

Oracle® Retail Data Store Private Endpoint Database Access Implementation Guide



G35995-02
August 2025



Copyright © 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
	Prerequisites	1
	What Is a Private Endpoint?	1
	Forward and Reverse Access	1
	Networking Expertise Required	2
	Private Endpoint Setup Timeline	3
2	Customer Responsibilities	
	On-Premises Network Configuration	1
	OCI Network Configuration	1
	Testing and Validation	2
	Additional Notes	2
3	Requesting a Private Endpoint	
	Obtain Your Tenancy OCID	1
	Create a Dedicated Sub-Compartment	1
	Creating a Private Subnet	2
	Additional Requirements for Single AD Regions	2
	Create Compartment Policies	2
	Security List Updates	3
	Submit Your Private Endpoint Request	4
	Submit Standby Information for Single AD Regions	4
	Notification Support	4
	Additional Requirements for HTTP/HTTPS Notification	4
	Include the Following in Your Request	5
	Single AD Region Disaster Recovery	5
4	Access Setup for the Credential Exchange Service	
	Setting Up Administrator Privileges to Oracle Retail Home	1
	Steps to Assign a User to an Admin Group	1
	Creating an OAuth 2.0 Client	2

Steps for Creating Client	2
Generating an Access Token	5
IDCS Service Host	6
Basic Auth Encoding	6
Combine the Client ID and Secret, then Encode in Base64	6
Output the Result	6
Using the cURL Command	6
Before Proceeding	7

5 Credential Exchange Service API

Base URL	1
On-Premise Access	1
Credential Exchange Service Endpoints	2
Fetching Credentials	2
Fetching Wallet	3
Registering Notification Endpoints	3
Unregistering Notification Endpoints	4
Listing Notification Endpoints	4
Troubleshooting	4

6 Verifying Your Private Endpoint from an OCI VM

Create an OCI VM	1
Connecting to Your VM	2
Configure Your VM	2
Fetching Credentials	3
Step 1: Generating Authentication Token	3
Step 2: Fetch	4
Step 3: Decode Wallet	4
Step 4: Set TNS ADMIN	4
Connecting to Your Database	5
Security Note – Securely Copying Your Wallet with SCP	5
Upload Wallet to the OCI VM (local → remote)	5
Download Wallet from the OCI VM (remote → local)	5
Guidelines	5
For Security and Cost Control	6

Preface

This guide describes the Implementation considerations for database access, for Retail Data Store Private Endpoint.

Audience

This guide is intended for administrators and describes the administration and implementation tasks for Oracle Retail Data Store Database Access.

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 following website <https://docs.oracle.com/en/industries/retail/html>

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

Oracle Retail Merchandising Cloud Services is hosted in the Oracle Cloud with the security features inherent to Oracle technology and a robust data center classification, providing significant uptime. The Oracle Cloud team is responsible for installing, monitoring, patching, and upgrading retail software.

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 Data Store (RDS) is accessible through the APEX Developer Environment as well as through custom APEX applications and services developed by the customer. Private endpoints extend access to RDS within your virtual cloud network (VCN) on Oracle Cloud Infrastructure or to other networks peered to the VCN such as your corporate network. That is, you can access RDS from hosts within your virtual cloud network (VCN) or from your on-premises network.

Prerequisites

To implement Private Endpoint access to Oracle Retail Data Store, your organization must have:

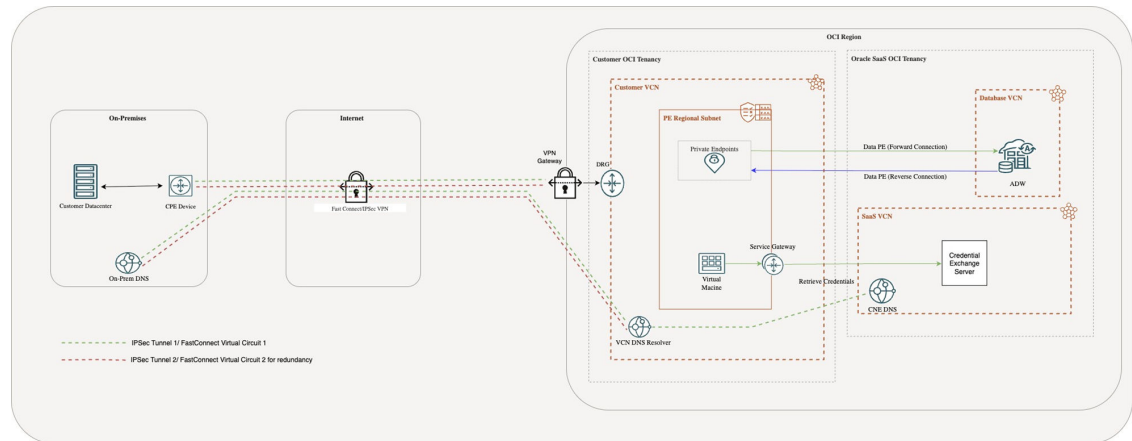
- A **paid Oracle Cloud Infrastructure (OCI) tenancy** with appropriate service limits.
- An **OCI Virtual Cloud Network (VCN)** with at least one subnet in the **same region as the RDS deployment**.
- **Networking expertise** or access to experienced resources familiar with OCI networking, including 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 RDS, 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 seen in Figure 1-1, private endpoints and reverse connections enable secure, non-internet communication between your network and Oracle Retail Data Store.

Figure 1-1 Overview

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

Table 1-1 Figure 1-1 Acronyms

RDS	Retail Data Store	Oracle Retail data warehouse offering accessed through APEX, REST, or SQL interfaces.
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 customer on-premises systems and OCI.
FastConnect	OCI FastConnect	Dedicated, private network connection between 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 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 RDS, 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 (discussed in more detail below). 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.

These responsibilities include, but are not limited to:

On-Premises Network Configuration

The customer's network team must:

- **Routing**
Ensure that routing rules allow traffic to and from the private endpoint over VPN or FastConnect via 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 RDS 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 (for example, 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 path 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 diagnostic tools such as **VCN Flow Logs** to verify traffic flow and troubleshoot issues.
- **Coordinate with Oracle Support**
If connectivity issues persist and 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 Oracle Retail Data Store (RDS) in Oracle Cloud Infrastructure (OCI). It outlines the prerequisites, guided console workflows, and required information you must provide when submitting your request.

You will begin by identifying your tenancy and creating a dedicated subcompartment and VCN. You'll then create a private subnet that RDS will use for private endpoint access. In single availability domain (AD) regions, you'll 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 will 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. The 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 RDS deployment.
2. Click the **navigation menu** (☰) in the upper left corner of the OCI Console, then go to **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 RDS deployment.
2. Click the **navigation menu** in the upper left corner of the OCI Console, then go to **Identity & Security** → **Compartments**
3. Locate and **click the name of your parent compartment** (for example, your tenancy root or project compartment)
4. Click **Create Compartment** in the **Create Compartment** dialog:
 - **Name:** Enter a name for your subcompartment (e.g., rds-vcn-compartment).
 - **Description:** (Optional) Enter a description to help identify its purpose.
 - **Parent Compartment:** Confirm it is the intended parent. Leave tags as-is unless your tenancy uses them.
5. Click **Create Compartment**.
6. Retain the name of your PE sub-compartment. Copy and retain the OCID for your sub-compartment

Creating a Private Subnet

1. **Sign in to the OCI Console** for your tenancy. Ensure you are in the same region as the RDS deployment.
2. Click on the **navigation menu** in the upper left corner of the OCI Console, then go to **Networking** → **Virtual Cloud Networks**
3. At the top of the VCN list page, click **Start VCN Wizard** from the **Actions** menu.
4. Select *Create VCN with Internet Connectivity* as the **Connection Type**. Click the **Start VCN Wizard** button.
5. In the *Create VCN with Internet Connectivity* panel:
 - **Name:** Enter a name for your VCN (for example, `rds-pe-vcn`).
 - **Compartment:** Select your dedicated private endpoint compartment.
 - **CIDR Block:** Accept the default (`10.0.0.0/16`) or define a custom IPv4 CIDR block.
 - Leave **IPv6 in this VCN** disabled.
 - Leave **DNS resolution** enabled.
 - Accept default configurations for both public and private subnets.
6. Click **Next**
7. Click **Create**
8. 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 (⋮), then select **Copy OCID** from the dropdown menu. Retain the OCID for your private subnet.

Additional Requirements for Single AD Regions

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

- create a dedicated private endpoint subcompartment in the standby region
- create private subnet in standby region

Use the same process as described above. Copy and retain 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 Retail Data Store is deployed.
2. Click the **navigation menu** in the upper left corner of the OCI Console, then go to: **Identity & Security** → **Policies**.
3. In **Applied Filters** select the compartment for your Private Endpoint.
4. Click the **Create Policy** button.
5. In the **Create Policy** panel:

- **Name:** Enter a name for the policy (for example, `rds-vcn-vnic-access`).
 - **Description:** (Optional) Provide a description (for example, `Allows RDS to manage VNICS in this compartment`).
 - **Compartment:** Select the compartment for your private endpoint.
6. Under Policy Builder, do the following:
- Select **Show manual editor**.
 - 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>
```

Security List Updates

When private endpoint setup is complete, Oracle Support provides you with details for each of your private endpoints. To access RDS from within OCI, you need to edit the security list Ingress Rules of your private subnet. See Table 3-1 for typical ingress rules. Your rules may be different.

Table 3-1 Typical Ingress Rules for Private Endpoint

Attribute	Value
STATELESS	No
SOURCE CIDR	(10.0.0.0/16)
IP PROTOCOL	TCP
SOURCE PORT RANGE	All
DESTINATION PORT RANGE	443, 1521-1522
TYPE AND CODE	(Blank)
ALLOWS	All
DESCRIPTION	(Optional)

The source should be the CIDR of the subnet where the private endpoints are deployed. Customers can create VCNs with various CIDR ranges, such as `10.0.0.0/16`, `192.168.0.0/16`, or `172.16.0.0/12`, and the source CIDR should match the CIDR of the subnet where the private endpoint is deployed.

Customers should ensure that there are no route rules that route traffic from the PE subnet/VCN to the DRG, NAT, or Internet Gateway, as this can cause issues with the private endpoint. The route tables should be configured to prevent traffic from being routed to these gateways.

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

Question: Do you anticipate using Oracle Integration Cloud (OIC) or any other capability that requires a wallet password? If you do not know the answer to this question at this time, leave it blank. You can request a wallet password in the future should you need one by submitting a Support Request.

Submit Standby Information for Single AD Regions

When you submit your request, provide: Tenancy OCID, Compartment Name, Compartment OCID, VCN OCID, 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 a notification to a private service endpoint.
- **None:** No notification is sent. Your system fetches 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:
 - Create a subcompartment.
 - 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.

4. Ensure the **Fully Qualified Domain Name (FQDN)** of the notification endpoint is resolvable from within OCI.

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. Refer to My Oracle Support [Doc ID: 2991525.1](#) for detailed steps on DNS updates during failover and failback in single AD regions.

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 the [Creating an OAuth 2.0 Client](#) section.

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 the [Generating an Access Token](#).

Setting Up Administrator Privileges to Oracle Retail Home

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

Two sets of groups 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 steps below assume you are using the Redwood UI rather than Classic UI.

1. **Sign in to the OCI Console** for your tenancy.
Use the identity domain where Retail Data Store is deployed.
2. From the **left navigation menu**, go to: **Identity & Security** → **Domains**

3. Click the domain where your Retail Data Store is deployed. If you're unsure which one to use, ask your tenancy administrator.
4. Click the **User Management** tab. **Groups** will be found at the bottom of the page.
5. Search for and select the group(s) to which you want to assign the user:
 - Production,
 - Non-production, or
 - 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.

It can take some time for group settings to propagate. It is not instantaneous. You will also 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 RDS 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 (for example, PROD, STG, UAT). You may have two IDCS servers, one for production and one for non-production environments. This 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.

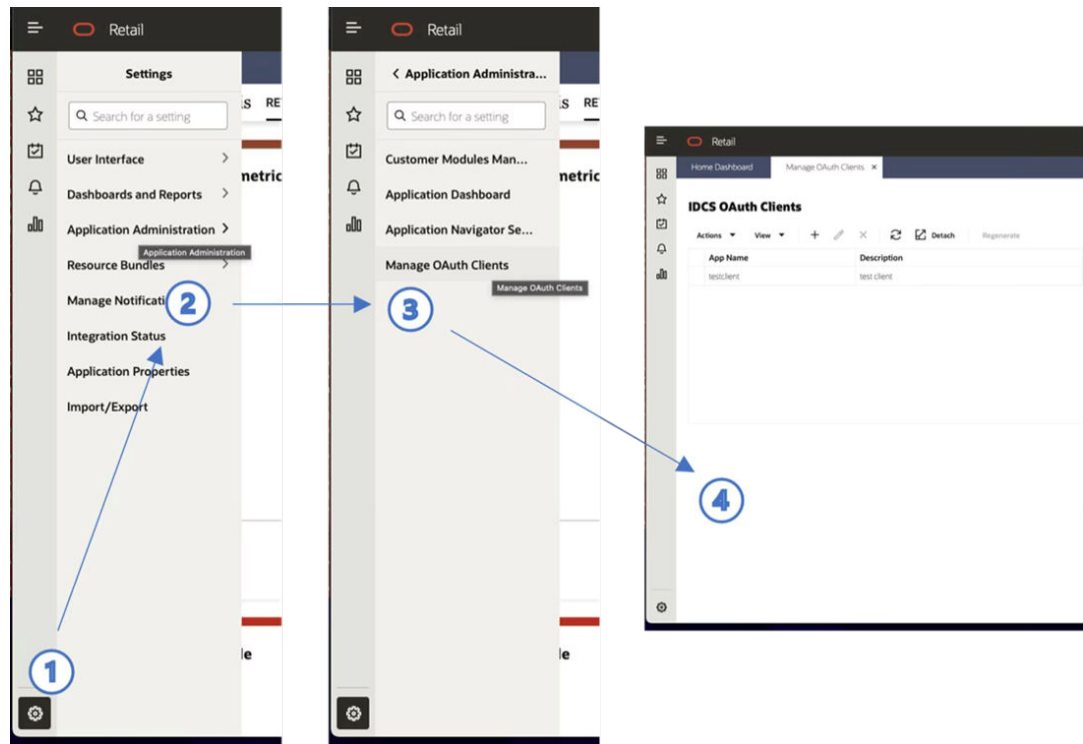
Steps for Creating Client

In a supported browser, navigate to Oracle Retail Home for one of your environments (for example, PROD, STG, UAT). Refer to Figure "OAuth 2.0 Client Creation Flow Diagram" for the first four steps.

1. Select the settings icon (a gear) in the lower left 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 back in to refresh your group associations
2. Select the **Application Administration** menu item
 3. Select the **Manage OAUTH Clients** menu item
 4. On the Manage OAUTH Clients page, select the + button.

Figure 4-1 OAuth 2.0 Client Creation Flow Diagram



5. A popup dialog will appear. See Figure "Create IDCS OAuth 2.0 Client". Provide an **App Name** and **Description**. Leave **Scope** blank.
6. Select **OK**.

Figure 4-2 Create IDCS OAuth 2.0 Client

Create IDCS OAuth 2.0 Client

⚠ Client ID and Client Secret will be displayed once

Submitting this form will provide you with a Client ID and Client Secret which will be displayed one time only

Assign Template

☐

App Name Required

Description Required

Scope 1 ×

[+ Add Scope](#)

Cancel OK

7. A new dialog window will appear with a **Display Name**, **Client ID**, and **Client Secret**. See Figure "New IDCS OAuth 2.0 Client". **Retain this information!** It will not be displayed again. Select **Done** when the information has been copied.

Figure 4-3 New IDCS OAuth 2.0 Client

New IDCS OAuth 2.0 Client

Display Name
RDS_IG_APP3

Client ID
RDS_IG_APP3_APPID

Client Secret
idcscs-d8866db6-29fd-4c3b-a39d-640e16abf6b6

Done

Consult the *Retail Home Application 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 environment. A single OAuth client can be used across all environments secured by a given IDCS.

Generating an Access Token

You will 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 form:

```
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 will be redirected to an IDCS authorization server URL. The host of that URL is the IDCS authorization server that you will use to obtain your access token. If you are already logged into Oracle Retail Home, log out and log back in. When 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 Base64 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 \\\nPOST "https://<idcs authorization server host>/oauth2/v1/token" \\\n--header "Content-Type: application/x-www-form-urlencoded" \\\n--header "Authorization: Basic ${ENCODED_CREDS}" \\\n--data-urlencode "grant_type=client_credentials" \\\n--data-urlencode "scope=urn:opc:idm: myscope")  
ACCESS_TOKEN=$(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 discovered 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 form:

```
https://home.retail.<region>.ocs.oracle.com/<subnamespace-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-Premise Access

If you are accessing the Credential Exchange Service from on-premise, you will 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-premise 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-premise 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-premise 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 Oracle Services Network. The credential CES API only accepts the traffic from the Private Network. Access from a Public Network will result in a HTTP 401 Status (Forbidden).

When you fetch credentials as JSON or a wallet ZIP file to an OCI VM as described in examples, you will need to copy it to your on-prem 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 in "JSON Format of Wallet and Credentials".

JSON Format of Wallet and Credentials

```
{
  "wallets": {
    "certificateEndDate": 1746276157000,
    "certificateStartDate": 1588596157000,
    "comment": null,
    "lastRotationDate": 1624305815466,
    "schemas": {
      "MFCS_RDS_CUSTOM": "password1",
      "CE_RDS_CUSTOM": "password2",
      ...
      RDS_CUSTOM_2": "password3",
      RDS_CUSTOM_3": "password3"
    },
  },
  "wallet": {
    "README": "...base64-encoded-file...",
    "cwallet.sso": "...base64-encoded-file...",
    "ewallet.pl2": "...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 wallet contents
walletName	Name of database wallet and instance, derived from tnsnames.ora within wallet
walletPassword	(Currently unused)
comment	(Currently unused)
certificateEndDate	Expiration date of wallet, derived from truststore certificate within wallet
certificateStartDate	Start date of wallet, derived from truststore certificate within wallet
lastRotationDate	Date of last rotation
schemas	Map of database credentials (username):(password)
wallet	Map of wallet file contents, (filename):(base64 encoded file)

Fetching 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.

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 form 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	<code><base-url>/api/data-pe/v1/rotation-notification</code>

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	<code><base-url>/api/data-pe/v1/rotation-notification?tenantId=abc123</code>

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 the a client ID and secret from the correct IAM service, that the token has not expired, that the token is valid (e.g., 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)

Table 5-2 (Cont.) HTTP Status Code

Problem	Solution
200	Response is: <pre>{ "msg": "Internal error, cannot connect to upstream service", "detail": "java.net.ConnectException: Connection refused (Connection refused)" }</pre>

6

Verifying Your Private Endpoint from an OCI VM

This section guides you through confirming that your RDS 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 RDS ADW database to verify connectivity.

At this point:

- Your RDS environments have been provisioned
- Each RDS environment has a ADW instance
- Your cloud service apps (for example, MFCS, XOffice) are replicating to RDS
- Your Private Endpoint setup is complete for each of your RDS environments (for example, PROD, STG, UAT)

In order to verify that your private endpoint connection is working properly, you will 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, then you should delete it to avoid future charges.

Create an OCI VM

This part explains how to provision a compute instance in OCI using the Redwood UI. 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 RDS deployment.
2. Click on the **navigation menu** in the upper left corner of the OCI Console, then go to **Compute** → **Instances**
3. Click **Create Instance**.
 - **Name**: Enter a name for your compute instance (that is, VM).
 - **Create in Compartment**: Select your dedicated private endpoint compartment.
 - **Placement**: Accept the default.
 - **Image**: Accept the default.
4. Click the **Next** button.

5. Accept the Security defaults.
6. Click the **Next** button.
 - **VNIC Name:** Enter a name for your VNIC (that is, VM).
 - **Primary network:** Pick **Select existing virtual cloud network**.
 - **Virtual cloud network compartment:** Select your dedicated private endpoint compartment.
 - **Virtual cloud network:** Select your dedicated private endpoint vcn.
 - **Subnet:** Pick **Select existing subnet**.
 - **Subnet compartment:** Select your dedicated private endpoint compartment.
 - **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.
 - **Private IPv4 addresses:** Pick *Automatically assign private IPv4 addresses*.
 - **Automatically assign public IPv4 addresses:** Enabled
 - **Add SSH keys**
 - Pick **Generate a key pair for me**.
 - Click **Download private key**.
 - Click **Download public key**.
 - Retain private and public keys, preferably in `~/.ssh`.
 - Set permissions, that is, `chmod 600 ~/.ssh/my_private.key`
 - Accept the Storage defaults. Click the **Next** button.
 - Review and click the **Create** button. Your Instance will take a moment to build.
 - Click the **navigation menu** in the upper left corner of the OCI Console, then go to **Compute** → **Instances**. Note the Public IP address of your newly created VM.

Connecting to Your VM

Here you will learn 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>
```

Configure Your VM

This section outlines how to prepare the VM with the tools needed to connect to your RDS database. It includes steps to install Java 11 (required for SQLcl), download and set up SQLcl, 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

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.

Step 1: Generating Authentication Token

In order to obtain a wallet for accessing to the RDS ADW database through your private endpoint, you will need an access token.

```
# Assign your 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 encode in 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://<ids authorization server host>/oauth2/v1/token" \
--header "Content-Type: application/x-www-form-urlencoded" \
--header "Authorization: Basic ${ENCODED_CREDS}" \
--data-urlencode "grant_type=client_credentials" \
--data-urlencode "scope=urn:opc:idm: myscope ")
# Extract the token using jq
ACCESS_TOKEN=$(echo ${RESPONSE} | jq -r .access_token)
echo ${ACCESS_TOKEN}
```

Step 2: Fetch

```
# 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
# chmod 600 sets permissions to rw-----
# so only the owner can read/write the file,
# preventing group/others from accessing it (used for sensitive files).
chmod 600 response.json
```

Step 3: Decode Wallet

```
# Extract wallet and schema credentials
cat response.json | jq -r '.wallets' > wallets.json
cat response.json | jq -r '.schemas' > schemas.json
# Set files to 600 so only the owner can read/write
# the files
chmod 600 wallets.json 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
# Set directories to 700 and files to 600
# so only the owner can read/write the files and directory
chmod 700 wallet chmod 600 wallet/*
```

Step 4: Set TNS ADMIN

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


Connecting to Your Database

This final part walks you through using the wallet and SQLcl to establish a secure connection to your RDS ADW database. You will select a schema and service name, set the TNS_ADMIN variable, and verify successful connectivity.

In order to connect to your database, you will 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`, something like `<name>_low`.

Login to your RDS ADW database using:

```
sql mfcs_rds_custom@<name>_low
```

Security Note – Securely Copying Your Wallet with SCP

You can use **SCP (Secure Copy Protocol)** 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 Wallet to the OCI VM (local → remote)

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

Download Wallet from the OCI VM (remote → 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.

For Security and Cost Control

- **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.

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