13_CVD: Verify GPU operation - Functional/Load Tests

Execute the following functional and load tests to verify that the GPU is operating correctly.

- GPU Functional Validation – Sample CUDA Application
- GPU Burn Test
- Sample PyTorch script, executed in OpenShift AI - compares CPU vs. GPU times

References:

- Sample GPU Application: https://docs.nvidia.com/datacenter/cloud-native/gpu-operator/latest/getting-started.html#verification-running-sample-gpu-applications
- GPU Burn Test: https://github.com/wilicc/gpu-burn
- Sample PyTorch script, executed in OpenShift AI: see Solution GitHub repo
- PyTorch Examples: https://github.com/pytorch/examples

Verify GPU Operation

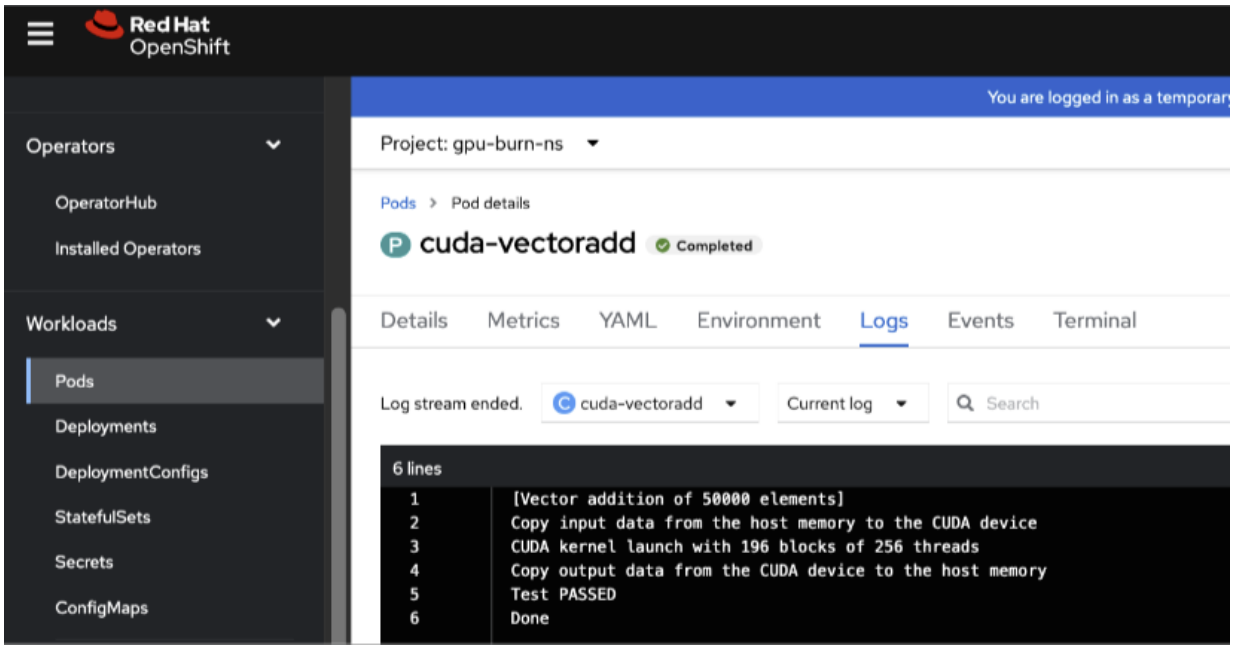
You can use the following tests to verify that GPU is operating correctly.

Sample CUDA Application

```
apiVersion: v1
kind: Pod
metadata:
name: vectoradd
spec:
restartPolicy: OnFailure
containers:
- name: vectoradd
image: nvidia/samples:vectoradd-cuda11.6.0-ubi8
resources:
limits:
nvidia.com/gpu: 1
securityContext:
capabilities:
add: ["SYS_ADMIN"]
```

You can deploy the above configuration on the OpenShift cluster as outlined below:

```
[administrator@FSV-AI-OCP-Installer OCP3]$ vi cuda-vectoradd.yaml
[administrator@FSV-AI-OCP-Installer OCP3]$ oc project
Using project "nvidia-gpu-operator" on server "https://api.ocp3.fsv.local:6443".
[administrator@FSV-AI-OCP-Installer OCP3]$ oc apply -f cuda-vectoradd.yaml
pod/cuda-vectoradd created
[administrator@FSV-AI-OCP-Installer OCP3]$
```



GPU Burn Test - Load/Stress Test

Output from gpu-burn test using CUDA12:

```
==========
== CUDA ==
==========
CUDA Version 12.0.0
Container image Copyright (c) 2016-2023, NVIDIA CORPORATION & AFFILIATES. All rights
reserved.
This container image and its contents are governed by the NVIDIA Deep Learning
Container License.
By pulling and using the container, you accept the terms and conditions of this
license:
https://developer.nvidia.com/ngc/nvidia-deep-learning-container-license
```

```
A copy of this license is made available in this container at /NGC-DL-CONTAINER-
LICENSE for your convenience.
*************************
** DEPRECATION NOTICE! **
*************************
THIS IMAGE IS DEPRECATED and is scheduled for DELETION.
https://gitlab.com/nvidia/container-images/cuda/blob/master/doc/support-policy.md
GPU 0: GRID A100D-40C (UUID: GPU-ef5a53d2-34d3-11b2-99cb-146bdf8cfaed)
Using compare file: compare.ptx
Burning for 60 seconds.
26.7% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:15 UTC 2023
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
30.0% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:22 UTC 2023
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
```

```
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 128 (9171 Gflop/s) errors: 0 temps: --
38.3% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
46.7% proc'd: 256 (18593 Gflop/s) errors: 0 temps: --
51.7% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:30 UTC 2023
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
55.0% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:37 UTC 2023
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
```

```
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 384 (18567 Gflop/s) errors: 0 temps: --
63.3% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
71.7% proc'd: 512 (18536 Gflop/s) errors: 0 temps: --
76.7% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:45 UTC 2023
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
```

```
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
80.0% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:48:52 UTC 2023
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
88.3% proc'd: 640 (18514 Gflop/s) errors: 0 temps: --
90.0% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
96.7% proc'd: 768 (18466 Gflop/s) errors: 0 temps: --
100.0% proc'd: 896 (18449 Gflop/s) errors: 0 temps: --
Summary at: Fri Dec 1 14:49:00 UTC 2023
Killing processes with SIGTERM (soft kill)
Using compare file: compare.ptx
```

```
Burning for 60 seconds.
Initialized device 0 with 40955 MB of memory (37077 MB available, using 33369 MB of
it), using FLOATS
Results are 268435456 bytes each, thus performing 128 iterations
Freed memory for dev 0
Uninitted cublas
done
Tested 1 GPUs:
GPU 0: OK
```


```
[administrator@FSV-AI-OCP-Installer OCP3]$ oc exec -it nvidia-driver-daemonset-413.92.202309261804-0-zshvt -- nvidia-smi
Fri Dec  1 14:54:19 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.60.13    Driver Version: 525.60.13    CUDA Version: 12.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  GRID A100D-40C      On   | 00000000:02:00.0 Off |                    0 |
| N/A   N/A    P0    N/A /  N/A |  34133MiB / 40960MiB |     99%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A    425634      C   ./gpu_burn                       34069MiB |
+-----------------------------------------------------------------------------+
[administrator@FSV-AI-OCP-Installer OCP3]$
```

Sample PyTorch script, executed in OpenShift AI

Execute the following in a Jupyter Notebook in OpenShift AI project workbench. When executed, it will return basic information, compares CPU vs. GPU times.

```
!nvidia-smi
```

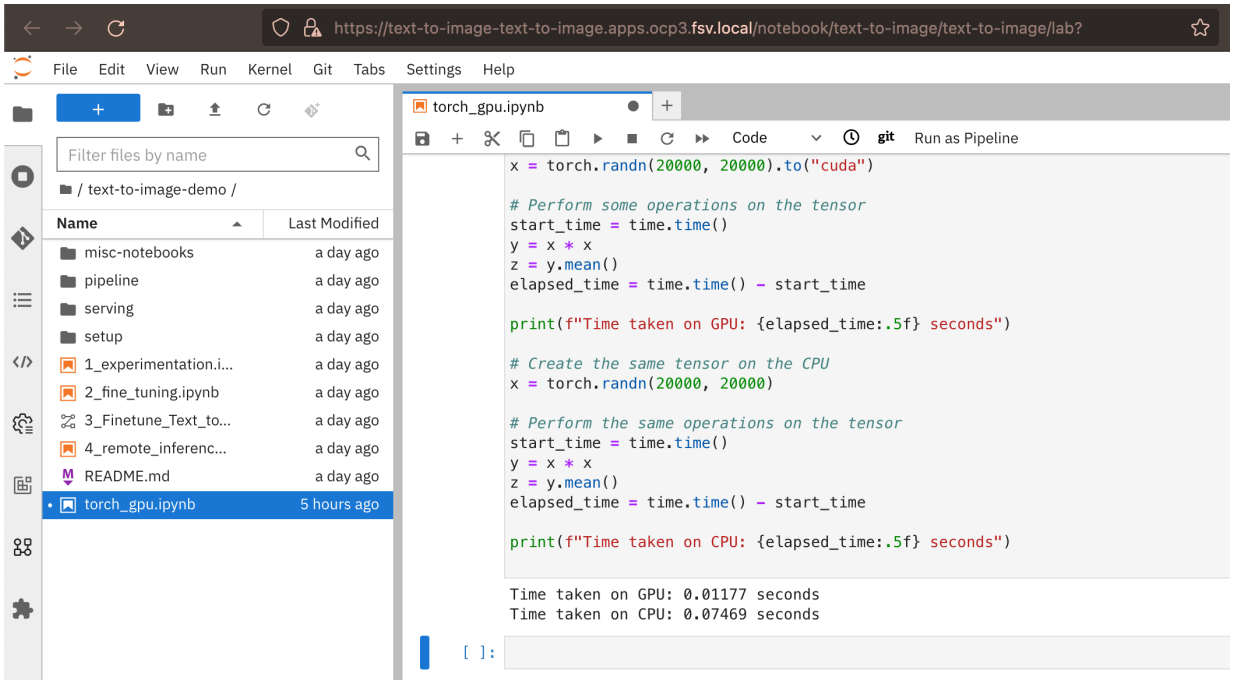
```
import torch

# Check how many GPUs are available
num_of_gpus = torch.cuda.device_count()
print(f"Number of available GPUs: {num_of_gpus}")

# Get a list of all the currently available GPUs
available_gpus = [torch.cuda.device(i) for i in range(num_of_gpus)]
print(f"List of available GPUs: {available_gpus}")
```

```
import torch

# Get a list of all the currently available GPUs
gpu_names = [torch.cuda.get_device_properties(i).name for i in
range(torch.cuda.device_count())]
print(f"List of available GPUs: {gpu_names}")
```

```
import torch

# Check if CUDA is available
if torch.cuda.is_available():
# Set the device to the first available GPU
device = torch.device("cuda:0")
print(f"Using {torch.cuda.get_device_name(device)}")

# Create a tensor on the GPU
x = torch.randn(3, 3).to(device)

# Perform some operations on the tensor
y = x + x

# Move the tensor back to the CPU
z = y.to("cpu")

# Print the result
print(z)
```

```
else:
print("CUDA is not available")
```

```
import torch
import time


if torch.cuda.is_available():
# Create a large tensor on the GPU
x = torch.randn(20000, 20000).to("cuda")

# Perform some operations on the tensor
start_time = time.time()
y = x * x
z = y.mean()
elapsed_time = time.time() - start_time

print(f"Time taken on GPU: {elapsed_time:.5f} seconds")
else:
print("CUDA is not available")
# Create the same tensor on the CPU
x = torch.randn(20000, 20000)

# Perform the same operations on the tensor
start_time = time.time()
y = x * x
z = y.mean()
elapsed_time = time.time() - start_time

print(f"Time taken on CPU: {elapsed_time:.5f} seconds")
```


https://text-to-image-text-to-image.apps.ocp3.fsv.local/notebook/text-to-image/text-to-image/lab?
File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ text-to-image-demo /
Name
Last Modified
misc-notebooks a day ago
pipeline a day ago
serving a day ago
setup a day ago
1_experimentation.i... a day ago
2_fine_tuning.ipynb a day ago
3_Finetune_Text_to... a day ago
4_remote_inferenc... a day ago
README.md a day ago
torch_gpu.ipynb 5 hours ago
torch_gpu.ipynb
Code
git
Run as Pipeline
x = torch.randn(20000, 20000).to("cuda")
# Perform some operations on the tensor
start_time = time.time()
y = x * x
z = y.mean()
elapsed_time = time.time() - start_time
print(f"Time taken on GPU: {elapsed_time:.5f} seconds")
# Create the same tensor on the CPU
x = torch.randn(20000, 20000)
# Perform the same operations on the tensor
start_time = time.time()
y = x * x
z = y.mean()
elapsed_time = time.time() - start_time
print(f"Time taken on CPU: {elapsed_time:.5f} seconds")
Time taken on GPU: 0.01177 seconds
Time taken on CPU: 0.07469 seconds
[ ]: