Oracle® Cloud

Connecting Oracle Analytics Cloud to Your Data

F25037-56
May 2024

Oracle Cloud Connecting Oracle Analytics Cloud to Your Data,

F25037-56

Primary Author: Rosie Harvey

Contributors: Oracle Analytics Cloud development, product management, and quality assurance teams

Contents

3 Connect to Data

4 Connect to Data for Pixel-Perfect Reports

B Frequently Asked Questions

C Troubleshoot

Preface

Learn how to connect to your data.

Topics:

Audience

This guide is intended for business intelligence analysts and administrators who use Oracle Analytics Cloud.

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.

Diversity and Inclusion

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

Related Documents

These related Oracle resources provide more information.

- Getting Started with Oracle Analytics Cloud

Conventions

Conventions used in this document are described in this topic.

Text Conventions

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

Videos and Images

Your company can use skins and styles to customize the look of the Oracle Analytics Cloud, dashboards, reports, and other objects. It is possible that the videos and images included in the product documentation look different than the skins and styles your company uses.

Even if your skins and styles are different than those shown in the videos and images, the product behavior and techniques shown and demonstrated are the same.

Part I

Getting Started with Connecting Oracle Analytics Cloud to Your Data

This part explains how you get started with connecting Oracle Analytics Cloud to your data.

Chapters:

1

Get Started with Data Sources in Oracle Analytics

Topics

About Data Sources

You can connect to many types of data source, such as Cloud databases, on-premises databases, and many commonly used applications, such as Dropbox, Google Drive, and Amazon Hive.

You create a connection for each data source that you want to access in Oracle Analytics. Once connected, you can visualize your data to create insights.

A data source is any tabular structure. You see data source values after you load a file or send a query to a service that returns results.

A data source can contain any of the following:

- **Match columns** - These contain values that are found in the match column of another source, which relates this source to the other, for example, Customer ID or Product ID.
- **Attribute columns** - These contain text, dates, or numbers that are required individually and aren’t aggregated, for example, Year, Category Country, Type, or Name.
- **Measure columns** - These contain values that should be aggregated, for example, Revenue or Miles driven.

You can analyze a data source on its own, or you can analyze two or more data sources together, depending on what the data source contains. If you use multiple sources together, then at least one match column must exist in each source. The requirements for matching are:

- The sources contain common values, for example, Customer ID or Product ID.
- The match must be of the same data type, for example, number with number, date with date, or text with text.

When you save a workbook, the permissions are synchronized between the workbook and the external sources that it uses. If you share the workbook with other users, then the external sources are also shared with those same users.

Any data that you upload (as a dataset) is stored securely in Oracle Cloud.

Data Sources and Subject Areas

You can combine data sources with subject areas to explore and analyze the data.

A subject area either extends a dimension by adding attributes or extends facts by adding measures and optional attributes. You can't define hierarchies in data sources.

A subject area organizes attributes into dimensions, often with hierarchies, and a set of measures, often with complex calculations, that can be analyzed against the dimension attributes, for example, the measure net revenue by customer segment for the current quarter, and the same quarter a year ago.

When you use data from a source such as an Excel file, the file adds information that is new to the subject area. For example, suppose you purchased demographic information for postal areas or credit risk information for customers and want to use this data in an analysis before adding the data to the data warehouse or an existing subject area.

Using a source as standalone means that the data from the source is used independently of a subject area. It's either a single file used by itself or it's several files used together and in both cases a subject area isn't involved.

You can extend a dimension by adding attributes from a data source to a subject area:

- You can only make matches to a single dimension.
- The set of values in matched columns are unique in the data source. For example, if the data source matches on ZIP code, then ZIP codes in the source are unique.
- You make matches between one or composite columns. An example of a one column match is that product key matches product key. For composite columns, an example is that company matches company and business unit matches business unit.
- All other columns must be attributes.

You can add measures from a data source to a subject area:

- You make matches to one or more dimensions.
- The set of values in matched columns aren't necessarily unique in the data source. For example, if the data source is a set of sales matched to date, customer, and product, then you can have multiple sales of a product to a customer on the same day.
- You make matches between one or composite columns. An example of a one column match is that product key matches product key. For composite columns, an example is that city and state from separate columns creates the composite City_State in a customers address.

A data source that adds measures can include attributes. You can use these attributes alongside external measures and not alongside curated measures in visualizations. For example, when you add a source with the sales figures for a new business, you can match these new business sales to an existing time dimension and nothing else. The data might include information about the products sold by this new business. You can show the sales for the existing business with those of the new business by time, but you can't show the old business revenue by new business products, nor can you show new business revenue by old business products. You can show new business revenue by time and new business products.

Data Sources and Measure Columns

You can work with data sources that either include or don't include a measure column.

- You can match tables with measures to other tables with a measure, a dimension, or both.
- When you match tables to other tables with measures, the tables don't need to be at the same grain. For example, you can match a table of daily sales table to a sales by quarter table.

A table with no measures is treated as a dimension.

- Matches can be between single or composite columns. A single column match might be the product key in one table matching the product key in another. A composite column match might be where company and business unit in one table matches company and business unit in the other table.
- All other columns must be attributes.

Dimension tables can be matched to other dimensions or they can be matched to tables with measures. For example, a table with Customer attributes can be matched to a table with demographic attributes provided both dimensions have unique Customer key columns and Demographic key columns.

Part II

Connecting Oracle Analytics Cloud to Your Data

This part describes how to set up the connections to your data.

Chapters:

2

Connect to On-premises Data Sources

You can connect to remote data sources (such as on-premises data sources) from Oracle Analytics Cloud through a private access channel or Data Gateway.

Topics:

Overview to Connecting to On-premises Data Sources

You can connect to remote on-premises data sources from Oracle Analytics Cloud. This enables you to deploy Oracle Analytics Cloud with large on-premises datasets without migrating the data to the cloud. Users can analyze the data in data visualizations, and in reporting dashboards and analyses.

 LiveLabs Sprint

You can connect to remote on-premises data sources over a *private access channel* or use *Data Gateway*. In most cases, using a private access channel is better than using Data Gateway because it provides direct and secure connectivity without having to install agents in-between. While a private access channel offers you on-going simplicity and better performance, it does require a Virtual Private Network (VPN) or some other direct network connectivity between Oracle Cloud and your data center, which is not a requirement for Data Gateway.

Before you choose your preferred approach, use Oracle Analytics Cloud's supported data source matrix to check whether you can use a *private access channel* or *Remote Data Connectivity* to connect to your on-premises data source. See List of Supported Databases in Oracle Analytics Cloud.

To find out how to set up a private access channel or Data Gateway, see:

- Connect to On-premises Data Sources Over a Private Access Channel
- Connect to On-premises Data Sources Using Data Gateway

Connect to On-premises Data Sources Over a Private Access Channel

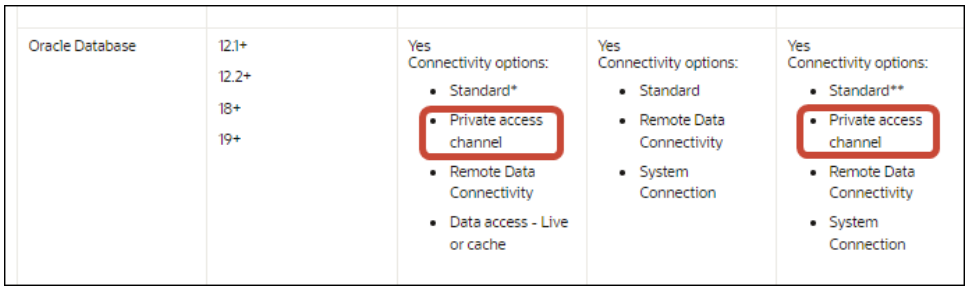
A private access channel enables a direct connection between Oracle Analytics Cloud and your private data sources.

Private access channels enable you to connect to private *data source* hosts. You can't use a private access channel to access any other type of private host. For example, you can't use private access channels to access private hosts that represent FTP servers, SMTP servers, printers , MapViewer configuration, or any other type of private host you might you use.

You use Oracle Cloud Infrastructure Console to set up a private access channel for Oracle Analytics Cloud and configure access to your on-premises data sources. See Connect to Private Data Sources Through a Private Access Channel and Top FAQs for Private Data Sources in *Administering Oracle Analytics Cloud on Oracle Cloud Infrastructure (Gen 2)*.

Supported Data Sources on Private Access Channels

To find out which data sources you can connect to using a private access channel, look for data sources with the connectivity option *Private access channel* in List of Supported Databases in Oracle Analytics Cloud.



Oracle Database	12.1+ 12.2+ 18+ 19+	Yes Connectivity options: • Standard* • Private access channel • Remote Data Connectivity • Data access - Live or cache	Yes Connectivity options: • Standard • Remote Data Connectivity • System Connection	Yes Connectivity options: • Standard** • Private access channel • Remote Data Connectivity • System Connection

Connect to On-premises Data Sources Using Data Gateway

You use Data Gateway to connect to remote on-premises data sources from Oracle Analytics Cloud.

Installing Data Gateway Agent

You can install Data Gateway on Linux or Windows platforms. See Install or Upgrade Data Gateway.

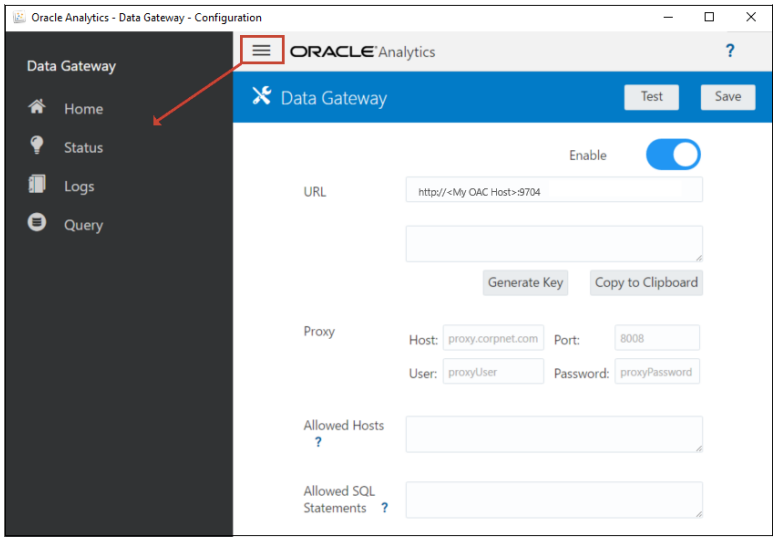
Sizing Data Gateway

Ask your sales account team for guidance on sizing Data Gateway.

Get Started with Data Gateway Agent

Data Gateway Agents enable you to use Oracle Analytics Cloud to visualize and model data in remote databases. You deploy Data Gateway in a subnet that gives visibility to both Oracle Analytics Cloud and the remote databases.

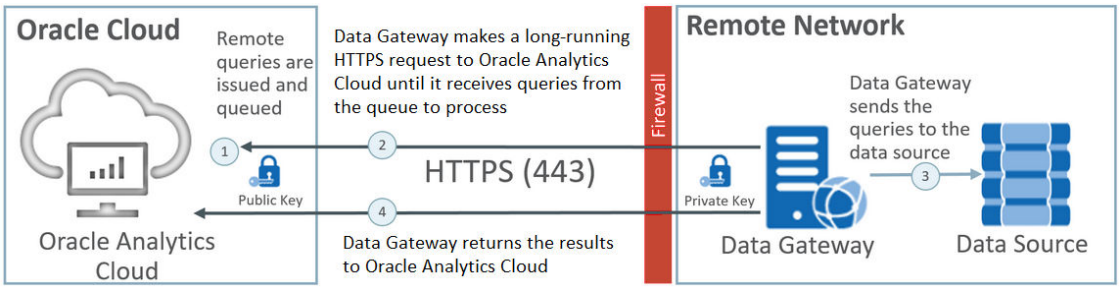
When you start a Data Gateway agent, you see the Home page. Click the **Navigator** to access the other Data Gateway agent pages using the navigator menu.



Navigator Option	Description	Find Out More
Home	Display the Home page, where you can configure the agent, enable or disable the agent, and test the agent's connection.	Configure Data Gateway for Data Visualization Configure and Register Data Gateway for Reporting
Status	Review the status of remote connection database requests between an agent and the remote database.	Diagnosing Connection Issues Using The Status Page
Logs	Display logging information for recent Data Gateway traffic, and enable and disable log polling.	Diagnosing Connection Issues Using The Logs Page
Query	Execute SQL queries to test the connection between the Data Gateway agent and the remote database.	See Diagnosing Connection Issues Using The Query Page.

Data Gateway Architecture

This diagram shows a typical architecture for a deployment of Data Gateway with Oracle Analytics Cloud. You install Data Gateway on a machine in the network where the data source is hosted and configure the Data Gateway agent for communication with your Oracle Analytics Cloud instance.



Data Gateway Functionality

Data Gateway agents poll Oracle Analytics Cloud for queries to run against your remote data sources. The results of these queries are returned to Oracle Analytics Cloud. For secure communication, Data Gateway traffic is signed with an encryption key and each packet is additionally encrypted by Transport Layer Security (TLS) and Secure Sockets Layer (SSL). You can use data from remote data sources in data flows. However, you can't save data to datasets using a remote connection.

Supported Operating Systems for Data Gateway

For a list of supported operating systems, see Download page for Oracle Analytics Cloud.

Supported Data Sources for Data Gateway

Look for databases with "Remote Data Connectivity" listed under Connectivity Options for Datasets, Semantic Modeler, or Model Administration Tool in List of Supported Databases in Oracle Analytics Cloud.

Deploying Multiple Data Gateway Agents

You can deploy multiple Data Gateway agents so that there's no single point of failure. Deploying multiple Data Gateway agents can also improve performance. When you register an agent using the Data Gateway agent Configuration dialog, note the following:

- Configure each agent in the same way.
- Each agent is capable of serving all remote queries. You can't target specific queries at specific agents.
- If you leave the **Allowed Hosts** field blank, the agent tries to reach a data source on any host based on the connection information it retrieves from a connection in Oracle Analytics Cloud. If you specify hosts in the **Allowed Hosts** field, the agent can only address those hosts specified.

Typical Workflow to Connecting to On-premises Data Sources with Data Gateway

Here are the common tasks for connecting to on-premises data sources with Data Gateway.

You can review frequently asked questions before you start. See Frequently Asked Questions about Data Gateway.

Task	Description	More Information
Perform the prerequisite tasks	Download Data Gateway and optionally download Model Administration Tool.	Before You Start with Data Gateway
Install Data Gateway	Install a Data Gateway agent on a local machine.	Install or Upgrade Data Gateway
Upgrade Data Gateway	To upgrade an earlier server installation of Data Gateway on Linux, install the latest update of Data Gateway into the existing installation folder on each machine where you've deployed Data Gateway.	Install or Upgrade Data Gateway
Configure remote data connectivity	Configure your on-premises environment and register one or more Data Gateway agents.	Configure Data Gateway for Data Visualization

Task	Description	More Information
Configure remote connectivity for reporting	(Optional) Perform additional configuration to enable remote connection from dashboards and analyses.	Configure and Register Data Gateway for Reporting
Test Data Gateway	Test your deployment by analyzing data in your on-premises database.	To connect from BI Analytics, see Connect to an On-premises Database from Oracle Analytics Cloud To connect from Publisher, see Set Up a JDBC Connection to an On-premises Data Source.
Monitor Data Gateway	Use the Status page in Data Gateway to monitor the jobs that Data Gateway uses to fetch remote data.	Diagnosing Connection Issues Using The Status Page
Manage Data Gateway	Review installation details, adjust logging levels, or uninstall Data Gateway.	Maintain Data Gateway
Manage Data Gateway agents	Add agents to improve performance or provide a backup, check the status of agents, and check for remote connection issues.	Manage Data Gateway Agents

Before You Start with Data Gateway

Download and install the required software.

- Download the latest update of Oracle Analytics Cloud Data Gateway from Oracle Technology Network. For a list of supported operating systems, see Download page for Oracle Analytics Cloud.
- To upgrade an earlier server installation of Data Gateway on Linux, install the latest update of Data Gateway into the existing installation folder on each machine where you've deployed Data Gateway. See Install or Upgrade Data Gateway.
- (Optional) If you're creating remote connections for analyses and dashboards, download and install Oracle Analytics Client Tools on a Windows machine to get the latest Model Administration Tool for Oracle Analytics Cloud from Oracle Technology Network. If you have a visualization-only deployment (for example, Oracle Analytics Cloud Professional Edition), you don't need Model Administration Tool.
- If you're deploying Data Gateway on Linux, unless you're performing a silent installation, make sure that you have an X server set up with the correct DISPLAY variable setting.

Download Data Gateway

Download Data Gateway from Oracle Technology Network (OTN) to a Linux or Windows machine where you want to install Data Gateway.

To find out the supported versions of operating systems, refer to the OTN download page.

1. Navigate to the OTN download page for Oracle Analytics Cloud.

   See Download page for Oracle Analytics Cloud.

2. Under Oracle Data Gateway <*Month Year*> Update, click **Oracle Analytics Cloud Data Gateway <Month Year> Update Self-contained Installer for Linux and Windows** to display the Oracle Software Delivery Cloud page.

3. Click the **Platforms** down arrow and select the platform(s) onto which you're deploying Data Gateway, then click outside of the drop-down list or press Enter.

   The ZIP files available for each Platform are selected by default.

4. If you selected "All" or "Microsoft Windows x64" in the **Platforms** option, de-select any components that you don't want to download.

   For example, you might de-select Oracle Analytics Power BI Connector.

5. Accept the Oracle Cloud Service License Agreement.
6. Click **Download** to start Oracle Download Manager, and follow the on-screen instructions.
7. When the download is complete, click **Open Destination**.
8. Extract the Oracle installer from the downloaded ZIP file.

   For example, for Linux, extract `DataGateway_<update>Linux64.bin`, or for Windows extract `DataGateway_<update>Windows64.exe`.

Download and Install Oracle Analytics Client Tools

Download and install Oracle Analytics Client Tools to enable remote connections from reporting dashboards and analyses. In addition, you might use Administration Tool (one of the client tools available for Microsoft Windows) to edit a semantic model (.rpd file) that isn't supported by Semantic Modeler.

You install Oracle Analytics Client Tools on Windows or Linux platforms.

- On Windows, the software pack installs the graphical user-interface version of Administration Tool , as well as command line utilities such as runcat.cmd (for catalog management).
- On Linux, the software pack installs `runcat.sh` and `datamodel.sh` command line utilities.

> **Note:**
>
> Oracle updates Oracle Analytics Client Tools with each Oracle Analytics Cloud update. Make sure that you're using the latest update of Oracle Analytics Client Tools.

1. Navigate to the download page for Oracle Analytics Client Tools.
2. Click the latest **Oracle Analytics Client Tools <Month Year> Update** link to display the Oracle Software Delivery Cloud page.
3. Click the **Platforms** down arrow, click **All**, then click outside of the drop-down list or press Enter.
4. In the Software column of the table, select the download pack for the platform you want.
   - For Windows, select **Oracle Analytics Client May2023-Win for (Microsoft Windows x64 (64-bit)), <Size in MB>**.
   - For Linux, select **Oracle Analytics Client May2023-Linux for (Linux x86-64), <Size in MB>**.

   Make sure that other components are de-selected (for example, Data Gateway and Power BI Connector).
5. Accept the Oracle Cloud Service License Agreement.

6. Click **Download** to start Oracle Download Manager, and follow the on-screen instructions.
7. When the download is complete, click **Open Destination**.
8. Extract and run the Oracle installer from the downloaded ZIP file.
   For example, extract and run the installer file `oac_client-<update ID>-win64.exe`, and follow the on-screen instructions.

   To start the tools on Windows, go to your Windows Start menu, click **Oracle Analytics Client Tools**, and then select the name of the tool you want to use. For example, to edit your semantic model, click **Model Administration Tool**.

   On Linux, use the `runcat.sh` and `datamodel.sh` command line utilities. See Using Oracle Analytics Client Tools on Linux.

Install or Upgrade Data Gateway

Install a Data Gateway agent on a machine in the network where the data source is hosted.

Note: Oracle Analytics no longer supports Data Gateway agents older than the November 2023 version. If you have an earlier version of Data Gateway, upgrade your environment by installing the latest version.

You can install Data Gateway agents interactively or silently using an Oracle Universal Installer response file. To deploy Data Gateway agents on multiple machines, repeat the installation and configuration steps for each machine.

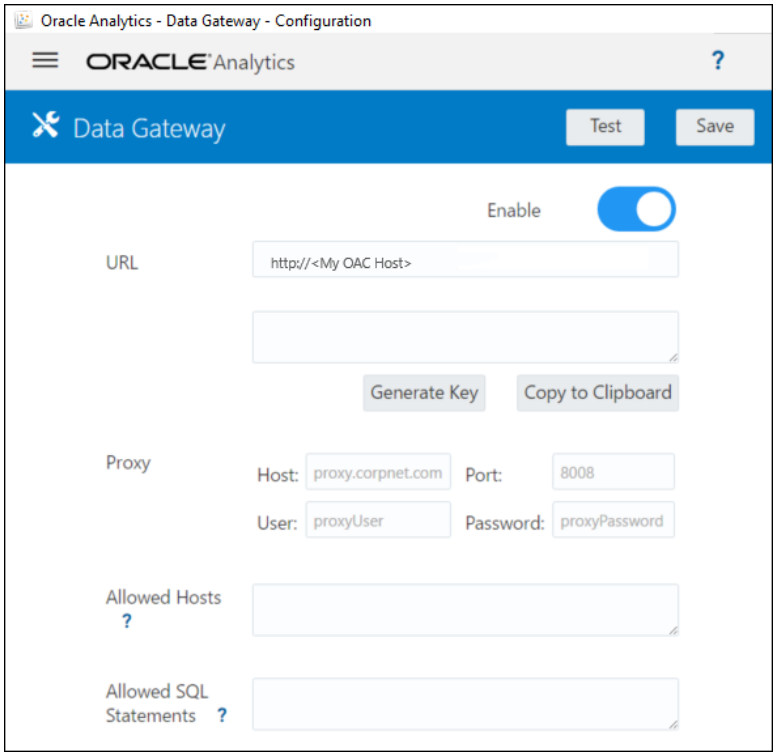
To upgrade an earlier server installation of Data Gateway on Linux, install the new version of Data Gateway into the existing installation folder. If you have an existing personal installation of Data Gateway on Windows, delete the installation and re-install using the instructions below.

1. Download Data Gateway from Oracle Technology Network (see Download Data Gateway).
2. Start the Data Gateway installer and follow the on-screen instructions.

   On Linux, run `DataGateway_<update>Linux64.bin` (before you start, make the installer file executable; for example, using chmod 777).

   On Windows run `DataGateway_<update>Windows64.exe` (as administrator).
3. On the Installation Complete page, under **Next Steps**, select **Start Jetty**.
4. To start the Data Gateway agent, open a web browser and enter the URL: `<Local hostname>:<port>/obiee/config.jsp`.

For example, on Windows, enter the URL `http://localhost:8080/obiee/config.jsp`.



5. If you're using a proxy, navigate to the Home page in the Data Gateway agent, and specify the **Proxy** details for **Host**, **Port**, **User**, and **Password**.

When the installation is complete, configure Data Gateway to communicate with your Oracle Analytics Cloud. instance. See Configure Data Gateway for Data Visualization or Configure and Register Data Gateway for Reporting.

Configure Data Gateway for Data Visualization

After you've installed Data Gateway, you configure your on-premises environment and register one or more Data Gateway agents for remote connectivity from visualization workbooks.

To deploy multiple Data Gateway agents, repeat steps 4 to 9 for each agent.

1. Log into Oracle Analytics Cloud as an administrator.
2. Copy your Oracle Analytics Cloud URL:
   a. In a browser, navigate to the home page of your Oracle Analytics Cloud instance.

      Use the same URL that end users use to connect to Oracle Analytics Cloud.

   b. In the browser's address bar, copy the URL up to and including <*domain*> (not the text after that).

      For example, if the URL is `https://oacinst-tenancy.analytics.ocp.oraclecloud.com/dv/ui`, then copy `https://oacinst-tenancy.analytics.ocp.oraclecloud.com`.

      You'll use this URL in Step 4 when you set up Data Gateway in Agent Configuration.
3. Enable Data Gateway in Console:
   a. From the Oracle Analytics Cloud Home page, click **Console**.
   b. Click **Remote Data Connectivity**.

c. Enable the **Enable Data Gateway** option.

   Keep this browser page open as you complete the following steps.

4. On the Data Gateway installation machine, start the Jetty server if it isn't already started.

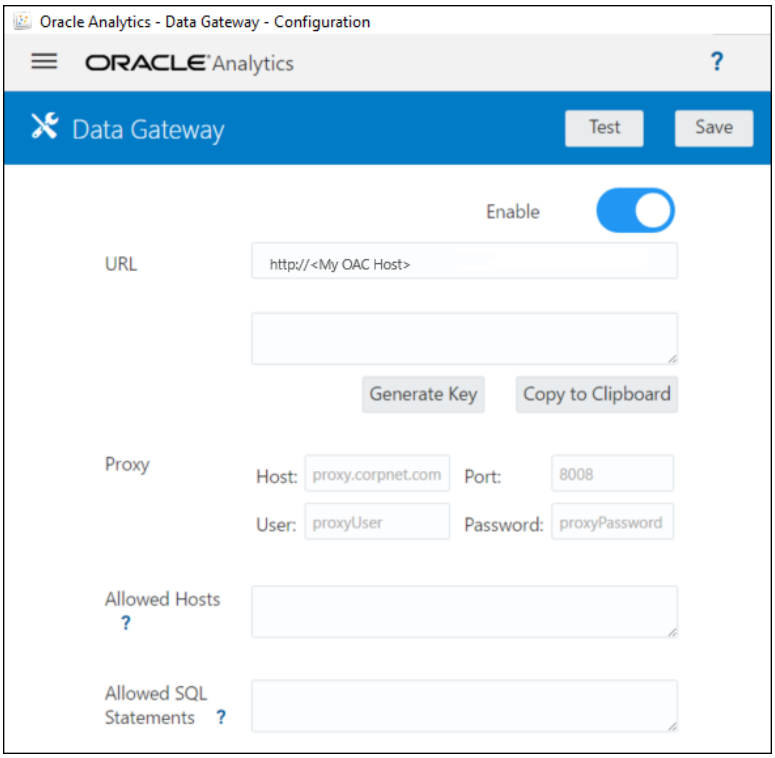
   For example, at the end of the installation, you might not have clicked **Start Jetty** under **Next Steps** on the Installation Complete page, or you might have rebooted the machine since the installation. See Start and Stop a Data Gateway Agent.

5. For each Data Gateway agent, use the Data Gateway agent Home page to generate an authorization key for that machine:

   If you're asked to log in, enter the same username and password that you specified in the Credentials page of the Data Gateway installer.

   a. To start the Data Gateway agent, open a web browser and enter the URL: `<Local hostname>:<port>/obiee/config.jsp` to display the Data Gateway agent Home page.

      For example, on Windows, you might enter the URL `http://localhost:8080/obiee/config.jsp`.



   b. In the **URL** field, enter the Oracle Analytics Cloud URL that you copied in Step 2.

   c. Click **Generate Key**, then click **Copy to Clipboard**.

      Leave the other fields blank.

> **Note:**
>
> Don't click **Test**, **Save**, or **Enable** yet.

6. Switch to the browser session where you have the Oracle Analytics Cloud Console page **Remote Data Connectivity** displayed, and add details of each Date Gateway agent that you've deployed.

a. Under **Data Gateway**, click **Add**.

b. In **Public Key**, paste in the key that you copied using the **Copy to Clipboard** option in Step 4.c.

   When you paste in the key, the **Name**, **ID**, and **Host** fields are completed with the details of the on-premises machine where you've installed Data Gateway.

c. Click **OK** to save the details.

7. Switch to the Data Gateway agent Home page.

8. Optional: Optionally use the **Allowed Hosts** field to restrict Data Gateway access to specific host machines. Leave the field blank to enable Data Gateway to access any host machine.

   You can specify host names and IP addresses with asterisk (*) wildcards, separated with a semi-colon.

   For example, `abcd*.example.com; 10.174.*.`

   By default the Data Gateway agent attempts to connect to a data source on any host specified in a remote connection in Oracle Analytics Cloud. The **Allowed Hosts** field allows you to constrain the target hosts and IP addresses that Data Gateway can connect to. However, you must configure Data Gateway so that all agents can serve all remote queries.

9. Optional: Optionally use the **Allowed SQL Statements** field to restrict Data Gateway to specific SQL or data manipulation language (DML) constructs. Leave the field blank to enable Data Gateway to execute any SQL statements or DML constructs on the data source.

   For example, specify `SELECT` to restrict Data Gateway to read-only access to the remote data source. Or specify `SELECT; ALTER SESSION` to restrict Data Gateway to using SELECT and ALTER SESSION operations.

   Make sure that the SQL in any semantic model connect scripts (or anywhere else) doesn't contain trailing whitespace or control characters (EOL - End of line, or CR - Carriage Return).

10. Click **Test**, **Save**, then **Enable**.

    To deploy multiple Data Gateway agents, repeat steps 4 to 9 for each agent.

    If the test fails, this means that the Data Gateway agent can't authenticate. Possible reasons include:

    - The agent key hasn't been copied to the **Remote Data Connectivity** page in Oracle Analytics Cloud Console.
    - The agent key has been regenerated in the agent, but the new key hasn't been copied to the **Remote Data Connectivity** page in Oracle Analytics Cloud Console.
    - There's no suitable network route from the agent to Oracle Analytics Cloud.

If you also want to connect remotely from reporting dashboards and analyses, perform the additional configuration steps in Configure and Register Data Gateway for Reporting.

Then you're ready to test your deployment by connecting remotely to an on-premises database.

Configure and Register Data Gateway for Reporting

Perform these optional steps to enable remote connectivity for Classic features such as analyses and dashboards.

If you're only deploying for data visualization (for example, Oracle Analytics Cloud Professional Edition), then you don't need to follow these steps.

Before you start, follow the configuration steps in Configure Data Gateway for Data Visualization.

1. On the machine where you've installed a Data Gateway agent, obtain the machine name and port number.

   In a server deployment:

   a. Execute the command `/<Data Gateway install folder>/domain/bin/status.sh`.

   b. In the command output, note the machine name contained within the **URL** displayed under **Data Gateway Status**, and note the **Data Gateway Jetty HTTP Port** value.

   In a personal deployment:

   a. Open the file: `%localappdata%\Temp\DataGateway\ports.properties`.

   b. Note the machine name and port number.

2. Start the Data Gateway agent.

3. If you want to model your data before you start connecting to it remotely, use Semantic Modeler or Model Administration Tool (if the database isn't supported by Semantic Modeler) to edit your semantic model.

4. If you're using Model Administration Tool, load the Java data source metadata.

   a. In Model Administration Tool, from the **File** menu, click **Open**, then **In the Cloud**, and use the Open in the Cloud dialog to specify the details of your semantic model.

   b. From the **File** menu, click **Load Java Data Sources**.

   c. At the Connect to Java Datasource Server dialog:

      - In the **Hostname** field, enter the machine name that you noted in Step 1. Fully qualify the hostname. For example, if you noted `machine` in Step 1, you might specify `machine.us.example.com`.
      - In the **Port** field, enter the port that you noted in Step 1. For example, `51811`.
      - In the **Username** and **Password** fields, enter `dummy` or any string (these credentials aren't validated because this is a public call to discover the capabilities advertised by the Data Gateway).

5. If you're using Model Administration Tool, set up a physical database connection:

   a. In the Physical layer, create a local (not remote) connection to your data source using the standard call interface appropriate for your data source, and model the data as required.

   b. When you're ready to make a remote connection to your semantic model and publish it back to the cloud, edit the connection that you created.

   c. On the General tab, in the **Call interface** field select `JDBC (Direct Driver)`, and in the **Connection String** field, specify the JDBC string and credentials in the semantic model connection. See **JDBC and JNDI Templates and Examples** below for a list of supported JDBC strings and driver classes.

d. On the Miscellaneous tab, in the **Use SQL Over HTTP** field enter true, and in the **RDC Version** field enter **2**, and specify the JDBC driver class.

e. Publish the semantic model to the cloud.

You're now ready to test your deployment by connecting remotely to an on-premises database.

JDBC and JNDI Templates and Examples

When you set up remote connectivity for analyses and dashboards, you might have to specify JDBC strings and driver classes, and JNDI connection details and context details.

JDBC String Patterns and Driver Classes

```
Oracle:
   Driver Class: oracle.jdbc.OracleDriver
   jdbc string: jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)
(HOST=[\"host-name\"])(PORT=[\"port\"]))
(CONNECT_DATA=(SERVICE_NAME=[\"service-name\"])))
Amazon Redshift:
   Driver Class: com.oracle.jdbc.redshift.RedshiftDriver
   JDBC String: jdbc:oracle:redshift://[\"host-name\"]:
[\"port\"];DatabaseName=[\"service-
name\"];EncryptionMethod=SSL;ValidateServerCertificate=false
Apache Hive
    Driver Class: com.oracle.bi.jdbc.hive.HiveDriver
    JDBC String: jdbc:oracle:hive://[\"host-name\"]:
[\"port\"];EncryptionMethod=SSL;ValidateServerCertificate=false
DB2
   Driver Class: com.oracle.bi.jdbc.db2.DB2Driver
   JDBC String: jdbc:oracle:db2://[\"host-name\"]:
[\"port\"];DatabaseName=[\"service-name\"]
Impala
   Driver Class: com.oracle.bi.jdbc.impala.ImpalaDriver
   JDBC String: jdbc:oracle:impala://[\"host-name\"]:
[\"port\"];EncryptionMethod=SSL;ValidateServerCertificate=false
MySQL
   Driver Class: com.mysql.cj.jdbc.Driver
   JDBC String: jdbc:mysql://[\"host-name\"]:[\"port\"]"[/database][?
properties]
SQL Server
   Driver Class: com.oracle.bi.jdbc.sqlserver.SQLServerDriver
   JDBC String: jdbc:oracle:sqlserver://[\"host-name\"]:
[\"port\"];DatabaseName=[\"service-name\"]
Teradata
   Driver Class: com.teradata.jdbc.TeraDriver
   JDBC String: jdbc:teradata://[\"host-name\"]/DBS_PORT=[\"port\"]
```

JNDI Templates for Native Drivers

```
Oracle:
      <Resource
      name="jdbc/myoracle"
      global="jdbc/myoracle"
      auth="Container"
      type="javax.sql.DataSource"
```

```
      driverClassName="oracle.jdbc.OracleDriver"
      url="jdbc:oracle:thin:@localhost:1521:orcl"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1"
      />

      <Resource
      name="jdbc/oracleolap"
      global="jdbc/oracleolap"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="oracle.jdbc.OracleDriver"
      url="jdbc:oracle:thin:@localhost:1522:orcl112"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1"
      />

      <Resource
      name="jdbc/oraclenorthwind"
      global="jdbc/oraclenorthwind"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="oracle.jdbc.OracleDriver"
      url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=example.com)(PORT=1234))
(CONNECT_DATA=(SERVICE_NAME=MATSDB.EXMAPLE.COM)))"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1"
      />

      DB2
      <Resource
      name="jdbc/db2northdb"
      global="jdbc/db2northdb"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="com.ibm.db2.jcc.DB2Driver"
      url="jdbc:db2://example.com:58263/NORTHDB"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1" />

      SQLServer:
      <Resource
      name="jdbc/sqlservernorthwind"
```

```
      global="jdbc/sqlservernorthwind"
      auth="Container"
      type="javax.sql.DataSource"

driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
      url="jdbc:sqlserver://
example.com:61045;DatabaseName=Northwind"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1" />

      Teradata:
      <Resource
      name="jdbc/teranorthwind"
      global="jdbc/teranorthwind"
      auth="Container"
      type="javax.sql.DataSource"
      driverClassName="com.teradata.jdbc.TeraDriver"
      url="jdbc:teradata://99.999.99.999"
      username="my_user"
      password="my_password"
      maxActive="15"
      maxIdle="1"
      maxWait="-1" />

      Mysql_community
      <Resource
      name="jdbc/CEmysql"
      auth="Container"
      type="com.mysql.jdbc.jdbc2.optional.MysqlDataSource"
      factory="com.mysql.jdbc.jdbc2.optional.MysqlDataSourceFactory"
      username="my_user"
      password="my_password"
      serverName="example.com"
      portNumber="3306"
      databaseName="my_database" />
```

JNDI Templates for DD Drivers

```
The JNDI for DD Drivers.
SQLServer:
<Resource
name="jdbc/DDsqlserver"
auth="Container"
type="com.oracle.bi.jdbcx.sqlserver.SQLServerDataSource"
factory="com.oracle.bi.jdbcx.sqlserver.SQLServerDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com\MSSQLSERVER16"
portNumber="61045"
databaseName="my_database" />

DB2:
```

```
<Resource
name="jdbc/DDdb2"
auth="Container"
type="com.oracle.bi.jdbcx.db2.DB2DataSource"
factory="com.oracle.bi.jdbcx.db2.DB2DataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="58263"
databaseName="my_database"
/>

Impala:
<Resource
name="jdbc/DDimpala"
auth="Container"
type="com.oracle.bi.jdbcx.impala.ImpalaDataSource"
factory="com.oracle.bi.jdbcx.impala.ImpalaDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="21050"
databaseName="my_database"
/>

Spark:
<Resource
name="jdbc/DDspark"
auth="Container"
type="com.oracle.bi.jdbcx.sparksql.SparkSQLDataSource"
factory="com.oracle.bi.jdbcx.sparksql.SparkSQLDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="10000"
databaseName="my_database"
/>

HIVE:
<Resource
name="jdbc/DDhive"
auth="Container"
type="com.oracle.bi.jdbcx.hive.HiveDataSource"
factory="com.oracle.bi.jdbcx.hive.HiveDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="10000"
databaseName="my_database"
/>

MySQL:
<Resource
name="jdbc/DDmysql"
auth="Container"
type="com.oracle.bi.jdbcx.mysql.MySQLDataSource"
```

```
factory="com.oracle.bi.jdbcx.mysql.MySQLDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="3306"
databaseName="my_database"
/>

MYSQL:
<Resource
name="jdbc/DDmysql"
auth="Container"
type="com.oracle.bi.jdbcx.mysql.MySQLDataSource"
factory="com.oracle.bi.jdbcx.mysql.MySQLDataSourceFactory"
user="my_user"
password="my_password"
serverName="example.com"
portNumber="3306"
databaseName="my_database"
/>
```

Add a JDBC Driver to Data Gateway

Add a JDBC driver to your Data Gateway installation so that you can model data in an on-premises database.

Before you start, ensure that Data Gateway and Model Administration Tool are installed on the same Windows computer in your on-premises environment.

1. Download the JDBC driver that you want to deploy.

   For example, to model data in a Snowflake database, download the latest Snowflake JDBC driver (for example, in file `snowflake-jdbc-3.9.0.jar`).

2. Copy the downloaded JDBC JAR file to the Data Gateway installation folder.
   - In a server deployment, copy the JAR file into: `<Data Gateway install_location>/domain/jettybase/thirdpartyDrivers`.
   - In a personal deployment on Windows, copy the JAR file into: `<Data Gateway_extract_path>\thirdpartyDrivers`.
   - In a personal deployment on MacOS, copy the JAR file into: `<Application->Show Package Contents>Resources->app.nw-> thirdpartyDrivers`.
3. Re-start Data Gateway. See Maintain Data Gateway.

DSN Formats for Specifying Data Sources

In Oracle Analytics you can model your on-premises data for many database types. Oracle Analytics supports direct access to some on-premises data sources through the semantic model. When you create the database connection using Model Administration Tool, in the **Data source name** field on the Connection Pool dialog (General tab) you use the appropriate DSN format for the database type you're connecting to.

```
Amazon Redshift:
       DRIVER=Oracle 7.1 Amazon Redshift Wire Protocol;HOST=["host-
```

```
name"];PORT=["port"];DB=["service-name"]
      SSL: DRIVER=Oracle 7.1 Amazon Redshift Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"];EM=6;CPV=TLSv1.2,TLSv1.1,TLSv1,
SSLv3, SSLv2;VSC=0
Apache Drill:
       DRIVER=MapR Drill ODBC Driver;Host=["host-
name"];Port=["port"];CastAnyToVarchar=true;ExcludedSchemas=sys,INFORMATION_SCH
EMA;AuthenticationType=Basic
Authentication;ConnectionType=Direct
Aster:
       DRIVER=Aster ODBC Driver;SERVER=["host-
name"];PORT=["port"];DATABASE=["service-name"]
DB2:
      DRIVER=Oracle 7.1 DB2 Wire Protocol;IpAddress=["host-
name"];PORT=["port"];DB=["service-name"]
      SSL:  DRIVER=Oracle 7.1 DB2 Wire Protocol;IpAddress=["host-
name"];PORT=["port"];DB=["service-name"];EM=1;VSC=0
Greenplum:
       DRIVER=Oracle 7.1 Greenplum Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
Hive:
       DRIVER=Oracle 8.0 Apache Hive Wire Protocol;HOST=["host-
name"];PORT=["port"]
      SSL: DRIVER=Oracle 8.0 Apache Hive Wire Protocol;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
Impala:
       DRIVER=Oracle 7.1 Impala Wire Protocol;HOST=["host-name"];PORT=["port"]
     SSL:  DRIVER=Oracle 7.1 Impala Wire Protocol;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
Informix:
       DRIVER=Oracle 7.1 Informix Wire Protocol;HOSTNAME=["host-
name"];PORTNUMBER=["port"];DATABASE=["service-name"]
MongoDB:
       DRIVER=Oracle 8.0 MongoDB;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
MySQL:
      DRIVER=Oracle 7.1 MySQL Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
PostgresSql:
       DRIVER=Oracle 7.1 PostgreSQL Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
Spark:
      DRIVER=Oracle 8.0 Apache Spark SQL;HOST=["host-name"];PORT=["port"]
      SSL:  DRIVER=Oracle 8.0 Apache Spark SQL;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
SQL Server:
     DRIVER=Oracle 7.1 SQL Server Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
     SSL:  DRIVER=Oracle 7.1 SQL Server Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-
name"];EM=1;VSC=0;CryptoProtocolVersion=TLSv1.2,TLSv1.1,TLSv1,SSLv3,SSLv2
Sybase:
       DRIVER=Oracle 7.1 Sybase Wire Protocol;NA=["host-name"],
["port"];DB=["service-name"]
Teradata:
       DRIVER=Oracle 7.1 Teradata;DBCName=["host-name"];port_name=["port"]
```

Connect to an On-premises Database from Oracle Analytics Cloud

After you've installed and deployed Data Gateway, you can start analyzing data in your on-premises database.

If you don't see a **Use Remote Data Connectivity** option on the Create Connection dialog, ask the Oracle Analytics administrator to enable one of the remote connectivity options in Console on the Remote Data Connectivity page.

1. Create a connection to your on-premises database:
   a. From the Home page, click **Create**, then **Connection**.
   b. Click a connection type that supports remote connectivity. For example, you want to connect remotely to an Oracle database.
   c. Use the Create Connection dialog to specify the connection details of your on-premises database.

      For example, for an on-premises Oracle Database, specify the Host, Port, Service Name, and credentials.
   d. Enable the **Use Remote Data Connectivity** option.
2. Create a workbook using the connection that you created in the Step 1.
   a. At the Oracle Analytics home page, click **Create**, then click **Workbook**.
   b. At the Add Dataset dialog, select the on-premises database and add columns from it to a visualization.
3. If you've also configured remote connection for reporting, in Classic Home, create an analysis based on the connection that you created in Step 1.
   a. At the Oracle Analytics home page, from the **Page menu**, then click **Open Classic Home**.
   b. From the top toolbar, click **Create**, then click Analysis. In the Select Subject Area drop-down list, select a subject area in the on-premises database and add columns from it to a visualization.

Maintain Data Gateway

Administrators maintain Data Gateway agents using these tasks. If you have multiple Data Gateway agents deployed, then repeat the task for each agent.

Maintenance Tasks for Data Gateway

Task	More Information
Manage one or more agents	See Manage Data Gateway Agents.

Task	More Information

Find out the Data Gateway agent URL and port.	Use the `DOMAIN_HOME/bin/status.sh` script to display the installation status and connection details. For example:
	
	`Data Gateway Jetty Home: <Jetty home>`
	`Data Gateway Domain Home: <Domain home>`
	`Data Gateway Jetty HTTP Port: <Port>`
	`Data Gateway Status: <Data Gateway status> (For example, UP.)`
	`URL: <URL for Data Gateway Agent Configuration page> (For example, http://example.com:8080/obiee/config.jsp.)`

Start and stop a Data Gateway agent.	See Start and Stop a Data Gateway Agent.
Change how much logging information is recorded for a Data Gateway agent.	See Adjust the Data Gateway Logging Level.
Upgrade or patch a Data Gateway agent	To upgrade an existing Data Gateway agent, install the new version of Data Gateway into the existing installation folder on each machine where you've deployed Data Gateway. See Install or Upgrade Data Gateway.
Review the auditing and diagnostics information that a Data Gateway agent has logged.	On the machine where you installed the Data Gateway agent, review the files in `/domain/jettybase/logs`.
Remove a Data Gateway agent from a machine.	Delete the Data Gateway installation folder.

Start and Stop a Data Gateway Agent

Start a Data Gateway agent so that you can connect a remote on-premises data source to Oracle Analytics Cloud.

On the machine where the Data Gateway agent is installed:

1. Start the Jetty server.
   On Linux, run the script `domain/bin/startjetty.sh`.

   On Windows, run the script `domain\bin\startjetty.cmd`.
2. To stop a Data Gateway agent, run the script `domain/bin/stopJetty.sh` or `domain/bin/stopJetty.cmd`.
3. To restart a Data Gateway agent, run the script `stopJetty` followed by `startjetty`.

Adjust the Data Gateway Logging Level

Increase or decrease the amount of logging information recorded by Data Gateway.

In a Server Deployment

1. Stop the Jetty server using `domain/bin/stopJetty.sh`.

2. In the folder `jetty/modules/log4j2-impl/resources/`, edit the file `log4j2.xml`.
3. In the file `log4j2.xml`, make these changes:
   - Line no. 2 - Change the configuration status to `debug ---> <Configuration status="debug" name="Jetty" >`
   - Line no. 7 - Change the root level to `debug --> <Root level="debug">`
   - Line no. 34 -Change the root level to `debug --> <Root level="debug">`
4. In the `startJetty.sh` file located in the domain/bin folder, add a property `-Dlog4j.configurationFile="<Full Path of the log4j2.xml>"` as shown.

   ```
   java -DSTOP.PORT=34954 -DSTOP.KEY=stop_jetty -DDOMAIN_HOME=$DOMAIN_HOME -
   DPUBLIC_KEY_FOLDER=/scratch/sunraj/Oracle/Middleware/Oracle_Home_RDG/
   domain/r dc_keys -DRDC_VERSION=V2 -Djetty.home=$JETTY_HOME
   Djetty.base=$JETTY_BASE -Djetty.http.port=8080 -Djetty.ssl.port=8443 -
   Dlog4j.configurationFile="<Full Path of the log4j2.xml>" -jar start.jar
   ```

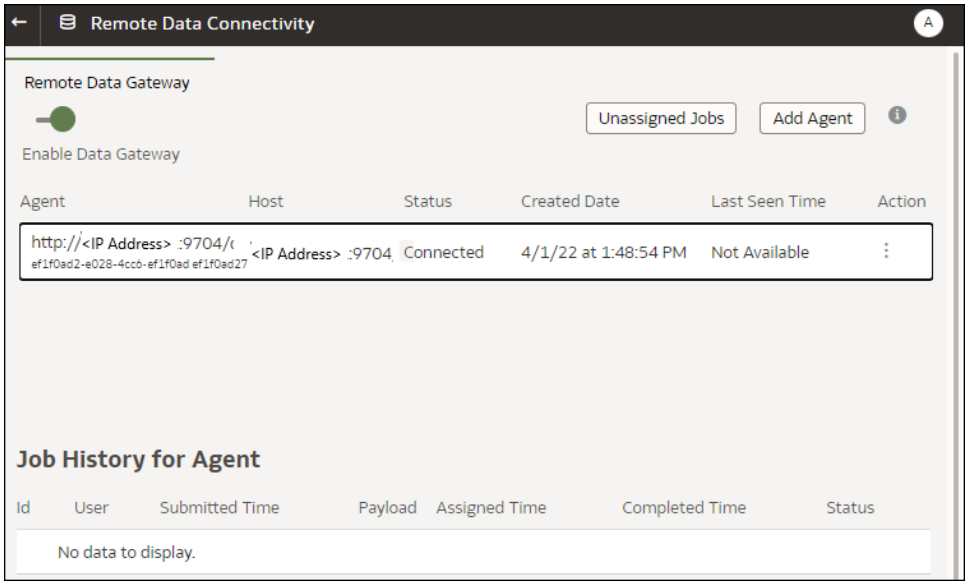
   For example, if the `log4j2.xml` path is `/scratch/user/Oracle/Middleware/Oracle_Home_RDG/jetty/modules/log4j2-impl/resources/log4j2.xml`, the format is `Dlog4j.configurationFile="/scratch/user/Oracle/Middleware/Oracle_Home_RDG/jetty/modules/log4j2-impl/resources/log4j2.xml"`
5. Start the Jetty server using `domain/bin/startJetty.sh`.

Manage Data Gateway Agents

Use the Console to manage Data Gateway agents. For example, you might add agents to improve performance or provide a backup, check the status of agents, and check for remote connection issues.

Data Gateway Agents enable you to use Oracle Analytics Cloud to visualize and model data in remote databases. You deploy Data Gateway in a subnet that gives visibility to both Oracle Analytics Cloud and the remote databases.

1. On the Home page, click **Navigator** then **Console**, then **Remote Data Connectivity**.



2. Use the Remote Data Connectivity page to manage agents.
   - To enable remote connection traffic between your Oracle Analytics Cloud instance and remote databases, enable the **Enable Data Gateway** option.
   - To enable or disable a specific agent, click the **Action** option for the agent (⋮), and select either **Enable Agent** or **Disable Agent**. If you disable an agent, in-progress jobs for the disabled agent are halted, and future connection jobs for the instance are automatically reassigned to other agents, if available.
   - To add an agent, click **Add Agent**. See Configure Data Gateway for Data Visualization.
   - To view remote connection traffic handled by an agent, select the agent to display a **Job History** list.
   - To check for remote queries that haven't been processed or assigned to an agent, click **Unassigned Jobs**.

3

Connect to Data

As an Oracle Analytics user with DV Content Author access, you can connect to the data sources used by your organization.

Topics

Manage Connections to Data Sources

You can create, update, remove, and share connections to data sources. As an Oracle Analytics user with DV Content Author access, you can perform these actions.

Topics:

- Share a Data Source Connection
- Database Connection Options
- Connect to Data With Upper, Lower, or Mixed-case Characters

Create a Connection to a Data Source

You can create a connection to enable you to analyze data in that data source.

1. On the Home page, click **Create**, and then click **Connection**.
2. In the Select Connection Type dialog box, click the icon for the connection type that you want. For example, **Oracle Database**.
3. Enter the required connection information such as host, port, username, password, and service name.
4. Optional: Select an **Authentication** option for your connection..
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.
5. If you're connecting to a remote database, click **Use Remote Data Connectivity**.

   Check with your administrator that you can access the remote database.
6. If you want to use these connection details in Semantic Modeler or Model Administration Tool, click **System connection**. See Database Connection Options.
7. Click **Save**.

   You can now begin creating workbooks or datasets using this connection. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Edit a Data Source Connection

You can update a data source's connection details.

If you're editing an SSL connection to an Oracle Database and you need to use a new `cwallet.sso` file, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.

1. On the Home page click the Navigator, select **Data**, and then click **Connections**.
2. Hover over the connection that you want to edit. To the right of the highlighted connection, click **Actions**, then select **Inspect**.
3. In the Inspect dialog, edit the connection details.

   You can't see the current password or Logical SQL for connections. If you need to change these, create a new connection.
4. Click **Save**.

Delete a Data Source Connection

You can remove a data source connection from Oracle Analytics Cloud. For example, you must delete a database connection and create a new connection when the database's password has changed.

If the connection contains any datasets, then you must delete the datasets before you can delete the connection.

1. Go to the Data page and select **Connections**.
2. Hover over the connection that you want to delete. To the right of the highlighted connection, click **Actions**, then select **Delete**.
3. Click **Yes**.

Share a Data Source Connection

You can assign access permissions to the data source connections that you create or administer.

1. On the Home page click the **Navigator**. Click **Data**, and then click **Connections**.
2. Hover over the connection that you want to share, click **Actions**, then select **Inspect**.
3. Click **Access**, and use the tabs to grant access:
   - **All** - Share the connection with individual users or roles.
   - **Users** - Share the connection with individual users.
   - **Roles** - Share the connection with application roles (for example, BI Consumer), so that all users with those roles can use the connection.
4. Use the **Add** box to search for and select a user or role.

   The user or role is displayed in the list below with the default privileges **Read-Only**.
5. To change the default privileges, select one of the following:
   - **Full Control** - The user or role can use the connection to create datasets, and modify, rename, or delete the connection. They can also change the privileges for the connection.
   - **Read-Write** - The user or role can use the connection to create datasets, and modify or rename the connection (but not delete it).
   - **Read-Only** - The user or role can use the connection to create datasets, but not change the connection details.
6. Click **Save**.

When users next log in, they can use connections that you've shared to visualize data from this database.

Database Connection Options

When you specify connection details using the Create Connection dialog or Inspect dialog, some database types have extra configuration options.

General Options

- When you create connections to Oracle Databases, you can connect in two ways using the **Connection Type** option:
  - **Basic** - Specify the **Host**, **Port**, and **Service Name** of the database.
  - **Advanced** - In the **Connection String** field specify the Single Client Access Name (SCAN) ID of databases running in a RAC cluster. For example:

```
sales.example.com =(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)
(FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.111)(PORT=1521))
(ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.222)(PORT=1521))
(ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.333)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME= salesservice.example.com)))
```

- **Enable Bulk Replication** - If you're loading a dataset for a workbook, then this option should be turned off and you can ignore it. This option is reserved for data analysts and advanced users for replicating data from one database to another database.

Authentication Options

- **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.
- **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
- (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.

System connection

Select **System connection** if you want to use these connection details in Semantic Modeler or Model Administration Tool.

- If you're modeling data using Semantic Modeler, choosing this option enables Semantic Modeler to connect to the data source. To find out which databases Semantic Modeler supports, look for a Yes in the 'Use in Semantic Modeler' column in the supported data sources list. See Supported Data Sources.
- If you're modeling data using Model Administration Tool, you can connect to some data sources by copying connection details from Oracle Analytics Cloud into Model Administration Tool. To find out which data sources support this, look for System Connection in the Connectivity Options in the Use in Model Administration Tool in Supported Data Sources. Click **Copy**, to copy the connection's **Object ID**. Business modelers can paste in the **Object ID** to identify and use a data connection in the Model Administration Tool (Connection Pool dialog).

Click **Copy**, to copy the connection's **Object ID**. Business modelers can paste in the **Object ID** to identify and use a data connection in the Model Administration Tool (Connection Pool dialog).

Note: If you don't click **System connection** when you initially create the connection, if you later want data modelers to use the connection details, then you have to create a new connection and select **System connection**. In other words, you can't edit the connection later and select this option.

Database Connection Limits

Understand your database connection requirements before your create the database connection.

The maximum number of database tables displayed in Oracle Analytics is 10,000. If you need additional tables, Oracle recommends that you ask your database administrator to create a database user with access to the specific objects you want to analyze and specify that user's credentials when you create the database connection.

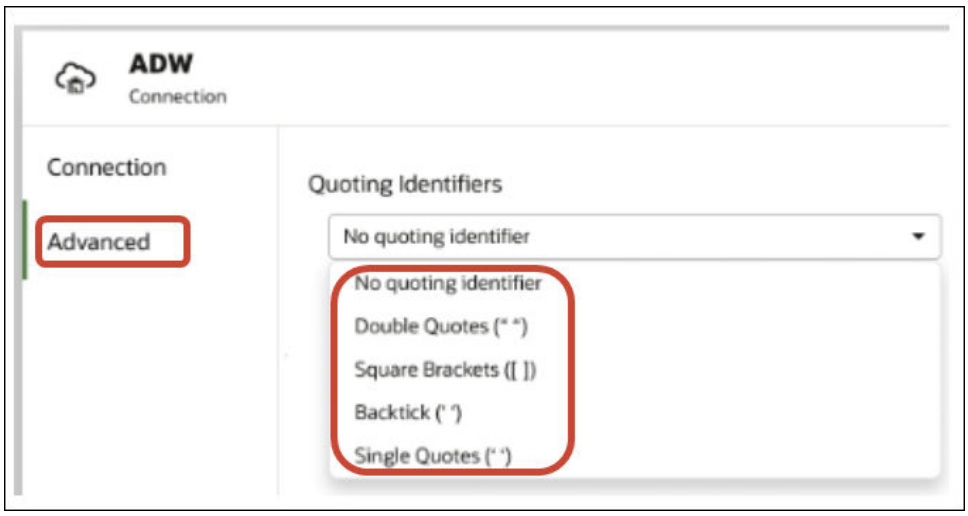
Connect to Data With Upper, Lower, or Mixed-case Characters

If you're connecting to an Oracle database, Oracle Autonomous Data Warehouse, Oracle Transaction Processing, Snowflake, SQL Server, or My SQL, you can change the default quoting identifier so that you can read data with upper, lower, or mixed-case characters in table or column names.

For example, you might choose double quotes as the quoting identifier. Oracle Analytics then adds double quotes to the underlying SQL statement `select "EfG_Field" from "AbCd";` instead of issuing `select EfG_Field from AbCd;`, (which would fail).

1. On the Home page, click **Create** and then click **Connection**.
2. Click one of the database types that support advanced properties.

   Supported databases are Oracle, Oracle Autonomous Data Warehouse, Snowflake, and My SQL.
3. Specify the connection details then save the connection.
4. On the home page, click the **Navigator**, then click **Data**, and then **Connections**.
5. Hover over the connection that you saved in Step 2 and click **Actions** then click **Inspect**.
6. Click **Advanced**, then use the Quoting Identifiers option to select the quoting identifiers used in the database.



For example, you might select **Double Quotes (" ")**. Oracle Analytics adds double quotes to the underlying SQL statement `select "EfG_Field" from "AbCd";` instead of issuing `select EfG_Field from AbCd;`.

> **Note:**
>
> You won't see the **Advanced** option if the connection was created with the **System connection** option selected. Advanced options aren't supported for system connections.

7. Click **Save**.

Manage Connections Using REST APIs

You can use Oracle Analytics Cloud REST APIs to programmatically manage connections to a range of data sources. For example, you might want to create a script that creates (or modifies) the same set of connections in both your test and production Oracle Analytics Cloud environments.

- About Connection REST APIs
- Typical Workflow for Managing Connections Using REST APIs
- How to Use REST APIs to Manage Data Source Connections
- Sample JSON Payloads for Data Sources

About Connection REST APIs

You can create, update, and delete connections for a range of data sources using REST APIs. This topic lists the types of data source connections you can manage with REST APIs.

> **Note:**
>
> *REST API for Oracle Analytics Cloud* provides detailed information about each REST API. See Connection REST Endpoints.

Supported Data Sources

- Oracle Database
- Oracle Autonomous Data Warehouse
- Oracle Essbase
- MySQL
- PostgreSQL
- Snowflake
- SQL Server
- Vertica

Connection Parameters

The connection parameters required for each data source are different. If you want to use the REST API to create or update a connection, you must know the JSON payload format required for your data source. See Sample JSON Payloads for Data Sources.

Typical Workflow for Managing Connections Using REST APIs

Here are the common tasks to start using Oracle Analytics Cloud REST APIs to programmatically manage connections. If you're using the REST APIs for the first time, follow these tasks as a guide.

Task	Description	REST API Documentation
Understand prerequisites	Understand and complete several prerequisite tasks. You must have permissions to create workbooks and connect to data in Oracle Analytics Cloud to manage data connections using REST APIs (**DV Content Author**).	Prerequisites
Understand OAuth 2.0 token authentication	Authentication and authorization in Oracle Analytics Cloud is managed by Oracle Identity Cloud Service. To access the Oracle Analytics Cloud REST APIs, you need an OAuth 2.0 access token to use for authorization.	OAuth 2.0 Token Authentication
Understand supported data sources	Read the topic that describes the types of data connections you can manage with REST APIs.	Supported Data Sources
Determine JSON payload formats	Read the topic that describes the JSON payload format for your data source and obtain the required connection parameters.	Sample JSON Payloads for Data Sources
Create a connection	Create a data connection for use in workbooks, reports, and dashboards.	Create a connection (Preview)
Update a connection	Update one or more properties of an existing data connection.	Update a connection (Preview)
Delete a connection	Delete a data connection.	Delete a connection (Preview)

How to Use REST APIs to Manage Data Source Connections

Use these samples and examples to help you manage data source connections with REST API requests using cURL.

- Sample cURL Format
- Examples

Sample cURL Format

Use the following cURL command format to create or update data source connections using REST APIs:

- **Simple JSON**

```
curl
--header "Authorization: Bearer <token>"
--header "Content-Type: application/json"
--request POST|PUT https://<hostname>/api/20210901/catalog/connections
--data "<data source connection payload>"
```

- **Multi-part/Form data**

```
curl
--header "Authorization: Bearer <token>"
--request POST|PUT https://<hostname>/api/20210901/catalog/connections
--form "cert=<security wallet file>"
--form "connectionParams=<data source connection payload>"
```

Where:

- **<token>** - OAuth 2.0 bearer token required to authenticate calls to Oracle Analytics Cloud REST APIs. See OAuth 2.0 Token Authentication.
- **<hostname>** - Host where Oracle Analytics Cloud is running.
- **<data source connection payload>** - Data source-specific connection information. See Sample JSON Payloads for Data Sources.
- **<security wallet file>** - Stores SSL-related information such as authentication and signing credentials, private keys, certificates, and trusted certificate. Required for some connection types, such as Oracle Database with SSL and Oracle Autonomous Data Warehouse (Mutual TLS).

Examples

The following examples show you how to create a connection to Oracle Autonomous Data Warehouse (ADW).

- Example 1 - Create a wallet-less (TLS) connection to Oracle ADW
- Example 2 - Create a connection to Oracle ADW that uses a credentials wallet file `cwallet.sso` (Mutual TLS)

Additional examples are available in *REST API for Oracle Analytics Cloud*. See Create a connection (Preview), Update a connection (Preview) and Delete a connection (Preview).

> **Note:**
>
> The JSON payloads in these examples are specific to Oracle ADW. The JSON payload format is different for other data sources. See Sample JSON Payloads for Supported Data Sources.

Example 1 Create a wallet-less (TLS) connection to Oracle ADW

In this example, you create a connection named `oracle_adw_walletless`. The request body includes simple JSON `application/json`.

```
curl
  --header "Authorization: Bearer <token>"
  --header "Content-Type: application/json"
  --request POST https://example.com/api/20210901/catalog/connections
  --data "{
     "version": "2.0.0",
     "type": "connection",
     "name": "oracle_adw_walletless",
     "description": "Sample Oracle ADW connection without a wallet created using
Connections API",
     "content": {
       "connectionParams": {
```

```
          "connectionType": "oracle-autonomous-data-warehouse",
          "connectionString": "(description= (retry_count=20)(retry_delay=3)
(address=(protocol=tcps)(port=1521)(host=adb.us-ashburn-1.oraclecloud.com))
(connect_data=(service_name=abcdefg1hijkl2m_adwwalletless_high.adb.oraclecloud.com))
(security=(ssl_server_dn_match=yes)))",
          "username": "ADMIN",
          "password": "<<password>>",
          "systemConnection": false,
          "remoteData": false,
          "sslType": "ServerSideSSL"
        }
      }
   }"
```

Response Body

```
{"connectionID":"J0FkbWluJy4nb3JhY2xlX2Fkd193YWxsZXRsZXNzJw=="}
```

Make a note of the Base64 encoded `connectionId` in the response body. Later on, you can use this value to update or delete the connection.

Example 2 Create a connection to Oracle ADW that uses a wallet file (Mutual TLS)

In this example, you create a connection named `oracle_adw_with_wallet`. The request body includes `multipart/form-data`, that is, requires both a wallet file from Oracle ADW (`cwallet.sso`) and Oracle ADW connection parameters.

```
curl
 --header "Authorization: Bearer <token>"
 --request POST https://<hostname>/api/20210901/catalog/connections
 --form "cert=@"/Users/scott/Downloads/Wallet_adw/cwallet.sso""
 --form "connectionParams= "{
    "version": "2.0.0",
    "type": "connection",
    "name": "oracle_adw_with_wallet",
    "description": "Sample Oracle ADW connection with a wallet created using Connections
API",
    "content": {
      "connectionParams": {
        "connectionType": "oracle-autonomous-data-warehouse",
        "connectionString": "(description= (retry_count=20)(retry_delay=3)
(address=(protocol=tcps) (port=1522)(host=adb.us-ashburn-1.oraclecloud.com))
(connect_data=(service_name=abcdefg1hijkl2m_walletadw_high.adwc.oraclecloud.com/))
(security=(ssl_server_dn_match=yes)))",
        "username": "ADMIN",
        "password": "<<password>>",
        "remoteData": "false",
        "systemConnection": false,
        "sslType": "ClientSideSSL"
       }
      }
    }"
```

Response Body

```
{"connectionID":"J2FkbWluJy4nb3JhY2xlX2Fkd193aXRoX3dhbGxldCc="}
```

Make a note of the Base64 encoded `connectionId` in the response body. Later on, you can use this value to update or delete the connection.

Sample JSON Payloads for Data Sources

To connect to data source you provide connection parameters. When you use the Connections REST API, you specify connection parameters in a JSON payload format. Use this table to determine the JSON payload for the data source you want to connect to.

Data Source	Request Type	Input Payload
Oracle Database (Non-SSL)	**application/ json**	**Basic Connection Type** `{ "version": "2.0.0", "type": "connection", "name": "oracle_db_non_ssl_basic", "description": "Sample non-SSL Oracle Database connection created using Connections API", "content": { "connectionParams": { "connectionType": "oracle-database", "host": "example.com", "port": "1521", "serviceName": "orcl", "username": "admin", "password": "<password>", "remoteData": false, "systemConnection": false } } }`  **Advanced Connection Type**  `{ "version": "2.0.0", "type": "connection", "name": "oracle_db_non_ssl_advanced ", "description": " Sample non-SSL Oracle Database connection created with the advanced connection string format using Connections API", "content": { "connectionParams": { "connectionString": "(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)(FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=example.com) (PORT=1521))) (CONNECT_DATA=(SERVICE_NAME=ORCLPDB1)))", "username": "admin", "password": "<password>", "connectionType": "oracle-database", "remoteData": false, "systemConnection": false } }`  `}`

Data Source	Request Type	Input Payload
		} }

Oracle Database with SSL	**multi-part/ form-data**	cwallet.sso (client credentials file)
		
		Basic Connection Type
		
		cert: <cwallet.sso file location>
		connectionParams: {
		"version": "2.0.0",
		"type": "connection",
		"name": "oracle_db_ssl",
		"description": "Sample Oracle Database connection with SSL created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "oracle-database",
		"host": "example.com",
		"port": "2484",
		"serviceName": "ORCLPDB1",
		"username": "admin",
		"password": "<password>",
		"systemConnection": false,
		"remoteData": false
		}
		}
		}

Data Source	Request Type	Input Payload

Oracle Autonomous Data Warehouse - Wallet-less (TLS)	**application/ json**	**Basic Connection Type**
		
		{
		"version": "2.0.0",
		"type": "connection",
		"name": "oracle_adw_walletless_basic",
		"description": "Sample Oracle ADW connection without a wallet created using Connections API",
		"content": {
		"connectionParams": { "connectionType": "oracle-autonomous-data-warehouse",
		"connectionString": "(description=(retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1521)(host=example.com))(connect_data=(service_name=example.com))(security=(ssl_server_dn_match=yes)))",
		"username": "admin",
		"password": "<password>",
		"systemConnection": false,
		"remoteData": false,
		"sslType": "ServerSideSSL"
		}
		}
		}

Data Source	Request Type	Input Payload

Oracle Autonomous Data Warehouse - Wallet (Mutual TLS)	**multipart/ form-data**	`cwallet.sso (client credentials file)`
		
		Basic Connection Type
		
		`cert: <cwallet.sso file location>`
		
		`connectionParams: {`
		` "version": "2.0.0",`
		` "type": "connection",`
		` "name": "oracle_adw_with_wallet",`
		` "description": "Sample Oracle ADW connection with wallet created using Connections API",`
		` "content": {`
		`  "connectionParams": {`
		`  "connectionType": "oracle-autonomous-data-warehouse",`
		`  "connectionString": "(description= (retry_count=20)(retry_delay=3) (address=(protocol=tcps) (port=1522) (host=example.com))`
		
		`(connect_data=(service_name=example.com))`
		
		`(security=(ssl_server_dn_match=yes)))",`
		`  "username": "admin",`
		`  "password": "<password>",`
		`  "remoteData": "false",`
		`  "systemConnection": false,`
		`  "sslType": "ClientSideSSL"`
		`  }`
		`  }`
		` }`

Data Source	Request Type	Input Payload

PostgreSQL (Non-SSL)	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "postgres_non_SSL",
		"description": "Sample PostgreSQL connection created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "postgresql-database",
		"host": "example.com",
		"port": "5432",
		"serviceName": "postgres",
		"username": "admin",
		"password": "<password>",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

PostgreSQL with SSL	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "postgres_SSL_Conn",
		"description": "Sample PostgreSQL connection with SSL created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "postgresql-database",
		"host": "example.com",
		"port": "5432",
		"serviceName": "postgres",
		"username": "admin",
		"password": "<password>",
		"sslType":"ServerSideSSL",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

Data Source	Request Type	Input Payload

SQL Server (Non-SSL)	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "SqlServer_non_ssl",
		"description": "Sample non-SSL SQL Server connection created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "sqlserver-database",
		"host": "example.com",
		"port": "1400",
		"serviceName": "sqlserver1",
		"username": "admin",
		"password": "<password>",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

SQL Server with SSL	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "SqlServer_ssl",
		"description": "Sample SQL Server connection with SSL created using the Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "sqlserver-database",
		"host": "example.com",
		"port": "60190",
		"serviceName": "sqlserver1",
		"username": "admin",
		"password": "<password>",
		"sslType":"ServerSideSSL",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

Data Source	Request Type	Input Payload

MySQL (Non-SSL)	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "MySql_no_SSL",
		"description": "Sample MySQL connection created using the Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "mysql-database",
		"host": "example.com",
		"port": "3307",
		"serviceName": "mysql1",
		"username": "admin",
		"password": "<password>",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

MySQL with SSL	**application/ json**	**Basic Connection Type**
		
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "MySql_ssl",
		"description": "Sample MySQL connection with SSL created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "mysql-database",
		"host": "example.com",
		"port": "3306",
		"serviceName": "mysql1",
		"username": "admin",
		"password": "<password>",
		"sslType":"ServerSideSSL",
		"remoteData": false,
		"systemConnection": false
		}
		}
		}

Data Source	Request Type	Input Payload

Oracle Essbase	**application/ json**	**Basic Connection Type**
		{
		"version":"2.0.0",
		"type": "connection",
		"name": "Oracle_Essbase",
		"description": "Sample Oracle Essbase connection created using Connections API",
		"content": {
		"connectionParams": {
		"connectionType": "oracle-essbase",
		"dsn":"example.com",
		"username":"admin",
		"password":"<password>",
		"remoteData": false,
		"systemConnection": false,
		"authentication":"current"/"private"/"sso"
		}
		}
		}

Connect to an Oracle Database

You can create a connection to a database and use the connection to access data, build a dataset, and visualize data.

Video

You can't use remote connections to save a dataset from a Data Flow.

1. On the Home page, click **Create**, then click **Connection**.
2. In Create Connection dialog, click the connection type, for example, **Oracle Database**.
3. Enter a unique name for the connection, and specify database connection details.
   - If you're not creating an SSL connection, specify the connection information for the database, such as the hostname, port, credentials, service name and so on.
   - If you're creating an SSL connection, in the **Client Wallet** field, click **Select** to browse for the `cwallet.sso` file. Ask your administrator for the location of the `cwallet.sso` file.
4. Use the **Connection Type** option to specify how you'd like to connect.
   - When you create connections to Oracle Databases, you can connect in two ways using the **Connection Type** option:
     - **Basic** - Specify the **Host**, **Port**, and **Service Name** of the database.
     - **Advanced** - In the **Connection String** field specify the Single Client Access Name (SCAN) ID of databases running in a RAC cluster. For example:
       ```
       sales.example.com =(DESCRIPTION= (ADDRESS_LIST= (LOAD_BALANCE=on)
       (FAILOVER=ON) (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.111)(PORT=1521))
       (ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.222)(PORT=1521))
       ```

```
(ADDRESS=(PROTOCOL=tcp)(HOST=123.45.67.333)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME= salesservice.example.com)))
```

- **Enable Bulk Replication** - If you're loading a dataset for a workbook, then this option should be turned off and you can ignore it. This option is reserved for data analysts and advanced users for replicating data from one database to another database.

5. If you're connecting to an on-premises database, click **Use Remote Data Connectivity**.

   Check with your administrator that you can access the on-premises database.

6. If you want data modelers to be able to use these connection details. click **System connection**. See Database Connection Options.

7. Click **Save**.

8. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to Oracle Analytic Views

You can create a connection to Analytic Views in an Oracle Database to access data, build a dataset, and visualize data.

Dataset authors can use this connection type to consume Oracle Analytic Views data, including multi-dimensional objects, without having to understand the complexities of Java database connectivity (JDBC).

1. On the Home page, click **Create**, then click **Connection**.

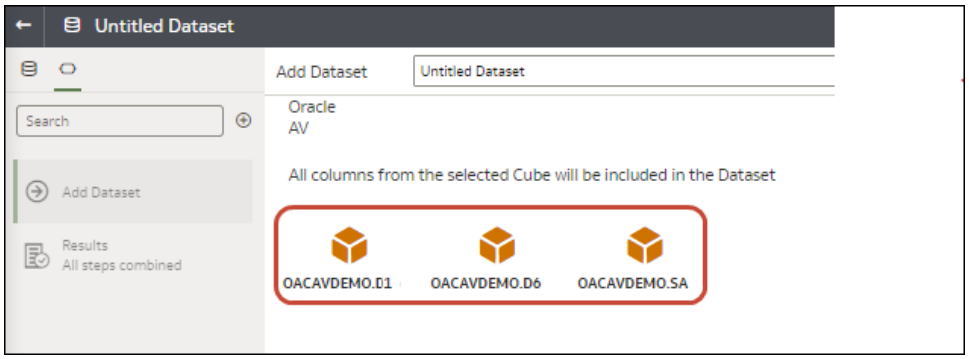
2. Click **Oracle Analytic Views** and enter the connection details.

   - For **Connection Type**, select **Basic** to connect by specifying **Host** as an Internet Protocol (IP) address, **Port**, and Service Name for the Oracle database. For example, **Host** = *<IP address>*, **Port** = 9018, and **Service Name** = PDBORCL.
   - Alternatively, select **Advanced** to connect by specifying a **Connection String**. For example, `(DESCRIPTION =(ADDRESS_LIST =(ADDRESS = (PROTOCOL = TCP) (HOST = <IP address>)(PORT = 9018))) (CONNECT_DATA = (SERVICE_NAME = PDBORCL)))`

3. Click **Save**.

   You can now create datasets using the connection and build workbooks.

When you create a dataset using the connection, select one of the cubes listed in the database. Then build a workbook using that dataset and start visualizing your data.



Connect to Oracle Autonomous Data Warehouse

You can create a connection to Oracle Autonomous Data Warehouse and use the connection to access data.

Before you start, ask your Autonomous Data Warehouse (ADW) administrator to configure ADW to allow access from Oracle Analytics Cloud. The configuration steps that administrators follow are different for public and private types of connection:

- For public connections, see Enable Access to Oracle Autonomous Data Warehouse Prerequisites.
- For private connections, see How do I connect to a private Oracle Autonomous Data Warehouse in a customer VCN?.

You can connect to Oracle Autonomous Data Warehouse using security certificates downloaded from Oracle Autonomous Data Warehouse to a wallet (known as mTLS, or Mutual Transport Layer Security), or without a wallet (known as TLS, or Transport Layer Security). The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Data Warehouse. The wallet file (for example, `wallet_ADWC1.zip`) that you upload must contain SSL trusted certificates, to enable SSL on your Oracle Database Cloud connections.

Tutorial

1. To connect using a credentials wallet file, (known as **Mutual TLS** connection), download the SSL certificates from Oracle Autonomous Data Warehouse.

   See Download Client Credentials (Wallets) in *Using Oracle Autonomous Database Serverless*.

   The credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Data Warehouse. The wallet file (for example, `wallet_ADWC1.zip`) that you upload must contain SSL certificates.

   To connect without using a credentials wallet file (known as **TLS** connection), skip Step 1 and go straight to the Step 2.
2. On the Home page, click **Create** then click **Connection**.
3. Click **Oracle Autonomous Data Warehouse**.
4. Enter a user-friendly **Connection Name** and **Description**.
5. For **Encryption Type**:
   - To connect without a credentials wallet file, select **TLS** as the **Encryption Type**, enter a **Connection String**, then enter a **Username** and **Password** of a user in Oracle Autonomous Data Warehouse.
   - To connect using a credentials wallet file, select **Mutual TLS** as the **Encryption Type**, then click **Select** and browse for and select the Client Credentials wallet file that you downloaded from Oracle Autonomous Data Warehouse (for example, wallet_ADWC1.zip), then enter a **Service Name**. The **Client Credentials** field displays the `cwallet.sso` file. See Selecting an Oracle Autonomous Data Warehouse Database Service Name.
     Tip: If you define the Oracle Autonomous Data Warehouse connection using an instance wallet, the high service name is selected by default. Change the name to low or medium to avoid limiting the number of concurrent connections.
6. If you're connecting to a remote database, click Use Remote Data Connectivity.

Check with your administrator that you can access the remote database.

7. If you want data modelers to be able to use these connection details. click **System connection**. See Database Connection Options.
8. Click **Save**.

You can now create datasets from the connection.

Selecting an Oracle Autonomous Data Warehouse Database Service Name

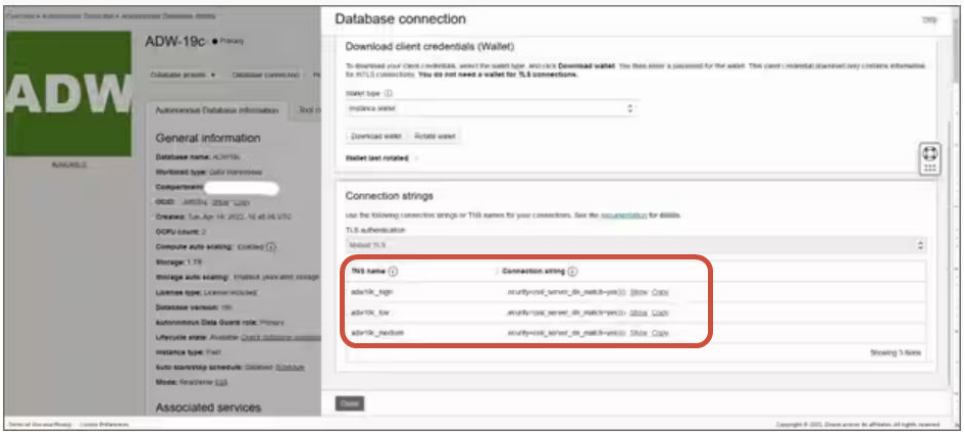
Selecting the correct prebuilt database service name is key to connecting to Oracle Autonomous Data Warehouse. Learn about the different prebuilt database service names and which one you should choose.

What are the prebuilt database service names?

Oracle Autonomous Data Warehouse provides three database service names for connections in the following format:

- `databasename_high` - Highest resources, lowest concurrency. Queries run in parallel.
- `databasename_medium` - Fewer resources, higher concurrency. Queries run in parallel.
- `databasename_low` - Least resources, highest concurrency. Queries run serially.

These names are contained in the tnsnames.ora file in the Oracle wallet. Click **Database Connections** in the Oracle Cloud Infrastructure Console to see the strings.



About Consumer Groups in Oracle Resource Manager

The database service names are mapped to consumer groups in the Resource Manager that limit the number of simultaneous connections and queries that can run in Oracle Autonomous Data Warehouse at the same time (concurrency) and the maximum number of parallel processes that are allowed per query (parallel_degree_limit). These limits are based on the number of ECPUs or OCPUs licensed and whether auto-scaling is enabled.

The following table shows sample concurrent connection values for a database with 32 ECPUs with ECPU auto-scaling disabled and enabled.

Database Service Name	Number of Concurrent Queries with ECPU Auto Scaling Disabled	Number of Concurrent Queries with ECPU Auto Scaling Enabled
high	3	9
medium	20 (.63 × number of ECPUs)	60 (1.89 × number of ECPUs)
low	Up to 4800 (150 x number of ECPUs)	Up to 4800 (150 x number of ECPUs)

Choosing the Optimal Database Service Name for Oracle Analytics

The largest number of simultaneous queries that can run for the high database service is three without auto-scaling and nine with auto-scaling enabled. This limit can be reached by three users connected to the high database service name running one query each or three reports in one Oracle Analytics dashboard for a single user.

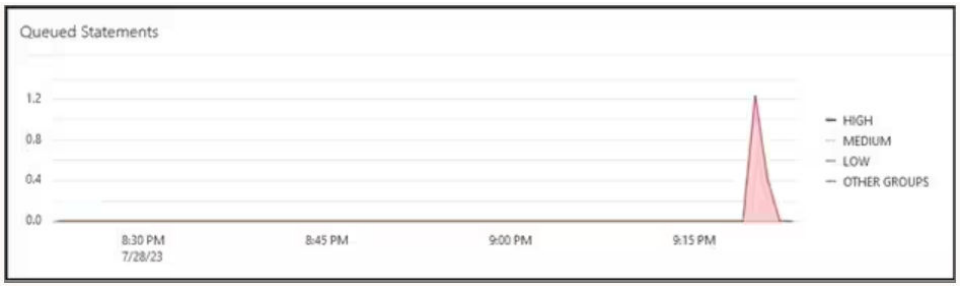
The low service name works well for most Oracle Autonomous Data Warehouse workloads with Oracle Analytics, but to utilize parallel queries, select the medium service name. The parallel degree limit for the low service name is one, meaning no parallelism. If you're connected to the low service name, even if the parallel degree is specified at the table or index level, the degree of parallelism is reduced to one and the query doesn't run in parallel. The parallel degree limit (per query) for medium and high equals two times the number of licensed CPUs.

Note: Connecting to a database that's part of Oracle Fusion Analytics Warehouse (Fusion Analytics) requires using the low service name to allow for the maximum number of concurrent queries.

Monitoring queued statements

If the maximum number of concurrent queries limit is reached, the excess queries are queued. Oracle Autonomous Data Warehouse provides a metric to check for queued statements.

Select **Database Actions** and **Database Dashboard** in the Oracle Autonomous Data Warehouse page of Oracle Cloud Infrastructure Console.

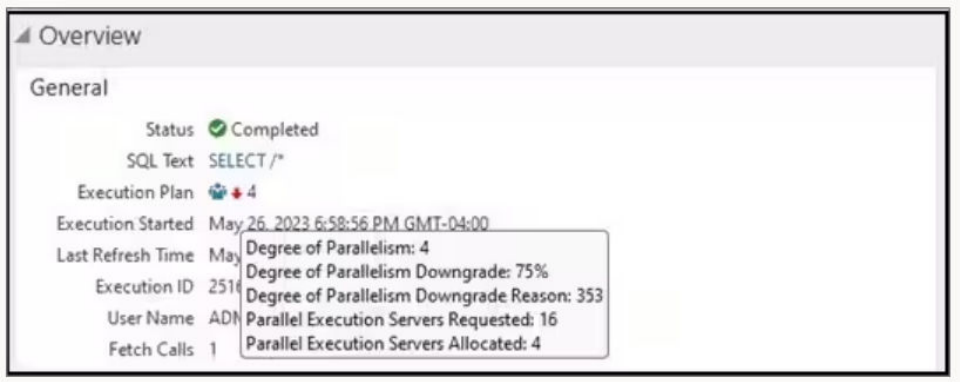


Select **Performance Hub,** and select **SQL Monitor Tab** to see the queued statement status, which is displayed as a grey clock. In this example, three queries are running with the high service name, one is queued, and one query is running with the medium service name. The queued statement executes when one of the three queries running with the high service name completes.

Status	Duration	SQL ID	SQL Plan Hash	User Name	Parallel
	8.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	
	5.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	4
	12.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	4
	29.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	4
	32.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	4
	31.00 sec	bsmqs5jr2ddnk	396151021	ADMIN	4

Monitoring parallelism

If the parallel degree limit is exceeded, you see the degree of parallelism (DOP) downgrade in the SQL monitor report. The degree of parallelism downgrade reason of 353 means that the Resource Manager downgraded the statement due to the maximum degree of parallelism limit.



For Oracle Database version 18 and higher, the downgrade reason codes are described in the following table:

ID	Reason Codes
352	DOP downgrade due to adaptive DOP.
353	DOP downgrade due to resource manager max DOP.
354	DOP downgrade due to insufficient number of processes.
355	DOP downgrade because slaves failed to join.

Resource Manager CPU Wait Event

A session waiting to be allocated CPU by the Resource Manager increments the resmgr:cpu quantum wait event. To reduce the occurrence of this wait event, verify that the low or medium service name is being used for the OAC connection or increase the number of CPUs allocated to ADW.

To see the number of waits and the average wait time, review Foreground Wait Events in the Automatic Workload Repository (AWR) report for the resmgr:cpu quantum wait event.

In this example, there were a total of 272 waits, waiting on average 588.91 milliseconds each for a total wait time of 160 seconds. It was determined that the reason was that the high database service name was being used for the OAC connection. These wait periods

disappeared once the customer switched to the medium service, and the periodic slowness of their dashboard was resolved.

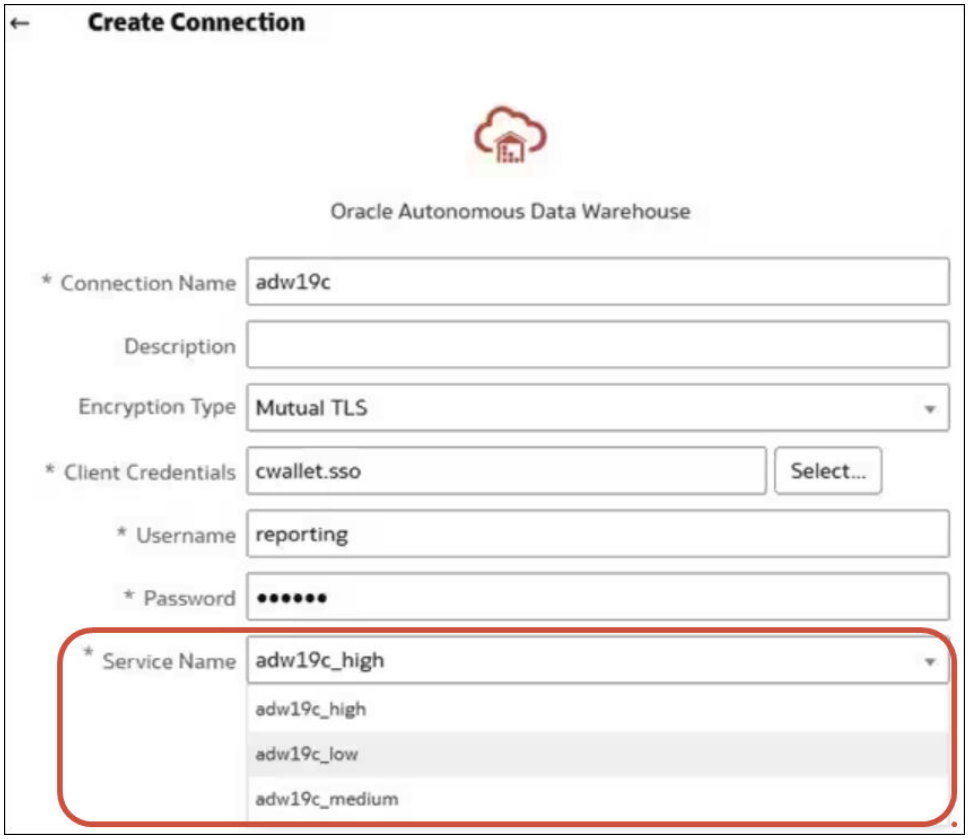
Foreground Wait Events

- s - second, ms - millisecond, us - microsecond, ns - nanosecond
- Only events with Total Wait Time (s) >= .001 are shown
- ordered by wait time desc, waits desc (idle events last)
- %Timeouts: value of 0 indicates value was < .5%. Value of null is truly 0

Event	Waits	%Time -outs	Total Wait Time (s)	Avg wait	Waits /txn	% DB time
resmgr:cpu quantum	272		160	588.91ms	0.01	0.26

Tip When Creating a Connection to Oracle Autonomous Data Warehouse in Oracle Analytics

In Oracle Analytics, when you define the Oracle Autonomous Data Warehouse connection using the instance wallet, the high service name is selected by default. Change the name to low or medium to avoid limiting the number of concurrent connections.



Connect to Oracle Autonomous Transaction Processing

You can create a connection to Oracle Autonomous Transaction Processing and use the connection to access data.

Before you start, ask your Autonomous Data Warehouse (ADW) administrator to configure ADW to allow access from Oracle Analytics Cloud. The configuration steps that administrators follow are different for public and private types of connection:

- For public connections, see Enable Access to Oracle Autonomous Data Warehouse Prerequisites.
- For private connections, see How do I connect to a private Oracle Autonomous Data Warehouse in a customer VCN?.

1. To connect using a credentials wallet file, (known as **Mutual TLS** connection), download the SSL certificates from Oracle Autonomous Data Warehouse.

   See Download Client Credentials (Wallets) in *Using Oracle Autonomous Database Serverless*.

   To connect without using a credentials wallet file (known as **TLS** connection), skip Step 1 and go straight to the Step 2.

   A credentials wallet file secures communication between Oracle Analytics and Oracle Autonomous Data Warehouse. The wallet file (for example, `wallet_ADWC1.zip`) that you upload must contain SSL certificates.
2. On the Home page, click **Create** then click **Connection**.
3. Click **Oracle Autonomous Transaction Processing**.
4. Enter a user-friendly **Connection Name** and **Description**.
5. For **Encryption Type**:

   To connect without a credentials wallet file, select **TLS**, then skip to the next step.

   To connect using a credentials wallet file, select **Mutual TLS**, then click **Select** and browse for and select the Client Credentials wallet file that you downloaded from Oracle Autonomous Data Warehouse (for example, wallet_ADWC1.zip). The **Client Credentials** field displays the `cwallet.sso` file.
6. For the **TLS** connection type (without a wallet), enter a **Connection String**, the **Username** and **Password** of a user in Oracle Autonomous Data Warehouse.
7. If you want data modelers to be able to use these connection details. click **System connection**. See Database Connection Options.
8. Click **Save**.
9. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to Analytic Views in Oracle Autonomous Data Warehouse

Connect to analytic views to visualize data in Oracle Autonomous Data Warehouse.

Before you start, ask your Oracle Analytics Cloud administrator to make analytic views available through a local subject area (semantic model).

1. In Oracle Analytics Cloud, on the Home page, click **Create** and then click **Dataset**.
2. Click **Local Subject Area**.
3. Select a subject area that's based on an analytic view.
4. Select the facts and measures that you want to analyze and add to the dataset.

   You can now visualize data in this dataset.

Connect to Oracle Fusion Cloud Applications Suite

You can connect to Oracle Fusion Cloud Applications Suite and create datasets that help you visualize, explore, and understand your applications data.

Topics:

About the Oracle Applications Connector

The "Oracle Applications" connection type () enables you to use Oracle Analytics to visualize data from applications in Oracle Fusion Cloud Applications Suite. For example, Oracle Fusion Cloud Financials.You can also use the "Oracle Applications" connection type to connect to your on-premises Oracle BI Enterprise Edition deployments (if patched to an appropriate level) or connect to another Oracle Analytics service.

You can connect to these applications in Fusion Applications Suite:

- Oracle Fusion Cloud Financials
- Oracle Fusion Cloud Human Capital Management
- Oracle Fusion Cloud Loyalty
- Oracle Fusion Cloud Procurement
- Oracle Fusion Cloud Project
- Oracle Fusion Cloud Supply Chain Planning
- Oracle Sales Automation

> **Note:**
>
> When you connect to applications in Fusion Applications Suite, you access the data from an Oracle Transactional Business Intelligence report. These reports are subject to caching in Oracle Transactional Business Intelligence, and the data available in Oracle Analytics is based on the cached data. You can't control the cache behavior in Oracle Transactional Business Intelligence from Oracle Analytics.

Connect to an Application in Oracle Fusion Cloud Applications Suite

Create a connection to an application in Oracle Fusion Cloud Applications Suite to visualize data from that application. For example, you can connect to Oracle Fusion Cloud Financials with Oracle Transactional Business Intelligence. You can also connect to Oracle BI Enterprise Edition.

1. On the Data page or Home page, click **Create**, then click **Connection**.
2. Click **Oracle Applications**. 
3. Enter the connection details.
   - For **Connection Name**, specify a user-friendly name to identify the connection details in Oracle Analytics.
   - For **Host**, enter the URL for Fusion Applications Suite with Oracle Transactional Business Intelligence or Oracle BI EE.

     > **Note:**
     >
     > Don't enter the URL of the Oracle Analytics application you're currently logged in to. If you want to visualize the data used in a local analysis, create a dataset based on the analysis (local subject area). See Create a Dataset from a Subject Area in Your Instance.

   - For **Username** and **Password**, specify the Oracle Applications user credentials.
4. Under **Authentication**, specify how you'd like to authenticate the connection:
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source. See Configure Impersonate User for the Use Active User Credentials Option. Make sure that the Oracle Analytics user exists in Oracle Transactional Business Intelligence .
5. Save the details.

   You can now create datasets from the connection.

The connection is visible only to you (the creator), but you can create and share datasets for it.

Configure Impersonate User for the Use Active User's Credentials Option

You can configure "impersonate user" to enable the active user's credentials option when you use the Oracle Applications connection type.

You can implement "impersonate user" for connections to Oracle Fusion Cloud Applications Suite with Oracle Transactional Business Intelligence, and Oracle BI EE. When you use "impersonate user", users can access data allowed for the active user's permissions,

privileges, and role assignments. Users aren't prompted to enter their user name and password.

Tips on setting up impersonation

- Provision one administrator user in the Oracle Fusion Cloud Enterprise Performance Management (EPM) identity domain with the necessary roles and permissions to impersonate other users.
- Specify the credentials for this EPM administrator user when you import the metadata using the BI Model Administration Tool. The credentials are stored in the connection pool of the RPD data model created by the BI Model Administration Tool.
- In addition, the SSO checkbox must be checked on the Planning server connection pool in the RPD.
- With this setup, only one native user needs to be provisioned in the EPM identity domain. Other end users log into Oracle Analytics using their SSO credentials, and Oracle Analytics passes the SSO username to Planning and Budgeting Cloud Service when connecting, along with the credentials of the native EPM administrator user that has impersonation permissions.
- **Note**: Logging into EPM using SSO credentials isn't supported by the REST API or the Planning ADM driver.

Provision Impersonate User for Connection to Oracle Fusion Cloud Applications Suite

You can provision the impersonate user functionality in Oracle Fusion Cloud Applications Suite when your connection target is an application in Oracle Fusion Cloud Applications Suite with Oracle Transactional Business Intelligence.

1. Log in to Oracle Fusion Applications as an administrator with privileges to create or modify a role.
2. Display the Security Console dialog, and display the **Users** tab.
3. Click **Add User Account** to create a user. For example, create a user called DV Admin.
4. Display the **Roles** tab, and click **Create Role**.
5. Enter a role name in the **Role Name** field. For example, enter `DV Access`.
6. Enter a code for the role name in the **Role Code** field. For example, enter `DV_ACCESS`.
7. Select `BI - Abstract Roles` in the **Role Category** field.
8. Skip the steps Function Security Policies and Data Security Policies.
9. In the Role Hierarchy step, click (+) **Add Role** and select the existing `BIImpersonator` role in the Add Role Membership dialog.
10. Select the user that you created (for example, DV Admin).
11. Click **Add User to Role** in the Add User dialog.
12. Click **Save and Close**.

    The DV Admin user is added to the BI Impersonator role, and you can use the DV Admin user in Oracle Analytics in conjunction with the **Use Active User's Credentials** option in the Create Oracle Application Connection dialog.

    You can now test the impersonate functionality.

Provision Impersonate User for Connections to Oracle BI EE On-Premises

You can provision the impersonate user functionality in Oracle Fusion Middleware when your Oracle Application connection target is an on-premises installation of Oracle BI EE.

1. Log in to Oracle Fusion Middleware Control for your Oracle BI EE instance using an administrator account.
2. Click the **Weblogic Domain** option, and select **Security** and **Application Policies**.
3. Click **Create** to display the Create Application Grant dialog.
4. Click (+) **Add** in the Permissions area.
5. Select **Resource Types**.
6. Select **oracle.bi.user** from the drop-down list.
7. Click **Continue**.
8. Enter an asterisk (*) in the **Resource Name** field.
9. Select "`impersonate`" in **Permission Actions**.
10. Click **Select**.
11. Click (+) **Add** in the Grantee section.
12. Select **User** from the **Type** drop-down list.

    You grant the newly created permission to either an application role or user.
13. Select **Includes** from the **Principal Name** drop-down list and enter an asterisk (*) into the field.
14. Click the **>** arrow to display a list of users.
15. Select the user that you want to give the permission to and click **OK**.

    You can now test the impersonate functionality.

Connect to Oracle Fusion Cloud Enterprise Performance Management (EPM)

You can create a connection to Oracle Fusion Cloud Enterprise Performance Management (EPM) and use the connection to access data.

Before you start, make sure that your product is supported. See Which Oracle EPM Business Processes Does Oracle Analytics Support?.

1. On the Home page, click **Create** and then click **Connection**.
2. Click **Oracle EPM Cloud** and enter the connection details.
3. For **URL**, enter the URL for the Oracle Fusion Cloud Enterprise Performance Management (EPM) data source.

   For EPM deployments on OCI Classic, enter the full URL in the format:

   ```
   https://epm-mySecurityDomain.epm.domain.mycloud.oracle.com/mySecurityDomain
   ```

   For example:

   ```
   https://example-idDomain.pbcs.dom1.oraclecloud.com/idDomain
   ```

For EPM deployments on OCI Gen 2, enter the EPM URL, excluding the application context (epmcloud or Hyperion):

```
https://epm-99999-plan.hap.fra.demoservices999.oraclepdemos.com/epmcloud
```

For example:

```
https://epm-99999-plan.hap.fra.demoservices999.oraclepdemos.com
```

4. Under **Authentication**, click **Use the active user's credentials**.

   Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source. Make sure that the Oracle Analytics user exists in Oracle Fusion Cloud Enterprise Performance Management (EPM).

   See Configure Impersonate User for the Use Active User's Credentials Option.

5. Save the details.

   You can now create datasets from the connection and visualize the data.

Visualize Data From Oracle Enterprise Performance Management (Oracle EPM)

Which Oracle EPM Business Processes Does Oracle Analytics Support?

When you integrate Oracle Analytics with Oracle Enterprise Performance Management, make sure that you're connecting to one of the supported business processes:

Oracle Analytics supports:

- Financial Consolidation and Close
- FreeForm
- Planning and Planning Modules
- Profitability and Cost Management
- Tax Reporting

Oracle Analytics doesn't support:

- Account Reconciliation
- Enterprise Data Management Cloud
- Narrative Reporting

Connect to Essbase

You can create, edit, and delete a connection to Essbase and use the connection to create datasets from Essbase cubes.

Topics:

- Create a Connection to Oracle Essbase
- Create a Connection to Oracle Essbase Data on a Private Network
- Enable Users to Visualize Oracle Essbase Cubes Using Single Sign-on

You can also connect to private Essbase data sources via Private Access Channel. See Connect to Private Data Sources Through a Private Access Channel.

Create a Connection to Oracle Essbase

You can create a connection to Oracle Analytics Cloud – Essbase and use the connection to access source data.

1. On the Home page, click **Create**, and then click **Connection**.
2. Click **Oracle Essbase**.
3. Enter the connection details.
4. For **DSN** (data source name), enter the agent URL for your data source.

   For Oracle Analytics Cloud – Essbase use the format:

   `https://fully_qualified_domain_name/essbase/agent`

   For example: `https://my-example.analytics.ocp.oraclecloud.com/essbase/agent`.

   With this URL, you can connect without having to open any ports or performing additional configuration. Oracle Analytics Cloud – Essbase must have a public IP address and use the default port.
5. For **Username** and **Password**, enter user credentials with access to the Essbase data source.
6. Under **Authentication**, specify how you'd like to authenticate the connection:
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.
7. Save the details.

   You can now create datasets from the connection.

Create a Connection to Oracle Essbase Data on a Private Network

You can create a connection to Oracle Essbase data on a private network and use the connection to access data.

You secure a connection to Oracle Essbase data held on a private network by using Data Gateway or Private Access Channel (for datasets or semantic models). For Data Gateway, your Administrator installs Data Gateway in your private network. Data Gateway then redirects queries to the Essbase host. For Private Access Channel, see Connect to Private Data Sources Through a Private Access Channel.

Before you start, make sure that your Oracle Essbase Marketplace instance has signed certificates.

1. On the Home page, click **Create**, and then click **Connection**.
2. Click **Oracle Essbase**.

3. Enter the connection details.
4. For **DSN** (data source name), enter the URL for your data source.

   These are the connectivity options to access Oracle Essbase on your private network:

   Note: When you specify the URL for your data source, <hostname>:<port> specify the hostname and port of the host accessible on the public internet that forwards traffic to your Remote Data Connector host.

   - The basic URL syntax:

     ```
     http(s)://<hostname>:<port>/essbase/capi/<private essbase host
     address>/<Oracle Essbase Agent port on the
           specified host>
     ```

     For example,

     ```
     https://myproxyhost.example.com:1234/essbase/capi/mylocalhost/1423
     ```

   - When Oracle Essbase is running on a secure port:

     ```
     http(s)://<hostname>:<port>/essbase/capi/<private essbase host
     address>/<Oracle Essbase Agent Secure port on the specified host>/secure
     ```

   - When Oracle Essbase is running on a secure port, with a self-signed certificate:

     ```
     http(s)://<hostname>:<port>/essbase/capi/<private essbase host
     address>/<Oracle Essbase Agent Secure port on the specified host>/
     secure/selfsigned
     ```

5. Under **Authentication**, specify how you'd like to authenticate the connection:
   - **Always use these credentials** - Oracle Analytics always uses the login name and password you provide for the connection. Users aren't prompted to log in.
   - **Require users to enter their own credentials** - Oracle Analytics prompts users to enter their own user name and password for the data source. Users can only access the data for which they have the permissions, privileges, and role assignments.
   - (Displayed if Oracle Analytics supports impersonation for this database type) **Use the active user's credentials** - Oracle Analytics doesn't prompt users to sign in to access the data. The same credentials they used to sign in to Oracle Analytics are also used to access this data source.
6. If you're connecting to an on-premises database, click **Use Remote Data Connectivity**.

   Check with your administrator that you can access the on-premises database.
7. Save the details.

   You can now create datasets from the connection.

Enable Users to Visualize Oracle Essbase Cubes Using Single Sign-on

With an Oracle Essbase data source, you can use impersonation to enable multiple users to visualize data in Oracle Essbase cubes without having to log in twice.

1. In Oracle Essbase, create a user with permissions to impersonate other users (using the *EssLoginAs* functionality).

2. In Oracle Analytics, on the Home page, click **Create**, then **Connection**, and click **Oracle Essbase**.
3. In the Create Connection page:
   a. In **DSN**, specify the agent URL for your Oracle Essbase data source.
   b. For **Username** and **Password**, enter the credentials for the user that you created in Step 1.
   c. Under **Authentication**, click **Use the active user's credentials**.
4. Share this connection with the multiple users who need to visualize data. See task below.

   If they've already logged in with their Single Sign-on credentials, they can access the cubes without having to log in again.

Share a Data Source Connection

You can assign access permissions to the data source connections that you create or administer.

1. On the Home page click the **Navigator**. Click **Data**, and then click **Connections**.
2. Hover over the connection that you want to share, click **Actions**, then select **Inspect**.
3. Click **Access**, and use the tabs to grant access:
   - **All** - Share the connection with individual users or roles.
   - **Users** - Share the connection with individual users.
   - **Roles** - Share the connection with application roles (for example, BI Consumer), so that all users with those roles can use the connection.
4. Use the **Add** box to search for and select a user or role.

   The user or role is displayed in the list below with the default privileges **Read-Only**.
5. To change the default privileges, select one of the following:
   - **Full Control** - The user or role can use the connection to create datasets, and modify, rename, or delete the connection. They can also change the privileges for the connection.
   - **Read-Write** - The user or role can use the connection to create datasets, and modify or rename the connection (but not delete it).
   - **Read-Only** - The user or role can use the connection to create datasets, but not change the connection details.
6. Click **Save**.

When users next log in, they can use connections that you've shared to visualize data from this database.

Connect to NetSuite

Connect to a NetSuite (NetSuite2.com) data source to visualize ERP and CRM data.

1. On the Home page, click **Create** and then click **Connection**.
2. Click **Oracle Netsuite**.
3. Enter the connection details.

To obtain the connection details for your NetSuite application, go to the NetSuite Portal home page, and navigate to **Settings** then **Set up SuiteAnalytics Connect**.

In **Role ID**, make sure that you specify the ID for a role name that doesn't contain spaces or special characters. Role names containing spaces or special characters can cause data flows to fail with an internal or syntax error.

4. Save the details.
5. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to Oracle Talent Acquisition Cloud

You can create a connection to Oracle Talent Acquisition Cloud and use the connection to access data.

1. On the Home page, click **Create** and then click **Connection**.
2. Click **Oracle Talent Acquisition** and enter the connection details.
3. For **Host**, enter the URL for the Oracle Talent Acquisition data source.

   For example, if the Oracle Talent Acquisition URL is `https://example.taleo.net`, then the connection URL that you must enter is `https://example.taleo.net/smartorg/Bics.jss`.
4. Select an **Authentication** option.
   - Select **Always use these credentials**, so that the login name and password you provide for the connection are always used and users aren't prompted to log in.
   - Select **Require users to enter their own credentials** when you want to prompt users to enter their user name and password to use the data from the Oracle Talent Acquisition Cloud data source. Users are required to log in see only the data that they have the permissions, privileges, and role assignments to see.
5. Click **Save**.
6. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to a Database Using Delta Sharing

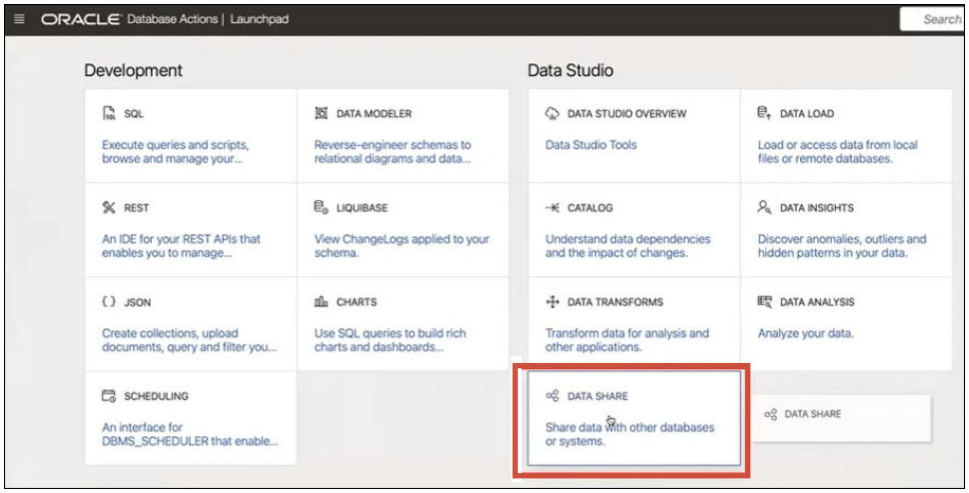
You can connect to some databases using the Delta Sharing protocol, for example, Oracle Autonomous Database, and visualize the data.

The Delta Sharing protocol provides secure data access without direct access to the source.

For a list of databases that support Delta Sharing, see List of Supported Databases in Oracle Analytics Cloud.

Use the connection to create a dataset and build workbooks. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created. Save the dataset and use it to build workbooks.

Before you start, ask the target database administrator to set up a delta sharing area and share it with you. For example, in Oracle Autonomous Database, an administrator creates a Data Share area, and shares it with you so that you receive a email containing an activation link. The link enables you to download a JSON file containing profile details required to create a connection in Oracle Analytics.



1. Contact the database administrator to request a data share.
2. In the activation email that you receive from the database administrator, click the activation link.
3. In the activation dialog, click **Get Profile Information**.

   A credentials file for the target database is downloaded to your local area in JSON format.
4. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
5. In Create Connection, click **Delta Share**.
6. Enter a **Connection Name** and optional **Description**.
7. In **Connection Type**, select a type suitable for your data source. For example:
   - For Oracle Autonomous Database, select **Client Credentials**.
   - For DataBricks, select **Bearer Token**.
8. Click **Import File** then select the JSON file containing connection details.

   Oracle Analytics populates the rest of the input fields with values from the import file.
9. Click **Save**.

You're now ready to create a workbook and start visualizing your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created and create a workbook.

Connect to Dropbox

You can create a connection to Dropbox and use the connection to access data.

LiveLabs Sprint

Before you start, set up a Dropbox application. See the Dropbox documentation.

1. Ask your Oracle Analytics administrator to allow connections to Dropbox.

   Your Oracle Analytics administrator must register the following domains as safe:

   ```
   api.dropboxapi.com
   ```

   ```
   *.dropbox.com
   ```

   See Register Safe Domains.

2. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
3. Click **Dropbox**.
4. Enter a **Connection Name** and optional **Description**.
5. Copy the URL displayed in the **Redirect URL** field.
6. In the Dropbox application, sign-in, and paste the **Redirect URL** URL into the Dropbox **OAuth 2 Redirect URIs** field, then click **Add**.
7. In Dropbox, copy the key in the **App Key** field.
8. In Oracle Analytics, paste the **App Key** key into the **Client ID** field.
9. In Dropbox, locate the **App Secret** field, then click **Show** and copy the value.
10. In Oracle Analytics, paste the **App Secret** value into the **Client Secret** field, then click **Authorize**.
11. In the Dropbox prompt to authorize the connection, click **Allow**.

    The Create Connection dialog refreshes and displays the name of the Dropbox account and associated email account.
12. Save the connection.
13. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to Google BigQuery

You can create a connection to a Google BigQuery database and use the connection to visualize data in a BigQuery project.

Before you start, note the following:

- Creating a connection to Google BigQuery is resource-intensive. It's best to create one connection and share it with other users, rather than having multiple users create their own connection.
- Oracle Analytics builds a cache of tables and schemas for each project in Google BigQuery. You should limit the projects, tables, and schemas to only those required for the analysis.
- Depending on the amount of Google BigQuery data, creating a connection can take up to several hours, so allow time for this process to complete.
- After the connection is created, allow time before you use the connection to start analyzing your data.

1. In Google BigQuery, create a Service Account.
    a. Add a role (for example, BigQuery User) with the permission `bigquery.jobs.create` to the Service Account.
    b. Add Users to the role.
    c. Add a JSON key.
2. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
3. Click **BigQuery**.
4. Enter the connection details.
    - In **Connection Name**, specify a user-friendly name to identify the connection details in Oracle Analytics.

- In **Project**, specify the BigQuery project you want to analyze using the name exactly as defined in Google BigQuery (case-sensitive).
- In **Service Account Email**, specify the email address used to log into Google BigQuery.
- In **Service Account Private Key**, upload the Service Account Private Key (JSON format).

5. Save the details.

   You can now create analytics dataset workbooks based on your Google BigQuery data. When you create a dataset, you navigate to BigQuery tables and select the data facts and measures to analyze. Alternatively, you can use a SQL Query to get the data directly.

Connect to Google Drive or Google Analytics

You can create a connection to Google Drive or Google Analytics and use the connection to access data.

Before you start, set up a Google Drive or Google Analytics application. See the Google documentation.

Use the latest Google Analytics connector, which supports column-listing.

1. Ask your Oracle Analytics administrator to allow connections to Google.

   Your Oracle Analytics administrator must register the following domains as safe:

   ```
   api.google.com
   ```

   ```
   *.google.com
   ```

   See Register Safe Domains.

2. On the Oracle Analytics Home page, click **Create**, then click **Connection**.
3. Click **Google Drive** or **Google Analytics** to display the Create Connection dialog.
4. Enter a **Connection Name** and optional **Description**.
5. Copy the URL displayed in the **Redirect URL** field.
6. In the Google application, on the Credentials page, paste the **Redirect URL** value into the Google "Authorized redirect URIs" field, then click **Add**.
7. In Google, on the Credentials page copy the "Client secret" value and the "Client ID" value.
8. In Oracle Analytics, paste the Google "Client secret" value into the **Client Secret** field.
9. In Oracle Analytics, paste the Google "Client ID" into the **Client ID** field.
10. In the Google application, copy the "Account ID" from the Account details and copy the "Property ID" from the Property details.

    In the Google administration settings, navigate to Account, then Account details to obtain the "Account ID", and navigate to Property details to obtain the "Property ID".

11. In Oracle Analytics, use the "Account ID" and "Property ID" that you copied in the previous step to specify the **Account ID** value and the **Property ID** value, then click **Authorize**.
12. In the Google prompt to authorize the connection, click **Allow**.

    The Create Connection dialog refreshes and displays the name of the Google account, and its associated email account.

13. Save the connection.

14. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to Snowflake Data Warehouse

You can create a connection to Snowflake Data Warehouse and use the connection to access data.

See format guidelines, `https://docs.snowflake.net/manuals/user-guide/connecting.html`.

1. On the Home page, click **Create**, and then click **Connection**.
2. Click **Snowflake Data Warehouse**.
3. Enter a Connection Name.
4. In **Hostname**, enter the host account name using one of the following formats:
   - For Amazon Web Services US West, use `<account>.snowflakecomputing.com`
   - For all other regions on Amazon Web Services, use `<account>.<region>.snowflakecomputing.com`
   - For all regions on Microsoft Azure, use `<account>.<region>.azure.snowflakecomputing.com`

   Where `account` is the Snowflake account name that you want to use to access the data, for example: `exampleaccountname.snowflakecomputing.com`.
5. For **Username** and **Password**, enter user credentials with access to the Snowflake data source.
6. For **Database Name**, enter the name of the database containing the schema tables and columns that you want to connect to.
7. For **Warehouse**, enter the name of the warehouse containing the database, schema tables and columns that you want to connect to. For example, `Example-WH`.
8. If you want data modelers to be able to use these connection details. click **System connection**. See Database Connection Options.
9. Click **Save**.
10. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connect to OCI Data Flow SQL Endpoints

OCI Data Flow SQL Endpoints enables business analysts and data scientists to analyze both structured and unstructured data in object storage with high performance and scalability.

OCI Data Flow SQL Endpoints enable you to analyze large volumes of event and time-series data in-place on the data lake without having to move and summarize it for performance.

Topics:

- Overview to Analyzing OCI Data Flow SQL Endpoints
- Download JDBC Connection Details for Data Flow SQL Endpoints to a JSON File
- Create a Connection to OCI Data Flow SQL Endpoints

For general information about OCI Data Flow SQL Endpoints, see SQL Endpoints in the Oracle Cloud Infrastructure documentation.

Overview to Analyzing OCI Data Flow SQL Endpoints

You use Oracle Analytics Cloud to analyze data from OCI Data Flow SQL Endpoints in object storage, data lakes, and applications.

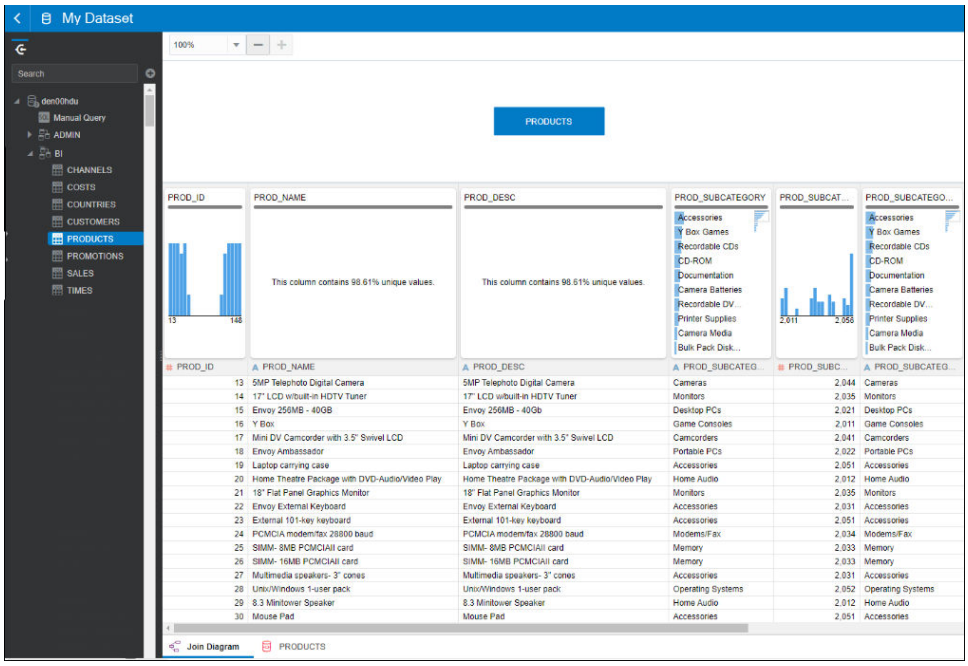
Data Flow SQL Endpoints are designed for developers, data scientists, and advanced analysts to interactively query data directly where it lives in a data lake.

Benefits of Using OCI Data Flow SQL Endpoints

- You can analyze large volumes of event and time-series data in-place on the data lake without having to move and summarize it for performance.
- You can consolidate data from multiple applications and data stores (for example, in Enterprise Resource Planning) into object storage and perform ad hoc queries regardless of where the data originates.
- You can dispense with extracts and pre-aggregation and work on live data at any level of granularity. So not only can you save the time and effort when preparing the data, you have more powerful analysis capabilities.

Best Practices for Performance

-



  To take advantage of the indexing and caching at the Spark Cluster tier, create a dataset based on a single table or view. Datasets based on multi-table joins are supported, but not recommended.

- When you configure the OCI Data Flow SQL Entpoints cluster, set `incrementalCollect` to true, for example:
  `spark.sql.thriftServer.incrementalCollect=true;`

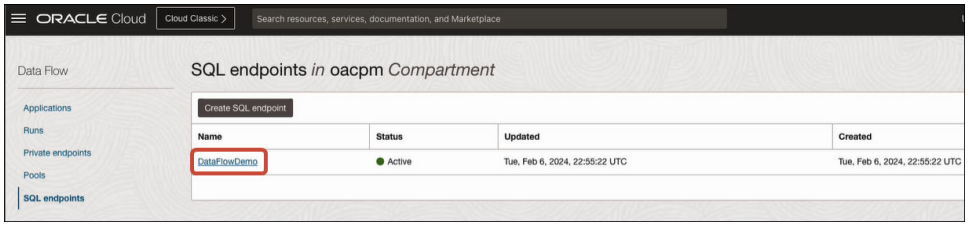
Visualizing Data From OCI Data Flow SQL Endpoints

In the Oracle Analytics Cloud workbook editor, add multiple OCI Data Flow SQL Endpoints tables or cubes. When you select a table or cube, you can add dimension columns and measure columns to your datasets for analysis.

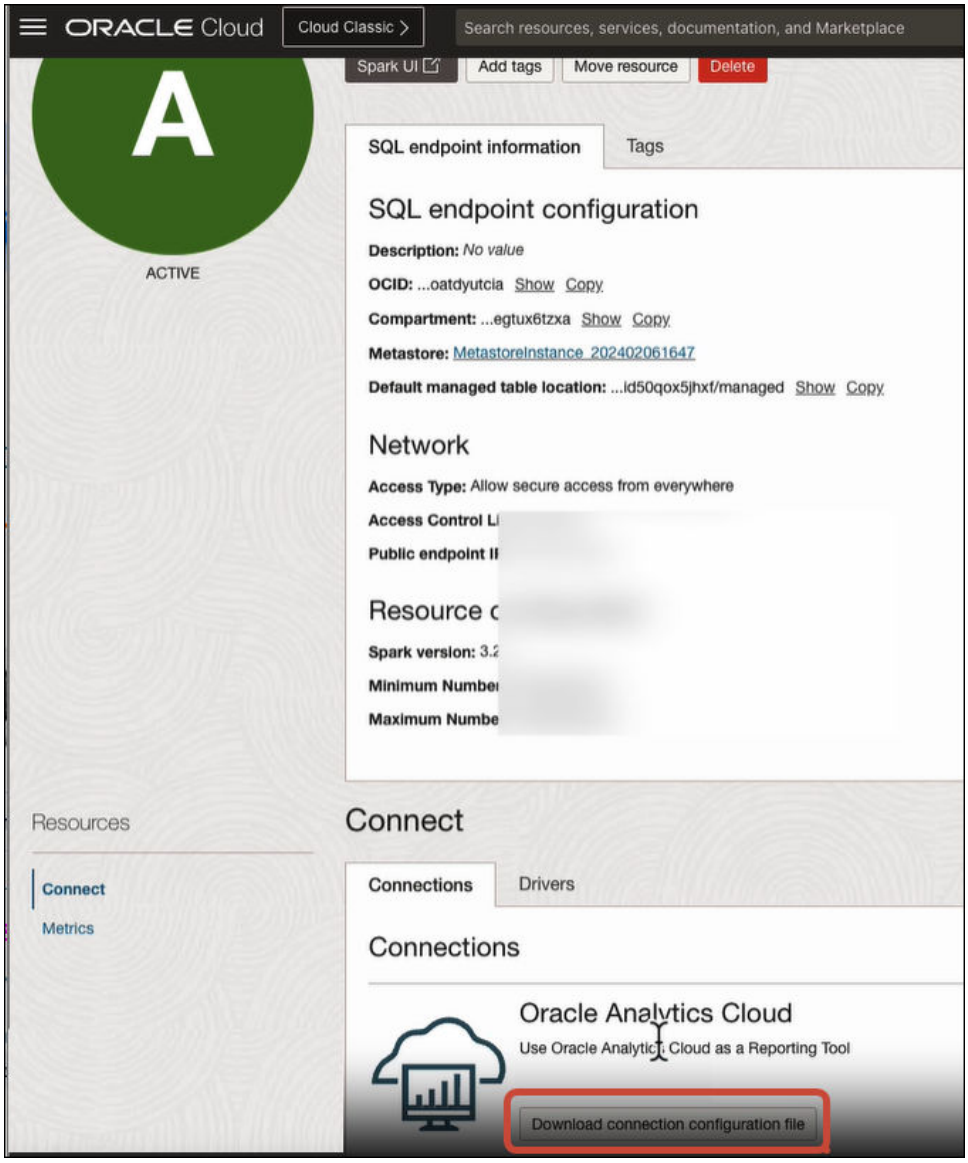
Download JDBC Connection Details for Data Flow SQL Endpoints to a JSON File

Download and configure JDBC connection details in a JSON file to use when you connect to OCI Data Flow SQL Endpoints from Oracle Analytics.

1. In OCI, navigate to Data Flow, SQL endpoints, and click the name of your data flow SQL endpoint.



2. Scroll down to the Connect area, and under Connections - Oracle Analytics Cloud, click **Download connection configuration file**.



The connection configuration file is downloaded and saved to your local area.

You're now ready to connect to your OCI Data Flow SQL Endpoints data source in Oracle Analytics. See Create a Connection to OCI Data Flow SQL Endpoints.

Create a Connection to OCI Data Flow SQL Endpoints

You can create a connection to a OCI Data Flow SQL Endpoints and use the connection to visualize data.

Before you start, in OCI Console download a JSON file containing connection details for the OCI tenancy where the OCI data flow is located. See Download JDBC Connection Details for Data Flow SQL Endpoints to a JSON File. In addition, copy an API Key from the User Tenancy area (in a Privacy Enhanced Mail (PEM) file).

1. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
2. Click **OCI Data Flow**.

3. In **Connection Name**, specify a user-friendly name to identify the connection in Oracle Analytics.
4. In **Connection Details**, click **Select**, navigate to your JDBC connection JSON file that you downloaded, then click **Open**.

   Oracle Analytics uses the JSON file to populate the **Host**, **Database**, **User OCID**, **Tenancy OCID**, and **Region** fields.
5. In **Private API Key**, click **Select**, navigate to your PEM file containing the API Key, then click **Open**.

   Oracle Analytics uses the PEM file to populate the **API Key Fingerprint** field.
6. Save the details.

You're now ready to create a workbook and start visualizing your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created and create a workbook.

Connect to Data from REST Endpoints

You can connect to data sources with REST endpoints and analyze the data. For example, connect to SaaS or PaaS applications or government data such as weather, spatial, or census data.

Connecting to data via REST endpoints enables you to analyze data from many transactional SaaS or PaaS applications without having to understand the internal format or structure of the data.

1. Store the data source's REST endpoint connection details in a JSON file. See Specifying REST Endpoint Connection Details in a JSON File.

   You can download example JSON files from the Oracle Analytics Public Library. See JSON Examples for Common Data Sources with REST Endpoints.
2. Create a connection in OAC by uploading the zipped JSON file. See Create a Connection to a Data Source with REST Endpoints.
3. Use the connection. For example, from the Home page, click **Create**, then **Dataset**, then select the connection that you created in Step 2.

See Troubleshooting Connection to Data Sources with REST Endpoints.

Specifying REST Endpoint Connection Details in a JSON File

Before you create a connection from Oracle Analytics Cloud to a data source's REST endpoints, create a zipped JSON file containing the connection details.

You can download a JSON template from the Oracle Analytics Public Library. Store the connection details in a JSON file based on the template, then compress the JSON file in ZIP format. You can also download example JSON files for a range of SaaS and PaaS applications.

JSON Template Format for Connection to REST Endpoints

```
{
    "name": "Connection name",
    "description": "Brief description",
    "baseURL": "URL, e.g. https://api.surveymonkey.com/v3",
    "endpoints": {
```

```
        "Endpoint 1": "Endpoint 1 URL, e.g. https://api.surveymonkey.com/v3/
users/endpoint1",
        "Endpoint 2": "Endpoint 2 URL, e.g. https://api.surveymonkey.com/v3/
question_bank/endpoint2",
        "Endpoint n": "Endpoint n URL, e.g. https://api.surveymonkey.com/v3/
users/endpointn"
    },
    "authentication": {
        "type": "Authentication type"
    }
}
```

JSON Example for Connection to Survey Monkey REST Endpoints

```
{
    "name": "Survey Monkey Connection",
    "description": "Survey Monkey connection",
    "baseURL": "https://api.surveymonkey.com/v3",
    "endpoints": {
        "Users": "https://api.surveymonkey.com/v3/users/me",
        "Questions": "https://api.surveymonkey.com/v3/question_bank/questions"
    },
    "authentication": {
        "type": "HttpHeader"
    }
}
```

About OAuth2 Authentication

Before you start, make sure that your target data source supports OAuth2 authentication; not all data sources do support OAuth2.

When you create the JSON file using one of the downloaded templates, specify the appropriate authentication type in the `authentication-type` field.

OAuth2 Types	Valid Values for the authentication-type field.
OAuth2 Code	OAuth2Code
OAuth2 Password Credentials	OAUTH2PasswordGrant
OAuth2 Implicit	OAUTH2ImplicitGrant
OAuth2 Client Credentials	OAUTH2ClientCredentials

In this example, the OAuth2 Password Credentials authentication method is specified using "OAUTH2PasswordGrant" as the `authentication-type` value.

```
{
    "name": "Name of the datasource",
    "description": "Description about datasource",
    "baseURL": "https://companyname.com",
    "endpoints": {
        "endpointExample1":"/example1",
        "endpointExample2":"/example2"
    },
    "authentication": {
        "//OAuth type": "OAuth2Code or OAUTH2ClientCredentials or OAUTH2ImplicitGrant or OAUTH2PasswordGrant",
        "//Others": "noauth or HttpHeader or BasicAuth",
        "type": "OAUTH2PasswordGrant"
    }
}
```

Other tips on creating the JSON file

- Remove unnecessary parameters and values from the file.
- Make sure that the Authentication type value is set to noauth, BasicAuth, HttpHeader, or one of the OAuth2 settings in the table above.

Create a Connection to a Data Source with REST Endpoints

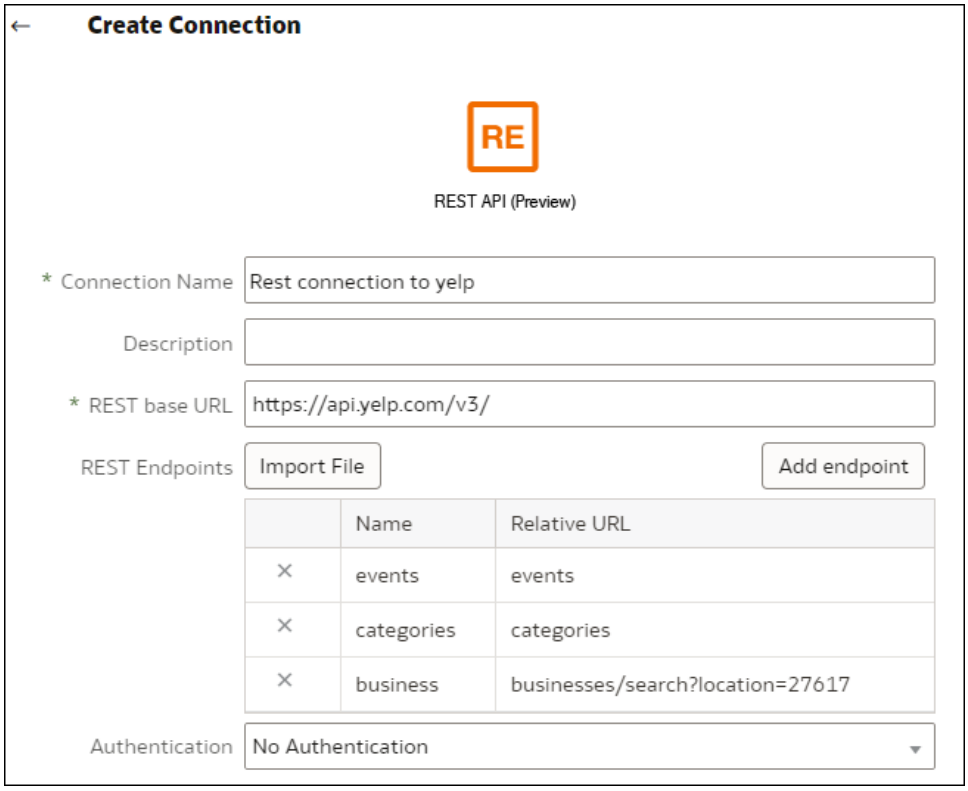
You can connect to data sources with REST endpoints and analyze the data. For example, connect to SaaS or PaaS applications or government data such as weather, spatial, or census data.

Connecting to data using REST endpoints enables you to analyze data from many transactional SaaS or PaaS applications without having to understand the internal format or structure of the data.

Before you start, create a JSON file for the data source you're connecting to. See Specifying REST Endpoint Connection Details in a JSON File.

1. On the Oracle Analytics Home page, click **Create**, then click **Connection**, then click **REST API (Preview)**.
2. In **Connection Name**, specify a user-friendly name to identify this connection in Oracle Analytics.
3. Click **Import File** and select the REST connector ZIP file that you created as a prerequisite.

   The **Description** and **REST base URL** fields are populated from the imported file, and the endoint table displays the **Name** and **Relative URL** for each available endpoint.



4. Optional: Edit the endpoints to match your business requirements. For example, you might delete endpoints that you don't need.
   - To edit an endpoint, double-click the **Name** or **Relative URL** value in the table and edit the text.
   - To add an endpoint, click **Add endpoint** to add a row to the table, and edit the default name and relative URL.
   - To remove an endpoint, click **Delete Row** next to the endpoint (that is, **X**).
5. In **Authentication**, select how you want to protect the connection.

   Tip: Make sure that the authentication type you select matches the authentication-type specified in the uploaded JSON file. See Specifying REST Endpoint Connection Details in a JSON File.
   - **No Authentication** - Allow connection without authentication. Use this option to connect to public endpoints.
   - **Basic** - authenticate the connection with a username and password.
   - **HttpHeader** - Authenticate the connection with a a security token.
   - **OAuth2 Code** - Connect to a client using an authorization code generated in your target application (known as the Authorization Code grant type). This is the most secure OAuth2 connection type.
   - **OAuth2 Password Credentials** - Connect to a trusted client using a password, (known as the Password Credentials or Resource Owner Password grant type). Use this connection type if you're using a trusted client.
   - **OAuth2 Implicit** - Connect to a client using a public code generated in your target application (known as the Authorization Code grant type). This is less secure than the OAuth2 Code connection type, but is easier to implement.

- **OAuth2 Client Credentials** - Connect to a client using a token, (known as the Client Credentials grant type).

For guidance on specifying OAuth2 connection details, see OAuth2 Authentication Values for REST Enabled Data Sources.

6. Click **Save**.

OAuth2 Authentication Values for REST Enabled Data Sources

When you connect to a REST enabled data source using one of the OAuth2 authentication types (that is, **OAuth2 Code**, **OAuth2 Password Credentials**, **OAuth2 Implicit**, or **OAuth2 Client Credentials**), you're prompted to specify connection details appropriate for authentication type that you use.

Connection dialog field or option	Description
Authorize	Click **Authorize** to test the connection and request necessary codes and tokens.
Auth URL	Enter the URL for the authentication page in the target application. For example, `https://example.com/login/oauth/authorize`.
Client ID	Enter the Client ID that you copy from your target application, such as Chimp, which is typically a string of integers and letters.
Client Secret	Enter the Client Secret that you copy from your target application, such as Chimp, which is typically a string of integers and letters.
Password	Enter the password used to log into your target application.
Scope	Enter `read:` or `write:` followed by the name of the target. For example, `read:org`.
Redirect URL	This is read-only field.
Token URL	Enter the authorization URL provided by the target application. For example, `https://example.com/login/oauth/access_token`.
Username	Enter the username used to log into your target application.

Troubleshooting Connection to Data Sources with REST Endpoints

Here're some tips on addressing issues you might encounter when you connect to REST endpoints.

Creating OAC Connections to data sources with REST endpoints

- **Failed to Import the file - Invalid JSON file provided**
  1. Extract the connection.json file from the uploaded ZIP file.
  2. Validate the JSON using any JSON validator, and fix any syntax errors.
  3. Re-create the connection ZIP file and retry the uploading using the **Import file** option on the Create Connection dialog.
- **Invalid REST Base URL** - Check the base URL using HTTP or CURL by appending each endpoint one at a time.
- **Failed to Import the file - Invalid Authentication Types** - In the JSON file, make sure that the `Authentication type` value is set to `noauth`, `BasicAuth`, or `HttpHeader`.
- **One Endpoint Table is Invalid** - Validate each endpoint URL using HTTP or CURL, and correct any errors.

- **Forbidden Access to some of the end points** - Using CURL or HTTP, validate each endpoint by appending a BASE URL to it using same user credentials. Provide access to any endpoints that don't have access or remove the endpoints from the JSON file.
- **Endpoint URL is invalid** - Using CURL or HTTP, validate each endpoint by appending a BASE URL to it. Correct any invalid endpoints or remove the endpoints from the JSON file.
- **Invalid Username/Password** - Using CURL or HTTP, validate the credentials for each endpoint.
- **Invalid JSON Response from REST data source** - Using HTTP or CURL, connect to the REST data source, extract the response received, and validate the response using a JSON validator. If necessary, contact the administrator of the data source to correct data issues.
- **URI too long** - Make sure that the URIs are no more than 8000 characters in length.

Connect to Remote Data Using Generic JDBC

You can connect to remote on-premises databases using generic JDBC connections.

> **Note:**
>
> - Oracle doesn't manage the license or usage compliance for any JDBC drivers that you deploy in your on-premises environment.
> - Oracle can't resolve issues with generic JDBC connections if you use JDBC drivers that aren't certified.
> - Oracle Analytics might be unable to list database objects with some JDBC drivers.

Before you start, check with your service administrator that remote connectivity is enabled in Oracle Analytics and that Data Gateway is installed on the system hosting your remote data source.

Check the driver documentation and the JAR file for specifying the URL of your JDBC data source. Avoid using instance-specific connection names such as host names, because the same connection can be configured against different databases in different environments (for example, development and production). When you create a connection using JDBC, the **Use Remote Data Connectivity** option is selected by default and greyed out because you can't use generic JDBC connections for local data sources.

1. Download the JDBC driver JAR file that you're deploying.
2. Navigate to the `\<Data Gateway installation>\` folder and copy in the JAR file that you downloaded in Step 1.
   - In a server deployment, copy the JAR file into: `<Data Gateway install_location>/domain/jettybase/thirdpartyDrivers`.
   - In a personal deployment on Windows, copy the JAR file into: `<Data Gateway_extract_path>\thirdpartyDrivers`.
   - In a personal deployment on MacOS, copy the JAR file into: `<Application->Show Package Contents>Resources->app.nw-> thirdpartyDrivers`.
3. Restart the Data Gateway agent.

4. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
5. Click **JDBC**.
6. Enter the URL of your JDBC data source in the **Host** field.
7. Enter the driver class name contained in the JAR file or from the download location in the **Driver Class** field.
8. Enter user credentials with access to the data source in the **Username** and **Password** fields.
9. Save the details.
10. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

Connecting to Data Sources Using Kerberos Authentication

You can connect from Oracle Analytics to Spark, Hive, and Impala using Kerberos.

Topics:

Create the Archive File Needed for a Database Connection with Kerberos Authentication

To connect to a data source using Kerberos authentication, you can supply connection details to Oracle Analytics using connection details stored in an archive file (for example, a compressed file with a *.zip file extension).

Storing connection details in an archive file (that is, a compressed file with a *.zip file extension) makes it easy to connect to a Spark or Hive data source without having to manually enter connection details.

The archive file requires a directory named kerberos that contains the following files :

- `kerberos/krb5conf`
- `kerberos/oac.keytab`
- `kerberos/service_details.json`

The `service_details.json` file contain values for Host, Port, and ServicePrincipalName with parameter values in quotation marks (`"value"`). For example:

```
{
 "Host" : "myHost.com",
 "Port" : "10000",
 "ServicePrincipalName" : "hive/myHostDB.com@BDA.COM"
}
```

1. Obtain the Kerberos configuration files from your database administrator, for example, to connect to Apache Hive.
2. Create a folder named `kerberos` to contain the Kerberos configuration files.
3. Copy the `krb5conf` file into the `kerberos` folder that you created.

4. Ensure the `.keytab` file is named `oac.keytab` (rename it if required), and copy the file into the folder you created.
5. Get or create the `service_details.json` file and save it in the folder you created.
6. Create an archive file containing the three files that you added to your folder, and provide an appropriate name, for example `SSLKerberos.zip`.

Connect to a Spark or Hive Database Using Kerberos Authentication

You can connect to a Spark or Hive database using Kerberos network authentication protocol.

Before you start, store the Kerberos connection details in an archive file (that is, a compressed file with a *.zip file extension).

To find out which database types support Kerberos authentication, look for databases with 'Supports Kerberos' in the **More Information** column in the Supported Data Sources list. See Supported Data Sources.

1. On the Home page, click **Create**, and then click **Connection**.
2. Select a Hive connection type (such as **Apache Hive** or **Hortonworks Hive**) or a Spark connection type .
3. Click **Authentication Type** and select **Kerberos**.
4. In the **Client Credentials** field, either drag and drop or click **Select** to browse for a prepared archive or CONF file.

   Do one of the following to get the appropriate configuration files for a SSL or a Non-SSL connection:

   - Ask your administrator to provide the appropriate archive or CONF files.
   - Prepare your own archive file.
5. If you added an archive file, enter the archive file password in the **ZIP Password** field.
6. If you added a `krb5conf` file, either drag and drop or click **Select** to browse for the `oac.keytab` file in the **Keytab** field.

   The **Host**, **Port**, and **Service Principal** fields automatically display values taken from the `service_details.json` file.
7. If you're connecting to an on-premises database, click **Use Remote Data Connectivity**.

   Your administrator can enable this checkbox in the Console.

   Check with your administrator that you can access the on-premises database.
8. If you're connecting to your data using SSL, click **Enable SSL**.
9. If you want data modelers to be able to use these connection details. click **System connection**. See Database Connection Options.
10. Click **Save**.

Connect to Oracle Service Cloud

Connect to an Oracle Service Cloud data source to visualize CRM data.

1. On the Home page, click **Create** and then click **Connection**.
2. Click **Oracle Service Cloud** and enter the connection details.

3. Save the details.
4. Use the connection to connect to your data. For example, from the Home page, click **Create**, then click **Dataset**, then select the connection that you just created.

4

Connect to Data for Pixel-Perfect Reports

This topic describes how to connect to data sources for creating pixel-perfect reports using Publisher.

Topics:

Overview to Connecting to Data for Pixel-Perfect Reports

You can use a variety of data sources for pixel-perfect reports.

The data can come from:

- Databases
- HTTP XML feeds
- Web Services
- Oracle BI Analyses
- OLAP cubes
- LDAP servers

You can connect to on-premises data sources over a Private Access Channel (PAC). See Connect to Private Data Sources Through a Private Access Channel.

About Private Data Source Connections

Private connections for OLAP, JDBC, Web Service, and HTTP data sources are supported in Oracle BI Publisher and can be created by users with data model creation privileges.

When you create a private data source connection, the private data source connection is available only to you in the data model editor data source menus.

Administrators have access to the private data source connections created by users. All private data source connections are displayed to Administrators when they view the list of OLAP, JDBC, Web Service, and HTTP data sources from the Administration page.

Private data source connections are distinguished by an **Allowed User** value on the Data Source Administration page. Administrators can extend access to other users to a private data source connection by assigning additional user roles to it.

For more information on assigning roles to data sources, see Grant Access to Data Sources Using the Security Region.

Grant Access to Data Sources Using the Security Region

When you set up data sources, you can also define security for the data source by selecting which user roles can access the data source.

You must grant access to users for the following:

- A report consumer must have access to the data source to view reports that retrieve data from the data source.
- A report designer must have access to the data source to create or edit a data model against the data source.

By default, a role with administrator privileges can access all data sources.

The configuration page for the data source includes a Security region that lists all the available roles. You can grant roles access from this page, or you can also assign the data sources to roles from the roles and permissions page.

About Proxy Authentication

Oracle BI Publisher supports proxy authentication for connections to various data sources

Supported data sources include:

- Oracle 10*g* database
- Oracle 11*g* database
- Oracle BI Server

For direct data source connections through JDBC and connections through a JNDI connection pool, Oracle BI Publisher enables you to select "Use Proxy Authentication". When you select Use Proxy Authentication, Oracle BI Publisher passes the user name of the individual user (as logged into Oracle BI Publisher) to the data source and thus preserves the client identity and privileges when the Oracle BI Publisher server connects to the data source.

Enabling this feature requires additional setup on the database. The database must have Virtual Private Database (VPD) enabled for row-level security.

For connections to the Oracle BI Server, Proxy Authentication is required. In this case, proxy authentication is handled by the Oracle BI Server, therefore the underlying database can be any database supported by the Oracle BI Server.

Choose JDBC or JNDI Connection Type

In general, a JNDI connection pool is recommended because it provides the most efficient use of your resources.

For example, if a report contains chained parameters, then each time the report is processed, the parameters initiate to open a database session every time.

About Backup Databases

When you configure a JDBC connection to a database, you can also configure a backup database.

A backup database can be used in two ways:

- As a true backup when the connection to the primary database is unavailable.
- As the reporting database for the primary. To improve performance you can configure your report data models to run against the backup database only.

To use the backup database in either of these ways, you must also configure the report data model to use it.

About Connection Creation and Closure Functions

You can define PL/SQL functions for Publisher to run when a connection to a JDBC data source is created (preprocess function) or closed (postprocess function).

The function must return a Boolean value. This feature is supported for Oracle databases only.

These two fields enable the administrator to set a user's context attributes before a connection is made to a database and then to dismiss the attributes after the connection is broken by the extraction engine.

The system variable :xdo_user_name can be used as a bind variable to pass the login username to the PL/SQL function calls. Setting the login user context in this way enables you to secure data at the data source level (rather than at the SQL query level).

For example, assuming the following sample function:

```
FUNCTION set_per_process_username (username_in IN VARCHAR2)
   RETURN BOOLEAN IS
     BEGIN
     SETUSERCONTEXT(username_in);
     return TRUE;
   END set_per_process_username
```

To call this function every time a connection is made to the database, enter the following in the **Pre Process Function** field: set_per_process_username(:xdo_user_name)

Another sample usage might be to insert a row to the LOGTAB table every time a user connects or disconnects:

```
CREATE OR REPLACE FUNCTION BIP_LOG (user_name_in IN VARCHAR2, smode IN
VARCHAR2)
RETURN BOOLEAN AS
   BEGIN
   INSERT INTO LOGTAB VALUES(user_name_in, sysdate,smode);
   RETURN true;
   END BIP_LOG;
```

In the **Pre Process Function** field enter: BIP_LOG(:xdo_user_name)

As a new connection is made to the database, it is logged in the LOGTAB table. The SMODE value specifies the activity as an entry or an exit. Calling this function as a **Post Process Function** as well returns results such as those shown in the table below.

NAME	UPDATE_DATE	S_FLAG
oracle	14-MAY-10 09.51.34.000000000	AMStart
oracle	14-MAY-10 10.23.57.000000000	AMFinish
administrator	14-MAY-10 09.51.38.000000000	AMStart
administrator	14-MAY-10 09.51.38.000000000	AMFinish
oracle	14-MAY-10 09.51.42.000000000	AMStart
oracle	14-MAY-10 09.51.42.000000000	AMFinish

Set Up a JDBC Connection to a Data Source

You can set up a JDBC connection to a data source.

1. From the Administration page, click **JDBC Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source in the **Data Source Name** field. This name is displayed in the Data Source selection list in the Data Model Editor.

   You can't create a new Oracle BI EE data source with the same name, nor can you delete the provisioned Oracle BI EE data source.
4. Select the driver type.
5. Select **Use Data Gateway** only if you want to connect to a remote data source.

   Your administrator must enable remote data connectivity and configure Data Gateway on your target on-premises database. If you select **Use Data Gateway**, the **Database Driver Class**, **Use System User**, **Pre Process Function**, **Post Process Function**, and **Use Proxy Authentication** settings aren't available for selection or update.
6. You can update the **Database Driver Class** field if required.
7. Enter the database connection string.

   Example connection strings:

   - Oracle database

     To connect to an Oracle database (non-RAC), use the following format for the connection string:

jdbc:oracle:thin:@[host]:[port]:[sid]

For example: jdbc:oracle:thin:@myhost.us.example.com:1521:prod

- Oracle RAC database

  To connect to an Oracle RAC database, use the following format for the connection string:

  jdbc:oracle:thin:@//<host>[:<port>]/<service_name>

  For example: jdbc:oracle:thin:@//myhost.example.com:1521/my_service

- Microsoft SQL Server

  To connect to a Microsoft SQL Server, use the following format for the connection string:

  jdbc:hyperion:sqlserver://[hostname]:[port];DatabaseName=[Databasename]

  For example: jdbc:hyperion:sqlserver://
  myhost.us.example.com:7777;DatabaseName=mydatabase

8. Enter the user name and password required to access the data source.
9. Optional: Enter a PL/SQL function to execute when a connection is created (Pre Process) or closed (Post Process).
10. Optional: Specify a client certificate for secured connection.

    The client certificates uploaded in Upload Center are listed for selection.
11. To enable Proxy Authentication, select **Use Proxy Authentication**.
12. Click **Test Connection**.
13. Optional: Enable a backup database for this connection:
    a. Select **Use Backup Data Source**.
    b. Enter the connection string for the backup database.
    c. Enter the user name and password for this database.
    d. Click **Test Connection**.
14. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

    When you set up a JDBC connection to Oracle BI EE data source, make sure you move the **BI Consumer** role from the **Available Roles** list to the **Allowed Roles** list.

    If you defined a backup data source, the security settings are passed to the backup data source.

Set Up a Secure JDBC Connection to Oracle Autonomous Data Warehouse

You can create a secure JDBC connection to Oracle Autonomous Data Warehouse.

Upload a JDBC client certificate and create an SSL based JDBC connection to Oracle Autonomous Data Warehouse.

1. Upload the JDBC client certificate (Oracle wallet file, cwallet.sso) to the server.
   a. From the Publisher Administration page, click **Upload Center**.
   b. Browse and select the Oracle wallet file, cwallet.sso.

c. Select **JDBC Client Certificate** from the **File Type** list.

d. Click **Upload**.

2. From the Publisher Administration page, click **JDBC Connection**.
3. Click **Add Data Source**.
4. Specify the following details for the connection:
   - **Data Source Name**: DBaaSConnection
   - **Driver Type**: Oracle 12c
   - **Database Driver Class**: oracle.jdbc.OracleDriver
5. Enter the JDBC connection string.

   Use TCPS strings. For example,
   `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=`*`server_name`*`)(PORT=`*`port`*`))(CONNECT_DATA=(SERVICE_NAME=`*`serviceName`*`)))`

   If you're using PAC (Private Access Channel), add `(ENABLE=broken)` to the DESCRIPTION parameter in the connect string. For example,
   `jdbc:oracle:thin:@(DESCRIPTION=(ENABLE=broken)(ADDRESS=(PROTOCOL=tcps)(HOST=`*`server_name`*`)(PORT=`*`port`*`))(CONNECT_DATA=(SERVICE_NAME=`*`serviceName`*`)))`
6. From the **Client Certificate** list, select the wallet file, cwallet.sso uploaded earlier.
7. Click **Test Connection**.
8. Click **Apply**.

Set Up a JDBC Connection to an On-premises Data Source

You can create a JDBC connection to an on-premises data source by using a data gateway agent.

Ensure that your administrator configures Data Gateway on your target on-premises database and enables data connectivity. See Overview to Connecting to On-premises Data Sources.

1. Enable Data Gateway in Console:

   a. From the Analytics Cloud Home page, click **Console**.

   b. Click **Remote Data Connectivity**.

   c. Enable the **Enable Data Gateway** option.

   d. Select and enable the data gateway agent you want to use.
2. From the Publisher Administration page, click **JDBC Connection**.
3. Click **Add Data Source**.
4. Enter a display name for the data source in the **Data Source Name** field. This name is displayed in the Data Source selection list in the Data Model Editor.
5. From the **Driver Type** list, select the driver for the database you want to connect. For example, select Oracle 12c for Oracle Database.
6. Select **Use Data Gateway**.

   When you select **Use Data Gateway**, the following settings aren't available for selection or update.
   - **Database Driver Class** (Default: oracle.jdbc.OracleDriver)
   - **Use System User**

- **Pre Process Function**
- **Post Process Function**
- **Client Certificate**
- **Use Proxy Authentication**

7. Enter the connection string for the database.
8. Enter the user name and password required to access the data source.
9. Click **Test Connection**.
10. (Optional) Enable a backup database for this connection:
    a. Select **Use Backup Data Source**.
    b. Enter the connection string for the backup database.
    c. Enter the user name and password for this database.
    d. Click **Test Connection**.
11. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

    If you have defined a backup data source, the security settings are passed to the backup data source.

Set Up a Connection to a Snowflake Data Warehouse

You can create a connection to Snowflake Data Warehouse and use the connection to access data for pixel-perfect reports.

1. From the Publisher Administration page, click **JDBC Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source in the **Data Source Name** field. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select **Snowflake** as the driver type.
5. In the **Database Driver Class** field, use the default **net.snowflake.client.jdbc.SnowflakeDriver**.
6. In the Connection String field, enter the following string:

   jdbc:snowflake://*accountName*.snowflakecomputing.com;db=*database name)*;warehouse=*(warehouse name)*;schema=*(schema name)*;

   If you want other properties for the connection, add the properties separated by semicolon (;) as shown in the example.

   For example: `jdbc:snowflake://hw11692.us-central1.gcp.snowflakecomputing.com;db=SNOWFLAKE_SAMPLE_DATA;warehouse=COMPUTE_WH;useProxy=true;proxyHost=www-proxy-adcq7-new.us.oracle.com;proxyPort=80`
7. Enter the user name and password required to access the data source.
8. Optional: Enter a PL/SQL function to execute when a connection is created (Pre Process) or closed (Post Process).
9. Optional: Specify a client certificate for secured connection.

   The client certificates uploaded in Upload Center are listed for selection.

10. To enable Proxy Authentication, select **Use Proxy Authentication**.
11. Click **Test Connection**.
12. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to a Vertica Data Warehouse

You can create a connection to Vertica Data Warehouse and use the connection to access data for pixel-perfect reports.

1. From the Publisher Administration page, click **JDBC Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source in the **Data Source Name** field. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select **Vertica** as the driver type.
5. In the **Database Driver Class** field, use the default **com.vertica.jdbc.Driver**.
6. In the Connection String field, enter the following string:

   jdbc:vertica://[host_name]:[port_number]/[service_name]
7. Enter the user name and password required to access the data source.
8. Optional: Enter a PL/SQL function to execute when a connection is created (Pre Process) or closed (Post Process).
9. Optional: Specify a client certificate for secured connection.

   The client certificates uploaded in Upload Center are listed for selection.
10. To enable Proxy Authentication, select **Use Proxy Authentication**.
11. Click **Test Connection**.
12. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Database Connection Using a JNDI Connection Pool

You can create a connection to database using a JNDI connection pool to access data for pixel-perfect reports.

Using a connection pool increases efficiency by maintaining a cache of physical connections that can be reused. When a client closes a connection, the connection gets placed back into the pool so that another client can use it. A connection pool improves performance and scalability by allowing multiple clients to share a small number of physical connections. You set up the connection pool in your application server and access it through Java Naming and Directory Interface (JNDI).

> **Note:**
>
> You can create JNDI connections to the user-defined data sources, but you can't create JNDI connections to the system-defined data sources. Only to create audit reports, you are allowed to create JNDI connections to the system-defined data sources to access the audit data source (AuditViewDataSource).

1. From the Publisher Administration page, click **JNDI Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Enter the JNDI name for the connection pool. For example, `jdbc/BIPSource`.
5. Select **Use Proxy Authentication** to enable Proxy Authentication.
6. Click **Test Connection**. You see a confirmation message if the connection is established.
7. Define security for this data source connection. Move the required roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to an OLAP Data Source

You can create connections to several types of OLAP databases to access data for pixel-perfect reports.

1. From the Publisher Administration page, click **OLAP Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the OLAP type.
5. Enter the connection string for the OLAP database.
6. Enter the user name and password for the OLAP database.
7. Click **Test Connection**.
8. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to a Web Service

You can create a connection to web service data source to access data for pixel-perfect reports.

If you want to use SSL connection for the web service data source, set the **Enable SSL for webservice, HTTP Datasource** runtime property to true.

Upload the SSL certificate in Upload Center before you define the SSL connection to the data source.

1. From the Publisher Administration page, click **Web Service Connection**.

2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the server protocol.
5. Enter the server name and the server port.
6. Enter the URL for the web service connection.
7. Optional: Enter the session timeout in minutes.
8. Select the security header from **WS-Security**.
   - 2002 — Enables the "WS-Security" Username Token with the 2002 namespace: `http://docs.oasis-open.org/wss/2002/01/oasis-200201-wss-wssecurity-secext-1.0.xsd`
   - 2004 — Enables the "WS-Security" Username Token with the 2004 namespace: `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText`
9. Optional: Enter the user name and password for the web service data source.
10. Optional: From the **SSL Certificate** list, select the SSL certificate you want to use for the connection.
11. If you're using a proxy-enabled server, select **Use System Proxy**.
12. Click **Test Connection**.
13. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.
14. Click **Apply**.

Set Up a Connection to an HTTP Data Source

You can create a connection to HTTP data source to build data models from XML, JSON, and CSV data over the web by retrieving data through the HTTP GET method.

If you want to use SSL connection for the HTTP data source, set the **Enable SSL for webservice, HTTP Datasource** runtime property to true.

Upload the SSL certificate in Upload Center before you define the SSL connection to the data source.

1. From the Publisher Administration page, click **HTTP Connection**.
2. Click **Add Data Source**.
3. Enter a display name for the data source. This name is displayed in the Data Source selection list in the Data Model Editor.
4. Select the server protocol.
5. Enter the server name and the server port.
6. Enter the URL context for the HTTP data source connection in the **Realm** field.

   For example, xmlpserver/services/rest/v1/reports
7. Enter the user name and password required to access the data source on the database.

8. If you want to use SSL connection, from the **SSL Certificate** list, select the SSL certificate you want to use for the data source.
9. If you're using a proxy-enabled server, select **Use System Proxy**.
10. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.

Set Up a Connection to a Content Server

You can create a connection to a Content Server to retrieve a text attachment stored inOracle WebCenter Content (earlier known as UCM) server, and display the attachment content in a pixel-perfect report.

1. From the Publisher Administration page, select the **Content Server** link.
2. Click **Add Data Source**.
3. Enter the name in the **Data Source Name** field.
4. Enter the URL in the **URI** field.
5. Enter the user name and password in the **Username** and **Password** fields, respectively.
6. Click **Test Connection**.
7. Define security for this data source connection. Move roles from the **Available Roles** list to the **Allowed Roles** list. Only users assigned the roles in the **Allowed Roles** list can create or view reports from this data source.
8. Click **Apply**.

View or Update a Connection to Data Source

You can view or update a connection to data source from the Publisher Administration page.

1. From the Publisher Administration page, select the **Data Source** type to update.
2. Select the name of the connection to view or update. All fields are editable. See the appropriate section for setting up the data source type for information on the required fields.
3. Select **Apply** to apply any changes or **Cancel** to exit the update page.

5

Manage Database Connections for Modeling Data

Administrators create and manage cloud database connections for modelling relational and non-relational data such as Essbase, Snowflake, or Oracle Enterprise Performance Management (Oracle EPM) data. Your business data doesn't have to be in one place. Connect to multiple cloud databases so that business modelers and analysts can analyze company data wherever it's stored.

Topics

Model Data in an Essbase Cube

Connect to an Essbase database so that you can model and visualize data from Essbase cubes.

You can only model Essbase data in Model Administration Tool.

Before you start, deploy Data Gateway with your Oracle Analytics instance and configure a Data Gateway agent to communicate with the remote Essbase deployment.

1. In the Physical Layer, create a database:
   a. Right-click in the Physical Layer pane and select **Create New Database**.
   b. At the Database dialog, specify a name to identify your database within Oracle Analytics.
   c. Select the **Database Type** (for example, Essbase 11), then click **OK**.
2. In the Physical Layer, create a connection pool:
   a. Right-click the new database, click **Create New Object**, then select **Database Connection Pool**.
   b. At the Connection Pool dialog, specify a name to identify the database pool within Oracle Analytics.
   c. In **Essbase Server**, specify the connection string for your Essbase server.

      For example, `http://<IP address>:<port number>/essbase/agent`.
   d. Select **Use Data Gateway**.
   e. Enter the **User name** and **Password** for the Essbase deployment.
   f. If prompted, re-enter the password for the Essbase deployment.

3. In the Physical Layer, import the Essbase metadata:
   a. Right-click the Essbase connection and select **Import Metadata**.
   b. At the Select Data Source page, click **Next**.
   c. At the Select Metadata Objects page, expand the database in the **Data source** box, select the cube you want, then click **Import selected**.

      For a large cube, the import can take two to three minutes.
   d. When the import is complete, expand the database in the **Repository View** to display the imported Essbase cube.
   e. Click **Finish**.
4. Using the physical layer that you've just created, create your business model and mapping layer, and your presentation layer.
5. Click **File**, **Cloud**, then **Publish**.
6. Create a dashboard or visualization workbook based on your Essbase cube.

   The new Essbase cube is now available as a subject area in Oracle Analytics.

   For example, in Oracle Analytics, create an analysis and in the Select Subject Area dialog you can access the new Essbase subject area. In Oracle Analytics, create a workbook and in the Add Dataset dialog, click **Subject Areas** to access the new Essbase subject area.

Model Data in Snowflake Data Warehouse

Configure your on-premises environment so that you can model data in a Snowflake database.

You can model Snowflake data in Semantic Modeler or Model Administration Tool. This task describes the process using Model Administration Tool.

Both local and remote connections to Snowflake from a semantic model require an installation of Data Gateway in order for the Model Administration Tool to import and model tables from a Snowflake data source. Configure the Data Gateway agent with a suitable driver to connect to Snowflake. Make sure that remote Data Gateway connections are available when queries are run. However, once the Snowflake tables are modeled and the semantic model is published to Oracle Analytics, for local connections, Data Gateway can be disabled or removed as it isn't used when queries are run from Oracle Analytics.

You can create either a local or a remote connection to Snowflake from the semantic model. When creating a local (not remote) connection, the connection pool in the semantic model uses a JDBC connection.

Before you start, install Data Gateway and Model Administration Tool on the same Windows computer in your on-premises environment.

1. Configure a local Data Gateway agent to facilitate a connection from the Developer Client Tool to Snowflake.
   a. Download the latest Snowflake JDBC driver (for example, in file snowflake-jdbc-3.9.0.jar).
   b. Copy the downloaded JAR file to the Data Gateway installation folder.

      In a server deployment, copy the JAR file into:

      ```
      <Data Gateway folder>/domain/jettybase/lib/ext
      ```

      In a personal deployment, copy the JAR file into:

      ```
      <install directory>\war\datagateway\WEB-INF\lib
      ```

c. Re-start Data Gateway.

2. Configure a connection to your Snowflake database.
   - For a local connection, see Create a Local Semantic Model Connection to Snowflake.
   - For a remote connection, see Create a Remote Semantic Model Connection to Snowflake:

You can now model your data using this connection.

Create a Local Semantic Model Connection to Snowflake

Connect to a local Snowflake database so that you can model Snowflake data.

1. In Model Administration Tool, enable the JDBC connection pool functionality by loading Java data sources. See Step 3 in Configure and Register Data Gateway for Reporting.
2. In Model Administration Tool, create a database and set the type to Snowflake.
3. Add a Connection Pools and specify these details on the General tab:
   - **Call Interface**: JDBC(Direct Driver).
   - **Require fully qualified table names**: Yes.
   - **Data source name**: Enter the connection string, for example: `jdbc:snowflake://xxxx.snowflakecomputing.com?db=ODEV&warehouse=xxxxxx&schema=xxxxxx`
   - **RDC Version**: Leave this field blank.
4. On the Miscellaneous tab, specify these details:
   - **JDS Server URL**: Leave this field blank (remove any entry in this field).
   - **Driver Class**: net.snowflake.client.jdbc.SnowflakeDriver.
   - **Use SQL over HTTP**: false.
5. Model your data using this connection.
6. Upload or publish your semantic model to Oracle Analytics Cloud when you have completed your model.

   Oracle Analytics Cloud connects to the Snowflake database without using Data Gateway.

You can now model your data using this connection.

Create a Remote Semantic Model Connection to Snowflake

Connect to a local Snowflake database so that you can model Snowflake data.

1. In Model Administration Tool, enable the JDBC connection pool functionality by loading Java data sources. See Step 3 in Configure and Register Data Gateway for Reporting.
2. In Model Administration Tool, create a database and set the type to Snowflake.
3. Add a Connection Pools and specify these details on the General tab:
   - **Call Interface**: JDBC(Direct Driver).
   - **Require fully qualified table names**: Yes.
   - **Data source name**: Enter the connection string, for example: `jdbc:snowflake://xxxx.snowflakecomputing.com?db=ODEV&warehouse=xxxxxx&schema=xxxxxx`
   - **RDC Version**: Set to 2.

4. On the Miscellaneous tab, specify these details:
   - **JDS Server URL**: Leave this field blank (remove any entry in this field).
   - **Driver Class**: `net.snowflake.client.jdbc.SnowflakeDriver`.
   - **Use SQL over HTTP**: true.
5. Model your data using this connection.
6. Upload or publish the semantic model to Oracle Analytics Cloud when you have completed your model.

   Note: Oracle Analytics Cloud connects to Snowflake using any configured Data Gateway agent.
7. Copy the Snowflake driver file to each Data Gateway agent installation folder.
   - In a server deployment, copy the JAR file into: `<Data Gateway install_location>/domain/jettybase/thirdpartyDrivers`.
   - In a personal deployment on Windows, copy the JAR file into: `<Data Gateway_extract_path>\thirdpartyDrivers`.
   - In a personal deployment on MacOS, copy the JAR file into: `<Application->Show Package Contents>Resources->app.nw-> thirdpartyDrivers`.
8. Re-start Data Gateway. See Maintain Data Gateway.

Model Data in Google BigQuery

Connect to a Google BigQuery database so that you can model and visualize data from Google BigQuery. You can model Google BigQuery data in Semantic Modeler or Model Administration Tool. These tasks describes the process using Model Administration Tool.

Topics

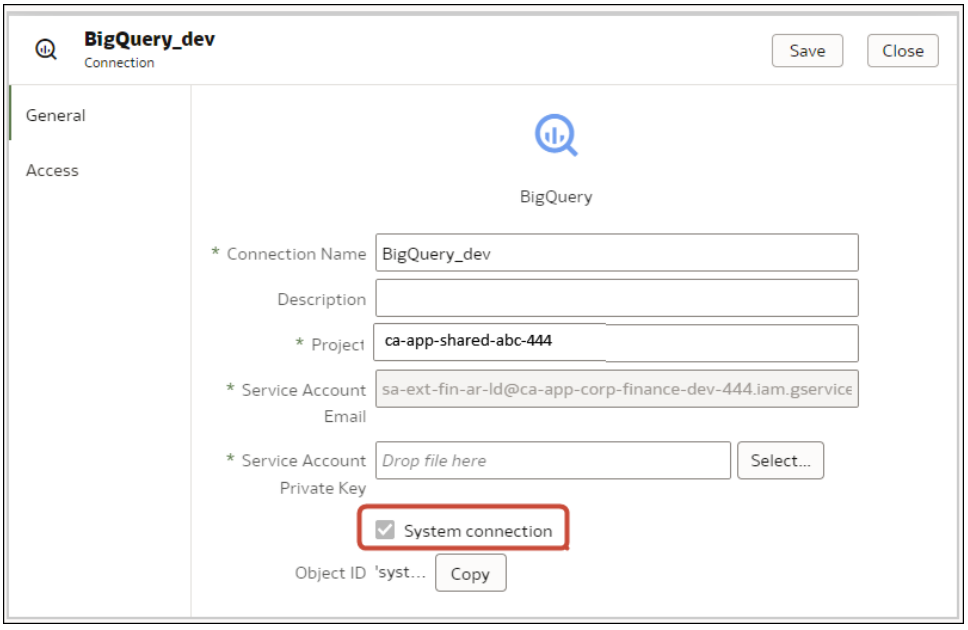
Create an Oracle Analytics Connection to Google BigQuery

You can create a system connection to a Google BigQuery database and use the connection to model the Google BigQuery project.

Before you start, download the Service Account Private Key (in JSON format) for your Google BigQuery service.

1. On the Oracle Analytics Home page, click **Create**, and then click **Connection**.
2. Click **BigQuery**.
3. Enter the connection details.
   - In **Connection Name**, specify a user-friendly name to identify the connection details in Oracle Analytics.
   - In **Project**, specify the BigQuery project name in lowercase of the project that you want to analyze.

- In **Service Account Private Key**, click **Select** and upload the Service Account Private Key (JSON format) for your BigQuery service. The **Service Account Email** is populated from the uploaded key details.
- In **System connection**, select this option.



4. Save the details.

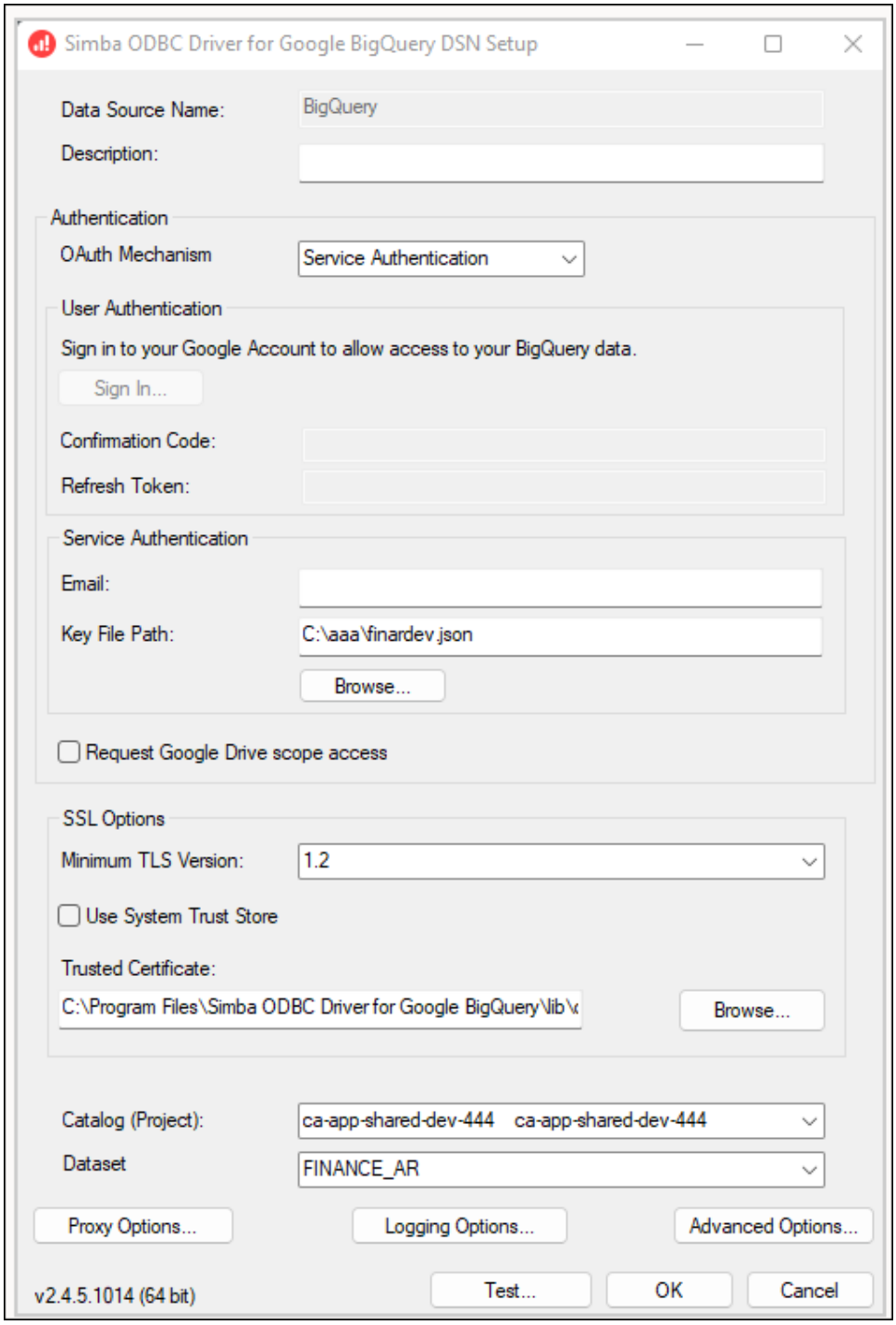
Download and Set Up BigQuery ODBC Driver

Install the ODBC driver required for connection to Google BigQuery, and configure it in Model Administration Tool to enable you to model the project.

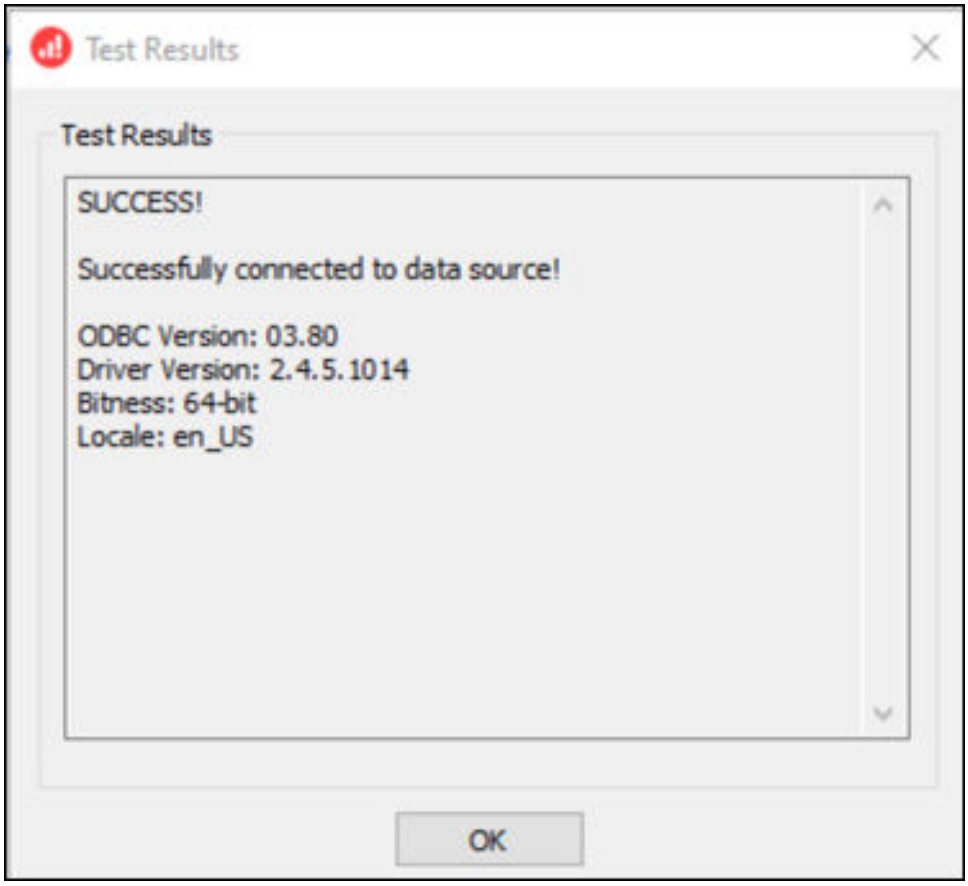
1. Download the Simba BigQuery ODBC driver from Google.

   For example, download it from the Google reference site.

2. Install the downloaded driver on the machine where Oracle Analytics Client Tools is installed.
3. Configure the ODBC driver using the DSN Setup dialog.



4. Click **Test** to test the connection.



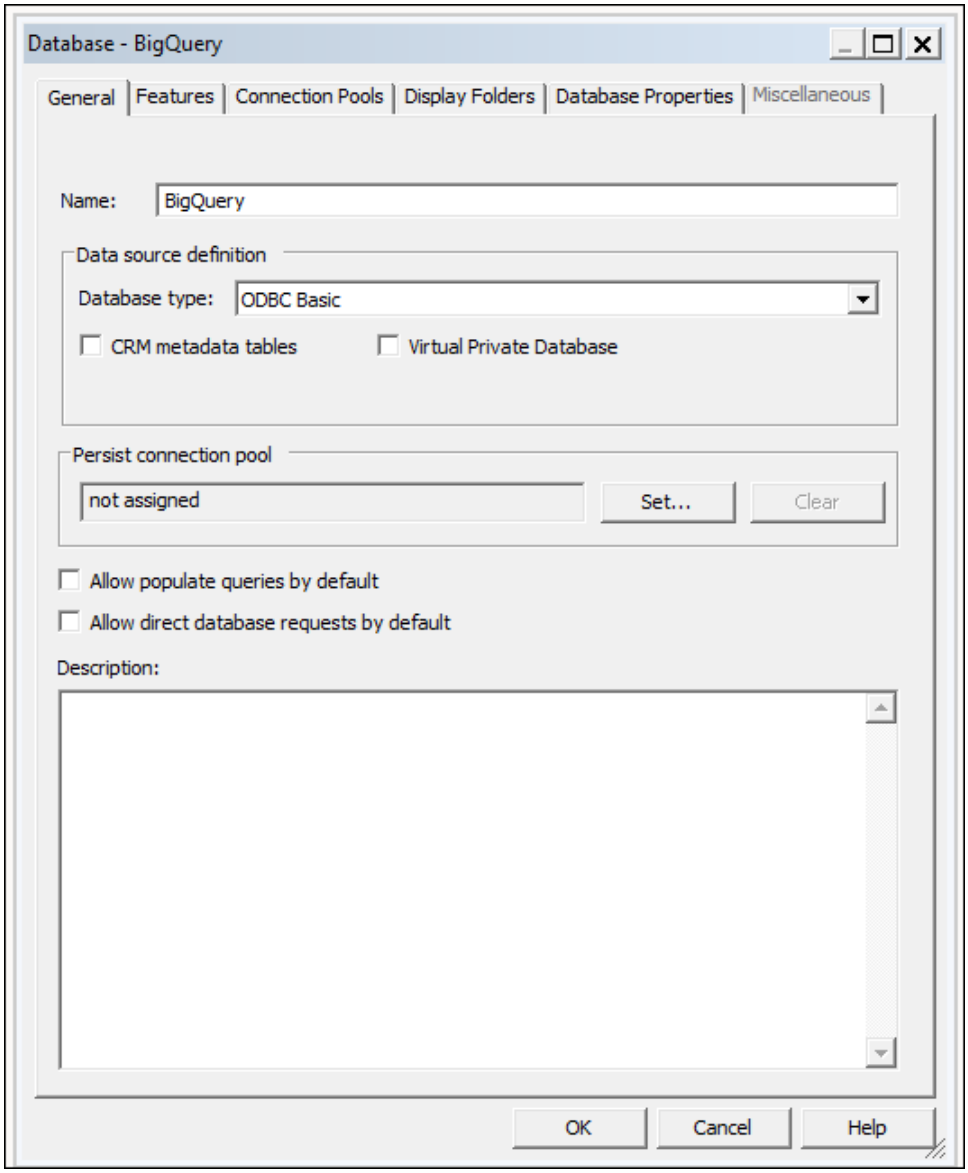
5. Save the details.

Build a Data Model from Google BigQuery Data Source

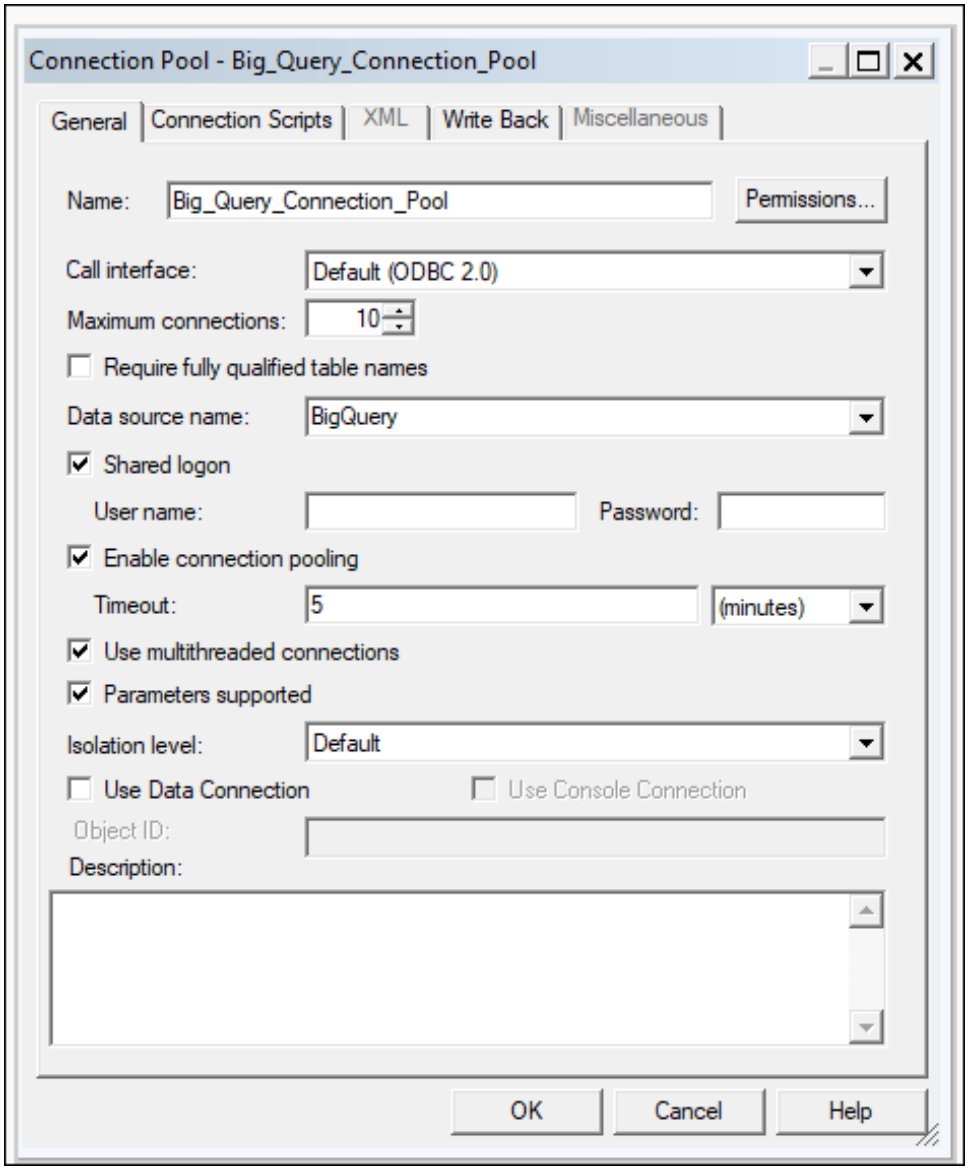
You build a data model for your Google BigQuery database so that you can deploy it to visualize data in a BigQuery project.

To build a data model, you need permissions in the BigQuery key. If the BigQuery key grants access to the dataset level, simply perform Import Metadata using the BigQuery ODBC driver by following the steps below. If the BigQuery key grants access to only specific tables or views, follow the steps below to create a physical schema.

1. In Model Administration tool, create a database in the repository and set the **Database type** to ODBC Basic.

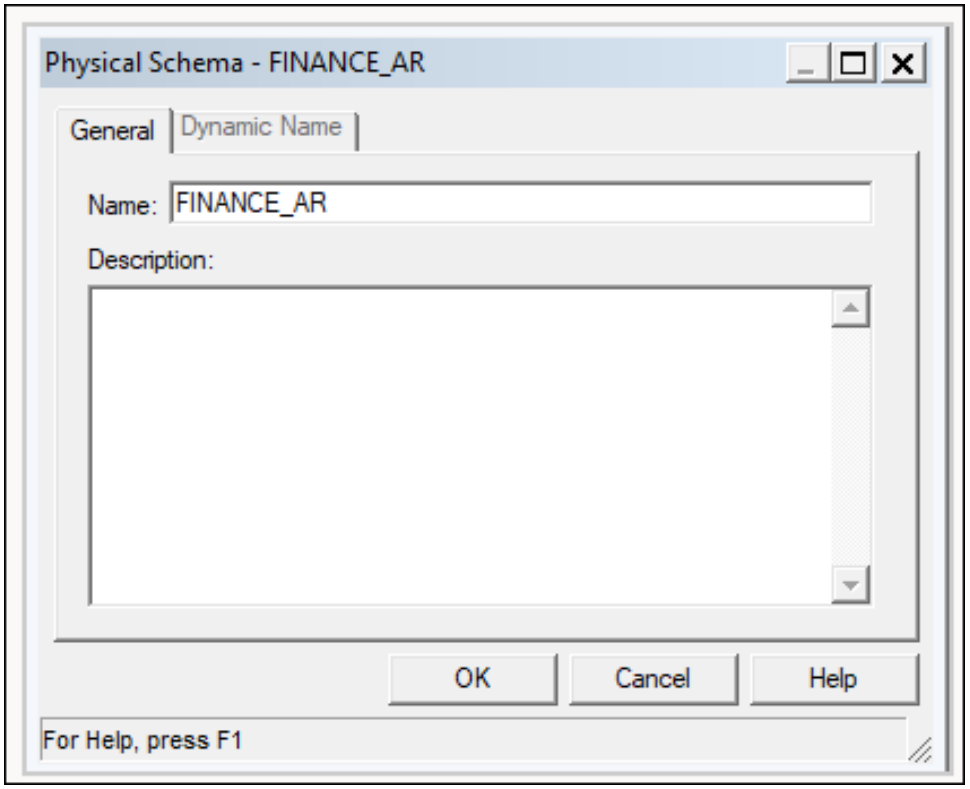


2. In the Connection Pools dialog, create a connection pool in the database.
   - In **Call interface**, select "Default (ODBC 2.0)".
   - In **Data source name** field, select the BigQuery ODBC driver that you created earlier.

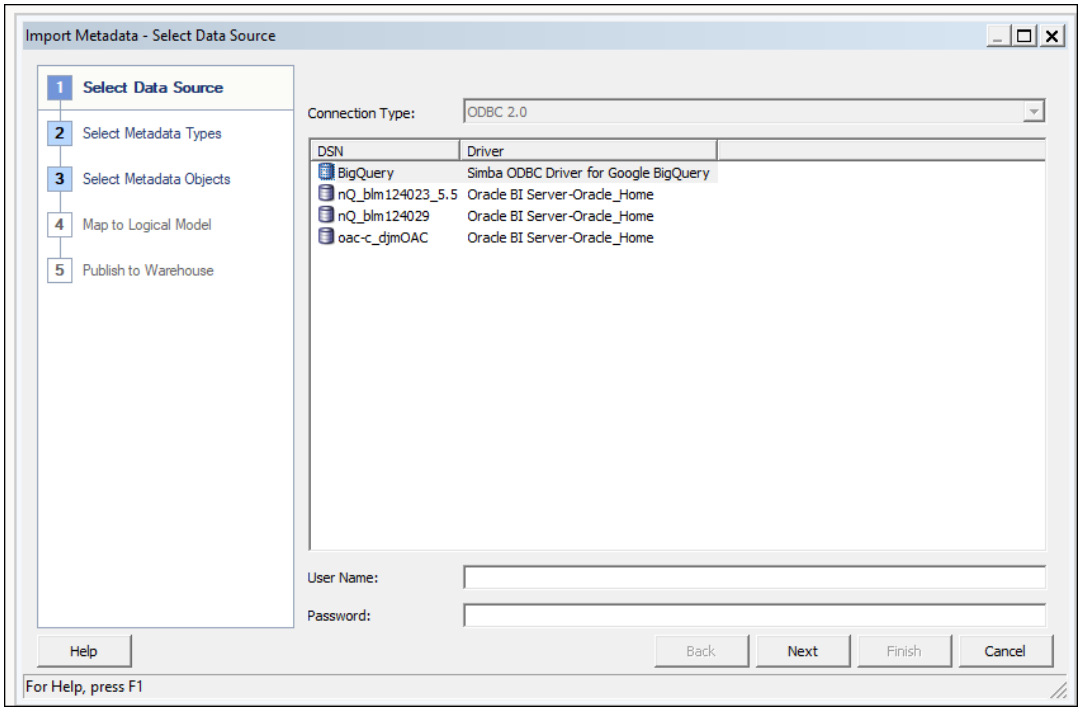


3. Create a physical schema in the database using the same name as the BigQuery dataset.

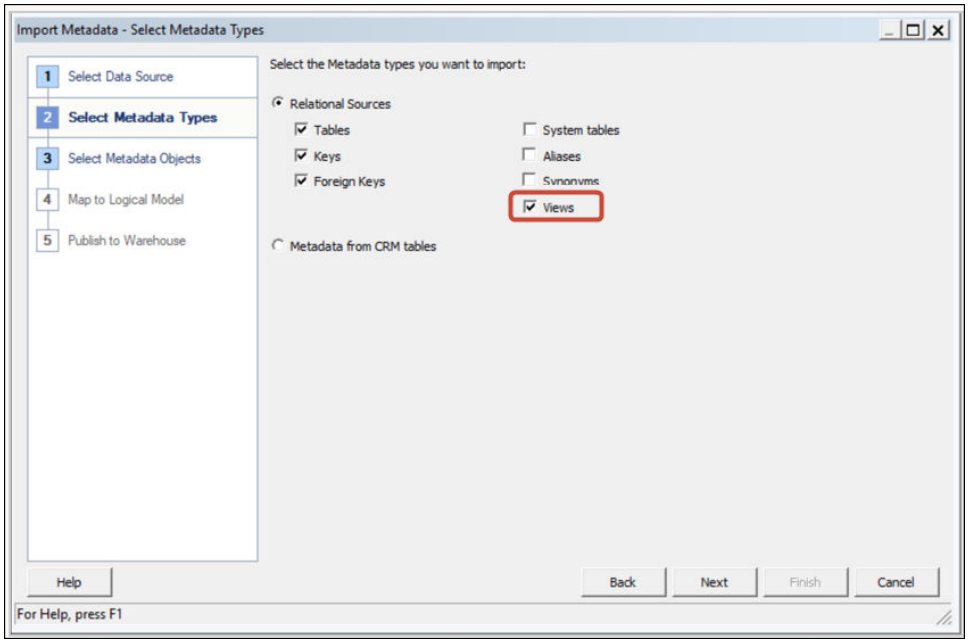
   BigQuery SQL requires that the dataset name prepend the table name, dataset.table. The dataset name is equivalent to a physical schema object in the repository file.



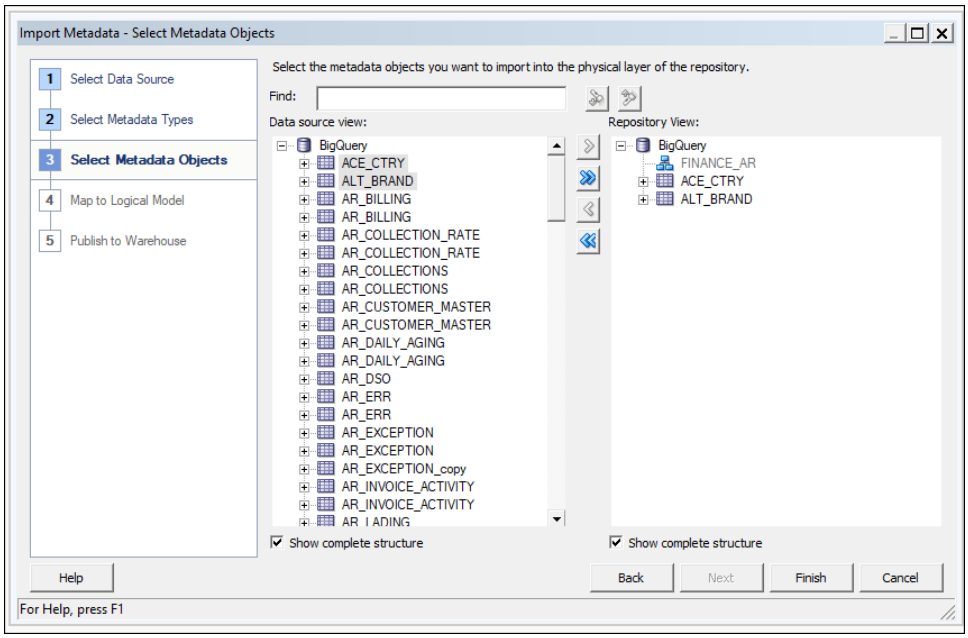
4. Right-click the connection pool and select **Import Metadata**.
5. On the Select Data Source dialog, select either ODBC 2.0 or ODBC 3.5 for the connection type, and select the BigQuery ODBC driver.



6. On the Select Metadata Types dialog, select **Views** and any other types you want to use for which your BigQuery key has permissions.

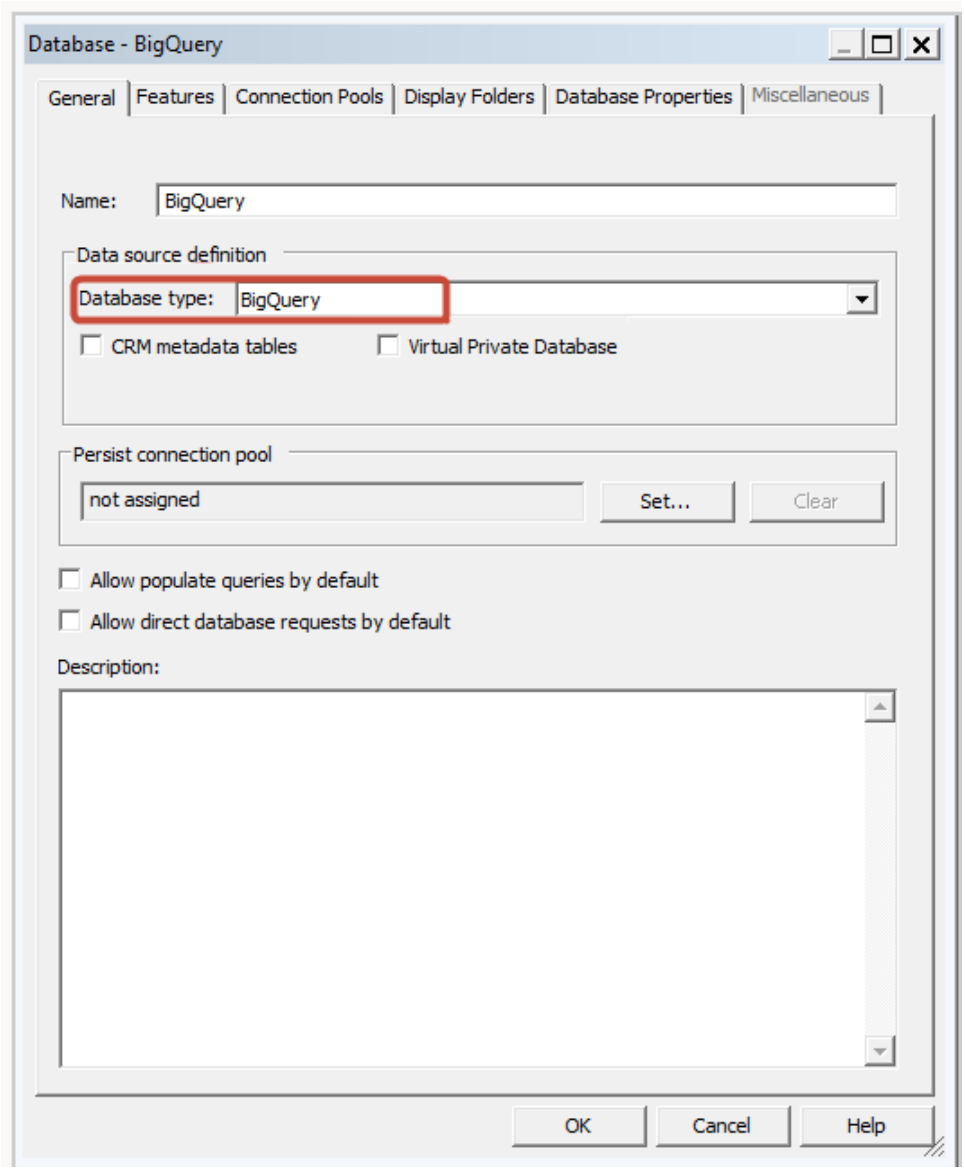


7. On the Select Metadata Types dialog, select the individual tables and then click **Import Selected**. This imports the BigQuery database and the underlying structures.

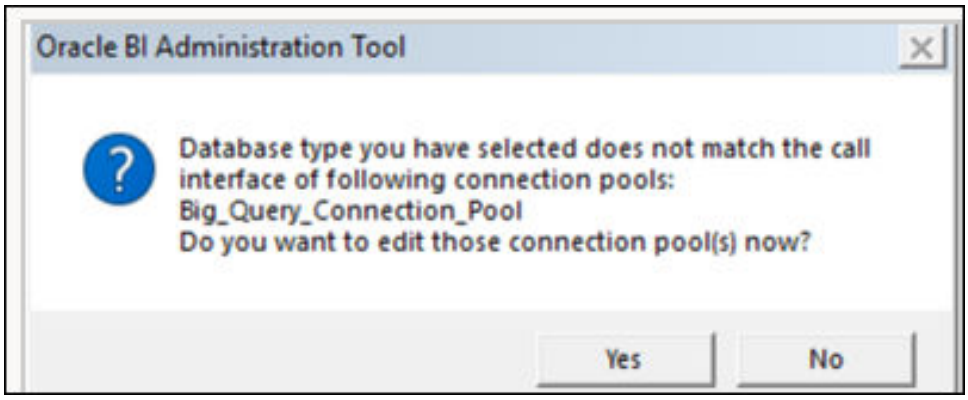


If you click **Import All**, you import only the database. If this happens, select **Import All** a second time to import the tables.

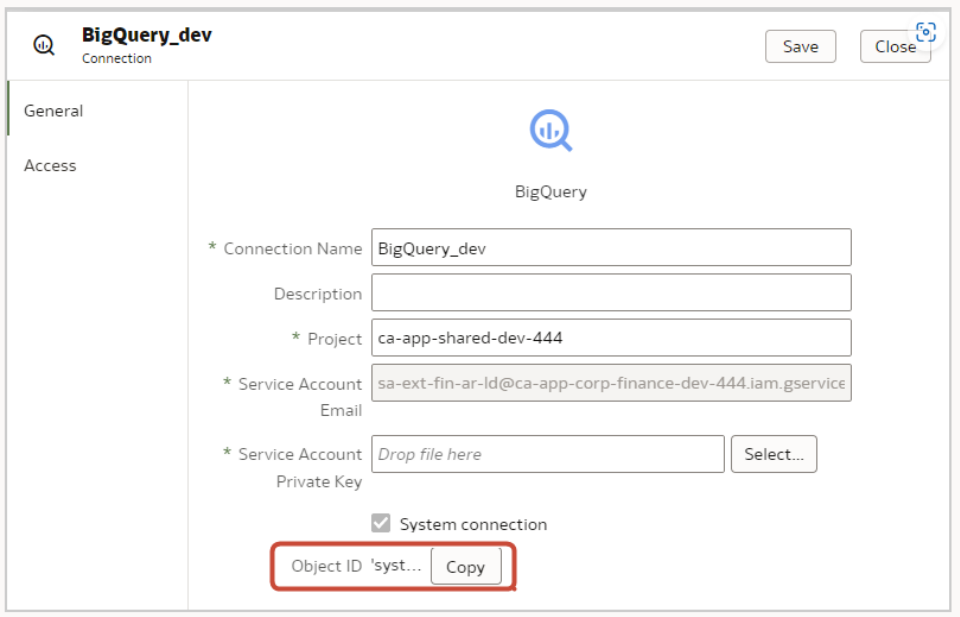
8. Click **Finish**.
9. Drag imported tables into physical schema.
10. Edit the physical database and change the database type to **BigQuery**.



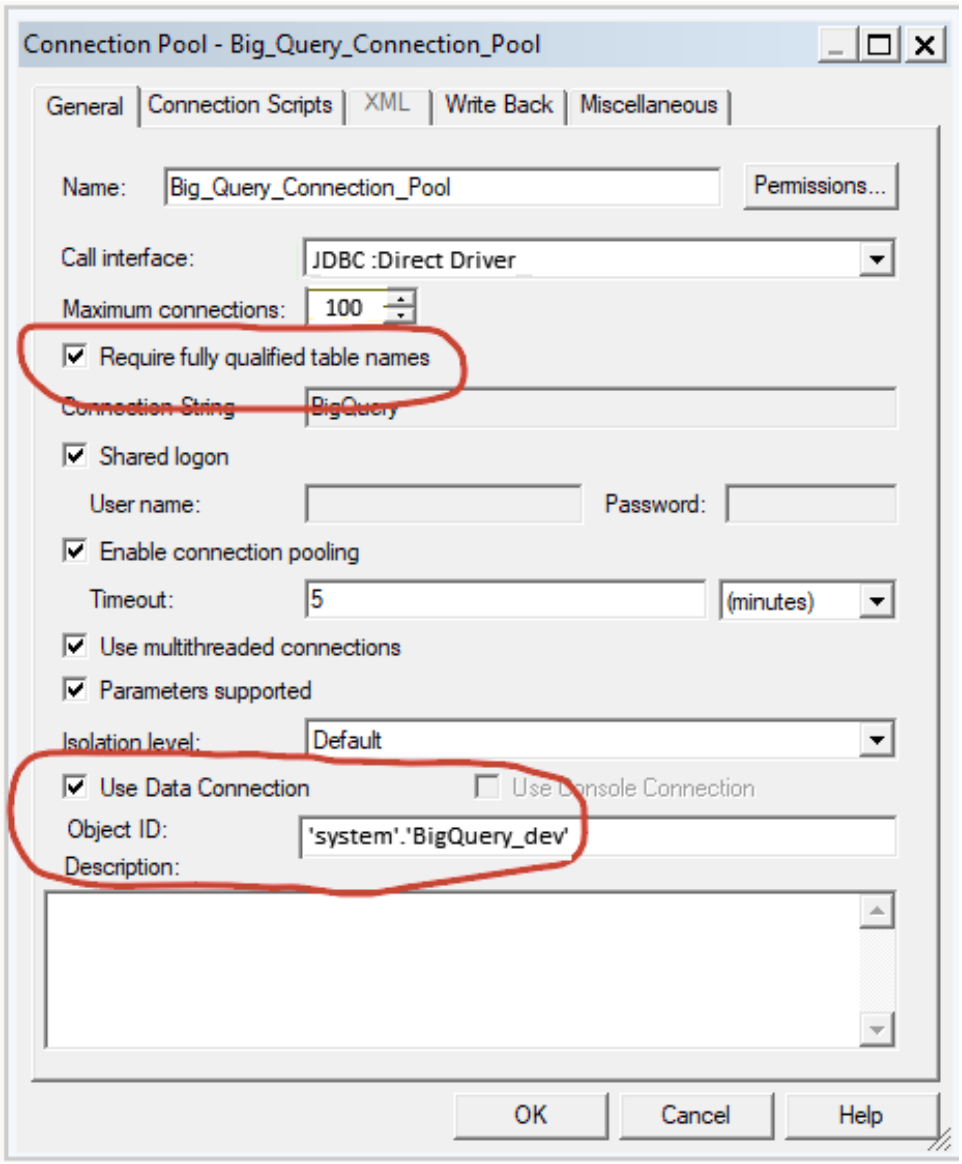
When changing the physical database, you see a message that states that the database type doesn't match the call interface set in the connection pool. Click **Yes**.



11. In the Connection Pool dialog, configure these settings:
    - In **Call interface**, change the call interface to JDBC (Direct Driver).
    - Select **Require fully qualified table names**.
    - Select **Use Data Connection**.
    - In Oracle Analytics, inspect the BigQuery connection and copy the Object ID. BigQuery is case-sensitive. To ensure that the data connection syntax is correct, use the **Copy** button.



    - In the Connection Pool dialog, paste the copied Object ID into the **Object ID** field.
    - Set **Maximum connections** to 100.



12. Save the details.

Model the metadata in the repository and upload the repository file (RPD) to Oracle Analytics.

Troubleshoot Repository Connection Issues for Google BigQuery

Here're some issues that you might encounter when connecting to Google BigQuery and work-arounds for them.

If 'Require fully qualified table names' isn't selected and a physical schema isn't part of the generated SQL, then queries fail with a message similar to 'Failed to read data from Java Datasource server'.

If the query is run against BigQuery using nqcmd or another SQL entry tool, the actual error message is displayed:

```
WITH SAWITH0 AS (select distinct T4.PROP_CD as c1 from FINOPS_RM_OCC_ACT T4)
select 0 as c1, D1.c1 as c2 from SAWITH0 D1 order by c2
```

```
[Simba][BigQuery] (70) Invalid query: Table "FINOPS_RM_OCC_ACT" must be
qualified with a dataset (e.g. dataset.table).
Statement preparation failed
```

The way to qualify the query with a dataset is to use a physical schema in the repository file.

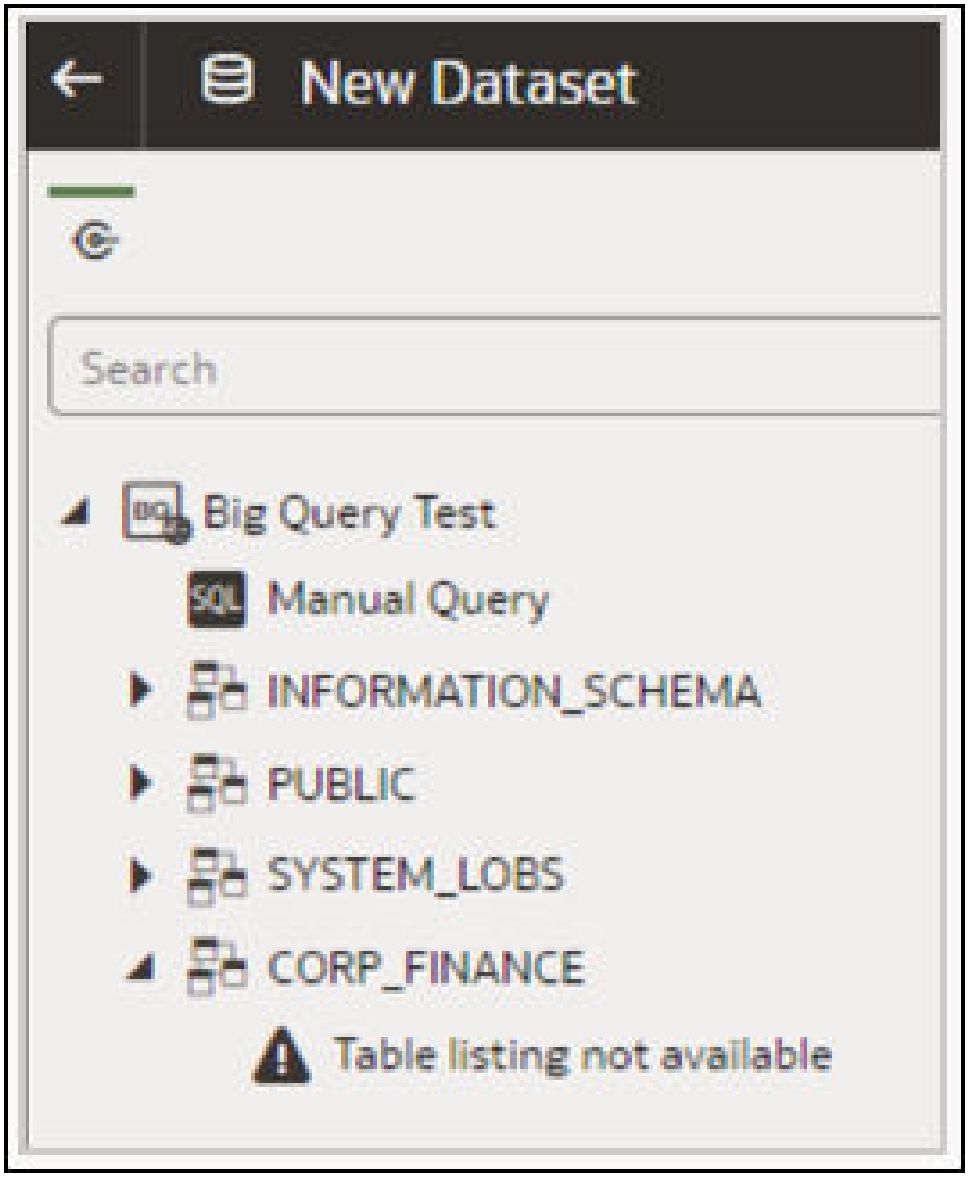
If the Oracle Analytics connection uses an uppercase project name, the connection is created successfully.

You might see two different problems.

1. Queries fail with a 404 Not Found message about a masked URL:

```
[2022-03-17T01:13:44.105+00:00] [OBIS] [TRACE:2] [USER-34] [] [ecid:
d6382db0-1e63-427e-893b-18bc00c0424e-0000de96,0:2:1:5] [sik: bootstrap] [tid:
856a6700] [messageId: USER-34] [requestid: 6358001e] [sessionid: 63580000]
[username: Testuser] -------------------- Query Status: [nQSError: 46164]
HTTP Server returned 404 (Not Found) for URL [masked_url]. [[
[nQSError: 46281] Failed to download metadata for dataset 'system'.'BigQuery
Test'.
[nQSError: 43119] Query Failed:
```

2. Within Oracle Analytics, you see datasets but the underlying tables aren't available.



In both cases, you can modify the connection so that the project name is lowercase.

When troubleshooting BigQuery connections in Oracle Analytics Cloud, use a third-party JDBC client to try to connect to BigQuery using the same Service Account Key.

If the connection still fails, there's a problem with the Service Account Key.

If the connection is successful, there's a problem with Oracle Analytics and you should contact Oracle Support.

This test is helpful in cases where the Service Account Key is not verified through ODBC.

DSN Formats for Specifying Data Sources

In Oracle Analytics you can model your on-premises data for many database types. Oracle Analytics supports direct access to some on-premises data sources through the semantic model. When you create the database connection using Model Administration Tool, in the **Data**

source name field on the Connection Pool dialog (General tab) you use the appropriate DSN format for the database type you're connecting to.

```
Amazon Redshift:
       DRIVER=Oracle 7.1 Amazon Redshift Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
      SSL: DRIVER=Oracle 7.1 Amazon Redshift Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"];EM=6;CPV=TLSv1.2,TLSv1.1,TLSv1,
SSLv3, SSLv2;VSC=0
Apache Drill:
       DRIVER=MapR Drill ODBC Driver;Host=["host-
name"];Port=["port"];CastAnyToVarchar=true;ExcludedSchemas=sys,INFORMATION_SCH
EMA;AuthenticationType=Basic
Authentication;ConnectionType=Direct
Aster:
       DRIVER=Aster ODBC Driver;SERVER=["host-
name"];PORT=["port"];DATABASE=["service-name"]
DB2:
      DRIVER=Oracle 7.1 DB2 Wire Protocol;IpAddress=["host-
name"];PORT=["port"];DB=["service-name"]
      SSL:  DRIVER=Oracle 7.1 DB2 Wire Protocol;IpAddress=["host-
name"];PORT=["port"];DB=["service-name"];EM=1;VSC=0
Greenplum:
       DRIVER=Oracle 7.1 Greenplum Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
Hive:
       DRIVER=Oracle 8.0 Apache Hive Wire Protocol;HOST=["host-
name"];PORT=["port"]
      SSL: DRIVER=Oracle 8.0 Apache Hive Wire Protocol;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
Impala:
       DRIVER=Oracle 7.1 Impala Wire Protocol;HOST=["host-name"];PORT=["port"]
     SSL:  DRIVER=Oracle 7.1 Impala Wire Protocol;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
Informix:
       DRIVER=Oracle 7.1 Informix Wire Protocol;HOSTNAME=["host-
name"];PORTNUMBER=["port"];DATABASE=["service-name"]
MongoDB:
       DRIVER=Oracle 8.0 MongoDB;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
MySQL:
      DRIVER=Oracle 7.1 MySQL Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
PostgresSql:
       DRIVER=Oracle 7.1 PostgreSQL Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
Spark:
      DRIVER=Oracle 8.0 Apache Spark SQL;HOST=["host-name"];PORT=["port"]
      SSL:  DRIVER=Oracle 8.0 Apache Spark SQL;HOST=["host-
name"];PORT=["port"];EM=1;VSC=0
SQL Server:
     DRIVER=Oracle 7.1 SQL Server Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-name"]
     SSL:  DRIVER=Oracle 7.1 SQL Server Wire Protocol;HOST=["host-
name"];PORT=["port"];DB=["service-
name"];EM=1;VSC=0;CryptoProtocolVersion=TLSv1.2,TLSv1.1,TLSv1,SSLv3,SSLv2
```

```
Sybase:
       DRIVER=Oracle 7.1 Sybase Wire Protocol;NA=["host-name"],
["port"];DB=["service-name"]
Teradata:
       DRIVER=Oracle 7.1 Teradata;DBCName=["host-name"];port_name=["port"]
```

Integrate with Oracle Enterprise Performance Management Platform Business Processes

Oracle Analytics Cloud integrates with Oracle Enterprise Performance Management (Oracle EPM) platform business processes, for example Oracle Planning and Budgeting Cloud Service.

There're two ways to integrate with Oracle EPM Platform:

- **Visualize data directly from Oracle EPM in Oracle Analytics Cloud** - Business users create visualization workbooks using data from cubes or plan types. For example, you can visualize data from Planning and Budgeting Cloud, Hyperion Planning, and Essbase Cubes. Self-service connection requires no special modelling or administration privileges. Business users simply create a connection using the **Oracle EPM** connection type, and create a visualization workbook.

  See Visualize Data from Oracle Enterprise Performance Management (Oracle EPM).

- **Model data for analyses in Oracle Analytics Cloud Classic** - Business analysts model Oracle EPM data first, then publish the semantic model for use by business users. For example, administrators or data modelers might create advanced calculations for an organization to use. Advanced data modeling requires administration privileges, and also requires Enterprise Edition.

  See Model Data in the Oracle EPM Platform.

Best Practices

Connectivity from Oracle Analytics Cloud to Oracle EPM Cloud is subject to concurrent query limits established by Oracle EPM Cloud. These limits are put in place to balance the needs of Oracle EPM application users and reporting applications that retrieve data from Oracle EPM.

Concurrency limits and other application workload on an Oracle EPM system affect the performance of Oracle Analytics Cloud applications that generate high concurrent query volume against Oracle EPM Cloud.

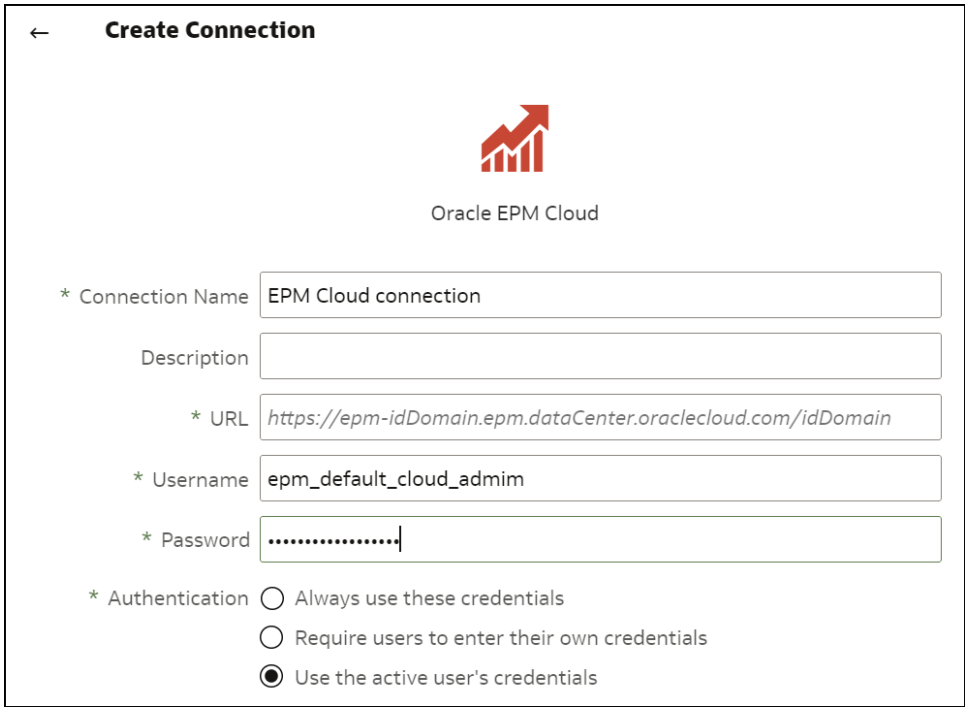
Reduce overall query volume by aligning with the Oracle EPM Cloud limits described in the following table:

Setting or Area	Recommended Number
Number of semantic model connections (set by **Max Connections**)	10
Number of visualizations per each workbook canvas	4

Visualize Data from Oracle Enterprise Performance Management (Oracle EPM)

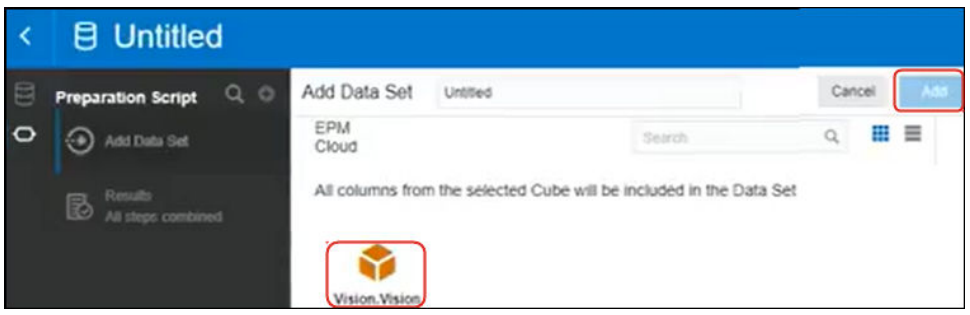
Connect to an application in Cloud EPM Platform and visualize the data in a workbook. Data is accessed as a live query.

1. In the Oracle Analytics Cloud homepage, click **Create**, then **Connection**, and select **Oracle EPM Cloud**.
2. In the Create Connection dialog, enter connection details for your Cloud EPM Platform application.
   - In **Authentication**, select **Use the active user's credentials**.



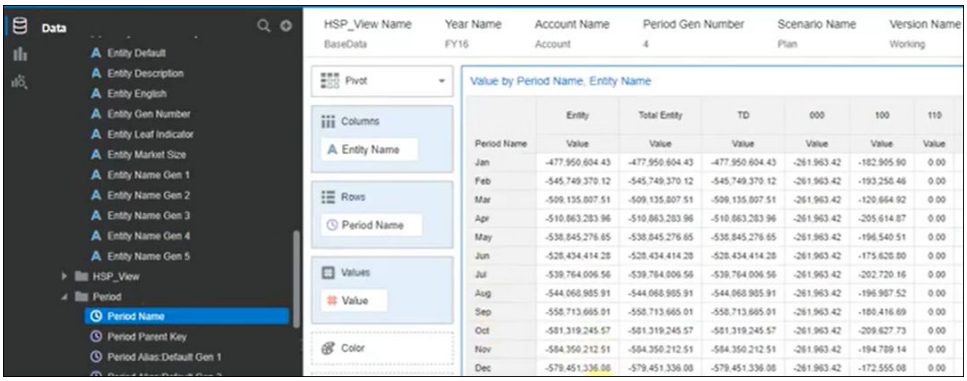
See Connect to Oracle Enterprise Performance Management Cloud.

3. From the home page, click Create, then Dataset.
4. At the Create Dataset dialog, select the Oracle EPM connection that you created in Step 2.
5. In the Add Dataset dialog, select the cube to visualize, then click Add.



6. Click Create Workbook, then add the data elements to the design canvas.

   Tip: Use filters to precisely locate the data in your cubes.



Model Data in the Oracle EPM Platform

Oracle Analytics Cloud Enterprise Edition integrates with Oracle Enterprise Performance Planning platform (Oracle EPM). You can build dashboards and analyses from Oracle EPM Cloud.

Topics

Note: You can only model EPM data in Model Administration Tool.

- Overview to Integration with Planning, Close and Tax Reporting on Oracle EPM Platform
- Prerequisites for Integration with Oracle EPM Platform
- Build and Upload a Semantic Model from Cloud EPM Platform

Overview to Integration with Planning, Close and Tax Reporting on Oracle EPM Platform

Oracle EPM Cloud business processes are used by companies to analyze data for planning, forecasting, and budgeting.

Report builders can analyze and build dashboards of data from Oracle EPM Cloud. Before they start, if you're creating a semantic model, you import the required application metadata from on-premises and cloud data sources, which enables your report builders to create dashboards and analyses.

- Oracle Analytics Cloud supports Planning, Financial Consolidation and Close, and Tax Reporting.

  If you have Oracle Enterprise Performance Management (EPM) Cloud Version 19.08 and higher, you can also use the Analytic Data Modeling (ADM) driver that's preinstalled with Oracle Analytics Cloud to take advantage of the advanced modeling capabilities provided in Oracle Planning and Budgeting Cloud Service:

  – Generation of number columns that support filtering.

  – Leaf indicator columns.

  - Separate columns per generation for member names and aliases (requires EPM 20.04).
  - Attribute dimensions.
  - Enhanced performance features.
- You can review the Service Description documents to understand licensing requirements for using this feature. See Service Description Documents.
- When you import data from Hyperion Planning data sources, both measures and dimensions are imported into your semantic model.

Prerequisites for Integration with Oracle EPM Platform

Before you start, make sure you have the required components in place and deployed correctly.

- Oracle Analytics Cloud - Enterprise Edition.
- A Windows 64–bit machine on which to run Model Administration Tool.
- Model Administration Tool for Oracle Analytics Cloud 5.6 or later.

  Download it from Oracle Technology Network and install it on a local Windows 64–bit machine. See Download and Install Analytics Client Tools for Oracle Analytics.
- A Javahost process running on the client. (Start a process using the command: `C:\oracle\oac-client-5.6\bi\bifoundation\javahost\bin\startOnClient.bat`.)
- If you're deploying Oracle Analytics Cloud, (that is, on Oracle Cloud Infrastructure with Oracle-Managed), configure these options:
  - Set the `JAVA_HOME` system environment variable to point to your JDK installation. For example, `C:\Program Files\Java\jdk1.8.0_162`.
  - `set INSTANCE_NAME=%DOMAIN_HOME%`
  - Edit the file `<BIClient_Home>\bi\bitools\bin\admintool.cmd`, and below the ESSBASEPATH setting add:`set INSTANCE_NAME=%DOMAIN_HOME%`
  - Run `<BIClient_Home>\bi\bifoundation\javahost\bin\startOnClient.bat` to start Javahost.
    If you need to restart the client, run `stopOnClient.bat`, then rerun `startOnClient.bat`.
- If you're deploying Oracle Analytics Cloud- Classic, (that is, Oracle Cloud Infrastructure Classic), configure these options:
  - Configure `<BIClient_Home>\bi\bitools\bin\admintool.cmd` with the instance name:

    ```
    INSTANCE_NAME=%DOMAIN_HOME%
    ```
  - Configure Oracle Analytics Cloud host and port information in `<BIClient_Home>\bi\config\fmwconfig\biconfig\OBIS\NQSConfig.INI`:

    ```
    [JAVAHOST] JAVAHOST_HOSTNAME_OR_IP_ADDRESSES = "host:9506";
    ```

Build and Upload a Semantic Model from Cloud EPM Platform

Build a semantic model on Cloud EPM Platform, then upload it to Oracle Analytics Cloud Enterprise Edition.

1. In your local environment, start Model Administration Tool, and create a semantic model.
2. Import the Planning and Budgeting metadata:
   a. From the **File** menu, select **Import Metadata** to start the import wizard.
   b. In the Select Data Source page, select *Hyperion ADM* from the **Connection Type** list.
   c. In the Select Metadata Objects page, enter connection details.

      For **Connection Type**, select **Hyperion ADM**.

      For **Provider Type**, select **Hyperion Planning**.

      For **URL**, specify the connection URL in the format:

      ```
      adm:thin:com.hyperion.ap.hsp.HspAdmDriver:<Server>%3A<Port>:<Application>
      ```

      Note: Specify the URL-encoded value %3A instead of a colon (:) to separate the host and port number.

      The connection URL is different on Oracle Cloud Infrastructure (Gen 1) and Oracle Cloud Infrastructure (Gen 2).

      - If your Planning and Budgeting application is deployed on Oracle Cloud Infrastructure (Gen 1), specify the connection URL as:

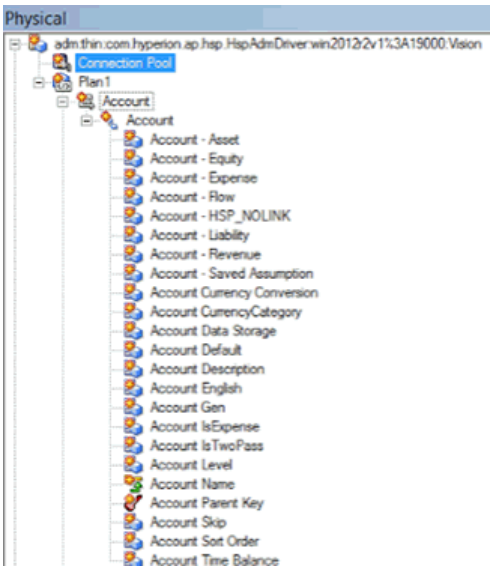
        ```
        adm:thin:com.hyperion.ap.hsp.HspAdmDriver:machine12345.oraclecloud.com%
        3A443:Vision?locale=en_US;tenantName=localhost;hubProtocol=https;
        ```

      - If your Planning and Budgeting application is deployed on Oracle Cloud Infrastructure (Gen 2), specify the connection URL as:

        ```
        adm:thin:com.hyperion.ap.hsp.HspAdmDriver:machine12345.oraclecloud.com%
        3A443:Vision;locale=en_US;tenantName=localhost;hubProtocol=https;
        ```

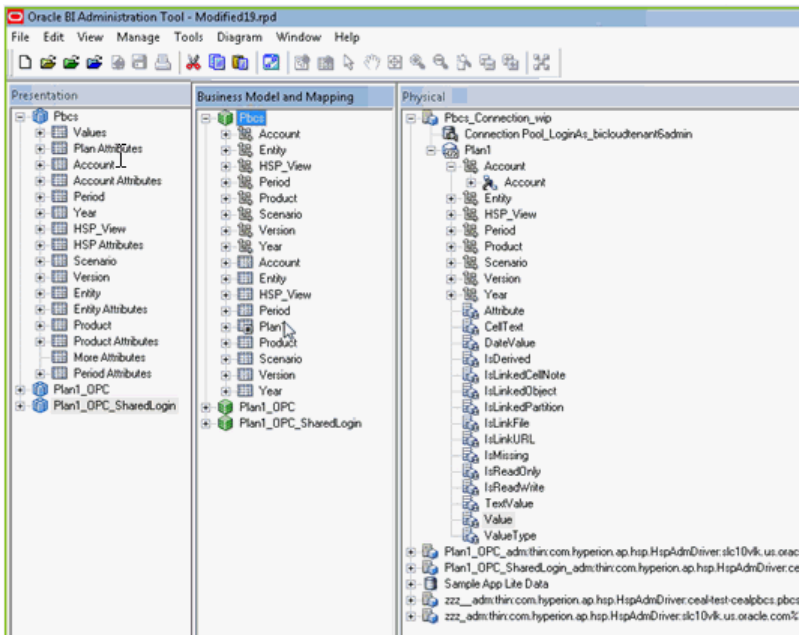
      For **User Name and Password**, specify the name and password of a user with administration privileges.

   d. When the import is complete, review the metadata in the Physical Layer.



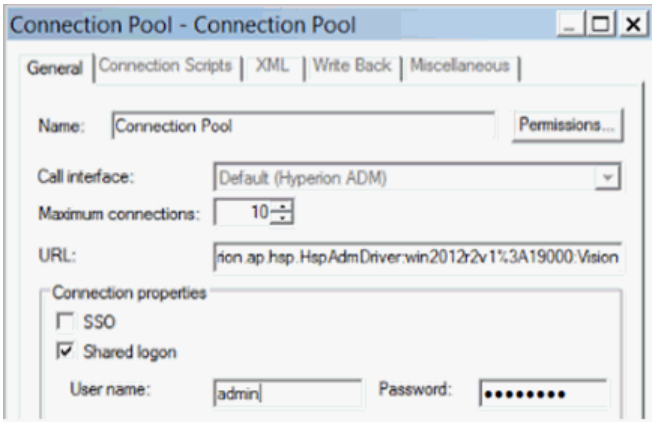
   e. Make sure that you've selected the `ANCESTOR_DIM_PROPERTY_SUPPORTED` database feature.
3. Complete your semantic model:

a. Cut and paste tables in the Physical layer to the Business Model and Mapping layer and to the Presentation layer.



b. Verify the semantic model and save it as an .rpd file.

4. In the Physical layer, edit the **Connection Pool** for this data source, specify a **URL** using the URL that you used on the Select Metadata Objects page in step 1, and select **Shared logon**.



5. Save the semantic model, and ignore the warning that the database features don't match the defaults.

6. Upload the semantic model to Oracle Analytics Cloud. From the **File** menu, click **Cloud**, then click **Upload**, and enter connection information for your Oracle Analytics Cloud instance.

Users can analyze data and build dashboards using the new semantic model.

6

Give Data Sources Access to Oracle Analytics Cloud Deployments

Some data sources, such as Oracle Autonomous Data Warehouse, require you to include the *IP address* of your Oracle Analytics Cloud deployment in their allowlist.

Topics:

7

Manage Database Connections for Model Administration Tool

Administrators create and manage cloud database connections for Model Administration Tool. Your business data doesn't have to be in one place. Connect to multiple cloud databases so business modelers and analysts can analyze company data wherever it is stored.

Topics

About Database Connections for Semantic Models

When you use Model Administration Tool to edit your semantic models and upload them to Oracle Analytics Cloud, you can refer to any database connections you define in the Console “by name” in the Connection Pool dialog. You don’t need to re-enter the connection details in Model Administration Tool.

See Connect to a Data Sources using a Connection Defined In Console.

You don't have to re-enter database connection information for semantic models pre-built with Oracle Analytics Server.

Connection information for these models is often already defined in the semantic model that you upload to Oracle Analytics Cloud. See About Uploading Semantic Models from Oracle Analytics Server.

Connect to Data in an Oracle Cloud Database

Administrators create database connections for Model Administration Tool so business analysts can analyze data stored in Oracle Cloud databases.

1. Click **Console**.
2. Click **Connections**.
3. Click **Create**.
4. Enter a meaningful **Name** and **Description** that you’ll remember and business modelers will recognize.
5. For **Connect Using**, select which properties you want to use to connect to the database.
6. Specify database connection information.
    a. In **Host**, specify the host name or IP address of the database you want to connect to.

b. In **Port**, specify the port number on which the database is listening for incoming connections.

c. In **Service Name**, specify the network service name of the database.

d. In **SID**, specify the name of the Oracle database instance.

e. In **TNS Descriptor**, specify the TNS connect descriptor that provides the location of the database and the name of the database service.

Use the format:

```
DESCRIPTION=(ADDRESS=(PROTOCOL=protocol)(HOST=host) (PORT=port))
(CONNECT_DATA=(SERVICE_NAME=service name))
```

For example:

```
DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myhost.example.om)(PORT=1521))
(CONNECT_DATA=(SERVICE NAME=sales.example.om))
```

7. For **Connect As**, enter the user name of the schema owner and then enter the **Password**.

You must connect to the database as the schema owner to access and display the tables in Data Modeler.

8. Select **Enable SSL**, to secure this connection using SSL.

If you haven't done so already, you must upload a wallet containing your SSL certificates.

9. Click **Test** to verify the connection.

10. Click **OK**.

Data modelers see the new connection in Model Administration Tool right away and can start to model the data.

Secure Database Connections with SSL

Use SSL to secure communication between Oracle Analytics Cloud and an Oracle database with SSL configured, Oracle Autonomous Data Warehouse, or Oracle Autonomous Transaction Processing. You must obtain and upload a wallet that contains SSL certificates, to enable SSL on your Oracle Database Classic Cloud Service connections.

1. Click **Console**.

2. Click **Connections**.

3. If you've not done so already, upload a wallet file containing SSL certificates to Oracle Analytics Cloud:

a. Click the Action menu, then **Upload Wallet**.

To update an existing wallet file, click **Replace Wallet**.

b. Click **Browse** and locate the wallet file.

Select a valid `cwallet.sso` file.

c. Click **OK**.

4. Enable SSL security on a database connection:

a. Create or edit a database connection.

b. In the Connection dialog, select **Enable SSL**.

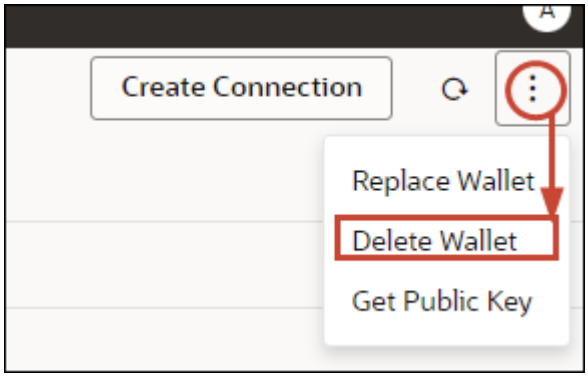
c. Click **OK**.

Delete the SSL Wallet Uploaded for Database Connections

If a database connection you configure for your data model requires SSL certification you must upload a wallet containing the required SSL certificates through the Console (Connections page). You can delete a wallet that you previously uploaded and no longer need.

For example, you might need to remove an existing wallet file if the Oracle Autonomous Data Warehouse that your data model connects to is set up to allow *wallet-less* connections.

1. Click **Console**.
2. Click **Connections**.
3. Click the Action menu, then **Delete Wallet**.



4. To confirm, click **Delete**.

Part III

Connecting to Oracle Analytics Cloud From Other Applications

This part describes how to connect to Oracle Analytics Cloud from other applications, for example, from Microsoft Power BI Desktop.

Chapters:

8

Connect to Oracle Analytics Cloud from Microsoft Power BI (Preview)

You can connect to Oracle Analytics Cloud from Microsoft Power BI and visualize Oracle Analytics content.

If you have an established Microsoft Power BI user base, you can leverage the visualization and publishing capabilities offered by Microsoft Power BI Desktop combined with the enterprise modelling capabilities of Oracle Analytics to create powerful insights into your data.

Topics:

About Microsoft Power BI Connectivity Support in Oracle Analytics Cloud (Preview)

You can use Microsoft Power BI Desktop to analyze content from Oracle Analytics Cloud.

In Microsoft Power BI, when you create visualizations based on Oracle Analytics Cloud workbooks and reports, you use cached data from Oracle Analytics Cloud subject areas. Data analysts can then share visualizations with other Microsoft Power BI users.

Prerequisites for Microsoft Power BI Integration (Preview)

Before you start, make sure that you have the following:

- A Windows machine with the latest version of Microsoft Power BI Desktop installed, or the April 2022 version at a miminum. Microsoft Power BI Pro or Premium aren't supported. To consume Oracle Analytics Cloud analyses in Microsoft Power BI Desktop using the Navigator (instead of copying SQL from Oracle Analytics Cloud Classic), use the latest January 2023 Update of Oracle Analytics Cloud and install V1.2 or higher of Power BI connector for Oracle Analytics Cloud.
- A Windows machine with the latest Oracle Analytics Client Tools installed. See Download page for Oracle Analytics Client Tools.
- User role privileges in Oracle Analytics Cloud:

  - To consume tables in subject areas, you need BIContentAuthor role privileges or higher.
  - To consume analysis reports, you need BIConsumer role privileges or higher.
- A semantic model in Oracle Analytics Cloud.
  If you used Model Administration Tool to develop your semantic model, make sure that subject areas and tables are available in the Presentation layer.

Configure a Microsoft Power BI Environment for Oracle Analytics Cloud Integration (Preview)

Configure your environment to integrate Microsoft Power BI Desktop with Oracle Analytics Cloud so that you can analyze data from Oracle Analytics Cloud.

Before you start, use the Oracle Analytics Model Administration Tool to create a data model in Oracle Analytics Cloud so that you can access the subject areas and tables in the presentation layer.

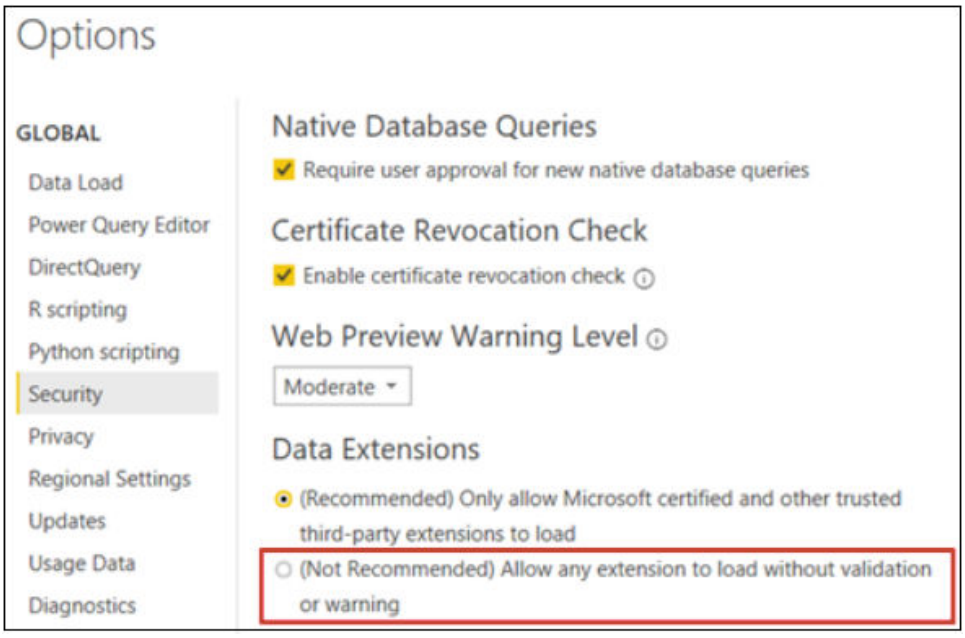
1. Install Microsoft Power BI Desktop.

   Install the minimum required version. See Prerequisites for Microsoft Power BI Integration (Preview).

   After you've installed it, configure these settings:

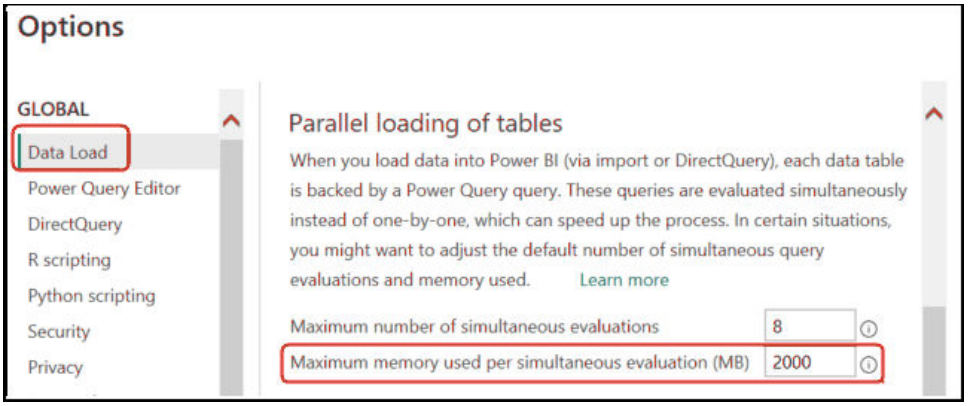
   a. In Power BI Desktop, navigate to Options and Settings.

   b. Under **GLOBAL** click **Security**, then under **Data Extensions**, select **(Not Recommended) Allow any extension to load without validation or warning**.



   c. Under GLOBAL, click **Data Load**, and set the value of **Maximum memory used per simultaneous evaluation (MB)** based on the available memory on your machine.

      Tip: To see how much memory your machine has available, hover over the information (*i*) icon next to **Maximum memory used per simultaneous evaluation (MB)**.



d. Under CURRENT FILE, click **Data Load**, and select **Enable parallel loading of tables**.

2. Install Oracle Analytics Client Tools in the same environment as Microsoft Power BI Desktop.
   a. Navigate to:

      Download page for Oracle Analytics Client Tools

   b. Click **Oracle Analytics Client Tools <Month Year> Update** to display the Oracle Software Delivery Cloud page, and select the latest version.
   c. Click the **Platforms** down arrow, click **Microsoft Windows x64 (64-bit)**, then click outside of the drop-down list or press Enter.
   d. In the Software column of the table, make sure that `Oracle Analytics Client...` is selected, and de-select other ZIP files (for example, `Windows Data Gateway...`).
   e. Accept the Oracle Cloud Service License Agreement.
   f. Click **Download** to start Oracle Download Manager, and follow the on-screen instructions.
   g. Unzip the ZIP file that you downloaded to extract the installer file `setup_bi_client-<update ID>-win64.exe`.
   h. Double-click the file `setup_bi_client-<update ID>-win64.exe` to start the installer.
   i. Follow the on-screen instructions.
3. In the same environment, download and install the Power BI connector for Oracle Analytics Cloud.
   a. Navigate to:

      Download page for Oracle Analytics Client Tools

   b. Click the download link for Oracle Analytics Cloud Connector for Microsoft Power BI to display the Oracle Software Delivery Cloud page, and select the latest version.
   c. Click the **Platforms** down arrow, click **Microsoft Windows x64 (64-bit)**, then click outside of the drop-down list or press Enter.
   d. In the Software column of the table, make sure that `Oracle Analytics Power BI Connector...` is selected, and de-select other ZIP files (for example, `Windows Data Gateway...`).
   e. Accept the Oracle Cloud Service License Agreement.

f. Click **Download** to start Oracle Download Manager, and follow the on-screen instructions.

g. On your local machine, create a `\Power BI Desktop\Custom Connectors` folder in `C:\Users\<username>\Documents\`.

   For example, `C:\Users\<username>\Documents\Power BI Desktop\Custom Connectors`.

h. Copy the downloaded OracleAnalyticsCloud-x.x.x.mez into the `\Power BI Desktop\Custom Connectors` folder.

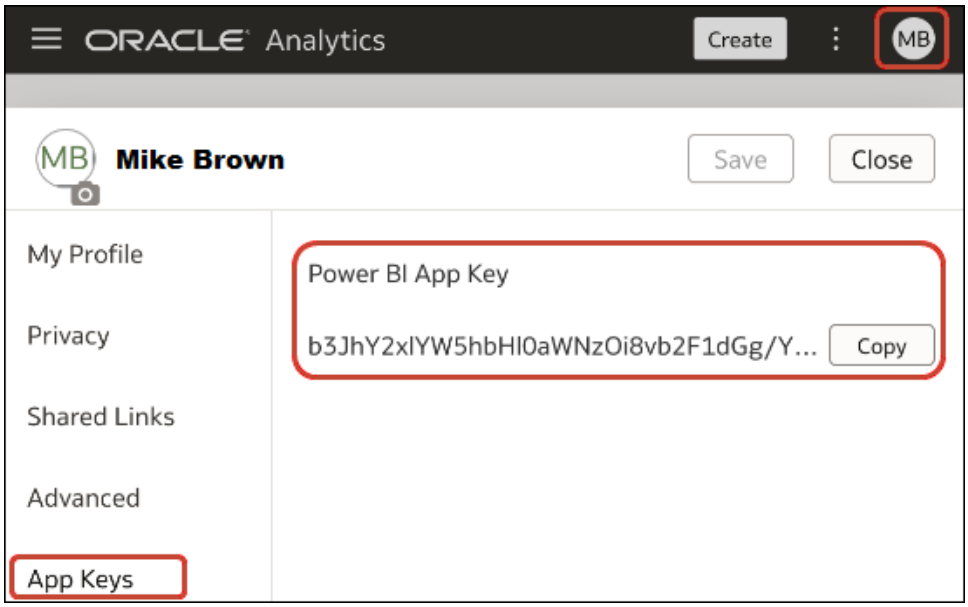
i. Start or restart Microsoft Power BI Desktop.

Connect to Oracle Analytics Cloud from Microsoft Power BI Desktop (Preview)

Data analysts connect to Oracle Analytics Cloud from Microsoft Power BI Desktop to analyze data from Oracle Analytics Cloud subject areas.

1. Obtain the **Power BI App Key** for your Oracle Analytics Cloud instance.

   In Oracle Analytics Cloud, the administrator obtains the user access key by clicking the administrator user profile, then **Profile**, then **App Keys**, and copying the **Power BI App Key**.

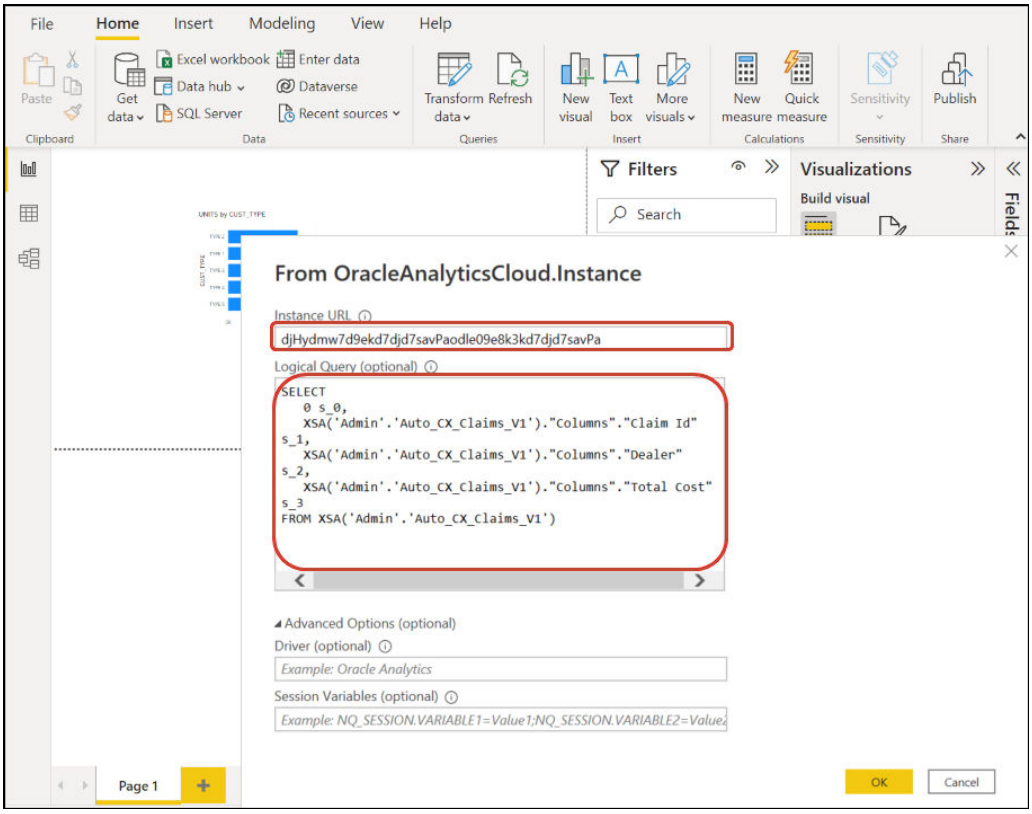


2. In Microsoft Power BI Desktop, connect to Oracle Analytics Cloud.

   a. From the home page, click **Get Data**, locate and select **Oracle Analytics (Beta)** in the list of connectors, then click **Connect**.

   b. In the **Instance URL** field, paste in or specify the **Power BI App Key** that you obtained in Step 1.

   c. In the **Logical Query (optional)** field, if you've copied SQL from a workbook or report, paste in or enter the SQL code.



Make sure that the SQL is copied from the same Oracle Analytics Cloud instance from which you obtained the **Power BI App Key**.

If you haven't copied SQL code from Oracle Analytics Cloud, you can skip Step 2.c and browse to reports or subject area tables manually after you click **OK**.

d. Specify advanced options (optional).

- In **Driver (optional)**, optionally specify an ODBC Driver Name from an Oracle Home if multiple Oracle Analytics Cloud Client Tool installations are present (the default is "Oracle Analytics").

- In **Session Variables (optional)**, optionally specify session variable values used in Oracle Analytics Cloud.

e. Click **OK**.

If you specified SQL code in the **Logical Query** field, Power BI displays the preview data.

If you didn't specify SQL code in the **Logical Query** field, use the Navigator to select an analysis or tables to preview.

f. Click **Load**.

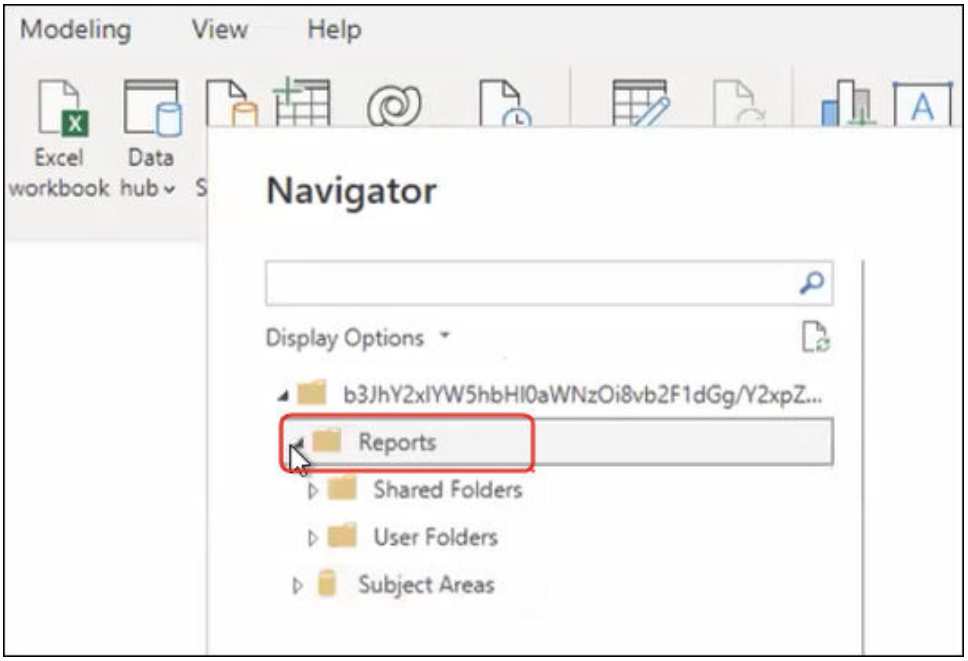
3. In Microsoft Power BI Desktop, create visualizations and save the project in a Power BI Desktop document (PBIX). See Integrate Oracle Analytics Cloud with Microsoft Power BI (Preview).

Integrate Oracle Analytics Cloud with Microsoft Power BI (Preview)

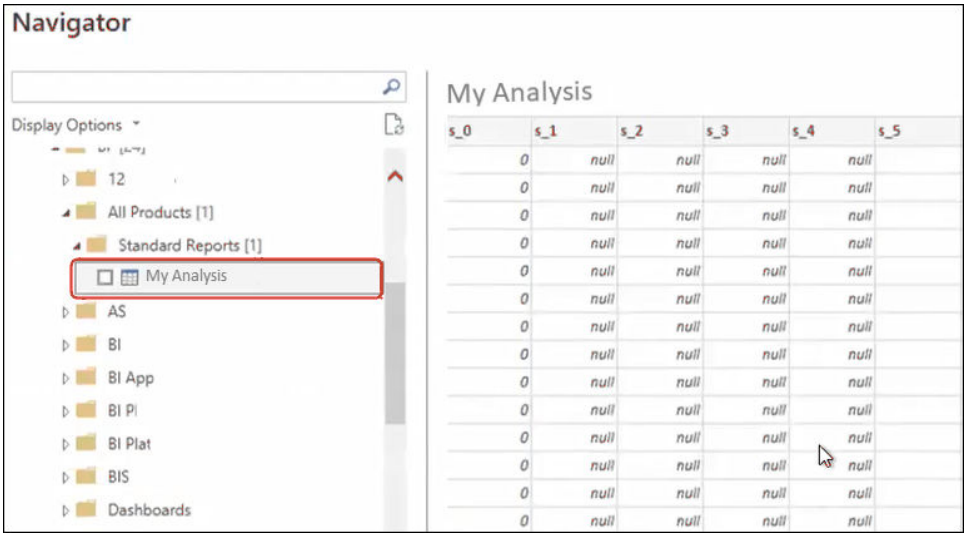
Follow these tips to integrate Oracle Analytics Cloud with Microsoft Power BI.

Integrate Oracle Analytics Cloud with Microsoft Power BI to leverage the visualization and publishing capabilities offered by Microsoft Power BI Desktop combined with the enterprise modelling capabilities of Oracle Analytics to create powerful insights into your data.

- In Power BI Desktop, create visualizations and save the project in a Power BI Desktop document (PBIX).
  - (Recommended approach for to consume reports) Use the Navigator to add reports directly – Analysts can use the Navigator to select reports (analyses) from Oracle Analytics Cloud to add them to a Power BI visualization. In the connect dialog, enter the Power BI App Key in the **Instance URL** field and leave the **Logical Query** box empty. You can then use the Navigator to browse to the Reports area.



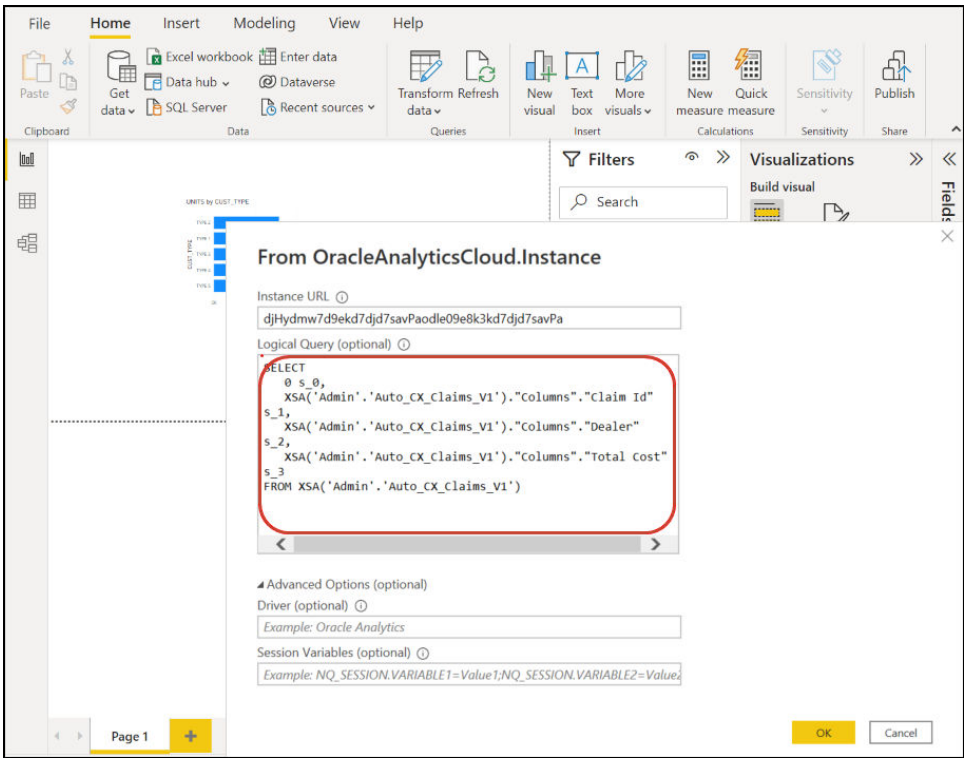
In the Reports area, select a report to add to your Power BI project. In the Reports areas, you can browse for available analyses to add to your Power BI project.



This method takes advantage of the Oracle Analytics Cloud data model.

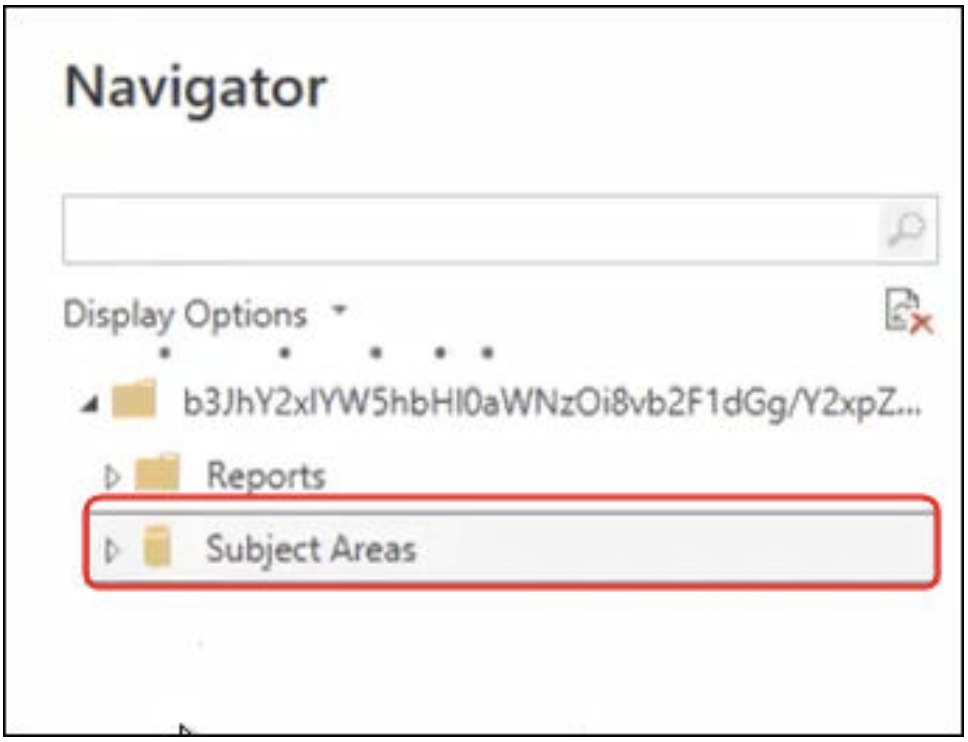
Alternatively, you can copy the **Logical SQL** code from the Advanced Tab of an analysis.

– (Recommended approach to consume workbooks) Copy SQL code from Oracle Analytics In a workbook, copy the SQL code from the Developer panel to add them to a Power BI visualization. Then, in the connect dialog enter the Power BI App Key in the **Instance URL** field and copy the SQL code into the **Logical Query** box. This method takes advantage of the Oracle Analytics Cloud data model.

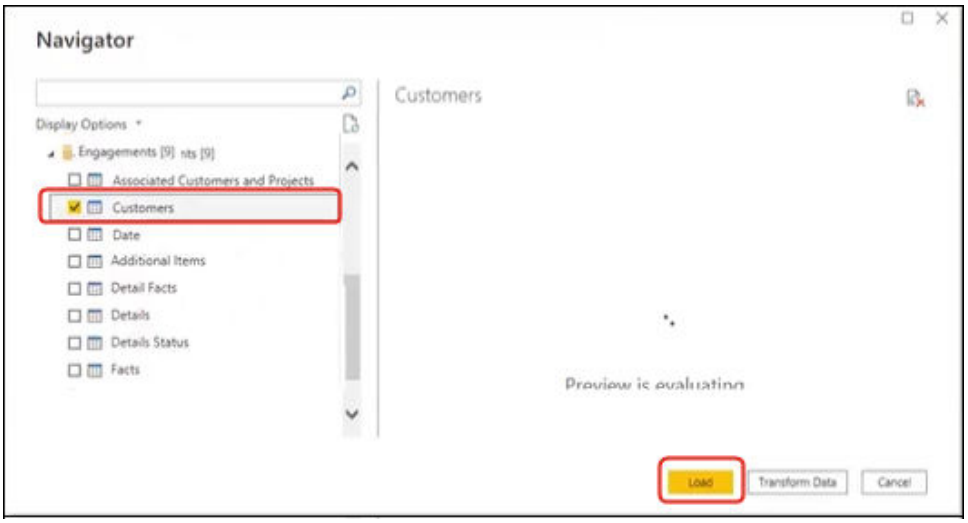


– Browse subject areas manually - Analysts can select subject area tables to load by browsing the subject areas manually. In the connect dialog, enter the Power BI App

Key in the **Instance URL** field and leave the **Logical Query** box empty. You can then use the Navigator to manually select tables in the Subject Areas folder.



In Oracle Analytics Cloud, curate subject areas specifically to meet the reporting needs of the Power BI users. Make sure that subject areas are curated with both fact and dimension columns in a single subject area table.



You can use this approach for basic data, however be aware that Microsoft Power BI makes assumptions and optimizations that affect your analysis results. As a result, it might be more difficult to debug and audit the queries that Microsoft Power BI generates.

In addition, when you access tables by browsing the subject areas, each table is loaded independently of other tables. Subject areas are often structured into dimension and fact tables that don't include related columns.

- Share the Power BI (PBIX) project with other users, who can manipulate any of the visualizations in the project.

Troubleshooting Power BI Connectivity and Performance (Preview)

Here're some tips on addressing connection and performance issues that you might encounter when you connect from Microsoft Power BI to Oracle Analytics Cloud .

Issue reported	Try this
`ODBC Driver Load Is Failing`	Make sure that: • Oracle Analytics Client Tools is installed in an Oracle_Home. • The PATH variable includes `server/bin`. • You have a 64-Bit installation of Microsoft Power BI Desktop.
`Unknown ODBC Error`	Download and install the latest Oracle Analytics Client Tools. See Download and Install Analytics Client Tools for Oracle Analytics.
I'm unable to see subject areas after signing-in successfully.	Check that the Oracle Analytics Cloud user has the 'BIContentAuthor' role, which is required to access the subject areas in Oracle Analytics Cloud. A user with only the 'BIConsumer' role can't access subject areas from Microsoft Power BI.
I encouter data loading issues when loading tables with large volumes of data.	In Microsoft Power BI on the client machine, go to **Options**, **Data Load**, and increase **Maximum memory used per simultaneous evaluation (MB)** value. To see how much memory your machine has available, hover over the information (*i*) icon next to **Maximum memory used per simultaneous evaluation (MB)**. For example, if the value is set to the default 432, you might increase it to 4000.
I encounter an authentication error after approximately 100 seconds.	Contact your Oracle Analytics Cloud administrator to adjust the Expiry Time of the Access Token for Oracle Analytics Cloud.

Frequently Asked Questions the Connector for Microsoft Power BI (Preview)

Here're some frequently asked questions (FAQs) to help you connect from Microsoft Power BI to Oracle Analytics Cloud .

Does the connector support Microsoft Power BI Pro or Premium (and Data Gateway)?

No. The connector supports Microsoft Power BI Desktop, but not the Pro or Premium versions.

Does the connector support live query?

No. Microsoft Power BI caches data for datasets.

Does the connector support Oracle Analytics Cloud datasets?

No. The concept of an enterprise semantic model at the scale that's commonly used in Oracle Analytics Cloud doesn't exist in Microsoft Power BI, which makes it an important point of integration between the two products.

Do Microsft Power BI and Oracle Analytics Cloud use subject areas in the same way?

No. Power BI is mostly optimized for cached data within its own store, and the integration with Oracle Analytics Cloud is based on that concept.

How does Oracle Analytics Cloud data security work with Microsoft Power BI?

Microsoft Power BI applies data security at the time that data is created. Therefore, the Oracle Analytics Cloud credentials determine the result set. Any row-level security on top of that must be done within Microsoft Power BI.

Should I use Microsoft Power BI or Oracle Analytics Cloud for my visualization needs?

Oracle recommends using Oracle Analytics Cloud wherever possible. Microsoft Power BI visualization provides additional analysis capabilities if you have a fragmented visualization eco system but still want to maintain enterprise semantics. Oracle Analytics Cloud provides the best combination of powerful enterprise analysis and line-of-business agility.

How do I get support for the functionality?

Contact Oracle Support if you have issues or questions relating to this feature at: https://support.oracle.com.

9

Query Semantic Models Remotely Using JDBC

You can query Oracle Analytics Cloud semantic models from an external client tool using a JDBC connection.

Topics

Overview to Querying Oracle Analytics Cloud Semantic Models Remotely

Java Data Base Connectivity (JDBC) is an industry standard API for accessing data sources. Use JDBC-compliant client tools to access semantic models defined in Oracle Analytics Cloud so that you can take advantage of its analytics engine and data abstraction features.

This feature requires Oracle Analytics Cloud 5.6 or later. Before you start, verify that the Oracle Analytics Cloud instance you're connecting to is 5.6 or later. If you're not sure, contact your Oracle representative.

JDBC uses OAuth to secure access to Oracle Analytics Cloud. OAuth is an authorization framework that enables an application to obtain limited access to a protected HTTP resource. In OAuth, applications are called clients; they access protected resources by presenting an access token to the HTTP resource.

Choosing an Assertion Type for Your JDBC Connection

When you register a BIJDBC application in Oracle Cloud Infrastructure, you specify an assertion type that best secures your connection.

Here's some guidance on choosing an assertion type using the **Allowed Grant Types** option on the Add Confidential Application page.

Assertion Type	Use this option when:	Considerations
(Recommended) Resource Owner	Your application has access to the user name and password of the end user connecting to Oracle Analytics Cloud. You want to create a 'gateway' connection to query data from the BI Server. Gateway accounts always use one single user name and password.	We recommend using this assertion type. It's simpler to configure.
JWT	You don't have the password and you need to connect to Oracle Analytics Cloud as different users.	This assertion type is more complex to configure. It also allows you to impersonate any BI user in the system, therefore you must ensure that the keys you generate are secured appropriately.

Typical Workflow to Query Oracle Analytics Cloud Semantic Models Remotely

If you're querying Oracle Analytics Cloud semantic models remotely for the first time, follow these tasks as a guide.

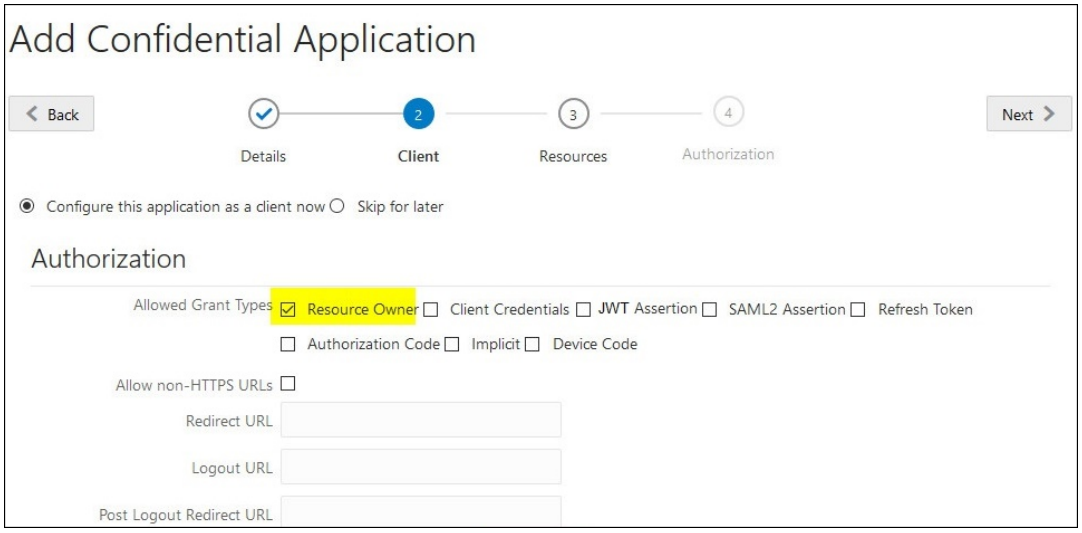
Task	Description	More Information
Decide how you want to secure your JDBC connection	Depending on your security requirements, choose either Resource Owner (recommended) or JSON Web Tokens (JWT) as the assertion type.	Choosing an Assertion Type for Your JDBC Connection
Register the BIJDBC application	Register the BIJDBC application to authenticate your JDBC connection.	(Recommended) Use Resource Owner assertion, see Register the BIJDBC Application Using Resource Owner Assertion. Alternatively, use JWT assertion: • First, generate a private key and certificate that JWT requires, see Generate the Client Private Key and Client Certificate File. • Then, use JWT assertion, see Register the BIJDBC Application using JWT Assertion.
Enable refresh security tokens	Configure your BIJDBC application to refresh security tokens.	Set Up Refresh Security Token
Download the JDBC driver	Download the JDBC driver for Oracle Analytics Cloud.	Download the JDBC Driver
Connect to Oracle Analytics Cloud	Connect to Oracle Analytics Cloud remotely using JDBC. Refer to the example as a guide, see Example: Connect to a Semantic Model Remotely Using SQuirrel.	Connect to Oracle Analytics Cloud Using a JDBC URL

Register the BIJDBC Application Using Resource Owner Assertion

You register the BIJDBC application in Oracle Cloud Infrastructure using Resource Owner assertion to authenticate your public JDBC connections.

1. Sign in to your Oracle Cloud account as an administrator.
2. Navigate to **Identity & Security**, and click **Domains**.

   If your cloud account doesn't offer identity domains, you don't see the **Domains** link. This means your cloud account federates with Oracle Identity Cloud Service. Click **Federation**, select **oracleidentitycloudservice**, and then click the **Oracle Identity Cloud Service Console URL**.
3. Navigate to the **Applications** tab and click **Add**.
4. In the Add Application dialog, click **Confidential Application**.
5. Specify a **Name** (for example, bi-jdbc-connection) , a **Description**, and then click **Next**.
6. Select **Configure this application as a client now**.
7. In **Allowed Grant Types**, click **Resource Owner**.



8. In the **Token Issuance Policy** section:
   a. Under **Authorized Resources**, select **Specific**.

      If your cloud account uses identity domains, select **Add resources**.
   b. Click **Add Scope**.
   c. Select the Oracle Analytics Cloud instance you want to connect to (for example, select AUTOANALYTICSINST_<*my_instance_ID*>).
   d. Click **Add.**
9. Click **Next** and then **Finish** to display an Application Added window.
10. Copy the **Client ID** and **Client Secret** to use later.
11. Close the Application Added page.
12. Click **Activate**, then click **Activate Application**.
13. Click **Save** to display a confirmation message.

Generate the Client Private Key and Client Certificate File

If you decide to secure your JDBC connection using the JWT assertion type, you generate a private key and certificate to authenticate the connection.

Note: You don't need a private key and certificate file if you're securing your JDBC connection using the Resource Owner assertion type.

See https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html.

1. Generate a key pair and key store.

   From a command prompt, issue a `keytool` command, using the command format:

   ```
   keytool -genkeypair -v -keystore <keystore name> -storetype <store type i.e
   PKCS12> -storepass <store pass> -keyalg <key algorithm> -keysize <key size> -
   sigalg <sig algorithm> -validity <validity days> -alias <alias name> -keypass
   <key pass>
   ```

   For example:

   ```
   keytool -genkeypair -v -keystore bijdbckeystore.jks -storetype PKCS12 -
   storepass password -keyalg RSA -keysize 2048 -sigalg SHA256withRSA -validity
   3600 -alias bijdbcclientalias -keypass password
   ```

2. Generate a public certificate.

   From a command prompt, issue a `keytool` command, using the command format:

   ```
   keytool -exportcert -v -alias <alias name> -keystore <keystore name> -
   storetype <store type, such as PKCS12> -storepass <store pass> -file
   <certificate file> -rfc
   ```

   For example:

   ```
   keytool -exportcert -v -alias bijdbcclientalias -keystore bijdbckeystore.jks -
   storetype PKCS12 -storepass password -file bijdbcclient.cert -rfc
   ```

3. Use OpenSS to extract the private key, in PKCS8 format, from the keystore file.

   Use the command format:

   ```
   openssl pkcs12 -in <keystore file name> -passin pass:<keystore password> -
   nodes -nocerts -nomacver > <PKCS8 key file path>
   ```

   For example:

   ```
   openssl pkcs12 -in bijdbckeystore.jks -passin pass:password -nodes -nocerts -
   nomacver |sed -n '/BEGIN PRIVATE KEY/,$p' > bijdbcclient.pem
   ```

4. Save the generated key and certificates in a location accessible to your client machine.

Register the BIJDBC Application using JWT Assertion

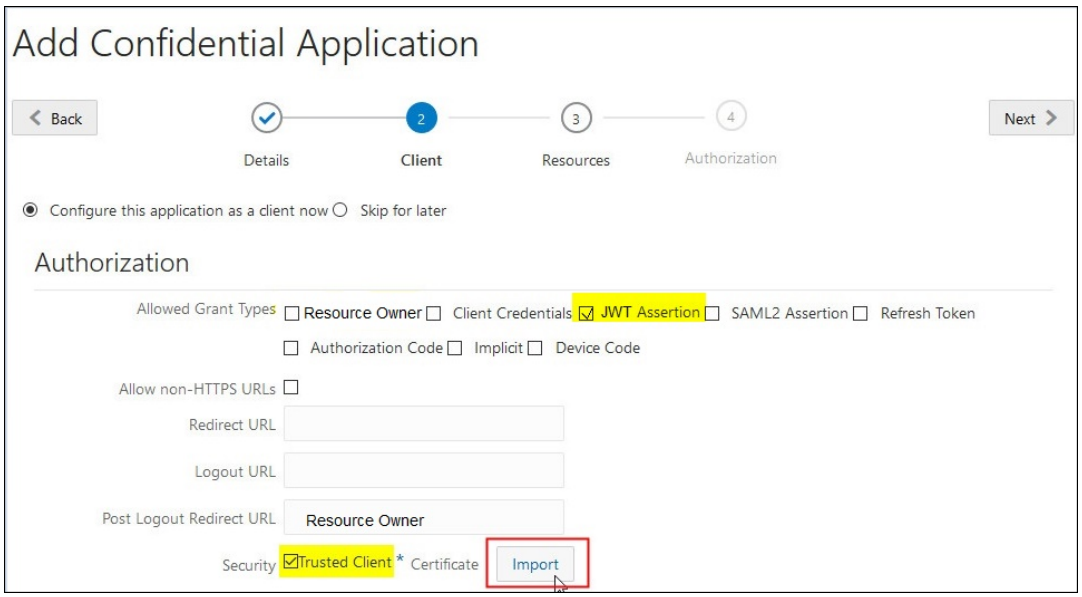
You register the BIJDBC application in Oracle Cloud Infrastructure using JWT assertion to authenticate your public JDBC connections.

Before you start, generate a Client Private Key and Client Certificate File as specified in the previous step.

1. In Oracle Cloud Infrastructure Console. navigate to **Identity & Security**, and click **Domains**.

If your cloud account doesn't offer identity domains, you don't see the **Domains** link. This means your cloud account federates with Oracle Identity Cloud Service. Click **Federation**, select **oracleidentitycloudservice**, and then click the **Oracle Identity Cloud Service Console URL**.

2. Navigate to the **Applications** tab and click **Add**.
3. In the Add Application dialog, click **Confidential Application**.
4. Specify a **Name** (for example, bi-jdbc-connection) , a **Description**, and then click **Next**.
5. Select **Configure this application as a client now**.
6. In **Allowed Grant Types**, click **JWT Assertion**.
7. For **Security**:
   a. Select **Trusted Client**.
   b. Click **Import** , enter a **Certificate Alias**, and then upload your client certificate file.



8. In the **Token Issuance Policy** section:
   a. Under **Authorized Resources**, select **Specific**.

      If your cloud account uses identity domains, select **Add resources**.
   b. Click **Add Scope**.
   c. Select the Oracle Analytics Cloud instance you want to connect to (for example, select AUTOANALYTICSINST_<*my_instance_ID*>).
   d. Click **Add.**
9. Click **Next** and then **Finish** to display an Application Added window.
10. Copy the **Client ID** and **Client Secret** to use later.
11. Close the Application Added page.
12. Click **Activate**, then click **Activate Application**.
13. Click **Save** to display a confirmation message.

Set Up Refresh Security Token

Configure your BIJDBC application to refresh security tokens.

1. In Oracle Cloud Infrastructure Console, navigate to **Identity & Security**, and click **Domains**.

   If your cloud account doesn't offer identity domains, you don't see the **Domains** link. This means your cloud account federates with Oracle Identity Cloud Service. Click **Federation**, select **oracleidentitycloudservice**, and then click the **Oracle Identity Cloud Service Console URL**.

2. Enable the **Refresh Token** option for the BIJDBC application that you created earlier.

   a. Navigate to **Applications**, and click the name of the BIJDBC application that you created earlier.

   b. Click **Edit OAuth Configuration**, then **Client Configuration**.

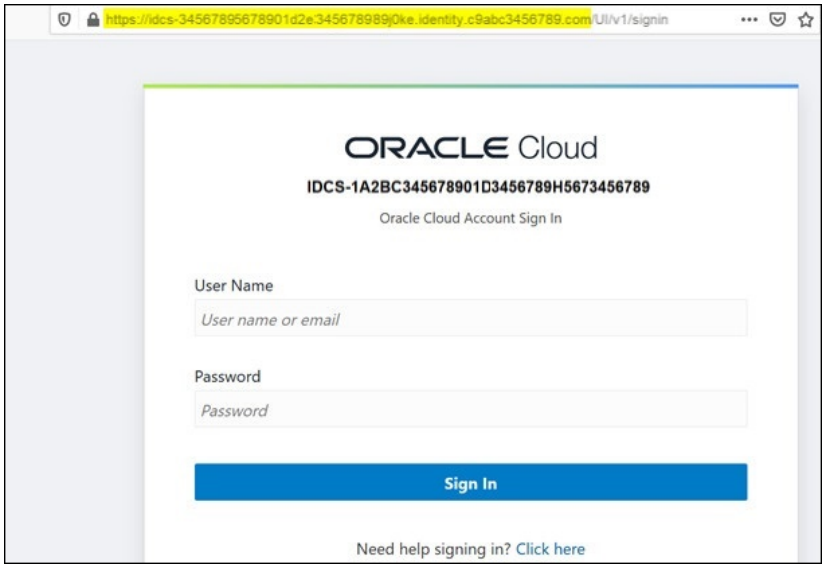
      If your cloud account uses Oracle Identity Cloud Service, select **Configuration** then **Client Configuration**.

   c. Select **Refresh Token**, and click **Save**.

If your Oracle Analytics Cloud instance was created after 12th May 2020, your BIJDBC application is now configured to refresh security tokens.

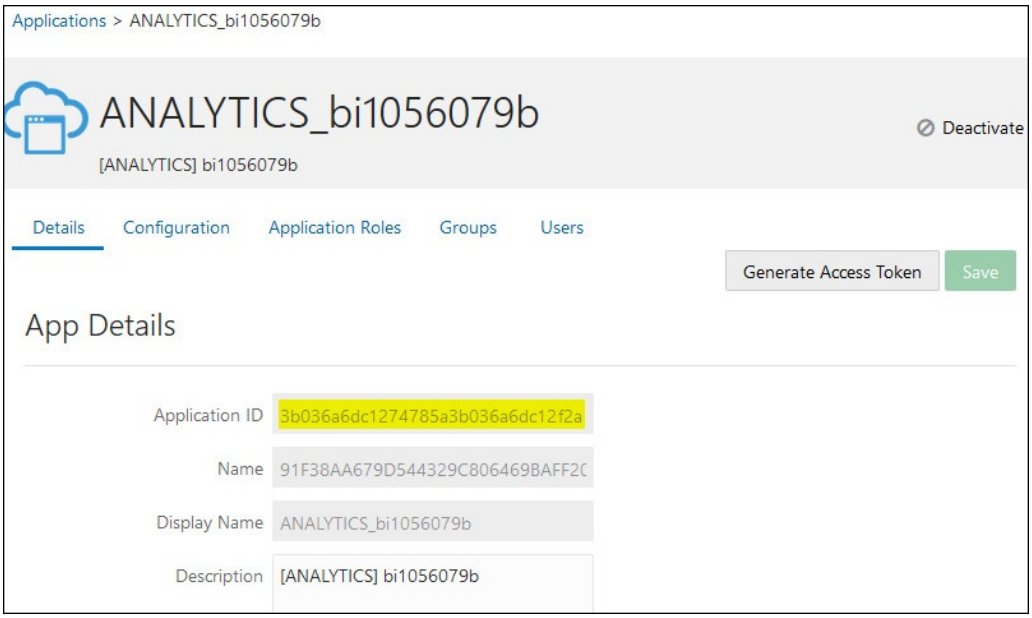
If your Oracle Analytics Cloud instance was created before 12th May 2020, perform the additional steps 2 to 4.

3. Make a note of your identity management host name displayed when you sign in to your Oracle Cloud account.

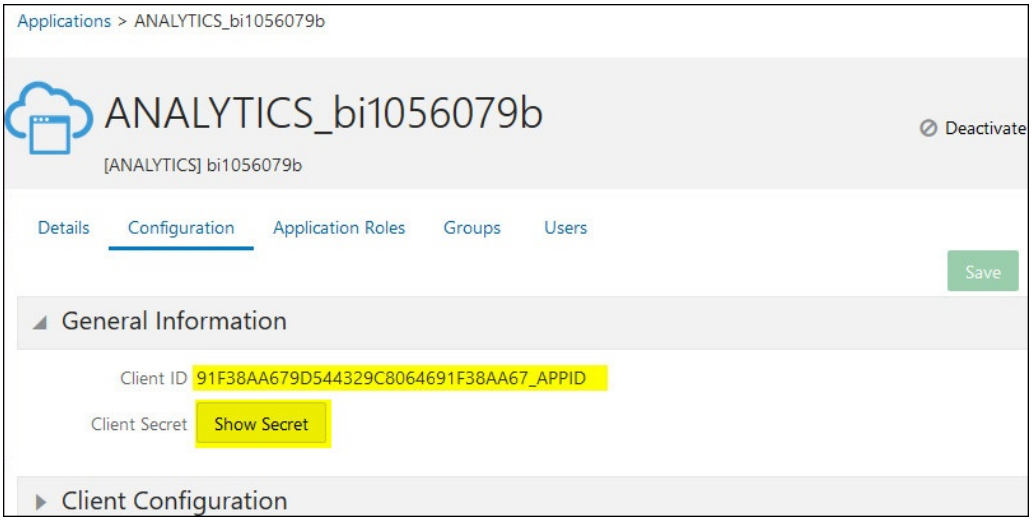


4. Navigate to the application associated with the Oracle Analytics Cloud you want to connect to, and make a note of the **Application ID**, **Client ID** and **Client Secret**.

   - **Application ID** (For the application associated with Oracle Analytics Cloud)



- **Client ID and Client Secret** (For the application associated with Oracle Analytics Cloud)



5. Use the REST API to generate the access token using client credentials.

   Use this REST API command format:

   ```
   curl --insecure -i -u '<Client-ID>:<Client-Secret>' -H "Content-Type:
   application/x-www-form-urlencoded;charset=UTF-8" --request POST https://<IDCS-
   Host>/oauth2/v1/token -d
   "grant_type=client_credentials&scope=urn:opc:idm:__myscopes__"
   ```

   Parameters:

   - **Client-ID**: Client ID of the application associated with your Oracle Analytics Cloud instance.
   - **Client-Secret**: Client secret for the application associated with your Oracle Analytics Cloud instance.
   - **IDCS-Host**: Host name you noted earlier.

   See Generate Access Token and Other OAuth Runtime Tokens to Access the Resource.

6. Use the REST API to update the flag.

   Use this REST API command format:

   ```
   curl --location --request PATCH 'https://<IDCS-Host>/admin/v1/Apps/
   <Application-Id>' \ --header 'Authorization: Bearer <Access-token>' \ --header
   'Content-Type: application/json' \ --data-raw '{ "schemas":
   [ "urn:ietf:params:scim:api:messages:2.0:PatchOp" ], "Operations": [{ "op":
   "replace", "path": "allowOffline", "value": true }] }'
   ```

   Parameters:

   - **IDCS-Host:** Host name you noted earlier.
   - **Application-Id:** ID of the application associated with your Oracle Analytics Cloud instance.
   - **Access-token:** Access token value you generated earlier.

Download the JDBC Driver

Obtain the JDBC driver JAR file (`bijdbc-all.jar`) from an Oracle Analytics Cloud Client Tools installation on a Windows machine.

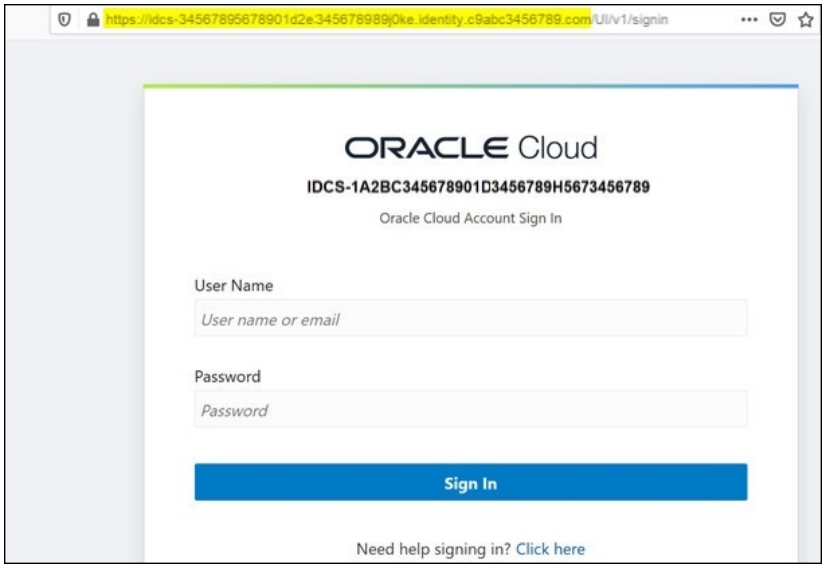
If you haven't done so already, download and install Oracle Analytics Cloud Client Tools on a Windows machine. If you want to connect to Oracle Analytics Cloud from an iOS machine, you must copy the JDBC driver file from the Windows installation folder to your iOS machine.

1. Download the latest Oracle Analytics Client Tools.
   a. Go to Download page for Oracle Analytics Client Tools.
   b. To start the download, click the **Oracle Analytics Client Tools** link that matches your Oracle Analytics Cloud environment.
   In most cases, this is the latest available update.
   c. Accept the Oracle license agreement if prompted, and then click the download link to download the software to your local machine.
2. Install Oracle Analytics Client Tools on your local machine.
   a. Unzip the file that you downloaded to extract the installer file `setup_bi_client-<update ID>-win64.exe` file.
   b. Double-click the file `setup_bi_client-<update ID>-win64.exe` to start the installer.
   c. Follow the on-screen instructions.
3. From the installation folder, copy the JDBC driver file `<OH>/bi/bifoundation/jdbc/bijdbc-all.jar`.
   To connect to Oracle Analytics Cloud from an iOS machine, copy the `bijdbc-all.jar` file to your iOS machine.

Connect to Oracle Analytics Cloud Using a JDBC URL

Determine the JDBC URL required to connect to your Oracle Analytics Cloud instance and test the connection.

1. Sign in to your cloud account and make a note of your identity management host name displayed on the sign in page.

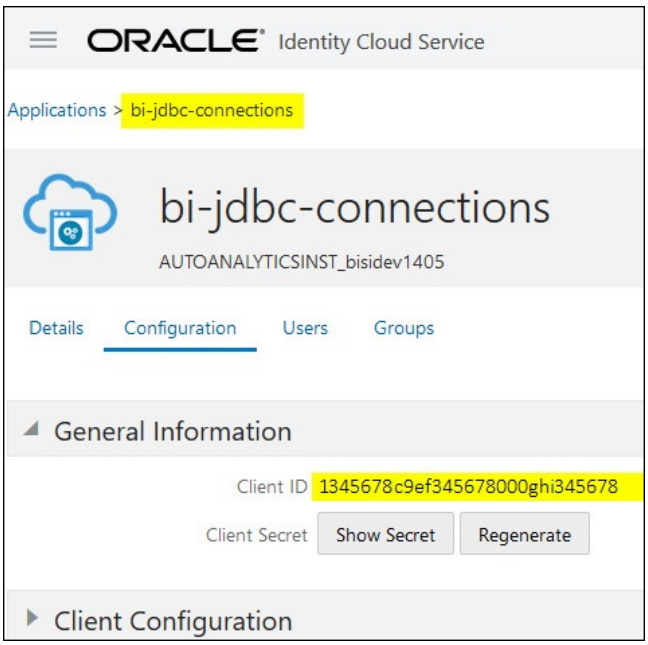


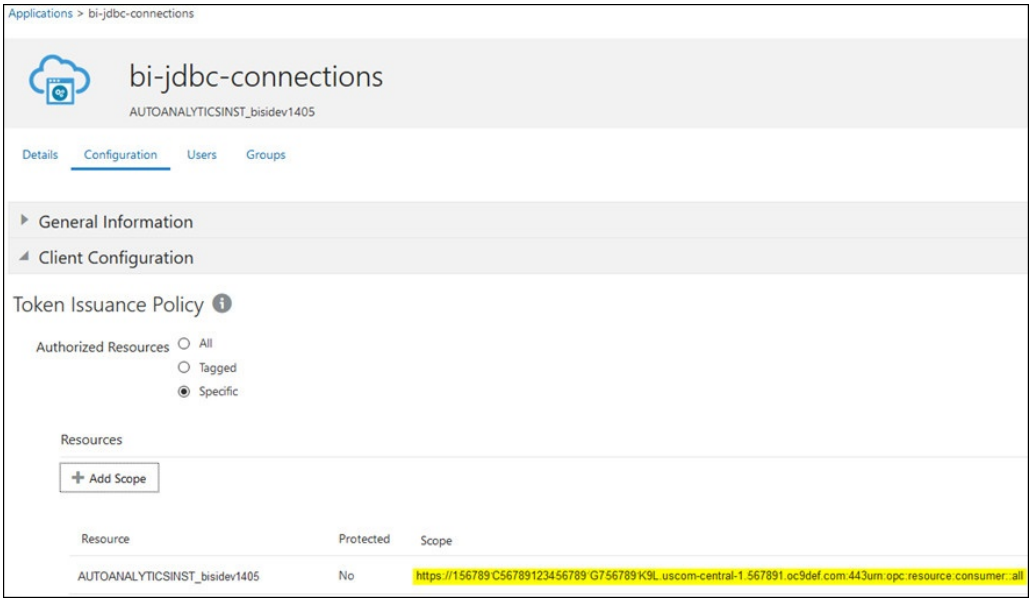
2. In Oracle Cloud Infrastructure Console, navigate to **Identity & Security**, and click **Domains**.

   If your cloud account doesn't offer identity domains, you don't see the **Domains** link. This means your cloud account federates with Oracle Identity Cloud Service. Click **Federation**, select **oracleidentitycloudservice**, and then click the **Oracle Identity Cloud Service Console URL**.

3. Navigate to the **Applications** tab and click the name of your BIJDBC application.

4. Make a note of the Client ID and Client Scope:



- 

5. Create a `bijdbc.properties` file for `OAuth` authentication and authorization and add credentials for your Oracle Analytics Cloud instance.

   To connect using Resource Owner assertion, use the following format for your `bijdbc.properties` file:

   ```
   idcsEndpointUrl=https://<IDCS_hostname>
   idcsClientId=<ID string>
   idcsClientScope=<ID string>
   idcsClientSecret=<secret>
   user=<firstname.lastname@example.com>
   password=<password>
   ```

   For example:

   ```
   idcsEndpointUrl=https://
   idcs-1a2bc345678901d2e34fgh56789j0ke.identity.c9abc1.oc9def.com
   idcsClientId=12a000dc9ef345678000ghij2kl8a34
   idcsClientScope=https://<host>.com:443urn:opc:resource:consumer::all
   idcsClientSecret=xyz
   user=myuser@office.com
   password=yourpassword
   ```

   If you're using JWT assertion, add the following:

   ```
   user=<firstname.lastname@example.com>
   idcsEndpointUrl=https://<IDCS_hostname>
   idcsClientId=<ID string>
   idcsClientScope=<ID string>
   certificateFile=<location>\jdbc\\bijdbcclient.cert
   privateKeyFile=<location>\jdbc\\bijdbcclient.pem
   ```

6. Determine the URL required to connect to your Oracle Analytics Cloud instance. The format you use depends on when and how the instance was deployed.

To connect to an instance deployed on	Create date
Oracle Cloud Infrastructure (Gen 2)	Any
Oracle Cloud Infrastructure	12th May 2020 or later

Use this URL format with `OAuth`:

```
jdbc:oraclebi:https://<host>:<port>/api/jdbc?BIJDBC_PROPERTIES_FILE=<fully
qualified location and name of properties file>
```

For example:

```
jdbc:oraclebi:https://<host>:443/api/jdbc?BIJDBC_PROPERTIES_FILE=D:\
\Workspace\\bijdbc\\bijdbc.properties
```

To connect to an instance deployed on	Create date
Oracle Cloud Infrastructure	Before 12th May 2020

Use this URL format with `OAuth`:

```
jdbc:oraclebi:https://<host>:<port>/bimodeler/api/jdbc?
BIJDBC_PROPERTIES_FILE=<fully qualified location and name of properties
file>
```

For example:

```
jdbc:oraclebi:https://abcdefghi123-jklmnopqrs4t-
je.analytics.ocp.oraclecloud.com:443/bimodeler/api/jdbc?
BIJDBC_PROPERTIES_FILE=D:\\Workspace\\bijdbc\\bijdbc.properties
```

7. Test the connection to the target Oracle Analytics Cloud instance.

   Use your favorite SQL command tool to connect to Oracle Analytics Cloud with the appropriate JBDC URL. For example:

   ```
   jdbc:oraclebi:https://abcdefghi123-jklmnopqrs4t-
   je.analytics.ocp.oraclecloud.com:443/api/jdbc?BIJDBC_PROPERTIES_FILE=D:\
   \Workspace\\bijdbc\\bijdbc.properties
   ```

Example: Connect to a Semantic Model Remotely Using SQuirrel

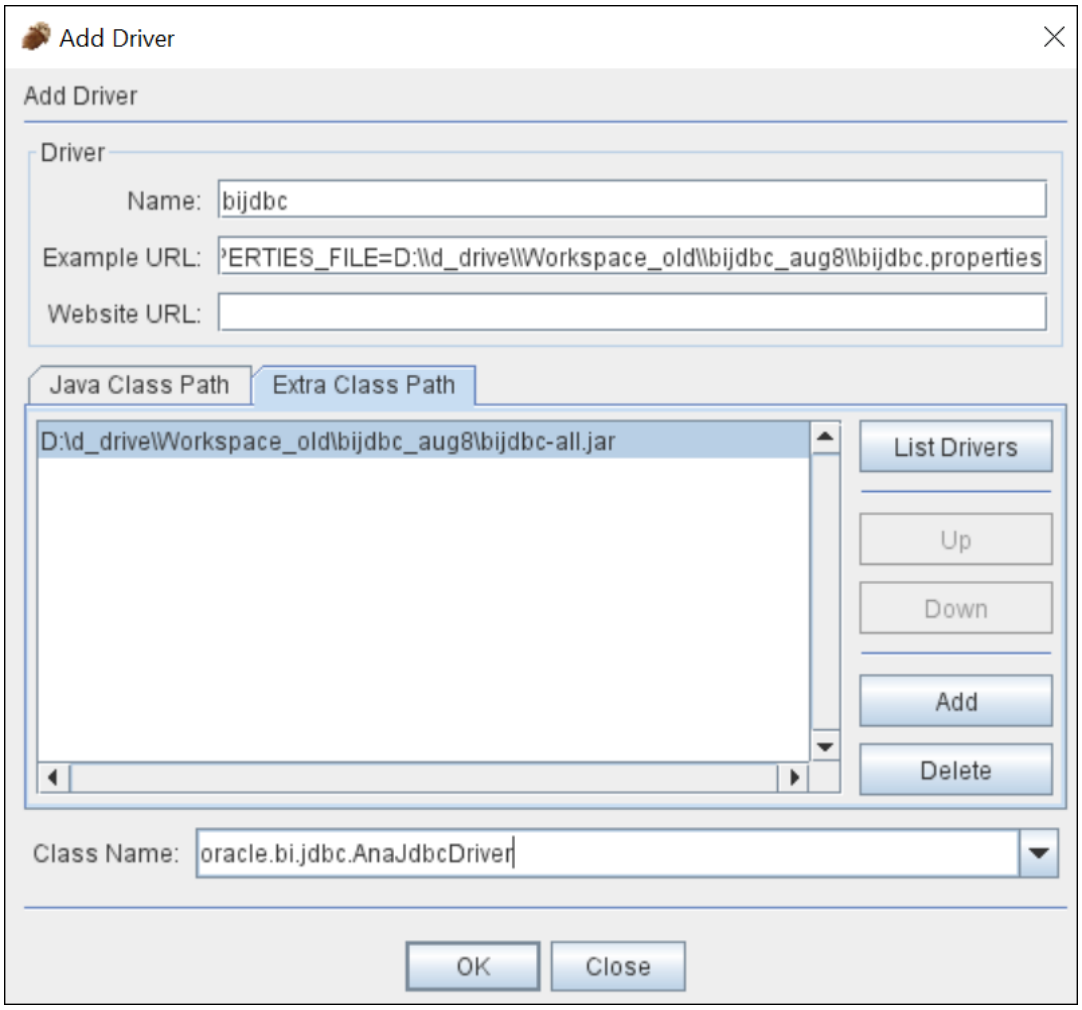
This example shows how to connect to a Oracle Analytics Cloud semantic model using JDBC with SQuirrel SQL Client tool.

1. Register the JDBC driver.
   a. In SQuirrel SQL Client, under **Drivers**, click **Create a New Driver**.
   b. In the **Example URL** field, specify the BIJDBC application URL with a fully qualified properties file.

      For example: `jdbc:oraclebi:https://abcdefghi123-jklmnopqrs4t-je.analytics.ocp.oraclecloud.com:443/bimodeler/api/jdbc?BIJDBC_PROPERTIES_FILE=D:\\Workspace\\bijdbc\\bijdbc.properties`

c. In the **Extra Class Path** tab, select the BIJDBC driver (JAR file) you downloaded from Client Installer.

d. Click **List Drivers**, and under **Class Name** select `oracle.bi.jdbc.AnaJdbcDriver`, then save the details.



2. Create a connection or (alias).

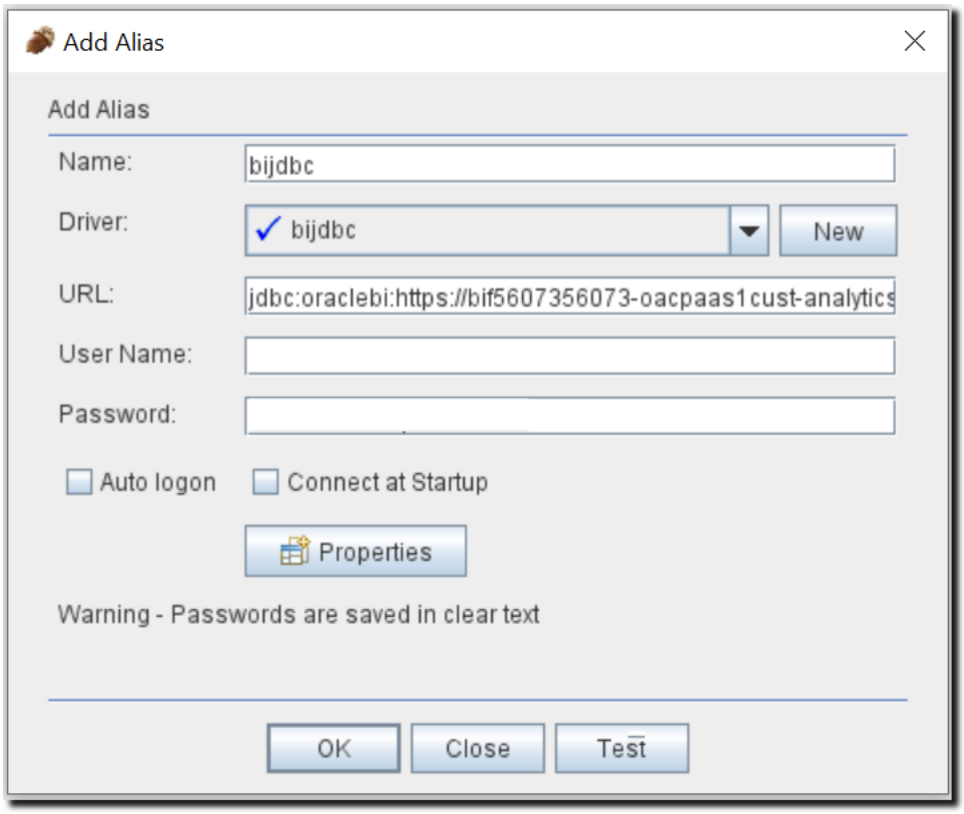
a. Under **Aliases**, click **Create a New Alias**.

b. In the **Driver** option, select `bijdbc`.

c. Edit the **URL**, specify credentials (if required), then click **Test**.

If credentials are provided in the property file, you don't need to specify **User Name** or **Password**.

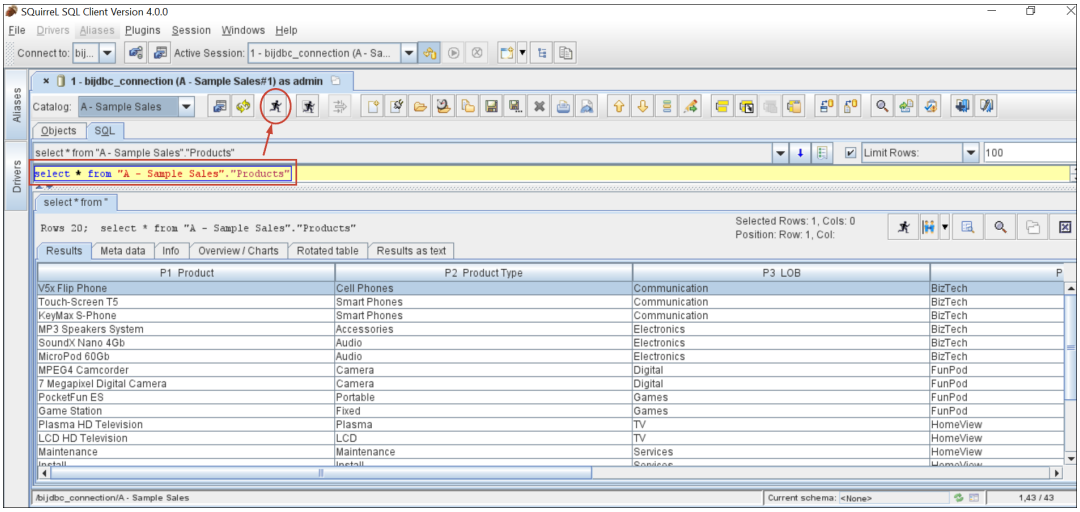
d. Validate the connection by connecting to the Alias and exploring the metadata in the **Objects** section.



3. In the **SQL** tab, enter a sample logical SQL query and click the **Run** button.

   For more information, see Logical SQL Reference Guide.

   If the connection is working, the **Results** tab shows the results of your query.



4. Check the **Results** tab to verify the rows returned by the query.

10

Connect to Databases Deployed on a Public IP Address

You can use Oracle Analytics Cloud to connect to databases with a public IP address so that end users can analyze that data in visualizations, analyses, and pixel-perfect reports.

For example, you might want to analyze data in a database deployed on Oracle Cloud Infrastructure or Oracle Cloud Infrastructure Classic.

Topics:

Connect to a Database Deployed on Oracle Cloud Infrastructure with a Public IP Address

Configure Oracle Analytics Cloud to connect to a database deployed on Oracle Cloud Infrastructure with a public IP address, so that end users can analyze that data in visualizations, analyses, and pixel-perfect reports.

Topics

Typical Workflow to Connect to a Database Deployed on Oracle Cloud Infrastructure

If you're connecting to an database deployment on Oracle Cloud Infrastructure for the first time, follow these tasks as a guide.

Task	Description	More Information
Verify the prerequisites	Verify that your environment satisfies the prerequisites required for this configuration.	Prerequisites
Record database information	Record connection information for database.	Record Database Information

Task	Description	More Information
Enable database access	Add an ingress rule to give Oracle Analytics Cloud access to the database.	Enable Database Access Through Port 1521
Connect to the database	Create and test your connections.	Connect to Your Database from Oracle Analytics Cloud

Prerequisites

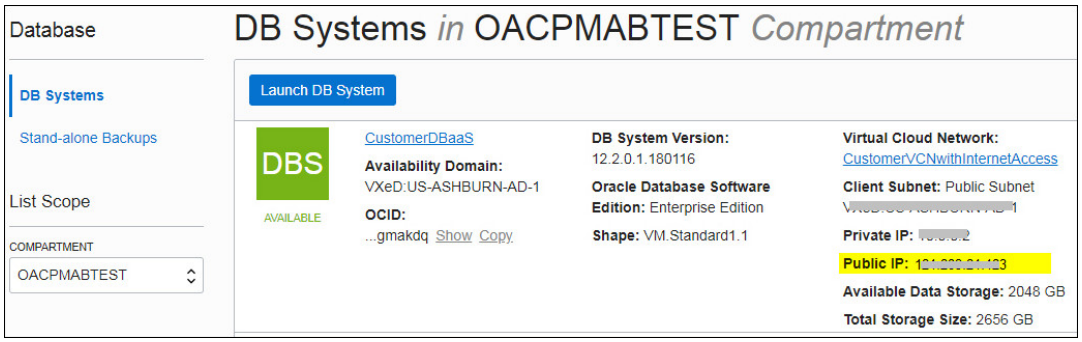
Before you start, make sure you have the required environment.

Step	Description	Important Information to Note
Set up Oracle Analytics Cloud	Deploy Oracle Analytics Cloud.	**Region** **Availability Domain**
Set up a Virtual Cloud Network (VCN) on Oracle Cloud Infrastructure	Set up a VCN for the database deployment on Oracle Cloud Infrastructure. **Note:** The VCN must be in the same Region and Availability Domain as Oracle Analytics Cloud.	**Virtual Cloud Network** **Subnet** Same: • **Region** • **Availability Domain**
Deploy a database: • Deploy the database on the VCN in Oracle Cloud Infrastructure • Populate the database with data • Set up a database user with permissions to read database tables	Deploy a database on the VCN in Oracle Cloud Infrastructure. **Note:** The database must be in the same Region and Availability Domain as the VCN.	**Public IP** **Database Unique Name** **Host Domain Name** **Database User/Password** Same: • **Region** • **Availability Domain** • **Virtual Cloud Network** • **Client Subnet**

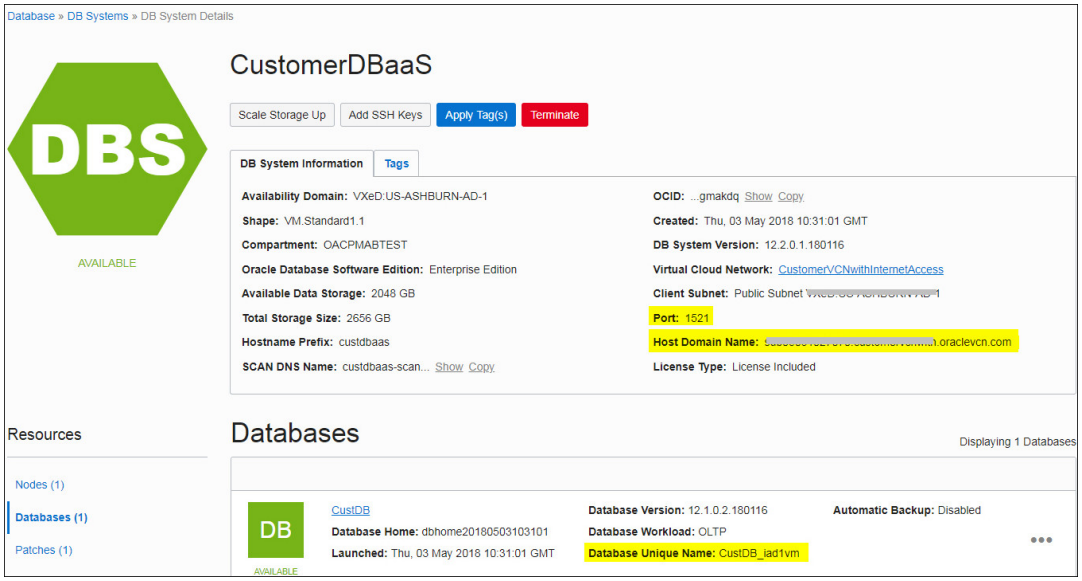
Record Database Information

All the information you need to connect to a database is available in the Oracle Cloud Infrastructure Console. Record the information now, so you have the required details when you set up the connection in Oracle Analytics Cloud.

1. In Oracle Cloud Infrastructure Console, click ☰ in the top left corner.
2. Click **Databases**. Under **MySQL**, click **DB Systems**.
3. Locate the database you want to connect to and record the **Public IP** address.



4. Click the name of the database you want to connect to and write down the values in these fields: **Database Unique Name**, **Host Domain Name**, **Virtual Cloud Network**, **Client Subnet**, and **Port**.

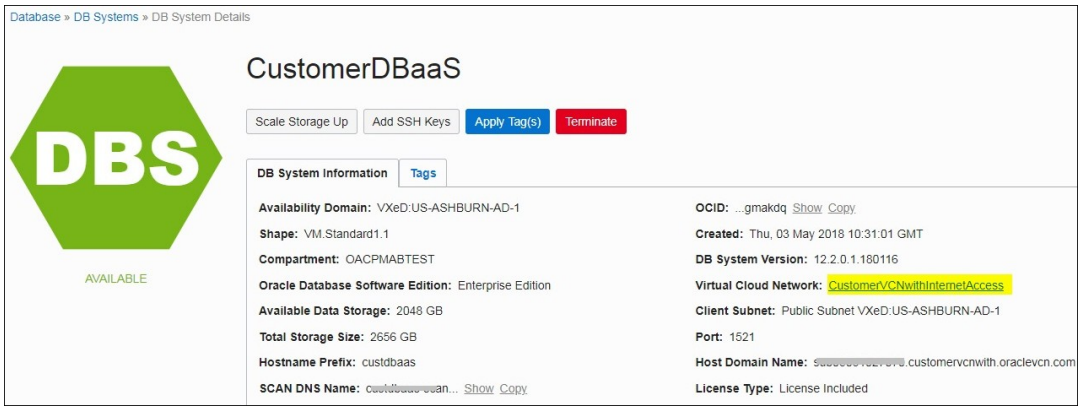


5. Find out the user name and password of a database user with permissions to read from this database, and write them down as you need these later. For example, the user SYSTEM.

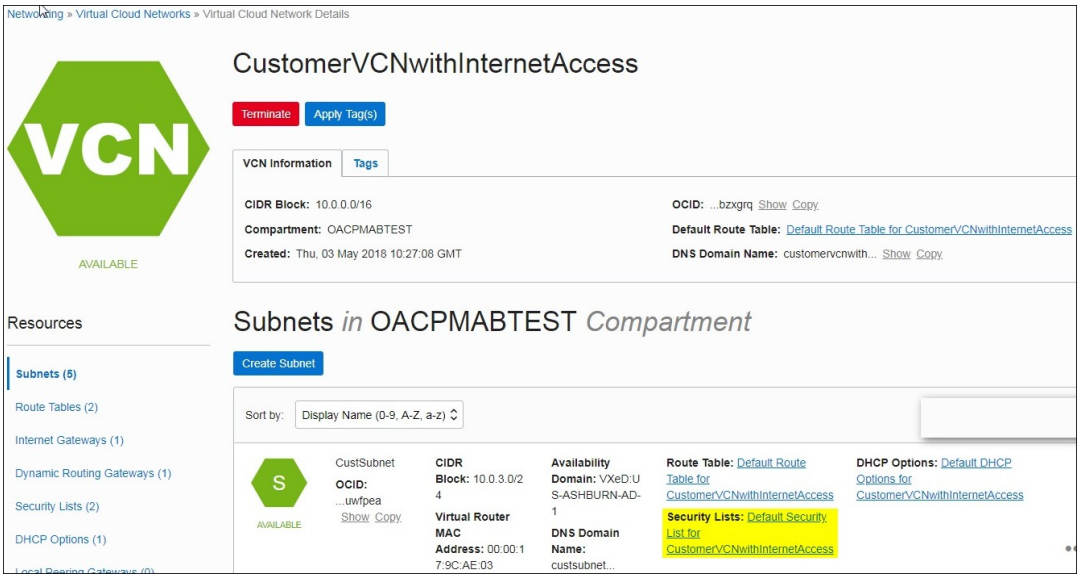
Enable Database Access Through Port 1521

Add an ingress rule that enables Oracle Analytics Cloud to access the database through port 1521.

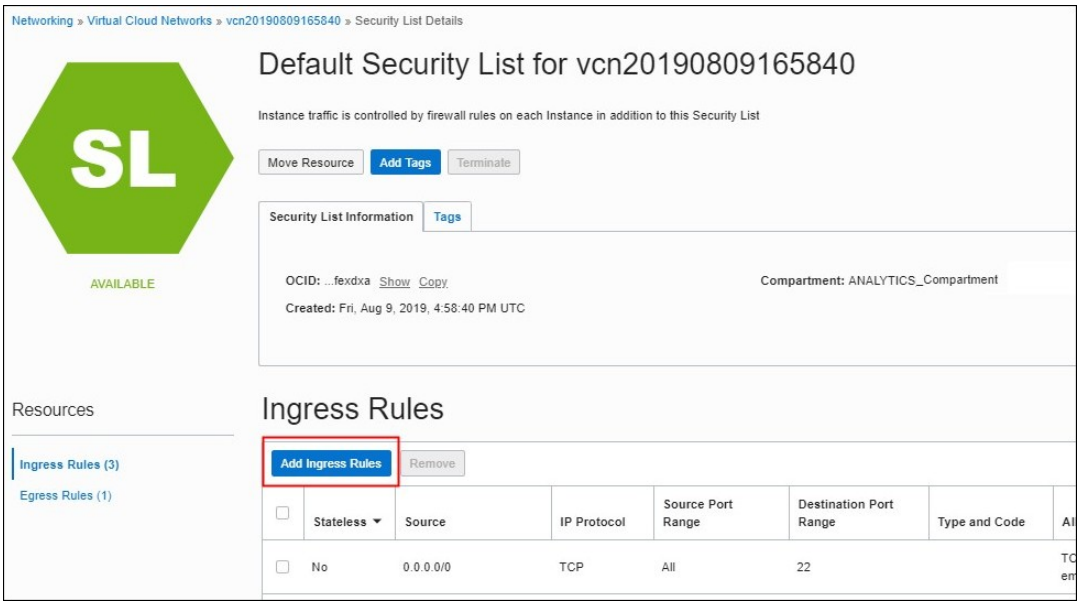
1. Make a note of the Oracle Analytics Cloud IP addresses that you want to allow access to.
2. In Oracle Cloud Infrastructure Console, click ☰ in the top left corner, and click **Databases**. Under **MySQL**, click **DB Systems**.
3. Click the database that you want to connect to.
4. Click the **Virtual Cloud Network** link.



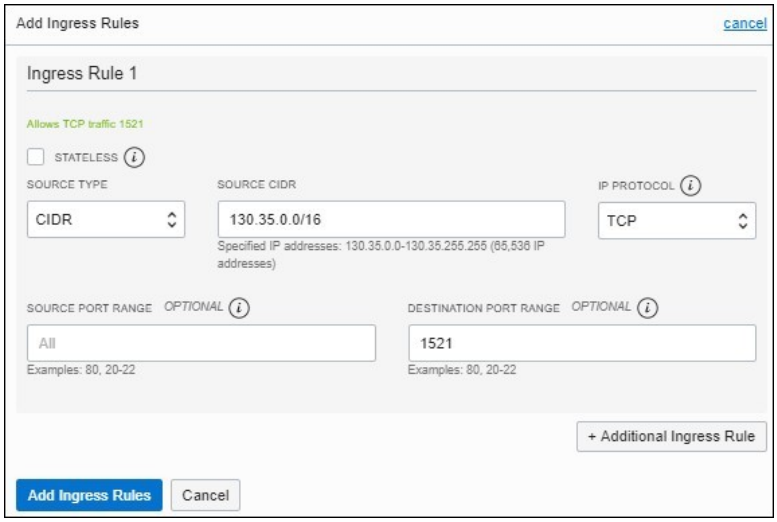
5. Navigate to the appropriate subnet, and under **Security Lists**, click **Default Security List For <VCN>**.



6. Click **Add Ingress Rules**.



7. For each IP address that you want to give access to, add an ingress rule to allow any incoming traffic from the public internet to reach port 1521 on this database node, with the following settings:
   - **SOURCE CIDR**: Enter the IP address that you wrote down in Step 1.
   - **IP PROTOCOL**: TCP
   - **SOURCE PORT RANGE**: All
   - **DESTINATION PORT RANGE**: 1521
   - **Allows**: TCP traffic for ports: 1521



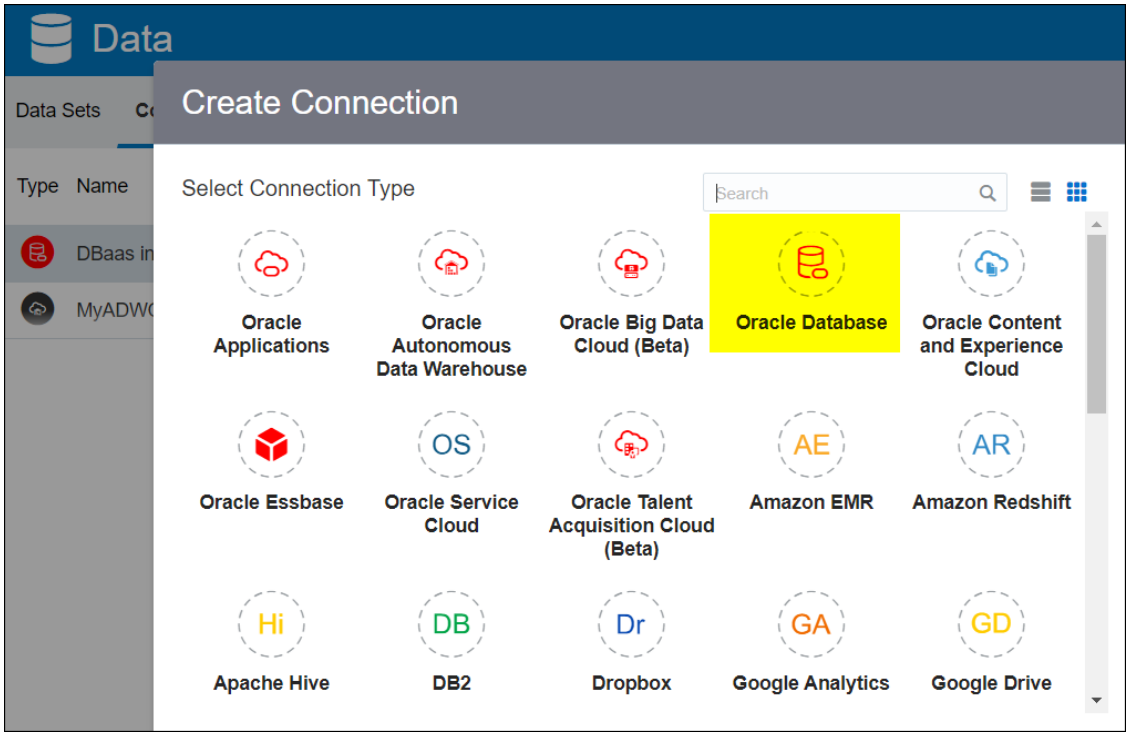
Connect to Your Database from Oracle Analytics Cloud

After enabling access to the database, use the database connection information you wrote down earlier to connect Oracle Analytics Cloud to the database. The way you connect to the database depends on what you want to do with the data.

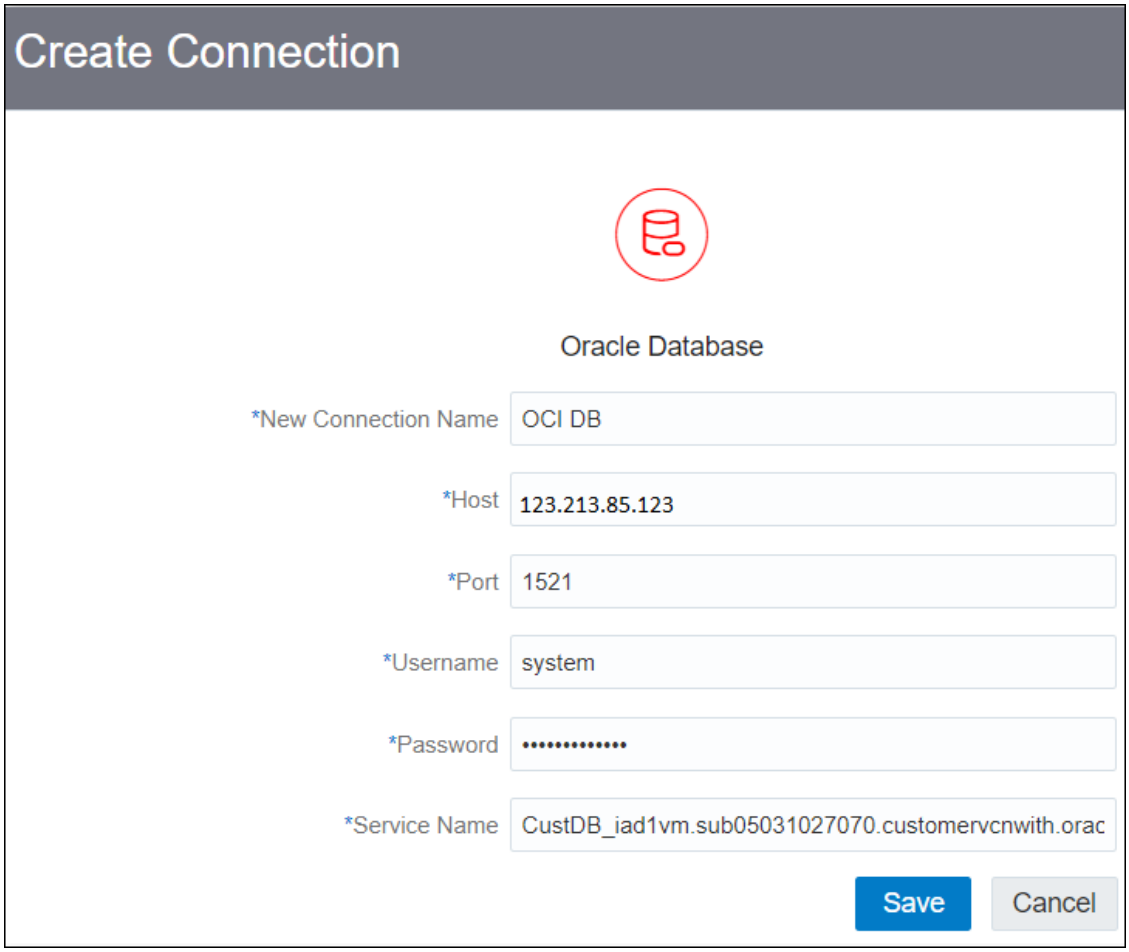
- Visualize the data.
- Model the data using Semantic Modeler, then generate analyses and dashboards.
- Model the data with Oracle Analytics Cloud Model Administration Tool, then generate analyses and dashboards.
- Publish the data in pixel-perfect reports.

Connect to Your Database for Data Visualization or Semantic Modeler

In Oracle Analytics Cloud, create an Oracle Database connection for data visualizations in the usual way. See Create Database Connections.



Use the database details you recorded earlier to fill in the Create Connection dialog.



Specify these values:

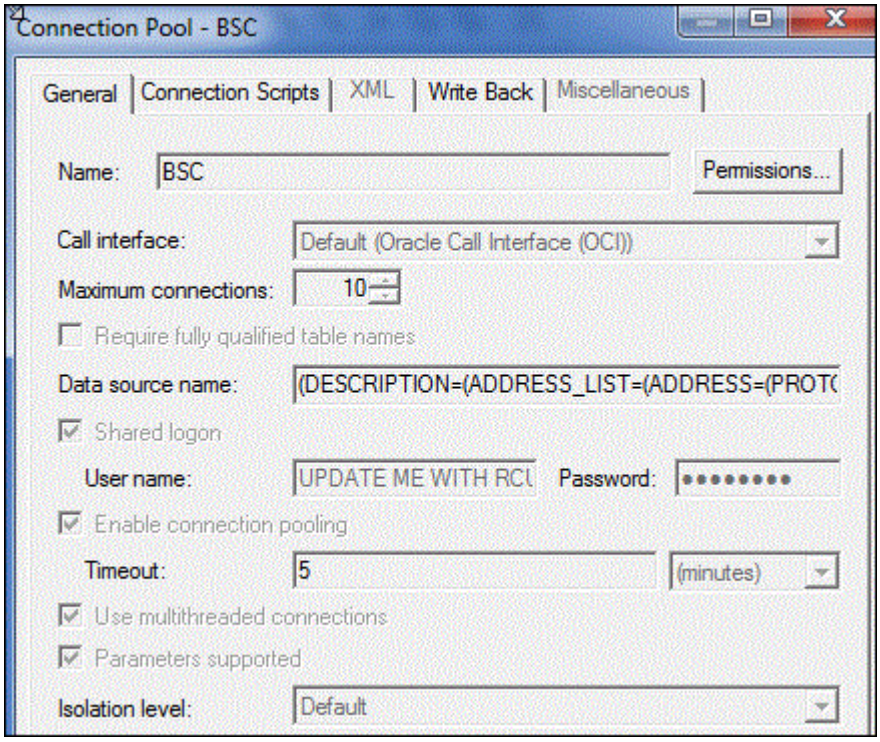
- **New Connection Name**: A name for the database you want to connect to.
- **Host**: The **Public IP** address for the database instance. For example, 123.213.85.123.
- **Port**: The port number that enables access to the database. For example, 1521.
- **Username**: The name of a user with read access to the database.
- **Password**: The password for the specified database user.
- **Service Name**: A concatenated name comprising **Database Unique Name** and **Host Domain Name**, separated with a period. For example, `CustDB_iad1vm.sub05031027070.customervcnwith.oraclevcn.com`.

Connect to Your Database in Model Administration Tool

In Model Administration Tool for Oracle Analytics Cloud, click **File**, then **Open**, then **In the Cloud** to open your semantic model. See Edit a Semantic Model in the Cloud.

When you sign in, use connection information for your Oracle Analytics Cloud to fill in the Open in the Cloud dialog.

Create a connection pool for your database. In the Physical pane, expand the **DBaaS** node, right-click the database icon, and click **Properties** to display the Connection Pool dialog. Use the database details you recorded earlier to specify **Call Interface**, **Data Source Name**, **User Name**, and **Password**.



Specify these values:

- **Call interface**: Select **Default (Oracle Call Interface (OCI)**).
- **Data Source Name**: Specify the connection details. For example:

  ```
  (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=129.213.85.177)
  (PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=CustDB_iad1vm.sub05031027070.customervcnwith.oraclevcn.com))
  ```

  For SERVICE_NAME, specify the concatenated **Database Unique Name** and **Host Domain Name** separated by a period, for example, db1_phx1tv.mycompany.com. To find both these names in Oracle Cloud Infrastructure Console, click **Databases**, under **MySQL** click **DB Systems**, and then click the name of your database.

Connect to Oracle Autonomous Data Warehouse with a Public IP Address

Configure Oracle Analytics Cloud to connect to Autonomous Data Warehouse over a public IP address so that end users can analyze that data in visualizations, analyses, dashboards, and pixel-perfect reports.

Topics

Typical Workflow to Connect to Oracle Autonomous Data Warehouse with a Public IP Address

If you're connecting Oracle Analytics Cloud to Autonomous Data Warehouse over a public IP address for the first time, follow these tasks as a guide.

Task	Description	More Information
Verify the prerequisites	Verify that your environment satisfies the prerequisites required for this configuration.	Prerequisites
Enable access to Autonomous Data Warehouse	Upload your Autonomous Data Warehouse Client Credentials file (wallet file) to Oracle Analytics Cloud.	Enable Access to Oracle Autonomous Data Warehouse
Connect to Autonomous Data Warehouse	Create and test your connections.	Connect to Oracle Autonomous Data Warehouse

Prerequisites

Before you start, make sure you have the required environment.

Step	Description	Important Information to Note
Set up Oracle Analytics Cloud	Deploy Oracle Analytics Cloud.	**Region** **Availability Domain**
Set up Oracle Autonomous Data Warehouse	Deploy Autonomous Data Warehouse. • Deploy Autonomous Data Warehouse on Oracle Cloud Infrastructure. • Populate Autonomous Data Warehouse with data. • Set up a database user with permissions to read database tables on Autonomous Data Warehouse	**Host Name** **Port Number** **Service Name** (Obtain these details from `tnsnames.ora` in the Autonomous Data Warehouse Client Credentials file.)

Enable Access to Oracle Autonomous Data Warehouse

To enable secure communication between Oracle Analytics Cloud and Autonomous Data Warehouse, you upload trusted SSL certificates to Oracle Analytics Cloud.

1. In Autonomous Data Warehouse Console, obtain the Client Credentials file.

   The Client Credentials file is a ZIP file containing the files `cwallet.sso` and `tnsnames.ora`. See Download Client Credentials (Wallets) in *Using Oracle Autonomous Data Warehouse*.

2. Extract the `cwallet.sso` file from the Client Credentials file.
3. Upload the `cwallet.sso` file to Oracle Analytics Cloud.
   a. Sign in to Oracle Analytics Cloud, open the **Console** and click **Connections**.
   b. Click **Upload Wallet** to upload a wallet for the first time or **Replace Wallet** to update an existing wallet.

c. Click **Browse** and locate the wallet file (`cwallet.sso`) you downloaded from Autonomous Data Warehouse.

d. Select the file and click **Open**.

e. Click **Update** and **OK** to update the existing wallet file.

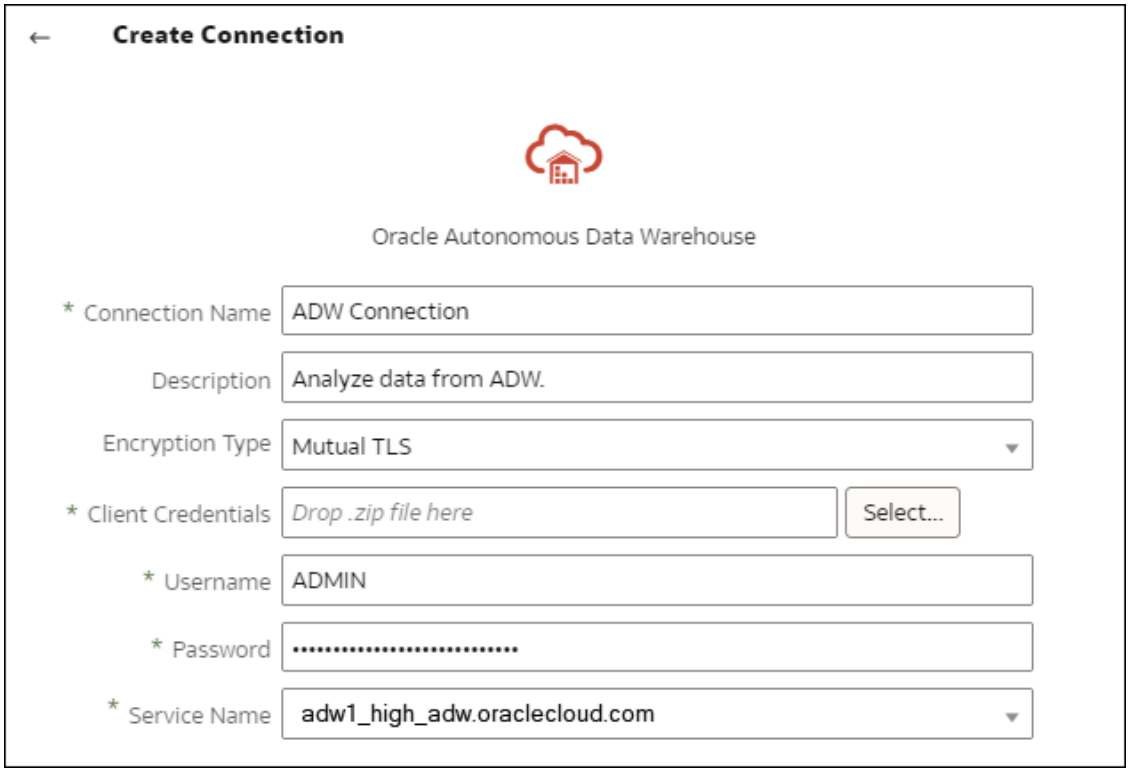
Connect to Oracle Autonomous Data Warehouse

After enabling access to Oracle Autonomous Data Warehouse, use the connection details you recorded earlier to connect Oracle Analytics Cloud to Autonomous Data Warehouse. The way you connect depends on what you want to do with the data.

- Visualize the data
- Model the data using Semantic Modeler, then generate analyses and dashboards.
- Model the data with Oracle Analytics Model Administration Tool, then generate analyses and dashboards.
- Publish the data in pixel-perfect reports.

Connect to Autonomous Data Warehouse for Data Visualization or Semantic Modeler

In Oracle Analytics Cloud, create an Autonomous Data Warehouse connection for data visualization. See Create Connections to Oracle Autonomous Data Warehouse.



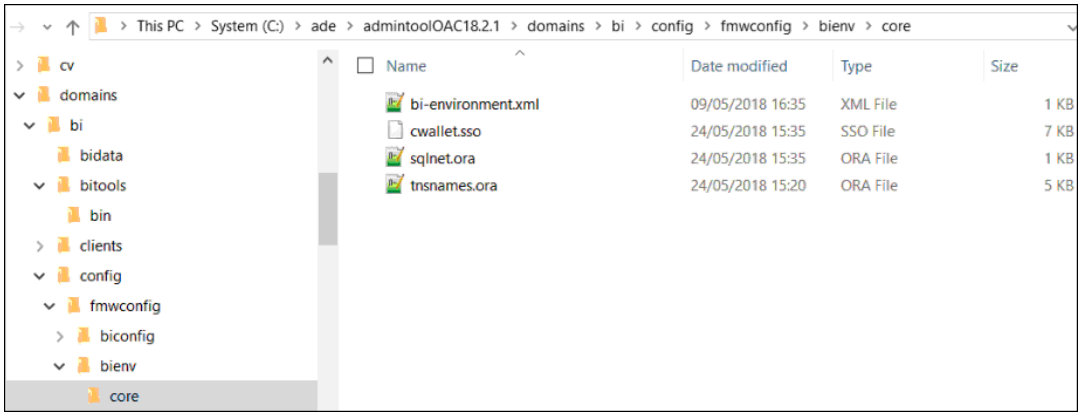
Now create a new workbook and dataset to visualize data from your Autonomous Data Warehouse.

Connect to Autonomous Data Warehouse in Model Administration Tool

You can use Model Administration Tool for Oracle Analytics Cloud to edit a semantic model connected to Autonomous Data Warehouse.

1. On the machine where you installed Oracle Analytics Cloud Client Tools, copy the `cwallet.sso`, `sqlnet.ora`, and `tnsnames.ora` from the zip file that you downloaded from Autonomous Data Warehouse to the folder:

   ```
   <Developer Client Tool installation
   folder>\domains\bi\config\fmwconfig\bienv\core
   ```

2. Edit `sqlnet.ora` so that the wallet location points to:

   ```
   <Developer Client Tool installation
   folder>\domains\bi\config\fmwconfig\bienv\core
   ```

   For example:

   ```
   WALLET_LOCATION = (SOURCE = (METHOD = file) (METHOD_DATA =
   (DIRECTORY="C:\ade\admintoolOAC18.2.1\domains\bi\config\fmwconfig\bienv\core")
   )) SSL_SERVER_DN_MATCH=yes
   ```

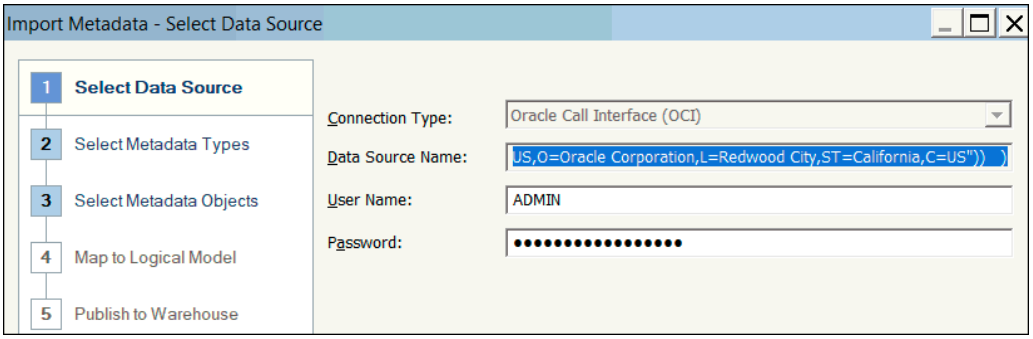
3. In Model Administration Tool, click **File**, then **Open**, then **In the Cloud** to open your semantic model. See Edit a Semantic Model in the Cloud.

   When you log in, use the connection information for your Oracle Analytics Cloud instance to fill in the Open in the Cloud dialog.

   - For **Port**, specify 443.
   - For **Host name**, specify the host domain name of your Oracle Analytics Cloud instance.
   - Select **SSL**. For **Trust Store** and **Password**, point to a local JDK/JRE cacerts keystore that trusts certificates signed by well-known CAs.

4. Connect to Autonomous Data Warehouse.

a. Click **File**, then **Import Metadata** to start the Import Metadata wizard, and follow the on-screen instructions.



b. On the Select Data Source page, for the **Data Source Name** value, specify a long TNS connection string from the downloaded `tnsnames.ora` file. Include the entire description, enclosed in brackets.

For example:

```
(description=(address=(protocol=tcps)(port=1522)
(host=adwc.example.oraclecloud.com))
(connect_data=(service_name=adwc1_high.adwc.oraclecloud.com))
(security=(ssl_server_cert_dn="CN=adwc.example.oraclecloud.com,OU=Oracle
BMCS US,O=Oracle Corporation,L=Redwood City,ST=California,C=US")) )
```

c. For **User Name** and **Password**, enter the credentials for the ADMIN user or another suitable Autonomous Data Warehouse user.

You're now ready to model the data in Model Administration Tool, publish the semantic model to Oracle Analytics Cloud, and create analyses and data visualizations using data from Autonomous Data Warehouse.

Connect to a Database Deployed on Oracle Cloud Infrastructure Classic with a Public IP Address

Configure Oracle Analytics Cloud to connect to Oracle Database Classic Cloud Service deployed on Oracle Cloud Infrastructure Classic so that end users can analyze that data in visualizations, analyses, and pixel-perfect reports.

Topics

Typical Workflow to Connect to a Database Deployed on Oracle Cloud Infrastructure Classic

If you're connecting Oracle Analytics Cloud to a database deployed on Oracle Cloud Infrastructure Classic for the first time, follow these tasks as a guide.

Task	Description	More Information
Verify the prerequisites	Verify that your environment satisfies the prerequisites required for this configuration.	Prerequisites
Record database information	Record connection information for Oracle Database Classic Cloud Service.	Record Database Information
Enable database access	Add access rules to enable Oracle Analytics Cloud access to the database.	Enable Database Access Through Port 1521
Connect to the database	Create and test your connections.	Connect to Your Database from Oracle Analytics Cloud

Prerequisites

Before you start, make sure you have the required environment.

Step	Description	Note Important Information
Set up Oracle Analytics Cloud	Deploy Oracle Analytics Cloud.	**Region** **Availability Domain**
Deploy Oracle Database Classic Cloud Service • Deploy Oracle Database Classic Cloud Service on the Virtual Cloud Network in Oracle Cloud Infrastructure Classic. • Populate Oracle Database Classic Cloud Service with data. • Set up a database user with permissions to read database tables.	Deploy Oracle Database Classic Cloud Service on the Virtual Cloud Network in Oracle Cloud Infrastructure Classic.	**Public IP** **Service Name** **Host Domain Name** **Database User/Password** Same: • **Region**

Record Database Information

All the information you need to connect to Oracle Database Classic Cloud Service is available in Oracle Cloud Infrastructure Console. Record the information now, so you have the required details when you set up the connection in Oracle Analytics Cloud.

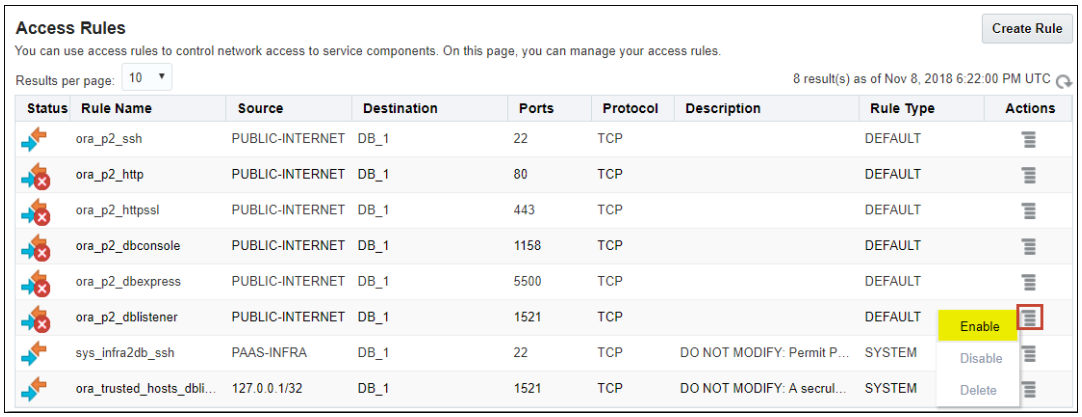
1. In Oracle Cloud Infrastructure Console, click  in the top left corner.
2. Click **OCI Classic Services**. Under **Classic Data Management Services**, click **Database Classic**.
3. Click the name of the database you want to connect to and from the Instance Overview section, record the Service Name from the **Connect String**. For example, `ucmdb906:1521/PDB1.504988564.oraclecloud.internal`.
4. Extract and record the Service Name of the database from the connect string value. For example, `PDB1.504988564.oraclecloud.internal`.

5. Record the IP address of the database displayed in the Resources section.
6. Find out the user name and password of a database user with permissions to read from this database, and write them down. For example, the user SYSTEM.

Enable Database Access Through Port 1521

Add an access rule that enables Oracle Analytics Cloud to access the database through port 1521.

1. In Oracle Cloud Infrastructure Console, click  in the top left corner.
2. Click **OCI Classic Services**. Under **Classic Data Management Services**, click **Database Classic**.
3. Select the database you want to connect to.
4. Click the **Manage service** icon and select **Access Rules**.
5. For port 1521, click **Actions** and select **Enable** to enable the port for the default Oracle listener.



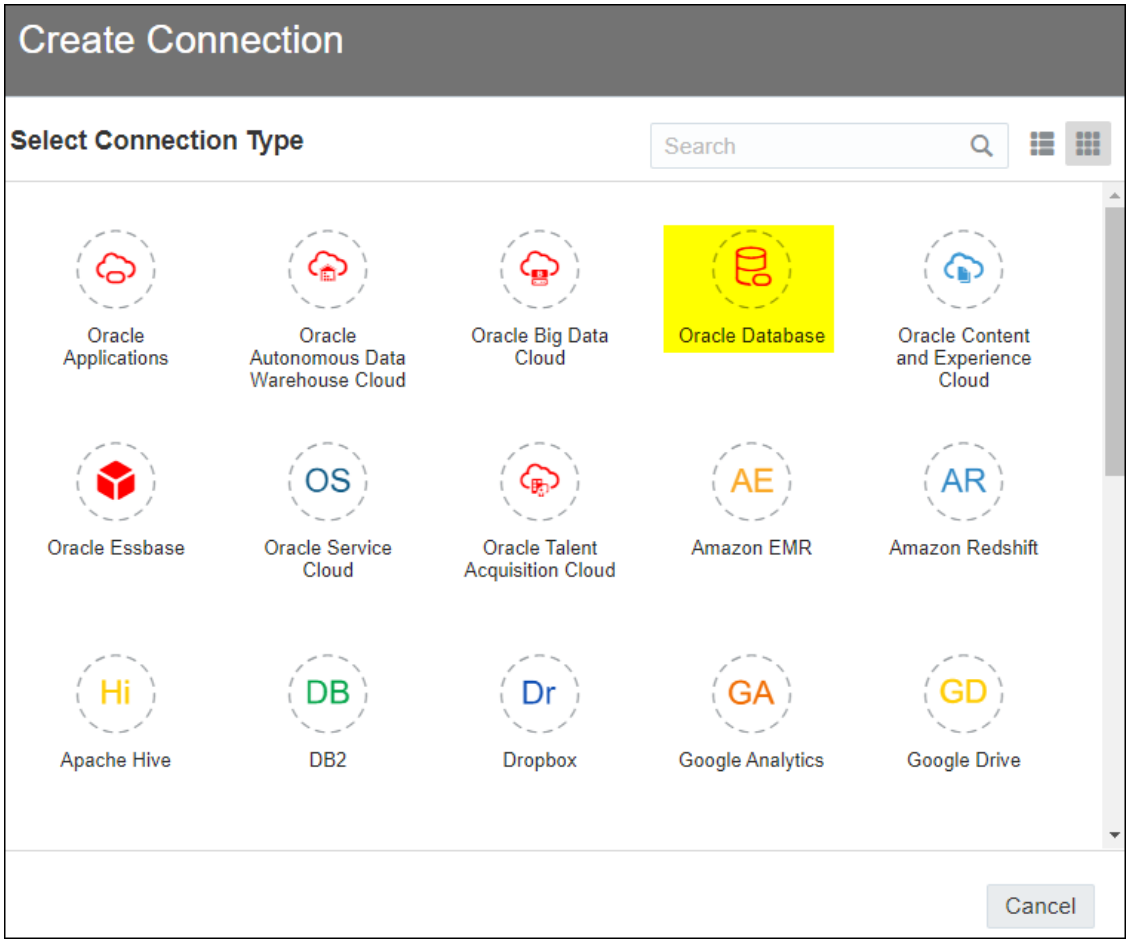
Connect to Your Database from Oracle Analytics Cloud

After enabling access to the database, use the database connection information you recorded earlier to connect Oracle Analytics Cloud to the database deployed in Oracle Cloud Infrastructure Classic. The way you connect to the database depends on what you want to do with the data.

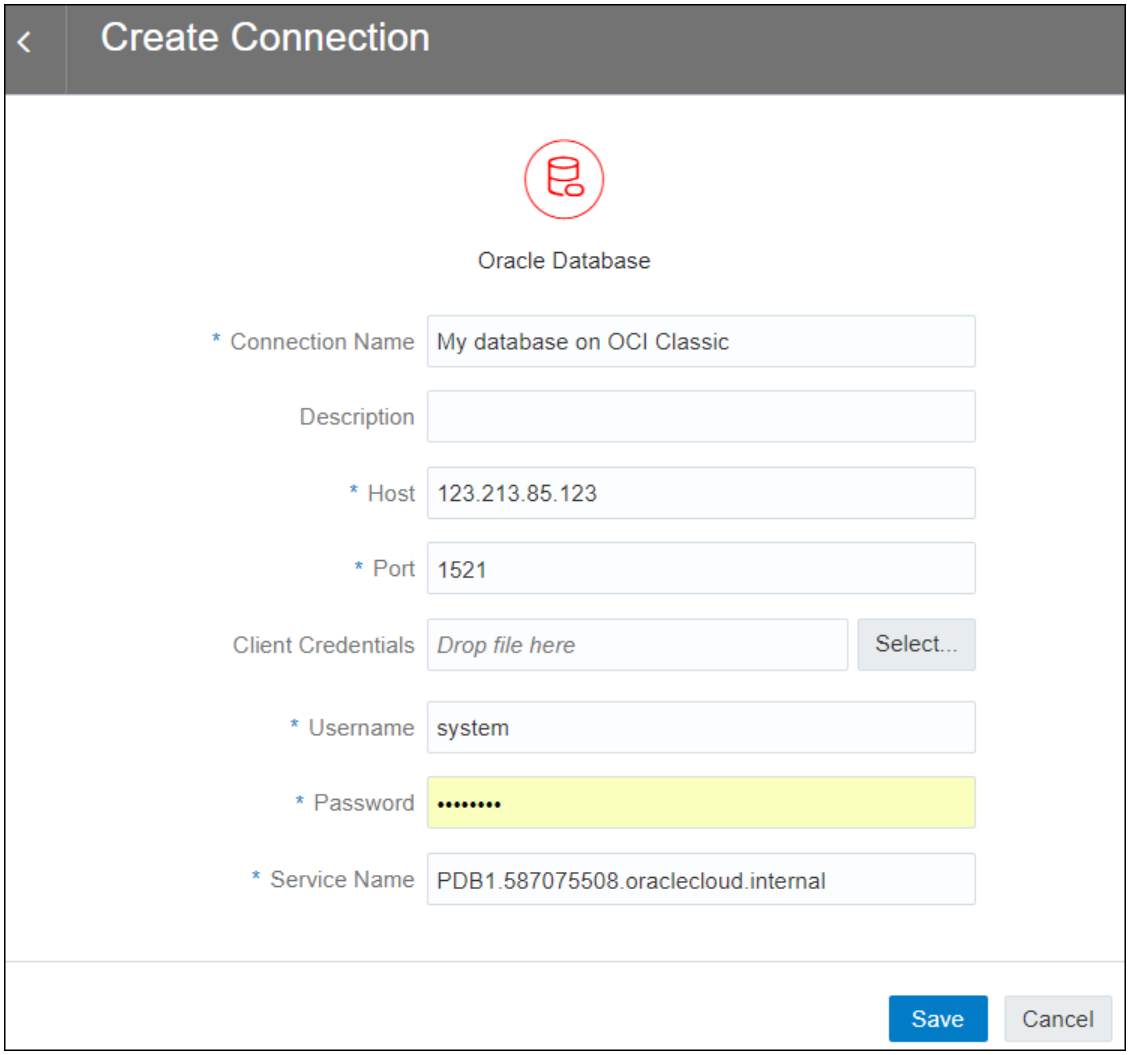
- Visualize the data.
- Model the data using Semantic Modeler or Data Modeler, then generate analyses and dashboards.
- Model the data with Oracle Analytics Model Administration Tool, then generate analyses and dashboards.

Connect to Your Database for Data Visualization or Semantic Modeler

In Oracle Analytics Cloud, create an Oracle Database connection for data visualizations in the usual way. See Create Database Connections.



Use the database details you recorded earlier to fill in the Create Connection dialog.



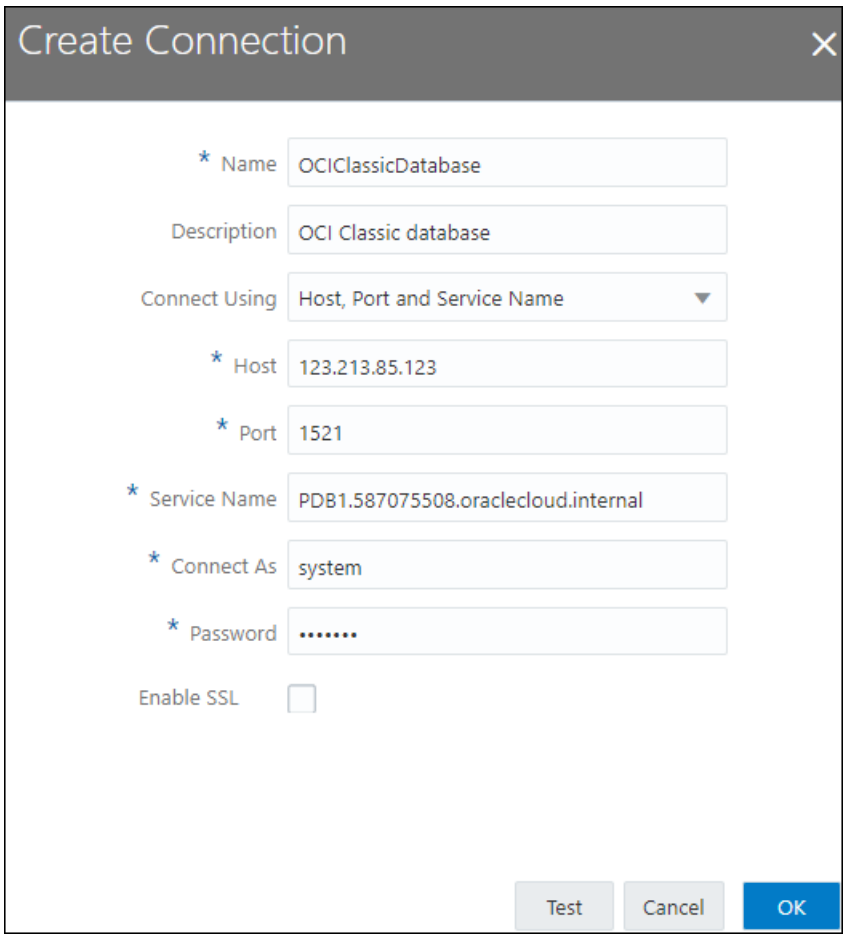
Specify these values:

- **Connection Name**: The name of the Oracle Database Classic Cloud Service you want to connect to.
- **Host**: The **Public IP** address for Oracle Database Classic Cloud Service. For example, 123.213.85.123.
- **Port**: The port number that enables access to Oracle Database Classic Cloud Service. For example, 1521.
- **Username**: The name of a user with read access to Oracle Database Classic Cloud Service.
- **Password**: The password for the specified database user.
- **Service Name**: The service name on the Database Classic page. For example, PDB1.123456789.oraclecloud.internal.

Connect to Your Database for Data Modeler

In Oracle Analytics Cloud Console, create a connection in the usual way. See Connect to Data in an Oracle Cloud Database.

Use the database details you recorded earlier to fill in the Create Connection dialog.



Specify these values:

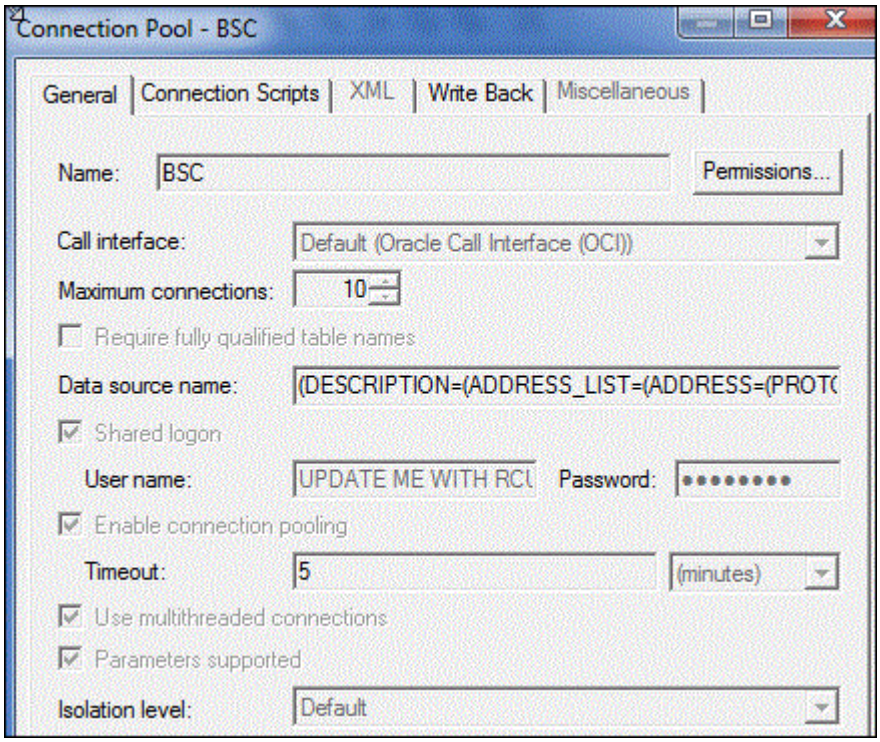
- **Name** and **Description**: The name of the Oracle Database Classic Cloud Service you want to connect to.
- **Connect Using**: Select **Host, Port, and Service Name**.
- **Host**: The **Public IP** address for Oracle Database Classic Cloud Service. For example, 123.213.85.123.
- **Port**: The port number that enables access to Oracle Database Classic Cloud Service. For example, 1521.
- **Service Name**: The service name from the Database Classic page. For example, PDB1.123456789.oraclecloud.internal.
- **Connect as**: The name of a user with read access to Oracle Database Classic Cloud Service.
- **Password**: The password for the specified database user.

Connect to Your Database in Oracle Analytics Model Administration Tool

In Model Administration tool for Oracle Analytics Cloud, click **File**, **Open**, and then **In the Cloud** to open your semantic model in the usual way. See Edit a Semantic Model in the Cloud.

When you sign in, use connection information for your Oracle Analytics Cloud to fill in the Open in the Cloud dialog.

Create a connection pool for your database. In the Physical pane, expand the database node, right-click the database icon, and click **Properties** to display the Connection Pool dialog. Use the database details you recorded earlier to specify **Call Interface**, **Data Source Name**, **User Name**, and **Password**.



Specify these values:

- **Call interface**: Select **Default (Oracle Call Interface (OCI)**.
- **Data Source Name**: Specify the connection details. For example:

  ```
  (DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=123.213.85.123)
  (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=PDB1.587075508.oraclecloud.internal))
  ```

  For SERVICE_NAME, use the Database Classic page to locate the service name. For example, `PDB1.587075508.oraclecloud.internal`.

You're now ready to model the data in Model Administration Tool, publish the semantic model to Oracle Analytics Cloud, and create analyses and data visualizations using data from Oracle Database Classic Cloud Service.

Part IV
Reference

Find answers to common questions, and troubleshoot connection issues.

Appendices:

- Data Sources and Data Types Reference
- Troubleshoot Private Access Channel Connectivity Issues

A

Data Sources and Data Types Reference

Find out about supported data sources, databases, JSON templates, and data types.

Topics

List of Supported Databases in Oracle Analytics Cloud

Oracle Analytics Cloud supports these databases. Follow the links for connectivity details about your data source.

- Oracle Database
- Oracle Analytic Views
- Oracle Applications
- Oracle Autonomous Data Warehouse (ADW)
- Oracle Autonomous Transaction Processing (ATP)
- OCI Data Flow SQL Endpoints
- OCI Object Storage
- OCI Resource (Preview)
- Oracle EPM Cloud (for Oracle Fusion Cloud Enterprise Performance Management)
- Oracle Essbase
- Oracle Hyperion Planning
- Oracle NetSuite
- Oracle Fusion Cloud B2C Service
- Oracle Talent Acquisition Cloud
- Amazon EMR
- Amazon Redshift
- Apache Hive
- CSV File
- Databricks
- Delta Share
- DropBox
- Google Analytics

- Google BigQuery
- Google Drive
- GreenPlum
- Hortonworks Hive
- IBM BigInsights Hive
- IBM DB2
- Impala (Cloudera)
- Informix
- JDBC (Generic)
- Local Subject Area in Oracle Analytics Cloud
- MapR Hive
- Microsoft Excel File
- Microsoft Azure SQL Database
- Microsoft Azure Synapse Analytics
- MongoDB
- MySQL
- MySQL HeatWave
- OData
- Pivotal HD Hive
- PostgreSQL
- REST API
- Salesforce
- Snowflake
- Spark
- SQL Server
- Sybase ASE
- Sybase IQ
- Teradata
- Vertica
- Key to Connectivity Table

Oracle Database

You can connect Oracle Analytics to an Oracle Database.

Supported Versions

12.1+, 12.2+, 18+, 19+

Prerequisites

Ensure that the appropriate security access rules are in place for Oracle Analytics Cloud to make a network connection to the database service on the database listening port.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Private access channel • Remote Data Connectivity • Data access - Live or cache	* For dataset connections, you can connect to multiple database instances. Upload a wallet for each connection.
Semantic Modeler	✓	• Standard** • Remote Data Connectivity • System connection	** For semantic model connections, you can only have one global wallet per semantic model connection.
Model Administration Tool	✓	• Standard • Private Access Channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✓	• Standard • Private access channel	-

For more information about this connectivity table, Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- Use the Oracle Database connection type to connect to Oracle Database Classic Cloud Service.

Useful Documentation Links

- Connect to an Oracle Database
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Analytic Views

You can connect Oracle Analytics to Oracle Analytic Views.

Supported Versions

Oracle Database 19c

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Data access - Live	-
Semantic Modeler	✗	-	-
Model Administration Tool	✓	• Standard	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

None.

Useful Documentation Links

- Connect to Oracle Analytic Views.
- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Applications

You can connect Oracle Analytics to Oracle Applications.

Supported Versions

Oracle Fusion Cloud Applications Suite, On-premises Oracle BI Enterprise Edition deployments, Another Oracle Analytics service

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Remote Data Connectivity • Data access - Cache only	* Remote connectivity for datasets is only available using Data Gateway for Linux.
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Connector supports several applications in Fusion Applications Suite.

Useful Documentation Links

- About the Oracle Applications Connector.
- Connect to an Application in Oracle Fusion Cloud Applications Suite
- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Autonomous Data Warehouse (ADW)

You can connect Oracle Analytics to Oracle Autonomous Data Warehouse (ADW).

Supported Versions

19c and later.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- For semantic model connections, you can only have one global wallet, therefore you can only connect to one instance.

Useful Documentation Links

- Connect to Oracle Autonomous Data Warehouse .
- You can also connect via Delta Sharing using the Delta Share connection type. See Connect to a Database Using Delta Sharing.
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Autonomous Transaction Processing (ATP)

You can connect Oracle Analytics to Oracle Autonomous Transaction Processing (ATP).

Supported Versions

19c and later.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	* For dataset connections, you can have one wallet per connection, therefore you can connect to multiple instances.
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard** • Private access channel • Remote Data Connectivity • System connection	** For semantic model connections, you can only have one global wallet per connection, therefore you can only connect to one instance.
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.

Useful Documentation Links

- Connect to Oracle Autonomous Transaction Processing
- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

OCI Data Flow SQL Endpoints

You can connect Oracle Analytics to OCI Data Flow SQL Endpoints.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

None.

Useful Documentation Links

- Connect to OCI Data Flow SQL Endpoints
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

OCI Object Storage

You can connect Oracle Analytics to OCI Object Storage and create datasets from files stored there.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Dataset from OCI Object Storage
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

OCI Resource (Preview)

You can connect Oracle Analytics to an OCI Resource database. Create a connection to an OCI Resource to integrate Oracle Analytics with OCI functions, OCI Vision, OCI Data Science, or OCI Language.

For example, you might register a language conversion function hosted in OCI so that you can convert English text into Spanish or German using an Oracle Analytics data flow.

Supported Versions

Not applicable.

Prerequisites

If applicable.

Connectivity

Use this connection type to register Oracle functions for use in data flows. See Create a Connection to Your OCI Tenancy.

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✗	-	-

Use Data Source With	Support	Connectivity Options	Notes
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle EPM Cloud (for Oracle Fusion Cloud Enterprise Performance Management)

You can connect Oracle Analytics to an Oracle EPM Cloud database.

Supported Versions

Latest version.

Prerequisites

Before you start, make sure that your product is supported. See Which Oracle EPM Business Processes Does Oracle Analytics Support?.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✓	• Standard	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- You can't use Oracle Fusion Cloud Enterprise Performance Management (EPM) datasets in Data Flows.
- You can't blend datasets that use Oracle Fusion Cloud Enterprise Performance Management (EPM) data sources.

Useful Documentation Links

- Connect to Oracle Fusion Cloud Enterprise Performance Management (EPM).
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Essbase

You can connect Oracle Analytics to Oracle Essbase.

Supported Versions

11.1.2.4.0+, 21c

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- For direct connections, see Create a Connection to Oracle Essbase.
- For remote connections via Data Gateway, see Create a Connection to Oracle Essbase Data on a Private Network .
- For remote connections via Private Access Channel, see Connect to On-premises Data Sources Over a Private Access Channel.
- You can't use Oracle Essbase datasets in Data Flows.
- You can't blend datasets that use Oracle Essbase data sources.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Hyperion Planning

You can connect Oracle Analytics to Oracle Hyperion Planning to model your data.

Supported Versions

11.1.2.4+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets		-	-
Semantic Modeler		-	-
Model Administration Tool		• Standard • Data access - Live only	-
Oracle Analytics Publisher		-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to an Oracle Database
- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle NetSuite

You can connect Oracle Analytics to Oracle NetSuite.

Supported Versions

Release 2019.2 (JDBC Driver 8.10.85.0)

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Specify NetSuite2.com as Data Source.

Useful Documentation Links

- Connect to NetSuite
- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Fusion Cloud B2C Service

You can connect Oracle Analytics to Oracle Fusion Cloud B2C Service.

Supported Versions

1.2

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Oracle Talent Acquisition Cloud

You can connect Oracle Analytics to Oracle Talent Acquisition Cloud.

Supported Versions

15b.9.3+, 17.4+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Oracle Talent Acquisition Cloud
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Amazon EMR

You can connect Oracle Analytics to an Amazon EMR database.

Supported Versions

4.7.2 (Running Amazon Hadoop 2.7.2 and Hive 1.0.0)

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Complex data types not supported.
- Amazon EMR (MapR) No Amazon Machine Image (AMI) 3.3.2 running MapR Hadoop M3 and Hive 0.13.1.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Amazon Redshift

You can connect Oracle Analytics to an Amazon Redshift database.

Supported Versions

1.0.1036 +

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Remote data connectivity • Data access - Live or cache	-

Use Data Source With	Support	Connectivity Options	Notes
Semantic Modeler	✓	• Standard • Remote data connectivity • System connection	-
Model Administration Tool	✓	• Remote data connectivity • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- JDBC and JNDI Templates and Examples
- Manage Connections Using REST APIs

Apache Hive

You can connect Oracle Analytics to an Apache Hive database.

Supported Versions

2.3.0+, 3.0+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

CSV File

You can connect Oracle Analytics to data in a comma-separated values (CSV) file.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create Datasets from Files

Databricks

You can connect Oracle Analytics to a Databricks database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Use the **Delta Share** connection type.

Useful Documentation Links

- Connect to a Database Using Delta Sharing
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Delta Share

Use the Delta Sharing protocol to connect to Oracle Autonomous Data Warehouse and Databricks.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	Not applicable	-	-
Semantic Modeler	Not applicable	-	-

Use Data Source With	Support	Connectivity Options	Notes
Model Administration Tool	Not applicable	-	-
Oracle Analytics Publisher	Not applicable	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Use the **Delta Share** connection type.
- See Oracle Autonomous Data Warehouse and Databricks.

Useful Documentation Links

- Connect to a Database Using Delta Sharing
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

DropBox

You can connect Oracle Analytics to a DropBox database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Dropbox
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Google Analytics

You can connect Oracle Analytics to a Google Analytics database.

Supported Versions

Universal Analytics, Google Analytics V4

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Google Drive or Google Analytics
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Google BigQuery

You can connect Oracle Analytics to a Google BigQuery database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Data access - Live or cache	-
Semantic Modeler	✓	• System connection	-
Model Administration Tool	✓	• System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Google BigQuery
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Google Drive

You can connect Oracle Analytics to a Google Drive database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Google Drive or Google Analytics
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

GreenPlum

You can connect Oracle Analytics to a GreenPlum database.

Supported Versions

4.3.8+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Hortonworks Hive

You can connect Oracle Analytics to a Hortonworks Hive database.

Supported Versions

1.2+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard	-
Model Administration Tool	✓	• Private access channel • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

IBM BigInsights Hive

You can connect Oracle Analytics to a IBM BigInsights Hive database.

Supported Versions

1.2+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Remote Data Connectivity • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-

Use Data Source With	Support	Connectivity Options	Notes
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

IBM DB2

You can connect Oracle Analytics to an IBM DB2 database.

Supported Versions

11.5+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✓	• Standard	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports SSL between Data Gateway and Oracle Analytics Cloud.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Impala (Cloudera)

You can connect Oracle Analytics to an Impala (Cloudera) database.

Supported Versions

2.7+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets		• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler		• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool		• Standard • Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher		-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Informix

You can connect Oracle Analytics to an Informix database.

Supported Versions

12.10+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

JDBC (Generic)

You can connect Oracle Analytics to a JDBC (Generic) database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Remote Data Using Generic JDBC
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Local Subject Area in Oracle Analytics Cloud

You can connect Oracle Analytics to data in a Local Subject Area in Oracle Analytics Cloud.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Dataset from a Local Subject Area

MapR Hive

You can connect Oracle Analytics to a MapR Hive database.

Supported Versions

1.2+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard	-
Model Administration Tool	✓	• Private access channel • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Microsoft Excel File

You can connect Oracle Analytics to data in a Microsoft Excel file.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	Only XLSX files (or XLS with unpivoted data).
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create Datasets from Files

Microsoft Azure SQL Database

You can connect Oracle Analytics to Microsoft Azure SQL Database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	Use the **SQL Server** connection type on the Create Connection page.
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Microsoft Azure Synapse Analytics

You can connect Oracle Analytics to a Microsoft Azure Synapse Analytics database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

MongoDB

You can connect Oracle Analytics to a MongoDB database.

Supported Versions

3.2.5

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Private access channel • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

MySQL

You can connect Oracle Analytics to a MySQL database.

Supported Versions

5.6+, 5.7+, 8.0+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	Supports Enterprise Edition Only.

Use Data Source With	Support	Connectivity Options	Notes
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Private access channel • Remote Data Connectivity • System connection	Supports all Editions.
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

MySQL HeatWave

You can connect Oracle Analytics to a MySQL HeatWave database.

Supported Versions

8.0.31+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Current latest cloud version supported.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

OData

You can connect Oracle Analytics to an OData database.

Supported Versions

2.0

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Cache only	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Pivotal HD Hive

You can connect Oracle Analytics to a Pivotal HD Hive database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard	-
Model Administration Tool	✓	• Standard	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

PostgreSQL

You can connect Oracle Analytics to a PostgreSQL database.

Supported Versions

9.0+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Private access channel • Data access - Live or cache	-
Semantic Modeler	✓	• System connection	-
Model Administration Tool	✓	• Private access channel • System connection	-

Use Data Source With	Support	Connectivity Options	Notes
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

REST API

You can connect Oracle Analytics to a REST API database.

Supported Versions

Not applicable.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel	Connect to wide range of data sources that have REST endpoints available
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source with REST Endpoints.
- JSON Examples for Common Data Sources with REST Endpoints.
- Manage Connections to Data Sources

- Manage Connections Using REST APIs

Salesforce

You can connect Oracle Analytics to a Salesforce database.

Supported Versions

Not applicable.

Prerequisites

Before you create a Salesforce connection, in the Salesforce application make sure that you enable API access in the administrative permissions for the Salesforce user.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Data access - Live or cache	-
Semantic Modeler	✗	-	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Snowflake

You can connect Oracle Analytics to a Snowflake database.

Supported Versions

Latest version.

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard • Private access channel • System connection	-
Oracle Analytics Publisher	✓	• Standard	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Connect to Snowflake Data Warehouse.
- Model Data in Snowflake Data Warehouse
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Spark

You can connect Oracle Analytics to a Spark database.

Supported Versions

1.6+, 3.0

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Private access channel • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-

Use Data Source With	Support	Connectivity Options	Notes
Model Administration Tool	✓	• Standard • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Supports saving output from data flows.
- Supports Kerberos authentication for datasets.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

SQL Server

You can connect Oracle Analytics to a SQL Server database.

Supported Versions

2014, 2016, 2017, 2019

Prerequisites

In SQL Server, set up a named connection using static port allocation. Dynamic port allocation isn't supported for connecting Oracle Analytics to SQL Server.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✓	• Standard	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Sybase ASE

You can connect Oracle Analytics to a Sybase ASE database.

Supported Versions

15.7+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard* • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity • System connection	-
Model Administration Tool	✗	-	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Sybase IQ

You can connect Oracle Analytics to a Sybase IQ database.

Supported Versions

16+

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets		• Standard • Data access - Live or cache	-
Semantic Modeler		-	-
Model Administration Tool		-	-
Oracle Analytics Publisher		-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Teradata

You can connect Oracle Analytics to a Teradata database.

Supported Versions

16.20, 17.x

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard • Remote Data Connectivity	-
Model Administration Tool	✓	• Standard • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✗	-	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- None.

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Vertica

You can connect Oracle Analytics to a Vertica database.

Supported Versions

9.x, 12.x

Prerequisites

None.

Connectivity

Use Data Source With	Support	Connectivity Options	Notes
Datasets	✓	• Standard • Private access channel • Remote Data Connectivity • Data access - Live or cache	-
Semantic Modeler	✓	• Standard	-

Use Data Source With	Support	Connectivity Options	Notes
Model Administration Tool	✓	• Standard • Private access channel • Remote Data Connectivity • System connection	-
Oracle Analytics Publisher	✓	• Standard	-

For more information about this connectivity table, see Key to Connectivity Table.

Other Connectivity Information

- Server-Side SSL support only – no support for Mutual-TLS.
- Remote connectivity for datasets is only available using Data Gateway.
- If you're connecting to an on-premises Vertica database using Data Gateway, you must copy the Vertica JDBC client driver JAR file to the machine where Data Gateway is installed:

  1. Stop the Jetty server. For example, use

     ```
     ./stopJetty.sh
     ```

     (on Linux) or

     ```
     stopJetty.cmd
     ```

     (on Windows). Run this command from

     ```
     Oracle/Middleware/Oracle_Home/domain/bin
     ```

     .
  2. Copy the Vertica JAR file to

     ```
     Oracle/Middleware/Oracle_Home/domain/jettybase/lib/ext
     ```

     .
  3. Start the Jetty server. For example, use .

     ```
     /startJetty.sh
     ```

     .

Useful Documentation Links

- Create a Connection to a Data Source
- Manage Connections to Data Sources
- Manage Connections Using REST APIs

Key to Connectivity Table

Use this guidance on connectivity options to connect Oracle Analytics to your data.

Key

- **Version Numbers**:
  - "1.x" means any version starting with a 1 (for example, this includes version 1.4.3 but not version 2.0).
  - "2.0.x" means any version starting with 2.0 (for example, this includes version 2.0.4 but not version 2.4).
  - "1.6+" means any version starting with 1 and is greater than or equal to (>=) 1.6 (for example, this includes version 1.8 but not version 2.4).
- A "Yes" () in the **Support** column means that you can connect to this type of data source using one or more of the options listed in **Connectivity Options**.
- **Connectivity Options**:
  - **Standard** means that the data source host is accessible via the public Internet.
  - **Private access channel** means that Oracle Analytics Cloud can access data on a private host over a private access channel. You can use a private access channel to connect to private data sources that are within your virtual cloud network (VCN) on Oracle Cloud Infrastructure or other networks peered to the VCN such as your corporate network. See Connect to Private Data Sources Through a Private Access Channel.
  - **Remote Data Connectivity:**
    * For datasets, this means that if your administrator has set up and enabled remote data connectivity, you can visualize on-premises data. You'll see a check box named **Use Remote Data Connectivity** on the Create Connection dialog that you select to indicate that the database is on-premises. See Configure Data Gateway for Data Visualization.
    * For Semantic Modeler or Model Administration Tool, this means that if your administrator has set up and enabled remote data connectivity, you can model on-premises data of that type. See Configure and Register Data Gateway for Reporting.

    Notes:
  - In data flows, you can add data from remote databases using a remote connection with Data Gateway. However, you can't save data to datasets that use remote connections.
  - **Data Access** options:
    Live only means that in a dataset, the table can only get its data directly from the data source.

    Cache only means that in a dataset, the table can only load or reload its data into cache.

    Live or Cache means that in a dataset the table can access its data in either live mode or cache mode.

    See Specify Whether a Dataset Table Is Cached or Live.

  - **System Connection** means that data modelers can connect to a semantic model using connection details copied from an Oracle Analytics Cloud connection. For supported data sources, data modelers copy the **Object ID** from the Inspect pane into Semantic Modeler. If you're using Model Administration Tool, copy the Object ID into the Connection Pool dialog. See Connect to a Data Source Using a Data Connection.
- When Oracle Analytics is deployed as part of other services such as Fusion Analytics Warehouse or NetSuite Analytics Warehouse, you can't connect to the semantic model. Therefore you can ignore the **Semantic Modeler** options.
- Oracle Analytics Cloud supports Transport Layer Security (TLS) for all data sources.
- In addition to the connections types listed on the Connections page, you can connect remotely to other on-premises data sources using generic JDBC. See Connect to Remote Data Using Generic JDBC.

Certification - Supported Data Types

Here're the supported data types for Oracle Analytics.

Topics:

Supported Base Data Types

When reading from a data source, Oracle Analytics attempts to map incoming data types to the supported data types.

For example, a database column that contains only date values is formatted as a DATE, a spreadsheet column that contains a mix of numerical and string values is formatted as a VARCHAR, and a data column that contains numerical data with fractional values uses DOUBLE or FLOAT.

In some cases Oracle Analytics can’t convert a source data type. To work around this data type issue, you can manually convert a data column to a supported type by entering SQL commands. In other cases, Oracle Analytics can't represent binary and complex data types such as BLOB, JSON, and XML.

Note that some data types aren't supported. You'll see an error message if the data source contains unsupported data types.

Oracle Analytics supports the following base data types:

- **Number Types** — SMALLINT, SMALLUNIT, TINYINT, TINYUINT, UINT, BIT, FLOAT, INT, NUMERIC, DOUBLE
- **Date Types** — DATE, DATETIME, TIMESTAMP, TIME
- **String Types** — LONGVARCHAR, CHAR, VARCHAR

Supported Data Types by Database

Oracle Analytics supports the following data types.

Database Type	Supported Data Types
Oracle	BINARY DOUBLE, BINARY FLOAT CHAR, NCHAR CLOB, NCLOB DATE FLOAT NUMBER, NUMBER (p,s), NVARCHAR2, VARCHAR2 ROWID TIMESTAMP, TIMESTAMP WITH LOCAL TIMEZONE, TIMESTAMP WITH TIMEZONE
DB2	BIGINT CHAR, CLOB DATE, DECFLOAT, DECIMAL, DOUBLE FLOAT INTEGER LONGVAR NUMERIC REAL SMALLINT TIME, TIMESTAMP VARCHAR
SQL Server	BIGINT, BIT CHAR DATE, DATETIME, DATETIME2, DATETIMEOFFSET, DECIMAL FLOAT INT MONEY NCHAR, NTEXT, NUMERIC, NVARCHAR, NVARCHAR(MAX) REAL SMALLDATETIME, SMALLINT, SMALLMONEY TEXT, TIME, TINYINT VARCHAR, VARCHAR(MAX) XML

Database Type	Supported Data Types
MySQL	BIGINT, BIGINT UNSIGNED CHAR DATE, DATETIME, DECIMAL, DECIMAL UNSIGNED, DOUBLE, DOUBLE UNSIGNED FLOAT, FLOAT UNSIGNED INTEGER, INTEGER UNSIGNED LONGTEXT MEDIUMINT, MEDIUMINT UNSIGNED, MEDIUMTEXT SMALLINT, SMALLINT UNSIGNED TEXT, TIME, TIMESTAMP, TINYINT, TINYINT UNSIGNED, TINYTEXT VARCHAR YEAR
Apache Spark	BIGINT, BOOLEAN DATE, DECIMAL, DOUBLE FLOAT INT SMALLINT, STRING TIMESTAMP, TINYINT VARCHAR
Teradata	BIGINT, BYTE, BYTEINT CHAR, CLOB DATE, DECIMAL, DOUBLE FLOAT INTEGER NUMERIC REAL SMALLINT TIME, TIMESTAMP VARCHAR

JSON Examples for Common Data Sources with REST Endpoints

Download these example JSON files for data sources listed below from the Oracle Analytics Public Library to enable you to connect to data sources with REST endpoints.

See Oracle Analytics Public Library.

- Aha
- Alpha Vantage
- IBM NLP
- Lexigram
- Mailchimp
- NY Times

- Oracle Analytics Publisher
- Quandl
- ServiceNow
- Stripe
- SurveyMonkey
- USDA-Nutrient
- US Geological Survey
- Yelp
- Zendesk

About the Oracle Applications Connector

The "Oracle Applications" connection type () enables you to use Oracle Analytics to visualize data from applications in Oracle Fusion Cloud Applications Suite. For example, Oracle Fusion Cloud Financials.You can also use the "Oracle Applications" connection type to connect to your on-premises Oracle BI Enterprise Edition deployments (if patched to an appropriate level) or connect to another Oracle Analytics service.

You can connect to these applications in Fusion Applications Suite:

- Oracle Fusion Cloud Financials
- Oracle Fusion Cloud Human Capital Management
- Oracle Fusion Cloud Loyalty
- Oracle Fusion Cloud Procurement
- Oracle Fusion Cloud Project
- Oracle Fusion Cloud Supply Chain Planning
- Oracle Sales Automation

> **Note:**
>
> When you connect to applications in Fusion Applications Suite, you access the data from an Oracle Transactional Business Intelligence report. These reports are subject to caching in Oracle Transactional Business Intelligence, and the data available in Oracle Analytics is based on the cached data. You can't control the cache behavior in Oracle Transactional Business Intelligence from Oracle Analytics.

B

Frequently Asked Questions

This reference provides answers to common questions asked by administrators and business intelligence analysts connecting to Oracle Analytics Cloud.

Topics

Frequently Asked Questions about Data Gateway

Here are answers to common questions about Data Gateway.

What operating systems does Data Gateway support?

You can deploy Data Gateway on Linux and Windows platforms. For a full list of supported operating systems, see Download page for Oracle Analytics Cloud.

What is the Data Gateway architecture?

See Overview to Connecting to On-premises Data Sources.

Where do I install Data Gateway?

You install Data Gateway in a subnet that gives visibility to both Oracle Analytics Cloud and the target data sources. Your network needs to allow outbound traffic from the node where Data Gateway is installed to the public internet on port 443 so that Data Gateway can communicate with Oracle Analytics Cloud. In addition, the network needs to allow outbound traffic from the Data Gateway agent to the data source. For example, you might test the network by opening a browser on the node where Data Gateway is installed and connect to Oracle Analytics Cloud. You might also test the connection from the same node to the data source using a generic JDBC tool.

Can I deploy multiple Data Gateway agents?

Yes. You can configure multiple Data Gateway agents to service the same Oracle Analytics Cloud service instance. However, all of these agents must be capable of servicing all remote queries. In other words, you cannot configure one agent to service queries for one data source only, and another agent to service queries for a different data source. In addition, in server deployments you can have multiple Data Gateway agents on each node (physical or virtual). For high availability, Oracle recommends at least two Data Gateway agents (that is, on two virual machines) per Oracle Analytics Cloud instance.

How do I configure high availability for Data Gateway?

On the Oracle Analytics Cloud side, high availability is provided natively. On the Data Gateway side, you set up high availability by deploying two Data Gateways for each Oracle Analytics Cloud instance.

Why is Data Gateway traffic egress only?

Data Gateway regularly communicates with Oracle Analytics Cloud to see whether Oracle Analytics Cloud has queries that need processing, known as long-polling. Data Gateway makes a long-running Transport Layer Security-encrypted HTTP request to Oracle Analytics Cloud and waits until Oracle Analytics Cloud has a query to process. If there're no queries from Oracle Analytics Cloud after two minutes, Data Gateway terminates and re-issues the request in order to avoid termination by the network as an idle or stale connection.

How does Data Gateway manage SSL certificates?

The HTTPS communication between Data Gateway and Oracle Analytics Cloud leverages the SSL certificate of your Oracle Analytics Cloud service instance. The same certificate encrypts your browser connections to Oracle Analytics Cloud.

How do I size Data Gateway?

Ask your sales account team for guidance on sizing Data Gateway.

Where is Data Gateway running? Do I install it on a virtual machine (VM)?

- At the Oracle Analytics Cloud end, Oracle Analytics Cloud manages the Data Gateway queue, therefore there's nothing additional to install.
- At the data source end, the Data Gateway agent typically runs on a server or Virtual Machine next to the data source. You can also run Data Gateway from a laptop or a compute instance in the cloud, as long as Data Gateway can connect to the data source.

How is Data Gateway network traffic secured?

When you install and set up Data Gateway, you generate a public key. This public key is used in conjunction with the private key for Oracle Analytics Cloud to encrypt all communication between Oracle Analytics Cloud and Data Gateway. Data Gateway's security features prevent "replay attacks" and "man-in-the-middle" attacks. The Transport Layer Security 1.2 encryption deployed by the HTTPS connection provides a further layer of encryption.

Can Data Gateway limit queries that affect performance or security?

Data Gateway doesn't limit the query row size. The query row size limit is determined by the number of Oracle Compute Units (OCPUs) that your Oracle Analytics Cloud service has.

What is the timeout setting for Data Gateway?

Data Gateway uses the same query timeout as Oracle Analytics Cloud. See Limits Querying Data (Data Visualization Workbooks, Classic Analyses and Dashboards).

C

Troubleshoot

This topic describes common connection issues and explains how to solve them.

Topics:

Troubleshoot Private Access Channel Connectivity Issues

This topic describes common problems that you might encounter and explains how to solve them.

Troubleshoot connectivity to an on-premises Oracle Database

Complete the following configuration in your on-premises for single node Oracle Database environment:

1. On the firewall, open the Oracle Database port e.g. 1521.
2. Set up a direct connection between your on-premises network and Oracle Cloud Infrastructure VCN.
3. Create a private DNS view, and then add a zone (in the view) for your custom domain. For example, ocivcn.companyabc.com.

Create a temporary Compute instance in the PAC subnet and then verify that you can resolve the on-premises database hostname and port, and ping the private IP address.

Hostname resolution check command:

```
$ nslookup <On-premises database hostname>
```

If you can't resolve the on-premises single node Oracle database hostname, it means the DNS servers configured in the subnet DHCP option can't resolve the hostname or the DNS zone configuration is invalid.

Connection check command:

```
nc -zv <On-premises database hostname> <port>
```

For example: `nc -zv onprem.db.xyz.com 1521`.

Note: If the nc package isn't available, use `yum install nc*`.

If you can't establish a connection, check VPN or FastConnect network connectivity between Oracle Cloud Infrastructure VCN and on-premises network.

Troubleshoot connectivity to an on-premises Oracle Essbase data source

Complete the following configuration in your on-premises Essbase environment:

1. On the firewall, open Essbase port ranges *32768-33768* and *1423*.
   Check your *essbase.cfg* file for any valid ports that Essbase is currently using.

Note: If you are using Palo Alto Networks firewall, don't create a rule on *App-ID* that is, oracle-essbase. Instead, create a firewall rule that includes the Essbase port ranges.

2. Set up a direct connection between your on-premises network and Oracle Cloud Infrastructure VCN.
3. Create a private DNS view, and then add a zone (in the view) for your custom domain. For example, `ocivcn.companyabc.com`.

Hostname resolution check command:

```
$ nslookup <On-premises Essbase hostname>
```

If you can't resolve the on-premises Essbase hostname, it means the DNS servers configured in the subnet DHCP option isn't able to resolve the hostname or the DNS zone configuration is invalid.

Connection check command:

```
nc -zv <On-premises Essbase hostname> <essbase port>
```

For example:

```
nc -zv onprem.essbase.xyz.com 1423

nc -zv onprem.essbase.xyz.com 33767
```

Note: If the nc package isn't available, use `yum install nc*`.

If you can't establish socket connection, check the following:

- VPN or FastConnect network connectivity between your Oracle Cloud Infrastructure VCN and on-premises network.

- Firewall rule exists for the entire Essbase port range 32768-33768.

Troubleshoot the maximum amount of query execution time in Planning and Budgeting Cloud Service

Oracle's Planning and Budgeting Cloud Service is one such cloud-based technology that provides businesses with an integrated solution for budgeting, forecasting, and planning. To ensure stability, it is crucial to set query execution time (QRYGOVEXECTIME) in Planning and Budgeting Cloud Service (Essbase). In this section, we will discuss the importance of setting QRYGOVEXECTIME.

What is QRYGOVEXECTIME?

`QRYGOVEXECTIME` is a parameter that controls the maximum amount of time that a query can run in Essbase.

Why is QRYGOVEXECTIME important?

`QRYGOVEXECTIME` is essential in Planning and Budgeting Cloud Service because it helps ensure the stability of both Oracle Analytics Cloud and Planning and Budgeting Cloud Service. Here are a few reasons why:

- Prevents long-running queries: Long-running queries can cause system instability, leading to performance issues and even system crashes. By setting QRYGOVEXECTIME (in PBCS), businesses can prevent long-running queries from executing, which can help ensure system stability.
- Limits resource consumption: Queries that run for an extended period can consume significant system resources, leading to performance degradation. By setting

QRYGOVEXECTIME, businesses can limit resource consumption by preventing queries from running indefinitely.

- Improves user experience: When users run a query that takes a long time to complete, it can lead to frustration and dissatisfaction. By limiting the maximum query execution time, businesses can improve the user experience by ensuring that queries run in a timely manner.

In conclusion, setting QRYGOVEXECTIME in PBCS / Essbase is an important step in safeguarding the stability of both Oracle Analytics Cloud and Essbase. By limiting query execution times, you can prevent resource contention, improve system stability, and improve overall performance. So, take the time to adjust this parameter to an appropriate value for your environment.

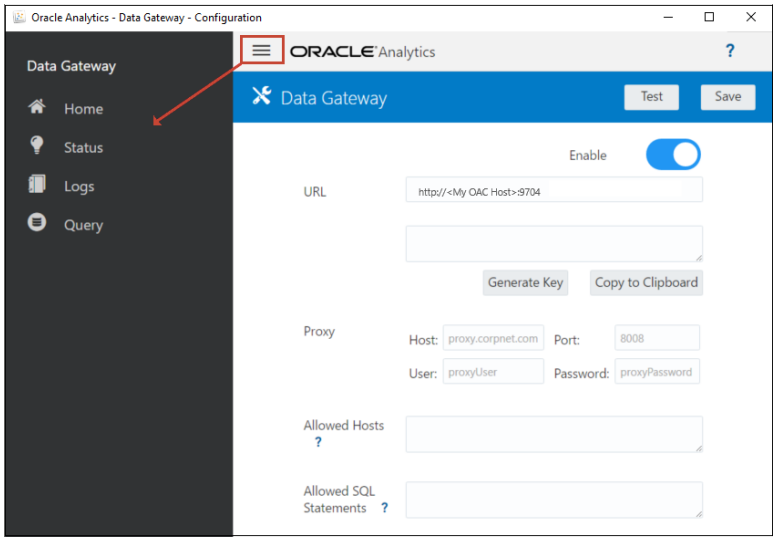
To implement this query execution time limits raise an SR with Oracle Support for Oracle Planning and Budgeting Cloud.

Troubleshoot Data Gateway

In a Data Gateway agent, use the Navigator options to display the Status, Logs, and Query pages to monitor remote connection traffice, and troubleshoot common connectivity and performance issues.

Topics

Click Navigator to access the Data Gateway pages.



Diagnosing Connection Issues Using The Status Page

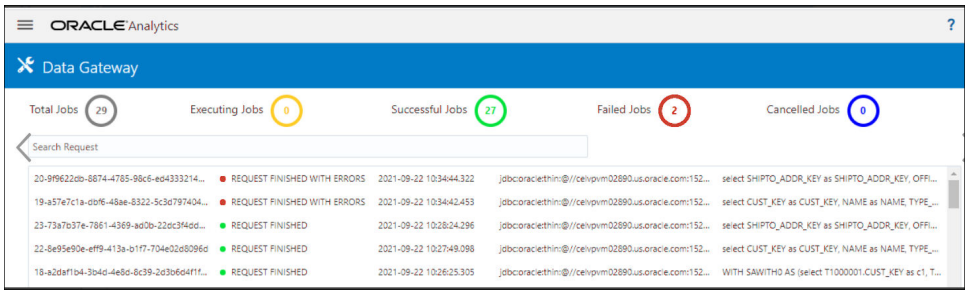
Use this page to review database requests that a Data Gateway agent makes with the remote database.

In a Data Gateway agent, click **Navigator** then **Status** to review database requests.

To diagnose issues, you typically search on date or job status:

- To search on date, in the **Search Request** field, enter all or part of the date and time in the format "YYYY-MM-DD HH-MM-SS". For example, enter "2022-03-28" to look for entries for 28th March, 2022.
- To search for failed jobs, in the **Search Request** field, enter "REQUEST FINISHED WITH ERRORS".

Clear the **Search Request** field to review all jobs.

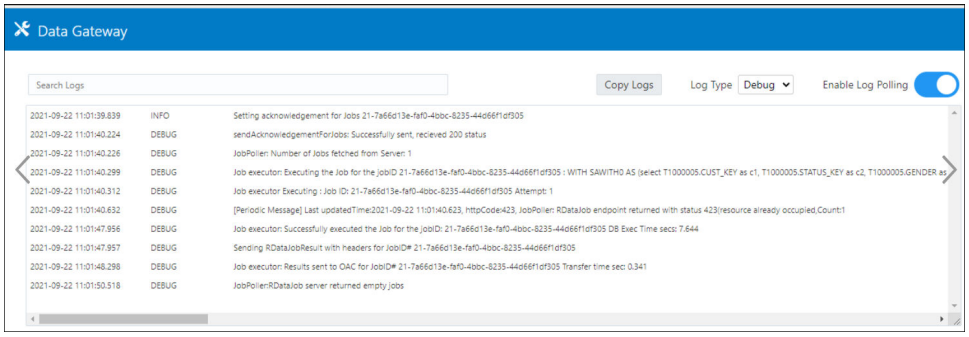


Click on a job to view the detailed status information.

Diagnosing Connection Issues Using The Logs Page

Use this page to review a Data Gateway agent's log entries so that you can analyze connection traffic.

In a Data Gateway agent, click **Navigator** then **Logs** to review log entries. Enable the **Enable Log Polling** option and select an appropriate logging level. For example, to diagnose connection issues, you might set **Log Type** to **Debug**.

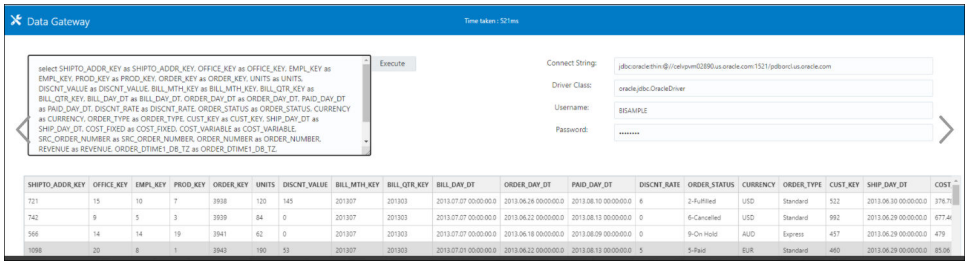


Oracle recommends that when you've completed your troubleshooting, you either disable **Enable Log Polling** or adjust the **Log Type** to capture less information.

Diagnosing Connection Issues Using The Query Page

Use this page to query a remote database from a Data Gateway agent to test the connection and assess performance.

In a Data Gateway agent, click **Navigator** then **Query** to execute a SQL statement directly from the Data Gateway agent to the (on-premises) database. For example, you might copy the **Query String**, **Connection String**, and **Driver Class** from a failed job listed on the Status page. Supply the database credentials and execute the query to review the outcome and the performance statistics (Time taken). **Note**: The remote database must support connection using a JDBC connection string.



Remote Connectivity Issues and Tips

Here're some connectivity issues that you might encounter and tips on how to resolve them.

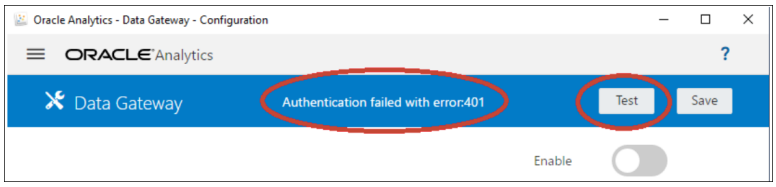
Dashboard and Analyses Issues

Issue reported	Do this
Error Code 603 - no agents are connected	Check that the Data Gateway agent is running and enabled in the Data Gateway Configuration page. With Data Gateway on Linux: Run $ <Data Gateway Install Directory>/domain/bin/status.sh and see if the "Data Gateway Status" is UP or DOWN. With Data Gateway on Windows: Check the Task Manager > Details tab for "datagateway.exe" processes.
[nQSError: 77031] Error occurs while calling remote service DatasourceService. Details: [JDSError : 78] Error Malformed URL	Review the Connection Pool in the semantic model and verify the settings in the General tab and Misc tabs.

Connections or Dataset Issues

Issue reported	Do this
Error Code 603 - no agents are connected	Check that the Data Gateway agent is running. With Data Gateway on Linux: Run $ *<Data Gateway Install Directory>*/domain/bin/status.sh and see if the "Data Gateway Status" is UP or DOWN. For Data Gateway agents on Windows, check the Details tab in Task Manager for "datagateway.exe" processes.
Failed to save the Connection. Invalid Connection details were supplied. Please enter the correct details and try again.	You see this error on the Connection dialog when you create a connection to DB2 or SQL Server. In the agent's Status page, you also see "`REQUEST FINISHED WITH ERRORS`", and if you click the requests, you see "`[JDSError : 110] JDS - Invalid connect string / URL to external source, Cause: Invalid Oracle URL specified`". Workaround: 1. Edit the `<Data Gateway Install Directory>/oracle_common/jdk1.8.0_333/jre/lib/security/java.security file`. 2. Locate this text on line number 720: `jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, \` 3. Change it to: `jdk.tls.disabledAlgorithms=SSLv3, TLSv1, TLSv1.1, RC4, DES, MD5withRSA, \` 4. Restart the agent using `<Data Gateway Install Directory>/domain/bin/stopJetty.sh` followed by `<Data Gateway Install Directory>/domain/bin/startJetty.sh`.
JDSError : 110 - Invalid connect string / URL to external source	Check that the Data Gateway agent can connect to the Data Source. For example, if you're connecting to an Oracle Database, then test using "telnet <hostname> <port>" on the machine where Data Gateway is installed.
No Columns display	Upgrade your Data Gateway agent. This issue typically happens if you're using an earlier Data Gateway agent update that doesn't match your Oracle Analytics Cloud update.
Use Remote Data Connectivity option is missing	Check that the **Enable Data Gateway** option is enabled on the Remote Data Connectivity page in Console.

General Issues

Issue reported	Do this
Agent state change failed with error: Agent name or Oracle Analytics Cloud URL not specified or the Key pair not generated	Click **Save**, then **Enable**. If the problem persists, restart the application. If necessary, check your network.
Authentication failed with error: 401" is returned on Test. Possible reasons include: - The Data Gateway agent key hasn't been copied to the Remote Data Connectivity page in Oracle Analytics Cloud Console. For example, you might have clicked the Test button before pasting the key in the OAC > Console > Remote Data Connectivity page. - The Data Gateway agent key has been regenerated, but the new key hasn't been copied to the Remote Data Connectivity page in Oracle Analytics Cloud Console. For example, you might have already registered a Data Gateway agent in the Remote Data Connectivity page in Console but its ID doesn't match the "id" of the key in the Data Gateway agent Home page.  	If the Data Gateway agent key hasn't been copied, paste the key in the Console to register the Agent. If the Data Gateway agent key has been regenerated, delete the Data Gateway agent in the Console, then paste the key in the Console to re-register the agent.
Authentication failed with error: 404" is returned on Save. This is usually encountered when the Oracle Analytics instance has been updated.	See: - Guidance For Using Remote Data Gateway on Environments Upgraded From Oracle Analytics Cloud 105.2 and Lower (Doc ID 2574387.1) - Oracle Analytics Cloud - Classic : How to Enable Remote Data Gateway in Customer Managed Oracle Analytics Cloud-Classic Instance Upgraded from 105.2 or Lower Release (Doc ID 2632064.1).
Invalid OAC URL /Unknown Host Exception or no error/message is returned on Test. Possible reasons include: - An incorrect URL is specified in the Data Gateway agent Home page. For example, you might have provided a URL like https://oacinst-tenancy.analytics.ocp.oraclecloud.com/dv/ui or https://oacinst-tenancy.analytics.ocp.oraclecloud.com/analytics - There is no suitable network route from the Data Gateway agent to the Oracle Analytics Cloud instance. For example, you might have a proxy server for Internet access and a firewall is blocking access from the Data Gateway agent to Oracle Analytics Cloud. If no Proxy Server is required, then confirm connectivity from the machine where Data Gateway is running to Oracle Analytics Cloud.	If an incorrect URL is specified in the Data Gateway agent Home page, update the URL in the **URL** field. For example, if the Oracle Analytics Cloud URL is `https://<instance details>.oraclecloud.com/dv/ui`, then specify this **URL**: `https://<instance details>.oraclecloud.com`. There is no suitable network route from the Data Gateway agent to the Oracle Analytics Cloud instance: - On Linux, issue the command $ sudo traceroute -T -p 443 https://<instance details>.oraclecloud.com - On Windows, issue the command C:\> telnet https://<instance details>.oraclecloud.com 443. If a Proxy Server is required, then check the proxy details for Data Gateway. See *Invalid Oracle Analytics Cloud URL (Data Gateway can't communicate with Oracle Analytics Cloud)*.

Issue reported	Do this
Invalid Oracle Analytics Cloud URL (Data Gateway can't communicate with Oracle Analytics Cloud)	• Check that you've enabled and configured Data Gateway in Oracle Analytics Cloud Console. • Make sure you can reach the Oracle Analytics Cloud URL from the environment where Data Gateway is running. For example, on Linux you might use a `traceroute` command, such as `sudo traceroute -T -p 443 <Fully qualified domain name of your Oracle Analytics Cloud instance>`. • Make sure there's nothing else blocking communication through the firewall. • If you're using a proxy, navigate to the Home page in the Data Gateway agent, and check the **Proxy** settings for **Host**, **Port**, **User**, and **Password**.
Performance is slow	Review the Logs page and search on : • date • failed jobs • Job ID • 'REMOTE' Once you've found log entries, click on a job and review the Request Status dialog to see the Time Taken in milliseconds. Ask your sales account team for guidance on sizing Data Gateway.
Test fails in Console page Remote Data Connectivity	If the test fails, the Data Gateway agent can't authenticate for various reasons including: • The Data Gateway agent key hasn't been copied to the **Remote Data Connectivity** page in Oracle Analytics Cloud Console. • The Data Gateway agent key has been regenerated, but the new key hasn't been copied to the **Remote Data Connectivity** page in Oracle Analytics Cloud Console. • There's no suitable network route from the Data Gateway agent to Oracle Analytics Cloud.