

## Setting Up Cloudera AI Inference service

Date published: 2020-07-16

Date modified: 2025-09-30



# 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 AI Inference service Overview.....</b>	<b>4</b>
Key Features.....	4
Key Applications.....	4
Terminology.....	5
Limitations and Restrictions.....	5
Supported Model Artifact Formats.....	6
 <b>Authorization of Cloudera AI Inference service.....</b>	 <b>6</b>
 <b>Cloudera AI Inference service Configuration and Sizing.....</b>	 <b>7</b>
 <b>Prerequisites for setting up Cloudera AI Inference service.....</b>	 <b>10</b>
Importing Models.....	10
Register an ONNX model to Cloudera AI Registry.....	11
 <b>Managing Cloudera AI Inference service.....</b>	 <b>11</b>
Managing Cloudera AI Inference service using the UI.....	11
Creating a Cloudera AI Inference service instance using the UI.....	11
Listing Cloudera AI Inference service instances using the UI.....	13
Viewing details of a Cloudera AI Inference service instances using the UI.....	13
Managing node groups using the UI.....	13
Deleting Cloudera AI Inference service instances using the UI.....	14
Obtaining Control Plane Audit Logs for Cloudera AI Inference service using the UI.....	15
Obtaining the kubeconfig of Cloudera AI Inference service using the UI.....	15
Managing Cloudera AI Inference service using CDP CLI.....	15
Creating a Cloudera AI Inference service instance.....	15
Listing Cloudera AI Inference service instances.....	18
Describing Cloudera AI Inference service instance.....	18
Managing Node Groups.....	19
Deleting Cloudera AI Inference service instance.....	21
Obtaining Control Plane Audit Logs for Cloudera AI Inference service.....	21
Obtaining the kubeconfig for the Cloudera AI Inference service Cluster.....	21

# Cloudera AI Inference service Overview

Cloudera AI Inference service provides a production-grade serving environment for hosting predictive and generative AI. It is designed to handle the challenges of production deployments, such as high availability, performance, fault tolerance, and scalability. The Cloudera AI Inference service allows data scientists and machine learning engineers to deploy their models quickly, without worrying about the infrastructure and maintenance. Cloudera AI Inference service supports running 100 or more model endpoints simultaneously, provided that the underlying compute resources are adequately and correctly sized.

Cloudera AI Inference service is built with tight integration of *NVIDIA NIM* and *NVIDIA Triton Inference Server*, providing industry-leading inference performance on NVIDIA GPUs. Model endpoint orchestration and management are built with the *KServe* model inference platform, a Cloud Native Computing Foundation (CNCF) open-source project. The platform provides standard-compliant model inference protocols, such as *Open Inference Protocol* for predictive models and *OpenAI API* for generative AI models.

Cloudera AI Inference service is seamlessly integrated with the *Cloudera AI Registries* enabling users to store, manage, and track their AI models throughout their lifecycle. This integration provides a central location to store model and application artifacts, metadata, and versions, making it easier to share and reuse AI artifacts across different teams and projects. It also simplifies the process of deploying models and applications to production, as users can select artifacts from the registry and deploy them with a single command.

## Key Features

The key features of Cloudera AI Inference service includes:

- **Easy to use interface:** Streamlines the complexities of deployment and infrastructure, meaningfully reducing time to value for AI use cases.
- **Real-time predictions:** Allows users to serve AI models in real-time, providing low latency predictions for client requests.
- **Monitoring and logging:** Includes functionality for monitoring and logging, making it easier to troubleshoot issues and optimize workload performance.
- **Advanced deployment patterns:** Includes functionality for advanced deployment patterns, such as canary and blue-green deployments, and supports A/B testing, enabling users to deploy new versions of models gradually and compare their performance before deploying them to production.
- **Optimized Performance:** Integrates with NVIDIA NIM microservices and NVIDIA Triton Inference Server to accelerate inference performance on NVIDIA accelerated infrastructure.
- **Model access:** Offers access to NVIDIA foundation models, tailored for NVIDIA hardware to increase inference throughput and to reduce latency.
- **REST API:** Provides APIs for deploying, managing, and monitoring of model endpoints. These APIs enable integration with continuous integration and continuous deployment (CI/CD) pipelines and other tools used in the Machine Learning Operations (MLOps) and Large Language Model Operations (LLMOps) workflows.

## Key Applications

Large Language Models deployed on Cloudera AI Inference service with NVIDIA NIM enable the following applications:

- **Chatbots & Virtual Assistants:** Empowers bots with human-like language understanding and responsiveness.
- **Content Generation & Summarization:** Generates high-quality content or distills lengthy articles into concise summaries with ease.
- **Sentiment Analysis:** Understands user sentiments in real-time, driving better business decisions.
- **Language Translation:** Breaks language barriers with efficient and accurate translation services.

## Terminology

Lists the Cloudera AI Inference service terminology and usage.

- **CML Serving App:** This is the term used by the Cloudera CLI to refer to a specific instance of Cloudera AI Inference service.
- **Model Endpoint:** This refers to a deployed model that has a URL endpoint accessible over the network.
- **Model Artifacts:** Files stored in Cloudera AI Registry that are necessary for deploying an instance of the model, such as model weights, metadata, and so on.
- **API standard:** The protocol that is exposed by a Model Endpoint. It can be either OpenAI (for NVIDIA NIM) or Open Inference Protocol for predictive models.
- **Cloudera Workload Authentication Token:** The bearer token used for authentication / authorization when accessing Cloudera AI Inference service API and model endpoints. Throughout this document this is referred to as “CDP\_TOKEN”.
- **Model ID:** This is the ID assigned to the model when it is registered to the Cloudera AI Registry.
- **Model Version:** This is the version of a registered model in the Cloudera AI Registry.

## Limitations and Restrictions

Lists the limitations and restrictions when using Cloudera AI Inference service.

- **API Stability:** Both the Cloudera AI Control Plane and Cloudera AI Inference service workload APIs and CLIs are under active development and are subject to change in a backward-incompatible way.
- **Cloud Platforms:** Cloudera AI Inference service is available on AWS and Azure.
- **Supported Instance Types:** Cloudera AI Inference service supports the same cloud instance types as those of Cloudera AI Workbenches with a few exceptions. See *Known Issues* for information on unsupported instance types. The type or size of the model you want to deploy determines the cloud compute instance type. Some highly optimized versions of Large Language Models, for instance, work only on specific GPU architectures.
- **No Non-Transparent Proxy Support:** Cloudera AI Inference service has not been tested with a non-transparent proxy (NTP) setup in a private cluster. However, it works in a vanilla private cluster.
- **User-Defined Route (UDR) Support in Azure:** Cloudera AI Inference service provides support for the UDR setup in Azure Kubernetes Service (AKS) clusters. Currently, the compute clusters UI does not support specification of subnets attached to UDRs. As a result, compute clusters utilizing UDR-attached subnets must be created using the CLI.



**Note:** When creating a cluster, ensure that the specified subnet is not used by another AKS cluster.

Example payload for creating compute clusters with a UDR-attached subnet using CLI:

```
{
  "environment": "[**ENVIRONMENT_NAME**]",
  "name": "[**CLUSTER_NAME**]",
  "network": {
    "subnets": [
      "[**SUBNET_NAME**]"
    ],
    "outboundType": "udr"
  },
  "skipValidation": false
}
```

- **Public Load Balancer:** By default, Cloudera AI Inference service uses a private load balancer for cluster ingress. If you use a public load balancer instead, set the `usePublicLoadBalancer` parameter value to `true` in the creation payload.  
  
If you are on AWS and use a private load balancer for cluster ingress, you must have a VPN connection between your corporate network and the Virtual Private Cloud (VPC) in which the Cloudera AI Inference service is deployed. The Cloudera AI Inference service UI requires VPN connection.
- **Logging:** All Kubernetes pod logs, including pods that are running model servers, are scraped by the platform log aggregator service (fluentd). Model endpoint logs can be viewed from the Cloudera AI Inference service GUI. To view logs of other pods, you must first obtain the kubeconfig of the cluster and use the `kubectl` command. Historical logs can be retrieved using the Generate Log Archive feature on the Cloudera AI Inference service administration UI.
- **Namespace:** Model endpoints can only be deployed in the serving-default namespace.

## Supported Model Artifact Formats

Lists Cloudera AI Inference service supported models:

- Text-generating Large Language Models (LLMs), embedding, ranking and object detection models packaged as *NVIDIA NIM*.
- Hugging Face transformer models supported by the vLLM engine.
- Predictive models in the ONNX representation, registered to Cloudera AI Registry from a Cloudera AI Workbench. See *Register an ONNX model to Cloudera AI Registry* as an example showing how to convert a sklearn model to ONNX and then register it to the Cloudera AI Registry. Refer to *Export a PyTorch model to ONNX* or *Getting Started Converting TensorFlow to ONNX* documentation regarding how models using these frameworks can be converted to the ONNX representation.

### Related Information

[Register an ONNX model to Cloudera AI Registry](#)

[Getting Started Converting TensorFlow to ONNX, ONNX Runtime documentation](#)

[Export a PyTorch model to ONNX, PyTorch documentation](#)

[NVIDIA NIM Large Language Models](#)

## Authorization of Cloudera AI Inference service

Cloudera AI Inference service implements role-based access control.

To create an instance of the service in a Cloudera environment, and to perform modifications to the instance, such as the addition of a node group, the user must have the following roles in the environment:

- EnvironmentAdmin
- MLAdmin

Model endpoints can be viewed, created, deleted, and modified by users having EnvironmentUser role along with either one of the following roles in the environment:

- MLAdmin
- MLUser

For information on how to grant the MLUser roles to users/groups, see *Granting Cloudera Data Platform Users Access to Cloudera AI Workbenches*.

### Related Information

[Granting Cloudera Users access to Cloudera AI Workbenches](#)

[Synchronize Users](#)


[Granting CDP Users Access to Cloudera AI Workbenches](#)

# Cloudera AI Inference service Configuration and Sizing

Consider the following factors for the configuration and sizing of Cloudera AI Inference service.

## Node Group Configuration

The configuration and size of Cloudera AI Inference service cluster is determined by the nature of the workloads you expect to deploy on the platform. Certain models might require GPUs, while other kinds of models might run only on CPUs. Similar considerations must be taken for the model endpoints. For example, you must determine the number of replicas of a model that are required to handle normal inference traffic, and there could be peak traffic for which additional replicas shall be spun up to keep user experience at acceptable levels.

 **Note:** Cloudera AI Inference service uses a rolling update strategy for all model endpoints. Consequently, during a rolling update, both the older and newer revisions of the model endpoint are running at the same time until the update is completed. Therefore, your cluster must be able to accommodate this expanded resource footprint in order for the rolling update to succeed.

For example, when you deploy the following NVIDIA NIM for Llama 3.1:

Import Model

aws eng registry-ml-5503aba1-b47

\* Select Model Size

Llama 3.1 8B Instruct

\* Select Optimization

Llama 3.1 8B Instruct A10G BF16 Throughput

PROFILE	PRECISION	GPU	COUNT
Throughput	BF16	A10G	2
GPU DEVICE	NIM VERSION	FEAT_LORA	
2237:10de	1.0.0	false	

In the above image, you can see that each replica of this model requires two A10G GPUs (due to the model being optimized for A10G with a tensor parallelism of two). You can assume that two replicas are required to handle normal traffic, but an additional replica would be required during peak traffic. Therefore, you must configure the model endpoint to autoscale between two and three replicas. Consequently, for normal traffic 4 A10G GPUs must be allocated, and two more for peak traffic. To ensure seamless rolling update for this model endpoint, and assuming updates are made during off-peak traffic, your cluster must be able to add eight A10G GPUs on-demand. An optimal node group configuration for this scenario on AWS would be one that has 0-2 instances of the g5.12xlarge instance type. One instance runs two replicas during normal load, and the second instance is added using autoscaling during peak traffic, and during rolling update. Another, less cost-efficient, possibility is to use a single g5.48xlarge instance type in the node group, in which case all the resource headroom is available on a single node. The larger node also helps to ensure that autoscaling model replicas and rolling updates are quicker as you do not have to wait for a new node to be spun up, and the container images to be pulled.

## Instance Volume Sizing

Cloudera AI Inference service service downloads model artifacts to the instance volume (also known as root volume) of the node where the model replica pod is scheduled. Large Language Model artifacts can be tens to hundreds of gigabytes large. The required instance volume size for a given node group can be estimated using the following formula:

$$S \approx S_o + \sum_i n_i r_i S_i + S_c$$

Where:

- $S_o$ : Size of storage required by the operating system, typically 30 to 40 GB.
- $S_i$ : Size of the  $i$ -th model replica artifacts on a node.
- $r_i$ : Number of replicas of the  $i$ -th model on the node.
- $S_c$ : Total size of all container images on the node. This is dominated by model runtime container images.

For instance, if you want to run 2 replicas of the instruction-tuned Llama 3.1 70b at FP16 precision on a node, you would need something like the following:

$$S \sim 40 + 2 * 148 + 20 = 356 \text{ GB}$$

Where it is assumed that the aggregate size of container images is 20 GB.

Cloudera recommends that the instance volume is slightly over-provisioned to ensure that you do not run out of disk space. In the above example, for instance, it is recommended to round up to 512 GB of instance volume. A larger instance volume also provides higher IOPs, which helps reduce model endpoint startup times. Note that the instance volume of an existing node group cannot be modified. You must first delete the node group and then add it back to the cluster with the new instance volume size.

## Choosing an NVIDIA NIM Profile

The following guidance here is in the context of Cloudera AI Inference service. See *Nvidia documentation* for information about NVIDIA NIM profiles.

NVIDIA NIM comes in three kinds of optimization profiles:

- **Latency**: This profile minimizes Time to First Token (TTFT) and Inter-Token Latency (ITIL) by using higher tensor parallelism, that is, using more GPUs.
- **Throughput**: This maximizes the token throughput per GPU by utilizing the minimum number of GPUs to host the model.
- **Generic**: Unlike the first two profiles, this profile uses the vLLM backend to load the model and run inference against it. This profile provides the most flexibility in terms of choosing GPU models at the cost of lower performance. Note that not all NIMs provide the generic profiles.

For a given precision, the latency profile provides the highest performance by utilizing the maximum number of GPUs while the generic profile offers the most flexibility by sacrificing performance and precision. The throughput profile strikes a good balance between the other two.

Cloudera AI Inference service lets you choose which NVIDIA NIM profile to deploy, so that only artifacts specific to the chosen profile are downloaded to your Cloudera AI Registry to save on storage costs.

The choice of profile is determined by the following:

- Hardware budget - which cloud instance types you have access to.
- Model performance requirement in terms of latency and throughput.

As an example, let us look at some of the available profiles for the instruction-tuned Llama 3.1 8b model. Each entry in the Optimization picker specifies the model name, GPU architecture, floating point precision, and profile type:



\* Select Model Size

Llama 3.1 8B Instruct

\* Select Optimization

Select Optimization

Llama 3.1 8B Instruct H100 FP8 Latency

Llama 3.1 8B Instruct H100 FP8 Throughput

Llama 3.1 8B Instruct A100 BF16 Throughput

Llama 3.1 8B Instruct H100 BF16 Latency

Llama 3.1 8B Instruct H100 BF16 Throughput

Llama 3.1 8B Instruct A10G BF16 Throughput

Llama 3.1 8B Instruct A100 BF16 Latency

Let us compare the difference between A100 BF16 Throughput and A100 BF16 Latency profiles:

\* Select Optimization

Llama 3.1 8B Instruct A100 BF16 Latency

PROFILE	PRECISION	GPU	COUNT
Latency	BF16	A100	2
GPU DEVICE	NIM VERSION	FEAT_LORA	
20b2:10de	1.0.0	false	

\* Select Optimization

Llama 3.1 8B Instruct A100 BF16 Throughput

PROFILE	PRECISION	GPU	COUNT
Throughput	BF16	A100	1
GPU DEVICE	NIM VERSION	FEAT_LORA	
20b2:10de	1.0.0	false	

As shown in the figures, the GPU count per model replica of the latency profile is double the size of the throughput profile.

Quantization, for example, FP8 vs BF16. A quantized model, if available, will have lower resource footprint and higher performance (latency and throughput) than a non-quantized one.

The actual latency and throughput as seen by a client application is affected by the end-end performance of the network between the client and the model server, which will include authentication and authorization checks, in addition to the performance of the chosen NVIDIA NIM profile, as well as the number of concurrent connections. For any chosen profile, end-to-end latency and throughput can be optimized by increasing the number of model endpoint replicas for the model.

### Related Information

[Nvidia Documentation Website](#)

## Prerequisites for setting up Cloudera AI Inference service

Consider the following prerequisites before setting up Cloudera AI Inference service.

### CDP CLI

To manage your Cloudera AI Inference service using the command-line interface (CLI), you must install the CDP CLI on your local machine. However, if you intend to manage the service solely using the user interface (UI), the CDP CLI is not required.

To create a Cloudera AI Inference service instance using CLI, you must first download and install the CDP CLI on your local machine. To access the download instructions, navigate to [Help Download CLI](#) from the left navigation menu in the Cloudera Console. You can use CDP CLI version 0.9.123 or later but it is recommended to use the latest available version of the CDP CLI for compatibility with Cloudera AI Inference service.

### Compute cluster-enabled Cloudera environment

Cloudera AI Inference service requires a compute cluster-enabled Cloudera environment. You must either create the compute cluster-enabled environment, or convert your existing environment.



**Note:** You cannot deploy Cloudera AI Inference service into the **default compute cluster** in the environment. You must create a separate compute cluster.

For information on enabling computer clusters on CDP, see *Using Compute clusters on AWS* or *Using Compute clusters on Azure*.

### Data Lake and Cloudera Manager supported versions

JSON Web Token based authentication from Cloudera AI Workbenches to Cloudera AI Inference service requires Data Lake version 7.2.18 or above, and Cloudera Manager version 7.12 or above.

### Cloudera AI Registry

A Cloudera AI Registry must first be deployed in the same Cloudera environment in which you plan to deploy Cloudera AI Inference service. That is, Cloudera AI Inference service can only deploy models registered to a Cloudera AI Registry in the same Cloudera environment. For this release, Cloudera AI Registry version CML2024.09-1 or higher version must be created before provisioning the Cloudera AI Inference service. If you have an existing Cloudera AI Registry in the environment, you must first upgrade it to version CML2024.09-1 before provisioning Cloudera AI Inference service.

### Related Information

[Installing Cloudera client](#)

[Using Compute Clusters in AWS environment](#)

[Using Compute Clusters in Azure environment](#)

[Setting up AI Registry](#)

## Importing Models

Model Hub offers a curated list of top-performing models from the NVIDIA NGC Catalog and Hugging Face.

The **Model Hub** page displays the list of different models along with their source type, tags, and description. You can import models listed on the **Model Hub** page into the **AI Registries**. Imported models can be seen on the **Registered Models** page, and can be deployed using Cloudera AI Inference service.

### Importing NVIDIA Foundation Model NIM

You can import NVIDIA NGC Catalog models listed on the **Model Hub** page into the Cloudera AI Registry. Imported models can be seen on the **Registered Models** page, and can be deployed using Cloudera AI Inference service.

For more information, see *Importing models from NVIDIA NGC*.

### Importing a Hugging Face Model

You can import Hugging Face models listed on the **Model Hub** page into the Cloudera AI Registry. Imported models can be seen on the **Registered Models** page, and can be deployed using Cloudera AI Inference service.

For more information, see *Importing a Huggingface Model from Model Hub*.

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

For more information, see *Importing a Huggingface Model from the Hugging Face website*.

### Related Information

[Importing models from NVIDIA NGC](#)

[Importing a Huggingface Model from Model Hub](#)

[Importing a Huggingface Model from the Hugging Face website](#)

## Register an ONNX model to Cloudera AI Registry

Register your ONNX model to Cloudera AI Registry.

To deploy a predictive ONNX model, first you must register your model to the Cloudera AI Registry running in the same environment as your Cloudera AI Inference service.

For information on how to register an ONNX model from a Cloudera AI Workbench, see *Deploying a Predictive Deep Learning Model*. For more information, see [Registering a model using the Cloudera AI Registry user interface](#).

### Related Information

[Deploying a Predictive Deep Learning Model](#)

[Registering a model using the AI Registry user interface](#)

## Managing Cloudera AI Inference service

Cloudera AI Inference service provides an UI and CLI interface to manage the life cycle of the service and associated infrastructure.

### Managing Cloudera AI Inference service using the UI

You can manage the life cycle of the Cloudera AI Inference service and associated infrastructure using the UI.

### Creating a Cloudera AI Inference service instance using the UI

You can create a Cloudera AI Inference service instance using the UI.

### Before you begin

Cloudera AI Inference service requires a compute cluster enabled Cloudera environment. You must either create the compute cluster enabled environment or convert your existing environment. For more information, see *Prerequisites for setting up Cloudera AI Inference service*.

## Procedure

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

The **Cloudera AI Workbenches** page displays.

2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.

The AI Inference Services page is displayed.

3. Click the Create AI Inference Service button.

The Create AI Inference Service page is displayed.

AI Inference Services / Create AI Inference Service

Configure AI Inference Service

\* Name  
cai-inf-marketing

Select Environment  
aws eng-ml-dev-aws-v2-public (us-west-2)

\* Select Compute Cluster  
Select Compute Cluster

**CPU Node Groups**

Instance Type: c4.2xlarge 8 CPU 15 GiB Autoscale Range: 0 to 100

**GPU Node Groups**

Instance Type: p3.2xlarge 8 CPU 1 GPU 61 GiB Autoscale Range: 0 to 100

**Network Settings**

Subnets for Worker Nodes: usw2-public-public-us-west-2a usw2-public-public-us-west-2b

Subnets for Load Balancer:

Create Cancel

4. In the Name textbox, enter a name for the Cloudera AI Inference service instance.
5. In the Select Environment dropdown list, select your Cloudera environment within which you want to create the service.
6. In the Select Compute Cluster dropdown list, select the compute cluster.
7. Optional: In CPU Node Groups, select the CPU instance type and autoscale range. You can add more than one CPU node group by clicking the + button.
8. Optional: In GPU Node Groups, select the GPU instance type and autoscale range. You can add more than one GPU node group by clicking the + button.
9. In Network Settings, select the subnets for worker nodes and load balancer. You can select multiple subnets.
10. In Load Balancer Source Ranges, specify the IP ranges that are allowed to access the load balancer.
11. Select the Enable Public IP Address for Load Balancer checkbox to make Cloudera AI Inference service available on the public internet. When disabled, it is assumed that connectivity is achieved through a corporate Virtual Private Cloud (VPC).
12. Recommended: In Static Subdomain, specify the static domain for the AI Inference Service.
13. Select the Skip Validation checkbox if you do not want to perform the validation checks before provisioning this AI Inference Service.
14. In Tags, add any custom key and value pairs for your own use.
15. Click Create.

### Related Information

[Installing Cloudera client](#)

[Using Compute Clusters in AWS environment](#)

[Using Compute Clusters in Azure environment](#)

[Setting up AI Registry](#)

## Listing Cloudera AI Inference service instances using the UI

You can view the list of all the Cloudera AI Inference service instances in your Cloudera tenant. It provides details like the name of the Cloudera AI Inference service, status, associated environment name, created date and time, and the cloud provider details.

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. You can use the filter bar at the top of the window to filter the list of Cloudera AI Inference service instances by service name and environment name.
4. Select a Cloudera AI Inference service instance to see its description.

## Viewing details of a Cloudera AI Inference service instances using the UI

You can view detailed configuration information about a specific Cloudera AI Inference service instance, including which node groups are currently configured.

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. Select a Cloudera AI Inference service instance to see its description.
4. You can view the configuration information like CRN, Environment CRN, Version, node configuration, events and logs, and so on.

## Managing node groups using the UI

You can add, modify, or delete node groups to or from your Cloudera AI Inference service instance.

### Adding a node group to an existing instance using the UI

You can add one or more node groups to your cluster if you do not have the right worker node hardware (for example, incorrect GPU models for a new workload) in your existing cluster configuration.

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. Select a Cloudera AI Inference service instance to which you want to add a node group.
4. Click Add CPU Node Group to add CPU nodes and select the instance type. Click Add GPU Node Group to add GPU nodes and select the instance type.

5. In the Autoscale Range Min - Max textbox, specify the autoscale range.

The screenshot shows the Cloudera AI Inference Services console. On the left is a navigation menu with options like AI HUB, Model Hub, DEPLOYMENTS, Model Endpoints, Registered Models, ADMINISTRATION, AI Workbenches, AI Inference Services (selected), AI Registries, and AI Workbench Backups. The main panel displays details for a specific AI Inference Service instance, including its CRN, Creation Date, Domain, Endpoint Public Access, and Version. Below this is a table of node groups with columns for Name, Instance Type, CPU, GPU, Memory, Count, and Autoscale Range (Min - Max). The table lists several node groups, including Cloudera AI CPU Workers and Cloudera AI GPU Workers. The 'Standard\_D12\_v2' node group is highlighted, and its 'Autoscale Range' is set to '0 - 2'. At the bottom of the table, there are buttons to '+ Add CPU Node Group' and '+ Add GPU Node Group'.


Name	Instance Type	CPU	GPU	Memory	Count	Autoscale Range Min - Max
Cloudera AI CPU Workers	Standard_D3_v2	4 CPU	-	14 GiB	1	1 - 5
Cloudera AI CPU Workers	Standard_D12_v2	4 CPU	-	28 GiB	0	0 - 1
Cloudera AI CPU Workers	Standard_D13_v2	8 CPU	-	56 GiB	0	0 - 1
Cloudera AI CPU Workers	Standard_D14_v2	16 CPU	-	112 GiB	0	0 - 1
Cloudera AI GPU Workers	Standard_D15_v2	20 CPU	-	140 GiB	0	0 - 1
Cloudera AI Infra	Standard_D8_v3	8 CPU	-	32 GiB	2	2 - 3
Platform Infra	Standard_D16_v3	16 CPU	-	64 GiB	2	2 - 4
Cloudera AI CPU Workers	Standard_D12_v2	4 CPU	-	28 GiB	0	0 - 2

6. Click Save.

### Modifying a node group in an existing instance using the UI

You can reconfigure an existing node group including the autoscaling range of an AI Inference service instance.


#### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. Select a Cloudera AI Inference service instance to which you want to modify a node group..
4. Click  in the node group you want to modify. You can modify the instance type and the autoscale range.
5. Click Save.

### Deleting a node group from an existing Cloudera AI Inference service instance using the UI

You can delete a node group from an existing Cloudera AI Inference service instance.

#### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. Select a Cloudera AI Inference service instance in which you want to delete a node group..
4. Click  in the node group you want to delete. A confirmation message will be displayed.
5. Click OK to delete the node group.

### Deleting Cloudera AI Inference service instances using the UI

You can delete a Cloudera AI Inference service instance if it is no longer needed..

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. From the Actions menu, click Delete. A confirmation message will be displayed.
4. Click OK to delete the instance.



**Caution:** If the deletion of the Cloudera AI Inference service instance fails a few times, you can try to delete it forcefully by selecting the Force Delete checkbox. It is not recommended to use the Force Delete option unless the graceful deletion is not successful.

## Obtaining Control Plane Audit Logs for Cloudera AI Inference service using the UI

You can obtain Control plane audit logs for Cloudera AI Inference service. You can use the audit log entries and the logs for troubleshooting purposes.

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. Select a Cloudera AI Inference service instance for which you want to obtain the event logs.
4. Click Events & Logs to view the event history.  
The **Events History** page is displayed.
5. In the **Events History** page, click View Event Logs to view the event history of the specific event. The log of the specific event is displayed.
6. Click Download Logs to download the logs of that specific event.

## Obtaining the kubeconfig of Cloudera AI Inference service using the UI

To troubleshoot Kubernetes-level issues in the cluster, such as installation or upgrade failure, or any other cluster level issue, you must first obtain the kubeconfig of the cluster.

### Procedure

1. In the **Cloudera** console, click the Cloudera AI tile.  
The **Cloudera AI Workbenches** page displays.
2. Click AI Inference Services under ADMINISTRATION on the left navigation menu.  
The AI Inference Services page is displayed.
3. From the Actions menu of the Cloudera AI Inference service instance, click Download Kubeconfig.  
The `[***INSTANCE-NAME****]/kubeconfig.yaml` file will be downloaded to your local machine.

## Managing Cloudera AI Inference service using CDP CLI

Cloudera AI Inference service provides a CLI interface to manage the life cycle of the service and associated infrastructure.

## Creating a Cloudera AI Inference service instance

The recommended way to create a Cloudera AI Inference service is to first generate the CLI input skeleton, customize the JSON file, and then pass the file to the creation command.

The following example present how to create a Cloudera AI Inference service by generating the CLI input skeleton, customizing the JSON file, and then passing the file to the creation command:

1. Generate the JSON skeleton payload and save it to a file:

```
$ cdp ml create-ml-serving-app --generate-cli-skeleton > /tmp/create-serving-app-input.json
```

2. Customize the JSON file to use it when creating a Cloudera AI Inference service instance. The following are the sample JSON files of AWS and Azure:

AWS JSON file example

```
{
  "appName": "my-aws-caii-cluster",
  "environmentCrn": "[***CDP-ENVIRONMENT-CRN***]",
  "clusterCrn": "[***COMPUTE-CLUSTER-CRN***]",
  "provisionK8sRequest": {
    "instanceGroups": [
      {
        "instanceType": "m5.4xlarge",
        "instanceTier": "ON-DEMAND",
        "instanceCount": 1,
        "name": "[***OPTIONAL-LEAVE BLANK***]",
        "rootVolume": {
          "size": 256
        },
        "autoscaling": {
          "minInstances": 0,
          "maxInstances": 5,
          "enabled": true
        }
      },
      {
        "instanceType": "p4de.24xlarge",
        "instanceCount": 1,
        "rootVolume": {
          "size": 1024
        },
        "autoscaling": {
          "minInstances": 0,
          "maxInstances": 5,
          "enabled": true
        }
      }
    ],
    "environmentCrn": "[***CDP-ENVIRONMENT-CRN***]",
    "tags": [
      {
        "key": "experience",
        "value": "cml-serving"
      }
    ]
  },
  "usePublicLoadBalancer": true,
  "skipValidation": false,
  "loadBalancerIPWhitelists": [
    ""
  ],
  "subnetsForLoadBalancers": [
    ""
  ],
  "staticSubdomain": "mydomain"
```



```
}
```

### Azure JSON file example

```
{
  "appName": "my-azure-caii-cluster",
  "environmentCrn": "[***CDP-ENVIRONMENT-CRN***]",
  "clusterCrn": "[***COMPUTE-CLUSTER-CRN***]",
  "provisionK8sRequest": {
    "instanceGroups": [
      {
        "instanceType": "Standard_D4s_v3",
        "instanceCount": 1,
        "rootVolume": {
          "size": 256
        },
        "autoscaling": {
          "minInstances": 0,
          "maxInstances": 5,
          "enabled": true
        }
      },
      {
        "instanceType": "Standard_ND96asr_A100_v4",
        "instanceCount": 1,
        "rootVolume": {
          "size": 1024
        },
        "autoscaling": {
          "minInstances": 0,
          "maxInstances": 5,
          "enabled": true
        }
      }
    ],
    "environmentCrn": "[***CDP-ENVIRONMENT-CRN***]",
    "tags": [
      {
        "key": "experience",
        "value": "cml-serving"
      }
    ]
  },
  "usePublicLoadBalancer": true,
  "skipValidation": false,
  "loadBalancerIPWhitelists": [
    ""
  ],
  "subnetsForLoadBalancers": [
    ""
  ],
  "staticSubdomain": "mydomain"
}
```

3. Use the JSON file created in the previous step to create the Cloudera AI Inference service instance:

```
$ cdp ml create-ml-serving-app --cli-input-json file:///tmp/create-serving-app-input.json
```

After a successful invocation of the create command, the CRN of the Cloudera AI Inference service instance that is created is displayed. The command adds the requested compute worker node groups to the existing Kubernetes cluster specified by the clusterCrn field in the request body, and installs the necessary software components.

A typical configuration with two worker node groups would take about 15-20 minutes to complete creation.



**Note:** In the above example, you set the staticSubdomain parameter to mydomain. Set this parameter, if necessary, to have a custom ingress subdomain for your Cloudera AI Inference service. If you leave this value blank, a randomly generated subdomain name, such as ml-92373b63-5f1, will be created.

## Listing Cloudera AI Inference service instances

You can list the Cloudera AI Inference service instances in your Cloudera tenant with the help of a command.

Use the following command to list Cloudera AI Inference service instances in your Cloudera tenant, across all environments:

```
$ cdp ml list-ml-serving-apps
```

A Cloudera AI Inference service instance that has been created successfully shall show the status as installation:finished similar to the following:

```
{
  "cloudPlatform": "AZURE",
  "appName": "serving-multi-gpu-az",
  "appCrn": "[***APPLICATION-CRN***]",
  "environmentCrn": "[***ENVIRONMENT-CRN***]",
  "environmentName": "eng-ml-env-azure-v2",
  "ownerEmail": "admin@example.com",
  "mlServingVersion": "1.2.0-b72",
  "isPrivateCluster": false,
  "creationDate": "2024-08-01T16:36:32.811000+00:00",
  "cluster": {
    "clusterName": "ml-92373b63-5f1",
    "domainName": "ml-92373b63-5f1.eng-ml-e.xcu2-8y8x.dev.cldr.
work",
    "liftieID": "liftie-wqq856vz",
    "isPublic": false,
    "ipAllowlist": "0.0.0.0/0",
    "authorizedIpRangesAllowList": false,
    "tags": [],
    "instanceGroups": [],
    "clusterCrn": "[***CLUSTER-CRN***]"
  },
  "status": "installation:finished",
  "usePublicLoadBalancer": false,
  "httpsEnabled": true,
  "subnetsForLoadBalancers": []
}
```

## Describing Cloudera AI Inference service instance

The describe command shows you all the detailed configuration information about a specific Cloudera AI Inference service, including which node groups are currently configured.

You can describe a Cloudera AI Inference service instance using the following command:

```
$ cdp ml describe-ml-serving-app --app-crn [***APPLICATION-CRN***]
```

The following is a sample output from the above command, edited for brevity:

```
{
  "app": {
    "cloudPlatform": "AZURE",
    "appName": "serving-multi-gpu-az",
    "appCrn": "[***APPLICATION-CRN***]",
    "environmentCrn": "[***ENVIRONMENT_CRN***]",
    "environmentName": "eng-ml-int-env-azure-v2",
    "ownerEmail": "admin@example.com",
    "mlServingVersion": "1.2.0-b69",
    "isPrivateCluster": false,
    "creationDate": "2024-09-03T18:57:16.288000+00:00",
    "cluster": {
      "clusterName": "ml-b0504a3f-bbc",
      "domainName": "ml-b0504a3f-bbc.eng-ml-i.svbr-nqvp.int.cldr.wor
k",
      "liftieID": "liftie-vjyrtsfn",
      "isPublic": false,
      "ipAllowlist": "0.0.0.0/0",
      "authorizedIpRangesAllowList": false,
      "tags": [],
      "instanceGroups": [
        ...
      ],
      "clusterCrn": "crn:[***CLUSTER-CRN***]"
    },
    "status": "installation:finished",
    "usePublicLoadBalancer": false,
    "httpsEnabled": true,
    "subnetsForLoadBalancers": []
  }
}
```

## Managing Node Groups

You can add or delete node groups to your cluster.

### Adding a Node Group to an existing instance

You can add one or more node groups to your cluster if you do not have the right worker node hardware (for example, incorrect GPU models for a new workload) in your existing cluster configuration.

#### Procedure

1. Generate the JSON skeleton to create the instance:

```
$ cdp ml add-instance-groups-ml-serving-app --generate-cli-skeleton > /tmp/add-instance-group.json
```

2. Edit the file to add the node group details:

```
{
  "appCrn": "[***APPLICATION-CRN***]",
  "instanceGroups": [
    {
      "instanceType": "g5.8xlarge",
      "instanceCount": 0,
      "rootVolume": {
```

```

        "size": 1024
      },
      "autoscaling": {
        "minInstances": 0,
        "maxInstances": 4,
        "enabled": true
      }
    },
    {
      "instanceType": "p3.8xlarge",
      "instanceCount": 0,
      "rootVolume": {
        "size": 1024
      },
      "autoscaling": {
        "minInstances": 0,
        "maxInstances": 4,
        "enabled": true
      }
    }
  ]
}

```

3. Use the JSON file created in the previous step to add the node group:

```
$ cdp ml add-instance-groups-ml-serving-app --cli-input-json file:///tmp/add-instance-group.json
```

### Modifying a node group in an existing instance

Follow the instructions to reconfigure an existing node group.

You can reconfigure an existing node group to change its autoscaling range, you can use the following command:

```
$ cdp ml modify-ml-serving-app --app-crn [***APPLICATION-CRN***] --instance-group-name [***GROUP-NAME***] --min [***MINIMUM-NODE***] --max [***MAXIMUM-NODE***] --instance-type [***INSTANCE-TYPE***]
```

Consider the following example:

```
$ cdp ml modify-ml-serving-app --app-crn <your-application-crn> --instance-group-name mlgpu1 --min 2 --max 4 --instance-type g5.24xlarge
```

The instance group name can be obtained using the `describe-ml-serving-app` command.

### Related Information

[Describing Cloudera AI Inference service instance](#)

### Deleting a node group from an existing Cloudera AI Inference service instance

Consider the guidelines here to delete a node group from an existing Cloudera AI Inference service instance.

Use the following command to delete a worker node group:

```
$ cdp ml delete-instance-group-ml-serving-app
```

Consider the following example:

```
$ cdp ml delete-instance-group-ml-serving-app --app-crn <YOUR-APP-CRN> --instance-group-name mlgpu2
```

### Related Information

[Describing Cloudera AI Inference service instance](#)

## Deleting Cloudera AI Inference service instance

Consider the following instructions for deleting Cloudera AI Inference service instance.

You can delete a Cloudera AI Inference service instance if it is no longer needed:

```
$ cdp ml delete-ml-serving-app --app-crn <app-crn>
```



**Note:** The above command only deletes the Cloudera AI Inference service and its associated cloud resources, such as cluster node groups and load balancers. The underlying compute cluster will not be deleted. If you want to delete the compute cluster itself once Cloudera AI Inference service has been deleted, you can do it from the **Compute Clusters** tab in the environment UI.



**Caution:** Always delete Cloudera AI Inference service first before deleting the underlying compute cluster.



**Caution:** If the deletion of the Cloudera AI Inference service instance fails a few times, you can try to delete it forcefully by using the `cdp ml delete-ml-serving-app --app-crn <app-crn> --force` command. It is not recommended to use the `--force` option unless the graceful deletion is not successful.

## Obtaining Control Plane Audit Logs for Cloudera AI Inference service

Consider the following instructions for obtaining Control plane audit logs for Cloudera AI Inference service.

Use the following commands to obtain audit log entries and the logs therein for troubleshooting purposes:

```
cdp ml get-audit-events --resource-crn [***APPLICATION-CRN***]
cdp ml get-logs --request-id [***REQUEST-ID***] --resource-crn [***APPLICATION-CRN***]
```

## Obtaining the kubeconfig for the Cloudera AI Inference service Cluster

Troubleshoot Kubernetes-level issues in the cluster.

To troubleshoot Kubernetes-level issues in the cluster, such as installation or upgrade failure, or any other cluster level issue, you must first obtain the kubeconfig of the cluster.

### Related Information

[Granting Remote Access to ML Workspaces](#)

### Obtaining kubeconfig on AWS

Follow the guidelines for obtaining kubeconfig on AWS.

### About this task

You need the user's Amazon Resource Name (ARN) to obtain the kubeconfig file. You can get the ARN from the user or look up a user's ARN in your AWS account. To obtain a user's ARN from the AWS account, complete the following steps:

### Procedure

1. Obtain the Amazon Resource Name (ARN) .

- a) Using AWS UI, go to your organization's AWS Account Identity and Access Management (IAM) Users and look up the user. The ARN is available on the **Summary** page.
- b) Using AWS CLI, run the following command to get the ARN:

```
$ aws sts get-caller-identity
```

Consider the following example:

```
# Sample output
```

```
{
  "UserId": "ABCDE12345FGHIJKLMNO6789",
  "Account": "88888888888888",
  "Arn": "arn:aws:iam::888888888888:user/joesmith"
}
```

2. Obtain your Cloudera AI Inference service CRN, run the following command:

```
$ cdp ml list-ml-serving-apps
```

3. Grant access to the Cloudera AI Inference service AWS cluster, run the following command:

```
$ cdp ml grant-ml-serving-app-access \
  --resource-crn [***APPLICATION-CRN***] \
  --identifier [***AWS_ARN***]
```

4. Obtain the kubeconfig used to access the above cluster by running:

```
$ cdp ml get-ml-serving-app-kubeconfig \
  --app-crn [***APPLICATION-CRN***] \
  | jq '.kubeconfig' | yq > myconfig.yaml
```

5. Send the downloaded Kubernetes configuration file to the user who has been granted access. To connect to the Amazon EKS cluster, you must have aws-iam-authenticator installed.
6. Get the logs of the Cloudera AI Inference service API server:

```
$ KUBECONFIG=myconfig.yaml kubectl get logs deployment/api -n cml-serving
```

### Obtaining kubeconfig on Azure

Consider the following guidelines for obtaining kubeconfig on Azure.

### Procedure

Locate the liftie cluster identifier for your Cloudera AI Inference service by issuing a describe command using Cloudera CLI:

```
$ cdp ml describe-ml-serving-app --app-crn [***APPLICATION-NAME***]
```

```
{
  "app": {
    "appName": "[***APPLICATION-NAME***]",
    "appCrn": "[***APPLICATION-CRN***]",
    "environmentCrn": "[***ENVIRONMENT-CRN***]",
    "environmentName": "[***ENVIRONMENT-NAME***]",
    "ownerEmail": "user@org.com",
    "mlServingVersion": "1.2.0-b72",
    "isPrivateCluster": false,
    "creationDate": "2024-05-07T14:27:16.817000+00:00",
    "cluster": {
      "clusterName": "ml-1fcaa8cf-a94",
      "domainName": "[***DOMAIN-NAME***]",
      "liftieID": "liftie-3544nrdr",
      "isPublic": false,
      "ipAllowlist": "0.0.0.0/0",
      "authorizedIpRangesAllowList": false
    },
    "status": "installation:finished",
    "usePublicLoadBalancer": false,
    "httpsEnabled": true
  }
}
```

```
}
```

The cluster identifier is in `app.cluster.lifetieID`.

Locate this entry in the Azure Portal's Kubernetes services page and do the following:

- a) Open the entry.
- b) Click on **Connect**.
- c) Copy the command under **Download cluster credentials** to your terminal and execute.

You are now authenticated and your `kubectl` context is set up to interact with the AKS cluster.