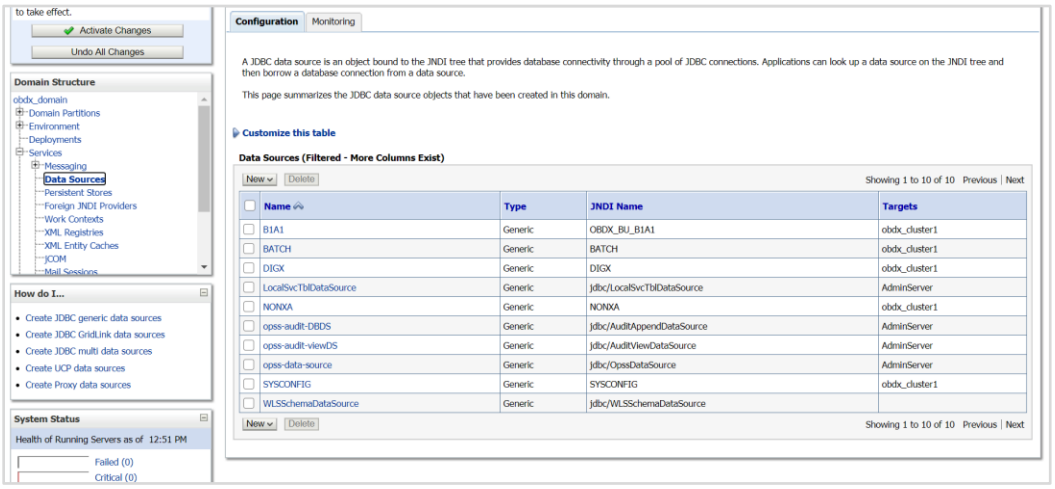
Password: - Database user ${ EHMS_SCHEMA_NAME } password



20. Test Configuration

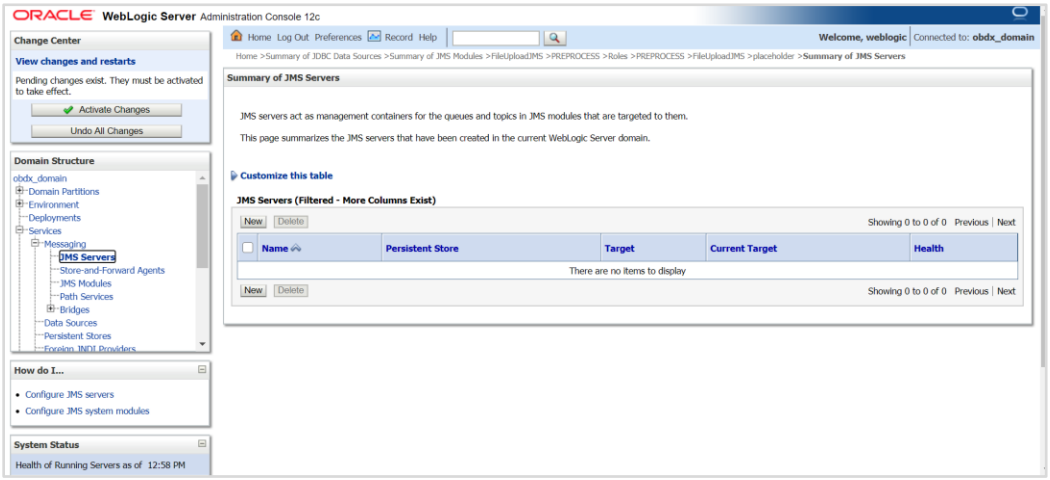


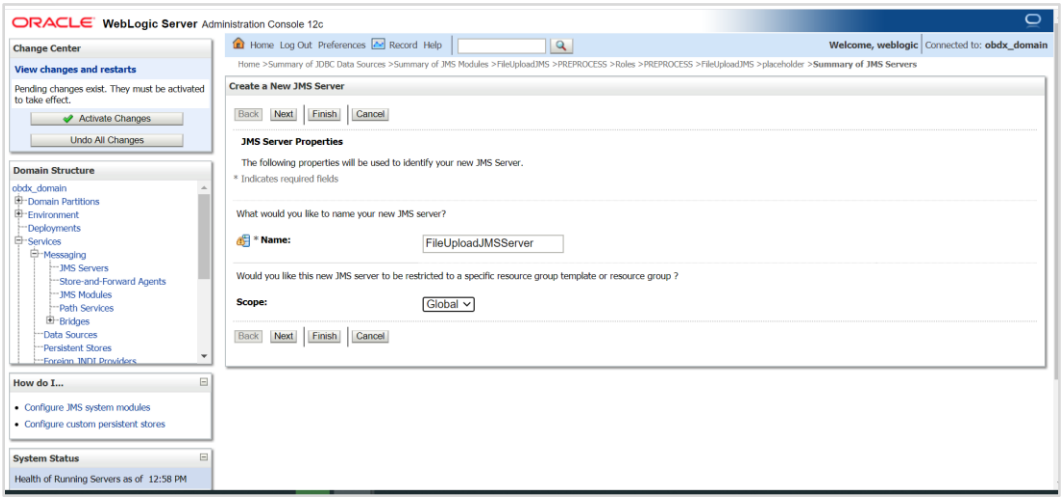
21. Set target as cluster and click on Finish



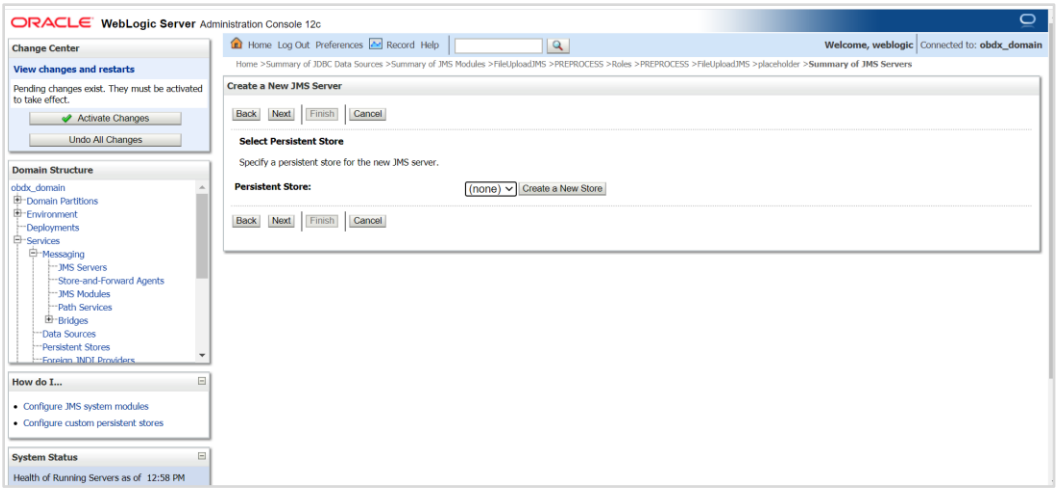
3.7 Create JMS server and JMS Module

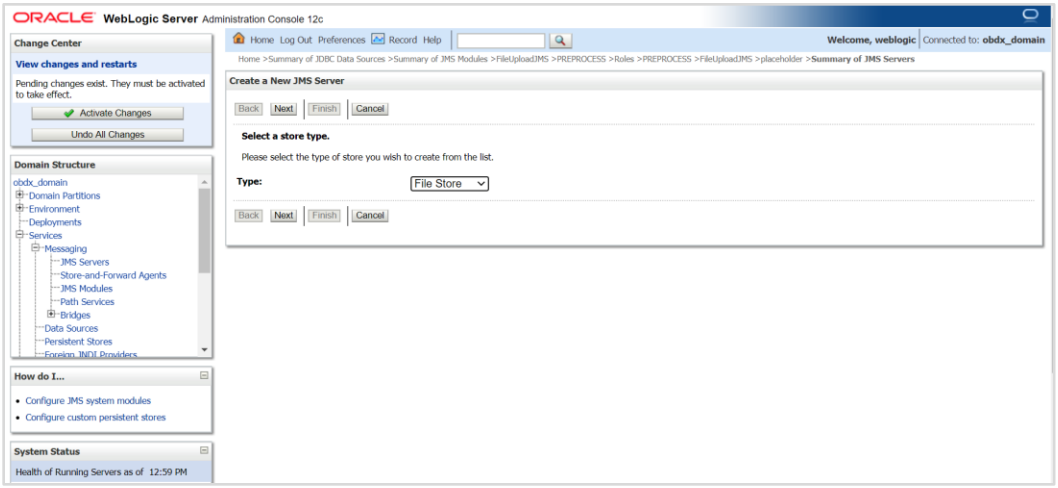
- Creating FileUploadJMS JSM Module
- Creating WLS_JMS_FILEUPLOAD_PS FileStore
- Creating FileUploadJMSServer JMS Server



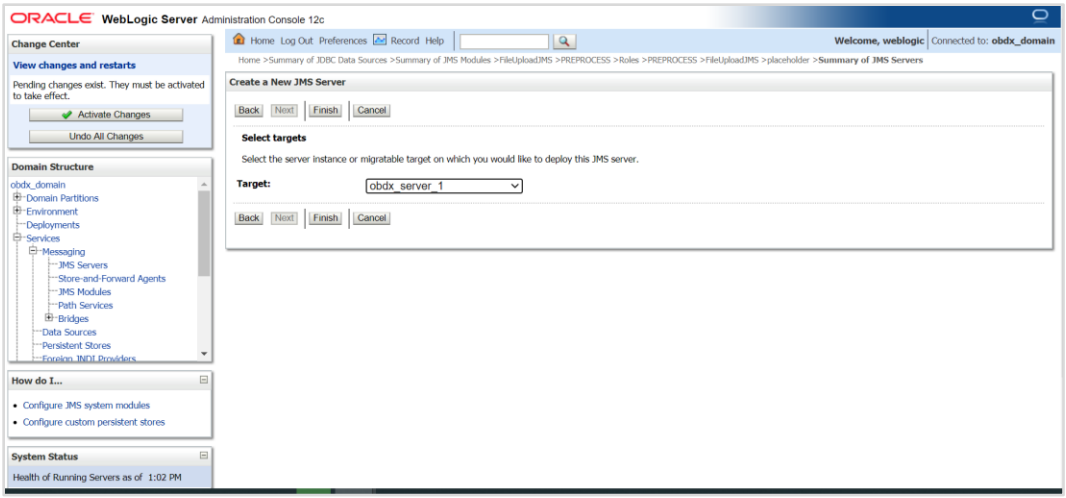


22. Click on JMS Servers → Name – FileUploadJMSServer -- > Click on Next

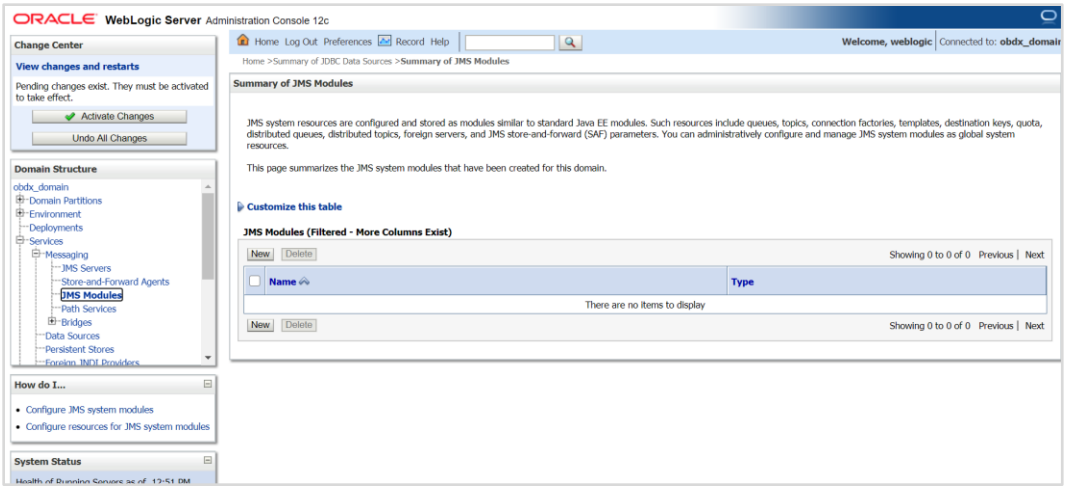




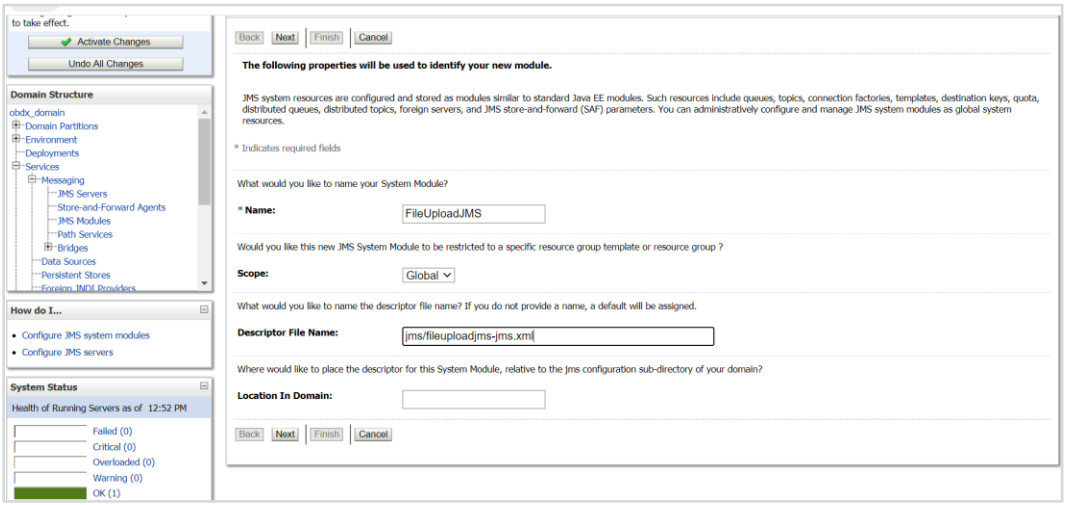
23. Select Type as File Store and click on Next



24. Select target as managed server and click on Finish



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

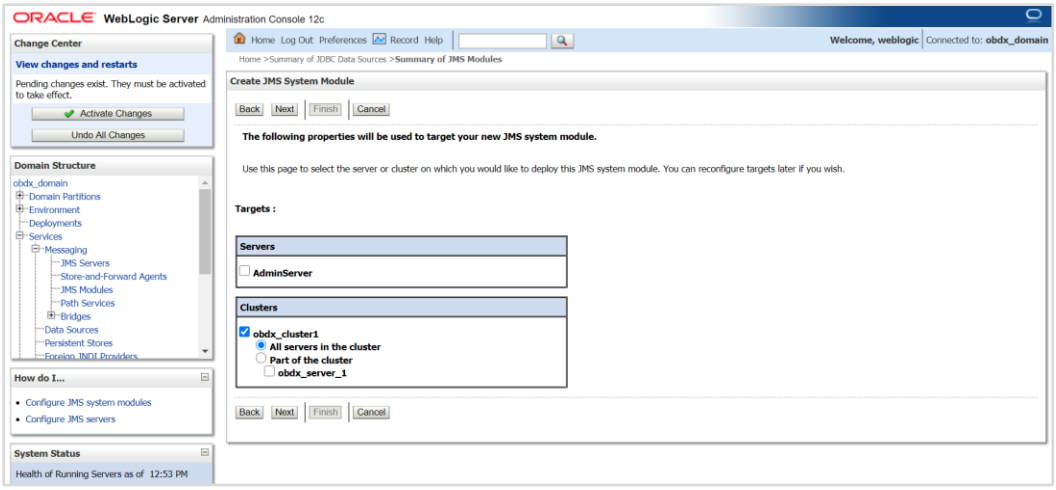


26. **Name :-** FileUploadJMS

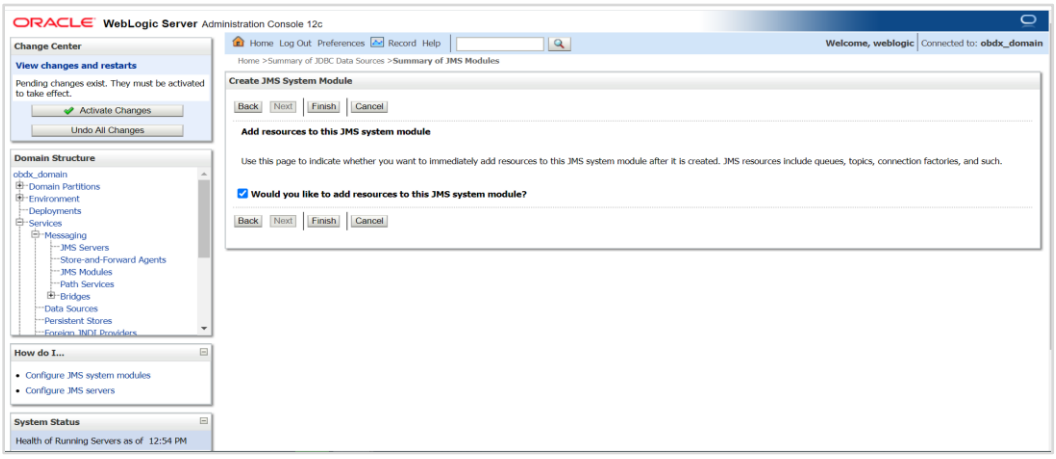
    Scope:- Global

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

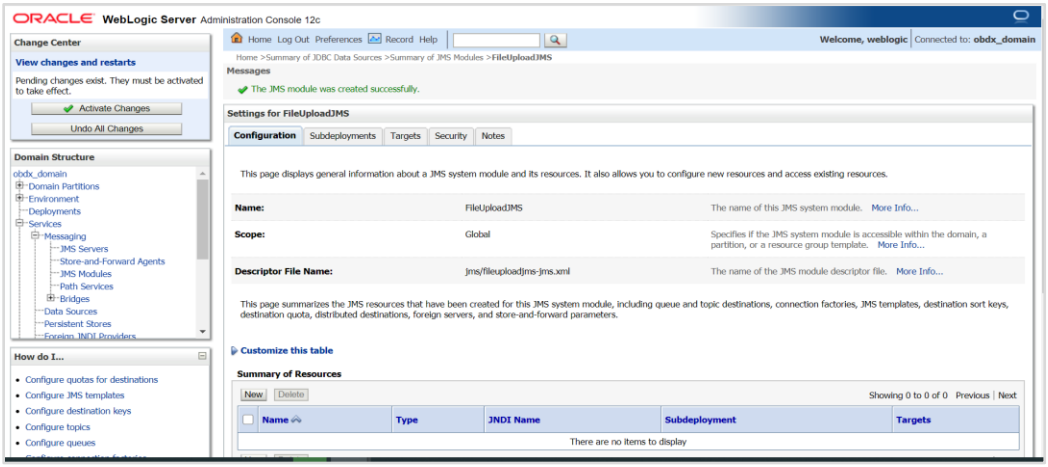
27. Click on Next



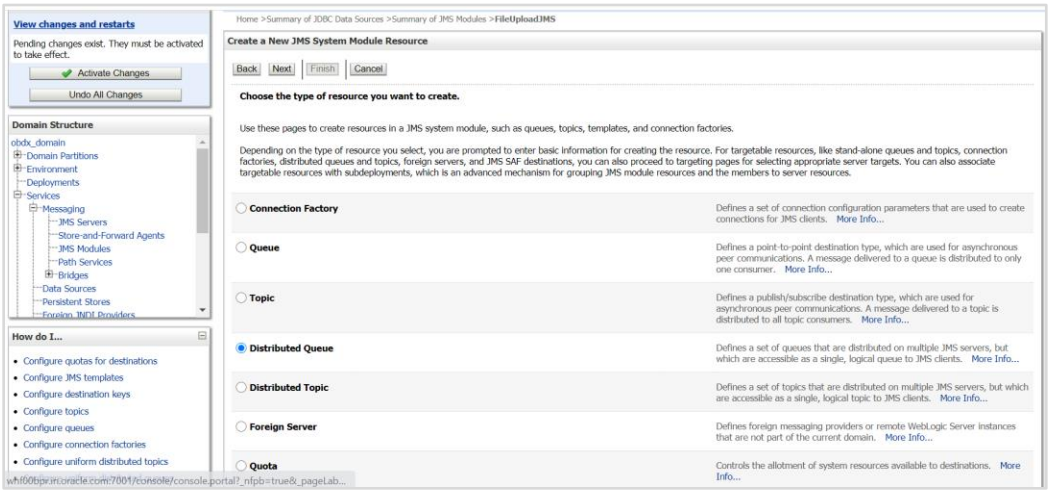
28. Set target as cluster → click on Next



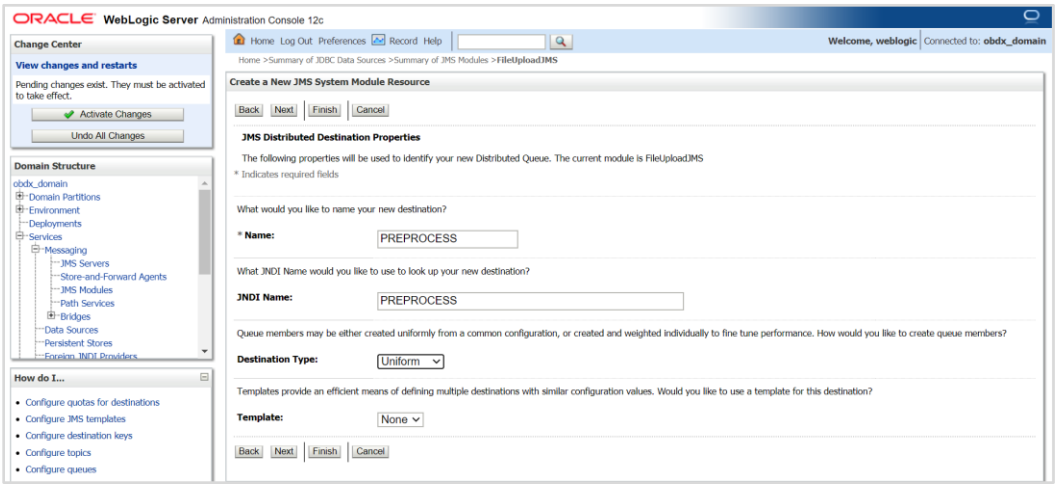
29. Select Would you like to add resources to this JMS system module and click on finish



30. Select new



31. Select Distributed Queue and click next



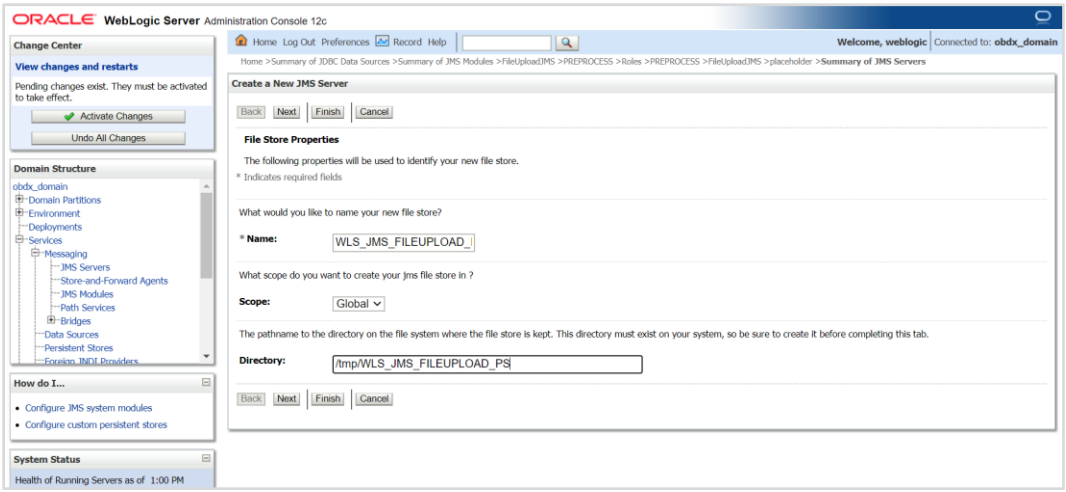
32. Provide

Name: - PREPROCESS

JNDI Name: - PREPROCESS

Destination Type: - Uniform

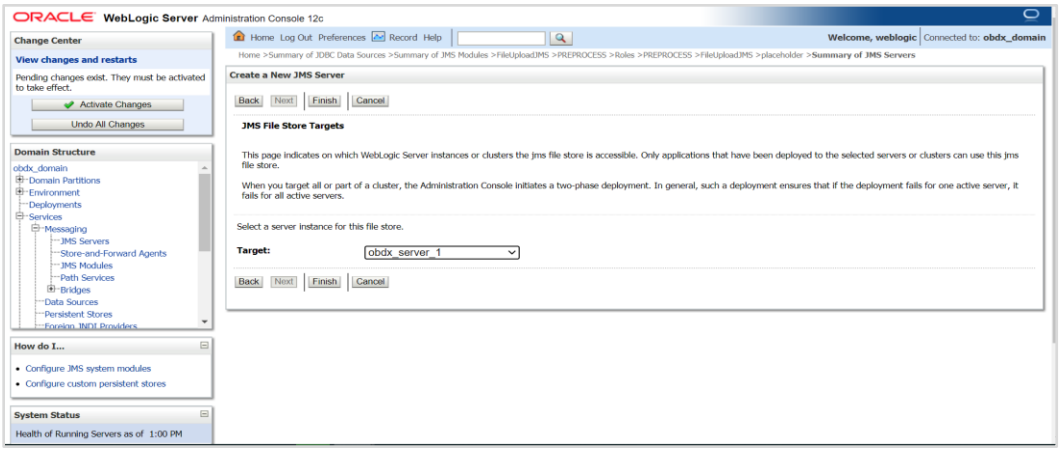
Template: - None



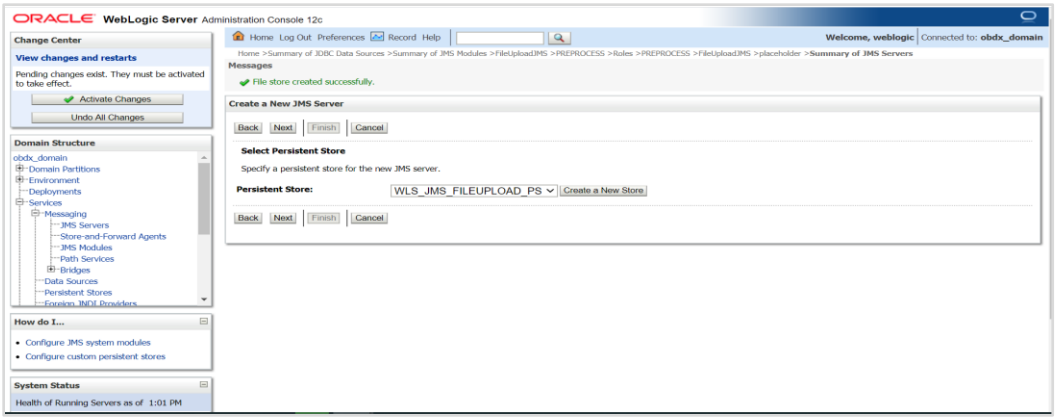
33. **Name :-** WLS_JMS_FILEUPLOAD_PS

    Scope :- Global

    Directory :- /tmp/WLS_JMS_FILEUPLOAD_PS

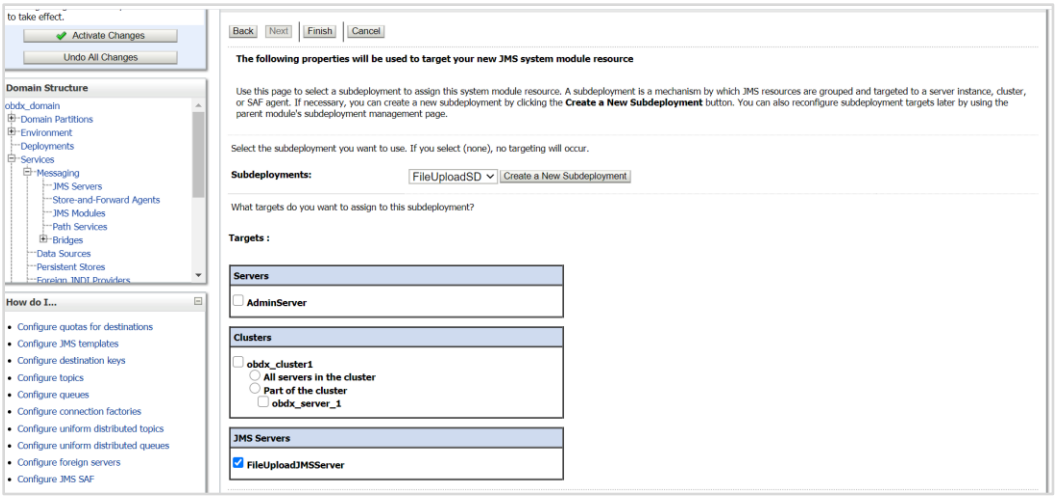


34. Select target as managed server

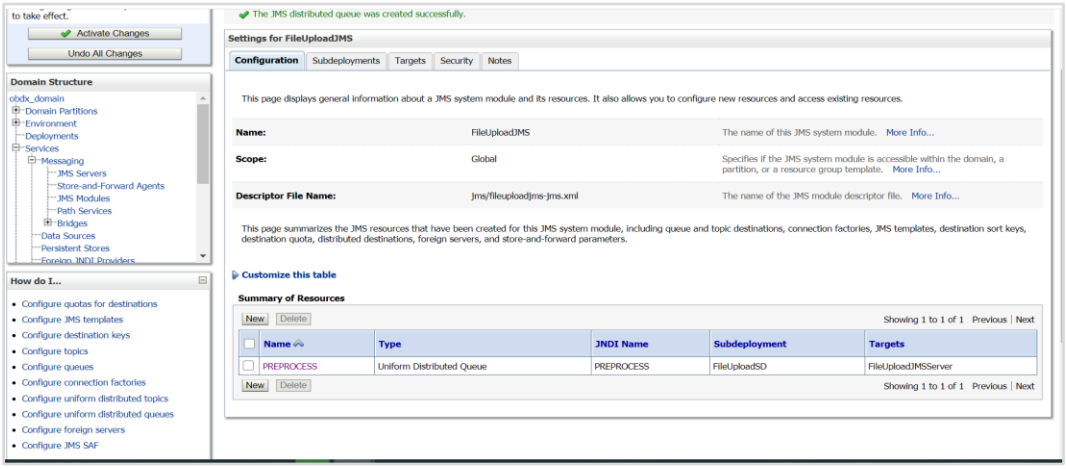


35. Select WLS_JMS_FILEUPLOAD_PS and click on Next

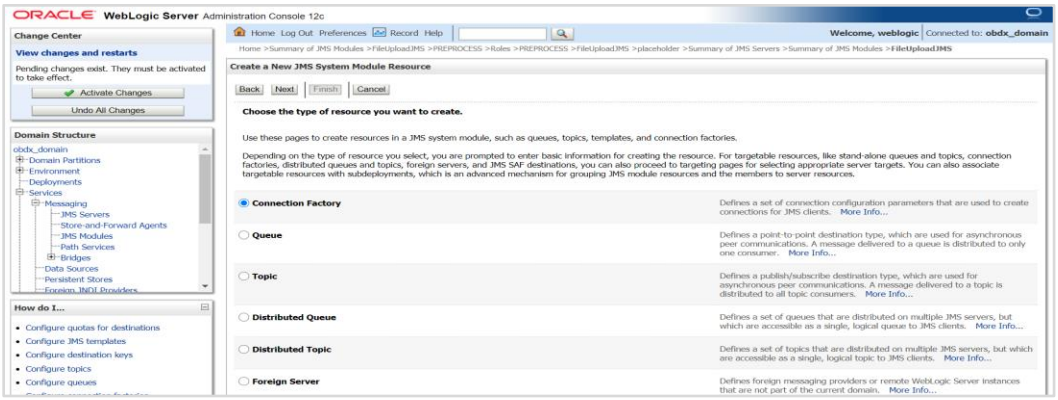
36. Select Create a New Subdeploymeny and create FileUploadSD



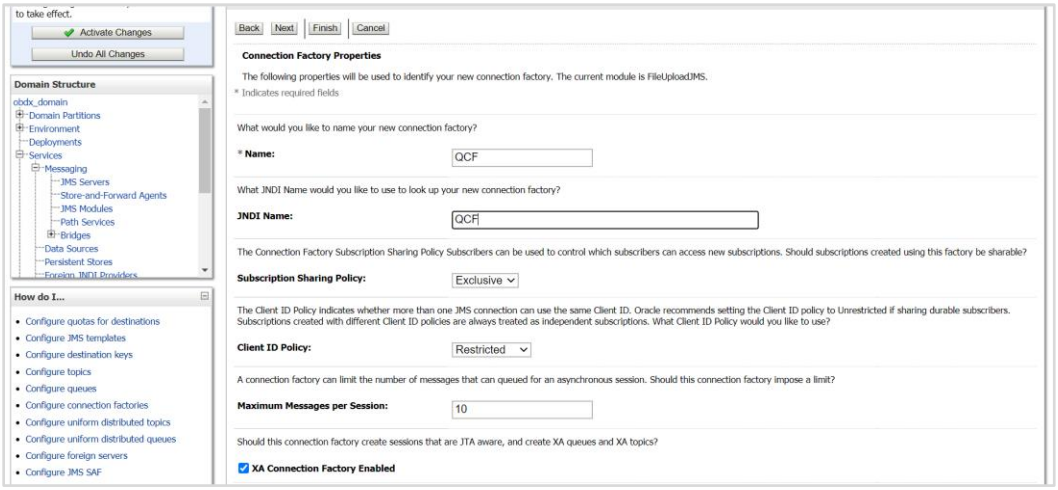
37. Select FileUploadJMSServer and click on Finish



38. Similarly Go into FileuploadJMS module and click on Next



39. Select Connection factory –> Click Next



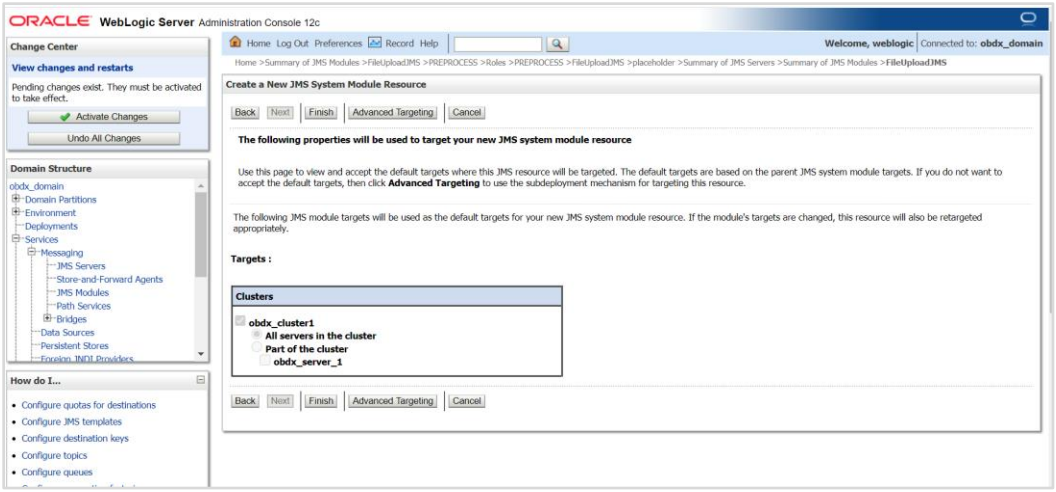
40. Provide

Name :- OCF

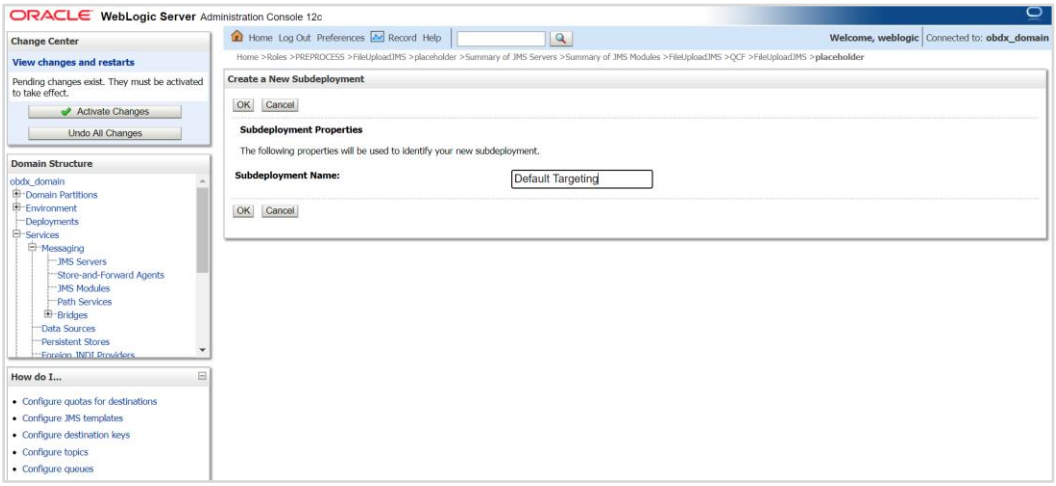
JNDI Name :- OCF

Subscription Sharing Policy :- Exclusive

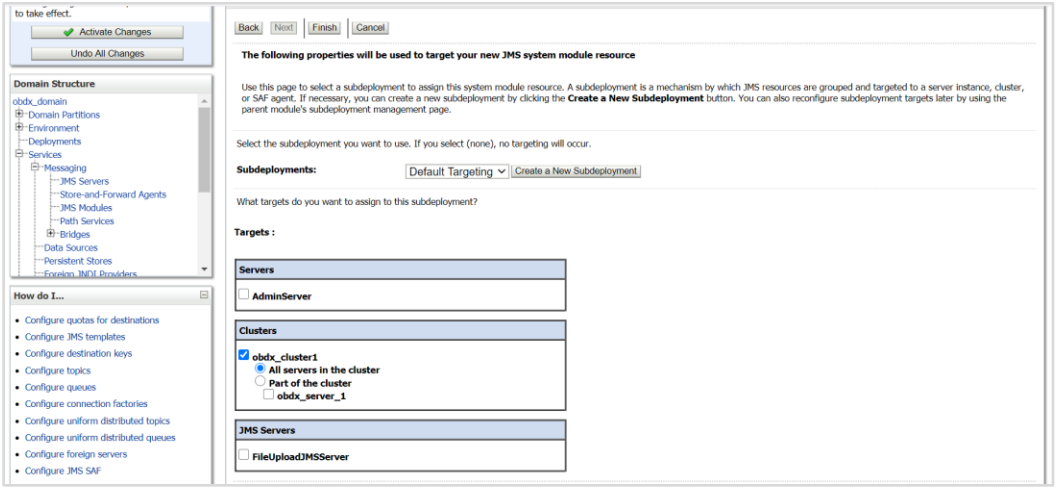
Client ID Policy :- Restricted



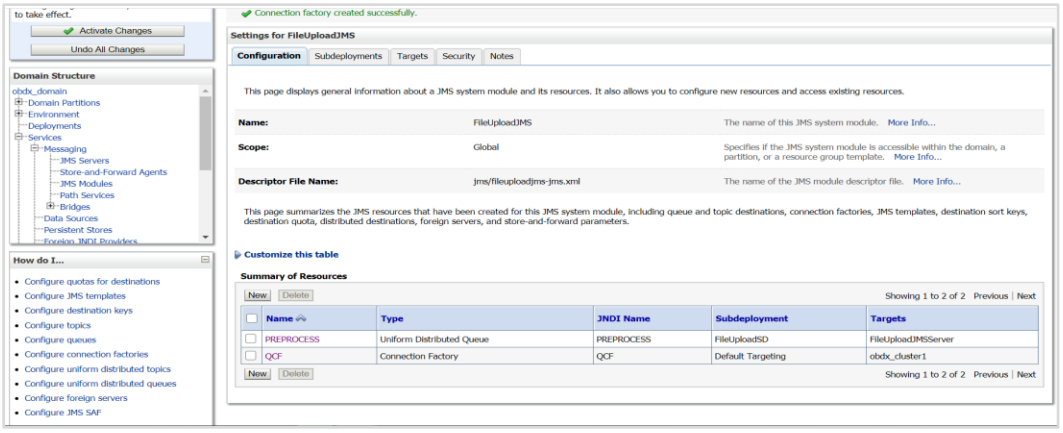
41. Click on Advanced targeting



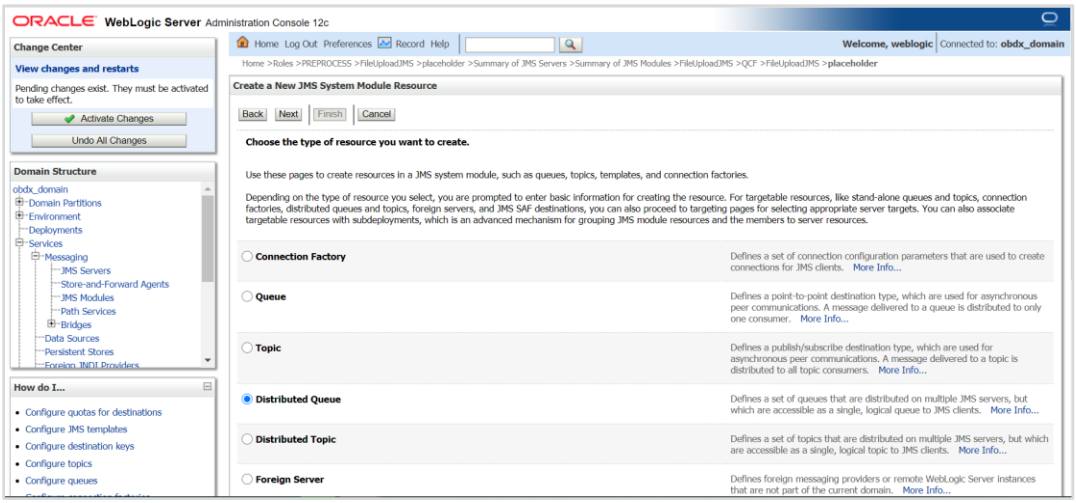
42. Provide Subdeployment Name as Default Targeting



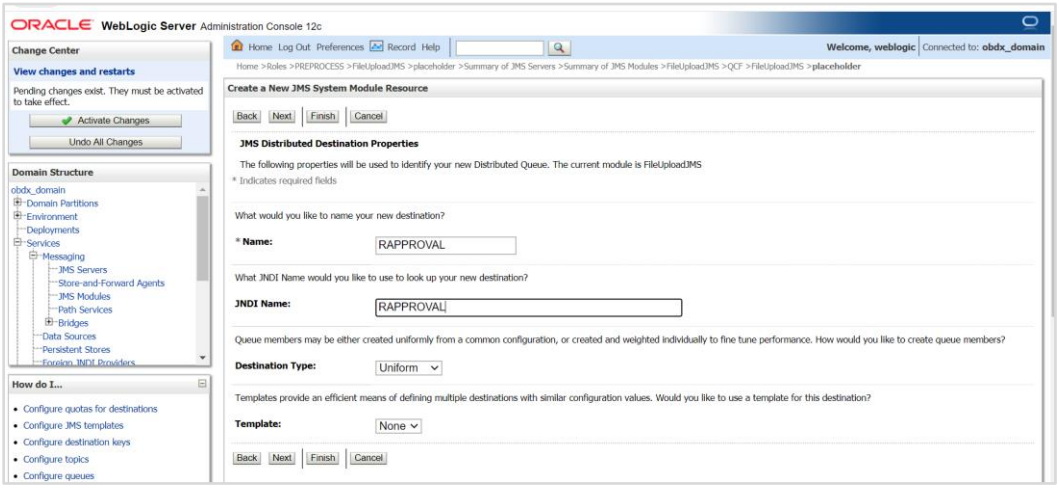
43. Select cluster and click on Finish



44. Go to FileUpload JMS click on New



45. Select Distributed Queue



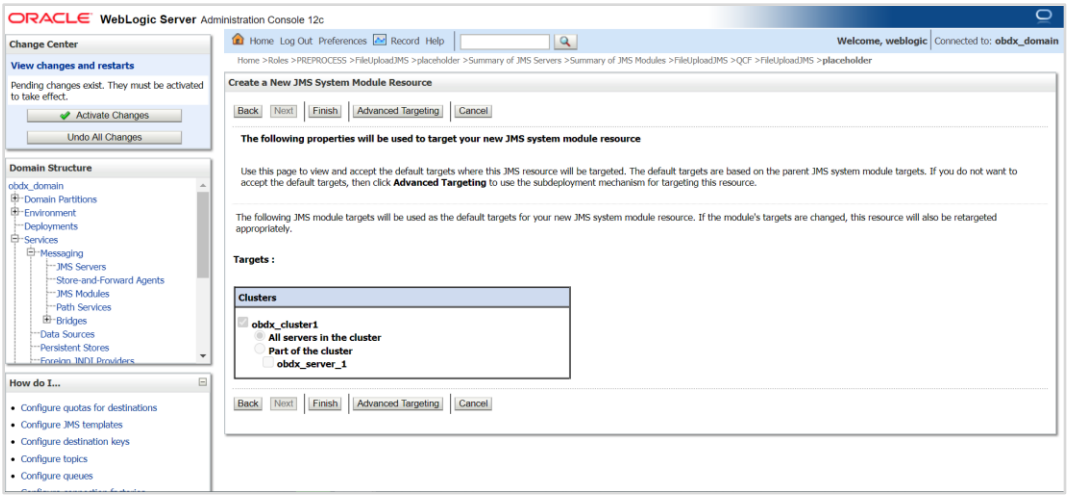
46. Provide

Name :- RAPPROVAL

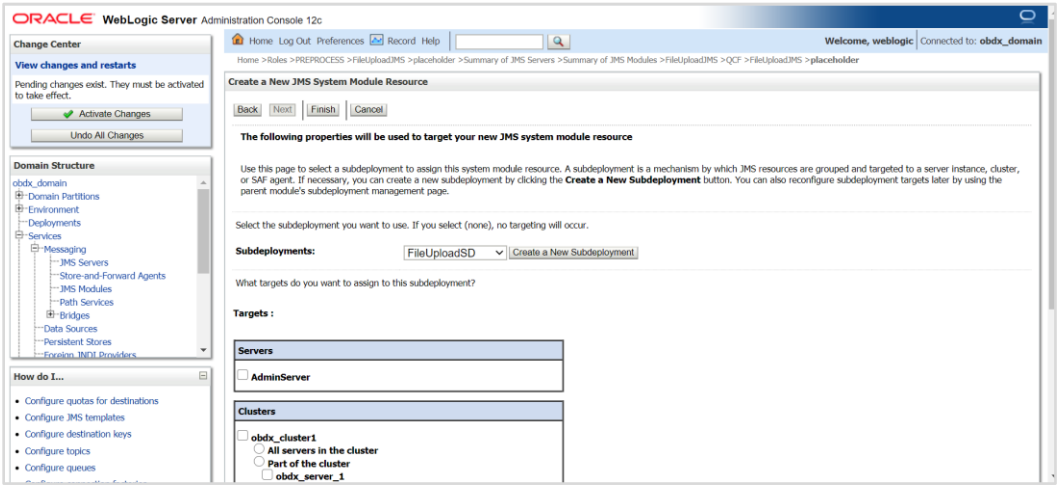
JNDI Name :- RAPPROVAL

Destination Type:- Uniform

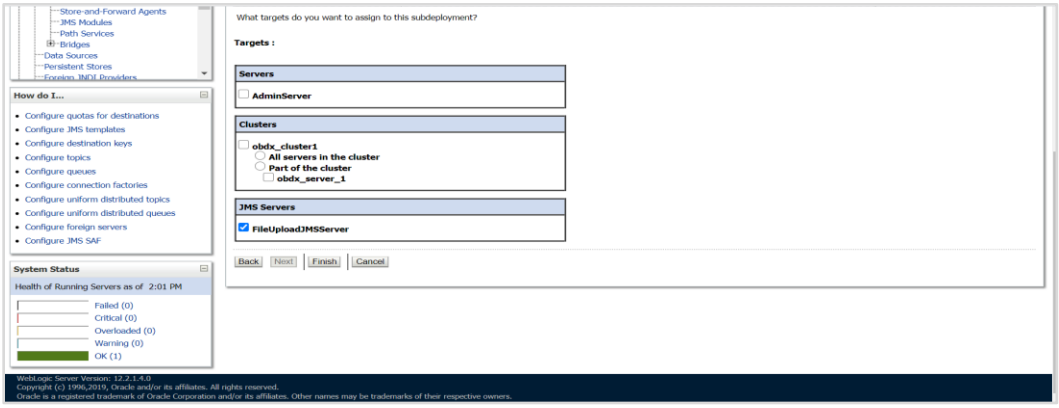
Template :- None



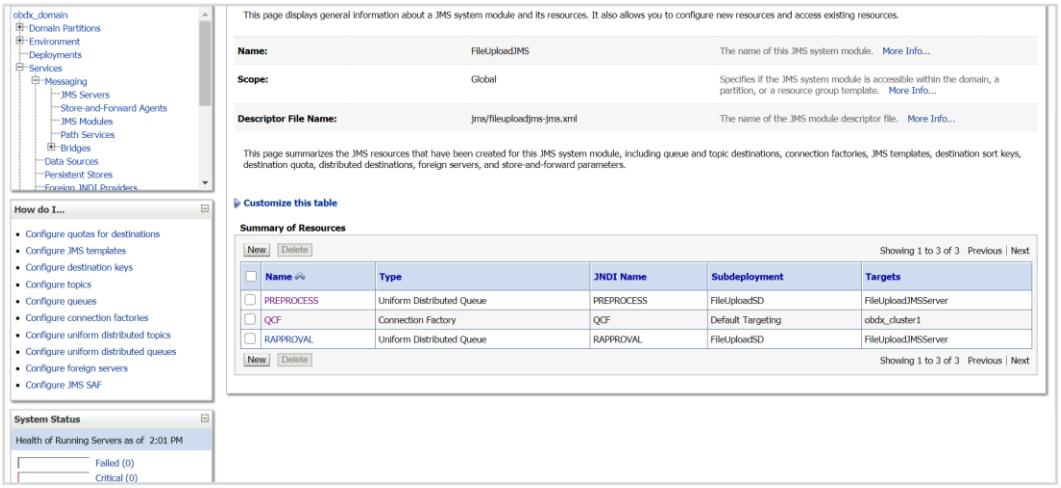
47. Select Advance targeting



48. Select Subdeployment :- FileUploadSD



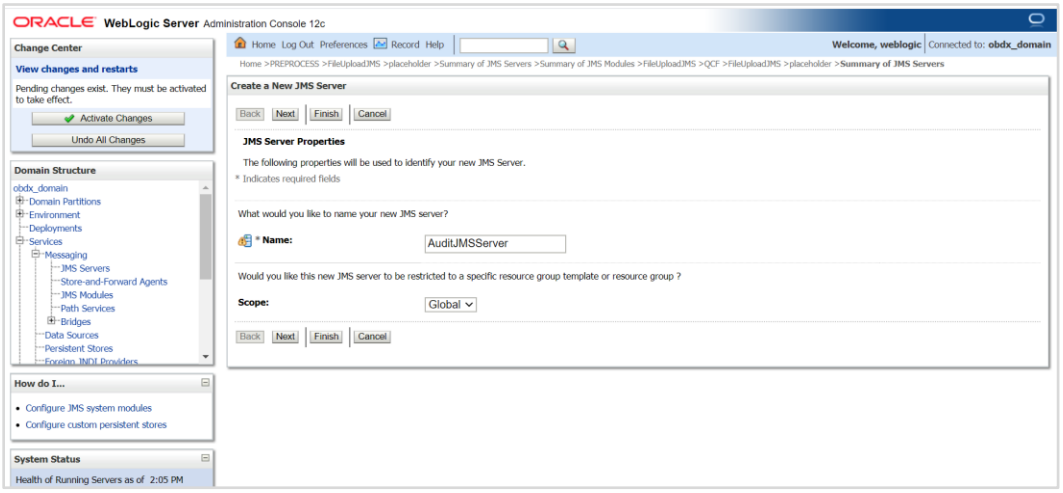
49. Select FileUploadJMSServer and click on Finish



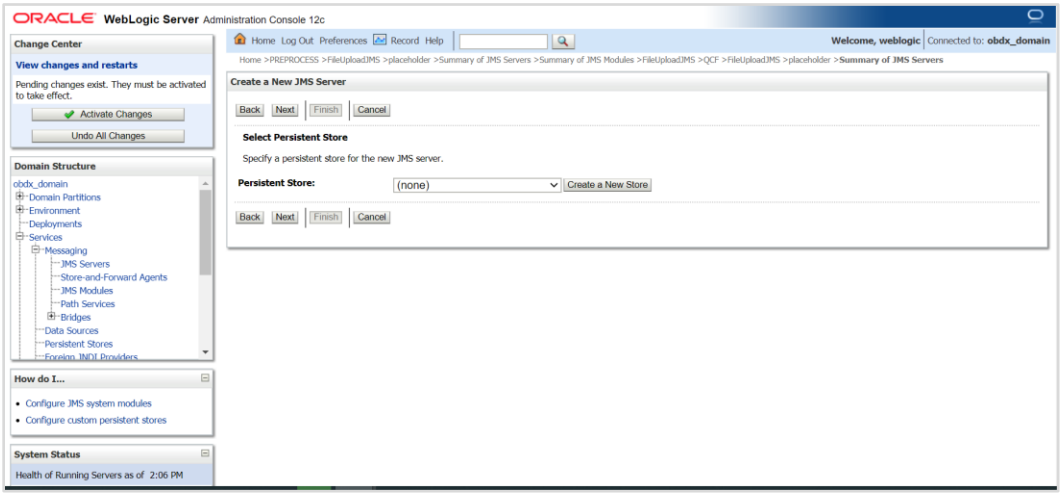
3.8 Creating WLS_JMS_AUDIT_PS FileStore

3.9 Creating AuditJMSServer JMS Server

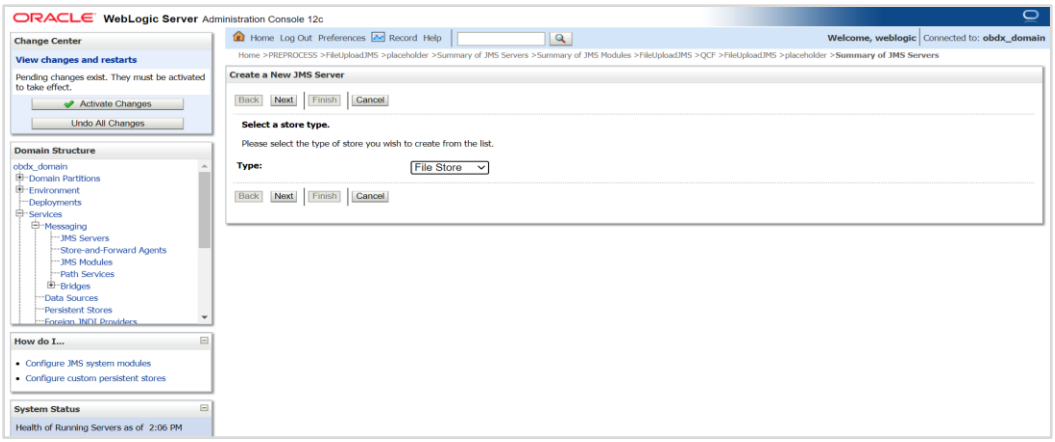
3.10 Creating WLS_JMS_REPORT_PS FileStore



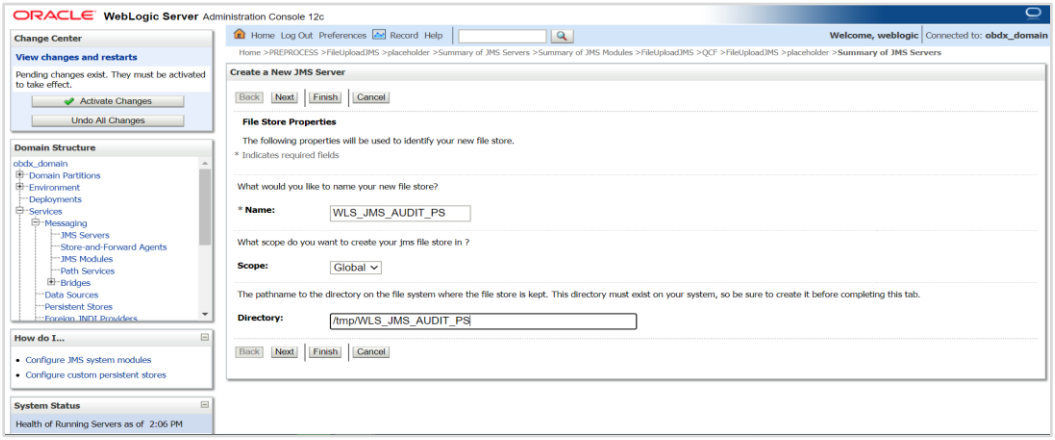
1. Click on JMS server and click on New
2. Provide Name as AuditJMSServer , Scope as Global



3. Click on Create a New Store



4. Select File Store

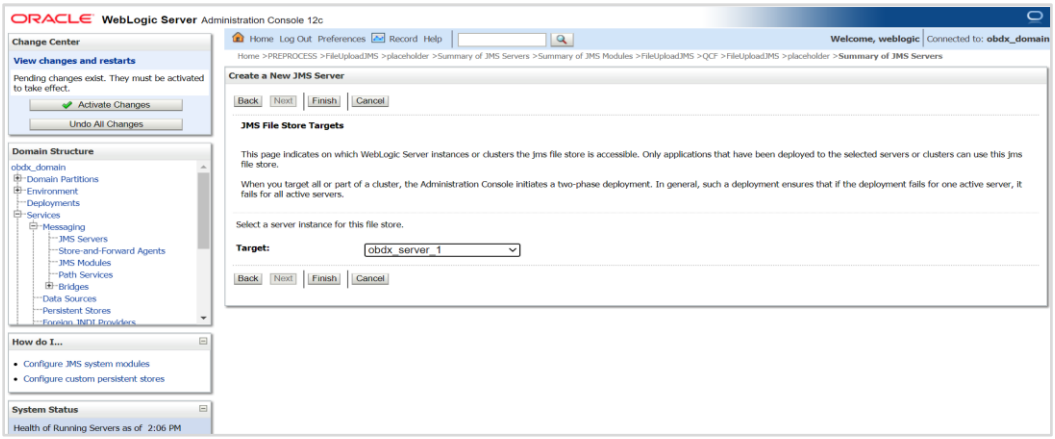


5. Provide

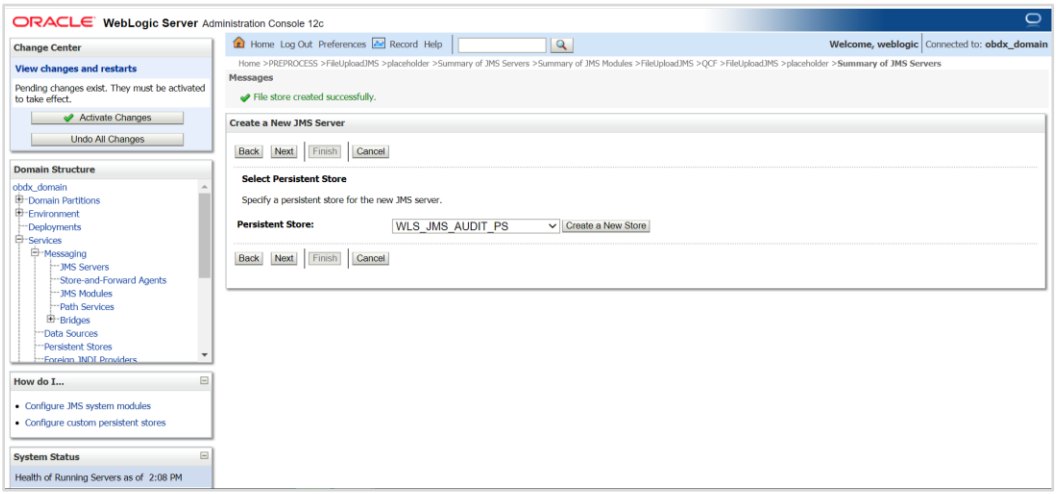
Name :- WLS_JMS_AUDIT_PS

Scope :- Global

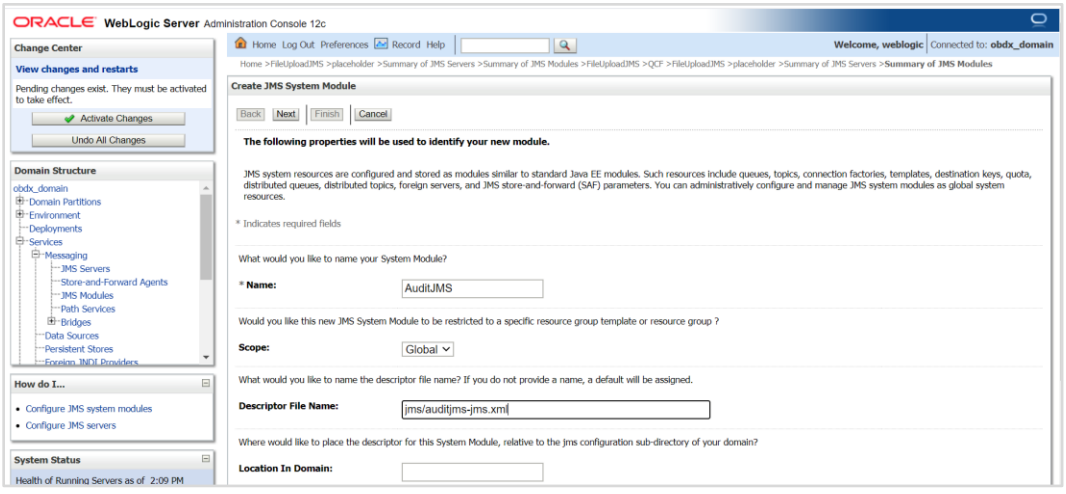
Directory :- /tmp/WLS_JMS_AUDIT_PS



6. Select Target as managed server and click on Finish



7. Select the new store created WLS_JMS_AUDIT_PS and click on Next

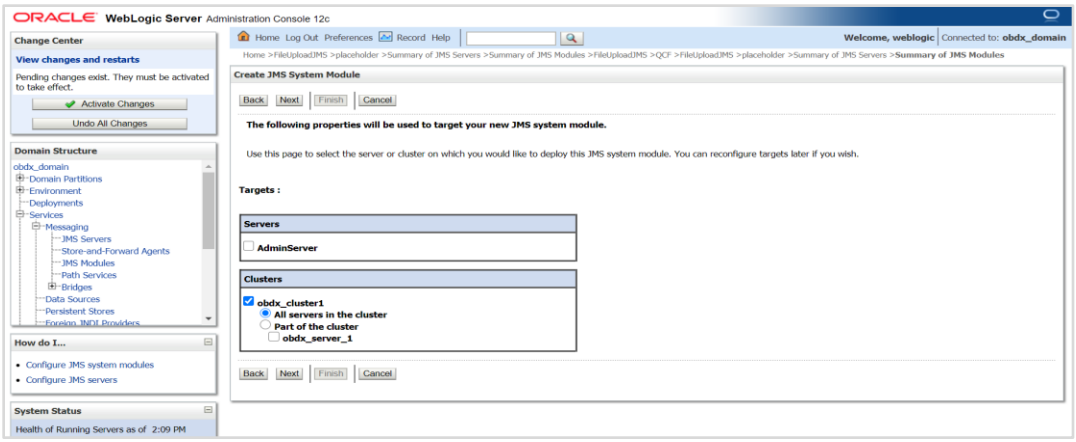


8. **Provide**

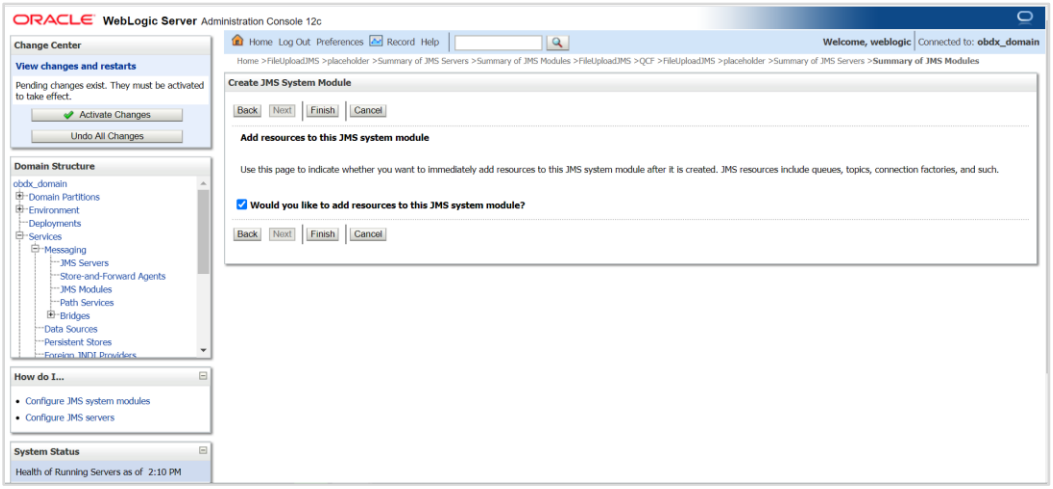
   Name :- AuditJMS

   Scope :- Global

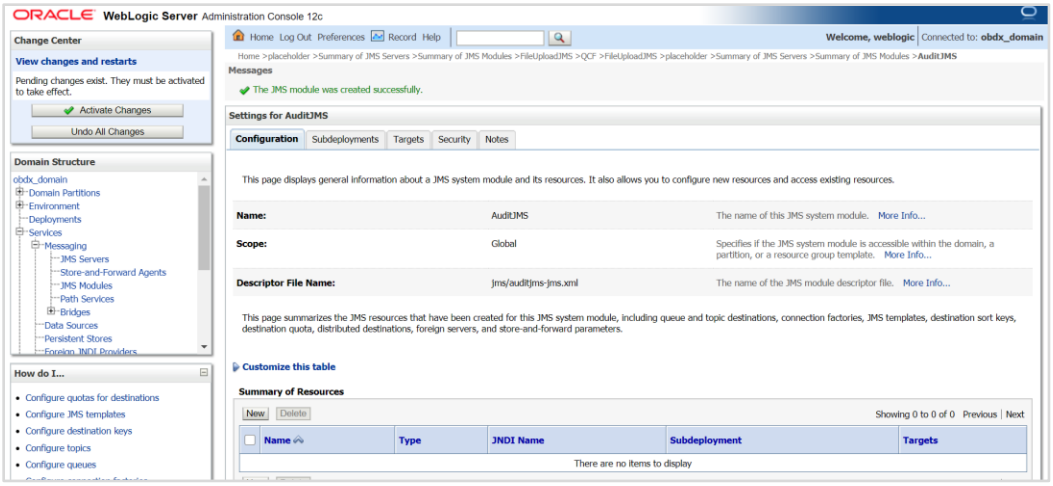
   Descriptor File Name:- jms/auditjms-jms.xml



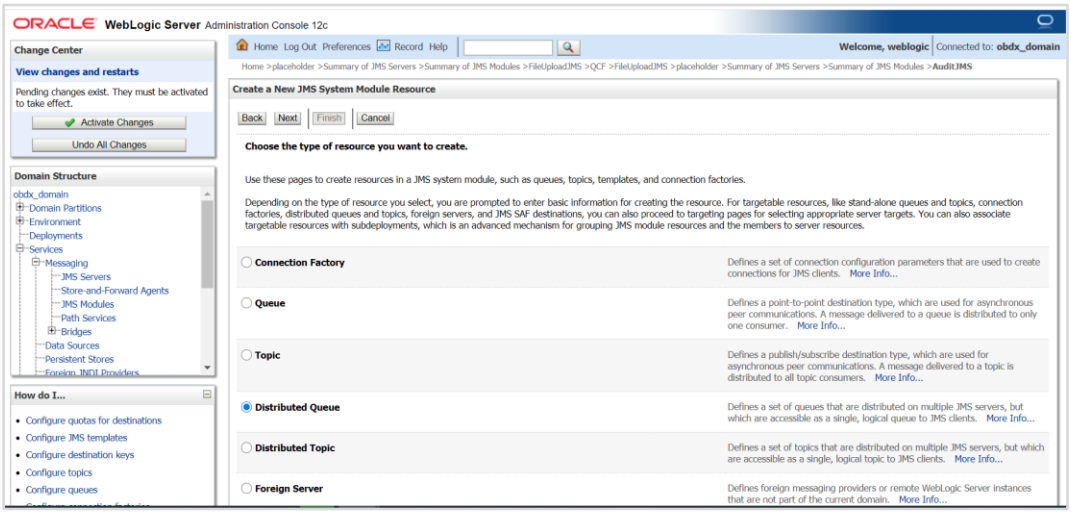
9. Select Cluster as a target



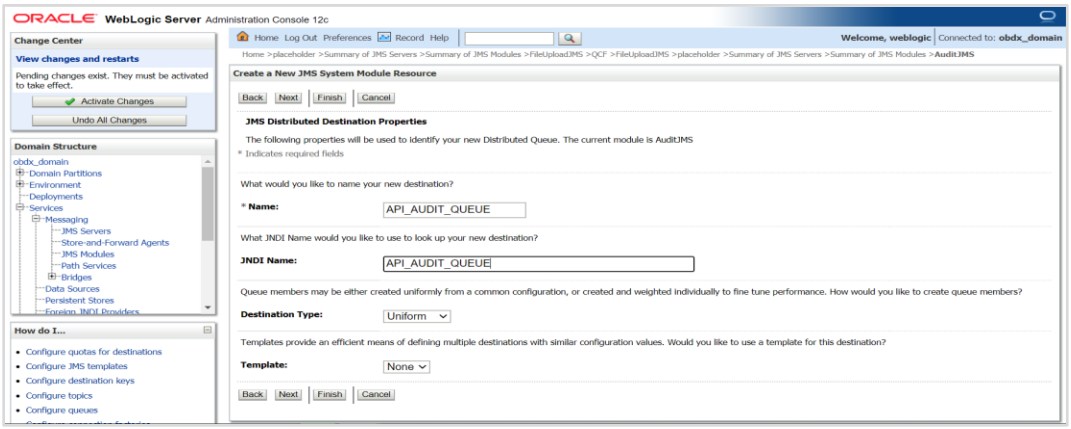
10. Select would you like to add resource to this JMS system module?



11. Click on new



12. Select Distributed Queue

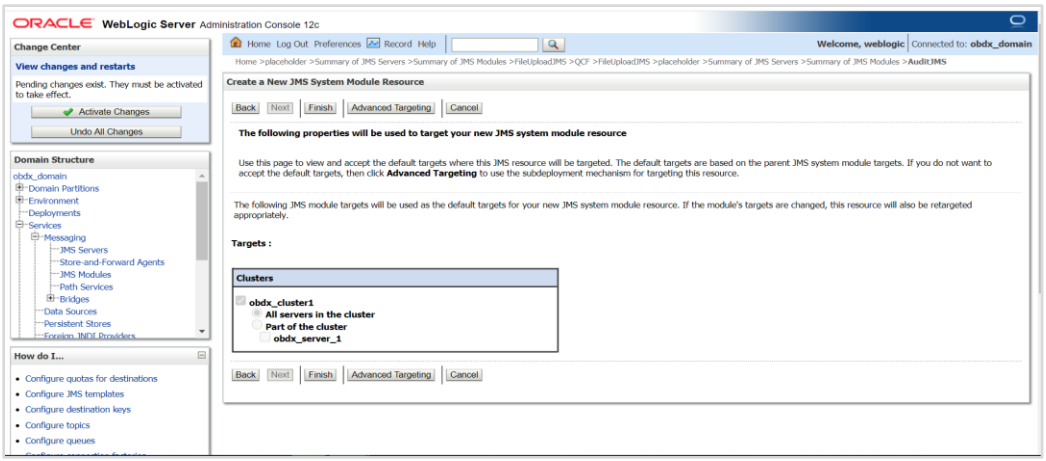


13. **Name:-** API_AUDIT_QUEUE

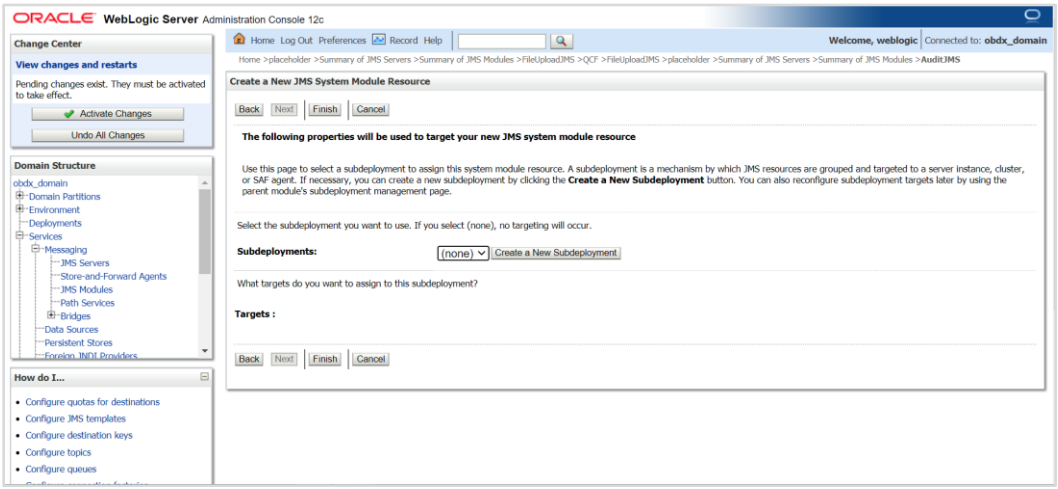
    JNDI Name:- API_AUDIT_QUEUE

    Destination Type :- Uniform

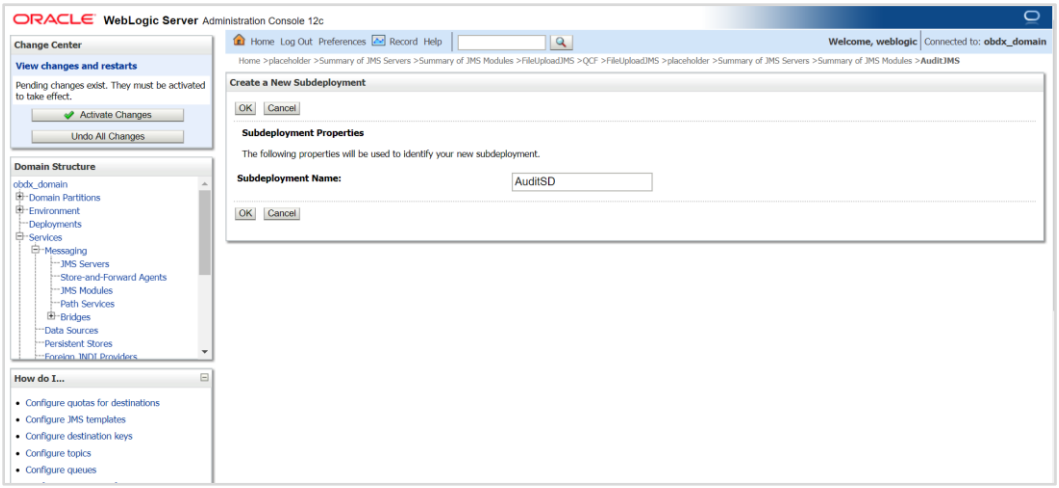
    Template:- None



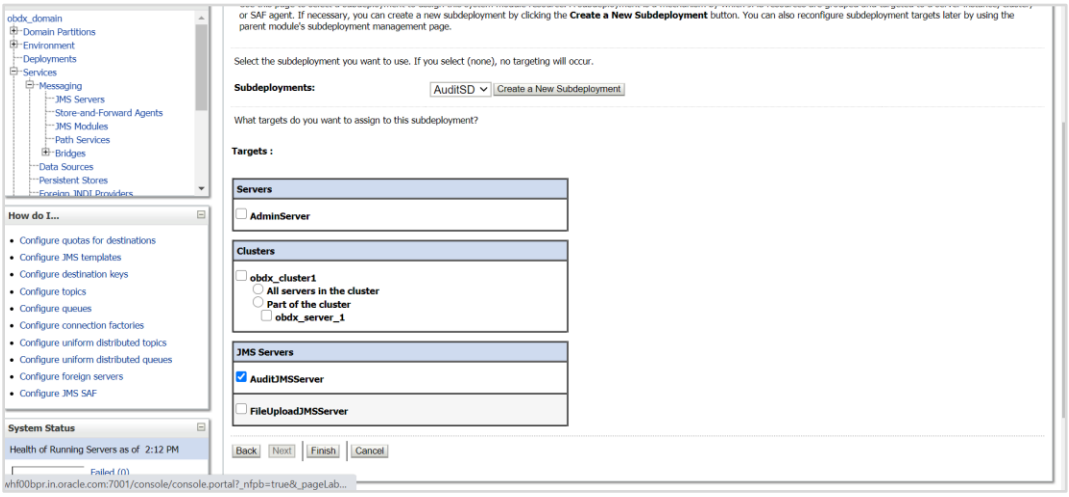
14. Select Advance targeting



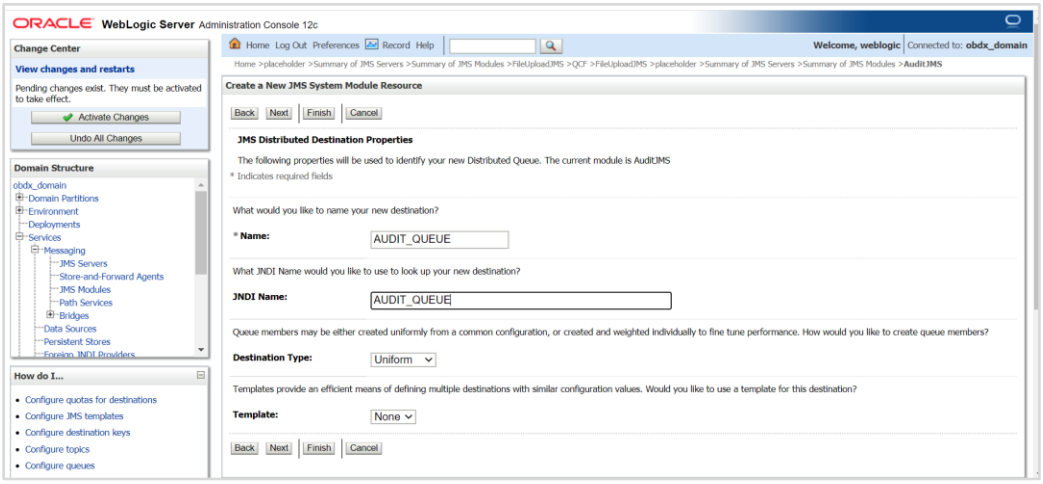
15. Click on Create a New Subdeployment

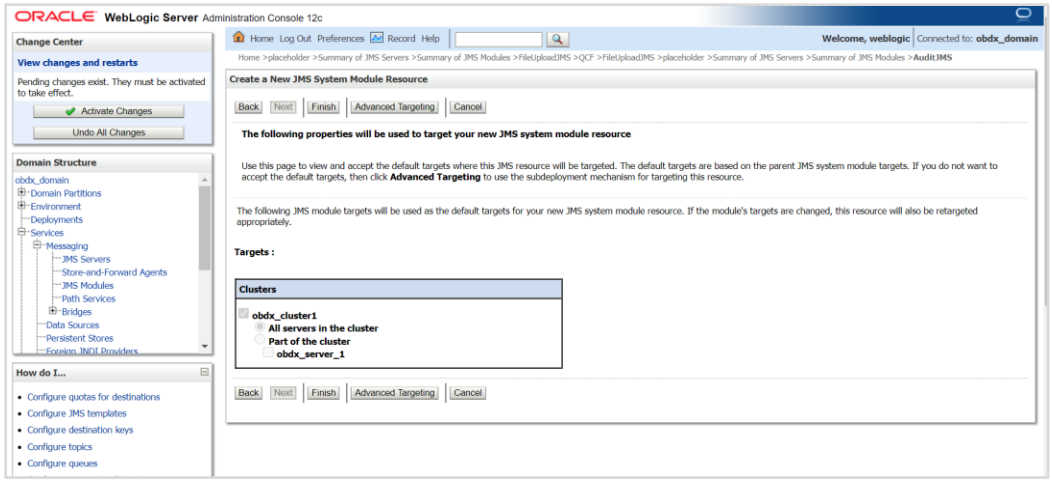


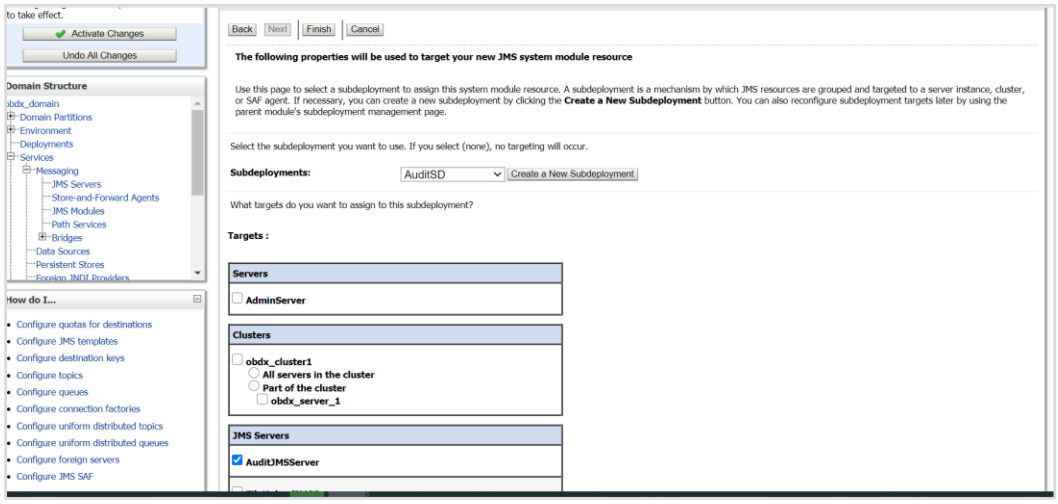
16. Provide Subdeployment Name as AuditSD

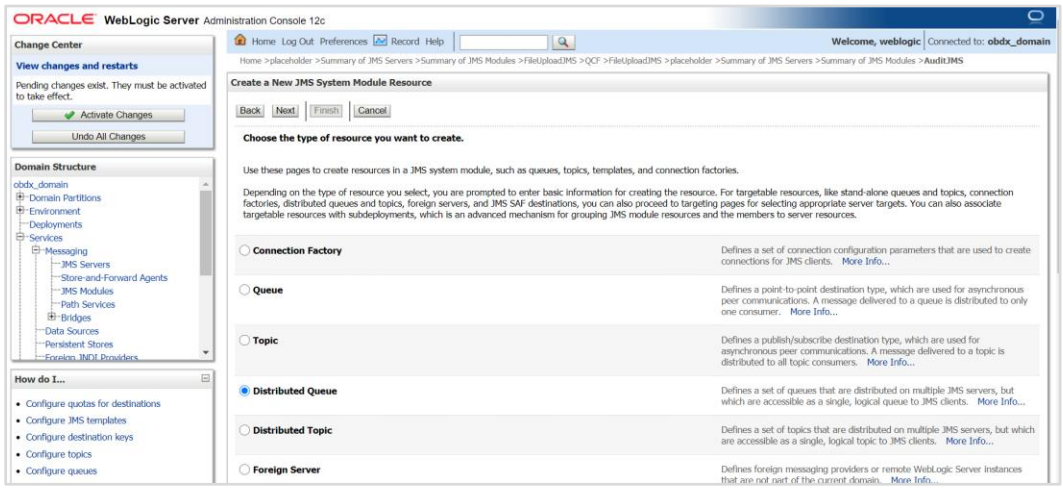


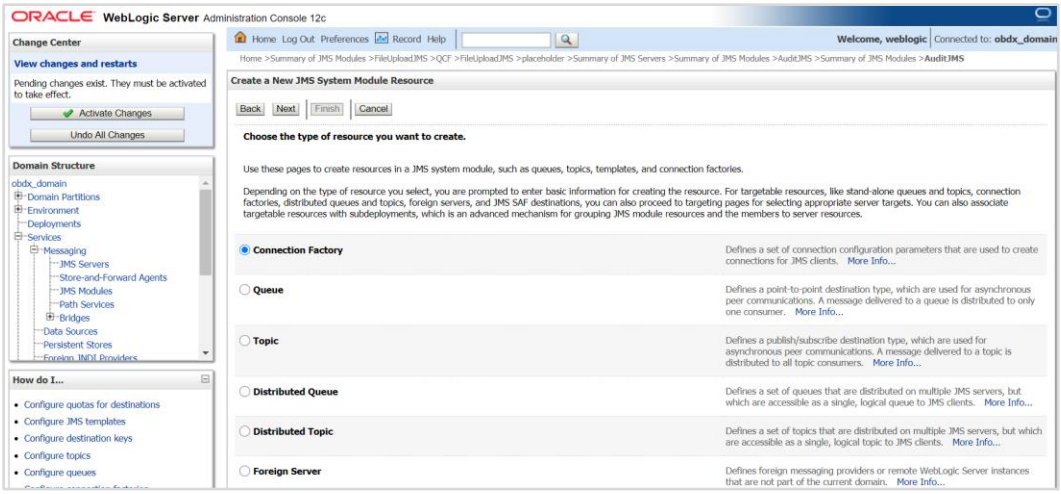
17. Select Target as AuditJMSServer



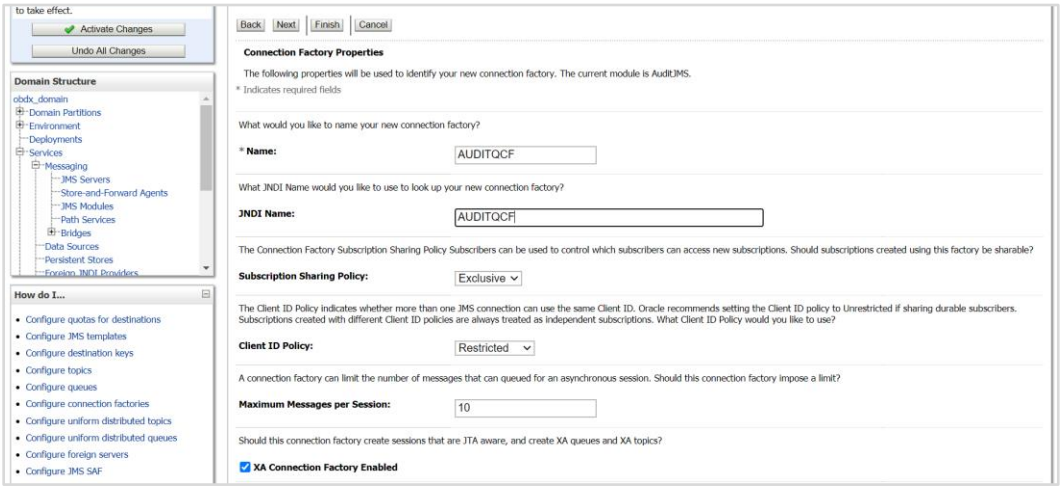

ORACLE WebLogic Server Administration Console 12c
Change Center
View changes and restarts
Pending changes exist. They must be activated to take effect.
Activate Changes
Undo All Changes
Domain Structure
obdx_domain
Domain Partitions
Environment
Deployments
Services
Messaging
JMS Servers
Store-and-Forward 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
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: obdx_domain
Home >placeholder >Summary of JMS Servers >Summary of JMS Modules >FileUploadJMS >QCF >FileUploadJMS >placeholder >Summary of JMS Servers >Summary of JMS Modules >AuditJMS
Create a New JMS System Module Resource
Back Next Finish Advanced Targeting Cancel
The following properties will be used to target your new JMS system module resource
Use this page to view and accept the default targets where this JMS resource will be targeted. The default targets are based on the parent JMS system module targets. If you do not want to accept the default targets, then click Advanced Targeting to use the subdeployment mechanism for targeting this resource.
The following JMS module targets will be used as the default targets for your new JMS system module resource. If the module's targets are changed, this resource will also be retargeted appropriately.
Targets :
Clusters
obdx_cluster1
All servers in the cluster
Part of the cluster
obdx_server_1
Back Next Finish Advanced Targeting Cancel


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
Back Next Finish Cancel
The following properties will be used to target your new JMS system module resource
Use this page to select a subdeployment to assign this system module resource. A subdeployment is a mechanism by which JMS resources are grouped and targeted to a server instance, cluster, or SAF agent. If necessary, you can create a new subdeployment by clicking the Create a New Subdeployment button. You can also reconfigure subdeployment targets later by using the parent module's subdeployment management page.
Select the subdeployment you want to use. If you select (none), no targeting will occur.
Subdeployments: AuditSD Create a New Subdeployment
What targets do you want to assign to this subdeployment?
Targets :
Servers
AdminServer
Clusters
obdx_cluster1
All servers in the cluster
Part of the cluster
obdx_server_1
JMS Servers
AuditJMSServer


ORACLE WebLogic Server Administration Console 12c
Change Center
View changes and restarts
Pending changes exist. They must be activated 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
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: obdx_domain
Home >placeholder >Summary of JMS Servers >Summary of JMS Modules >FileUploadJMS >QCF >FileUploadJMS >placeholder >Summary of JMS Servers >Summary of JMS Modules >AuditJMS
Create a New JMS System Module Resource
Back Next Finish Cancel
Choose the type of resource you want to create.
Use these pages to create resources in a JMS system module, such as queues, topics, templates, and connection factories.
Depending on the type of resource you select, you are prompted to enter basic information for creating the resource. For targetable resources, like stand-alone queues and topics, connection factories, distributed queues and topics, foreign servers, and JMS SAF destinations, you can also proceed to targeting pages for selecting appropriate server targets. You can also associate targetable resources with subdeployments, which is an advanced mechanism for grouping JMS module resources and the members to server resources.
Connection Factory
Defines a set of connection configuration parameters that are used to create connections for JMS clients. More Info...
Queue
Defines a point-to-point destination type, which are used for asynchronous peer communications. A message delivered to a queue is distributed to only one consumer. More Info...
Topic
Defines a publish/subscribe destination type, which are used for asynchronous peer communications. A message delivered to a topic is distributed to all topic consumers. More Info...
Distributed Queue
Defines a set of queues that are distributed on multiple JMS servers, but which are accessible as a single, logical queue to JMS clients. More Info...
Distributed Topic
Defines a set of topics that are distributed on multiple JMS servers, but which are accessible as a single, logical topic to JMS clients. More Info...
Foreign Server
Defines foreign messaging providers or remote WebLogic Server instances that are not part of the current domain. More Info...



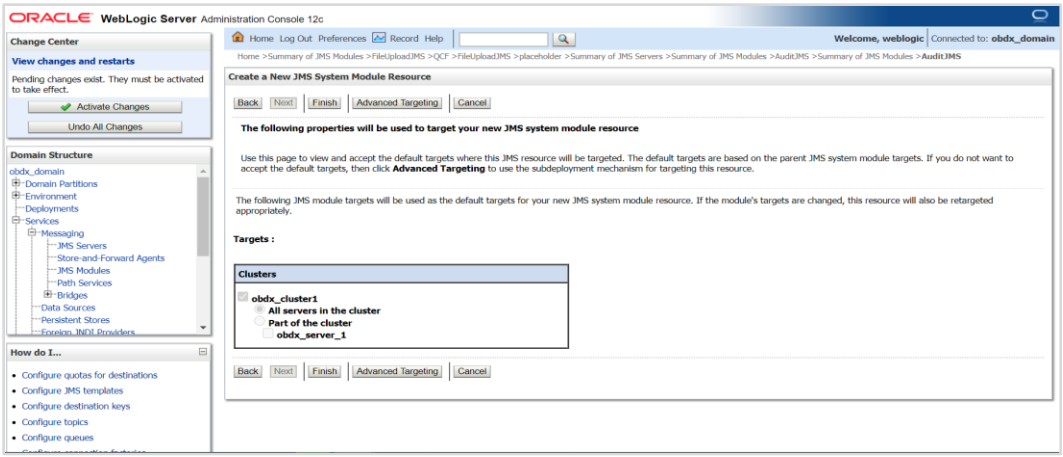
18. Click on connection Factory



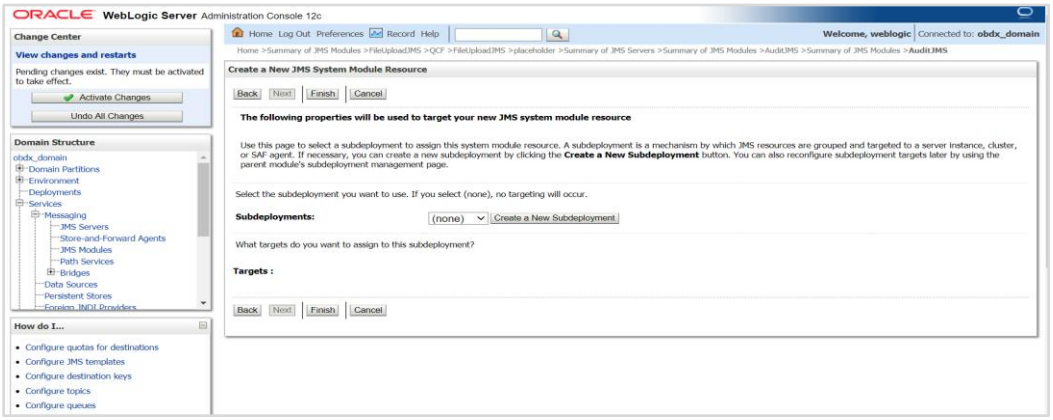
19. Provide

Name :- AUDITQCF

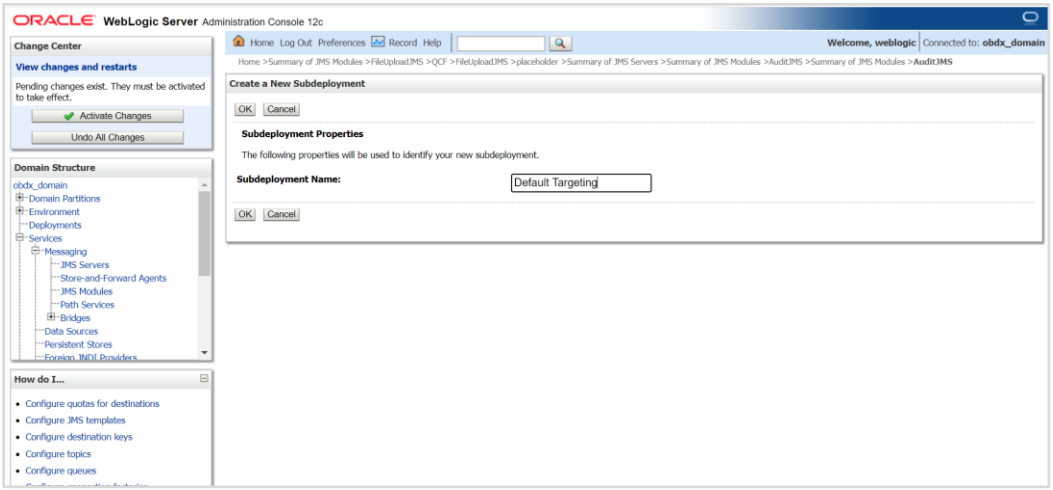
JNDI Name :- AUDITQCF



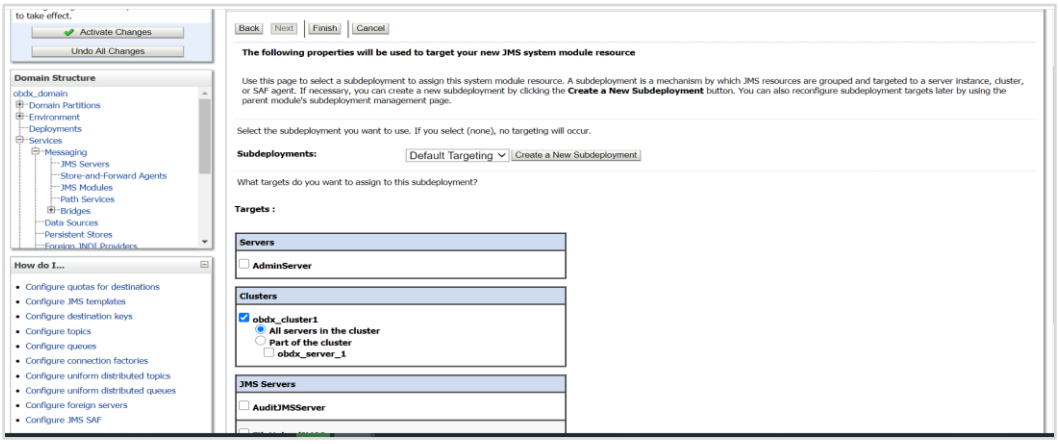
20. Click on Advanced Targeting



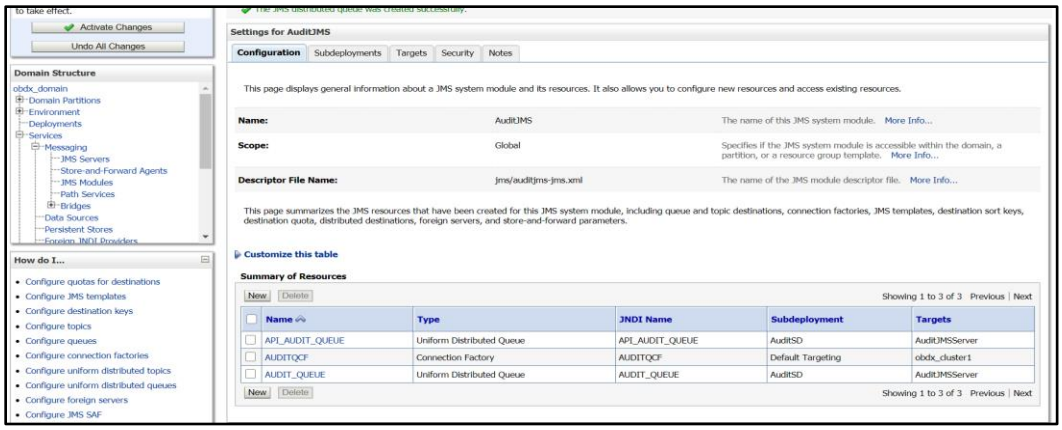
21. Click on Create a New Subdeployment



22. Give Subdeployment Name as Default Targeting

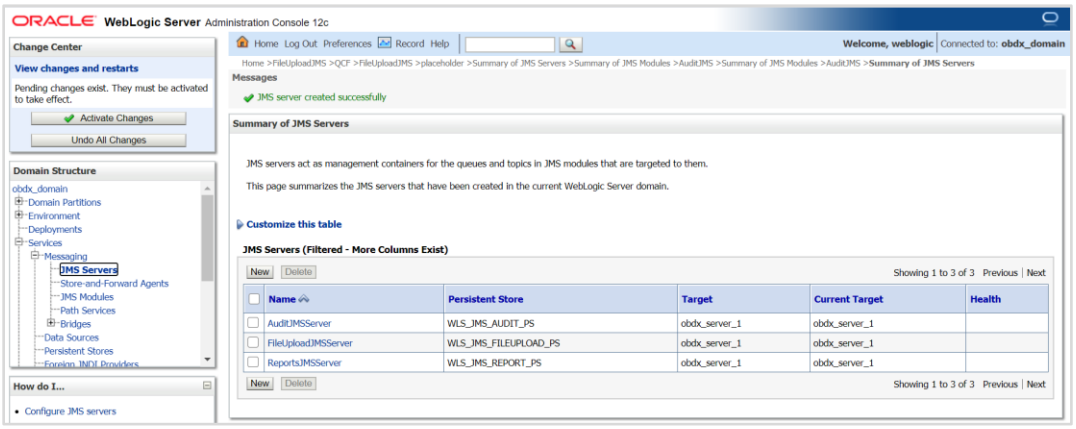


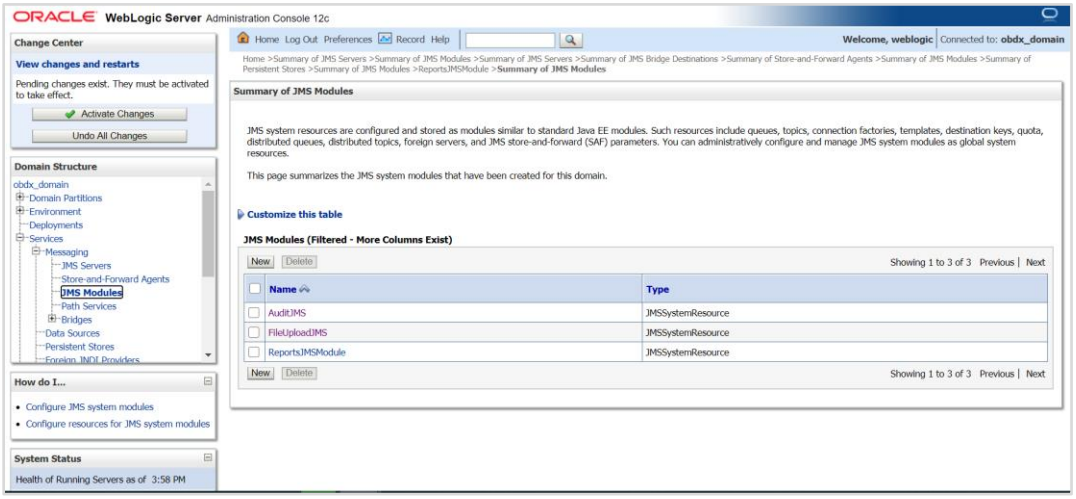
23. Under AuditJMS module Create Uniform Distrubuted Queue and connection Factory as show below in the screen shot

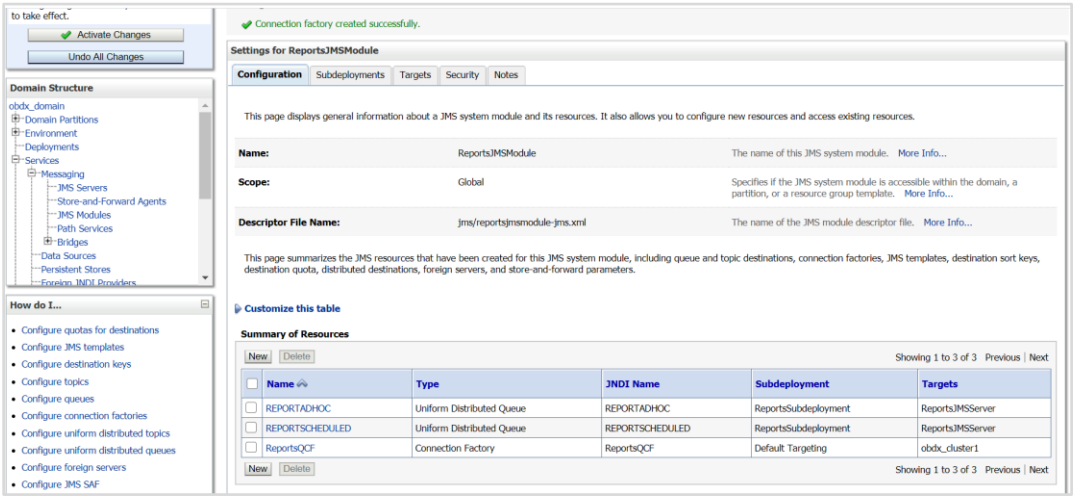


3.11 Creating ReportsJMSServer JMS Server

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







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

   REPORTADHOC – Uniform Distributed Queue

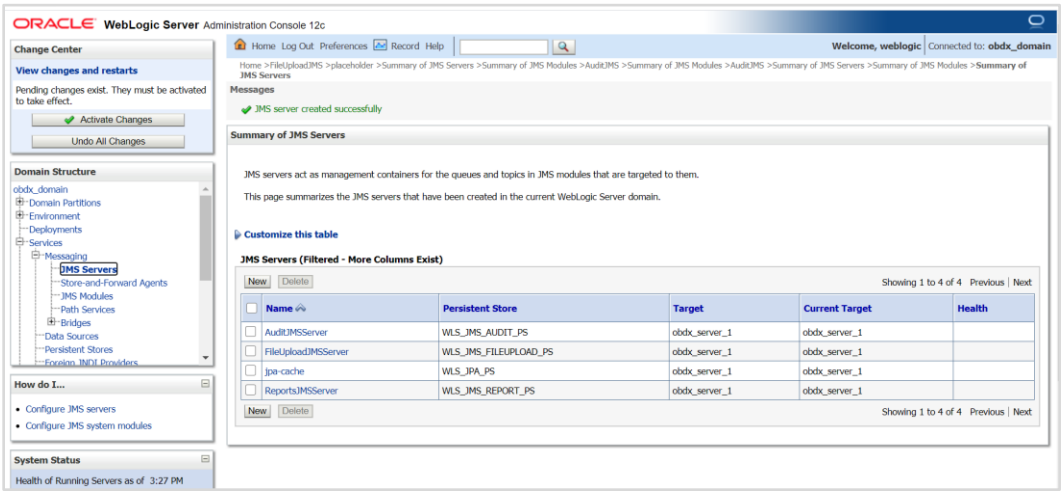
   REPORTSCHEDULED -- Uniform Distributed Queue

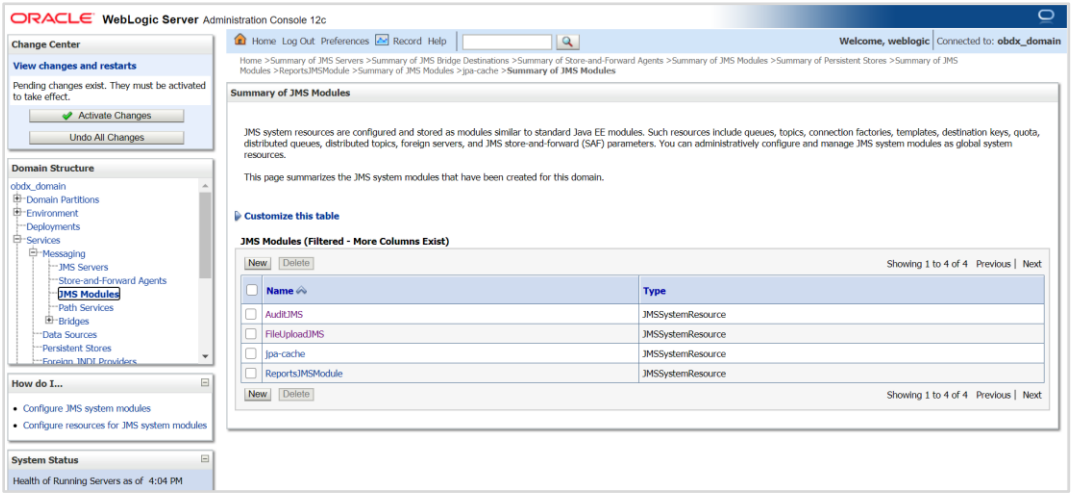
   ReportsQCF – Connection Factory

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

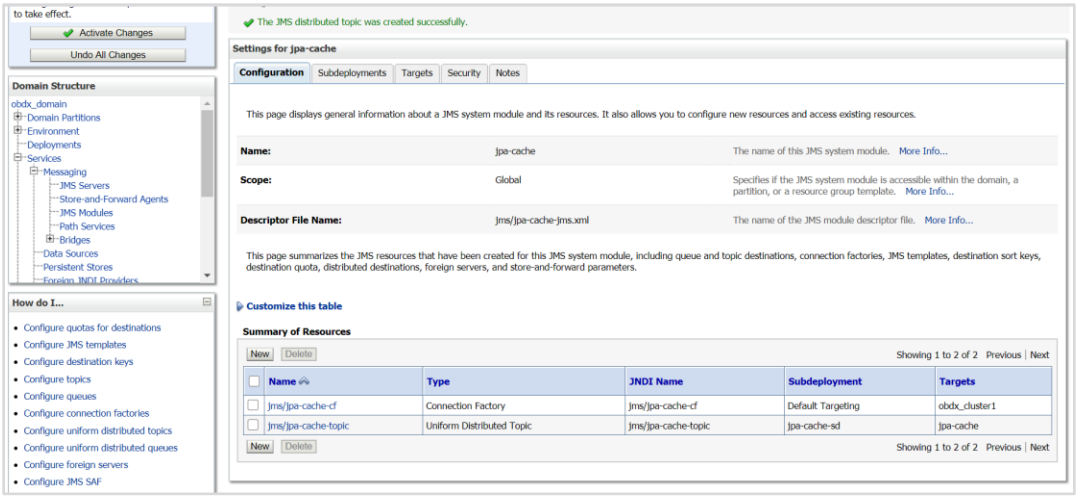




1. Under jpa-cache JMS Module create connection Factory and Uniform Distrbuted topic as shown in below screen shot

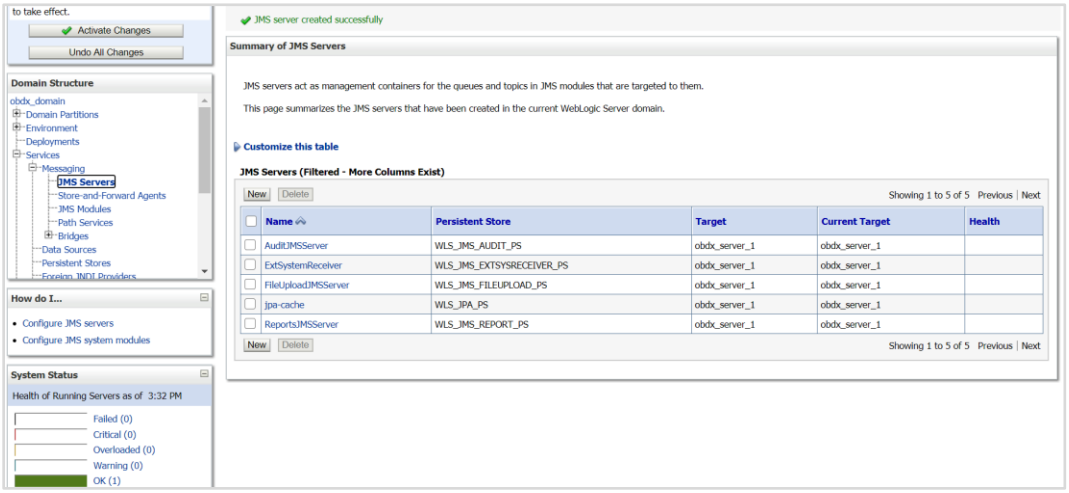
   Jms/jpa-cache-cf --- Connection Factory

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

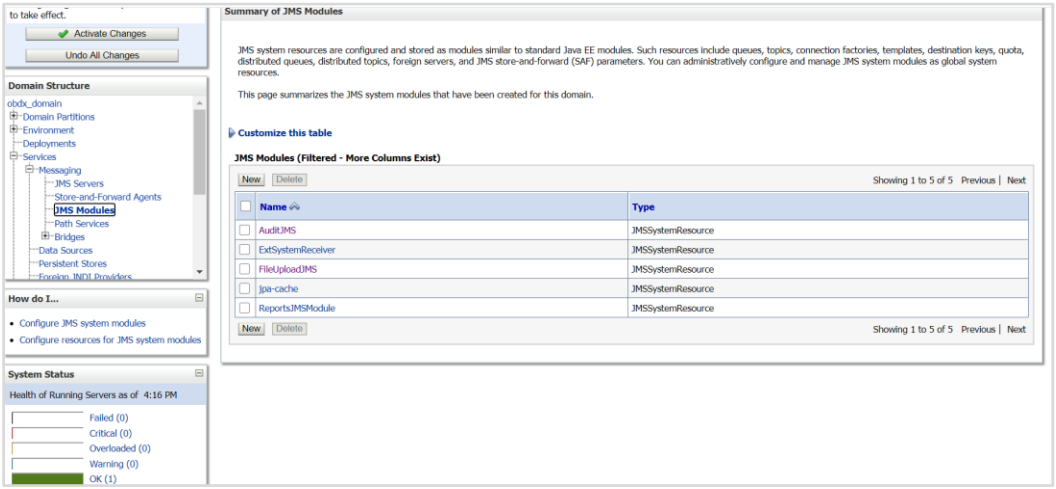

to take effect.
Activate Changes
Undo All Changes
Domain Structure
obdx_domain
Domain Partitions
Environment
Deployments
Services
Messaging
JMS Servers
Store-and-Forward Agents
JMS Modules
Path Services
Bridges
Data Sources
Persistent Stores
Foreign JNDI Providers
How do I...
Configure quotas for destinations
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
Name
Type
JNDI Name
Subdeployment
Targets
jms/jpa-cache-cf
Connection Factory
jms/jpa-cache-cf
Default Targeting
obdx_cluster1
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

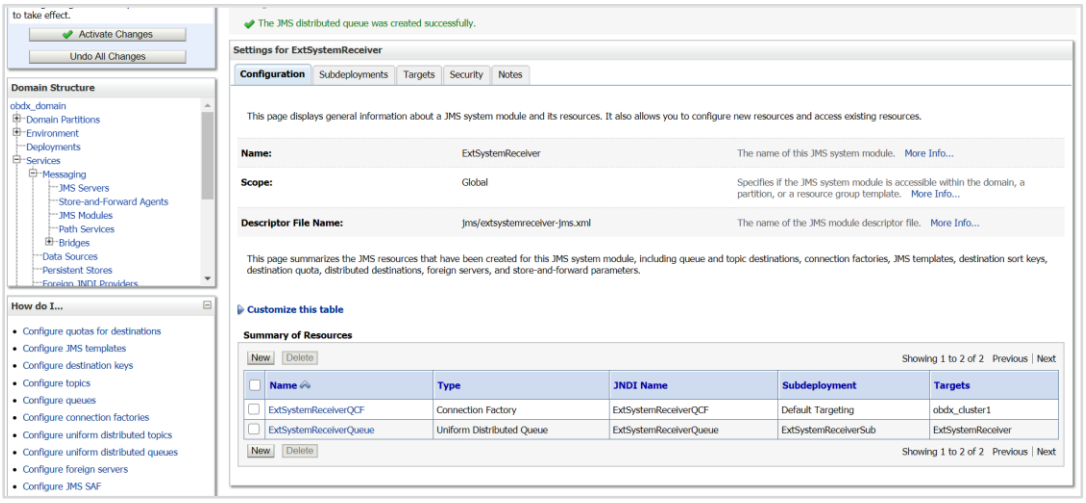
1. Create ExtSystemReceiver JMS Server Persistent store file store as WLS_JMS_EXTSYSRECEIVER_PS as show in below screen shot.



2. Create ExtSystemReceiver JMS Module as below

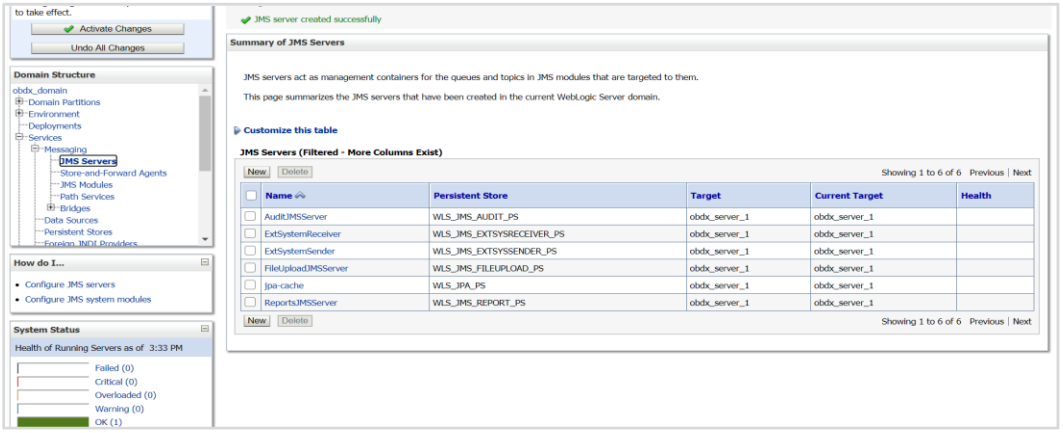


3. Create ExtSystemReceiverQCF – connection Factory and ExtSystemReceiverQueue – uniform Distributed Queue in ExtSystemReceiver JMS Module refer below screen shot

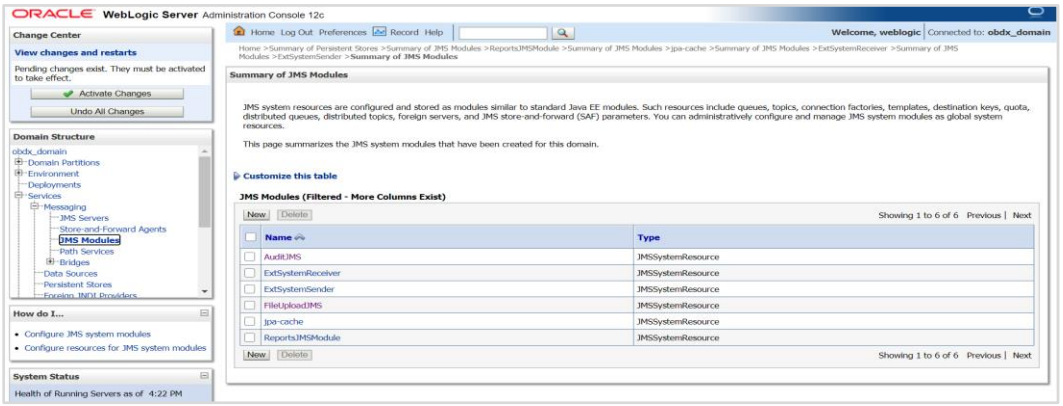


3.15 Creating ExtSystemSender JMS Server Persistent Store FileStore as WLS_JMS_EXTSYSSENDER_PS

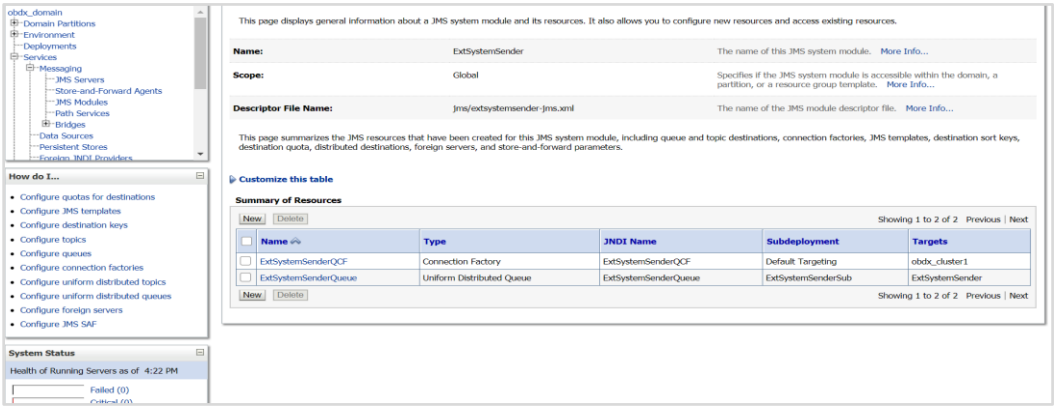
As show below create JMS Server ExtSystemSender



1. Create ExtSystemSender JMS Module

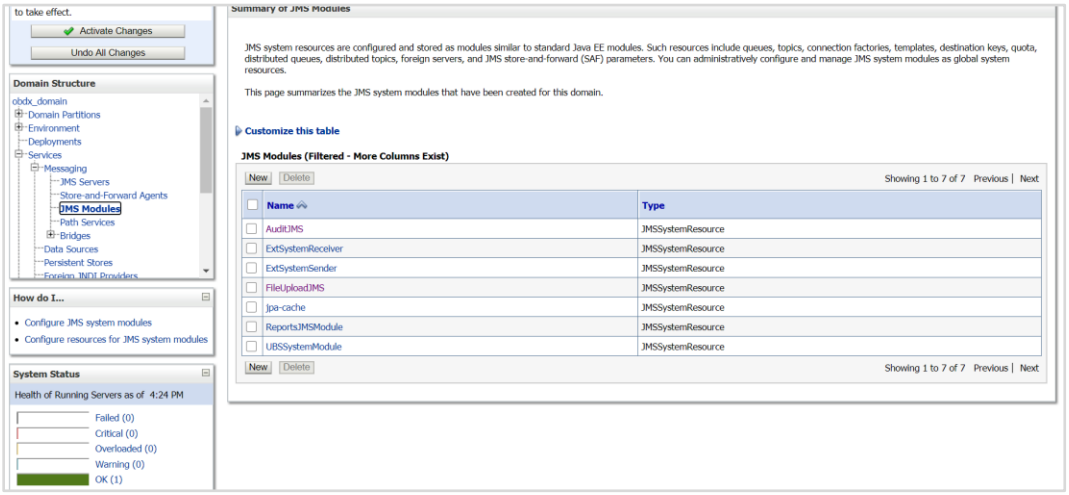


2. Under ExtSystemSender JMS Module create ExtSystemSenderQCF – connection Factory and ExtSystemSenderQueue – Uniform Distributed Queue as show below

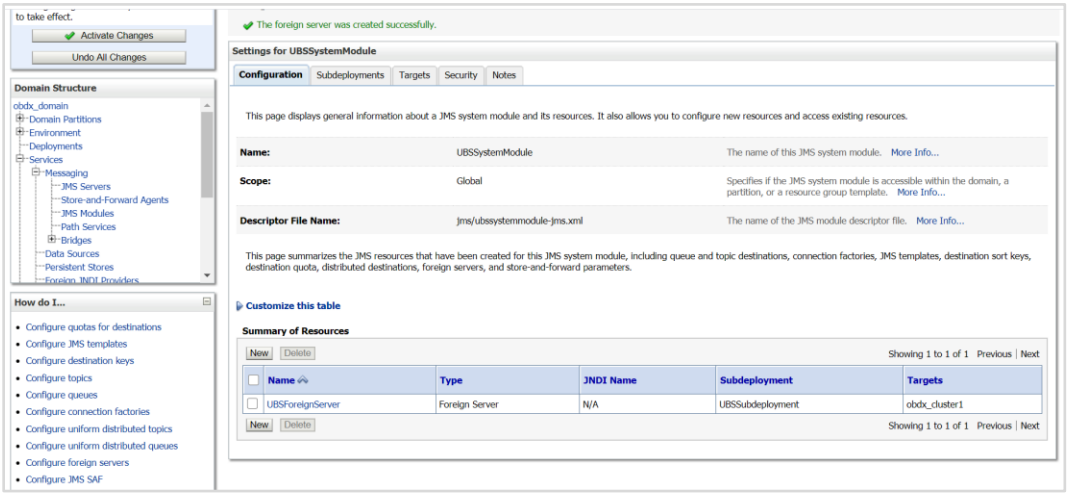


3.16 Creating UBSForeignServer JMS Server

1. In JMSModule create UBSSystemModule

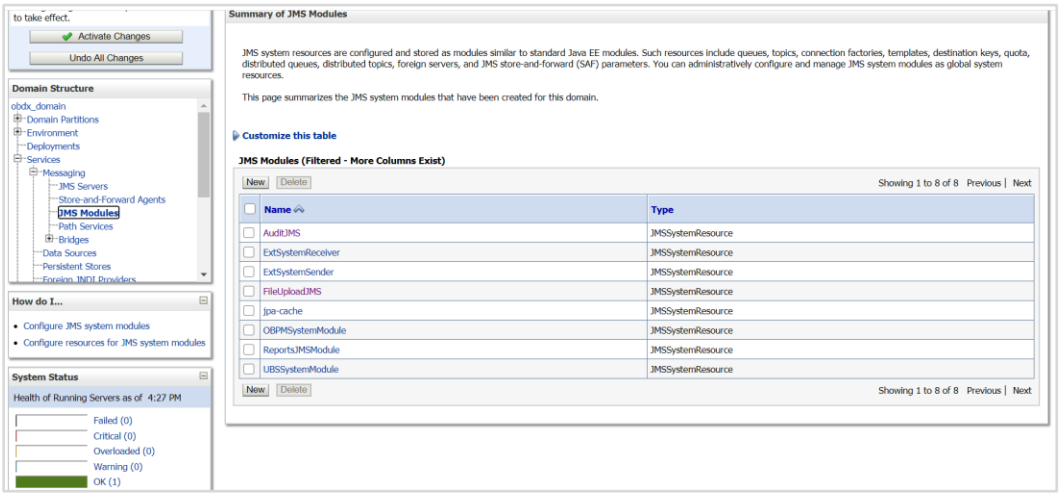


2. Under UBSSystemModule create UBSForeignServer – Foreign Server as shown below

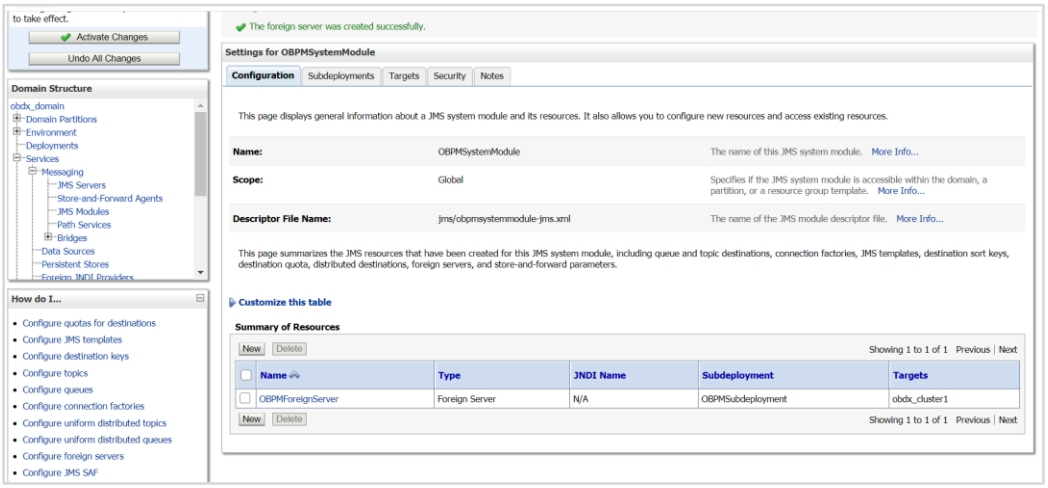


3.17 Creating OBPMForeignServer JMS Server

1. In JMSModule create OBPMSystemModule



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



Home

4. Deploying Applications

Deployment of Lib and Apps

${MW_HOME}/wlserver/common/deployable-libraries/jax-rs-2.0.war (Target - obdx_cluster, AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.domain.ear (Target - obdx_cluster, AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.framework.ear (Target - obdx_cluster, AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.extsystem.domain.ear (Target - obdx_cluster, AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.core.domain.ear (Target - obdx_cluster ,AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.thirdparty.app.domain.ear (Target - obdx_cluster,AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.rest.idm.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/BatchResourceAdapter.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/AuditMDBEAR.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/com.ofss.digx.app.connector.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.mdb.report.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.timer.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.oauth.ear (Target - obdx_cluster,AdminServer)

${OBDX INSTALLER}/installables/app/components/obdx/deploy/obdx.app.oauth.rest.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/cz/obdx.cz.app.domain.ear (Target - obdx_cluster,AdminServer)

${OBDX INSTALLER}/installables/app/cz/obdx.cz.extsystem.domain.ear (Target - obdx_cluster,AdminServer)

${OBDX INSTALLER}/installables/app/cz/obdx.cz.thirdparty.app.domain.ear (Target - obdx_cluster,AdminServer)

${OBDX INSTALLER}/installables/app/components/ubs/deploy/obdx.app.soap.ear (Target - obdx_cluster)

${OBDX INSTALLER}/installables/app/components/thp/deploy/ExtxfaceSimulatorMDB.ear (Target - obdx_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