

CUDA Engine - Technical Preview

Date published: 2020-02-28

Date modified: 2020-12-15



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

CUDA Engine.....	4
Enabling GPUs for CDSW with the CUDA engine.....	4
Set up the CDSW hosts.....	4
Site Admins: Add the CUDA Engine to your Cloudera Data Science Workbench Deployment	
Project Admins: Enable the CUDA Engine for your Project.....	5
Test the CUDA Engine.....	5

CUDA Engine

To make it easier for users to get started with using GPUs on CDSW, the Cloudera engineering team is working on a new custom engine that comes enabled with CUDA out of the box.


Previously, users were expected to [build their own CUDA engine](#).



Important: This engine is still under development and is not recommended for use in production environments.

Compatibility Information

The first version of this engine is built on top of CDSW base engine:13 and ships with CUDA 10.1. It has been tested with CDSW 1.8.x.

Custom Engine	CDSW	NVIDIA Driver
cuda-engine:10 - Based on CDSW engine:13 - Ships CUDA 10.1  Note: CUDA 10.1 is not supported for TensorFlow 2.4 or above.	1.8.x, 1.9.x	418.39 (or higher) NVIDIA/CUDA compatibility matrix

Enabling GPUs for CDSW with the CUDA engine

This section gives you a modified set of steps to be used when you want to use the CUDA engine to enable GPUs for CDSW workloads.

Set up the CDSW hosts

The first few steps of this process are the same as that required for the traditional GPU set up process.

Before you begin

If you have already performed these steps for an existing project that uses GPUs, you can move on to the next section: Site Admins: Add the Custom CUDA Engine to your Cloudera Data Science Workbench Deployment.

Procedure

1. [Set Up the Operating System and Kernel](#)
2. [Install the NVIDIA Driver on GPU Hosts](#) - Make sure you are using a driver that is compatible with the CUDA engine.
3. [Enable GPU Support in Cloudera Data Science Workbench](#)

Site Admins: Add the CUDA Engine to your Cloudera Data Science Workbench Deployment

Before you begin

Required CDSW Role: Site Administrator

About this task

After you've created the CUDA engine, a site administrator must add this new engine to Cloudera Data Science Workbench.

Procedure

1. Sign in to Cloudera Data Science Workbench.
2. Click Admin.
3. Go to the Engines tab.
4. Under Engine Images, add `docker.repository.cloudera.com/cdsw/cuda-engine:10` to the list of images.
5. Click Update.

Airgapped CDSW Deployments: Once these steps have been performed, the CUDA engine will be pulled from Cloudera's public Docker registry. If you have an airgapped deployment, you will also need to manually distribute this image to every CDSW host. For sample steps, see [Distribute the Image](#).

Project Admins: Enable the CUDA Engine for your Project

Project administrators can use the following steps to make it the CUDA engine the default engine used for workloads within a particular project.

Procedure

1. Navigate to your project's Overview page.
2. Click Settings.
3. Go to the Engines tab.
4. Under Engine Image, select the CUDA-capable engine image from the dropdown.

Test the CUDA Engine

You can use the following simple examples to test whether the new CUDA engine is able to leverage GPUs as expected.

Procedure

1. Go to a project that is using the CUDA engine and click New Session.
2. Launch a new session with GPUs.
3. Run the following command in the workbench command prompt to verify that the driver was installed correctly:

```
! /usr/bin/nvidia-smi
```

4. Use any of the following code samples to confirm that the new engine works with common deep learning libraries.

Pytorch

```
!pip3 install torch
from torch import cuda
assert cuda.is_available()
assert cuda.device_count() > 0
print(cuda.get_device_name(cuda.current_device()))
```

Tensorflow

```
!pip3 install tensorflow-gpu==2.0.0
from tensorflow.python.client import device_lib
assert 'GPU' in str(device_lib.list_local_devices())
device_lib.list_local_devices()
```

Keras

```
!pip3 install keras
from keras import backend
```

```
assert len(backend.tensorflow_backend._get_available_gpus()) > 0
print(backend.tensorflow_backend._get_available_gpus())
```