

Cloudera Data Engineering Prerequisites

Date published: 2020-07-30

Date modified: 2024-03-05



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

Prerequisites for Cloudera Data Engineering on CDP Private Cloud.....	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

Prerequisites for Cloudera Data Engineering on CDP Private Cloud

Prerequisites for Cloudera Data Engineering (CDE) on Cloudera Data Platform (CDP) Private Cloud.

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

- [Installing on OpenShift](#)
- [Installing using the Embedded Container Service](#)

CDP Private Cloud Base cluster requirements

The CDP Private Cloud Base 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 CDE Private Cloud 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 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. CDE 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 1.5.3 PVC CDE, 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 ECS or OCP cluster as a worker node.

For information about GPU hardware requirements, see *Additional resource requirements for CDE*.

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 ECS 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: OCP 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 CDP Private Cloud Data Services Cluster](#)

[NVIDIA Device Plugin](#)

Testing GPU setup

Before you create a CDE 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 labels and taints, see *Node Labels* and *Taints and Tolerations*.

For information about using labels and taints when creating CDE 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.

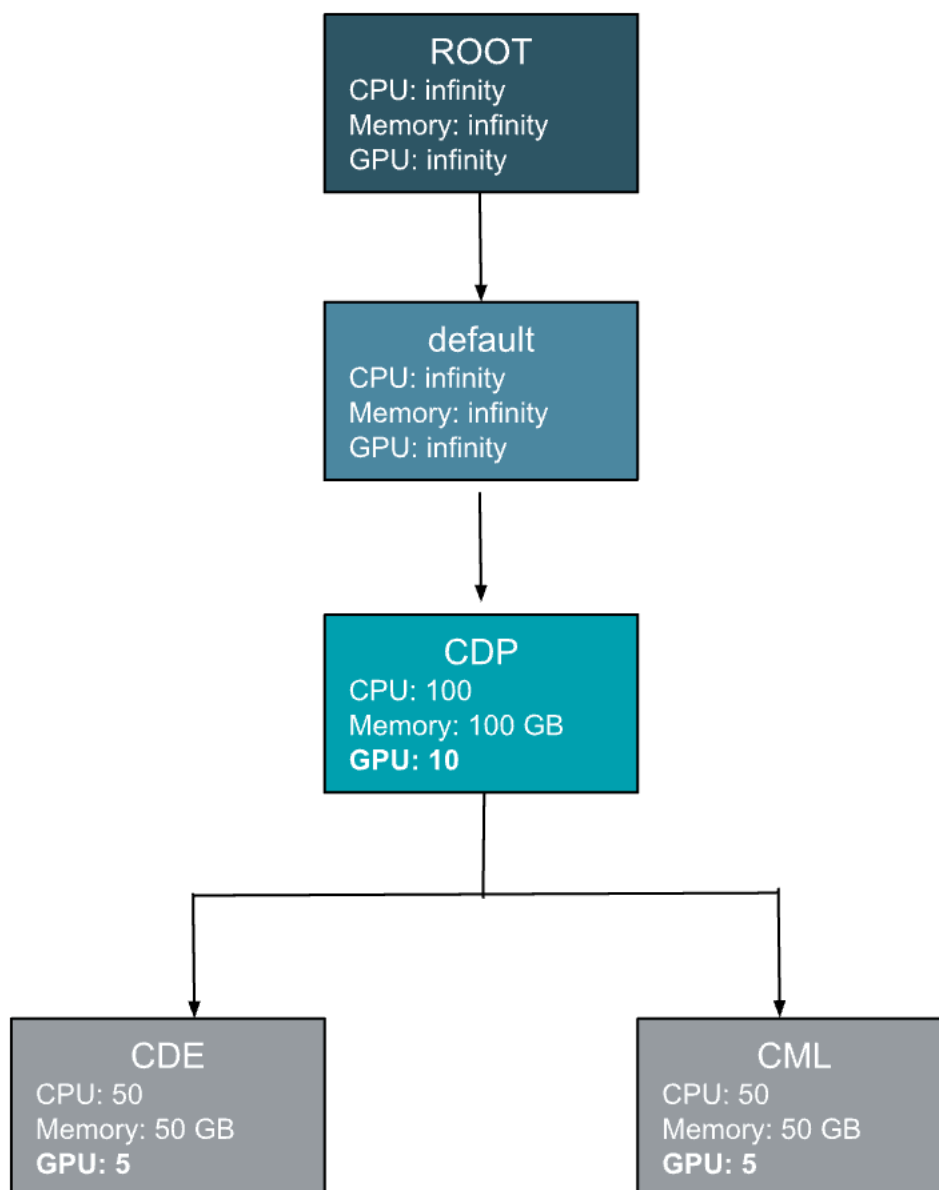
Type	CDP Base version	CUDA Version	Base Image	Spark GPU Runtime
Job	7.1.7	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-spark-runtime-gpu-3.2.3-7.1.7.2035:1.20.1-b48
Job	7.1.9	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	dex-spark-runtime-gpu-3.3.2-7.1.9.0:1.20.1-b48
Session	7.1.7	CUDA 12.1.1	nvidia/cuda:12.1.1-base-ubi8	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, CML and CDE can share or dedicatedly set resource quota for their experience. Cloudera recommends 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 recommends 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:



Related Information

[Managing cluster resources using Quota Management](#)