Oracle® Banking APIs

Single Sign-on Configuration-SAML

Release 25.1.0.0.0
G28246-01
April 2025

Oracle Banking APIs Single Sign-on Configuration-SAML, Release 25.1.0.0.0

G28246-01

Contents

List of Figures

Preface

Purpose

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

Audience

This document is intended for the following audience:

- Customers
- Partners

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at Critical Patches, Security Alerts and

Bulletins. All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by Oracle Software Security Assurance.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Related Resources

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

- Oracle Banking APIs Installation Manuals

Screenshot Disclaimer

Personal information used in the interface or documents is dummy and does not exist in the real world. It is only for reference purposes.

Acronyms and Abbreviations

The list of the acronyms and abbreviations used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
OBAPI	Oracle Banking APIs

1 Introduction

This document covers step-by-step details on configuration required at IDCS side (Application and User) and WebLogic console configurations for SAML and SQL Authentication Providers. Document also includes the configuration required on OHS to enable different URL's for internal and external user login.

2
Configuration

To enable SAML authentication it involves configuration at WebLogic server (console) and IDCS console.

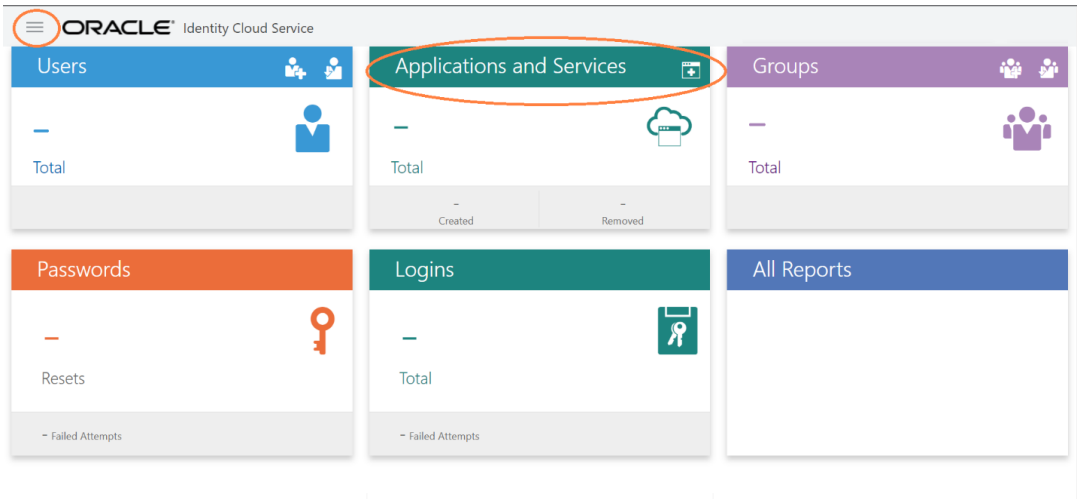
- Identity Provider Configuration at IDCS
- SAML Authentication Provider configuration
- SQL Authentication Provider configuration
- OHS Configuration
- Database Configuration
- IDCS OAuth Integration
- WebLogic configuration for OAuth
- OBAPI configuration for OAuth
- Default Admin Configuration
- Logout Configurations

2.1 Identity Provider Configuration at IDCS

Steps to configure Identity Provide at IDCS

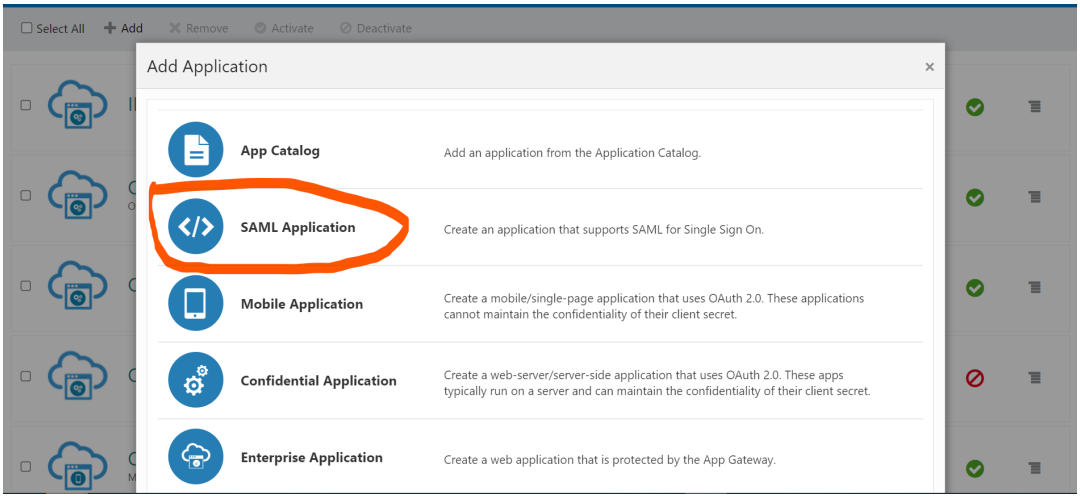
1. Login to Oracle Identity Cloud Service (IDCS) console with admin login. In dashboard click on **Add Application** in Application and Services widget or navigate through the breadcrumb menu as highlighted in screenshot.

Figure 2-1 Dashboard



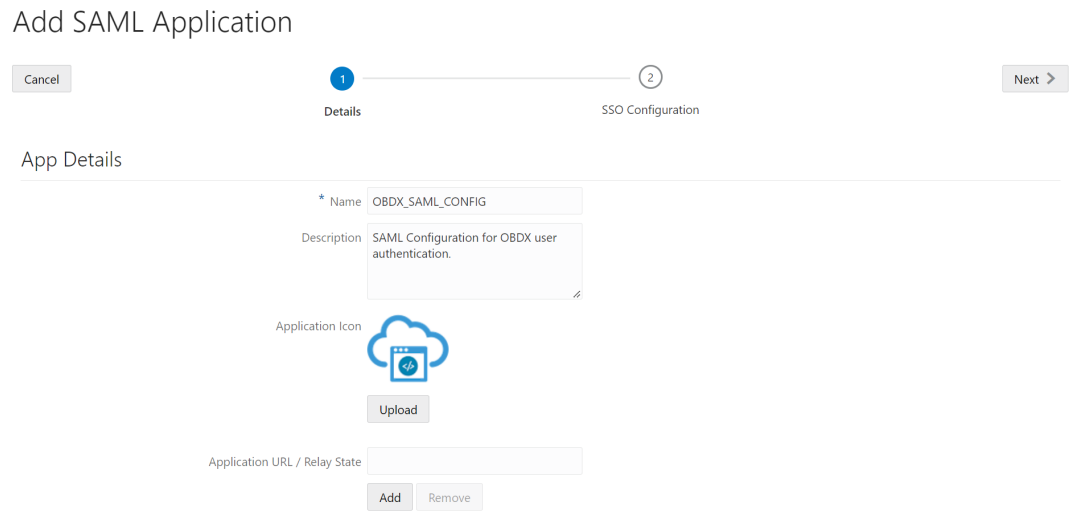
2. In popup window select **SAML** Application.

Figure 2-2 Add Application



3. In **Add SAML** Application page provide below mentioned fields and click on **Next**.
   a. Name
   b. Description

Figure 2-3 Add SAML Application

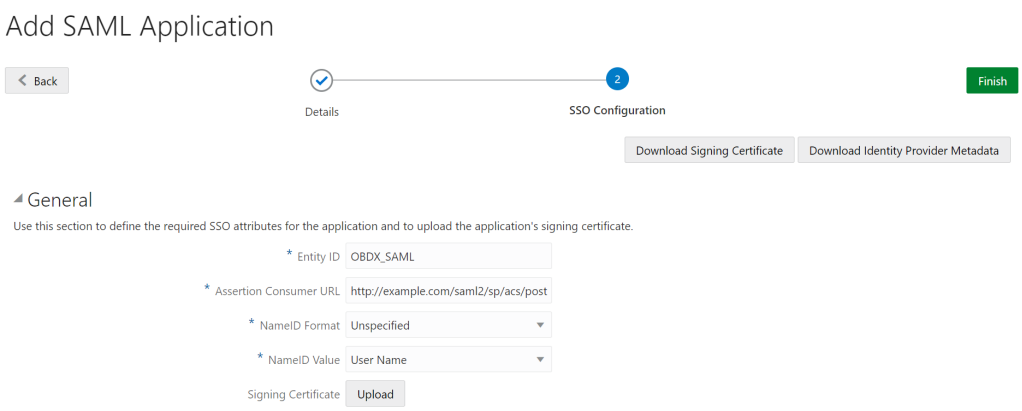


4. Fill below mentioned fields as per section.
   a. General
      i. Entity Id - A unique identifier / name for the service provider.
      ii. Assertion Consumer URL - End point to which assertion will be sent by IDCS. Recommended URL format `<OHS_URL>/saml2/sp/acs/pos`

      e.g. `<PROTOCOL>://<OHS_HOST>:<OHS_PORT>/saml2/sp/acs/post`

      http://whf000xxx.bank.com:9999/saml2/sp/acs/post
      iii. NameID Format- Select value as “Unspecified”.
      iv. NameID Value- Select value as “User Name”.

Figure 2-4 Add SAML Application



b. Advance Settings

i. Signed SSO :- Select value as “Assertion”

ii. Enable Single Logout: - This field should be checked.

iii. Logout Binding: - Select value as “Redirect”.

iv. Single Logout URL: - End point which IDCS will make call to do single logout functionality.
Recommended URL format `<OHS_URL>/digx-infra/sso-logout`

e.g. `<PROTOCOL>://<OHS_HOST>:<OHS_PORT>/digx-infra/sso-logout`

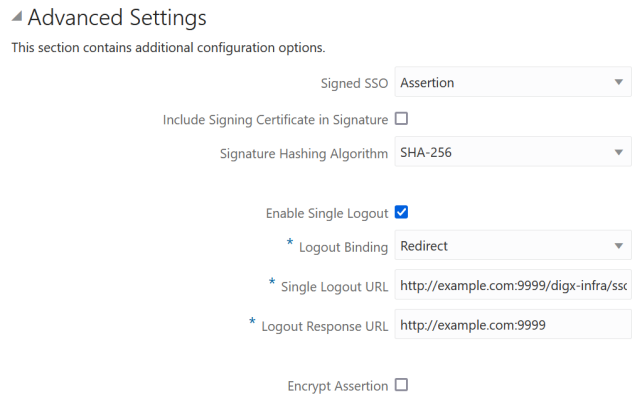
http://whf000xxx.bank.com:9999/digx-infra/sso-logout

v. Logout Response URL: -
Recommended URL format `<OHS_URL>/digx-infra/sso-logout`

e.g. `<PROTOCOL>://<OHS_HOST>:<OHS_PORT>/digx-infra/sso-logout`

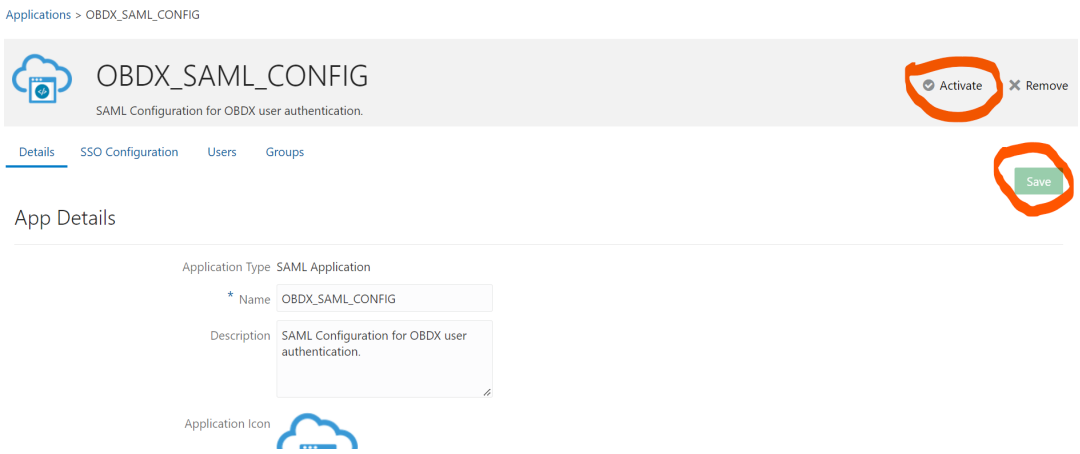
http://whf000xxx.bank.com:9999/digx-infra/sso-logout

Figure 2-5 Add SAML Application



5. Click on **Finish / Save**.
6. Click on **Activate** button to activate your application.

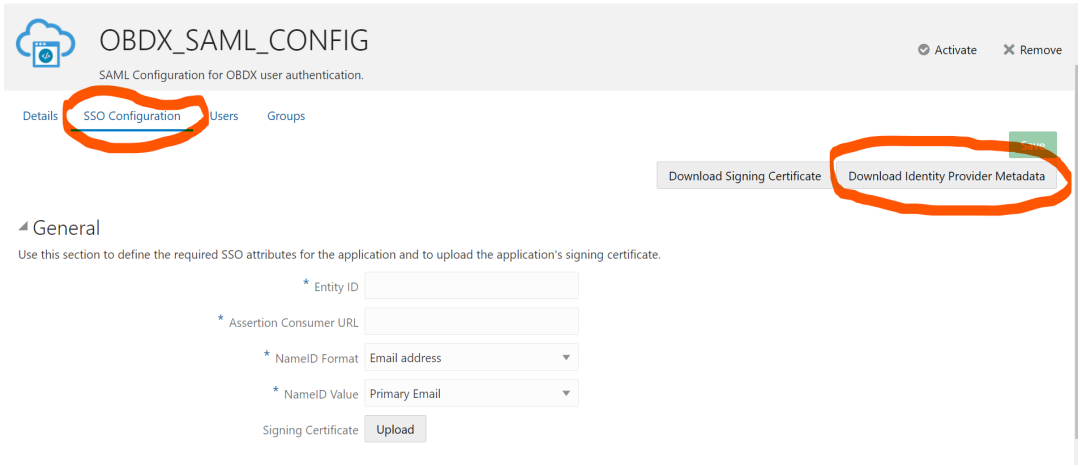
Figure 2-6 Edit Application



7. Navigate to Dashboard and search the application you have created.
8. Navigate to **SSO Configuration** tab and click on **"Download Identity Provider Metadata"**.

   Keep the downloaded xml file, it will be required to upload in WebLogic console. Same is explain in WebLogic console configuration steps.

Figure 2-7 Edit Application



9. Copy / FTP the downloaded IDC metadata xml file to WebLogic server using winscp / putty.
10. Navigate to **Users** tab in application to add the users related to application.
11. Click on **Assign Users or Assign (+)** button to search and add the users into application.

    If user is not available follow steps mentioned in Section 1.3 to create new user.

Figure 2-8 Edit Application

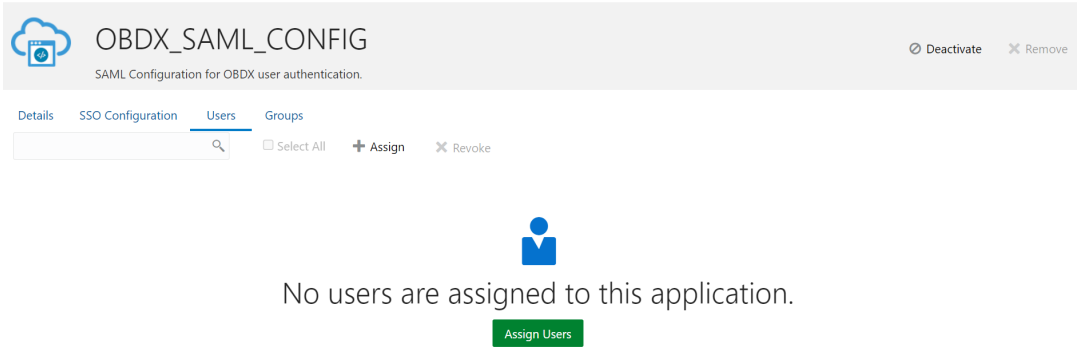
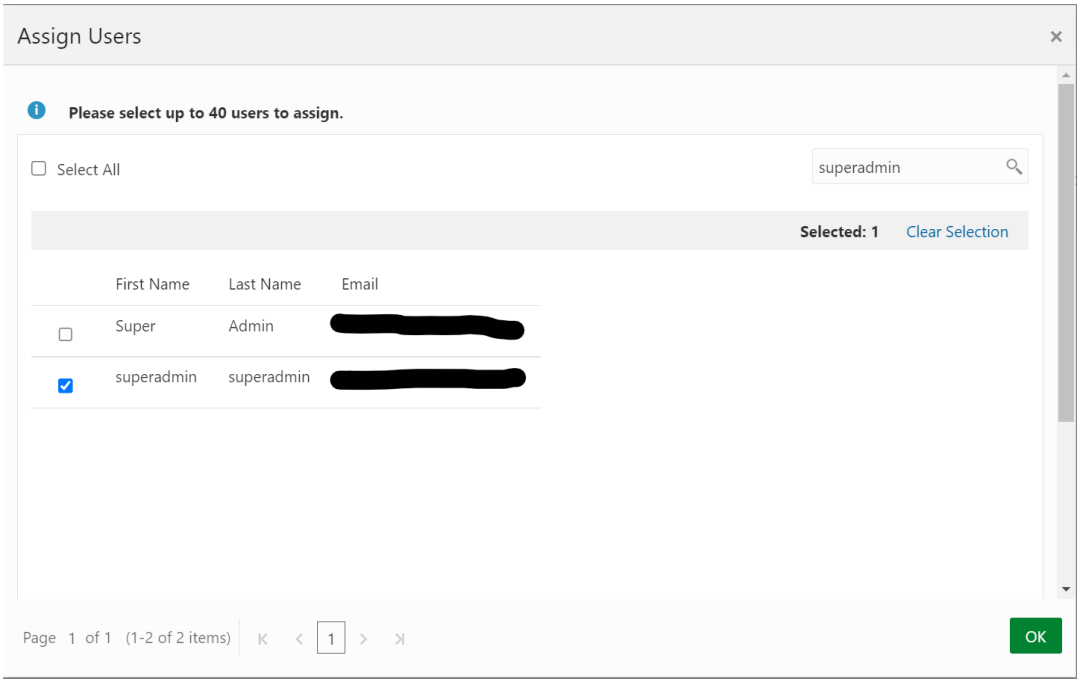


Figure 2-9 Assign Users



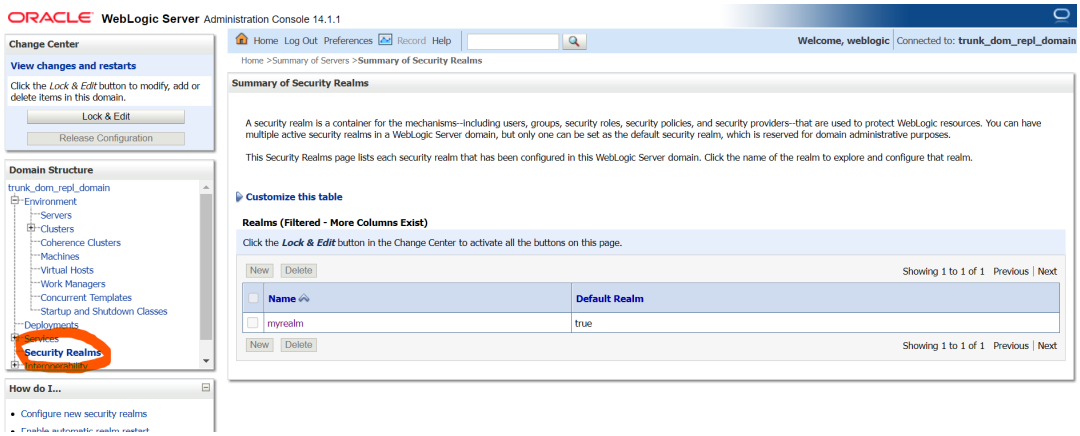
12. Logout from IDSC console.

2.2 SAML Authentication Provider configuration

Steps to configure SAML Authentication Providers changes into WebLogic console.

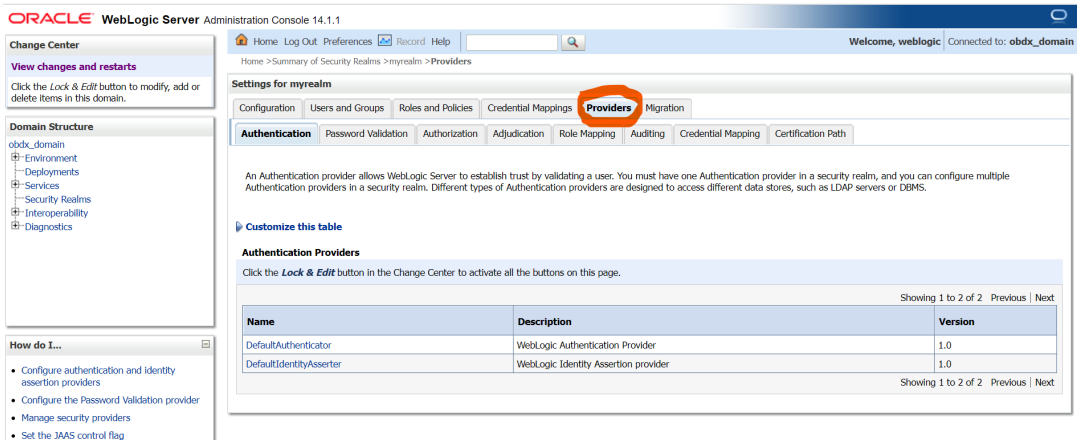
1. Login to WebLogic console with admin login and navigate to “Security Realms”.

Figure 2-10 Security Realms



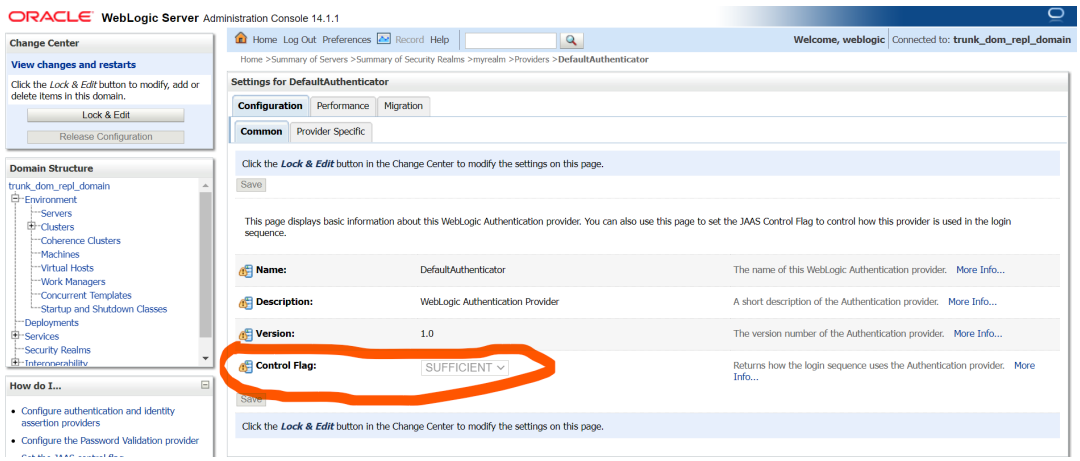
2. → Click on myrealm or your realm name present in screen. Navigate to "**Providers**" tab.

Figure 2-11 Providers



3. Select **"DefaultAuthenticator"** and change the Control Flag value to **"SUFFICIENT"**.

Figure 2-12 Default Authenticator



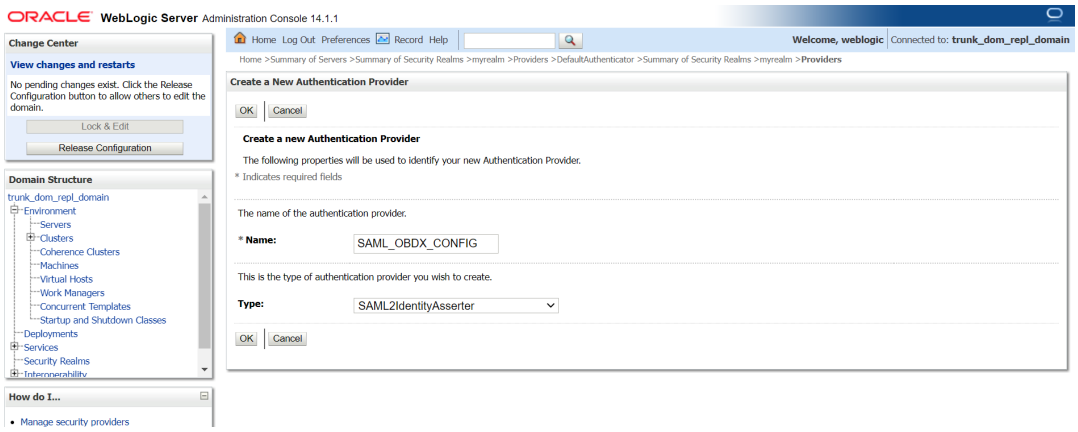
4. Again, navigate to **“Security Realms”** → myrealms → Providers and click on **New** button to create new Authentication Provider.

   Fill the below mentioned fields with appropriate values and click on **OK**.

   a. Name: Name of authentication provider.

   b. Type : Select value as “SAML2IdentityAsserter”.

Figure 2-13 Create Authentication Provider

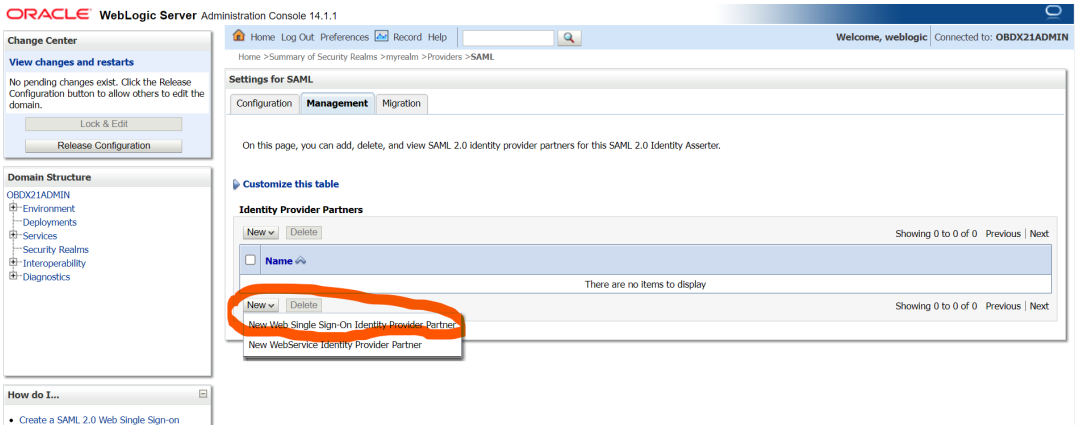


5. Restart Admin Server.

6. Login to WebLogic console and navigate to **“Security Realms”** → myrealms → Providers newly created authentication provider (e.g. SAML_OBDX_CONFIG) and navigate to **“Management”** tab.

7. Click on **New** button to add the Identity Provider Partner and select **“New Web Single Sign-On Identity Provider Partner”**.

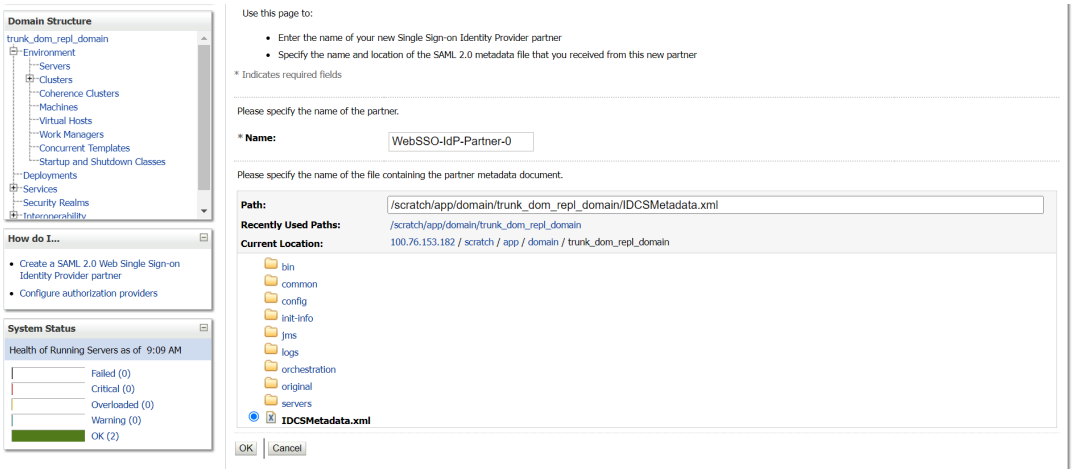
Figure 2-14 Management



8. Provide the name for the identity partner and select the IDC metadata xml copied to WebLogic server.

   Click **OK** button to save.

Figure 2-15 Create a SAML 2.0 Web Single Sign-on Identity Provider Partner



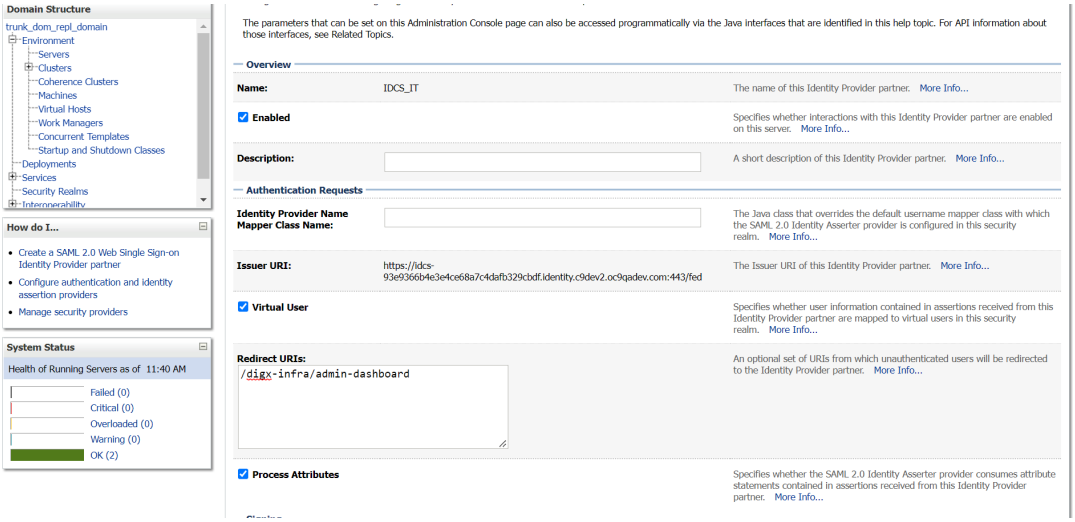
9. Open the newly added Identity Provider Partner and select below mentioned checkboxes and field and click on **Save**.

   a. Enable: Checked

   b. Virtual User: Checked

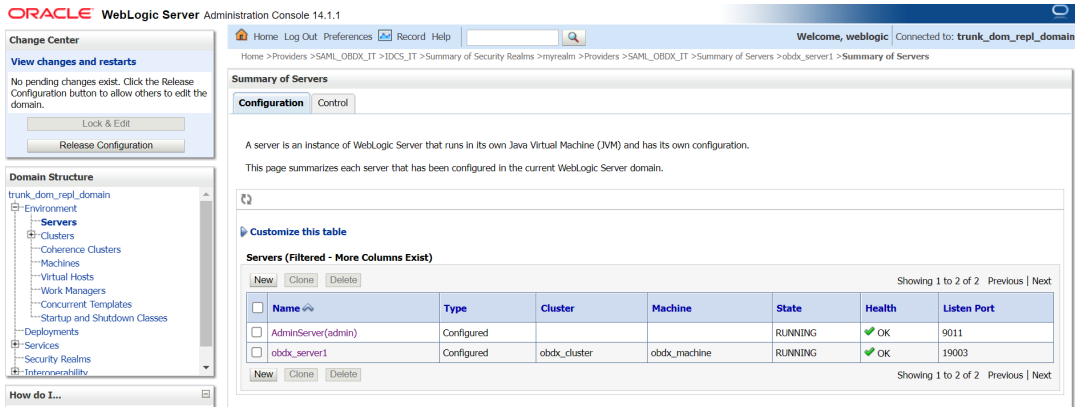
   c. Redirect URIs: /digx-infra/admin-dashboard

Figure 2-16 Settings for Create a SAML 2.0 Web Single Sign-on Identity Provider Partner



10. Navigate to “Environment” → “Servers” and select the server on which SSO authentication application will be deployed.

Figure 2-17 Servers



11. Navigate to “Federation Services” → “SAML 2.0 General” and provide values to below mentioned fields. Click on **Save**.

    a. Published Site URL: Recommended URL format `<OHS URL>/saml2`
    e.g. `<PROTOCOL>://<OHS_HOST>:<OHS_PORT>/saml2`

    `http://whf000xxx.bank.com:9999/saml2`

    b. Entity Id: Value should match with Entity Id provided in SAML configuration in IDCS console.

    c. Recipient Check Enabled: unchecked.

Figure 2-18 SAML 2.0 General

12. Navigate to “Federation Services” → “SAML 2.0 Service Provider” and provide values to blow mentioned fields and click on **Save**.

    a. Enabled: Check box should be checked.

    b. Preferred Binding: Post

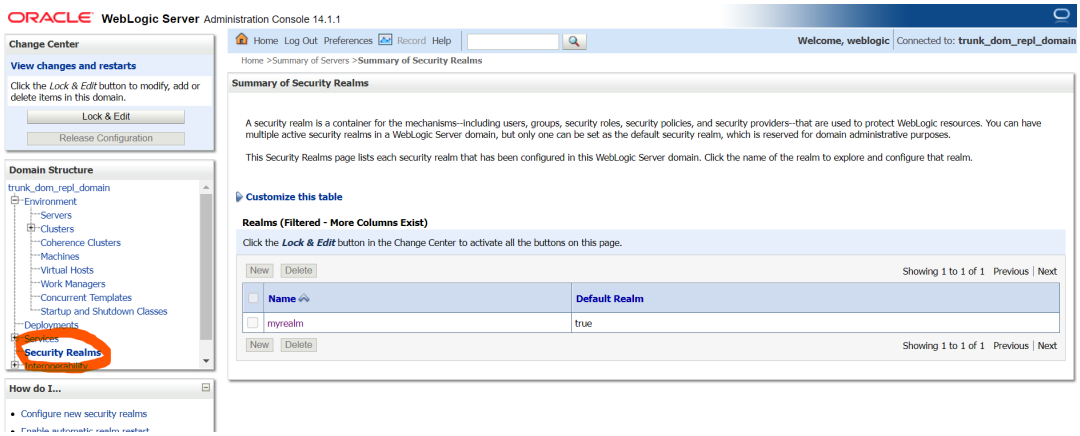
    c. Default URL: <OHS_URL>/digx-infra/admin-dashboard

2.3 SQL Authentication Provider configuration

Steps to configure SQL Authentication Providers changes into WebLogic console.

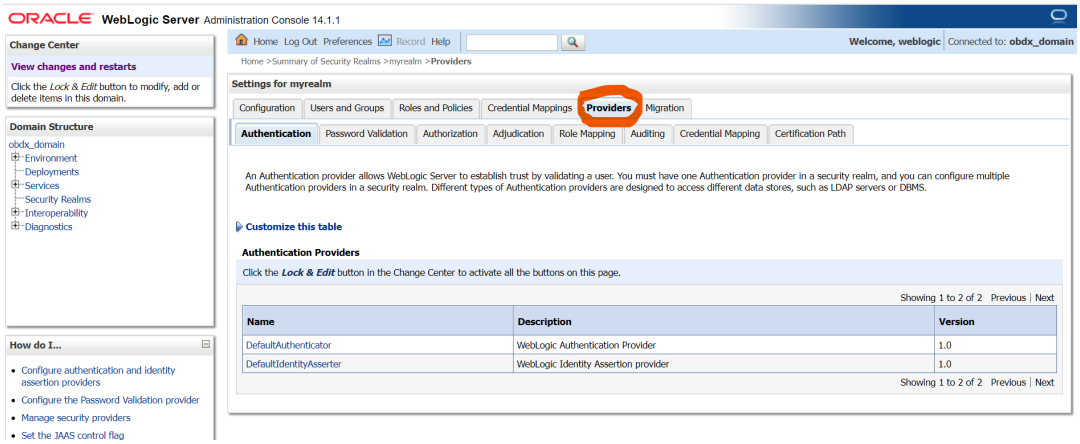
1. Login to WebLogic console with admin login and navigate to “Security Realms”.

Figure 2-19 Security Realms



2. → Click on myrealm or your realm name present in screen. Navigate to **“Providers”** tab.

Figure 2-20 Providers



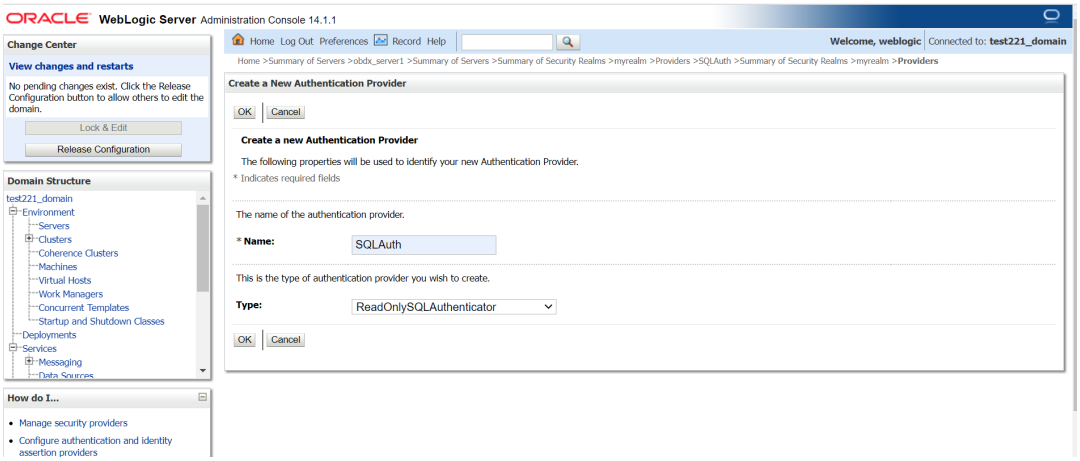
3. Click on **New** button to create new Authentication Provider.

   Fill the below mentioned fields with appropriate values and click on **OK**.

   a. Name: Name of authentication provider.

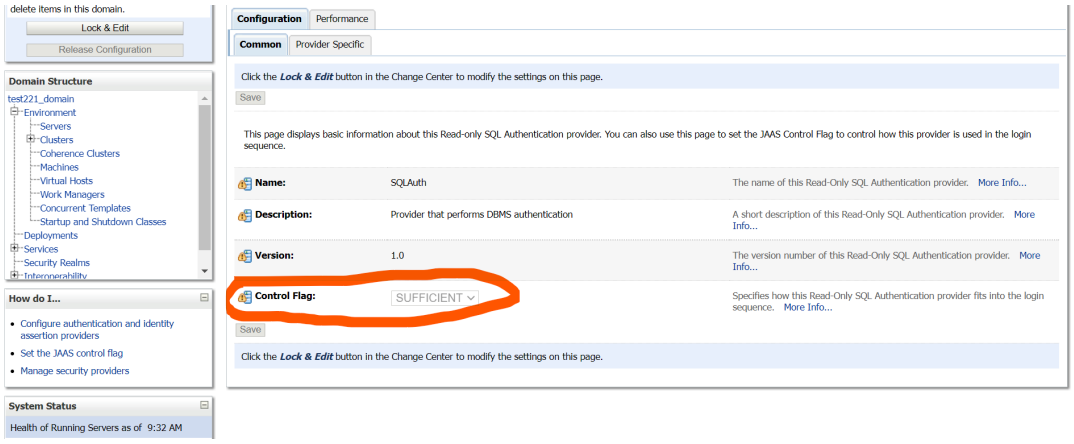
   b. Type :Select value as “ReadOnlySQLAuthenticator”.

Figure 2-21 Create New Authentication Provider



4. Open newly created authentication provider (e.g. SQLAuth). Select the value of Control Flag as **"SUFFICIENT"**.

Figure 2-22 Settings for Read Only SQL Authentication Provider

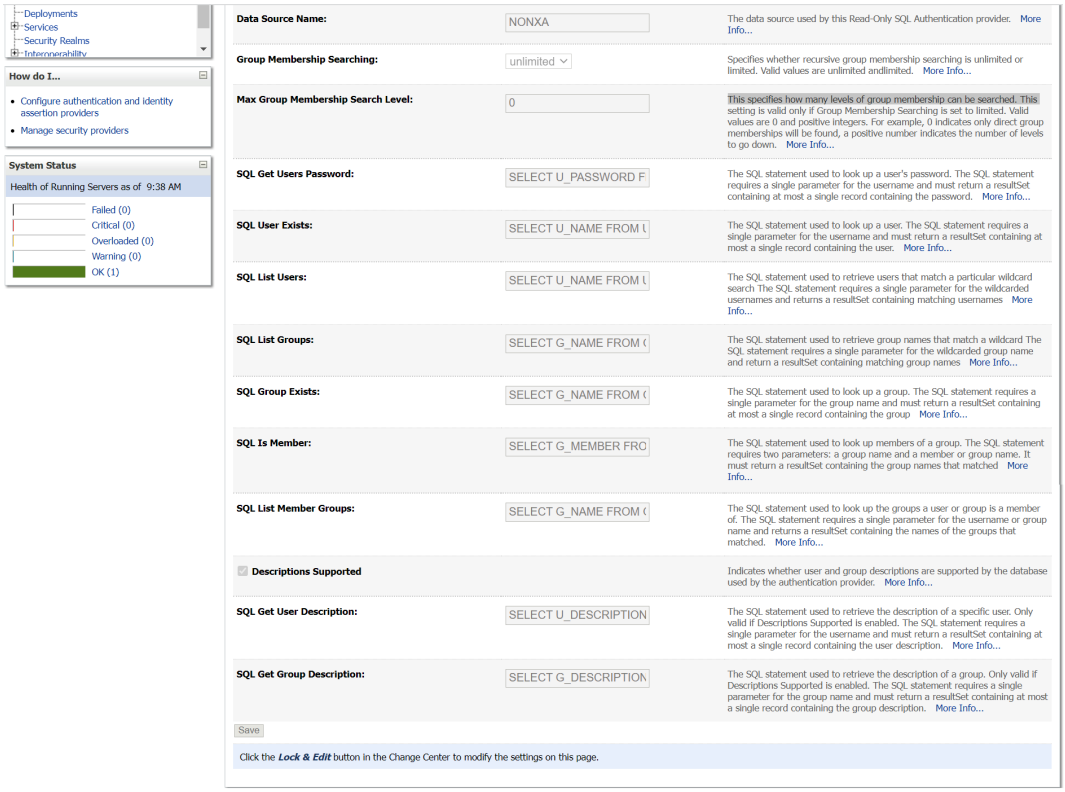


5. Navigate to "Provider Specific" tab to configuration related to SQL Authentication.
6. Provide the values to fields mentioned below with given value in case it is not auto populated.
   a. Data Source Name: NONXA
   b. SQL Get Users Password: SELECT U_PASSWORD FROM USERS WHERE U_NAME = ?
   c. SQL User Exists: SELECT U_NAME FROM USERS WHERE U_NAME = ?
   d. SQL List Users: SELECT U_NAME FROM USERS WHERE U_NAME LIKE ?
   e. SQL List Groups: SELECT G_NAME FROM GROUPS WHERE G_NAME LIKE ?
   f. VI. SQL Group Exists: SELECT G_NAME FROM GROUPS WHERE G_NAME = ?
   g. SQL Is Member: SELECT G_MEMBER FROM GROUPMEMBERS WHERE G_NAME = ? AND G_MEMBER = ?
   h. SQL List Member Groups: SELECT G_NAME FROM GROUPMEMBERS WHERE G_MEMBER = ?

i. SQL Get User Description: - SELECT U_DESCRIPTION FROM USERS WHERE U_NAME = ?

j. SQL Get Group Description: - SELECT G_DESCRIPTION FROM GROUPS WHERE G_NAME = ?

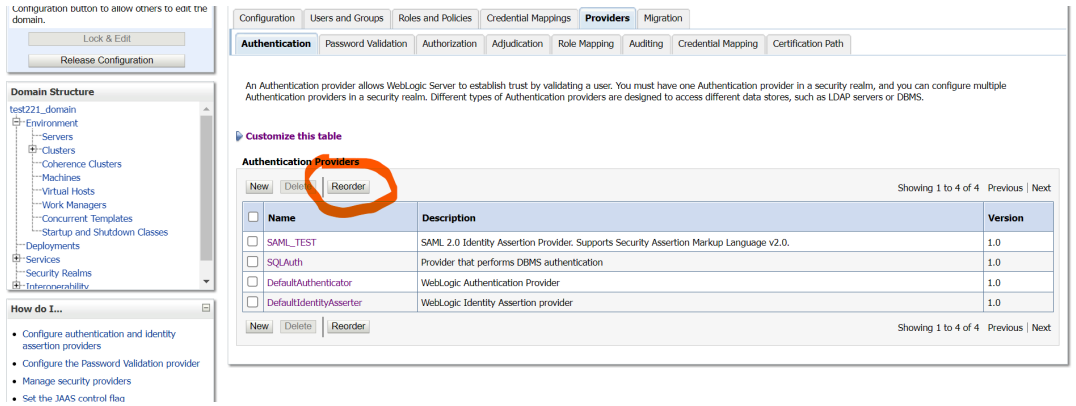
Figure 2-23 Settings for Read Only SQL Authentication Provider



7. Click on **Save**.

8. Navigate to **"Security Realms"** → myrealms → Providers and click on **Reorder** button.

Figure 2-24 Authentication



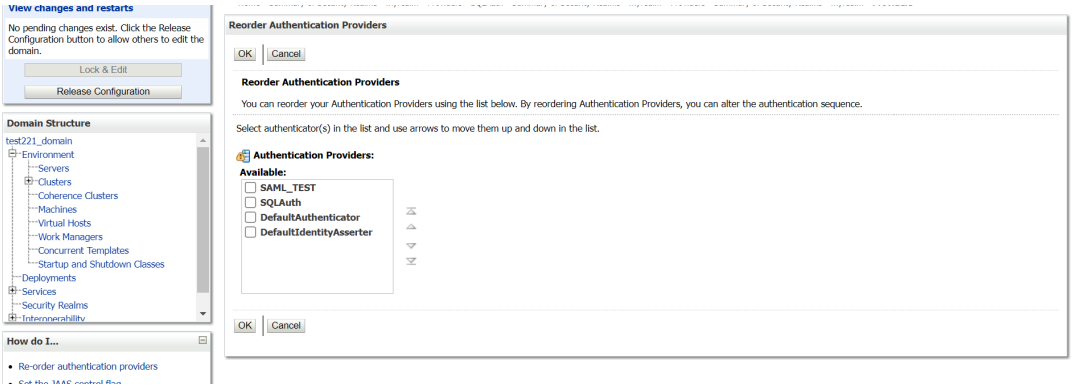
9. Reorder the authentication providers as given below.

a. SAML Authentication Provider

b. SQL Authentication Provider

c. Default Authenticator

Figure 2-25 Reorder Authentication Providers



10. Restart all the servers in domain including Admin Server.

> **Note:**
>
> Accessing /saml2 uri from OHS (<OHS_URL>/saml2), /saml2 uri has to be proxy bypassed from OHS

2.4 OHS Configuration

Provides details on configuration required on OHS to enable different URL's for internal and external users. i.e authentication with OBDX or external service provider.

1. Open obdx.conf file from OHS server. You can find the location of obdx.conf file from httpd.conf file.
2. Verify if proxypass URLs are configured in obdx.conf file. If not then add entries as mentioned in below format.

```
ProxyPassMatch "/digx(.*)" "<PROTOCOL>://<WL_HOST_NAME>:<WL_PORT>/digx$1"
ProxyPassReverse "/digx(.*)" "<PROTOCOL>://<WL_HOST_NAME>:<WL_PORT>/digx$1"
ProxyPassMatch "/saml2(.*)" "<PROTOCOL>://<WL_HOST_NAME>:<WL_PORT>/saml2$1"
ProxyPassReverse "/saml2(.*)" <PROTOCOL>://<WL_HOST_NAME>:<WL_PORT>/saml2$1
ProxyPassMatch "/digx(.*)" "http://whf000xxx.bank.com:19003/digx$1"
ProxyPassReverse "/digx(.*)" "http://whf000xxx.bank.com:19003/digx$1"
ProxyPassMatch "/saml2(.*)" "http://whf000xxx.bank.com:19001/saml2$1"
ProxyPassReverse "/saml2(.*)" "http://whf000xxx.bank.com:19001/saml2$1"
```

3. Add below virtual configuration into obdx.conf file.

```
##Virtual HostsListen <PORT_1><VirtualHost *:<PORT_1>>    ServerName
        <HOST_NAME>
        RewriteEngine On
        RewriteOptions inherit
          <Directory
```

```
        "${DocumentRoot}">
        Options FollowSymLinks
        AllowOverride all
        </Directory></VirtualHost> Listen <PORT_2><VirtualHost
          *:<PORT_2>>
        ServerName <HOST_NAME>
        RewriteEngine On
          RewriteRule    "^(.*)/config\.js$"
          "<SERVER_PROTOCOL>://<HOST_NAME>:<PORT_2>/framework/js/
configurations/config-admin.js"  [R]
          <Directory
        "${DocumentRoot}">
        Options FollowSymLinks
        AllowOverride all
        </Directory>
    </VirtualHost>
```

> **Note:**
>
> Replace the <PORT_1> & <PORT_2> with the ports which are expose to outside world. Replace <SERVER_PROTOCOL> and <HOST_NAME> with appropriate values. E.g. http and whfxxx.sample.com (if hostname is not available then <HOST_NAME> value can be IP address.)

```
# All other request passed through this rules.
ProxyPassMatch "/digx(.*)" "http://whf00qiw.in.oracle.com:19001/digx$1"
ProxyPassReverse "/digx(.*)" "http://whf00qiw.in.oracle.com:19001/digx$1"
ProxyPassMatch "/saml2(.*)" "http://whf00qiw.in.oracle.com:19001/saml2$1"
ProxyPassReverse "/saml2(.*)" "http://whf00qiw.in.oracle.com:19001/saml2$1"

##Virtual Hosts
Listen 8888
<VirtualHost *:8888>
    ServerName whf00qiw.in.oracle.com
    RewriteEngine On
    RewriteOptions inherit

    <Directory "${DocumentRoot}">
        Options FollowSymLinks
        AllowOverride all
        #Require all granted
    </Directory>
</VirtualHost>

Listen 9999
<VirtualHost *:9999>
        ServerName whf00qiw.in.oracle.com
        RewriteEngine On
        RewriteRule    "^(.*)/config\.js$"  "http://whf00qiw.in.oracle.com:9999/framework/js/configurations/config-admin.js"  [R]

    <Directory "${DocumentRoot}">
        Options FollowSymLinks
        AllowOverride all
        #Require all granted
    </Directory>
</VirtualHost>
```

4. Save obdx.conf file and restart ohs server.

2.5 Database Configuration

To enable SSO for external users below configuration need to be done in database.

1. To enable SSO authentication for user type / enterprise role execute below query on intended database environment. Replace <USER_TYPE> with the user type / enterprise role for which SSO authentication to be enabled.

```
UPDATE DIGX_FW_CONFIG_ALL_B SET PROP_VALUE = 'External' WHERE PROP_ID =
'<USER_TYPE>' AND CATEGORY_ID = ' AuthenticationConfiguration ';
```

For example: `UPDATE DIGX_FW_CONFIG_ALL_B SET PROP_VALUE = 'External' WHERE PROP_ID = 'administrator' AND CATEGORY_ID = 'AuthenticationConfiguration';`

2. Execute below query for redirection after authentication from SSO service provider back to OBDX. Replace the value of `<OHS_URL_FOR_ADMIN_USER_LOGIN>` with the OHS_URL with port enable for external / admin user login, the virtual host enabled in section 3.4, step 3.

```
INSERT INTO DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE,
FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE,
LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS, OBJECT_VERSION_NUMBER,
EDITABLE, CATEGORY_DESCRIPTION) values ('SSO_PUBLIC_URL', 'dayoneconfig',
'<OHS_URL_FOR_ADMIN_USER_LOGIN>', 'N', null, 'Public SSO URL', 'ofssuser',
to_timestamp('29-09-22 10:05:56.000000000 AM', 'DD-MM-RR fmHH12:fmMI:SSXFF
AM'), 'ofssuser', to_timestamp('29-09-22 10:05:56.000000000 AM', 'DD-MM-RR
fmHH12:fmMI:SSXFF AM'), 'A', 1, 'N', null);
```

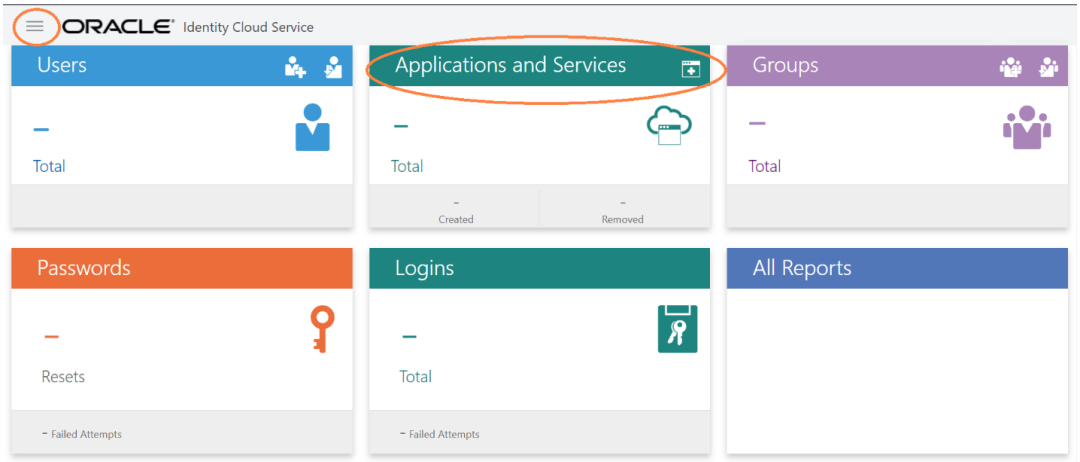
For Example: `INSERT INTO DIGX_FW_CONFIG_ALL_B (PROP_ID, CATEGORY_ID, PROP_VALUE, FACTORY_SHIPPED_FLAG, PROP_COMMENTS, SUMMARY_TEXT, CREATED_BY, CREATION_DATE, LAST_UPDATED_BY, LAST_UPDATED_DATE, OBJECT_STATUS, OBJECT_VERSION_NUMBER, EDITABLE, CATEGORY_DESCRIPTION) values ('SSO_PUBLIC_URL', 'dayoneconfig', 'http:// whf000xxx.bank.com:9999', 'N', null, 'Public SSO URL', 'ofssuser', to_timestamp('29-09-22 10:05:56.000000000 AM', 'DD-MM-RR fmHH12:fmMI:SSXFF AM'), 'ofssuser', to_timestamp('29-09-22 10:05:56.000000000 AM', 'DD-MM-RR fmHH12:fmMI:SSXFF AM'), 'A', 1, 'N', null);`

2.6 IDCS OAuth Integration

To fetch the user information from external SSO provider, application need to be registered as a client in IDCS. Below steps providers details on registering the application in IDCS.

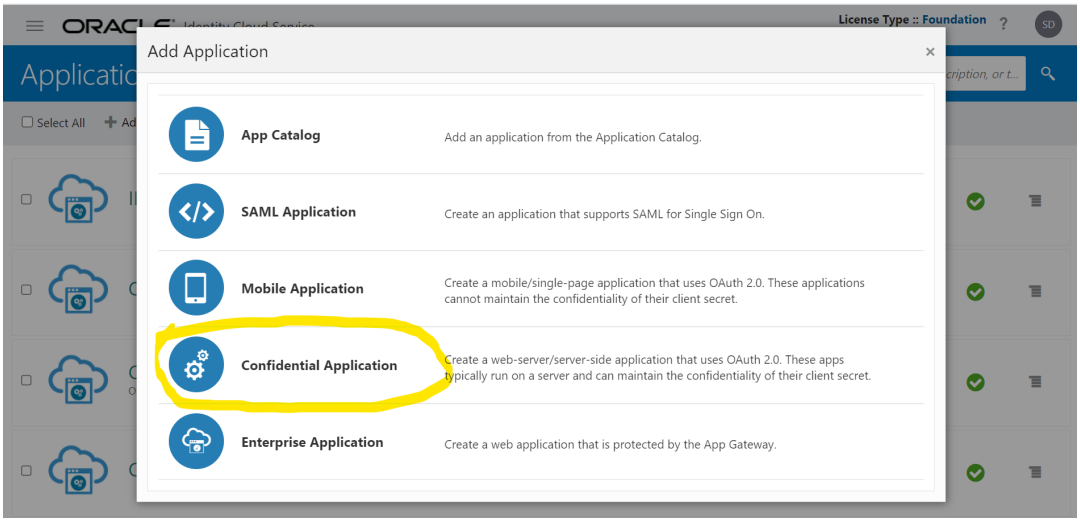
1. Login to Oracle Identity Cloud Service (IDCS) console with admin login. In dashboard click on **Add Application** in **Application and Services** widget or navigate through the breadcrumb menu as highlighted in screenshot.

Figure 2-26 Dashboard



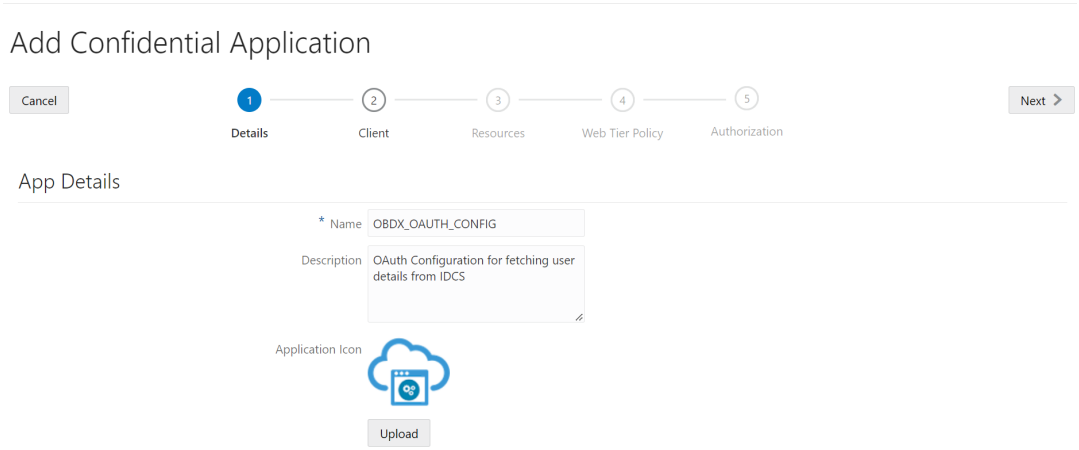
2. In popup window select Confidential Application.

Figure 2-27 Add Application



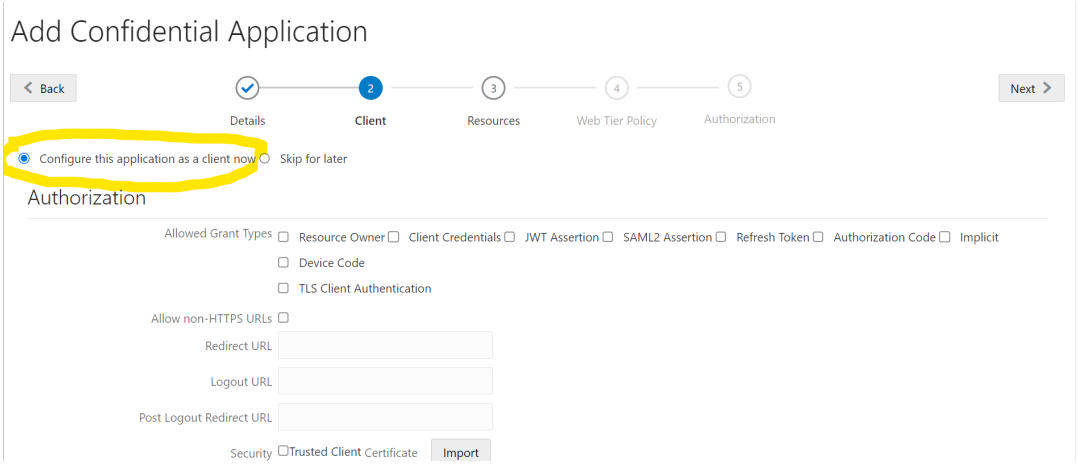
3. In **Add Confidential Application** page provide below mentioned fields and click on **Next**.
   a. Name
   b. Description

Figure 2-28 Add Confidential Application



4. Select Configure this application as a client now option in screen as shown in below screenshot.
   a. Name
   b. Description

Figure 2-29 Add Confidential Application

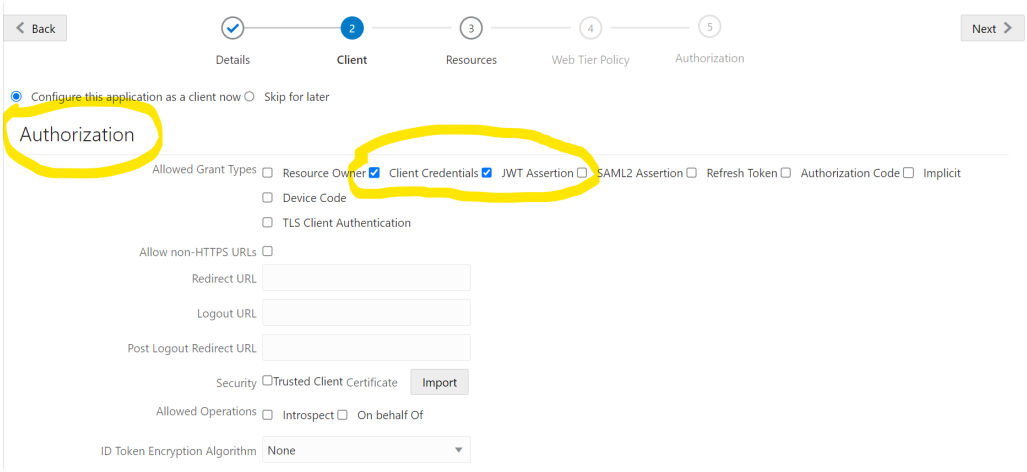


5. Fill below mentioned fields as per section.

   a. Authorization

      i. Allowed Grant Types:- Select checkbox as “Client Credentials” and “JWT Assertion”

Figure 2-30 Add Confidential Application

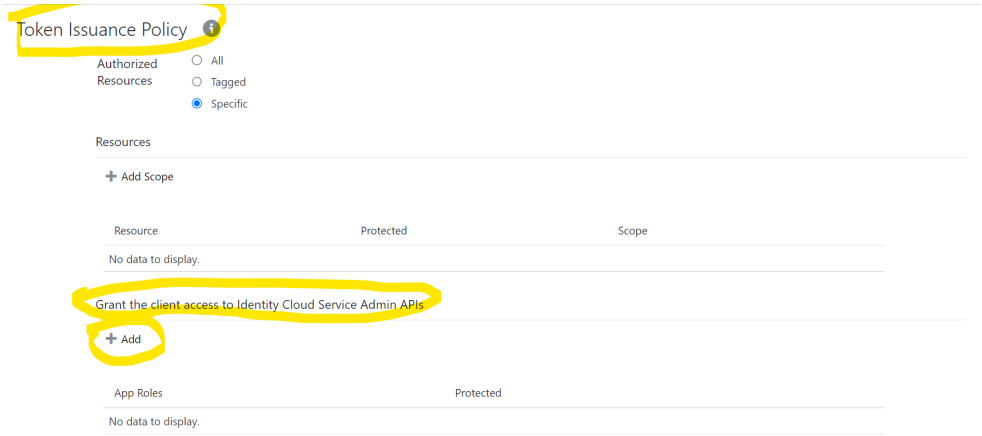


   b. Token Issuance Policy

      i. Authorized Resources :Select value as “Specific”

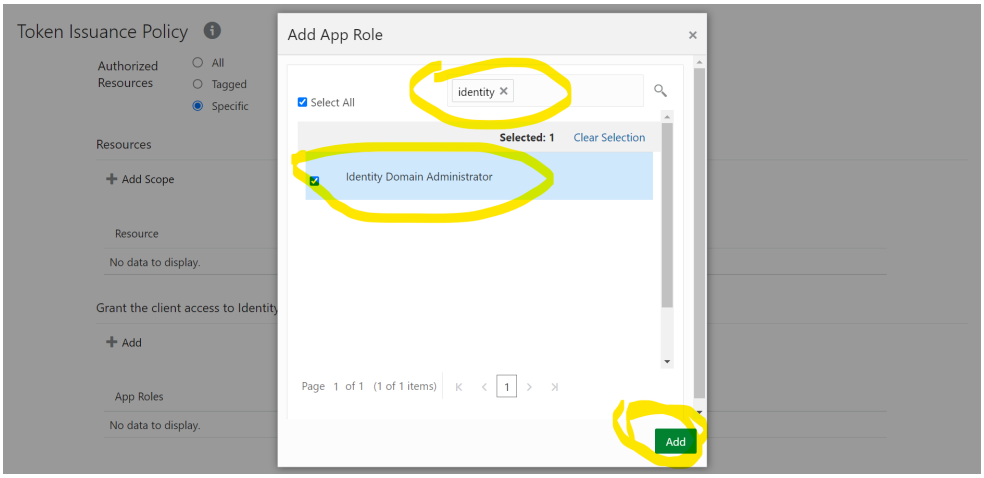
      ii. Grant the client access to Identity Cloud Service Admin APIs: Click on **Add** button

Figure 2-31 Add Confidential Application



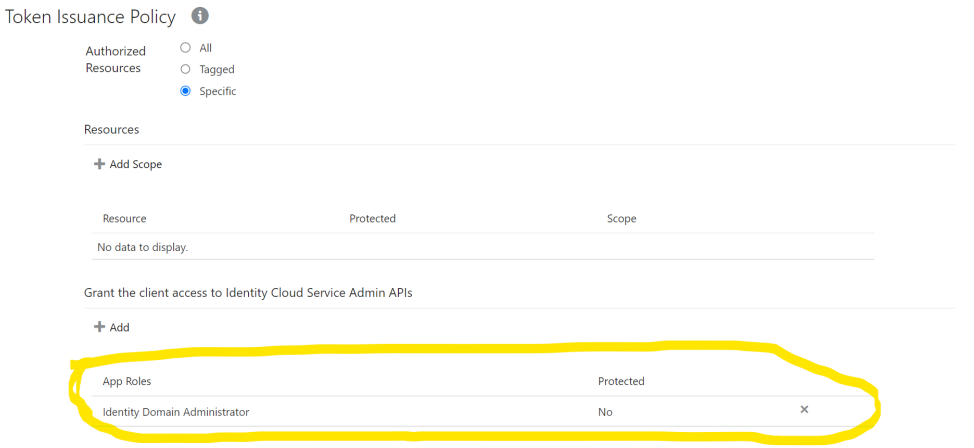
iii. In popup window search for "**Identity Domain Administrator**" and click on **Add**.

Figure 2-32 Add App Role



iv. Verify a row added in table for **App Roles** as shown like below screenshot.

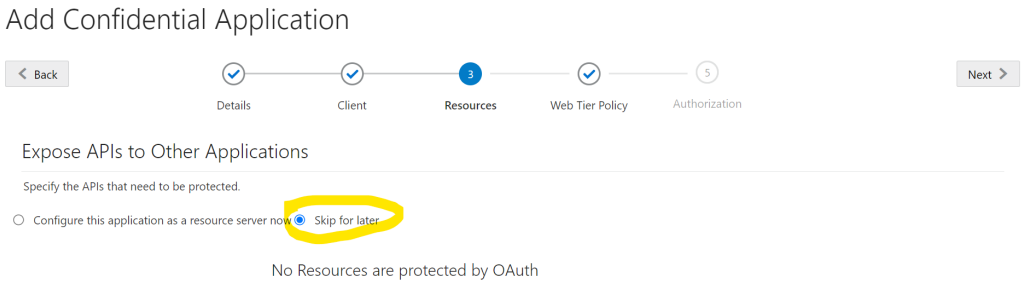
Figure 2-33 Add Confidential Application



v. Click on **Next** button on top.

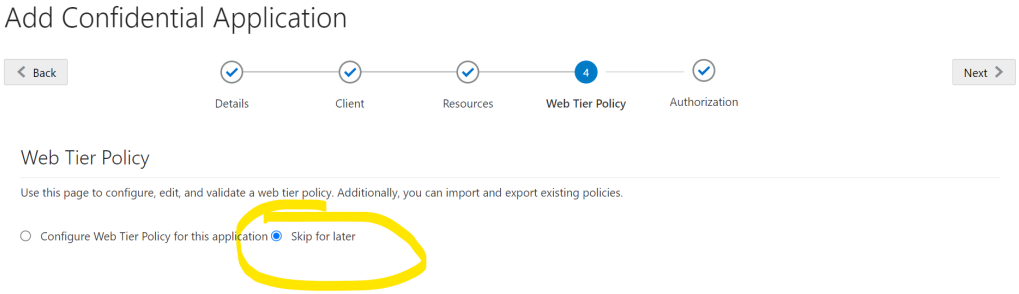
c. Expose APIs to Other Applications: Select "**Skip for later**" and click on **Next**.

Figure 2-34 Add Confidential Application



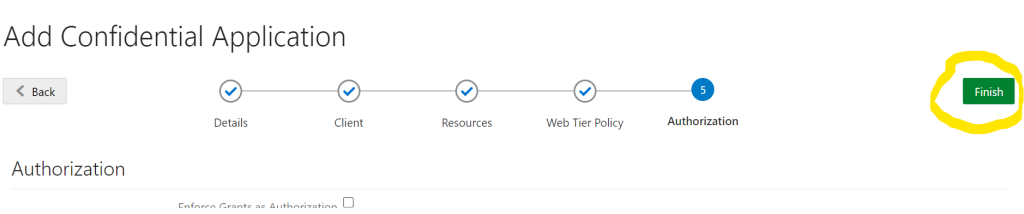
d. Web Tier Policy: Select **"Skip for later"** and click on **Next** button.

Figure 2-35 Add Confidential Application



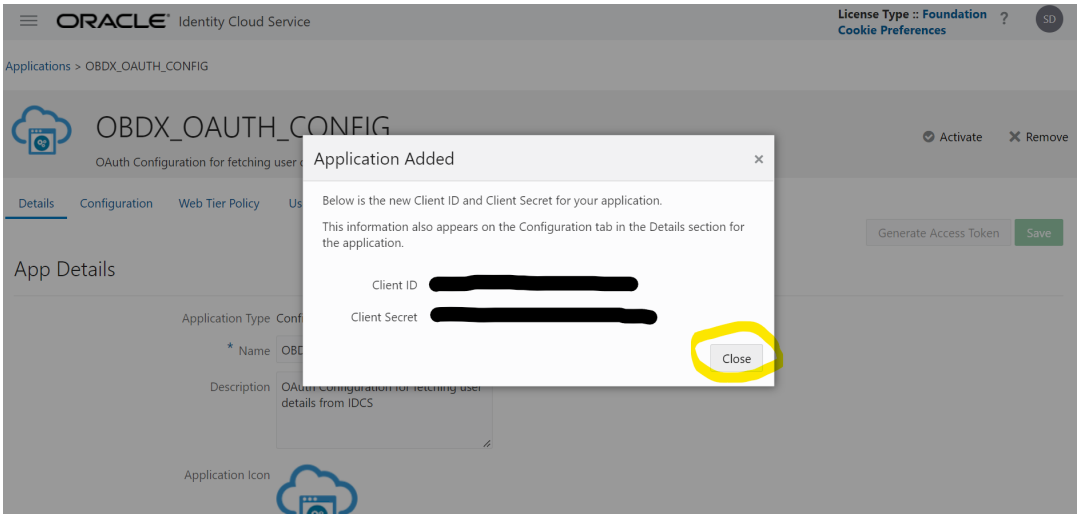
e. Click on **"Finish"**.

Figure 2-36 Add Confidential Application



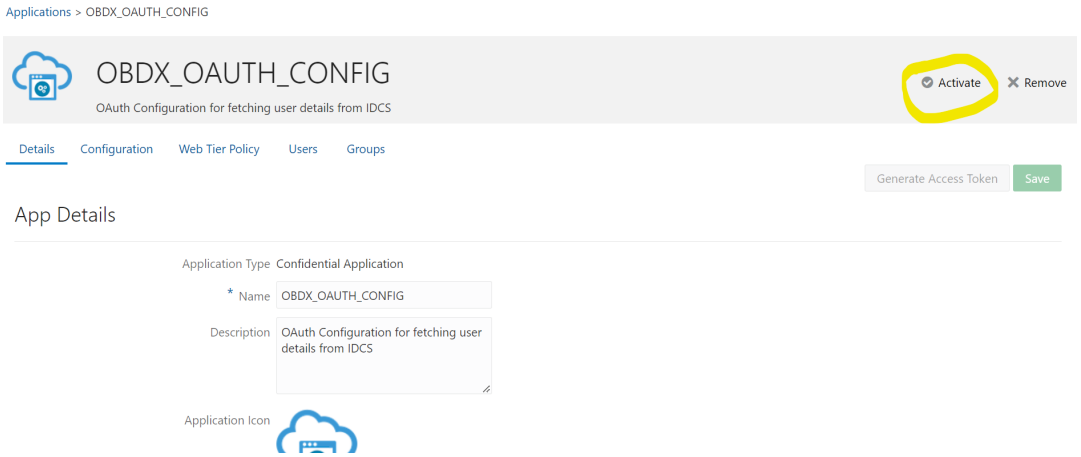
6. After finish click a popup window will open with "Client ID" and "Client Secret" as shown in below screenshot. Copy the Client Id and Client Secret to text file to keep it handy as it will be required in further steps. Once copied click on "Close".

Figure 2-37 Add Confidential Application



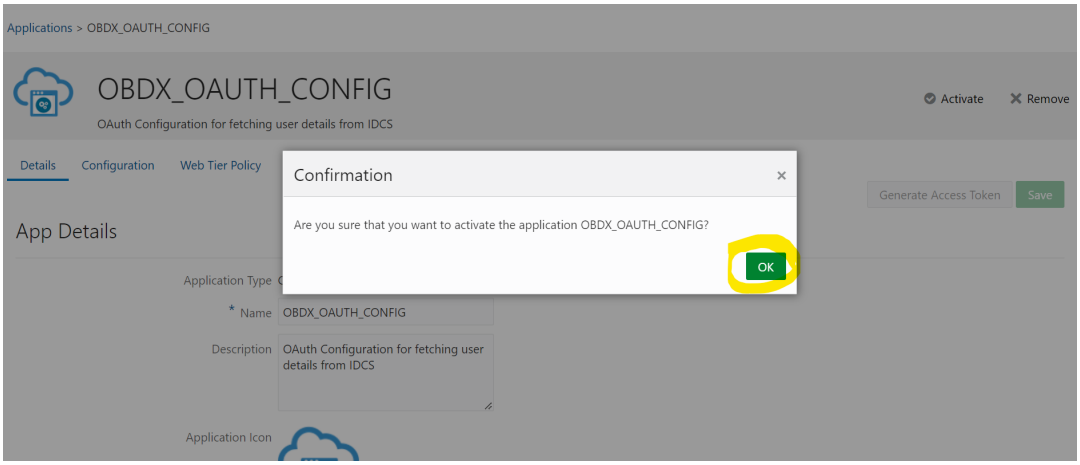
7. Click on **"Activate"** button to activate the application.

Figure 2-38 Edit Application



8. Popup window asking confirmation to activate the application will open, click on "**OK**" to activate the application.

Figure 2-39 Edit Application



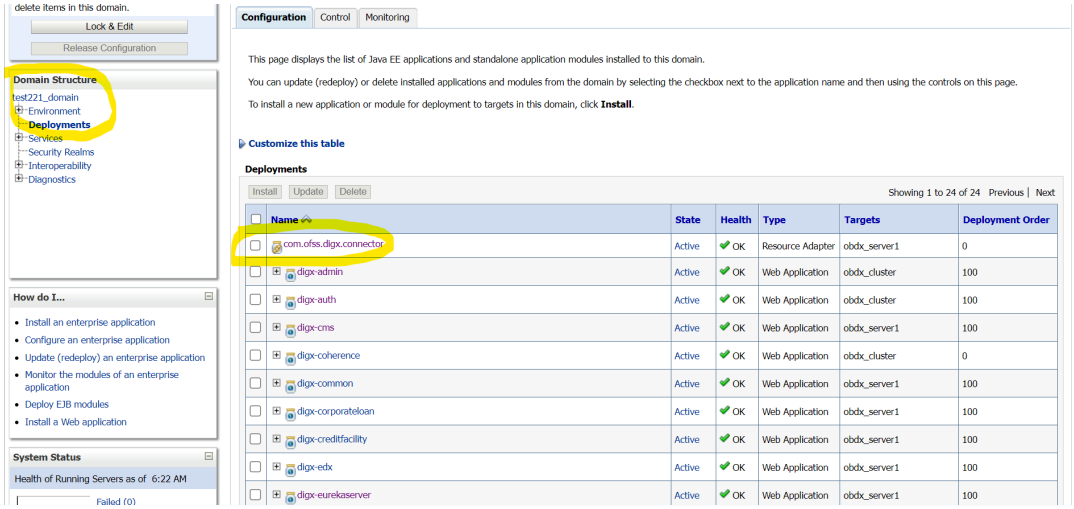
9. Logout from IDCS console.

2.7 WebLogic configuration for OAuth

To enable OAuth support on WebLogic server follow below mentioned steps.

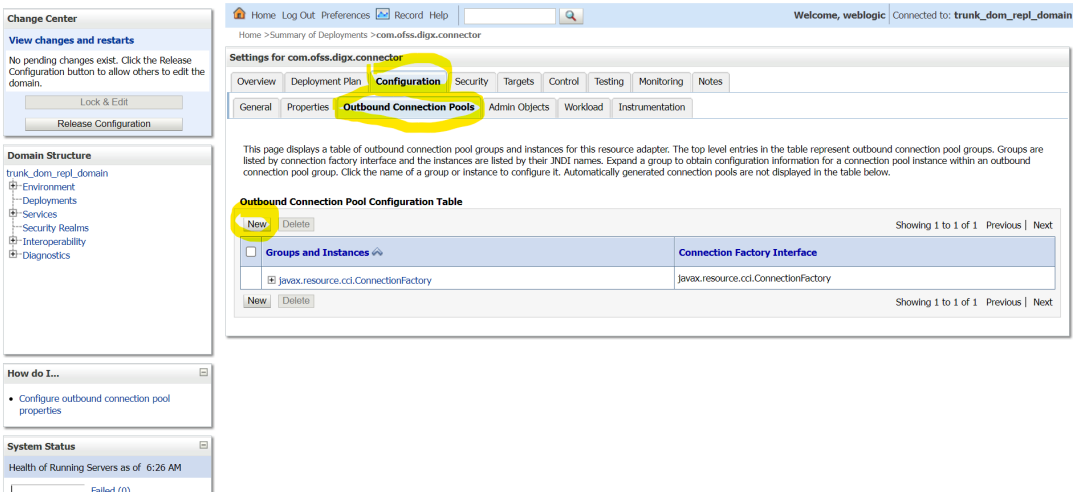
1. Login to WebLogic console with admin login and navigate to “Domain Structure” → “Deployments”.
2. Click on “com.ofss.digx.connector”

Figure 2-40 Deployments



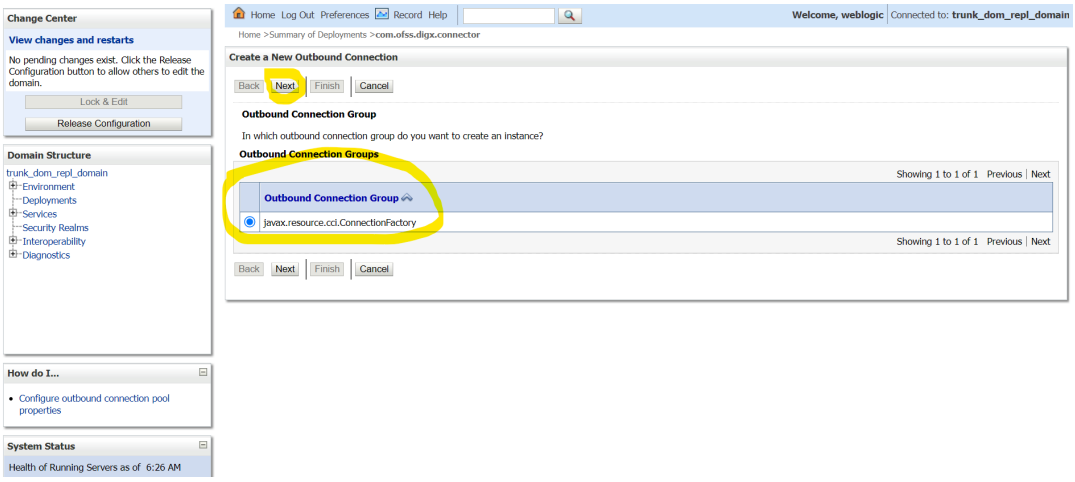
3. Navigate to “Configuration” → “Outbound Connection Pools” tab and click on **New**.

Figure 2-41 Outbound Connection Pools Configuration



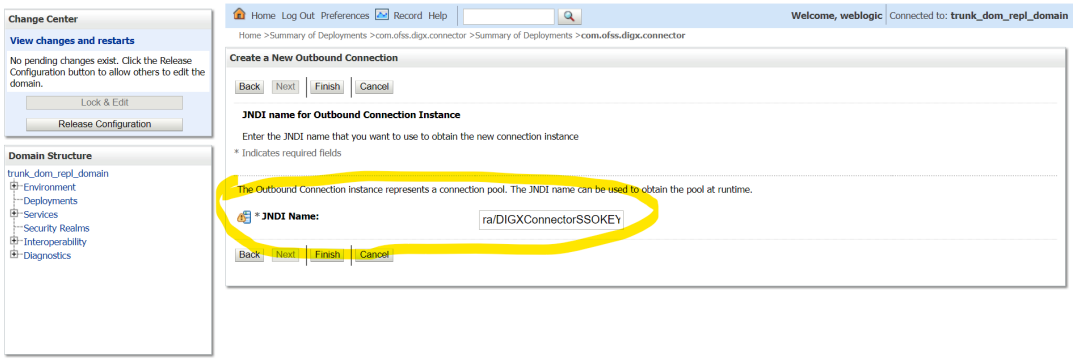
4. Select “javax.resource.cci.ConnectionFactory” and click on **Next**.

Figure 2-42 Outbound Connection Group Configuration



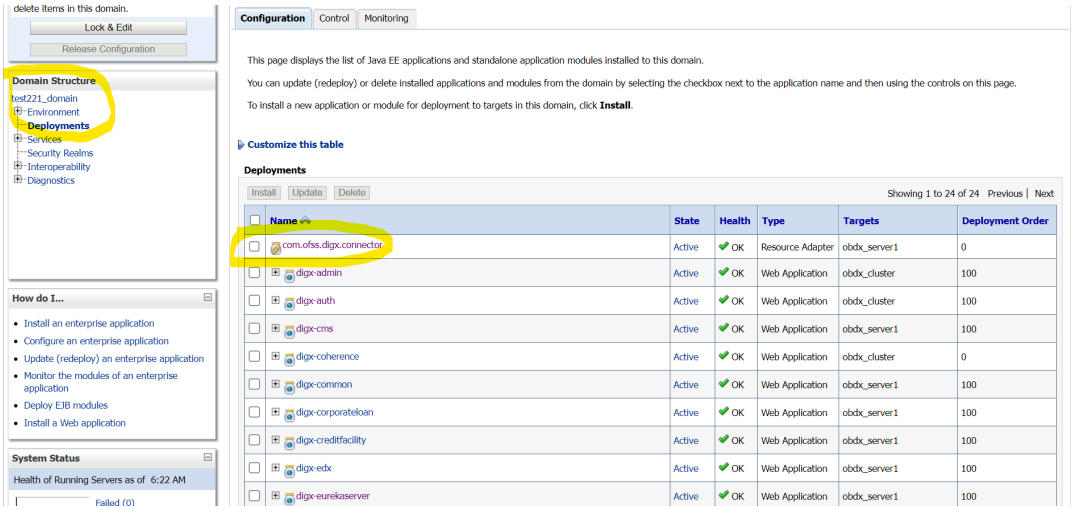
5. Enter JNDI name as ra/DIGXConnectorSSOKEY and click on **Finish**.

Figure 2-43 JNDI Configuration for Outbound Connection



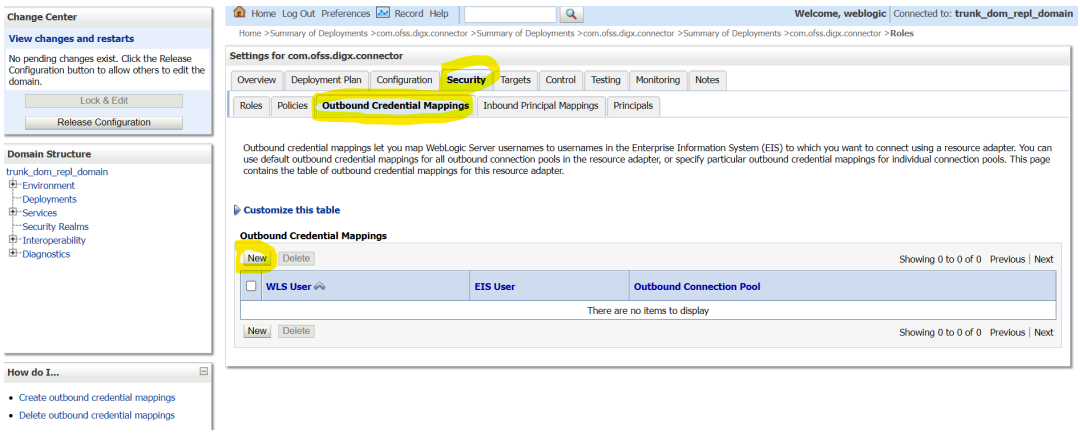
6. Again navigate to “Domain Structure” → “Deployments”.
7. Click on **“com.ofss.digx.connector”**.

Figure 2-44 Deployments



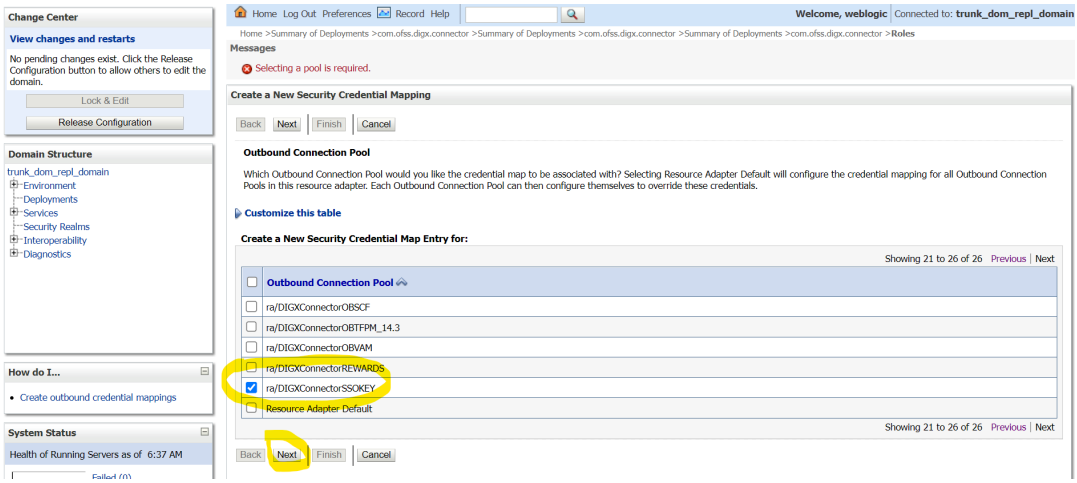
8. Navigate to “Security” → “Outbound Credentials Mapping” tab and click on **New**.

Figure 2-45 Outbound Credentials Mappings



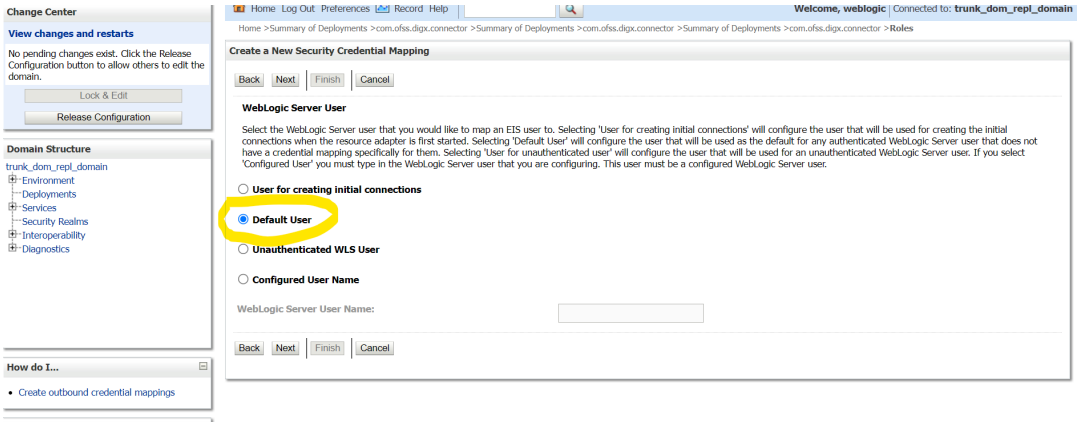
9. Select “ra/DIGXConnectorSSOKEY” by navigating using next button. Once selected as shown in below screenshot, click on **Next**.

Figure 2-46 Create New Security Credentials Mappings



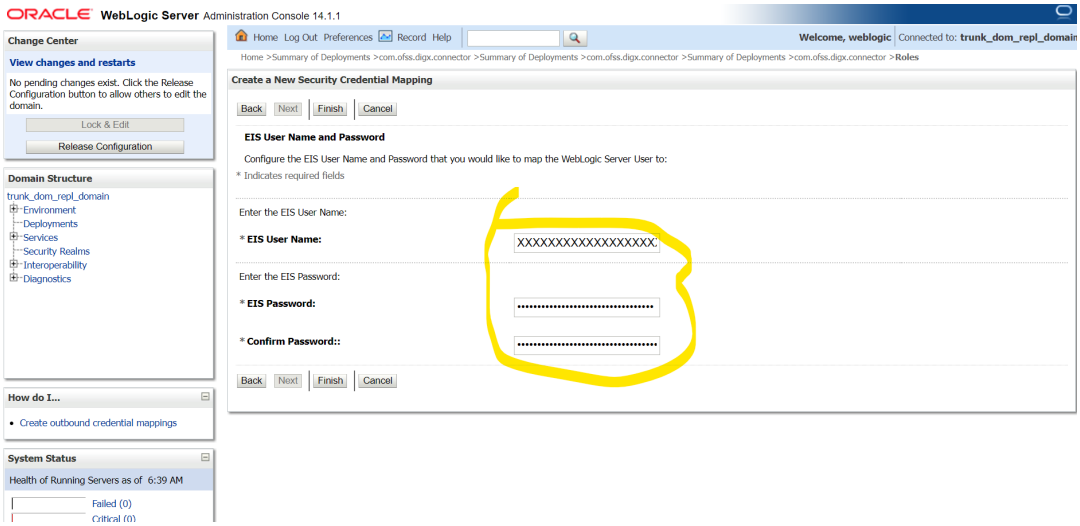
10. Select “Default User” and click on **Next**.

Figure 2-47 Create New Security Credentials Mappings



11. Provide the below mentioned field values as given below.
    a. EIS User Name: - Client ID save in txt file generated from IDCS in section 3.5, step 6.
    b. EIS Password: - Client Secret save in txt file generated from IDCS in section 3.5, step 6.
    c. EIS User Name: - Client Secret save in txt file generated from IDCS section 3.5, step 6.

Figure 2-48 Configure EIS UIS Username / Password



12. Click on **Finish** to save the configuration.

2.8 OBAPI configuration for OAuth

To enable IDCS out of the box support for OAuth follow below mentioned steps.

```
update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_URL> where prop_id = 'SSO_PROVIDER_URL';
```

1. Replace <SSO_PROVIDER_URL> with respective SSO provider URL.
2. Restart all the managed servers.

For configuring any other service provider, a custom class needs to be written which implements com.ofss.digx.app.sms.service.user.external.IExternalUser interface.

The entry for the new custom class has to be made in database using the below script -

```
update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_CLASS> where prop_id = 'SSO_PROVIDER_CLASS';
```

3. Replace <SSO_PROVIDER_CLASS> with the fully qualified name of the new custom class.
4. Also below queries need to be executed as well if there are any changes in the configuration-

```
update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_TOKEN_SCOPE> where prop_id = 'SSO_PROVIDER_TOKEN_SCOPE';update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_TOKEN_URI> where prop_id = 'SSO_PROVIDER_TOKEN_URI';update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_URL> where prop_id = 'SSO_PROVIDER_URL';update DIGX_FW_CONFIG_ALL_B set prop_value = <SSO_PROVIDER_USER_READ_URI> where prop_id = 'SSO_PROVIDER_USER_READ_URI';
```

5. Restart all the servers in domain.

2.9 Default Admin Configuration

OBAPI installer comes pre-shipped admin user with name “superadmin”,so in order to login into the OBAPI application for completing Day 1 maintenances the same user need to be created in SSO Provider with same name post SSO integration.

2.10 Logout Configurations

Below query needs to be executed as part of the logout configurations.

```
Insert into DIGX_FW_CONFIG_ALL_B (PROP_ID,CATEGORY_ID,PROP_VALUE,
FACTORY_SHIPPED_FLAG,PROP_COMMENTS,SUMMARY_TEXT,CREATED_BY,CREATION_DATE,
LAST_UPDATED_BY,LAST_UPDATED_DATE,OBJECT_STATUS,OBJECT_VERSION_NUMBER,
EDITABLE,CATEGORY_DESCRIPTION)


values
('SSO_LOGOUT_URL','dayoneconfig','<LOGOUT_URL>','Y',null,'SSO logout Url',
'ofssuser',sysdate,'ofssuser',sysdate,'A',1,'N',null);
```

Replace <LOGOUT_URL> with respective url.

Index