Machine Learning

Applied ML Prototypes (AMPs)

Date published: 2020-07-16
Date modified: 2024-05-30

https://docs.cloudera.com/

Legal Notice

© Cloudera Inc. 2024. 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

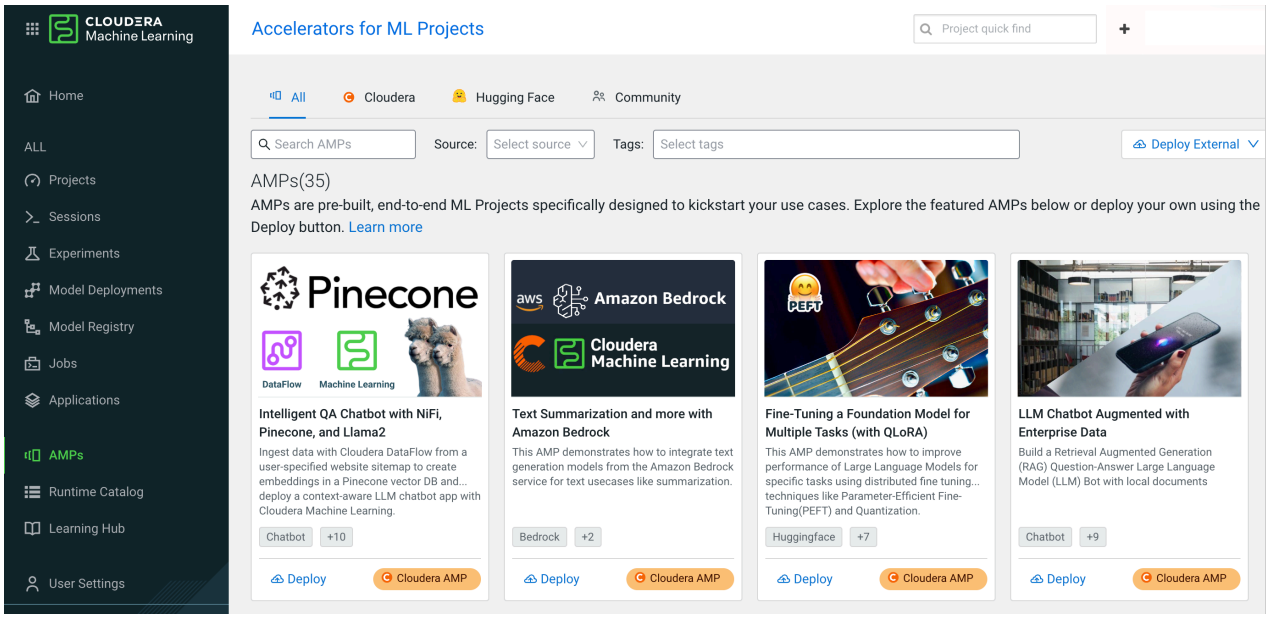
Accelerators for ML Projects (AMPs)

Accelerators for ML Projects (AMPs) provide reference example machine learning projects in Cloudera Machine Learning. 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 the machine learning field demonstrate how to fully use the power of Cloudera Machine Learning. AMPs show you how to create Cloudera Machine Learning projects to solve your own use cases.

AMPs are available to install and run from the Cloudera Machine Learning 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 on page 25



Using AMPs

It's simple to get started with AMPs.

1. Log in to your Cloudera Machine Learning workspace, 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.

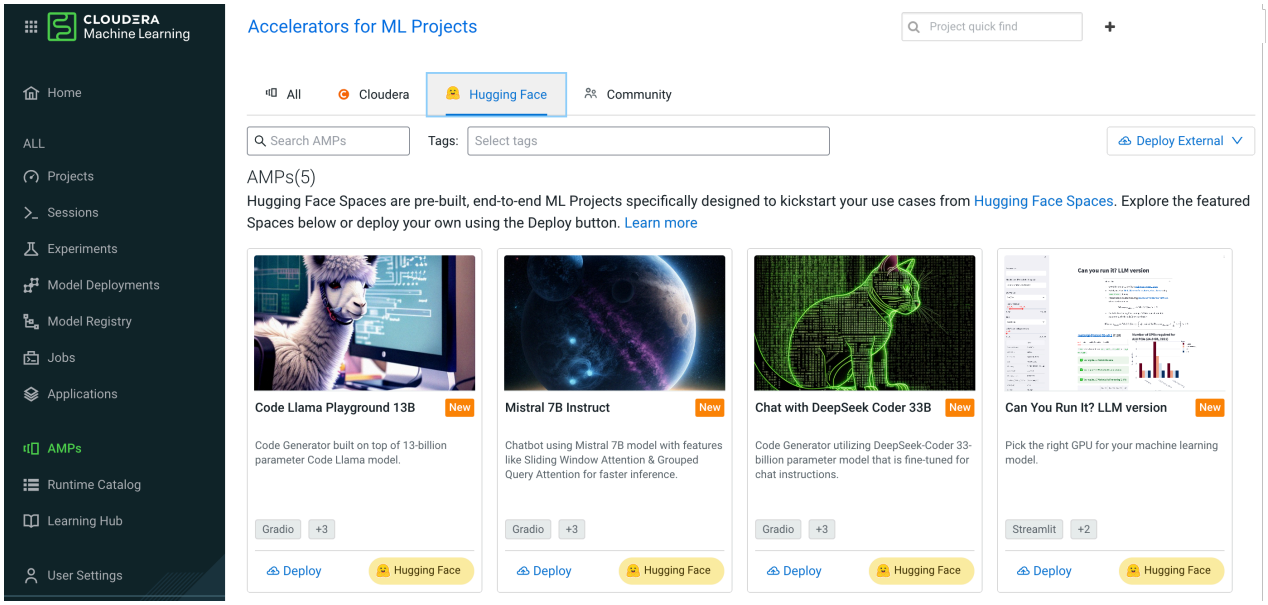
Note: If nothing appears in the AMPs panel, an administrator may need to reconfigure and refresh the catalog. In Site Administration AMPs , click Refresh. The administrator can also refresh periodically to add newly developed AMPs to the panel.

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 CML workspace. In the HuggingFace tab in AMPs are several HF spaces that are tested and supported on CML.



HuggingFace is an online community for machine learning (Hugging Face). 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 CML community.

There are a few things to keep in mind when launching HuggingFace Spaces or Community AMPs in your CML workspace.

- 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 CML.
- 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 workspace. CML will run scripts to download the project from the HuggingFace space and install it in CML. 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 CML, 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

One great use for AMPs is to showcase reference examples specific to your business by creating your own AMPs in-house. Once a data science project has been built in Cloudera Machine Learning, you can package it and have the Cloudera Machine Learning Admin add it to the AMP Catalog.

Each individual AMP requires a project metadata file, which defines the environmental resources needed by the AMP, and the setup steps to install the AMP in a Cloudera Machine Learning workspace. See AMP Project Specification for details.

Note: You can store your AMPs in a git repo hosted on Github, Github Enterprise, or GitLab servers (not limited to github.com or gitlab.com.)

Additionally, only simple authentication is supported, such as passing an API key, or including the username and password, as part of the URL. If additional authentication steps are required, then that git host is not supported.

You can also look at an example for a Cloudera AMP, such as: .project-metadata.yaml.

Related Information

AMP Project Specification

Custom AMP Catalog

An AMP catalog is a collection of AMPs that can be added to a workspace as a group. Cloudera Machine Learning 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

Catalog File Specification
Creating New AMPs

Add 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.

Field Name	Type	Example	Description
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 .git...	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.
git_ref	string	git_ref: 9e56b6578e37185777380e2a74b3/83d9ac6ac3	Optional. Git ref (branch name, 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 CML's real-time model dep
loyment capability and interact with a
     basic Flask-based web application.
     image_path: >-
```

```
      https://raw.githubusercontent.com/cloudera/Applied-ML-Prototypes/maste
r/images/churn-prediction.jpg
    tags:
      - Churn Prediction
      - Logistic Regression
      - Explainability
      - Lime
    git_url: "https://github.com/cloudera/CML_AMP_Churn_Prediction"
    is_prototype: true
```

AMP Project Specification

AMP projects include a project metadata file that provides configuration and setup details. These details may include environment variables and tasks to be run on startup.

YAML File Specification # Version 1.0

The project metadata file is a YAML file. It must be placed in your project's root directory, and must be named .project-metadata.yaml. The specifications for this file are listed below. You can also look at an example for one of the Cloudera AMPs, such as:.project-metadata.yaml.

Fields

Fields for this YAML file are in snake_case. String fields are generally constrained by a fixed character size, for example string(64) is constrained to contain at most 64 characters. Click Show to see the list of fields.

Field Name	Type	Example	Description
name	string(200)	ML Demo	Required: The name of this project prototype. Prototype names do not need to be unique.
description	string(2048)	This demo shows off some cool applications of ML.	Required: A description for this project prototype.
author	string(64)	Cloudera Engineer	Required: The author of this prototype (can be the name of an individual, team, or organization).
date	date string	"2020-08-11"	The date this project prototype was last modified. It should be in the format: "YYYY-MM-DD" (quotation marks are required).
specification_version	string(16)	0.1	Required: The version of the YAML file specification to use.
prototype_version	string(16)	1.0	Required: The version of this project prototype.

Field Name	Type	Example	Description
shared_memory_limit	number	0.0625	Additional shared memory in GB available to sessions running in this project. The default is 0.0625 GB (64MB).
environment_variables	environment variables object	See below	Global environment variables for this project prototype.
feature_dependencies	feature_dependencies	See below	A list of feature dependencies of this AMP. A missing dependency in workspace blocks the creation of the AMP.
engine_images	engine_images	See below	Engine images to be used with the AMP. What's specified here is a recommendation and it does not prevent the user from launching an AMP with non recommended engine images.
runtimes	runtimes	See below	Runtimes to be used with the AMP. What's specified here is a recommendation and it does not prevent the user from launching an AMP with non recommended runtimes.
tasks	task list	See below	A sequence of tasks, such as running Jobs or deploying Models, to be run after project import.

Example

```
name: ML Demo
description: >-
This demo shows off some cool applications of ML.
author: Cloudera Engineer
date: '2020-08-11T17:40:00.839Z'
specification_version: 1.0
environment_variables:
...
tasks:
...
```

Environment variables object

The YAML file can optionally define any number of global environment variables for the project under the environment field. This field is an object, containing keys representing the names of the environment variables, and

values representing details about those environment variables. Click Show to see the list of fields in the Environment variables object.

Field Name	Type	Example	Description
default	string	"3"	The default value for this environment variable. Users may override this value when importing this project prototype.
description	string	The number of Model replicas, 3 is standard for redundancy.	A short description explaining this environment variable.
required	boolean	true	Whether the environment variable is required to have a non-empty value, the default is false.

Example: This example creates four environment variables.

```
environment_variables:
  AWS_ACCESS_KEY:
    default: ""
    description: "Access Key ID for accessing S3 bucket"
  AWS_SECRET_KEY:
    default: ""
    description: "Secret Access Key for accessing S3 bucket"
    required: true
  HADOOP_DATA_SOURCE:
    default: ""
    description: "S3 URL to large data set"
    required: false
  MODEL_REPLICAS:
    default: "3"
    description: "Number of model replicas, 3 is standard for redundancy"
    required: true
```

Feature Dependencies

AMPs might depend on some optional features of a workspace. The feature_dependencies field accepts a list of such features. Unsatisified feature dependencies prevent the AMP from being launched in a workspace, and display an appropriate error message. The supported feature dependencies are as follows:

- model_metrics

Runtimes Specification

The runtimes field accepts a list of runtimes objects defined as follows. This Runtimes specification can be added per task or per project.

```
- editor: the_name_of_the_editor # case-sensitive string required. e.g. Work
bench, Jupyter, etc. (how it appears in the UI)
  kernel: the_kernel # case-sensitive string required. e.g. Python 3.6, Pyth
on 3.8, R 3.6, etc. (how it appears in the UI)
  edition: the_edition # case-sensitive string required. e.g. Standard, N
vidia GPU, etc. (how it appears in the UI)
```

```
  version: the_short_version # case-sensitive string optional. e.g. 2021.
03, 2021.05, etc. (how it appears in the UI)
  addons: the_list_addons_needed # list of case-sensitive strings optional.
 e.g Spark 2.4.7 - CDP 7.2.11 - CDE 1.13, etc. (how it appears in the UI)
```

This example specifies the Runtimes the Workbench version for Python 3.8.

```
runtimes:
  - editor: Workbench
    kernel: Python 3.8
    edition: Standard
    addons: ['Spark 2.4.7 - CDP 7.2.11 - CDE 1.13']
```

Engine Images Specification

The engine_images field accepts a list of engine_image objects defined as follows:

```
- image_name: the_name_of_the_engine_image # string (required)
  tags: # list of strings (optional)
    - the_tag_of_engine_image
    - ...
```

This example specifies the official engine image with version 11 or 12:

```
engine_images:
  - image_name: engine
    tags:
      - 12
      - 11
```

This example specifies the most recent version of the dataviz engine image in the workspace:

```
engine_images:
  - image_name: cmldataviz
  - image_name: cdswdataviz
```

Note that when specifying CDV images, both cmldataviz and cdswdataviz must be specified. When tags are not specified, the most recent version of the engine image with the matching name is recommended. The following rule is used to determine the most recent engine_image with the matching name:

Official Engine (engine) and CDV (cmldataviz and cdswdataviz) images

Since the officially released engine images follow semantic versioning (where a newer version is always larger than any older version, when compared with >), the most recent engine image is the one with the largest tag. For example, engine:14 will be recommended over engine:13 and cmldataviz:6.3.4-b13 is recommended over cmldataviz:6.2.1-b12.

Custom engine images

There is no way for Cloudera Machine Learning to determine the rules for customer custom engine image tags, and therefore there is no reliable way to determine the most recent custom engine image. You should use the engine image that has the correct matching name and has the newest id. The newest id means that the engine image is the most recently added engine image.

Task list

This defines a list of tasks that can be automatically run on project import. Each task will be run sequentially in the order they are specified in this YAML file. Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	create_job	Required: The type of task to be executed. See below for a list of allowed types.
short_summary	string	Creating a Job that will do a task.	A short summary of what this task is doing.
long_summary	string	Creating a Job that will do this specific task. This is important because it leads up to this next task.	A long summary of what this task is doing.

Jobs

Create Job

Example

```
- type: create_job
  name: howdy
  entity_label: howdy
  script: greeting.py
  arguments: Ofek 21
  short_summary: Creating a job that will greet you.
  environment_variables:
  SAMPLE_ENVIRONMENT_VARIABLE: CREATE/RUN_JOB
  kernel: python3
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	create_job	Required: Must be create_job.
name	string	howdy	Required: Job name.
entity_label	string	howdy	Required: Uniquely identifies this job for future tasks, i.e. run_job tasks. Entity labels must be lowercase alphanumeric, and may contain hyphens or underscores.
script	string	greeting.py	Required: Script for this Job to run.
kernel	string	python3	Required: What kernel this Job should use. Acceptable values are python2, python3, r, and scala. Note

Field Name	Type	Example	Description
			that scala might not be supported for every cluster.
arguments	string	Ofek 21	Command line arguments to be given to this Job when running.
environment_variables	environment variables object	See above	See above
cpu	number	1.0	The amount of CPU virtual cores to allocate for this Job, the default is 1.0.
memory	number	1.0	The amount of memory in GB to allocate for this Job, the default is 1.0.
gpu	integer	0	The amount of GPU to allocate for this Job, the default is 0.
timeout	integer	10	The amount of time in minutes to wait before timing out this Job, the default is 10.
timeout_kil	boolean	false	Whether or not to stop this Job when it times out, the default is false.

Run Job

Example run job task:

```
- type: run_job
  entity_label: howdy
  short_summary: Running the job that will greet you.
  long_summary: >-
    Running the job that will greet you. It will greet you by the name
    which is the first and only command line argument.
```

Most Job run tasks should just contain the type and entity_label fields. Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	run_job	Required: Must be run_job.
entity_label	string	howdy	Required: Must match an entity_label of a previous create_job task.

However, they can optionally override previously defined fields. Click Show to see the list of fields.

Field Name	Type	Example	Description
script	string	greeting.py	Required: Script for this Job to run.
kernel	string	python3	Required: What kernel this Job should use. Acceptable values are python2, pyth on3, r, and scala. Note that scala might not be supported for every cluster.
arguments	string	Ofek 21	Command line arguments to be given to this Job when running.
environment_variables	environment variables object	See above	See above
cpu	number	1.0	The amount of CPU virtual cores to allocate for this Job, the default is 1.0
memory	number	1.0	The amount of memory in GB to allocate for this Job, the default is 1.0.
gpu	integer	0	The amount of GPU to allocate for this Job, the default is 0.
shared_memory_limit	number	0.0625	Limits the additional shared memory in GB that can be used by this Job, the default is 0.0625 GB (64MB).

Models

Note: All models have authentication disabled, so their access key alone is enough to interact with them.

Resources object

Models may define a resources object which overrides the amount of resources to allocate per Model deployment.

Click Show to see the list of fields.

Field Name	Type	Example	Description
cpu	number	1.0	The number of CPU virtual cores to allocate per Model deployment.

Field Name	Type	Example	Description
memory	number	2.0	The amount of memory in GB to allocate per Model deployment.
gpu	integer	0	The amount of GPU to allocate per Model deployment.

For example:

```
resources:
  cpu: 1
  memory: 2
```

Replication policy object

Models may define a replication policy object which overrides the default replication policy for Model deployments.

Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	fixed	Must be fixed if present.
num_replicas	integer	1	The number of replicas to create per Model deployment.

For example:

```
replication_policy:
  type: fixed
  num_replicas: 1
```

Model examples list

Models may include examples, which is a list of objects containing a request and response field, each containing a valid object inside, as shown in the example:

```
examples:
  - request:
      name: Ofek
      age: 21
    response:
      greeting: Hello Ofek (21)
  - request:
      name: Jimothy
      age: 43
    response:
      greeting: Hello Coy (43)
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
request	string	See above	Required: An example request object.
response	string	See above	Required: The response to the above example request object.

Create Model

Example:

```
- type: create_model
  name: Say hello to me
  entity_label: says-hello
  description: This model says hello to you
  short_summary: Deploying a sample model that you can use to greet you
  access_key_environment_variable: SHTM_ACCESS_KEY
  default_resources:
  cpu: 1
  memory: 2
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	create_model	Required: Must be crea te_model.
name	string	Say hello to me	Required: Model name
entity_label	string	says-hello	Required: Uniquely identifies this model for future tasks, i.e. build_mo del and deploy_model tasks. Entity labels must be lowercase alphanumeric, and may contain hyphens or underscores.
access_key_environment_variable	string	SHTM_ACCESS_KEY	Saves the model's access key to an environment variable with the specified name.
default_resources	resources object	See above	The default amount of resources to allocate per Model deployment.
default_replication_policy	replication policy object	See above	The default replication policy for Model deployments.

Field Name	Type	Example	Description
description	string	This model says hello to you	Model description.
visibility	string	private	The default visibility for this Model.

Build Model

Example

```
- type: build_model
  entity_label: says-hello
  comment: Some comment about the model
  examples:
    - request:
        name: Ofek
        age: 21
    response:
      greeting: Hello Ofek (21)
  target_file_path: greeting.py
  target_function_name: greet_me
  kernel: python3
  environment_variables:
    SAMPLE_ENVIRONMENT_VARIABLE: CREATE/BUILD/DEPLOY_MODEL
```

Field Name	Type	Example	Description
type	string	build_model	Required: Must be buil d_model.
entity_label	string	says-hello	Required: Must match an entity_label of a previous create_model task.
target_file_path	string	greeting.py	Required: Path to file that will be run by Model.
target_function_name	string	greet_me	Required: Name of function to be called by Model.
kernel	string	python3	What kernel this Model should use. Acceptable values are python2, pyth on3, r, and scala. Note that scala might not be supported for every cluster.
comment	string	Some comment about the model	A comment about the Model.
examples	model examples list	See above	A list of request/response example objects.

Field Name	Type	Example	Description
environment_variables	environment variables object	See above	See above

Deploy Model

Example:

```
- type: deploy_model
  entity_label: says-hello
  environment_variables:
  SAMPLE_ENVIRONMENT_VARIABLE: CREATE/BUILD/DEPLOY_MODEL
```

Most deploy model tasks should just contain the type and entity_label fields. Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	deploy_model	Required: Must be depl oy_model.
entity_label	string	says-hello	Required: Must match an entity_label of a previous deploy_model task.

However, they can optionally override previously defined fields. Click Show to see the list of fields.

Field Name	Type	Example	Description
cpu	number	1.0	The number of CPU virutal cores to allocate for this Model deployment.
memory	number	2.0	The amount of memory in GB to allocate for this Model deployment.
gpu	integer	0	The amount of GPU to allocate for this Model deployment.
replication_policy	replication policy object	See above	The replication policy for this Model deployment.
environment_variables	environment variables object	See above	Overrides environment variables for this Model deployment.

Applications

Start Application

Example:

```
- type: start_application
```

```
  subdomain: greet
  script: greeting.py
  environment_variables:
    SAMPLE_ENVIRONMENT_VARIABLE: START_APPLICATION
  kernel: python3
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
bypass_authentication	boolean	True	When enabled, allows unauthenticated access to an application
type	string	start_application	Required: Must be start_application.
subdomain	string	greet	Required: Application subdomain, which must be unique per Application, and must be alphanumeric and hyphen-delimited. Application subdomains are also converted to lowercase.
kernel	string	python3	Required: What kernel this Application should use. Acceptable values are python2, python3, r, and scala. Note that scala might not be supported for every cluster.
entity_label	string	greeter	Uniquely identifies this application for future tasks. Entity labels must be lowercase alphanumeric, and may contain hyphens or underscores.
script	string	greeting.py	Script for this Application to run.
name	string	Greeter	Application name, defaults to 'Untitled application'.
description	string	Some description about the Application	Application description, defaults to 'No description for the app'.
cpu	number	1.0	The number of CPU virutal cores to allocate for this Application.

Field Name	Type	Example	Description
memory	number	1.0	The amount of memory in GB to allocate for this Application.
gpu	integer	0	The amount of GPU to allocate for this Application.
shared_memory_limit	number	0.0625	Limits the additional shared memory in GB that can be used by this application, the default is 0.0625 GB (64MB).
environment_variables	environment variables object	See above	See above
static_subdomain	boolean	True	When enabled, subdomain will not get randomized.

Experiments

Run Experiment

Example:

```
- type: run_experiment
  script: greeting.py
  arguments: Ofek 21
  kernel: python3
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	run_experiment	Required: Must be run_experiment.
script	string	greeting.py	Required: Script for this Experiment to run.
entity_label	string	test-greeter	Uniquely identifies this experiment for future tasks. Entity labels must be lowercase alphanumeric, and may contain hyphens or underscores.
arguments	string	Ofek 21	Command line arguments to be given to this Experiment when running.
kernel	string	python3	What kernel this Experiment should use.

Field Name	Type	Example	Description
			Acceptable values are python2, python3, r, and scala. Note that scala might not be supported for every cluster.
comment	string	Comment about the experiment	A comment about the Experiment.
cpu	number	1.0	The amount of CPU virtual cores to allocate for this Experiment.
memory	number	1.0	The amount of memory in GB to allocate for this Experiment.
gpu	number	0	The amount of GPU to allocate for this Experiment.

Sessions

Run Sessions

Example:

```
- type: run_session
  name: How to be greeted interactively
  code: |
    import os
    os.environ['SAMPLE_ENVIRONMENT_VARIABLE'] = 'SESSION'

    !python3 greeting.py Ofek 21

    import greeting
    greeting.greet_me({'name': 'Ofek', 'age': 21})
  kernel: python3
  memory: 1
  cpu: 1
  gpu: 0
```

Click Show to see the list of fields.

Field Name	Type	Example	Description
type	string	run_session	Required: Must be run_session.
	string	See above for code, greeting.py for script	Required: Either the code or script field is required to exist for the run Session task, not both. code is a direct block of code that will be run by the Session,

Field Name	Type	Example	Description
			while script is a script file that will be executed by the Session.
kernel	string	python3	Required: What kernel this Session should use. Acceptable values are python2, python3, r, and scala. Note that scala might not be supported for every cluster.
cpu	number	1.0	Required: The amount of CPU virtual cores to allocate for this Session.
memory	number	1.0	Required: The amount of memory in GB to allocate for this Session.
entity_label	string	greeter	Uniquely identifies this session for future tasks. Entity labels must be lowercase alphanumeric, and may contain hyphens or underscores.
name	string	How to be greeted interactively	Session name.
gpu	integer	0	The amount of GPU to allocate for this Session.

Restarting a failed AMP setup

If any task fails after initiating an AMP, you can resume it from the last unsuccessful step using the Resume option. You can use the Redeploy option to re-import and execute tasks based on the modified .project-metadata.yaml file.

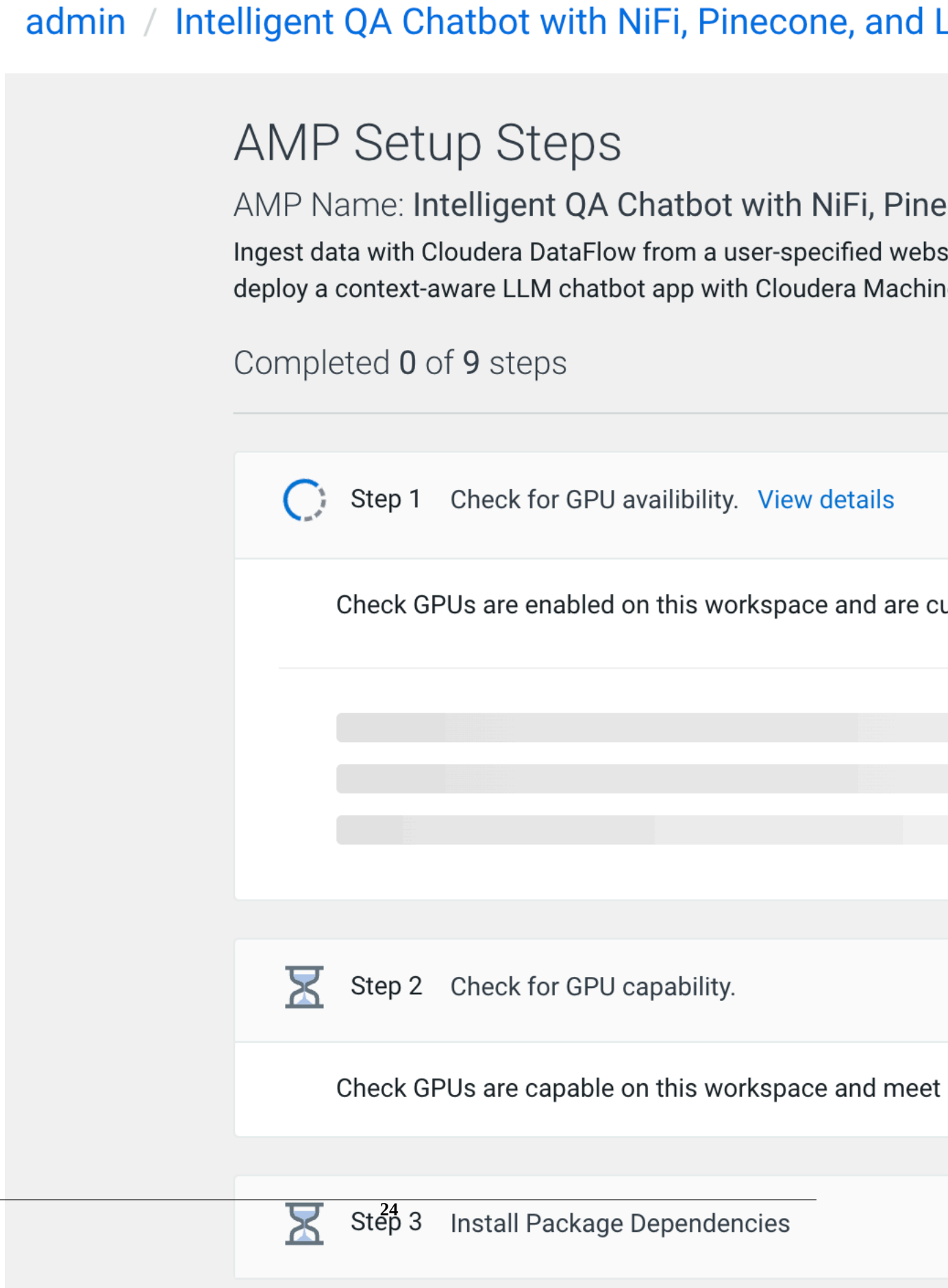
Resuming an AMP

The resume action lets you resume the tasks from the last failed step. This eliminates the need to create a new project to launch the AMP again. You can resume an AMP setup process only from the failure state.

Redeploying an AMP

When you redeploy, it stops all the long-running workloads (Model Deployment or Application) linked to the current prototype, deletes the existing tasks, and re-imports tasks from the .project-metadata.yaml file.

- In the Accelerated ML Projects page, after launching a project, in the AMP Setup Steps window, you can use the Redeploy or Resume option to restart the AMP setup process.



Host names required by AMPs

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

- *.storage.googleapis.com
- *.raw.githubusercontent.com
- *.pypi.org
- *.pythonhosted.org
- github.com
- *.github.com
- huggingface.co
- *.huggingface.co

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 AMPs catalog included at installation and default AMPs may be inaccessible. There are a few options to work around this issue.

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 on page 25.

Option 2: Clone AMPs catalog and projects internally and configure the CML workspace to use their internal catalog

1. Clone the AMPs catalog repository to local: https://github.com/cloudera/Applied-ML-Prototypes.

   Note: Catalog links must be https, not http.

2. Clone all the git projects mentioned in the git_url fields in https://github.com/cloudera/Applied-ML-Prototypes/blob/master/amp-catalog-cloudera-default.yaml.
3. Update the git_url and image_path links to point to your local internal github repository clones (from Step 2) in local amp-catalog-cloudera-default.yaml (from Step 1).
4. Update the AMPs catalog in the CML workspace to point to the local amp-catalog-cloudera-default.yaml (from Step 1). Follow the steps in Add a catalog on page 7.
5. Complete the steps in “Accessing python packages required by AMPs on airgapped setups ” (see below).

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

1. Browse the AMPs catalog page and go to the github project associated with the AMP: https://cloudera.github.io/Applied-ML-Prototypes/#/.

   Note: Catalog links must be https, not http.

2. Download the AMP project as zip file.
3. Go to CML workspace and create a project by choosing file upload method and providing the above AMP zip file.
4. Complete the steps in “Accessing python packages required by AMPs on airgapped setups ” (see below).

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.