

**ANNEXURE – 1**

# **Oracle Banking Origination**

Release 14.6.1.0.0

**Part Number F61868-01**

August 2022

# Table of Contents

<b>1. ANNEXURE - 1</b> .....	<b>1-1</b>
1.1 INTRODUCTION .....	1-1
1.2 PLACEHOLDER UPDATE FOR ORACLE BANKING MICROSERVICES ARCHITECTURE SERVICES.....	1-1
1.2.1 Method 1 – Via setUserOverrides.sh file.....	1-1
1.2.2 Method 2 – Via passing the -D params in the Server start argument.....	1-5
1.2.3 Method 3 – Using env files and setUserOverrides.sh file.....	1-7
1.2.4 Method 4 – Workflow Configuration .....	1-15
1.3 HOW TO CREATE DOMAIN AND CLUSTER CONFIGURATION.....	1-20
1.3.1 Domain Creation Configuration .....	1-20
1.3.2 Post Domain Creation Configurations .....	1-29
1.4 HOW TO CREATE DATASOURCE.....	1-31
1.5 HOW TO DEPLOY APPLICATION .....	1-36
1.6 HOW TO UNDEPLOY APPLICATION.....	1-41
1.7 HOW TO RESTART SERVERS.....	1-42
1.8 HOW TO CHECK PORT NUMBER .....	1-45
1.9 WEBLOGIC EMBEDDED LDAP SETUP.....	1-46
1.9.1 Configuration of Weblogic LDAP.....	1-46
1.9.2 Creation of Users .....	1-47
1.9.3 Oracle Banking Microservices Architecture Security Config Table Entries .....	1-50
1.10 ORACLE ANALYTIC SERVER SETUP .....	1-51
1.10.1 Prerequisite .....	1-51
1.10.2 Start BI Server .....	1-51
1.10.3 Upload BI Reports .....	1-51
1.10.4 Test BI Reports .....	1-51

## 1.1 Introduction

This guide is a supporting document for the installation of Oracle Banking Microservices Architecture applications. You can find the reference in the respective installation guides.

## 1.2 Placeholder Update for Oracle Banking Microservices Architecture Services

The Placeholder update can be performed in the following methods:

- Method 1 – Via **setUserOverrides.sh** file
- Method 2 – Via passing the **-D params** in the Server start argument
- Method 3 – Using **env** files and **setUserOverrides.sh** file
- Method 4 – Via Workflow creation in Plato O

### 1.2.1 Method 1 – Via setUserOverrides.sh file

Perform the following steps:

1. Create a file called **setUserOverrides.sh** inside the Weblogic bin location.
2. The following formats of the **setUserOverrides.sh** file and the list of parameters that need to be passed in order to run Oracle Banking Microservices Architecture services properly.

**NOTE:** Below are the list of **-D params** (ENV Variables) which needs to be set for all the individual services. Set a single **-Dparam** as follows:

```
JAVA_OPTIONS="{JAVA_OPTIONS} -DParam =<ParamValue>"  
export JAVA_OPTIONS
```

#### **//Common Properties**

```
-Dplato.services.config.port= <CONFIG_SERVICE_PORT>  
-Dplato.services.config.uri=  
http://<CONFIG_SERVICE_HOSTNAME>:<CONFIG_SERVICE_PORT>  
-Deureka.client.serviceUrl.defaultZone=  
http://<DISCOVERY_SERVICE_HOST>:<DISCOVERY_SERVICE_PORT>/plato-  
discovery-service/eureka  
-Dplato.services.entityservices.port= <PLATO_ORCH_SERVICE_PORT>  
-Dplato.service.logging.path= <LOGGING PATH>
```

#### **//Flyway Common Placeholders**

```
-Dflyway.domain.placeholders.eureka.host= <DISCOVERY_SERVICE_HOST>  
-Dflyway.domain.placeholders.eureka.port= <DISCOVERY_SERVICE_PORT>  
-Dflyway.domain.placeholders.plato-api-gateway.server.port=  
<API_GATEWAY_PORT>  
-Dflyway.domain.placeholders.zipkin.host= <ZIPKIN_HOSTNAME>  
-Dflyway.domain.placeholders.zipkin.port= <ZIPKIN_PORT>
```

#### **//SMS - Needed for other services also**

-Dflyway.domain.placeHolders.sms.username= <SMS\_SCHEMA\_USERNAME>  
-Dflyway.domain.placeHolders.sms.password= <SMS\_SCHEMA\_PASSWORD>  
-Dflyway.domain.placeHolders.sms.jdbcUrl= <SMS\_SCHEMA\_URL>  
-Dflyway.domain.placeHolders.sms.schemas= <SMS\_SCHEMA\_NAME>

**//Plato Config Service - Needed for other services also**

-Dflyway.domain.placeHolders.plato-config.username= <PLATO\_DB\_USERNAME>  
-Dflyway.domain.placeHolders.plato-config.password= <PLATO\_DB\_PASSWORD>  
-Dflyway.domain.placeHolders.plato-config.jdbcUrl= <PLATO\_DB\_URL>  
-Dflyway.domain.placeHolders.driver.className= oracle.jdbc.driver.OracleDriver  
-Dflyway.domain.placeHolders.plato-config.schemas= <PLATO\_DB\_SCHEMANAME>

**//Plato Api Gateway - Needed for other services also**

-Dflyway.domain.placeHolders.api-gateway.username=  
<SECURITY\_DB\_USERNAME>  
-Dflyway.domain.placeHolders.api-gateway.password=  
<SECURITY\_DB\_PASSWORD>  
-Dflyway.domain.placeHolders.api-gateway.jdbcUrl= <SECURITY\_DB\_URL>  
-Dflyway.domain.placeHolders.api-gateway.schemas= <SECURITY\_SCHEMANAME>  
-Dflyway.domain.placeHolders.apigateway.host= <APIGATEWAY\_HOSTNAME>  
-Dflyway.domain.placeHolders.apigateway.port= <APIGATEWAY\_PORT>  
-Dflyway.domain.placeHolders.USER.STORE= <USER.STORE>  
-Dflyway.domain.placeHolders.LDAP.CORS.allowed.origin= <LDAP\_CORS>  
-Dflyway.domain.placeHolders.LDAP.credential.SALT=  
<LDAP\_CREDENTIALS\_SALT>  
-Dflyway.domain.placeHolders.JWT.EXPIRY.seconds= <JWT\_EXPIRY\_SECONDS>  
-Dflyway.domain.placeHolders.LDAP.url = <LDAP\_SERVER\_URL >  
-Dflyway.domain.placeHolders.LDAP.userId = <LDAP\_SERVER\_USER>  
-Dflyway.domain.placeHolders.LDAP.server.base = <LDAP\_SERVER\_BASE>  
-Dflyway.domain.placeHolders.LDAP.server.credential = <LDAP\_CREDENTIALS>  
-Dflyway.domain.placeHolders.LDAP.usersearch.base = <LDAP\_USER\_BASE>  
-Dflyway.domain.placeHolders LDAP.user.prefix = <LDAP\_USER\_PREFIX>  
-Dflyway.domain.placeHolders.LDAP.provider = <LDAP\_PROVIDER>  
-Dflyway.domain.placeHolders.TOKEN.autoregenerate =  
<TOKEN\_AUTOREGENERATION>  
-Dflyway.domain.placeHolders.SSO.enabled = <SSO\_ENABLED>  
-Dflyway.domain.placeHolders.TOKEN.regeneration.enabled =  
<TOKEN\_ALWAYSNEW\_GENERATION >

**//Plato Discovery Service**

-Dflyway.domain.placeHolders.plato-discovery-  
service.server.port=<DISCOVERY\_SERVICE\_PORT>

**//Plato UI-Config Services**

```
-Dflyway.domain.placeHolders.plato-ui-config-  
services.server.port=<UICONFIG_SERVICE_PORT>  
  
-Dflyway.domain.placeHolders.plato-ui-  
config.username=<UICONFIG_SCHEMA_USERNAME>  
  
-Dflyway.domain.placeHolders.plato-ui-  
config.password=<UICONFIG_SCHEMA_PASSWORD>  
  
-Dflyway.domain.placeHolders.plato-ui-config.jdbcUrl=<UICONFIG_SCHEMA_URL>  
  
-Dflyway.domain.placeHolders.plato-ui-  
config.schemas=<UICONFIG_SCHEMA_NAME>
```

### **//Plato Feed Services**

```
-Dflyway.domain.placeHolders.plato-feed-  
services.feed.upload.directory=<FEED_SERVICE_UPLOAD_PATH>  
  
-Dflyway.domain.placeHolders.plato-feed-  
services.server.port=<FEED_SERVICE_PORT>  
  
-Dflyway.domain.placeHolders.plato-feed-  
services.username=<FEED_DB_USERNAME>  
  
-Dflyway.domain.placeHolders.plato-feed-  
services.password=<FEED_DB_PASSWORD>  
  
-Dflyway.domain.placeHolders.plato-feed-services.jdbcUrl=<FEED_DB_URL>  
  
-Dflyway.domain.placeHolders.plato-feed-  
services.schemas=<FEED_SCHEMA_NAME>
```

### **//Plato Batch Server**

```
-Dflyway.domain.placeHolders.plato-batch-  
server.server.port=<BATCH_SERVER_PORT>  
  
-Dflyway.domain.placeHolders.plato-batch-  
server.plato.eventhub.kafka.brokers=<EVETNHUB_KAFKA_BROKERS>  
  
-Dflyway.domain.placeHolders.plato-batch-  
server.plato.eventhub.zk.nodes=<ZK_NODES>  
  
-Dflyway.domain.placeHolders.plato-batch-  
server.username=<BATCH_SCHEMA_USERNAME>  
  
-Dflyway.domain.placeHolders.plato-batch-  
server.password=<BATCH_SCHEMA_PASSWORD>  
  
-Dflyway.domain.placeHolders.plato-batch-server.jdbcUrl=<BATCH_SCHEMA_URL>  
  
-Dflyway.domain.placeHolders.plato-batch-  
server.schemas=<BATCH_SCHEMA_NAME>
```

### **// Plato-Alerts-Management-Services**

```
-Dflyway.domain.placeHolders.plato-alerts-management-  
services.server.port=<ALERTS-MANAGEMENT-SERVER-PORT>  
  
-Dflyway.domain.placeHolders.plato-alerts-management-  
services.plato.eventhub.kafka.brokers=<EVETNHUB_KAFKA_BROKERS>  
  
-Dflyway.domain.placeHolders.plato-alerts-management-  
services.plato.eventhub.zk.nodes=<ZK_NODES>  
  
-Dflyway.domain.placeHolders.plato-alerts-management-  
services.username=<ALERTS_SCHEMA_USERNAME>
```

-Dflyway.domain.placeholders.plato-alerts-management-services.password=<ALERTS\_SCHEMA\_PASSWORD>

-Dflyway.domain.placeholders.plato-alerts-management-services.jdbcUrl=<ALERTS\_SCHEMA\_URL>

-Dflyway.domain.placeholders.plato-alerts-management-services.schemas=<ALERTS\_SCHEMA\_NAME>

#### **//Plato Orch Service**

-Dflyway.domain.placeholders.plato-orch-service.server.port=<ORCH\_SERVICE\_PORT>

-Dflyway.domain.placeholders.plato-orchestrator.hostname=<CONDUCTOR-EUREKA-HOSTNAME >

#### **//Conductor**

-Dconductor.properties=<CONDUCTOR\_CONFIG\_FILE\_PATH>

#### **//Common core NLP services**

-Dflyway.domain.placeholders.cmc-nlp-annotator-

services.server.port=<CMC\_NLP\_ANNOTATOR\_SERVICES\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-dashboard-widget-

services.server.port=<CMC\_NLP\_DASHBOARD\_SERVICES\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-model-mngmnt-

services.server.port=<CMC\_NLP\_MODEL\_MANGEMENT\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-online-processing-

services.server.port=<CMC\_NLP\_ONLINE\_PROCESSING\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-tag-maint-

services.server.port=<CMC\_NLP\_TAG\_MAINTENANCE\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-text-extraction-

services.server.port=<CMC\_NLP\_TEXT\_EXTRACTION\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-txn-log-

services.server.port=<CMC\_NLP\_TXN\_LOG\_SERVICES\_PORT>

-Dflyway.domain.placeholders.cmc-nlp-util-

services.server.port=<CMC\_NLP\_UTIL\_SERVICES\_PORT>

#### **// Common core NLP Poller service**

-Dflyway.domain.placeholders.cmc-fc-ai-ml-services.server.port=<Server\_Port>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.postingPath=<Posting\_Path>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-services.server.pollingPath=<Polling\_Path>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.pollingEmail=<Polling\_Email>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.emailServerHost=<Email\_Server\_Host>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.emailServerPort=<Email\_Server\_PORT>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.pollingFrequency=<Polling\_Frequency>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.pollerInitialDelay=<Poller\_Initial\_Delay>

-Dflyway.domain.placeholders.cmc-fc-ai-ml-  
 services.server.emailPassword=<Poller\_Email\_Password>

### 1.2.2 Method 2 – Via passing the -D params in the Server start argument

All the above mentioned -D parameters can be passed through the Server start argument in respective managed server. Perform the following steps:

1. Navigate to the Server **Configuration** tab and click managed server to which you want to pass the values.

**Summary of Servers**

**Configuration** Control

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.  
 This page summarizes each server that has been configured in the current WebLogic Server domain.

[Customize this table](#)

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

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

New Clone Delete Showing 1 to 2 of 2 Previous | Next

<input type="checkbox"/>	Name	Type	Cluster	Machine	State	Health	Listen Port
<input type="checkbox"/>	AdminServer(admin)	Configured		whf00dkx	RUNNING	OK	7001
<input type="checkbox"/>	managed1_server	Configured		whf00dkx	RUNNING	OK	7003

New Clone Delete Showing 1 to 2 of 2 Previous | Next

2. Select **Server Start** tab in the next screen.

Settings for managed1\_server

Configuration Protocols Logging Debug Monitoring Control Deployments Services Security Notes

General Cluster Services Keystores SSL Federation Services Deployment Migration Tuning Overload Concurrency Health Monitoring **Server Start**

Web Services Coherence

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Save

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

**Java Home:**  The Java home directory (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

**Java Vendor:**  The Java Vendor value to use when starting this server. [More Info...](#)

**BEA Home:**  The BEA home directory (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

**Root Directory:**  The directory that this server uses as its root directory. This directory must be on the computer that hosts Node Manager. If you do not specify a Root Directory value, the domain directory is used by default. [More Info...](#)

**Class Path:**  The classpath (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

3. Edit the **Arguments** field and pass all the environment parameters required for the service to run.

Node Manager is a WebLogic Server utility that you can use to start, suspend, shut down, and restart servers in normal or unexpected conditions. Use this page to configure the startup settings that Node Manager will use to start this server on a remote machine.

**Java Home:**  The Java home directory (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

**Java Vendor:**  The Java Vendor value to use when starting this server. [More Info...](#)

**BEA Home:**  The BEA home directory (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

**Root Directory:**  The directory that this server uses as its root directory. This directory must be on the computer that hosts Node Manager. If you do not specify a Root Directory value, the domain directory is used by default. [More Info...](#)

**Class Path:**  The classpath (path on the machine running Node Manager) to use when starting this server. [More Info...](#)

**Arguments:**  The arguments to use when starting this server. [More Info...](#)

```
-Deureka.server.enable-self-preservation=false
-Dspring.flyway.enabled=false -Dflyway.enabled=false
-Deureka.client.serviceUrl.defaultZone=http://whf00dkx:7003
/plato-discovery-service/eureka -Dserver.port=7003
```

4. Save the configuration and restart the managed server. After you restart, the service can be started or deployed properly.



### 1.2.3 Method 3 – Using env files and setUserOverrides.sh file

Perform the following steps:

1. Copy the **setUserOverrides.sh** file to each of the <domain>/bin folder. The example of the file is given below:

```
#!/bin/bash
# shellcheck disable=SC1090
# Common functions

set -e -x

config_file=""
PLATO_CONFIG_MANAGED_SERVER_NAME=""

# This file is used only for PLATO-CONFIG service
plato_config_file="${DOMAIN_HOME}/bin/plato-config-deploy.env"

# This file is used for rest of the services
domain_config_file="${DOMAIN_HOME}/bin/domain-config-deploy.env"

if [ -f "$plato_config_file" ]; then
    PLATO_CONFIG_MANAGED_SERVER_NAME=`cat ${DOMAIN_HOME}/bin/plato-config-deploy.env | grep "PLATO_CONFIG_MANAGED_SERVER_NAME" | cut -d=' ' -f2`
fi

if [ "${SERVER_NAME}" = "${PLATO_CONFIG_MANAGED_SERVER_NAME}" ]; then
    # This will get executed only for Plato-config service entries
    config_file="${plato_config_file}"
else
    # This will get executed for all other services
    config_file="${domain_config_file}"
fi

if [ -f "$config_file" ]
then
    while read -r prop || [ -n "$prop" ]
    do
        case "$prop" in \#*) continue ;; esac
        if [ -z "${prop}" ]; then
            continue
        else
            PLACEHOLDERS=${PLACEHOLDERS}" $(echo -D$prop)
            PLACEHOLDERS=${PLACEHOLDERS}"
        fi
    fi
```

```

done < "$config_file"
else
    echo "$config_file not found. please provide the property file to set -D parameter"
    exit 1
fi

PLACEHOLDERS="{PLACEHOLDERS}"

JAVA_OPTIONS="{JAVA_OPTIONS}{PLACEHOLDERS}"

export JAVA_OPTIONS

echo "{JAVA_OPTIONS}"

```

2. Place the **env** files containing all the key value pairs of the **-D params** in the respective <domain>/env folder.

**NOTE:** The plato-config-deploy.env file contains all the key value pairs specific only to the plato-config-service and need to be placed in the bin folder of the plato-domain. The domain-config-deploy.env file contains the key-value pairs for the rest of the services and should be placed in each <domain>/bin folder.

The sample for each of the files are given below:

#### **plato-config-deploy.env**

```

### Managed server name of plato-config service ###
PLATO_CONFIG_MANAGED_SERVER_NAME=

### plato config flyway connection entries ###
flywayTask=migrate
flyway.enabled=true
spring.flyway.enabled=false
plato-config.flyway.domain.db.username=
plato-config.flyway.domain.db.password=
plato-config.flyway.domain.db.jdbcUrl=
plato-config.flyway.domain.schemas=
plato-
config.flyway.domain.locations=db/migration/domain/plato,db/migration/domain/sms,db/migra
tion/domain/cmc,db/migration/domain/obvam

#### Kafka properties for all services ####
flyway.domain.placeholders.plato.eventhub.broker.hosts=
flyway.domain.placeholders.plato.eventhub.zookeeper.hosts=

#### Kafka Security for all services ####
flyway.domain.placeholders.plato.eventhub.broker.hosts=
flyway.domain.placeholders.plato.eventhub.zookeeper.hosts=
flyway.domain.placeholders.kafka.ssl.truststore.location=

```

```
flyway.domain.placeholders.kafka.ssl.truststore.password=  
flyway.domain.placeholders.kafka.broker.username=  
flyway.domain.placeholders.kafka.broker.password=  
  
### common entries for all services ###  
flyway.domain.placeholders.driver.className=oracle.jdbc.driver.OracleDriver  
  
### eureka entries for all services ###  
flyway.domain.placeholders.eureka.host=  
flyway.domain.placeholders.eureka.port=  
  
### zipkin entries for all services ###  
flyway.domain.placeholders.zipkin.host=  
flyway.domain.placeholders.zipkin.port=  
  
### plato config flyway placeholder entries ###  
flyway.domain.placeholders.plato-config.username=  
flyway.domain.placeholders.plato-config.password=  
flyway.domain.placeholders.plato-config.jdbcUrl=  
flyway.domain.placeholders.plato-config.schemas=  
flyway.domain.placeholders.plato-config.sessionIdleTimeout=  
flyway.domain.placeholders.plato-config.sessionIdleWarningTime=  
flyway.domain.placeholders.plato-config.environment=  
  
### plato api-gateway flyway placeholder entries ###  
flyway.domain.placeholders.api-gateway.host=  
flyway.domain.placeholders.api-gateway.username=  
flyway.domain.placeholders.api-gateway.password=  
flyway.domain.placeholders.api-gateway.jdbcUrl=  
flyway.domain.placeholders.api-gateway.schemas=  
flyway.domain.placeholders.plato-api-gateway.server.port=  
  
### plato api-gateway LDAP flyway placeholder entries ###  
flyway.domain.placeholders.USER.STORE=  
flyway.domain.placeholders.LDAP.CORS.allowed.origin=  
flyway.domain.placeholders.LDAP.credential.SALT=  
flyway.domain.placeholders.JWT.EXPIRY.seconds=  
flyway.domain.placeholders.LDAP.url=  
flyway.domain.placeholders.LDAP.userId=  
flyway.domain.placeholders.LDAP.server.base=  
flyway.domain.placeholders.LDAP.server.credential=  
flyway.domain.placeholders.LDAP.usersearch.base=
```

```
flyway.domain.placeholders.LDAP.user.prefix=  
# Allowed values for LDAP provider are: EMBEDDED_WEBLOGIC, PLATO  
# If LDAP is running in weblogic then value should be EMBEDDED_WEBLOGIC  
# If spring based LDAP(which is run through a jar provided) is used, then the value should be  
PLATO  
flyway.domain.placeholders.LDAP.provider=  
flyway.domain.placeholders.TOKEN.autoregenerate=  
flyway.domain.placeholders.SSO.enabled=  
flyway.domain.placeholders.TOKEN.regeneration.enabled=  
  
### plato-ui-config flyway placeholder entries ###  
flyway.domain.placeholders.plato-ui-config.username=  
flyway.domain.placeholders.plato-ui-config.password=  
flyway.domain.placeholders.plato-ui-config.jdbcUrl=  
flyway.domain.placeholders.plato-ui-config.schemas=  
flyway.domain.placeholders.plato-ui-config-services.server.port=  
flyway.domain.placeholders.apigateway.host=  
flyway.domain.placeholders.apigateway.port=  
  
### plato-discovery flyway placeholder entries ###  
flyway.domain.placeholders.plato-discovery-service.server.port=  
  
### plato-orch flyway placeholder entries ###  
flyway.domain.placeholders.plato-orch-service.server.port=  
flyway.domain.placeholders.plato-orchestrator.hostname=  
  
### plato-feed flyway placeholder entries ###  
flyway.domain.placeholders.plato-feed-services.username=  
flyway.domain.placeholders.plato-feed-services.password=  
flyway.domain.placeholders.plato-feed-services.jdbcUrl=  
flyway.domain.placeholders.plato-feed-services.jndi=jdbc/PLATOFEED  
flyway.domain.placeholders.plato-feed-services.schemas=  
flyway.domain.placeholders.plato-feed-services.feed.upload.directory=  
flyway.domain.placeholders.plato-feed-services.server.port=  
  
### plato-batch flyway placeholder entries ###  
flyway.domain.placeholders.plato-batch-server.username=  
flyway.domain.placeholders.plato-batch-server.password=  
flyway.domain.placeholders.plato-batch-server.jdbcUrl=  
flyway.domain.placeholders.plato-batch-server.schemas=  
flyway.domain.placeholders.plato-batch-server.server.port=  
flyway.domain.placeholders.plato-batch-server.plato.eventhub.kafka.brokers=
```

```
flyway.domain.placeholders.plato-batch-server.plato.eventhub.zk.nodes=  
flyway.domain.placeholders.plato-batch-server.jndi=jdbc/PLATOBATCH  
  
### plato-alerts-management flyway placeholder entries ###  
flyway.domain.placeholders.plato-alerts-management-services.username=  
flyway.domain.placeholders.plato-alerts-management-services.password=  
flyway.domain.placeholders.plato-alerts-management-services.jdbcUrl=  
flyway.domain.placeholders.plato-alerts-management-services.schemas=  
flyway.domain.placeholders.plato-alerts-management-services.server.port=  
  
### sms flyway placeholder entries ###  
flyway.domain.placeholders.sms-core-services.server.port=  
flyway.domain.placeholders.sms.username=  
flyway.domain.placeholders.sms.password=  
flyway.domain.placeholders.sms.jdbcUrl=  
flyway.domain.placeholders.sms.schemas=  
  
### cmncore flyway placeholder entries ###  
flyway.domain.placeholders.cmncore.username=  
flyway.domain.placeholders.cmncore.password=  
flyway.domain.placeholders.cmncore.jdbcUrl=  
flyway.domain.placeholders.cmncore.schemas=  
flyway.domain.placeholders.cmc-corebanking-adapter-service.server.port=  
flyway.domain.placeholders.cmc-currency-services.server.port=  
flyway.domain.placeholders.cmc-account-services.server.port=  
flyway.domain.placeholders.cmc-base-services.server.port=  
flyway.domain.placeholders.cmc-external-virtual-account-services.server.port=  
flyway.domain.placeholders.cmc-branch-services.server.port=  
flyway.domain.placeholders.cmc-customer-services.server.port=  
flyway.domain.placeholders.cmc-external-chart-account-services.server.port=  
flyway.domain.placeholders.cmc-external-system-services.server.port=  
flyway.domain.placeholders.cmc-advice-services.server.port=  
flyway.domain.placeholders.cmc-facilities-services.server.port=  
flyway.domain.placeholders.cmc-txn-code-services.server.port=  
flyway.domain.placeholders.cmc-settlement-services.server.port=  
flyway.domain.placeholders.cmc-businessoverrides-services.server.port=  
flyway.domain.placeholders.cmc-resource-segment-orchestrator-service.server.port=  
flyway.domain.placeholders.cmc-screenclass-services.server.port=  
flyway.domain.placeholders.cmc-datasegment-services.server.port=  
flyway.domain.placeholders.cmc-settlements-services.server.port=
```

```

flyway.domain.placeholders.cmc-transactioncontroller-services.server.port=
flyway.domain.placeholders.cmc-report-services.server.port=
flyway.domain.placeholders.cmc-nlp-annotator-services.server.port=
flyway.domain.placeholders.cmc-nlp-dashboard-widjet-services.server.port=
flyway.domain.placeholders.cmc-nlp-model-mngmnt-services.server.port=
flyway.domain.placeholders.cmc-nlp-online-processing-services.server.port=
flyway.domain.placeholders.cmc-nlp-tag-maint-services.server.port=
flyway.domain.placeholders.cmc-nlp-text-extraction-services.server.port=
flyway.domain.placeholders.cmc-nlp-txn-log-services.server.port=
flyway.domain.placeholders.cmc-nlp-util-services.server.port=
flyway.domain.placeholders.cmc-batch-services.server.port=
flyway.domain.placeholders.cmc-fc-ai-ml-services.server.port=
flyway.domain.placeholders.cmc-fc-ai-ml-services.postingPath=
flyway.domain.placeholders.cmc-fc-ai-ml-services.pollingEmail=
flyway.domain.placeholders.cmc-fc-ai-ml-services.emailServerPort=
flyway.domain.placeholders.cmc-fc-ai-ml-services.emailServerHost=
flyway.domain.placeholders.cmc-fc-ai-ml-services.pollingFrequency=
flyway.domain.placeholders.cmc-fc-ai-ml-services.pollerInitialDelay=
flyway.domain.placeholders.cmc-fc-ai-ml-services.emailPassword=
flyway.domain.placeholders.cmc-fc-ai-ml-services.pollingPath=

### biPublisher related cmc-report-service entries ###
flyway.domain.placeholders.weblogic.userid=
flyway.domain.placeholders.weblogic.password=
flyway.domain.placeholders.biPublisher.host=
flyway.domain.placeholders.biPublisher.port=
flyway.domain.placeholders.runReportTemplate=
flyway.domain.placeholders.emailTemplate=
flyway.domain.placeholders.dms.host=
flyway.domain.placeholders.dms.port=

### flyway jndi connection details for shared services placeholder entries ###
flyway.domain.placeholders.plato.jndi=jdbc/PLATO
flyway.domain.placeholders.plato-config.jndi=jdbc/PLATO
flyway.domain.placeholders.plato-sec.jndi=jdbc/PLATO_SECURITY
flyway.domain.placeholders.plato-ui-config.jndi=jdbc/PLATO_UI_CONFIG
flyway.domain.placeholders.sms.jndi=jdbc/sms
flyway.domain.placeholders.cmncore.jndi=jdbc/CMNCORE

### flyway jndi connection details for obvam services placeholder entries ###

```

flyway.domain.placeholders.eie.jndi=jdbc/EIE  
flyway.domain.placeholders.eie.server.port=  
flyway.domain.placeholders.eie.schemas=  
  
flyway.domain.placeholders.elm.jndi=jdbc/ELM  
flyway.domain.placeholders.elm.server.port=  
flyway.domain.placeholders.elm.schemas=  
  
flyway.domain.placeholders.vam.jndi=jdbc/VAM  
flyway.domain.placeholders.vam.server.port=  
flyway.domain.placeholders.vam.schemas=  
  
flyway.domain.placeholders.vac.jndi=jdbc/VAC  
flyway.domain.placeholders.vac.server.port=  
flyway.domain.placeholders.vac.schemas=  
  
flyway.domain.placeholders.vab.jndi=jdbc/VAB  
flyway.domain.placeholders.vab.server.port=  
flyway.domain.placeholders.vab.schemas=  
  
flyway.domain.placeholders.vae.jndi=jdbc/VAE  
flyway.domain.placeholders.vae.server.port=  
flyway.domain.placeholders.vae.schemas=  
  
flyway.domain.placeholders.eda.jndi=jdbc/EDA  
flyway.domain.placeholders.eda.server.port=  
flyway.domain.placeholders.eda.schemas=  
  
flyway.domain.placeholders.vai.jndi=jdbc/VAI  
flyway.domain.placeholders.vai.server.port=  
flyway.domain.placeholders.vai.schemas=  
  
flyway.domain.placeholders.van.jndi=jdbc/VAN  
flyway.domain.placeholders.van.server.port=  
flyway.domain.placeholders.van.schemas=  
  
flyway.domain.placeholders.vap.jndi=jdbc/VAP  
flyway.domain.placeholders.vap.server.port=  
flyway.domain.placeholders.vap.schemas=  
  
flyway.domain.placeholders.vas.jndi=jdbc/VAS  
flyway.domain.placeholders.vas.server.port=  
flyway.domain.placeholders.vas.schemas=  
  
flyway.domain.placeholders.vat.jndi=jdbc/VAT  
flyway.domain.placeholders.vat.server.port=

flyway.domain.placeholders.vat.schemas=  
flyway.domain.placeholders.vaj.server.port=  
flyway.domain.placeholders.platoorch.domain.jndi=jdbc/PLATO-O  
flyway.domain.placeholders.platoorch.domain.schemas=  
flyway.domain.placeholders.plato.alerts.email.userId=  
flyway.domain.placeholders.plato.alerts.email.password=  
flyway.domain.placeholders.plato.alerts.cmc.userId=  
flyway.domain.placeholders.plato.alerts.cmc.branchCode=  
flyway.domain.placeholders.plato.alerts.cmc.appId=  
flyway.domain.placeholders.plato-rule.hostname=  
flyway.domain.placeholders.plato-rule-service.server.port=  
flyway.domain.placeholders.platorule.domain.jndi=  
flyway.domain.placeholders.platorule.domain.schemas=  
flyway.domain.placeholders.obrh.import.data.disable-modify=  
flyway.domain.placeholders.cmc-obrh-services.kafka.server.path=  
flyway.domain.placeholders.cmc-obrh-services.zookeeper.server.path=  
flyway.domain.placeholders.cmc.schemas=  
flyway.domain.placeholders.cmc-nlp-opennlp-services.server.port=  
flyway.domain.placeholders.cmc-nlp-maintenance-services.server.port=  
flyway.domain.placeholders.cmc-nlp-pipeline-services.server.port=  
flyway.domain.placeholders.cmc-nlp-docview-services.server.port=  
flyway.domain.placeholders.cmc-ml-indb-services.server.port=  
flyway.domain.placeholders.cmc-obrh-services.kafka.enabled=  
flyway.domain.placeholders.cmc-sla-services.server.port=  
flyway.domain.placeholders.cmc-obcbs-services.schemas=  
flyway.domain.placeholders.obcbs.server.port=  
flyway.domain.placeholders.orch.cmc.brn=  
flyway.domain.placeholders.orch.cmc.user=  
flyway.domain.placeholders.orch.enableDynamicAllocation=



```
flyway.domain.placeholders.orch.enableSLA=  
flyway.domain.placeholders.report-service.server.port=  
flyway.domain.placeholders.report-service.hostname=  
flyway.domain.placeholders.report-service.domain.jndi=jdbc/PLATOREPORT  
flyway.domain.placeholders.report-service.template-metadata-directory=  
flyway.domain.placeholders.report-service.output-directory=  
flyway.domain.placeholders.report-service.fop-config-file=
```

```
### generic entries for all services ###
```

```
spring.cloud.config.uri=  
apigateway.url=  
service.logging.environment=  
service.logging.path=
```

#### **domain-config-deploy.env**

```
### domain config flyway connection entries ###
```

```
flywayTask=migrate  
flyway.enabled=true  
spring.flyway.enabled=false
```

```
### generic entries for all services ###
```

```
spring.cloud.config.uri=  
apigateway.url=  
service.logging.environment=  
service.logging.path=
```

### **1.2.4 Method 4 – Workflow Configuration**

Follow the below steps to create a workflow:

1. Metadata of the workflow creation. The sample DSL for workflow creation is given below:

```
{  
  "name": "initialTest",  
  "description": "Test workflow",  
  "version": 4,  
  "tasks": [  
    {  
      "name": "TEST",  
      "taskReferenceName": "TESTING3",  
      "description": "TESTING2",
```

```

        "inputParameters": {
            "FUNCTIONAL_CODE": "TEST_FA_ILS_REGTN2",
            "processRefNo":
                "${workflow.input.transactionModel.txnIdentification.processRefNo}",
            "processName": "Testing Process2",
            "processCode":
                "${workflow.input.transactionModel.txnIdentification.processName}",
            "transactionModel": "${workflow.input.transactionModel}",
            "stage": "TESTING2",
            "priority":
                "${workflow.input.transactionModel.transactionData.moduleData.taskPriority}",
            "applicationDate":
                "${workflow.input.transactionModel.txnIdentification.applicationDate}",
            "applicationNumber":
                "${workflow.input.transactionModel.txnIdentification.processRefNo}",
            "processRefNumber":
                "${workflow.input.transactionModel.txnIdentification.processRefNo}",
            "branch": "${workflow.input.transactionModel.txnIdentification.branchCode}",
            "user": "${workflow.input.transactionModel.txnIdentification.currentUser}",
            "customerNumber":
                "${workflow.input.transactionModel.transactionData.moduleData.customerId}",
            "amount":
                "${workflow.input.transactionModel.transactionData.moduleData.amount}",
            "currencyCode":
                "${workflow.input.transactionModel.transactionData.moduleData.currency}",
            "TASK_OUTCOMES": [
                "PROCEED"
            ],
            "moduleCode": "OBTFPM",
            "customFilter": [
                {
                    "key": "contractRefNo",
                    "label": "Back Office Reference"
                },
                {
                    "key": "otherRefNo",
                    "label": "External Reference"
                }
            ]
        },
        "type": "WAIT",

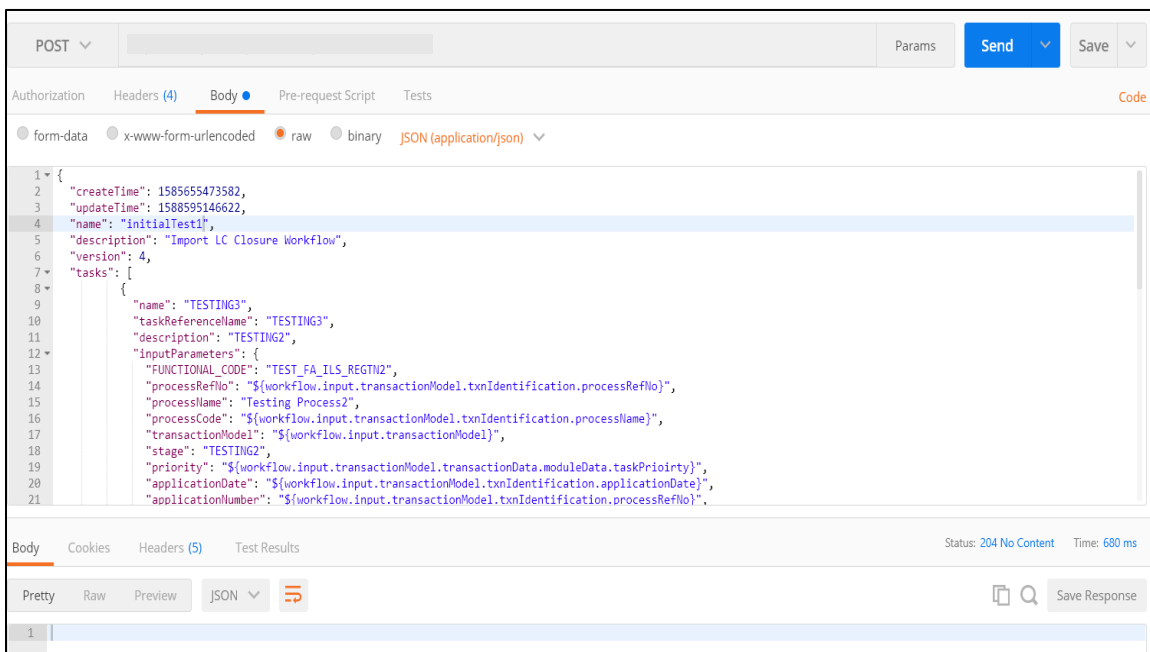
```

```

        "startDelay": 0,
        "optional": false,
        "asyncComplete": false
    }
  ],
  "outputParameters": {
    "stage": "CLMO_FA_SNPOAR_APPEN",
    "taskOutcome": "PROCEED_WITH_PARTICIPANT"
  },
  "schemaVersion": 2,
  "restartable": true,
  "workflowStatusListenerEnabled": false
}

```

Call the API (/api/metadata/workflow) and pass the DSL in body. The following screen depicts the sample workflow:



## 2. Workflow Creation

Call the API (/api/workflow) to create the workflow. This API provides the information to the workflow metadata which we have created using previous call.

Body:

```

{
  "name": "initialTest",
  "description": "initialTest",
  "version": 4,
  "input": {
    "transactionModel": {

```

```
    "transactionData": {
      "moduleData": {
        "amount": 122,
        "currency": "GBP",
        "customerId": "001506",
        "customerName": "MARKS AND SPENCER",
        "taskPriority": "H"
      }
    },
    "txnIdentification": {
      "branchCode": "000",
      "currentStage": "TEST_FA_ILS_REGTN2",
      "currentUser": "SWAGATIKA",
      "key1": "Desk",
      "moduleCode": "TRMO",
      "processName": "Testing Process2",
      "processRefNo": "300ILCI012260",
      "applicationDate": 1588582461960,
      "taskOutcome": "PROCEED",
      "taskPriority": "H"
    }
  }
}
```

The following screen depicts the sample workflow:

POST <https://webf00for7000/latam/ccb/aml/workflow> Params Send Save

Authorization Headers (4) Body Pre-request Script Tests Code

form-data  x-www-form-urlencoded  raw  binary JSON (application/json)

```
1-21 {
  "name": "initialTest1",
  "description": "Import LC Closure Workflow",
  "version": 4,
  "input": {
    "transactionModel": {
      "transactionData": {
        "moduleData": {
          "amount": 122,
          "currency": "GBP",
          "customerId": "001506",
          "customerName": "MARKS AND SPENCER",
          "taskPriority": "H"
        }
      },
      "txnIdentification": {
        "branchCode": "000",
        "currentStage": "TEST_FA_ILS_REGTN2",
        "currentUser": "SNAGATIKA",
        "key1": "Desk",
        "moduleCode": "TRMD",

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 187272 ms

Pretty Raw Preview Text Save Response

```
1 151d78d7-6711-46ae-be15-bf7b550c4b36
```

## 1.3 How to Create Domain and Cluster Configuration

This section contains the following sub-sections:

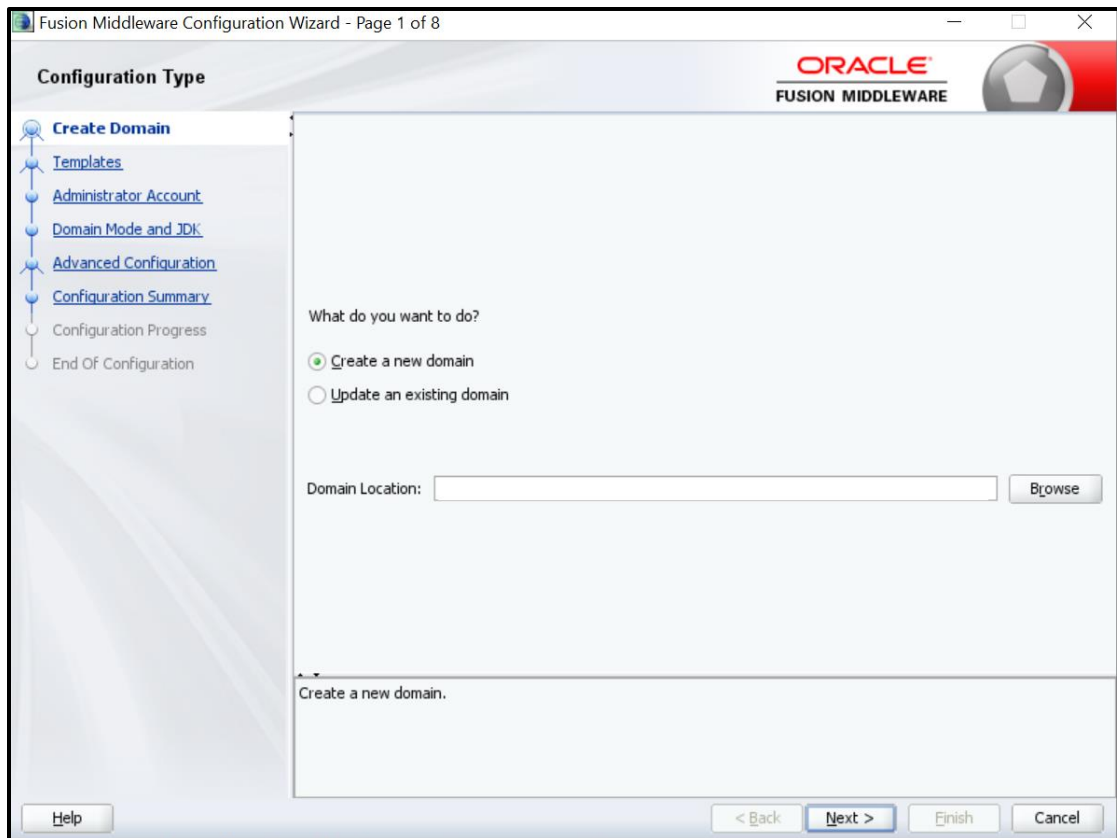
- Domain Creation Configuration
- Post Domain Creation Configurations

### 1.3.1 Domain Creation Configuration

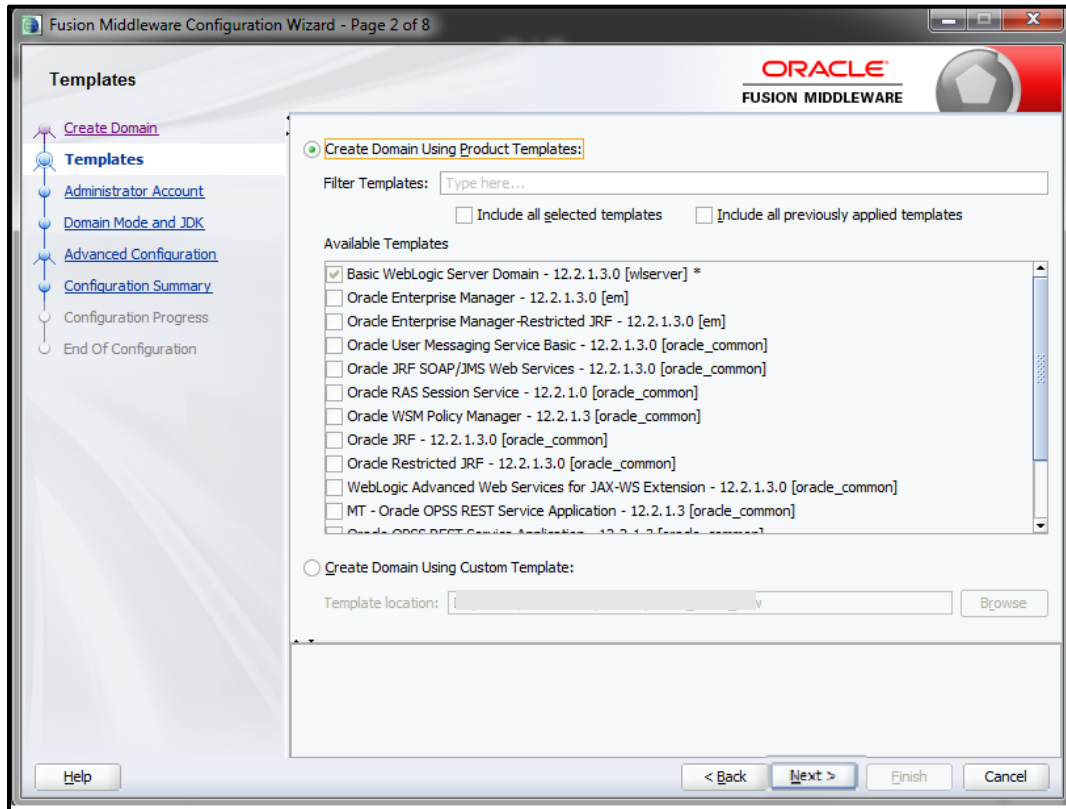
Perform the following steps for domain and cluster configuration:

**NOTE:** Name need not to be same as provided in Screenshot.

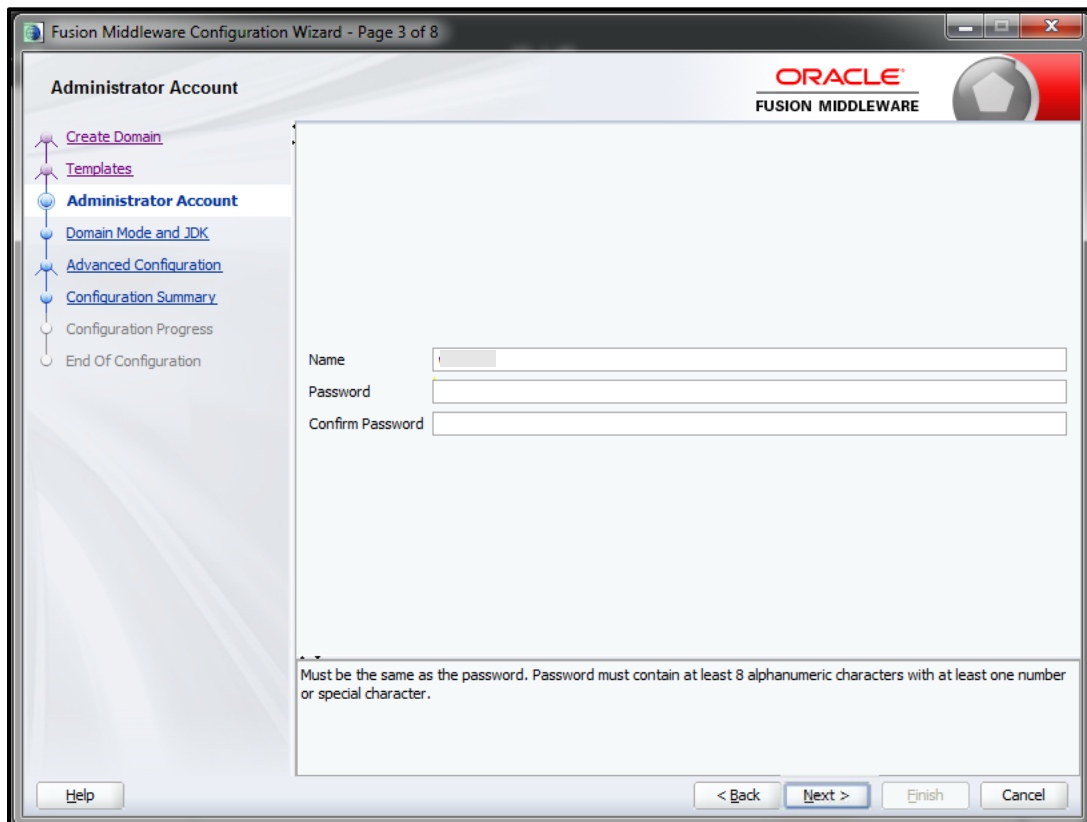
1. Open **/oracle\_common/common/bin** and run **config.cmd** (or **.sh** if operating system is linux). Create domain with required cluster and server configurations. Refer to the screenshots below.
2. Select **Create a new domain** and provide domain name. For example, **platoinfra\_domain**.



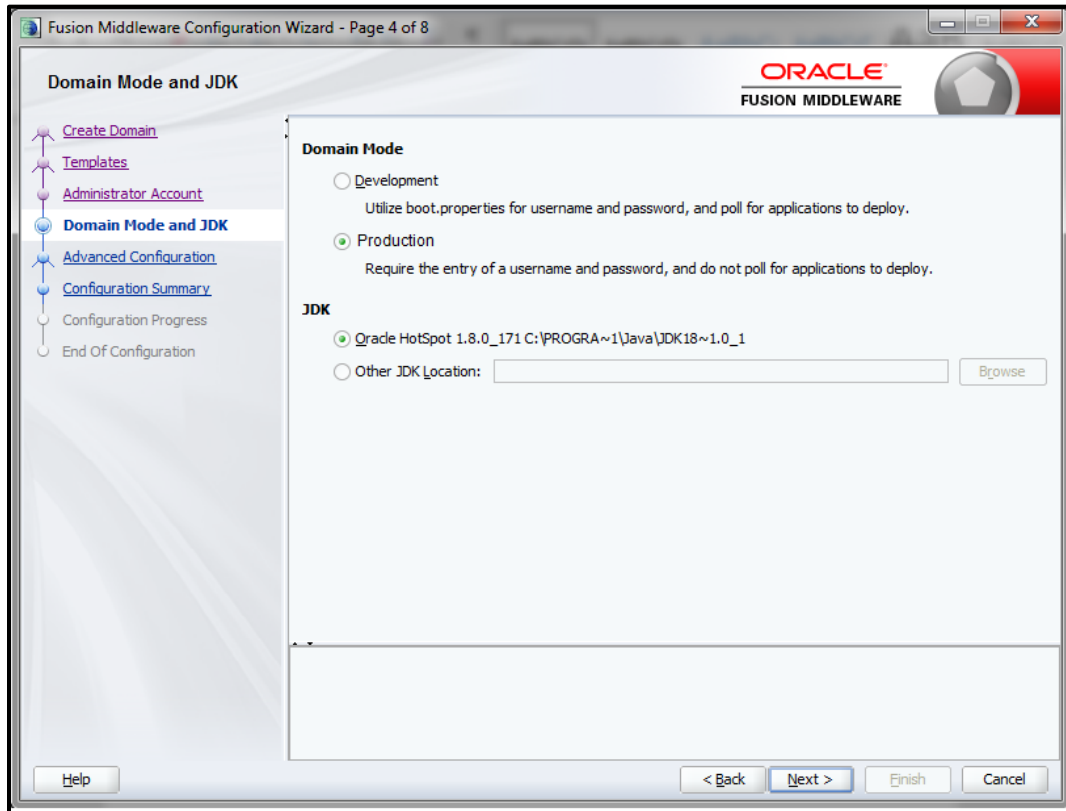
3. Click **Next** to create simple domain with default templates.



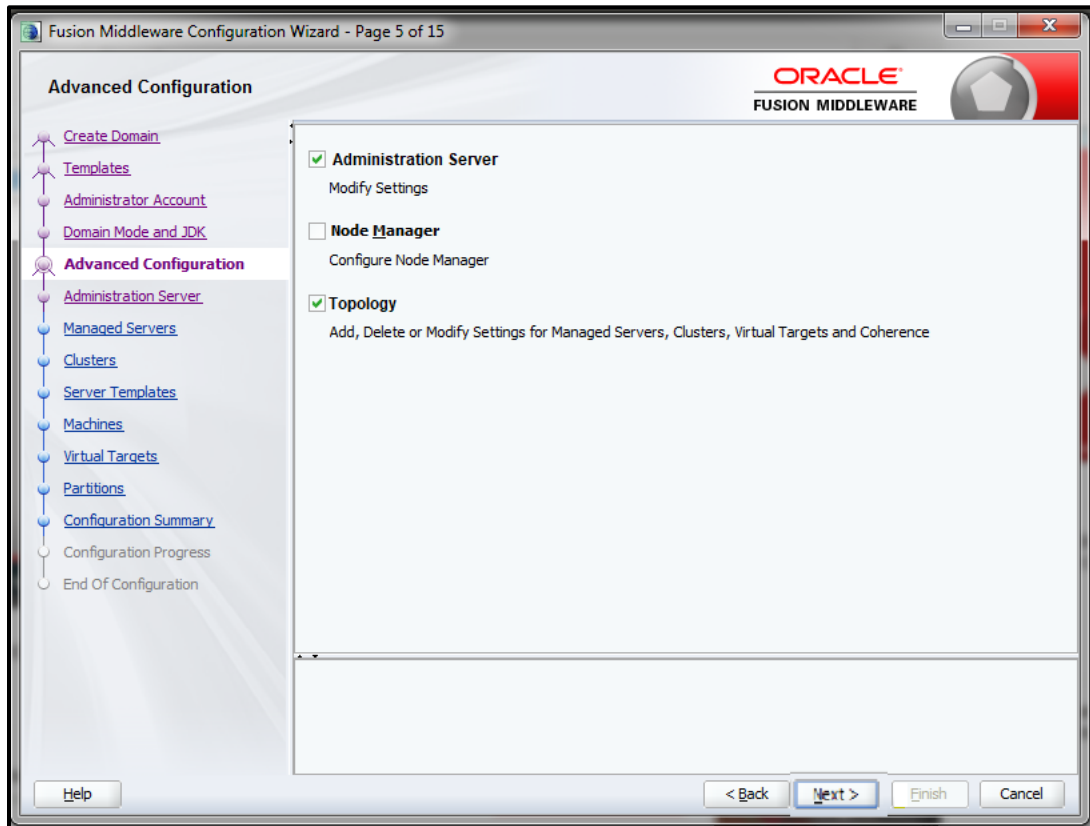
4. Set password and confirm, click **Next** to proceed.



5. Select **Domain Mode** as **Production** and select **JDK**.

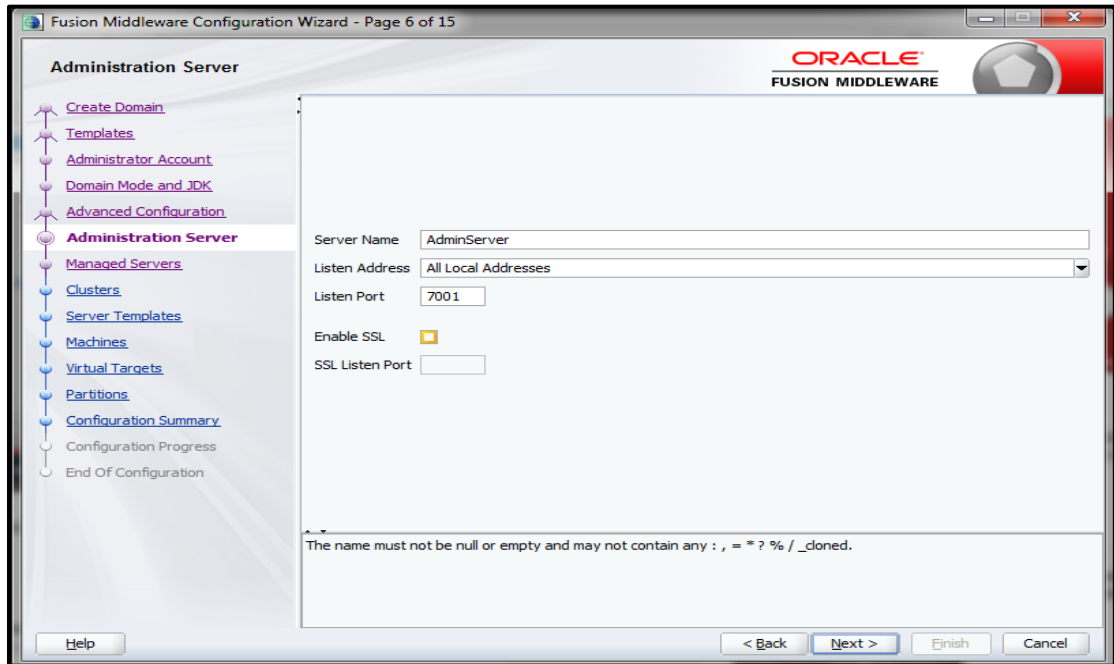


6. Select **Administration Server** and **Topology** in advanced configurations.

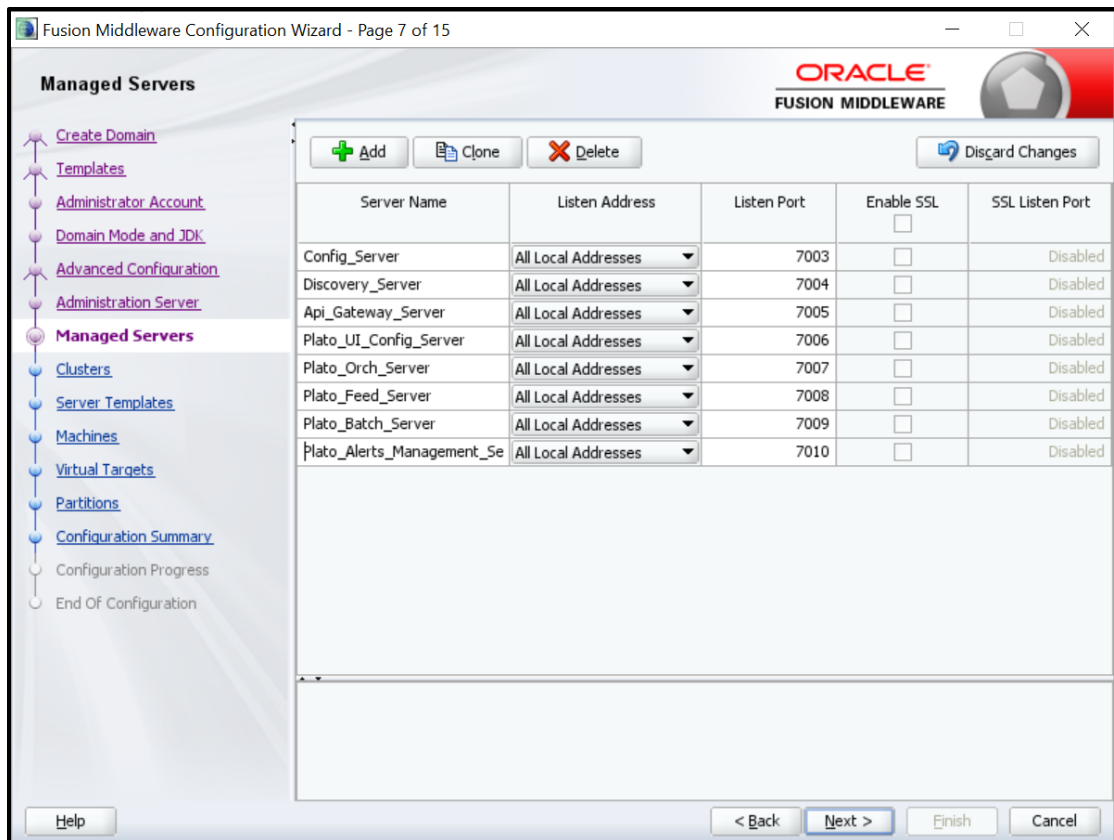




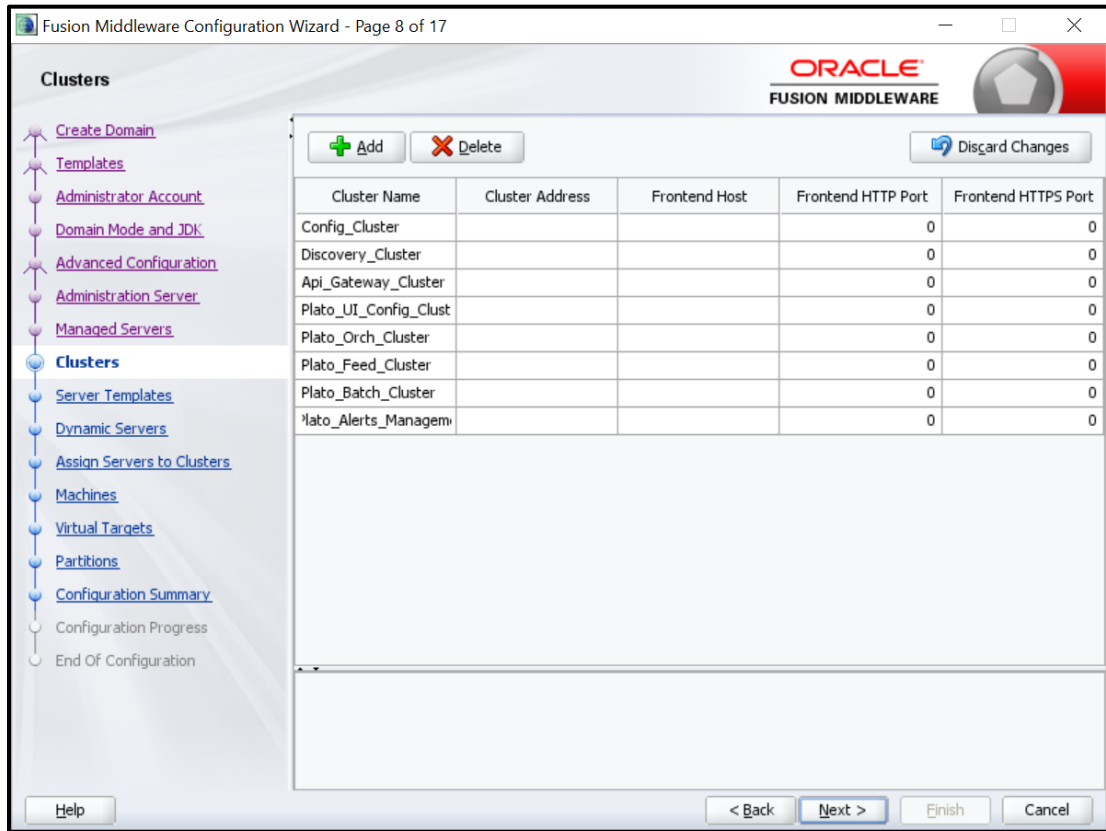
- Edit the port and host configurations as required and click **Next**.



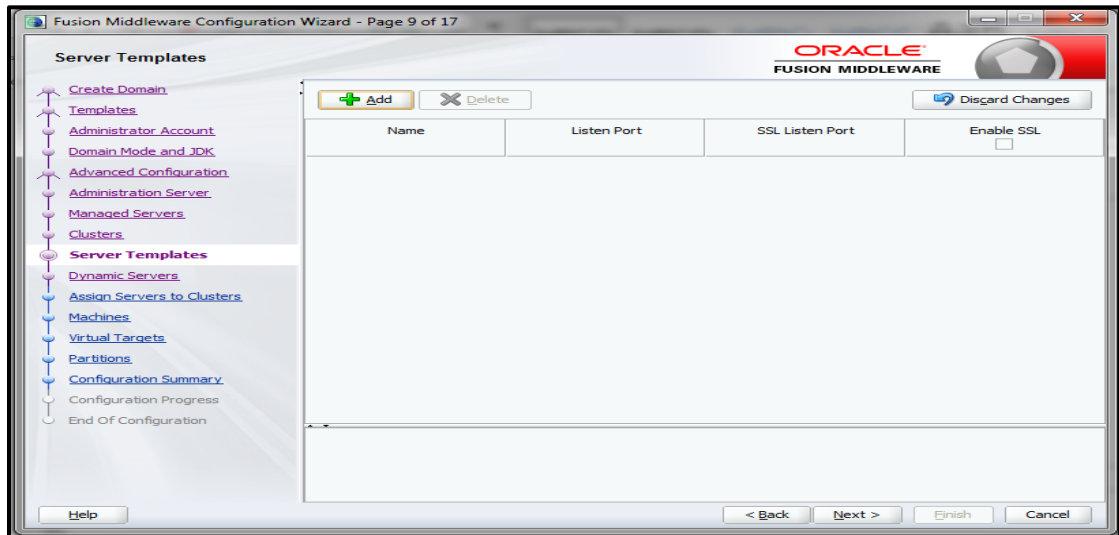
- Add managed servers and provide meaningful **Server Name**, edit listen address and port as required.



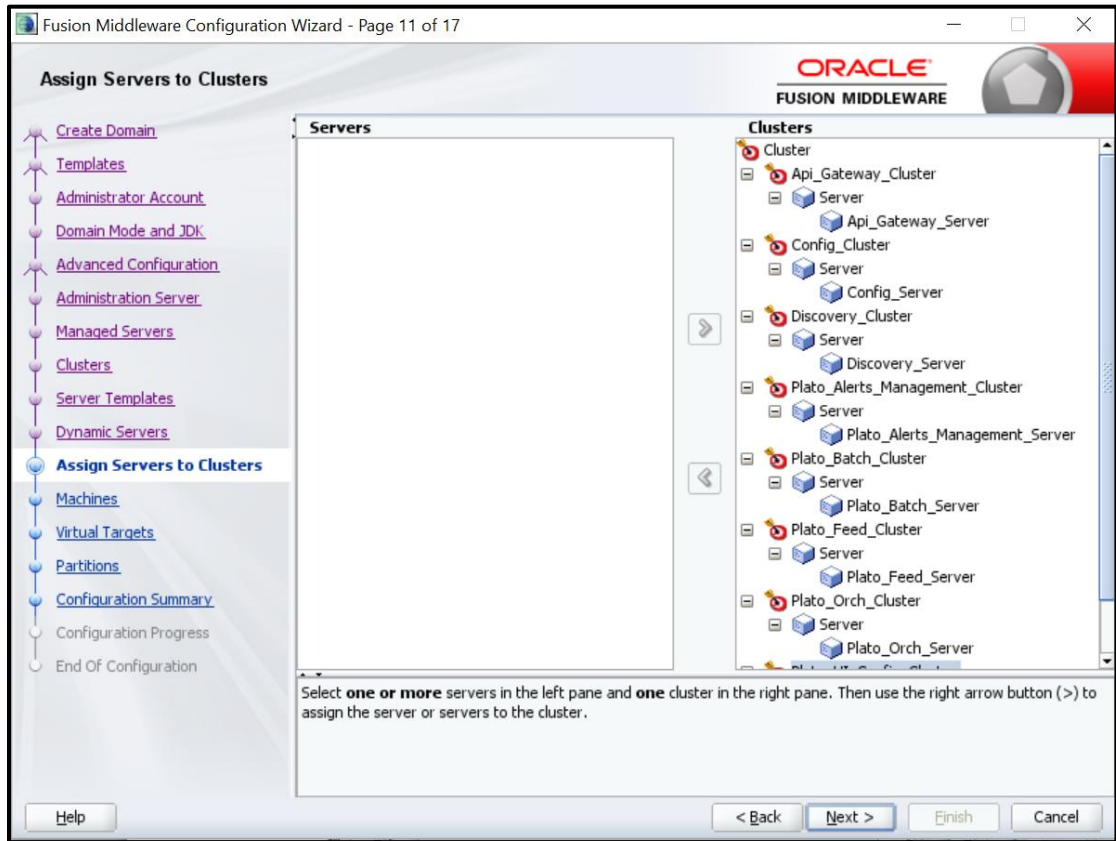
9. Add clusters one for each **managed servers**.



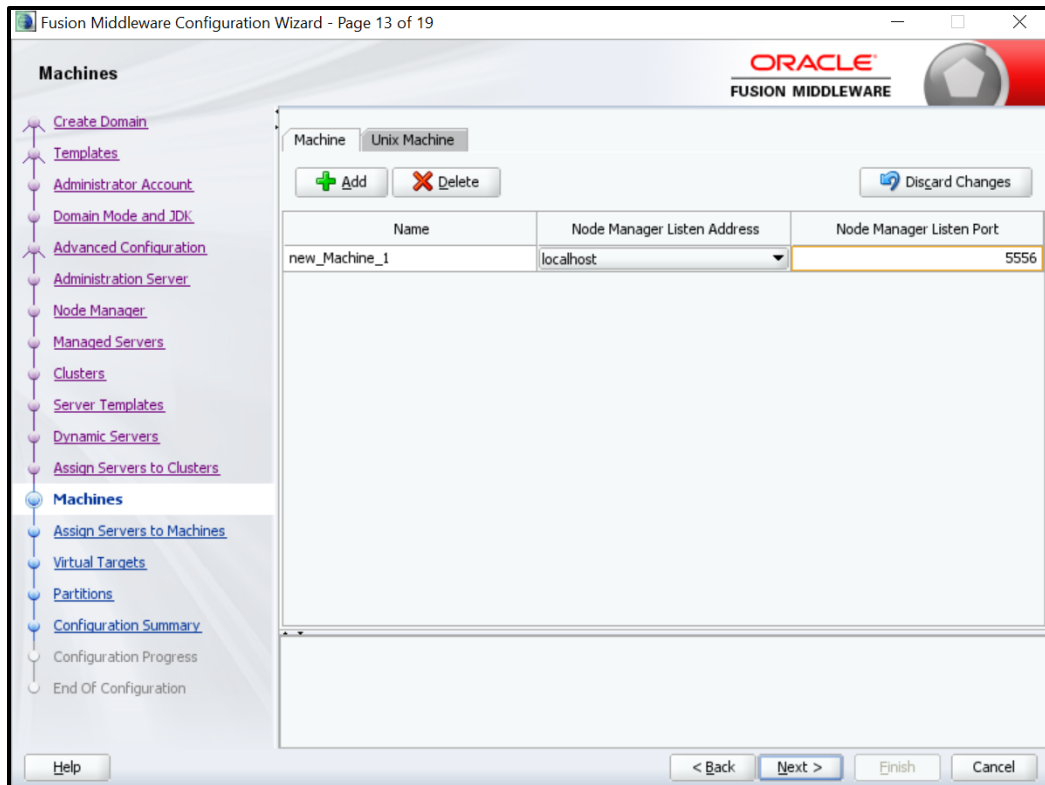
10. Skip **Server Templates** and **Dynamic Servers**.



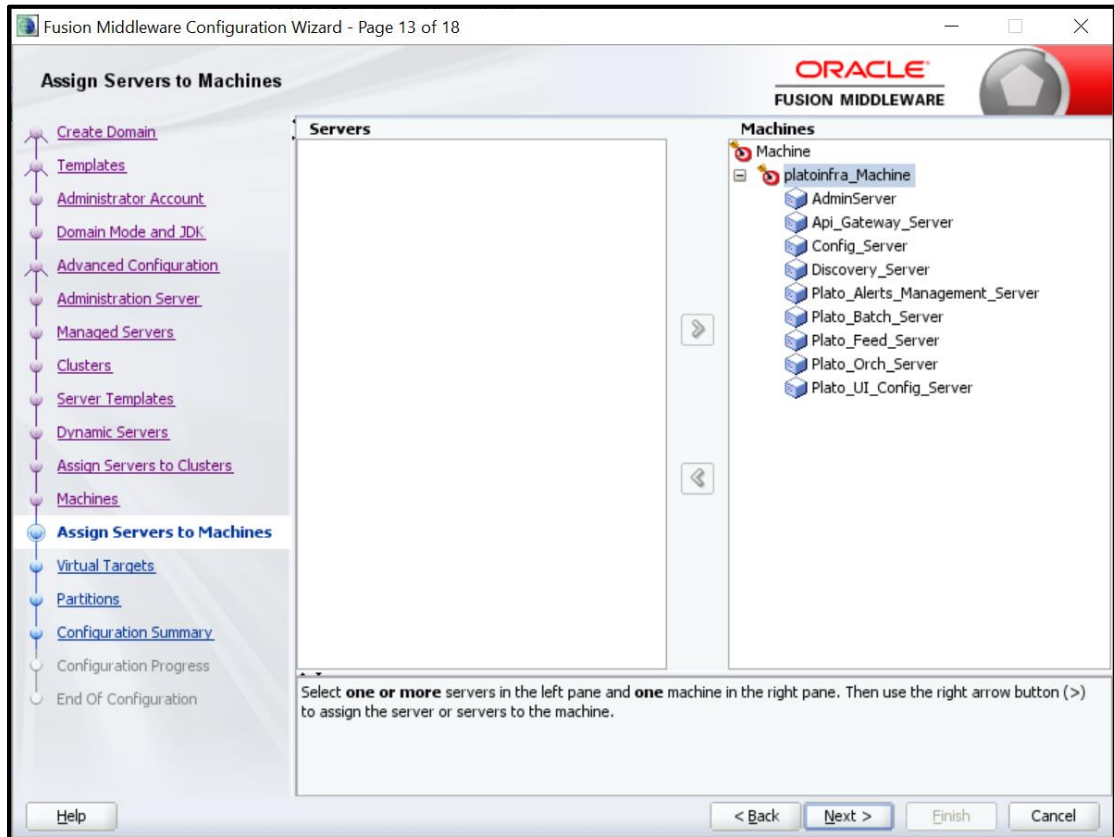
11. Assign clusters with servers.



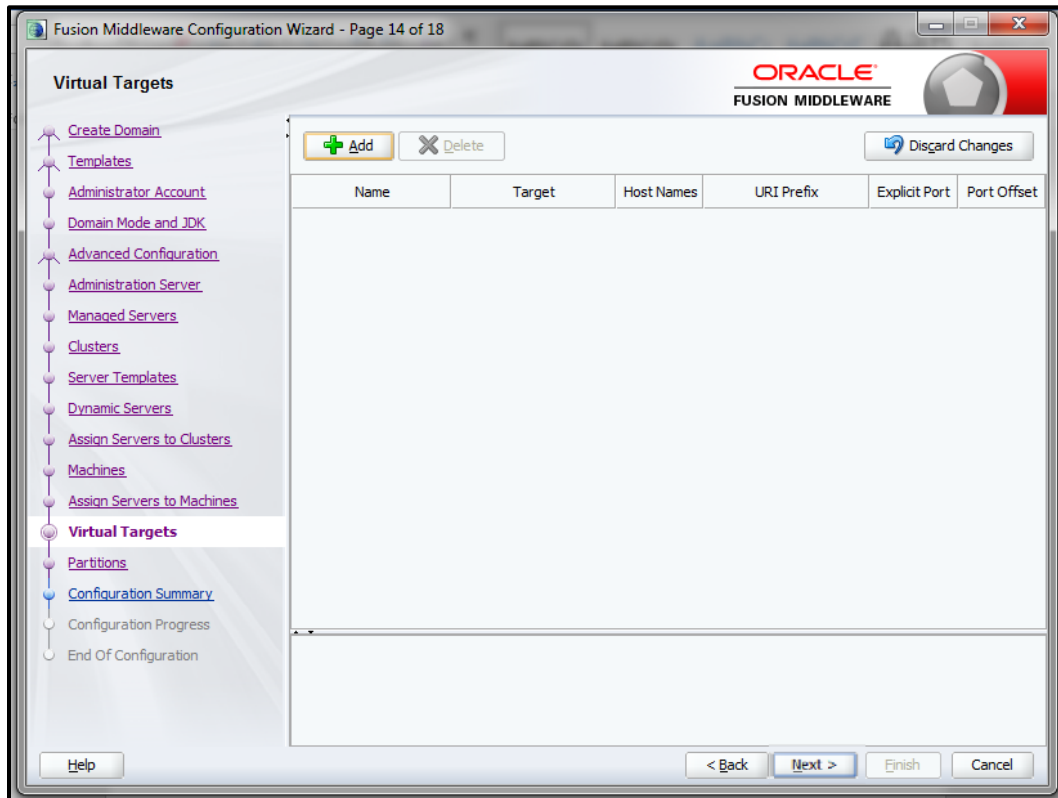
12. Add **Machine/Unix Machine** based on operating system and configure **Name**, **Node Manager Listen Address** and **Node Manager Listen Port** as required.

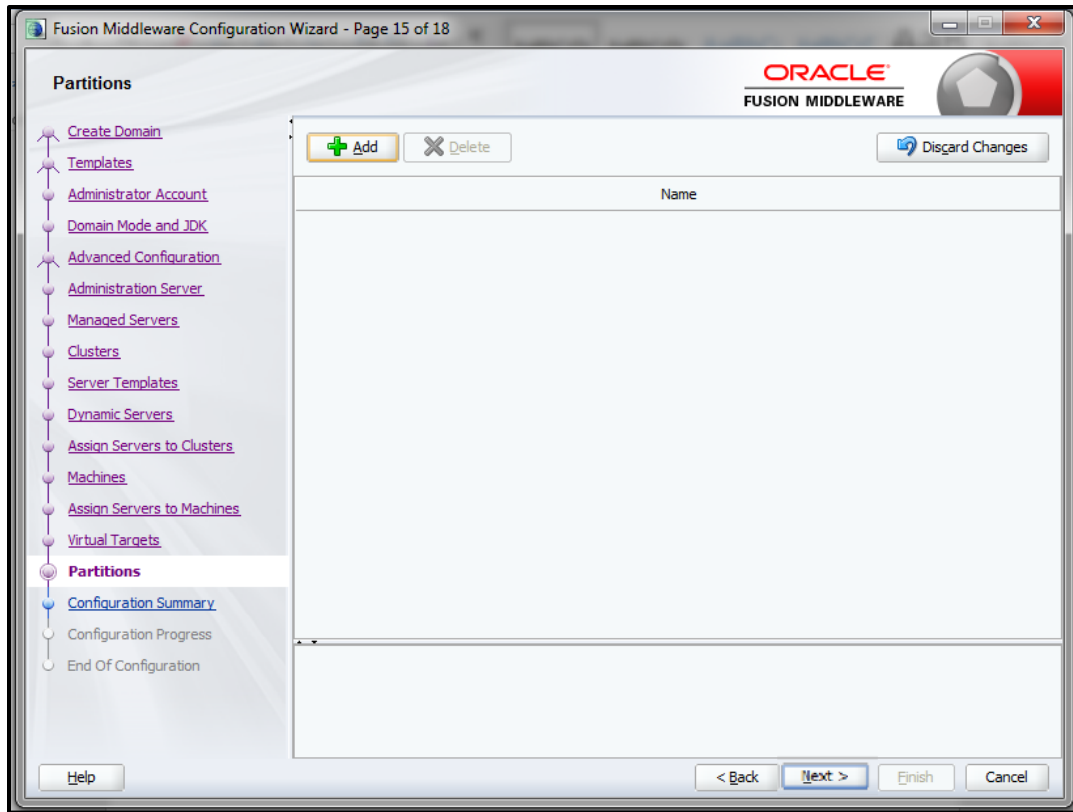


13. Map all managed servers under the machine created.

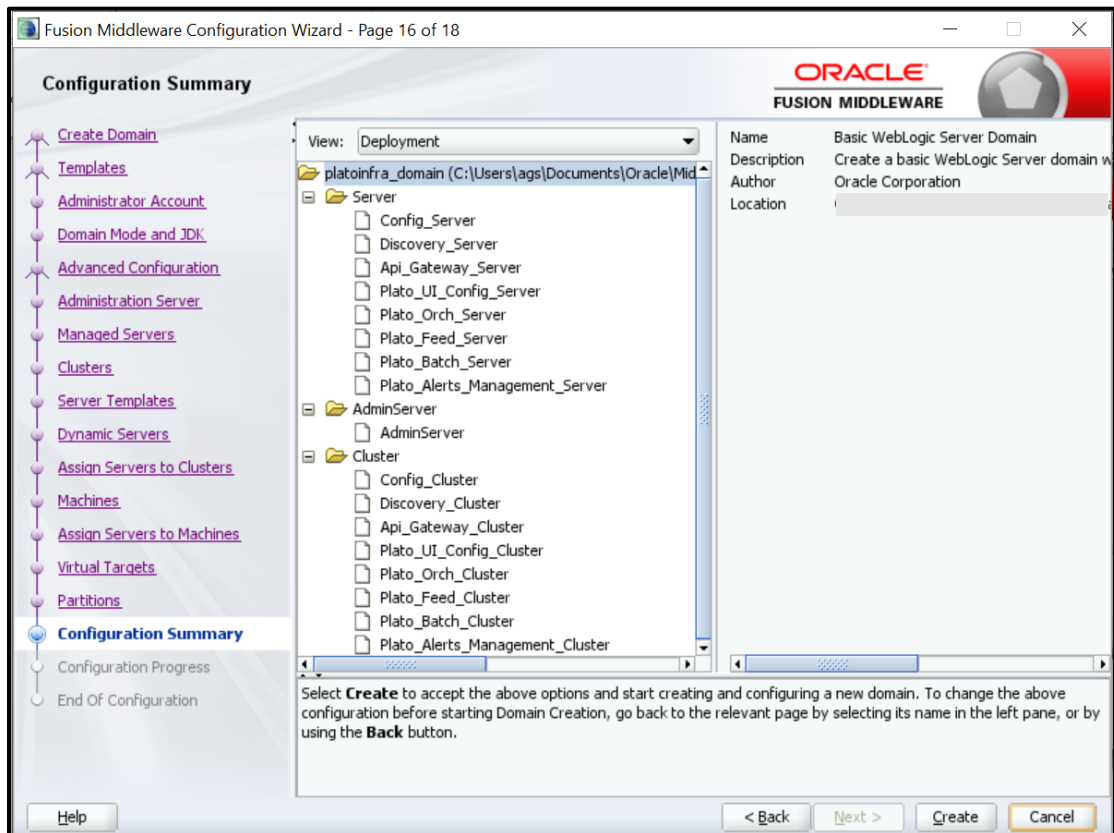


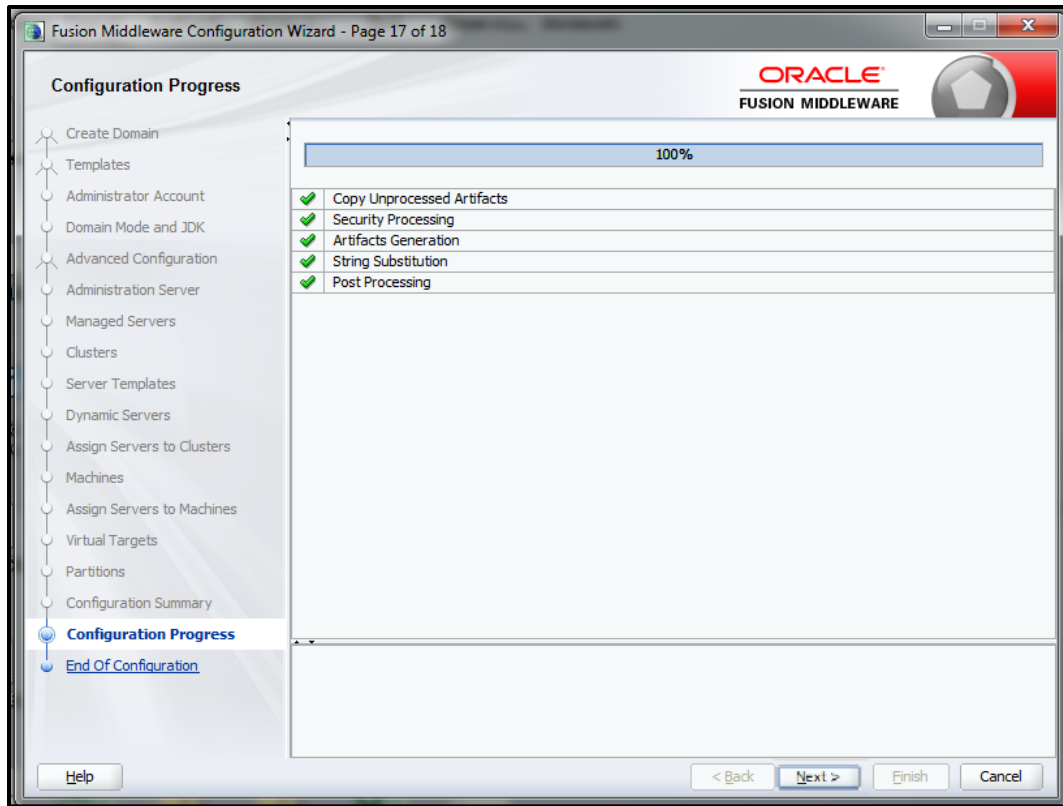
14. Skip or configure **Virtual Targets** and **Partitions** as required.



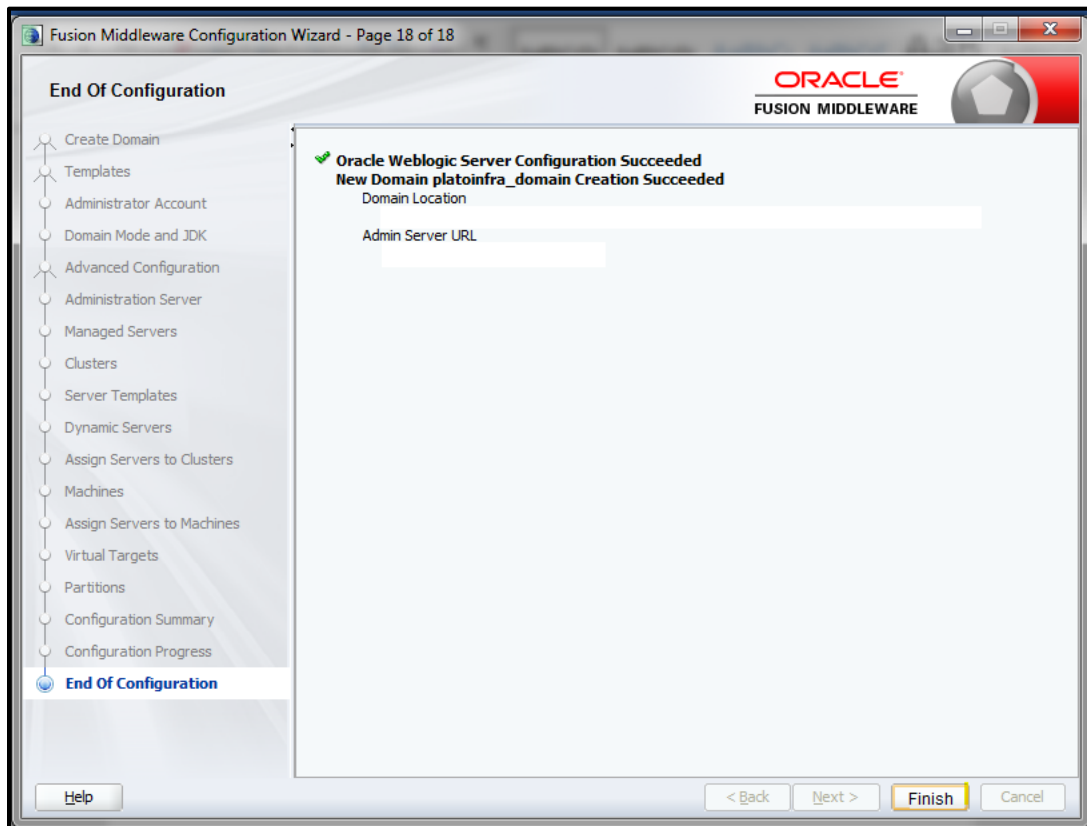


15. Check the **Configuration Summary** and confirm creating domain.





16. Click **Finish** to complete the procedure.



## 1.3.2 Post Domain Creation Configurations

Once finished, refer oracle fusion middleware documents for more details on how to start admin server, node manager and managed servers.

1. Open `/user_projects/domain/platoinfra_domain/bin`
2. Perform all the Environment Setup steps such as setting -D parameters, Embedded Weblogic Setup and changes required for OAuth.
3. Run `startWeblogic.cmd` (or `.sh` if operating system is linux).
4. Open `/user_projects/domains/platoinfra_domain/bin`.
5. Run `setNMJavaHome.cmd` (or `.sh` if operating system is linux).
6. Open `/user_projects/domains/platoinfra_domain/nodemanager`.
7. Edit `nodemanager.properties` as required (securelistner = false if ssl and keystore is not given).
8. In admin console, select the following options in sequential order:
  - a. **Machines**
  - b. **platoinfra\_Machine**
  - c. **Node Manager**
  - d. **Type**
  - e. **Plain**
  - f. **Save**
9. Open `/user_projects/domains/platoinfra_domain/bin`.
10. Run `startNodeManager.cmd` (or `.sh` if operating system is linux )
11. Start all managed servers.
12. Login to console and verify servers and clusters. Refer to the screenshots below:

The screenshot displays the Oracle WebLogic Administration Console interface. On the left, the 'Domain Structure' tree is visible, showing the hierarchy from 'Domain Partitions' down to 'Servers'. The 'Servers' folder is expanded, showing various server types like 'AdminServer', 'Apl\_Gateway\_Server', etc. The main content area shows the 'Configuration' tab for the 'Servers' page. It includes a table of servers with the following data:

Name	Type	Cluster	Machine	State	Health	Listen Port
AdminServer(admin)	Configured		platoinfra_Machine	RUNNING	OK	7001
Apl_Gateway_Server	Configured	Apl_Gateway_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7005
Config_Server	Configured	Config_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7003
Discovery_Server	Configured	Discovery_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7004
Plato_Alerts_Management_Server	Configured	Plato_Alerts_Management_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7010
Plato_Batch_Server	Configured	Plato_Batch_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7009
Plato_Feed_Server	Configured	Plato_Feed_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7008
Plato_Orch_Server	Configured	Plato_Orch_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7007
Plato_UI_Config_Server	Configured	Plato_UI_Config_Cluster	platoinfra_Machine	SHUTDOWN	Not reachable	7006

**Change Center** Home Log Out Preferences [Add](#) [Record](#) [Help](#)   Welcome, Connected to: platoinfra\_domain

Home > Summary of Deployments > Summary of Servers > Summary of Clusters

**Summary of Clusters**

This page summarizes the clusters that have been configured in the current WebLogic Server domain.

A cluster defines groups of WebLogic Server servers that work together to increase scalability and reliability.

**Customize this table**

**Clusters (Filtered - More Columns Exist)**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

Showing 1 to 8 of 8 Previous | Next

<input type="checkbox"/>	Name <input type="text"/>	Cluster Address	Cluster Messaging Mode	Migration Basis	Default Load Algorithm	Replication Type	Cluster Broadcast Channel	S
<input type="checkbox"/>	Apl_Gateway_Cluster		Unicast	Database	Round Robin	(None)		A
<input type="checkbox"/>	Config_Cluster		Unicast	Database	Round Robin	(None)		C
<input type="checkbox"/>	Discovery_Cluster		Unicast	Database	Round Robin	(None)		D
<input type="checkbox"/>	Plato_Alerts_Management_Cluster		Unicast	Database	Round Robin	(None)		P
<input type="checkbox"/>	Plato_Batch_Cluster		Unicast	Database	Round Robin	(None)		P
<input type="checkbox"/>	Plato_Feed_Cluster		Unicast	Database	Round Robin	(None)		P
<input type="checkbox"/>	Plato_Orch_Cluster		Unicast	Database	Round Robin	(None)		P
<input type="checkbox"/>	Plato_UI_Config_Cluster		Unicast	Database	Round Robin	(None)		P

Showing 1 to 8 of 8 Previous | Next

**Change Center** Home Log Out Preferences [Add](#) [Record](#) [Help](#)   Welcome, Connected to: platoinfra\_domain

Home > Summary of Deployments > Summary of Servers > Summary of Clusters > Summary of Machines

**Summary of Machines**

A machine is the logical representation of the computer that hosts one or more WebLogic Server instances (servers). WebLogic Server uses configured machine names to determine the optimum server in a cluster to which certain tasks, such as HTTP session replication, are delegated. The Administration Server uses the machine definition in conjunction with Node Manager to start remote servers.

This page displays key information about each machine that has been configured in the current WebLogic Server domain.

**Customize this table**

**Machines**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

Showing 1 to 1 of 1 Previous | Next

<input type="checkbox"/>	Name <input type="text"/>	Type
<input type="checkbox"/>	platoinfra_Machine	Machine

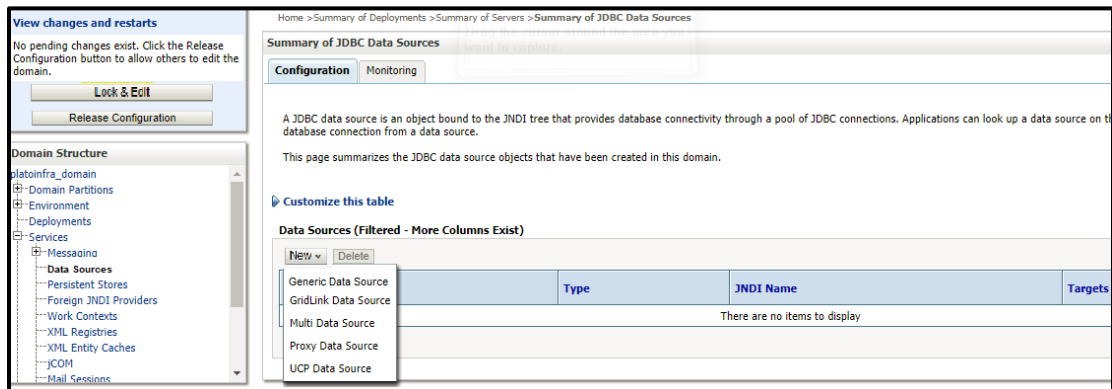
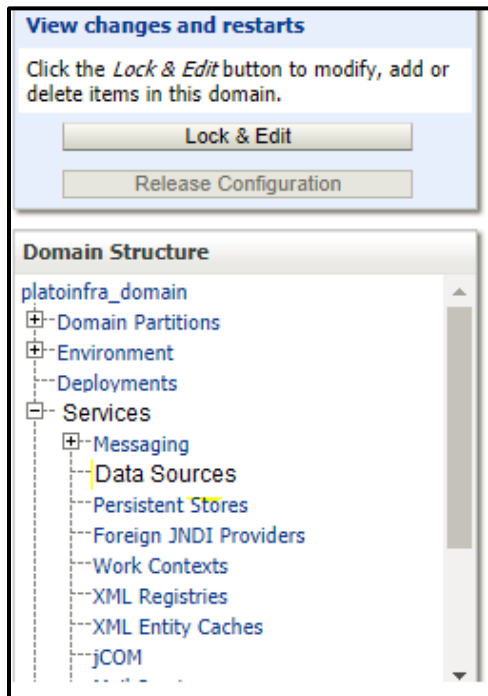
Showing 1 to 1 of 1 Previous | Next



## 1.4 How to Create Datasource

Perform the following steps to create data source:

1. Start **AdminServer**, **Node Manager** and make sure all the **managed servers** (targets) are in running mode.
2. Select the following options in sequential order:
  - a. **Services**
  - b. **Datasources**
  - c. **New**
  - d. **Generic Datasource**



3. Give datasource **Name** and **JNDI Name**, and click **Next**.

**Create a New JDBC Data Source**

Back Next Finish Cancel

**JDBC Data Source Properties**

The following properties will be used to identify your new JDBC data source.  
\* Indicates required fields

What would you like to name your new JDBC data source?

**\* Name:** PLATO

What scope do you want to create your data source in ?

**Scope:** Global ▼

What JNDI name would you like to assign to your new JDBC Data Source?

**JNDI Name:**  
jdbc/PLATO

What database type would you like to select?

**Database Type:** Oracle ▼

Back Next Finish Cancel

4. Select **Thin for Service Connections (Instant)** and click **Next**.

**Create a New JDBC Data Source**

Back Next Finish Cancel

**JDBC Data Source Properties**

The following properties will be used to identify your new JDBC data source.

**Database Type:** Oracle

What database driver would you like to use to create database connections? Note: \* indicates that the driver is explicitly supported by Oracle WebLogic Server.

**Database Driver:** \*Oracle's Driver (Thin) for Service connections; Versions:Any ▼

Back Next Finish Cancel

5. Uncheck support for Global Transactions.

Home > Summary of Deployments > Summary of Servers > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

**Transaction Options**

You have selected non-XA JDBC driver to create database connection in your new data source.

Does this data source support global transactions? If yes, please choose the transaction protocol for this data source.

**Supports Global Transactions**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the *Logging Last Resource (LLR)* transaction option. Emulate Two-Phase Commit.

**Logging Last Resource**

Select this option if you want to enable non-XA JDBC connections from the data source to emulate participation in global transactions using JTA. Select this option only if your application conditions.

**Emulate Two-Phase Commit**

Select this option if you want to enable non-XA JDBC connections from the data source to participate in global transactions using the one-phase commit transaction processing. With this option, the data source can participate in the global transaction.

**One-Phase Commit**

Back Next Finish Cancel

6. Give database connection details and click **Next** to test connection.

Home > Summary of Deployments > Summary of Servers > Summary of JDBC Data Sources

Create a New JDBC Data Source

Back Next Finish Cancel

**Connection Properties**

Define Connection Properties.

What is the name of the database you would like to connect to?

**Database Name:**

What is the name or IP address of the database server?

**Host Name:**

What is the port on the database server used to connect to the database?

**Port:**

What database account user name do you want to use to create database connections?

**Database User Name:**

What is the database account password to use to create database connections?

**Password:**

**Confirm Password:**

Additional Connection Properties:

**oracle.jdbc.DRCPConnectionClass:**

Home > Summary of Deployments > Summary of Servers > Summary of JDBC Data Sources

**Messages**

✔ Connection test succeeded.

**Create a New JDBC Data Source**

Test Configuration | Back | Next | Finish | Cancel

**Test Database Connection**

Test the database availability and the connection properties you provided.

What is the full package name of JDBC driver class used to create database connections in the connection pool?  
(Note that this driver class must be in the classpath of any server to which it is deployed.)

**Driver Class Name:**

7. Select targets to deploy data source.

The screenshot shows the Oracle WebLogic Console interface. On the left is a navigation tree with 'platoinfra\_domain' selected. The main area is titled 'Servers' and lists several clusters with their target selection options:

- AdminServer**:  AdminServer
- Api\_Gateway\_Cluster**:  All servers in the cluster,  Part of the cluster,  Api\_Gateway\_Server
- Config\_Cluster**:  All servers in the cluster,  Part of the cluster,  Config\_Server
- Discovery\_Cluster**:  All servers in the cluster,  Part of the cluster,  Discovery\_Server
- Plato\_Alerts\_Management\_Cluster**:  All servers in the cluster,  Part of the cluster,  Plato\_Alerts\_Management\_Server
- Plato\_Batch\_Cluster**:  All servers in the cluster,  Part of the cluster,  Plato\_Batch\_Server
- Plato\_Feed\_Cluster**:  All servers in the cluster,  Part of the cluster,  Plato\_Feed\_Server

At the bottom left, there is a 'System Status' section showing 'Health of Running Servers as of 6:15 PM' with a bar chart indicating 1 OK server.

8. View created Data Sources, verify JNDI Name and Targets.

Summary of JDBC Data Sources

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 1 of 1 | Previous | Next

<input type="checkbox"/>	Name	Type	JNDI Name	Targets
<input type="checkbox"/>	PLATO	Generic	jdbc/PLATO	Api_Gateway_Server, Config_Server, Discovery_Server, Plato_Alerts_Management_Server, Plato_Batch_Server, Plato_Feed_Server, ...

New | Delete | Showing 1 to 1 of 1 | Previous | Next

- Click **Activate Changes** after confirming details.

**Change Center**

**View changes and restarts**

Pending changes exist. They must be activated to take effect.

**Activate Changes**

**Undo All Changes**

**Domain Structure**

platoinfra\_domain

Domain Partitions

Home Log Out Preferences Record Help

Home > Summary of Deployments > Summary of Servers > Summary of JDBC Data Sources

**Summary of JDBC Data Sources**

**Configuration** Monitoring

A JDBC data source is an object bound to the JNDI tree that provides database connectivity from a data source.

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

**Change Center**

**View changes and restarts**

Click the **Lock & Edit** button to modify, add or delete items in this domain.

**Lock & Edit**

**Release Configuration**

**Domain Structure**

- Coherence Clusters
- Resource Groups
- Resource Group Templates
- Machines
- Virtual Hosts
- Virtual Targets
- Work Managers
- Concurrent Templates
- Resource Management
- Startup and Shutdown Classes

**Services**

- Deployments
- Messaging
- Data Sources**

**How do I...?**

- Create JDBC generic data sources
- Create JDBC GridLink data sources

Welcome, Connected to: platoinfra\_domain

Home > Summary of Deployments > Summary of Servers > Summary of Clusters > Summary of Machines > Summary of JDBC Data Sources

**Messages**

**Activate Changes**

All changes have been activated. No restarts are necessary.

**Summary of JDBC Data Sources**

**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)**

Click the **Lock & Edit** button in the Change Center to activate all the buttons on this page.

Name	Type	JNDI Name	Targets
PLATO	Generic	jdbc/PLATO	Api_Gateway_Server, Config_Server, Discovery_Server, Plato_Alerts_Management_Server, Plato_Batch_Server, Plato_Feed_Server, ...

Showing 1 to 1 of 1 Previous Next

- Post creating Data Source, click **Data source** under **Services** menu.
- Select the specific Data sources one by one.
- Navigate to **Connection Pool** tab under **Configuration** section and update the initial capacity, Minimum capacity and Max capacity as explained in the below screenshot

Configuration settings are mirrored. Future changes will automatically be attributed as you modify, add or delete items in this domain.

**Domain Structure**

- platoinfra\_domain
- Domain Partitions
- Deployments
- Messaging
- Services
- Data Sources

**How do I...?**

- Configure binding options for a JDBC data source
- Configure the statement cache for a JDBC connection pool
- Configure credential mapping for a JDBC data source
- Configure connection harvesting for a connection pool
- Storage connection properties

**System Status**

Health of Running Servers as of 12:46 PM

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

**Settings for administrators**

**Configuration** Targets Monitoring Control Security Roles

**Connection Pool** Oracle JDBC Transaction Diagnostics Identity Options

**Basic**

The connection pool within a JDBC data source contains a group of JDBC connections that applications request, use, and then return to the pool. The connection pool and the connections within it are created when the connection pool is registered, usually when starting up WebLogic Server or when deploying the data source to a new target.

Use this page to define the configuration for this data source's connection pool.

**URL:** jdbc:oracle:thin:@uhf005zz:1521:OBTFPM19CFDB

The URL of the database to connect to. The format of the URL varies by JDBC driver. [More info...](#)

**Driver Class Name:** oracle.jdbc.OracleDriver

The full package name of JDBC driver class used to create the physical database connections in the connection pool. (Note that this driver class must be in the classpath of any server to which it is deployed). [More info...](#)

**Properties:**

The list of properties passed to the JDBC driver that are used to create physical database connections. For example: server=obsemer1. List each property-value pair on a separate line. [More info...](#)

**System Properties:**

The list of system properties passed to the JDBC driver that are used to create physical database connections. For example: server=obsemer1. List each property-value pair on a separate line. [More info...](#)

**Encrypted Properties:**

The list of encrypted properties passed to the JDBC driver that are used to create physical database connections. For example: password=obsemer1. [More info...](#)

**Password:** [REDACTED]

The password attribute passed to the JDBC driver when creating physical database connections. [More info...](#)

**Confirm Password:** [REDACTED]

**Initial Capacity:** 5

The number of physical connections to create when creating the connection pool in the data source. If needed to create this number of connections, creation of the data source will fail. [More info...](#)

**Maximum Capacity:** 12

The maximum number of physical connections that this connection pool can contain. [More info...](#)

**Minimum Capacity:** 1

The minimum number of physical connections that this connection pool can contain after it is initialized. [More info...](#)

**Statement Cache Type:** LRU

The algorithm used for maintaining the prepared statements stored in the statement cache. [More info...](#)

## 1.5 How to Deploy Application

Steps to Deploy archives as application on weblogic is same for all the above except for managed server and domain, where we deploy will differ. Perform the following steps to see how deployment of archive as application is done on weblogic:

1. Navigate to left menu and select **Deployments**.

The screenshot shows the Oracle WebLogic Administration Console. On the left, the 'Domain Structure' tree is expanded to 'Deployments'. The main content area shows the 'Home Page' with sections for 'Information and Resources', 'Domain Configurations', and 'Deployed Resources'. The 'Domain Configurations' section includes links for 'Domain', 'Domain Partitions', and 'Environment'. The 'Deployed Resources' section includes a link for 'Deployments'.

2. Click on **Lock and Edit** and then click **Install**.

The screenshot shows the 'Summary of Deployments' page. The 'Configuration' tab is active. The page displays instructions on how to update, delete, or install applications and modules. Below the instructions, there is a table with columns for Name, State, Health, Type, Targets, Scope, and Domain Partitions. The table is currently empty, displaying 'There are no items to display'.

The screenshot shows the 'Install Application Assistant' dialog. The 'Locate deployment to install and prepare for deployment' step is active. The dialog prompts the user to select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. The 'Path' field is empty, and the 'Current Location' is set to 'localhost \'. There are 'Back', 'Next', 'Finish', and 'Cancel' buttons at the top.

3. Click **Upload your file(s)** to select archive, **Choose File** and click **Next**.

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Lock & Edit  
Release Configuration

**Domain Structure**

- platoinfra\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
  - Security Realms
  - Interoperability
  - Diagnostics

**Install Application Assistant**

Back Next Finish Cancel

**Upload a deployment to the Administration Server**

Click the Browse button below to select an application or module on the machine from which you are currently browsing. When you have located the file, click the Next button to upload this deployment to the Administration Server.

**Deployment Archive:** Choose File No file chosen

**Upload a deployment plan (this step is optional)**

A deployment plan is a configuration which can supplement the descriptors included in the deployment archive. A deployment will work without a deployment plan archive now. This deployment plan archive will be a directory of configuration information packaged as a .jar file. See related links for additional information.

**Deployment Plan Archive:** Choose File No file chosen

Back Next Finish Cancel

4. After archive is uploaded, click **Next**.

Home Log Out Preferences Record Help Welcome, Connected to: platoinfra\_domain

Home > Summary of Deployments > Summary of Servers > Summary of Clusters > Summary of Machines > Summary of JDBC Data Sources > Summary of Deployments

**Install Application Assistant**

Back Next Finish Cancel

**Upload a deployment to the Administration Server**

Click the Browse button below to select an application or module on the machine from which you are currently browsing. When you have located the file, click the Next button to upload this deployment to the Administration Server.

**Deployment Archive:** Browse... plato-discovery-service-5.1.0.war

**Upload a deployment plan (this step is optional)**

A deployment plan is a configuration which can supplement the descriptors included in the deployment archive. A deployment will work without a deployment plan, but you can also upload a deployment plan archive now. This deployment plan archive will be a directory of configuration information packaged as a .jar file. See related links for additional information about deployment plans.

**Deployment Plan:** Browse... No file selected.

Back Next Finish Cancel

Home Log Out Preferences Record Help Welcome, Connected to: platoinfra\_domain

Home > Summary of Deployments > Summary of Servers > Summary of Clusters > Summary of Machines > Summary of JDBC Data Sources > Summary of Deployments

**Messages**

✔ The file plato-discovery-service-5.1.0.war has been uploaded successfully to C:\Users\ags\Documents\Oracle\Middleware\Oracle\_Home\user\_projects\domains\platoinfra\_domain\servers\AdminServer\upload

**Install Application Assistant**

Back Next Finish Cancel

**Locate deployment to install and prepare for deployment**

Select the file path that represents the application root directory, archive file, exploded archive directory, or application module descriptor that you want to install. You can also enter the path of the application directory or file in the Path field.

**Note:** Only valid file paths are displayed below. If you cannot find your deployment files, Upload your file(s) and/or confirm that your application contains the required deployment descriptors.

**Path:** \_\_\_\_\_

**Recently Used Paths:** (none)

**Current Location:** \_\_\_\_\_

plato-discovery-service-5.1.0.war

Back Next Finish Cancel

5. Select option **Install this deployment as an application** and click **Next**.

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

**Domain Structure**

- temp\_domain
  - Domain Partitions
  - Environment
  - Deployments
  - Services
  - Security Realms
  - Interoperability
  - Diagnostics

**Install Application Assistant**

Back Next Finish Cancel

**Choose installation type and scope**

Select if the deployment should be installed as an application or library. Also decide the scope of this deployment.

The application and its components will be targeted to the same locations. This is the most common usage.

**Install this deployment as an application**

Application libraries are deployments that are available for other deployments to share. Libraries should be available on all of the targets running their referencing applications.

**Install this deployment as a library**

Select a scope in which you want to install the deployment.

**Scope:** Global

Back Next Finish Cancel

6. Select target servers/clusters on which application has to be deployed and the **Next**.

The screenshot shows the 'Available targets for plato-discovery-service-5.1.0' dialog. On the left, the 'Domain Structure' tree is visible, showing a hierarchy from 'platoinfra\_domain' down to 'Resource Management'. The main area is divided into sections for 'Servers', 'Clusters', and 'Discovery\_Cluster'. Under 'Clusters', the 'Discovery\_Cluster' is checked, and within it, 'Part of the cluster' and 'Discovery\_Server' are selected. Other clusters like 'Api\_Gateway\_Cluster' and 'Plato\_Alerts\_Management\_Cluster' are not selected.

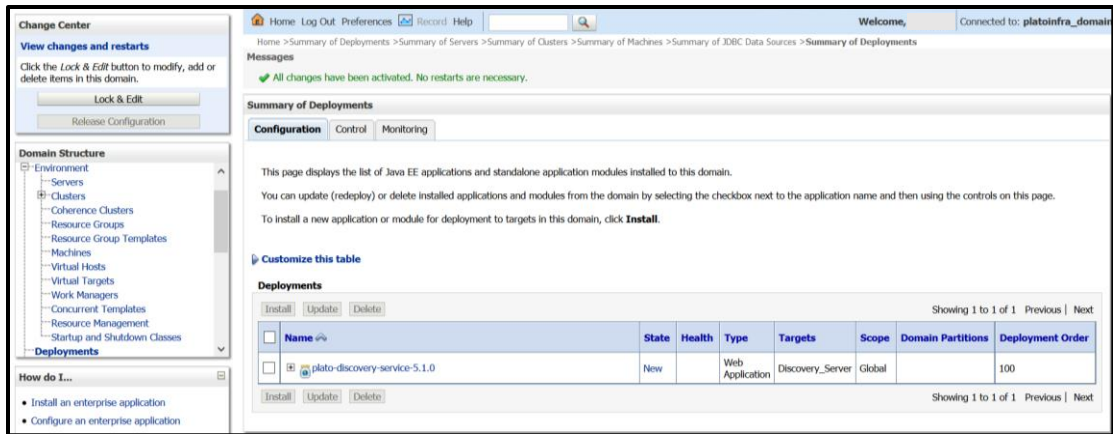
The screenshot shows the 'Install Application Assistant' dialog. The 'Name' field is filled with 'plato-discovery-service-5.1.0'. Under the 'Security' section, the 'DD Only: Use only roles and policies that are defined in the deployment descriptors.' radio button is selected. Other options like 'Custom Roles' and 'Advanced' are not selected. The 'Source Accessibility' section is also visible.

7. Click **Finish** and then click **Save and Activate Changes**.

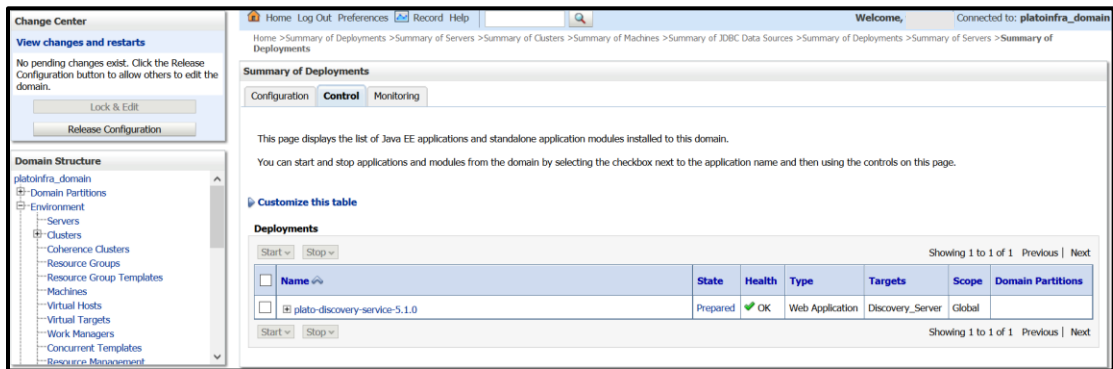
The screenshot shows the 'Summary of Deployments' page. The 'plato-discovery-service-5.1.0' application is listed in a table. The table has columns for Name, State, Health, Type, Targets, Scope, Domain Partitions, and Deployment Order. The application is in the 'Installing' state. Below the table, there are 'Install', 'Update', and 'Delete' buttons for each application.

Name	State	Health	Type	Targets	Scope	Domain Partitions	Deployment Order
plato-discovery-service-5.1.0	Installing	distribute	Web Application	Discovery_Server	Global		100

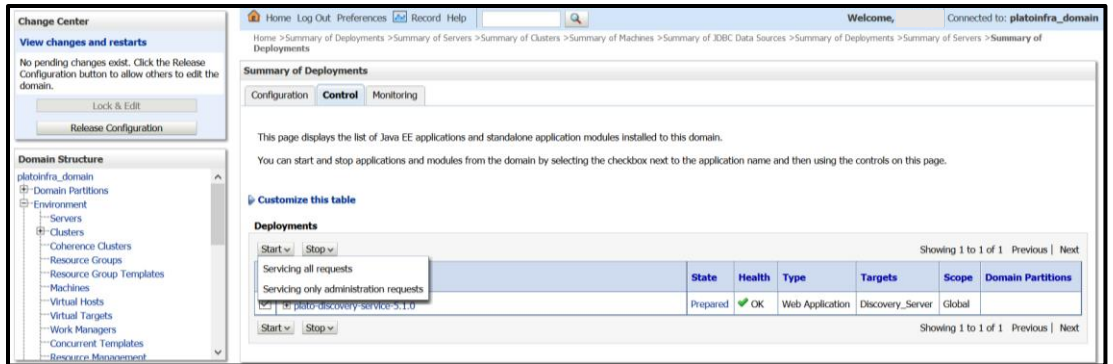




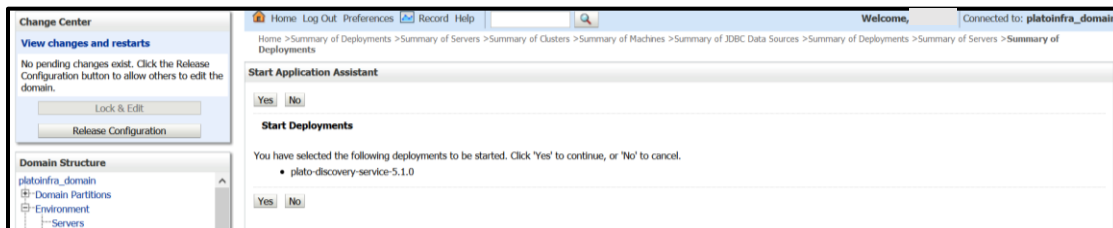
- Click **Deployments** and then **Control** to changes the state of application from prepared to active status.



- Under **Deployment**, click **Start** dropdown and select **Start all requests**.



- Click **Yes**.



11. The status is displayed as **Active** in the state column.

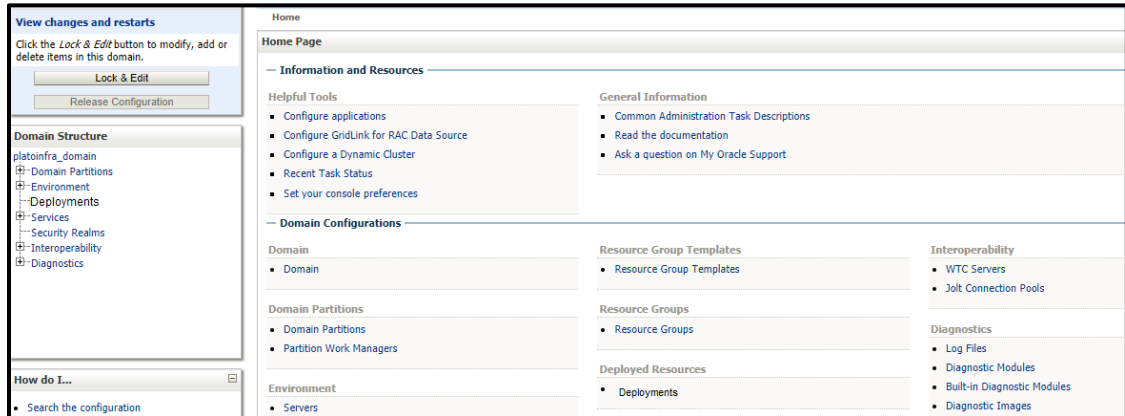
The screenshot shows the Oracle JBoss Management Console interface. On the left is the 'Change Center' and 'Domain Structure' sidebar. The main content area is titled 'Summary of Deployments' and includes a 'Configuration' tab. Below the introductory text is a table of deployments. The table has columns for Name, State, Health, Type, Targets, Scope, Domain Partitions, and Deployment Order. One deployment is listed: 'plato-discovery-service-5.1.0' with a state of 'Active' and a health status of 'OK'.

<input type="checkbox"/>	Name	State	Health	Type	Targets	Scope	Domain Partitions	Deployment Order
<input type="checkbox"/>	plato-discovery-service-5.1.0	Active	OK	Web Application	Discovery_Server	Global		100

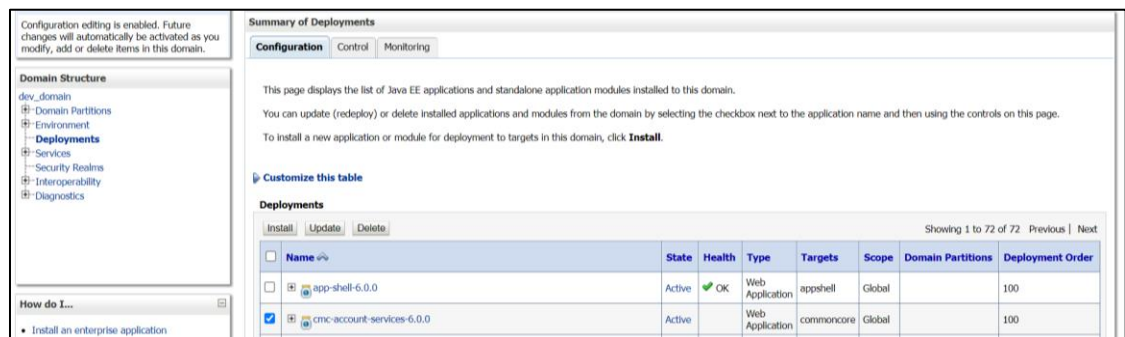
## 1.6 How to Undeploy Application

Login into weblogic server with the proper credentials.

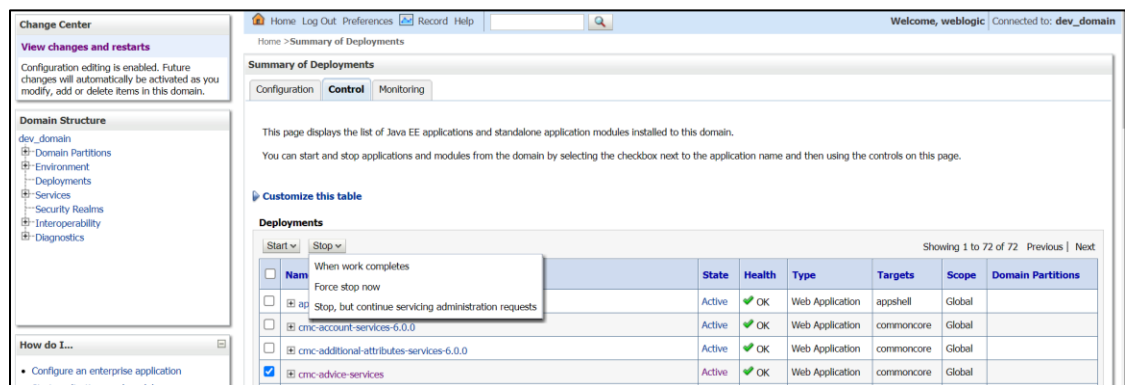
1. Navigate to left menu and select **Deployments**.



2. Click **Lock and Edit** and then select the service that needs to be undeployed in Deployments.



3. On **Control** tab, click **Stop**, and select **Force stop now** from the dropdown list.



- Once it changes to prepared state, click **Configuration** tab.

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

**Summary of Deployments**

Configuration | Control | Monitoring

This page displays the list of Java EE applications and standalone application modules installed to this domain. You can update (redeploy) or delete installed applications and modules from the domain by selecting the checkbox next to the application name and then using the controls on this page. To install a new application or module for deployment to targets in this domain, click **Install**.

Customize this table

Install | Update | Delete

Showing 1 to 72 of 72 Previous | Next

Name	State	Health	Type	Targets	Scope	Domain Partitions	Deployment Order
app-shell-6.0.0	Active	OK	Web Application	appshell	Global		100
cmc-account-services-6.0.0	Active		Web Application	commoncore	Global		100
cmc-additional-attributes-services-6.0.0	Active		Web Application	commoncore	Global		100
cmc-advice-services	Prepared		Web Application	commoncore	Global		100

- Select the service again and click on Delete to undeploy the service.

## 1.7 How to Restart Servers

Perform the following steps to restart servers:

- Navigate to left menu and select **Environment**, and then click **Servers**.

**Change Center**

View changes and restarts

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

**Domain Structure**

- temp\_domain
  - Domain Partitions
  - Environment
    - Servers**
    - Clusters
      - Coherence Clusters
      - Resource Groups
      - Resource Group Templates
      - Machines
      - Virtual Hosts
      - Virtual Targets
      - Work Managers
      - Concurrent Templates
      - Resource Management

2. Click **Control** tab.

The screenshot shows the Oracle WebLogic Server console. The 'Summary of Servers' page is active, and the 'Control' tab is selected. The 'Discovery\_Server' is highlighted in the table below.

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Api_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Discovery_Server	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Plato_Alerts_Management_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Plato_Batch_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Plato_Feed_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Plato_Orch_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Plato_UI_Config_Server	platoinfra_Machine	SHUTDOWN	None

3. Select servers to **Shutdown**

The screenshot shows the Oracle WebLogic Server console. The 'Summary of Servers' page is active, and the 'Control' tab is selected. The 'Discovery\_Server' is highlighted in the table, and the 'Shutdown' dropdown menu is open, showing options: 'When work completes' and 'Force shutdown now'.

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Api_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input checked="" type="checkbox"/> Discovery_Server	platoinfra_Machine	RUNNING	None

4. Click **Yes** to confirm shutdown.

The screenshot shows the Oracle WebLogic Server console. The 'Server Life Cycle Assistant' dialog box is open, and the 'Yes' button is selected. The dialog box contains the following text:

**Forcibly Shutdown Servers**

You have selected the following servers to be immediately shut down. Press 'Yes' to continue or 'No' to cancel.

- Discovery\_Server

Yes No

5. The status displayed as shown below:

The screenshot shows the Oracle WebLogic Server console. The 'Summary of Servers' page is active, and the 'Control' tab is selected. The 'Discovery\_Server' is highlighted in the table, and its status is 'FORCE\_SHUTTING\_DOWN'.

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Api_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Discovery_Server	platoinfra_Machine	FORCE_SHUTTING_DOWN	TASK IN PROGRESS

- Once shutdown is completed, navigate to **Control**, select the servers to **Start**, and click **Yes** to confirm action.

Summary of Servers

Configuration **Control**

Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed Servers in Standby mode requires the domain-wide administration port.

Customize this table

Servers (Filtered - More Columns Exist)

Start Resume Suspend Shutdown Restart SSL Showing 1 to 9 of 9 Previous Next

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Apl_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input checked="" type="checkbox"/> Discovery_Server	platoinfra_Machine	SHUTDOWN	TASK COMPLETED

Server Life Cycle Assistant

Yes No

**Start Servers**

You have selected the following servers to be started. Press 'Yes' to continue or 'No' to cancel.

- Discovery\_Server

Yes No

Messages

✔ A request has been sent to the Node Manager to start the selected servers.

Summary of Servers

Configuration **Control**

Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed Servers in Standby mode requires the domain-wide administration port.

Customize this table

Servers (Filtered - More Columns Exist)

Start Resume Suspend Shutdown Restart SSL Showing 1 to 9 of 9 Previous Next

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Apl_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Discovery_Server	platoinfra_Machine	SHUTDOWN	TASK IN PROGRESS

- When all requested servers are running, navigate to **Summary of Deployments**, and check if deployments are in active state.

Summary of Servers

Configuration **Control**

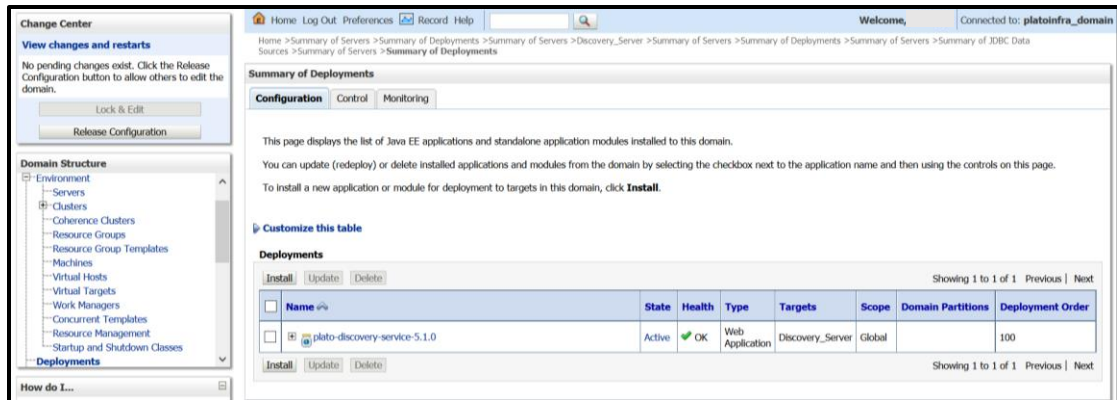
Use this page to change the state of the servers in this WebLogic Server domain. Control operations on Managed Servers require starting the Node Manager. Starting Managed Servers in Standby mode requires the domain-wide administration port.

Customize this table

Servers (Filtered - More Columns Exist)

Start Resume Suspend Shutdown Restart SSL Showing 1 to 9 of 9 Previous Next

Server	Machine	State	Status of Last Action
<input type="checkbox"/> AdminServer(admin)	platoinfra_Machine	RUNNING	None
<input type="checkbox"/> Apl_Gateway_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Config_Server	platoinfra_Machine	SHUTDOWN	None
<input type="checkbox"/> Discovery_Server	platoinfra_Machine	RUNNING	TASK COMPLETED



## 1.8 How to Check Port Number

Perform the following steps to check port numbers:

1. Specify the **User id** and **Password**, and login to **WebLogic console**.
2. Click **Environment** and then click **Server**.



3. Under Servers (Filtered - More Columns Exist) section, you will be able to see all the server listed.

Name	Type	Cluster	Machine	State	Health	Listen Port
AdminServer(admin)	Configured			RUNNING	OK	7020
managed_server1	Configured		Machine1	RUNNING	OK	7023

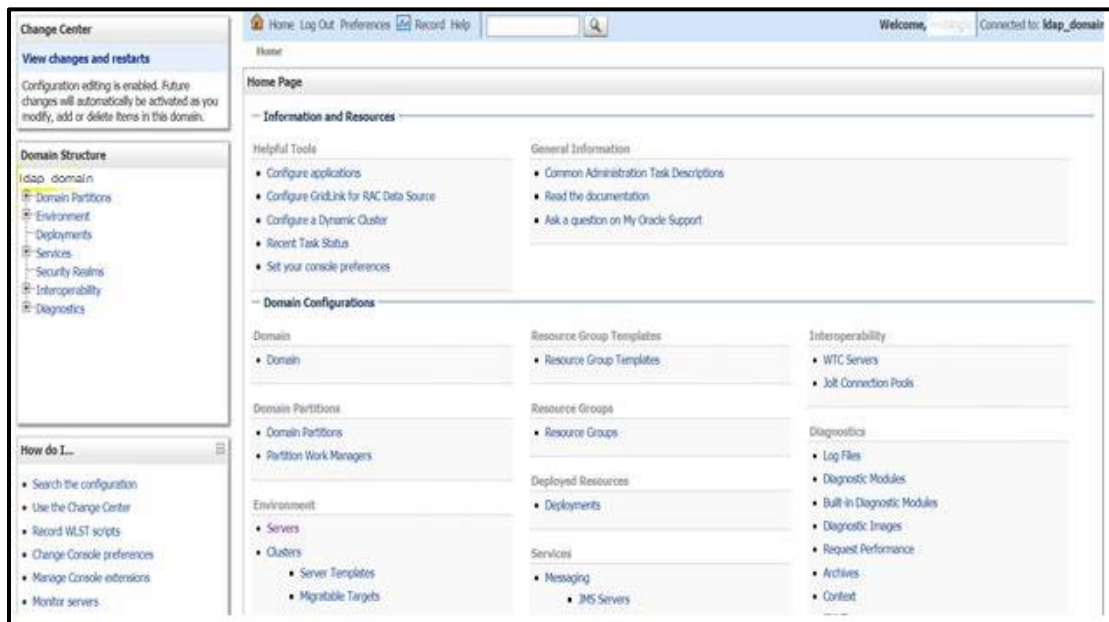
## 1.9 Weblogic Embedded LDAP Setup

The following changes are to be made for configuring the Weblogic-Embedded LDAP server for Oracle Banking Microservices Architecture:

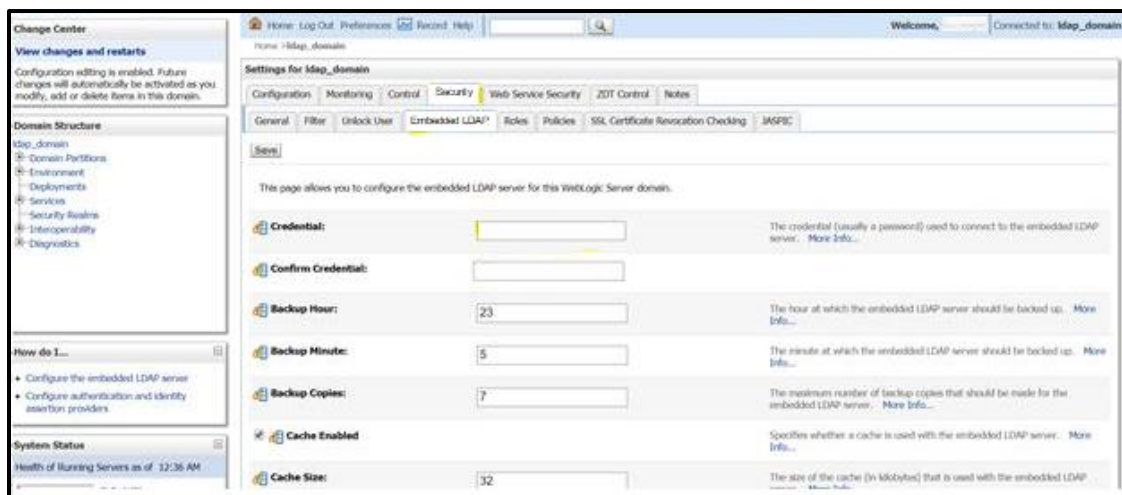
- Configuration of Weblogic LDAP
- Creation of Users
- Plato Security Config Table Entries

### 1.9.1 Configuration of Weblogic LDAP

1. Open the **Administration Console** for the **Weblogic**, and click domain name in left panel.

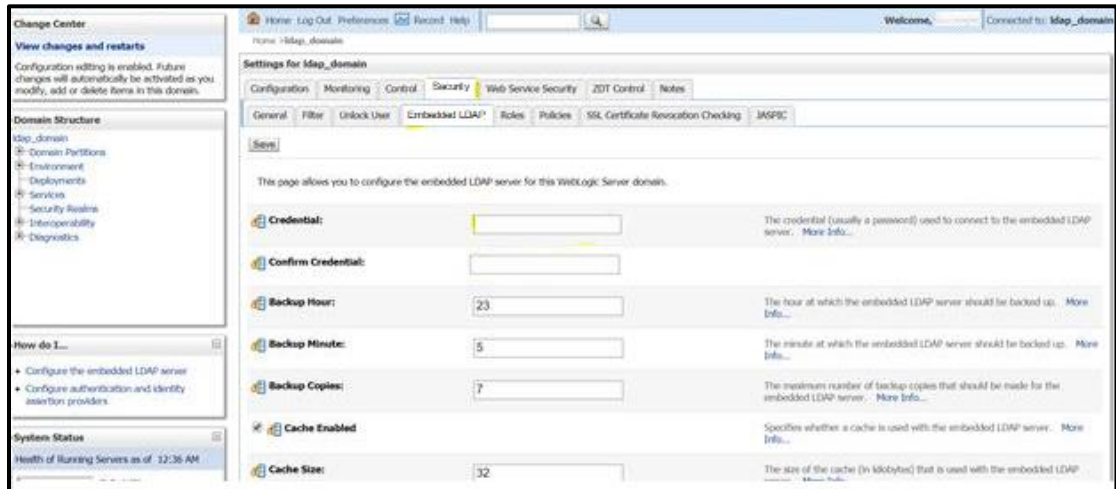


2. Under Settings for ldap\_domain, click **Security** tab, and then click **Embedded LDAP** tab.



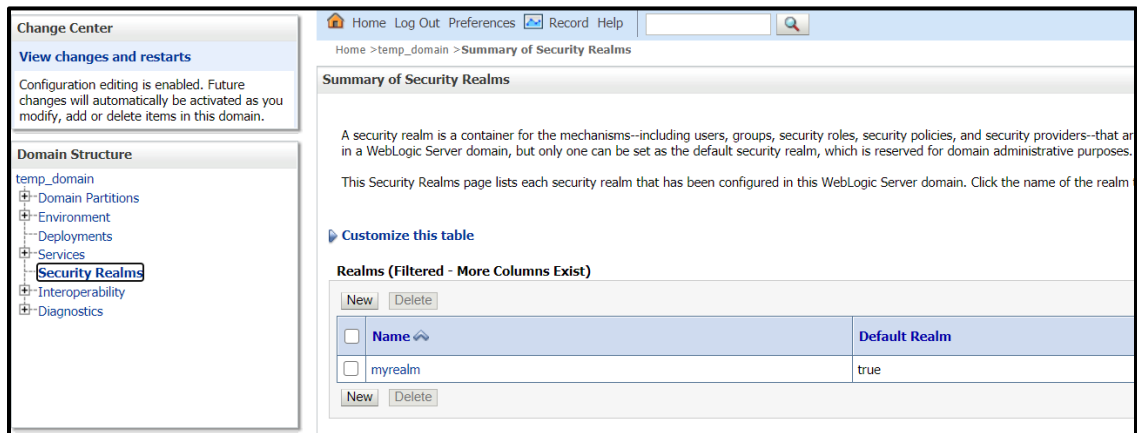


3. Set the **Credential** for Weblogic Embedded LDAP store. This is needed in the **Security Config** table.

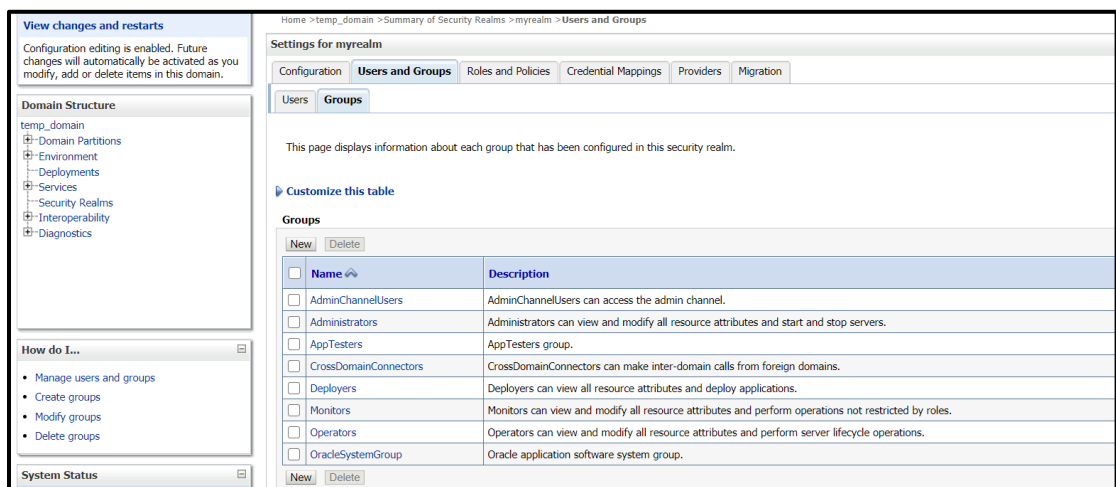


## 1.9.2 Creation of Users

1. Navigate to left menu, and click **Security Realms**.
2. In the **Summary of Security Realms** window, click **myrealm**.



3. Under **Settings for myrealm**, click **Users and Groups**.
4. Click **Groups** tab. Click **New** to make a new group.



5. Add the relevant details and click **OK**. The new group will be created.

View changes and restarts  
Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure  
temp\_domain  
 + Domain Partitions  
 + Environment  
 + Deployments  
 + Services  
 + Security Realms  
 + Interoperability  
 + Diagnostics

How do I...  
 • Create groups  
 • Modify groups  
 • Delete groups

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups

Create a New Group

OK | Cancel

Group Properties  
The following properties will be used to identify your new Group.  
\* Indicates required fields

What would you like to name your new Group?

\* Name:

How would you like to describe the new Group?

Description:

Please choose a provider for the group.

Provider:

OK | Cancel

6. Click **Users** tab, and click **New** to create user.

Change Center  
View changes and restarts  
Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure  
temp\_domain  
 + Domain Partitions  
 + Environment  
 + Deployments  
 + Services  
 + Security Realms  
 + Interoperability  
 + Diagnostics

How do I...  
 • Create users  
 • Modify users  
 • Delete users  
 • Create groups  
 • Manage users and groups

System Status  
Health of Running Servers as of 12:48 AM  
Failed (0)

Home Log Out Preferences Record Help

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

Users Groups

This page displays information about each user that has been configured in this security realm.

Customize this table

Users (Filtered - More Columns Exist)

New Delete

Name	Description
LCMUser	This is the default service account for WebLogic Server Lifecycle Manager configuration updates.
OracleSystemUser	Oracle application software system user.
weblogic	This user is the default administrator.

New Delete

7. Enter the required details for the user. After completing click **OK**. The user will be created.

Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Domain Structure  
temp\_domain  
 + Domain Partitions  
 + Environment  
 + Deployments  
 + Services  
 + Security Realms  
 + Interoperability  
 + Diagnostics

How do I...  
 • Create users  
 • Modify users  
 • Delete users  
 • Create groups  
 • Manage users and groups

System Status  
Health of Running Servers as of 12:48 AM  
Failed (0)

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups

Create a New User

OK | Cancel

User Properties  
The following properties will be used to identify your new User.  
\* Indicates required fields

What would you like to name your new User?

\* Name:

How would you like to describe the new User?

Description:

Please choose a provider for the user.

Provider:

The password is associated with the login name for the new User.

\* Password:

\* Confirm Password:

OK | Cancel

- Click on the newly created user to assign the newly created user to some group.

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups

Messages  
 ✓ User created successfully

Settings for myrealm

Configuration **Users and Groups** Roles and Policies Credential Mappings Providers Migration

**Users** Groups

This page displays information about each user that has been configured in this security realm.

[Customize this table](#)

Users (Filtered - More Columns Exist)

New Delete

<input type="checkbox"/>	Name ↕	Description
<input type="checkbox"/>	ADMINUSER1	ADMINUSER1
<input type="checkbox"/>	LCMUser	This is the default service account for WebLogic Server Lifecycle Manager configuration updates.
<input type="checkbox"/>	OracleSystemUser	Oracle application software system user.
<input type="checkbox"/>	weblogic	This user is the default administrator.

New Delete

- Under **Setting for ADMINUSER1** (or whatever your user's name is) window, click **Groups** tab.

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups > ADMINUSER1

Settings for ADMINUSER1

General Passwords Attributes **Groups**

Save

Use this page to configure group membership for this user.

Parent Groups:

Available:

- AdminChannelUsers
- Administrators
- AppTesters
- CrossDomainConnectors
- Deployers
- Monitors
- Operators
- OracleSystemGroup

Chosen:

Save

- Select the groups you want to assign to the user and click single right button as shown below.

Home > temp\_domain > Summary of Security Realms > myrealm > Users and Groups > ADMINUSER1

Settings for ADMINUSER1

General Passwords Attributes **Groups**

Save

Use this page to configure group membership for this user.

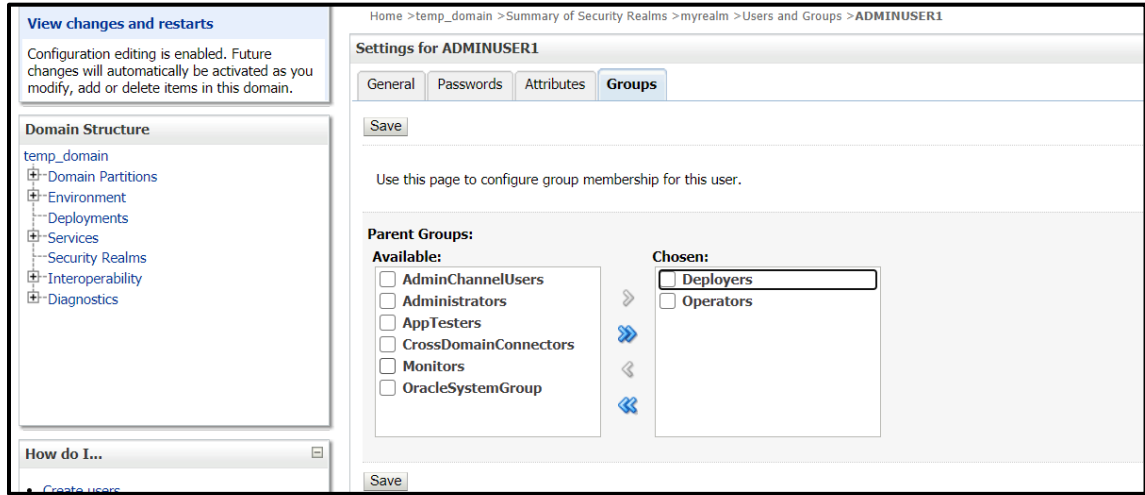
Parent Groups:

Available:

- AdminChannelUsers
- Administrators
- AppTesters
- CrossDomainConnectors
- Deployers
- Monitors
- Operators
- OracleSystemGroup

Chosen:

11. Click **Save**.



### 1.9.3 Oracle Banking Microservices Architecture Security Config Table Entries

Connection details for the embedded LDAP of weblogic (assuming the admin server is running on 10.99.99.10:7001) are given below:

**Connection Details:**

**URL:** ldap:// 10.99.99.10:7001

**Server Base:** dc={DOMAIN\_NAME} ( in our case it would be dc=ldap\_domain)

**User Search Base:** ou=people,ou=myrealm

**Server User:** cn=admin

**Server Credentials:** As setup in step Point 3 under 1.8.1

**Security Config Table Entries:**

ID	VALUE	Description
LDAP_URL	ldap:// 10.99.99.10:7001	Valid LDAP Server address with port.
LDAP_SERVER_USER	cn=admin	LDAP server login username
LDAP_SERVER_BASE	dc=ldap_domain	LDAP Server Base
LDAP_SERVER_CREDENTIAL	ylksiMFjVbfcP A7Qheh8Q==	LDAP server credentials in encrypted form(For Encryption steps, refer to Encrypted Utility section below)
LDAP_USER_SEARCH_BASE	ou=people,ou=myrealm	LDAP User Search Base
LDAP_PROVIDER	EMBEDDED_WEBLOGIC	Which LDAP Provider to be used. Also, if this row is not present in this table, then In-House Spring Plato LDAP will be used.

## 1.10 Oracle Analytic Server Setup

This section contains the following sub-sections:

- Prerequisite
- Start BI Server
- Upload BI Reports
- Test BI Reports

### 1.10.1 Prerequisite

Perform the following steps:

- Machine should have Java JDK1.8.0\_271 has installed
- Oracle Analytics Server 5.5.0

### 1.10.2 Start BI Server

Perform the following steps to start BI server:

1. Start the weblogic server and analytics server.
2. Check the weblogic console whether analytics server is running.

### 1.10.3 Upload BI Reports

Perform the following steps to upload BI reports:

1. Login to the Analytics server console.
2. Open the OSDC and check for the report Catalog object **{unzip folder}\REP\{reportfilename}.xdrz** or any other Catalog objects listed below:

Catalog Object	Extensions Supported
Data Model	.xdmz
Folder	.xdrz
Report	.xdoz
Style Template	.xssz
Subtemplate	.xsbz

3. Upload the catalog object to Analytics Server.

### 1.10.4 Test BI Reports

Perform the following steps to generate BI reports:

1. Open the application, and go the **Reports** section of the application.
2. Choose the report generation criteria. For example, **Start Date** or **End Date**.
3. Choose the format of the report.
4. Generate the report.

**NOTE:** If the format of the report selected is PDF, a PDF report will be generated.



## ANNEXURE - 1

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

<https://www.oracle.com/industries/financial-services/index.html>

Copyright © 2021, 2022, 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.