Cloudera Data Engineering 1.5.5

Cloudera Data Engineering Prerequisites

Date published: 2020-07-30
Date modified: 2025-11-08

https://docs.cloudera.com/

Legal Notice

Contents

Prerequisites for Cloudera Data Engineering on premises..... 4

Using GPUs in Cloudera Data Engineering (Technical Preview)..... 4

GPU nodes setup..... 4
Testing GPU setup..... 5
Managing heterogenous GPU nodes..... 5
Spark GPU Runtime Images..... 6
Quota Management..... 6

Integrating a third-party certificate manager..... 9

Prerequisites for Cloudera Data Engineering on premises

Prerequisites for Cloudera Data Engineering on premises.

Before deploying Cloudera Data Engineering, make sure you have reviewed and complied with the requirements in the installation guide for your environment:

- Installing on OpenShift
- Installing using the Cloudera Embedded Container Service

Cloudera Base on premises cluster requirements

The Cloudera Base on premises cluster that you are using for the Cloudera Data Engineering service must have the Apache Ozone service enabled before creating an environment.

Red Hat OpenShift Container Platform requirements

For Cloudera Data Engineering on premises running on Red Hat OpenShift Container Platform (OCP), you must configure a route admission policy.

You must configure the OpenShift cluster for running applications in multiple namespaces with the same domain name. Run the following commands. If you have not installed the oc command line utility, install it using the instructions in the OpenShift documentation. For instructions on downloading the OCP kubeconfig file, see Downloading the kubernetes Configuration.

```
export KUBECONFIG=</path/to/ocp-kubeconfig>

oc -n openshift-ingress-operator patch ingresscontroller/default --patch '{
"spec":{"routeAdmission":{"namespaceOwnership":"InterNamespaceAllowed"}}}' -
-type=merge
```

Related Information

Compatibility for Cloudera Data Engineering and Cloudera Runtime components

Using GPUs in Cloudera Data Engineering (Technical Preview)

A GPU is a specialized processor that can be used to accelerate highly parallelized computationally-intensive workloads. Cloudera Data Engineering leverages the Spark RAPIDS library to accelerate the Spark jobs and sessions using Nvidia GPUs.

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.

With Cloudera on premises1.5.5 version of Cloudera Data Engineering, GPU nodes have been verified to be working with CentOS 7.9 and RHEL 8.8.

GPU nodes setup

You can add the GPU hardware to the existing or new Cloudera Embedded Container Service or OCP cluster as a worker node.

For information about GPU hardware requirements, see *Additional resource requirements for Cloudera Data Engineering*.

You must install `nvidia-container-toolkit` on the worker node. For more on `nvidia-container-runtime` migration to `nvidia-container-toolkit`, see Migration Notice. For information about the installation, NVIDIA Installation Guide. If using Red Hat Enterprise Linux (RHEL), use `dnf` to install the package. For an example with RHEL 8.8, see Installing the NVIDIA Container Toolkit.

You can use following options to advertise the GPUs in the Kubernetes cluster:

- Nvidia device plugin: In Cloudera Embedded Container Service installation, if the Nvidia drivers are correctly installed, the Nvidia-device-plugin automatically advertises the GPU resource to the scheduler. Platform administrator need not deploy the Nvidia device plugin.
- Node Feature Discovery Operator (NFD) and GPU Operator: OpenShift Container Platform administrators must install NFD and GPU Operator for advertising the GPU resource to the Kubernetes scheduler.

If the Nvidia drivers are correctly installed, the above options should advertise the GPU resource to the scheduler. For more information, see *NVIDIA Device Plugin* documentation.

Related Information

Additional resource requirements for Cloudera Data Engineering

Adding hosts to a Cloudera on premises Data Services Cluster

NVIDIA Device Plugin

Testing GPU setup

Before you create a Cloudera Data Engineering Data Service, as a Kubernetes administrator, you must ensure that GPUs are advertised.

You can test if the GPU resources are advertised by running a sample Pod:

```
$ cat <<EOF | kubectl apply -f -
    apiVersion: v1
    kind: Pod
    metadata:
    name: gpu-pod
    spec:
    restartPolicy: Never
    containers:
    - name: cuda-container
    image: nvcr.io/nvidia/k8s/cuda-sample:vectoradd-cuda10.2
    resources:
    limits:
    nvidia.com/gpu: 1 # requesting 1 GPU
    EOF
```

If you get an output similar to the following, it means that the GPU resources are ready for scheduling.

```
// Log Output
        $ kubectl logs gpu-pod
        [Vector addition of 50000 elements]
        Copy input data from the host memory to the CUDA device
        CUDA kernel launch with 196 blocks of 256 threads
        Copy output data from the CUDA device to the host memory
        Test PASSED
        Done
```

Managing heterogenous GPU nodes

If you have heterogeneous GPU nodes and want to run Spark jobs or sessions on a specific GPU node, then the Kubernetes platform administrator must add the node labels and taints.

You can use the below commands to manage labels and taint the node.

- Add Node Label

```
kubectl label nodes worker-node1 nvidia.com/gpu=a100
```

- Remove Node Label

```
kubectl label nodes worker-node1 nvidia.com/gpu-
```

If you want to control running CPU workloads on GPU nodes, it is recommended to set node taint.

- Add Taint

```
kubectl taint nodes worker-node1 nvidia.com/gpu=true:NoSchedule
```

- Remove Taint

```
kubectl taint nodes worker-node1 nvidia.com/gpu=true:NoSchedule-
```

After you add label and taint to the nodes, data engineers can provide node selectors and tolerations during the Spark job submission. For more information about adding node lables and taints, see *Node Labels* and *Taints and Tolerations*.

For information about using labels and taints when creating Cloudera Data Engineering Jobs, see *Creating jobs in Cloudera Data Engineering*.

Related Information

Node Labels

Taints and Tolerations

Creating jobs in Cloudera Data Engineering

Spark GPU Runtime Images

The Cloudera Spark GPU Runtime uses the "nvidia/cuda" option as a base image. The Spark RAPIDS is built against Cloudera Spark distribution and is compatible with the Spark version that Cloudera offers.

The following table provides information about the Spark GPU Runtime images.

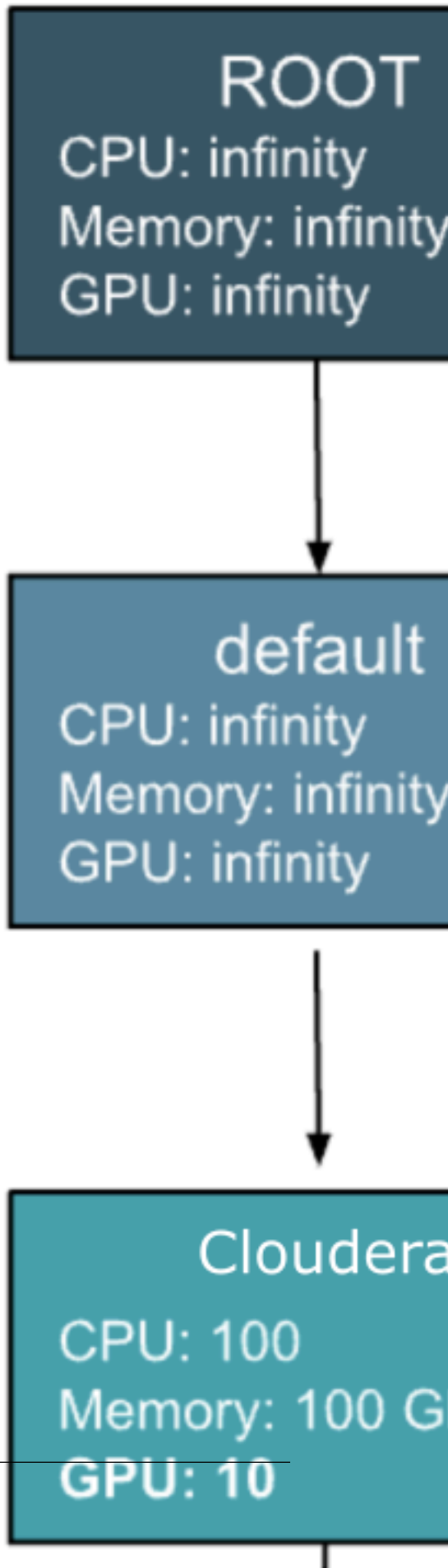
Type	Cloudera Base version	CUDA Version	Base Image	Spark GPU Runtime
Job	7.1.7	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-spark-runtime-gpu-3.2.3-7.1.7.2035:1.20.1-b48
Job	7.1.9	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-spark-runtime-gpu-3.3.2-7.1.9.0:1.20.1-b48
Session	7.1.7	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-livy-runtime-gpu-3.2.3-7.1.7.2035:1.20.1-b48
Session	7.1.9	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-livy-runtime-gpu-3.3.2-7.1.9.0:1.20.1-b48

Quota Management

GPU resources are limited in the cluster and all Data Services, that is, Cloudera AI and Cloudera Data Engineering can share or dedicatedly set resource quota for their experience. Cloudera recommendeds to set the GPU resource quota so that each data service can use the allocated GPU resources effectively. Sharing GPU resources might lead to resource contention and delayed POD allocation.

Cloudera recommendeds to preallocate GPU resources quota for each Data Service using resource pool. For information about configuring resource pool, see *Managing cluster resources using Quota Management*.

Following is an example illustration of GPU resource allocation:


ROOT
CPU: infinity
Memory: infinity
GPU: infinity
default
CPU: infinity
Memory: infinity
GPU: infinity
Cloudera
CPU: 100
Memory: 100 G
GPU: 10

Related Information

Managing cluster resources using Quota Management

Integrating a third-party certificate manager

If a third-party certificate manager with a valid cluster issuer is integrated before the installation of Cloudera Data Services on premises, it allows you to validate the certificates required to initialize the Cloudera Data Engineering Services and Virtual Clusters.

Note:

- Using a third-party certificate manager is available only for new installations performed on Cloudera Embedded Container Service.
- Cloudera supports only Venafi Trust Protection Platform (TPP) as the certificate issuer.
- Venafi TPP must be installed and available before installing Cloudera Data Services on premises.

When you start a Cloudera Data Services on premises service installation, make sure that you have installed a cluster issuer to use third-party certificates. For installing a cluster issuer, see Setting up Certification Manager using Venafi TPP. A valid cluster issuer must have:

- The `issuer.cdp.cloudera.com/type=longlived` annotation
- The following label set:

```
issuer.cdp.cloudera.com/project=<***CDP_NAMESPACE***>
```

You must have a unique valid cluster issuer created. If multiple cluster issuers are found, the Cloudera Data Engineering Service creations fail.

Note:

- You can configure the duration by setting the `issuer.cdp.cloudera.com/duration` annotation in the cluster issuer.
- If the cluster issuer is not in ready state, then the Cloudera Data Engineering Service or the Virtual Cluster creation fails.
- If a Cloudera Data Engineering Service is created using a cluster issuer, and that cluster issuer is not in ready state when the Virtual Cluster is created under the Service, then Virtual Cluster installation fails.