Cloudera AI

Using quota management (Tech Preview)

Date published: 2020-07-16
Date modified: 2025-10-31

https://docs.cloudera.com/

Legal Notice

Contents

Quota Management overview

Quota Management enables you to control how resources are allocated within your Cloudera AI Workbench on user and on group level.

In order to prevent a single session, job, or other workload from consuming all the available cluster resources, you can limit the number of CPUs, the amount of memory allocated by user, business units, or Data Service by defining resource pools that set resource limits.

Pools are organized in a hierarchical manner, defining nodes in a hierarchy based on resource limits.

A regular workflow of managing quotas constitutes of the following activities:

1. Enabling Quota management in Cloudera AI
2. Provisioning a workbench using a configured resource pool

Note:

Quota Management cannot be enabled for an existing workbench. It is recommended for Cloudera AI administrators to provision a new workbench to enable and test this feature.

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

Related Information

Enabling Quota Management in Cloudera AI
Provisioning a workbench
Limitations for Quota Management

Enabling Quota Management in Cloudera AI

To enable Quota Management in Cloudera AI it needs to be configured. Follow the recommended configuration guidelines.

Before you begin

Set up the Kubernetes and kubectl as defined in Prerequisites.

Note: The Quota Management feature cannot be enabled for an existing workbench. Cloudera recommends that the Administrator provisions a new workbench and enables the Quota Management feature.

Procedure

1. Configure the kubectl using the kubeconfig file.
2. Edit the Cloudera AI control plane deployment:

```
kubectl get deploy dp-mlx-control-plane-app -n <cdp-namespace> -o yaml
> <file-name>
```

This will save the Cloudera AI control plane deployment specification.

3. Edit the Cadence Worker deployment with the following command:

```
kubectl get deploy dp-cadence-worker -n <cdp-namespace> -o yaml > <caden
ce-file-name>
```

   The command saves both control plane and Cadence Worker deployment specification.
4. Take a backup of the above files.
5. Search for the environment variable `ENABLE_UMBRA_INTEGRATION` in both files, and change the value to true.
6. Save the deployment files and run the following commands:

```
kubectl apply -f <file-name>
```

```
kubectl apply -f <cadence-file-name>
```

   Wait for the new pods to come up.
7. Verify the new pods by running the following command:

```
kubectl get pods -n <cdp-namespace>
```

Provisioning a workbench

Configuring the quota for a specified workbench requires additional configuration settings besides provisioning the workbench for Quota Management.

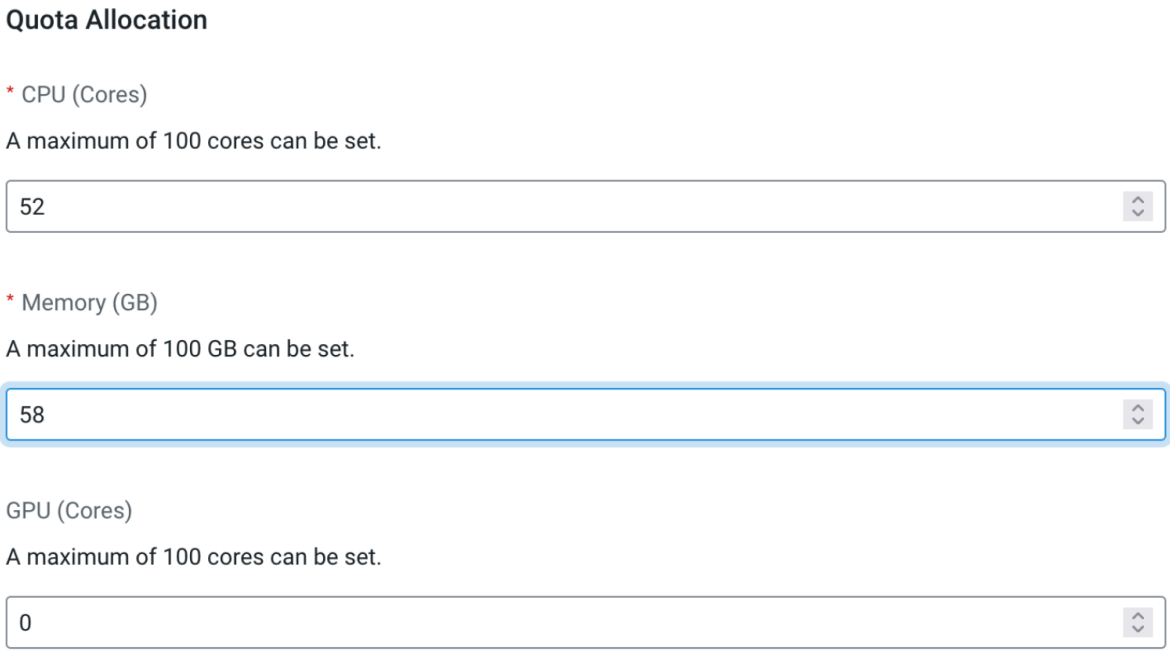
Cloudera AI reserves 50 GB Memory and 50 CPU for the Cloudera AI Workbench installation.

The minimum requirement for a Cloudera AI Workbench to be provisioned is 58 GB of Memory and 52 CPU.

Note: Starting from Cloudera AI version 1.5.5 you no longer need to create a resource pool before creating the workbench.

During the provisioning of the workbench, a service pool, named root.env-name.cml is created within the environment you selected while provisioning the workbench.

The service pool does not have a quota assigned by default. You are required to update the quota for the service pool through the **Resource Utilization** page.



In this example out of 100 Cores and 100 GB, 50 Cores and 50 GB will be reserved for the Cloudera AI Workbench.

Out of these resources, 50 cores and 50 GBs will be used for Cloudera AI, while 2 cores and 8 GBs are available to run workloads.

The rest of the resources is available in root.default.cml resource pool and can be allocated to other Cloudera AI Workbenches or to Cloudera.

Related Information

Managing cluster resources using Quota Management

Upgrade Cloudera AI Workbench to the latest version

To use the latest Quota Management feature, upgrade Cloudera AI Workbench to the latest version.

Note:

Upgrade your Cloudera AI Workbench only if you have already enabled the Quota Management Technical Preview feature in your existing setup. If you have upgraded your Cloudera AI Workbench, you need to enable the Quota Management feature again.

However, Cloudera does not support upgrading Cloudera AI Workbenches from earlier versions without the Quota Management feature enabled to later versions where the feature is enabled.

In Cloudera AI on premises 1.5.5, the root.default pool is deprecated, and the existing hierarchy under the default pool is migrated based on the environment used while provisioning the AI workbench.

After upgrading to the latest version of Cloudera AIon premises, the environment and service pool will be created with an unset quota. To ensure the proper functioning of the Cloudera AI service, it is essential to update the quota for the service pool. For more information, refer to Updating the quota for the service pool after the upgrade.

Related Information

Upgrade from 1.5.2 or 1.5.3 to 1.5.4 (OCP)

Updating the quota for the service pool after the upgrade

Learn about updating the quota for the service pool following the upgrade.

Procedure

1. In the Cloudera console, click the Cloudera Management Console tile.
2. Click **Resource Utilization** in the left navigation panel.
3. Select the Quotas tab.
4. Find the root.<env>.cml service pool.
5. Update the quota for the root.<env>.cml service pool.

Quota for Cloudera AI workloads

Quota management is implemented for both user and group level. A Cloudera AI Workbench is allocated a set amount of resources based on configured parameters at provisioning time. Within a workbench, resources available for workloads can be further subdivided into quotas at user and/or team level.

User and team naming limitations

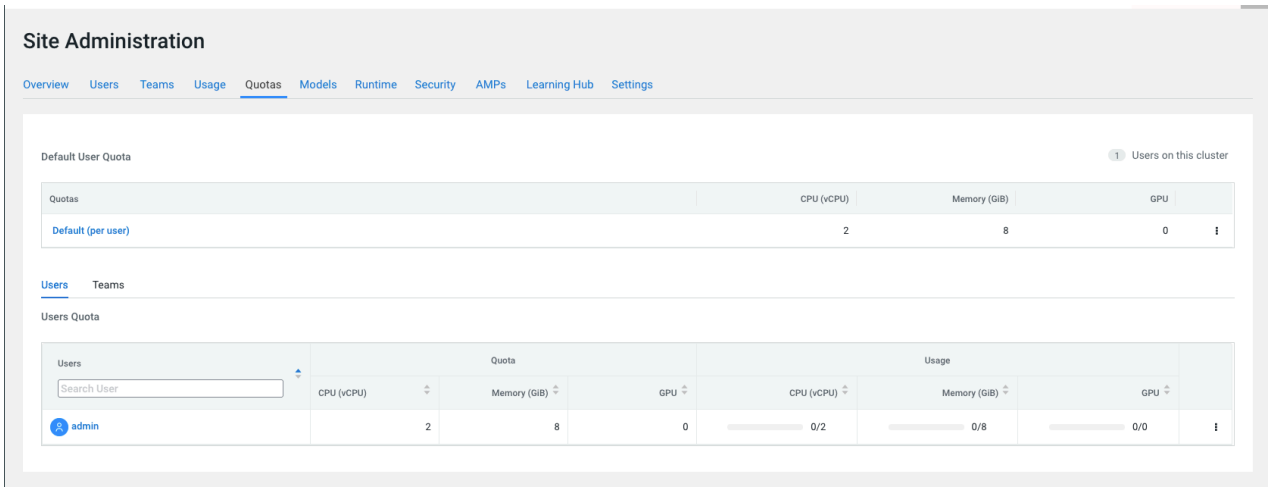
Follow the kubernetes label naming conventions when naming users and teams:

- The name must be 63 characters or less.
- The name must begin and end with an alphanumeric character ([a-z0-9A-Z]).
- The name can contain dashes (-), underscores (_), dots (.), and alphanumerics, except for the beginning and ending characters.

User quota

By default, 8 GB memory and 2 vCPU cores are configured for each user. Such resources are sufficient for running simple sessions but might not be sufficient for the spark workloads if the executors cannot find enough resources. The Cloudera AI administrators can configure custom quota for the user on the Site Administration Page.

Figure 1: User quota settings



If the quota for a user is used up, the workload remains in the pending state until the required resources are available.

If the quota for the users is modified, it will be reflected when the next workload is submitted.

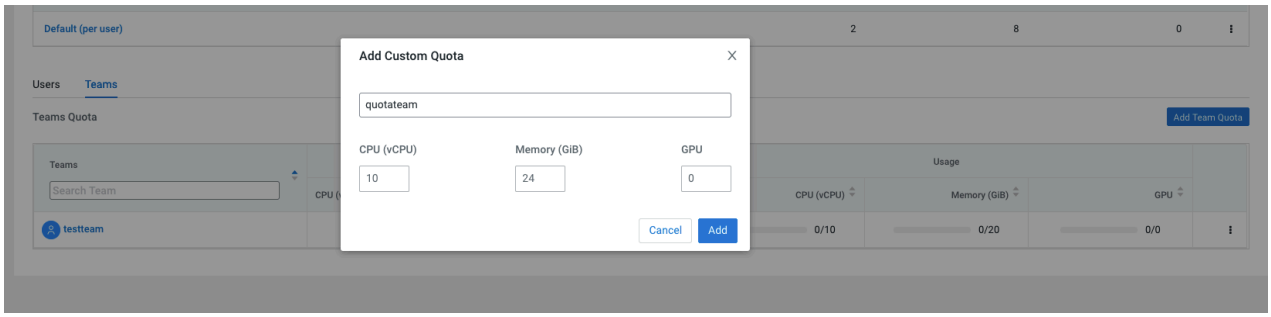
GPU resources can be edited if the workbench is provisioned with GPU resources.

Group quota

There is no quota configured by default for a team or group.

1. As a Cloudera AI administrator configure a custom quota for a group under Site Administration Quotas tab.
2. Configure a quota for a group, at the Quotas Teams tab.
3. Click the Add Team Quota and a popup enables you to add a custom team quota.

Figure 2: Custom quota



If the quota for the group is used up, the workload remains in PENDING state until the required resources are available.

If the quota for the group is modified, it will be reflected when the next workload is submitted.

Quota enforcement

The resources are allocated based on the project context from which the workloads are created.

- If a workload is created in a project, which is developed in users' context, it will always take the user quota.
- If a workload is created in a project which is developed in a team's context, it will take resources from the team's quota, provided the team has a quota configured.
- If a workload is created in a project which is developed in a team's context and the team does not have a quota configured, it will take the resources from the users' quota.

Related Information

Creating a team

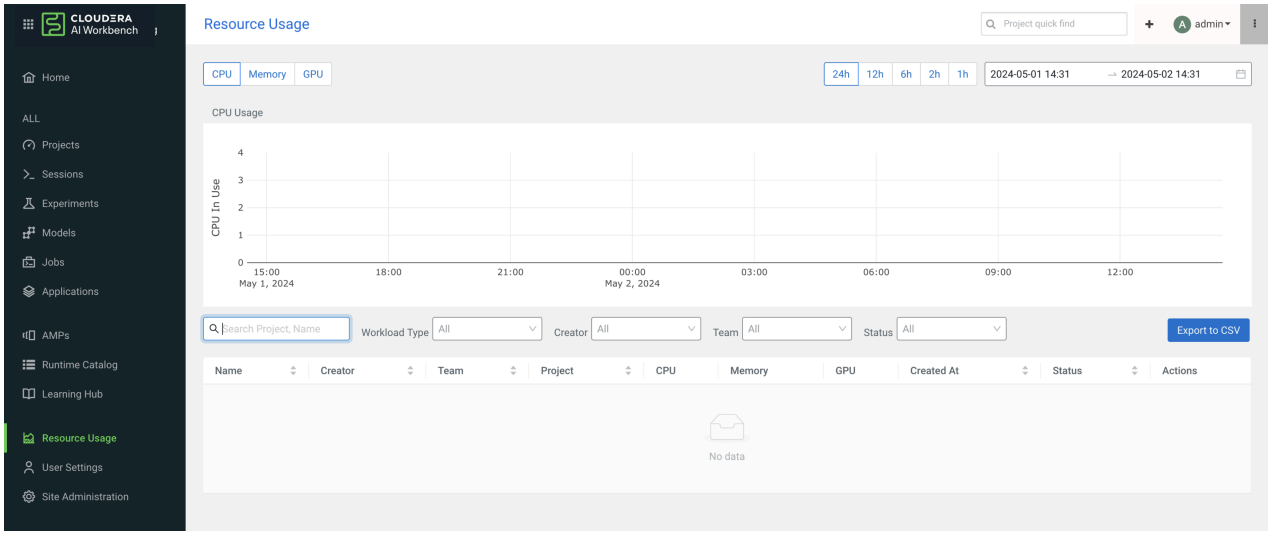
Resource Usage Dashboard

The Resource Usage Dashboard is developed on top of Quota Management to depict the resource usage metrics. The dashboard can only be accessed when Quota Management is enabled.

The Resource Usage Dashboard consists of two parts.

- The Time series chart shows each resource’s usage, based on the filtered time range.
- The Workloads' usage table shows how much resources are utilized by each workload, based on the filtered time range.

Figure 3: Resource Usage Dashboard



The resource options on the top left side (CPU, Memory and GPU) are used to show or filter the selected resource usage, based on a filtered time range.

The date and time range options on the top right side are global filters which are used to filter both the time-series chart and the workloads' usage table.

There is also an export button to download the Workloads' usage table into CSV format.

Limitations for Quota Management

Follow the recommendations on the limitations of Cloudera AI Quota Management.

Note: Do not use the root.default resource pool in the Technical Preview (TP) release to provision the Cloudera AI Workbench. Create at least one resource pool for Cloudera AI.

If the resource pool gets deleted, the underlying Cloudera AI Workbench and workloads will also be deleted. However, the stale entries will still be available in the Cloudera AI UI, reflecting that the workbench still exists but the underlying applications on the cluster are deleted.

If the quota for users or groups is modified, it will be reflected when the next workload (session or job) is submitted.

Yunikorn Gang scheduling

Yunikorn Gang Scheduling is the default scheduling mechanism in Cloudera AI. Yunikorn schedules the workload pods when Quota Management is enabled.

Gang scheduling is a scheduling mechanism in Yunikorn where Yunikorn schedules an application (a set of pods) only when all the resources necessary for bringing up all the pods in the application are available in the cluster. If the resources for bringing up all the pods are not available, the application will not be scheduled, thus helping to use cluster resources efficiently.

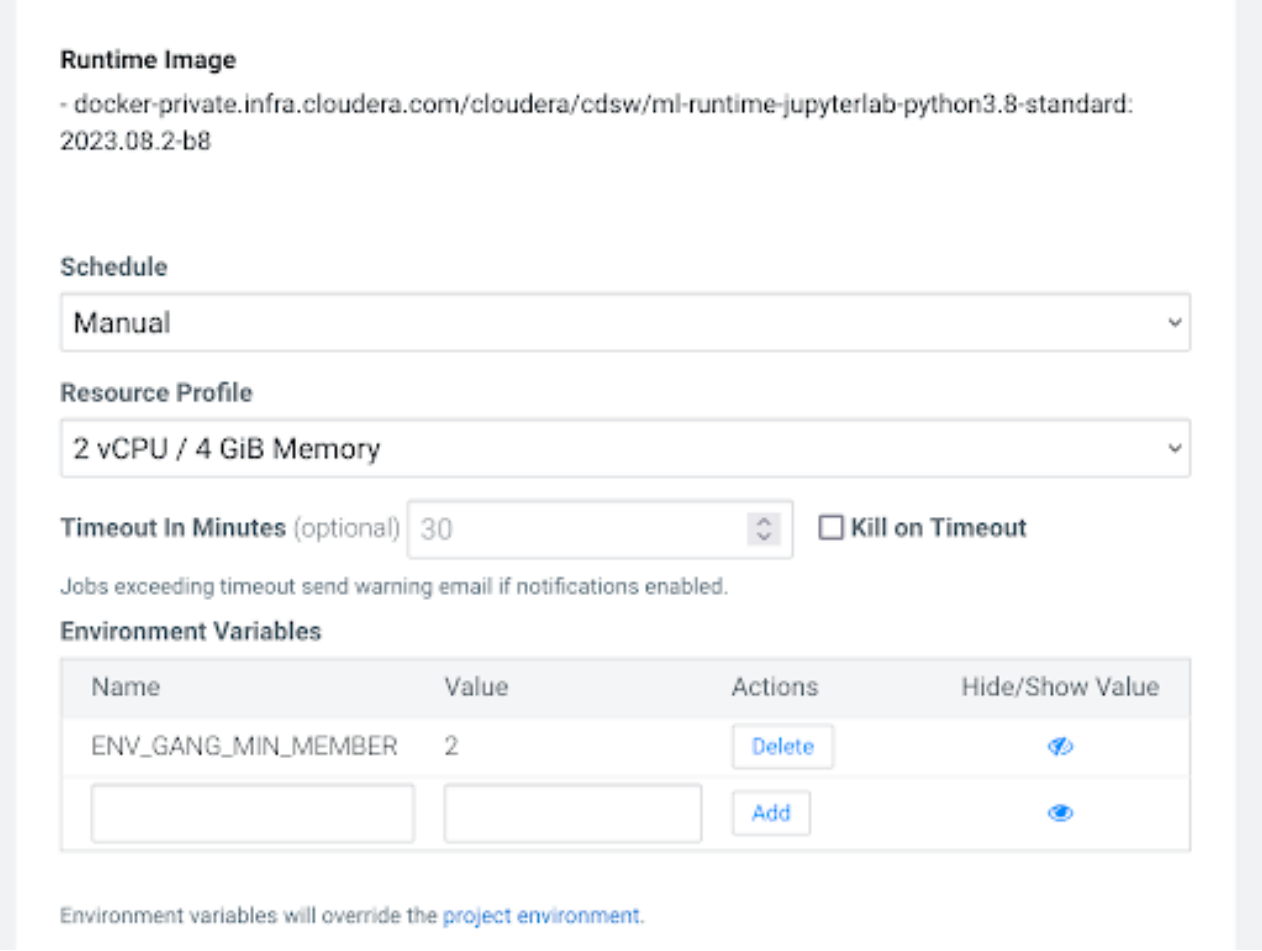
Most workloads need only a single pod and work without any additional configuration in the Cloudera AI. Those workloads however, that take additional worker pods, need additional configuration of Gang parameters to work optimally.

For example, in case there is a Spark job which creates additional executor pods, the driver pod is created in the default setting with the Gang Scheduler and the executor pods use the regular scheduling (that is, the pods will not be scheduled with a Gang Scheduler). However, this is not optimal to run the workloads so. It is important to specify

the expected number of worker pods that the Spark driver pod will schedule. Complete the configuration with the following environment variable:

ENV_GANG_MIN_MEMBER - It sets the number of worker pods apart from the driver pod.

Figure 4: Environment variable



Note:

Although the environment variable can also be configured at project level, it is recommended to configure it at workload level (jobs, models and applications).

Model scheduling

For model workloads, Cloudera AI configures the same number of driver pods as the number of replicas configured for the model. Consequently, the models will not be displayed if the system cannot allocate the resources for all the replicas configured. It is therefore recommended to increase the resource quota or set the replicas properly.

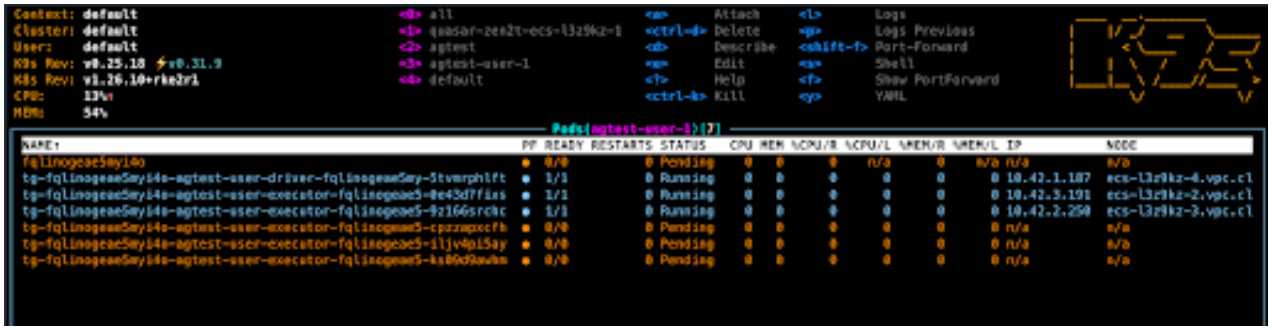
You can override this behavior by using the following environment variable: ENV_DISABLE_GANG_SCHEDULER.

If the value for this variable is set to true for the workload, the Gang Scheduling will be disabled only for that workload. (This is not recommended.)

Placeholder pods

Yunikorn completes the Gang Scheduling work by creating placeholder pods for all the resources needed for the workload. If you configure the ENV_GANG_MIN_MEMBER variable with the value 5 and the system cannot allocate resources for all the 6 placeholder pods (5 workers + 1 driver), the driver pod will remain in PENDING state.

Figure 5: Gang Scheduling pods

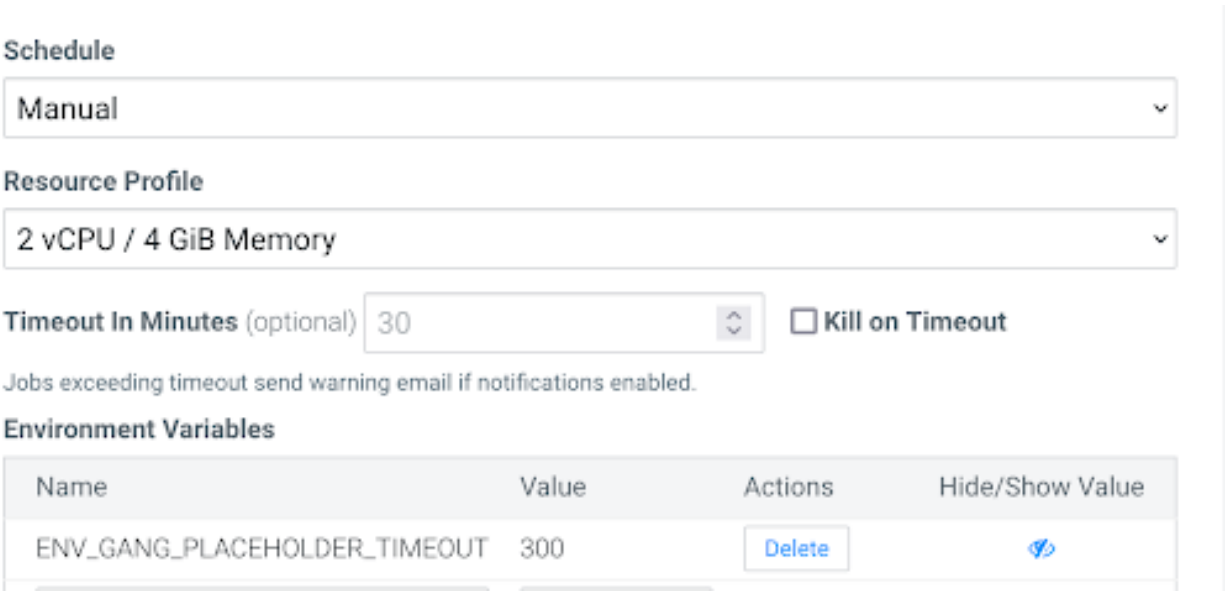


By default, the scheduler keeps placeholder pods in Cloudera AI for 60 seconds. If the scheduler cannot allocate resources within 60 seconds, the workload will be terminated with a FAILED status, and all the pods will be terminated.

Increase the default placeholder timeout value for a workload if needed by configuring the following environment variable:

ENV_GANG_PLACEHOLDER_TIMEOUT - It sets the number of seconds until scheduling.

Figure 6: Environment Gang placeholder



In the example, the placeholder pods will be terminated after 300 seconds.

Configuring resources

By default, the resources for the executors are set similarly as those of the drivers. If the executor resources are different from those of the driver, the values can be configured using the following environment variables:

- Set the executor resources for CPUs with: ENV_GANG_CPU_REQUEST.
- Set the executor resources for memory with: ENV_GANG_MEMORY_REQUEST.
- Set the executor resources for GPUs with: ENV_GANG_GPU_REQUEST.

Only configure the resources that are different from that of the driver. If the CPU and memory capacity needs for the executor are the same as for the driver, only configure the ENV_GANG_GPU_REQUEST variable.

Related Information

Troubleshooting guide for Gang Scheduler