

Nvidia GPU workload test on SUSE/Rancher stack.

To create Nvidia GPU container based driver review >

https://github.com/alexarnoldy/technical-reference-documentation/blob/nvidia-operator-on-B CI/kubernetes/start/nvidia/adoc/gs_rke2_nvidia-gpu-operator.adoc

or

https://documentation.suse.com/trd/kubernetes/pdf/gs_rke2-slebc_i_nvidia-gpu-operator_en.pdf

In this test example container-based Nvidia GPU driver was created for SLES15 and pushed on the local repo.

DRIVER_VERSION="535.104.05"

OPERATOR_VERSION="v23.9.0"

CUDA_VERSION="12.2.2"

GPU-OPERATOR config has a reference of the local repo.

```
nvidiaDriverCRD:
  deployDefaultCR: true
  driverType: gpu
  enabled: false
  nodeSelector: {}
rdma:
  enabled: false
  useHostMofed: false
repoConfig:
  configMapName: ''
  repository: isv-registry.susealliances.com
  resources: {}
startupProbe:
  failureThreshold: 120
  initialDelaySeconds: 60
  periodSeconds: 10
```

To run a workload test: (need to use a VPN for the cluster access)

Go to

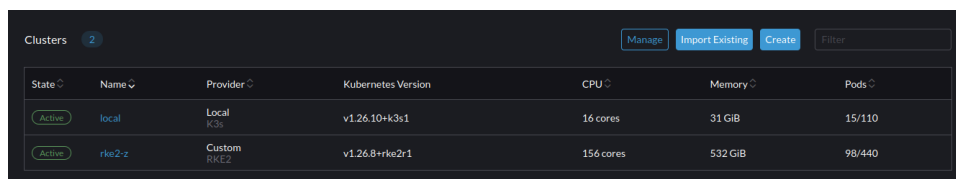
<https://le1.isv.suse> from your browser (need to be on lab VPN and have the following records in your local laptop /etc/hosts file:

192.168.150.16 le1.isv.suse le1

=====

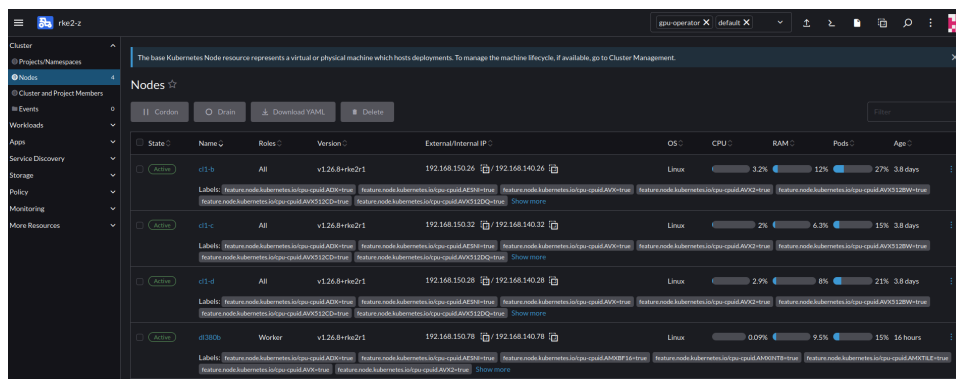
Login to the Rancher server dashboard (admin/Suse_1234567)

Go to the rke2 cluster



State	Name	Provider	Kubernetes Version	CPU	Memory	Pods
Active	local	Local K3s	v1.26.10+k3s1	16 cores	31 GiB	15/110
Active	rke2-z	Custom RKE2	v1.26.8+rke2r1	156 cores	532 GiB	98/440

Check deployed nodes in the cluster and their roles:

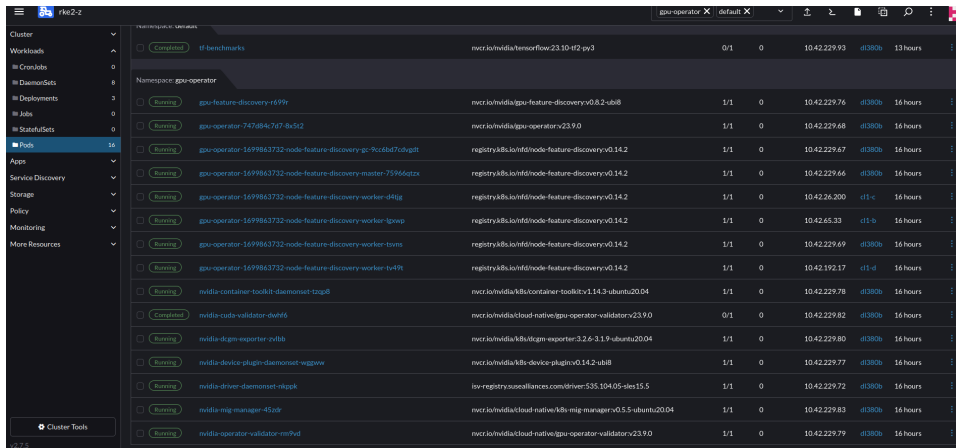


State	Name	Roles	Version	External/Internal IP	OS	CPU	RAM	Pods	Age
Active	c1-b	All	v1.26.8+rke2r1	192.168.150.26 / 192.168.140.26	Linux	3.2%	12%	27%	3.8 days
Active	c1-c	All	v1.26.8+rke2r1	192.168.150.32 / 192.168.140.32	Linux	2%	6.3%	15%	3.8 days
Active	c1-d	All	v1.26.8+rke2r1	192.168.150.28 / 192.168.140.28	Linux	2.9%	8%	21%	3.8 days
Active	d100b	Worker	v1.26.8+rke2r1	192.168.150.78 / 192.168.140.78	Linux	0.0%	9.5%	15%	16 hours

In this RKE2 setup we have 3 SLE15.5 based nodes with all-roles (control-plane, etcd, Master) and 1 SLE15.5 worker node with H100 GPU installed.

Validate that gpu-operator deployed:

Click Workload > Pods



From Kubectl shell execute the following:

```
kubectl exec -it \
"${for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done}" \
-n gpu-operator \
nvidia-smi
```

```
> kubectl exec -it \
> "${for EACH in \
> $(kubectl get pods -n gpu-operator \
> -l app=nvidia-driver-daemonset \
> -o jsonpath={.items..metadata.name}); \
> do echo ${EACH}; done}" \
> -n gpu-operator \
> nvidia-smi
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
Tue Nov 14 00:31:46 2023

+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2   |
+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap       | Memory-Usage  | GPU-Util  Compute M. |
|====+=====+====+=====+=====+=====+=====+=====+
| 0  NVIDIA H100 PCIe                     On          | 00000000:B4:00:0  Off |           0          |
| N/A   49C    P0              50W / 310W | 2MiB / 81559MiB |    0%      Default  |
+-----+-----+-----+-----+-----+-----+
|                                     MIG M. |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                               |
|  GPU   GI   CI        PID   Type   Process name                        GPU Memory |
|  ID   ID                                     Usage                        |
+-----+-----+-----+-----+-----+-----+
| No running processes found              |
+-----+
```

No workload is listed in the above screenshot.

Generate some test workload >

Deploy tf-benchmarks.yaml file with

```
kubectl apply -f tf-benchmarks.yaml
```

from your master node.

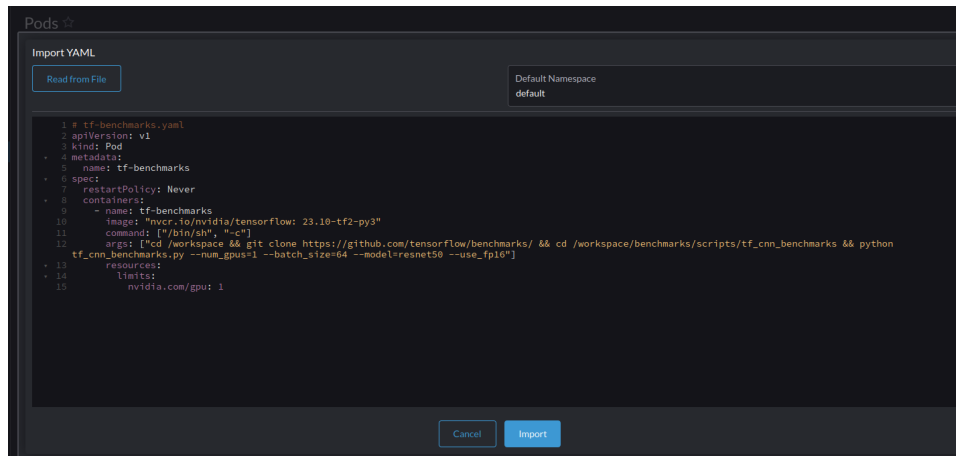
Or from the Rancher Dashboard click <Import Yaml> and paste the following:

=====

```
# tf-benchmarks.yaml
apiVersion: v1
kind: Pod
metadata:
  name: tf-benchmarks
spec:
  restartPolicy: Never
  containers:
  - name: tf-benchmarks
    image: "nvcr.io/nvidia/tensorflow:23.10-tf2-py3"
    command: ["/bin/sh", "-c"]
    args: ["cd /workspace && git clone https://github.com/tensorflow/benchmarks/ && cd
/workspace/benchmarks/scripts/tf_cnn_benchmarks && python tf_cnn_benchmarks.py --num_gpus=1
--batch_size=64 --model=resnet50 --use_fp16"]
    resources:
      limits:
        nvidia.com/gpu: 1
```

===== ref >

<https://developer.nvidia.com/blog/getting-kubernetes-ready-for-the-a100-gpu-with-multi-instance-gpu/> can be used for diff. Nvidia tests including MIG strategy which we used with DGX.



Click import which will create tf-benchmark pod.

While pod is in the training mode, run the same nvidia-smi command to validate the workload:

(You'll have about 10 sec to catch the output)

```
kubectl exec -it \
"${for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done}" \
-n gpu-operator \
nvidia-smi
```

```
kubectl exec -it "${for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done}" -n gpu-operator nvidia-smi
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
Tue Nov 14 00:56:01 2023
```

NVIDIA-SMI 535.104.05 Driver Version: 535.104.05 CUDA Version: 12.2									
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	NVIDIA H100 PCIe	On	00000000:B4:00:0	Off	0				
N/A	55C	P0	208W / 310W	79583MiB / 81559MiB	83%	Default			
						Disabled			

Processes:						
GPU	GI	CI	PID	Type	Process name	GPU Memory
ID	ID	ID				Usage
0	N/A	N/A	90376	C	python	79574MiB

```

kubectrl exec -it "$(for EACH in \
$(kubectrl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done)" -n gpu-operator nvidia-smi
kubectrl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectrl exec [POD] -- [COMMAND] instead.
Tue Nov 14 00:56:02 2023
+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2   |
+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id  Disp.A | Volatile Uncorr. ECC |
| Fan   Temp   Perf              Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+
| 0     NVIDIA H100 PCIe                  On         | 00000000:B4:00.0 Off |                    0 |
| N/A   56C    P0              271W / 310W | 79583MiB / 81559MiB | 92%      Default  |
|                               |              | Disabled |
+-----+-----+
Processes:
+-----+
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
| ID   ID                                     Usage      |
+-----+-----+
| 0     N/A  N/A      90376   C     python                          79574MiB |
+-----+

```

Also, you can check logs from tf-benchmarks pod

kubectrl logs tf-benchmarks

```

~ Kubectrl: rke2-z [X] [v]
t1114 00:55:59.159159 140146836006720 session_manager.py:529] Done running local_init_op.
2023-11-14 00:56:00.091664: I tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:432] Loaded cuDNN version 8905
TensorFlow: 2.13
Model: resnet50
Dataset: imagenet (synthetic)
Mode: training
SingleSess: False
Batch size: 64 global
          64 per device
Num batches: 100
Num epochs: 0.00
Devices: ['/gpu:0']
NUMA bind: False
Data format: NCHW
Optimizer: sgd
Variables: parameter_server
=====
Generating training model
Initializing graph
Running warm up
Done warm up
Step    Img/sec total_loss
1       images/sec: 2256.9 +/- 0.0 (jitter = 0.0)    7.602
10      images/sec: 2504.5 +/- 58.3 (jitter = 18.5)  7.853
20      images/sec: 2575.7 +/- 32.7 (jitter = 8.9)   8.015
30      images/sec: 2602.4 +/- 22.7 (jitter = 8.2)   7.937
40      images/sec: 2615.8 +/- 17.3 (jitter = 9.8)   8.139
50      images/sec: 2623.2 +/- 14.0 (jitter = 8.6)   8.050
60      images/sec: 2628.5 +/- 11.8 (jitter = 8.3)   7.793
70      images/sec: 2626.6 +/- 10.6 (jitter = 8.0)   7.856
80      images/sec: 2628.8 +/- 9.3 (jitter = 7.7)    8.007
90      images/sec: 2632.2 +/- 8.3 (jitter = 8.2)    7.847
100     images/sec: 2633.9 +/- 7.5 (jitter = 8.2)    8.091
-----
total images/sec: 2630.41
-----

```

Software Updates 3:36 PM
You have 62 new updates

Or simply click on pod's View Logs >

The screenshot displays the AWS CloudWatch console interface. At the top, the 'Metrics' tab is selected for the 'gpu-operator' namespace. The 'All metrics' list is expanded, showing a table of metrics for the 'm5.xlarge' instance type. The 'TensorFlow' metric is highlighted, showing a value of 2.13. The 'm5.xlarge' instance is selected, and the 'All metrics' list is expanded, showing various metrics for the 'm5.xlarge' instance type. The 'TensorFlow' metric is highlighted, showing a value of 2.13.

Metric Name	Value
2023-11-13T17:56:05.264514568-87:08 TensorFlow	2.13
2023-11-13T17:56:05.264529333-87:08 Model	resnet50
2023-11-13T17:56:05.264531142-87:08 Dataset	imagenet (synthetic)
2023-11-13T17:56:05.264534466-87:08 Mode	Training
2023-11-13T17:56:05.264539096-87:08 SingleLoss	False
2023-11-13T17:56:05.264536593-87:08 Batch size	64 global
4 per device	
2023-11-13T17:56:05.264539665-87:08 Num batches	100
2023-11-13T17:56:05.264541019-87:08 Num epochs	0.00
2023-11-13T17:56:05.264542579-87:08 Devices	[/gpu0]
2023-11-13T17:56:05.264544183-87:08 Hunk limit	False
2023-11-13T17:56:05.264546449-87:08 Data format	NCM
2023-11-13T17:56:05.264546833-87:08 Optimizer	sgd
2023-11-13T17:56:05.264548186-87:08 Variables	parameter_server
2023-11-13T17:56:05.264549570-87:08 -----	
2023-11-13T17:56:05.264551041-87:08 Generating training model	
2023-11-13T17:56:05.264552638-87:08 Initializing graph	
2023-11-13T17:56:05.264553868-87:08 Running warm up	
2023-11-13T17:56:05.264556288-87:08 Done warm up	
2023-11-13T17:56:05.264556288-87:08 Step	img/sec total_loss
2023-11-13T17:56:05.264558016-87:08 1	images/sec: 2256.9 +/- 0.0 (jitter = 0.0)
2023-11-13T17:56:05.264558016-87:08 18	images/sec: 2264.5 +/- 0.3 (jitter = 18.5)
2023-11-13T17:56:05.264558016-87:08 20	images/sec: 2276.7 +/- 32.7 (jitter = 8.9)
2023-11-13T17:56:05.264559238-87:08 39	images/sec: 2692.4 +/- 22.7 (jitter = 8.2)
2023-11-13T17:56:05.264561009-87:08 49	images/sec: 2615.8 +/- 17.3 (jitter = 8.6)
2023-11-13T17:56:05.264561009-87:08 58	images/sec: 2623.2 +/- 14.0 (jitter = 8.5)
2023-11-13T17:56:05.264566770-87:08 69	images/sec: 2626.5 +/- 11.8 (jitter = 8.3)
2023-11-13T17:56:05.264568161-87:08 78	images/sec: 2626.6 +/- 9.0 (jitter = 8.0)
2023-11-13T17:56:05.264569671-87:08 88	images/sec: 2622.8 +/- 9.3 (jitter = 7.7)
2023-11-13T17:56:05.264571186-87:08 98	images/sec: 2632.2 +/- 9.3 (jitter = 8.2)
2023-11-13T17:56:05.264572593-87:08 100	images/sec: 2633.9 +/- 7.5 (jitter = 8.2)
2023-11-13T17:56:05.264573570-87:08 total	images/sec: 2630.41

If you want to rerun the test, simply re-deploy pod again.

To view GPU metrics modify Prometheus yamls in rancher-monitoring during Rancher-Monitoring installation (already done):

prometheus:

prometheusSpec:

serviceMonitorSelectorNilUsesHelmValues: false

additionalScrapeConfigs:

- job_name: gpu-metrics

scrape_interval: 1s

metrics_path: /metrics

scheme: http

kubernetes_sd_configs:

- role: endpoints

namespaces:

names:

- gpu-operator

relabel_configs:

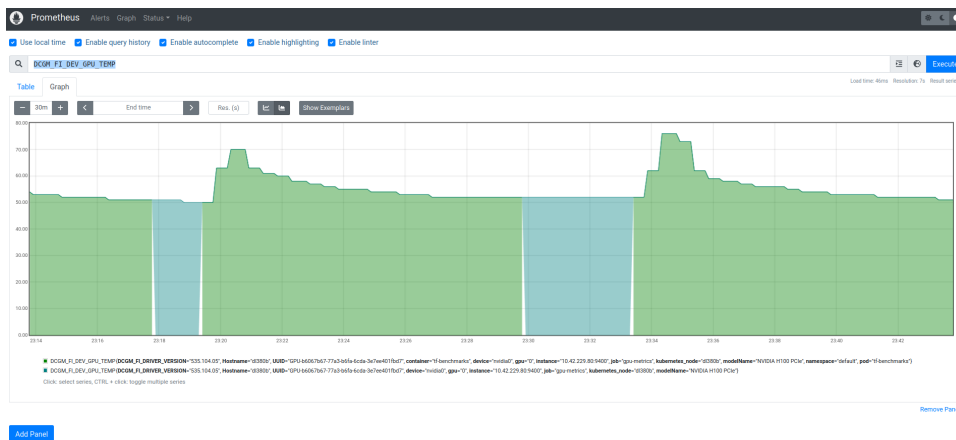
- source_labels: [__meta_kubernetes_pod_node_name]

action: replace

target_label: kubernetes_node

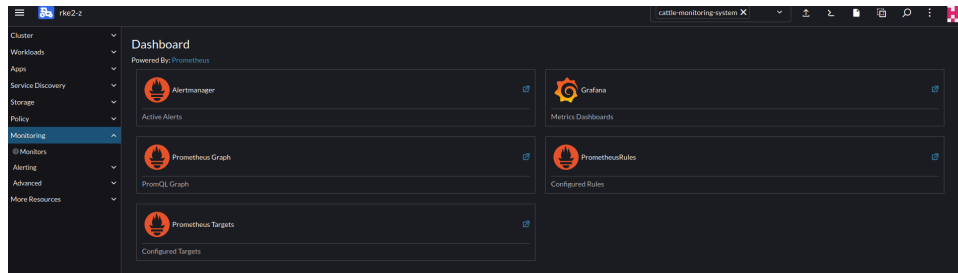
=====

In Prometheus panel enter DCGM_FI_DEV_GPU_TEMP

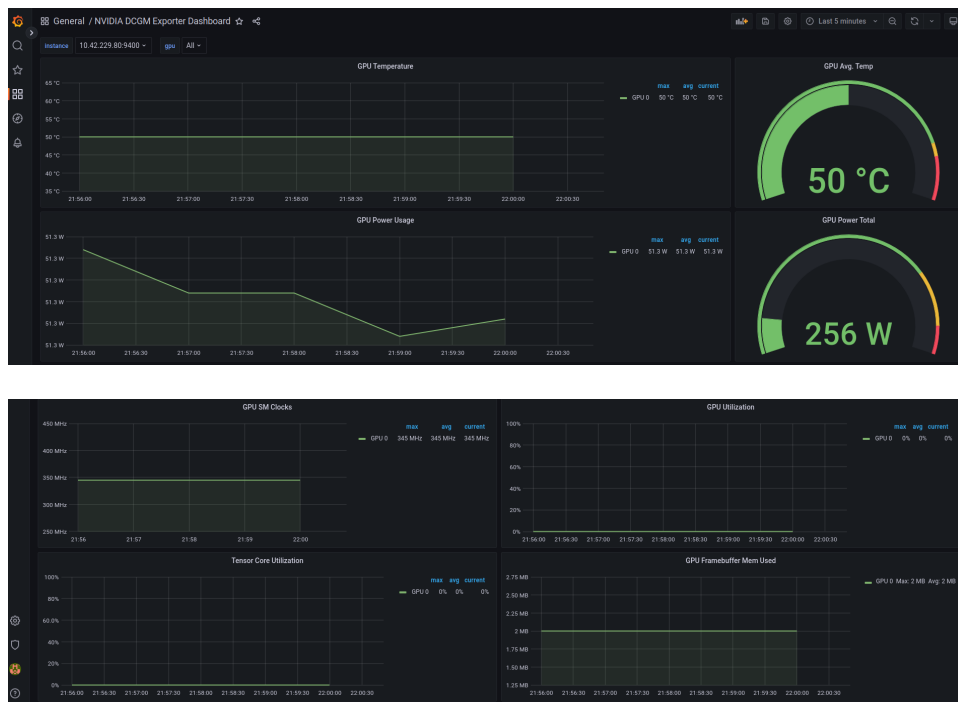


Import NVIDIA DCGM Exporter Dashboard from Grafana (already done)

Open Grafana >



Search for NVIDIA DCGM dashboard





Diff. test values will give you diff numbers.

For ex. changing arguments can increase the GPU utilization:

```
python tf_cnn_benchmarks.py --num_gpus=1 --batch_size=1024 --model=resnet50
--variable_update=parameter_server --use_fp16
```



```
=====
Generating training model
Initializing graph
Running warm up
Done warm up
Step    Img/sec total_loss
1       images/sec: 3738.7 +/- 0.0 (jitter = 0.0)      7.853
10      images/sec: 3733.4 +/- 4.1 (jitter = 21.8)     7.844
20      images/sec: 3734.9 +/- 2.7 (jitter = 16.2)     7.756
30      images/sec: 3734.4 +/- 2.0 (jitter = 14.2)     7.744
40      images/sec: 3732.7 +/- 1.8 (jitter = 14.4)     7.694
50      images/sec: 3732.3 +/- 1.6 (jitter = 13.8)     7.660
60      images/sec: 3730.9 +/- 1.5 (jitter = 13.4)     7.658
70      images/sec: 3730.0 +/- 1.4 (jitter = 13.7)     7.589
80      images/sec: 3727.8 +/- 1.4 (jitter = 15.8)     7.580
90      images/sec: 3725.6 +/- 1.5 (jitter = 15.9)     7.538
100     images/sec: 3723.4 +/- 1.6 (jitter = 18.4)     7.532
-----
total images/sec: 3722.94
-----
```

```
c13-b:/etc/rancher/rke2 # kubectl exec -it "$(for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done)" -n gpu-operator nvidia-smi
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [POD] -- [COMMAND] instead.
Tue Nov 14 06:19:22 2023

+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2   |
+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf              Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|=====-=+==+=====+=====+=====+=====+=====+=====+
|  0   NVIDIA H100 PCIe                  On          | 00000000:B4:00.0 Off |                    0 |
| N/A   66C    P0              304W / 310W | 79583MiB / 81559MiB |    99%    Default  |
|                                           MIG M.         Disabled |
+-----+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                  GPU Memory |
| ID   ID   ID           |                   |           Usage            |
|=====-=+==+=====+=====+=====+=====+=====+=====+
|  0   N/A  N/A       84759    C      python                      79574MiB |
+-----+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                  GPU Memory |
| ID   ID   ID           |                   |           Usage            |
|=====-=+==+=====+=====+=====+=====+=====+=====+
|  0   N/A  N/A       84759    C      python                      79574MiB |
+-----+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                  GPU Memory |
| ID   ID   ID           |                   |           Usage            |
|=====-=+==+=====+=====+=====+=====+=====+=====+
|  0   N/A  N/A       84759    C      python                      79574MiB |
+-----+-----+
+-----+
```

Another example:

```
Import YAML
Read from File
Default Namespace
default

1 # tf-benchmarks.yaml
2 apiVersion: v1
3 kind: Pod
4 metadata:
5   name: tf-benchmarks
6 spec:
7   restartPolicy: Never
8 containers:
9   - name: tf-benchmarks
10    image: "mcr.io/nvidia/tensorflow:23.10-tf2-py3"
11    command: ["python", "-c"]
12    args: ["cd /workspace && git clone https://github.com/tensorflow/benchmarks/ && cd /workspace/benchmarks/scripts/tf_cnn_benchmarks && python tf_cnn_benchmarks.py
13    --num_gpus=1 --batch_size=1024 --model=inception3 --variable_update=parameter_server --use_fp16"]
14 resources:
15   limits:
16     nvidia.com/gpu: 1

Cancel Import
```



```
cli-b:/etc/rancher/rke2 # kubectl exec -it "${for EACH in \
$(kubectl get pods -n gpu-operator \
-l app=nvidia-driver-daemonset \
-o jsonpath={.items..metadata.name}); \
do echo ${EACH}; done}" -n gpu-operator nvidia-smi
Tue Nov 14 06:33:24 2023
+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2               |
+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap       |      Memory-Usage | GPU-Util  Compute M. |
|============================================+==================+|
| 0   NVIDIA H100 PCIe                   On          | 00000000:B4:00:0  Off |                    0 |
| N/A   67C   P0              302W / 310W | 79583MiB / 81559MiB |    100%    Default  |
|                                           MIG M.         |
+-----+-----+
+-----+
| Processes: |
| GPU   GI   CI        PID   Type   Process name                      GPU Memory |
| ID   ID   ID              |                 |           Usage         |
|-----+-----+|
| 0   N/A  N/A       92335    C      python                          79574MiB |
+-----+
```

More tf_cnn_benchmarks tests are available at >

https://github.com/tensorflow/benchmarks/tree/master/scripts/tf_cnn_benchmarks