ORACLE

Integrate Oracle Analytics Server with Oracle Identity Cloud Service or IAM Identity Domain for Single Sign-On using App Gateway

Describes how to integrate Oracle Analytics Server with IDCS or IAM Domain for SSO using App Gateway.

May 2023, version 1.0

Public

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document since its initial publication.

DATE	REVISION
May 2023	Initial publication
March 2024	Updated publication

Authors: Veera Raghavendra Rao Koka.

Table of Contents

Introduction

App Gateway is a software appliance enabling you to integrate applications hosted either on a compute instance in Oracle Cloud or in an on-premises server with Oracle Identity Cloud Service (IDCS) or IAM Identity Domain for Single Sign-On (SSO) purposes.

Using the App Gateway, you can integrate the Oracle Analytics Server (OAS) with IDCS and IAM Identity Domain for SSO purposes.

App Gateway acts as a reverse proxy protecting OAS resources by restricting unauthorized network access. App Gateway intercepts any HTTP request to OAS and ensures that the users are authenticated with IDCS or IAM Identity Domain before forwarding the request to OAS. App Gateway propagates the authenticated user's identity to OAS.

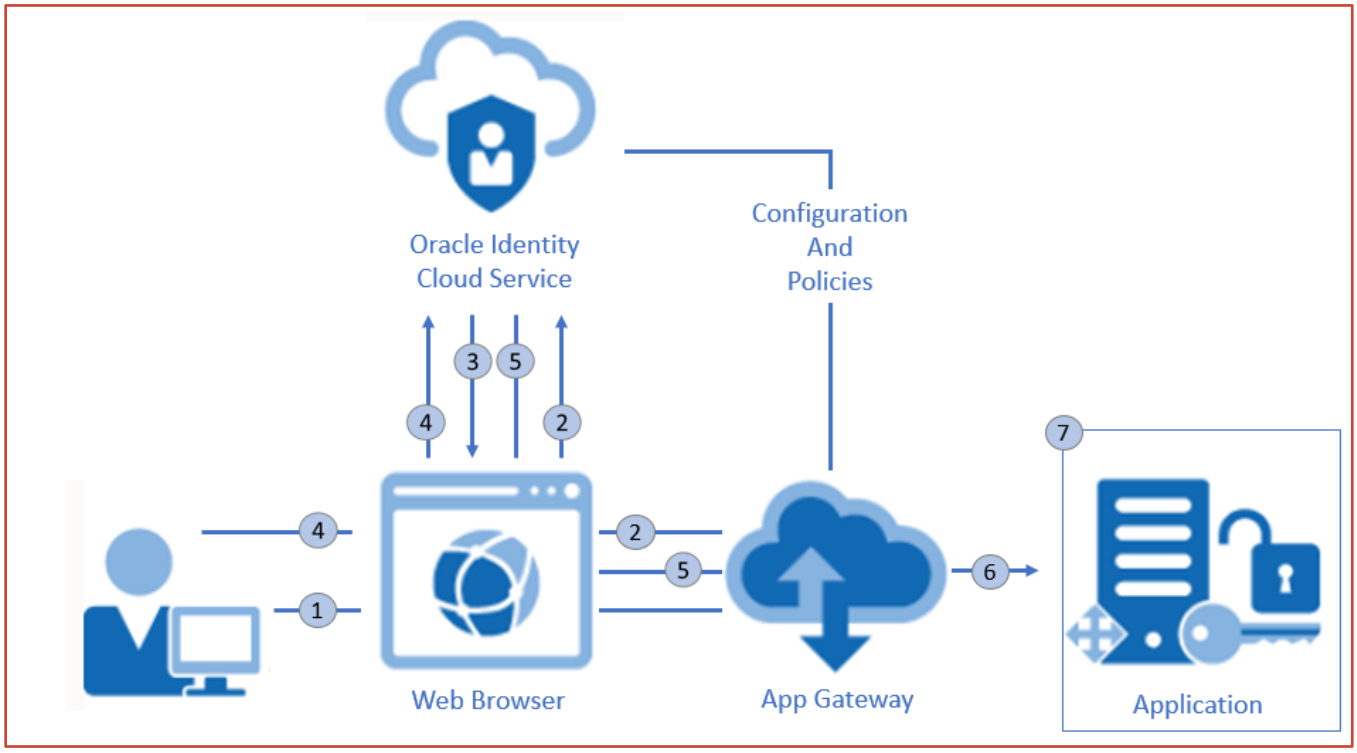
If the user is not authenticated with IDCS or IAM Identity Domain, then App Gateway redirects the user to the IDCS Sign-In page for credential validation.

This whitepaper describes the configuration of SSO for a single node OAS server running on Oracle Cloud using the App Gateway docker image on the same compute as the OAS server.

Another blog post will cover the configuration of SSO for a clustered OAS environment and the high availability of App Gateway.

The subsequent sections of this whitepaper will discuss the prerequisites and limitations concerning the configuration of SSO for OAS on-premises with App Gateway.

How App Gateway Works



Refer to the documentation to understand How IDCS App Gateway Works and How IAM App Gateway Works.

References

App Gateway is available for IDCS and IAM Identity Domain. Functionality is same except for the way you navigate, user interface, and download.

Understand App Gateway

See IDCS App Gateway documentation.

See IAM App Gateway documentation.

Prerequisites

- Oracle Analytics Server (5.9 and higher)
- OCI Compute Instance for App Gateway
- OCI Load Balancer (optional if App Gateway is running on a separate compute)

Configuration

SSO configuration of OAS delegates authentication to IDCS or IAM Identity Domain using App Gateway. For the authorization, OAS requires the users and groups from the IDCS or IAM Identity Domain to be available in OAS for application role management.

OAS to read the users and groups from IDCS or IAM Identity Domain and list them in OAS, you need to configure **Oracle Identity Cloud Integrator** as an authentication provider in the Oracle WebLogic Server of the OAS server.

Optional Configuration

Configuring OCI Load Balancer is an optional step, since we are suggesting to deploy the App Gateway docker image on the OAS compute instance, it would be better to have a load balancer in a public subnet and OAS compute on a private subnet.

Also block the direct access to the OAS server and ports when using OCI load balancer.

When the App Gateway docker image is deployed on a separate compute than the OAS server, you can avoid the use of a load balancer and block direct access to the OAS server and ports.

Install Docker Engine

Install on Oracle Linux 7

```
yum install docker-engine
systemctl enable docker
systemctl start docker
```

Docker commands as non-root user

```
sudo usermod -a -G docker oracle
```

Install on Oracle Linux 8

https://oracle-base.com/articles/linux/docker-install-docker-on-oracle-linux-ol8

```
dnf install -y dnf-utils zip unzip
dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
```

```
dnf remove -y runc
dnf install -y docker-ce --nobest
systemctl enable docker.service
systemctl start docker.service
```

Docker commands as non-root user

```
sudo usermod -a -G docker oracle
```

Oracle Identity Cloud Integrator

Oracle Analytics Server is now certified to use Oracle Identity Cloud Integrator to list Users and Groups from IDCS to OAS. Refer to Configure Oracle Identity Cloud Integrator as the Authentication Provider.

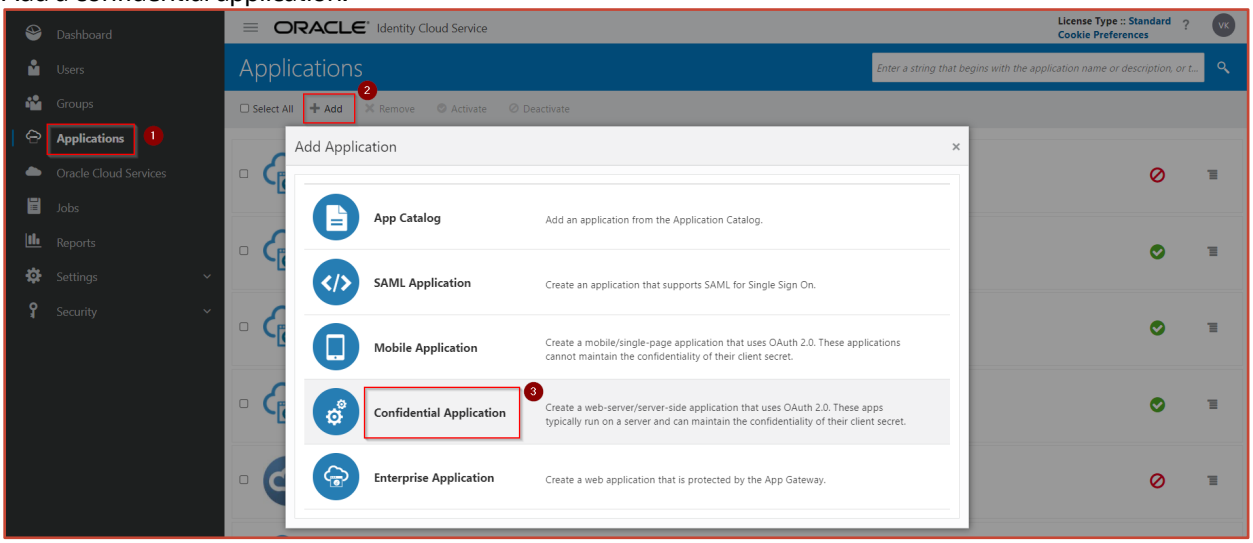
Oracle Analytics Server on Oracle Cloud can be integrated with IDCS or IAM Identity Domain for authentication and authorization using an **Oracle Identity Cloud Integrator** authentication provider type available in the Oracle WebLogic Server.

Since Oracle Identity Cloud Integrator is a System for Cross-domain Identity Management (SCIM) connector, it will not list the users and groups in the Oracle WebLogic Server; instead, it lists directly in the OAS console > Users and Roles.

It is also used as an Oracle WebLogic Identity Asserter to receive the user's identity sent by the App Gateway to fulfil SSO to OAS.

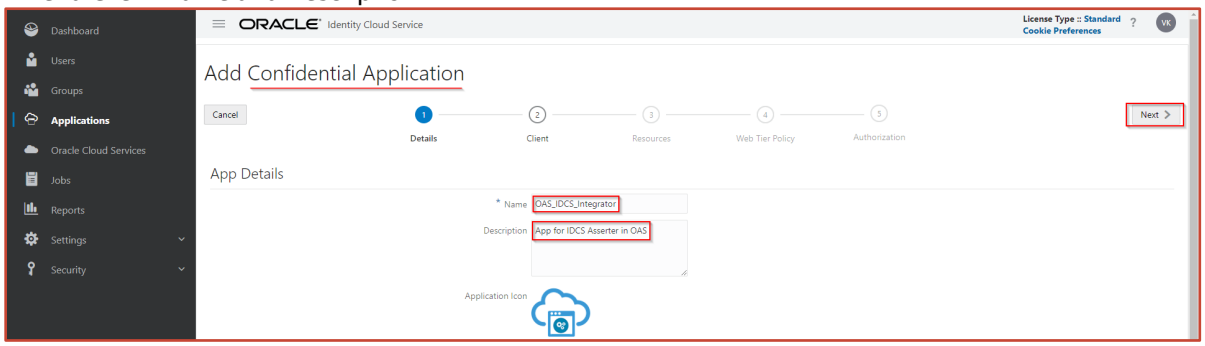
Create a Confidential Application for OAuth Client

1. Log in to Oracle Identity Cloud Service console as Identity Domain Administrator.
2. In the Oracle Identity Cloud Service console, create an OAuth client.
3. Add a confidential application.



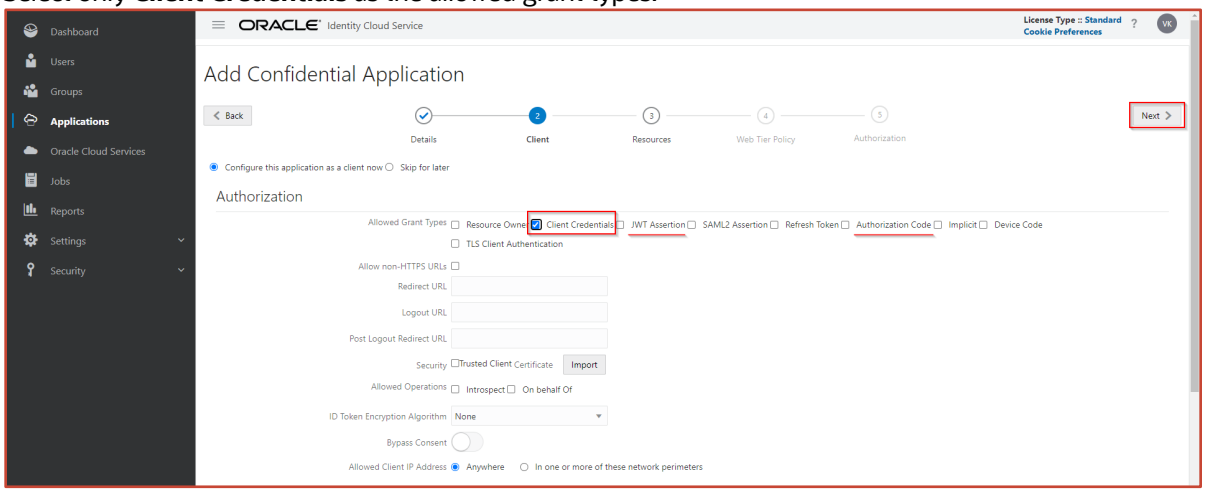
4. In the Application Wizard:

a. Enter a Client Name and Description

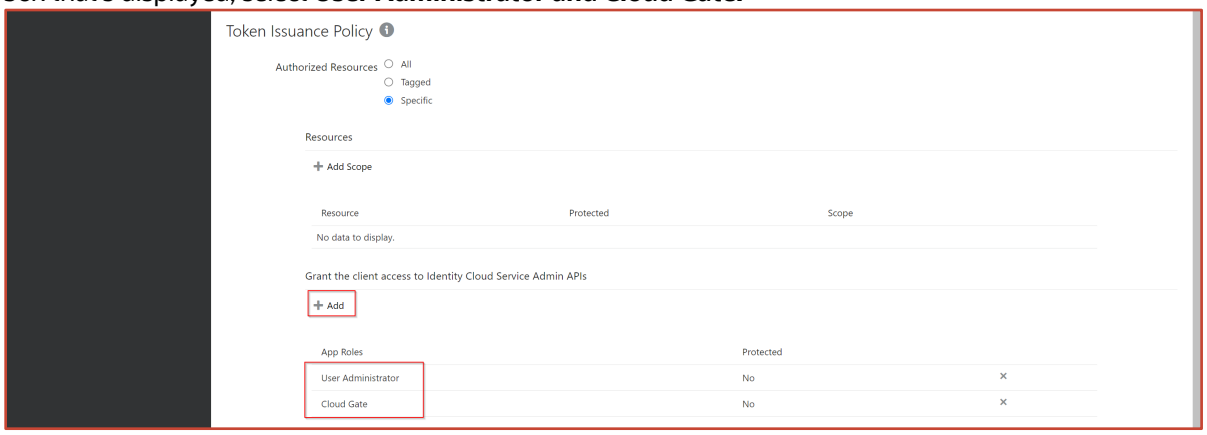


b. Select **Configure this application as a client now** to configure authorization settings.

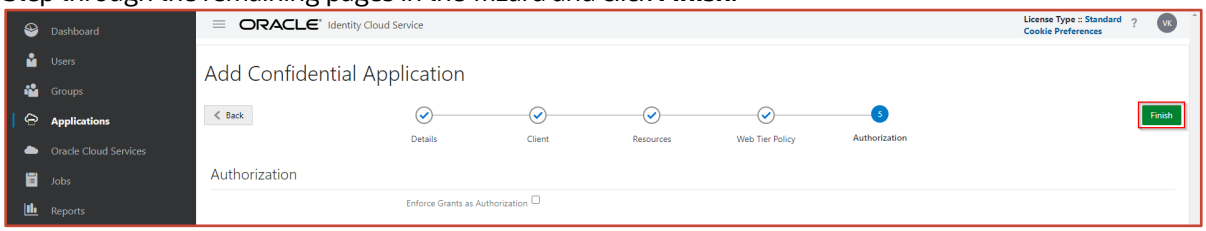
c. Select only **Client Credentials** as the allowed grant types.



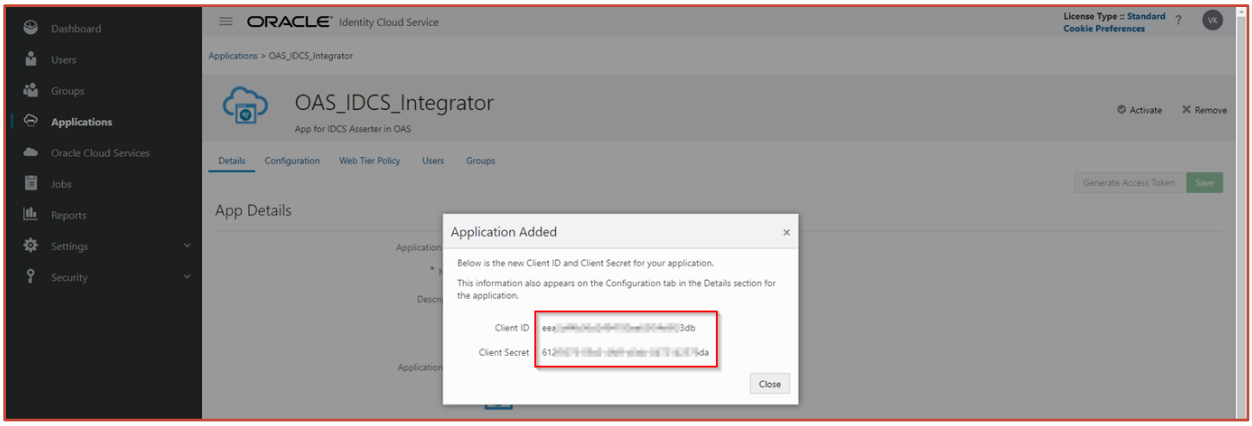
d. Assign the client to the User Administrator and Cloud Gate application roles. To do so, Click on Add button under **Grant the client access to Identity Cloud Service Admin APIs** section and then, in the box that's displayed, select **User Administrator and Cloud Gate.**



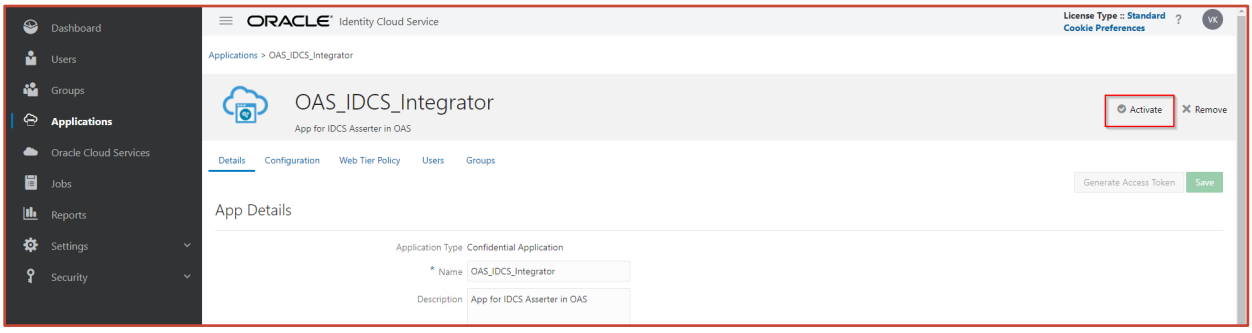
e. Step through the remaining pages in the wizard and click **Finish**.



5. Record the **Client ID** and **Client Secret** and note down the **Oracle Identity Cloud Service URL**.



6. Activate the application.



Required Configuration Attributes

To configure the Oracle Identity Cloud Integrator provider in Oracle WebLogic Server, you must provide the following attributes from the OAuth client:

- Tenant — *The name of the primary tenant in the Oracle Identity Cloud Service where you provisioned the OAuth client.*
- ClientId — *The OAuth client ID used to access the Oracle Identity Cloud Service identity store.*
- ClientSecret — *The OAuth Client Secret (password) used to generate access tokens.*
- Client tenant (Optional) — *The name of the OAuth client tenant in which the Client Id resides. This attribute isn't required if the Client tenant is the same as the primary tenant.*

Sample Data

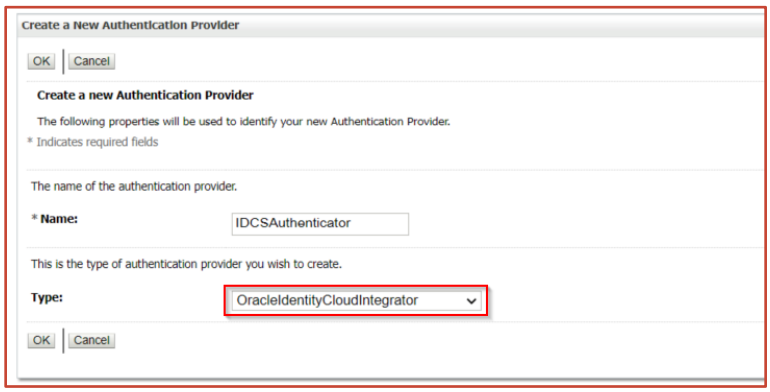
IDCS DNS Name: https://idcs-f5xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx403.identity.oraclecloud.com
Host: identity.oraclecloud.com
Tenant: idcs-f565xxxxxxxxxxxxxxxxxxxxxxxxxxxy403
Client ID: eea4xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx33db
Client Secret: 6xxxxxx6-0xx2-4xx9-axxb-0xxxxxxxxxa
Client Tenant: idcs-f565xxxxxxxxxxxxxxxxxxxxxxxxxxxy403

Configure the Oracle Identity Cloud Integrator Provider in the Oracle WebLogic Server

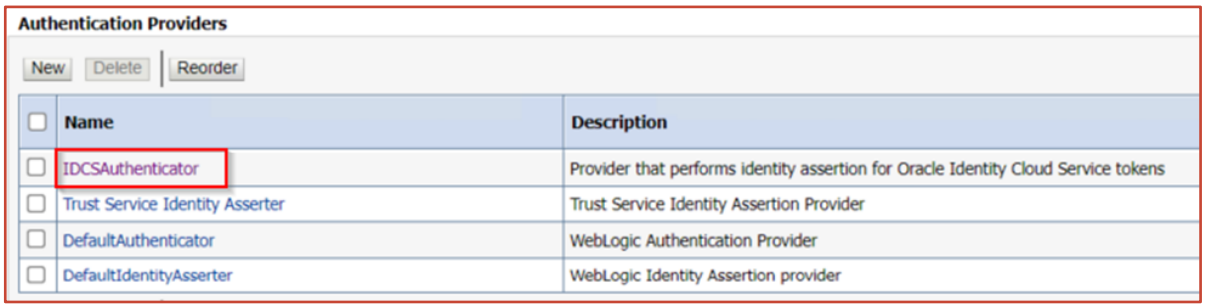
1. Log into Oracle WebLogic Server administration console.

   For example: `http://oas1.sub03xxxxxxxx91.oasvcn.oraclevcn.com:9500/console`

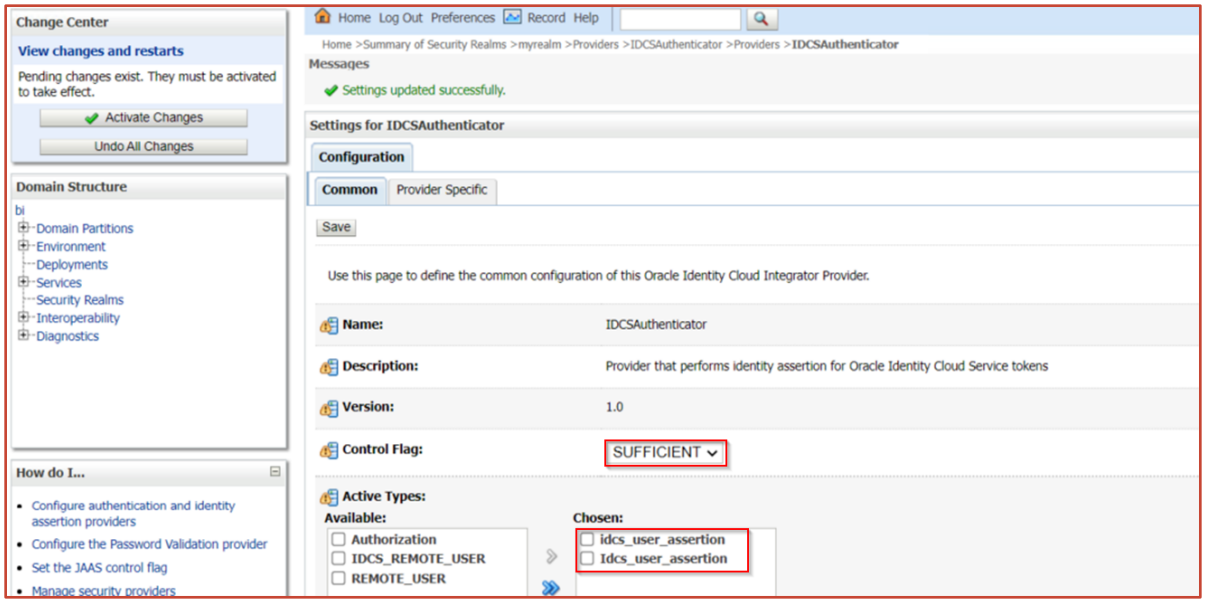
2. Click **Lock and Edit**.
3. Create a new Authentication Provider. Navigate to Security Realms > myrealm > Providers > New
4. For Type of Authentication Provider, select **OracleIdentityCloudIntegrator**.



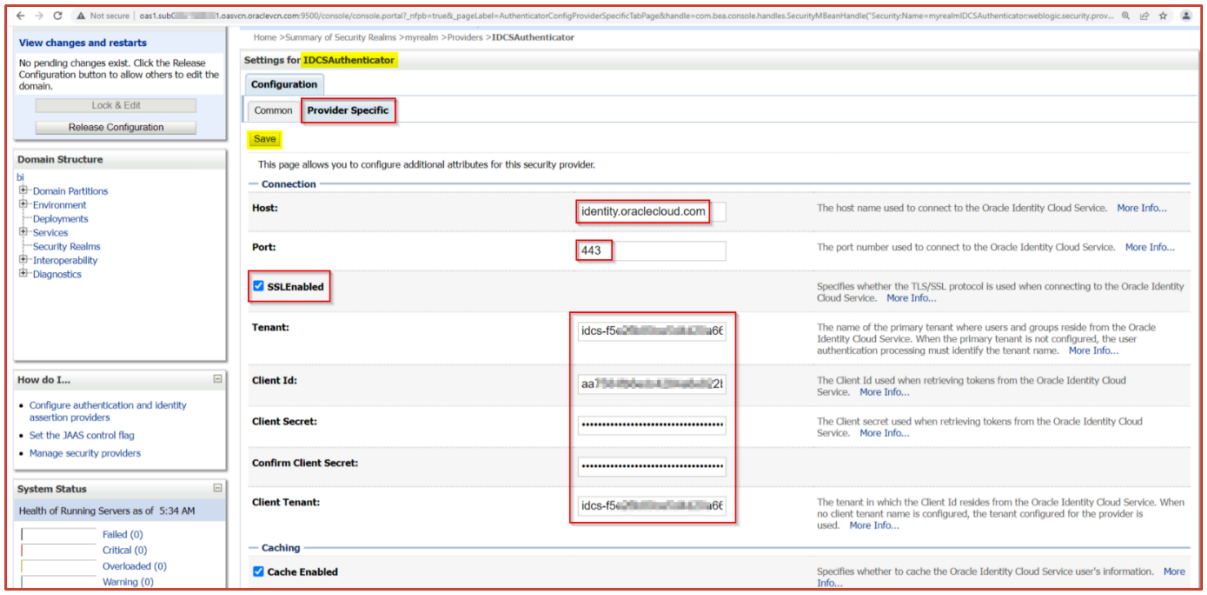
5. Reorder the providers.



6. Change the Control Flag for IDCSAuthenticator and DefaultAuthenticator to **SUFFICIENT**.
7. Go to the **Provider Specific** tab of IDCSAuthenticator and select the two types shown.



8. Using the details from the confidential application created for OAuth client in Oracle Identity Cloud Service, configure the **OracleIdentityCloudIntegrator.**



9. Click **Save** and **Activate Changes.**

Configuring TLS/SSL for the Oracle Identity Cloud Integrator Provider

The Oracle Identity Cloud Integrator provider supports one-way SSL. To secure the connection using TLS/SSL, you need to establish trust between Oracle WebLogic Server and Oracle Identity Cloud Service. To do so, you may need to obtain the Oracle Identity Cloud Service SSL certificate and import it into the Oracle WebLogic Server trust store.

If the Oracle Identity Cloud Service uses a well-known certificate authority (CA) such as Symantec, VeriSign, DigiCert, etc, and your WebLogic domain is using Java Standard Trust, then Oracle WebLogic Server trusts it by default and importing the certificate isn't required. If, however, your domain is configured for custom trust, you may need to import the Intermediate CA and root CA certificates into your trust store, regardless of whether Oracle Identity Cloud Service is using a well-known CA.

Note: In this example SSL offloading is done at the App Gateway Server.

To configure TLS/SSL:

On the Oracle Identity Cloud Integrator provider, set the following attributes:

- SSLEnabled — true
- idcsPort — The appropriate SSL port for Oracle Identity Cloud Service, for example 443

Configure hostname verification in the Oracle WebLogic Server

Configure hostname verification using the wild card host name verifier to allow WebLogic Server to accept certificates containing wildcards:

Add the property in the EXTRA_JAVA_PROPERTIES section of the DOMAIN_HOME/bin/setDomainEnv.sh script as:

```
-
Dweblogic.security.SSL.hostnameVerifier=weblogic.security.utils.SSLWLSWildcardHostnameVerifier
```

```
EXTRA_JAVA_PROPERTIES="-Djavax.management.builder.initial=weblogic.management.jmx.mbeanserver.WLSMBeanServerBuilder ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES

EXTRA_JAVA_PROPERTIES="-Dweblogic.security.SSL.hostnameVerifier=weblogic.security.utils.SSLWLSWildcardHostnameVerifier ${EXTRA_JAVA_PROPERTIES}"
export EXTRA_JAVA_PROPERTIES

# Set final environment user overrides, if available.
```

Change the idstore from ldap to scim in jps-config.xml

Edit the file `/u01/data/domains/bi/config/fmwconfig/jps-config.xml`

```
   <jpsContexts default="default">
      <jpsContext name="default">
         <serviceInstanceRef ref="credstore.db"/>
         <serviceInstanceRef ref="keystore.db"/>
         <serviceInstanceRef ref="policystore.db"/>
         <serviceInstanceRef ref="audit.db"/>
         <serviceInstanceRef ref="trust"/>
         <serviceInstanceRef ref="pdp.service"/>
         <serviceInstanceRef ref="attribute"/>
         <serviceInstanceRef ref="idstore.ldap"/>
      </jpsContext>
      <jpsContext name="bootstrap_credstore_context">
         <serviceInstanceRef ref="bootstrap.credstore"/>
         <serviceInstanceRef ref="keystore"/>
      </jpsContext>
      <jpsContext name="bootstrap_credstore_context_local">
         <serviceInstanceRef ref="bootstrap.credstore.local"/>
      </jpsContext>
   </jpsContexts>
</jpsConfig>
```

Change ldap to scim:

```
   <jpsContexts default="default">
      <jpsContext name="default">
         <serviceInstanceRef ref="credstore.db"/>
         <serviceInstanceRef ref="keystore.db"/>
         <serviceInstanceRef ref="policystore.db"/>
         <serviceInstanceRef ref="audit.db"/>
         <serviceInstanceRef ref="trust"/>
         <serviceInstanceRef ref="pdp.service"/>
         <serviceInstanceRef ref="attribute"/>
         <serviceInstanceRef ref="idstore.scim"/>
      </jpsContext>
      <jpsContext name="bootstrap_credstore_context">
         <serviceInstanceRef ref="bootstrap.credstore"/>
         <serviceInstanceRef ref="keystore"/>
      </jpsContext>
      <jpsContext name="bootstrap_credstore_context_local">
         <serviceInstanceRef ref="bootstrap.credstore.local"/>
      </jpsContext>
   </jpsContexts>
</jpsConfig>
```

Optionally, obtain the root CA certificate from the Oracle Identity Cloud Service's server and import it into the appropriate trust store in your WebLogic Server domain.

This step is **not required** if the Oracle Identity Cloud Service uses a **well-known CA**.

- If your domain uses a KSS trust store, import the certificate as described in Configuring the OPSS Keystore Service for Custom Identity and Trust: Main Steps.
- If your domain uses the JKS trust store, see Importing Certificates into the Trust and Identity Keystores.

Restart all the Oracle Analytics Server Services

`$DOMAIN_HOME/bitools/bin/stop.sh`

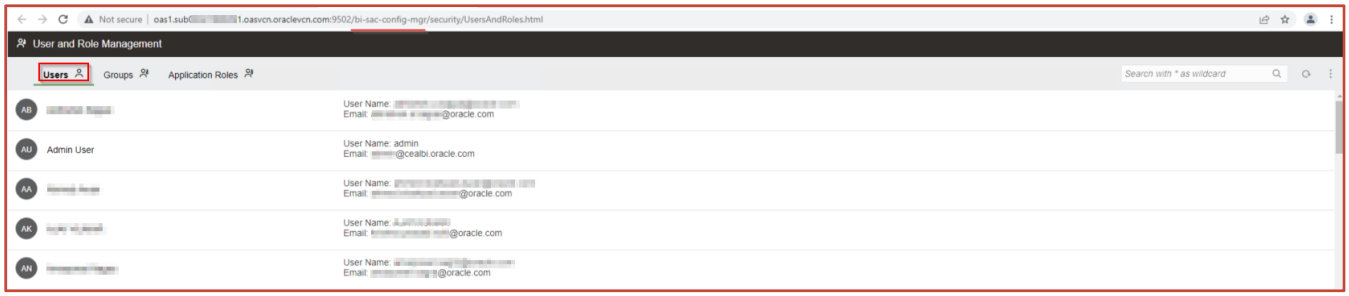
`$DOMAIN_HOME/bitools/bin/start.sh`

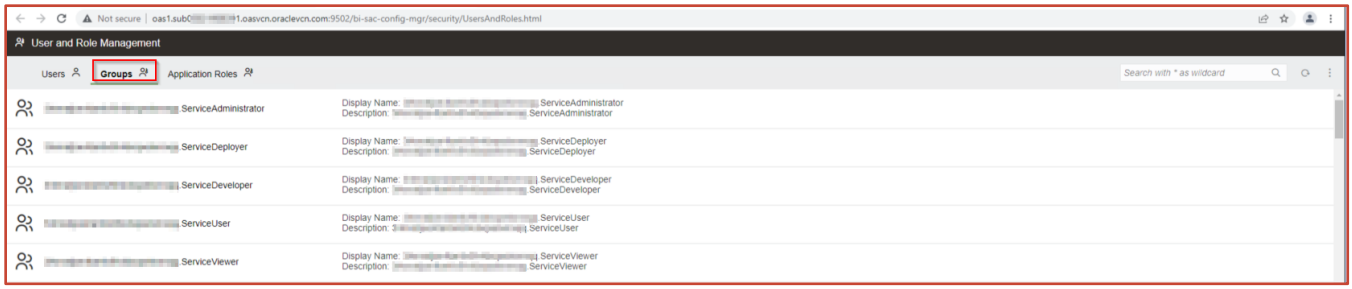
Managing Users and Groups from IDCS or IAM Domain

Users and groups from IDCS or IAM Domain aren't listed in Oracle WebLogic Server administration console. You manage users and groups from the Console in Oracle Analytics Server.

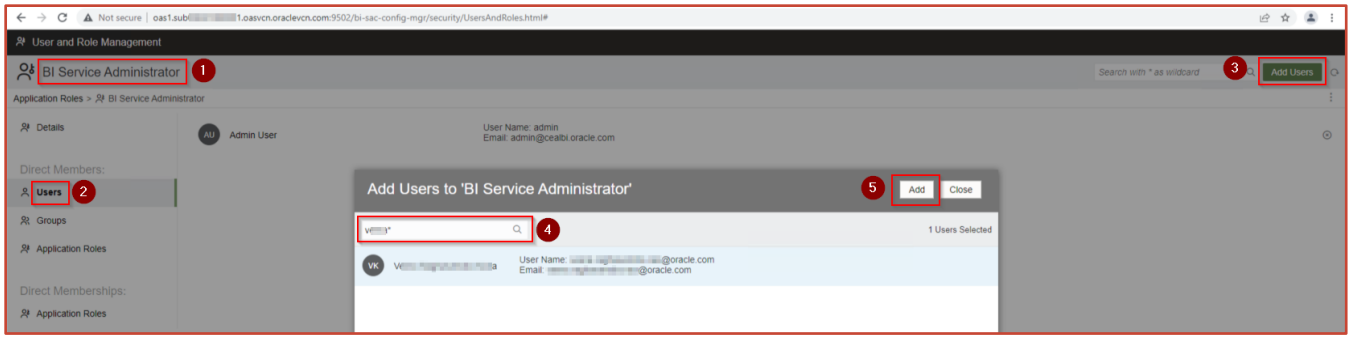
Cannot create users and groups in Oracle Analytics Server Console > Users and Roles but can add the IDCS or IAM Domain users and groups to OAS Application Roles.

Sign-in to Oracle Analytics Server and navigate to **Console** > **Users and Roles**.



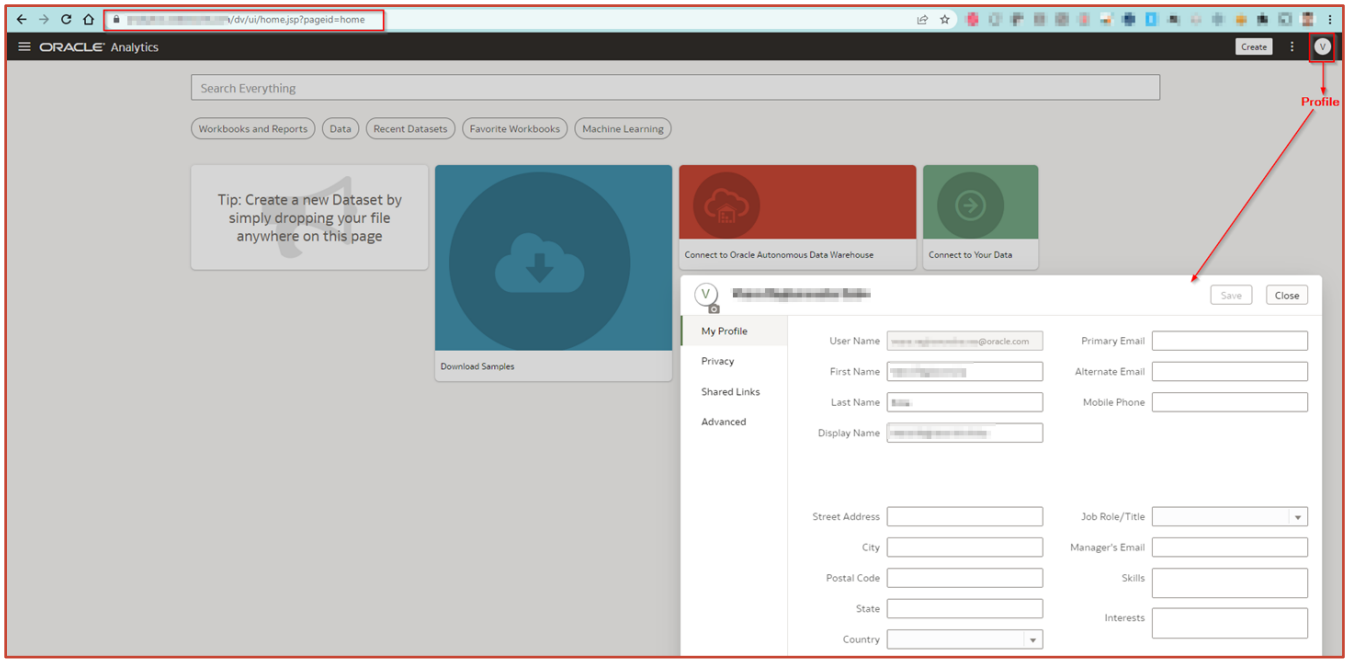


Assign users and groups in Oracle Identity Cloud Service to one or more application roles in Oracle Analytics Server.



Log into Oracle Analytics Server as an Oracle Identity Cloud Service User

NOTE: Here you enter IDCS/IAM Domain native username and password in the OAS login page.



NOTE: If you have federated users, they cannot login to OAS using OAS login page as they don't have a password in IDCS/IAM Domain to validate.

To allow all users federated and non-federated (native) in IDCS/IAM Domain able to login to OAC, you need to configure Single Sign-On for OAS with IDCS/IAM Domain so that the federated users further select the external SAML Identity Provider such as Azure, Okta, ADFS, any SAML 2.0 IdP and login using the external SAML IdP login page or method.

To configure Single Sign-On for OAS with IDCS/IAM Domain, use the App Gateway.

Oracle Identity Cloud Service App Gateway

Oracle Analytics Server is now certified for the use of IDCS App Gateway for Single Sign-On refer to Configure SSO with Oracle Identity Cloud Service and App Gateway.

Install IDCS App Gateway Server; there are three approaches:

1. Install App Gateway on Oracle Cloud Infrastructure.
2. Install App Gateway Using Oracle VM Virtual Box Software.
3. Deploy the Oracle App Gateway Docker Container.

Install App Gateway on OCI Compute or Oracle VM Virtual Box, refer to Configuring SSO for OBIEE12c/OAS Running On On-Premise or On OCI Compute with IDCS Using App Gateway (Doc ID 2611016.1).

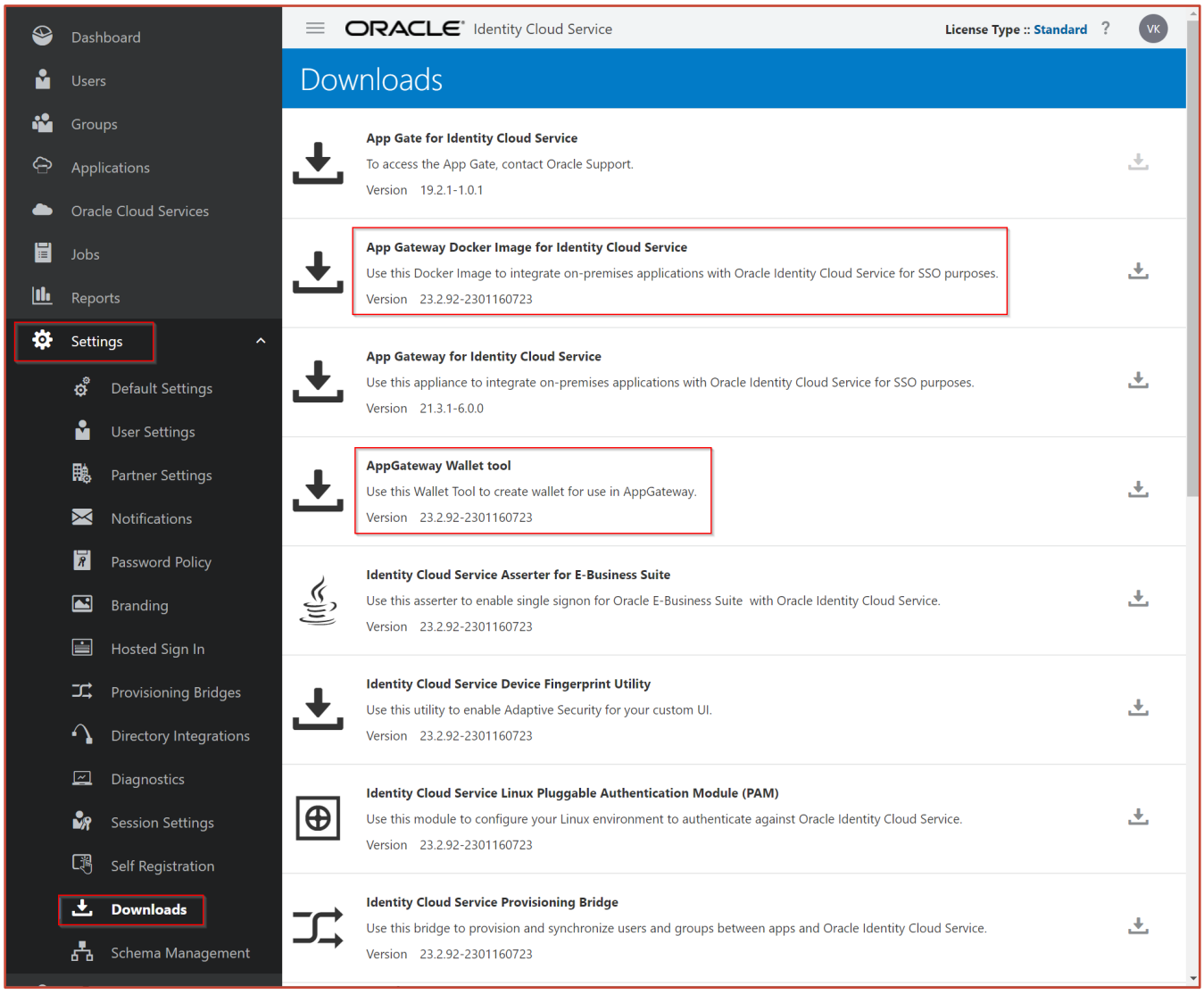
Here we demonstrate the deployment of IDCS App Gateway docker container.

App Gateway Docker Image

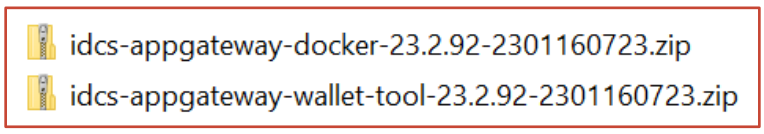
Reference: https://docs.oracle.com/en/cloud/paas/identity-cloud/uaids/manage-oracle-identity-cloud-service-app-gateways1.html

This approach uses IDCS App Gateway Docker Image

1. Log in to the Oracle Identity Cloud Service (IDCS) admin console.
2. Navigate to Settings → Downloads and Download the App Gateway Docker Image for Identity Cloud Service.
3. Also Download App Gateway Wallet Tool.



Download the App Gateway Wallet Tool from the IDCS admin console.



Copy the Zip files to the OCI Compute Instance such as the OAS Server.

Register an App Gateway

Before installing the binary file for App Gateway that appears on the **Downloads** page, you must register your App Gateway using the Identity Cloud Service console.

To register an App Gateway, you must add hosts and associate each host to an enterprise application your App Gateway will protect:

In the **Hosts** pane, you define host identifiers.
Each host identifier represents a domain name and port number App Gateway uses to proxy an enterprise application.
In the **Apps** pane, you associate an enterprise application with a host identifier.

To register an App Gateway, you must have either the **Identity Domain Administrator** or **Security Administrator** roles.

1. In the Identity Cloud Service console, expand the **Navigation Drawer** > click **Security** > click **App Gateways >** click **Add**.
2. In the **Details** pane, specify your App Gateway's name, then click **Next** (**>**).
3. In the **Hosts** pane, click **Add**.
4. In the Add Host dialog, provide a name in the Host Identifier field.
5. Enter the Host and Port values that the App Gateway server will respond to HTTP requests. App Gateway server uses the port number in the above step to respond to HTTP requests.
6. Select the SSL Enabled check box to have your App Gateway listen to HTTP requests in secure mode (HTTPS). Otherwise, clear this check box, and your App Gateway will only listen to non-secure HTTP requests.
7. If you select the SSL Enabled check box, then populate the Additional Properties text area with the following values to specify the certificate key pair the App Gateway server will use, protocols, and ciphers for SSL:

   ```
   ssl_certificate /usr/local/example.com.rsa.crt;
   ssl_certificate_key /usr/local/example.com.rsa.key;
   ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
   ssl_ciphers HIGH:!aNULL:!MD5;
   ```

   The **/usr/local/example.com.rsa.crt** is the full path of a certificate file in the App Gateway server.
   The **/usr/local/example.com.rsa.key** is the secret key of that certificate file.
   You must upload both files to the App Gateway server after you install the App Gateway binary file.

Note: Starting with App Gateway OVA version 20.4.1-4.0.0, App Gateway will work only in SSL/HTTPS mode.
Note the following considerations:

a. If there is no load balancer in front of App Gateway, then populate the **Additional Properties** as specified above.
b. If App Gateway is running behind a load balancer, then the load balancer must be listening over SSL/HTTPS.

c. If the load balancer is listening over SSL/HTTPS and the App Gateway is not SSL enabled, the load balancer must pass the header (Name: is_ssl Value: ssl) to App Gateway.

8. In the Add Host dialog, click Save.
9. In the Hosts pane, click Next >.
10. If you have previously registered an enterprise application in Oracle Identity Cloud Service, click Add in the Apps pane. See Assign an Enterprise Application to an App Gateway
11. Click Finish.
12. In the App Gateway Details page, note the value of the Client ID.
13. Click Show Secret and note the value of the Client Secret.

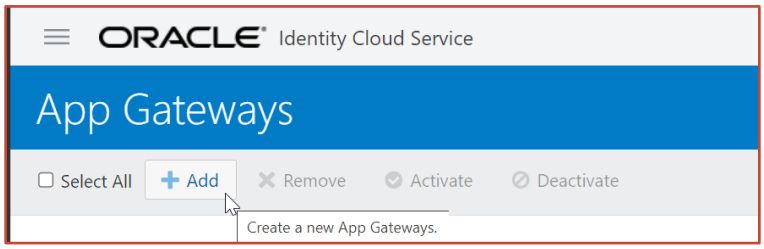
    The **Client ID** and **Client Secret** are equivalent to a credential (for example, an ID and password) that your App Gateway server uses to communicate with Oracle Identity Cloud Service. You'll need these values when you configure the App Gateway server.
14. In the Navigation Drawer, click App Gateways.
15. In the App Gateways page, select your App Gateway, click Activate, and then click OK in the Confirmation window to activate your App Gateway.

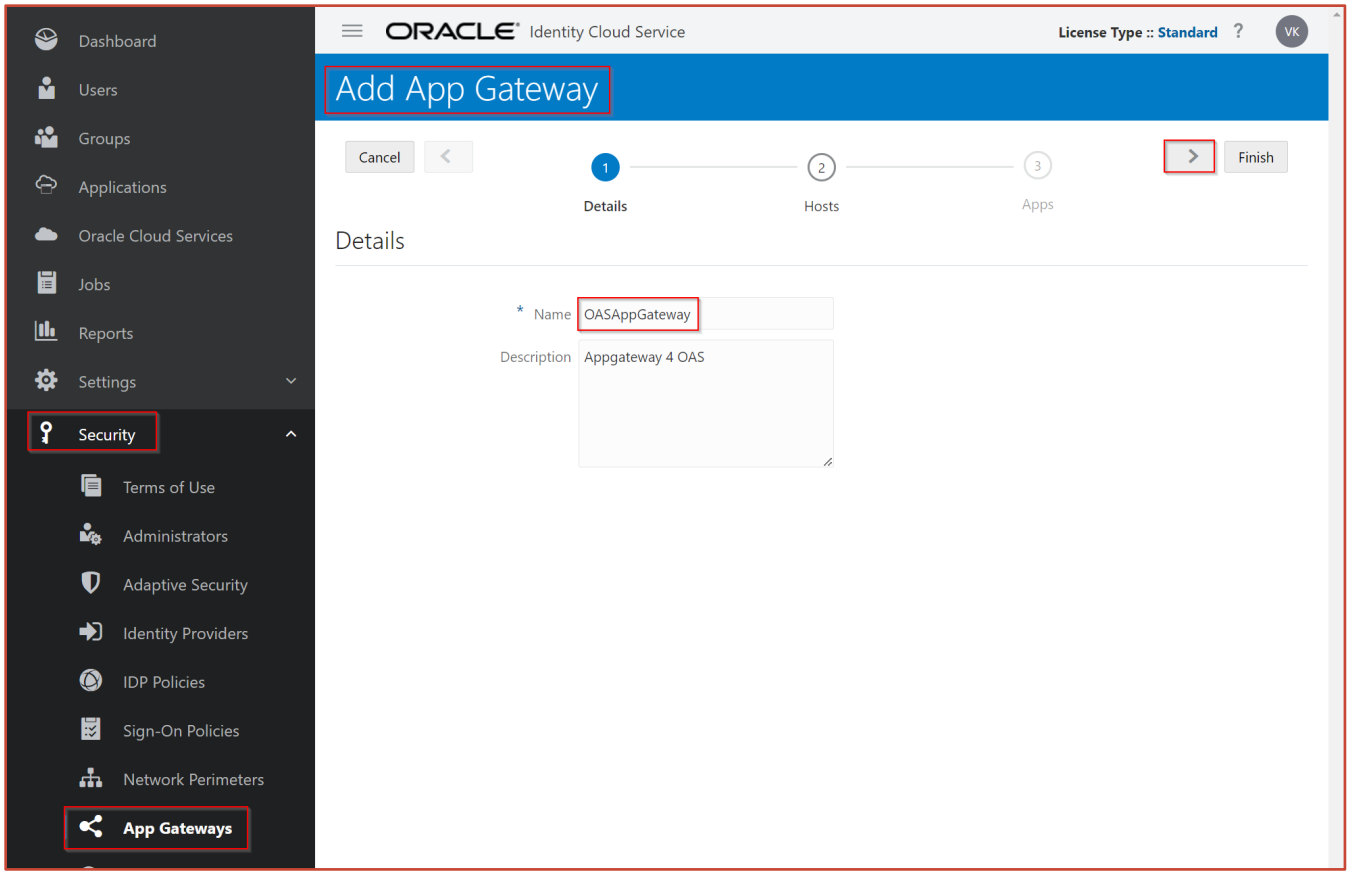
Configured SSL certificates must be copied to the location specified in **Additional Properties**.

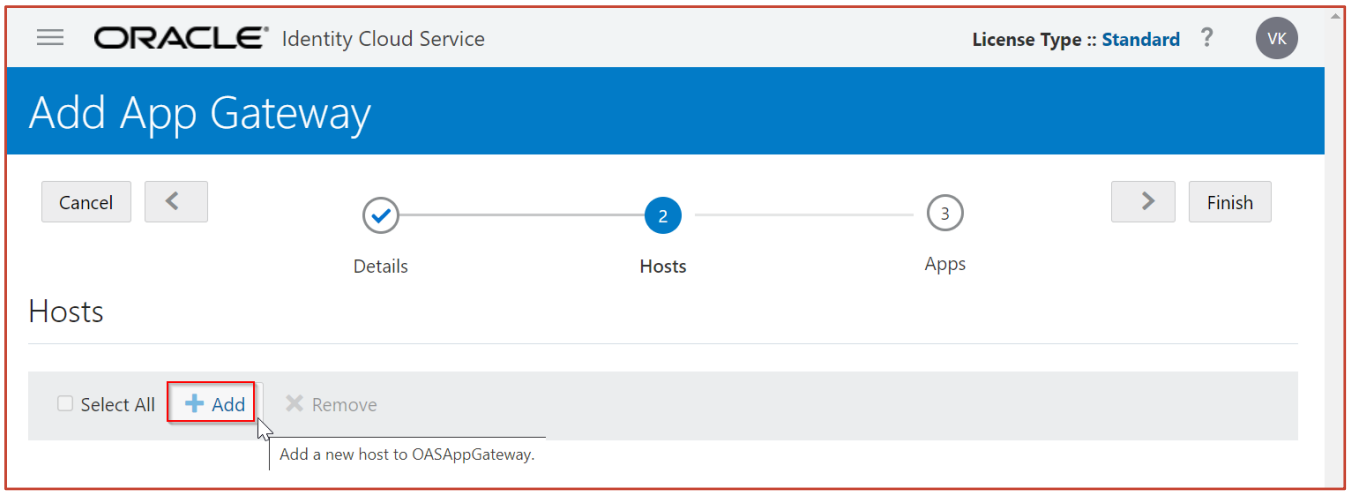
Go to **Security**, **App Gateways**, **<Gateway>**, **Hosts**, **Additional Properties,** and note the location.

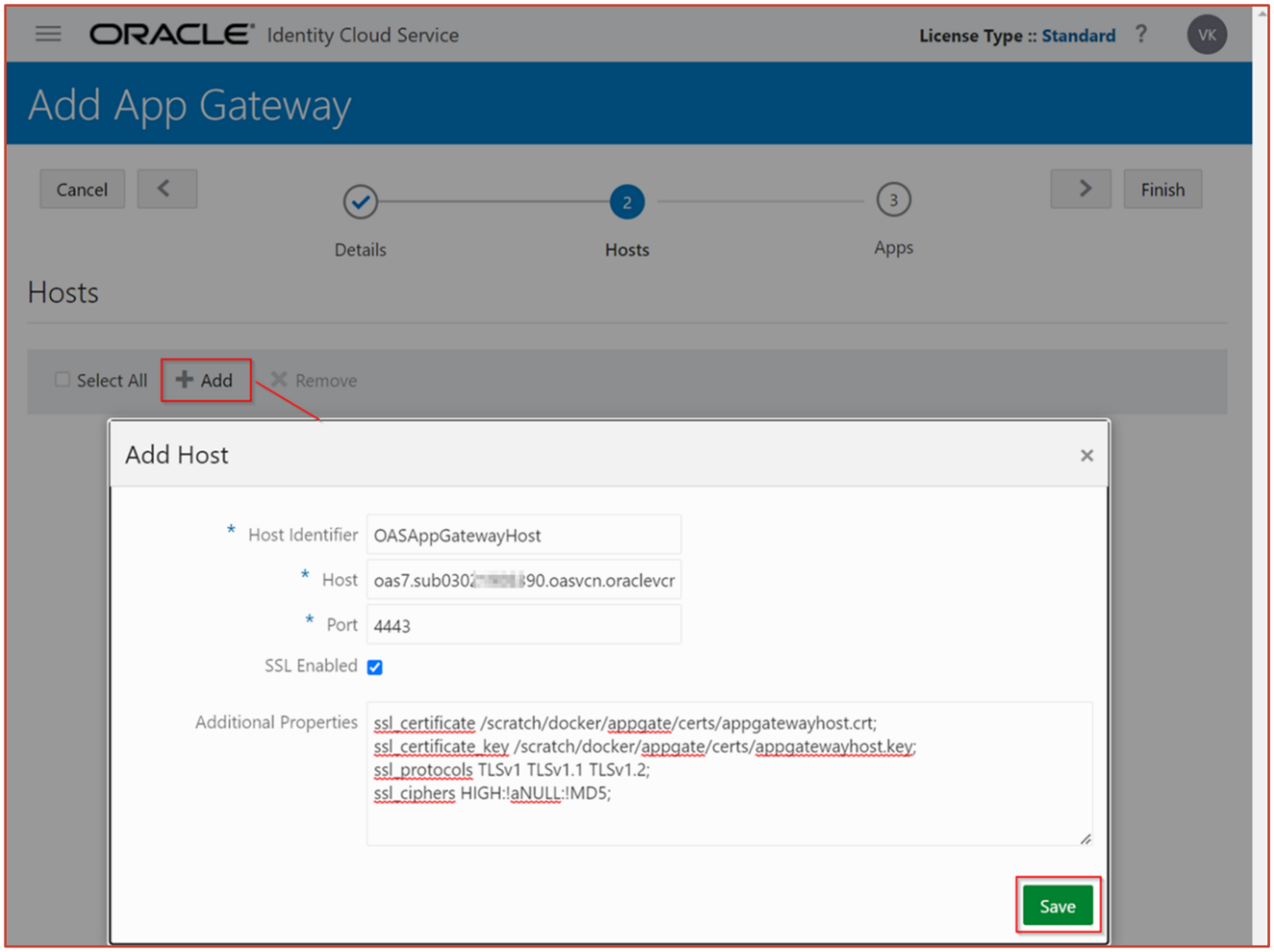
Run commands like the following.

Note: The location of the certificate depends on the location you specified in the App Gateways Host.



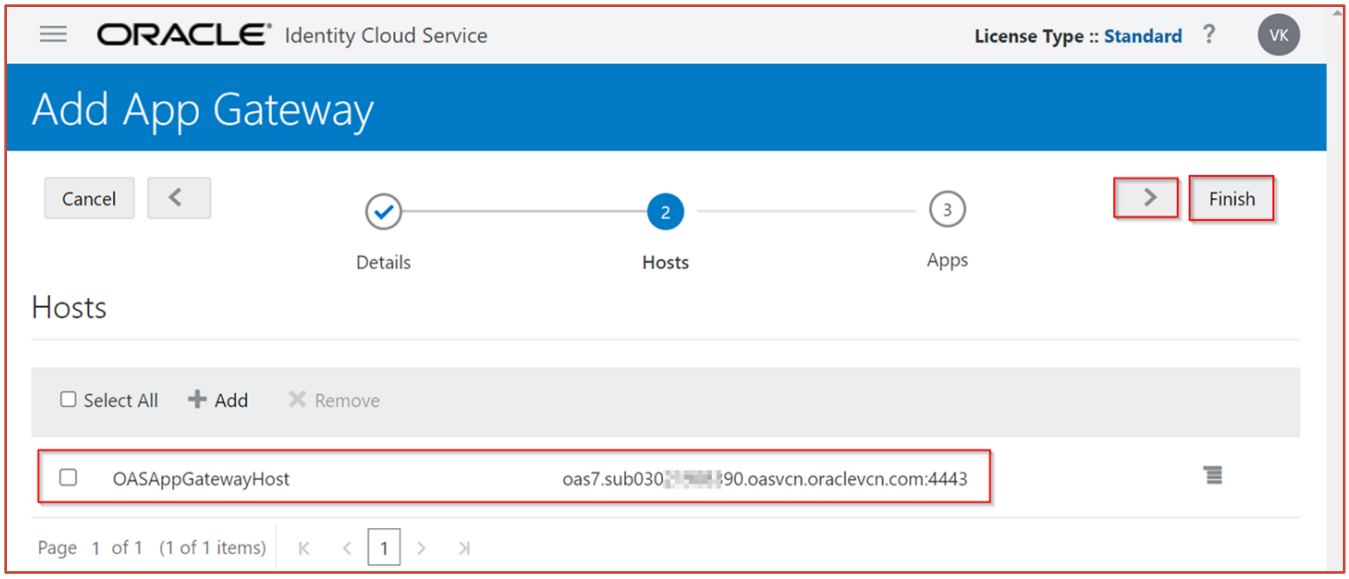




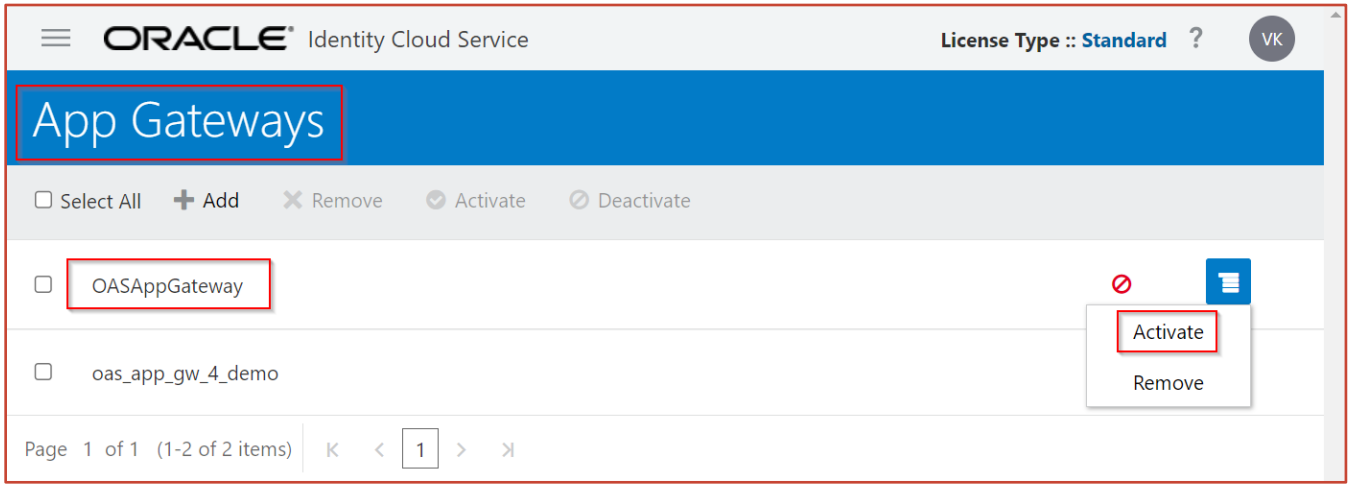



```
ssl_certificate /scratch/docker/appgate/certs/appgatewayhost.crt;
ssl_certificate_key /scratch/docker/appgate/certs/appgatewayhost.key;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers HIGH:!aNULL:!MD5;
```

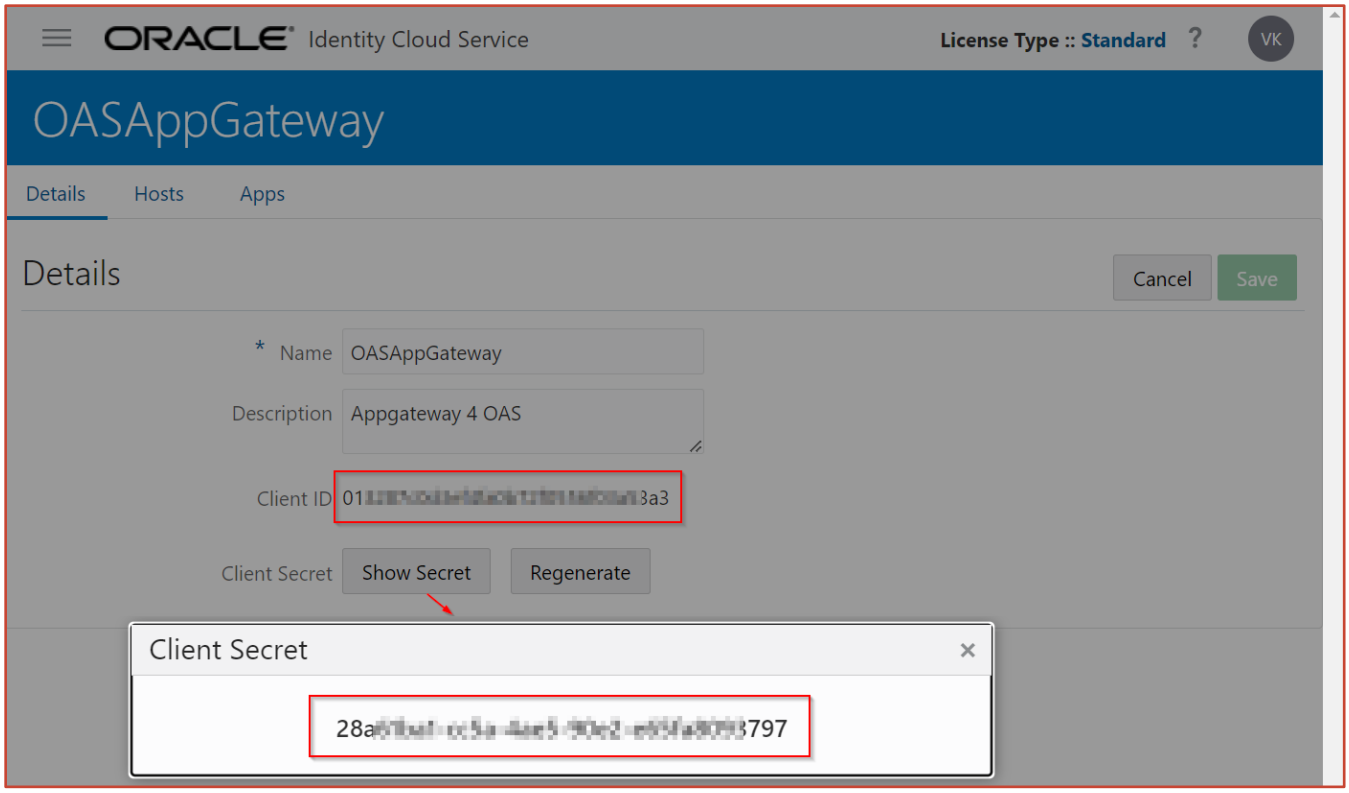
Note: Change the SSL protocols based on your application security specifications.



Click on Finish; You Can Add the Enterprise Application later.

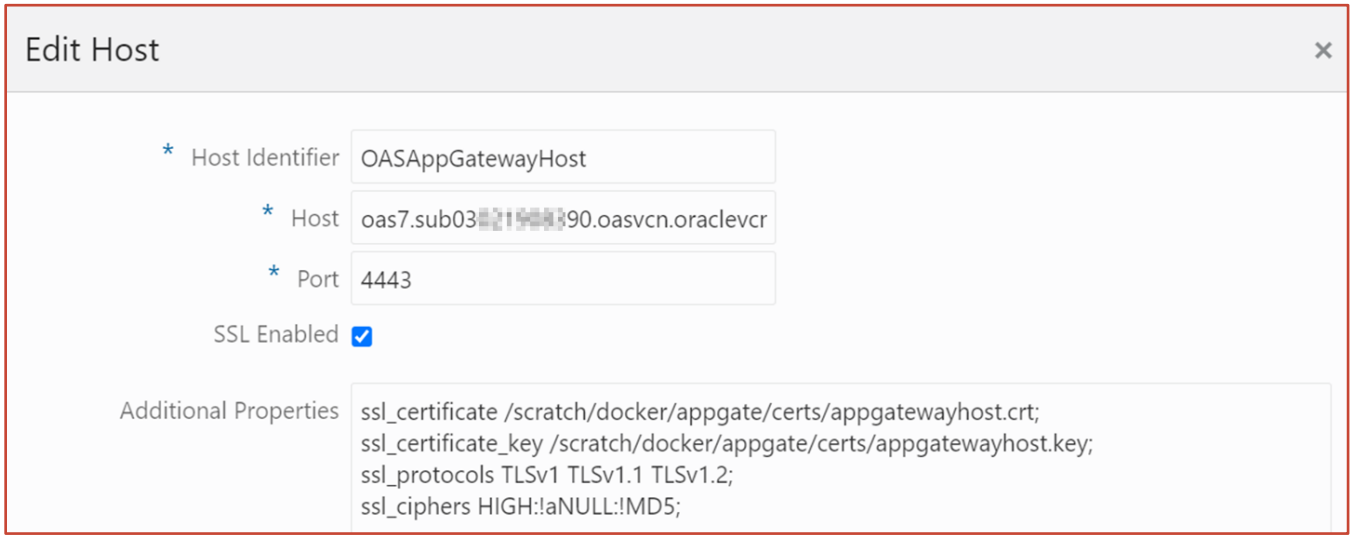


Activate the App Gateway and open the App Gateway.



Copy the Client ID and Client Secret values of the Registered App Gateway for later use in the Docker Container deployment.

Click on the Hosts tab and select the App Gateway Host.



Create a Wallet

```
sudo su oracle

mkdir -p /u01/data/AppGateway/wallet

unzip /tmp/idcs-appgateway-wallet-tool-23.2.92-2301160723.zip -d
/u01/data/AppGateway/wallet
```

Get the **Client ID** and **Client Secret** values of the Registered App Gateway

Run the below command:

```
env LD_LIBRARY_PATH=./lib ./cgwallettool --create -i <ClientID>

Enter the Client Secret:

<ClientSecret>
```

```
[oracle@oas7 wallet]$ ls
cgwallettool  FileInfo.json  lib  README
[oracle@oas7 wallet]$ env LD_LIBRARY_PATH=./lib ./cgwallettool --create -i 01328[redacted]a53a3

Enter the Client Secret:
28a[redacted]97

================================================================================

 IDCS AppGateway Wallet Tool - Create Wallet

    Create a Wallet file and populate it with:
        -- Client ID:     01328[redacted]a53a3
        -- Client Secret: **************

    Wallet file (cwallet.sso) will be written to: ./
    NOTE: An existing file will be re-used. Not replaced.

================================================================================

SUCCESS! A cwallet.sso Wallet file can be found in: ./

[oracle@oas7 wallet]$ ls
cgwallettool  cwallet.sso  FileInfo.json  lib  README
```

Copy the **`cwallet.sso`** file from `/u01/data/AppGateway/wallet` to **`/u01/data/AppGateway`** folder.

Set the File permissions to **644** or **755.**

Load the Docker Image

```
sudo su oracle

mkdir -p /u01/data/AppGateway/wallet

unzip idcs-appgateway-docker-23.2.92-2301160723.zip -d /u01/data/AppGateway/docker
```

```
[oracle@oas7 docker]$ ls
appgateway-23.2.92-2301160723.tar.gz  FileInfo.json
```

Run the below commands:

```
docker load < appgateway-23.2.92-2301160723.tar.gz
```

```
[oracle@oas7 docker]$ docker load < appgateway-23.2.92-2301160723.tar.gz
62d9c4cbe8f4: Loading layer [==================================================>]    142MB/142MB
1eab6675b5fe: Loading layer [==================================================>]  12.09MB/12.09MB
26b2ce35ed2d: Loading layer [==================================================>]  42.87MB/42.87MB
9c22c28e018c: Loading layer [==================================================>]  80.21MB/80.21MB
a4e858de6097: Loading layer [==================================================>]  11.31MB/11.31MB
dd3b106e3c79: Loading layer [==================================================>]  2.881MB/2.881MB
0d7ee6ab38e9: Loading layer [==================================================>]  17.41kB/17.41kB
50d26cb6fdb0: Loading layer [==================================================>]  64.51kB/64.51kB
41d4bed46fbf: Loading layer [==================================================>]  28.67kB/28.67kB
700d9ad94139: Loading layer [==================================================>]  3.878MB/3.878MB
cdea03083ee3: Loading layer [==================================================>]  116.4MB/116.4MB
158d8d2fe1ef: Loading layer [==================================================>]  62.46kB/62.46kB
b43e2db918c7: Loading layer [==================================================>]  4.608kB/4.608kB
11a3d4e14bb1: Loading layer [==================================================>]  120.4MB/120.4MB
e964f5f11347: Loading layer [==================================================>]   2.46MB/2.46MB
Loaded image: idcs/idcs-appgateway:23.2.92-2301160723
```

Command: `docker images`

```
[oracle@oas7 docker]$ docker images
REPOSITORY              TAG                    IMAGE ID       CREATED       SIZE
idcs/idcs-appgateway    23.2.92-2301160723     f444ed3e3655   4 weeks ago   525MB
```

Create the Environment Variables for App Gateway

Set the following environment variables in the appgateway-env file to run the App Gateway Docker container.

Get the nameserver value of the App Gateway Host Server

`cat /etc/resolv.conf`

```
nameserver 169.254.169.254
```

`cat /u01/data/AppGateway/appgateway-env`

```
CG_APP_TENANT=idcs-f5exxxxxxxxxxxxxxxxxxxxxxxx0403
IDCS_INSTANCE_URL=https://idcs-f5exxxxxxxxxxxxxxxxxxxxxxxx0403.identity.oraclecloud.com
NGINX_DNS_RESOLVER=169.254.169.254
```

```
CG_APP_TENANT=idcs-f5e[redacted]0403
IDCS_INSTANCE_URL=https://idcs-f5e[redacted]0403.identity.oraclecloud.com
NGINX_DNS_RESOLVER=169.254.169.254
```

```
[oracle@oas7 AppGateway]$ ls -l
total 8
-rw-rw-r--. 1 oracle oracle 176 Feb 14 11:00 appgateway-env
-rwxr-xr-x. 1 oracle oracle 557 Feb 14 11:12 cwallet.sso
drwxrwxr-x. 2 oracle oracle 100 Feb 14 10:32 docker
drwxrwxr-x. 3 oracle oracle 120 Feb 14 10:24 wallet
```

Prerequisite

Before you use the Bridge Network configuration, add/update iptables to true in the file `/etc/docker/daemon.json`.

The above config allows the Docker daemon to edit the iptables filter rules required for port mapping.

```
docker run -it -p 8080:4443 -d --name appgateway --env-file
/u01/data/AppGateway/appgateway-env --env HOST_MACHINE=`hostname -f` --volume
/u01/data/AppGateway/cwallet.sso:/usr/local/nginx/conf/cwallet.sso --net=bridge-net
idcs.docker.example.com/idcs/appgateway:RELEASE-BUILDNUMBER
```

Note: Docker internally updates the iptables/firewalld with the routes for the port when we run the above command.

`cat /etc/docker/demon.json`

```
{"iptables": "true"}
```

NOTE: We started the Docker Container with the network as host and ignored the above step.

Start docker

```
docker run -dit --name appgateway --env-file /u01/data/AppGateway/appgateway-env --env
HOST_MACHINE=`hostname -f` --volume
/u01/data/AppGateway/cwallet.sso:/usr/local/nginx/conf/cwallet.sso --net=host idcs/idcs-
appgateway:23.2.92-2301160723
```

Test if the docker container started

Command: `docker ps`

```
[oracle@oas7 AppGateway]$ docker ps
CONTAINER ID   IMAGE                                      COMMAND                  CREATED          STATUS          PORTS     NAMES
9344b142c901   idcs/idcs-appgateway:23.2.92-2301160723   "/bin/sh -c '$CLOUDG…"   22 seconds ago   Up 21 seconds             appgateway
```

Generate SSL Certificate and Private Key for App Gateway to run on HTTPS

Copy the **`generate_certs.sh`** script to the `/u01/data/AppGateway` folder and execute the script

Using the below generate_certs.sh script, generate the Private key and Certificate Signing Request

Get the CSR file Signed by the Organization Certificate Authority.

CA will give you the Signed Certificate or use the Self-Signed Certificate.

NOTE: You can also use the script and generate a self-signed certificate.

----------generate_certs.sh----------

```
#!/bin/bash
host=`hostname -f`
# Generating new server key
openssl genrsa -aes256 -passout pass:Oracle123 -out server.key 2048
# Taking backup of the server.key
cp server.key server-orig.key
# Removing the PassPhrase from the server.key
openssl rsa -passin pass:Oracle123 -in server-orig.key -out server.key
# deleting the backup of the key
rm server-orig.key
# Generating server certificate sign request, i.e., server.csr
openssl req -subj "/C=US/ST=California/L=RedwoodShores/O=Oracle Cloud Services/OU=Oracle
Analytics Server/CN=$host" -out server.csr -key server.key -new -sha256
# Get the CSR signed by a well-known or Company Certificate Authority
# Signing the CSR and generating a self-signed certificate
openssl x509 -req -days 365 -sha256 -in server.csr -signkey server.key -out server.crt
```

Download the Script to Generate SSL Certificates

generate_certs.sh

chmod +x generate_certs.sh

Rename the server.crt and server.key to the respective filenames, such as **appgatewayhost.crt** and **appgatewayhost.key**

NOTE: These filenames should match those in the Host entry while registering the App Gateway in IDCS Administration Console.

Set the File permissions of the Certificate and Private Key to **644** or **755.**

While starting the Docker Container, mount the wallet, SSL certificate, and private key at runtime.

```
[oracle@oas7 AppGateway]$ ls -l
total 24
-rw-rw-r--. 1 oracle oracle  176 Feb 14 11:00 appgateway-env
-rwxr-xr-x. 1 oracle oracle 1415 Feb 14 11:59 appgatewayhost.crt
-rwxr-xr-x. 1 oracle oracle 1675 Feb 14 11:59 appgatewayhost.key
-rwxr-xr-x. 1 oracle oracle  557 Feb 14 11:12 cwallet.sso
drwxrwxr-x. 2 oracle oracle  100 Feb 14 10:32 docker
-rwxrwxr-x. 1 oracle oracle  791 Feb 14 11:52 generate_certs.sh
-rw-rw-r--. 1 oracle oracle 1102 Feb 14 11:59 server.csr
drwxrwxr-x. 3 oracle oracle  120 Feb 14 10:24 wallet
```

Stop the Docker Container

```
docker stop appgateway
```

```
docker rm appgateway
```

Start the Docker Container

```
docker run -dit --name appgateway --env-file /u01/data/AppGateway/appgateway-env --env
HOST_MACHINE=`hostname -f` --volume
/u01/data/AppGateway/cwallet.sso:/usr/local/nginx/conf/cwallet.sso --volume
/u01/data/AppGateway/appgatewayhost.crt:/scratch/docker/appgate/certs/appgatewayhost.crt
--volume
/u01/data/AppGateway/appgatewayhost.key:/scratch/docker/appgate/certs/appgatewayhost.key
--net=host idcs/idcs-appgateway:23.2.92-2301160723
```

Open the Required Port for App Gateway

Open the port specified in the Host entry while registering the App Gateway in IDCS Administration Console.

For example, 4443.

```
sudo su root
```

```
sudo firewall-cmd --zone=public --permanent --add-port=4443/tcp
```

```
sudo firewall-cmd --complete-reload
```

NOTE: Also Open the same port for example, 4443 in the Security List of the Private/Public Subnets of the VCN on OCI, respectively.

Assign an Enterprise Application to an App Gateway

Refer to identity cloud documentation.

Update the App Gateway registration in the Oracle Identity Cloud Service console and add an enterprise application that interacts with App Gateway.

1. In the Identity Cloud Service console, expand the **Navigation Drawer**, click **Security**, click **App Gateways**, and then click the name of your App Gateway.
2. Click the **Apps** tab, and then click **Add**.
3. In the **Assign an App to gate** window, map App Gateway to an enterprise application using the values below, and then click **Save**.
   - **Application**: Select the enterprise application you want to protect using this App Gateway. See About Enterprise Applications.
     Note: The enterprise application must be in activated status.
   - **Select a Host**: Select the host identifier to which the App Gateway will proxy the enterprise application.
   - **Resource Prefix**: Enter the URL prefix used by App Gateway to proxy the enterprise application.
     For example, use / to represent every request since root path will be forwarded to your selected enterprise application.
   - **Origin Server**: This is the actual base URL where the application is hosted. If the application is not directly accessible but accessible through a web proxy, then enter the URL of the web proxy. See the example diagram below.
   - **Additional Properties**: This field is used to provide additional configuration for the application. The values specified into the field are nginx directives or statements, which will be part of the location block in nginx.conf. Examples of when you would do this are:
     a. If protected applications need to do further redirects or to access resources after successful authentication with App Gateway, you can use this field to populate the host header with the correct value and pass it to the application.
        For example, if a user accesses the application using `https://myappgateway.example.com:4443/dv`, the browser passes the host header to App Gateway with the value set to Host: myappgateway.example.com:4443. This value is passed by App Gateway to the downstream application, and you can achieve this by setting either of the values below in Additional Properties:
        ```
        proxy_set_header host "myappgateway.example.com:4443";
        ```
        or
        ```
        proxy_set_header host $http_host;
        ```

        `$http_host` is a variable and its value is populated with the host header App Gateway receives from the browser or from any client.

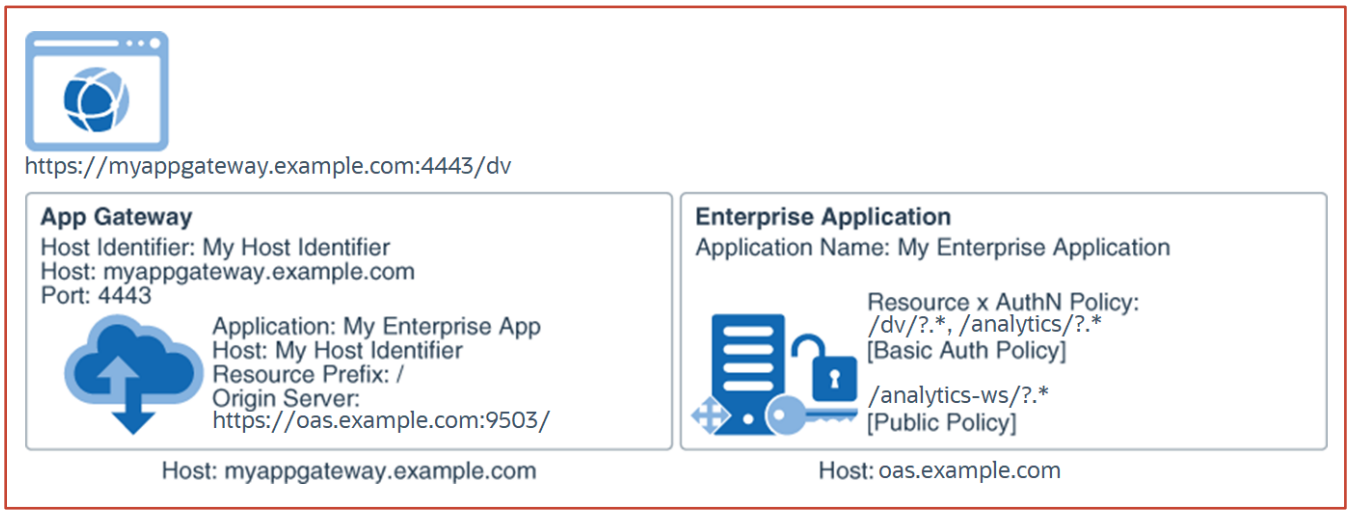
        Note: If there are load balancers sitting behind App Gateway, then it is the job of the load balancers to forward the actual host header to app gateway so that `$http_host` is populated with the correct value and App Gateway can forward it to the application.

     b. If the application is accessible through a web proxy, then enter the values below:
        ```
        proxy_set_header host "oas.example.com";
        ```

        The "`oas.example.com`" domain is the domain name where the application is hosted, also known as origin server.

        In this case, App Gateway can't pass the host header received from browser or other client and applications cannot do further redirects via App Gateway.

The following figure provides examples of the mappings that you configure between App Gateway and your enterprise application:



Note: You can assign multiple enterprise applications to the same App Gateway; consequently, the same App Gateway server will protect these applications.

Make sure for each application, the value of Resource Prefix differs. For example, if you have `http://oas.example.com:9502/analytics` and `http://bip.example.com:9000/xlmpserver`, both accessible through `http://myappgateway.example.com:4443/` App Gateway URL, then enter `/analytics` as **Resource Prefix** when you register application 1, and `/xmlpserver` as **Resource Prefix** when you register application 2.

After you assign the application to your App Gateway, you may need to restart the App Gateway server for the changes to be effective immediately.

Create an Enterprise Application

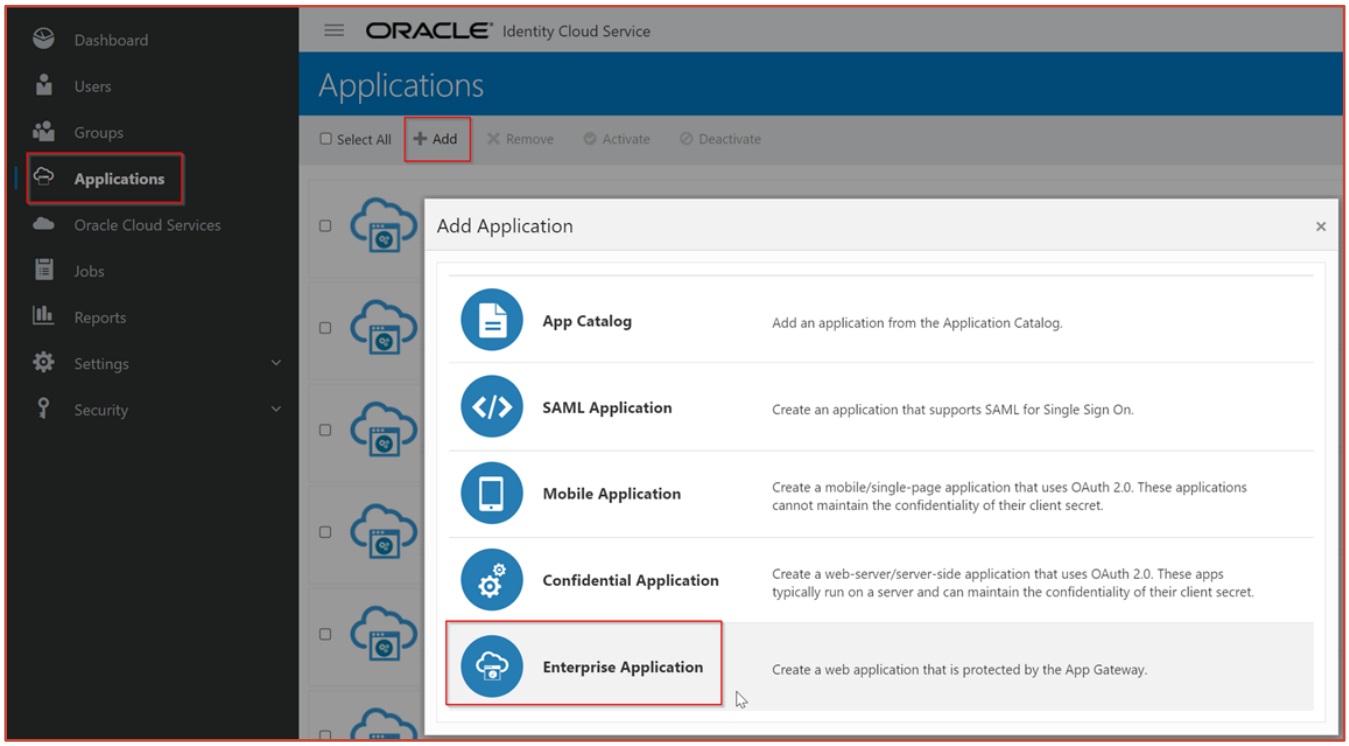
An enterprise application enables you to secure web applications that are protected by the Oracle App Gateway. See documentation.

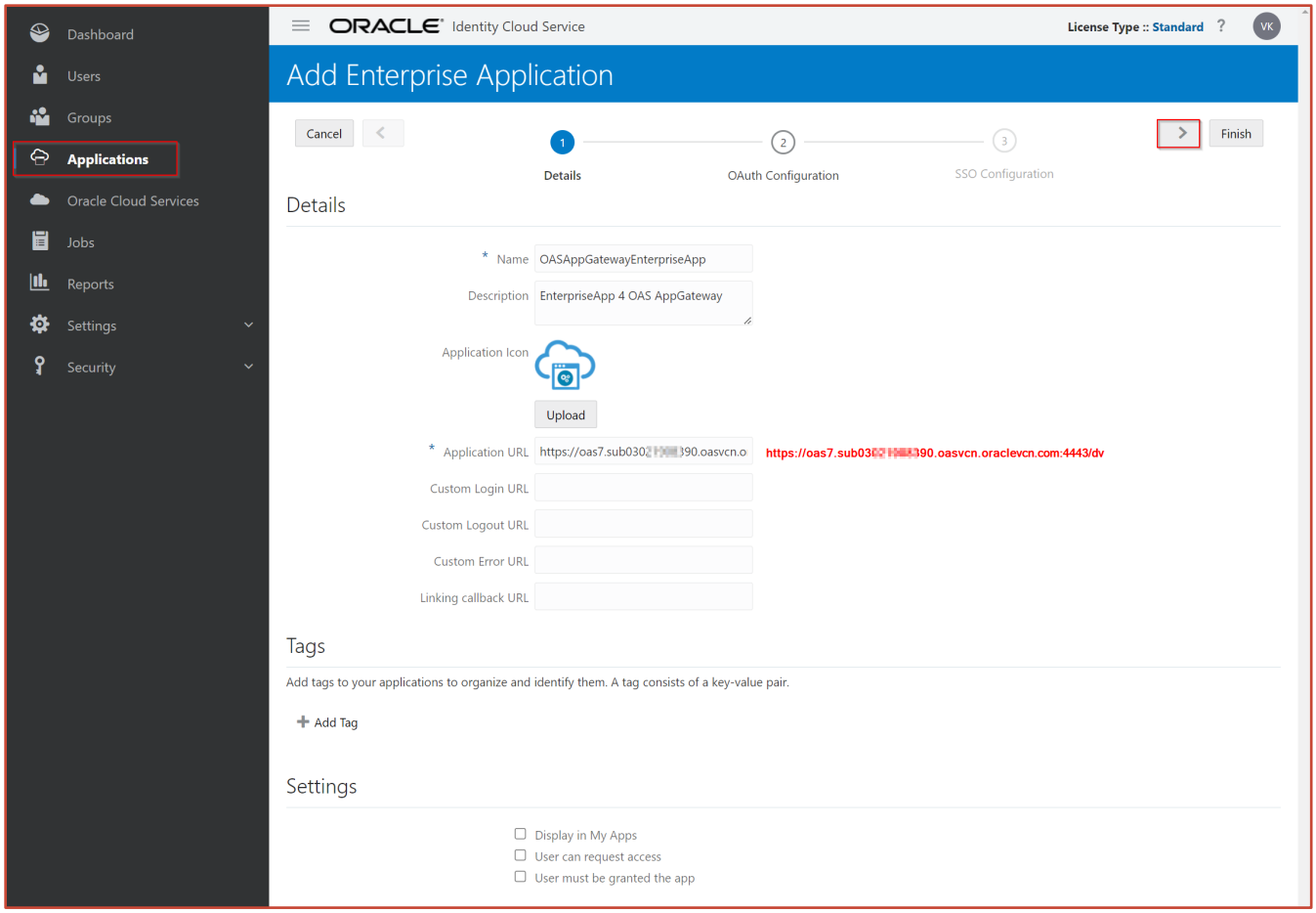
To add an enterprise application in Oracle Identity Cloud Service, you need to configure the list of application resources (web application's URLs or URL patterns), create an authentication policy for each resource, and create an authorization policy for each resource. For each authentication policy, you define an authentication method, and header variables for App Gateway to include in the request before forwarding the request to the application.

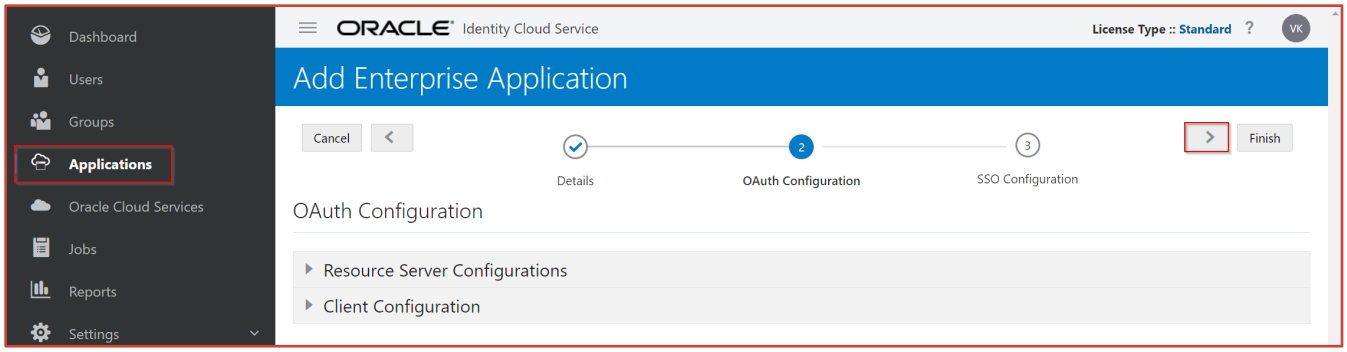
1. Sign into the Identity Cloud Service console as an application administrator.
2. Expand the **Navigation Drawer**, click **Applications**, and then click **Add**.
3. In the **Add Application** page, click **Enterprise Application**.
4. In the **Details** pane of the **Add Enterprise Application** page, provide a name for the application, enter the application URL, complete all other fields as necessary, and then click the **Next >** icon.

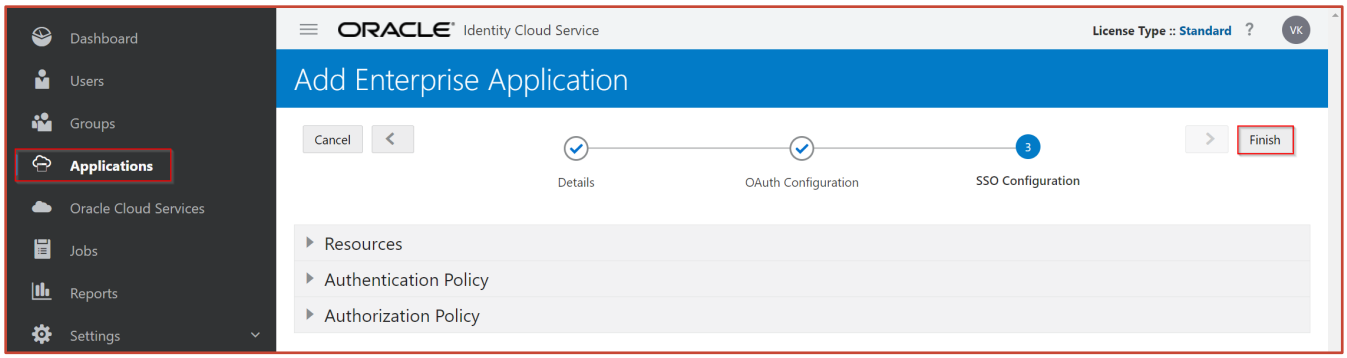
   Note: The application URL is the URL that you want users to use to access your enterprise application. Use the host name and port number of the App Gateway. If you have multiple instances of App Gateway, then use the host name and port number of the load balancer.

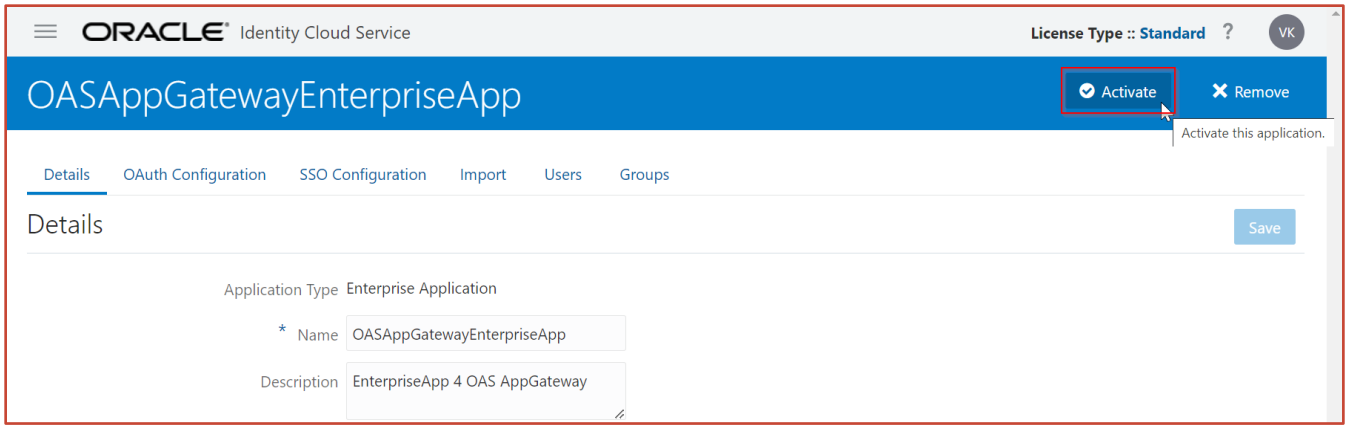
5. In the **OAuth Configuration** pane, click the **Next >** icon.
   Use the **OAuth Configuration** pane to configure the enterprise application to act as a confidential application by providing client and resource server configurations.
6. In the **SSO Configuration** pane, click **Finish**.
   You configure the **Resources**, **Authentication Policy** and **Authorization Policy** sections under the **SSO Configuration** pane later.
7. Click **Activate**, and then click **OK** in the **Confirmation** widow to activate the application.











Activate the Enterprise Application created.

Under the **SSO Configuration** tab, **Add Resources.**

Add Resources

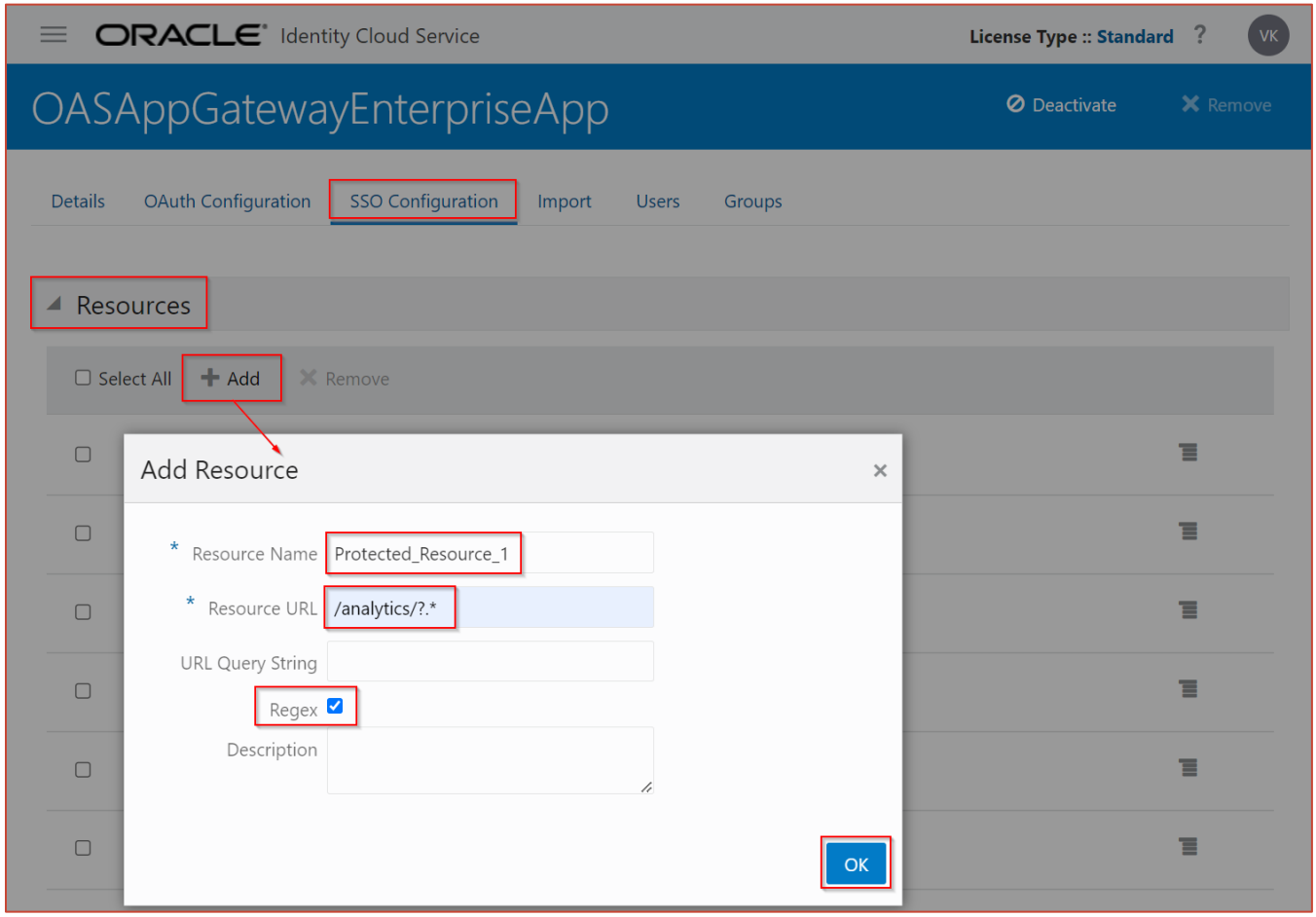
For Protected, Public, and Excluded list of Resources for OAS, Refer to OAS Documentation.

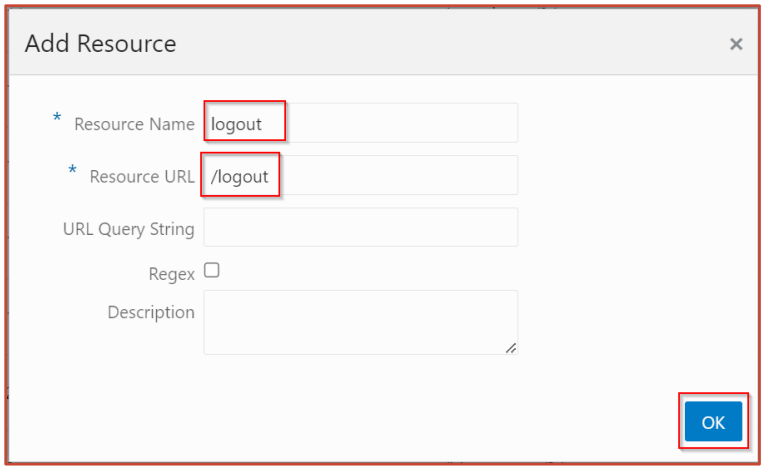
Protect Oracle Analytics Server URLs or Make Them Public

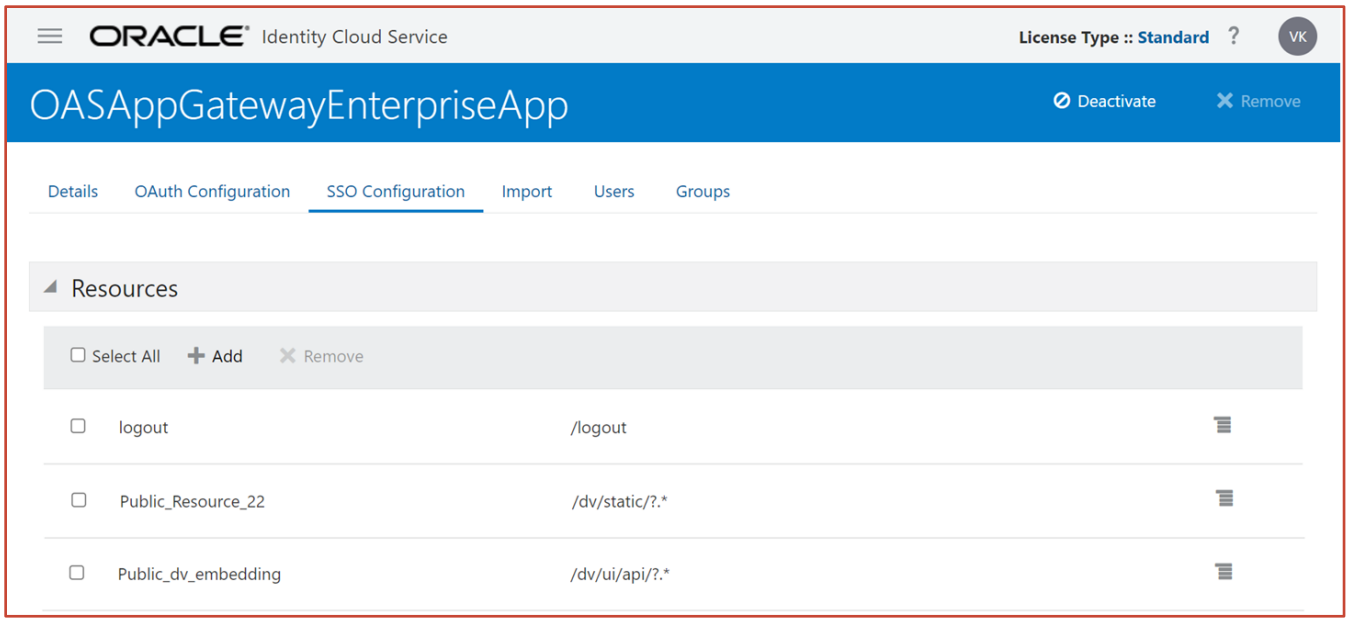
Resource	Public	Protected
/analytics/?.*	-	Yes
/analytics/saw.dll/wsdl?.*	Yes	-
/analytics-bi-adf/?.*	Yes	-
/analytics-ws?.*	Yes	-
/api/?.*	Yes	-
/aps/?.*	Yes	-
/aps/JAPI/?.*	Yes	-
/aps/SmartView/?.*	-	Yes
/bicomposer/?.*	-	Yes
/bicontent/?.*	-	Yes
/bi-lcm/?.*	Yes	-
/biinfer/?.*	-	Yes

Resource	Public	Protected
/bi-sac-config-mgr/?.*	-	Yes
/bisearch/?.*	-	Yes
/bi-security-login/?.*	Yes	-
/biserviceadministration/?.*	-	Yes
/biservices/?.*	Yes	-
/cds/?.*	-	Yes
/dv/?.*	-	Yes
/mapviewer/?.*	-	Yes
/mapviewer/dataserver/?.*	Yes	-
/mapviewer/foi/?.*	Yes	-
/mapviewer/mcserver/?.*	Yes	-
/mapviewer/wms/?.*	Yes	-
/mapviewer/wmts/?.*	Yes	-
/mobile/?.*	-	Yes
/security/?.*	-	Yes
/xmlpserver/?.*	-	Yes
/xmlpserver/Guest?.*	Yes	-
/xmlpserver/report_service/?.*	Yes	-
/xmlpserver/ReportTemplateService.xls?.*	Yes	-
/xmlpserver/services/?.*	Yes	-
/bimajel/?.*	-	Yes

Resource	Public	Protected
/analytics/res/?.*	Yes	-
/dv/public/?.*	Yes	-
/dv/ui/api/?.*	Yes	-
/dv/static/?.*	Yes	-
/logout	-	Yes

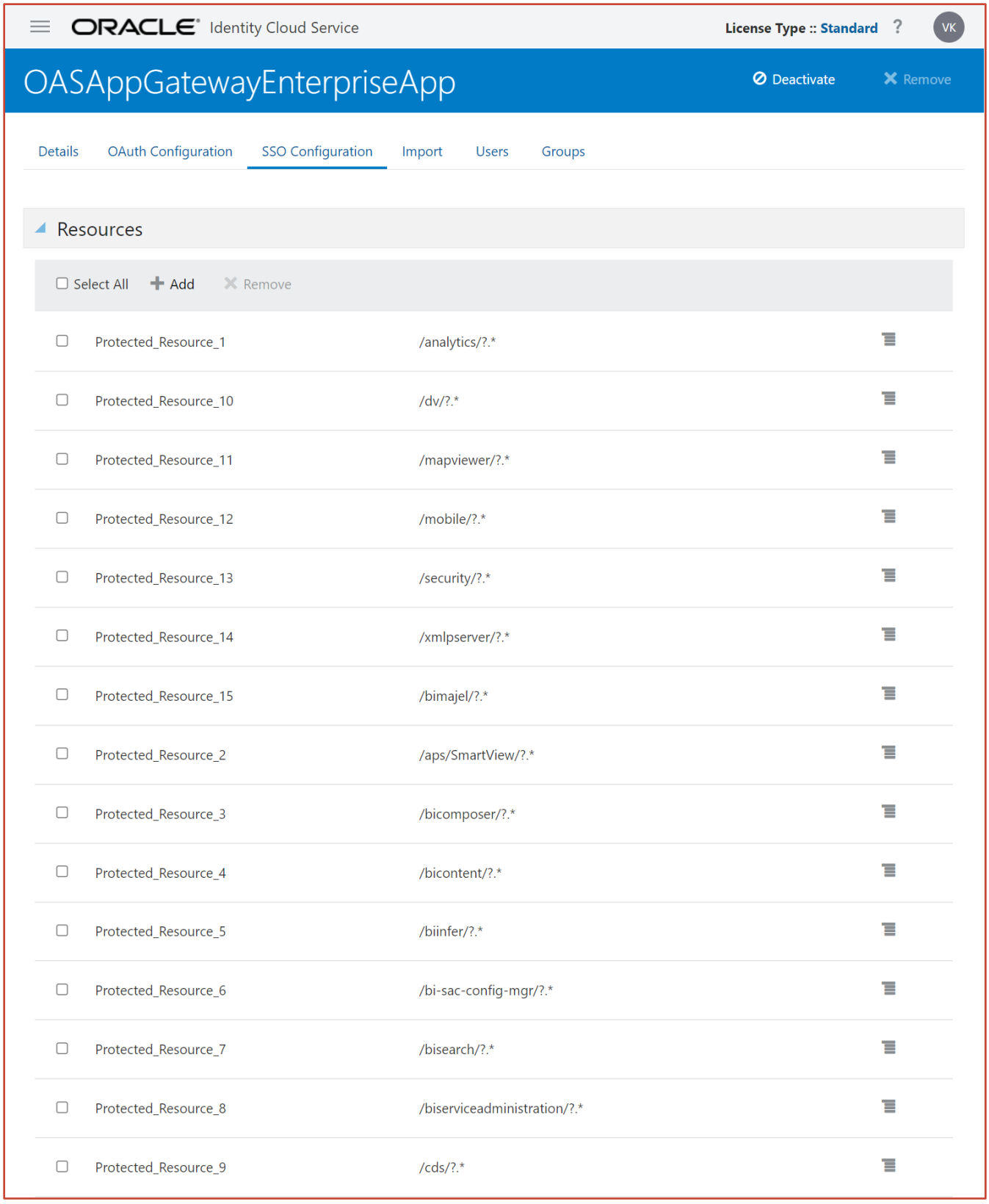






NOTE: Resource **`/dv/static/?.*`** and **`/dv/ui/api/?.*`** are needed for embedding DV Projects in a Public Portal.

NOTE: Resource **`/logout`** is needed to LogOff from Oracle Analytics Server using IDCS App Gateway.

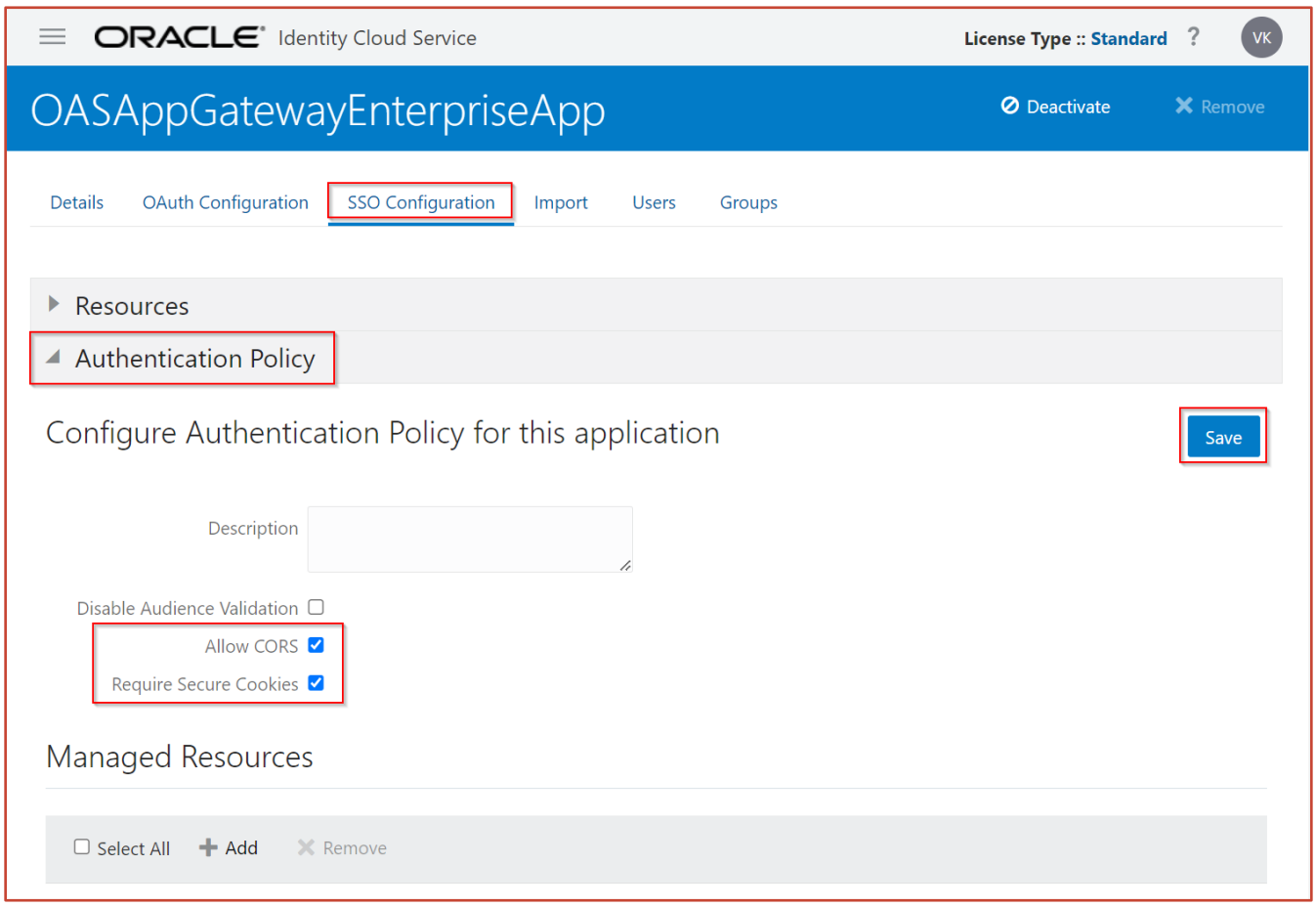


☐	Public_Resource_1	/analytics/saw.dll/wsdl?.*
☐	Public_Resource_10	/mapviewer/dataserver/?.*
☐	Public_Resource_11	/mapviewer/foi/?.*
☐	Public_Resource_12	/mapviewer/mcserver/?.*
☐	Public_Resource_13	/mapviewer/wms/?.*
☐	Public_Resource_14	/mapviewer/wmts/?.*
☐	Public_Resource_15	/xmlpserver/Guest?.*
☐	Public_Resource_16	/xmlpserver/report_service/?.*
☐	Public_Resource_17	/xmlpserver/ReportTemplateService.xls?.*
☐	Public_Resource_18	/xmlpserver/services/?.*
☐	Public_Resource_19	/analytics/res/?.*
☐	Public_Resource_2	/analytics-bi-adf/?.*
☐	Public_Resource_20	/dv/public/?.*
☐	Public_Resource_3	/analytics-ws?.*
☐	Public_Resource_4	/api/?.*
☐	Public_Resource_5	/aps/?.*
☐	Public_Resource_6	/aps/JAPI/?.*
☐	Public_Resource_7	/bi-lcm/?.*
☐	Public_Resource_8	/bi-security-login/?.*
☐	Public_Resource_9	/biservices/?.*

Page 1 of 1 (1-35 of 35 items) 1

Configure Authentication Policy

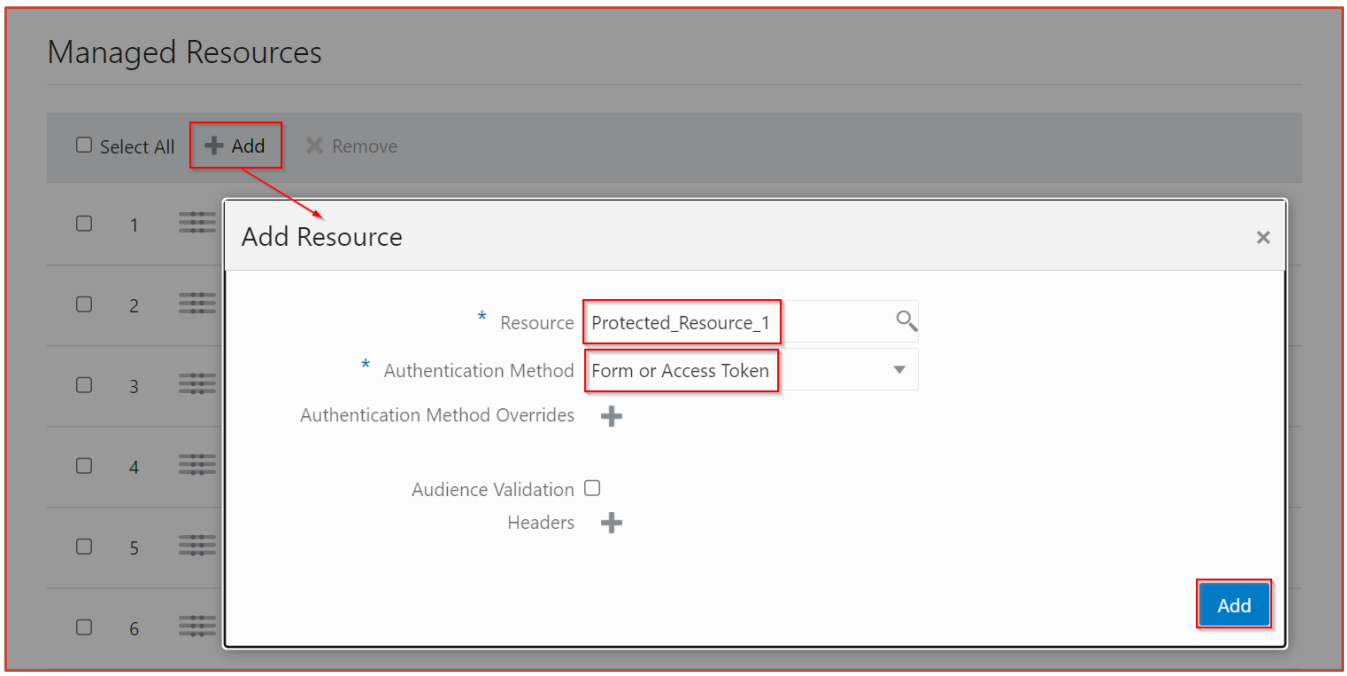
Create a Web Tier Policy to define Protected and Public Resources



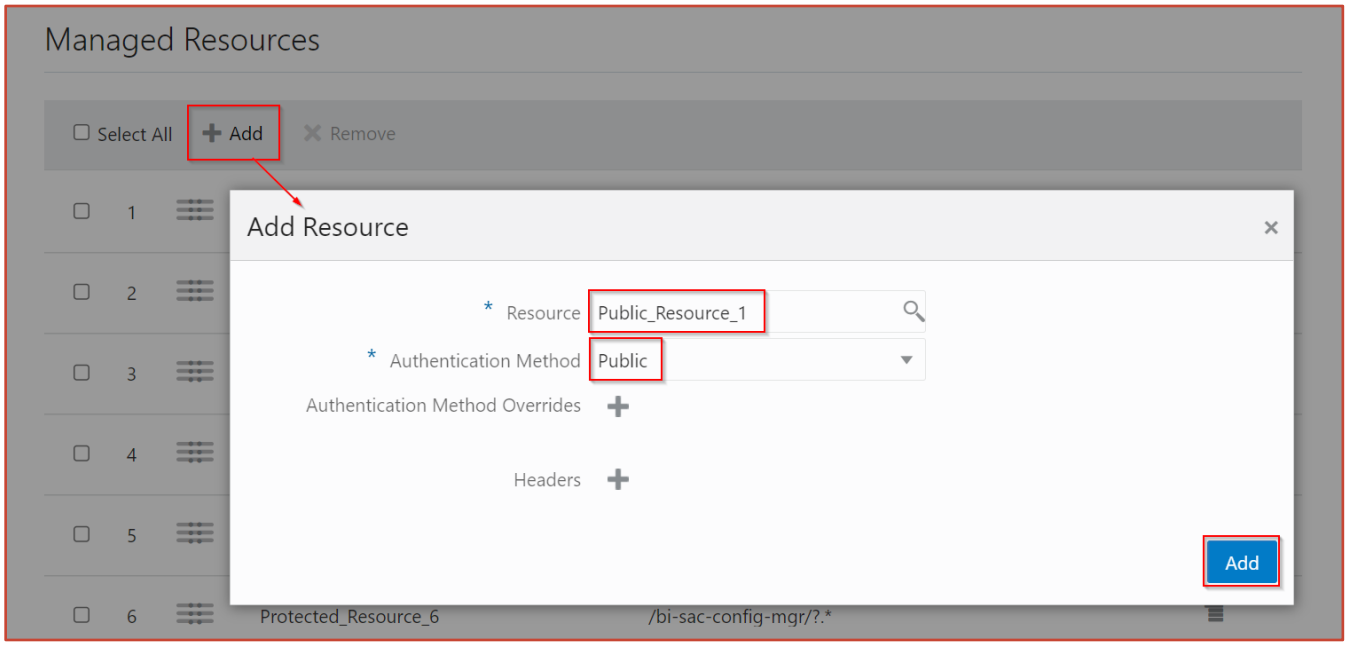
Allow CORS

Configure Allow Cross-Origin Resource Sharing (**CORS**) to allow client applications that run on one domain to obtain data from another domain.

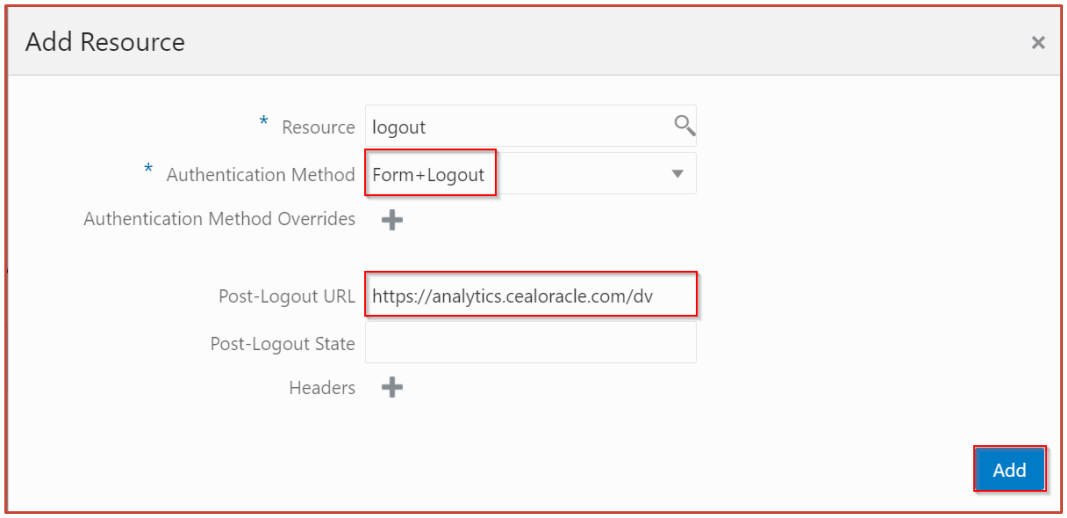
Protected Resource, Authentication Method: **Form or Access Token**.



Public Resource, Authentication Method: **Public**.



For Logout URL use the Authentication Method as Form + Logout



NOTE: The order/sequence of the Resources of type Regex (Regular Expression) Added to the Authentication Policy play a key role in Protecting and Unprotecting the child resources.

For example, below is the order of the Resources added to Authentication Policy

/dv/?.* -- Protected

/dv/ui/api/?.* --Public

/dv/static/?.* --Public

Since the /dv/* is a protected resource and is defined prior to the other two public resources in the Authentication Policy, the /dv/ui/api/* and /dv/static/* will be satisfying the regular expression condition /dv/* and the public resources will be treated as a protected resource and the Authentication challenge will be triggered for the public resources.

To AVOID such case, please change the order/sequence of the resources added in the Authentication Policy.

For example, below will be the correct order

/dv/static/?.* --Public

/dv/ui/api/?.* --Public

/dv/?.* --Protected.

NOTE: The same rule is applicable for protected resources to be treated as public resources

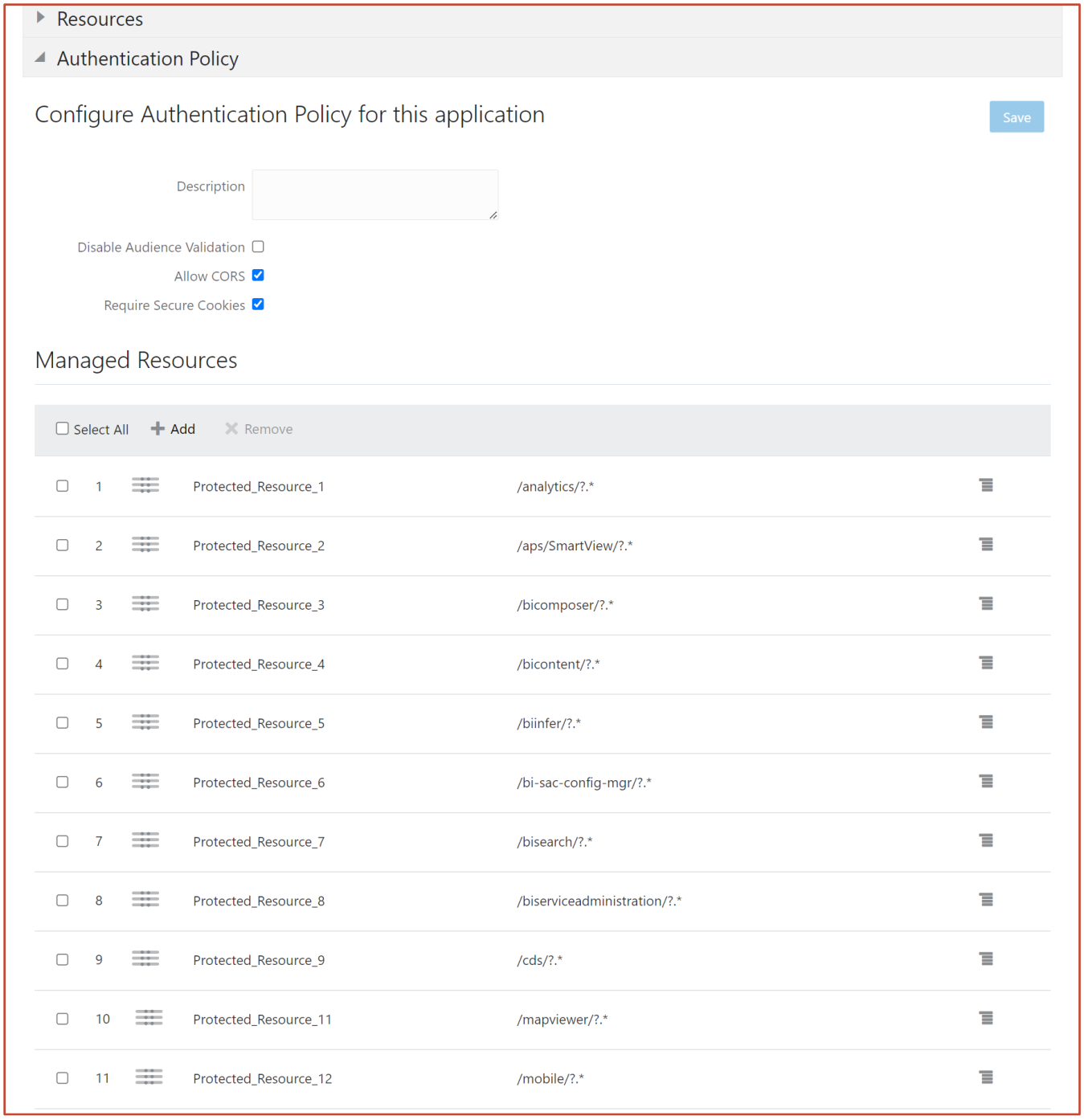
For example, as below

/abc/?.* --Public

/abc/xyz/?.* --Protected

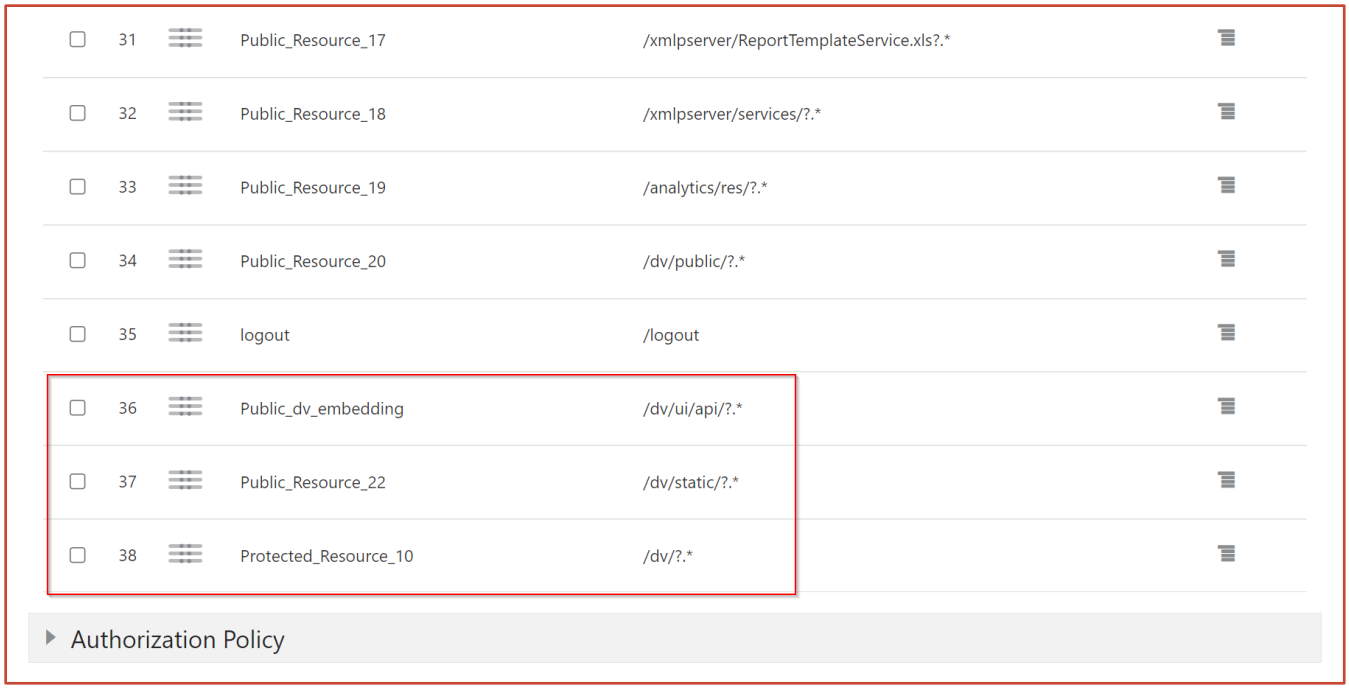
Since the /abc/xyz/* satisfies the regular expression condition of public resource i.e., /abc/* the protected resource is treated as public.

The solution is to understand such dependencies and change the order of the resources added to the Authentication Policy.

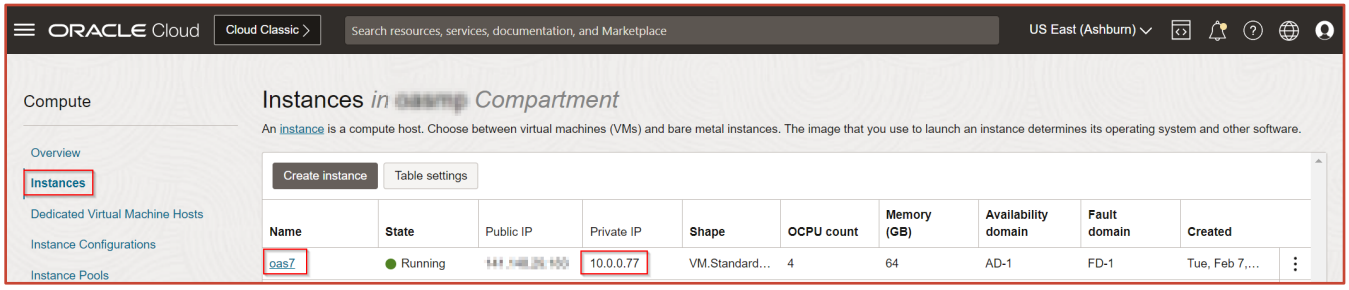


☐	12	Protected_Resource_13	/security/?.*	
☐	13	Protected_Resource_14	/xmlpserver/?.*	
☐	14	Protected_Resource_15	/bimajel/?.*	
☐	15	Public_Resource_1	/analytics/saw.dll/wsdl?.*	
☐	16	Public_Resource_2	/analytics-bi-adf/?.*	
☐	17	Public_Resource_3	/analytics-ws?.*	
☐	18	Public_Resource_4	/api/?.*	
☐	19	Public_Resource_5	/aps/?.*	
☐	20	Public_Resource_6	/aps/JAPI/?.*	
☐	21	Public_Resource_7	/bi-lcm/?.*	
☐	22	Public_Resource_8	/bi-security-login/?.*	
☐	23	Public_Resource_9	/biservices/?.*	
☐	24	Public_Resource_10	/mapviewer/dataserver/?.*	
☐	25	Public_Resource_11	/mapviewer/foi/?.*	
☐	26	Public_Resource_12	/mapviewer/mcserver/?.*	
☐	27	Public_Resource_13	/mapviewer/wms/?.*	
☐	28	Public_Resource_14	/mapviewer/wmts/?.*	
☐	29	Public_Resource_15	/xmlpserver/Guest?.*	
☐	30	Public_Resource_16	/xmlpserver/report_service/?.*	

Spinning Tool

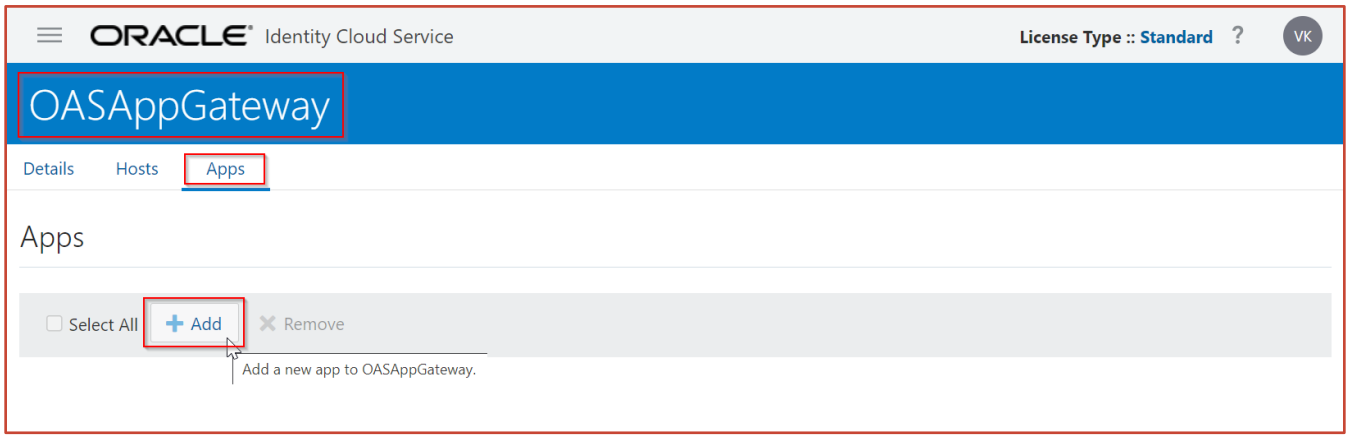


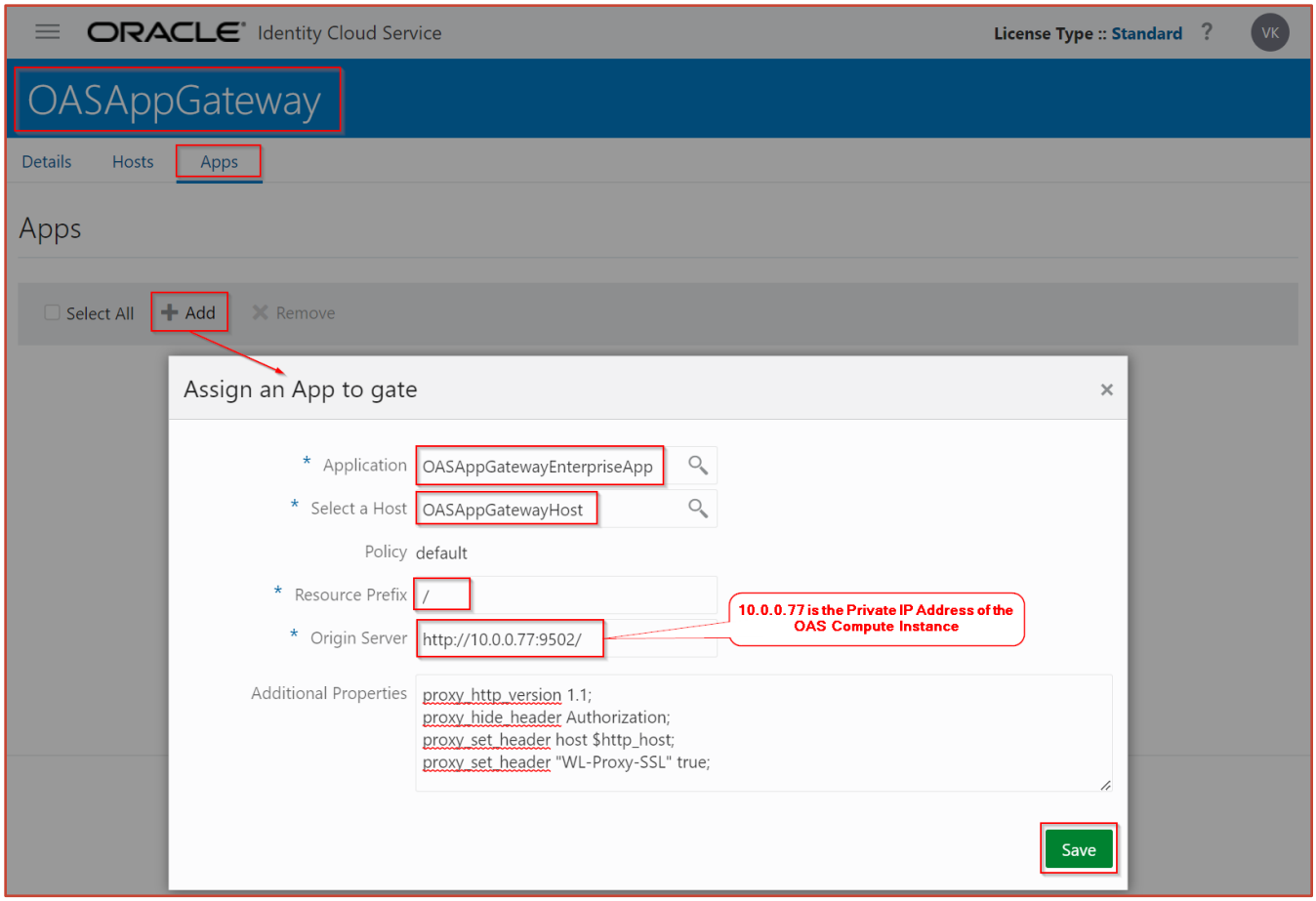
Get the **Private IP Address** of The Oracle Analytics Server Compute Instance on Oracle Cloud



Add the Enterprise Application to the Registered App Gateway

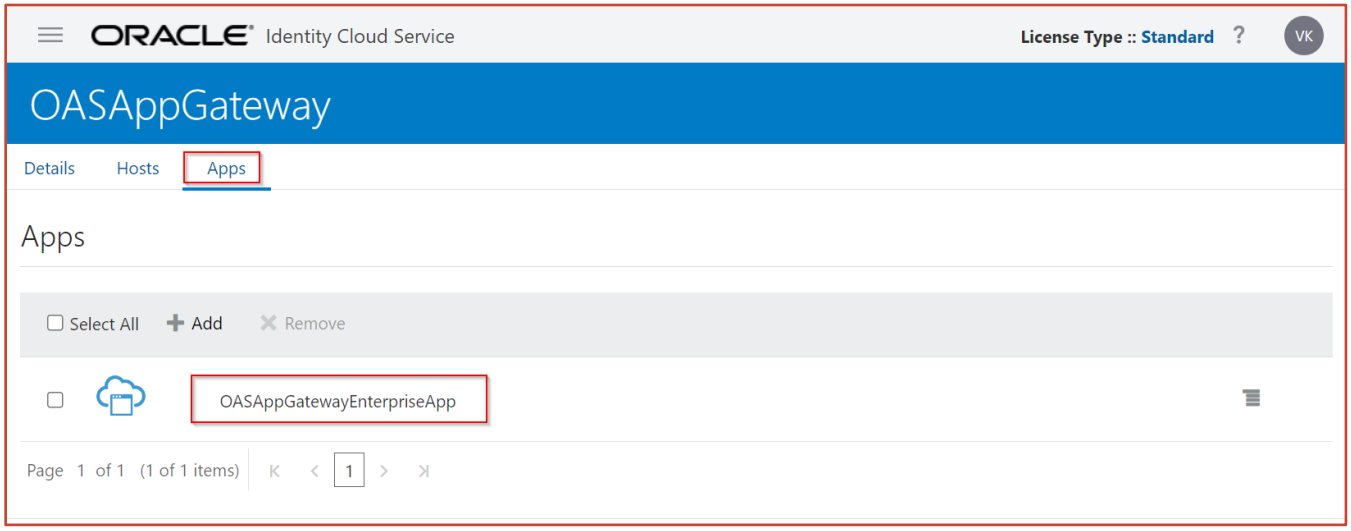
Navigate to Security → App Gateways → <Registered App Gateway> → Apps (Tab)

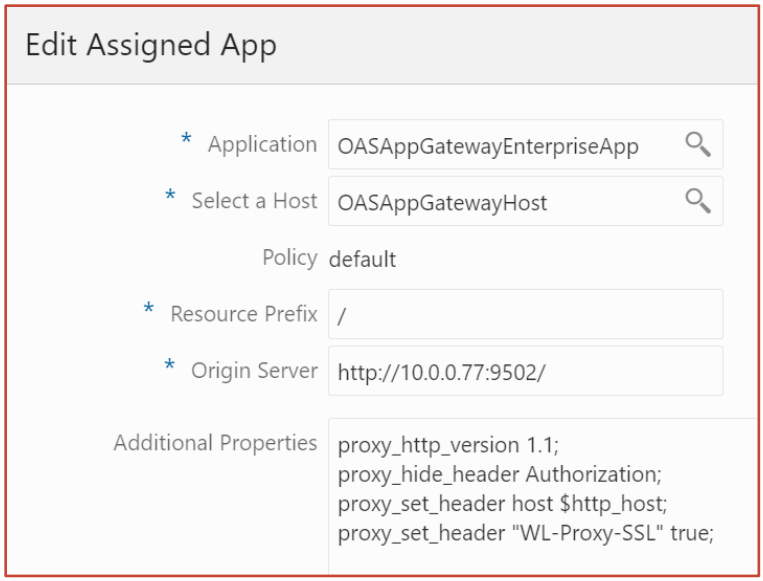




NOTE: Need to add the OAS server Private IP as the Origin Server for example, `http://<OAS_Private_IP>:9502/`

App gateway running on same compute as OAS or different compute in OCI, will always use private IP to communicate internally with in the OCI.



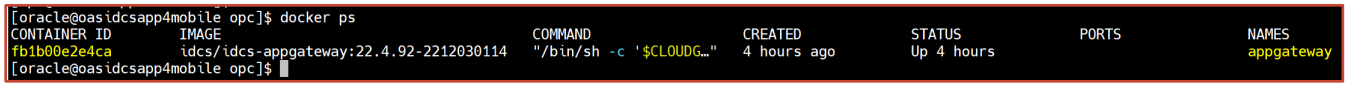



```
proxy_http_version 1.1;
proxy_hide_header Authorization;
proxy_set_header host $http_host;
proxy_set_header "WL-Proxy-SSL" true;
```

Start and Stop App Gateway

To start and stop App Gateway Server and App Gateway Agent, you can use scripts installed in the App Gateway docker container where your App Gateway runs.

On the App Gateway Docker Container Host, connect to the App Gateway docker container using below command
Get the Docker Container ID or the Name
Command: `docker ps`




```
docker exec -it appgateway bash
```
or
```
docker exec -it fb1b00e2e4ca bash
```

And then run the following command:

1. To start App Gateway server.
   `/scratch/oracle/idcs-cloudgate/latest/bin/cg-start`
2. To start App Gateway agent.
   `/scratch/oracle/idcs-cloudgate/latest/bin/agent-start`
3. To stop App Gateway server.
   `/scratch/oracle/idcs-cloudgate/latest/bin/cg-stop`
4. To stop App Gateway agent.
   `/scratch/oracle/idcs-cloudgate/latest/bin/agent-stop`

When you start the App Gateway Server, App Gateway contacts Oracle Identity Cloud Service to retrieve the **port number** you configured during the App Gateway registration in Oracle Identity Cloud Service console.

The App Gateway Server starts using this **port number**. The App Gateway Agent is responsible for synchronizing the App Gateway configuration

(hosts and applications) from Oracle Identity Cloud Service to the App Gateway server.

To check the running status of the App Gateway server, run the following command:
`/scratch/oracle/idcs-cloudgate/latest/bin/cg-status`

From Browser to test the App gateway Server Status use below URL

`https://myappgateway.example.com:4443/cloudgate/v1/about`

We can create a custom script to control the App Gateway docker and its cloud gate and agent services inside the docker container

In case we make any changes to the Registered App Gateway or Enterprise Application we need to either restart the cloud gate and agent services inside the docker container or restart the docker.

```
----------------------------------appgateway-ctl.sh---------------------------------
#!/bin/bash
# chkconfig: 345 99 01
# description: Shut down docker appgateway gracefully on system shutdown

case "$1" in
        start|restart)
                [ -z $( docker ps -f name=^appgateway$ -q ) ] || docker stop appgateway
                [ -z $( docker ps -af name=^appgateway$ -q ) ] || docker rm appgateway
                docker run -dit --name appgateway --env-file
/u01/data/AppGateway/appgateway-env --env HOST_MACHINE=`hostname -f` --volume
/u01/data/AppGateway/cwallet.sso:/usr/local/nginx/conf/cwallet.sso --volume
/u01/data/AppGateway/appgatewayhost.crt:/scratch/docker/appgate/certs/appgatewayhost.crt
--volume
/u01/data/AppGateway/appgatewayhost.key:/scratch/docker/appgate/certs/appgatewayhost.key
--net=host idcs/idcs-appgateway:23.2.92-2301160723
        ;;
        stop)
                [ -z $( docker ps -f name=^appgateway$ -q ) ] || docker stop appgateway
                [ -z $( docker ps -af name=^appgateway$ -q ) ] || docker rm appgateway
        ;;
        restartservices)
                if [ $( docker ps -q -f name=^appgateway$ ) ]; then
                        echo "appgateway docker is running"
                        echo "restarting the cloudgate and agent services inside the
docker container"
                        docker exec -it appgateway /scratch/oracle/idcs-
cloudgate/latest/bin/cg-stop
                        docker exec -it appgateway /scratch/oracle/idcs-
cloudgate/latest/bin/agent-stop
                        docker exec -it appgateway /scratch/oracle/idcs-
cloudgate/latest/bin/cg-start
                        docker exec -it appgateway /scratch/oracle/idcs-
cloudgate/latest/bin/agent-start
                else
                        echo "appgateway docker is not running."
                fi
        ;;
```

```
        status)
                [ -z $( docker ps -f name=^appgateway$ -q ) ] || echo "appgateway docker
is running."
                [ ! -z $( docker ps -f name=^appgateway$ -q ) ] || echo "appgateway
docker is not running."
        ;;
        *)
                echo "Usage: $0 {start|stop|restart|status|restartservices}"
                echo "start|stop|restart|status options are for docker appgateway"
                echo "restartservices option is for restarting the cloudgate and agent
services inside the running docker container"
                exit 1
        ;;
esac
```

Download the Script to Manage the App Gateway Docker Container

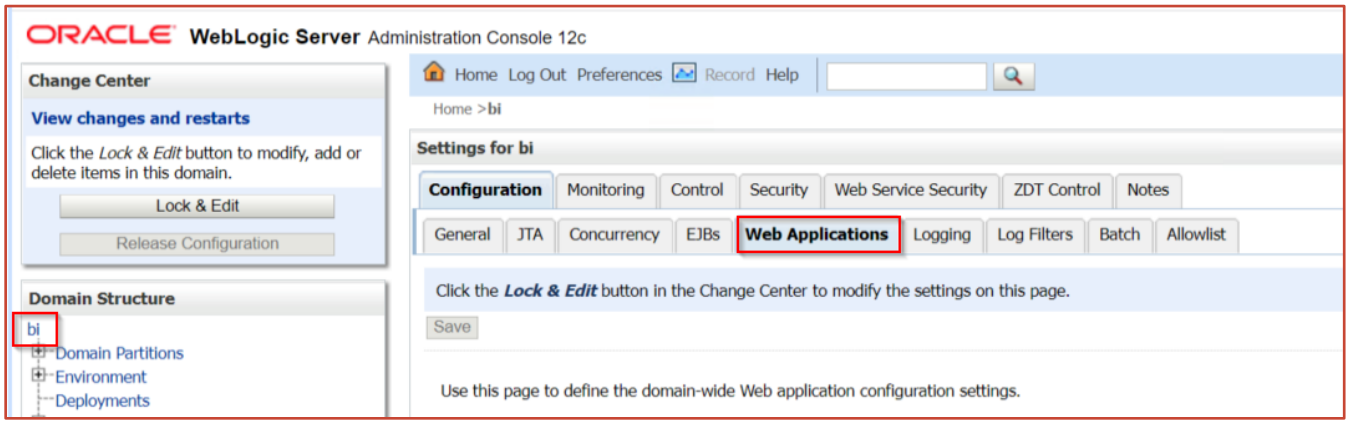
appgateway-ctl.sh

```
chmod +x appgateway-ctl.sh
```

Configuration on OAS Server

Enable the WebLogic Plugin

In the WebLogic admin console enable WebLogic-Plugin at the Domain level. For example "bi".

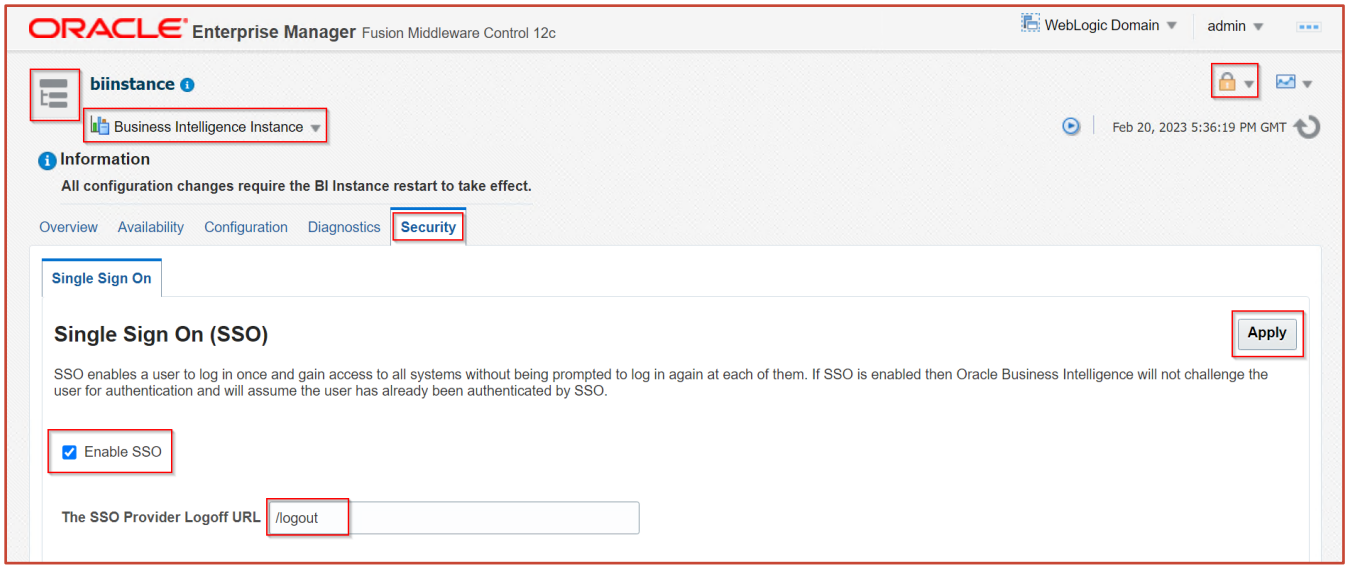


Config the SSO Logout URL

Login to WebLogic FMW EM and change the SSO Provider Logoff URL

from "`/bi-security-login/logout?redirect=/dv`" to "`/logout`"

Lock and Edit → Set the logout URL to "`/logout`" → Apply → Activate Changes.



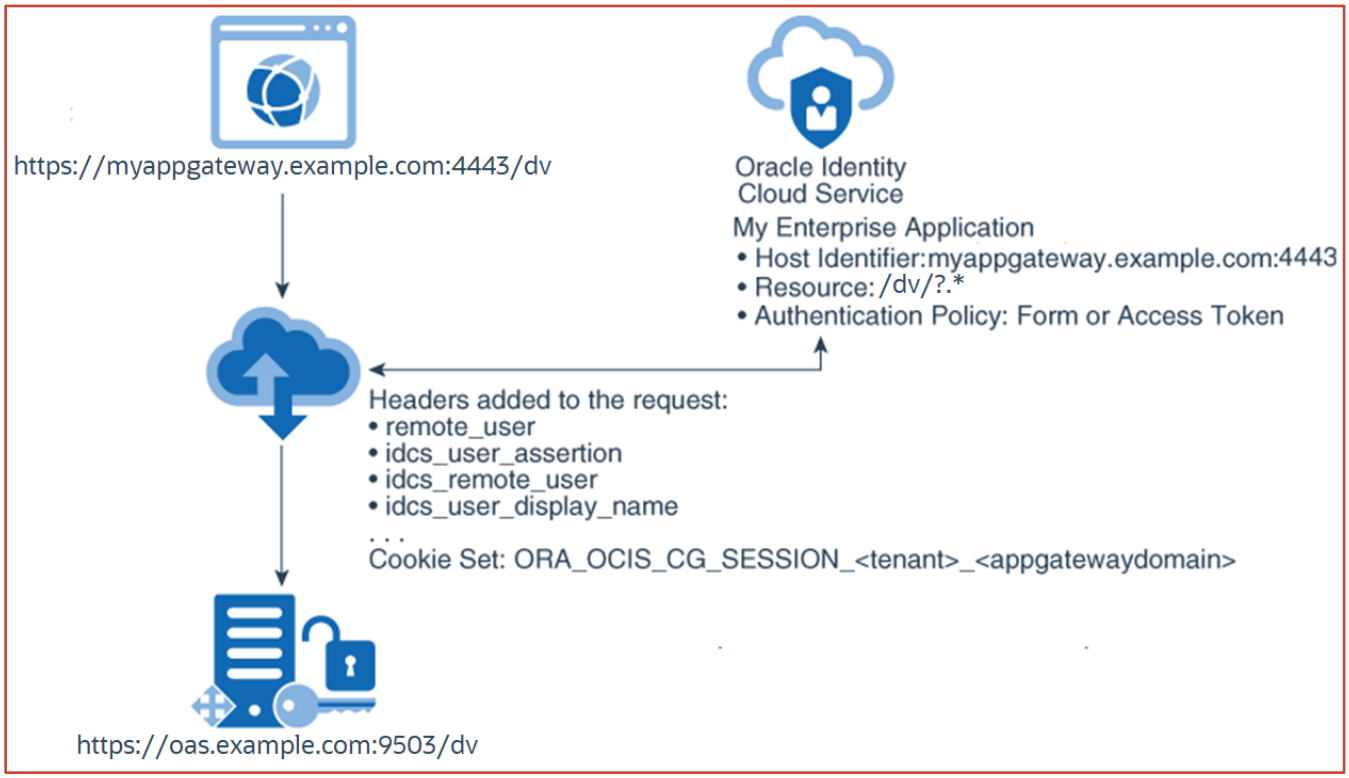
Restart the OAS Services.

Test Access to Your Application Using App Gateway

After you configure the App Gateway server to communicate with your Oracle Identity Cloud Service instance, and start the server, test access to your enterprise application.

The following diagram provides an example of how App Gateway and Oracle Identity Cloud Service interact when an HTTP request to an application resource is sent by the user browser through App Gateway.

Because App Gateway proxies your web application, use the App Gateway base URL to access the application instead of the application's actual URL.



1. Open a new web browser and access your application using the App Gateway URL.
   In this example, the URL is: `https://myappgateway.example.com:4443/dv`
   The actual application `https://oas.example.com:9503/dv` isn't accessible by the user browser.
2. App Gateway intercepts the request and communicates with Oracle Identity Cloud Service to verify if the URL corresponds to an enterprise application.
   In this example, Enterprise Application is registered, and the authentication policy for this enterprise application is **Form or Access Token**.
3. App Gateway verifies the request contains a valid Oracle Identity Cloud Service's access token in the Authorization Bearer header or Oracle Identity Cloud Service's session cookie, indicating the user has already signed into Oracle Identity Cloud Service.
4. If the user hasn't signed into Oracle Identity Cloud Service, then App Gateway redirects the user browser to Oracle Identity Cloud Service **Sign In** page.
5. If the user has signed in, then App Gateway adds header variables and a cookie to the request, and then forwards the request to the application.

The application receives the request, uses the header variables to identify the user and to present the content of the /dv page.

Configuring WebLogic to prevent direct access to BI

Follow the WebLogic documentation to configure a connection filter so that only the App Gateway server and machines running BI components are allowed to access the WebLogic server:

1. Instructions on configuring the default connection filter are contained in the section entitled **Using Connection Filters** in the WebLogic documentation at: Using Network Connection Filters.
2. Instructions on writing an appropriate filter rule are contained in the section entitled **Guidelines for Writing Connection Filter Rules** in WebLogic documentation at: Guidelines for Writing Connection Filter Rules.
3. Your filter rule should look like this:

```
[App Gateway server IP Address] * [WebLogic Admin Server Port] allow
[App Gateway server IP Address] * [WebLogic Managed Server Port] allow
[BI component server IP Address] * [WebLogic Admin Server Port] allow
[BI component server IP Address] * [WebLogic Managed Server Port] allow
[Another BI component server IP Address (if it exists)] * [WebLogic Managed Server
Port] allow
0.0.0.0/0 * * deny
```

Test that you can access the WebLogic Administration Console and Analytics URLs via the web server, but not directly from any other machine.

Protecting direct HTTP access to OBIPS

Follow the guidance in the **Managing Security for Oracle Analytics Server** documentation guide, SSO Implementation Considerations.

For convenience, an extract from the OAS document is shown below.

When implementing an SSO solution with Oracle Analytics Server you should consider the following:

When accepting trusted information from the App Gateway server or servlet container, you must secure the machines that communicate directly with Presentation Services. In the `instanceconfig.xml` file, specify the list of App Gateway server or servlet container IP addresses in the `Listener\Firewall` node. The `Firewall` node must include the IP addresses of all the Oracle BI Scheduler instances, Oracle Presentation Services instances, and Oracle Analytics Server Java Host instances.

If any of these components are co-located with Oracle BI Presentation Services, you must add the 127.0.0.1 address in `Firewall` node. Setting the list of App Gateway server or servlet container IP addresses does not control end-user browser IP addresses. When using mutually-authenticated SSL, you must specify the Distinguished Names (DNs) of all trusted hosts in the `Listener\TrustedPeers` node.

For example:

```
<Listener port="XXXX" ssl="false">
   <Firewall>
      <Allow address="127.0.0.1"/>
      <Allow address="XXX.XXX.X.XXX"/>
      <Allow address="XXX.XXX.X.XXY"/>
   </Firewall>
</Listener>
```

Configure Load Balancer (Optional)

Create a Load Balancer in Oracle Cloud from the OCI Administration Console and configure SSL Certificate of the corresponding DNS Name of the Load Balancer.

While Configuring the OCI Load Balancer, the Backend Server should be the IP Address of the IDCS App Gateway Docker Host i.e., OAS Server (in case you use the OAS Server for hosting App Gateway Docker Container) at 4443 port.

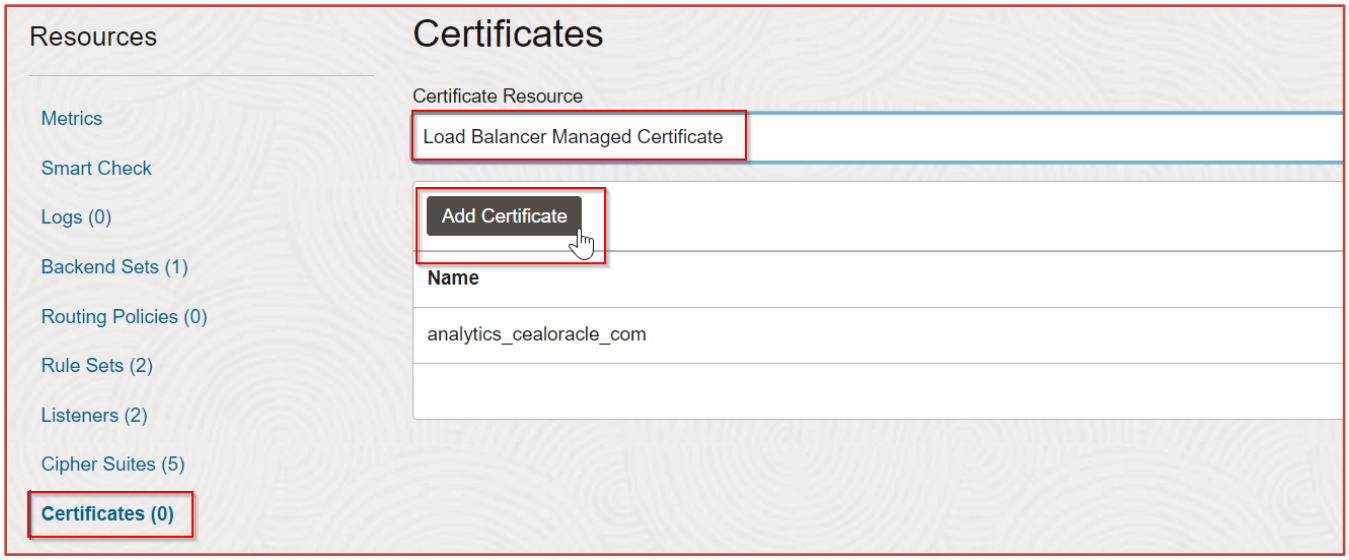
If the SSL is offloaded at Load Balancer and the IDCS App Gateway is running in non-ssl port, Load Balancer should set the RequestHeader IS_SSL as "ssl" and RequestHeader WL-Proxy-SSL as "true".

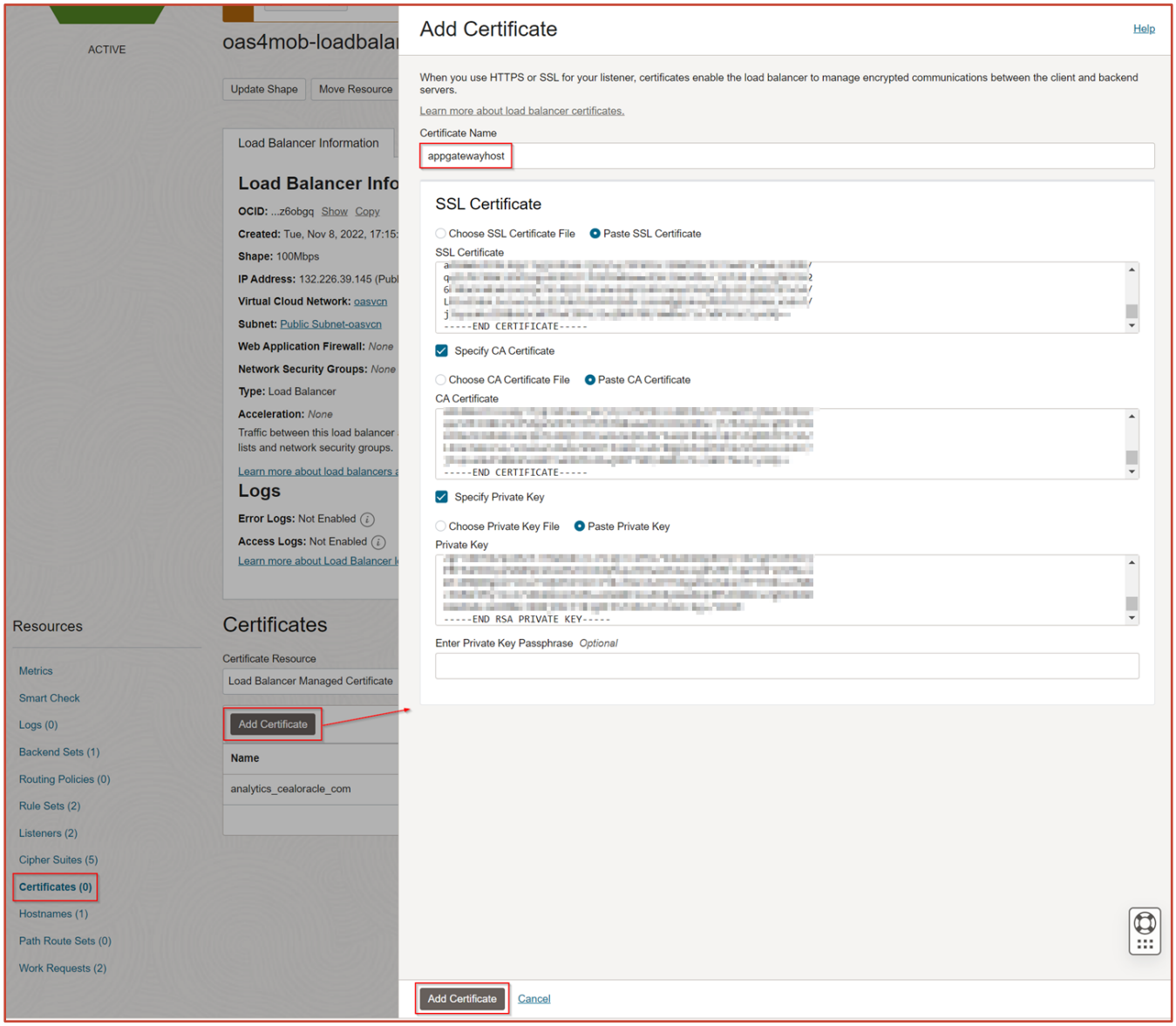
To create the OCI Load Balancer and Offload SSL at Load balancer, Please refer the Blog SSL Offloading at Oracle Cloud Infrastructure (OCI) Load Balancer for Oracle Analytics Server on Oracle Cloud Marketplace.

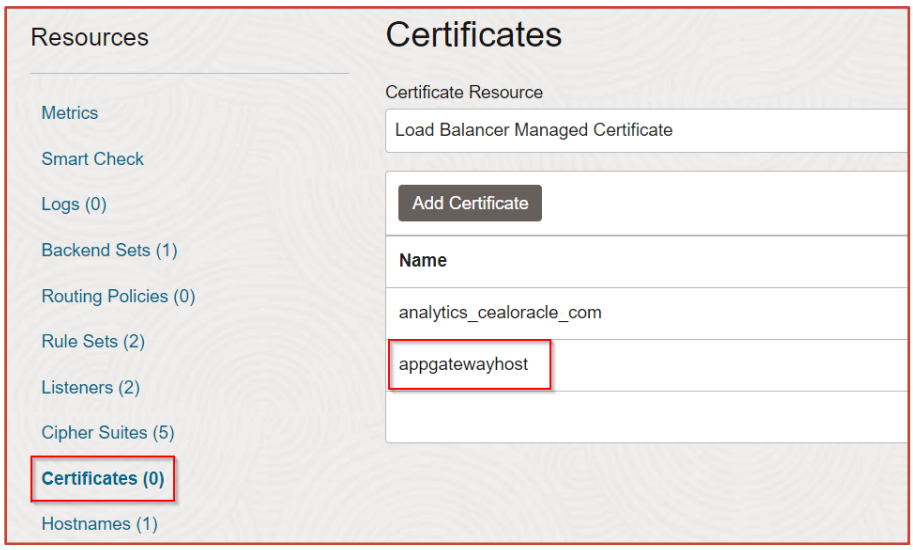
Suppose the OCI Load Balancer's Backend Server i.e.. In that case, the IDCS App Gateway docker host (e.g., OAS Server) is configured to run on HTTPS protocol, follow the steps below at the Load Balancer Configuration.

Get the SSL Certificates (e.g. appgatewayhost.crt and appgatewayhost.key) Configured at IDCS App Gateway Docker Container and create a Certificate in the OCI Load Balancer Configuration pages.

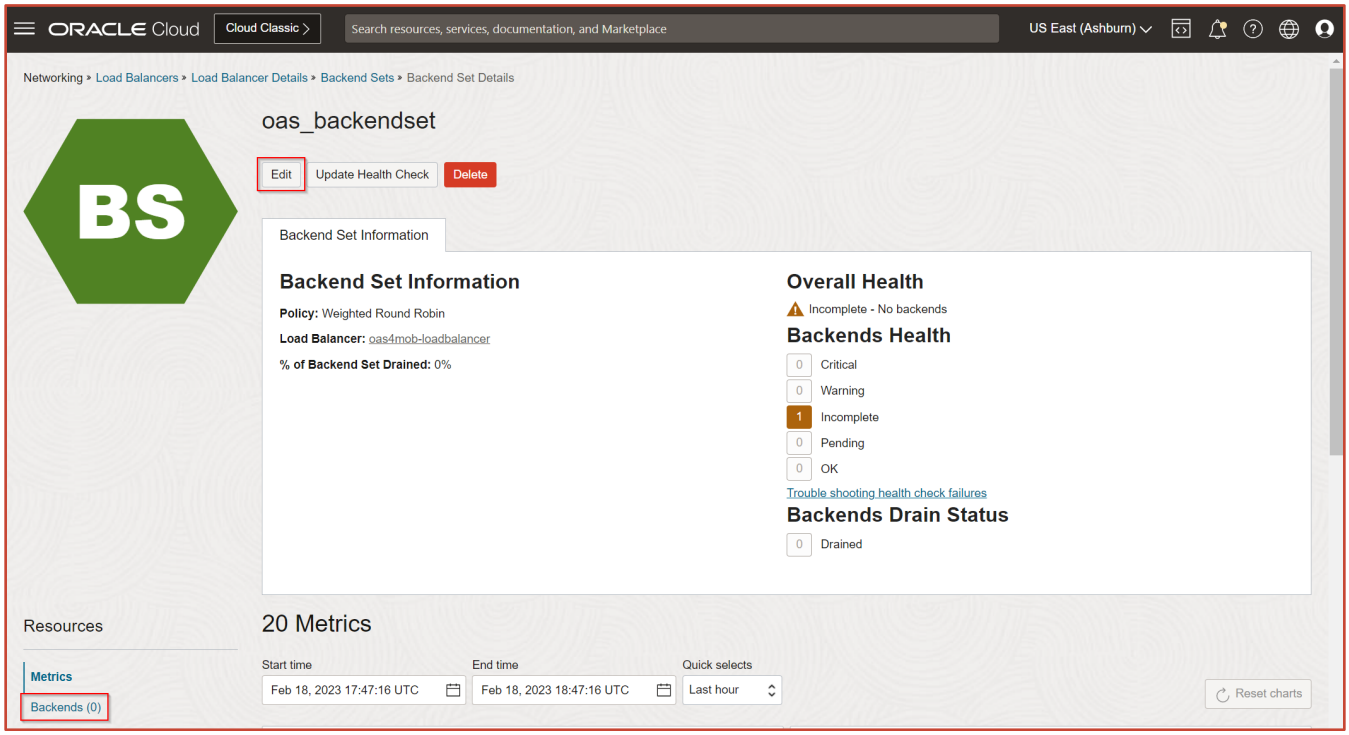
Use this certificate for the Backend Set configuration while adding the App Gateway Docker Host as the backend to the Load Balancer's Backend Set.

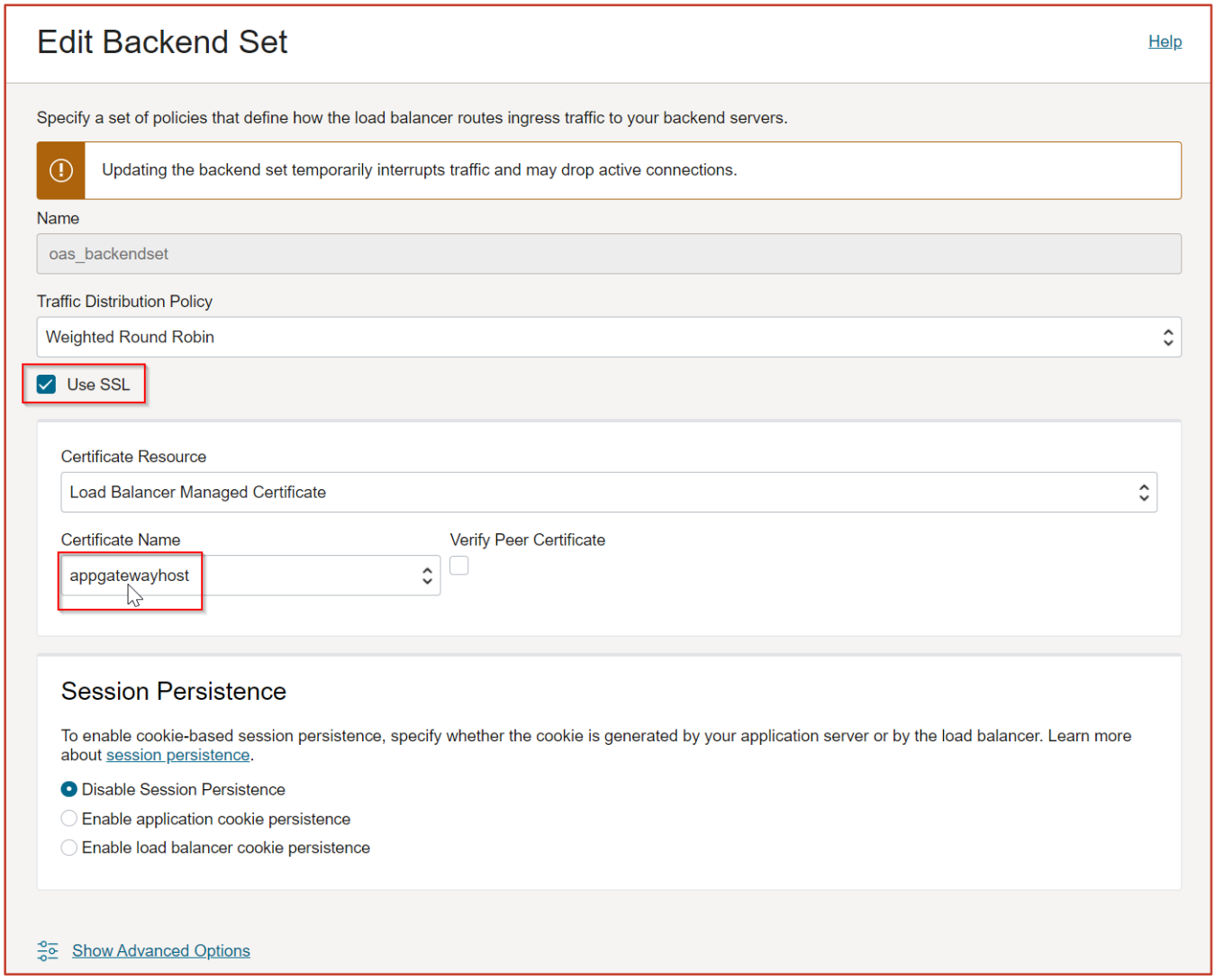






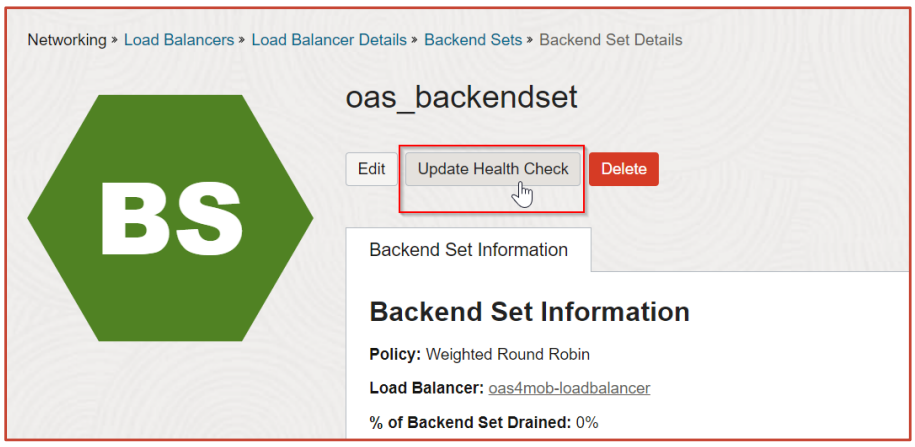
Edit the Backend Set

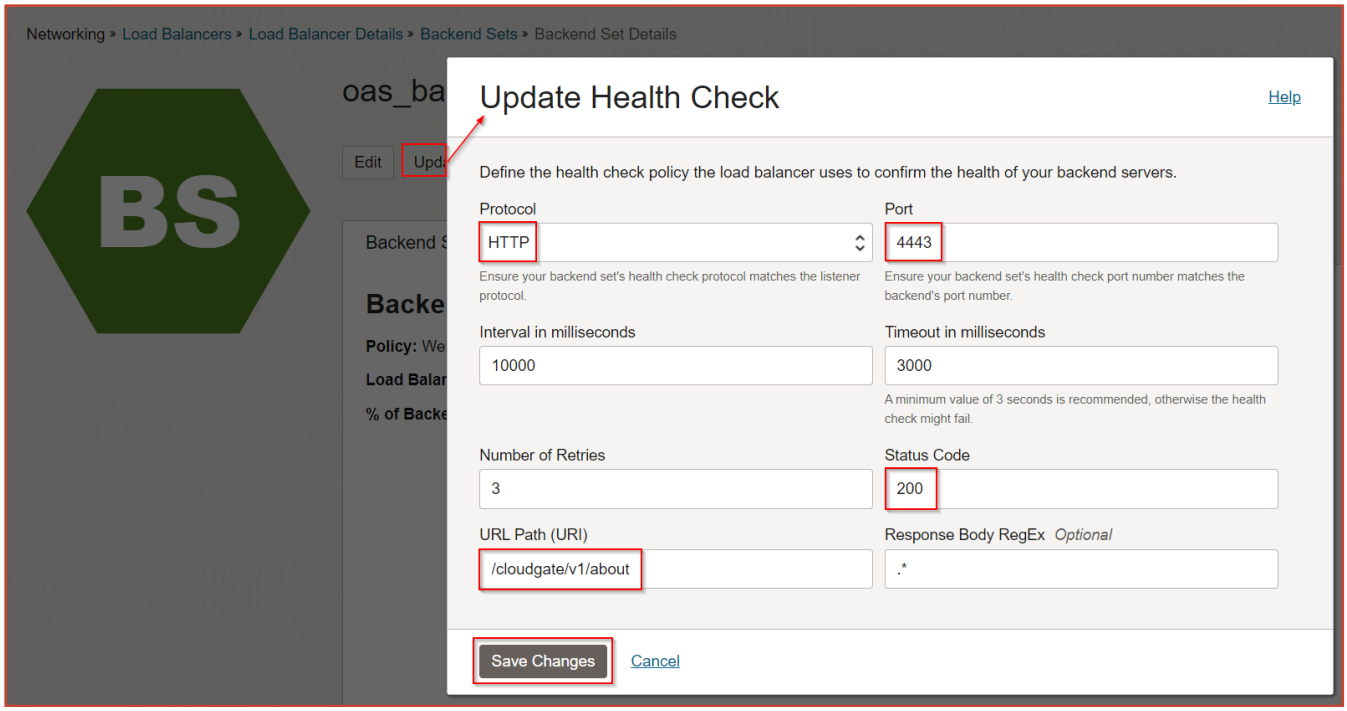




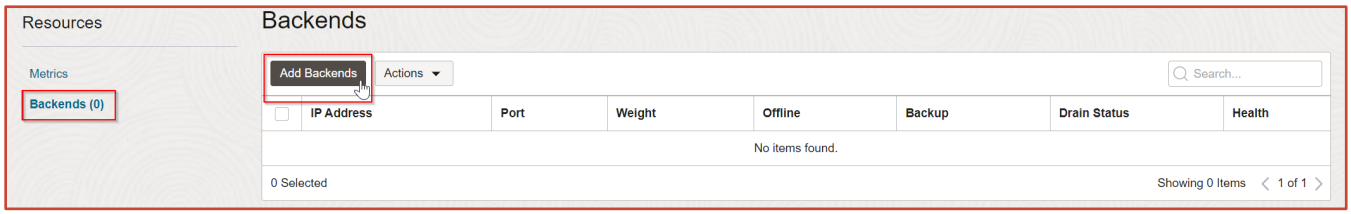
Click on Save Changes.

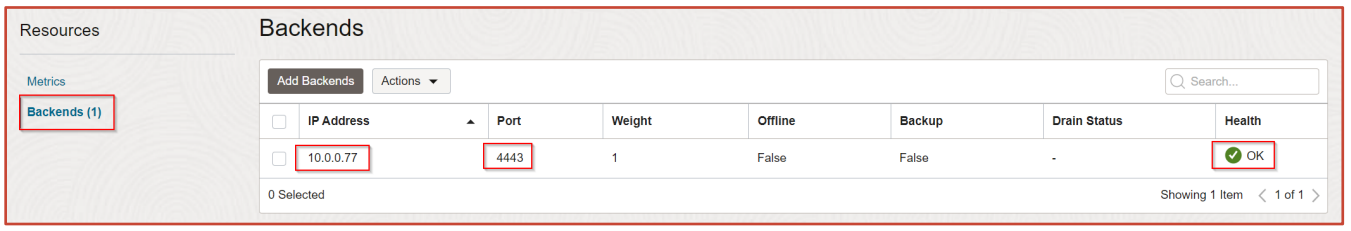
Update the Health Check of the Backend Set as well

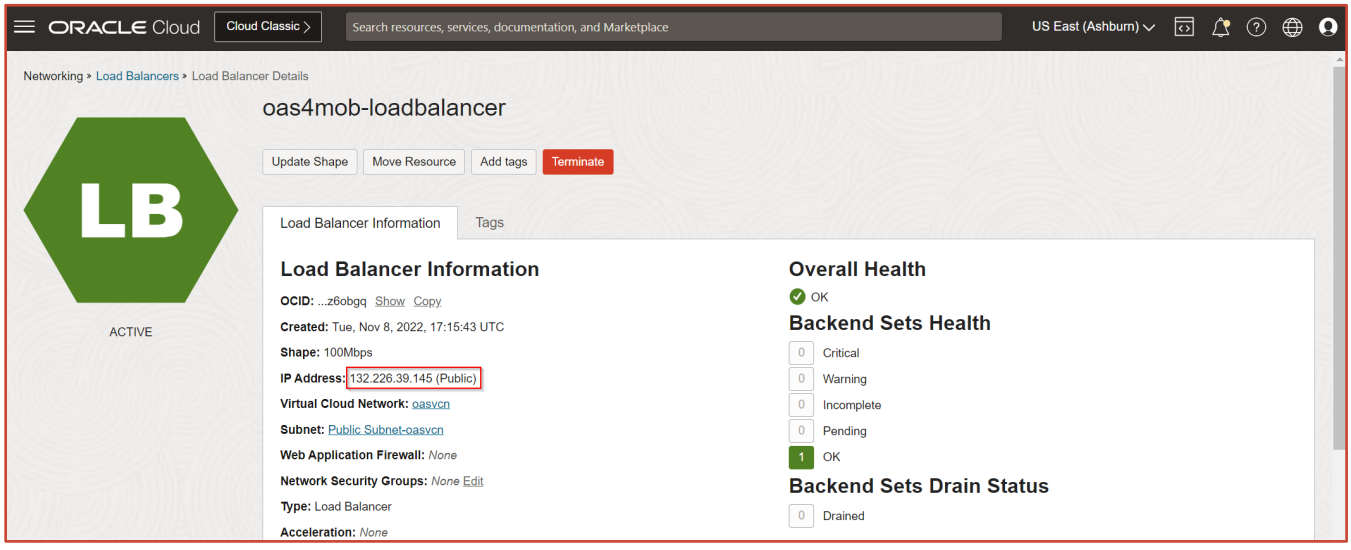




Add the App Gateway Docker Host with the port specified in the Registered App Gateway as the Backend.

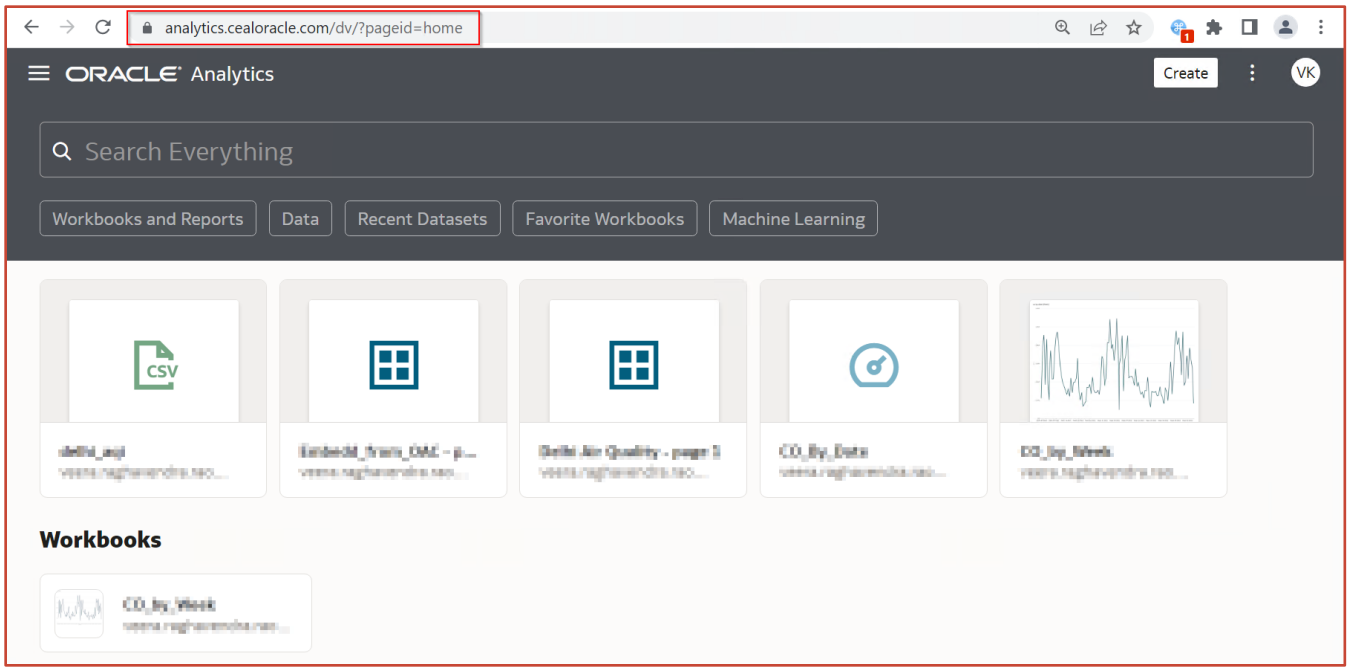






Map the Load Balancer Public IP Address to the DNS Name in your DNS Resolver and Domain Provider.

e.g., https://analytics.cealoracle.com/dv



Summary

Here we have covered the Single Sign-On Configuration of the Oracle Analytics Server on Oracle Cloud using IDCS App Gateway for a Single Node OAS.

Following blogs, cover the App Gateway SSO configuration for the clustered nodes of OAS on Oracle Cloud and High Availability of the App Gateway Servers.

Refer to Single Sign-On for Oracle Analytics Server Cluster on Oracle Cloud using App Gateway.

Refer to Single Sign-On for Oracle Analytics Server on Oracle Cloud using App Gateway High Availability.

I hope this helps you to configure the Single Sign-On Configuration of the Oracle Analytics Server on Oracle Cloud using IDCS App Gateway when integrating Oracle Analytics Server with IDCS or with the IAM Identity Domain of OCI.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle