

# Managing Cloudera Accelerators for Machine Learning Projects (AMPs)

Date published: 2020-07-16

Date modified: 2025-10-31

# CLOUDERA

# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>Cloudera Accelerators for Machine Learning Projects.....</b>	<b>4</b>
<b>HuggingFace Spaces and Community AMPs.....</b>	<b>5</b>
Enable users to deploy external spaces.....	5
Launch a HuggingFace or Community project.....	5
Launch a HuggingFace Space outside the catalog.....	6
<b>Creating New AMPs using API.....</b>	<b>6</b>
<b>Custom AMP Catalog.....</b>	<b>7</b>
<b>Adding a catalog.....</b>	<b>7</b>
<b>Catalog File Specification.....</b>	<b>8</b>
<b>Host names required by AMPs.....</b>	<b>9</b>
<b>AMPs in airgapped environments.....</b>	<b>10</b>
<b>Configuring Traefik readTimeout for large file uploads.....</b>	<b>11</b>

# Cloudera Accelerators for Machine Learning Projects

Cloudera Accelerators for Machine Learning Projects (AMPs) provide reference example projects in Cloudera AI. More than simplified quickstarts or tutorials, AMPs are fully-developed expert solutions created by Cloudera's research arm, Fast Forward Labs.

These solutions to common problems in this field demonstrate how to fully use the power of Cloudera AI. AMPs show you how to create Cloudera AI projects to solve your own use cases.

AMPs are available to install and run from the Cloudera AI user interface. As new AMPs are developed, they will become available to you for your study and use.



**Note:** In an airgapped installation, the default AMPs catalog included at installation and default AMPs may be inaccessible. See [AMPs in airgapped environments](#).

The screenshot shows the Cloudera AI Workbench interface. On the left is a sidebar with navigation options: Home, Projects, Sessions, Experiments, Models, Jobs, Applications, AMPs (highlighted), Runtime Catalog, Learning Hub, User Settings, and Site Administration. The main area is titled 'Accelerators for ML Projects' and contains a search bar, filters for 'All', 'Cloudera', and 'Hugging Face', and a 'Deploy External' button. Below the filters, there are 10 AMP tiles arranged in a grid. Each tile has a header image, a title, a brief description, and a 'Deploy' button. The AMPs include: Knowledge Graph powered RAG based QA application, Fine Tuning Studio, PromptBrew by Verta, Chat with your Documents, Pinecone Intelligent QA Chatbot, Amazon Bedrock Text Summarization, Fine-Tuning a Foundation Model for Multiple Tasks, LLM Chatbot Augmented with Enterprise Data, Churn Modeling with scikit-learn, and Deep Learning for Image Analysis.

## Using AMPs

Get started with AMPs:

1. Log in to your Cloudera AI workbench, and in the left panel click AMPs.
2. Click on an AMP tile to read its description.
3. Click Configure Project and provide any configuration values required by the AMP. The Description field explains how to determine these configuration values. After you click Launch Project, the installation process may take several minutes.
4. When the installation is complete, click Overview to read the documentation for the AMP and explore the code and project structure.



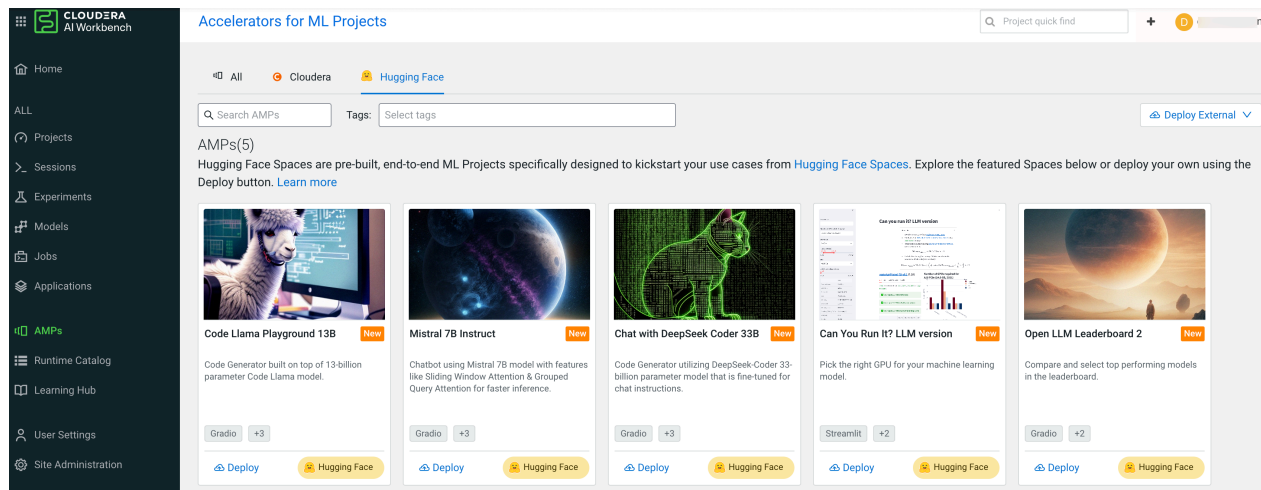
**Note:** If nothing appears in the AMPs panel, an administrator may need to reconfigure and refresh the catalog. In Site Administration AMPs, click Refresh. The administrator can also refresh periodically to add newly developed AMPs to the panel.

## Related Information

[Custom AMP Catalog](#)

## HuggingFace Spaces and Community AMPs

You can run HuggingFace Spaces or Community AMPs, which are both types of machine learning projects, in your Cloudera AI Workbench. In the HuggingFace tab in AMPs are several HF spaces that are tested and supported on Cloudera AI.



HuggingFace is an online community for machine learning ([Hugging Face](https://huggingface.co)). It serves as a repository for machine learning projects and datasets. Spaces are individual projects or applications. Community AMPs are machine learning prototypes created by the Cloudera AI community.

There are a few things to keep in mind when launching HuggingFace Spaces or Community AMPs in your Cloudera AI Workbench.

- To launch a HuggingFace project, you first need to create a free account on HuggingFace and obtain an access token.
- Make sure that the Hugging Face project actually works. Many of the projects are community-created projects that may be experimental or no longer maintained.
- The HuggingFace Space must use the gradio or streamlit libraries to run on Cloudera AI.
- The Community tab is not visible by default. To enable this tab, in Site Administration Settings, select: Allow users to deploy community AMPs.

### Enable users to deploy external spaces

By default, users cannot deploy Hugging Face spaces outside the curated collection in the Hugging Face tab.

To enable users to deploy external Hugging Face spaces, in Site Administration Settings, select:

- Allow users to deploy external Hugging Face Space

When selected, the ability to deploy external HuggingFace spaces is available.

### Launch a HuggingFace or Community project

You can launch a HuggingFace Space or community project by clicking Deploy for the respective space.

In Configure Project, you can set some parameters for the space.

If needed, enter your Hugging Face access token. The environment variables HF\_TOKEN and HUGGING\_FACE\_HUB\_TOKEN both require the same value.



**Note:** Do not reduce any of the values specified for cpu, memory, or gpu.

Click Launch Project to deploy the project in your workbench. Cloudera AI will run scripts to download the project from the HuggingFace space and install it in Cloudera AI. This process can take some time.

## Launch a HuggingFace Space outside the catalog

It is also possible to deploy Hugging Face spaces that are not included in the AMP catalog. Note that there are several caveats on which HuggingFace spaces may be able to work in Cloudera AI, as described above.

Click Deploy External to open the UI for launching an external project, and follow the instructions. The project is imported from HuggingFace or the specified repository

## Creating New AMPs using API

You can create AMPs using API V2.

1. Get the valid runtime\_identifier by using the GET request using filter parameters like kernel, editor, and edition.

For example, if you want to create the AMP and it requires the following runtime based on project-metadata.yaml of the AMP, then use the filter like {"kernel": "Python 3.9", "editor": "Workbench", "edition": "Standard"} to get a smaller list of runtimes:

```
curl -X GET "https://<workbenchURL>/api/v2/runtimes?search_filter=%7B%22kernel%22%3A%22Python%203.9%22%2C%22editor%22%3A%22Workbench%22%2C%22edition%22%3A%22Standard%22%7D" -H "accept: application/json" -H "Authorization: Bearer <API Token>"
```

This command will display a list of runtimes that match the filter.

2. Using the runtime\_identifier obtained in the previous step, you can create an AMP. Following is the sample payload used to trigger the creation of one of the AMPs. You can add additional key-value pairs as needed under the environment object based on the project-metadata.yaml file of the AMP you are trying to create.

```
{
  "configure_prototype_request": {
    "execute_amp_steps": true,
    "run_import_tasks": true,
    "runtime_identifier": "docker.repository.cloudera.com/cloudera/cds/ml-runtime-workbench-python3.9-standard:2024.05.2-b14"
  },
  "create_project_request": {
    "description": "amps v2",
    "environment": {
      "HF_TOKEN": "<HF TOKEN>"
    },
    "git_url": "https://github.com/cloudera/CML_AMP_Image_Analysis",
    "name": "amps_v2",
    "template": "git",
    "visibility": "private"
  }
}
```

Example: The following is an example of creating an AMP that requires multiple environment variables with IBM WatsonXAI.

```
{
  "configure_prototype_request": {
```

```

    "execute_amp_steps": true,
    "run_import_tasks": true,
    "runtime_identifier": "docker.repository.cloudera.com/cloudera/cdsw/ml
-runtime-pbj-workbench-python3.9-standard:2024.10.1-b12"
  },
  "create_project_request": {
    "description": "watson amps v2",
    "environment": {
      "IBM_WATSONXAI_ENDPOINT": "https://us-south.ml.cloud.ibm.com",
      "IBM_API_KEY": "<Put valid key>",
      "IBM_PROJECT_ID": "<Put valid project ID>"
    },
    "git_url": "https://github.com/agupta-git/CML_AMP_watsonxai",
    "name": "watson_v2",
    "template": "git",
    "visibility": "private"
  }
}

```

3. You can now see that a new project has been created in your Cloudera AI Workbench which is carrying out the tasks listed in its project-metadata.yaml file. After all the tasks are executed, your new AMP-project will be ready to use.

## Custom AMP Catalog

An AMP catalog is a collection of AMPs that can be added to a workbench as a group. Cloudera AI ships with the Cloudera AMP catalog, containing AMPs developed by Cloudera Fast Forward Labs, but you can also create and add your own catalog, containing AMPs developed by your organization.

To create an AMP catalog, you need to create a YAML file called the catalog file. This file is hosted on GitHub or another git server. This could be either a public or a private git server.

The catalog file contains information about each AMP in the catalog, and provides a link to the AMP repository itself. The catalog file specification is shown in Catalog File Specification.

You can also look at the Cloudera catalog file for an example. To view the file, click directly on the URL for Cloudera in Catalog Sources.

For more details on creating the AMPs that you will include in your catalog, see Creating New AMPs.

One use case you might consider is creating a fork of the Cloudera AMP catalog, in order to host it internally. In this case, you will need to edit the URLs in the catalog and project metadata files to point to the appropriate internal resources.

### Related Information

[Creating New AMPs](#)

## Adding a catalog

The collection of AMPs available to end users can draw from one or more sources. For example, you might have an internal company catalog in addition to the default Cloudera catalog. The catalog must be provided with a catalog file and one or more project metadata YAML files.

### About this task

Specify Catalog File URL if your git hosting service allows you to access the raw content of the repo without authenticating. (That is, the source files can be retrieved with a curl command, and do not require logging into a web

page). Otherwise, specify the Git Repository URL. To use a git repository as a catalog source, the catalog file and the AMP files must be in a repository that can be cloned with `git clone` without authentication.

### Procedure

1. As an Administrator, go to Site Administration > AMPs .
2. Select Git Repository URL or Catalog File URL to specify a new source. Paste or enter the URL to the new source, and file name for the catalog file if necessary.
3. Click Add Source.  
The catalog YAML file is loaded, and the projects found there are displayed in Catalog Entries.
4. If there are projects that are not yet ready for use, or that should not be displayed in the catalog, deselect Enabled in the Catalog Entries.

## Catalog File Specification

The Catalog file is a YAML file that contains descriptive information and metadata for the displaying the AMP projects in the Project Catalog.

### Fields

Fields are in snake\_case. Each project in the catalog uses the following fields:

Field Name	Type	Example	Description
name	string	name: Cloudera	Required. Name of the catalog, displayed as Source in the Prototype Catalog tab.
entries	string	entries:	Required. Contains the entries for each project.
title	string	title: Churn Modeling	Required. The title of the AMP, as displayed in the Prototype Catalog.
label	string	label: churn-prediction	Required.
short_description	string	short_description: Build an scikit-learn model...	Required. A short description of the project. Appears on the project tile in the Prototype Catalog.
long_description	string	long_description: >- This project demonstrates...	Required. A longer description that appears when the user clicks on the project tile.
image_path	string	image_path: >- https://raw.githubusercontent.com	Required. Path to the image file that displays in the Prototype Catalog.
tags	string	tags: - Churn Prediction - Logistic Regression	Required. For sorting in the Prototype Catalog pane.
git_url	string	git_url: "https:..."	Required. Path to the git repository for the project.

Field Name	Type	Example	Description
git_ref	string	git_ref: 9e56b6578e37185777380e2447413e83d9016a3	Optional. Git ref (branch name or commit hash/tag name).
is_prototype	boolean	is_prototype: true	Optional. Indicates the AMP should be displayed in the Prototype Catalog. Use if coming_soon is not used.
coming_soon	boolean	coming_soon: true	Optional. Displays the AMP in the Prototype Catalog with a “COMING SOON” watermark. Use if is_prototype is not used.

Example:

```
name: Cloudera

entries:
- title: Churn Modeling with scikit-learn
  label: churn-prediction
  short_description: Build an scikit-learn model to predict churn using
customer telco data.
  long_description: >-
    This project demonstrates how to build a logistic regression classific
ation model to predict the probability
    that a group of customers will churn from a fictitious telecommunicatio
ns company. In addition, the model is
    interpreted using a technique called Local Interpretable Model-agnos
tic Explanations (LIME). Both the logistic
    regression and LIME models are deployed using Cloudera AI's real-time
model deployment capability and interact with a
    basic Flask-based web application.
  image_path: >-
    https://raw.githubusercontent.com/cloudera/Applied-ML-Prototypes/mas
ter/images/churn-prediction.jpg
  tags:
    - Churn Prediction
    - Logistic Regression
    - Explainability
    - Lime
  git_url: "https://github.com/cloudera/CML_AMP_Churn_Prediction"
  is_prototype: true
```

## Host names required by AMPs

If you are using a non-transparent proxy in AWS, you need to allow the following host names in order for AMPs to work.

- \*.storage.googleapis.com
- \*.raw.githubusercontent.com
- \*.pypi.org
- \*.pythonhosted.org

- \*.pypi.python.org
- github.com
- \*.github.com
- huggingface.co
- \*.huggingface.co
- \*.pinecone.io
- download.pytorch.org
- objects.githubusercontent.com

These host names should be specified in the proxy, along with any other endpoints that are needed for your workspace.

## AMPs in airgapped environments

In an airgapped installation, the default Cloudera Accelerators for Machine Learning Projects catalog included at installation and default AMPs may be inaccessible. There are a few options to work around this issue.



**Important:** Cloudera Accelerators for Machine Learning Projects do not work in Cloudera AI on premises version 1.5.4 environment where non-transparent proxy is configured. To use AMPs in airgapped environment with non-transparent proxy configured, upgrade to Cloudera AI on premises version 1.5.4 CHF1 (1.5.4-h2-b25) or higher version. For more details see the [TSB 2024-766: AMPs and Feature Announcements not working with non-transparent proxy on Cloudera AI on premises 1.5.x](#) Knowledge Base article.

### Option 1: Set up a proxy with AMPs public endpoints whitelisted

Set up a proxy and whitelist the public endpoints listed in [Host names required by AMPs](#).

### Option 2: Clone AMPs catalog and projects internally and configure the Cloudera AI Workbench to use their internal catalog

1. Clone the AMPs catalog repository to local: [AMP Catalog | Github documentation](#).



**Note:** Catalog links must be https, not http.

2. Clone all the git projects mentioned in the git\_url fields in [AMPs | Github documentation](#).
3. Update the git\_url and image\_path links to point to your local internal github repository clones (See [Step 2](#)) in local amp-catalog-cloudera-default.yaml (See [Step 1](#)).
4. Update the AMPs catalog in the Cloudera AI Workbench to point to the local amp-catalog-cloudera-default.yaml (See [Step 1](#)). Follow the steps in [Adding a catalog](#) [Adding a catalog](#) on page 7.
5. Complete the steps as instructed in [Accessing python packages required by AMPs on airgapped setups](#).

### Option 3: Download the desired AMPs in a zip file and upload it as a Cloudera AI project

1. Browse the AMPs catalog page and go to the github project associated with the AMP: [Cloudera Accelerators for Machine Learning Projects | Cloudera Github documentation](#).



**Note:** Catalog links must be https, not http.

2. Download the AMP project as zip file.
3. Go to Cloudera AI Workbench and create a project by choosing file upload method and providing the above AMP zip file.
4. Complete the steps as instructed in [Accessing python packages required by AMPs on airgapped setups](#).

### Accessing python packages required by AMPs on airgapped setups

You must whitelist or allow traffic to the public pypi repository to install python packages required for AMPs to successfully deploy.

- \*.pypi.org
- \*.pythonhosted.org

If you do not want to allow access to these domains from your environment, you must host your internal pypi repository and upload all the necessary pip packages used by AMPs and configure your cluster to use the internal pypi as default.

## Configuring Traefik readTimeout for large file uploads

When uploading large files, the default readTimeout of 60 seconds in Traefik might not suffice. In such cases, you can configure Traefik to disable the default timeout.

1. Run the following Kubernetes command to edit the Traefik ConfigMap:

```
kubectl edit configmap traefik-config -n mlx
```

2. In the ConfigMap file, locate and modify the traefik.toml section to include the following custom readTimeout:

```
[entryPoints.websecure.transport]
  [entryPoints.websecure.transport.respondingTimeouts]
    readTimeout = 0
```



**Note:** The readTimeout value is set to 0 to indicate no timeout, ensuring uninterrupted uploads for large files.

For example, after adding the above changes, your traefik.toml file should look similar to:

```
# traefik.toml
[global]
  checkNewVersion = false
  sendAnonymousUsage = false
[entryPoints.web]
  address = ":80"
  [entryPoints.web.http.redirects.entryPoint]
    to = "websecure"
    scheme = "https"
[entryPoints.websecure]
  address = ":443"
  [entryPoints.websecure.transport]
    [entryPoints.websecure.transport.respondingTimeouts]
      readTimeout = 0
  [entryPoints.websecure.http.tls]
```

3. Save and apply the updated ConfigMap to your cluster for the changes to take effect.

```
kubectl rollout restart deployment/traefik -n mlx
```