Preventing unprivileged access to GPUs in Kubernetes

Kevin Klues <kklues@nvidia.com>

Last Updated:
04-Sep-2020

Table of Contents

Overview

This document walks through the steps necessary to take advantage of the features described in the following document:

> Read list of GPU devices from volume mounts instead of NVIDIA_VISIBLE_DEVICES

This feature prevents users from exploiting the use of **`NVIDIA_VISIBLE_DEVICES`** to bypass the **`k8s-device-plugin`** when requesting access to GPUs in Kubernetes.

This document assumes you already have a GPU capable Kubernetes cluster up and running. Instructions for doing so can be found across the documents below:

> Install NVIDIA Container Toolkit
> Install Kubernetes
> Install NVIDIA Device Plugin

Configuring the `nvidia-container-toolkit`

First, install version v**`1.3.0-rc.2+`** of the **`nvidia-container-toolkit`**

For **`apt-get`**:

Add the repo:

```
DISTRIBUTION=$(. /etc/os-release;echo $ID$VERSION_ID)
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | \
  sudo apt-key add -
curl -s -L \
  https://nvidia.github.io/nvidia-docker/${DISTRIBUTION}/nvidia-docker.list | \
  sudo tee /etc/apt/sources.list.d/nvidia-docker.list
```

Enable access to the experimental repos:

```
sudo sed -i -e '/experimental/ s/^#//g' \
  /etc/apt/sources.list.d/nvidia-docker.list
sudo apt-get update
```

Install the latest version of **nvidia-container-toolkit**

```
sudo apt-get update
sudo apt-get install -y nvidia-container-toolkit
```

For **yum**:

Add the repo:

```
DISTRIBUTION=$(. /etc/os-release;echo $ID$VERSION_ID)
curl -s -L
https://nvidia.github.io/nvidia-docker/$DISTRIBUTION/nvidia-docker.repo | \
  sudo tee /etc/yum.repos.d/nvidia-docker.repo
```

Enable access to the experimental repos:

```
sudo yum-config-manager --enable libnvidia-container-experimental
sudo yum-config-manager --enable nvidia-container-runtime-experimental
```

Install the latest version of **nvidia-container-toolkit**

```
sudo yum install -y nvidia-container-toolkit
```

Once installed, open the file **/etc/nvidia-container-runtime/config.toml**, uncomment the lines in red below, and set them to the values shown:

```
disable-require = false
#swarm-resource = "DOCKER_RESOURCE_GPU"
accept-nvidia-visible-devices-envvar-when-unprivileged = false
accept-nvidia-visible-devices-as-volume-mounts = true

[nvidia-container-cli]
#root = "/run/nvidia/driver"
#path = "/usr/bin/nvidia-container-cli"
environment = []
#debug = "/var/log/nvidia-container-toolkit.log"
#ldcache = "/etc/ld.so.cache"
load-kmods = true
#no-cgroups = false
#user = "root:video"
ldconfig = "@/sbin/ldconfig.real"
```

```
[nvidia-container-runtime]
#debug = "/var/log/nvidia-container-runtime.log"
```

Configuring the k8s-device-plugin

First, install version **v0.7.0-rc.7+** of the **k8s-device-plugin**

1) Add the **nvidia-device-plugin** repository:

```
$ helm repo add nvdp https://nvidia.github.io/k8s-device-plugin
$ helm repo update
```

2) Verify that the **v0.7.0-rc.7** version of the plugin is available:

Note: Since this is a pre-release, you need to pass the **--devel** flag to **helm search repo** in order to see the release listed.

```
$ helm search repo nvdp --devel
NAME                          CHART VERSION  APP VERSION      DESCRIPTION
nvdp/nvidia-device-plugin     0.7.0-rc.7     0.7.0-rc.7       A Helm chart for ...
```

3) Deploy the **k8s-device-plugin** with the following settings:

```
$ helm install \
    --version=0.7.0-rc.7 \
    --generate-name \
    --set securityContext.privileged=true \
    --set deviceListStrategy=volume-mounts \
    nvdp/nvidia-device-plugin
```

Testing in docker

1) Setting **NVIDIA_VISIBLE_DEVICES** as an unprivileged user

```
$ docker run \
    nvidia/cuda:9.0-base \
    nvidia-smi -L
docker: Error response from daemon: OCI runtime create failed:
container_linux.go:349: starting container process caused "process_linux.go:449:
container init caused \"process_linux.go:432: running prestart hook 1 caused
\\\"error running hook: exit status 1, stdout: , stderr: insufficient privileges to
read device list from NVIDIA_VISIBLE_DEVICES envvar\\\n\\\"\"": unknown.
ERRO[0001] error waiting for container: context canceled
```

2) Setting **NVIDIA_VISIBLE_DEVICES** as a privileged user

```
$ docker run \
    --cap-add SYS_ADMIN \
    nvidia/cuda:9.0-base \
    nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-f22fb098-d1b3-3806-2655-ba25f02229c1)
GPU 2: Tesla V100-SXM2-16GB-N (UUID: GPU-f613f823-1032-b3ec-a876-50f2e35e6f9e)
GPU 3: Tesla V100-SXM2-16GB-N (UUID: GPU-3109fa37-4445-73c7-b695-1b5a4d13f58e)
GPU 4: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)
GPU 5: Tesla V100-SXM2-16GB-N (UUID: GPU-a27fb382-bad2-c02a-95ba-f6a1da38e76c)
GPU 6: Tesla V100-SXM2-16GB-N (UUID: GPU-f5bb8d07-ee19-1787-4d9a-a84c4ac6b086)
GPU 7: Tesla V100-SXM2-16GB-N (UUID: GPU-1ba0ca0e-6d1d-d9db-07d8-c1c5a8c32814)
```

3) Setting the device list via volume mounts as an unprivileged user

```
$ docker run \
    -v /dev/null:/var/run/nvidia-container-devices/GPU-edfee158-11c1-52b8-0517-92f30e7fac88 \
    -v /dev/null:/var/run/nvidia-container-devices/GPU-e28a6529-288c-7ddf-8fea-68c4833cda70 \
    nvidia/cuda:9.0-base nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)

$ docker run \
    -v /dev/null:/var/run/nvidia-container-devices/0 \
    -v /dev/null:/var/run/nvidia-container-devices/4 \
    nvidia/cuda:9.0-base nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)
```

4) Setting the device list via volume mounts as a privileged user

```
$ docker run \
    --cap-add SYS_ADMIN \
    -v /dev/null:/var/run/nvidia-container-devices/GPU-edfee158-11c1-52b8-0517-92f30e7fac88 \
    -v /dev/null:/var/run/nvidia-container-devices/GPU-e28a6529-288c-7ddf-8fea-68c4833cda70 \
    nvidia/cuda:9.0-base nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)

$ docker run \
    --cap-add SYS_ADMIN \
    -v /dev/null:/var/run/nvidia-container-devices/0 \
    -v /dev/null:/var/run/nvidia-container-devices/4 \
    nvidia/cuda:9.0-base nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)
```

Testing in kubernetes

1) Running an unprivileged pod requesting GPUs with **NVIDIA_VISIBLE_DEVICES** unset

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --limits=nvidia.com/gpu=2 \
    test-pod -- nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-3109fa37-4445-73c7-b695-1b5a4d13f58e)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)

$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --limits=nvidia.com/gpu=2 \
    test-pod -- bash -c 'export'
declare -x CUDA_PKG_VERSION="9-0=9.0.176-1"
declare -x CUDA_VERSION="9.0.176"
declare -x HOME="/root"
declare -x HOSTNAME="test-pod"
declare -x KUBERNETES_PORT="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP_ADDR="10.96.0.1"
declare -x KUBERNETES_PORT_443_TCP_PORT="443"
declare -x KUBERNETES_PORT_443_TCP_PROTO="tcp"
declare -x KUBERNETES_SERVICE_HOST="10.96.0.1"
declare -x KUBERNETES_SERVICE_PORT="443"
declare -x KUBERNETES_SERVICE_PORT_HTTPS="443"
declare -x LD_LIBRARY_PATH="/usr/local/nvidia/lib:/usr/local/nvidia/lib64"
declare -x NVIDIA_DRIVER_CAPABILITIES="compute,utility"
declare -x NVIDIA_REQUIRE_CUDA="cuda>=9.0 "
declare -x NVIDIA_VISIBLE_DEVICES="/var/run/nvidia-container-devices"
declare -x OLDPWD
declare -x
PATH="/usr/local/nvidia/bin:/usr/local/cuda/bin:/usr/local/sbin:/usr/local/bin:/usr/
sbin:/usr/bin:/sbin:/bin"
declare -x PWD="/"
declare -x SHLVL="1"
declare -x TERM="xterm
```

2) Running an unprivileged pod not requesting GPUs with **NVIDIA_VISIBLE_DEVICES** unset

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    test-pod -- nvidia-smi -L
pod default/test-pod terminated (StartError)
failed to create containerd task: OCI runtime create failed: container_linux.go:349:
starting container process caused "process_linux.go:449: container init caused
\"process_linux.go:432: running prestart hook 0 caused \\\"error running hook: exit
status 1, stdout: , stderr: insufficient privileges to read device list from
NVIDIA_VISIBLE_DEVICES envvar\\\\n\\\"\"": unknown
```

3) Running an unprivileged pod requesting GPUs with **NVIDIA_VISIBLE_DEVICES** set

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --env NVIDIA_VISIBLE_DEVICES=all \
    --limits=nvidia.com/gpu=2 \
```

```
    test-pod -- nvidia-smi -L
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-3109fa37-4445-73c7-b695-1b5a4d13f58e)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)

$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --env NVIDIA_VISIBLE_DEVICES=all \
    --limits=nvidia.com/gpu=2 \
    test-pod -- bash -c 'export'
declare -x CUDA_PKG_VERSION="9-0=9.0.176-1"
declare -x CUDA_VERSION="9.0.176"
declare -x HOME="/root"
declare -x HOSTNAME="test-pod"
declare -x KUBERNETES_PORT="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP_ADDR="10.96.0.1"
declare -x KUBERNETES_PORT_443_TCP_PORT="443"
declare -x KUBERNETES_PORT_443_TCP_PROTO="tcp"
declare -x KUBERNETES_SERVICE_HOST="10.96.0.1"
declare -x KUBERNETES_SERVICE_PORT="443"
declare -x KUBERNETES_SERVICE_PORT_HTTPS="443"
declare -x LD_LIBRARY_PATH="/usr/local/nvidia/lib:/usr/local/nvidia/lib64"
declare -x NVIDIA_DRIVER_CAPABILITIES="compute,utility"
declare -x NVIDIA_REQUIRE_CUDA="cuda>=9.0 "
declare -x NVIDIA_VISIBLE_DEVICES="all"
declare -x OLDPWD
declare -x
PATH="/usr/local/nvidia/bin:/usr/local/cuda/bin:/usr/local/sbin:/usr/local/bin:/usr/
sbin:/usr/bin:/sbin:/bin"
declare -x PWD="/"
declare -x SHLVL="1"
declare -x TERM="xterm
```

4) Running an unprivileged pod not requesting GPUs with **NVIDIA_VISIBLE_DEVICES** set

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --env NVIDIA_VISIBLE_DEVICES=0,4 \
    test-pod -- nvidia-smi -L
pod default/test-pod terminated (StartError)
failed to create containerd task: OCI runtime create failed: container_linux.go:349:
starting container process caused "process_linux.go:449: container init caused
\"process_linux.go:432: running prestart hook 0 caused \\\"error running hook: exit
status 1, stdout: , stderr: insufficient privileges to read device list from
NVIDIA_VISIBLE_DEVICES envvar\\\\n\\\"\"": unknown
```

5) Running an unprivileged pod not requesting GPUs with **NVIDIA_VISIBLE_DEVICES** nulled

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --env NVIDIA_VISIBLE_DEVICES="" \
    --restart=Never \
    test-pod -- nvidia-smi -L
failed to create containerd task: OCI runtime create failed: container_linux.go:349:
starting container process caused "exec: \"nvidia-smi\": executable file not found
```

```
in $PATH": unknown

$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --env NVIDIA_VISIBLE_DEVICES="" \
    --restart=Never \
    test-pod -- bash -c 'export'
declare -x CUDA_PKG_VERSION="9-0=9.0.176-1"
declare -x CUDA_VERSION="9.0.176"
declare -x HOME="/root"
declare -x HOSTNAME="test-pod"
declare -x KUBERNETES_PORT="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP="tcp://10.96.0.1:443"
declare -x KUBERNETES_PORT_443_TCP_ADDR="10.96.0.1"
declare -x KUBERNETES_PORT_443_TCP_PORT="443"
declare -x KUBERNETES_PORT_443_TCP_PROTO="tcp"
declare -x KUBERNETES_SERVICE_HOST="10.96.0.1"
declare -x KUBERNETES_SERVICE_PORT="443"
declare -x KUBERNETES_SERVICE_PORT_HTTPS="443"
declare -x LD_LIBRARY_PATH="/usr/local/nvidia/lib:/usr/local/nvidia/lib64"
declare -x NVIDIA_DRIVER_CAPABILITIES="compute,utility"
declare -x NVIDIA_REQUIRE_CUDA="cuda>=9.0 "
declare -x NVIDIA_VISIBLE_DEVICES=""
declare -x OLDPWD
declare -x
PATH="/usr/local/nvidia/bin:/usr/local/cuda/bin:/usr/local/sbin:/usr/local/bin:/usr/
sbin:/usr/bin:/sbin:/bin"
declare -x PWD="/"
declare -x SHLVL="1"
declare -x TERM="xterm"
```

6) Running a privileged pod not requesting GPUs with **NVIDIA_VISIBLE_DEVICES** set

```
$ kubectl run -it --rm \
    --image=nvidia/cuda:9.0-base \
    --restart=Never \
    --overrides='{
      "spec": {
        "containers": [{
          "name": "test-pod",
          "image": "nvidia/cuda:9.0-base",
          "env": [{
            "name": "NVIDIA_VISIBLE_DEVICES",
            "value": "0,4"
          }],
          "securityContext": {
            "capabilities": {
              "add": ["SYS_ADMIN"]
            }
          },
          "command": ["nvidia-smi", "-L"]
        }]
      }
    }' \
    test-pod
GPU 0: Tesla V100-SXM2-16GB-N (UUID: GPU-edfee158-11c1-52b8-0517-92f30e7fac88)
GPU 1: Tesla V100-SXM2-16GB-N (UUID: GPU-e28a6529-288c-7ddf-8fea-68c4833cda70)
```