Oracle® Cloud

Using Oracle AI Data Platform Workbench

G50054-09
April 2026

Oracle Cloud Using Oracle AI Data Platform Workbench,

G50054-09

Primary Author: Adam Donald

Contents

11 Notebooks

12 Workflows

13 Compute

About This Content

This guide is describes how to create, access, manage, and develop data resources in Oracle AI Data Platform Workbench.

Audience

This guide is intended for data scientists and developers who use Oracle AI Data Platform Workbench to:

- Create secure, single-pane of glass data platforms for data governance and management.
- Create coding solutions for managing, analyzing, and enriching data

Conventions

The following text conventions are used in this document:

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

Part I

Introduction to Oracle AI Data Platform

Oracle AI Data Platform simplifies cataloging, ingesting, and analyzing data for data professionals in an organization. The Oracle AI Data Platform service provides the platform and the framework to create data analytics pipelines.

Users are able to create an AI Data Platform, build data catalogs across data in their data estate and set role-based access control (RBAC) policies, and use notebooks powered by Spark to prepare, analyze, and enrich their data. The data catalog enables seamless usage and ingestion of metadata from supported external services. OCI and Oracle Analytics users can connect to Oracle AI Data Platform and analyze their data. All these capabilities are provided in a single pane of glass experience with enterprise security features across all capabilities.

Oracle AI Data Platform is an enterprise-grade, unified platform that simplifies the cataloging, preparation, and analysis of data across an organization's data estate. It brings together data, AI, analytics, and governance services within a single, cohesive experience enabling users to build and operationalize AI-powered applications securely and at scale. Oracle AI Data Platform unifies Oracle Autonomous AI Lakehouse, Oracle Analytics, OCI Object Storage, OCI Generative AI and Oracle Fusion Data Intelligence into a single, governed platform to build, deploy, and scale data and AI applications on your enterprise data.

Within this platform, **Oracle AI Data Platform Workbench** provides a dedicated development environment for data engineers, scientists, and AI developers to design, orchestrate, and deploy data pipelines, models, set role-based access control (RBAC) policies, and use open source technologies such as Spark to prepare, analyze, and enrich their data.

Topics:

1

Overview of Oracle AI Data Platform and Workbench

This chapter provides information and procedures for new users getting started with Oracle AI Data Platform Workbench.

Topics:

What is Oracle AI Data Platform Workbench Used For?

Oracle AI Data Platform Workbench provides an integrated environment for building, orchestrating, and operationalizing data and AI workflows.

Oracle AI Data Platform Workbench is designed for enterprises that need to:

- **Streamline Data Discovery and Governance:** AI Data Platform Workbench provides a centralized metadata repository (Master Catalog) that enhances searchability and governance of structured and unstructured data.
- **Enable Secure Data Collaboration:** Through RBAC-based access control, AI Data Platform Workbench allows different teams to work on shared datasets while maintaining strict security policies.
- **Accelerate Data Preparation and Processing:** With built-in notebooks and workflow orchestration, users can clean, transform, and enrich data efficiently.
- **Support Advanced Analytics and AI/ML:** AI Data Platform Workbench integrates with Apache Spark, allowing data scientists and analysts to run complex computations and model training directly within their data lake.
- **Ensure Seamless Integration Across Data Sources:** AI Data Platform Workbench supports external catalogs from Autonomous Database (ADB), Object Storage (OS), and third-party data sources, enabling users to query and analyze data without duplication.

Managed Integration with Open Source

Oracle AI Data Platform Workbench leverages and extends open-source technologies to provide a powerful yet managed experience.

Some key integrations include:

- **Apache Spark:** AI Data Platform Workbench's compute layer is powered by Spark, enabling scalable, distributed data processing.
- **Delta Lake Support:** AI Data Platform Workbench leverages Delta Lake to enhance data reliability, ACID transactions, and schema evolution.

- **Iceberg & Hudi Compatibility via Delta Uniform:** Through Delta Uniform, AI Data Platform Workbench extends support for Apache Iceberg and Apache Hudi, enabling interoperability across different storage formats. This ensures users can adopt a unified table format strategy while maintaining efficient query execution and data governance.
- **JDBC Integration for BI Tools:** AI Data Platform Workbench provides JDBC drivers, allowing seamless connectivity with external BI tools like Oracle Analytics Cloud (OAC) and third-party visualization platforms.

Personas for Oracle AI Data Platform Users

Oracle AI Data Platform serves a variety of users across different roles within an organization, each with unique needs and requirements.

Here's a general overview of the key personas who interact with AI Data Platforms:

- **Data Engineers** - Data engineers work with large-scale data pipelines, transforming raw data into usable formats for analysis. They rely on AI Data Platform's robust capabilities to design and manage data workflows, ingest data from various sources, and ensure data quality. They are highly focused on automating processes, optimizing compute resources, and integrating different data systems seamlessly.
- **Data Analysts** - Data analysts use AI Data Platform to discover, analyze, and generate insights from data. They require an intuitive interface and tools for querying and analyzing large datasets. AI Data Platform empowers them with interactive notebooks and seamless integration with business intelligence (BI) tools, helping them transform raw data into actionable insights for decision-makers.
- **Data Scientists** - Data scientists leverage AI Data Platform's scalable compute capabilities for machine learning and advanced analytics tasks. They need access to diverse datasets, powerful processing tools, and the ability to run complex models. AI Data Platform's Spark-powered notebooks, AI/ML integration, and support for open-source libraries enable data scientists to build, test, and deploy models within the platform.
- **Data Stewards** - These users ensure that all data is handled in compliance with industry regulations and organizational policies. They focus on maintaining data privacy, auditing access, and monitoring data usage across the organization. AI Data Platform helps them manage metadata, enforce role-based access controls (RBAC), and ensure proper governance through cataloging, lineage tracking, and security policies.

Common Use Cases for Oracle AI Data Platform

Oracle AI Data Platform serves a variety of use cases across industries and business functions.

Medallion Architecture

- Implement a Medallion Architecture with bronze, silver, and gold layers.
- Use Delta Uniform and Iceberg for efficient data storage and query optimization.
- Enable zero-copy access to external data sources for seamless analytics.

ETL & Data Engineering

- Use Spark-based workflows and notebooks to process, transform, and enrich raw data.
- Automate data pipelines with low-code/no-code workflow orchestration.
- Handle large-scale batch processing and real-time data ingestion.

Machine Learning, AI and Data Science

- Train and deploy machine learning models using Spark-powered notebooks.
- Enable large-scale feature engineering and data transformation.
- Provide managed execution environments for Python and PySpark workloads.

Building AI Agents Leveraging Enterprise Data

- Create conversational AI agents to assist with retrieving and developing data.

Enterprise Data Catalog & Governance, Delta Sharing

- Centralized metadata management for structured and unstructured data.
- Role-based access control (RBAC) for secure data access and collaboration.
- Integration with external catalogs, including Autonomous Database (ADB) and Object Storage.
- Oracle AI Data Platform supports Delta Sharing, enabling secure, real-time, and governed data sharing across organizational boundaries.

Analytics, Business Intelligence & Reporting

- Connect OCI Oracle Analytics Cloud (OAC) and third-party BI tools via JDBC like Tableau, Power BI.

Multi-Cloud & Hybrid Data Integration

- Enable federated query execution across multiple OCI services.
- Integrate with third-party cloud storage and databases for hybrid analytics.
- Maintain data sovereignty and compliance across multiple environments.

2

Get Started with Oracle AI Data Platform Workbench

Oracle AI Data Platform Workbench helps you quickly build, orchestrate, and manage end-to-end data workflows across your organization. This guide walks you through the essential steps to create your first AI Data Platform Workbench and start working with data in a governed and scalable environment.

Topics:

Create Your First AI Data Platform Workbench

Before you create your first Oracle AI Data Platform Workbench, you need to ensure you have the correct IAM permissions.

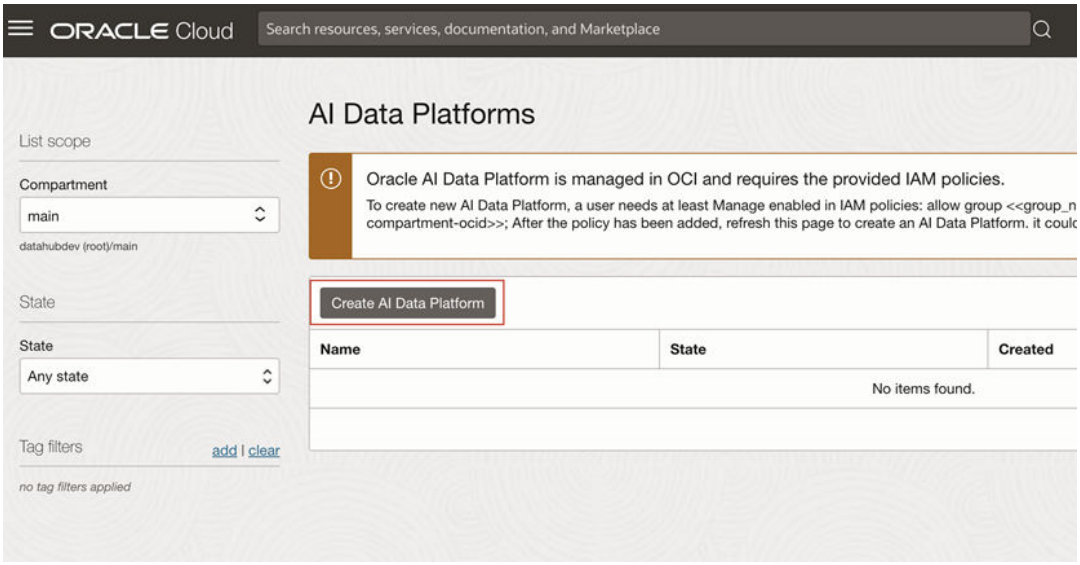
 LiveLabs Sprint

 Video

Before creating or using an AI Data Platform Workbench, check the following:

- You have the appropriate permissions for the compartment where you want to create the AI Data Platform Workbench. For more information, see Managing Compartments.
- If you already have tag-defaults set for your compartment you may face issues while creating an AI Data Platform Workbench instance in that scope. For more information, see Known Issues.

1. In the Home section of OCI, click **Analytics & AI**.
2. Click **AI Data Platform Workbenches**.



3. Click **Create AI Data Platform Workbench**. Oracle AI Data Platform Workbench performs a check to validate that all required IAM policies are in place.
4. Provide a name and description for your AI Data Platform Workbench instance.
5. Provide a name and description for your first AI Data Platform Workbench workspace.
6. An Oracle Autonomous AI Lakehouse instance is required for AI features. Select whether to create a new instance or use an existing instance.
   - For **Create new**, provide a new password for the ADMIN user of the instance.
   - For **Choose existing**, select the compartment and instance and enter the password for the ADMIN user of the existing instance. Your existing instance must be 26ai or above.
7. Select one of the following from **Add Policies**:
   - **Standard** (Recommended): Applies access settings broadly at the tenancy level.
   - **Advanced:** Allows you to configure fine-grained access at the compartment level.
8. Review the missing policies, if any, and apply them as necessary.
9. Expand **Optional Policies**. Review and add any additional policies your instance requires. For more information, see IAM Policies for Oracle AI Data Platform Workbench.
10. Click **Advanced Options** to provide metadata tags for your AI Data Platform Workbench. Click **Add tag** to add multiple.
11. Click **Create AI Data Platform Workbench**.

Access an Oracle AI Data Platform Workbench

You can access any Oracle AI Data Platform Workbench that you or a role you belong to have been provided permissions for.

You require at least USE permissions from the AI Data Platform Workbench IAM policies to access an AI Data Platform Workbench. For more information, see IAM Policies for Oracle AI Data Platform Workbench.

1. In the Home section of OCI, click **Analytics & AI**.

2. Click **AI Data Platform Workbench** then click **AI Data Platform Workbenches**.
3. Click the AI Data Platform Workbench you want to access.

Access Oracle AI Data Platform Workbench from a URL

You can access your Oracle AI Data Platform Workbench instance directly from a URL.

The URL for your Oracle AI Data Platform Workbench uses the following format, where *tenant_name* is the name of the OCI account you use to log in and *domain_name* is the domain you select:

```
https://<hash>.datalake.oci.oraclecloud.com/#?
&tenant=<<tenant_name>>&domain=<<domain_name>>
```

You can also get a direct link to your instance by right-clicking your AI Data Platform Workbench instance name on the home page and clicking **Copy Link**.

You can share your AI Data Platform Workbench instance URL with other users. Those users need to have appropriate permissions to access the instance. For more information, see About Permissions.

Understanding the Workspace and UI Flow

Once you've created or accessed your AI Data Platform Workbench, the user interface helps you navigate and manage all key components seamlessly inside the workbench.

Left Navigation Overview

When inside an AI Data Platform Workbench, the left sidebar contains:

- Home - Brings you back to the dashboard
- Master Catalog - View all registered data assets across workspaces
- Workspace - Direct access to workspace details
- Select Workspace - Dropdown to switch between different workspaces
  - Inside a workspace, you'll see:
    - Workflow - Design and schedule data workflows
    - Compute - Manage Spark compute environments
- Data Sharing - Share data with other services supporting Delta Share protocol
- Auto populate catalog - Automatically ingest metadata from connected sources
- Notifications - Search and filter system messages you receive
- Roles - Configure access control (RBAC)
- Audit logs - Search and filter the history of objects in your AI Data Platform Workbench

Next Steps After Setup

Once you have created your Oracle AI Data Platform Workbench, you can prepare it for regular use.

When you have your Oracle AI Data Platform Workbench created and have accessed it for the first time, these are the next steps to ensure its ready for use:

- Review IAM Policies and Roles - Ensure correct access is set up for users and groups via the Roles tab. For more information, see Roles.
- Set Up Compute - Configure Spark compute in the Compute tab to run your workflows efficiently. For more information, see Compute.
- Ingest and Organize Data - Start by creating a Catalog to organize your data sources. For more information, see Data Management.
- Explore Notebooks - Use notebooks for ad hoc exploration, transformations, or machine learning tasks. For more information, see Notebooks.

Edit an AI Data Platform Workbench

You can edit the details of an Oracle AI Data Platform Workbench you own.

1. In the Home section of OCI, click **Analytics & AI**.
2. Click **AI Data Platform Workbench** then click **AI Data Platform Workbenches**.
3. Click ⋯**Actions** next to your AI Data Platform Workbench and click **Edit**.
4. Modify the details and click **Save**.

Delete an AI Data Platform Workbench

You can delete unused or redundant AI Data Platform Workbenches you own.

1. In the Home section of OCI, click **Analytics & AI**.
2. Click **AI Data Platform Workbench** then click **AI Data Platform Workbenches**.
3. Click ⋯ **Actions** next to your AI Data Platform Workbench and click **Delete**.
4. Select whether to delete:
   - All managed data in the Master Catalog
   - All files and folders in all Workspaces
5. Enter `Delete` where prompted. Click **Delete**.

IAM Policies for Oracle AI Data Platform Workbench

Oracle AI Data Platform Workbench is managed in OCI and requires the provided IAM policies.

To create new AI Data Platform Workbench instances, a user needs at least MANAGE enabled in IAM policies:

```
allow group <aidpAdminIdentityDomain>/<aidpAdminGroup> to manage ai-data-
platforms in compartment id <aidpCompartmentId>
```

Oracle AI Data Platform Workbench allows users two different combination of policies either of which they can choose from to set up their instance.

Option 1: Tenancy-level Policies (Broad Scope)

With this option, your policies are defined at the **tenancy** (root) level, giving your Oracle AI Data Platform Workbench broad access across compartments.

- Minimizes the need to write new IAM policies every time you add new workloads, data sources, or compartments.
- Easiest onboarding experience; requires the least changes after initial setup.
- Users have a broader scope of permissions.
- May not meet strict least-privilege requirements in regulated environments.

1. **Allow Oracle AI Data Platform Workbench service to view OCI IAM resources to configure role-based access control of AI Data Platform managed resources:**

   ```
   allow any-user TO {AUTHENTICATION_INSPECT, DOMAIN_INSPECT, DOMAIN_READ,
   DYNAMIC_GROUP_INSPECT, GROUP_INSPECT, GROUP_MEMBERSHIP_INSPECT,
   USER_INSPECT, USER_READ} IN TENANCY where all
   {request.principal.type='aidataplatform'}
   ```

2. **Allow Oracle AI Data Platform Workbench service to create OCI logging log group and provide logs to users:**

   ```
   allow any-user to manage log-groups in compartment id <aidpCompartmentId>
   where ALL { request.principal.type='aidataplatform' }
   allow any-user to read log-content in compartment id <aidpCompartmentId>
   where ALL { request.principal.type='aidataplatform' }
   ```

3. **Allow Oracle AI Data Platform Workbench service to provide metrics to users:**

   ```
   allow any-user to use metrics in compartment id <aidpCompartmentId> where
   ALL {request.principal.type='aidataplatform',
   target.metrics.namespace='oracle_aidataplatform'}
   ```

4. **Allow Oracle AI Data Platform Workbench service on create and manage OCI Object Store Bucket for workspace and managed data in Master Catalog:**

   ```
   allow any-user to manage buckets in tenancy where all
   { request.principal.type='aidataplatform', any {request.permission =
   'BUCKET_CREATE', request.permission = 'BUCKET_INSPECT', request.permission
   = 'BUCKET_READ', request.permission = 'BUCKET_UPDATE'}}
   ```

5. **Allow Oracle AI Data Platform Workbench service to govern/manage data in Workspace and Master Catalog with restricted access to per AI Data Platform Workbench instance level:**

   ```
   allow any-user to {TAG_NAMESPACE_USE} in tenancy where all
   {request.principal.type = 'aidataplatform'}
   allow any-user to manage buckets in tenancy where all
   { request.principal.id=target.resource.tag.orcl-aidp.governingAidpId, any
   {request.permission = 'BUCKET_DELETE', request.permission = 'PAR_MANAGE',
   request.permission = 'RETENTION_RULE_LOCK', request.permission =
   'RETENTION_RULE_MANAGE'} }
   allow any-user to read objectstorage-namespaces in tenancy where all
   { request.principal.type='aidataplatform', any {request.permission =
   ```

```
'OBJECTSTORAGE_NAMESPACE_READ'}}
allow any-user to manage objects in tenancy where all
{ request.principal.id=target.bucket.system-tag.orcl-
aidp.governingAidpId  }
```

6. **Allow Oracle AI Data Platform Workbench service to configure Compute Cluster to access data in a private network (Optional):**

```
allow any-user to manage vnics in compartment id <aidpCompartmentId> where
all { request.principal.type='aidataplatform'}
allow any-user to use subnets in compartment id <aidpCompartmentId> where
all { request.principal.type='aidataplatform'}
allow any-user to use network-security-groups in compartment id
<aidpCompartmentId> where all { request.principal.type='aidataplatform'}
```

7. **Allows the Object Storage service to automatically apply lifecycle actions (such as permanent deletion or archival) to your Oracle AI Data Platform Workbench workspace data, reducing manual maintenance effort and supporting compliance with data retention best practices (Optional):**

```
allow service objectstorage-<<region_identifier>> to manage object-family
in compartment id <<aidp-compartment-ocid>>
```

Option 2: Compartment-Level Policies (Fine-grained Scope)

With this option, your policies are defined at the **compartment** level, meaning the compartment where your AI Data Platform instance is created.

- Provides you a tighter security boundary; limits your AI Data Platform Workbench's access to a single compartment by default.
- You can add new compartment policies incrementally when workflows need to span additional compartments.
- Requires you to make manual IAM updates whenever you need your AI Data Platform Workbench to access a different compartment.
- Requires more operational overhead during expansion.

1. **Allow Oracle AI Data Platform Workbench service to view OCI IAM resources to configure role-based access control of AI Data Platform managed resources:**

```
allow any-user TO {AUTHENTICATION_INSPECT, DOMAIN_INSPECT, DOMAIN_READ,
DYNAMIC_GROUP_INSPECT, GROUP_INSPECT, GROUP_MEMBERSHIP_INSPECT,
USER_INSPECT, USER_READ} IN TENANCY where all
{request.principal.type='aidataplatform'}
```

2. **Allow Oracle AI Data Platform Workbench service to create OCI logging log group and provide logs to users:**

```
allow any-user to manage log-groups in compartment id <aidpCompartmentId>
where ALL { request.principal.type='aidataplatform' }
allow any-user to read log-content in compartment id <aidpCompartmentId>
where ALL { request.principal.type='aidataplatform' }
```

3. **Allow Oracle AI Data Platform Workbench service to provide metrics to users:**

```
allow any-user to use metrics in compartment id <aidpCompartmentId> where
ALL {request.principal.type='aidataplatform',
target.metrics.namespace='oracle_aidataplatform'}
```

4. **Allow Oracle AI Data Platform Workbench service on create and manage OCI Object Store Bucket for workspace and managed data in Master Catalog:**

```
allow any-user to manage buckets in compartment id <aidpCompartmentId>
where all { request.principal.type='aidataplatform', any
{request.permission = 'BUCKET_CREATE', request.permission =
'BUCKET_INSPECT', request.permission = 'BUCKET_READ', request.permission =
'BUCKET_UPDATE'}}
```

5. **Allow Oracle AI Data Platform Workbench service to govern/manage data in Workspace and Master Catalog with restricted access to per AI Data Platform Workbench instance level:**

```
allow any-user to {TAG_NAMESPACE_USE} in tenancy where all
{request.principal.type = 'aidataplatform'}
allow any-user to manage buckets in compartment id <aidpCompartmentId>
where all { request.principal.id=target.resource.tag.orcl-
aidp.governingAidpId, any {request.permission = 'BUCKET_DELETE',
request.permission = 'PAR_MANAGE', request.permission =
'RETENTION_RULE_LOCK', request.permission = 'RETENTION_RULE_MANAGE'} }
allow any-user to read objectstorage-namespaces in compartment id
<aidpCompartmentId> where all { request.principal.type='aidataplatform',
any {request.permission = 'OBJECTSTORAGE_NAMESPACE_READ'}}
allow any-user to manage objects in compartment id <aidpCompartmentId>
where all { request.principal.id=target.bucket.system-tag.orcl-
aidp.governingAidpId  }
```

6. **Allow Oracle AI Data Platform Workbench service to configure Compute Cluster to access data in a private network (Optional):**

```
allow any-user to manage vnics in compartment id <aidpCompartmentId> where
all { request.principal.type='aidataplatform'}
allow any-user to use subnets in compartment id <aidpCompartmentId> where
all { request.principal.type='aidataplatform'}
allow any-user to use network-security-groups in compartment id
<aidpCompartmentId> where all { request.principal.type='aidataplatform'}
```

7. **Allows the Object Storage service to automatically apply lifecycle actions (such as permanent deletion or archival) to your Oracle AI Data Platform Workbench workspace data, reducing manual maintenance effort and supporting compliance with data retention best practices (Optional):**

```
allow service objectstorage-<<region_identifier>> to manage object-family
in compartment id <<aidp-compartment-ocid>>
```

Additional Policies for External Tables

If your AI Data Platform Workbench instance needs to access data stored in a different compartment, you must grant additional policies for that external compartment. These policies

allow AI Data Platform Workbench to inspect, read, and manage buckets and objects in the external compartment to use it inside AI Data Platform Workbench workspace.

```
allow any-user to manage buckets in compartment id <external-data-
CompartmentId> where all { request.principal.type='aidataplatform', any
{request.permission = 'BUCKET_INSPECT', request.permission = 'BUCKET_READ',
request.permission = 'BUCKET_UPDATE'}}
allow any-user to manage buckets in compartment id <external-data-
CompartmentId> where all { request.principal.id=target.resource.tag.orcl-
aidp.governingAidpId, any {request.permission = 'PAR_MANAGE',
request.permission = 'RETENTION_RULE_LOCK', request.permission =
'RETENTION_RULE_MANAGE'} }
allow any-user to manage objects in compartment id <external-data-
CompartmentId> where all { request.principal.id=target.bucket.system-tag.orcl-
aidp.governingAidpId }
allow service objectstorage-<<region_identifier>> to manage object-family in
compartment id <external-data-CompartmentId>
```

> **Note**
>
> If you are using a custom identity domain (non-default), you must prefix the group name with the domain name in your IAM policy. For example:

```
allow group <aidpAdminIdentityDomain>/<aidpAdminGroup> to manage ai-data-
platforms in compartment id <aidpCompartmentId>
```

For more information on IAM policies, see IAM Policies Overview.

To see and login to an AI Data Platform Workbench, you need to be granted access by the administrator of that AI Data Platform Workbench.

Pricing

AI Data Platform Unit (AIDP Unit) is the unit of measurement used to meter consumption in your Oracle AI Data Platform Workbench.

AIDP Unit is a measure of the AI, data management, and data processing work done in your AI Data Platform Workbench instance. The rate at which AIDP Units are charged depends on the use of OCPU and memory resources per hour.

You can see how different cluster configurations are priced by using our Pricing Widget.

Table 2-1 AI Data Platform Units Mapping

AI Data Platform Definition	AIDP Units Charged
AMD OCPU per Hour (with included 100 GB Block storage per OCPU)	67 AIDP Units
ARM OCPU per Hour (with included 100 GB Block storage per OCPU)	27 AIDP Units
Intel OCPU per Hour (with included 100 GB Block storage per OCPU)	87 AIDP Units

Table 2-1 (Cont.) AI Data Platform Units Mapping

AI Data Platform Definition	AIDP Units Charged
NVIDIA GPU per Hour (with included Block Volume, CPU, Memory)	4110 AIDP Units
AMD Memory GB per hour	3 AIDP Units
ARM Memory GB per hour	
Intel Memory GB per hour	

You pay AIDP Units to AI Data Platform Workbench for all your compute clusters and features based on the definitions in the table above. For object storage, logging, metrics, or any databases on OCI, you pay directly to OCI.

You select Compute Shape options between AMD, Intel, ARM, and NVIDIA GPUs. More specific or granular shapes are not currently available. Your selection of compute shapes is subject to the availability of compute shapes in the OCI region where your AI Data Platform Workbench instance is located. Oracle reserves the right to change the specific compute shapes available in an OCI region.

Compute clusters are a combination of compute OCPU and memory. When you create a compute cluster in AI Data Platform Workbench, you see an estimated cost of that compute cluster per hour in AIDP Units. For example:

- The cost Per Hour for an AMD 2 OCPU 32 GB Memory cluster will be 2 OCPU x 67 AIDP Units + 32 GB Memory x 3 AIDP Units = 230 AIDP Units.
- The cost Per Hour for an Intel 4 OCPU 32 GB Memory cluster will be 4 OCPU x 87 AIDP Units + 32 GB Memory x 3 AIDP Units = 444 AIDP Units.
- The cost Per Hour for an ARM 4 OCPU 32 GB Memory cluster will be 4 OCPU x 27 AIDP Units + 32 GB Memory x 3 AIDP Units = 204 AIDP Units.
- The cost Per Hour for a NVIDIA GPU cluster with 2 GPU will be 2 GPU x 4110 AIDP Units = 8220 AIDP Units.

By default, one Master Catalog Default Cluster is present in each AI Data Platform Workbench instance. This cluster is responsible for the essential AI Data Platform Workbench functions, like search crawls, refreshing catalog objects, creating, editing, and deleting objects, testing connections etc. Default Master Catalog Compute Clusters have AMD 2 OCPU 32 GB Memory. For the Default Master Catalog Compute Cluster, you are charged hourly for a minimum of 230 AIDP Units. Deleting an AI Data Platform Workbench instance halts the ongoing cost of that Default Master Catalog Compute Cluster. If you add more capacity to the Default Master Catalog Compute cluster to meet your performance and scalability needs, the AIDP Units cost of the Default Master Catalog scales up relative to the increase.

Sample Pricing Scenario

In this scenario, you have 2 compute clusters running for a month (all calculations are in USD):

- 1 Default Master Catalog Compute Cluster: AMD 2 OCPU with 32 GB Memory
- 1 custom Cluster for workloads: Intel 4 OCPU with 32 GB Memory

In this case, the cost calculation for your AMD Cluster (for Default Master Catalog Compute Cluster) is:

- 2 OCPUs 32 GB Memory Per Hour = (2*67 + 32*3) AIDP Units =230 AIDP Units
- 230 AIDP Units/hour = 171,120 AIDP Units/month

- $0.230/hour = $171.12/month

The cost calculation for your Intel Cluster (for workloads) is:

- 4 OCPUs 32 GB Memory Per Hour = (4*87 + 32*3) AIDP Units = 444 AIDP Units
- AIDP Units 444 AIDP Units/hour = 330,370 AIDP Units/month
- $0.444/hour = $330.37/month

Adding your monthly costs together, you total monthly costs would amount to: $171.12/month + $330.37/month = $501.58/month.

3

Features of Oracle AI Data Platform Workbench

Oracle AI Data Platform Workbench is a modern data platform designed to simplify data ingestion, processing, and analytics at scale. It provides a seamless integration of compute, storage, and cataloging capabilities to enable efficient data management.

Key features of AI Data Platform Workbench include:

Workspace

A workspace in AI Data Platform Workbench acts as an isolated environment where users can manage and organize their data lake resources, including workflows, notebooks, and libraries. Workspaces enable efficient collaboration and governance by keeping resources grouped logically.

Compute

AI Data Platform Workbench provides scalable CPU and GPU compute resources for executing data processing and analytics workloads. Users can leverage Spark-based execution environments for high-performance processing, supporting batch and interactive workloads.

Notebook

AI Data Platform Workbench includes notebooks as an interactive development environment for writing and executing code. It supports Python and SparkSQL enabling users to transform, analyze, and visualize data directly within AI Data Platform.

Workflow

The workflow component allows users to define and orchestrate data pipelines made of notebooks, Python tasks, if-else, and other job tasks. Users can create, schedule, and monitor workflows for ETL, data transformations, and analytics automation.

Master Catalog

The Master Catalog serves as the central metadata repository for all structured and unstructured datasets within an AI Data Platform Workbench. It provides unified governance and data discovery, allowing users to search and manage datasets across different schemas and storage locations.

Catalog

A catalog in an AI Data Platform Workbench is a logical grouping of schemas, tables, volumes, and models, providing a structured way to organize datasets. Users can create multiple catalogs for different projects or teams to ensure effective data segmentation.

Schema

A schema defines the structure within a catalog, organizing tables and views under a common namespace. Schemas help in logically structuring data for different applications and analytics workloads.

Table

A table in an AI Data Platform Workbench represents structured datasets that can be queried and processed. Tables support various storage formats, including Delta Uniform, ensuring compatibility with multiple query engines.

View

A view is a virtual table in an AI Data Platform Workbench that provides a queryable representation of data stored in underlying tables. Views allow for simplified access to transformed datasets without requiring data duplication.

Volume

A volume is a storage abstraction in an AI Data Platform Workbench that provides a managed space for persisting raw, processed, and curated data. It supports efficient data access and integration with Object Storage.

Auto Populate

The Auto Populate feature simplifies metadata management by automatically detecting and registering new datasets located in OCI Object Storage. This reduces manual effort in keeping data catalogs up to date.

Role-Based Access Controls (RBAC)

AI Data Platform implements RBAC to enforce fine-grained access control across different resources. Users can define roles and permissions for workspaces, catalogs, and datasets to ensure secure collaboration.

Audit Log

Audit logs in Oracle AI Data Platform Workbench capture detailed records of user activities. These logs help monitor usage, ensure compliance, and investigate issues such as unauthorized access or configuration changes.

Three-Part Namespace

AI Data Platform Workbench adopts a three-part namespace (Catalog.Schema.Table) for accessing datasets, enabling a structured and consistent way to reference data across the platform. This standardization improves interoperability and ease of access.

Part II
Data Management

Oracle AI Data Platform Workbench enables organizations to effectively manage all their data and metadata across OCI Object Storage and other external sources like Autonomous Database, Kafka, etc. It provides centralized metadata across your data estate and enables enterprises to define unified access control for their structured, semi-structured and unstructured data.

AI Data Platform Workbench helps enterprises solve their data management use cases seamlessly. Various personas work in tandem in an organization to deliver value to the business leaders.

- Data stewards need to discover data assets and entities to understand where data is located, how it is structured and used, thus effectively managing the data/metadata life cycle.
- Data admins organize data in catalogs, schemas, tables, and volumes, to ensure efficient and secure storage, organization, and retrieval of data.
- Data engineers and analysts need to share data with other analysts or business leaders to unlock the true value of data

Discover Data

All data assets in the AI Data Platform Workbench can be discovered seamlessly using:

- Master Catalog Explorer
- Catalog Explorer in Workspace while working with notebooks, sql/python files
- SQL grammar like SHOW, LIST and DESCRIBE
- APIs

Organize Data

You can organize the data in catalogs, schemas, tables, volumes:

- **Standard Catalog:** A standard catalog is a logical container for schemas (databases), users can create tables, views and volumes in a schema. Standard catalogs manage the lifecycle of metadata of all child objects.
- **External Catalog:** An external catalog is backed by external data sources like Autonomous Database. In case of external catalog, the only metadata is synced from the external source and users can query the data residing in an external source using the 3-part name like: catalog_name.schema.name.table_name. In case of external catalog the metadata lifecycle is managed by the external source and the Master Catalog keeps a copy of the metadata. External Catalog only harvests the metadata from the external source, the data is not copied into your AI Data Platform.

You can choose to let AI Data Platform Workbench manage the metadata lifecycle, by creating:

- External tables, by defining a table, its schema, and referring to a location in OCI Object Storage, or
- External volume, by defining a volume referring to a location in OCI Object Storage and then further storing files and folder in the volume

You can also choose to let AI Data Platform Workbench manage the data and metadata lifecycle, by creating:

- Managed table and AI Data Platform Workbench manages the OCI Object Storage location in customer's tenancy
- Managed volume and AI Data Platform Workbench manages the OCI Object Storage location in customer's tenancy so that users can store files and folders (semi-structured or unstructured data) in the volume

Data Sharing

Data Share in AI Data Platform Workbench enables users to share data assets with users in the organization as well as outside the organization. Data Sharing in AI Data Platform Workbench is built on top of open source Delta Share protocol and to ensure that Data is shared in a secure manner, you can enforce permissions on who can share and create recipients.

Auto Populate

The Auto Populate feature simplifies metadata management by automatically detecting and creating data entities in a selected standard catalog. This automates the process of manually creating huge number of tables by enabling users to create metadata extractors by pointing to data location in OCI Object Storage.

Topics:

4

Manage with Master Catalog

This chapter helps you use and understand the master catalog, standard and external catalogs, schema, tables, and volumes.

Topics:

Master Catalog

Master Catalog in AI Data Platform Workbench is the top level entity that enables you to manage your data and metadata by providing a centralized view.

Video

Master Catalog is a container for both standard and external catalogs. You create catalogs with their data assets in Oracle Autonomous AI Lakehouse, OCI Object Storage, and Kafka. Master Catalog allows you to enforce permissions on its child objects.

Standard and external catalogs have different functions and use cases:

- **Standard catalog:** A standard catalog is a logical container for schemas (databases), users can create tables, views and volumes in a schema. Standard catalog manages the lifecycle of metadata of all child objects.
- **External catalog:** An external catalog is backed by external data sources like Oracle Autonomous AI Lakehouse, Kafka, etc. In case of external catalog, the metadata is synched from the external source and users can query the data in an external source using the 3-part name like: *catalog_name.schema_name.table_name*. In case of external catalog the metadata lifecycle is managed by the external source and the Master Catalog keeps a copy of the metadata.

Use Cases for Master Catalog

Master catalogs can be leveraged to help with data preparation and analysis, storing unstructured data, and more.

Query and Analyze Data Using SQL Syntax

Create managed or external tables in a standard catalog to query and analyze data using familiar SQL-like syntax, making it easier to explore and understand the data stored in AI Data Platform.

Data Preparation

Leverage structured format of data stored in managed/external tables for preparing data for machine learning models, making it easier to clean, transform, and feature engineer data. This facilitates efficient data access and processing for feature engineering and model training

Time Travel

Open table formats support schema evolution. The structure of the data can change over time without rewriting the entire dataset. These tables can be versioned and users can run time travel queries allowing you to query historical versions of data, facilitating retrospective analysis and data recovery.

ACID Transaction Support

Open table formats support full Create, Read, Update, and Delete (CRUD) operations, ensuring data consistency and enabling data updates. Tables can be used to store and manage transactional data, enabling applications to track changes to data.

Efficiently Read and Write Data

Tables in AI Data Platform Workbench can be partitioned, allowing for efficient data access and processing, especially for large datasets.

Store and Process Unstructured Data

Create managed or external volumes to store unstructured data so that they can be processed using Apache Spark.

Standard Catalogs

Standard catalogs are created and managed inside of your AI Data Platform Workbench.

 LiveLabs Sprint

You create standard catalogs inside your master catalog. Users with the required permissions can view, modify, or create schema, volumes, and tables inside standard catalogs. Standard catalogs can contain external tables and volumes.

Create a Standard Catalog

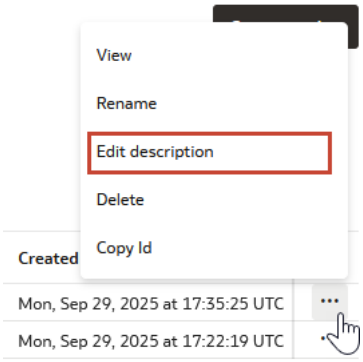
You can create a catalog that connects to data in your master catalog.

1. Click **Create** in the left navigation pane and select **Catalog**. You can also navigate to the **Master catalog** and click **Create Catalog in Master Catalog**.
2. Fill in the name and description fields.
3. From the **Catalog Type** drop-down list, select **Standard Catalog**.
4. Click **Browse** and select the compartment where you want to create the bucket for your catalog. Click **Select**.
5. Click **Create**.

Edit a Standard Catalog Description

You can edit the description for standard catalogs after creation if their contents or purpose has changed.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** and click **Edit description**.



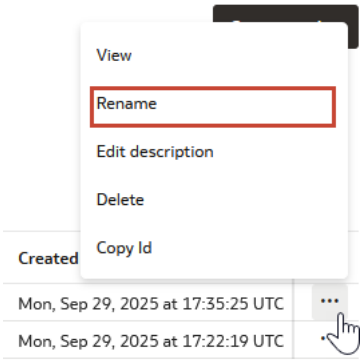
3. Make your changes to the catalog description and click **Save**.

Rename a Standard Catalog

You can rename your catalogs to provide a descriptive label when the contents or purpose of the catalog has changed.

You can't rename the default catalog.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** and click **Rename**.



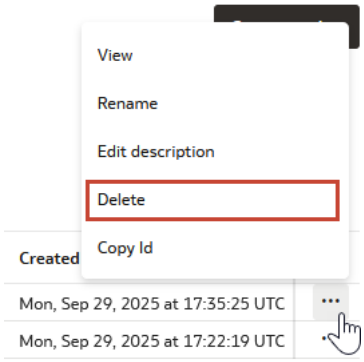
3. Make your changes to the catalog name and click **Save**.

Delete a Standard Catalog

You can delete standard catalogs to remove all locally stored metadata as well as data.

Deleting a standard catalog deletes both the data and the stored metadata.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** then click **Delete**.



3. Select **Confirm deletion of the catalogs**.
4. Click **Delete**.

External Catalogs

External catalogs are catalogs where the data is sourced from a location outside Oracle AI Data Platform Workbench.

LiveLabs Sprint

External catalogs connect to sources like Oracle Autonomous AI Lakehouse, Oracle AI Database, and Oracle Autonomous AI Transaction Processing to extract data into your AI Data Platform Workbench. External catalogs use the credentials provided during the external catalog creation for querying the external source. For more information on data sources, see Internal Sources.

For example, if you create an external catalog for an Autonomous AI Lakehouse instance where the Autonomous AI Lakehouse user credentials used have access to *schema1* but not *schema2*, only *schema1* appears in the external catalog. Users with permissions for the external catalog can only query the schema the Autonomous AI Lakehouse user has access to.

> **Note**
>
> AI Data Platform Workbench does not support harvesting data from schemas and tables shared across Pluggable Databases (PDBs) or from Oracle-maintained schemas and tables.

Required Permissions for Autonomous AI Lakehouse and Autonomous AI Transaction Processing

When you create an external catalog in AI Data Platform Workbench the user credentials you use to connect should have at least the following permissions:

- CREATE SESSION to connect to the database
- SELECT access on the required objects (tables/views/external tables) via least-privilege grants or a dedicated read role
- READ, WRITE on DIRECTORY DATA_PUMP_DIR

If you are inserting data or creating a new table in the external catalog, ensure the user is part of the DWROLE. For more information, refer to the Oracle Autonomous AI Transaction Processing documentation, Manage User Privileges on Autonomous AI Database - Connecting with a Client Tool.

Limitations

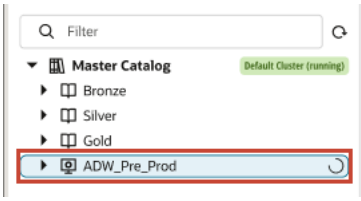
External catalogs support Query and DML. DDL is not supported, even when the credentials used to create the external catalog has permissions to execute DDL statements.

Create an External Catalog

You can connect a catalog from your AI Data Platform Workbench to a external source.

1. Click **Create** in the left navigation pane and select **Catalog**. You can also navigate to the **Master catalog** and click ⊕ **Create Catalog in Master Catalog**.
2. Fill in the name and description fields.
3. From the **Catalog Type** drop-down list, select **External Catalog**.
4. Select the external source type.
   - For Oracle Autonomous AI Lakehouse, provide either a wallet file, or the instance configuration.
   - For Oracle Autonomous AI Transaction Processing, provide either a wallet file, or the instance configuration.
   - For Oracle AI Database, provide either a wallet file, or the instance configuration.
   - For Kafka, provide the bootstrap server. Separate multiple servers with a comma. **(Coming soon)**
5. Fill in the user name and password.
6. SSL is enabled by default. Clear the box to disable SSL.
7. Click **Create**.

   External catalogs that are extracting data from an external source display a spinning circle icon.



You can also monitor progress from **Job Runs**.

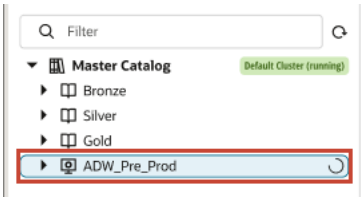
Create an External Catalog for Private Networks

You can create an external catalog that accesses data sources in a private network.

1. Click **Create** in the left navigation pane and select **Catalog**. You can also navigate to the **Master catalog** and click **Create Catalog in Master Catalog**.
2. Fill in the name and description fields.
3. From the **Catalog Type** drop-down list, select **External Catalog**.
4. Select the external source type.
   - For Oracle Autonomous AI Lakehouse, provide either a wallet file, or the instance configuration.
   - For Oracle Autonomous AI Transaction Processing, provide either a wallet file, or the instance configuration.
   - For Oracle AI Database, provide either a wallet file, or the instance configuration.
   - For Oracle Exadata Database Service, provide host, port, and service name (SID).
   - For Kafka, provide the bootstrap server. Separate multiple servers with a comma. **(Coming soon)**
5. Fill in the user name and password.
6. SSL is enabled by default. Clear the box to disable SSL.
7. Select **Enable private network**.
8. Select the workspace with the desired private network configuration.

   For information on setting up a workspace configured for private networks, see Create a Workspace with Private Network Access Enabled.
9. Click **Create**.

   External catalogs that are extracting data from an external source display a spinning circle icon.



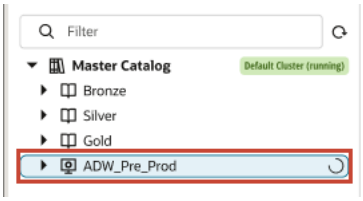
You can also monitor progress from **Job Runs**.

Refresh External Data Catalogs

You can trigger a refresh of all external catalogs to update their contents from the source.

1. On the Home page, click **Master Catalog**.
2. Select the external catalog you want to refresh.
3. Click **Refresh**.

When you click refresh, workflows start in the background to extract and update metadata from external catalogs. Catalogs that are extracting data from an external source display a spinning circle icon.

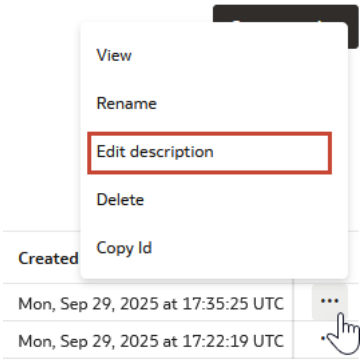


You can also monitor progress from **Job Runs**.

Edit an External Catalog Description

You can edit the description for external catalogs after creation if their contents or purpose has changed.

1. On the Home page, click **Master catalog**.
2. Next to your external catalog, click ··· **Actions** and click **Edit description**.



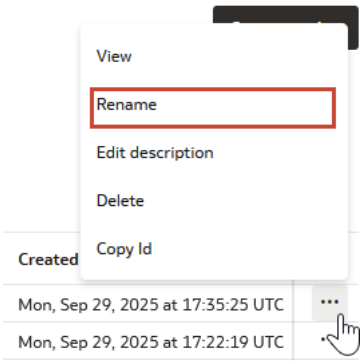
3. Make your changes to the catalog description and click **Save**.

Rename an External Catalog

You can rename your external catalogs to provide a descriptive label when the contents or purpose of the catalog has changed.

You can't rename the default catalog.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** and click **Rename**.



3. Make your changes to the catalog name and click **Save**.

Edit an External Catalog Configuration

You can edit the configuration of an external catalogs to update the required password.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** and click **Edit configuration**.
3. Enter the new password for the external catalog and click **Save**.

Access External Catalog Tables in Notebook

You can access tables from external catalogs in your notebooks using three-part names.

Push down is enabled by default and operations like aggregations, projection, filters, column pruning, topN, limit, offset, boolean, logical, and relational operators are pushed down to the source.

1. Navigate to your notebook's code interface.
2. To read data from your external catalog table use:

```
df = spark.read.table("<<catalog_name>>.<<schema_name>>.<<table_name>>")
```

3. To write data to your external catalog, there are four options:

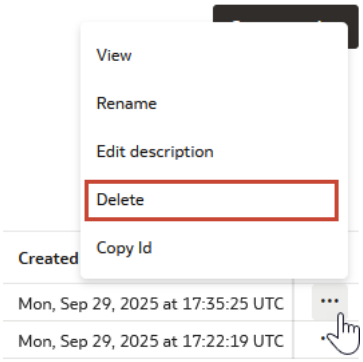
```
Option#1
df.write.saveAsTable("<<catalog_name>>.<<schema_name>>.<<table_name>>")

Option#2
df.write.mode("append").insertInto("<<catalog_name>>.<<schema_name>>.<<tabl
e_name>>")

Option#3
df.write.mode("overwrite").saveAsTable("<<catalog_name>>.<<schema_name>>.<<
table_name>>")

Option#4
df.write.option('write.mode', 'MERGE').option('write.merge.keys','merge
column(s)').insertInto('<<catalog_name>>.<<schema_name>>.<<table_name>>')
```

Delete an External Catalog

You can delete external catalogs to remove all locally stored metadata.

Deleting an external catalog only deletes the locally stored metadata. Data in the data source is not impacted.

1. On the Home page, click **Master catalog**.
2. Next to your catalog, click ··· **Actions** then click **Delete**.



3. Select **Confirm deletion of the catalogs**.
4. Click **Delete**.

Schema

Schema in data catalogs are constructs to organize data.

Catalogs are logical containers for schema, which are also referred to as databases. Schema can contain tables, which contain structured data, and volumes, which contain unstructured data.

A default schema is created in all standard catalogs created in the Master Catalog. To create additional schema, see Create a Schema.

You can manage permissions to control who has access to your schema. For more information, see Schema Permissions.

Master catalog's default catalog, which is called 'default', contains a reserved schema named oci_ai_models. If you have the requisite permissions on OCI Generative AI models in the region, the model is displayed in the oci_ai_models schema and you can drag and drop the models onto a notebook to auto-generate code for batch inference. For more information, see OCI Generative AI (Pretrained Foundation Models).

> **Note**
>
> You can't change the name of a schema in a standard catalog.

Create a Schema

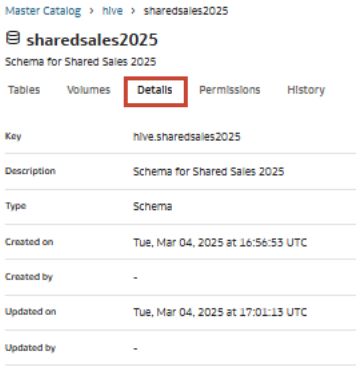
You can create schema in catalogs you own or are shared with you.

1. Click **Create** in the left navigation pane and select **Schema**. You can also navigate to the catalog in which you want to create a schema, click the Schema tab, and click **Create Schema**.
2. Provide a name and description for your schema and click **Create**.

View Schema Details

You can view schema information and resources from the schema Details tab.

1. On the home page, click **Master Catalog**.
2. Navigate to your schema, then click the **Details** tab.

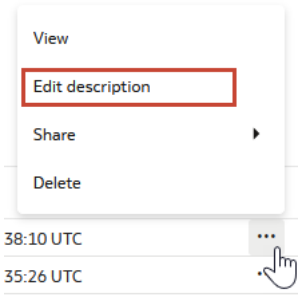


3. Click **Details**.

Edit a Schema Description

You can edit the description for a schema to provide an updated summary of its contents.

1. On the Home page, click **Master Catalog**.
2. Next to the schema you want to change the description for, click **Actions** and click **Edit Description**.



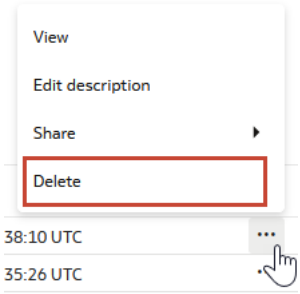
3. Provide a new description. Click **Save**.

Delete a Schema

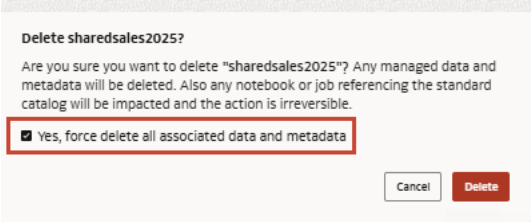
You can delete schema and all their metadata and AI Data Platform-managed data when they are no longer needed.

Deleting a schema also deletes all metadata and AI Data Platform-managed data in managed tables and managed volume associated with that schema. Make sure you backup or move any data you want to retain to another location before deleting a schema.

1. On the Home page, click **Master catalog**.
2. Next to the schema you want to delete, click ··· **Actions** and click **Delete**.



3. Select **Yes, force delete all associated data and metadata**.



4. Click **Delete**.

Tables

Tables define the structure for your data.

You can load new data into your tables or reference data in an existing location. You can define fine-grained access control permissions on tables by creating table permissions.

Tables can either be external or managed.

External tables

An external table defines a structure for data that's stored in a location not managed by Oracle AI Data Platform Workbench. When you create an external table in the AI Data Platform Workbench, the metadata life cycle is managed by AI Data Platform Workbench. When you delete an external table, only the table definition is deleted. The data referenced by the external table isn't deleted.

Ensure your users have the following IAM policies required to create external tables:

```
allow group <GroupName> to read buckets in compartment id <external-data-
CompartmentId>
allow group <GroupName> to inspect objects in compartment id <external-data-
CompartmentId>
```

Additional IAM policies are required for external tables. For more information, see IAM Policies for Oracle AI Data Platform Workbench.

Managed Tables

A managed table defines a structure for data that's stored within the AI Data Platform and can only be accessed by AI Data Platform Workbench users.

When you delete a managed table, the table definition and the table data is deleted.

Supported Table Formats

Format	Description	Usage
Comma-separated-values (CSV)	Data is stored as a text file with a specified row based file format to structure the data. Typically, the first row in the file is a header row that contains columns names for the data.	Used to exchange tabular data between systems. Each row in the file is a row in a table.
JavaScript Object Notation (JSON)	Data is stored in a standard text-based format for representing structured data based on JavaScript object syntax. JSON supports lists of objects or hierarchical structures.	Used in stream applications. JSON simplifies the storage of related data with complex relationships in a single document and avoids chaotic list conversion to a relational data model. Note that JSON isn't splittable.
Avro	Data is stored in a row based binary format while the schema is stored in JSON format to minimize file size and maximize efficiency. Avro has reliable support for schema evolution by managing added, missing, and changed fields. This lets old software to read new data, and new software to read old data. Also known as the data serialization system.	Used for data storage as avro files are splittable and compressible. The serialized row-based storage is ideal for heavy write transaction, such as inserting data into AI Data Platform. Avro is also a good choice when schema evolution is critical during high speed writes.

Format	Description	Usage
Parquet	Data is stored in a columnar data format and is highly compressible and splittable. Parquet is optimized for the paradigm Write Once Read Many (WORM). It writes slowly but reads incredibly quickly, especially when you only access a subset of columns.	Used for solving Big Data problems as compression algorithms work better with columnar data format. You can store Big Data in various formats, such as images, videos, documents, and structured data tables. Parquet is a good choice for heavy workloads when reading portions of data. For example, when the dataset has many columns, but you only want to access a subset of columns. Ideal when you're dependent on Spark or when you want several services to access the same data stored in Object Storage.
Optimized Row Columnar (ORC)	Data is stored in collections of rows in a single file in columnar format.	Used for parallel processing of row collections across a cluster. Ideal when read transactions are more than write transactions or when compression is priority.
Delta	Data is stored in a columnar format that extends Parquet data files with a JSON file-based transaction log for ACID transactions and scalable metadata handling.	Used for transaction support.

Limitations

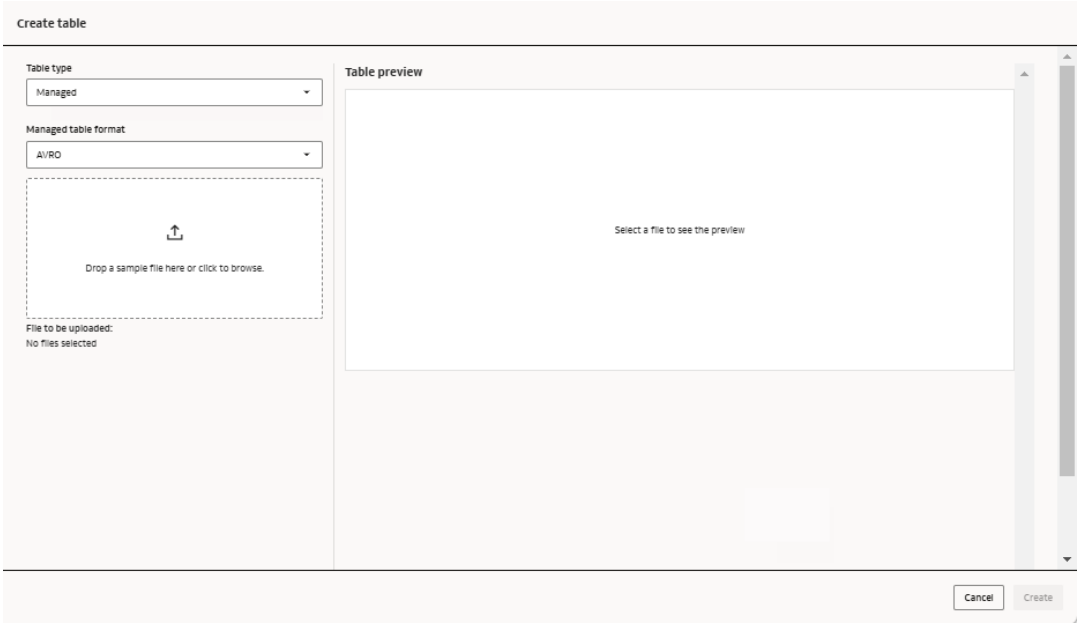
The following limitations apply to tables in Oracle AI Data Platform:

- You cannot define an external table on any data files or directories within/on a volume.
- You cannot define an external table on a bucket and/or its directory that is already used for another external table or external volume
- Views cannot be viewed/listed in the Master Catalog.

Create a Managed Table

You can create tables for schema you manage.

1. Navigate to the schema you want to create a table for.
2. Select the **Tables** tab.
3. Click ⊕ **Create Table**.

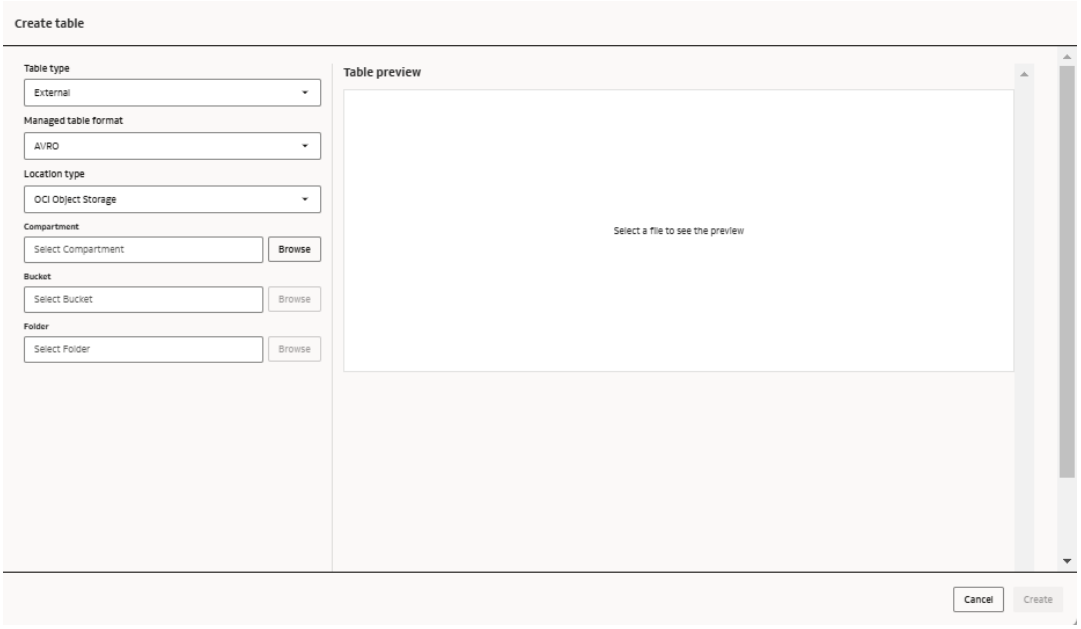


4. Select **Managed** for your **Table Type**.
5. Select the format for your table from **Managed table format**.
6. Either drag and drop a file with your table data or click to browse to the file location.
7. Provide a name and description for your table.
8. Optional: To add partitions, expand **Partition keys (optional)**. Click **Add Partition** and select a data column.
9. Optional: To add table properties to the data catalog's metadata, expand **Table properties (optional)**. Click **Add Property** and provide the property and its value.
10. Click **Create**.

Create an External Table

You can create create an external table with data in OCI Object Storage.

1. Navigate to the schema you want to create a table for.
2. Select the **Tables** tab.
3. Click ⊕ **Create Table**.



4. Select **External** for your **Table Type**.
5. Select the compartment, bucket, and folder from OCI Object Storage where data is stored. The objects you can select are based on the logged in user's IAM permissions.
6. Provide a name and description for your table.
7. Optional: To add table properties to the data catalog's metadata, expand **Table properties (optional)**. Click **Add Property** and provide the property and its value.
8. Click **Create**.

Edit a Table

You can modify details of tables you manage.

1. Navigate to your schema.
2. Select the **Tables** tab.
3. Next to the table you want to edit click ⋯ **Actions**.
   - Click **Rename** to change your table's name. Enter a new name and press `Enter`.
   - Click **Edit Description** to change your table's description. Provide the new description and click **Save**.

View Table Details

You can view the details of tables in schema.

1. Navigate to your schema. Click the **Tables** tab.
2. Click the name of the volume you want to view details for. You can also click ⋯ **Actions** next to the volume then click **View**.
3. Click the **Details** tab.

Delete a Table

You can delete tables from schema you manage.

1. Navigate to the schema you want to delete your table from.
2. Click the **Tables** tab.
3. Next to the table you want to delete, click ⋯ **Actions** and click **Delete**.
4. Click **Delete**.

Volumes

Volumes are containers to store data in its original form and can store semi-structured or unstructured data.

You can load new data into your volume or reference data in an existing location. You can define access control permissions on a volume by creating volume permissions.

Volumes can either be external or managed.

External volumes

An external volume refers to an existing location in OCI Object Storage. When users create an external volume, AI Data Platform Workbench manages only the metadata of the volume, the data life cycle is managed by the customer.

Ensure your users have the following IAM policies required to create external volumes:

```
allow group <GroupName> to read buckets in compartment id <external-data-
CompartmentId>
allow group <GroupName> to inspect objects in compartment id <external-data-
CompartmentId>
```

When you delete an external volume, only the volume definition is deleted. The data referenced by the external volume isn't deleted.

Managed volumes

A managed volume creates a location in OCI Object Storage for storing the data. AI Data Platform Workbench manages both the metadata and data of the volume.

When you delete a managed volume, the volume definition and the volume data is deleted.

Limitations

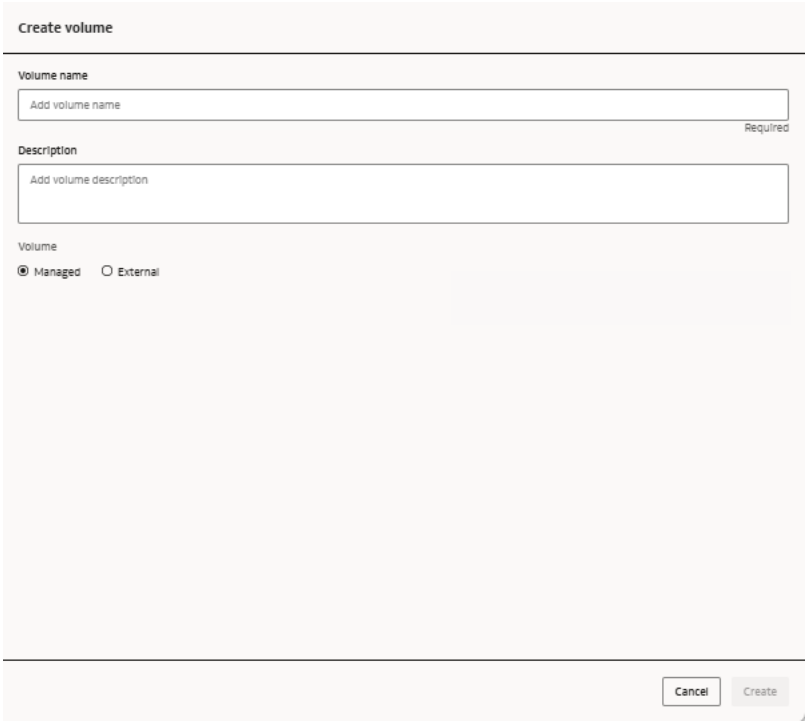
You cannot define an external volume on a bucket and its directory if they are already used for another external table or external volume.

Create a Managed Volume

You can create managed volumes for schema you manage.

1. Navigate to the schema you want to create a volume for.
2. Select the **Volumes** tab.

3. Click ⊕ **Create Volume**.
4. Provide a volume name and description.
5. Select the **Managed** option.

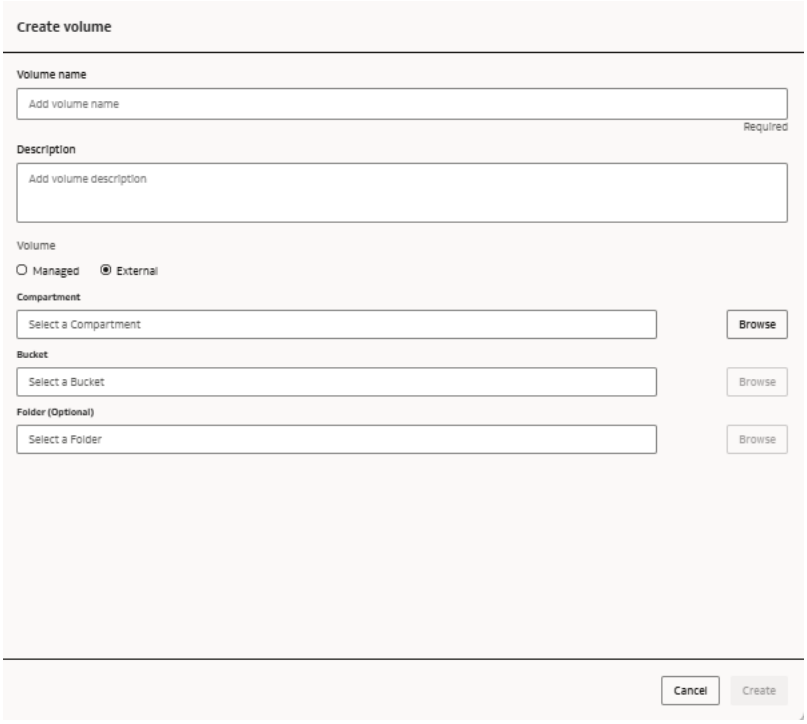


6. Click **Create**.

Create an External Volume

You can create external volumes for schema you manage.

1. Navigate to the schema you want to create a volume for.
2. Select the **Volumes** tab.
3. Click ⊕**Create Volume**.
4. Select the **External** option.



5. Select the compartment, bucket, and folder for your external volume. The objects you can select are based on the logged in user's IAM permissions.
6. Click **Create**.

Edit a Volume

You can modify details of volumes you manage.

1. Navigate to your schema.
2. Select the **Volumes** tab.
3. Next to the volume you want to edit click ··· **Actions**.
   - Click **Rename** to change your volume's name. Enter a new name and press `Enter`.
   - Click **Edit Description** to change your volume's description. Provide the new description and click **Save**.

View Volume Details

You can view the details of volumes in schema.

1. Navigate to your schema. Click the **Volumes** tab.
2. Click the name of the volume you want to view details for. You can also click ··· **Actions** next to the volume then click **View**.
3. Click the **Details** tab.

Delete a Volume

You can delete volumes from schema you manage.

1. Navigate to the schema you want to delete your volume from.
2. Click the **Volumes** tab.
3. Next to the volume you want to delete, click ··· **Actions** and click **Delete**.
4. Click **Delete**.

Cross-Tenancy External Tables and Volumes

Cross-tenancy external tables and volumes allow you to securely access and query data stored in disparate tenancies without the need for complex ETL pipelines or manual data movement.

AI Data Platform Workbench enables users to create cross-tenancy external tables and volumes, a powerful capability designed to eliminate data silos and streamline collaboration.

The benefits of cross-tenancy are:

- **Zero Data Duplication:** You access live data where it resides, saving on storage costs and ensuring "single source of truth" integrity.
- **Simplified Governance:** You manage permissions across boundaries using IAM policies and AI Data Platform Workbench access controls.

Cross-Tenancy Access Requirements

Setting up cross-tenancy access for external tables and volumes requires specific IAM policies configured in a provider tenancy and a consumer tenancy.

In the provider tenancy, you need to create an IAM Dynamic Group in the Oracle Cloud Infrastructure (OCI) console that includes your specific AI Data Platform Workbench resource as a member. For more information, see Managing Dynamic Groups.

After you create the IAM Dynamic Group, you need to configure IAM policies in the provider tenancy:

- Define resources in IAM for consumer tenancy, user group and dynamic groups
- Write **admit** IAM policy for the consumer tenancy resources

```
define tenancy <consumer_tenancy_name1> as <consumer tenancy OCID>
define group <group_name1> as <consumer user group>
define dynamic-group <dynamic_group_name1> as <consumer dynamic group OCID>

admit dynamic-group <dynamic_group_name1> of tenancy <consumer_tenancy_name1>
to manage object-family in tenancy
admit dynamic-group <dynamic_group_name1> of tenancy <consumer_tenancy_name1>
to { OBJECTSTORAGE_NAMESPACE_READ } in tenancy
admit group <group_name1> of tenancy <consumer_tenancy_name1> to manage
object-family in tenancy
```

After configuring the provider tenancy IAM policies, you need to configure your consumer tenancy IAM policies:

- Define the resource in IAM for provider tenancy
- Write **endorse** IAM policy for the local consumer tenancy resources

```
define tenancy <provider_tenancy_name1> as <provider tenancy OCID>
```

```
endorse dynamic-group <dynamic_group_name> to manage object-family in tenancy
<provider_tenancy_name1>
endorse dynamic-group <dynamic_group_name> to
{ OBJECTSTORAGE_NAMESPACE_READ } in tenancy <provider_tenancy_name1>
endorse group <group_name> to manage object-family in tenancy
<provider_tenancy_name1>
```

Once both provider and consumer tenancy IAM policies are configured, you can create cross-tenancy external tables and volumes using SQL grammar. For more information, see SQL Grammar.

5

Work with Files

You can store files in volumes in AI Data Platform Workbench and users can organize these files in folders within a volume. AI Data Platform Workbench provides you multiple ways to access data stored in volumes and workspaces.

AI Data Platform Workbench supports multiple methods for accessing data stored in volumes:

- **POSIX-style paths:** Allow users to provide access to data relative to the driver root (/). Users can read/write data to volumes or workspace folders.
- **URI-style paths:** Allow users to provide access to data using a URI scheme. For example, if you want to read data in OCI Object Storage, you should provide a valid URI scheme to read/write that data.

Here are some examples:

Source	Access Pattern	Example

Volume	POSIX	**Example 1**
		
		`df_csv = spark.read.csv("/Volumes/ <<catalog_name>>/ <<schema_name>>/ <<volume_name>>/ <<file_name>>.csv", header=True, inferSchema=True, sep=",")`
		
		Example 2
		
		`import pandas as pd df_panda_csv=pd.read_csv("/Volumes/ <<catalog_name>>/ <<schema_name>>/ <<volume_name>>/ <<file_name>>.csv", header=0, sep=",")`
		
		Example 3
		
		`import os os.listdir("/Volumes/ <<catalog_name>>/ <<schema_name>>/ <<volume_name>>/")`

	URI	`df = spark.read.format("csv") .option("header",True).load("file:///Volumes// <<catalog_name>>/ <<schema_name>/ <<volume_name>>/ <<folder_path>>/ <<file_name>>.csv")df.show()`

Source	Access Pattern	Example

Workspace	POSIX	**Example 1**
		
		`df_csv = spark.read.csv("/Workspace/<<folder_path>>/<<file_name>>.csv", header=True, inferSchema=True, sep=",") df_csv.show()`
		
		Example 2
		
		`import pandas as pd df_panda_csv=pd.read_csv("/Workspace/<<folder_path>>/<<file_name>>.csv", header=0, sep=",") df_panda_csv.head()`
		
		Example 3
		
		`import osos.listdir("/Workspace/<<folder_path>>/")`

	URI	`spark.read.format("json").load("file:///Workspace/<<folder_path>>/<<file_name>>.json").show()`
OCI Object Storage	URI	`df_csv = spark.read.csv("oci://<<bucket_name>>@<<namespace>>/<<folder/file>>", header=True, inferSchema=True, sep=",")`

6

Auto Populate Catalog

This chapter contains information about creating and managing automated extractors to pull data into your catalogs.

Topics:

About Auto Populate

You can automate the process of extracting metadata from sources directly to your data catalogs.

Manually creating schema, tables, and partitions from your data sources is time consuming and complicated. Oracle AI Data Platform Workbench offers the ability to automatically extract metadata from data sources and create entities in catalogs that you specify in the metadata extractor.

You automatically populate this metadata in your catalog by creating a metadata extractor. As part of creating the extractor, you specify the target catalog to extract metadata to and the source for the metadata. You can choose to have the extractor create tables in a specified schema, or let the system suggest where the tables are created if no schema is specified or detected.

Auto populate can extract metadata from the following file types:

- CSV
- JSON
- Avro
- ORC
- Parquet
- Delta Lake

> **Note**
>
> When you specify the source folder when creating a metadata extractor, all the files in the leaf folder must be the same data format.

> **Note**
>
> Auto-populate only supports underscores (_) as special characters in column names.

You can opt to either manually review entities that are extracted or let the system automatically create the entities from the extracted metadata. When extracting metadata, entities that cause errors are captured in the log. You can view the log to see which entities encountered errors and take action to correct.

Manually reviewing entities allows you to accept or reject entities on an individual basis. You can view entities are already approved or rejected in the Reviewed Entities tab.

Extractors display their status to let you know what stage they are currently at and if user intervention is required.

Extractor Status	Description
Not Started	The extractor has not started. Start the extractor to begin.
Running	Extractor is in progress
Ready for review	The extractor has run and you have chosen manual approval. Extracted entities must be reviewed and either accepted or approved.
Reviewing	The extractor has run and you have chosen manual approval. Some entities have been reviewed or approved by a user, but entities remain that require review.
Completed	The extractor has run and entities have either been approved automatically or manually approved by a user

You can view and use metadata extractors created by other users if you have the requisite permissions.

Create Metadata Extractor

You can create metadata extractors to automate extracting entities like schema and tables to your catalogs.

1. On the Home page, click **Auto populate catalog**.
2. Click ⊕ **Create Metadata Extractor**.
3. Enter a name for the metadata extractor.
4. Select the target catalog from the **Catalog** dropdown.
5. Select the appropriate source type from **Source Type** dropdown.
6. Next to **Compute**, click **Browse** and choose the cluster the extractor should use. Click **Select**.
7. For **Object Storage URI**, select whether to browse your compartments for the bucket or folder you want to extract metadata to or if you want to specify the URI for the bucket or folder.
   - For **Select bucket or folder**, click **Browse** to select the compartment, bucket, and folder you want metadata extracted to.

- For **Enter URI manually**, enter the URI in the field provided.

8. Select whether entities are created with manual approval or automatically approved by the system.
9. Optional: Select the schema where external tables are created. If no schema specified, the system creates tables in schema based on folder structure, or in the default schema if no schema is detected.

Manually Review Extracted Metadata Entities

When you choose the manual method of creating entities in a metadata extractor, you need to review the extracted entities and approve or reject adding them to your catalog.

1. On the Home page, click **Auto populate catalog**.
2. Click the name of the metadata extractor.
3. Click the **Entities awaiting review** tab.
4. For each entity, select **Approve** or **Reject**.
5. Optional: Select **Approval All** or **Reject All** to set all entities under review to the selected status.
6. Click **Submit**.

View Reviewed Entities

You can see entities that have been manually or automatically reviewed as part of metadata extraction and see log details, table details, or column schema for that entity.

1. On the Home page, click **Auto populate catalog**.
2. Click on the name of the metadata extractor.
3. Click the **Reviewed entities** tab.
4. Next to an entity, click ··· **Actions**.
   - Click **View table details** to see the table details for the selected entity.
   - Click **View column schema** to see the column schema for the selected entity.
   - Click **View logs** to see the metadata extractor logs for the selected entity.

View Metadata Extractor Details

You can view the details of a metadata extractor to see its status, metadata creation method, base location, and creation details.

1. On the Home page, click **Auto populate catalog**.
2. Click the name of the metadata extractor.
3. Click the **Details** tab.

Delete Metadata Extractor

You can delete metadata extractors that are no longer needed.

1. On the Home page, click **Auto populate catalog**.

2. Next to the metadata extractor you want to delete, click ··· **Actions** and click **Delete**
3. Click **Delete**.

7

Share Data

This chapter provides details on managing data shares, assets, and recipients in your AI Data Platform Workbench.

Topics:

Data Sharing

Data Sharing in Oracle AI Data Platform Workbench enables secure and efficient real-time data sharing across and within the organization without data duplication.

AI Data Platform Workbench uses Delta Sharing protocol to securely share data. To share your data, you must complete these steps:

1. Create a share.
2. Add assets that you want to share. An asset can be a schema or table.
3. Create and add recipients to a share. A share and recipient can be access controlled using permissions model.
4. Activate the recipients.

From the Data Sharing page you can see all Shares in your AI Data Platform Workbench, all recipients of Shares in your AI Data Platform Workbench, your Shares, and Shares others have made you a recipient for.

From the **Shares** tab, you can see the details of Shares in your AI Data Platform Workbench, including the number of assets in a Share, number of recipients, and who owns the share. You can create a new Share by clicking **Create share** next to the search bar.

Clicking on a Share lets you see the shared assets, recipients, details, and RBAC permissions. From the **Assets** tab you can add assets to the share. From the Recipients tab you can manage the recipients for this specific share, create new recipients, or share with existing recipients. The Details tab provides the name, description and creation details of the Share. From the Permissions tab you can manage the RBAC permissions for that Share.

From the **Recipients** tab at the top level of Data Sharing, you can see details of the Share recipients in your AI Data Platform Workbench. You can create new recipients by clicking **Create recipient** next to the search bar.

Clicking on a recipients name shows Shares the recipient is added to, details for that recipient, and the RBAC permissions for that recipient.

Handling Token Expiry

Oracle AI Data Platform can regenerate bearer tokens for data sharing for active recipients. If the current bearer token has not expired, you can extend the duration of the bearer token by calling the `updateSecretExpiry` API using a curl command:

```
curl
    --header 'Authorization: Bearer <token>’
    --header 'Content-Type: application/json'
    --data '{"existingTokenExpireInSeconds":"<duration>"}'
    --request POST ‘<delta_share_endpoint>/updateSecretExpiry'
```

Where:

- `<token>` is the bearer token value in the share profile
- `<duration>` is the duration of the new token in seconds
- `<delta_share_endpoint>` is the Delta Share endpoint. You can get this value for a recipient by using `sql("DESCRIBE recipient <<recipient_name>>")` and checking the value of the recipient's `delta_share_enpoint`.

Limitations

- You can send the activation token email using an email application or by copying the activation link and composing an email using their existing email service.
- You cannot share volumes or workspace files (data, notebook, sql or python files) using data sharing.

Create a Share

You must create a Share as a container for managing shared assets as part of AI Data Platform Workbench data sharing.

1. On the Home page, click **Data Sharing**.
2. Click ⊕ **Create Share**.
3. Enter a name and description for your Share.
4. Click **Create**.

Modify a Share

You can update the name and description of your Shares after creation.

1. On the Home page, click **Data Sharing**.
2. Next to the share to modify, click ··· **Actions** then click **Edit**.
3. Enter a new name or description for your Share. Click **Save**.

Delete a Share

You can delete a Share that is no longer needed.

1. On the Home page, click **Data Sharing**.
2. Next to the share to delete, click ··· **Actions** then click **Delete**.
3. Click **Delete**.

View Share Details

1. On the Home page, click **Data Sharing**.
2. Click to the **Shares** tab.
3. Click the share name and click the **Details** tab.

Add an Asset to a Share

After creating a Share, you can add assets to be shared with recipients.

1. On the Home page, click **Data Sharing**.
2. Click the name of the Share to add assets to.
3. On the Assets tab, click ⊕ **Add asset**.
4. Select **Workspace** or **Catalog** as the asset source.
5. Use the search bar or navigate to the asset you want to share. Select it and click **Add**.

Remove an Asset from a Share

You can remove assets you no longer want to be a part of a Share.

1. On the Home page, click **Data Sharing**.
2. Click the name of the Share to remove assets from.
3. On the **Assets** tab, next to asset you want to remove click ··· **Actions** then click **Delete**.
4. Click **Delete**.

Add Recipients to Data Sharing

You must add recipients before you can include them in Shares.

1. On the Home page, click **Data Sharing**.

2. Click the **Recipients** tab.
3. Click **Create recipient**.
4. Enter a name and description for the recipient.
5. Optional: Enter the email address to send an activation link to for the recipient.
6. Click **Create**.

Add an Existing Recipient to a Share

You can add recipients that already exist in your AI Data Platform Workbench to a Share.

1. On the Home page, click **Data Sharing**.
2. Click the name of a Share to add an existing recipient to.
3. Click the **Recipients** tab.
4. Click **Share with existing recipients**.
5. Search for a user and select the box next to the recipient name.
6. Click **Share**.

Manually Activate a Recipient

You can choose to manually activate recipients if there are issues preventing them from using the email activation links.

1. On the Home page, click **Data Sharing**.
2. Click to the **Recipient** tab.
3. Click the recipient name and click the **Details** tab.
4. Next to **Activation status** click **Activate**.
5. Copy the activation link from the prompt and paste into a separate window.
6. Verify the Activation status is now set to **Active**.

Resend Activation Link to a Recipient

If the activation link was lost, deleted, or otherwise not received by a recipient, you can resend the activation link to the provided email address.

The recipient must have an email address as part of their details to send them an activation link. If the email is incorrect or missing, you may need to create a new recipient.

1. On the Home page, click **Data Sharing**.
2. Click to the **Recipient** tab.
3. Click the recipient name and click the **Details** tab.
4. Next to **Activation email link** click **Send activation link**. Your configured email application opens.
5. Review and send the email to the recipient.

Remove Recipient from a Share

You can remove a recipient from a Share, keeping the recipient but removing their access to assets in that Share.

1. On the Home page, click **Data Sharing**.
2. Click the share you want to remove recipients from.
3. Click the **Recipients** tab.
4. Next to the recipient you want to remove, click ··· **Actions** then click **Remove**.

Modify a Recipient

You can update the name, description, and email address for a recipient after creation.

1. On the Home page, click **Data Sharing**.
2. Click to the **Recipient** tab.
3. Next to the recipient to update, click ··· **Actions** then click **Edit**.
4. Enter a new name or description for your recipient. Click **Save**.

View Recipient Details

You can see information on a recipient, such as name, email address for activation link, active status, and creation details from the Details tab.

1. On the Home page, click **Data Sharing**.
2. Click to the **Recipient** tab.
3. Click the recipient name and click the **Details** tab.

Delete a Recipient

You can delete recipients that are no longer needed in your AI Data Platform Workbench.

1. On the Home page, click **Data Sharing**.
2. Click the **Recipients** tab.
3. Next to the recipient to delete, click ··· **Actions** then click **Delete**.
4. Click **Delete**.

Part III
Data Governance

This section covers the ways you can manage the users and roles that can see and develop your data.

Permission Inheritance

The permission model in Oracle AI Data Platform is designed to support permission inheritance so that permissions granted at the parent level automatically flow to the child objects.

For example, User A is granted SELECT permissions for Catalog A, and Catalog A has the following hiearchy:

```
|CatalogA
|-------schema1
|------------table1
```

In this example, User A has SELECT permissions for Catalog A, schema1, and table1.

Permission Expansion

Permissions granted at the child level do no require explicit permissions at the parent object level. For example, you manage a catalog with the following hiearchy:

```
|CatalogA
|-------schema1
|------------table1_1
|-------schema2
|-------------table2_1
|-------------table2_2
```

In this example, if you grant User A SELECT permissions for table2_2, User A only sees the following hiearchy:

```
|CatalogA
|-------schema2
|-------------table2_2
```

User A is only granted limited list permissions for Catalog A and schema2. They can see objects that contain table2_2, but do not have access to do anything more to them.

Limitations

Known limitations of permissions inheritance and expansion are:

- Permissions inheritance does not show up for ADMIN permissions for any object type
- Permissions inheritance does not work for volumes
- Permissions expansion does not work for volumes and workspace files and folders

Topics:

- Permissions Model
- Audit Log
- Roles

8

Permissions Model

This chapter describes the permissions model Oracle AI Data Platform uses to manage access.

Topics:

About Permissions

Oracle AI Data Platform Workbench permissions follow a similar model for all objects that use them.

You can manage permissions for each object from its **Permissions** tab.

AI Data Platform Workbench has two layers of security - access to OCI resources using IAM policies and access to Data Platform objects. Users must have access to OCI resources first before granting them access to AI Data Platform Workbench objects. Users of AI Data Platform Workbench require access to navigate to resources in OCI console and IAM permissions to list compartments and buckets. To access an AI Data Platform Workbench instance, you require at least USE IAM policy permissions. These IAM policies are needed even if you have AI_DATA_PLATFORM_ADMIN role on an AI Data Platform Workbench instance.

Permissions in AI Data Platform Workbench follow a hierarchy where permissions granted for a parent object or space grant permissions to contained objects and spaces.

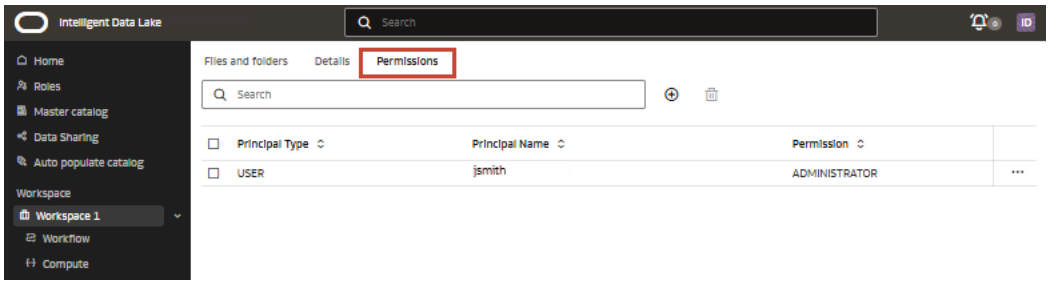
Permission to Create Workspaces

Permissions to create workspaces are included in the AI_DATA_PLATFORM_ADMIN role by default. If you want users other than the administrator to be able to create workspaces, you need to provide CREATE_WORKSPACE permissions to that user. You can assign CREATE_WORKSPACE to a user from the Workspace Listing screen.

Workspace Permissions

You can set role-based action controls for a workspace you own that apply to all its contents.

Workspace permissions are managed from the Permissions tab, located at the top of your workspace home page.



A user can be granted the following permissions:

- USER: You can create folders/files in root, and have MANAGE permissions on the Shared Folder.
- PRIVILEGED_USER: You have USER permissions and can also create compute.
- ADMINISTRATOR: You have ADMIN permissions on all workspace objects and can update or delete a workspace and manage permissions.

> **Note**
>
> The USER permission for workspaces also grants users the MANAGE permission on all objects in the Shared Folder. A Shared Folder cannot be deleted, renamed, or moved.

Permissions can be granted to users, groups, or roles. You can either select users from a list of AI Data Platform Workbench users or add a used or role by the OCID.

Create Workspace Permissions

You can grant access to your workspaces to users, roles, or groups.

You must have administrator privileges in the workspace to grant access to others.

1. On the Home page, click **Workspace**.
2. Next to your workspace, click ··· **Actions** then click **Permissions**.
3. Click ⊕**New Permission**.
4. Select the permissions level and principal type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click **Search** and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Workspace Permissions

You can change permission settings for any workspace where you have administrator privileges.

1. On the Home page, click **Workspace**.
2. Next to your workspace, click ··· **Actions** then click **Permissions**.
3. Next to the permission, click **Actions** and click **Edit**.
4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Workspace Permissions

You can delete a workspace permission to remove access and actions for all contained users.

1. On the Home page, click **Workspace**.
2. Next to your workspace, click ··· **Actions** then click **Permissions**.
3. Next to your permission, click **Actions** and click **Delete**.
4. On the confirmation window, click **Delete**.

Workspace Folder Permissions

You can manage which users, roles, and groups can view and modify files and folders in your workspaces.

Workspace folder permissions grant the following actions:

- **READ:** Users can read/list files and folders.
- **USE:** Users can read/write to folders and contained files, and run permitted job types (.ipynb, .py, .sql, .scala, etc).
- **MANAGE:** Users have Read and Use permissions and can rename files/folders and modify files.
- **ADMIN:** Users all permissions and can create, modify, or delete other user permissions.

An admin can grant permission to any principal who has at least a workspace USER permissions.

Operation	READ	USE	MANAGE	ADMIN
List	Yes	Yes	Yes	Yes
View object	Yes	Yes	Yes	Yes
Create folder	No	Yes	Yes	Yes
Create file	No	Yes	Yes	Yes
Rename folder	No	No	Yes	Yes
Move folder	No	No	Yes	Yes
Delete folder	No	No	No	Yes
Manage user permissions	No	No	No	Yes

Permissions granted on a folder in the workspace will cascade to the child objects by default, but you have the option to turn off cascading. If a user has ADMIN permissions on a folder, that

user will have ADMIN permissions on all child files and folders. When an ADMIN grants permission on a folder with cascade, all child resources current and future have same permission. When an ADMIN grants permission on a folder without cascade, permission is applicable only on the current object.

For example, given the following folder structure, an ADMIN wants to grant USE access to USER2 for just `File1.csv`:

- WORKSPACE1
  - F1
    - F2
      - File1.csv
    - F3
      - File2.csv

Assuming USER2 is already a user in the workspace, the ADMIN grants USE permission to USER2 on `File1.csv`. For the user to be able to work with `File1.csv` the ADMIN must also explicitly grant READ permission on folders `F1` and `F2`, without cascading. As a result, the user sees this folder structure:

- WORKSPACE1
  - F1
    - F2
      - File1.csv

Create File and Folder Permissions

You can set individual permissions for files and folders in your workspaces.

1. Navigate to the file or folder you want to set permissions for.
2. Click ··· **Actions** and click **Permissions**.
3. Click ⊕ **Create Permission**.
4. Select a permission level, principal type, and the user from the dropdown menus.
5. Click **Save**.

Modify File and Folder Permissions

You can modify existing permissions for files or folders in your workspace.

1. Navigate to the file or folder you want to set permissions for.
2. Click ··· **Actions** and click **Permissions**.
3. Next to permission you want to modify, click ··· **Actions** and click **Edit**.
4. Change the permissions details as needed and click **Save**.

Delete File and Folder Permissions

You can delete permissions for files and folders in your workspace.

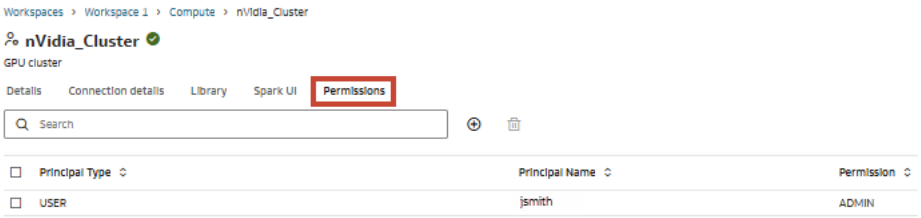
1. Navigate to the file or folder you want to set permissions for.

2. Click ··· **Actions** and click **Permissions**.
3. Next to the permission you want to delete, click ··· **Actions** and click **Delete**.
4. Click **Delete**.

Compute Cluster Permissions

You can control which users and roles have view, read, and administrator access to your compute clusters.

You create and manage user permissions from the **Permissions** tab in your cluster.



As an administrator, you can grant permissions to any principal who has at least User workspace permissions.

Operation	Read	Use	Admin
List cluster	Yes	Yes	Yes
Attach cluster to notebook/job	Yes	Yes	Yes
View driver logs, Spark UI	Yes	Yes	Yes
View cluster metrics	Yes	Yes	Yes
Start/Restart cluster	No	Yes	Yes
Terminate cluster	No	Yes	Yes
Edit cluster	No	No	Yes
Attach/Upload library to cluster	No	No	Yes
Grant/Revoke permissions	No	No	Yes

Create Cluster Permissions

You can control which users and roles can see and modify your clusters.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Permissions** tab.
3. Click ⊕ **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.

6. Click **Create**.

Modify Cluster Permissions

You can modify permissions for users and roles assigned to your cluster.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Permissions** tab.
3. Next to the user or role you want to modify, click ⋯ **Actions** then click **Edit**.
4. Select a new permission level from the dropdown. Click **Save**.

Delete Cluster Permissions

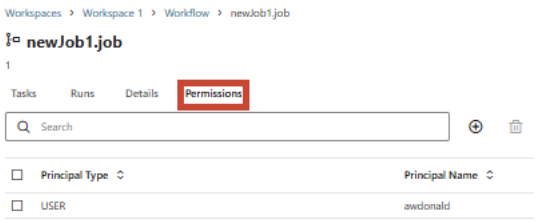
You can remove permissions that are no longer needed for users or roles on your cluster.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Permissions** tab.
3. Next to the user or role you want to delete, click ⋯ **Actions** then click **Delete**.
4. Click **Delete**.

Job Permissions

Job permissions control which users and roles have access to your jobs.

You manage the users and roles that have access to your job from the Permissions tab in your job.



The following permission levels are available to job users:

- Read
- Use
- Manage
- Admin

Each permission level has access to a different set of operations, outlined below.

Operation	Read	Use	Manage	Admin
List	Y	Y	Y	Y
View details	Y	Y	Y	Y
Execution status	Y	Y	Y	Y

Operation	Read	Use	Manage	Admin
Attach/Detach compute	N	Y	Y	Y
Run	N	Y	Y	Y
View task log	N	Y	Y	Y
Rename job	N	N	Y	Y
Edit job	N	N	Y	Y
Terminate workflow	N	N	Y	Y
Move file	N	N	N	Y
Delete job	N	N	N	Y
Grant/Revoke permissions	N	N	N	Y

Create Job Permissions

You can create permissions to control which users and roles have access to your jobs.

You can only grant access to jobs that you own.

1. Navigate to the job you want to grant access to.
2. Click **Permissions**.
3. Click ⊕ **New Permissions**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Job Permissions

You can grant or revoke permissions by changing the permission levels for existing users or roles.

1. Navigate to your workspace and click **Workflow**.
2. Click your job, then click the **Permissions** tab.
3. Next to the user or role you want to modify, click ··· **Actions** then click **Edit**.
4. Select a new permission level from the dropdown. Click **Save**.

Delete Job Permissions

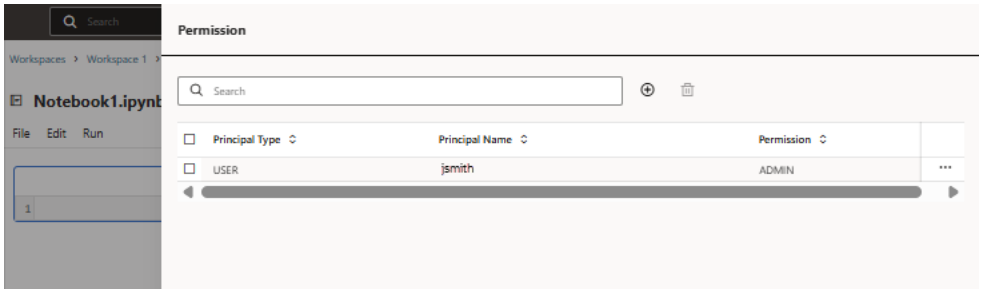
You can remove permissions that are no longer needed for users or roles in your job.

1. Navigate to your workspace and click **Workflow**.
2. Click your job, then click the **Permissions** tab.
3. Next to the user or role you want to delete, click ··· **Actions** then click **Delete**.
4. Click **Delete**.

Notebook Permissions

Notebooks permissions determine which users, roles, and groups can view and modify your notebook.

Permissions for a notebook are viewed by clicking **Actions** on the top right of your notebook, and clicking **Permissions**. From the **Permission** page, you can view, create, modify, or delete permissions for your notebook. You can filter the displayed list of users by entering a user in the **Search** bar.



The following permission levels are available to notebook users:

- Read
- Use
- Manage
- Admin

Each permission level has access to a different set of operations, outlined below.

Operation	Read	Use	Manage	Admin
List	Y	Y	Y	Y
View details	Y	Y	Y	Y
Execution status	Y	Y	Y	Y
Attach/Detach compute	N	Y	Y	Y
Run workflow	N	Y	Y	Y
View log	N	Y	Y	Y
Rename notebook	N	N	Y	Y
Edit notebook	N	N	Y	Y
Terminate workflow	N	N	Y	Y
Move file	N	N	N	Y
Delete notebook	N	N	N	Y
Grant/Revoke permissions	N	N	N	Y

Create Notebook Permissions

You can set individual permissions for notebooks you own.

1. Navigate to the notebook you want to set permissions for.

2. Click **Actions** and click **Permissions**.
3. Click ⊕ **Create Permission**.
4. Select a permission level, principal type, and the user from the dropdown menus.
5. Click **Save**.

Modify Notebook Permissions

You can modify existing permissions for notebooks you own.

1. Navigate to the notebook you want to set permissions for.
2. Click **Actions** and click **Permissions**.
3. Next to permission you want to modify, click **Actions** and click **Edit**.
4. Change the permissions details as needed and click **Save**.

Delete Notebook Permissions

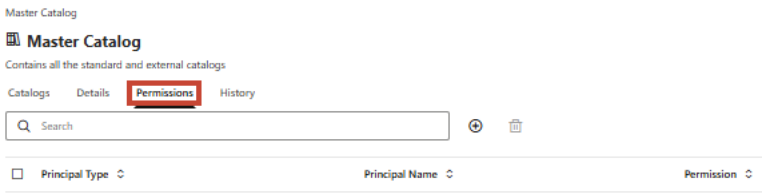
You can delete permissions for notebooks you administer.

1. Navigate to the notebook you want to set permissions for.
2. Click **Actions** and click **Permissions**.
3. Next to the permission you want to delete, click **Actions** and click **Delete**.
4. Click **Delete**.

Master Catalog Permissions

Permissions at the master catalog level determine who can create new standard and external catalogs and grant permissions to others.

You manage permissions for the Master catalog from the **Permissions** tab.



The user that creates your AI Data Platform Workbench is automatically granted ADMIN permissions for the Master catalog. There are two permission levels for Master catalog:

- **CREATE_CATALOG:** User can create standard and external catalogs.
- **ADMIN:** User can view all catalogs, create, edit, or delete catalogs and their child objects, and grant or revoke permissions.

Master Catalog Permission Inheritance

ADMIN permissions for the Master Catalog confer ADMIN permissions on all child objects in the Master Catalog. When a user with CREATE_CATALOG permissions creates a catalog,

they are automatically given ADMIN permission for the newly created catalog and all its child objects.

Create Master Catalog Permissions

You can set permissions to manage who can create, edit, and delete catalogs and grant permissions to others.

1. On the Home page, click **Master Catalog**.
2. Click the **Permissions** tab.
3. Click **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Master Catalog Permissions

You can modify the permissions of users or roles for the Master catalog.

1. On the Home page, click **Master Catalog**.
2. Click the **Permissions** tab.
3. Next to the permission, click **Actions** and click **Edit**.
4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Master Catalog Permissions

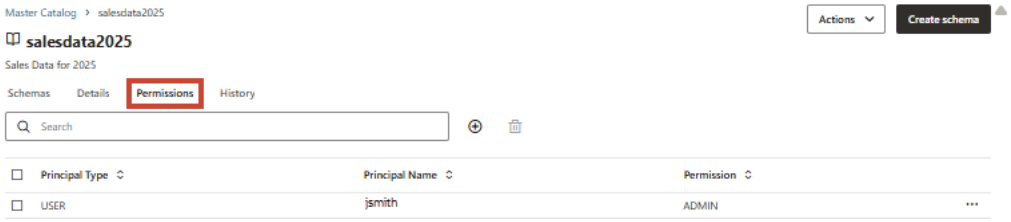
You can delete catalog permissions to remove access and actions for all contained users or roles.

1. On the home page, click **Master Catalog**.
2. Click the **Permissions** tab.
3. Next to your permission, click **Actions** and click **Delete**.
4. Click **Delete**.

Standard Catalog Permissions

You can manage permissions for standard catalogs to determine which users, roles, and groups can view and modify your catalogs.

You can set permissions for standard catalogs from the **Permissions** tab of your catalog. You can filter the list of users and roles that have access to your catalog by entering a name in the **Search** bar.



Permissions set at the catalog level cascade down to any children of the catalog. Permissions set at the schema level apply to any child objects of the schema.

- **SELECT:** Users can read/list catalogs, schema, and volumes. Users can run select queries on views and tables.
- **MANAGE:** Users have all Select permissions at the Standard catalog level and can alter schema, tables, and views and write to volumes. Users can also insert, update, and delete data in tables.
- **CREATE_SCHEMA:** Users have all Manage permissions at the Standard catalog level and can create new schema in the catalog.
- **ADMIN**: Users have all Create_Schema permissions at the Standard catalog level and can delete schema, as well as manage other user permissions

Operation	SELECT	MANAGE	CREATE_SCHEMA	ADMIN
Read/List	Yes	Yes	Yes	Yes
Run queries	Yes	Yes	Yes	Yes
Edit schema/tables/ volumes/views	No	Yes	Yes	Yes
Create schema	No	No	Yes	Yes
Delete schema	No	No	No	Yes
Manage permissions	No	No	No	Yes

Master Catalog Permission Inheritance

Users with CREATE_CATALOG or ADMIN permissions at the Master catalog level are treated as having the following permissions in standard catalogs:

- SELECT
- MANAGE
- CREATE_SCHEMA
- ADMIN

External Catalog Permissions

You can manage permissions for external catalogs to determine which users, roles, and groups can view and modify your catalogs.

Users with ADMIN permissions for an external catalog can grant permissions to:

- Any IAM user principal or IAM group. Users are loaded in the following order:

1. All users from the selected domain who have opened an AI Data Platform instance at least once
2. All remaining users in the selected domain, in alphabetical order

- Roles the ADMIN user can view.

External catalog permissions grant the following actions:

Operation	MANAGE	ADMIN
Read/List & Perform DML operations * DDL **(Coming soon)**	Yes	Yes
Edit catalog name	No	Yes
Edit catalog properties (password, etc.)	No	Yes
Drop catalog	No	Yes
Manage permissions	No	Yes
* External catalog permissions are limited to the permissions of the user used to connect to the external source. If the user of the external source used to create the external catalog has read-only permission, MANAGE permission of the external catalog is also limited to read-only permission.		

Master Catalog Permission Inheritance

Users with CREATE_CATALOG or ADMIN permissions at the Master catalog level are treated as having the following permissions in external catalogs:

- MANAGE
- ADMIN

Create Catalog Permissions

You can grant permissions to view and modify catalogs, schema, tables, and volumes.

1. On the Home page, click **Master Catalog**.
2. Navigate to the catalog you want to create a new permission for and click the **Permissions** tab.
3. Click ⊕ **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Catalog Permissions

You can modify the permissions of users or roles for catalogs you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your catalog, then click the **Permissions** tab.
3. Next to the permission, click ··· **Actions** and click **Edit**.

4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Catalog Permissions

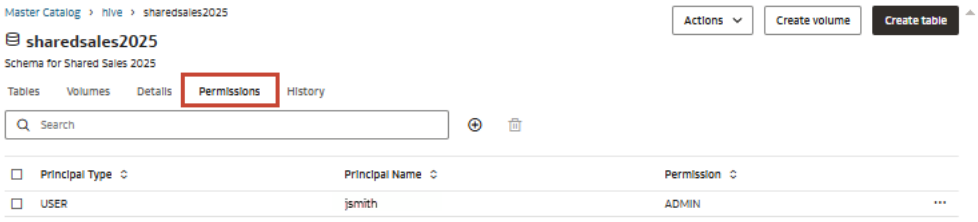
You can delete catalog permissions to remove access and actions for all contained users or roles.

1. On the home page, click **Master Catalog**.
2. Navigate to your catalog, then click the **Permissions** tab.
3. Next to your permission, click ··· **Actions** and click **Delete**.
4. Click **Delete**.

Schema Permissions

Schema permissions determine which users, roles, and groups can view and modify your schema and their child objects.

You control the users and roles that can access your schema from the schema **Permissions** tab.



Permissions set at the schema level apply to any child objects of the schema.

Schema permissions grant the following actions:

- **SELECT:** Users can read/list tables, view, and volumes in the schema. Users can run select queries on views and tables.
- **WRITE:** Users have Select permissions and can alter tables or data in tables, write to volumes, and alter views.
- **CREATE_MODEL:**: Users can create models in a schema.
- **CREATE_TABLE:**: Users can create tables in a schema.
- **CREATE_VIEW:** Users can create views in a schema.
- **CREATE_VOLUME:** Users can create volumes in a schema.
- **ADMIN:** Users have Select, Write, and all Create permissions and can create, modify, or delete other user permissions.

Operation	SELECT	WRITE	CREATE_ MODEL	CREATE_ TABLE	CREATE_ VIEW	CREATE_ VOLUME	ADMIN
Read/List	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Operation	SELECT	WRITE	CREATE_ MODEL	CREATE_ TABLE	CREATE_ VIEW	CREATE_ VOLUME	ADMIN
Run queries/ Read volumes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Edit tables/ volumes/ views	No	Yes	Yes	Yes	Yes	Yes	Yes
Create model	No	No	Yes	No	No	No	Yes
Create table	No	No	No	Yes	No	No	Yes
Create view	No	No	No	No	Yes	No	Yes
Create volume	No	No	No	No	No	Yes	Yes
Delete schema	No	No	No	No	No	No	Yes
Manage permissions	No	No	No	No	No	No	Yes

Schema Permission Inheritance

Schema Permission	Catalog Level Permission		
SELECT	SELECT	MANAGE	ADMIN
WRITE	X		
CREATE_VIEW	X	X	
CREATE_VOLUME	X	X	
CREATE_TABLE	X	X	
ADMIN	X	X	

Create Schema Permissions

You can control which users and roles have access to schema you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your schema, then click the **Permissions** tab.
3. Click **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Schema Permissions

You can modify the permissions of users or roles for schema you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your schema, then click the **Permissions** tab.
3. Next to the permission, click ··· **Actions** and click **Edit**.
4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Schema Permissions

You can delete schema permissions to remove access and actions for all contained users or roles.

1. On the home page, click **Master Catalog**.
2. Navigate to your catalog, then click the **Permissions** tab.
3. Next to your permission, click ··· **Actions** and click **Delete**.
4. Click **Delete**.

Table Permissions

Table permissions determine which users, roles, and groups can view and modify your tables.

Table permissions grant the following actions:

- **SELECT:** Users can read/list tables. Users can run select queries on tables.
- **INSERT:** Users can read/list tables and write to tables.
- **UPDATE:** Users can read/list tables and can run updates on table data.
- **DELETE:** Users can read/list tables and can delete data from the table.
- **ALTER:** Users can read/list tables and can modify table names or descriptions.
- **ADMIN:** Users have all permissions and can create, modify, or delete other user permissions.

Operation	SELECT	INSERT	UPDATE	DELETE	ALTER	ADMIN
List table	Yes	Yes	Yes	Yes	Yes	Yes
Read table data	Yes	No	No	No	No	Yes
Write data to table	No	Yes	No	No	No	Yes
Update data in table	No	No	Yes	No	No	Yes
Delete data from table	No	No	No	Yes	No	Yes
Alter table metadata	No	No	No	No	Yes	Yes
Delete table	No	No	No	No	No	Yes
Manage user permissions	No	No	No	No	No	Yes

Table Permission Inheritance

Table Permission	Schema Level Permission		
SELECT	SELECT	MANAGE	ADMIN
INSERT	X		
UPDATE	X		
DELETE	X		
ALTER	X		
ADMIN	X	X	

Create Table Permissions

You can control which users and roles have access to tables you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your table, then click the **Permissions** tab.
3. Click ⊕ **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Table Permissions

You can modify the permissions of users or roles for tables you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your table, then click the **Permissions** tab.
3. Next to the permission, click ⋯ **Actions** and click **Edit**.
4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Table Permissions

You can delete table permissions to remove access and actions for all contained users or roles.

1. On the home page, click **Master Catalog**.
2. Navigate to your catalog, then click the **Permissions** tab.
3. Next to your permission, click ⋯ **Actions** and click **Delete**.
4. Click **Delete**.

Volume Permissions

Volume permissions determine which users, roles, and groups can view and modify your volumes.

Volume permissions grant the following actions:

- **READ:** Users can list folders/files and read files from volume.
- **WRITE:** Users can list folders/files, read files, create folder and files and write to files in a volume.
- **ADMIN:** User will have READ/WRITE permissions on the volume, delete/create a volume, and will be able to grant/revoke permissions on the volume.

Operation	READ	WRITE	ADMIN
List volume	Yes	Yes	Yes
Read volume data	Yes	Yes	Yes
Write data to volume	No	Yes	Yes
Delete data from volume	No	Yes	Yes
Create folder	No	Yes	Yes
Delete volume	No	No	Yes
Create volume	No	No	Yes
Manage user permissions	No	No	Yes

Create Volume Permissions

You can control which users and roles have access to volumes you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your volume, then click the **Permissions** tab.
3. Click ⊕ **New Permission**.
4. Select the permissions level and user type from the dropdowns.
5. Select whether to add the user by user name or OCID.
   - For **User name**, click Search and enter a user name. Select the user from the list.
   - For **Enter OCID**, enter the OCID of the user.
6. Click **Create**.

Modify Volume Permissions

You can modify the permissions of users or roles for volumes you own.

1. On the home page, click **Master Catalog**.
2. Navigate to your volumes, then click the **Permissions** tab.
3. Next to the permission, click ··· **Actions** and click **Edit**.
4. Select the new permission level from the **Permissions** dropdown and click **Save**.

Delete Volume Permissions

You can delete volume permissions to remove access and actions for all contained users or roles.

1. On the home page, click **Master Catalog**.

2. Navigate to your volume, then click the **Permissions** tab.
3. Next to your permission, click ··· **Actions** and click **Delete**.
4. Click **Delete**.

9

Audit Log

Audit logs in Oracle AI Data Platform Workbench capture detailed records of user activities. These logs help monitor usage, ensure compliance, and investigate issues such as unauthorized access or configuration changes.

Audit logs are accessed from the **Audit logs** link in the left navigation pane of your AI Data Platform Workbench home page. You can track the operations of the following objects in your AI Data Platform Workbench:

- Auto-Populate extractors
- Compute clusters
- External Catalogs
- Roles
- Standard Catalogs
  - Schema
  - Tables
  - Views
  - Volumes
    - Files
    - Folders
- Workspaces
  - Workspace files
  - Workspace folders

Audit logs tracks the attributes of each object logged, including:

- Object name
- Object type
- Operation
- Operation request details
- Date, time, and timezone
- Details of the user that initiated the object (Initiated by)
- Details of where the object was initiated (UI, Terraform, SDK, API, Notebook, etc.)
- Status of the object (Success, failure)

Filters and search options enable you to narrow the parameters of which objects are displayed to help you locate logs for specific objects or periods of time.

Who can view Audit Logs?

Users assigned the AUDITOR role can view the entire audit trail. When a new AI Data Platform Workbench is created, the AUDITOR role is automatically created and the

AI_DATA_PLATFORM_ADMIN role is made a member by default. Any users added to the AI_DATA_PLATFORM_ADMIN role are also automatically added as members of the AUDITOR role.

Users with administrator permissions for a specific object, such as Catalog, Schema, Tables, or Volumes, are able to view the audit logs for that object type.

> **Note**
>
> Log groups are created in AI Data Platform Workbench, but management of log groups, such as deleting log groups that aren't in use, takes place in Oracle Cloud Infrastructure. See Logging Overview for more info.

Search and Filter Audit Logs

You can search and filter records in audit logs to narrow the list of displayed logs and identify specific patterns, trends, or issues with your Oracle AI Data Platform Workbench objects.

1. On your home page, click **Audit Logs**.
2. Use the provided search fields to locate an object by name or payload.
3. Use the provided filters to help narrow your search by the associated value.

10

Roles

This chapter describes how to use Oracle AI Data Platform Workbench role-based access controls (RBAC) to manage user roles and access.

Topics:

About Roles

Oracle AI Data Platform Workbench lets you manage your users and permissions using role based access controls (RBAC).

 LiveLabs Sprint

 Video

You manage RBAC through the Roles interface, where you can create new roles, modify existing, or delete unused roles. After you've provisioned a role, you can assign members by individual user, group, or other roles. You can review and modify the assigned members for any role you have created. You can check the permissions assigned to the role from the Permissions tab.

By default, Oracle AI Data Platform has two system roles, AI_DATA_PLATFORM_ADMIN and AUDITOR:

- AI_DATA_PLATFORM_ADMIN is automatically assigned to the user that created the data platform. This user has administrator permissions to all data platform objects and can grant or revoke permissions to other users, groups, or AI Data Platform Workbench roles. To create an Oracle AI Data Platform Workbench instance you need MANAGE AI Data Platform IAM permission.
- AUDITOR users are able to view the entire audit trail of objects in your AI Data Platform Workbench. The AI_DATA_PLATFORM_ADMIN is automatically made a member of the AUDITOR role when you create your AI Data Platform Workbench instance. Any users added to the AI_DATA_PLATFORM_ADMIN role are added to the AUDITOR role as well.

> **Note**
>
> Your AI Data Platform Workbench instance can only have one AI_DATA_PLATFORM_ADMIN system role. If the AI_DATA_PLATFORM_ADMIN role needs to pass to another user, a user with MANAGE AI Data Platform IAM permissions can reassign it to another user by logging in to OCI and viewing the details of the AI Data Platform Workbench instance.

RBAC permissions are passed down to contained objects. Permissions granted at the Master Catalog level cascade down to all contained objects.

Map Active Directory Groups to IAM Groups

To map Active Directory (AD) groups to Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) groups, you need to establish a federation between your AD and your OCI tenancy.

To map AD groups to IAM groups, see Federating with Microsoft Active Directory.

This process involves creating mappings between AD groups and corresponding IAM groups in OCI, allowing users in your AD groups to access OCI resources with appropriate permissions. Once federated, your AD groups are visible in OCI and you can add group mappings by following the steps in **To add group mappings for an identity provider** under Managing Identity Providers in the Console.

Once you have added group mappings, you can assign permissions to IAM groups in AI Data Platform.

Create a Role

You can create new role as part of RBAC management.

1. On the Home page, click **Roles**.
2. Click ⊕ **New Role**.
3. Provide a name and description for the role.
4. Click **Create**.

Modify a Role

You can modify settings of a role you own.

1. Navigate to **Roles**.
2. Next to the role you want to modify, click ··· **Actions** then **Edit**.
3. Make your changes to the role, then click **Save**.

Delete a Role

You can delete Oracle AI Data Platform Workbench roles that you own.

1. Navigate to **Roles**.
2. Next to the role you want to delete, click ··· **Actions** then **Delete**.

3. Click **Delete**.

Assign Members to a Role

You can assign users, groups, or other roles to a role you created.

1. Navigate to **Roles** and click the role you want to add members to.
2. Click **Members** then click **Add Members**.
3. From the Principle Type, select **User**, **Group**, or **Role**.
   - For **User**, you search for an individual user by name or you provide the OCID of the user.
     - To assign a user by name, you select the compartment and domain then select the user from the list. Enter the user name in the search bar to narrow results.
     - To assign a user by OCID, enter their OCID in the provided field.
   - For **Group**, you search for a group name or you provide the OCID of the group.
     - To assign a group by name, you select the compartment and domain then select the group from the list. Enter the group name in the search bar to narrow results.
     - To assign a group by OCID, enter their OCID in the provided field.
   - For **Role**, you select a role from the list provided.
4. Click **Create**.

Remove Member from Role

You can remove assigned members from roles you own.

1. Navigate to **Roles** and click the role you want to remove members from.
2. Click **Members**.
3. Next to the member you want to remove, click **Actions** then **Remove assignee**.
4. Click **Delete**.

Part IV
Data Engineering

The section explains the methods for developing your data in AI Data Platform Workbench.

Data engineers focus on building and maintaining the systems that data analysts use to access and manipulate data. They use big data technologies like Apache Spark and programming languages including Python and SQL to process and manage data located in object storage, databases, and data warehouses. They are responsible for the initial stages of the data analytics and data science workflow, such as collecting, storing, and transforming data. Their work ensures that the data is accessible and is of high quality so that other data scientists and analysts can use it for their work. Data Engineers also use CI/CD principles for data pipelines and code to manage version control and promote collaboration with data scientists, analysts, and other stakeholders.

Topics:

11

Notebooks

This chapter provides information on using and managing notebooks in your workspace.

Topics:

Develop Code in Notebooks

Data engineers and data scientists can use notebooks in their Oracle AI Data Platform Workbench as a common tool for interactively developing code and exploring data.

Oracle AI Data Platform Workbench currently supports Python, SQL, and Scala languages in notebooks. Notebooks can be scheduled or configured to run as part of a workflow. To run notebooks, you need to attach a compute cluster.

Your AI Data Platform Workbench comes with integrated managed notebooks for an intuitive developer experience.

You can use the sample code in the Oracle AI Data Platform Workbench Samples Git repository for examples of code you can use with your notebook.

Auto-save

Notebooks are automatically saved every two minutes.

Importing and Exporting Notebooks

You can currently import a notebook file (*.ipynb) from your local machine to your workspace.

Exporting notebooks is not currently supported.

Create a Notebook

You can create a notebook in any workspace you have administrator permissions.

1. On the Home page, navigate to your workspace.
2. Click **Create** and click **Notebook**.
3. Fill in the name and description, then click **Create**.

Attach an Existing Cluster to a Notebook

Notebooks require an attached cluster to provide compute power for developed code.

1. On the Home page, navigate to your workspace and open your notebook.
2. Click **Actions** then click **Attach an existing cluster**.
3. Click on the cluster you want to use from the list.

   Your notebook will show **Cluster: (ClusterName) running** when it has been successfully attached. This can take up to several minutes.

Create a Cluster for a Notebook

You can create a new cluster directly from the notebook interface and attach it immediately.

For more information, see About Compute Clusters.

1. On the Home page, navigate to your workspace and open your notebook.
2. Click **Actions** then click **Create cluster**.
3. Select **Runtime version**.
4. Select the driver options for your cluster.
5. Select the worker options for your cluster. These options apply to all cluster workers.
6. Select whether the number of workers is static or scales automatically.
   - If **Static amount**, specify the number of workers.
   - If **Autoscale**, specify the minimum and maximum number of workers the cluster can scale to.
7. For **Run duration**, select whether the cluster will stop running after a set duration of inactivity. If **Idle timeout** is selected, specify the idle time, in minutes, before the cluster will time out.
8. Click **Create**.

Default Language

You can use notebooks to develop and run Apache Spark code in Python, SQL, or Scala.

The default language for notebooks is Python. You can change the default language for the whole notebook or for individual cell(s) to SQL, Scala, or Markdown or raw text. You can combine Python, SQL, and Scala code in different cells within the same notebook.

Notebooks have syntax highlighting for Python, SQL, and Scala. New notebook cells will be created based on the default language of the notebook.

Manage Notebooks

You can rename and delete notebooks your own. You can also clone notebooks to copy their contents and work with its code in a new notebook.

You rename and delete your notebooks from their action menu in your workspace. You clone your notebook by opening it and selecting the **Clone** option from the File menu.

Rename a Notebook

If the name of your notebook is no longer helpful or relevant, you can change it at any time.

1. On the Home page, navigate to your workspace.
2. Next to the notebook you want to rename, click **Actions** then **Rename**.
3. Enter a new name and click **Save**.
4. Optional: You can also change the name of an open notebook by clicking the name and entering a new one.

Delete a Notebook

You can delete notebooks that you have administrator permissions for.

1. Navigate to your workspace.
2. Next to the notebook you want to delete, click **Actions** then **Delete**.
3. Click **Delete**.

Clone a Notebook

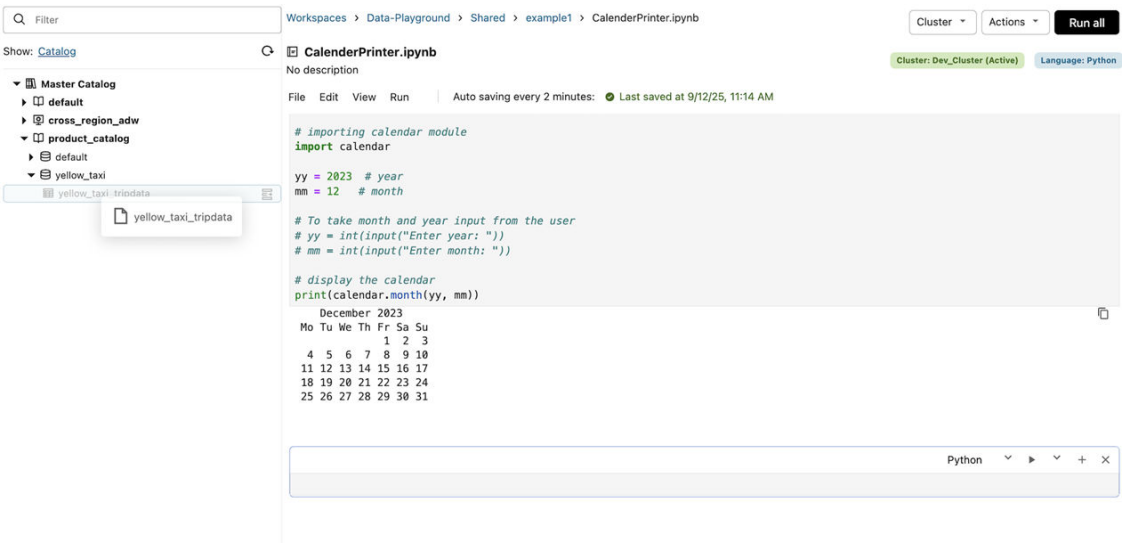
You can clone an existing notebook to create a copy of the content of that notebook you can modify while retaining the original.

1. Open the notebook you want to clone.
2. In the notebook toolbar, click **File** then click **Clone**.
3. Enter a new name for the cloned notebook.
4. Click **Browse** to select the workspace folder to save the cloned notebook into. If no folder is selected, the cloned notebook is created in the same folder as the notebook you are cloning.
5. Select whether to include or exclude outputs. Outputs are included by default. Clear the selection to exclude outputs.
6. Click **Clone**. The cloned notebook is created in the workspace folder you specified.

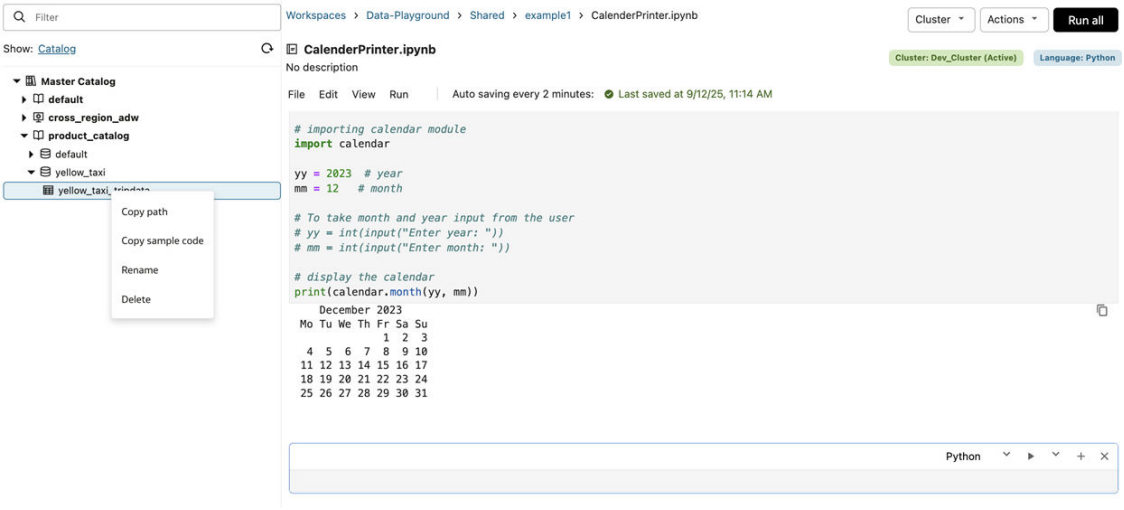
Browse Resources While Editing Notebook

When you are in a notebook, you can browse the Catalog or workspace objects on the left side without leaving your notebook.

If you drag and drop any object from the left hand pane to the notebook, the object name or the full path is copied and pasted to the notebook cell (depending on the context).



You also have a button and context menu options available for each catalog or workspace object in the left hand pane. The context menu at the left navigation has options to copy sample code, copy name, or copy path and so that you can paste to your notebook cell.



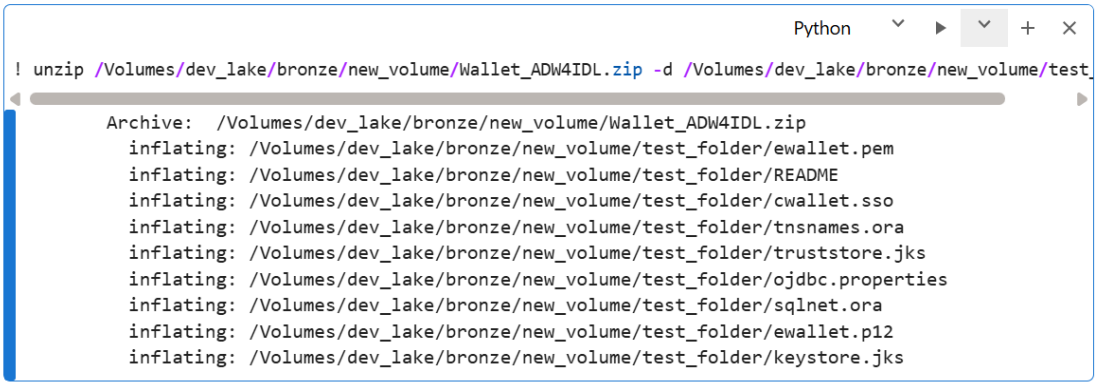
Run Notebooks

You can run code in notebooks you own or from notebooks that are shared with you.

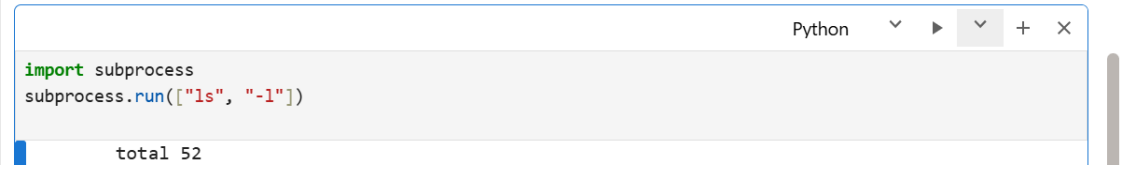
Code can be run from a notebook using three methods: running on demand, running as a one-off manual run, or creating a scheduled notebook job. Jobs run on demand are run only once.

Running Terminal Commands Within a Notebook

You can run basic terminal commands or shell commands within a notebook by prefixing with an '!'. For example, you can use the **unzip** command to extract from ZIP files in the workspace.



You can also use the **subprocess** module in Python for shell script execution.



You can also use native Python modules like **zipfile** for tasks like unzipping files as an alternative to shell commands.

Limitations

Currently, Oracle AI Data Platform Workbench does not have native support for pip install, CI/CD, Git, or version control systems.

Run Options for Notebook Cells

The Run menu in notebooks provides you with options for running cells in a notebook.

<Enter Section Title Here>

You can find all options for running cells in your notebook from the **Run** menu at the top of your notebook.

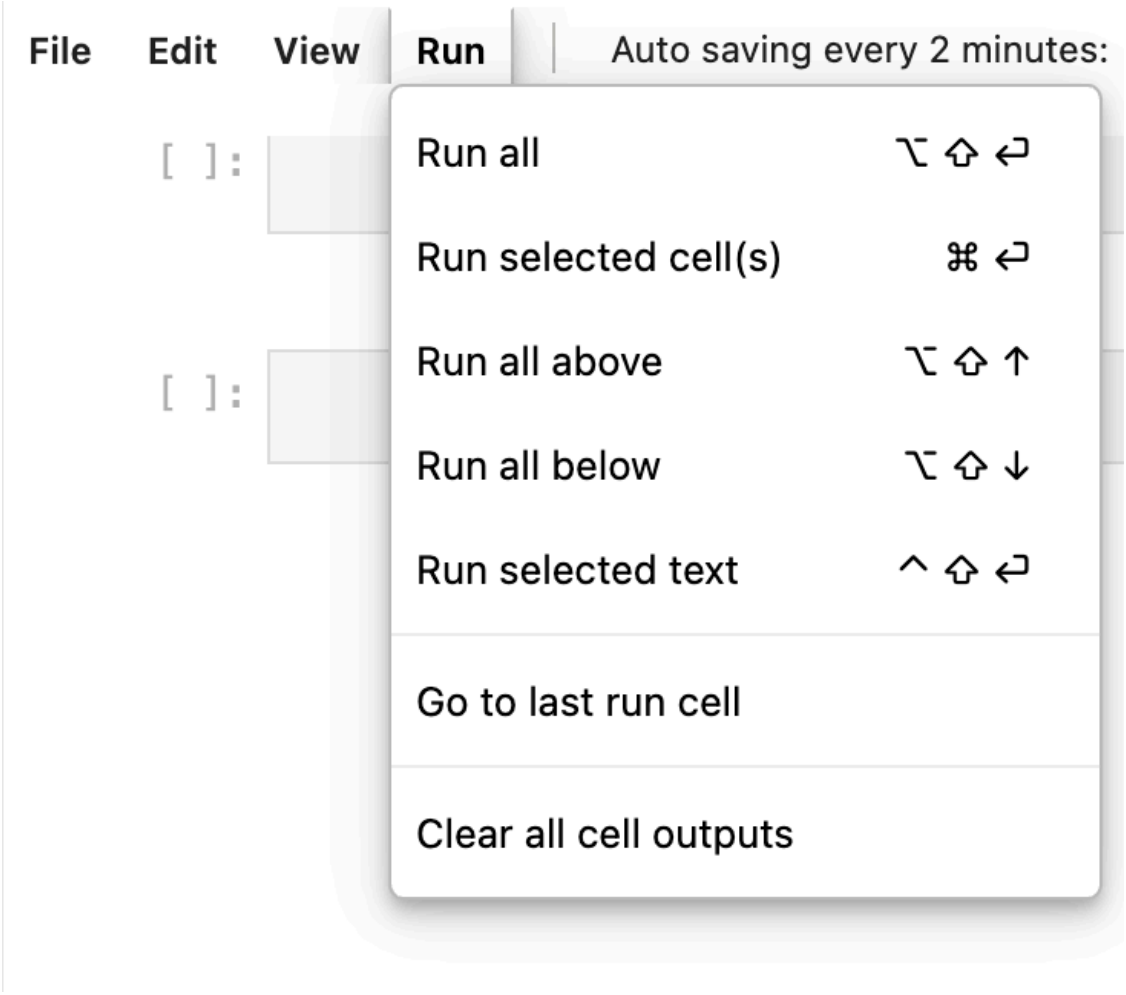


Table 11-1 Run Options for Notebook Cells

Option	Description
Run all	Runs all cells in the notebook sequentially.
Run selected cell(s)	Runs the currently selected cell or cells.
Run all above	Runs the currently selected cell and any cells that appear above the currently selected cell in the notebook.
Run all below	Runs the currently selected cell and any cells that appear below the currently selected cell in the notebook.
Run selected text	Runs the selected segment of code in a cell.
Go to last run cell	Navigates you to the most recently run cell in the notebook.
Clear all cell outputs	Removes outputs from all cells in the notebook.

Run Code from a Notebook

You can choose to run all code developed in a notebook at once, or one cell at a time.

Keyboard shortcuts for running code in a notebook are:

- **MacOS**: `Cmd + Return`
- **Windows**: `Ctrl + Enter`

You can run code in a single cell by clicking the ▸ **Play** button, or run the whole notebook by clicking **Run all**.

1. On the Home page, click **Workspace**.
2. Navigate to your notebook.
3. Click **Run all**.
4. Check the status of your notebook job run by clicking **Workflow** then **Job Runs**.

Run Code from Another Notebook

You can use %run magic command in a notebook to include code from another notebook.

In the following example, you bring in code from a notebook named **called-notebook.ipynb**, to a notebook **caller-notebook.ipynb**.

1. Install the **nbconvert** Python library.
2. Use the `%run` command in a cell, as in the following example:

```
%run /Workspace/folder1/called-notebook.ipynb
```

After following these steps, the notebook named **called-notebook.ipynb** is immediately run using your user principal (i.e. **caller-notebook.ipynb**) and using the attached cluster of **caller-notebook.ipynb**. All the functions and variables defined in **called-notebook.ipynb** immediately become available in the notebook named **caller-notebook.ipynb**.

Share Notebook Output with oidlUtils

You can capture and share content output by your notebook tasks by using utilities available in oidlUtils.

oidlUtils is a set of utilities available to all users of Oracle AI Data Platform Workbench. When sharing content between notebooks, oidlUtils can be called to pass arguments to one notebook and pass output back to the caller notebook, and it can be called in a job task in a notebook to pass output back to the parent task, meaning the task calling on the notebook. Used this way, you can capture and use structured output returned by your notebook tasks.

These oidlUtils modules are available for use with notebooks:

Module	Description	Example
notebook	Orchestrate notebook task runs and return a single structured result (commonly a JSON string) to the caller.	`oidlUtils.notebook.run( notebook_path: str, timeout_seconds: int = 0, (Optional) parameters: dict (Optional) ) -> str`

Module	Description	Example
notebook	Allow a notebook to exit a task run and return a single string result (commonly a JSON payload) to the caller notebook or job/task output API.	`oidlUtils.notebook.exit( value: str)`

Example 1: Notebook to Notebook Sharing

In this example you use Notebook A to invoke Notebook B. Notebook B returns a result payload that Notebook A is set up to capture and use.

Notebook A

```
result = oidlUtils.notebook.run("NotebookB", 0)
print("Output from Notebook B:", result)
```

Notebook B

```
import json

payload = {
"status": "SUCCESS",
"rows_processed": 1234,
"output_table": "sales_gold",
"run_id": "run_2026_02_11"
}

json_payload = json.dumps(payload)
oidlUtils.notebook.exit(str(json_payload))
```

Output from Notebook B

```
{"status": "SUCCESS", "rows_processed": 1234, "output_table": "sales_gold",
"run_id": "run_2026_02_11"}
```

Example 2: Passing Output through Job Tasks

In this example, you return a JSON file when your notebook is run as a task: `oidlUtils.notebook.exit(json.dumps(payload))`.

```
import json

payload = {
    "status": "SUCCESS",
    "output_table": "sales_gold",
    "rows_processed": 1234
}

oidlUtils.notebook.exit(json.dumps(payload))
```

Next you run the job with your notebook task and get the task output through the API call: `endpoint = f"https://<workspace-url>/jobs/runs/get-output?run_id={task_run_id}"` and `response = requests.get(endpoint, headers=headers).json()`.

```
import requests

task_run_id = "<task_run_id>"

endpoint = f"https://<workspace-url>/jobs/runs/get-output?
run_id={task_run_id}"
response = requests.get(endpoint, headers=headers).json()
```

Last, you capture the output returned by the notebook with `job_result = response['notebook_output']['result']`.

```
job_result = response["notebook_output"]["result"]
payload = json.loads(job_result)

print(payload["output_table"])   # Output : sales_gold
print(payload["rows_processed"]) # Output : 1234
```

Notebook Navigation

You can create and maintain a table of contents that can be used to organize and navigate your notebook.

You can click the table of contents icon in the top left of your notebook to display a notebook outline. The table of contents is automatically generated based on markdown headings you can create, enabling easy organization and navigation.

You can added formatted text, headings, lists, and documentation as markdown to organize and explain notebook content for yourself and other users.

Create a Markdown Cell

You can create markdown cells to provide headings in your notebook table of contents for ease of organization and navigation.

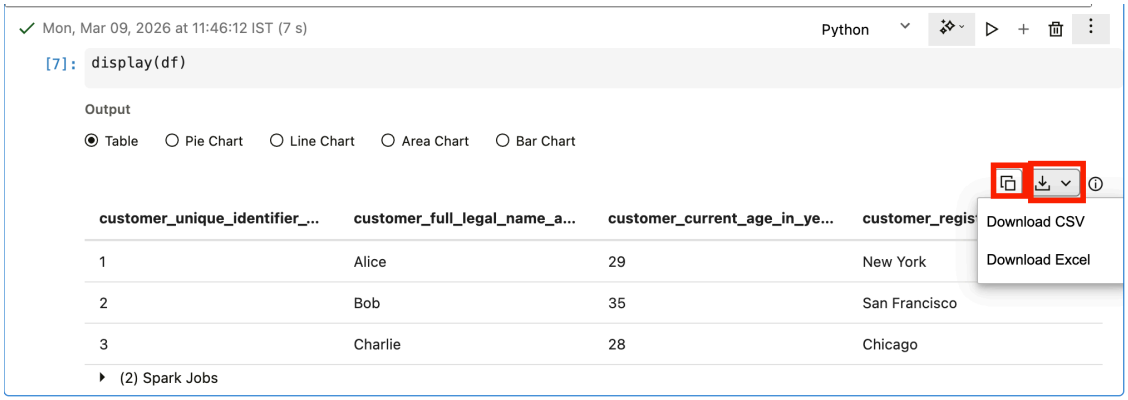
1. From the Home page, navigate to your notebook.
2. From the cell type drop-down list, select **Markdown**.
3. Add your markdown to the cell.
   - Markdown cells can include formatted text, headings, lists, and other documentation.
   - To create a heading, start a line with # followed by a space. Heading 1 uses one #, heading 2 uses ##. Add additional # for each additional heading level.

Notebook Output and Results

You can see notebook outputs and results in a new cell that appears right after the cell with code.

While a cell is in progress, you can cancel the execution of the cell. If a notebook is run as a workflow job, the output is not visible in the same notebook. In that case the output is visible in the output area of the corresponding workflow job run.

You can download the output from the output cell as a CSV or Excel file. You can also copy the contents of the output cell directly to your clipboard.



Download Notebook Output

You can download the resulting output of a notebook cell directly from the results panel.

1. In the top-right of the cell you want to download output from, click **Download**.
2. From the menu, select the format you want to download output as.

Copy Notebook Output

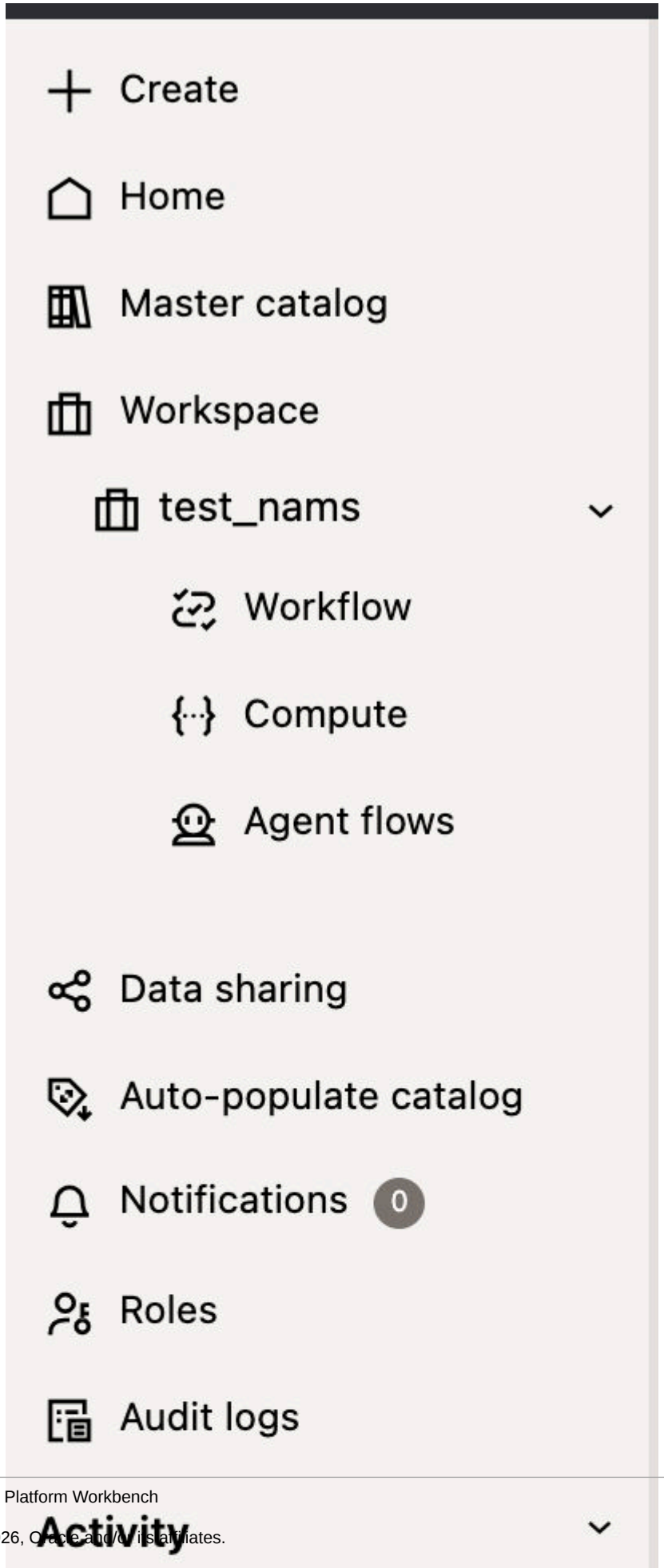
You can copy the resulting output of a notebook cell directly from the results panel.

1. In the top-right of the cell you want to download output from, click **Copy**.
2. The contents of the output cell are copied directly to your clipboard. You can paste it elsewhere in your notebook or to an external location.

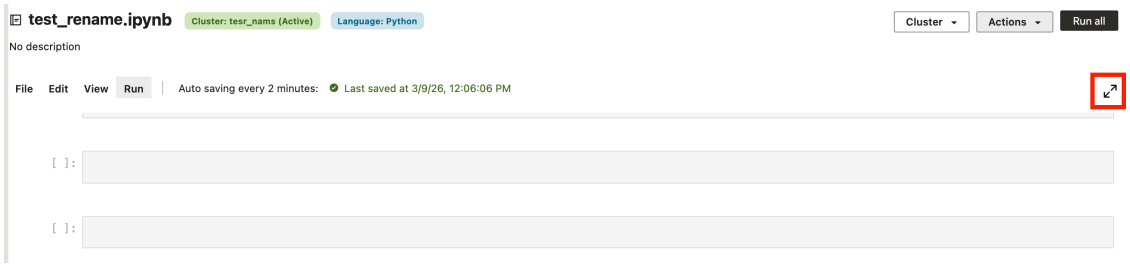
Notebook Appearance

You can alter the screen space available for working in notebooks by minimizing the left navigation panel or expanding the notebook view.

You can minimize the Oracle AI Data Platform Workbench left navigation panel while working on your notebook to increase the usable screenspace by clicking **Minimize panel** on the bottom-right of the navigation panel.



You can also expand the notebook by clicking **Expand** at the top right of the notebook, expanding the space available and making larger cells and outputs easier to read.



Cell Run Numbers

Each notebook cell displays a run number that indicates the order in which cells were run. This number updates every time the cell is run. You can run cells in any order, so the run numbers may not match the physical order of cells in your notebook.

Manage Job Runs

You can create job runs to manage how and when code is run from your notebooks.

Manual job runs can be run again or later set up to run on a schedule. Scheduled job runs are automatically triggered based on the schedule you set. Unless a schedule is configured, manual jobs are only run once.

Create a Manual Run Job from a Notebook

You can create an unscheduled job that you can run manually from code you've developed in your notebook.

1. On the Home page, click **Workspace**.
2. Navigate to your notebook.
3. Click **Actions**, then click **Schedule**.
4. Provide a name and description for the job.
5. Click **Browse** and select the location to store your job. Click **Select**.
6. Select a compute cluster from the **Cluster** dropdown.
7. For **Schedule**, select **Manual Run**.
8. Click **Create**.

Create a Scheduled Job Run from a Notebook

You can create a scheduled job that runs automatically from code you've developed in your notebook.

1. On the Home page, click **Workspace**.
2. Navigate to your notebook.
3. Click **Actions**, then click **Schedule**.

4. Provide a name and description for the job.
5. Click **Browse** and select the location to store your job. Click **Select**.
6. Select a compute cluster from the **Cluster** dropdown.
7. For **Schedule**, select **Schedule**.
8. Select a **Schedule Status**.
   - Select **Active** if you want the schedule to be enabled immediately.
   - Select **Paused** if you want to manually enable the scheduled run at a later time.
9. Provide a time zone for the schedule to be based on.
10. Select the **Schedule Type**.
    - For **Calendar**, you must specify the frequency and which hours or days the schedule will repeat on.
    - For **Cron Expression**, you must provide the schedule in the form of a cron expression.
11. Check the listed run time at the bottom to confirm your schedule is correct. Click **Create**.

Notebook Keyboard Shortcuts

You can use keyboard shortcuts to simplify using commands in your notebook.

Windows	macOS	Action
Ctrl + Enter	Cmd + Return	Execute cell
Shift + Enter	Shift + Return	Execute cell and advance to next cell
Ctrl + S	Cmd + S	Save notebook
Ctrl + N	Ctrl + N	New notebook
Ctrl + Z	Cmd + Z	Undo
Ctrl + Y	Cmd + Y	Redo
Ctrl + C	Cmd + C	Copy
Ctrl + X	Cmd +X	Cut
Ctrl + V	Cmd + V	Paste
Ctrl + Alt + F	Ctrl + Option + F	Find and Replace
Ctrl + Shift + A	Ctrl + Shift + A	Insert cells above
Ctrl + Shift + B	Ctrl + Shift + B	Insert cells below
Ctrl + Alt + Up	Ctrl + Option + Up	Move cell up
Ctrl + Alt + Down	Ctrl + Option +Down	Move cell down
Ctrl + D	Ctrl + D	Delete cell
Alt + Shift + Enter	Option + Shift + Return	Run All
Alt + Shift + Up	Option + Shift + Up	Run all above cells

Migrate Existing Apache Spark Code to Oracle AI Data Platform Workbench

You can adapt your Apache Spark code to migrate it for use in Oracle AI Data Platform Workbench notebooks.

If you are migrating existing Spark code from other platforms, you can use the following guidelines to adapt your code for use in notebooks.

Table 11-2 Apache Spark to AI Data Platform Migration Guidelines

Guideline	Details
Remove SparkSession creation commands	AI Data Platform Workbench automatically creates a SparkContext for each compute cluster. We recommend removing the session creation commands or replacing them with `SparkSession.builder().getOrCreate()`.
Remove session termination commands, like `sys.exit()` or `spark.stop()`	All purpose compute clusters are shared clusters, so if any users stop the SparkSession, by using `sys.exit()` or `spark.stop()` for example, the cluster needs to be restarted for everyone. To avoid disruption, we recommend avoiding those commands in the notebooks.

12

Workflows

Workflows in Oracle AI Data Platform Workbench provide a powerful and flexible way to automate data processing tasks. With workflows, users can define and orchestrate complex data pipelines that can run on-demand and based on a pre defined schedule. Workflows can be composed of multiple tasks, each performing a specific action, and can include advanced features such as dependencies, triggers, and error handling.

Video

Topics:

Key Features of AI Data Platform Workflows

- Automation: Automate complex data tasks and processes.
- Orchestration: Define the sequence and dependencies of tasks in a pipeline.
- Scheduling: Run workflows on a schedule or trigger based on specific events.
- Monitoring: Track workflow status, logs, and execution history.
- Parameterization: Pass parameters to customize the behavior of workflows and tasks.

Core Concepts

- Job: A collection of tasks executed in sequence or parallel to complete a data processing job.
- Task: The individual steps that make up a workflow. Tasks can include actions like running Python code, executing a notebook, if-else task or running another job task.
- Job Run: An instance of a job execution. A job can be triggered multiple times, each time representing a new job run.
- Trigger: Defines the conditions under which a workflow is executed, such as on a schedule, or if it's manually triggered.
- Dependencies: Define the order of task execution or specify conditions under which certain tasks run.
- Parameters: Values passed to workflows or tasks to customize their execution. Parameters can be defined at the job, task, or runtime level.

Benefits/Use Cases of Using Workflows

- Streamlined Automation - Simplify the execution of recurring data tasks by automating them through workflows.
- Parallel Processing - Speed up data processing by running tasks in parallel.

- Customizable Execution - Modify workflows at runtime with parameters to meet specific needs.
- Improved Efficiency - Reduce manual interventions and errors, enabling smoother operations.

Workflows in an AI Data Platform Workbench enable a wide range of use cases, including automated ETL pipelines, data integration from multiple sources, and advanced analytics. Users can automate data quality monitoring, machine learning model training, and deployment. These capabilities drive efficiency and scalability for modern data-driven workflows.

Best Practices

- Task Modularization - Break down workflows into reusable tasks to simplify management and improve maintainability.
- Efficient Resource Allocation - Optimize workflows for better performance by running tasks in parallel when appropriate.
- Error Handling - Use retries, error notifications, and fallback mechanisms to ensure workflows run reliably.
- Compute Assignment - Assign specific compute resources to each task based on workload size, optimizing performance and cost.

By following these best practices, you can design workflows that are scalable, reliable, and efficient, ensuring optimal performance and easier management in Oracle AI Data Platform.

Configure Jobs

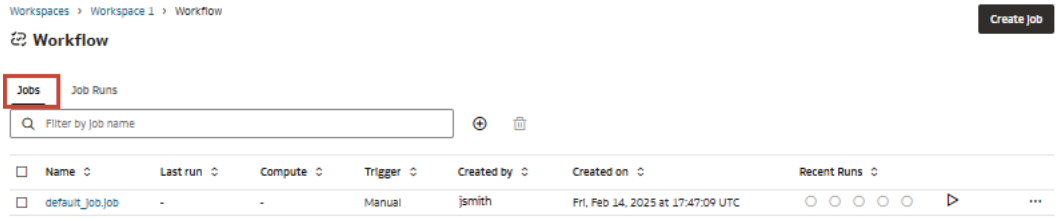
This section covers configuring jobs and job runs in your AI Data Platform Workbench.

Topics:

About Jobs

You create workflows for your data by constructing jobs.

You can track and manage all jobs in your AI Data Platform Workbench from the **Jobs** tab in the Workflow home page. From the **Jobs Run** tab you can see job history and the status of runs currently in progress.



Jobs are way of organizing and orchestrating groups of tasks as part of workflow. You can use workflows for common data processing, such as ETL workflows, Python scripting, running notebooks, and machine learning.

Jobs can vary in complexity. One job may have only a single task that runs a notebook, while another has more than a hundred tasks and nested jobs carrying out complex tasks with multiple conditions and dependencies.

Running a job starts the tasks inside with the sequences and conditions you specified. Jobs can even be nested inside of other jobs, embedding their sequence of tasks as a single node.

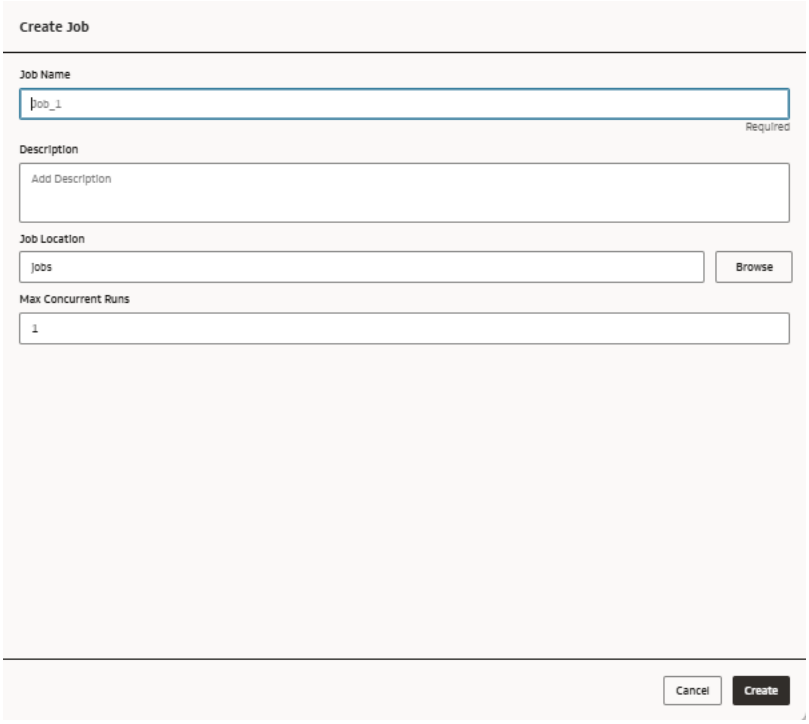
Every job run generates a job file that is stored in a user-specified location during job creation. This job file acts as a reference for managing and tracking job executions. You can run, schedule, and view the JSON representation of the job directly from the file in the workspace, ensuring transparency and reproducibility. This approach allows seamless job monitoring, versioning, and integration within automated workflows.

Jobs can configured to run on a calendar schedule, scheduled to run based on a cron expression, or run immediately.

Create a Job

You need to create a job to begin using workflows in Oracle AI Data Platform Workbench.

1. Navigate to your workspace and click **Workflow**.
2. On the **Jobs** tab, click **Create Job**. You can also click **Create Job** on the top right.



3. Provide a name and description for your job.
4. Click **Browse** and select the location to save the job in your AI Data Platform Workbench. Click **Select**.
5. Enter a number for **Max Concurrent Runs**.
6. Click **Create**.

Change Job Location

You can change the location of a job after creation.

1. On the Home page, click **Workflow**.
2. Click the job you want to change the location for.
3. Click the **Details** tab.
4. Next to Location, click **Browse**.
5. Pick a new location for the job and click **Select**.

Delete a Job

You can delete jobs that you no longer need.

1. Navigate to your workspace and click **Workflows**.
2. Next to the job you want to delete click **Options** then click **Delete**.
3. Click **Delete**.

Schedule a Job Using a Calendar

Workflow jobs can be scheduled to run on an automated basis.

1. On the Home page, click **Workflow**.
2. Click the job you want to make a schedule for.
3. Click the Details tab.
4. Next to schedule, click **Add**.
5. Choose whether your schedule begins as **Active** or **Paused**.
   - Select **Active** if you want the schedule in effect immediately.
   - Select **Paused** if you want to activate the schedule at a later time.
6. Select the **Time Zone** the schedule uses as a basis.
7. From **Schedule Type**, select **Calendar**.
8. Select whether the schedule will be run hourly, daily, weekly, or monthly. You need to provide additional information for **Hourly**, **Weekly**, and **Monthly** options.
   - For **Hourly**, select the hours on which the schedule repeats.
   - For **Weekly**, select the days of the week the schedule repeats.
   - For **Monthly**, select the days of the month the schedule repeats.
9. Enter the time of day the schedule runs. The time of day is in 24 hour format, beginning at 00:00 and ending at 23:59.
10. Check the listed run time at the bottom to confirm your schedule is correct. Click **Create**.

Schedule a Job Using a Cron Expression

Workflow jobs can be scheduled to run on an automated basis using cron expressions to determine date and times.

1. On the Home page, click **Workflow**.
2. Click the job you want to make a schedule for.
3. Click the Details tab.
4. Next to schedule, click **Add**.
5. Choose whether your schedule begins as **Active** or **Paused**.
   - Select **Active** if you want the schedule in effect immediately.
   - Select **Paused** if you want to activate the schedule at a later time.
6. Select the **Time Zone** the schedule uses as a basis.
7. From **Schedule Type**, select **Cron Expression**.
8. Enter the cron expression.
9. Check the listed run time at the bottom to confirm your schedule is correct. Click **Create**.

Run a Workflow on Demand

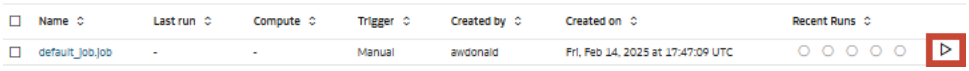
You can choose to run a workflow job immediately.

1. On the Home page, click **Workflow**.
2. Click on the job you want to run.
3. Click **Run Now**.

Run a Workflow Job from the Jobs Page

You can quickly run listed jobs directly from the Jobs page.

1. On the home page, click **Workflow**.
2. In the **Jobs** tab, next to the listed job you want to run, click **Run Now**.



Change a Job Run Schedule

You can edit the schedule of a job run after creation to modify time, timezone, or frequency of that schedule.

1. On the Home page, click **Workflow**.
2. Click the job you want to make a schedule for.
3. Click the Details tab.
4. Next your schedule, click **Edit**.
5. Modify the schedule as needed. Check the listed run time at the bottom to confirm your schedule is correct.
6. Click **Save**.

Pause or Activate a Job Run Schedule

After setting a schedule for a job run you can choose to pause it then reactivate when needed.

1. On the Home page, click **Workflow**.
2. Click the job you want to make a schedule for.
3. Click the **Details** tab.
4. Next to your schedule click to **Pause** or **Activate** your schedule.

   The option displayed is dependent on the current status of the schedule. If active, **Pause** is displayed. If paused, **Activate** is displayed.

Repair Failed Job Runs

You can attempt to repair a run that has failed by reviewing the timeline and details and rerunning the job with optional parameters to assist your diagnostic.

1. Navigate to your workflow.
2. Click **Job Runs**.
3. Next to the failed job run, click **Actions** and click **Repair run**.
4. Review the task details to determine possible causes of failure.
5. Click **Repair run** and select which tasks to rerun.
6. Add optional parameters that will only apply to this repair run.

- If you select **Key/Value** as the **Parameter** type, click **Add** and enter parameters.
- If you select **JSON**, enter parameter values in the space provided.

7. Click **Run repair**.

Configure Tasks

This section covers the creation and configuration of tasks.

Topics:

About Tasks

Tasks are short, functional blocks of code you can piece together into a flow as part of job or promote to jobs themselves.

Tasks are the primary building blocks of all workflows in Oracle AI Data Platform Workbench. The type of task determines the type of code it uses. As part of a job, you connect tasks to determine their sequence and priority when the job is run.

Task type	Description
Notebook task	A task that has been saved to a notebook you can access
Python task	A task using a snippet of Python programming language
If/else condition	A task that uses if/else conditions
Nested job task	A task that uses an existing job and its tasks as a nested task
Jar task	A task that can run Scala or Java code compiled into Java Archive (JAR) files.

When you have more than one task, you can create sets of task dependencies where the success or failure of one task can trigger subsequent tasks in sequence. You can only create dependencies in jobs that have more than one task. See Create a Notebook Task.

Tasks can be run in parallel with one another. You can do this by making two or more tasks dependent on the success or failure of another task in the same workflow, causing them to run at the same time.

Tasks can fail due to transient issues, such as network disruptions, resource unavailability, or temporary service failures. In these cases, your AI Data Platform Workbench automatically retries the task based on retry policies you configure when the task is created. As part of these policies, you define:

- **Retry Count:** The maximum number of retry attempts.
- **Retry Interval:** The wait time between retries.

In addition to standard task retries, AI Data Platform Workbench also supports **Retry on timeout**. If a task exceeds its execution time limit due to resource constraints or slow processing and you want to retry only for these scenarios, you can choose to automatically trigger a retry. These retry policies enhance workflow resilience, ensuring tasks have a higher chance of successful execution without manual intervention.

When and How to Use Compute Logs

You should check your compute logs if your task fails with resource or system related errors, such as out-of-memory errors or CPU use exceeding limits.

Review Spark logs if you see long wait times, unexpected retries, or job performance bottlenecks. These logs provide insight into the driver and worker nodes of the compute cluster backing your task and can help identify the source of possible problems.

For guidance on how to check your logs, see Monitor a Specific Job Run.

You must have appropriate compute-level RBAC permission to view metadata and logs for the compute instance associated with the job. Contact your administrator to obtain these permissions if you are unable to view compute logs. For more information, see About Permissions.

Create a Python Task

You can create a task as part of workflow job that uses Python scripting.

1. On the Home page, click **Workflow**.
2. Click on the job you want to make a task for.
3. Click **Add task**.
4. Enter a task name.
5. For **Task type** select **Python**.
6. Click **Browse** and navigate to the Python script you want to add as a task. Click **Select**.
7. Select a compute cluster for the Python task, if one is not already attached.
8. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

9. From **Depends on**, select any tasks you want to make this task dependent on. Select the conditional response to that dependency from the **Run if** dropdown.
10. Add additional parameters by providing their **Key** and **Value**. Click **Add parameter** to provide multiple parameters.

Create a Notebook Task

You create tasks using notebooks you have built in your AI Data Platform Workbench.

1. On the Home page, click **Workflow**.
2. Click on the job you want to make a task for.
3. Click **Add task**.
4. Enter a task name.
5. For **Task type** select **Notebook**.
6. Click **Browse** and navigate to the notebook you want to add as a task. Click **Select**.
7. Select a compute cluster for the notebook task, if one is not already attached.
8. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

9. From **Depends on**, select any tasks you want to make this task dependent on. Select the conditional response to that dependency from the **Run if** dropdown.
10. Add additional parameters by providing their **Key** and **Value**. Click **Add parameter** to provide multiple parameters.

Create a Nested Job Task

You can use another workflow job and its contained tasks as a nested task within another workflow.

1. On the Home page, click **Workflow**.
2. Click on the job you want to make a task for.
3. Click **Add task**.
4. Enter a task name.
5. For **Task type** select **Nested job task**.
6. From the **Jobs** drop-down list, select an existing job you want to make a task from.
7. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

8. From **Depends on**, select any tasks you want to make this task dependent on. Select the conditional response to that dependency from the **Run if** dropdown.

9. Add additional parameters by providing their **Key** and **Value**. Click **Add parameter** to provide multiple parameters.

Create an If/Else Task

You can create a task that uses if/else conditions based on catalog data to determine if the task triggers.

1. On the Home page, click **Workflow**.
2. Click on the job you want to make a task for.
3. Click **Add task**.
4. Enter a task name.
5. For **Task type** select **If/Else**.
6. Enter the conditions that determine if the task triggers. Click **Add** to set multiple conditions.
7. Enter the condition expression.
8. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

9. From **Depends on**, select any tasks you want to make this task dependent on. Select the conditional response to that dependency from the **Run if** dropdown.
10. Add additional parameters by providing their **Key** and **Value**. Click **Add parameter** to provide multiple parameters.

Create a Jar Task

You can create tasks that run Scala or Java code compiled into Java Archive (JAR) files.

> **Note**
>
> Dependent library files must use JDK, Scala, or Spark versions compatible with the Oracle AI Data Platform Workbench cluster runtime at time of creation to avoid unexpected behavior.

1. On the Home page, click **Workflow**.
2. Click on the job you want to make a task for.
3. Click **Add task**.
4. Enter a task name.
5. For **Task type** select **JAR task**.

6. For **Main class name** specify the full name of the class that contains the main method you want to execute. For example, `ProcessTransaction`. This class must be included in one of the files added as a dependent library.
7. For **Dependent libraries**, click **Add**.
8. Select a source for your dependent library files. At least one library that contains the main class method specified above must be included.
   - For **Workspace** or **Volume**, browse your AI Data Platform Workbench workspace or volume, select the file or files you want to add as a library and click **Add**.
   - For **Upload file to workspace**, browse your local machine for the file or files to upload as a library and click **Add**.
9. In **Command line arguments**, provide whitespace-delimited arguments that you want to pass to the main class.
10. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

11. From **Depends on**, select any tasks you want to make this task dependent on. Select the conditional response to that dependency from the **Run if** dropdown.
12. Add additional parameters by providing their **Key** and **Value**. Click **Add parameter** to provide multiple parameters.

Modify a Task

You can change existing attributes of a task, such as name, type, and parameters to alter how it functions in your job.

1. On the Home page, click **Workflow**.
2. Click on the job you want to configure tasks for.
3. In the **Tasks** tab, click the task you want to edit.
4. In the **Task Details** pane on the right, modify the task attributes as needed. Changes are saved automatically.

View Task Logs

You can view the run logs of individual tasks in a job.

1. On the Home page, click **Workflows**.
2. Click **Job Runs**.
3. Click the job you want to see the task logs for.
4. Click the task nodes to see the logs for that task.

Delete a Task

You can delete a task by removing the task node from a job.

1. On the Home page, click **Workflow**.
2. Click on the job you want to delete tasks from.
3. On the task node, click **Actions** then click **Remove node**.
4. Click **Delete**.

Parameterization

This section covers parameters and how they are used in jobs and tasks.

Topics:

About Parameters

You can customize job runs by passing parameters that alter the behavior of the job, task, or job run.

Parameters can be provided at three different levels of a workflow: job level, task level, and job run level. Parameters take the following precedence in the case of conflicts: Job Run > Task > Job.

1. **Job Run Level Parameters:** Job run level parameters specified in the job run payload take precedence over task-level parameters defined in the job configuration. This means that any parameters specified during job run override the task-specific defaults.
2. **Task Level Parameters:** If job run parameters for a specific task are not provided, the task uses the parameters defined at the task level in the job configuration. Job-level parameters with the same name override task-level parameters for that specific task. So if you want a task specific parameter, you should name it differently then the job parameter.
3. **Job Level Parameters:** If neither task-level nor runtime parameters are provided, the default values set at the job level apply. Job-level parameters are considered defaults that can be used if no more specific parameters are available.

> **Note**
>
> Job parameters are immutable in task contexts. This means if there is a job with parameter *JobParamA* with resolved value *JobParamRuntimeValueA* then TaskA execution cannot change the value of *JobParamA*. The value of *JobParamA* remains *JobParamRuntimeValueA* for all tasks and the entire job run. As a result, if you wanted to share information between tasks you could use intermediate storage or output parameters to achieve that.

When task names, task value keys, or job parameter names contain special characters (such as !@$%), you must surround those identifiers with backticks (` `). Only alphanumeric and underscore characters are usable without surrounding the identifier in backticks.

For example:

```
{
  "VariableWithSpecialChars": "{{job.parameters.`param$@`}}"
}
```

System Parameters are templated parameters whose value is provided by the system as part of workflow runs and subsequent task runs. You don't have to provide any value, default or otherwise, for these templated parameters. AI Data Platform Workbench has a fixed list of valid templated parameters / dynamic value references that are supported in workflow. System parameters are entered by surrounding them with two curly brackets. For, example `{{job.id}}`.

Table 12-1 Supported System Parameters

Parameter	Description
{{hub.id}}	The unique identifier assigned to the hub
{{hub.region}}	The region of the hub
{{workspace.id}}	The unique identifier assigned to the workspace
{{workspace.url}}	The URL of the workspace
{{job.id}}	The unique identifier assigned to the job
{{job.name}}	The name of the job at the time of the job run
{{job.run_id}}	The unique identifier assigned to the job run
{{job.repair_count}}	The number of repair attempts on the current job run
{{job.start_time.[argument]}}	A value based on the time (in UTC timezone) that the job run started. The return value is based on the argument option. See Options for date and time values.
{{job.parameters.[name]}}	The value of the job-level parameter with the key [name]
{{job.trigger.type}}	The trigger type of the job run. The possible values are Manual and Scheduled.
{{job.trigger.file_arrival.location}}	If a file arrival trigger is configured for this job, the value of the storage location
{{job.trigger.time.[argument]}}	A value based on the time (in UTC timezone) that the job run was triggered, rounded down to the closest minute for jobs with a cron schedule. The return value is based on the argument option. See Options for date and time values.
{{task.name}}	The name of the current task
{{task.run_id}}	The unique identifier of the current task run
{{task.execution_count}}	The number of times the current task was run (including retries and repairs)
{{task.notebook_path}}	The notebook path of the current notebook task
{{tasks.[task_name].run_id}}	The unique identifier assigned to the task run for [task_name]

Table 12-1 (Cont.) Supported System Parameters

Parameter	Description
{{tasks.[task_name].result_state}}	The result state of task [task_name]. The possible values are success, failed, excluded, canceled, skipped, timed out, upstream_canceled, and upstream_failed.
{{tasks.[task_name].error_code}}	The error code for task [task_name] if an error occurred running the task. Examples of possible values are RunExecutionError, ResourceNotFound, and UnauthorizedError. For successful tasks, this evaluates to an empty string.
{{tasks.[task_name].execution_count}}	The number of times the task [task_name] was run (including retries and repairs)
{{tasks.[task_name].notebook_path}}	The path to the notebook for the notebook task [task_name]
{{tasks.[task_name].values.[value_name]}}	The task value with the key [value_name] that was set by task [task_name]

Table 12-2 Options for Date and Time

Argument	Description
iso_weekday	Returns a digit from 1 to 7, representing the day of the week in the time stamp
is_weekday	Returns `true` if the timestamp is on a weekday
iso_date	Returns the date in ISO format
iso_datetime	Returns the date and time in ISO format
year	Returns the year part of the timestamp
month	Returns the month part of the timestamp
day	Returns the day part of the timestamp
hour	Returns the hour part of the timestamp
minute	Returns the minute part of the timestamp
second	Returns the second part of the timestamp
timestamp_ms	Returns the timestamp in milliseconds

Pass Parameters Between Tasks and Notebook

You can pass parameters from a task to a notebook and vice versa. This enables dynamic workflow behavior, allowing notebooks to adjust their processing based on runtime values.

The oidlUtils.parameters package provides the necessary functionality to handle these parameter operations. The oidlUtils package is a utility library in AI Data Platform Workbench that simplifies tasks such as parameter management, task value passing, and other workflow operations. It is commonly used in notebooks and tasks to get and set parameters across workflow stages.

Task key values are either strings or JSONs. For example, to use a string as your task key value, your parameter call would look like this:

```
oidlUtils.parameters.setTaskValue(key="payload", value="abc", "defaultValue")
```

To use a JSON as your task key value, your parameter call would look like this:

```
oidlUtils.parameters.setTaskValue(key="payload", value=json.dumps(payload),
"defaultValue")
```

Example Workflow: Passing Parameters

In this scenario, we have two notebooks in a workflow. Notebook 1 receives parameters from a task, processes them, and sets output parameters that are passed to Notebook 2 in the next task.

Notebook 1: Get and Set Parameters

```
# Get parameter if already set in the task
param_key = "param1"
param_value = oidlUtils.parameters.getParameter(param_key, "defaultValue")
print(param_value)
print("Param {} value is {}".format(param_key, param_value))


# Set parameter value in the task
output_parameter_key = "output_parameter"
output_param_value = oidlUtils.parameters.getParameter(output_parameter_key,
"defaultValue2")
print("Param {} value is {}".format(output_parameter_key, output_param_value))
oidlUtils.parameters.setTaskValue(output_parameter_key, "1234")
```

The first notebook retrieves a parameter (param1) passed from the task and then sets a new parameter (output_param_2), which will be used in the next task.

Notebook 2: Read the Output Parameter

```
output_param_2= "output_parameter"
param_value = oidlUtils.parameters.getTaskValue("GetSetParameter",
output_param_2, "defaultValue")
print("Param {} value is {}".format(output_param_2, param_value))
```

The second notebook receives the `output_param_2` from Notebook 1 via the workflow task and processes it.

1. Task 1: Notebook 1
   - In the first task, parameters can be passed to Notebook 1 from the job or task itself.
   - Notebook 1 processes the parameters as input parameters param1 and sets new output parameters (e.g., output_param_2).
2. Task 2: Notebook 2
   - In the second task, Notebook 2 receives the output parameter from Task 1 by referencing it directly in the notebook code as shown above by passing the name of first task "GetSetParameter" as set in the workflow.
   - The value of output_param_2 is passed into Notebook 2 where it can be used for further processing.

This approach makes it easy to pass values dynamically between tasks and notebooks, allowing your workflows to be more flexible and adaptive.

Add Parameters to a Job

You can provide your jobs different parameters to track.

1. On the Home page, click **Workflow**.
2. Click the job you want add parameters to and click the **Details** tab.
3. Under Job Parameters, provide the key and value to track. To add multiple parameters, click **Add Parameter**.

   Changes you made are saved automatically.

Delete Parameters from a Job

You update your jobs to remove parameters that are no longer needed.

1. On the Home page, click **Workflow**.
2. Click the job you want remove parameters from and click the **Details** tab.
3. Under Job Parameters, click **Delete** next to the parameter you want to remove.

   Changes you made are saved automatically.

Add Parameters to a Task

You can add parameters to tasks to alter their behavior.

1. On the Home page, click **Workflow**.
2. Click the job that contains the tasks you want add parameters to and click the **Tasks** tab.
3. Click the task you want to add parameters to.
4. Under **Parameters**, enter the key and value for your parameter. To add multiple parameters, click **Add Parameter**.

   Changes you made are saved automatically.

Delete Parameters from a Task

You can remove parameters you don't need from your tasks.

1. On the Home page, click **Workflow**.
2. Click the job that contains the tasks you want delete parameters from and click the **Tasks** tab.
3. Click the task you want to delete parameters from.
4. Click **Delete** next to each parameter you want removed.

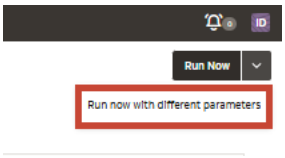
   Changes you made are saved automatically.

Run a Job with Different Parameter Values

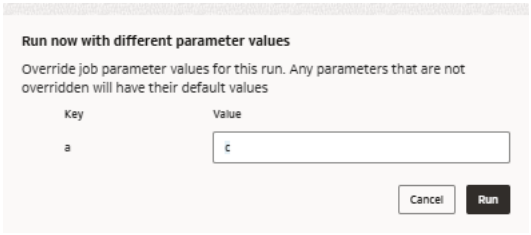
You can choose to run a job immediately with modified parameter values.

1. On the Home page, click **Workflow**.
2. Click on the job you want to run.

3. Click the down arrow next to **Run Now**. Click **Run now with different parameters**.



4. Enter new values. These parameter values only apply to this job run.



5. Click **Run**.

Monitor Jobs

This section covers how to track and search job runs in your AI Data Platform Workbench.

Topics:

About Jobs Runs

You can monitor the status of previous job runs, job runs in progress, or job runs that have been interrupted from the Job Runs page.

You can find a history of job runs in Job Runs tab of your job.

All jobs that have been run on your AI Data Platform Workbench are listed in the Job Runs page. The job run information is split into columns tracking different aspects of each job run. You can sort those columns in ascending or descending order by clicking the column header. You can filter the job runs displayed by using the dropdown menus, using the Search bar to filter by name, or combining multiple filter options.

Column name	Description
Job Name	Name of the job
Job Run Detail	Click **View** to see the details page of the job
Run as	User or role that triggered the run
Trigger	Trigger type. Either Manual or Scheduled
Duration	Length of time taken to run the job
Start time	Start time
End time	End time
Status	Current job status

The **Status** displays the current state of a job, using one of the values from the following list.

Status	Description
Success	Job run was successful
Failed	Job run failed
Canceled	Job run was canceled
Skipped	Job run was canceled because a previous run of the same job was already active
Timed Out	Job run exceeded the configured time limit and was stopped

View All Workflow Job Runs

You can see the recent history of your run workflow jobs from the **Job Runs** page.

1. On the Home page, click **Workflows**.
2. Click **Job Runs**.
3. Optional: Filter the results by selecting date ranges, run status, job name or any combination of filters to find the workflow job runs you need.

View a Job's Run History

You can see records of all previous runs for a job and filter the results for greater detail.

1. On the Home page, click **Workflow**.
2. Click the job you want to view the run history for.
3. Click the **Runs** tab.
4. Optional: Filter the results by selecting date ranges or run status from the dropdowns.

Monitor a Specific Job Run

You can track the status and history of a specific job run from the Job Runs page.

1. From the Job Runs page, click on **View Job Run Detail** for the job run you want to inspect.
2. Click the Details tab to review run-level metadata such as job parameters, compute configuration, schedule, and start/end time.
3. Use the buttons at the bottom of the Details page to see more specific information about the job run.
   - Click **View Details** to inspect compute configurations, like driver and worker shape.
   - Click **Spark UI** to inspect Spark stages, tasks, and resource usage for the run.
   - Click **Logs** to view driver and worker logs and review errors, warnings, and other runtime messages.
   - Click **Metrics** to monitor additional metrics related to compute, like CPU and memory usage.

To learn more about monitoring and troubleshooting activity at the compute level, see Monitor Compute.

13

Compute

This chapter covers the use of computing resources in AI Data Platform Workbench workspaces.

Topics:

AI Data Platform Workbench compute clusters are computing resources available in the workspace. You can use compute clusters to run workloads, such as data engineering, data science, and data analytics workloads.

You can use existing compute or, if you have the necessary permissions, create new compute to support your workloads. You can view the compute you have access to using the Compute section of the workspace.

Manage Compute

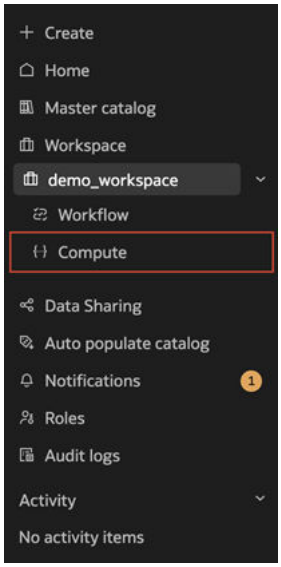
This section covers the basic functions of creating, changing, or removing compute clusters in your AI Data Platform Workbench.

Topics:

About Compute Clusters

All-purpose compute clusters provide you the compute resources to process your workloads in an AI Data Platform Workbench instance.

You manage your compute clusters from the Compute page in your Oracle AI Data Platform Workbench.



Types of Compute

Two types of compute exist in your AI Data Platform Workbench: all-purpose compute clusters and Default Master Catalog Compute Cluster.

You can only create all-purpose compute clusters in your AI Data Platform Workbench. All-purpose compute clusters are suitable for a versatile range of workloads and can be attached to your notebooks and used in workflows. Unless otherwise specified, any references to 'compute cluster' or 'cluster' in documentation refer to all-purpose compute clusters.

When you create a new all purpose compute cluster, you can choose either the Quickstart or Custom configuration. The Quickstart configuration is optimized to provide fast startup, while Custom configuration allows you to fine-tune your all purpose compute cluster to suit the specific workloads you need it to process. In both Quickstart and Custom configuration options, you can view cost projections and modify idle timeout options.

> **Note**
>
> Installing custom libraries to a Quickstart configured all purpose compute cluster automatically changes it to the Custom configuration. This can impact startup performance.

Default Master Catalog Compute Cluster is present in all AI Data Platform Workbench instances. This cluster is responsible for essential AI Data Platform Workbench functions, like search crawls, refreshing catalog objects, creating, editing, and deleting objects, and testing connections.

Cluster Runtime

All-purpose compute clusters can be created with an Apache Spark 3.5 runtime. The runtime environment is compatible with:

- Spark 3.5.0
- Delta 3.2.0 (pre-included)
- Python 3.11

- Scala 2.12
- Hadoop 3.3.4
- Java 17

Maintenance Updates for Compute Clusters

AI Data Platform Workbench compute automatically applies maintenance updates without user intervention. The maintenance updates cover any necessary security patches or bug fixes for operating system and AI Data Platform Workbench internal components.

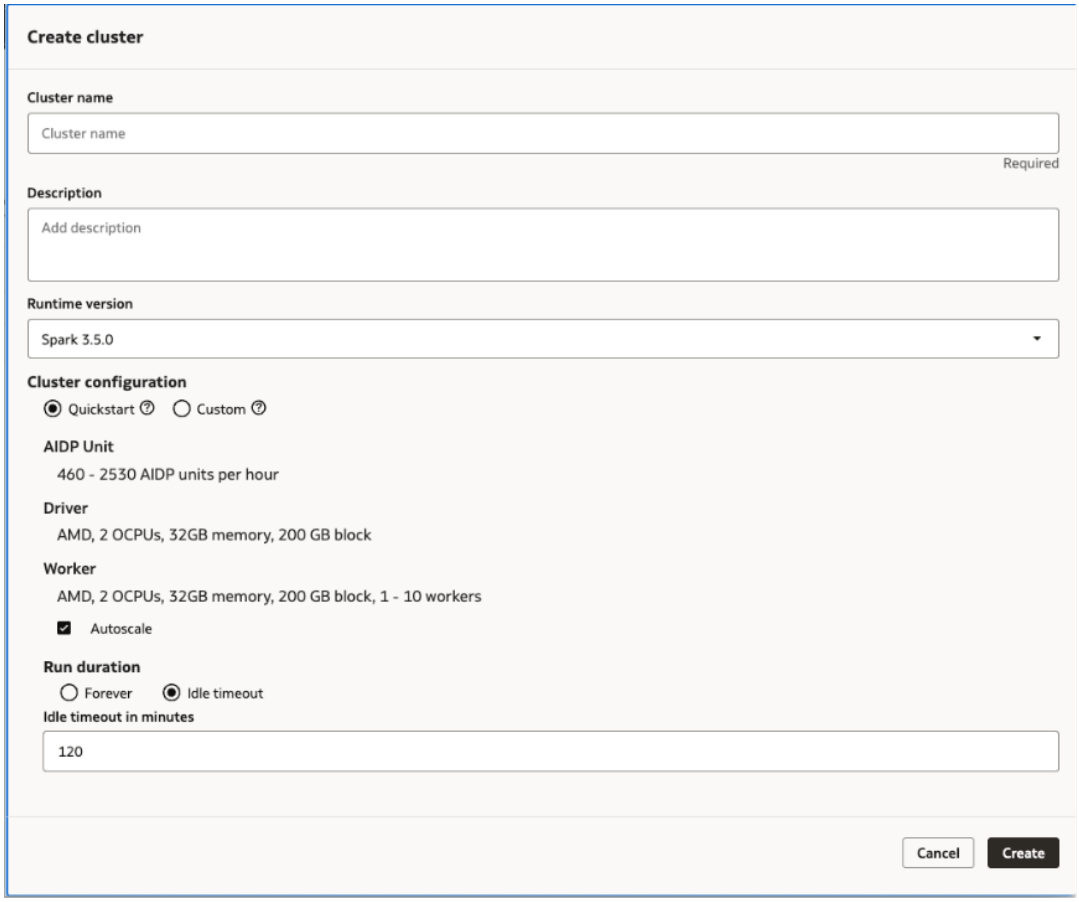
AI Data Platform Workbench verifies there are no running clusters before applying these monthly maintenance updates.

Create a Quickstart Cluster

You can choose to create an all-purpose compute cluster with preconfigured settings to process data and AI workloads in your AI Data Platform Workbench.

The quickstart configuration is an Apache Spark cluster with 1 driver and up to 10 workers, each with AMD 2 OCPU and 32 GB memory. Autoscaling is enabled by default for quickstart configuration. You can set your clusters to be constantly active or you can set an interval of inactivity after which the cluster will automatically stop (idle timeout). Stopped clusters will resume when called on by an attached workflow or notebook.
You can edit your cluster at any time after creation.

1. Click **Create** in the left navigation panel then click **Compute**. You can also navigate to your workspace and click **Compute**, then click **Create Cluster**.



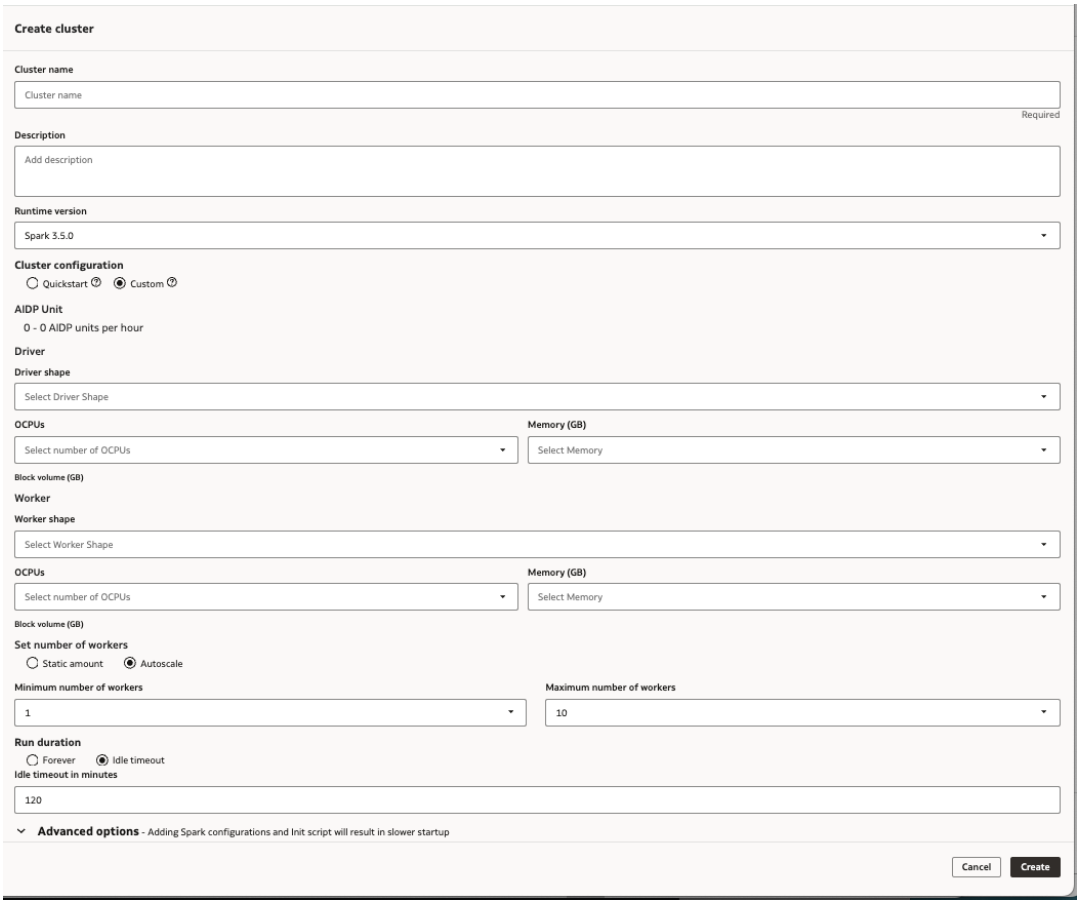
2. Provide a name and description to identify your cluster.
3. Select **Runtime version**.
4. Select **Quickstart** as your cluster configuration.
5. Select whether the number of workers is static or scales automatically. Autoscaling is enabled by default for quickstart configuration.
6. For **Run duration**, select whether the cluster will stop running after a set duration of inactivity. If **Idle timeout** is selected, specify the idle time, in minutes, before the cluster will time out.
7. Click **Create**.

Create a Custom Cluster

You can create an all-purpose compute cluster with configuration settings of your own choosing to process data and AI workloads in your AI Data Platform Workbench.

Custom clusters are intended for advanced users who want to leverage the full range of configuration options to suit their needs. You should select the driver and worker options that best fit the workloads you are going to process. You can set your clusters to be constantly active or you can set an interval of inactivity after which the cluster will automatically stop (idle timeout). Stopped clusters will resume when called on by an attached workflow or notebook. You can edit your cluster at any time after creation.

1. Click **Create** in the left navigation panel then click **Compute**. You can also navigate to your workspace and click **Compute**, then click **Create Cluster**.



2. Provide a name and description to identify your cluster.
3. Select **Runtime version**.
4. Select the driver options for your cluster.
5. Select the worker options for your cluster. These options apply to all cluster workers.
6. Select whether the number of workers is static or scales automatically.
   - If **Static amount**, specify the number of workers.
   - If **Autoscale**, specify the minimum and maximum number of workers the cluster can scale to.
7. For **Run duration**, select whether the cluster will stop running after a set duration of inactivity. If **Idle timeout** is selected, specify the idle time, in minutes, before the cluster will time out.
8. Click **Create**.

Create an NVIDIA GPU Cluster

You can choose to use an NVIDIA GPU in an All Purpose Compute Cluster to accelerate any workload in your unified AI and data pipeline.

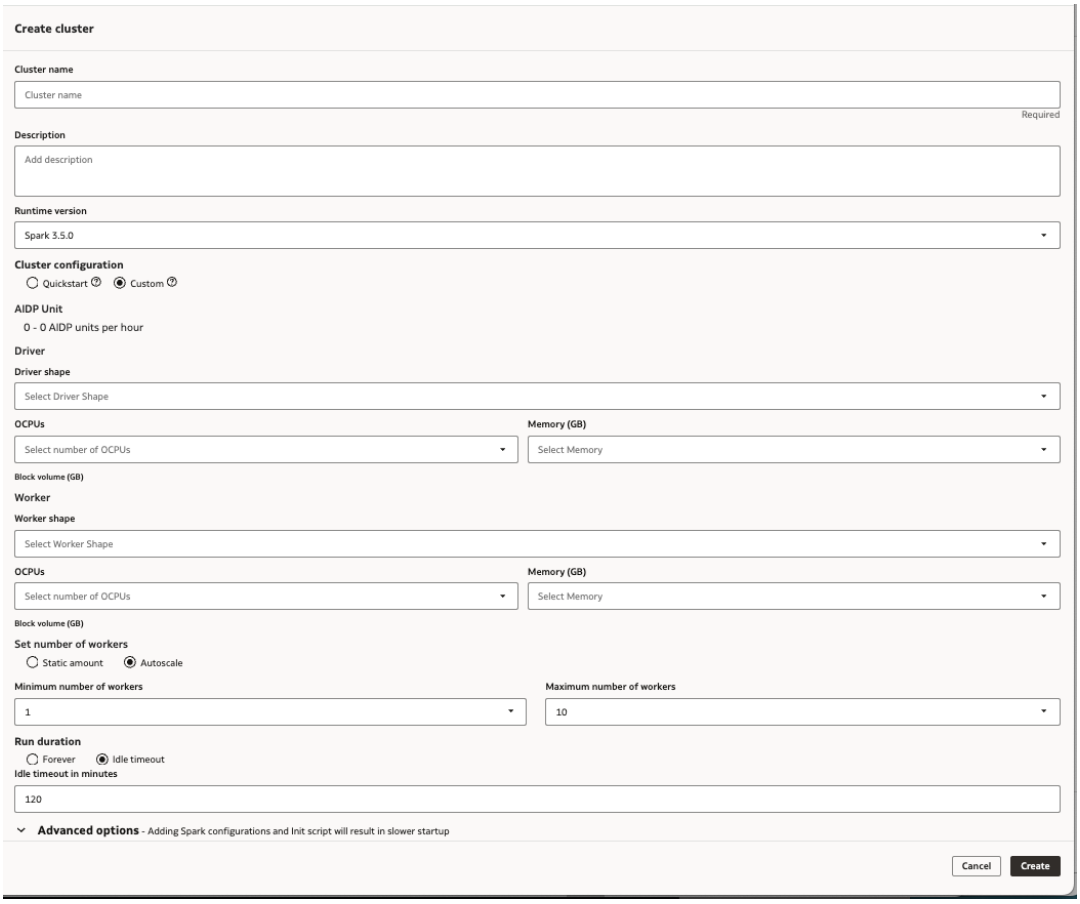
NVIDIA GPU shapes use the following configurations:

Table 13-1 NVIDIA GPU Shapes

GPU Count	OCPU	Block storage (GB)	GPU memory (GB)	CPU memory (GB)
1	15	1500	24	240
2	30	3000	48	480

> **Note**
>
> When you use NVIDIA GPU shapes, both the Driver and Worker shape must be an NVIDIA GPU. Mixing CPU and GPU shapes for the same cluster is currently not supported.

1. Click **Create** in the left navigation panel then click **Compute**. You can also navigate to your workspace and click **Compute**, then click **Create Cluster**.



2. Provide a name and description to identify your cluster.
3. Select **Runtime version**.
4. Select **Custom** as your cluster configuration.
5. For your cluster driver options:

- Select **NVIDIA GPU** as the Driver Shape.
- Select 1 or 2 as the GPU count.

6. For your cluster worker options:
   - Select **NVIDIA GPU** as the Worker Shape.
   - Select 1 or 2 as the GPU count.
7. Select whether the number of workers is static or scales automatically.
   - If **Static amount**, specify the number of workers.
   - If **Autoscale**, specify the minimum and maximum number of workers the cluster can scale to.
8. For **Run duration**, select whether the cluster will stop running after a set duration of inactivity. If **Idle timeout** is selected, specify the idle time, in minutes, before the cluster will time out.
9. Click **Create**.

NVIDIA GPU Cluster Tuning

You can tune your NVIDIA GPU clusters to optimize their performance by using recommendations from the GPU provider and by installing optional libraries.

Tuning GPU clusters can help optimize the performance of those clusters when called on by jobs in your AI Data Platform Workbench.

For NVIDIA GPU-based clusters, you can follow NVIDIA's Tuning Guide for recommendations and steps you can take to optimize performance.

You also have the option of installing Spark RAPIDS libraries to assist with optimization:

- Spark RAPIDS library is a RAPIDS accelerator for Apache Spark and provides a set of plugins that leverage GPUs to accelerate processing.
- Spark RAPIDS ML library enables GPU-accelerated, distributed machine learning on Apache Spark and provides several PySpark ML compatible algorithms powered by the RAPIDS cuML library.

The Spark RAPIDS library is commonly used first for feature engineering and data cleaning, and then cross validation is performed at scale using the Spark RAPIDS ML library. You can use these libraries for use cases like fraud detection (time series), web clickstream, and A/B experimentation.

Table 13-2 Recommended Spark Configurations

Setting	Value	Note
spark.executor.instances	4	Number of worker x GPU count per worker If the number of workers is 4, and GPU count per worker is 1, then recommended spark.executor.instances config is 4 x 1 = 4
spark.executor.cores	16	GPU count/ worker / CPU cores, maximum of 16

Table 13-2 (Cont.) Recommended Spark Configurations

Setting	Value	Note
spark.executor.memory	32 GB	2GB / core or 80% of CPU memory / GPU count per worker (whichever is less)
spark.task.resource.gpu.amount	0.0625	1 / spark.executor.cores
spark.rapids.sql.concurrentGpuTasks	3	GPU memory / 8GB, maximum of 4
spark.rapids.shuffle.multiThreaded.writer.threads	32	CPU cores / GPU count per worker
spark.rapids.shuffle.multiThreaded.reader.threads	32	CPU cores / GPU count per worker
spark.shuffle.manager	com.nvidia.spark.rapids.spark350.RapidsShuffleManager	-
spark.rapids.shuffle.mode	MULTITHREADED	-
spark.plugins	com.nvidia.spark.SQLPlugin	-
spark.executor.resource.gpu.amount	1	-
spark.sql.files.maxPartitionBytes	2 GB	Optional, recommended for large datasets
spark.rapids.sql.batchSizeBytes	2 GB	Optional, recommended for large datasets
spark.rapids.memory.host.spillStorageSize	32 G	Optional, recommended for large datasets
spark.rapids.memory.pinnedPool.size	8 G	Optional, recommended for large datasets
spark.sql.adaptive.coalescePartitions.minPartitionSize	32 MB	Optional, recommended for large datasets
spark.sql.adaptive.advisoryPartitionSizeInBytes	160 MB	Optional, recommended for large datasets
spark.rapids.filecache.enabled	True	Optional, recommended if workloads will be reusing datasets

Modify a Cluster

You can change settings or add additional parameters for your clusters.

1. Navigate to your workspace and click **Compute**.
2. Next to the compute cluster you want to modify, click ⋯ **Actions** then click **Edit**.
3. Modify the attributes of your compute cluster or add additional parameters as needed.
4. Click **Save**.

Delete a Cluster

You can delete compute clusters that are unused or no longer needed.

1. Navigate to your workspace and click **Compute**.
2. Next to the cluster you want to delete, click ⋯ **Actions** and click **Delete**.

3. Click **Delete**.

View Cluster Details

You can review the shape and settings of a cluster at any time.

1. Navigate to your workspace and click **Compute**.
2. Click the name of the cluster you want to view details for.
3. Click the **Details** tab.

Maintenance Updates for Compute Clusters

Oracle AI Data Platform compute automatically applies maintenance updates without user intervention.

The maintenance updates cover any necessary security patches or bug fixes for operating system and AI Data Platform internal components. AI Data Platform verifies there are no running clusters before applying these monthly maintenance updates.

Connect to Compute

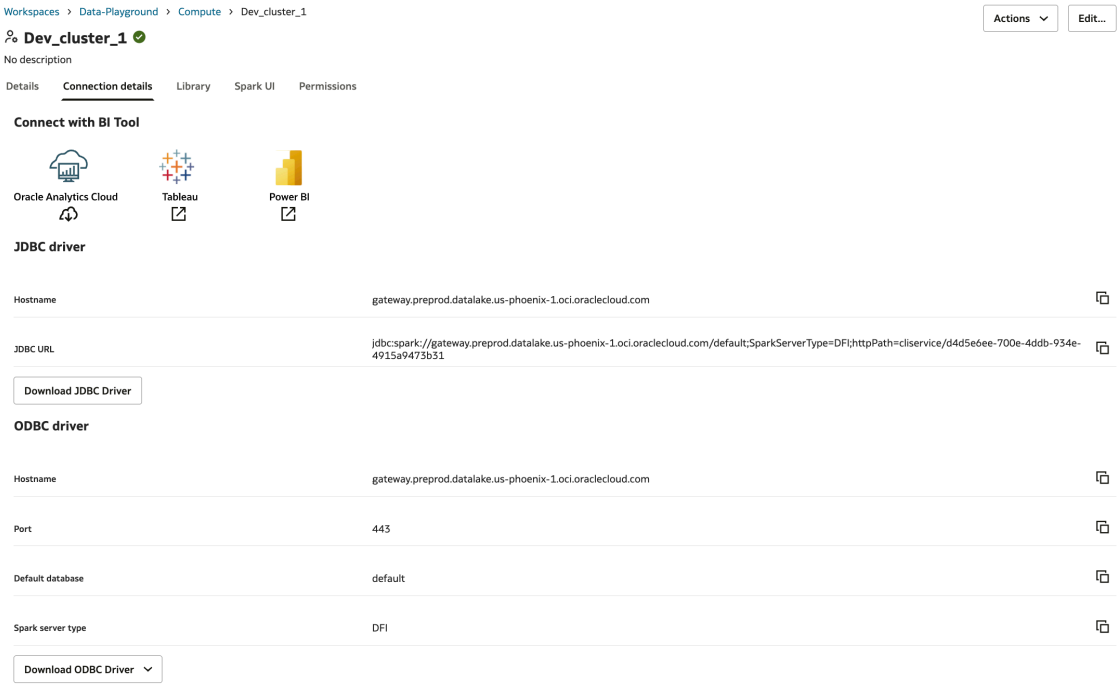
This section covers connecting compute in your AI Data Platform Workbench to other business intelligence tools.

Topics:

Connections

You can connect your AI Data Platform Workbench with Oracle Analytics Cloud or other business intelligence tools.

You can connect to Oracle AI Data Platform Workbench from different business intelligence tools using a custom JDBC or ODBC provided by the AI Data Platform Workbench. AI Data Platform Workbench also supports connections from third party commercial BI tools (Tableau, Power BI) or open source BI tools like DBeaver. In order to connect from these tools, you need the connection details of the compute cluster in your AI Data Platform Workbench, which you can find in the Connection details tab. The required information varies depending on the product you are connecting from.



Connect Oracle Analytics to AI Data Platform Workbench

You can connect to the catalog or tables managed by your AI Data Platform Workbench instance from an Oracle Analytics Cloud instance.

Video

In order to create a connection from Oracle Analytics to Oracle AI Data Platform Workbench, you need to do the following:

1. Get an API Key from OCI.
2. Get an Oracle Analytics Connection Configuration File.
3. Combine the API key and configuration file to create the connection. See Create an Oracle Analytics Connection File.
4. Use the connection in Oracle Analytics Cloud. See Connect Oracle AI Data Platform Workbench to Oracle Analytics.

You can perform step 1 and 2 in any order, but they must both be performed before continuing.

Oracle recommends creating a user in OCI that is dedicated to the Oracle Analytics use-case. That user needs to have appropriate permissions to the tables in your AI Data Platform Workbench that you want to access from Oracle Analytics. Generate the needed API key in OCI and download the PEM file for the user created for this purpose.

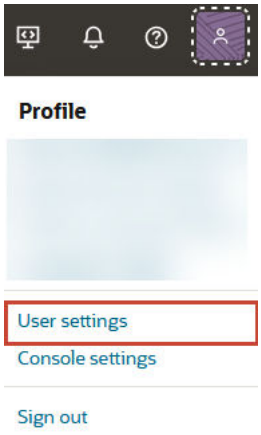
> **Note**
>
> - To connect to your AI Data Platform Workbench from Oracle Analytics, **you must provide the catalog name during connection setup.**
> - The catalog is currently necessary for the system to locate the correct schema and data objects within the AI Data Platform cluster.
> - This requirement will be alleviated in a future release.

Get an API Key from OCI

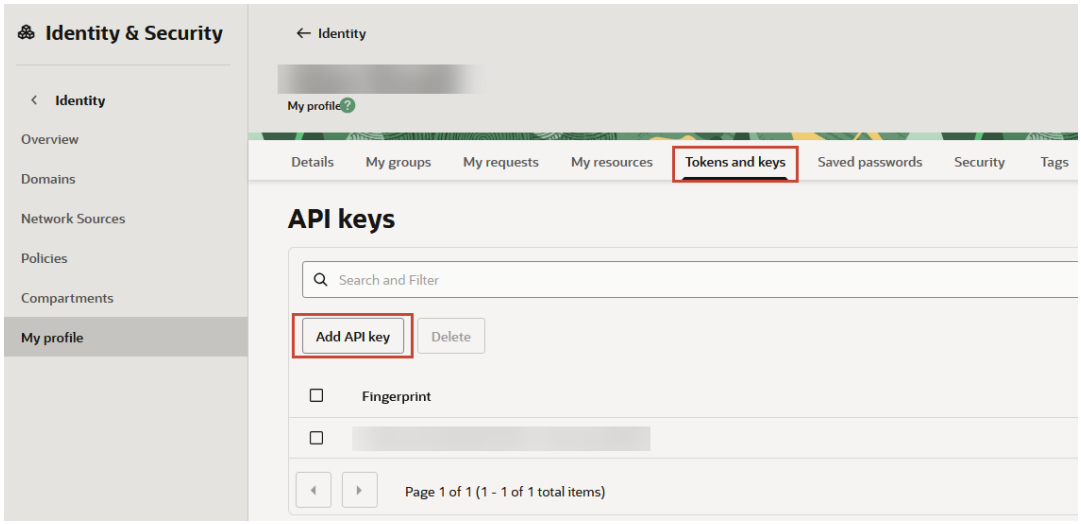
You need an API key from OCI for a user that has access to your Oracle AI Data Platform Workbench when creating a connecting to Oracle Analytics.

The user you create the API key for must be the creator of the AI Data Platform Workbench instance or have Admin permissions for the instance. You must also have a valid AI Data Platform Workbench cluster in the instance.

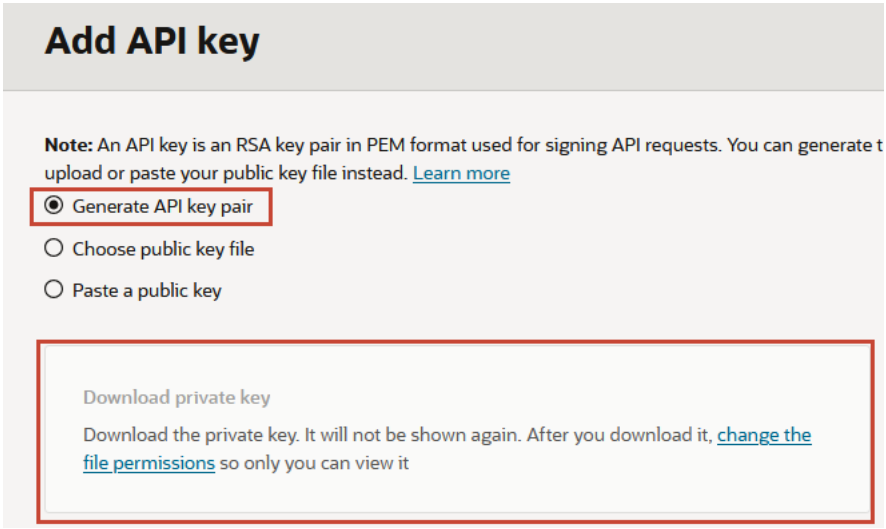
1. In OCI, click the user icon, then click **User Settings**.



2. In My profile, click **Tokens and keys**, then click **Add API key**.



3. Select **Generate API key pair**, then click **Download private key**.



4. Click **Add**.

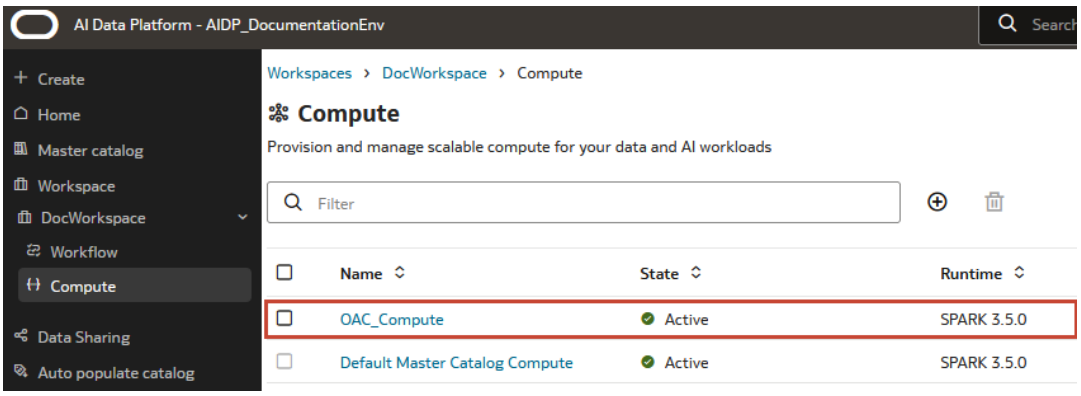
Save the private key (.pem file) you receive to a secure and persistent location. It is required for connections and can only be downloaded once. This file is used as your fingerprint when connecting Oracle Analytics to your AI Data Platform Workbench.

Get an Oracle Analytics Connection Configuration File

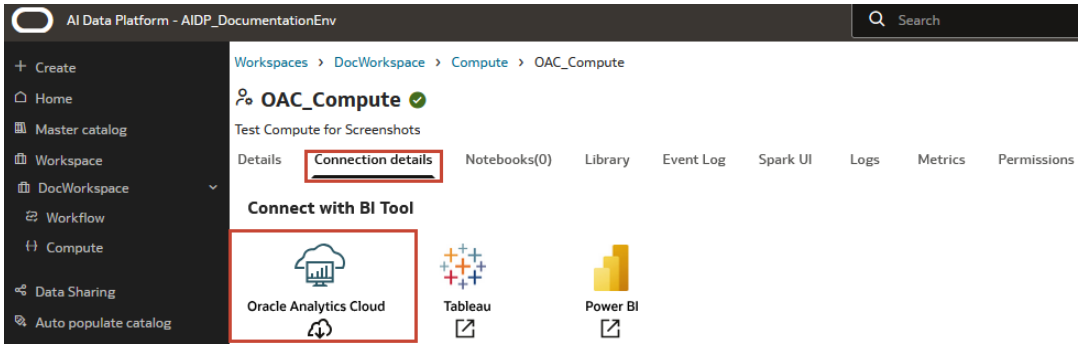
You need to download a config.json file from your Oracle AI Data Platform Workbench instance to connect it to Oracle Analytics.

You need a compute cluster with at least 2 OCPUs, 32 GB of memory for both driver and worker nodes, and at least 2 workers.

1. Navigate to your workspace and click **Compute**.



2. Click the cluster you want to connect to Oracle Analytics.
3. In the **Connection details** tab, click the **Download** icon underneath the Oracle Analytics Cloud logo. A `config.json` file is downloaded to your machine.



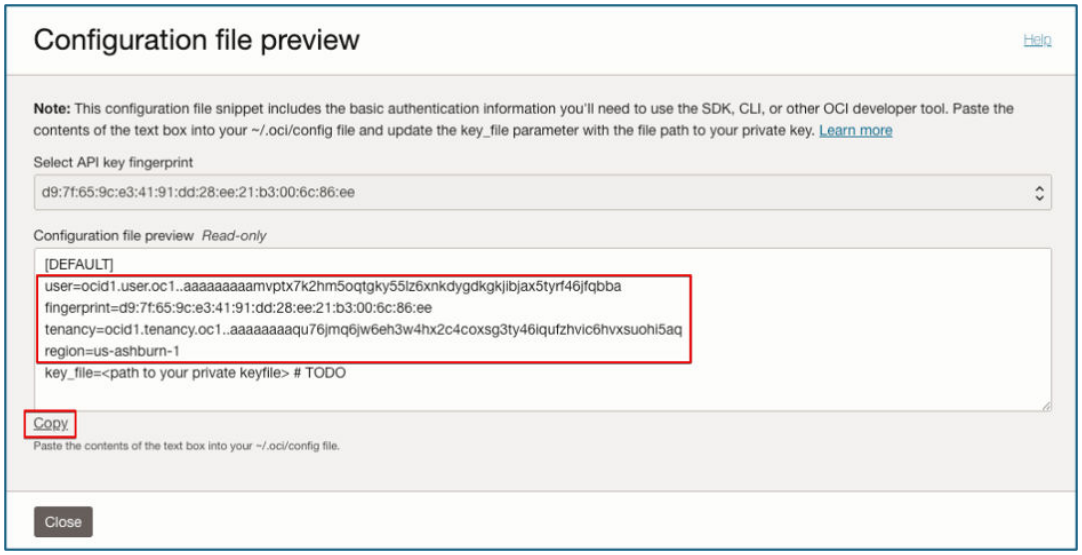
The `config.json` file you've downloaded is missing the fingerprint key value pair. It must be combined with the fingerprint key value from an API key belonging to an OCI user that created the AI Data Platform Workbench or a user with Admin permissions.

Create an Oracle Analytics Connection File

You need to manually add the fingerprint key value pair to the `config.json` file downloaded from your Oracle AI Data Platform Workbench so the file can be used to connect to Oracle Analytics.

You need to have created an API key for an OCI user that created the AI Data Platform Workbench instance or has Admin permissions for the instance. You also need to have downloaded the `config.json` from the cluster you want to connect to Oracle Analytics.

1. Open the `config.json` file and the API key's configuration file.
2. Edit the `config.json` to add the fingerprint value from the API key configuration file.



3. Save your changes to the `config.json` file.

```
{
  "username": "ocid1.user.oc1..aaaaaaaaamvptx7k2hm5oqtqky551z6xnkdyqdkqkjibjax5tyrf46jfqbba",
  "tenancy": "ocid1.tenancy.oc1..aaaaaaaaaqu76jmq6jw6eh3w4hx2o4coxsg3ty46iqufzhvic6bvxsuohi5aq",
  "region": "us-ashburn-1",
  "fingerprint": "96:1e:f8:5f:75:22:32:54:09:90:66:b7:18:ca:39:61",
  "dsn": "jdbc:spark://gateway.datalake.us-ashburn-1.oci.oraclecloud.com/default;SparkServerType=IDL;httpPath=cliservice/d5bba507-940a-407b-bd59-667cdb3efd70"
}
```

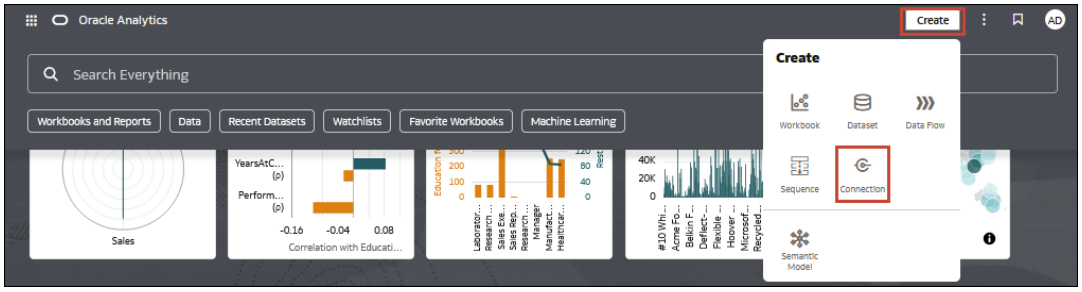

In a future release, the `config.json` will include the fingerprint of the API key's public key and you will no longer have to manually add it. In that release, the latest API key available to the user will be used in the `config.json`.

Connect Oracle AI Data Platform Workbench to Oracle Analytics

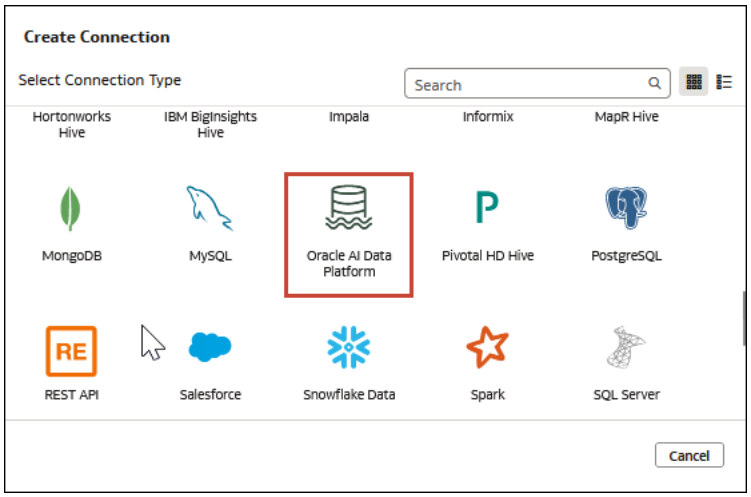
You can connect to the catalog or tables managed by your AI Data Platform Workbench instance from an Oracle Analytics instance.

Oracle Analytics connections to AI Data Platform Workbench instances only support one catalog per connection. You need to create additional connections to use more than one catalog.

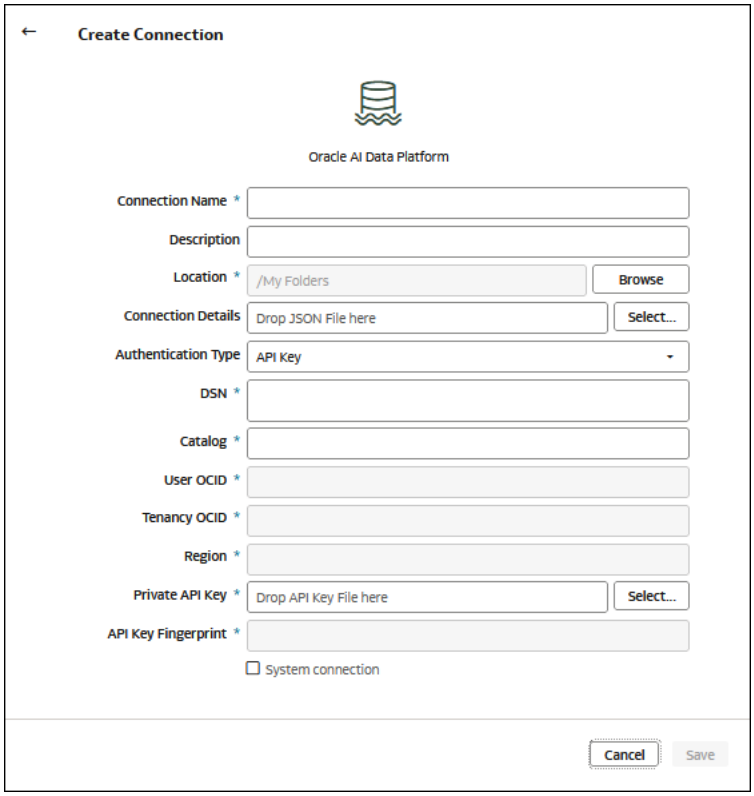
1. On the Oracle Analytics home page, click **Create** then click **Connection**.



2. Click **Oracle AI Data Platform**.



3. Enter a name and description for your connection.



4. For Connection Details, click **Select**, navigate to your config.json file, and click **Open**. For more information, see Create an Oracle Analytics Connection File.
5. Select **API Key** as the authentication type.
6. Enter the Catalog name to be used in your datasets.
7. For Private API Key, click **Select**, navigate to your private API key (.PEM file), and click **Open**. For more information, see Get an API Key from OCI.
8. Click **Save**.
9. Test your new connection by creating a new dataset. For more information, see Create a Dataset from a Connection.

If you have followed the instructions and are still **having issues saving your connection or listing your schemas**, a cluster restart may be required.

Download JDBC Driver

You can download the JDBC driver from the Connection details tab and configure it for different BI tools.

1. Navigate to your workspace and click **Compute**.
2. Click the cluster you want to connect to a JDBC compatible BI tool and click **Connections**.
3. Click **Download JDBC Driver**.

Configure DBeaver

You need to install DBeaver and prepare files downloaded from Oracle AI Data Platform Workbench before you can connect it to a compute cluster in your AI Data Platform Workbench.

1. Download and install the DBeaver client. You can use either Community or Enterprise versions, but they must be at least 22.x.DBeaver is only available for Windows, Mac OS X, Eclipse Plugin, and Linux.
2. Unzip the file from Download JDBC Driver.
3. From the unzipped files, unzip the `simbaSpark.zip` driver file.

Configure DBeaver with the Spark Simba JDBC Driver

You can configure DBeaver to connect to a compute cluster in your Oracle AI Data Platform Workbench.

You must have installed DBeaver and extracted the simbaSpark.zip files downloaded from AI Data Platform Workbench.

1. Open DBeaver.
2. Click **Database Navigator**.
3. Click **Driver Manager**.
4. Click **New**.
5. Enter `AI Data Platform` as the **Driver Name**.
6. Click **Libraries**.
7. Click **Add Folder**.
8. Browse to the location of the sparkSimba.zip extract.
9. Click **Find Class**.
10. Select **OK**.
11. Set **Driver class** to `com.simba.spark.jdbc.Driver`.
12. Click **Settings**.
13. Set **Class Name** to `com.simba.spark.jdbc.Driver`.
14. Click **OK**.

Create a Database Connection in DBeaver

To connect DBeaver to a compute cluster in your Oracle AI Data Platform Workbench, you need to first create a database connection in DBeaver.

You must have configured DBeaver with the Spark Simba JDBC driver downloaded from AI Data Platform Workbench. For more information, see Configure DBeaver with the Spark Simba JDBC Driver.

1. Open DBeaver.
2. Click **Database**.
3. Click **New Database Connection**.

4. Click **All**.
5. Select **AI Data Platform**.
6. Click **Next**.
7. Enter the URL of the JDBC driver. You can find the JDBC URL on the **Connection details** tab of your compute cluster in AI Data Platform Workbench.
8. Click **Finish**.

Connect DBeaver to Oracle AI Data Platform Workbench using JDBC

Once DBeaver is configured and has a database connection to your AI Data Platform Workbench, you can complete the connection between DBeaver and AI Data Platform Workbench.

1. Open DBeaver.
2. Click **Connect**.
3. Choose to connect with an authorization token or an API key.
   - **Connect using authorization token**
     - Use a token by not specifying any profile in the URL if you don't have a DEFAULT profile. For example: `jdbc:spark://gateway.aidp.me-riyadh-1.oci.oraclecloud.com/default;SparkServerType=AIDP;httpPath=cliservice/cf18b4ef-b83e-41dd-82b6-8d391584f6c5`
       The URL opens a browser window.

       Sign in to the tenancy where the AI Data Platform Workbench instance is created.

       For more information, see Token-based Authentication for the CLI.
   - **Connect using an API key,** by specifying the OCI Profile with `ociProfile=<profile_name>` in the connection URL.
     - Use API key authentication to connect to an AI Data Platform Workbench instance. Use API key by specifying the OCI Profile with `ociProfile=<profile_name>` in the connection URL. For example, to use OCI Profile name Demo: `jdbc:spark://gateway.aidp.me-riyadh-1.oci.oraclecloud.com/default;SparkServerType=AIDP;httpPath=cliservice/cf18b4ef-b83e-41dd-82b6-8d391584f6c5 ;ociProfile=Demo`

       For more information, see Required Keys and OCIDs.
4. DBeaver creates a connection for reading metadata, and a connection for all the other operations. If you're limited for connections, you can disable the second one so that DBeaver uses one connection for all operations.
   a. Click **Preferences**.
   b. Click **Common**.
   c. Click **Metadata**.
   d. Deselect **Open separate connection for metadata reads**.

Connect Tableau to AI Data Platform using JDBC

You can connect data in your AI Data Platform to Tableau to use with its analytics and visualizations.

1. Navigate to your workspace and click **Compute**.
2. Click the cluster you want to connect to Tableau and click the Connection Details tab.
3. Click **Download JDBC Driver**.
4. Extract the files from the downloaded .zip file.
5. Open the main folder from the extracted files and extract the Simba JAR file to following folder depending on your OS type:
   - Linux: `/opt/tableau/tableau_driver/jdbc`
   - MacOS: `~/Library/Tableau/Drivers/`
   - Windows: `C:\Program Files\Tableau\Driver`

   For example, in MacOS, the JAR file location would be `~/Library/Tableau/Drivers/SparkJDBC42.jar`.
6. Open the Tableau client.
7. In the **Connect To a Server** list, search for JDBC databases.
8. Copy the JDBC URL.
9. Select **SQL92** for Dialect.
10. Authenticate your connection with one of the following methods:
    - Authenticate with an API key
      - Use the default OCI configuration file and the default profile. For example:

        ```
        jdbc:spark://gateway.datalake.uk-london-1.oci.oraclecloud.com/
        default;SparkServerType=AIDP;httpPath=cliservice/1ee500ba-faad-4267-
        adcb-b6c7ce08d5a0
        ```

      - OR, if not using the default location, append the OCI configuration file location in the connection JDBC URL. For example:

        ```
        jdbc:spark://gateway.datalake.uk-london-1.oci.oraclecloud.com/
        default;SparkServerType=AIDP;httpPath=cliservice/1ee500ba-faad-4267-
        adcb-b6c7ce08d5a0;OCIConfigFile=<config_file_absolute_path>
        ```

      - OR, if not using the default profile, append the OCI profile in the connection JDBC URL. For example:

        ```
        jdbc:spark://gateway.datalake.uk-london-1.oci.oraclecloud.com/
        default;SparkServerType=AIDP;httpPath=cliservice/1ee500ba-faad-4267-
        adcb-
        b6c7ce08d5a0;OCIConfigFile=<config_file_absolute_path>OCIProfile=<pr
        ofile_name>
        ```

    - Authenticate using an authorization token:
      - To use an authorization token, do not have a config file in the default path (for example, `/Users/xyz/.oci/config`) and do not specify any profile in the URL. For example:

        ```
        jdbc:spark://gateway.datalake.uk-london-1.oci.oraclecloud.com/
        default;SparkServerType=AIDP;httpPath=cliservice/1ee500ba-faad-4267-
        adcb-b6c7ce08d5a0
        ```

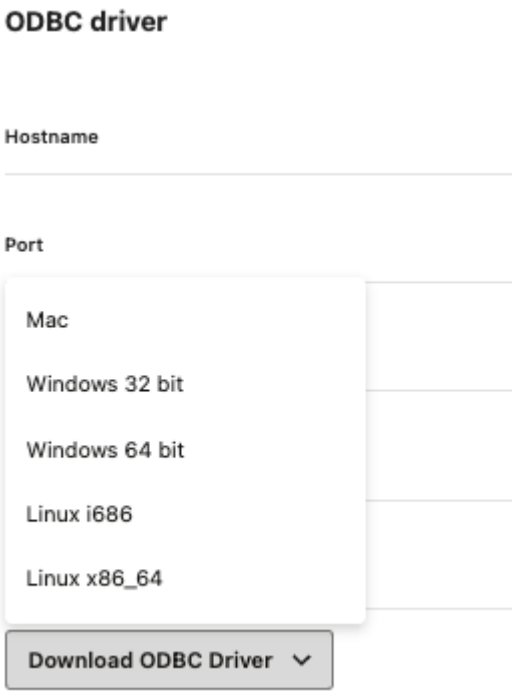
For more information, see Token-based Authentication for the CLI.

11. Click **Sign In**.
12. Optional: If you are using token-based authentication, sign in to the tenancy where your AI Data Platform was created in the browser window that appears.

Download ODBC Driver

You can download the ODBC driver from the Connection details tab and configure it for different BI tools.

1. Navigate to your workspace and click **Compute**.
2. Click the cluster you want to connect to an ODBC compatible BI tool and click **Connections**.
3. Click **Download ODBC Driver**.
4. Select the appropriate OS from the list.



Manage Libraries

This section covers how to use and manage libraries connected to your compute clusters.

Topics:

- Libraries
- Notebook Scoped Libraries
- Install a Library from a Workspace or Volume
- Install a Library from an Uploaded File

- Uninstall a Library
- Notebook Scoped Libraries

Libraries

You can add cluster scoped libraries to make third-party or custom code available to your compute clusters while running notebooks or workflow jobs.

Cluster scoped libraries can be installed to extend the out of the box capabilities of compute clusters and applies to all notebooks and workflow jobs using that cluster. For example, visualization options, connectivity options (e.g. JDBC JARs), extractions (e.g. extracting text from PDF) or transformations.

The option for installing cluster scoped libraries is available in the Library tab of your cluster after the cluster status changes to Active. Your library file should be a .jar file or a Wheel (*.whl) file or a requirements.txt file.

You can also add initialization scripts during the creation of a cluster or by modifying an existing cluster. For more information, see Modify a Cluster.

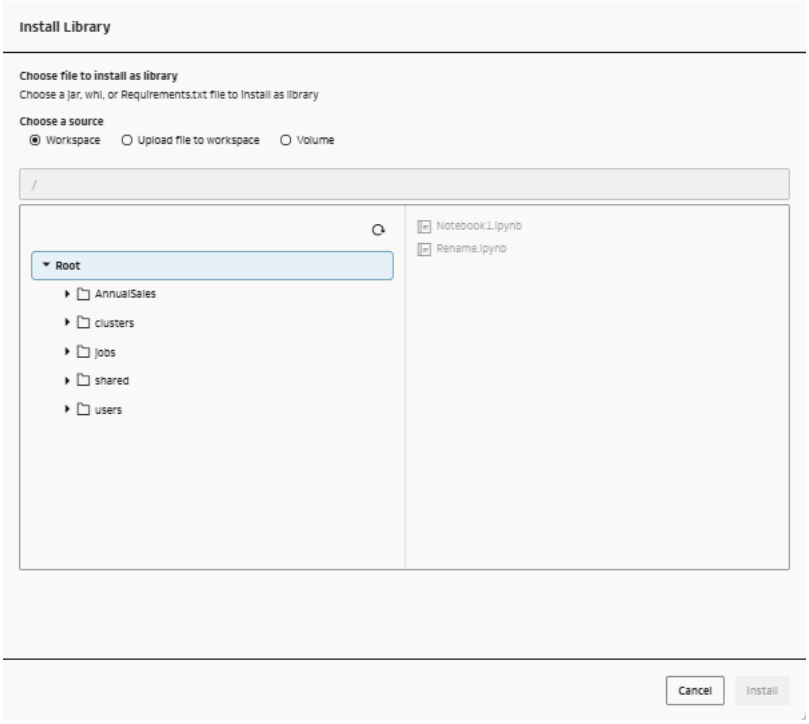
Install a Library from a Workspace or Volume

You can install a library that is in your workspace or volume to expand cluster options for attached notebooks and jobs.

Libraries can only be added from a workspace or a volume where you have appropriate permissions. You can view libraries that are installed on a cluster at any time from the cluster's Library tab.

If the library file you want to install is not already available in your workspace or volume, you can upload the library from your local machine to your workspace first and then install at the cluster.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Library** tab.
3. Click ⊕ **Install Library**.
4. Select whether your library is part of a **Workspace** or **Volume**.



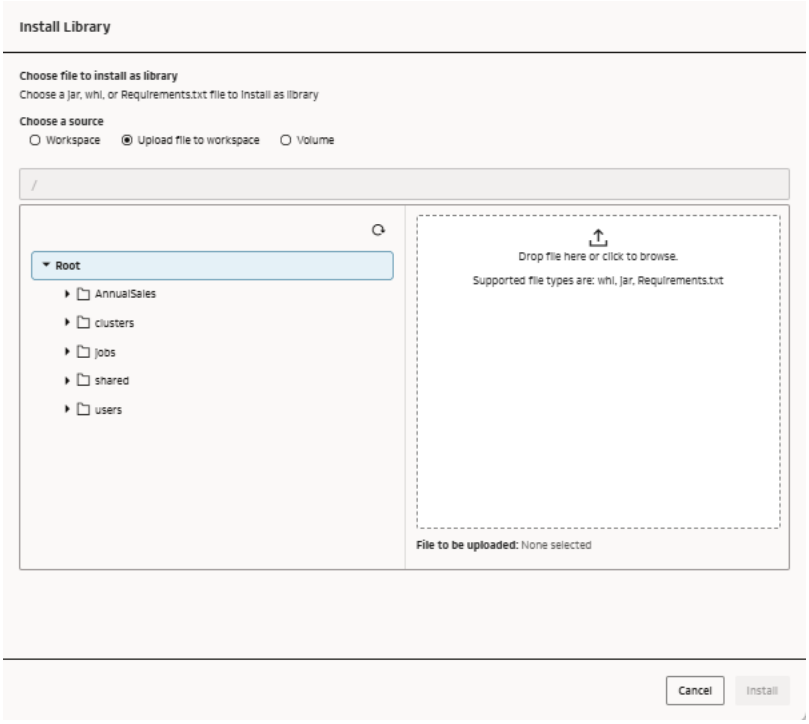
5. Navigate to the library and select it. Click **Install**.
6. Once the library is installed, restart the cluster by clicking **Actions**, then **Restart**.

   When the cluster status is Active again, you can use the library in your code inside a notebook or workflow job.

Install a Library from an Uploaded File

You can install a library to your workspace from an uploaded file to expand cluster options for attached notebooks and jobs.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Library** tab.
3. Click ⊕ **Install Library**.
4. Select **Upload file to workspace**.



5. Browse to the file that contains your library or drag and drop it into the window.

   Your library file must be a .whl or .jar format or a text file with the name `requirements.txt`. For more information on the requirements.txt file, see Requirements File Format.
   Here is an example of a requirements.txt file:

   ```
   plotly==6.0.1
   pandas==2.2.3
   matplotlib==3.10.1
   ```

6. Click **Install**.
7. Once the library is installed, restart the cluster by clicking **Actions**, then **Restart**.

   When the cluster status is Active again, you can use the library in your code inside a notebook or workflow job.

Uninstall a Library

You can uninstall an unwanted or no longer needed library from clusters you own.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Library** tab.
3. Next to the library, click ··· **Actions** then click **Uninstall**.
4. Click **Uninstall**.

Notebook Scoped Libraries

You can use notebook scoped libraries in Oracle AI Data Platform Workbench for creating custom Python environments specific to a Notebook.

Notebook scoped libraries are not cluster scoped, so they do not apply to other notebooks sharing the same cluster. Notebook scoped libraries only apply to current notebook and all workflow jobs that use the current notebook as 'Notebook tasks'.

> **Note**
>
> `!pip`-based commands are currently only supported in notebooks with `*.ipynb` extensions. Python files with `*.py` extensions are not supported and installing JAR files as notebook scoped libraries is not supported.

Supported Commands

You can use the following commands for managing your notebook scoped libraries:

Install and uninstall a package by using the package name:

```
!pip install <package>
!pip uninstall <package>
```

Install and uninstall all the libraries from a requirements.txt file:

```
!pip install -r /Workspace/path/to/requirements.txt
!pip uninstall -r /Workspace/requirements.txt
```

Install and uninstall wheel files with *.whl extension from a workspace path:

```
!pip install /Workspace/path/to/my-package.whl
!pip uninstall /Workspace/path/to/my-package.whl
```

List the already installed notebook scoped libraries in the current notebook:

```
!pip list
```

Monitor Compute

This section explains the different methods and metrics you can use to monitor compute in your Oracle AI Data Platform Workbench.

Topics:

View Spark UI

You can view the Spark Web UI to see to monitor the status and resource consumption of your all-purpose compute clusters.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Spark UI** tab.
3. Optional: Click to pop-out button on the top right to view the Spark UI in a separate window.

View Driver and Worker Logs

You can view the Driver and Worker Logs of your All Purpose Compute Clusters for troubleshooting or debugging.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Logs** tab.
3. Filter your logs to see more specific information.

4. Click **Download** to save a local copy of your filtered data.

View Metrics

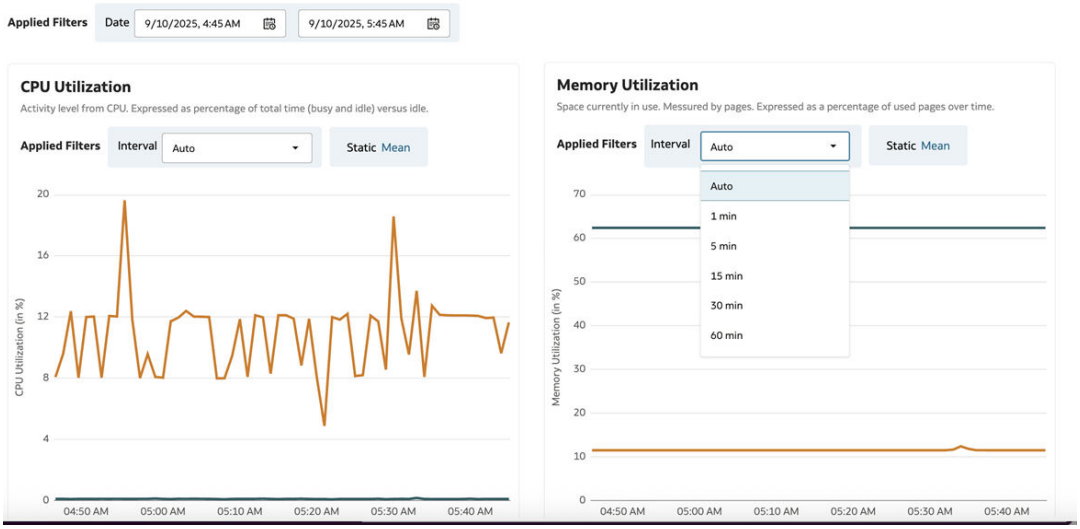
You can monitor the infrastructure metrics of your compute clusters for troubleshooting or for making any sizing adjustments.

You can view status and history for the following metrics:

- CPU Utilization
- Memory Utilization
- Disk read
- Disk write
- File system utilization
- Garbage Collector CPU utilization
- Network received
- Network transmitted
- Active tasks
- Total failed tasks
- Total task tasks
- Total completed tasks
- Total number of tasks

- Total shuffle read bytes
- Total shuffle write bytes
- Total task duration in seconds
- SQL: Peak concurrent queries
- SQL: Peak concurrent connections

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Metrics** tab.



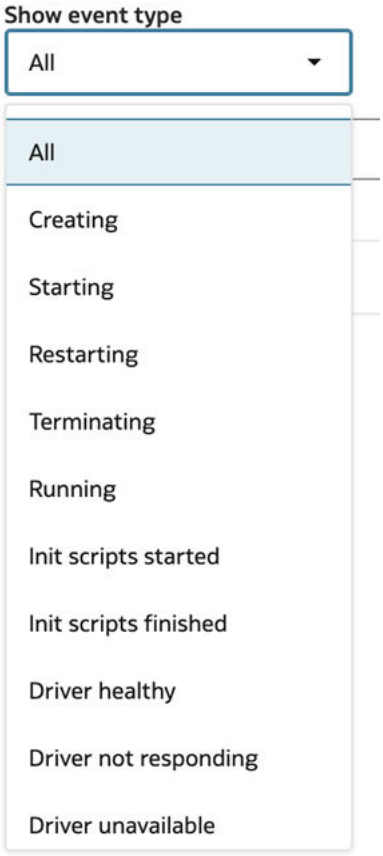
3. Select time frames using the **Date** filter to view metrics over a specific period.
4. Select an option from the **Interval** dropdown to filter information for a specific metric.

View Event Logs

You can view the Event Logs to monitor different cluster related operations, like creation of clusters, restarts of clusters, init script execution, or monthly maintenance updates.

Oracle AI Data Platform Workbench retains the last 14 days of event logs.

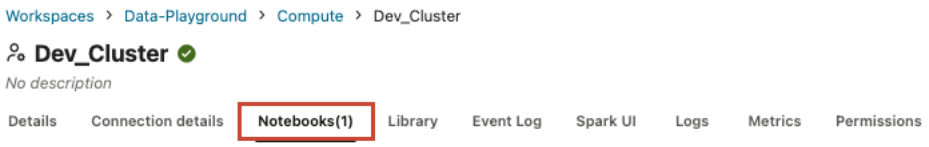
1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Event Logs** tab.
3. Filter your logs to see more specific information.



View Notebooks

You can view all the notebooks the current cluster is attached with. This view includes notebook count, notebook status, and provides you a quick way to navigate to the appropriate notebooks.

1. Navigate to your workspace and click **Compute**.
2. Click your cluster, then click the **Notebooks** tab.



   The notebook state is **Active** if code is running from that notebook. The notebook state is **Idle** if no code is running from that notebook.
3. Click the name of a notebook to go to it.

14

Ingest Data

This chapter explains how AI Data Platform Workbenches can ingest data from different internal and external sources.

Topics:

Oracle AI Data Platform Workbench enables seamless ingestion of data from both external and internal sources using Spark-based notebooks. Whether you're pulling data from cloud services, on-premise databases, or Oracle-native platforms, AI Data Platform Workbench provides flexible, code-driven ingestion methods that support your data engineering workflows at scale.

With AI Data Platform Workbench's ingestion connectors, you can:

- Use drag and drop features to generate notebook code for fast connection setup.
- Ingest batch or near real-time data from a wide variety of systems.
- Leverage Spark and JDBC-based patterns to read, write, and process data efficiently.
- Register external sources as catalogs for direct querying without duplication.

Explore the following sections to learn more:

- **External Sources -** Ingest data from MySQL, PostgreSQL, Kafka, and more.
- **Internal Sources -** Connect with Oracle-native systems like Oracle Autonomous AI Lakehouse, Fusion via BICC, and Oracle Database/Oracle AI Database.

Internal Sources

Oracle AI Data Platform Workbench supports ingestion from internal Oracle sources using built-in ingestion connectors. These connectors enable users to seamlessly extract data using Spark-based notebooks and integrate it into their workflows and data pipelines.

Ingestion connectors abstract the complexities of connection setup, providing optimized access patterns for both batch and near real-time ingestion from Oracle-native services.

AI Data Platform Workbench provides sample code templates in the Oracle AI Data Platform Workbench Samples Git repository to support ingesting data from several internal sources using Spark in notebooks.

Table 14-1 Internal Sources

Source	Access Type	Integration Method	Decription	External Catalog Support	Sample Code Available
Fusion	Extract Only	Preconfigured Spark Templates	Extracts data from Fusion SaaS applications via BICC into AI Data Platform Workbench tables or volumes.	No	Yes
REST Endpoints	Read Only	JDBC via Spark Notebook	Reads from APIs for ingesting semi-structured data like JSON.	No	Yes
MySQL HeatWave	Read Only	JDBC via Spark Notebook	Move data between AI Data Platform Workbench and MySQL HeatWave using JDBC.	No	Yes
Oracle Autonomous AI Lakehouse	Read/Write + Zero-Copy	JDBC or External Catalog	Ingest from or register Oracle Autonomous AI Lakehouse as an external catalog for querying data directly without duplication.	Yes	Yes
Oracle Autonomous AI Transaction Processing	Read/Write + Zero-Copy	JDBC or External Catalog	Ingest from or register as an external catalog for querying data directly without duplication.	Yes	Yes
Oracle Database	Read/Write	JDBC or External Catalog	Supports data ingestion from on-prem or OCI databases.	Yes	Yes
Exadata	Read/Write	JDBC or External Catalog	Access Exadata systems for high-performance reads and writes using JDBC.	No	Yes

Table 14-2 Spark SQL to , Oracle Autonomous AI Database, Exadata Data Type mapping

Spark SQL Type	Oracle AI Database, Oracle Autonomous AI Database, Exadata Data Type
ByteType	NUMBER(38,10)
ShortType	NUMBER(38,10)
IntegerType (INT)	NUMBER(38,10)
LongType	NUMBER(38,10)
FloatType	FLOAT(126)
DoubleType	NUMBER(38,10)
DecimalType(p,s)	NUMBER(p,s)
StringType	VARCHAR2(4000 CHAR)
BinaryType	BLOB
BooleanType	VARCHAR2(4000 CHAR)
DateType	DATE
TimestampType	TIMESTAMP(6)
ArrayType	VARCHAR2(4000 CHAR)
MapType	Not supported
StructType	VARCHAR2(4000 CHAR)
CalendarIntervalType	Supported if converted to String/VARCHAR2

External Sources

Oracle AI Data Platform Workbench supports ingestion of data from a wide range of sources using Spark-based notebook connectors. These connectors enable users to ingest and process data directly from external sources in a flexible, code-driven manner.

AI Data Platform Workbench provides sample code templates in the Oracle AI Data Platform Workbench Samples Git repository to support ingesting data from several external systems using Spark in notebooks. These templates are pre-built and customizable, allowing users to quickly connect, read, and write data from various commonly used systems.

Table 14-3 External Ingestion Sources

Source	Access Type	Integration Method	Decription	External Catalog Support	Sample Code Available
MySQL	Read/Write	JDBC via Spark Notebook	Ingest and export data between AI Data Platform Workbench and MySQL databases using JDBC connectors.	No	Yes

Table 14-3 (Cont.) External Ingestion Sources

Source	Access Type	Integration Method	Decription	External Catalog Support	Sample Code Available
PostgreSQL	Read/Write	JDBC via Spark Notebook	Supports bidirectional data movement with PostgreSQL via JDBC.	No	Yes
MS SQL Server	Read/Write	JDBC via Spark Notebook	Connect and transfer data from Microsoft SQL Server using Spark and JDBC.	No	Yes
Kafka	Read	Kafka Consumer in Spark Notebook	Stream ingestion from Kafka topics	No	Yes
Hive	Read/Write	JDBC via Spark Notebook	Ingest and export data between AI Data Platform Workbench and Hive databases using JDBC connectors	No	Yes

15

Integration

You can configure Oracle AI Data Platform Workbench to integrate seamlessly with other Oracle services. This section covers enabling services integrated with AI Data Platform Workbench and leveraging the shared data to build advanced AI/ML pipelines.

Topics:

Fusion Data in Oracle AI Data Platform Workbench with BICC

You can connect your Oracle AI Data Platform Workbench directly to raw Fusion data using the Business Intelligence Cloud Connector (BICC).

You can use Oracle Business Intelligence Cloud Connector (BICC) to extract business intelligence and other data in bulk and load it into designated external storage areas. To learn more about creating an extract using BICC, see Creating a Business Intelligence Cloud Extract.

To create a connection with BICC to your Oracle AI Data Platform Workbench, you need the following prerequisites:

Fusion Applications Prerequisites

You need to ensure you have the required permissions for your Fusion Applications instance. The user logging into the Fusion Applications instance must have:

- Administrator permissions for the instance
- ORA_ASM_APPLICATION_IMPLEMENTATION_ADMIN_ABSTRACT role or a role that includes it

Oracle AI Data Platform Workbench Prerequisites

In the OCI compartment where your AI Data Platform Workbench instance resides, you need to have:

- An Object Storage Bucket
- Object Storage bucket name
- Object Storage namespace
- Object Storage host name
- OCID value of your AI Data Platform Workbench instance tenancy
- OCID value of the user with the API key created to access the Object Storage Bucket

Create an Oracle Business Intelligence Cloud Connector Connection to Oracle AI Data Platform Workbench

To use the Oracle Business Intelligence Cloud Connector (BICC) to access your Fusion data directly from your Oracle AI Data Platform Workbench.

1. Open any browser and enter `https://<saas cloud host name>:<saas cloud port number>/biacm` to sign into your Fusion Applications instance.
2. Click **Configure External Storage**.
3. Click the OCI Object Storage Connection tab, then click **Add**.
4. Enter the required OCI parameters. See the required OCI parameters in Oracle AI Data Platform Workbench Prerequisites.
5. Click **Generate API Signing Key**, then click **Export Public Key**.
6. Open Oracle Cloud in another window and log in as the user with access to the Object Storage Bucket.
7. In the top-right, click **Profile**, then click **User Details**.
8. Click **API Key** and add the exported public API key from BICC.
9. Return to the BICC window. Click the Console tab, then click **Test the Connection**.
10. Click **Save**.

Add Fusion Data Sources with a BICC Connection

You can use your BICC connection to your Fusion Applications instance to connect data sources to your Oracle AI Data Platform Workbench.

1. Open any browser and enter `https://<saas cloud host name>:<saas cloud port number>/biacm` to sign into your Fusion Applications instance.
2. Click **Manage Jobs**, then click **Add** to create a new job.
3. Select the offerings and required public virtual objects you want to add to your AI Data Platform Workbench. Click **Save**.
4. Click **Manage Job Schedules**, then click **Add** to create a new schedule.
5. Set the schedule to run immediately or on a ongoing basis. Click **Save**.
6. Once the job has run, check the Object Storage Bucket in OCI to confirm the data is available. Data is exported as zipped CSV files.

Extract Fusion Data from a BICC Connection to a Notebook

Once you have connected Fusion Data to your Oracle AI Data Platform Workbench from a BICC connection, you can extract that data from a notebook.

1. From your home page, navigate to a notebook.
2. In your notebook, enter Spark code to extract the data. For example:

```
spark.read.format("datalake") \
    .option("type", "FUSION_BICC") \
    .option("fusion.service.url", "https://<saas cloud host name>:<saas
```

```
cloud port number>") \
    .option("user.name", "john.smith") \
    .option("password", "**password**") \
    .option("schema", "Financial") \
    .option("fusion.external.storage", "FA4IDL") \
    .option("datastore",
"CrmAnalyticsAM.GeographiesAnalyticsAM.Geography") \
    .load().show()
```

3. Read the data to a data frame and save the data frame as a delta table in a catalog.

Oracle GoldenGate Integration

Oracle AI Data Platform Workbench integrates with Oracle GoldenGate to enable real-time replication of operational data into your AI Data Platform Workbench for analytics and AI-driven workloads.

With the integration of GoldenGate, you can continuously stream and merge transactional changes from source systems into AI Data Platform Workbench target tables.

For full details on configuration parameters, prerequisites, and troubleshooting, please refer to the GoldenGate documentation Connect to Oracle AI Data Platform.

How it Works

- GoldenGate captures change data from source databases and uses stage and merge data flow to enable high throughput, low impact data replication into AI Data Platform Workbench.
- GoldenGate can run initial instantiation and sync the initial load with cdc replication without any data loss. During initial instantiation or cdc replication, GoldenGate can automatically create target tables.

This allows AI Data Platform Workbench to always reflect the most up-to-date view of your operational data for downstream analytics, machine learning, and data engineering.

Requirements to Integrate with Oracle GoldenGate

- An OCI tenancy with an active AI Data Platform Workbench instance
- Database privileges on target AI Data Platform Workbench tables
- The SIMBA JDBC driver for Apache Spark, downloadable from the AI Data Platform Workbench console.

Configuration Summary

When you set up Oracle GoldenGate integration with Oracle AI Data Platform Workbench as part of the replicat process, you need to do the following:

- Select **AIDP** from the **Target** dropdown.
- Configure the replicat properties marked as *TODO* in the properties file.
- Set `gg.target=aidp` to enable AI Data Platform Workbench at the target.
- Specify a staging location. If no staging location is specified, OCI Object Storage is used.

16

Streaming

The section covers the use of streaming data or continuously produced data in Oracle AI Data Platform Workbench.

Topics:

About Streaming

You can process streaming data or continuously produced data in near real-time in Oracle AI Data Platform Workbench using the Apache Spark Structured Streaming capability.

Both notebooks and workflows support Apache Spark structured streaming. You can use the following sources and sinks for reading stream data from, writing stream data to, and for checkpoint locations.

Table 16-1 Supported Sources and Sinks

Source or Sink	Supported?
Volume path (/Volume/bronze/bucket1)	Supported for all formats
Workspace path (/Workspace/folder1/)	Supported for all formats
Tables in catalogs with three part names (catalog.schema.table)	Supported for Delta format only Not supported for Parquet, CSV, JSON, ORC formats Example 1: Supported code • `streaming_df = spark.readStream.format("delta").table('stdcatalog.stdschema.deltatable')` • `streaming_df.writeStream.format("delta").outputMode("append").option("checkpointLocation", "/Volumes/checkpoints1/").toTable("stdcatalog.stdschema.deltatable")` Example 2: Unsupported code • `spark.readStream.option("withEventTimeOrder", "true").format("format") .table("stdcatalog.stdschema.samplecsv")`
Kafka	Supported for any Kafka compatible streams without three-part-naming convention Not supported for Kafka based catalog following three-part-naming convention)
OCI Streaming service	Supported
OCI Object storage path (using oci://)	Unsupported

Table 16-1 (Cont.) Supported Sources and Sinks

Source or Sink	Supported?
Oracle Autonomous AI Lakehouse, Oracle AI Database, Oracle Autonomous AI Transaction Processing	Unsupported for streaming (readStream or writeStream)

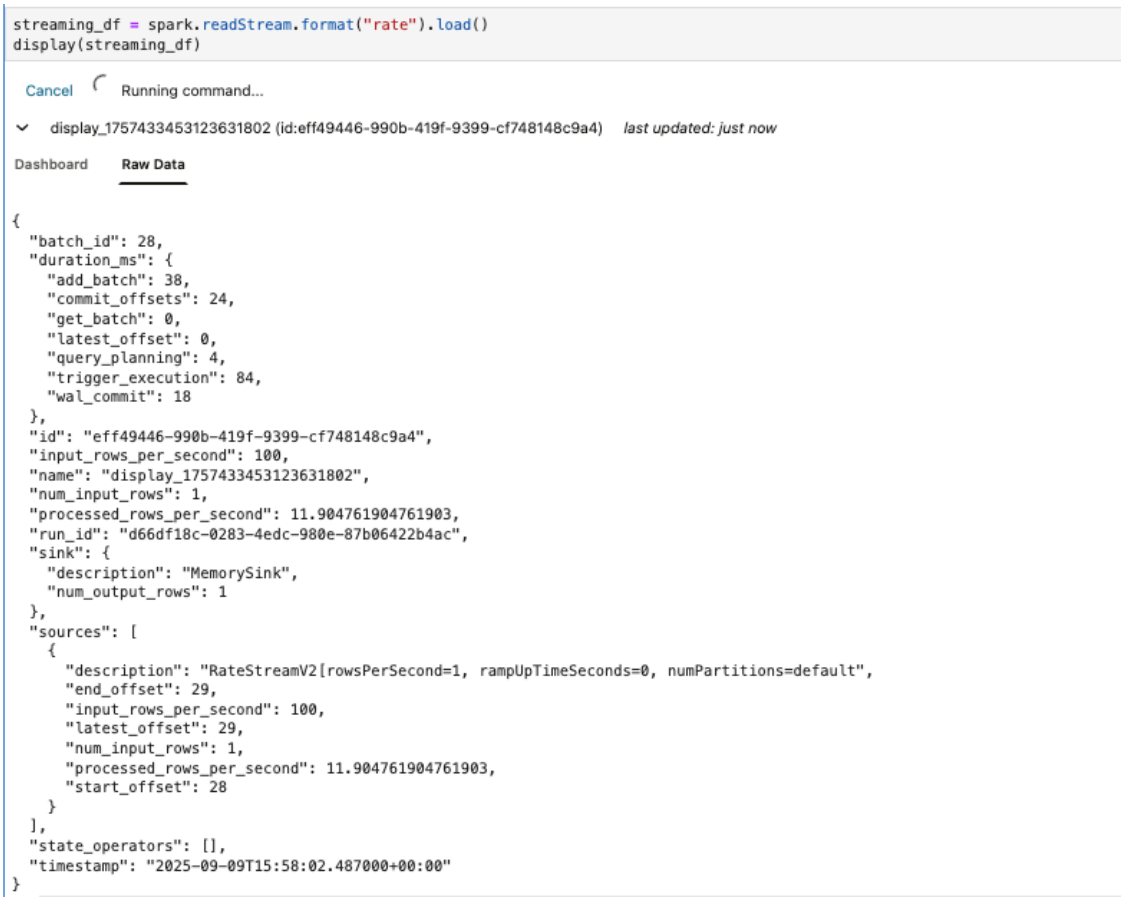
Structured Streaming Using Notebooks

You can write Python code to process stream data in a notebook. Either volume paths or workspace paths are valid as a checkpoint location, but object Storage paths (oci:// format) are not supported as a checkpoint location. We recommend using volume paths as a checkpoint location.


```
# Triggger Once example
# Source => Volume File Source, Sink => Volume File Sink, checkpoint location => volume location
# Place the csv file in your volume location  ex: </Volumes/default/default/manVol/data/>

# Define input and checkpoint paths
input_path = "/Volumes/default/default/manVol/data/*.csv"
output_path = "/Volumes/default/default/manVol/TriggerOnceOutput"
checkpoint_path = "/Volumes/default/default/manVol/TriggerOnceOutoutCheckpoint"

userSchema = spark.read.option("header", "true").csv(input_path).schema

# Read streaming data from a file source
df = spark.readStream \
    .format("csv") \
    .schema(userSchema) \
    .option("header", "true") \
    .load(input_path)

word_counts = df.groupBy("vendor_id").count()

#display(word_counts)

# Write the streaming data to a console sink with Trigger.Once()
query = word_counts.writeStream \
    .outputMode("complete") \
    .format("delta") \
    .trigger(once=True) \
    .option("checkpointLocation", checkpoint_path) \
    .start(output_path)
```

You can see Apache Spark streaming-related events, like input rate, processing rate, and batch duration from the Dashboard tab in your notebook while running streaming code.

You can also view the raw streaming-related events from the Raw Data tab while you incrementally develop your code.

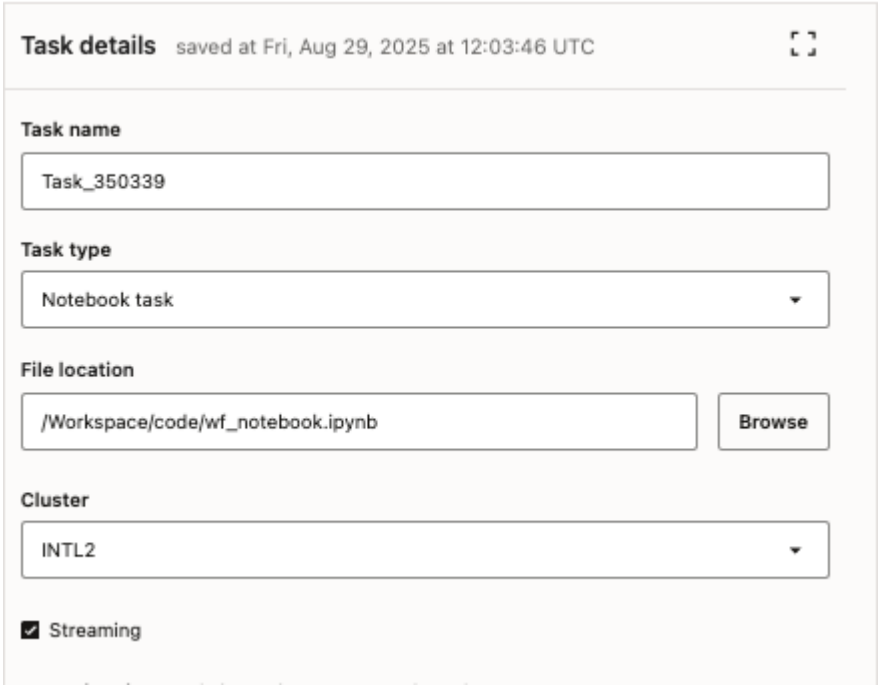


Configuring Spark Structured Streaming using Workflows

You can configure a streaming task inside a workflow for continuous processing of stream data.

You first need to create a job and then add one Notebook or Python task to that job to begin using workflows with streaming in Oracle AI Data Platform Workbench.

1. Navigate to your workspace and click **Workflow**.
2. Click ⊕**Create Job**.
3. Provide a name and description for your job.
4. Click **Browse** and select the location to save the job in your AI Data Platform Workbench. Click **Select**.
5. Enter 1 for **Max Concurrent Runs**.
6. Click **Create**.
7. Click the job you just created.
8. Click **Add task**.
9. Provide a name for your task.
10. Select **Notebook** or **Python** for **Task type**.
11. Click **Browse** and navigate to the Notebook or Python script you want to add as a Streaming task. Click **Select**.
12. Select a compute cluster for the Notebook or Python task, if one is not already attached.
13. Select the **Streaming** checkbox. Selecting Streaming disables execution timeout and task dependencies as options.



14. Select the number of retries a task should attempt on failure. If you select more than 0, you must also specify how much time the job run should wait between retries and if retries should be attempted on timeout.

15. Click **Run Now**.

After a Streaming task is started, it continues to run until you manually stop it. During regular monthly maintenance, the Streaming task is stopped and restarted by the service without requiring any action from your end.

Part V
Machine Learning and AI

This section covers the ways you can integrate Oracle AI Data Platform with machine learning and generative AI.

Topics:

About Machine Learning and AI

AI Data Platform enables you to build supported AI and machine-learning models using by seamlessly accessing data in catalogs, transforming them, and leveraging this to create models using libraries like Scikit-learn, PyTorch, TensorFlow, Keras, and SciPy. You can use your own custom libraries and train custom models in AI Data Platform to help solve use cases.

AI Data Platform also provides access to OCI Generative AI foundational models, enabling you to run batch inference on pre-trained OCI Generative AI chat models like:

- cohere.command-r-08-2024,
- cohere.command-r-plus-08-2024,
- meta.llama-3.3-70b-instruct,

and embedding models like:

- cohere.embed-english-v3.0,
- cohere.embed-multilingual-v3.0

For a full list of models, see Pretrained Foundational Models in Generative AI.

As new pre-trained models are added to the OCI Generative AI service, those models are made available in AI Data Platform's default catalog's, *oci_ai_models* schema.

17

OCI Generative AI (Pretrained Foundation Models)

Generative AI is a fully managed Oracle Cloud Infrastructure service that provides a set of state-of-the-art, customisable large language models (LLMs) that cover a wide range of use cases, including chat, text generation, summarization, and creating text embeddings.

Oracle AI Data Platform Workbench users can access Generative AI models if they have the requisite permissions and the pre-trained model is hosted in the same region as the data lake. For more information on permissions, see Getting Access to Generative AI. For more information on where Generative AI models are hosted, see Regions with Generative AI.

You can use Generative AI models in AI Data Platform Workbench for the following use cases:

- Use the pre-trained chat models to create text for any purpose.
- Extract specific pieces of data from text.
- Generate executive summaries for documents that are too long to read, or summarize any type of text.
- Classify text into predefined categories.

You can also run batch inferences on Spark Data Frames using the pre-trained models in a language of your choice, like SQL or Python. For more information on pretrained models, see Pretrained Foundational Models in Generative AI.

Prerequisites for Generative AI

You must meet the following prerequisites to use Generative AI in AI Data Platform Workbench:

- User must have USE permissions on the base models
- AI Data Platform is in the same region where the Generative AI models are hosted

If the prerequisites are met, the models are listed in the *default.oci_ai_models* schema. You can then list the models in the catalog explorer while working in a notebook and drag drop the models to generate sample code or use the model for batch inference. Alternatively, you can choose to write your code in an AI Data Platform Workbench notebook to invoke the model.

You can use the following methods to invoke a Generative AI model:

SQL

```
select *, query_model(model_name, concat("What is the sentiment for this
review: ", review)) as sentiment from
<<catalog_name>>.<<schema_name>>.<<table_name>>
```

Where:

- *model_name* is the generative AI model you want to invoke: `default.oci_ai_models.<model_name>`
- *review* is the column name that is used to create the prompt

- *sentiment* is the output column name
- *<<catalog_name>>.<<schema_name>>.<<table_name>>* is the table in 3-part name pattern

PySpark

```
df.withColumn("sentiment", query_model(model_name, "What is the sentiment for
this review: "+review))
```

Where:

- *model_name* is the generative AI model you want to invoke: `default.oci_ai_models.<model_name>`
- *review* is the column name that is used to create the prompt
- *sentiment* is the output column name
- *df* is the input data frame

Request Limit

Description	Limit Name	Service Limit
Maximum number of chat requests per minute allowed per compartment for on-demand inferencing	max-on-demand-chat-request-per-minute-count	500

Part VI
Administration

This section covers the administration of your Oracle AI Data Platform Workbench and its contents.

In order to create and manage AI Data Platform Workbench instances, you will need to have an existing cloud account (tenancy) with Oracle Cloud Infrastructure (OCI) and have familiarity with the essential OCI concepts. For more information, see Service Essentials.

Workspaces

A workspace in AI Data Platform Workbench acts as an isolated logical container where users can manage and organize their data resources, including workflows, notebooks, and libraries. Workspaces enable efficient collaboration and governance by keeping resources grouped logically.

Networking

The data assets that are read from and written to by your AI Data Platform Workbench can be in an OCI virtual cloud network. For more information, see Virtual Cloud Network.

AI Data Platform Workbench workspaces can be enabled for a private network where one or more target data assets belong to. Enabling for private networks allows workspaces to access target data assets located in the private network. In order to enable private network for a workspace, you will need three networking related details from your administrator: the existing virtual cloud network of the data asset, the subnet and the network security group.

In order to create an external catalog for a data asset that is in a private network, you will first need to create a workspace enabled for the same private network.

Security and Identity
In order to create, manage or use Oracle AI Data Platform Workbenches, you need to have either a local user or federated user credential for console. The existing security features (e.g. Multi factor authentication or MFA) and access control features (e.g. pre-requisite policies, sign-on policies) are applicable for Oracle AI Data Platform in OCI Console. For more information, review the following OCI documentation:

- User Credentials
- Federating with Identity Providers
- Securing IAM
- Managing Access to Resources

On top of OCI Security and Identity and Access Management, Oracle AI Data Platform Workbench implements additional Role Based Access Control (RBAC) to enforce fine-grained access control across different resources. In your AI Data Platform Workbench, you can define roles and permissions for workspaces, catalogs, and datasets to ensure secure collaboration.

Limitations

Oracle AI Data Platform Workbench implements and manages limits for the resources created and managed by Oracle AI Data Platform Workbench instances.

Topics:

18

Workspaces

Workspaces are containers for organizing your notebooks and workflows.

Topics:

About Workspaces

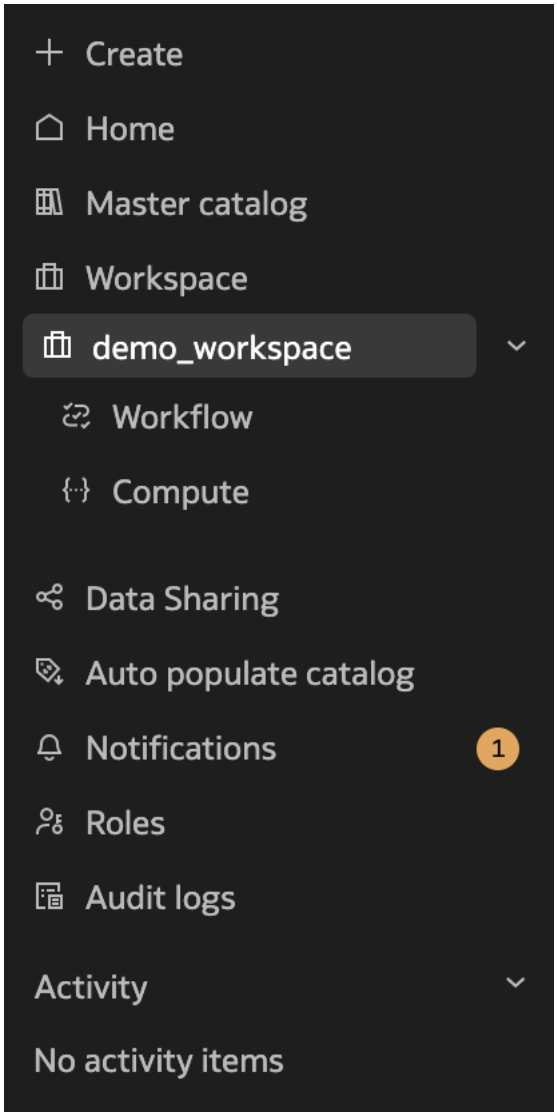
Workspaces are isolated logical containers that organizes your Oracle AI Data Platform Workbench notebooks, files, folders, workflows and compute clusters.

Video

A workspace establishes a mapping between the file-like resources you need (e.g. notebooks, files, folders) with corresponding workflow resources (e.g. jobs, job runs) and corresponding compute clusters.

One AI Data Platform Workbench instance can have multiple workspaces. A default workspace is created while creating your AI Data Platform instance. You can create additional workspaces within the same AI Data Platform Workbench instance depending on your business needs. For example, you can create different workspaces for different teams or business units (e.g. Sales, Marketing, Engineering).

From OCI console, clicking on your AI Data Platform Workbench instance takes you to the AI Data Platform Workbench Homepage. At the left-most navigation pane of the Homepage, you need to select a specific workspace before accessing the workspace files, folder, notebooks, workflows and compute clusters.



Adding Resources to Workspaces

Within a workspace, new files, folders, notebooks and jobs can be created from scratch using the ⊕ button.

New files, folders or notebooks can also be uploaded from the local machine. Multi-file upload is coming soon.

Folders

You can use folders in your workspaces to help organize files and other folders containing code, libraries, configurations, metadata or data.

When you create a new workspace, it automatically begins with a Shared folder. You can create additional folders to help organize your files.

You can manage permissions to your folders and assign who can view and modify the content of each folder. For more information, see Workspace Folder Permissions.

When you delete a folder, you will delete all data and metadata contained in that folder.

Editing and renaming workspace resources

You can edit the contents of and rename the files, folders, notebooks or jobs within a workspace.

When you click on a workspace resource, depending on the resource type, the corresponding editor will open where they can be edited and renamed inline. For example, notebooks will open in a notebook editor.

You can rename folder names and descriptions inline at the top of the workspace details page.

Create a Workspace

You create a workspace as a container for your files, folders, notebooks, workflows and compute clusters.

To create a workspace, you need the CREATE_WORKSPACE permission or the AI_DATA_PLATFORM_ADMIN role. The CREATE_WORKSPACE permission can be granted directly from the Workspace Listing page.

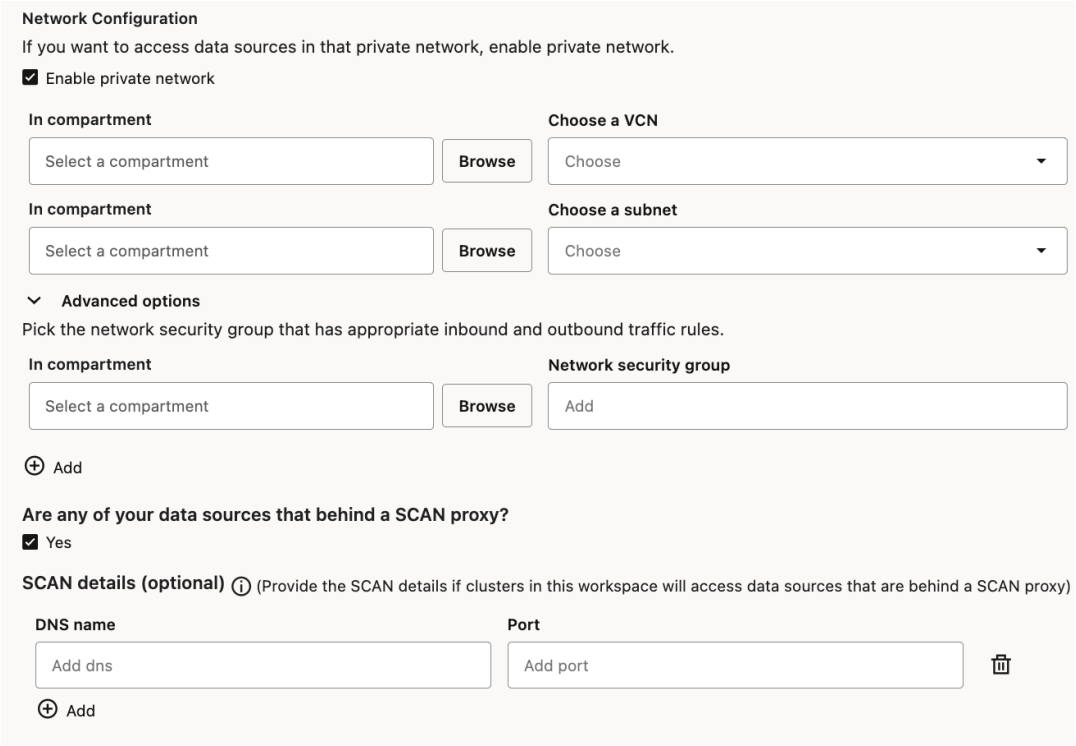
1. On the Home page, click **Workspace**.
2. Click ⊕ **Create Workspace**.
3. Fill in the required details and select the default catalog you want your workspace to pull data from.
4. Click **Create**.

Create a Workspace with Private Network Access Enabled

You can create a workspace that can access data assets in a private network. By default, a workspace is not enabled for accessing data assets in a private network and can only access data assets that are not in a private network. A workspace can only access data assets residing in one private network at a time.

In order to create a Workspace that can access existing data assets in an existing private network, you need to have the IAM policies to inspect the VCN of the data asset, the subnet of the data asset, and, if network-security-group is used, to inspect the network-security-group of the data asset within their respective compartments. For more information, see IAM Security Policies.

1. On the Home page, click **Workspace**.
2. Click ⊕ **Create Workspace**.
3. Select **Enable private network**.



4. Select the VCN from the corresponding OCI compartment.
5. Select the subnet from the corresponding OCI compartment.
6. Provide the details of the network security group. Click **Add** if there are multiple.
7. Optional: Select **Yes** if any of your data sources are behind a SCAN proxy. Add DNS name and Port for your SCAN proxy.

   This is a prerequisite if clusters in this workspace need to access Oracle databases that are behind SCAN proxies.
8. Click **Create**.

Workspaces, External Catalogs, and Data Assets in a Private Network

In order to create an external catalog for a data asset that is in a private network, you will first need to create a workspace enabled for the same private network.

During the creation of an external catalog, you need to pick the workspace that is enabled for the private network of the data asset. For more information, see Create an External Catalog for Private Networks.

Clusters in a workspace can read-write to this external catalog only if the workspace is already enabled for the private network of the data asset.

In order to access data assets in a private network, the VCN of that private network needs to have security rules allowing ingress traffic from all source IP inside that VCN to all destination ports. If that security rule is not already configured, you need to create a new Ingress security rule. For more information, see Security Rules.

If your data asset is in another VCN within the same region, the two VCN needs to be configured with Local VCN peering (within region) using Local Peering Groups or using Local VCN Peering through an upgraded Dynamic Routing Gateway (DRG).

If your data asset is in another VCN in a different region, the two VCN needs to be configured with Remote VCN peering (across regions) using Remote Peering Connections (RPCs) or using Remote VCN Peering through an Upgraded Dynamic Routing Gateways (DRG).

Edit a Workspace

Workspace attributes can be changed after the workspace is created.

Most changes to the network configuration of a workspace first require all existing compute clusters to be manually stopped.

1. On the Home page, click **Workspaces**.
2. Next to the workspace you want to edit, click ··· **Actions** and click **Edit**.
3. Make your changes to the workspace and click **Edit**.

Delete a Workspace

Deleting your workspace removes it and all your contained jobs, tasks, and compute clusters.

1. On the Home page, click **Workspaces**.
2. Next to the workspace you want to delete, click ··· **Actions** and click **Delete**.
3. On the confirmation page, click **Delete**.

Create a Folder

You can create folders in workspaces where you have appropriate permissions.

1. In the workspace you want to create a folder, click ⊕ **Create** then **Folder**.
2. Provide the folder name and a folder description. Click **Create**.

Delete a Folder

You can delete folders and their contained data and metadata.

1. Navigate to the folder you want to delete in your workspace.
2. Click ··· **Actions** then click **Delete**.
3. Click **Delete**.

Move a Folder

You can move folders and their contained data and metadata to another location in your workspace.

1. Navigate to the folder you want to move to a new location in your workspace.
2. Click ··· **Actions** then click **Move**.
3. Select a new location for your folder and its contents.

4. Click **Move**.

Copy a Folder

You can create a duplicate of a folder and its contents in your workspace.

1. Navigate to the folder you want to copy to another location in your workspace.
2. Click ··· **Actions** then click **Copy**.
3. Select a location to copy the folder and its contents to.
4. Click **Copy**.

19

Notifications

Oracle AI Data Platform Workbench keeps you updated about the status of your long running operations through notifications to the users or administrators of workspaces and catalogs in your AI Data Platform Workbench instance.

Notifications keep you in the loop on the events that matter most to you related to operations you initiated. You get an instant heads-up when something finishes, breaks, or needs your attention.

Notifications in Oracle AI Data Platform Workbench are one of two types, based on the audience:

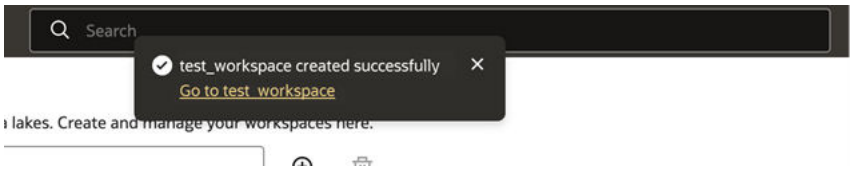
- Notifications to users of workspace or catalogs, for the success, failure, or in-progress status of user initiated operations
- Notifications to administrators of workspaces or catalogs for unexpected events (warnings, errors) detected by the Oracle AI Data Platform service

Notifications tell you what's happening with your long-running or asynchronous tasks. Notifications let you know if something you started, scheduled, or set up in a workflow has succeeded, failed, or is still in progress. Notifications can be related to different user-triggered actions, such as:

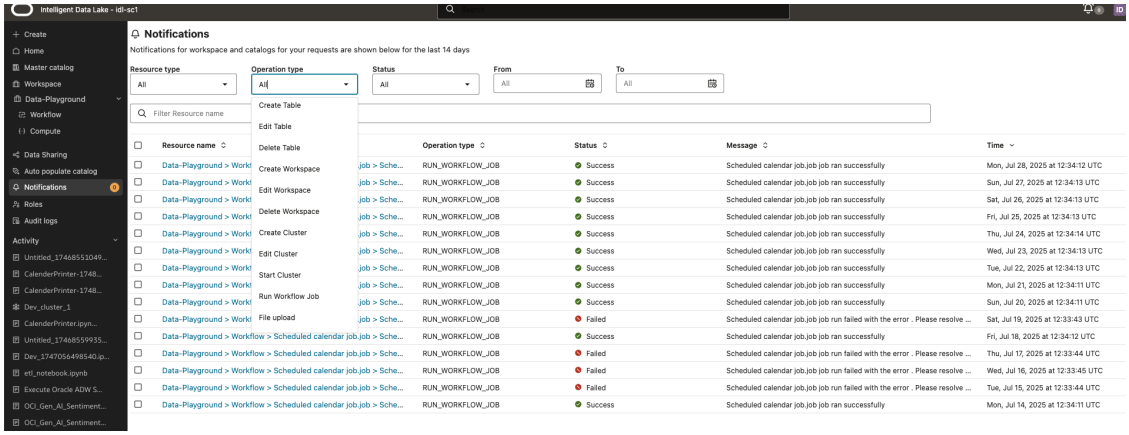
- Creating catalogs, schema, tables, volumes, workspaces, and compute clusters
- Starting or stopping compute clusters
- Starting, progress, and completion of file uploads

Oracle AI Data Platform Workbench also sends notifications to workspace or catalog admins about unexpected errors or warnings affecting resources across all the workspaces or catalogs they manage. These are typically error or warning notifications regarding health, capacity, and performance related issues that users don’t directly trigger. For example, your AI Data Platform Workbench sends notifications when an external catalog can’t authenticate, or certain cluster shapes become unavailable, or if a Default Master Catalog cluster hits any issue. These notifications can provide you early warning signals when workflow jobs are running with no data or clusters are slowing down under heavy load, along with clear next-step guidance, like advising an increase OCPUs or contacting Oracle support. These proactive messages let you fix problems before they disrupt your workloads.

Notifications appear as brief toast messages and disappear automatically after a short period. When you see a notification, you can click on it to view more details, you can ignore it and let it disappear, or you can click the X to close and dismiss it.



All the notifications from the last 14 days are stored in your Notifications page, so you don't need to worry about missing a notification while you are away or attend to every notification immediately while you're occupied with other tasks.



The count in the notification page indicates the number of notifications that are new for you. After you visit the notification page, the notification count will reset. In the notification page, you can filter by resource types, operation types, status and the date-time ranges. Filtering helps you narrow down to the displayed notifications so you can locate specific notifications you want to review. If you are not able to view the full content of a notification message, you can hover over it to view the complete text. Clicking on the notification takes to you the respective resource details page.

20 Limits

Oracle AI Data Platform Workbench imposes limits on its resources to ensure efficient and error-free use.

All limits are independently applied. The following examples illustrate how limits are applied:

- **Example 1:**
  - You have 1 AI Data Platform Workbench instance with of 10 workspaces and 5 workspaces enabled for private networks, or
  - You have 2 AI Data Platform Workbench instances with 5 workspaces each (totaling 10 workspaces). The first AI Data Platform has 5 workspaces enabled for private networks. Then the second one can have only 3 workspaces enabled for private networks, to remain within the limit for up to 8 workspaces enabled for private networks.
- **Example 2:**
  - You have 1 all-purpose compute clusters with 100 driver and worker nodes, or
  - You have 25 all-purpose compute clusters each with 4 nodes (totalling 100 nodes across all clusters).

The following table lists various numerical limits for Oracle AI Data Platform Workbench resources in a single tenancy.

Resource	Metric	Scope	Limit
AI Data Platform Workbench	Maximum number of AI Data Platform Workbenches	Region	100
Workspace	Maximum number of workspaces Includes default workspace and workspaces enabled for private network	Region	100
Workspaces enabled for private network	Number of workspaces enabled for private network	Region	8
Catalog	Maximum number of catalogs that can be created	AI Data Platform	1000
Schema	Maximum number of schema that can be created	Catalog	10000
Table	Maximum number of tables that can be created	Schema	100000
Volume	Maximum number of volumes that can be created	Schema	10000

Resource	Metric	Scope	Limit
Jobs	Maximum number of jobs that can be created per hour	Workspace	10000
Jobs: Concurrent tasks	Maximum number of tasks that are running simultaneously	Workspace	1000
Jobs: Saved jobs	Maximum number of jobs that can be saved	Workspace	12000
Jobs: Tasks	Maximum number of tasks Excludes tasks inside a pipeline task.	Job	100
Jobs: Nesting level	Maximum levels of nesting	Job	3
Jobs: Schedule	Minimum frequency	Job	30 minutes
Jobs: Log retention period	Maximum number of days JobRun logs will be retained	AI Data Platform	30
Workspace file	Maximum number of files Includes notebooks	Workspace	10000
Workspace folder	Maximum number of folders	Workspace	10000
Workspace file: File size	Maximum size allowed for a single file	Workspace	500 MB
Workspace folder: folder depth	Maximum depth of folders	Workspace	25
Workspace: Object name length	Maximum length of an object name	Workspace	500 characters
Compute: Clusters	Maximum number of clusters	Region	200
Compute:Driver & Worker node	Maximum total number of driver and worker nodes	Region	500
Compute:OCPU per node: (driver or worker)	Maximum number of OCPU	Node (Driver or worker)	64
Compute: Memory per OCPU	Memory per OCPU	OCPU	Up to 16 times OCPU (in GB)
Compute: Block storage per OCPU	Block storage per OCPU	OCPU	100 times OCPU (in GB)
Compute: NVIDIA GPU	Maximum number of NVIDIA GPU	Region	3
Compute: Logs retention	Maximum number of days compute logs will be retained	AI Data Platform	30

21

Known Issues

This page lists current limitations, bugs, and behavior inconsistencies identified in Oracle AI Data Platform Workbench.

These issues are actively being tracked and will be addressed in future updates.

When reporting any issues regarding Oracle AI Data Platform Workbench resources (e.g. Workspace, Compute), you should also share the ID of the resource. You can find the resource ID by going to the page where the resource is listed, clicking ··· **Actions** and clicking **Copy ID**.

Table 21-1 Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
1	AI Data Platform Workbench Instance	UPDATE	Updating tags on the AI Data Platform Workbench instance does not retroactively apply to sub resources that were already created in the same compartment by AI Data Platform Workbench.	Tag updates only apply to newly created resources in the customer tenancy after the tag change. If consistent tag values are required across all resources, manually update tags on existing sub resources.
2	AI Data Platform Workbench Instance	UPDATE	When a user selects or creates an external table using a bucket that has already been tagged/associated with another AI Data Platform Workbench instance the service blocks reuse. Deleting the external table from the original instance does not allow reusing the same bucket in the new AI Data Platform Workbench instance.	No current workaround. Once a bucket is chosen and the external table is created, the association is fixed. Customers must provision a new bucket if they need to create a new external table with a different AI Data Platform Workbench instance and copy the data to this new location.

Table 21-1 (Cont.) Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
3	AI Data Platform Workbench Instance	CREATE	A single bucket location cannot be referenced by multiple AI Data Platform Workbench instances for external tables at the same time. Attempting to reuse the same bucket results in validation failure.	No current workaround. Each AI Data Platform Workbench instance requires its own dedicated bucket for external tables. Customers must create separate buckets per instance for creating new external tables in a new AI Data Platform Workbench instance.
4	AI Data Platform Workbench Instance	UPDATE	Cannot edit tag namespaces after instance creation.	No current workaround

Table 21-1 (Cont.) Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
5	Auto-populate	CREATE	Auto-populate creation fails if the bucket has already been tagged with another AI Data Platform Workbench.	• If the tagged AI Data Platform Workbench is still present and active, the bucket can't be used in the new AI Data Platform Workbench's extraction. This is to prevent the two AI Data Platform Workbenches from overwriting each other's data. • If the tagged AI Data Platform Workbench is deleted, the bucket can be reused for the new AI Data Platform Workbench. The governingAIDPId tag needs to be removed first, then you can retry auto-populate creation.
6	Compute	UPDATE	Updating a cluster is currently only possible while it is active. Updating a stopped cluster is currently not possible.	To update a stopped cluster, you can either start the cluster and update it or delete and create that cluster again with the changes.
7	Workspace	MOVE	From a workspace page, moving multiple items by selecting multiple items at one go is currently not possible.	Users can move one item at a time by selecting that item first, then selecting **Move** from the Actions Menu.

Table 21-1 (Cont.) Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
8	External Catalog	REFRESH	When a user creates an external catalog, it runs a background job to harvest the metadata from the external source. The background job takes a long time for the metadata from the external system to reflect in the external catalog. Clicking on the refresh icon in the master catalog while the first job is still in progress causes problems in the processing of the background job.	You should not click the Refresh button until the job has finished. Users can see the progress of the background job in the **History** tab. When the job is complete, you can click Refresh if all the expected objects have not appeared in the external catalog.
9	External Catalog	REFRESH	When a user creates an external catalog, it runs a background job to harvest the metadata from the external source. If a column in a table in that external source has "$" in the name it will not be harvested.	No current workaround
10	External Catalog	CREATE	Users cannot create external catalogs with regional wallet files.	You can create external catalogs using instance wallets or provide the instance details and create an external catalog.
11	Data Ingestion	CREATE	Users cannot write to an Oracle Autonomous AI Lakehouse table using a notebook without first creating an external catalog	Users are expected in general to first create an external Autonomous AI Lakehouse catalog and then write it to an Autonomous AI Lakehouse table.

Table 21-1 (Cont.) Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
12	External/Managed Table	CREATE	Users cannot create external tables when the input data is in multi-line JSON format.	No current workaround
13	Data Sharing	CONSUMPTION	Users cannot receive a share using Delta Sharing protocol.	Users can load the shared data into a data frame using the delta-sharing library's load_as_spark(<<table_path>>).
14	Data Sharing	CONSUMPTION	Users cannot consume data shared by AI Data Platform Workbench in Autonomous AI Lakehouse	No current workaround
15	Auto Populate	CREATE	The Auto Populate feature does not support table creation when the target location contains delimited data files with a delimiter other than a comma.	No current workaround
16	Workflow	CREATE/ SCHEDULE	Directly accessing OCI Object Storage (or GenAI service) is currently not supported for scheduled jobs	Configure external volumes for Object Storage access in the workflow job. There is no current workaround to access generative AI models

Table 21-1 (Cont.) Known Issues

	Resource	Category	Error and How to Find the Problem	Workaround
17	Workspace configured for Private Network	Access AWS S3 Buckets	For workspaces configured for Private Network, currently there is a known issue about AWS S3 access from Notebooks and Workflows.	File a support ticket if you need to access AWS S3 buckets from notebooks and workflows for Workspaces configured for private networks. Please indicate the workspace key in that ticket. AI Data Platform team will enable a temporary workaround for your tenancy while we work on a permanent solution.
18	Compute	Monitor compute notebooks	All notebooks attached to a compute cluster as displayed are Active regardless of actual status	No current workaround AI Data Platform team is working on a fix. Notebooks will display the correct status when the fix is deployed.

Part VII
Reference

This section provides additional reference material on Oracle AI Data Platform Workbench.

Topics:

22

SQL Grammar

Oracle AI Data Platform Workbench users can use SQL to automate their DDL workloads.

Topics:

Catalog SQL Grammar

Catalog objects support the listed SQL grammar for DDL workloads.

Oracle AI Data Platform Workbench supports all standard Spark SQL data types. For more information, see Apache Spark Documentation - Supported Data Types.

Table 22-1 Standard and External Catalog SQL Grammar

Operation	Grammar

Create Catalog	**Catalog**
	
	`CREATE CATALOG [ IF NOT EXISTS ] <<catalog_name>> [ PROPERTIES (DESCRIPTION = description ) ] OPTIONS ( { option_name = option_value } [ , ... ] )`
	
	External Catalog
	
	`CREATE EXTERNAL CATALOG [ IF NOT EXISTS ] <<catalog_name>> [PROPERTIES ( DESCRIPTION description ) ]OPTIONS ( { option_name = option_value } [ , ... ] )`
	
	OPTIONS will have connection details
	External Catalog - Oracle Autonomous AI Lakehouse Example
	
	`wt = base64 encoded wallet contents create_sql="create external catalog if not exists catalog_adw options ('wallet.content' = '{wt}', 'type' = 'ORACLE_ADW', 'user.name' = 'ADMIN', 'tns' = 'adw23ai_high', 'password' = 'xxxxx','wallet.password' = 'xxxxx')"`
	
	Response
	
	`Catalog <<catalog_name>> created successfully`
	
	Error
	
	`<<SQL Command>> failed due to <<reason>>`

Table 22-1 (Cont.) Standard and External Catalog SQL Grammar

Operation	Grammar

Alter Catalog	**Alter catalog name**
	
	`ALTER CATALOG old_catalog_name RENAME new_catalog_name;`
	
	Alter catalog description
	
	`ALTER CATALOG <catalog-name> set properties (DESCRIPTION=<property-value>)`
	
	Alter catalog options (conn)
	
	`ALTER CATALOG <catalog-name> set options (option_name = option_value)`
	
	Response
	
	`Catalog <<catalog_name>> updated successfully`
	
	Error
	
	`<<SQL Command>> failed due to <<reason>>`

Delete Catalog	`DROP CATALOG [ IF EXISTS ] catalog_name`
	
	By default during DROP catalog, all child objects will also get deleted
	Response
	
	`Catalog <<catalog_name>> dropped successfully`
	
	Error
	
	`<<SQL Command>> failed due to <<reason>>`

Table 22-1 (Cont.) Standard and External Catalog SQL Grammar

Operation	Grammar

List Catalogs	`SHOW CATALOGS [ [ LIKE ] [ regex_pattern ] [ TYPE = EXTERNAL CATALOG | CATALOG ]`
	
	regex_pattern: A regular expression pattern that is used to filter the results of the statement.
	Response:
	
	Catalog | **Type**
	<<catalog_name>> | Catalog | External Catalog
	<<catalog_name>> | Catalog | External Catalog
	<<catalog_name>> | Catalog | External Catalog
	
	Error
	
	`<<SQL Command>> failed due to <<reason>>`

Table 22-1 Standard and External Catalog SQL Grammar

Operation	Grammar

Describe Catalog	`DESC CATALOG <<catalog_name>>`
	
	`DESCRIBE CATALOG <<catalog_name>>`
	
	Response (Standard Catalog):
	
	Attribute / **Value**
	Name — Standard catalog name
	Type — Standard Catalog
	Description — Standard catalog description
	Created by — Principal that created the standard catalog
	Created on — Date and time created
	Updated by — Principal that last updated the standard catalog
	Updated on — Date and time last updated
	
	Response (External catalog):
	
	Attribute / **Value**
	Name — External catalog name
	Type — External Catalog
	Source type — Source of external catalog (e.g. Oracle Autonomous AI Lakehouse)
	Description — External catalog description
	Created by — Principal that created the external catalog
	Created on — Date and time created
	Updated by — Principal that last updated the external catalog
	Updated on — Date and time last updated
	Connection details — Connection .json file
	
	Error:
	
	`<<SQL Command>> failed due to <<reason>>`

Schema SQL Grammar

Schema support the listed SQL grammar for DDL workloads.

Oracle AI Data Platform Workbench supports all standard Spark SQL data types. For more information, see Apache Spark Documentation - Supported Data Types.

Table 22-2 List of Schema SQL Grammar

Operation	Grammar

Create Schema	`CREATE SCHEMA [IF NOT EXISTS] catalog_name.schema_name`
	
	Response
	
	`<<SQL Command>> was successfully executed`
	
	Error
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Alter Schema	**Alter Schema Description**
	
	`ALTER SCHEMA <schema-name> set dbproperties (DESCRIPTION=<property-value>)`
	
	Response
	
	`<<SQL Command>> was successfully executed`
	
	Error
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Delete Schema	`DROP SCHEMA [ IF EXISTS ] <<schema_name>>` By default during DROP schema, all child objects will also get deleted

Table 22-2 List of Schema SQL Grammar

Operation	Grammar

List Schemas	`SHOW SCHEMAS [ { FROM | IN } catalog_name ] [ [ LIKE ] regex_pattern ]`
	
	Examples:
	• `SHOW SCHEMAS FROM defaultcatalog1 LIKE 'd*'`
	• `SHOW SCHEMAS IN defaultcatalog1 LIKE 'd*'`
	Response:
	
	| | **Schema** |
	| --- | --- |
	| 1 | <<schema_1>> |
	| 2 | <<schema_2>> |
	| 2 | <<schema_3>> |
	
	Error
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Describe Schema (get details)	`DESCRIBE SCHEMA <<catalog_name>>.<<schema_name>>`
	`DESCRIBE SCHEMA <<schema_name>>`
	`DESCRIBE SCHEMA <<schema_name>> in Catalog <<catalog_name>>`
	
	| **Attribute** | **Value** |
	| --- | --- |
	| Catalog name | Catalog name |
	| Schema | Schema name |
	| Description | Schema description |
	| Created by | User that created the catalog |
	| Created on | Date and time created |
	| Updated by | User that last updated the catalog |
	| Updated on | Date and time last updated |
	| Location | Location in the catalog |

Drop Schema	`drop schema [ IF EXISTS ] <<schema_name>> cascade`

Volume SQL Grammar

Volume objects support the listed SQL grammar for DDL workloads.

Oracle AI Data Platform Workbench supports all standard Spark SQL data types. For more information, see Apache Spark Documentation - Supported Data Types.

Table 22-3 Volume SQL Grammar

Operation	Grammar
Create volume	`CREATE [ EXTERNAL ] VOLUME [IF NOT EXISTS ] <<catalog_name.schema_name.volume_name>> [ LOCATION location_path ] [ PROPERTIES (DESCRIPTION = description ) ]`
Alter volume properties	`ALTER VOLUME <<volume_name>> { RENAME TO <<new_volume_name>> | [ set properties (DESCRIPTION = description) ] }`

Drop volume	`DROP VOLUME [ IF EXISTS ] <<volume_name>>`
	
	OR
	
	`DROP VOLUME <<catalog_name>>.<<schema_name>>.<<volume_name>>`
	
	By default during DROP volume, all child objects will also get deleted

List Volumes	`SHOW VOLUMES [ { FROM | IN } catalog_name.schema_name ] [ [ LIKE ] regex_pattern } ]`

Table 22-3 Volume SQL Grammar

Operation	Grammar

Describe Volume	`DESCRIBE VOLUME volume_name`
	
	Attribute / **Value**
	Catalog name / Catalog name
	Schema name / Schema name
	Volume name / Volume name
	Description / User defined description of volume
	Location / Location in catalog
	Volume type / Type of volume
	
	Error:
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Table SQL Grammar

Table objects support the listed SQL grammar for DDL workloads.

Oracle AI Data Platform Workbench supports all standard Spark SQL data types. For more information, see Apache Spark Documentation - Supported Data Types.

Operation	Grammar

Create Table	`CREATE [EXTERNAL] TABLE [IF NOT EXISTS] <catalog_name>.<schema-name>.<table-name>`
	`[ ( <column1-name><column1-type> [comment <column1-comment>], ... ) ]`
	`USING [HIVE|DELTA, CSV, TXT, ORC, JDBC, PARQUET, etc.]`
	`[ options ( <key1>=<val1>[, ...]) ]`
	`[ PARTITIONED BY (<par-column-name>[, ...]) ]`
	`[ CLUSTERED BY ( <clus-column-name>[, ...])`
	`[ SORTED BY ( <sort-column-name> [ asc | desc ][, ...]) ]`
	`INTO <num_buckets> buckets]`
	`[ LOCATION '<path>']`
	`[TBLPROPERTIES ( DESCRIPTION = 'some-description', '<property-name>'='<property-value>'[, ...]) ]`
	
	Response:
	
	`<<SQL Command>> was successfully executed`
	
	Error:
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Create Managed Table	**Create Managed Table**
	
	`CREATE TABLE <catalog>.<schema>.<table-name>`
	`[ ( <column1-name><column1-type> [comment <column1-comment>], ... ) ]`
	`USING <format>;`
	
	Response:
	
	`<<SQL Command>> was successfully executed`
	
	Error:
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Operation	Grammar

Create Managed Table with Data	`create datatable <<catalog_name>>.<<schema_name>>.<<table_name>> [ ( <column1-name><column1-type> [comment <column1-comment>], ... ) ] tblproperties ('lakehouse_storage_format'='PARQUET' ) using parquet with select ( <column1-name>], ... ) from parquet.'oci://bucket@namespace/folder/'`
	
	Response:
	
	`<<SQL Command>> was successfully executed`
	
	Error:
	
	`Error: <<SQL Command>> failed due to <<reason>>`

Create Table with Uniform Support	`CREATE [EXTERNAL] TABLE [IF NOT EXISTS] <catalog_name>.<schema-name>.<table-name> [ ( <column1-name> <column1-type> [comment <column1-comment>], ... ) ] [TBLPROPERTIES ('delta.universalFormat.enabledFormats' = 'iceberg') ]`

Operation	Grammar

Alter Table	`ALTER TABLE table_old_name RENAME TO table_new_name`
	`ALTER TABLE table_name ADD COLUMNS ( col_spec [ , ... ])`
	`ALTER TABLE table_name DROP { COLUMN | COLUMNS } [ ( ] col_name [ , ... ] [ ) ]`
	`ALTER TABLE table_name RENAME COLUMN col_name TO col_name`
	`ALTER TABLE table_name ADD [IF NOT EXISTS] ( partition_spec [partition_spec ... ] )`
	`ALTER TABLE table_name DROP [ IF EXISTS ] partition_spec [PURGE]`
	`ALTER TABLE table_name set tblproperties (description ='some-description')`

Drop Table	`DROP TABLE [ IF EXISTS ] table_name [ PURGE ]`
	
	Response:
	
	`<<SQL Command>> was successfully executed`
	
	Error:
	
	`Error: <<SQL Command>> failed due to <<reason>>`

List Tables in a schema	`SHOW TABLES in catalog_name.schema_name [ LIKE <regex_pattern> ]`
	
	regex_pattern: A regular expression pattern that is used to filter the results of the statement.
	Response:
	
	`<<namesake>>,tableName,isTemporary`
	
	Error:
	
	`<<SQL Command>> failed due to <<reason>>`

Operation	Grammar

Describe Table	`DESCRIBE TABLE [ FORMAT ] catalog_name.schema_name.table_name [ PARTITION (<partition_col_name> = <partition_col_val>, ...) ] [ catalog_name.schema_name.table_name .column_name ]`
	
	Format: If `EXTENDED` is specified as the format, additional metadata information (such as parent database, owner, and access time) is returned.
	DESCRIBE TABLE catalog.schema.table Response:
	
	`col_name,data_type,comment`
	
	DESCRIBE TABLE catalog.schema.table column Response:
	
	`info_name,info_value`

View SQL Grammar

View objects support the listed SQL grammar for DDL workloads.

Oracle AI Data Platform Workbench supports all standard Spark SQL data types. For more information, see Apache Spark Documentation - Supported Data Types.

Operation	Grammar
Create View	`CREATE VIEW [IF NOT EXISTS] <<catalog_name.schema_name.view_name>> as select ... from table[TBLPROPERTIES ( DESCRIPTION = 'some-description', '<property-name>'='<property-value>'[, ...]) ]`

Alter View (Coming Soon)	**Rename view**
	`ALTER VIEW <schema-name>.<old_view_name> RENAME TO <schema-name>.<new_view_name>`
	Change view description
	`ALTER VIEW <schema-name>.<view_name> set TBLPROPERTIES (DESCRIPTION = 'some-description)`

Delete View	`DROP VIEW [IF EXISTS] <<catalog_name.schema_name.view_name>>`

<table>
<tr><th>Operation</th><th>Grammar</th></tr>
<tr><td>List Views</td><td>SHOW VIEWS FROM <<catalog_name.schema_name>>
Response:
<table><tr><th>Attribute</th><th>Type</th></tr><tr><td>Catalog</td><td>Catalog name</td></tr><tr><td>Schema</td><td>Schema name</td></tr><tr><td>View</td><td>View name</td></tr></table></td></tr>
<tr><td>Describe a view</td><td>DESCRIBE VIEW <<catalog_name.schema_name.view_name>>
Response :
<table><tr><th>Attribute</th><th>Value</th></tr><tr><td>Catalog</td><td>Catalog name</td></tr><tr><td>Schema</td><td>Schema name</td></tr><tr><td>View</td><td>View name</td></tr><tr><td>Description</td><td>View description</td></tr><tr><td>View Definition</td><td>View creation grammar</td></tr><tr><td>Created by</td><td>User that created the view</td></tr><tr><td>Created on</td><td>Date and time view was created</td></tr></table></td></tr>
</table>

DML Queries

You can run select, insert and delete queries on data using Oracle AI Data Platform notebooks, SL, Python, and Spark scripts.

Oracle AI Data Platform Workbench currently supports Spark 3.5 with Delta Lake 3.2.0. For more information on DML queries, see:

- Apache Spark SQL Syntax - DML Statements
- Delta Lake - Table deletes, updates, and merges
- Delta Lake - Table utility commands
- Delta Lake - Use liquid clustering for Delta tables