# Oracle® Retail AI Foundation Cloud Service

## AI Foundation Private Endpoint Cloud Service: Database Access Implementation Guide

Release 25.2.401.0
G42483-01
October 2025

Oracle Retail AI Foundation Cloud Service AI Foundation Private Endpoint Cloud Service: Database Access Implementation Guide, Release 25.2.401.0

G42483-01

# Contents

# Preface

### Audience

This document is intended for the users and administrators of Oracle Retail AI Foundation Cloud Service.

### 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 customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

### Customer Support

To contact Oracle Customer Support, access My Oracle Support at the following URL:

https://support.oracle.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

### Oracle Help Center (docs.oracle.com)

Oracle Retail product documentation is available on the Oracle Help Center at https://docs.oracle.com/en/industries/retail/index.html.

(Data Model documents can be obtained through My Oracle Support.)

### Comments and Suggestions

Please give us feedback about Oracle Retail Help and Guides. You can send an e-mail to: retail-doc_us@oracle.com

### Oracle Retail Cloud Services and Business Agility

Included in the service is continuous technical support, access to software feature enhancements, hardware upgrades, and disaster recovery. The Cloud Service model helps to free customer IT resources from the need to perform these tasks, giving retailers greater business agility to respond to changing technologies and to perform more value-added tasks focused on business processes and innovation.

Oracle Retail Software Cloud Service is acquired exclusively through a subscription service (SaaS) model. This shifts funding from a capital investment in software to an operational

expense. Subscription-based pricing for retail applications offers flexibility and cost effectiveness.

1

# Introduction

The Oracle Retail AI Foundation Cloud Service (AIFCS) database is accessible through Innovation Workbench including APEX and Notebook development environments. Private endpoints extend access to AIFCS within the Virtual Cloud Network (VCN) on Oracle Cloud Infrastructure or to other networks peered to the VCN, such as your corporate network. You can access AIFCS data from hosts within the VCN or from the on-premises network.

## Prerequisites

To implement Private Endpoint access to Oracle Retail AI Foundation Cloud service, your organization must have:

- A *paid Oracle Cloud Infrastructure (OCI) tenancy* with appropriate service limits.
- An *OCI Virtual Cloud Network (VCN)* with at least one subnet in the same region as the AIFCS deployment.
- *Networking expertise* or access to experienced resources familiar with OCI networking, including VPN or FastConnect setup and DNS configuration.

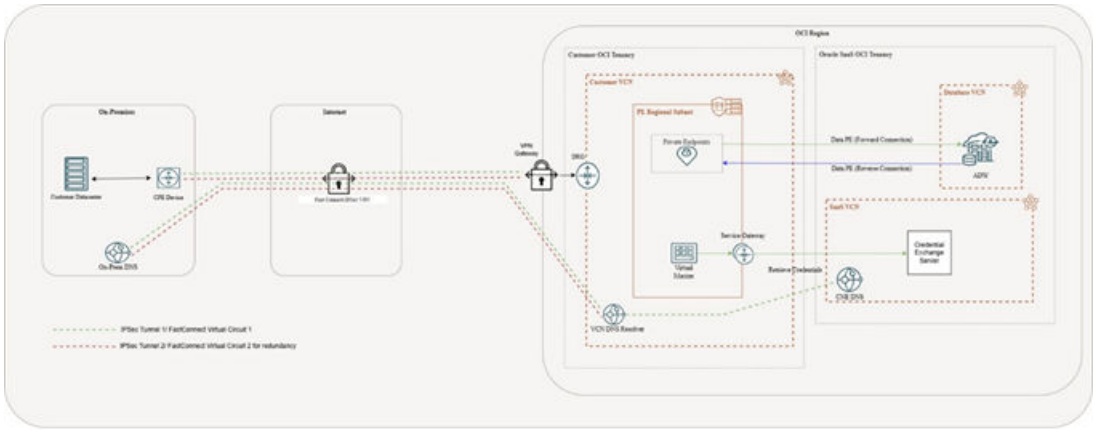
## What is a Private Endpoint?

With a private endpoint, traffic does not go over the internet. A private endpoint is a private IP address within your VCN that you can use to access a given service within Oracle Cloud Infrastructure. The service sets up the private endpoint in a subnet of your choice within the VCN. You can think of the private endpoint as just another Virtual Network Interface Card (VNIC) in your VCN. You control access to it as you would for any other VNIC by using security rules. When you set up a private endpoint for AIFCS, however, the VNIC is set up for you, and its availability is maintained on your behalf. Your only responsibility is to maintain the subnet and the security rules.

## Forward and Reverse Access

As shown in Figure 1-1, private endpoints and reverse connections enable secure, non-internet communication between your network and AIFCS.

**Figure 1-1 AIFCS Access through a Private Endpoint**



The diagram shows how AIFCS is accessed using a private endpoint deployed in the customer's VCN. Forward connections allow customer systems or services to access AIFCS and related SaaS services. Reverse connections (such as for Credential Exchange Service) enable Oracle-hosted services to securely reach designated targets within the customer's network.

**Table 1-1 Acronyms**

| Acronym | Name | Description |
|---|---|---|
| PE | Private Endpoint | A private IP address in your VCN used to access Oracle services without going over the internet. |
| DRG | Dynamic Routing Gateway | Network gateway that connects your on-premises network to your OCI VCN using VPN or FastConnect. |
| VPN | Virtual Private Network | A secure encrypted tunnel between the customer on-premises systems and OCI. |
| FastConnect | OCI FastConnect | Dedicated, private network connection between the customer on-premises data center and OCI. |
| CPE | Customer-Premises Equipment | Device on the customer's side that connects to the VPN or FastConnect. |
| ADW | Autonomous Data Warehouse | Oracle's cloud-native data warehouse service. |
| CNE DNS | Cloud Native Environment DNS | Internal DNS resolver used by Oracle-hosted Kubernetes clusters and services. |
| VCN | Virtual Cloud Network | A customizable private network in OCI, similar to a traditional data center network. |
| VCN DNS Resolver | VCN DNS Resolver | DNS resolution service for resources within a VCN. |

# Networking Expertise Required

Effectively using a private endpoint requires substantial networking expertise. For additional information, consult the Oracle documentation on OCI networking, OCI private access, FastConnect, and site-to-site VPN.

# Private Endpoint Setup Timeline

When you request a private endpoint for AIFCS, you receive an endpoint for each of your environments: production, stage, and so on. You also receive a second private endpoint that gives you access to a Credential Exchange Service (see Access Setup for the Credential Exchange Service). Establishing a private endpoint requires some lead time and a short outage on each environment (two to eight hours, depending on environment size). The outage on each environment precedes the availability of the endpoint by several days. In short, the time between your request for private endpoint access and its availability is measured in days, not hours or minutes. Oracle support will contact you to schedule environment outages.

2

# Customer Responsibilities

Once a Private Endpoint (PE) is set up in Oracle Cloud Infrastructure (OCI), customers are responsible for configuring their network to ensure connectivity. This includes both *on-premises network configuration* and *OCI-side configuration* within the VCN hosting the private endpoint.

The customer responsibilities include, but are not limited to the following.

## On-Premises Network Configuration

The customer's network team must do the following:

- **Routing**
  Ensure that routing rules allow traffic to and from the private endpoint over VPN or FastConnect using the Dynamic Routing Gateway (DRG).
- **Firewall Rules**
  Configure firewall rules to permit necessary traffic between on-premises systems and OCI services through the private endpoint.
- **DNS Resolution**
  Validate that Fully Qualified Domain Names (FQDNs) such as those for Autonomous Data Warehouse (ADW) and other AIFCS components resolve to private endpoint IP addresses. This may involve forwarding DNS queries from on-premises DNS servers to the OCI VCN DNS Resolver or Oracle-hosted CNE DNS, depending on the service.

## OCI Network Configuration

The customer must also configure networking within OCI:

- **Security Lists and NSGs**
  Update the subnet's security lists and/or Network Security Groups (NSGs) to allow inbound and outbound traffic between the private endpoint and other OCI resources.
- **Route Tables**
  Ensure that the VCN's route tables are updated to direct traffic correctly to and from the private endpoint, and if applicable, the DRG.
- **Subnet Association**
  Verify that the subnet containing the private endpoint is correctly associated with the DRG, and that the subnet has sufficient address space to accommodate Oracle-managed VNICs.
- **Reverse Connectivity for Oracle-Initiated Connections**
  Some Oracle services (such as Credential Exchange Service) initiate connections back to designated resources within the customer VCN. Ensure that the subnet allows *inbound traffic* from Oracle over the reverse connection patch and that DNS resolution supports these services.

## Testing and Validation

Before considering the setup complete:

- **Connectivity Testing**
  Confirm access to the private endpoint from on-premises systems. Use tools such as `traceroute`, `telnet`, and `nslookup` to validate routing, port accessibility, and DNS resolution.
- **OCI Diagnostics**
  Use OCI diagnostics tools such as VCN Flow Logs to verify traffic flow and troubleshoot issues.
- **Coordinate with Oracle Support**
  If connectivity issues persist and the OCI-side configurations appear to be correct, contact Oracle Support for assistance.

  Testing both the forward and reverse paths is essential to ensure full functionality.

# Additional Notes

- **Reverse Path Awareness**
  Services such as the Credential Exchange Server rely on Oracle-initiated traffic into the customer's VCN. Failure to allow reverse traffic may not affect initial setup, but will result in operational failures later.
- **High Availability**
  For production workloads, Oracle recommends redundant VPN tunnels or FastConnect circuits. Customers should verify that both paths are configured, operational, and tested.
- **Subnet Ownership**
  While Oracle provisions the VNICs used for private endpoints, the subnet is owned and maintained by the customer. Customers must ensure subnet health and configuration, including availability of private IP addresses and security policies.

For detailed guidance on OCI networking best practices, refer to the Oracle Cloud Infrastructure Networking Documentation.

3

# Requesting a Private Endpoint

This chapter describes the steps required to request a Private Endpoint for AIFCS in OCI. It outlines the prerequisites, guided console workflows, and required information you must provide when submitting your request.

You begin by identifying your tenancy and creating a dedicated subcompartment and VCN. You then create a private subnet that AIFCS will use for private endpoint access. In single Availability Domain (AD) regions, you also configure a matching standby environment.

You are responsible for creating the required IAM policies that grant the ORACLE_INDUSTRY_SAAS service permission to manage networking resources in your compartment. Once all components are in place, you collect and submit a list of OCIDs to Oracle Support to complete the private endpoint provisioning process.

Optional guidance is provided for customers who want to receive credential rotation notifications through a private HTTP or HTTPS endpoint. This chapter concludes with a reference to disaster recovery instructions for single AD regions.

## Obtain your Tenancy OCID

1. Sign in to the OCI Console for your tenancy. Ensure you are in the same region as the AIFCS deployment.
2. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Governance & Administration** > **Tenancy Details**.
3. Copy and retain your tenancy OCID in the *Tenancy Information* panel.

## Create a Dedicated Sub-Compartment

1. Sign in to the OCI Console for your tenancy. Ensure you are in the same region as the AIFCS deployment.
2. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Identity & Security** > **Compartments**.
3. Locate and click the name of your parent compartment (that is, your tenancy root or project compartment).
4. Click **Create Compartment**.
5. In the Create Compartment dialog:
   a. **Name:** Enter a name for your subcompartment (for example, aif-vcn-compartment).
   b. **Description:** (Optional) Enter a description to help identify its purpose.
   c. **Parent Compartment:** Confirm it is the intended parent.
   d. Leave tags as is unless your tenancy uses them.
6. Click **Create Compartment**.

7. Retain the name of your PE sub-compartment. Copy and retain the OCID for your sub-compartment.

# Create a Private Subnet

1. Sign in to the OCI Console for your tenancy. Ensure you are in the same region as the AIFCS deployment.
2. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Networking** > **Virtual Cloud Networks**.
3. At the top of the VCN list page, click **Start VCN Wizard** in the **Actions** menu.
4. Select *Create VCN with Internet Connectivity* for the **Connection Type**. Click **Start VCN Wizard**.
5. In the *Create VCN with Internet Connectivity* panel:
   a. **Name:** Enter a name for your VCN (for example, aif-pe-vcn).
   b. **Compartment:** Select your dedicated private endpoint compartment.
   c. **CIDR Block:** Accept the default (10.0.0.0/16) or define a custom IPv4 CIDR block.
   d. **Ipv6 in this VCN:** Leave disabled.
   e. **DNS resolution:** Leave enabled.
   f. Accept the default configurations for both public and private subnets.
6. Click **Next**.
7. Click **Create**.
8. To view your VCN, click **View VCN**.
9. Copy and retain your VCN OCID.
10. Click the **Subnets** tab.
11. At the end of the row for your private subnet, click the three-dot menu (...), and then select **Copy OCID** from the drop-down menu. Retain the OCID for your private subnet.

# Additional Requirements for Single AD Regions

If you are in a single AD region, you need to:

- Create a dedicated private endpoint subcompartment in the standby region.
- Create a private subnet in the standby region.

Use the same process already described. Copy and retain the Tenancy OCID, Compartment Name, Compartment OCID, VCN OCID, and Private Subnet details for your standby region.

# Create Compartment Policies

1. Sign in to the OCI Console for your tenancy. Use the identity domain where AIFCS is deployed.
2. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Identity & Security** > **Policies**.

3. In **Applied Filters**, select the compartment for your Private Endpoint.
4. Click **Create Policy**.
5. In the *Create Policy* panel:
   a. **Name:** Enter a name for the policy (for example, aif-vcn-vnic-access).
   b. **Description:** (Optional) Provide a description (for example, Allows AIFCS to manage VNICs in this compartment).
   c. **Compartment:** Select the compartment for your private endpoint.
6. Under Policy Builder, do the following:
   a. Select **Show manual editor**.
   b. Paste the following policy statements with the appropriate compartment name. These policies grant the ORACLE_INDUSTRY_SAAS service access only within the specified compartment. Be sure to replace <Your PE Compartment Name> with the exact name of your compartment.

```
Allow service ORACLE_INDUSTRY_SAAS to manage vnics in compartment <Your
PE Compartment Name>
Allow service ORACLE_INDUSTRY_SAAS to use subnets in compartment <Your
PE Compartment Name>
Allow service ORACLE_INDUSTRY_SAAS to use network-security-groups in
compartment <Your PE Compartment Name>
Allow service ORACLE_INDUSTRY_SAAS to inspect work-requests in
compartment <Your PE Compartment Name>
```

# Submit your Private Endpoint Request

Submit your request for a private endpoint with the following information, which you gathered during the compartment and private subnet setup process:

- Tenancy OCID
- Compartment Name
- Compartment OCID
- VCN OCID
- Private Subnet OCID

# Submit Standby Information for Single AD Regions

When you submit your request, provide the Tenancy OCID, Compartment Name, Compartment OCID, VCN OCI, and Private Subnet for your standby region as well.

### Notification Support

Oracle uses the Credential Exchange Service to notify you of database credential rotation and securely deliver database credentials to your environment. You may choose one of the following notification methods:

- **Email:** Oracle sends a notification to a specified email address.
- **HTTP or HTTPS Endpoint:** Oracle sends notification to a private service endpoint.
- **None:** No notification is sent. Your system fetches the credentials when needed.

If you choose HTTP or HTTPS, you must complete additional setup steps. Oracle will use the Credential Exchange Service to initiate connections to your private endpoint, which must be reachable through the Oracle network.

### Additional Requirements for HTTP/HTTPS Notification

This section is required only if you choose HTTP or HTTPS notification. If you choose email or no notification, you may skip this section.

If using an HTTP or HTTPS endpoint, you must:

1. Create a dedicated private subnet in a separate compartment specifically for the notification endpoint.
2. Repeat the subnet setup process described earlier in this chapter:
    a. Create a subcompartment.
    b. Create a VCN and a private subnet, and retain the associated OCIDs.
3. If necessary, add an ingress rule to the subnet's network security group or security list to allow traffic from the Credential Exchange Service to your notification endpoint.

### Include the Following in Your Request

When submitting your private endpoint request, state that you wish to receive credential rotation notifications through an HTTP or HTTPS endpoint. Provide the following details about your notification subnet:

- **VCN OCID**
- **Private Subnet OCID**
- **Private Subnet CIDR**
- **Fully Qualified Domain Name (FQDN) of the notification endpoint**

If you are unsure whether you will use HTTP or HTTPS notification or if the endpoint details are not yet available, you may choose to submit this information in a later request.

# Single AD Region Disaster Recovery

In the event of a Disaster Recovery in a single AD region, the customer must perform a number of DNS updates. When the disaster is mitigated, the customer must reverse those updates. For detailed steps on DNS updates during failover and failback in single AD regions, see Oracle Retail Cloud Services Private Connection Setup Guidance on My Oracle Support at Doc ID 2991525.1.

4

# Access Setup for the Credential Exchange Service

Before you can use the Credential Exchange Service, you must obtain OAuth 2.0 credentials that authorize access to it. These *access credentials* are distinct from the *schema or wallet credentials* returned by the service itself.

There are two parts to using OAuth 2.0 with the service:

1. **Creating an OAuth2 Client**
   A one-time setup step where a client is registered with the authorization server and issued credentials (that is, a client ID and client secret). See Creating an OAuth 2.0 Client.
2. **Obtaining and Using Access Tokens**
   A runtime step. Each time you call the service, you must include a valid, unexpired token in the request header. See Generating an Access Token.

## Setting Up Administrator Privileges to Oracle

If you do not have administrator privileges to Oracle Retail Home, you need to obtain them before setting up access. In order to obtain the privileges, you need to have permission to *manage IAM users and groups* in the OCI tenancy.

There are two sets of groups that control administrative access:

- **Production environments**

  ```
  RETAIL HOME ADMIN, PLATFORM SERVICES ADMINISTRATOR,
  PLATFORM_SERVICES_ADMINISTRATOR_ABSTRACT
  ```

- **Non-production environments (STG or UAT)**

  ```
  RETAIL HOME ADMIN PREPROD, PLATFORM SERVICES ADMINISTRATOR PREPROD,
  PLATFORM_SERVICES_ADMINISTRATOR_ABSTRACT_PREPROD
  ```

> **Note**
>
> These groups are pre-created by Oracle.

## Steps to Assign a User to an Admin Group

The following steps assume you are using the Redwood UI rather than the Classic UI.

1. Sign in to the OCI Console for your tenancy. Use the identity domain where AI Foundation Cloud Service is deployed.
2. From the left navigation menu, select **Identity & Security** > **Domains**.
3. Click the domain where your AI Foundation Cloud Service is deployed. If you are unsure which one to use, ask your tenancy administrator.

4. Click the **User Management** menu item. Groups are found at the bottom of the page.
5. Search for and select the groups to which you want to assign the user:
   - Production
   - Non-production
   - Both
6. In the group details, click the **Users** menu item.
7. Search for users by name or email. Select the appropriate user by checking the box next to their name, then click **Assign user to group**.
8. Repeat for additional users and groups as needed.

> **Note**
>
> It can take some time for group settings to propagate. It is not instantaneous. You need to log out of Oracle Retail Home and log in again before the group setting will take effect.

# Creating an OAuth 2.0 Client

OAuth2 clients are registered with an Oracle Identity Cloud Service (IDCS) server, not with individual AIFCS environments. This is a key architectural point: *the client is associated with the IDCS server*, and the server governs access to all environments under its domain (such as PROD, STG, and UAT). You may have two IDCS servers, one for production and one for non-production environments. The distinction matters.

As a result:

- A single OAuth2 client can be used to obtain access tokens that are valid for *any environment* managed by the same IDCS server. It does not matter which Oracle Retail Home environment you use to create the client so long as the environment is secured by the correct IDCS server.
- Any valid token issued by the IDCS server is accepted by the Credential Exchange Service across all of its environments secured by that server.

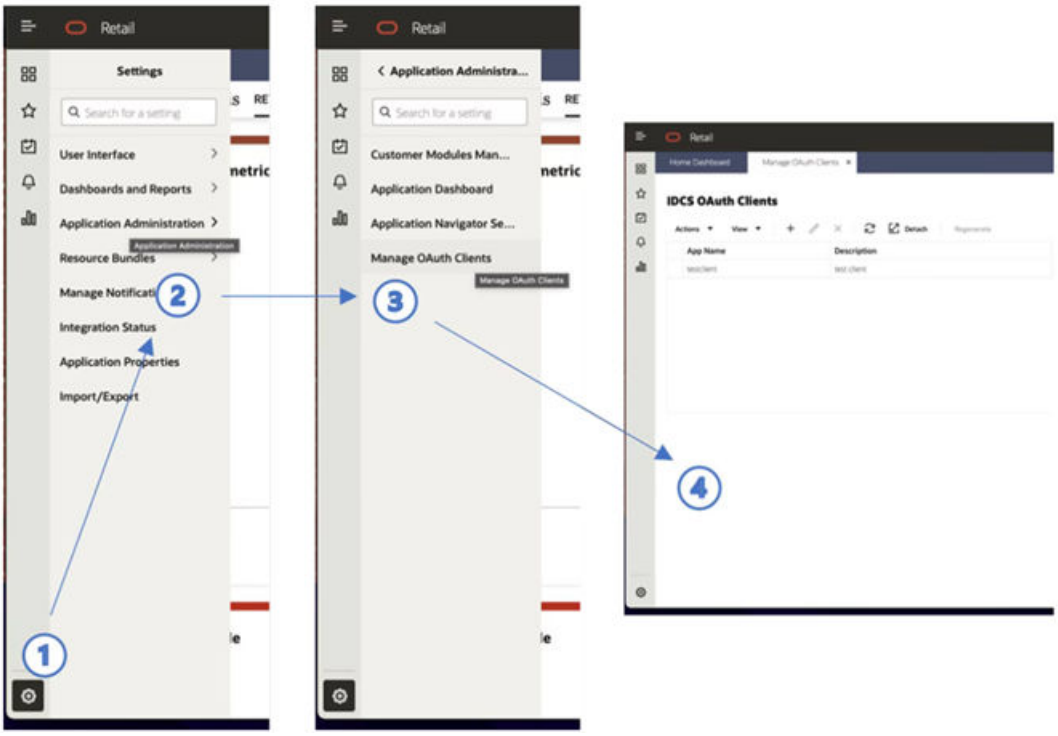
> **Note**
>
> Only one OAuth2 client is required per IDCS server. There is no need to create separate clients for each environment.
>
> This approach simplifies configuration and reduces operational overhead.

# Steps for Creating a Client

In a supported browser, navigate to Oracle Retail Home for one of your environments (that is, PROD, STG, or UAT).

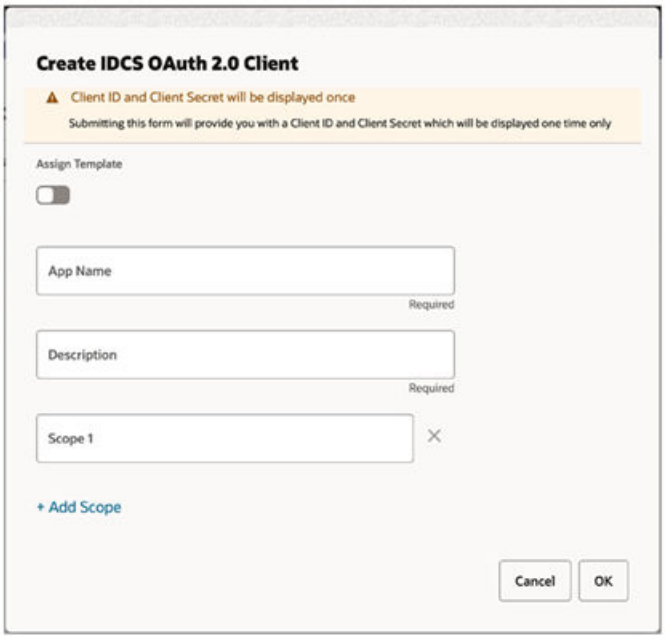
Figure 4-1 shows the first four steps for creating a client.

**Figure 4-1 OAuth 2.0 Client Creation Flow Diagram**




1. Click the settings icon in the lower left corner of the Oracle Retail Home screen. If you do not see the settings icon, you may be experiencing one of several problems:
   - You do not have administrator privileges.
   - Groups have not propagated yet.
   - You need to log out and log in to refresh your group associations.
2. Click **Application Administration**.
3. Click **Manage OAUTH Clients**.
4. On the Manage OAUTH Clients page, click **+** to add a client. The Create IDCS OAuth 2.0 Client dialog opens.

**Figure 4-2 Create IDCS OAuth 2.0 Client**




5. Provide the App Name and Description. Leave Scope blank.
6. Click **OK**. The New IDCS OAuth 2.0 Client dialog opens. It shows the Display Name, Client ID, and Client Secret.

**Figure 4-3 New IDCS OAuth 2.0 Client**


7. It is important to retain this information. It will not be displayed again. When the information has been copied, click **Done**.

Consult the *Oracle Retail Home Administration Guide* for additional details on managing OAUTH clients.

Remember that the OCI IAM service is rate-limited (see API Rate Limits). Best practice is to reuse tokens until they expire (one hour). If you encounter a 429 error when requesting a token or authenticating, you have hit the rate limit. When you encounter a rate limit, back off for one minute to reset the rate limiter.

Before proceeding:

1. Verify that a client ID and secret can be created in Retail Home.
2. Retain the client ID and secret for future use.

Remember, there is no need to create multiple OAuth clients for each OCI IAM service. A single OAuth client can be used across all environments secured by a given IDCS.

# Generating an Access Token

You need an IDCS Authorization Server endpoint URL, and an OAuth 2.0 client ID and client secret to perform the steps described below. There are multiple techniques for generating an access token. The example below employs cURL and assumes Oracle Linux 8.

The cURL command for generating an access token has five components:

1. The IDCS Authorization Server endpoint URL
2. A content type
3. A client ID and client secret
4. A grant type
5. A scope

Only the IDCS Authorization Server endpoint URL, client ID, and client secret are customer specific. Content type, grant type, and scope are the same for all customers.

# IDCS Service Host

The IDCS endpoint URL has the following format:

```
https://<idcs authorization server host>/oauth2/v1/token
```

To obtain your IDCS service host, navigate to Retail Home. When you navigate to Retail Home, you are redirected to an IDCS authorization server URL. The host of that URL is the IDCS authorization server that you use to obtain your access token. If you are already logged in to Oracle Home, log out and log back in. Once you log back in, you will be redirected to an IDCS authorization server.

# Basic Auth Encoding

To fetch a token, you need to use Basic Auth and Base 64 to encode the credentials. For example, you could use the following script to encode the Basic Auth credentials:

```
CLIENT_ID=″your_client_id″
CLIENT_SECRET=″your_client_secret″
```

Combine the client ID and secret, then encode in Base64.

```
ENCODED_CREDS=$(echo -n “${CLIENT_ID}:${CLIENT_SECRET}″ | base64 -w 0)
```

Output the result.

```
Echo “Encoded Base64 credentials: $ENCODED_CREDS″
```

Replace your_client_id and your_client_secret with the credentials obtained when creating your OAuth 2.0 client.

# Using the cURL Command

You can use cURL to generate a token using the following shell script:

```
RESPONSE=$(curl --location --request \
POST "https://<idcs authorization server host>/oauth2/v1/token" \
--header "Content-Type: application/x-www-form-urlencoded" \
--header "Authorization: Basic ${ENCODED_CREDS}" \
--data-urlencode "grant_type=client_credentials" \
--data-urlencode "scope=urn:opc:idm: myscopes ")

ACCESS_TOKEN=$(echo ${RESPONSE} | jq -r .access_token) echo ${ACCESS_TOKEN}
```

# Before Proceeding

1. Verify that you can generate an access token for each IDCS server.
2. Retain the access token if you plan to use it within the next hour.

5

# Credential Exchange Service API

## Base URL

The base URL of the Credential Exchange Service (CES) can be obtained from your Oracle Retail Home endpoint. If you do not know your Oracle Retail Home endpoint, contact your administrator.

The Oracle Retail Home endpoint has the following format:

```
https://home.retail.<region>.ocs.oracle.com/<sub-namespace-name>
```

The sub-namespace-name can be further decomposed into:

```
rgbu-common-<customerID>-<env>-rh
```

You can extract region, customerID, and env from your Oracle Retail Home URL. The host port of the base URL is 443. The base URL for the CES is:

```
https://home.retail.<region>.ocs.oracle.com:443/rgbu-common-<customerID>-<env>
```

If you are still uncertain as to how to construct the base URL or the base URL you have constructed is not working as expected, submit a Support Request for further assistance.

## On-Premises Access

If you are accessing the Credential Exchange Service from on-premises, you need to ensure that *network connectivity to the CES private endpoint* is in place. This typically requires:

- **Private connectivity** (such as Oracle FastConnect or a VPN) between your on-premises network and the Oracle Cloud Infrastructure (OCI) Virtual Cloud Network (VCN) where the CES Private Endpoint resides.
- **Ingress rules** on the subnet security list or Network Security Groups (NSGs) that allow traffic from your on-premises IP range to the CES Private Endpoint over *TCP port 443*.
- **DNS resolution** of the `home.retail.<region>.ocs.oracle.com` domain to the private IP address of the CES endpoint. This may involve a custom DNS resolver rule in OCI or DNS forwarding from your on-premises network.

Contact your organization's network and cloud administrators to confirm that all required routing, security, and DNS configurations are in place to support private access to CES from on-premise systems.

## Credential Exchange Service Endpoints

To access the CES, your PE subnet's VCN must be attached to a service gateway that forwards traffic to the Oracle Services Network. The credential CES API only accepts the traffic from the Private Network. Access from a Public Network will result in an HTTP 401 Status (Forbidden).

When you fetch credentials as JSON or a wallet ZIP file to an OCI VM as described in the examples, you need to copy it to your on-premises system.

# Fetching Credentials

| Method | Endpoint |
|---|---|
| GET | <base-url>/api/data-pe/v1/fetch-credentials |

Returns the wallet and credentials for the schemas exposed by the Database Private Endpoint. Credentials are serialized into JSON and, within that payload, Oracle Wallet file contents are base64 encoded. The JSON format of the response is shown below.

**JSON Format of Wallet and Credentials**

```
{
   "wallets": {
   "certificateEndDate": 1746276157000,
   "certificateStartDate": 1588596157000,
   "comment": null,
   "lastRotationDate": 1624305815466,
   "schemas": {
   "MFCS_RDS_CUSTOM": "password1",
   "CE_RDS_CUSTOM": "password2",
   ...
   “RASE01": "password3",
   “RABE01USER": "password3"
   },
   "wallet": {
                     "README": "...base64-encoded-file...",
                     "cwallet.sso": "...base64-encoded-file...",
                     "ewallet.p12": "...base64-encoded-file...",
                     "keystore.jks": "...base64-encoded-file...",
                     "ojdbc.properties": "...base64-encoded-file...",
                     "sqlnet.ora": "...base64-encoded-file...",
                     "tnsnames.ora": "...base64-encoded-file...",
                     "truststore.jks": "...base64-encoded-file..."
                    },
       "walletName": "Wallet_RDSADWABC123",
       "walletPassword": null
       }
}
```

**Table 5-1 Serialized Wallet and Credential Format**

| Content | Purpose |
|---|---|
| wallets | Map of the wallet contents |
| walletName | Name of the database wallet and instance, derived from tnsname.ora within the wallet |
| walletPassword | (currently unused) |
| comment | (currently unused) |
| certificateEndDate | Expiration date of the wallet, derived from the truststore certificate within the wallet |

**Table 5-1 (Cont.) Serialized Wallet and Credential Format**

| Content | Purpose |
|---|---|
| certificateStartDate | Start date of the wallet, derived from the truststore certificate within the wallet |
| lastRotationDate | Date of last rotation |
| schemas | Map of the database credentials, (username):(password) |
| wallet | Map of the wallet file contents, (filename):(base64 encoded file) |

# Fetching the Wallet

| Method | Endpoint |
|---|---|
| GET | `<base-url>/api/data-pe/v1/fetch-wallet` |

This REST call returns the wallet as a compiled zip file for use with the Database Private Endpoint. The wallet does not contain credentials; these need to be fetched from the `fetch-credentials` endpoint.

# Notification Endpoints

### Registering Notification Endpoints

| Method | Endpoint |
|---|---|
| PUT | `<base-url>/api/data-pe/v1/rotation-notification` |

**JSON Payload:** `{"usecase": "credentialRotationNotification", "endpoint": "http://example.org:80/foo/bar/baz/notification1" }`

This method inserts unique endpoints into the notification endpoint list. Duplicates are silently ignored (intended for repeat registrations from restarted callback services). The notification endpoint can be a URL in the format of HTTP, HTTPS, or MAILTO (for example, mailto:foo@bar.baz).

Registered HTTP or HTTPS endpoints are called with an HTTP POST containing a JSON payload describing the scope of the change: `{usecase:"credentialRotation", change:"<all|credentials|wallet>" }`

> **Note**
>
> If an HTTP or HTTPS endpoint is registered, you may need to add an ingress rule for <base_url> to ensure that the endpoint is reachable.

Registered MAILTO endpoints are sent a notification email. SMTP notifications are sent from the regional OCI Email Delivery Service to the email address that the customer specifies.

After receiving this notification, the consuming applications should refresh their credentials using the `fetch-credentials` or `fetch-wallet` endpoints.

### Unregistering Notification Endpoints

| Method | Endpoint |
|---|---|
| DELETE | `<base-url>/api/data-pe/v1/rotation-notification` |

**JSON Payload:** `{"usecase": "credentialRotationNotification", "endpoint": "http://example.org:80/foo/bar/baz/notification1" }`

Removes endpoints from a list. Non-existent endpoints are silently ignored.

### Listing Notification Endpoints

| Method | Endpoint |
|---|---|
| GET | `<base-url>/api/data-pe/v1/rotation-notification?tenantId=abc123` |

Returns endpoints[...] containing a list of registered endpoints, or empty endpoints [] if none exist.

**Example:**

```
{"endpoints": [ "http://example.org:80/foo/bar/baz/notification", "mailto:nobody@example.org" ] }
```

# Troubleshooting

**Table 5-2 HTTP Status Code**

| Problem | Solution |
|---|---|
| 404 | Incorrect API URL. Verify API URL. |
| 401 | Invalid, expired, or missing token. Verify that you are using a client ID and secret from the correct IAM service, that the token has not expired, that the token is valid (for example, echo CURL script), and the token is not missing. |
| 403 | Internal error. Submit a support ticket with the details of the API invocation (for example, a cURL script). |
| 200 | Response is: {“msg”:”Internal error. Cannot connect to upstream service”. “detail”:”java.net.ConnectException: Connection refused (Connection refused)”} |

6

# Verifying your Private Endpoint from an OCI VM

This chapter guides you through confirming that your AIFCS Private Endpoint is correctly configured and operational. You will create a temporary OCI VM, configure it with the required tools, fetch the necessary credentials, and connect to your AIFCS ADW database to verify connectivity.

At this point:

- Your AIFCS environments have been provisioned.
- Each AIFCS environment has an ADW instance.
- Your Private Endpoint setup is complete for each of your AIFCS environments (that is, PROD, STG, and UAT).

In order to verify that your private endpoint connection is working properly, you need an appropriately configured VM with:

- Java 11 Open JDK
- SQLcl
- A database wallet for each of your environments

> **Note**
>
> You are creating this VM for verification. If you do not intend to use it beyond verification, you should delete it to avoid future charges.

## Create an OCI VM

This section explains how to provision a compute instance in OCI. It covers selecting the appropriate compartment, network configuration, and SSH key setup to ensure the VM can access your private endpoint.

1. Sign in to the OCI Console for your tenancy. Ensure you are in the same region as the AIFCS deployment.
2. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Identity & Security** > **Compartments**.
3. Click **Create Instance**.
   a. **Name**: Enter a name for your compute instances (that is, VM).
   b. **Create in Compartment**: Select your dedicated private endpoint compartment.
   c. **Placement**: Accept the default.
   d. **Image**: Accept the default.
4. Click **Next**.

5. Accept the Security defaults. Click **Next**.
   a. **VNIC Name**: Enter a name for your VNIC (that is, VM).
   b. **Primary network**: Choose *Select existing virtual cloud network.*
      i. **Virtual cloud network compartment**: Select your dedicated private endpoint compartment.
      ii. **Virtual cloud network**: Select your dedicated private endpoint VCN.
   c. **Subnet**: Choose *Select existing subnet.*
      i. **Subnet compartment**: Select your dedicated private endpoint compartment.
      ii. **Subnet**: Select your public subnet. Although database traffic will flow through the private endpoint, the VM requires a public IP to allow SSH access for verification purposes.
   d. **Private IPv4 addresses**: Select *Automatically assign private IPv4 addresses.*
   e. **Automatically assign public IPv4 addresses**: Enabled
   f. **Add SSH keys**:
      i. Select *Generate a key pair for me.*
      ii. Click **Download private key**.
      iii. Click **Download public key**.
      iv. Retain the private and public keys, preferably in *~/.ssh/my_private.key*
   g. Accept the Storage defaults. Click **Next**.
   h. Review and then click **Create**. Your instance will take a moment to build.
   i. Click the navigation menu  in the upper left corner of the OCI Console, and then select **Compute & Instances**. Note the Public IP address of your newly created VM.

# Connecting to Your VM

This section describes how to securely access the newly created OCI VM over SSH from your local machine. The instructions provide the necessary command format and remind you to use the correct private key permissions.

The following example was run on MacOSX:

1. Open a terminal window.
2. SSH into your VM.

```
ssh -i <private_key> opc@<public_ip>
```

# Configuring Your VM

This section describes how to prepare the VM with the tools needed to connect to your AIFCS database. It includes steps to install Java 11 (required for SQLcl), download and set up SQLCI, and configure your environment for easy execution.

1. Install Java 11 Open JDK (needed for SQLcl).

```
sudo dnf install java-11-openjdk
```

2. Install SQLcl.

```
# In HOME directory create a tools directory
mkdir tools
# Set current working directory to ~/tools
cd tools
# Download and unzip SQLcl
curl -O https://download.oracle.com/otn_software/java/sqldeveloper/sqlcl-
latest.zip
unzip sqlcl-latest.zip
# Return to HOME directory
cd
# For persistence, append the export line to .bashrc
echo 'export PATH=$PATH:$HOME/tools/sqlcl/bin' >> ~/.bashrc
# Update Linux environment
source ~/.bashrc
```

# Fetching Credentials for your VM

You will retrieve the database wallet and schema credentials required for secure access. The steps guide you through generating an authentication token, fetching credentials from the Credential Exchange Service, decoding the wallet files, and configuring them for SQLcl.

1. Generate the authentication token. In order to obtain a wallet for accessing the AIFCS database through your private endpoint, you need an access token.

```
#Assign you client id and secret to environment variables.
#Replace your_client_id and your_client_secret with the
#credentials obtained when creating your OAuth 2.0 client.
CLIENT_ID=”your_client_id”
CLIENT_SECRET=”your_client_secret”
#Combine the client ID and secret, then encodei n Base64
#Assign the result to an environment variable
ENCODED_CREDS=$(echo -n “${CLIENT_ID}:${CLIENT_SECRET}” | base64 -w 0)

echo “Encoded Base64 credentials: $ENCODED_CREDS

#Use cURL to obtain the authentication token
RESPONSE=$(curl --location --request\
POST https://<idcs authorization server host>/oauth2/v1/token \
--header “Content-Type: application/x-www-form-urlencoded” \
--header “Authorization: Basic #{ENCODED_CREDS}” \
--data-urlencode “grant_type=client_credentials” \
--data_urlencode “scope=urn:opc:idm: myscopes “{

#Extract the token using jq
ACCESS_TOKEN=$(echo ${RESPONSE} | jq -r .access_token)
echo ${ACCESS_TOKEN}
```

2. Fetch the token.

```
# Assign your region, customer_id, and env to environment variables.
REGION=your_region
CUSTOMER_ID=your_customer_id
ENV=your_env
```

```
CES_URL=https://home.retail.${REGION}.ocs.oracle.com:443/rgbu-common-$
{CUSTOMER_ID}-
${ENV}
curl --location "${CES_URL}/api/data-pe/v1/fetch-credentials" \
--header "Authorization: Bearer ${ACCESS_TOKEN}" > response.json
```

3. Decode the wallet.

```
# Extract wallet and schema credentials
cat response.json | jq -r '.wallets' > wallets.json
cat response.json | jq -r .schemas > schemas.json
# Create a wallet directory
mkdir -p wallet
# Extract wallets to the wallet directory
cat wallets.json | jq -r '.wallet.README' | base64 -d > wallet/README
cat wallets.json | jq -r '.wallet."cwallet.sso"' | base64 -d > wallet/
cwallet.sso
cat wallets.json | jq -r '.wallet."ewallet.p12"' | base64 -d > wallet/
ewallet.p12
cat wallets.json | jq -r '.wallet."keystore.jks"' | base64 -d > wallet/
keystore.jks
cat wallets.json | jq -r '.wallet."ojdbc.properties"' | base64 -d >
wallet/ojdbc.properties
cat wallets.json | jq -r '.wallet."sqlnet.ora"' | base64 -d > wallet/
sqlnet.ora
cat wallets.json | jq -r '.wallet."tnsnames.ora"' | base64 -d > wallet/
tnsnames.ora
cat wallets.json | jq -r '.wallet."truststore.jks"' | base64 -d >
wallet/truststore.jks
```

4. Set the TNS Admin.

```
#Make wallet accessible to SQLcl
export TNS_ADMIN=$HOME/wallet
echo ‘TNS_ADMIN=$HOME/wallet ‘ >> ~/.bashrc
```

# Connecting to your Database

This section describes using the wallet and SQLcl to establish a secure connection to your AIFCS ADW database. You will select a schema and service name, set the TNS_AMDIN variable, and verify successful connectivity.

In order to connect to your database, you need:

- A schema name and password

  Choose a schema name and password from your schemas.json file.

- A service name

  Choose a service name from tnsnames.ora, for example *<name>_low.*

Log in to your AIFCS ADW database using:

```
sql RASE01@<name>_low
```

# Securely Copying your Wallet with SCP

You can use Secure Copy Protocol (SCP) to securely transfer your database wallet to and from your OCI VM over an encrypted SSH connection. Always keep your private key and wallet files protected.

**Upload the Wallet to the OCI VM (local to remote)**

```
scp -i ~/.ssh/my_private.key /path/to/wallet.zip opc@<public_ip>:/home/opc/
```

**Download the Wallet from the OCI VM (remote to local)**

```
scp -i !/.ssh/my_private.key opc@<public_ip>:/home/opc/wallet.zip /path/to/local/
```

**Guidelines**

- Use the -i option to specify the SSH private key generated when you provisioned the VM.
- Transfer files only to secure directories such as /home/opc/.
- After copying, set restrictive permissions on the wallet: `chmod 600 /home/opc/wallet.zip`
- Extract and use the wallet only where needed.
- Delete the wallet files from both the VM and your local machine after verification to prevent unauthorized access.

# Security and Cost Control

These practices help prevent unauthorized access and reduce exposure of sensitive data:

- **Delete the verification VM** once you have confirmed that the private endpoint is functioning correctly. Leaving it running may expose it to unnecessary risk and incur additional charges.
- **Securely remove any downloaded credentials**, including the database wallet and schema password files, after verification is complete.
- **Do not share the private key, wallet files, or access tokens** with unauthorized users. Store them in a secure location and limit access to only those who require it.
- **Rotate OAuth2 client secrets** if they were exposed during testing, following your organization's security policies.