

# Containers: How do I configure Kubernetes to use NVIDIA GPUs on a Bright 8.0 cluster?

Kubernetes 1.6 allows NVIDIA GPUs to be used from within containers.

However, one GPU cannot be shared among multiple containers. This means that if there are 3 GPUs, then only 3 containers are able to run at a time, with each container assigned one GPU. Other PODs that do not require any GPU resources can still run independently.

## Prerequisites

- You need at least one compute node with an Nvidia GPU;
- You should be running on a Bright 8.0 cluster;
- Your Linux distribution must be supported by Kubernetes.

## Installation

Suppose that your nodes with GPUs are in the category **gpu-cat** and software image **gpu-image**.

Install the cuda package in the software image:

```
yum install --installroot=/cm/images/gpu-image cuda-driver
```

Install kubernetes with **cm-kubernetes-setup**. Select to run PODs in the **gpu-cat** category. At the end of the setup reboot the compute nodes in that category.

```
cmsh -c "device; foreach -c gpu-cat (reboot)"
```

Add a flag to the Kubernetes::Node role:

```
cmsh -c 'category use gpu-cat; roles; use kubernetes::node; set options "--feature-gates=Accelerators=true"; commit'
```

## Example

# Containers: How do I configure Kubernetes to use NVIDIA GPUs on a Bright 8.0 cluster?

You can verify that the GPUs are detected by using `kubectl describe node <my-node>`:

```
kubectl describe node node001
```

under "Capacity" you will see the GPU:

**Capacity:**

```
alpha.kubernetes.io/nvidia-gpu: 1
```

```
cpu: 2
```

Then you can try to create a POD that use that resource:

```
kind: Pod
apiVersion: v1
metadata:
  name: gpu-pod
spec:
  containers:
  - name: gpu-container
    image: gcr.io/tensorflow/tensorflow:latest-gpu
    imagePullPolicy: Always
    command: ["python"]
    args: ["-u", "-c", "import tensorflow"]
    resources:
      requests:
        alpha.kubernetes.io/nvidia-gpu: 1
      limits:
        alpha.kubernetes.io/nvidia-gpu: 1
    volumeMounts:
    - name: bin
      mountPath: /usr/local/nvidia/bin
    - name: lib
      mountPath: /usr/local/nvidia/lib
  restartPolicy: Never
  volumes:
  - name: bin
    hostPath:
      path: /cm/local/apps/cuda-driver/libs/current/bin
  - name: lib
    hostPath:
      path: /cm/local/apps/cuda-driver/libs/current/lib64
```

# Containers: How do I configure Kubernetes to use NVIDIA GPUs on a Bright 8.0 cluster?

The idea is to mount into the container the cuda driver and binaries installed in the host. The container image we are using here comes from Google and contains TensorFlow. In the "resources" section we require an Nvidia GPU to be used, so that the POD will be scheduled wherever one is available. This specific image includes the mounted paths in the **\$PATH** and **\$LD\_LIBRARY\_PATH** environment variables, so the **tensorflow** python module will be able to access them.

Create it with:

```
module load kubernetes
```

```
kubectl create -f gpu-pod.yaml
```

You can verify that everything went well by looking at the pods:

```
watch kubectl get pods --show-all
```

If the pod terminates successfully, then the cluster is ready to go. Please refer to the "Bright Machine Learning manual" for more examples. You will be able to run them inside a container managed by Kubernetes.

Unique solution ID: #1356

Author: Michele Bertasi

Last update: 2017-07-26 09:40