Cloudera AI

Using Cloudera AI Registry

Date published: 2020-07-16
Date modified: 2026-01-06

https://docs.cloudera.com/

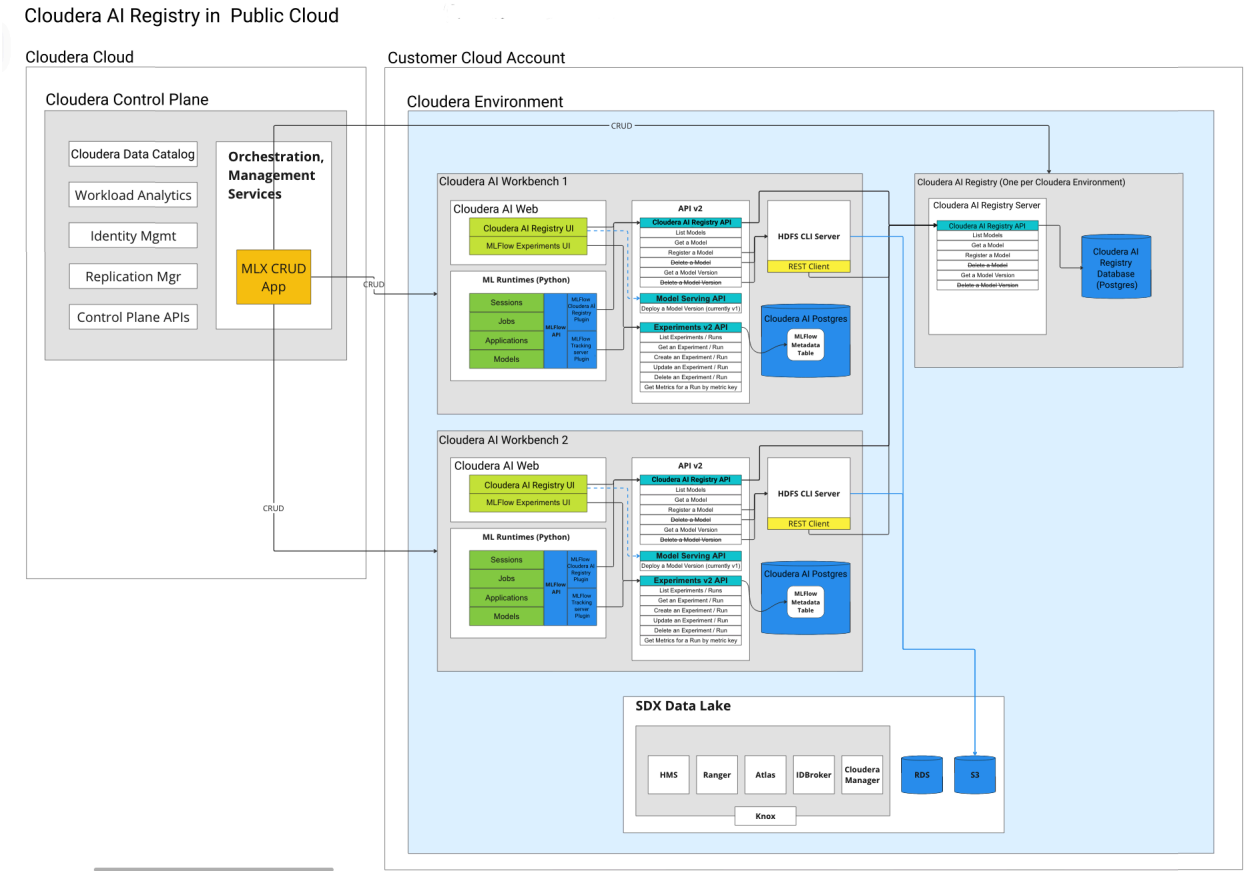
Legal Notice

Contents

Using Cloudera AI Registry

Cloudera AI Registry is the core enabler for MLOps, or DevOps for machine learning.

Cloudera AI 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.



Cloudera AI 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.

Cloudera AI Registry instances 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 Cloudera AI Registry integrates MLFlow and maintains compatibility with the open source ecosystem.

Limitations

- Upgrade to the General Availability (GA) version of Cloudera AI Registry might not be supported. Alternatively, upgrade to the GA version of Cloudera AI Registry might require reinstalling Cloudera AI Registry which could

result in loss of Cloudera AI Registry data configured with the technical preview (TP) version of Cloudera AI Registry.

Cloudera AI Registry standalone API

You can use the standalone Cloudera AI Registry API to communicate with the Cloudera AI Registry using the REST client or CLI client.

The Cloudera AI Registry standalone API supports the following functionalities:

- GET/PATCH/DELETE for the model and model version
- GET a curated list of NGC models
- Import external model from NVIDIA NGC or HuggingFace to Cloudera AI Registry through the POST method

Currently, the Cloudera AI Registry Standalone API does not support uploading the models through POST method from the local machine.

Cloud Platforms

Cloudera AI Registry API is available only on AWS and Azure.

API definition

The Swagger definition is available in the Cloudera AI API documentation.

Prerequisites for Cloudera AI Registry standalone API

To set up the Cloudera AI Registry standalone API, configure the Cloudera AI Inference service and import pretrained Models.

Prerequisites for Cloudera AI Inference service

Cloudera AI Registry is a prerequisite for Cloudera AI Inference service because the Cloudera AI Inference service needs to deploy the models that are stored in the Cloudera AI Registry.

- To use the Cloudera AI Inference service, the latest Cloudera AI Registry must be present in the same Cloudera environment before the Cloudera AI Inference service is created.
- If there is an older Cloudera AI Registry in the environment that is created before May 14, 2024, follow the *Upgrade Cloudera AI Registry* instructions to upgrade the Cloudera AI Registry to the latest version before you create the Cloudera AI Inference service.
- If the Cloudera AI Registry is recreated, upgraded, or cert-renewed while the Cloudera AI Inference service is present, then follow the steps listed in the *Manually updating Cloudera AI Registry configuration* topic to ensure that the configuration of Cloudera AI Registry and Cloudera AI Inference service are synchronized.

Prerequisites to import pretrained models

You must add the URL details to allow them in the firewall rules.

NVIDIA GPU Cloud (NGC)

Add the following URL details so they can be allowed in the firewall's rules.

- prod.otel.kaizen.nvidia.com (NVIDIA open telemetry)
- api.ngc.nvidia.com
- files.ngc.nvidia.com

Hugging Face

Add the following URL details so they can be allowed in the firewall's rules.

- huggingface.co

- cdn-lfs.huggingface.co
- *.cloudfront.net (CDN)

Note: If required, you must allow more URLs based on your requirements.

Authenticating clients for interacting with Cloudera AI Registry API

Clients that interact with the Cloudera AI Registry Standalone API and with model endpoints must obtain a JSON Web Token (JWT) from the Cloudera control plane, which must be passed as a Bearer token in HTTP requests sent to the serving API and endpoints.

To obtain JWT, run the following Cloudera CLI command:

```
$ CDP_TOKEN=$(cdp iam generate-workload-auth-token --workload-name DE | jq -
r
   '.token')
```

In this comment, *DE* is the workload name.

Then pass CDP_TOKEN in the HTTP request header as follows

```
$ curl -H "Authorization: Bearer ${CDP_TOKEN}" <URL>
```

The token obtained using this method expires in one hour.

Role-based authorization

Cloudera AI Registry implements role-based access control.

Users must have the following roles to create an instance of the service in a Cloudera environment:

- EnvironmentAdmin
- MLAdmin (admin user)

Registered Models can be viewed, created, deleted, and modified by users having EnvironmentUser role along with either one of the following roles:

- MLAdmin (admin user)
- MLUser

For more information about the access control for the registered models, see *Model access control*.

Using the REST Client

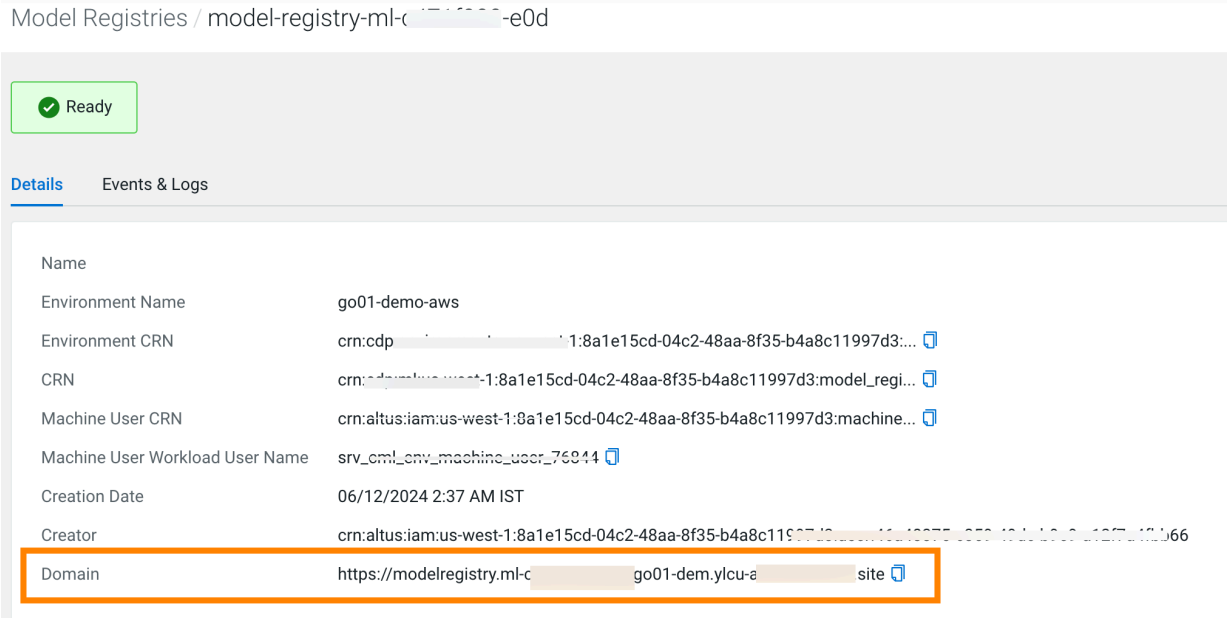
You need the domain information to use the REST client to interact with the registry.

Before you begin

To obtain the domain information, perform the following:

1. In the **Cloudera** console, click the **Cloudera AI** tile.
2. Click AI Registries in the left navigation menu. The AI Registries page displays.

3. Click on the name of the Cloudera AI Registry to display the Cloudera AI Registry information. The Domain name is displayed in the Details tab.



Get all Models

```
curl -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/models | jq
        {
        "models": [
        {
        "created_at": "2024-04-18T15:54:15.543Z",
        "creator": {
        "user_name": "csso_cheyuanl"
        },
        "id": "5bwt-qqe2-e1vg-chqj",
        "name": "foo",
        "tags": null,
        "updated_at": "2024-04-18T15:54:15.543Z",
        "visibility": "private"
        },
```

Get a Model

```
curl -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/models/fx0k
-baf7-yszl-jrt2 | jq
          {
          "created_at": "2024-04-18T15:54:24.940Z",
          "creator": {
          "user_name": "csso_cheyuanl"
          },
          "id": "fx0k-baf7-yszl-jrt2",
          "model_versions": [
          {
          "artifact_uri": "abfs://data@engmldevenvazuresan.dfs.core.windows
.net/modelregistry/fx0k-baf7-yszl-jrt2/y8d8-qluc-00md-h2pw/model.tar.gz",
          "created_at": "2024-04-18T15:54:24.942Z",
          "model_id": "fx0k-baf7-yszl-jrt2",
          "status": "READY",
          "tags": null,
          "updated_at": "2024-04-18T15:54:24.942Z",
```

```
          "user": {
          "user_name": "csso_cheyuanl"
          },
          "version": 1
          }
          ],
          "name": "foo2",
          "tags": null,
          "updated_at": "2024-04-18T15:54:24.940Z",
          "visibility": "private"
          }
```

Get a Model version

```
curl -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/models/fx0k
-baf7-yszl-jrt2/versions/1 | jq
          {
          "artifact_uri": "abfs://data@engmldevenvazuresan.dfs.core.windows.
net/modelregistry/fx0k-baf7-yszl-jrt2/y8d8-qluc-00md-h2pw/model.tar.gz",
          "created_at": "2024-04-18T15:54:24.942Z",
          "model_id": "fx0k-baf7-yszl-jrt2",
          "status": "READY",
          "tags": null,
          "updated_at": "2024-04-18T15:54:24.942Z",
          "user": {
          "user_name": "csso_cheyuanl"
          },
          "version": 1
          }
```

Patch a Model

```
curl -XPATCH -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/mod
els/fx0k-baf7-yszl-jrt2 -d '{
  "visibility": "public"
}'
```

Patch a Model Version

```
curl -XPATCH -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/mod
els/fx0k-baf7-yszl-jrt2/version/1 -d '{
  "tags": [{"key": "k1", "value": "v1"}, "key": "k2", "value": "v2"}]
}'
```

Delete a Model

```
curl -XDELETE -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/mo
dels/vuu6-gcfx-ydio-rit0
```

Delete a Model version

```
curl -XDELETE -s -H "Authorization: Bearer ${CDP_TOKEN}" ${DOMAIN}/api/v2/mo
dels/vuu6-gcfx-ydio-rit0/versions/1
```

Cloudera AI Registry CLI Client

Cloudera AI Registry client is a command line tool (CLI) that can be used to interact to a Cloudera AI Registry server. It can be downloaded from any Cloudera AI Registry server <domain>/apiv2/cli/<os>. Here, <os> is either Linux, Darwin for Mac, or Windows based on the operating system the Cloudera AI Registry CLI is installed on.

The swagger CLI is downloaded from `https://<domain>/apiv2/cli/<os>`. The following are some of the example usage of CLI.

Usage

After you download the CLI and add it to the path, you can use the `modelregistrycli` commands.

```
modelregistrycli help
Usage:
  modelregistrycli [command]

Available Commands:
  completion  Generate completion script
  help        Help about any command
  operations
Flags:
      --Authorization string
      --config string          config file path
      --debug                  output debug logs
      --dry-run                do not send the request to server
  -h, --help                   help for modelregistrycli
      --hostname string        hostname of the service (default "localhost")
      --scheme string          Choose from: [http] (default "http")
```

Create a Model

Create an imported model request

```
$ modelregistrycli --Authorization "Bearer nil" --hostname localhost:8188 op
erations CreateModel --body '{
    "name": "tiny",
    "createModelVersionRequestPayload": {
        "metadata": {
            "model_repo_type": "HF"
        },
        "downloadModelRepoRequest": {
            "source": "HF",
            "repo_id": "prajjwal1/bert-tiny"
        }
    }
}'
***Output:***

{"created_at":"2024-04-03T18:02:15.331Z","creator":{"user_name":"admin"},
"description":"model to classify catAndDogClassifier","id":"1w6s-8m6t-ngdr-3
qvr","model_versions":null,"name":"catAndDogClassifier","tags":[{"key":"cat"
,"value":"1"}],"updated_at":"2024-04-03T18:02:15.331Z","visibility":"public"
}
```

Get Models

```
$ modelregistrycli --Authorization "Bearer nil" --hostname localhost:8188 op
erations GetModels

***output***:
```

```
{"models":[{"created_at":"2024-04-03T18:02:15.331Z","creator":{"user_name
":"admin"},"description":"model to classify catAndDogClassifier","id":"1w6s-
8m6t-ngdr-3qvr","model_versions":null,"name":"catAndDogClassifier","tags":nu
ll,"updated_at":"2024-04-03T18:02:15.331Z","visibility":"public"},{"created_
at":"2024-04-03T18:08:43.130Z","creator":{"user_name":"admin"},"description"
:"create request model request with model version example","id":"8fts-rgpn-r
9xo-xlh0","model_versions":null,"name":"chain-classifier","tags":null,"updat
ed_at":"2024-04-03T18:08:43.130Z","visibility":"public"}]}
```


Get Model by ID

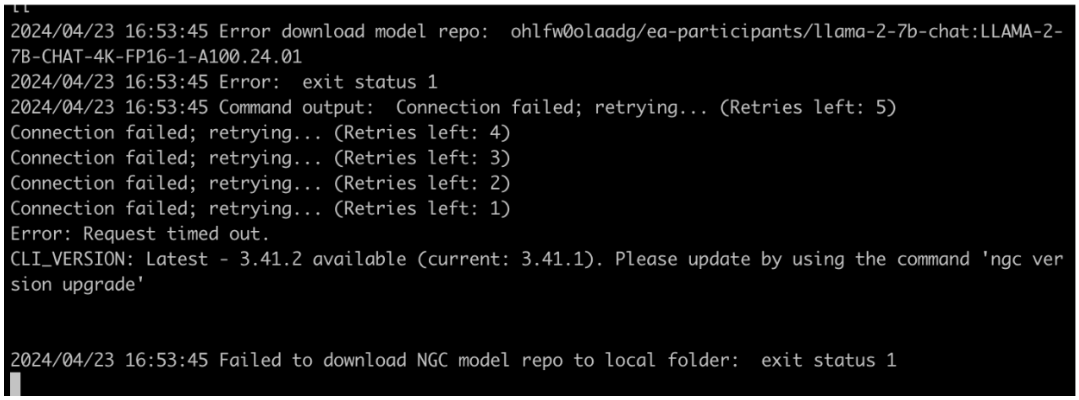

```
$ modelregistrycli --Authorization "Bearer nil" --hostname localhost:8188 op
erations GetModel --model_id '8fts-rgpn-r9xo-xlh0'
{"created_at":"2024-04-03T18:08:43.130Z","creator":{"user_name":"admin"},
"description":"create request model request with model version example","id"
:"8fts-rgpn-r9xo-xlh0","model_versions":[{"artifact_uri":"http://localhost:9
000/8fts-rgpn-r9xo-xlh0/r5eg-m0gp-i4qs-8b07/model.tar.gz","created_at":"2024
-04-03T18:08:43.132Z","model_id":"8fts-rgpn-r9xo-xlh0","notes":"create reque
st model request with model version example","status":"REGISTERING","tags":[
{"key":"chain","value":"2"}],"updated_at":"2024-04-03T18:08:43.132Z","user":
{"user_name":"admin"},"version":1}],"name":"chain-classifier","tags":null,"u
pdated_at":"2024-04-03T18:08:43.130Z","visibility":"public"}
```


Known issues with Cloudera AI Registry standalone API

These are some of the known issues you might run into while using Cloudera AI Registry standlone API.

NGC model download timeout

The NGC model import might time out, and the corresponding model version status is shown as “failed”. You can access the logs found in the API v2 pod by performing the steps mentioned in the *Debugging the model import failure* troubleshooting section.



Retry the model import request again.

Model import failure

You can download the models concurrently only if their combined size is below approximately 400 GB. Exceeding this limit may result in import failures and unexpected behavior.

Request Throttling

Currently, there is no request throttling mechanism implemented. As a result, excessive concurrent requests may lead to model import failures. To minimize the risk, it is recommended to limit concurrent requests to a maximum of 5, which is considered a safe threshold.

Model Import progress indicator

A progress bar is not available for model imports. For reference, importing a 70 GB model typically takes approximately 1 hour. Users should plan accordingly and monitor the process through alternative options, if necessary.

Troubleshooting issues with Cloudera AI Registry API

Learn about some of the recommended series of steps to perform when troubleshooting issues related to the Cloudera AI Registry API.

Cloudera AI Inference service cannot discover Cloudera AI Registry

Learn about scenarios or issues that may be resolved by using the steps in the *Manually updating Cloudera AI Registry configuration* solution.

- ML Serving installed before Cloudera AI Registry

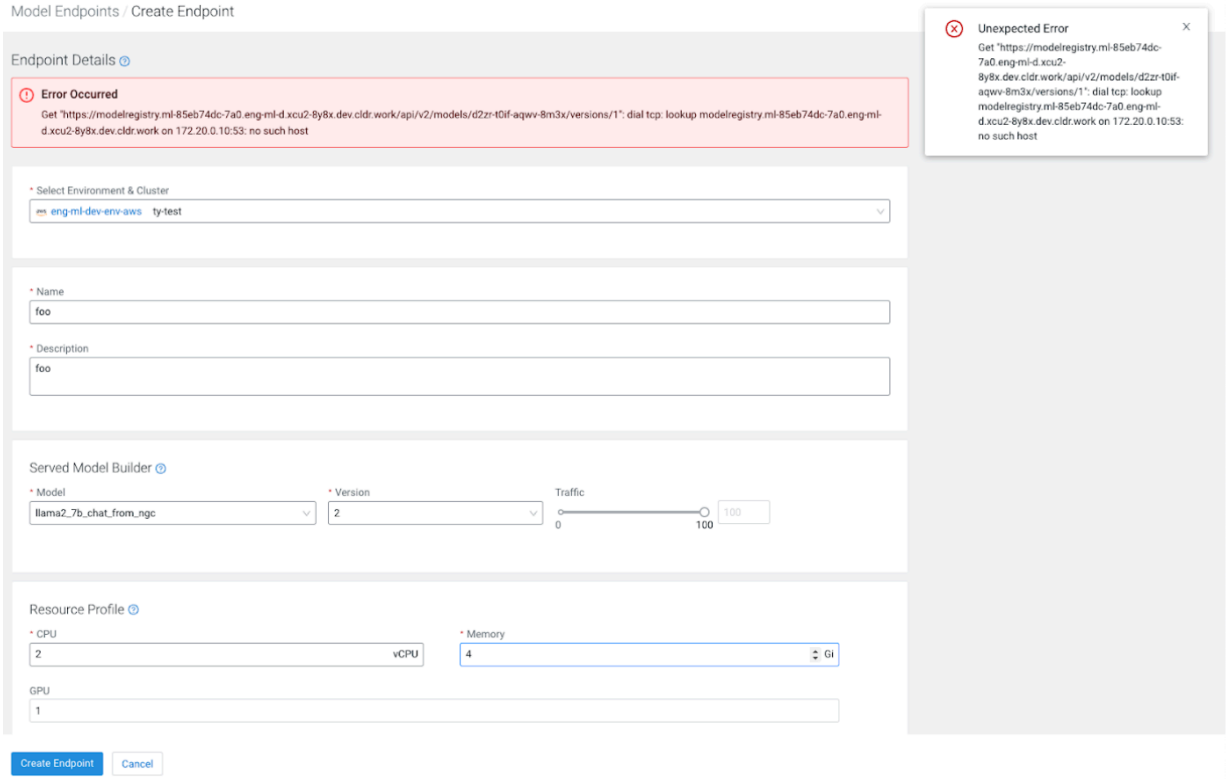
  If the Cloudera AI Registry has not been created yet, when ML Serving is installed it will not be able to generate the ConfigMap.

- Cloudera AI Registry upgraded after ML Serving installed

  When the Cloudera AI Registry is upgraded, it will require an updated ConfigMap.

- Cloudera AI Registry certificate expired

  After 90 days, Cloudera AI Registry will have a certificate update, it will require an updated ConfigMap.



Manually updating Cloudera AI Registry configuration

If you upgrade the Cloudera AI Registry after creating your Cloudera AI Inference service cluster, the Cloudera AI Registry configuration stored by the Cloudera AI Inference service will get out of synchronization. Follow the steps below to manually reconcile the configuration, so that Cloudera AI Inference service will be able to connect to Cloudera AI Registry.

Important: You are required to access multiple clusters when performing the below steps. For information configuring and switching KubeConfig between multiple clusters, see Configure Access to Multiple Clusters

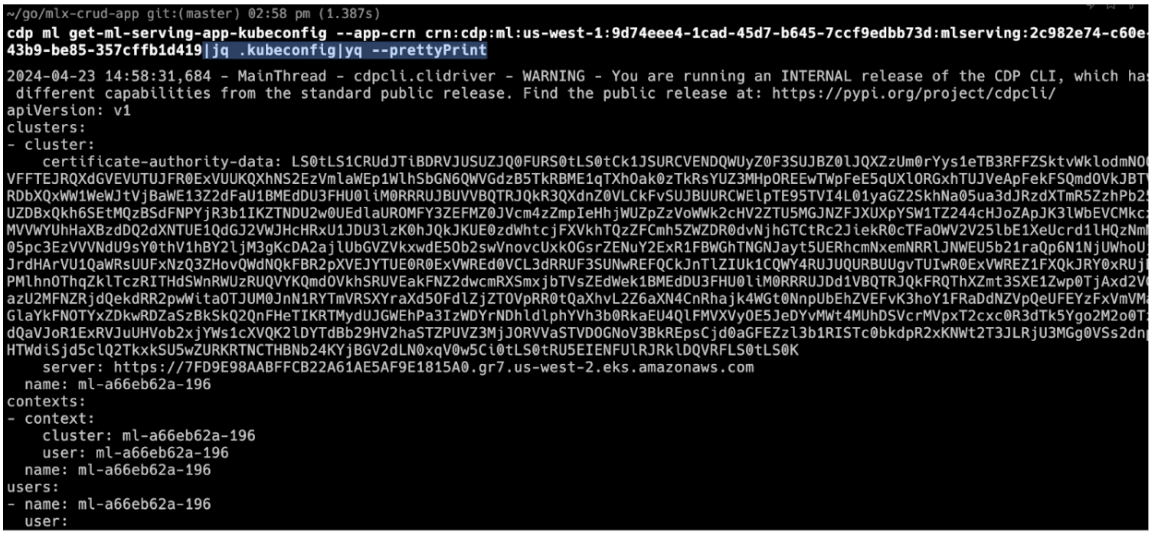
1. Get KubeConfig for ML Serving Cluster.
   a. In a terminal where Cloudera CLI is installed, run the cdp ml list-ml-serving-apps command.
   b. Find the appCrn of the ML Serving instance you wish to update.

   c. Retrieve the KubeConfig with this command:

```
cdp ml get-ml-serving-app-kubeconfig --app-crn <YOUR-APP-CRN> |jq .kubec
onfig|yq --prettyPrint
```

   Here:

   - `cdp ml get-ml-serving-app-kubeconfig --app-crn <YOUR-APP-CRN>` : returns raw json response containing KubeConfig.
   - `|jq .kubeconfig:` returns only the actual KubeConfig (removes json wrapper).
   - `|yq --prettyPrint`: prints the kKubeConfig response into a format both machine and user-readable.



   d. Direct the output to the standard KubeConfig location:

```
cdp ml get-ml-serving-app-kubeconfig --app-crn <YOUR-APP-CRN> |jq .kubec
onfig|yq --prettyPrint>~/.kube/config
```

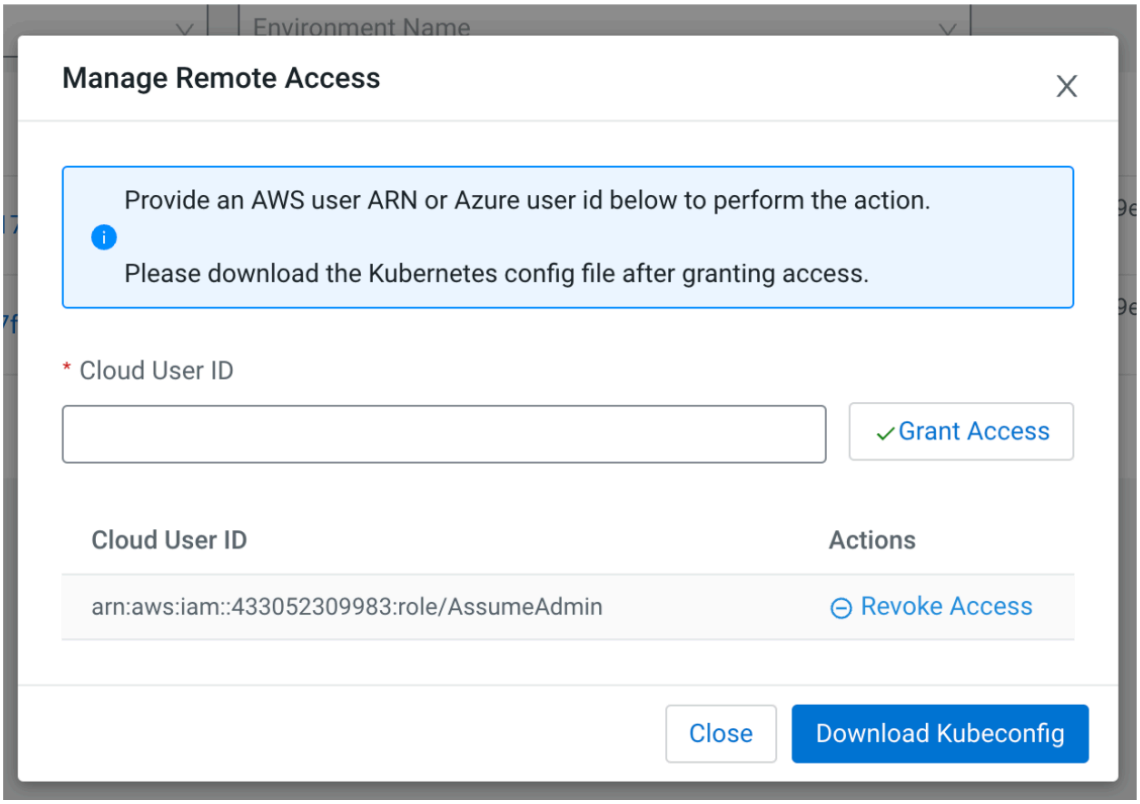
   Make sure this is the active KubeConfig by running the `kubectl config view` command. The output of the above command shall now show the KubeConfig data returned above.

2. Grant your user ML Serving App access.

   Add your cloud provider user identifier to the ML Serving App access control list. Use the following command, with your ARN, or other cloud provider user identifier.

```
cdp ml grant-ml-serving-app-access --resource-crn crn:cdp:ml:us-west-1:9
d74eee4-1cad-45d7-b645-7ccf9edbb73d:mlserving:2c982e74-c60e-43b9-be85-35
7cffb1d419 --identifier <Your Cloud Provider User ID>
```

3. Get new ConfigMap data.
   a. Find the new Cloudera AI Registry ConfigMap. Use the grant-model-registry-access API in CDP CLI to add your user name to the new Cloudera AI Registry, or use the UI, as shown:



   b. After your user ARN has been granted access to the Cloudera AI Registry, get the ConfigMap data in the following way:
      - Download the Cloudera AI Registry KubeConfig.
      - Set the Cloudera AI Registry KubeConfig (save it to ~/.kube/config).
      - `kubectl get cm -n mlx`: lists all the available ConfigMaps.
      - `kubectl describe cm jwks-rootca -n mlx`: Returns the TLS Certificate for Cloudera AI Registry.
      - `cdp ml list-model-registries`: The response will contain the domain for the updated Cloudera AI Registry.
   c. Copy the entire rootca.pem output from the command.

```
kubectl describe cm jwks-rootca -n mlx
```

```
Data
====
rootca.pem:
----
-----BEGIN CERTIFICATE-----
MIIFmDCCA4CgAwIBAgIQU9C87nMpOIFKYpfvOHFHFDANBgkqhkiG9w0BAQsFADBm
MQswCQYDVQQGEwJVUzEzMDEGA1UEChMqKFNUQUdJTkcpIEludGVybmV0IFNlY3Vy
aXR5IFJlc2VhcmNoIEdyb3VwMSIwIAYDVQQDExkoU1RBR0lORykgUHJldGVuZCBQ
ZWFyIFgxMB4XDTE1MDYwNDExMDQzOFoXDTM1MDYwNDExMDQzOFowZjELMAkGA1UE
BhMCVVMxMzAxBgNVBAoTKihTVEFHSU5HKSBJbnRlcm5ldCBTZWN1cml0eSBSZXNl
YXJjaCBHcm91cDEiMCAGA1UEAxMZKFNUQUdJTkcpIFByZXRlbmQgUGVhciBYMTCC
AiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALbagEdDTa1QgGBWSYkyMhsc
ZXENOBaVRTMX1hceJENgsL0Ma49D3MilI4KS38mtkmdF6cPWnL++fgehT0FbRHZg
jOEr8UAN4jH6omjrbTD++VZneTsMVaGamQmDdFl5g1gYaigkkmx8OiCO68a4QXg4
wSyn6iDipKP8utsE+x1E28SA75HOYqpdrk4HGxuULvlr03wZGTIf/oRt2/c+dYmD
oaJhge+GOrLAEQByO7+8+vzOwpNAPEx6LW+crEEZ7eBXih6VP19sTGy3yfqK5tPt
TdXXCOQMKAp+gCj/VByhmIr+0iNDC540gtvV303WpcbwnkkLYC0Ft2cYUyHtkstO
fRcRO+K2cZozoSwVPyB8/J9RpcRK3jgnX9lujfwA/pAbP0J2UPQFxmWFRQnFjaq6
rkqbNEBgLy+kFL1NEsRbvFbKrRi5bYy2lNms2NJPZvdNQbT/2dBZKmJqxHkxCuOQ
FjhJQNeO+Njm1Z1iATS/3rts2yZlqXKsxQUzN6vNbD8KnXRMEeOXUYvbV4lqfCf8
mS14WEbSiMy87GB5S9ucSV1XUrlTG5UGcMSZOBcEUpisRPEmQWUOTWIoDQ5FOia/
GI+Ki523r2ruEmbmG37EBSBXdxIdndqrjy+QVAmCebyDx9eVEGOIpn26bW5LKeru
mJxa/CFBaKi4bRvmdJRLAgMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNVHRMB
Af8EBTADAQH/MB0GA1UdDgQWBBS182Xy/rAKkh/7PH3zRKCsYyXDFDANBgkqhkiG
9w0BAQsFAAOCAgEAncDZNytDbrrVe68UT6py1lfF2h6Tm2p8ro42i87WWyP2LK8Y
nLHC0hvNfWeWmjZQYBQfGC5c7aQRezak+tHLdmrNKHkn5kn+9E9LCjCaEsyIIn2j
qdHlAkepu/C3KnNtVx5tW07e5bvIjJScwkCDbP3akWQixPpRFAsnP+ULx7k0aO1x
qAeaAhQ2rgo1F58hcflgqKTXnpPM02intVfiVVkX5GXpJjK5EoQtLceyGOrkxlM/
sTPq4UrnypmsqSagWV3HcUlYtDinc+nukFk6eR4XkzXBbwKajl0YjztfrCIHOn5Q
CJL6TERVDbM/aAPlv8kJ1sWGLuvvWYzMYqLzDul//rUF10qEMWaXVZV51KpS9DY/
```

The domain of the new Cloudera AI Registry is contained in the list-model-registries response:

```
~ 05:43 pm (1.773s)
cdp ml list-model-registries
2024-04-23 17:43:36,327 - MainThread - cdpcli.clidriver - WARNING - You are running an INTERNAL release of the CDP CLI, which has
 different capabilities from the standard public release. Find the public release at: https://pypi.org/project/cdpcli/
{
    "modelRegistries": [
        {
            "id": 990,
            "creator": "crn:altus:iam:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:user:1f77129b-9cb0-4a23-970f-a5f8af88fdea",
            "status": "installation:finished",
            "environmentCrn": "crn:cdp:environments:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:environment:3715dcbe-48a8-4292
-94a0-f34e44c3bf35",
            "createdAt": "2024-04-23T22:25:58.569000+00:00",
            "crn": "crn:cdp:ml:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:model_registry:dfcff53f-4f0f-4a54-9eaf-4f1cd384387a
",
            "environmentName": "eng-ml-dev-env-aws",
            "workspaceName": "model-registry-ml-17bf05df-050",
            "machineUserCrn": "crn:altus:iam:us-west-1:9d74eee4-1cad-45d7-b645-7ccf9edbb73d:machineUser:cml_env_machine_user_3715
dcbe-48a8-4292-94a0-f34e44c3bf35/5ce78dc4-e84d-498d-a2a0-ebe71d0dae2d",
            "serviceName": "model-registry-ml-17bf05df-050",
            "domain": "https://modelregistry.ml-17bf05df-050.eng-ml-d.xcu2-8y8x.dev.cldr.work"
        },
```

4. Apply ConfigMap Update.
   a. Update the KubeConfig back to the ML Serving KubeConfig.
   b. Edit the ConfigMap of the ML Serving Cluster:
      - kubectl edit cm modelregistry-config-controlplane -n serving: Update tls.crt with the data from above.
      - kubectl describe cm api-config -n serving: Update model.registry.url with the data from above.
   c. Restart the deployment to force the Cloudera Manager changes to take effect.
      - kubectl scale deployment api -n serving --replicas=0
      - kubectl scale deployment api -n serving --replicas=1

Debugging the model import failure

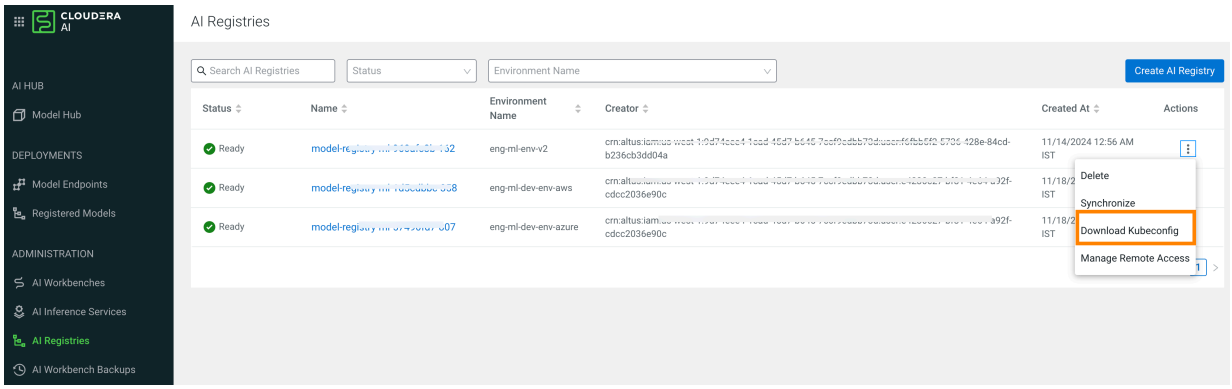
To debug errors that occurred on the Cloudera AI Registry server, you can access the logs found in the API v2 pod.

About this task

Access logs from Cloudera AI Registry Kubernetes cluster.

You can obtain the kubeconfig for the Cloudera AI Registry cluster.

1. In the **Cloudera** console, click the **Cloudera AI** tile.
2. Click AI Registries in the left navigation menu. The AI Registries page displays.
3. In the **Actions** menu, click and select **Download Kubeconfig**.



In AWS, you need to add your identity under Manage Remote Access to access the Kubernetes cluster.

You must add your identity under Manage Remote Access. For information on granting remote access, see *Granting Remote Access to Cloudera AI Workbench*. After the kubeconfig is set up, run the following `kubectl` command to get logs for the Cloudera AI Registry pod:

```
kubectl logs <AI registry pod name> -n mlx
```

Importing a Hugging Face Model (Technical Preview)

If your desired Hugging Face model is unavailable on the Model Hub page, you can import those models from the *Hugging Face* website. After you import the model, the newly imported model will be listed on the Registered Models page.

Note: This feature is in Technical Preview and not recommended for production deployments. Cloudera recommends that you try this feature in test or development environments.

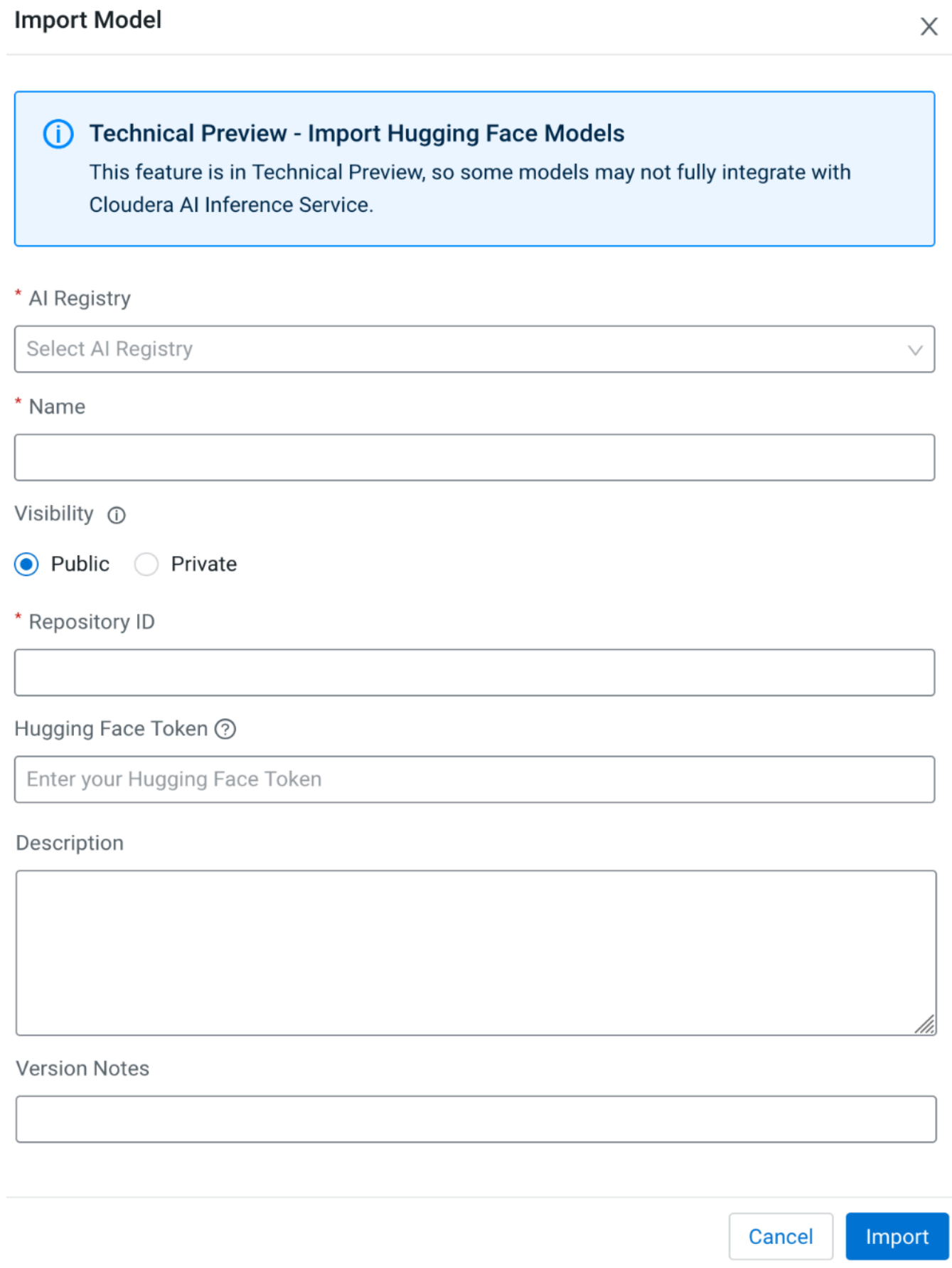
Procedure

1. In the **Cloudera** console, click the **Cloudera AI** tile.

   The **Cloudera AI Workbenches** page displays.
2. Click **Registered Models** under **Deployments** in the left navigation menu.

   The **Registered Models** page displays. The page lists all the models of different Cloudera AI Registries along with the associated metadata.

3. Click Import Model. The Import Model page displays.


Import Model
Technical Preview - Import Hugging Face Models
This feature is in Technical Preview, so some models may not fully integrate with Cloudera AI Inference Service.
* AI Registry
Select AI Registry
* Name
Visibility
Public
Private
* Repository ID
Hugging Face Token
Enter your Hugging Face Token
Description
Version Notes
Cancel
Import

4. In the AI Registry drop-down list, select the AI registry to which you want to import the model.
5. In the Name field, enter a new name for the model you are importing.
6. Select the Visibility as Public or Private. If you select Public, the model is available for other users. If you select Private, the model is displayed on the Registered Models page only for the user who imported it.
7. In the Repository ID field, enter the ID of the Hugging Face model. You can obtain the ID of a model from the Hugging Face website.
8. In the Hugging Face Token field, enter the token obtained from the Hugging Face website.
9. In the Description field, enter a description for the model.
10. In the Version Notes field, enter notes about this version of the model.
11. Click Import.

Results

You can view this newly imported model on the Registered Models page.