Model Registry (Preview)

Date published: 2023-01-31
Date modified: 2023-08-02

Legal Notice

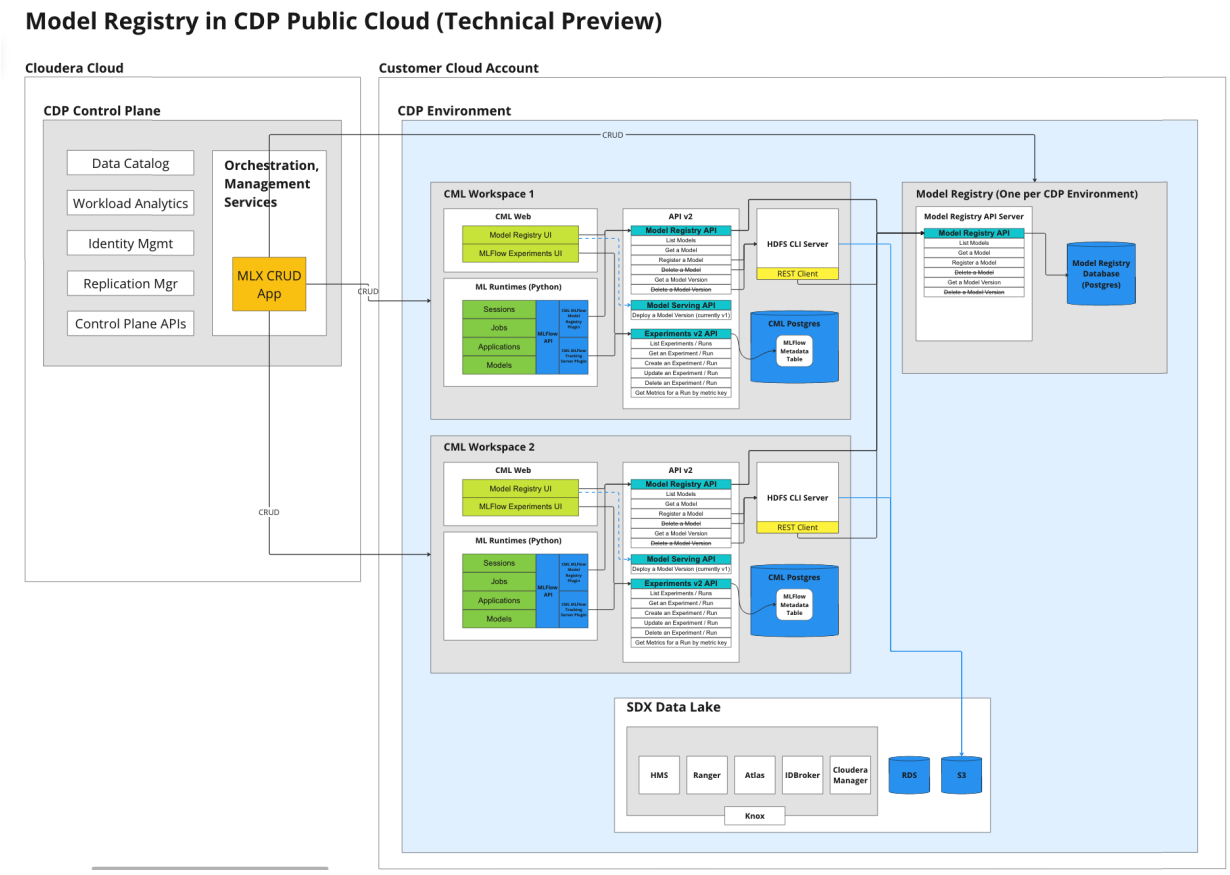
This document has been released as part of a technical preview for features described herein. Technical preview components are provided as a convenience to our customers for their evaluation and trial usage. These components are provided ‘as is’ without warranty or support. Further, Cloudera assumes no liability for the usage of technical preview components, which should be used by customers at their own risk.

Contents

Overview

The Model Registry is the core enabler for MLOps, or DevOps for machine learning.

The Model Registry stores and manages machine learning models and associated metadata, such as the model's version, dependencies, and performance. The registry enables MLOps and facilitates the development, deployment, and maintenance of machine learning models in a production environment.



Model Registry includes functionality for the following tasks:

- Storing and organizing different versions of a machine learning model and its associated metadata.
- Tracking the lineage of a model, including who created it, when it was created, and any changes made to it over time.
- Providing APIs for accessing and deploying models, as well as for querying and searching the registry.
- Integrating with CI/CD pipelines and other tools used in the MLOps workflow.

Model registries help organizations improve the quality and reliability of their machine learning models by providing a centralized location for storing and managing models, as well as enabling

traceability and reproducibility of model development. They also make deploying and managing models in a production environment easier by providing a single source for model versions and dependencies.

The Model Registry integrates MLFlow and maintains compatibility with the open source ecosystem.

Limitations

- Upgrade to the GA (General Availability) version of Model Registry might not be supported. Alternatively, to upgrade to the GA version of Model Registry might require reinstalling Model Registry which could result in loss of Model Registry data configured with the technical preview version of Model Registry.

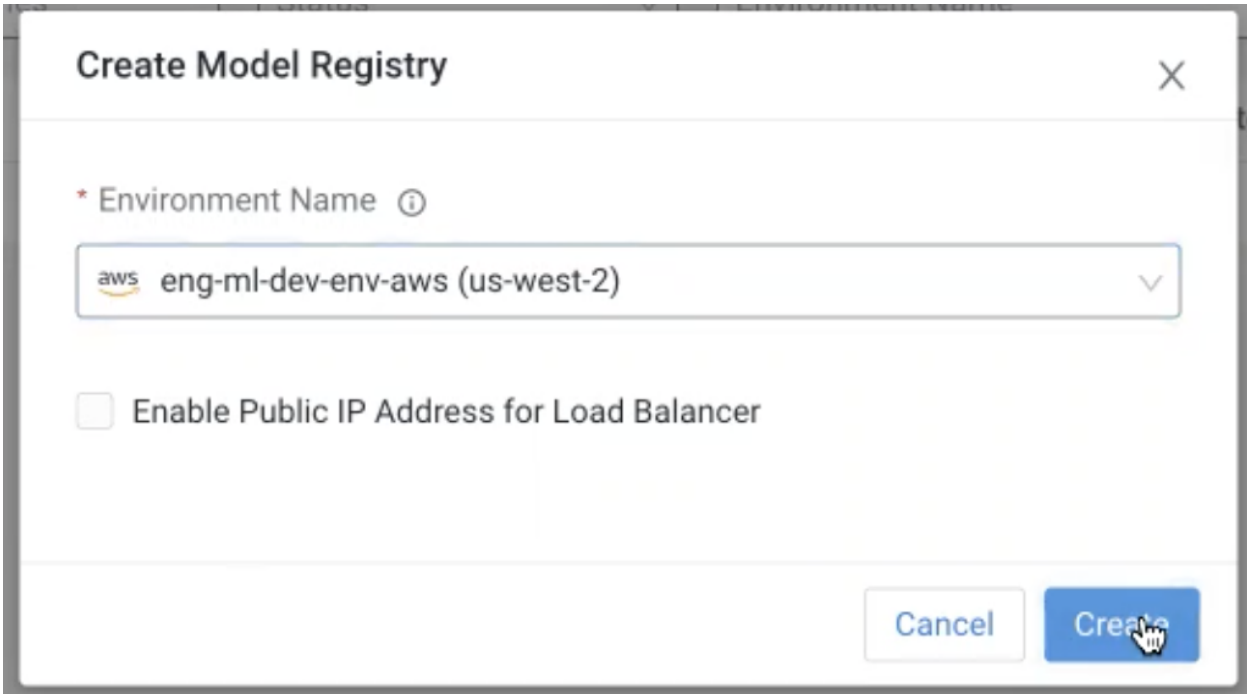
Setting up Model Registry

Prerequisites

- Before you can start using Model Registry you must have the Model Registry entitlement which is **ML_MODEL_REGISTRY**.
- You must have permission to access a project in which the model is created before you can register it.

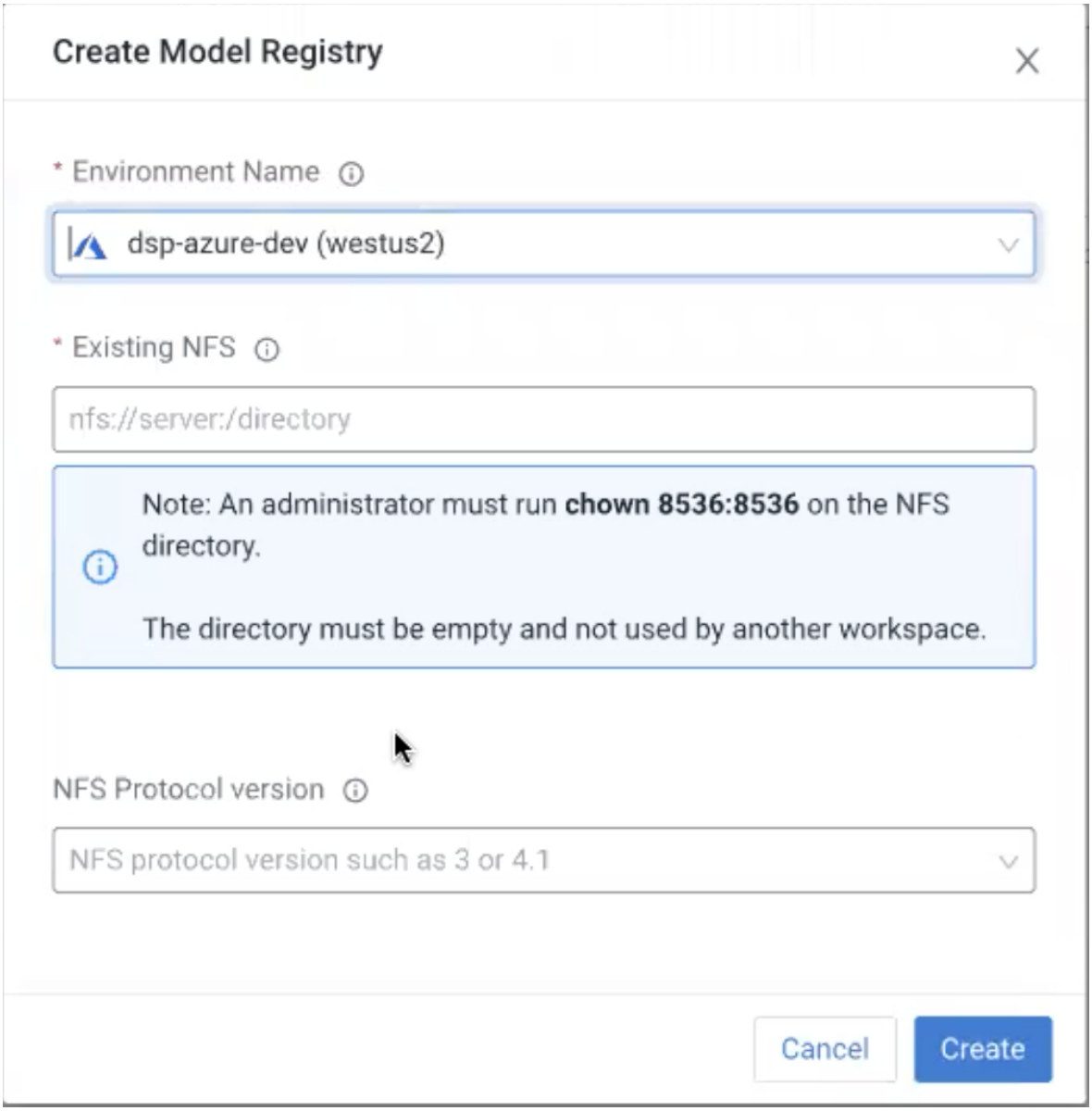
Creating a Model Registry

1. Navigate to Machine Learning on the CDP Control Plane.
2. Click **Model Registries** in the left navigation pane.
3. Click **Create Model Registry**.
4. Choose your environment from the **Environment Name** drop down list.
5. Depending on your environment, complete one of the following:
    a. If your environment is in AWS, Model Registry displays the following dialog box:



i. Click **Create** to create the Model Registry.

b. If your environment is in Azure, Model Registry displays the following dialog box:



i. Enter your NFS directory in the **Existing NFS** field.
ii. Choose the NFS Protocol version.
iii. Click **Create** to create the Model Registry.

Creating a Model Registry on an Azure UDR Private Cluster

Use the following template CDP CLI command to create a UDR private cluster on Azure with a Model Registry. You must replace the following template items with your own information.

- <environment CRN>
- <environment name> (in two places)
- <existing NFS name>
- <subnet>

Model registries are also supported on Azure private clusters with UDR. For more information about UDR, see the Preview Feature documentation.

If you have not yet downloaded the CDP CLI tool, see the documentation.

The required CDP CLI version is version 0.9.93 or higher.

CDP CLI command to create a Model Registry

This CDP CLI command performs has three key sections:

1. Enables support for private clusters in Azure ("privateCluster": true,)
2. Enables UDR for the private cluster ("outboundTypes": ["OUTBOUND_TYPE_UDR"],)
3. Specifies the subnet for the UDR-enabled private cluster ("subnets")

```
cdp ml create-model-registry --cli-input-json '{
  "environmentCrn": "<environment CRN>",
  "environmentName": "<environment name>",
  "createWorkspacePayload": {
     "environmentName": "<environment name>",
     "workspaceName": "modelregistry",
     "privateCluster": true, # This setting enables the support for private cluster in azure.
     "outboundTypes": ["OUTBOUND_TYPE_UDR"], # Required for enabling UDR.
     "skipValidation": true,
     "disableTLS": false,
     "disableSSO": false,
     "existingNFS": "<existing NFS name>",
     "nfsVersion": "3",
     "xEntitlements": [
        "ML_MODEL_REGISTRY",
        "ML_ENABLE_PRIVATE_CLUSTER"
     ],
```

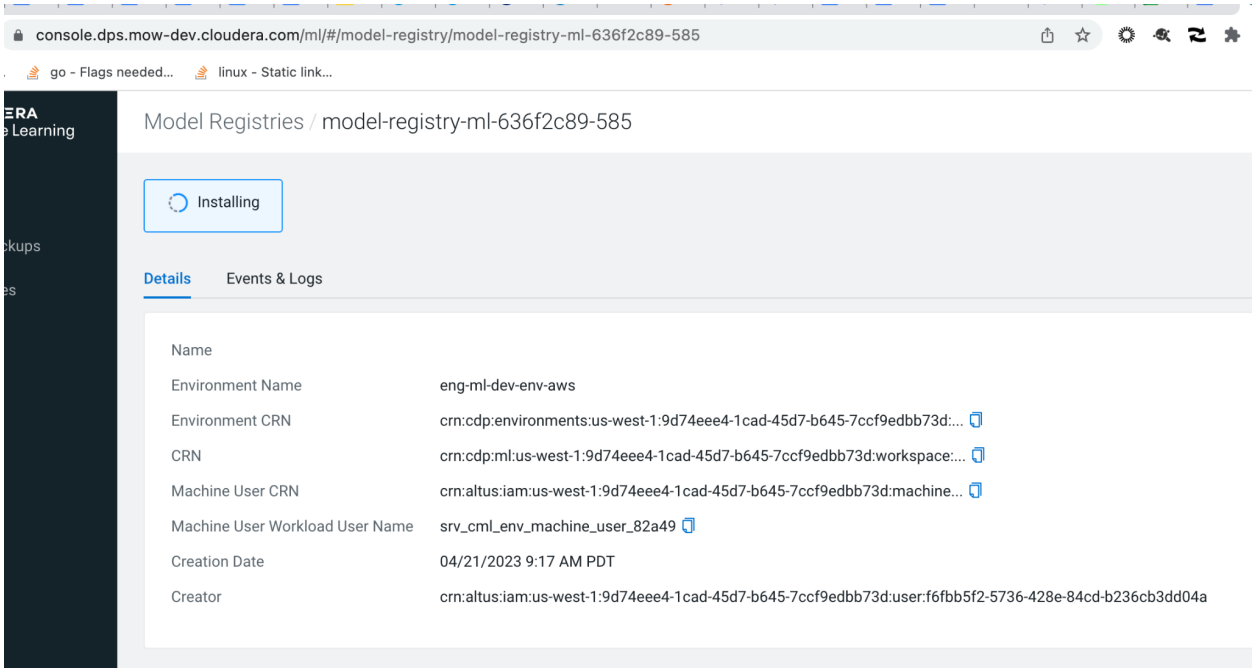
```
   "provisionK8sRequest": {
     "environmentName": "<environment name>",
     "network": {
       "topology": {
         "subnets": [
           "<subnet>" # subnet with a default route configuration to forward the traffic to the
network
appliance or firewall. This is required to enable UDR.
         ]
       }
     },
     "instanceGroups": [{
       "autoscaling": {
          "minInstances": 1,
          "maxInstances": 5
         },
         "instanceType": "Standard_DS5_v2",
         "rootVolume": {
           "size": 256
         }
       }
       ]
   }
  }
}' --profile eu-stage
```

Setting up access for Model Registry in a RAZ-enabled environment

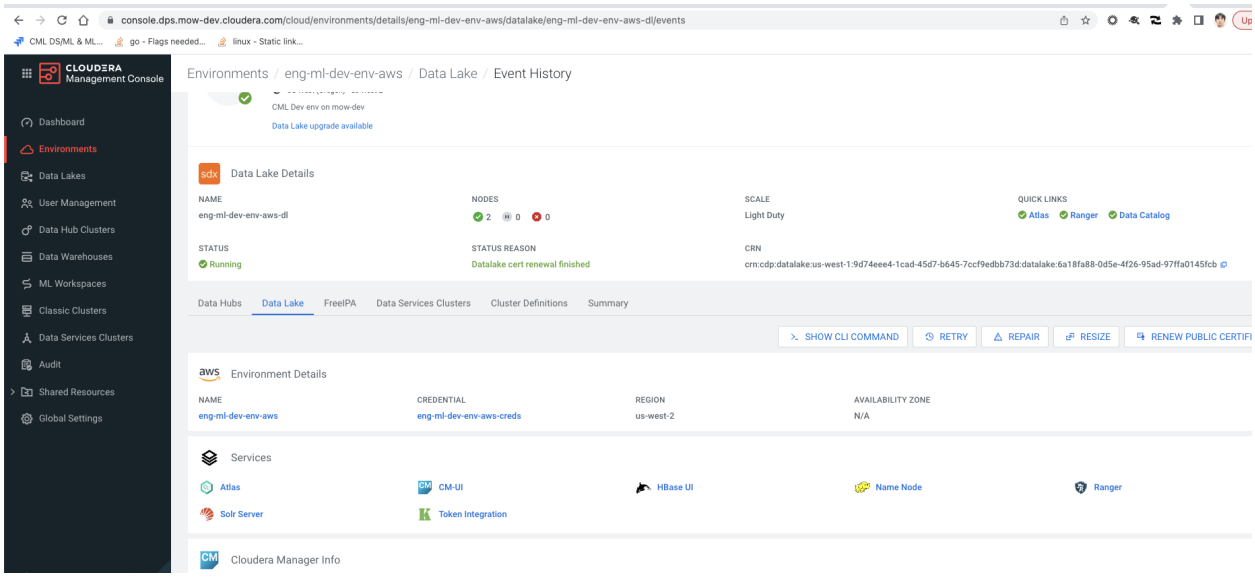
In a RAZ-enabled environment you need to set up the S3-Ranger policy by manually adding the machine user name in the S3 Ranger policy.

To set up the S3-Ranger policy, complete the following:

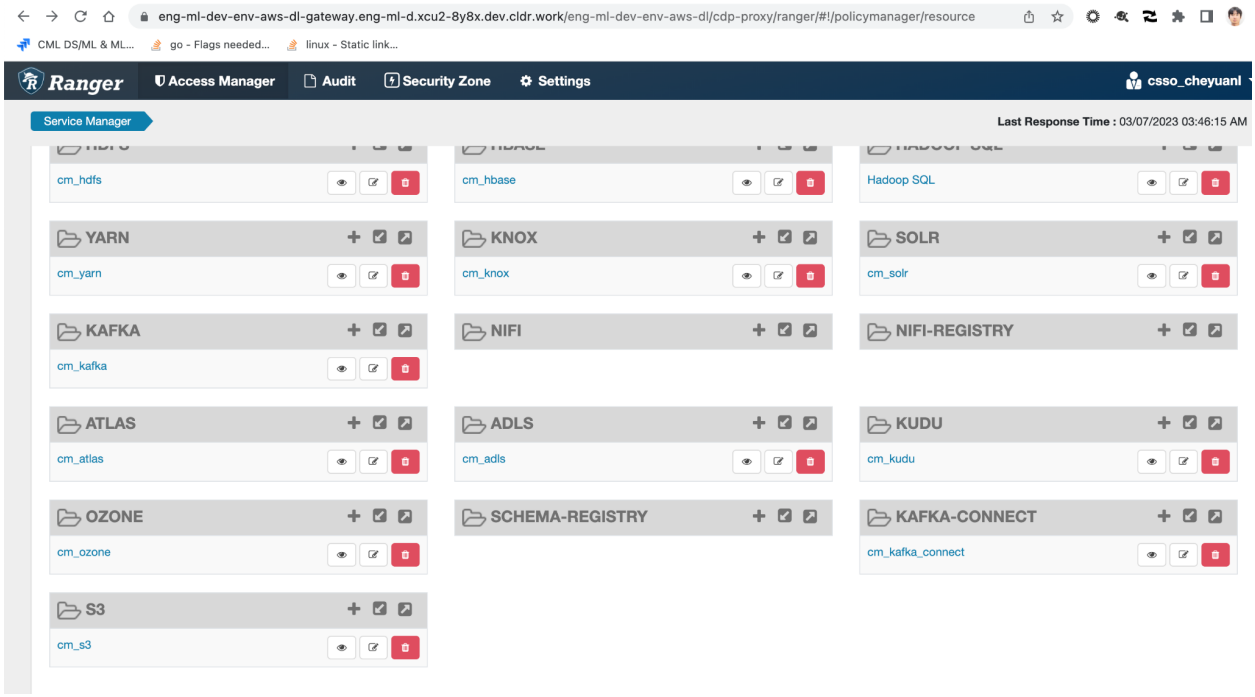
1. On the **Models Registry Details** page, find and copy the Machine User Workload User Name in the **Machine User Workload User Name** field.
   For example, in the following screenshot, the **Machine User Workload User Name** field contains srv_cml_env_machine_user_82a49. Copy the Machine User Workload User Name which is **82a49**.



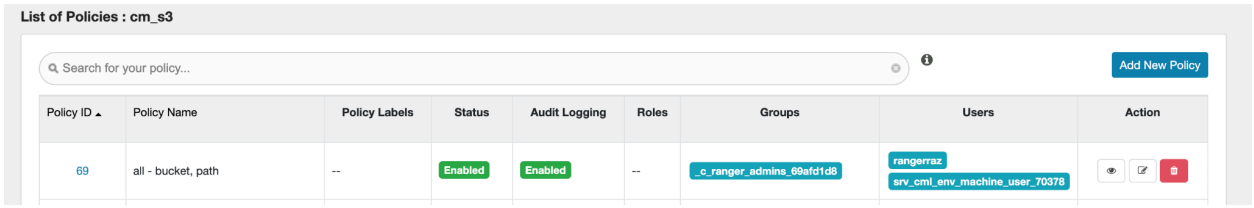
2. Go to the Ranger UI in the Datalake of the environment.



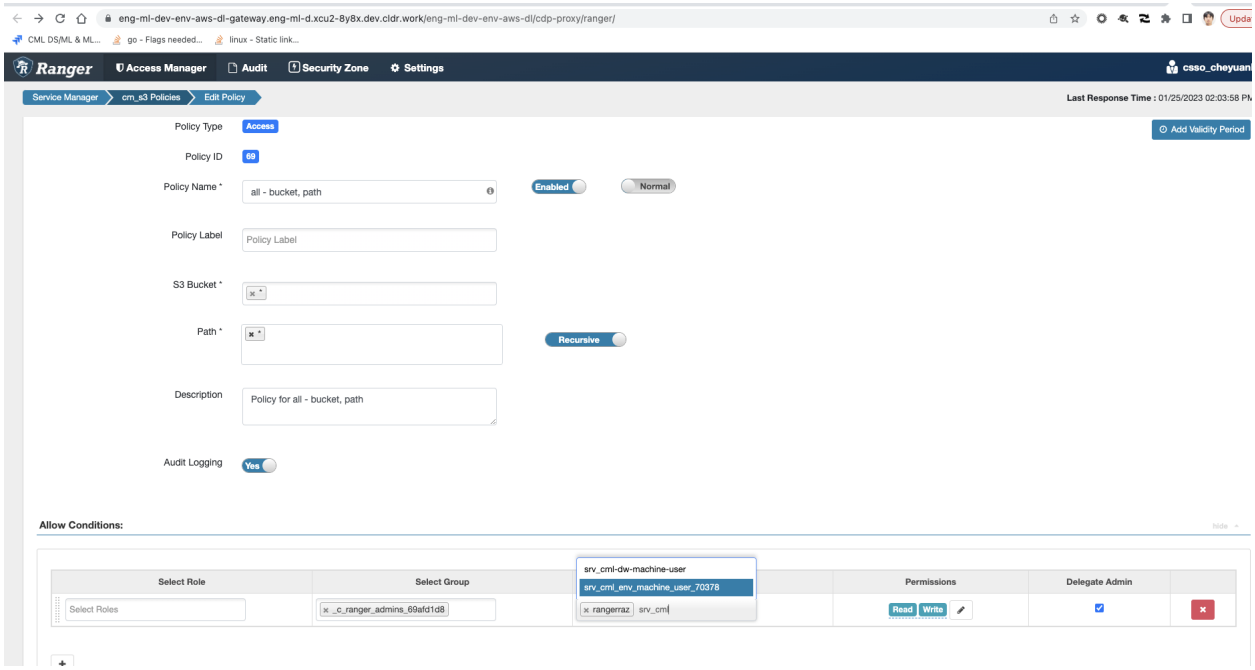
3. Depending on your environment, select **cm_s3** (AWS) or **cm_adls** (Azure).



4. Go to the policy named **all - bucket, path** which controls the access to the object store bucket.



5. Enter the Machine User Workload User Name to the **Select User** field in the **allow conditions** section.
   For example, using the Machine User Workload User Name from Step 2, add the value which is **82a49.**

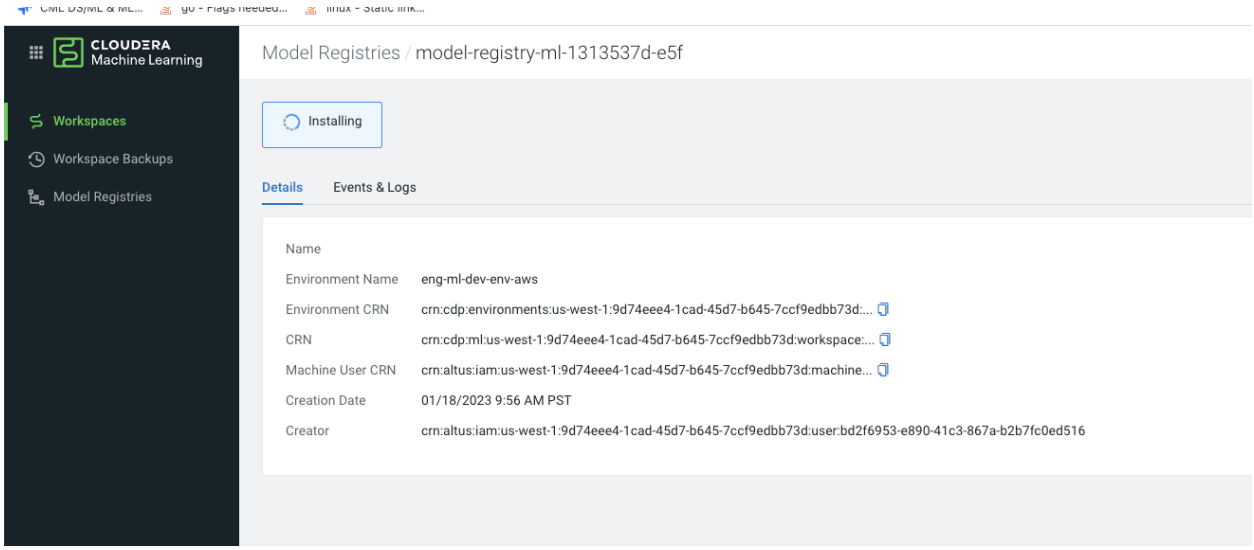


Setting up access for Model Registry in a non-RAZ-enabled environment

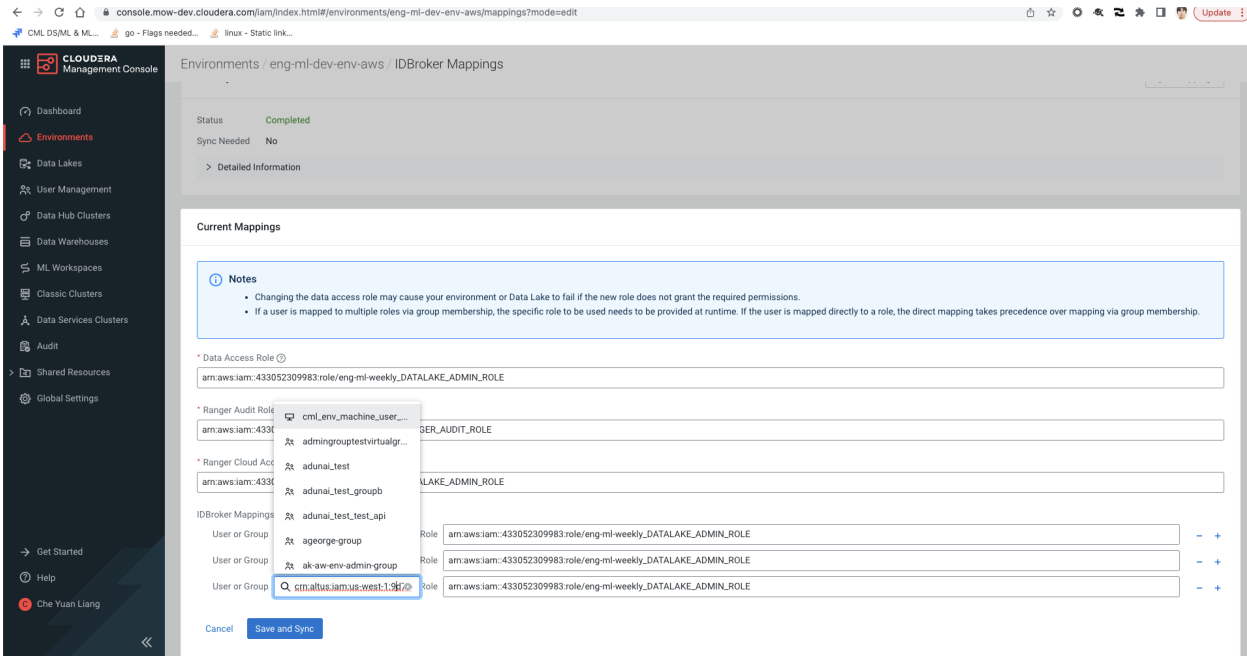
In a non-RAZ-enabled environment you need to add the Machine User CRN to the IDBroker mapping in order to access the S3/ADLS buckets.

To add the Machine User CRN to the IDBroker mapping complete the following:

1. Locate the Machine User CRN in the **Model Registry Details** page.



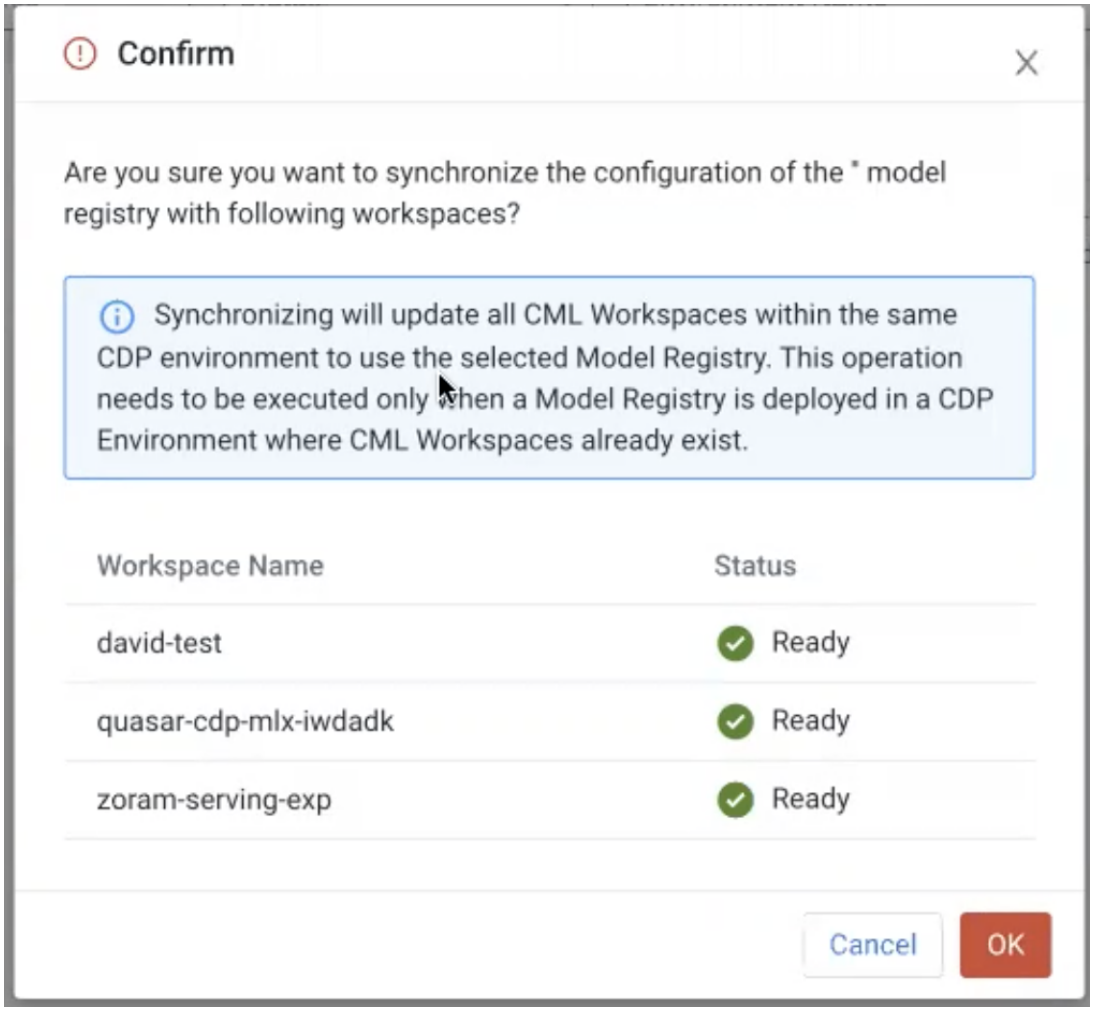
2. Copy the **entire** Machine User CRN mapping.
3. Navigate to the **Environment->Manage access->idbroker** page and add or choose the Machine User CRN mapping to the **Data Access Role** field.
4. Click **Save and Sync**.



Synchronizing the model registry with a workspace

If you deploy a model registry in an environment that contains one or more CML workspaces, you must synchronize the model registry with the workspaces.

1. Click **Model Registry** to display the **Model Registries** window.
2. Choose the registry model you want to synchronize with the workspaces in the environment.
3. From the **Actions** menu, click **Synchronize**.
   Model Registry displays the **Confirm** dialog box listing all of the workspaces in the environment.

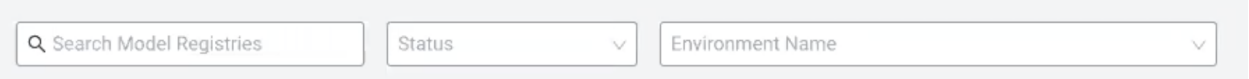


4. Click **OK**.

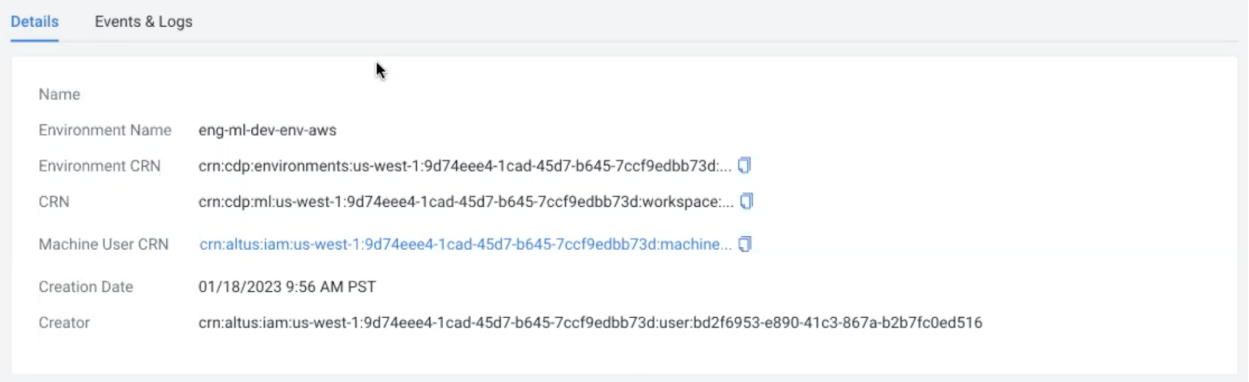
Viewing Details for Model Registry

You can view detailed information for Model Registry.

1. Select **Model Registry** from the left navigation pane.
   On the main **Model Registry** page, you can see all the models currently registered, their environment name, respective owners, location of creation, and the last updated time, if known.
2. You can use the filter bar at the top of the window to filter the list of model registries by name, status, and environment name.



3. Select a model registry to see its description.
   CML displays the **Details** page which lists the environment name, environment CRN, CRN, machine user CRN, creator, and creation date.



4. You can also click the **Events & Logs** tab to display information on the events and logs for the model registry.

Model Registry permissions

Model registry permissions for the following actions are separate from workspace permissions, but they are inherited from environment level workspace permissions.

- create
- delete
- getKubeconfig
- grant/ list/revoke access

Therefore, if you have the MLAdmin role on an environment, you can perform these actions for model registry, but an MLUser cannot.
Remote access to a model registry works similarly to workspace remote access. In addition to downloading the kubeconfig file, you need to use Grant/List/RevokeModelRegistryAccess endpoints to manage what cloud user identity can access the Kubernetes cluster using your cloud credential.

Deleting Model Registry

If you no longer want access to Model Registry, you can delete it.

1. Click **Model Registry** to display the **Model Registries** window.
2. Choose the registry model you want to delete.
3. From the **Actions** menu, choose **Delete**.
   Model Registry displays the **Confirm** dialog box.
4. Click **OK**.

Registering and deploying a registered model

Creating a model file using MLflow

Refer to MLflow documentation on how to create a model file:
https://docs.cloudera.com/machine-learning/cloud/experiments/topics/ml-exp-v2-mlflow-model-artifact.html

Registering a model using the User Interface

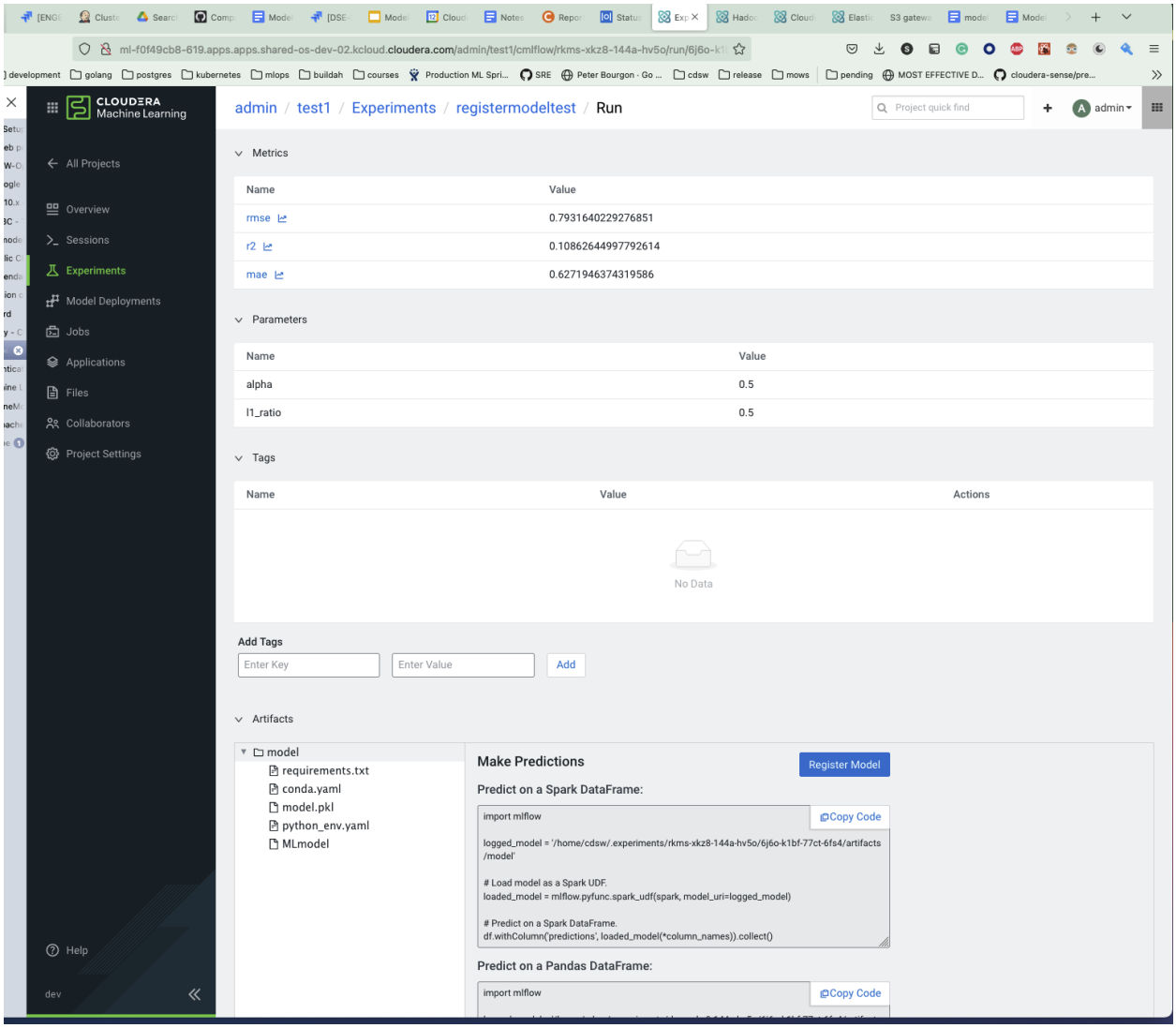
You can register a model using the user interface or the MLFlow SDK.

Registering a model enables you to upload and share the model. Registering a model stores the model archives in the model registry with a version tag. The first time you register a model, Model Registry automatically creates a model repository with the first version of the model.

Prerequisites:

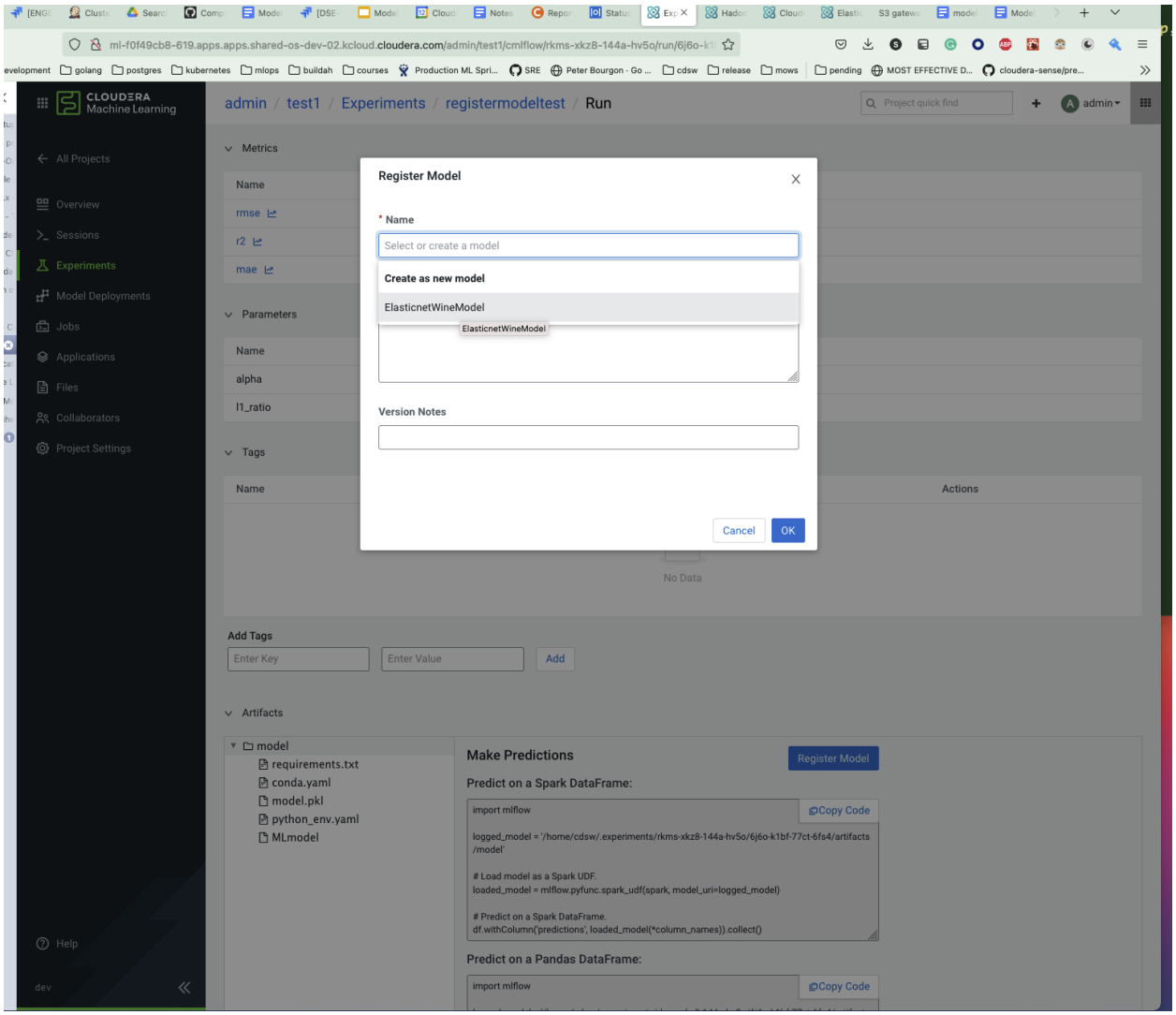
- You must have permission to access a project in which the model is created before you can register it.

1. Click **Projects** in the left navigation pane to display the **Projects** page.
2. Select the project that contains the model you want to register.
   CML displays all of the models under the specific project along with their source, deployment status, replicas, memory, and a drop-down function for actions that can be made pertaining to that model for deployment.
3. Click the **Experiments** tab in the left navigation pane and select the experiment that contains the model you want to register.
   The system displays the **Experiment Detail** page.



4. Select the run that contains the model you want to register.

5. Select **Register Model** to begin the registration process.
   Register Model displays the **Register Model** dialog box.



6. Enter the name for your registered model.
   You can also enter optional information for the description, version notes, version tags.
7. Click **OK** to complete the registration.

Registering a model using the MLflow SDK

You can also register a model using the MLflow SDK or the user interface.

Registering a model enables you to upload and share the model. Registering a model stores the model archives in the model registry with a version tag. The first time you register a model, Model Registry automatically creates a model repository with the first version of the model.

To register a model using MLflow SDK you must specify the `registered_model_name` and assign a value.:

```
mlflow.<model_flavor>.log_model()
```

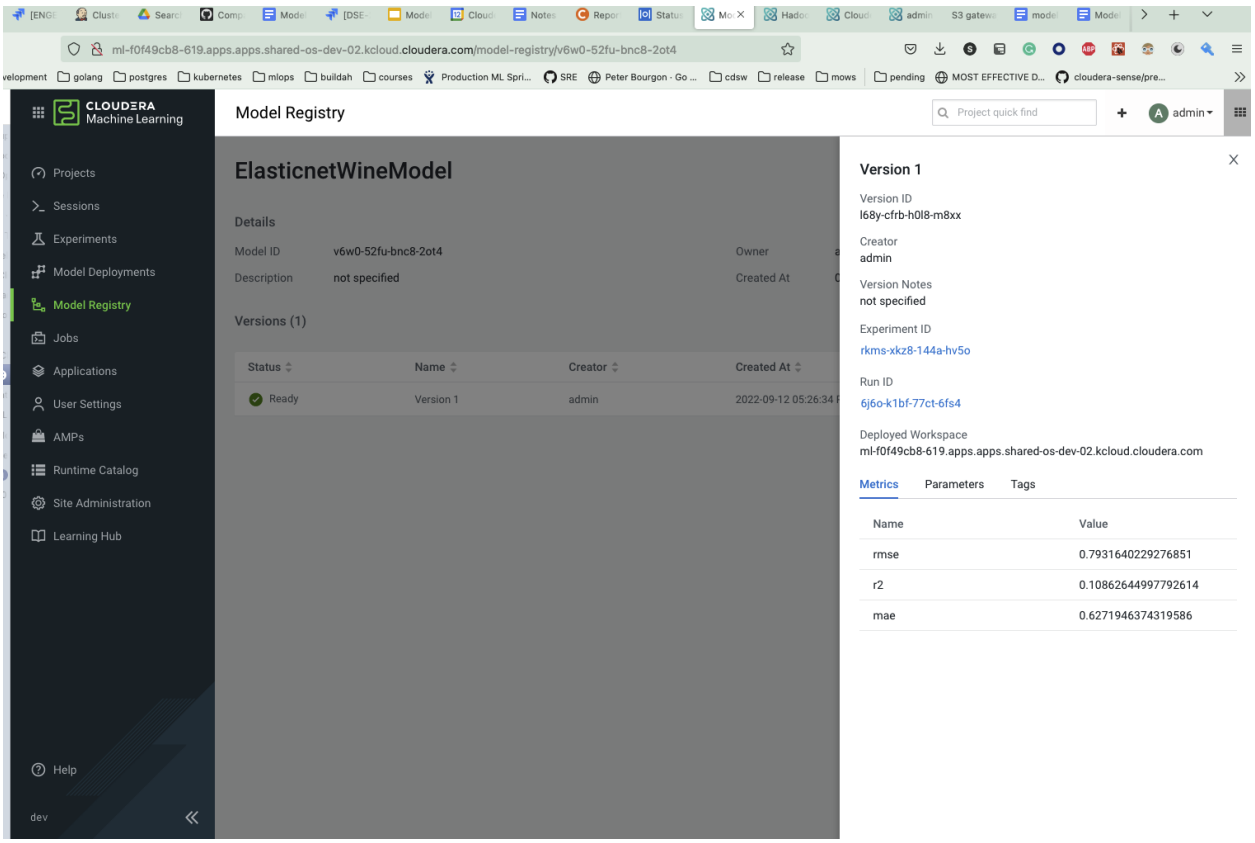
For example:

```
mlflow.sklearn.log_model(lr, "model", registered_model_name="ElasticnetWineModel")
```

If you run this Python code again with the same `model_name` it will create an additional version for the `model_name`.

Viewing registered model information

1. From the **Projects** page in CML, select **Model Registry** from the navigation pane.
   On the main **Model Registry** page, you can see all the models currently registered, their respective owners, location of creation, and the last updated time, if known.
2. Select a registered model to see its description.
   CML displays the **Details** page which outlines the model description, ID, owner, and versions. Different versions of the same model can be deployed in the workspace.



Creating a new version of a registered model

You can easily create a new version of a registered model.

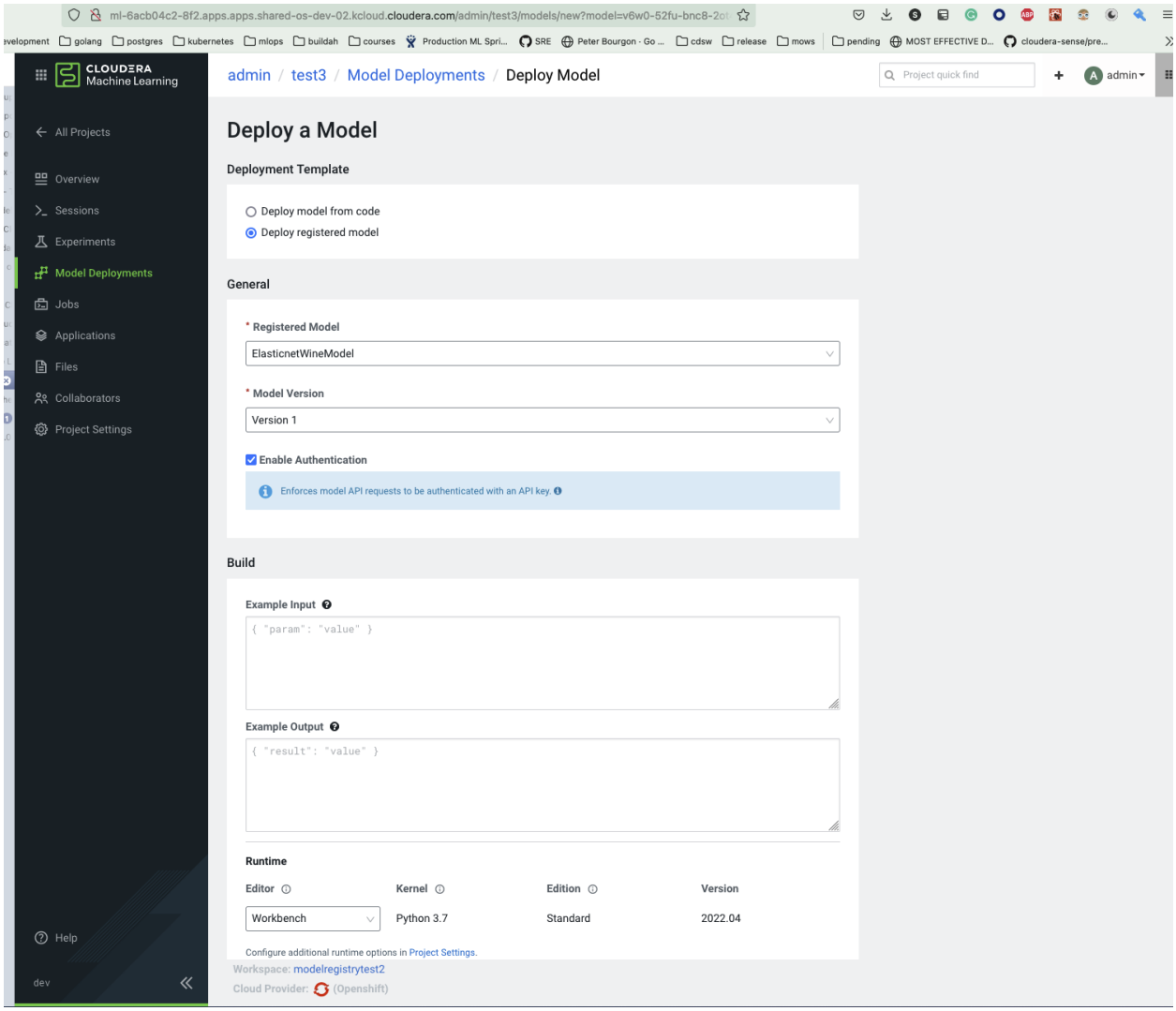
1. Click **Projects** in the left navigation pane to display the **Projects** page.
2. Select the project that contains the model for which you want to create a new version.
3. Click **Experiments** in the left navigation pane and select the experiment that contains the model you want to register.
   The system displays the **Experiment Detail** page.
4. Select the run that contains the model you want to register.
5. Scroll down the page to find the **Artifacts** section and click **model**.
6. Click **Register Model**.
7. From the **Name** field, choose the model for which you want to create a new version.
8. Click **OK**.

You can also create a new model version using MLflow SDK. Simply run the Python code to register a model again with the same `model_name`. This will create an additional version for the `model_name`.

Deploying a model from the Model Registry page

You can deploy a model one or more times to create different versions of the model. You can also deploy a model you created in one workspace to a different workspace.

1. Select **Model Registry** from the left navigation pane.
2. Select the model you want to deploy.
   Model Registry displays the **Model Version List** page.
3. Select the model version you want to deploy.
   Model Registry displays a side window that lists the version information. Dismiss this window to proceed.
4. Under the **Actions** menu, click **Deploy**.
5. Select the Project you want to deploy to in the dialog box and click **Go**.
   You can select either the project the model is located in or another project to deploy the model to.
   Model Registry displays the **Deploy a Model** page with the detailed model information auto-populated.



6. If you enable authentication, you will need to enter an API key to access and use the model in the case you have deployed the model to a shared project.
7. Click **OK**.

Deploying a model from the destination project page

You can deploy a model one or more times to create different versions of the model. You can also deploy a model you created in one workspace to a different workspace.

1. Navigate to the Project you want to deploy to.
2. Click **Model Deployments** in the left navigation pane.
3. Make sure you have clicked the **Deploy registered model** checkbox at the top of the window.
4. Select the registered model you want to deploy from the **Registered Model** field.

5. If you enable authentication, the user will need to enter an API key to access and use the model in the case you have deployed the model to a shared project.
6. Select **Deploy Model** at the bottom of the window.