KubeCon | CloudNativeCon
Europe 2021
Virtual

MULTI-INSTANCE GPUs IN CONTAINERS AND KUBERNETES

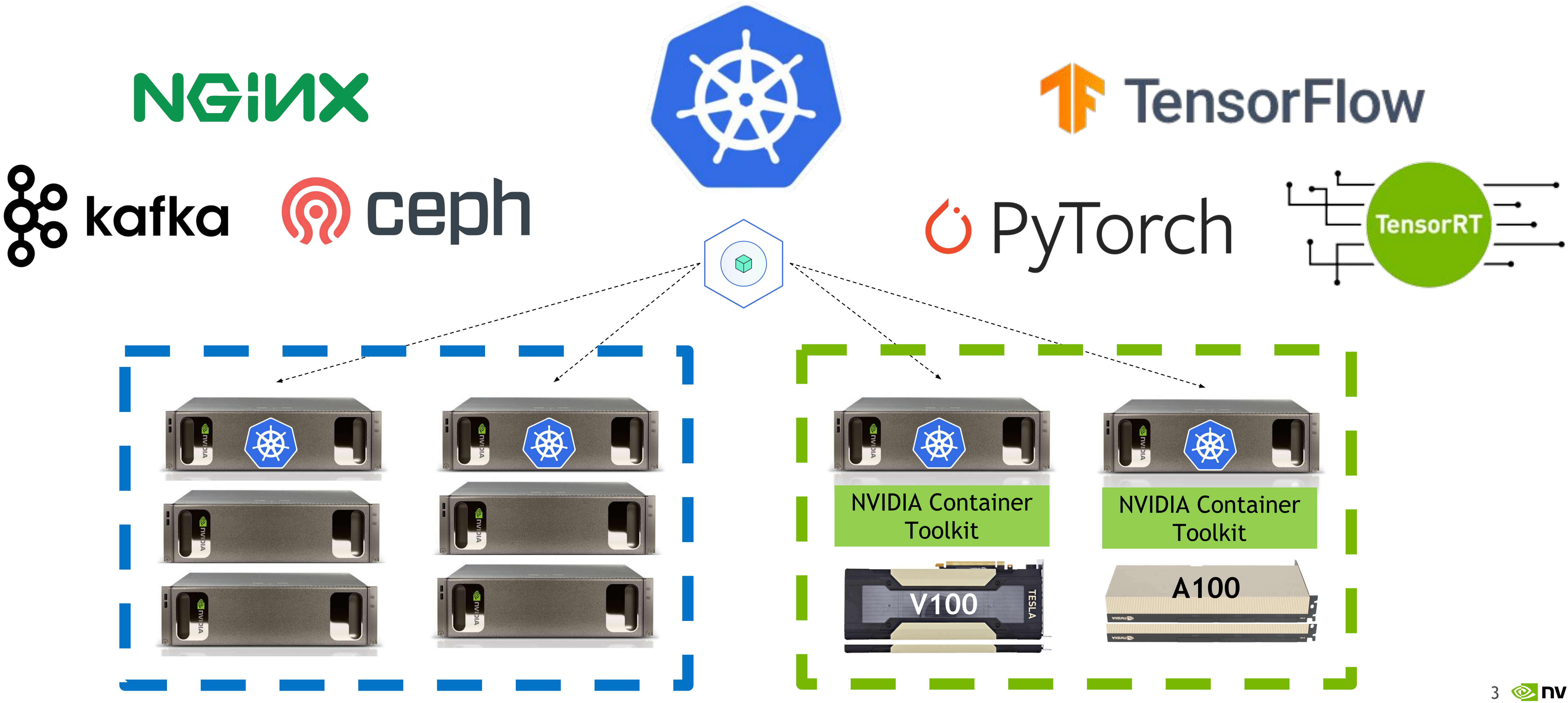
Kevin Klues
kklues@nvidia.com

GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines

GPUs AND KUBERNETES

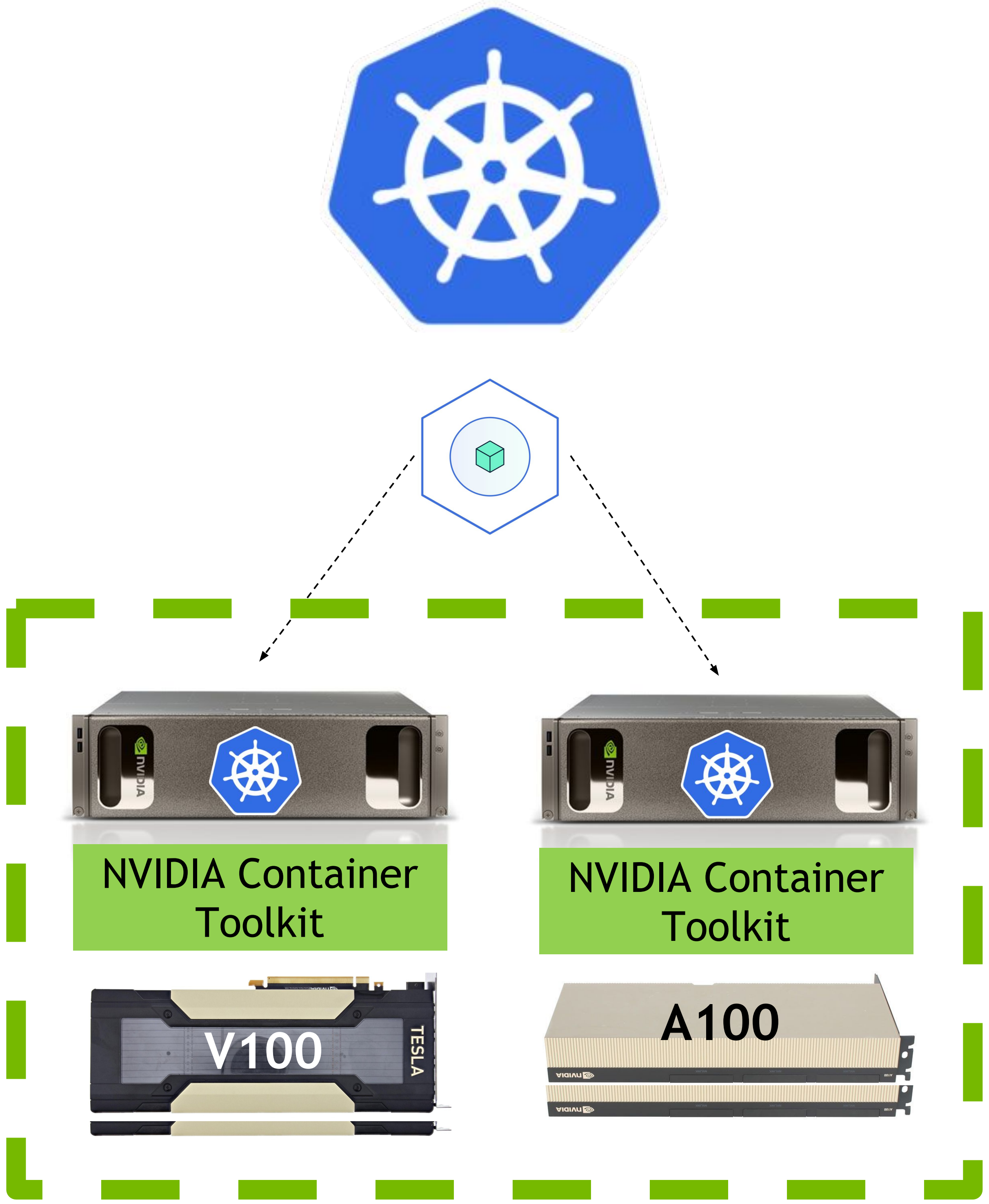
Seamlessly scale up training and inference to a cluster of GPU machines



GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines

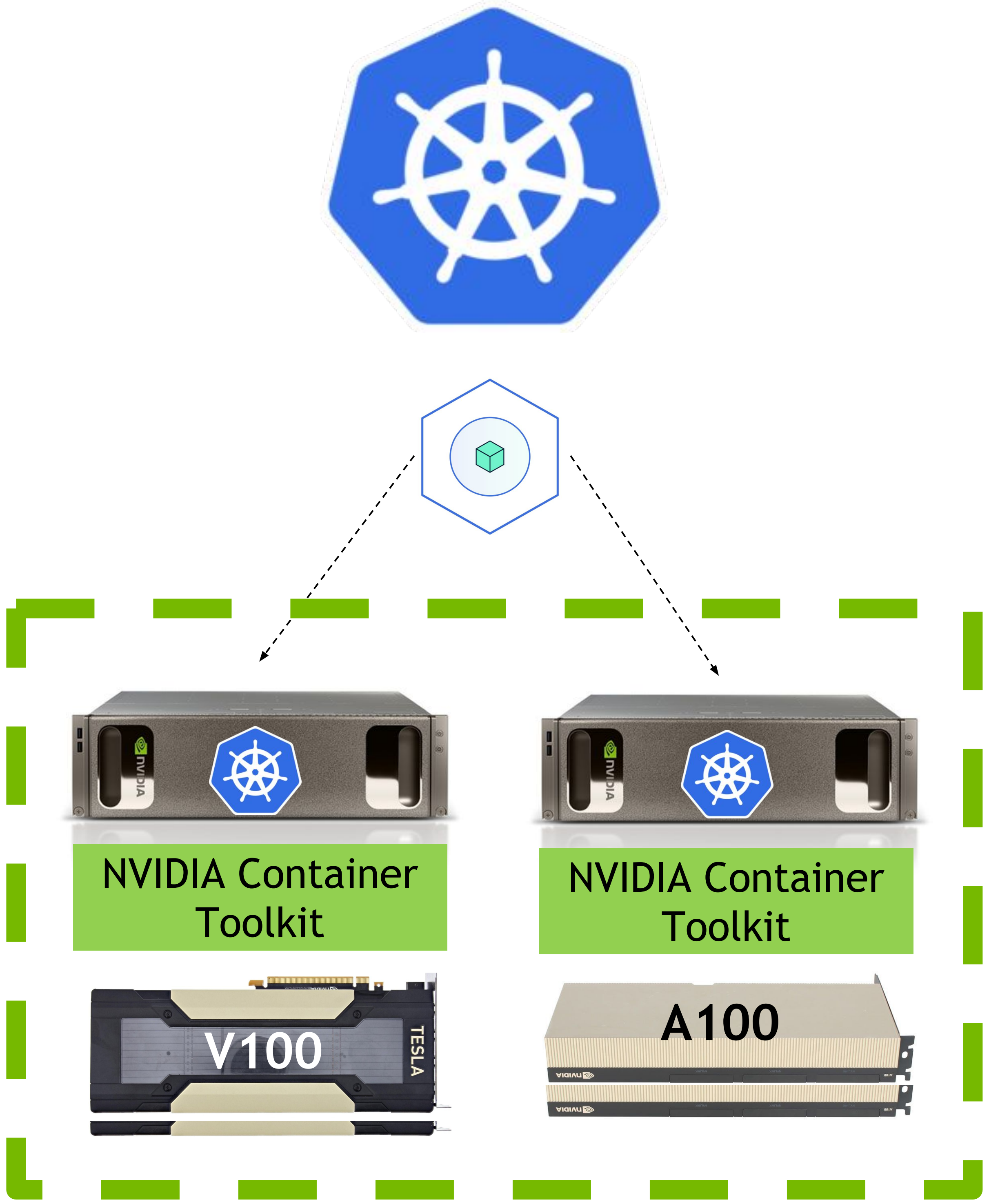
```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 4
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
    nvidia.com/cuda.runtime: 11.0
    nvidia.com/cuda.driver: 450.51.06
```



GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines

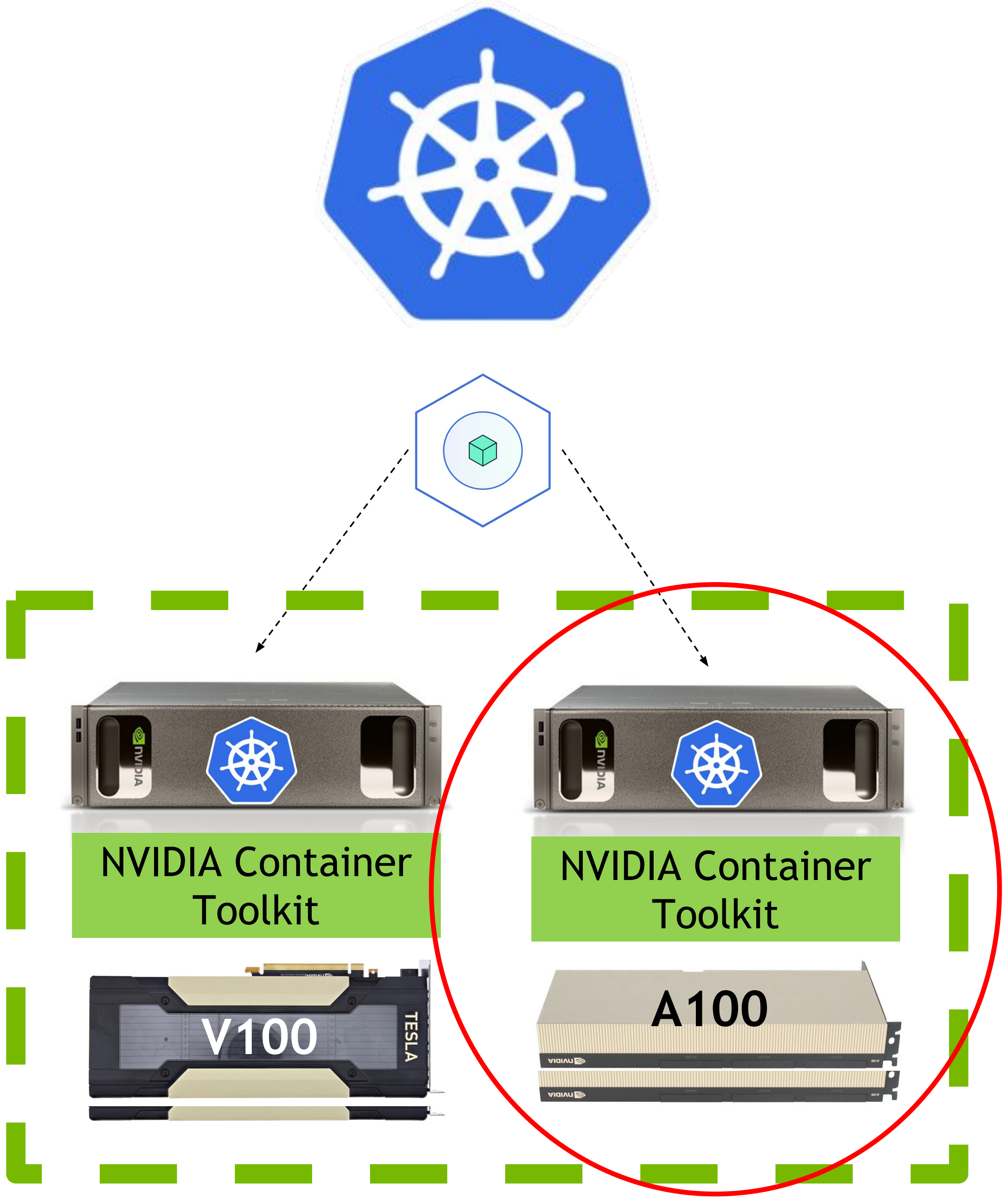
```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 4
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
    nvidia.com/cuda.runtime: 11.0
    nvidia.com/cuda.driver: 450.51.06
```



GPUs AND KUBERNETES

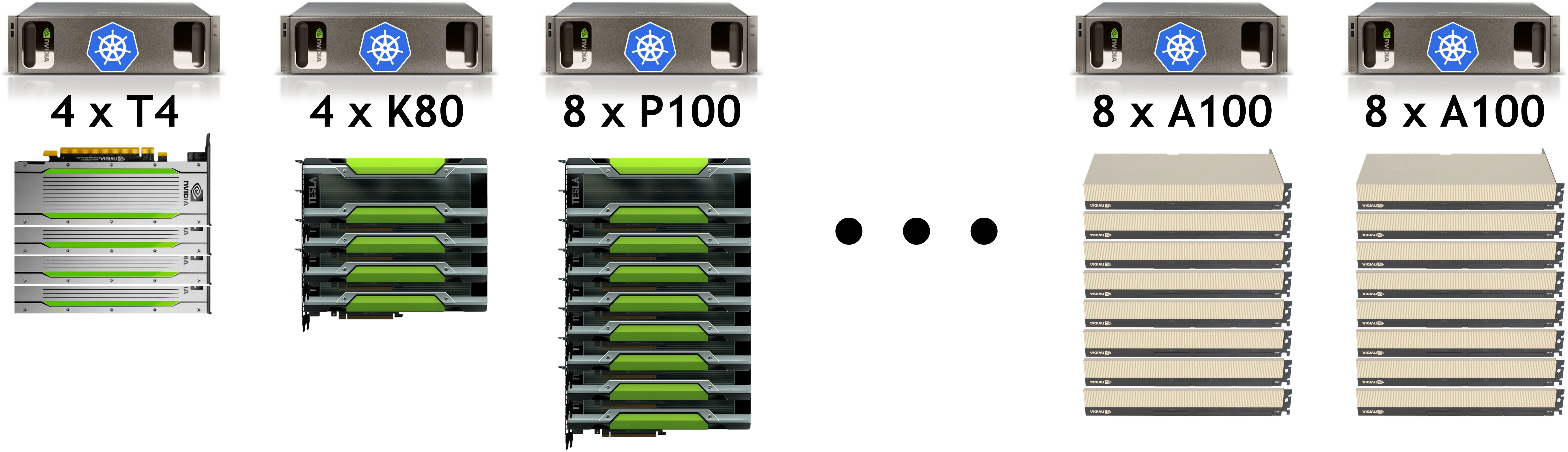
Seamlessly scale up training and inference to a cluster of GPU machines

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 4
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
    nvidia.com/cuda.runtime: 11.0
    nvidia.com/cuda.driver: 450.51.06
```



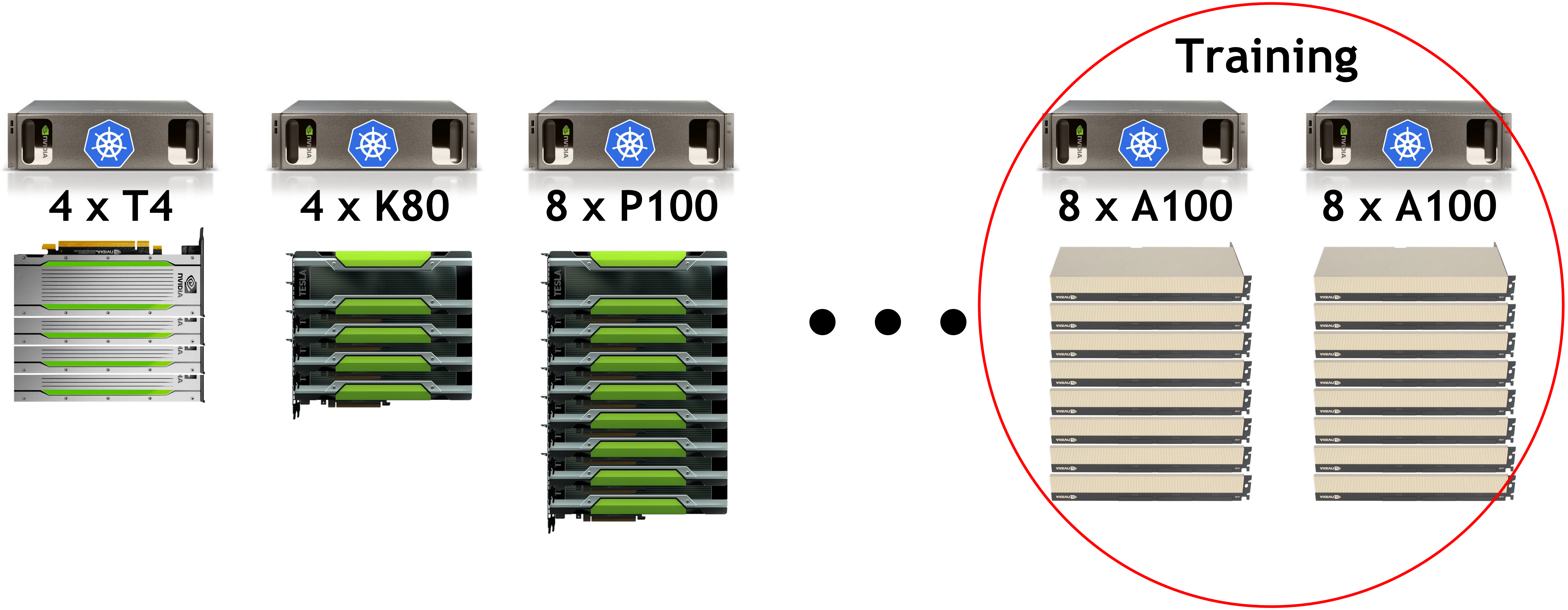
GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



Training

4 x T4

4 x K80

8 x P100

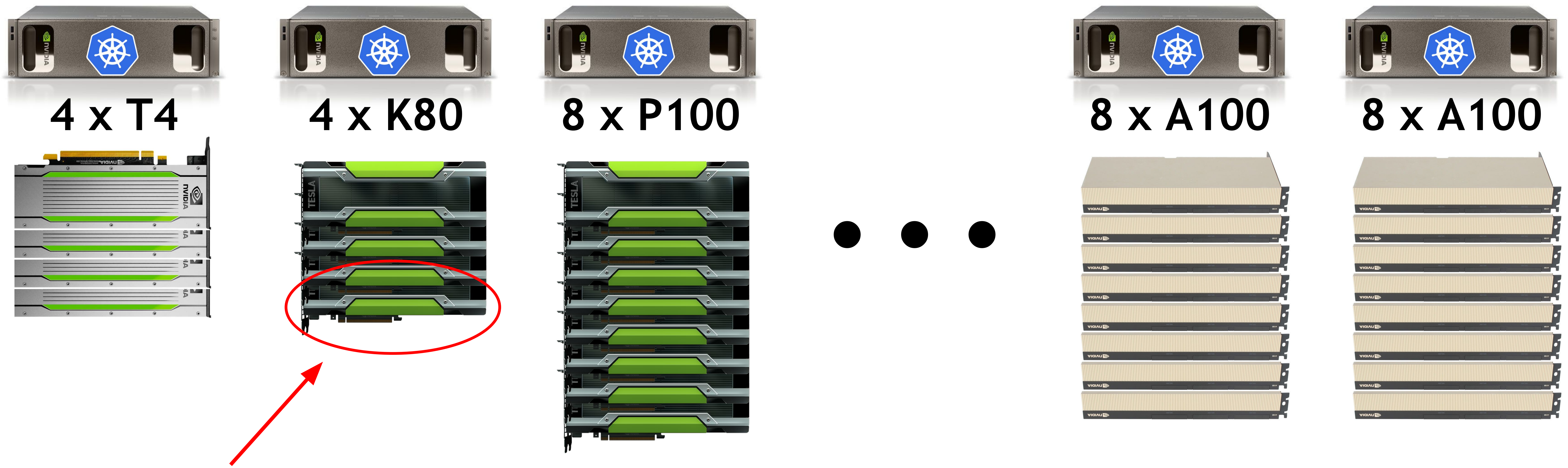
• • •

8 x A100

8 x A100

GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



Inference

4 x T4

4 x K80

8 x P100

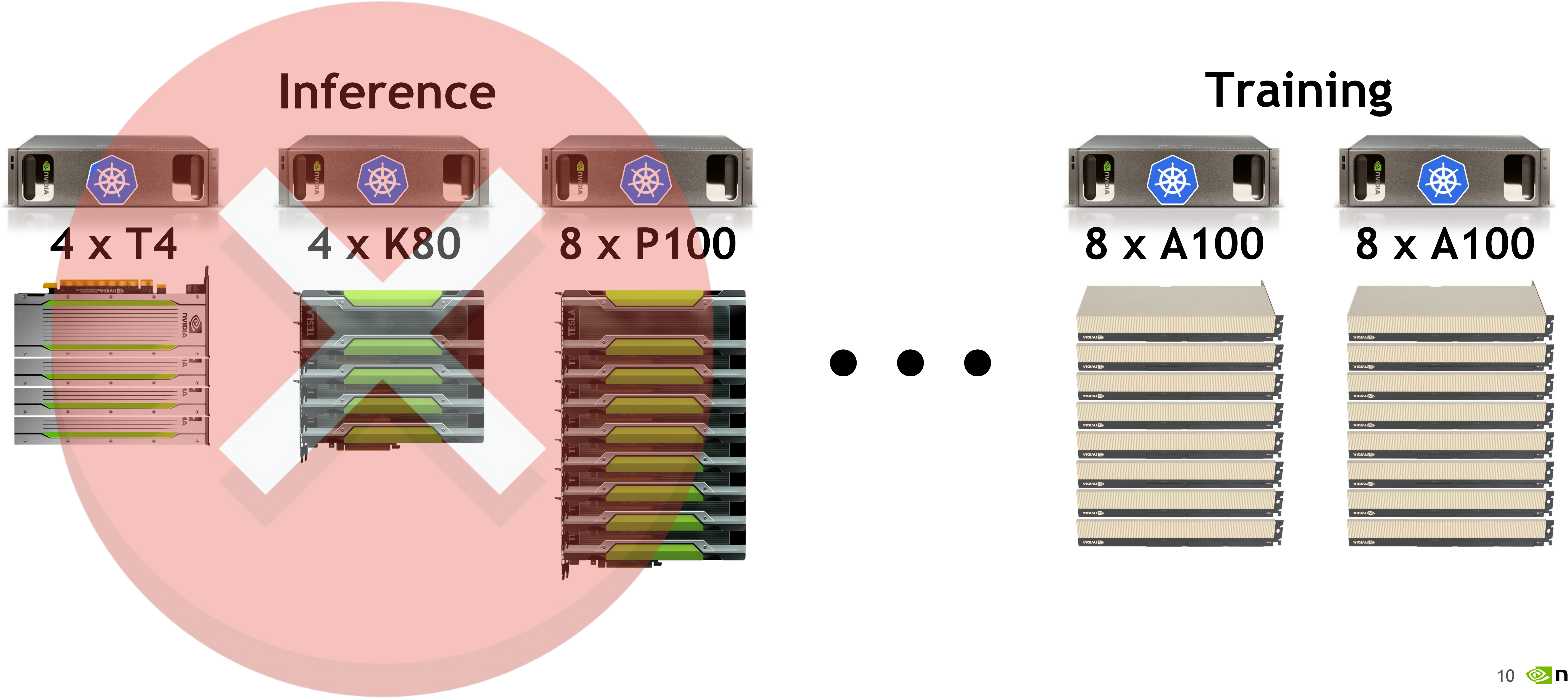
Training

8 x A100

8 x A100

GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



Inference

4 x T4

4 x K80

8 x P100

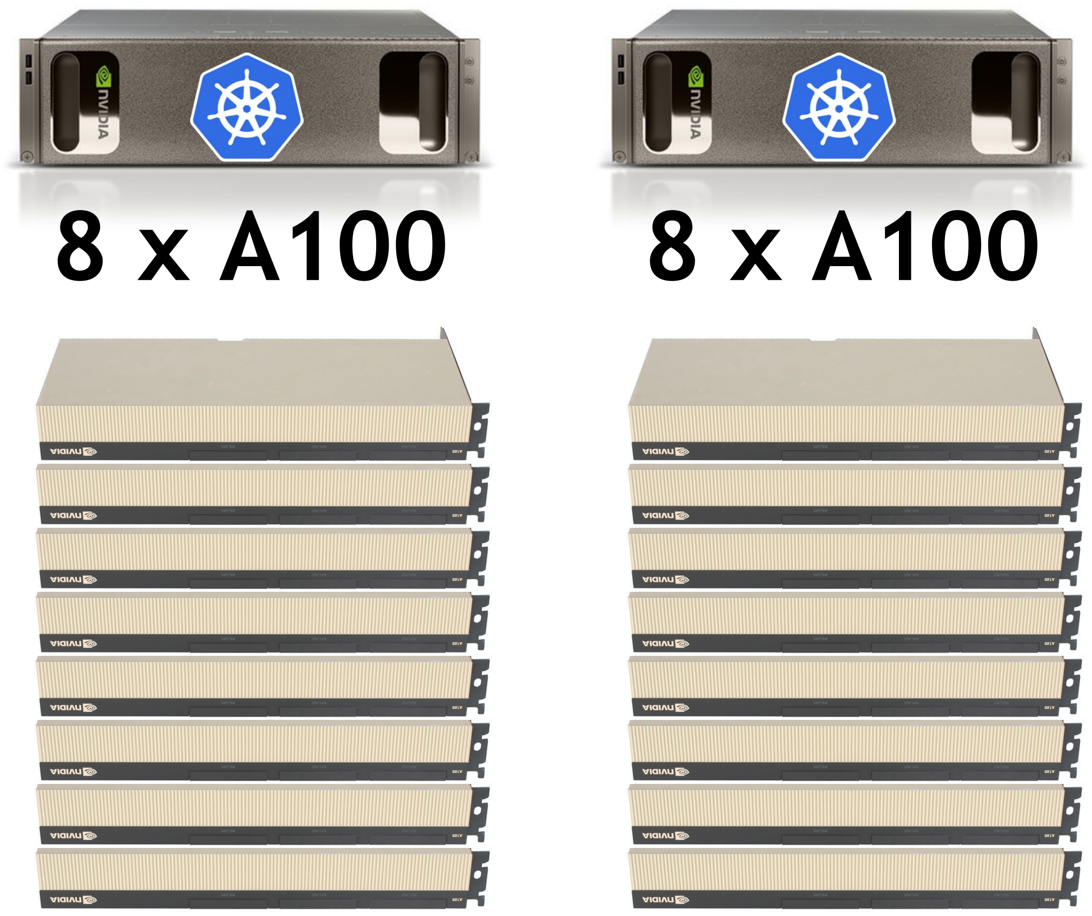
Training

8 x A100

8 x A100

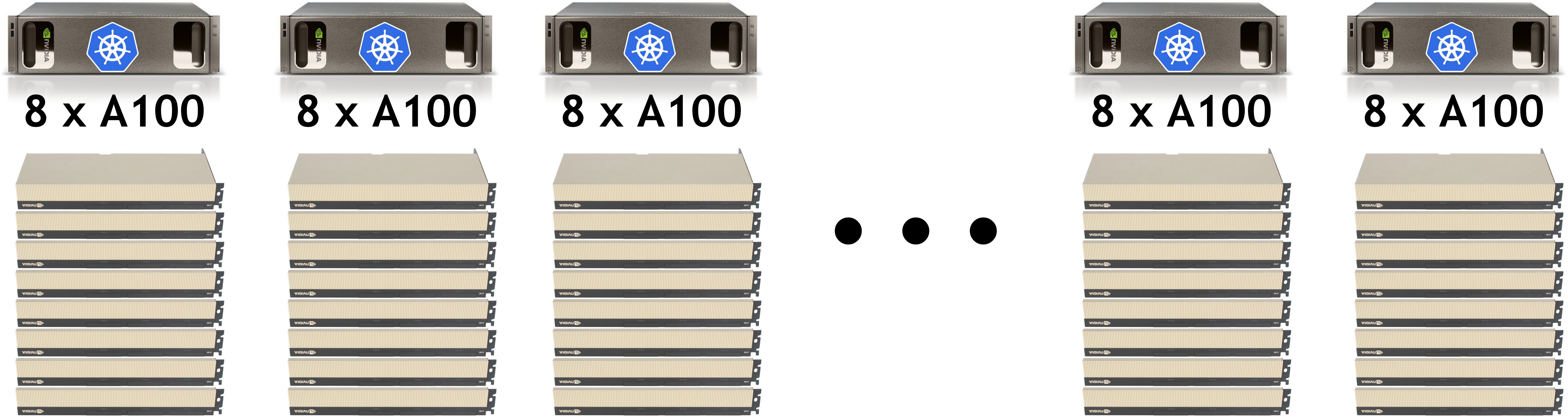
GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines



8 x A100

8 x A100

8 x A100

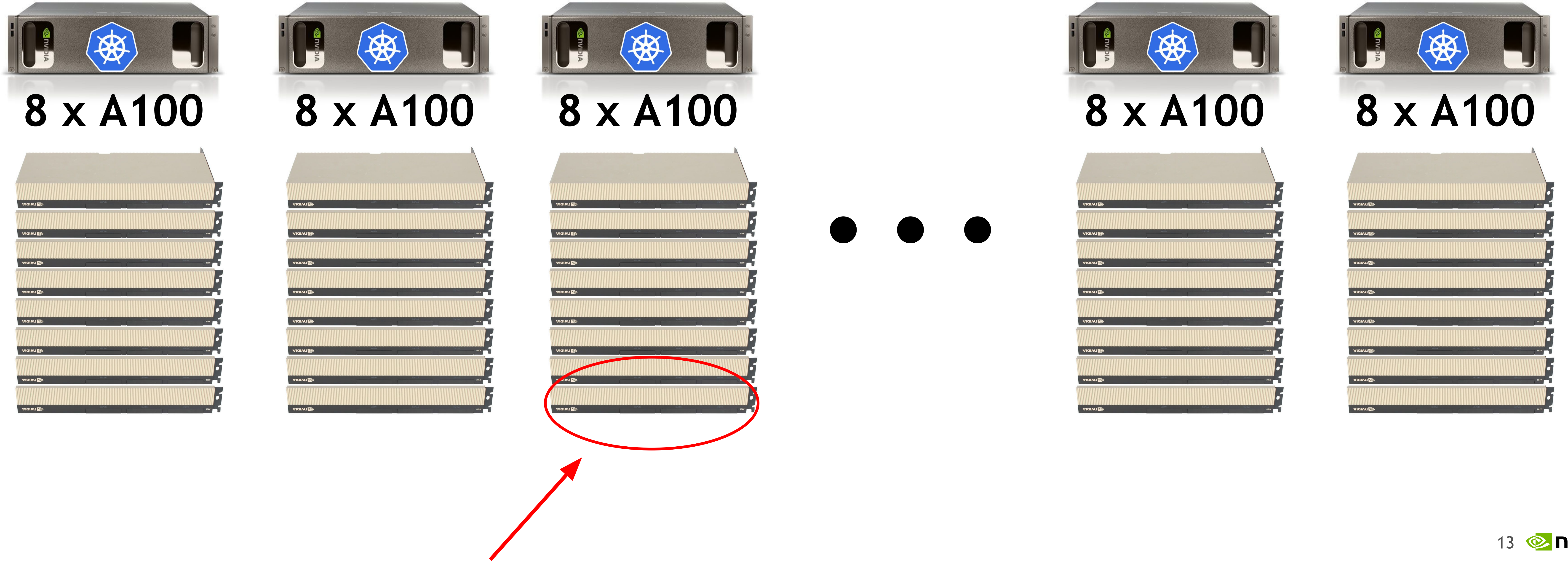
• • •

8 x A100

8 x A100

GPUs AND KUBERNETES

Seamlessly scale up training and inference to a cluster of GPU machines

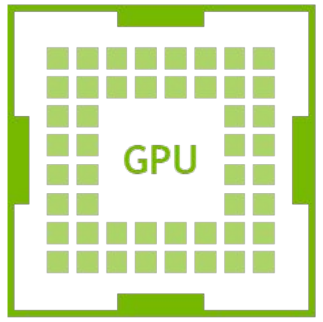


WHAT ARE MULTI-INSTANCE GPUs?

Slices of a full GPU with dedicated memory and compute resources

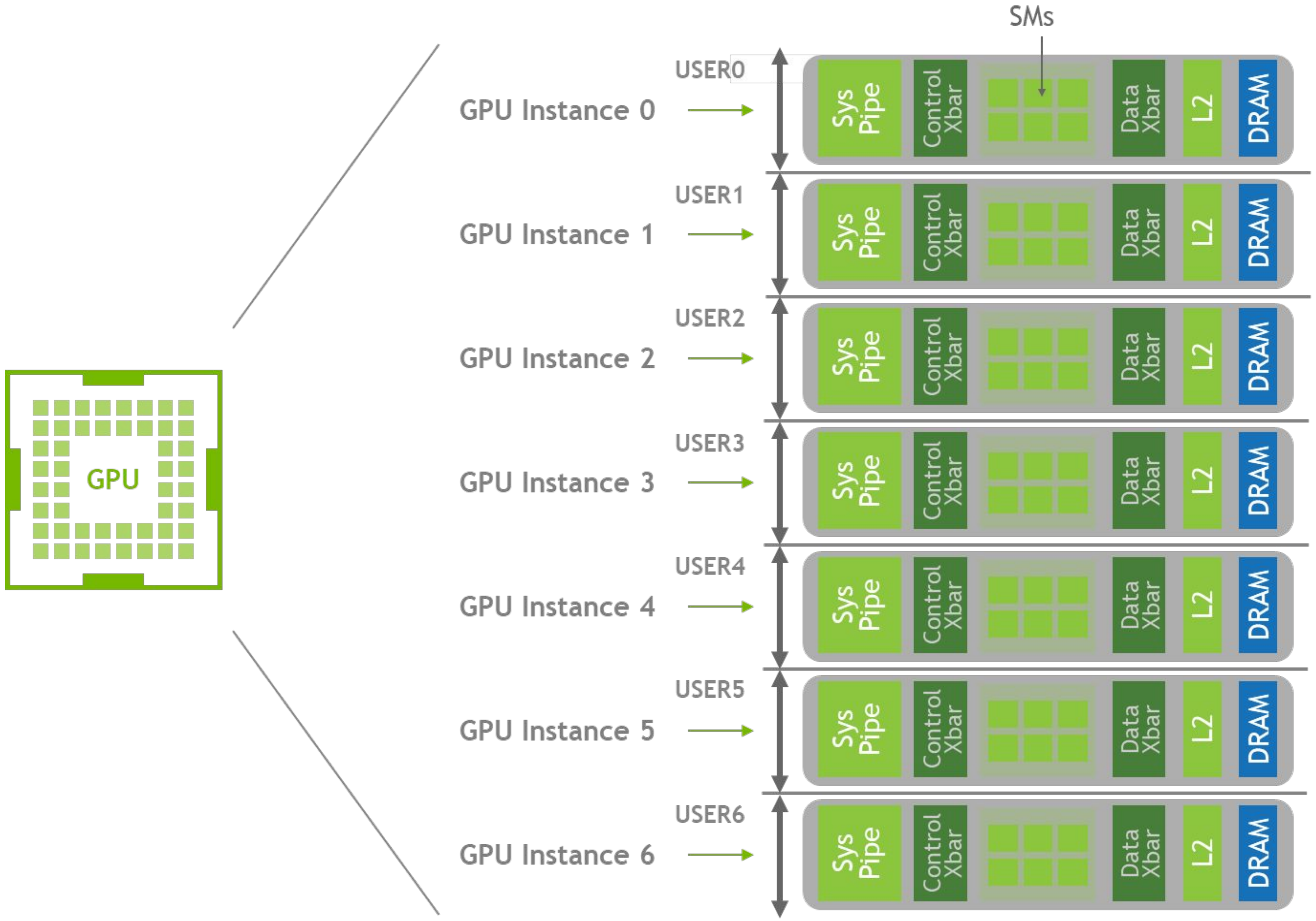
WHAT ARE MULTI-INSTANCE GPUs?

Slices of a full GPU with dedicated memory and compute resources



WHAT ARE MULTI-INSTANCE GPUs?

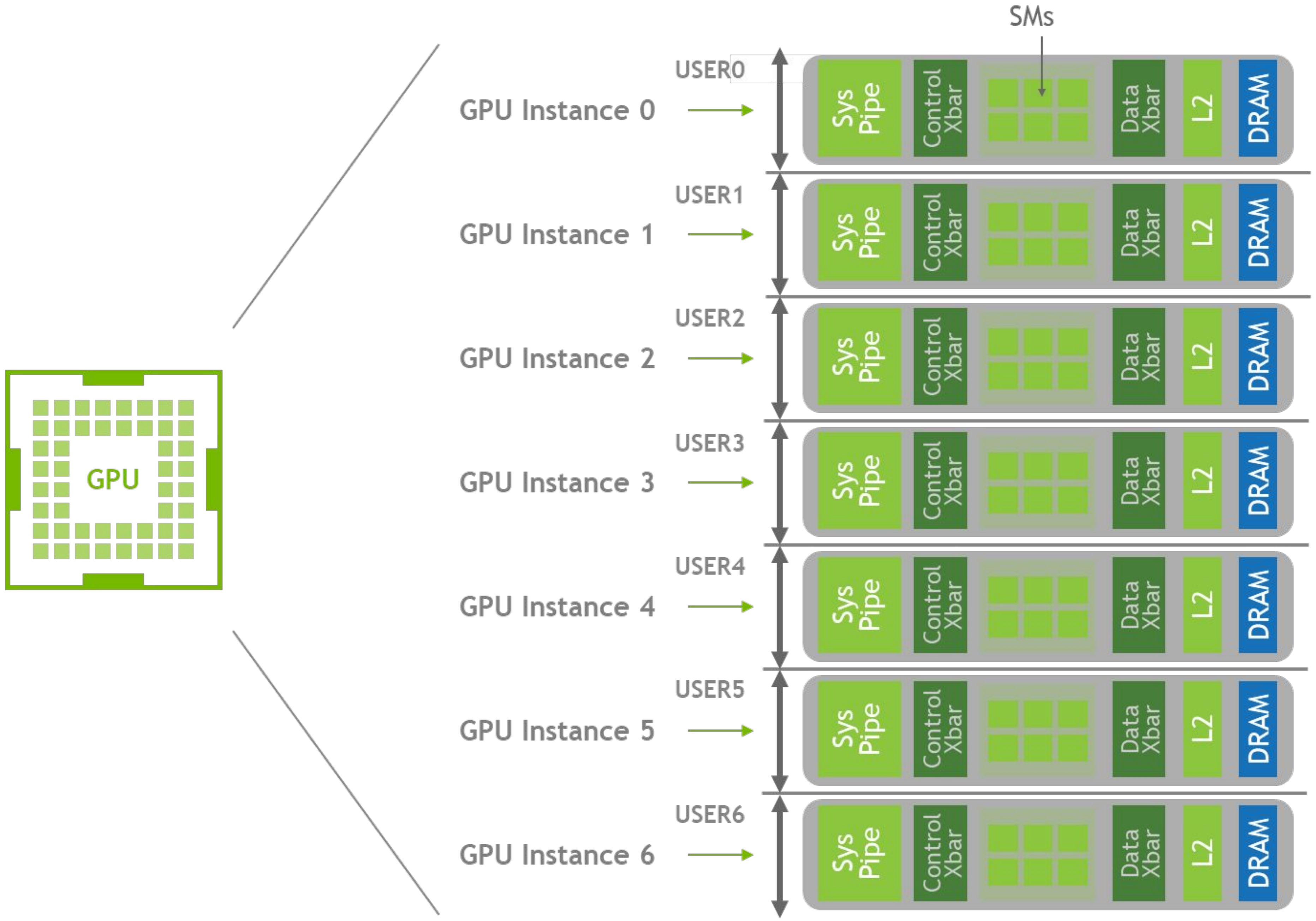
Slices of a full GPU with dedicated memory and compute resources



Create multiple “GPU Instances” on a single GPU:
Dedicated SM, Memory, L2 cache, Bandwidth for hardware QoS & isolation

WHAT ARE MULTI-INSTANCE GPUs?

Slices of a full GPU with dedicated memory and compute resources

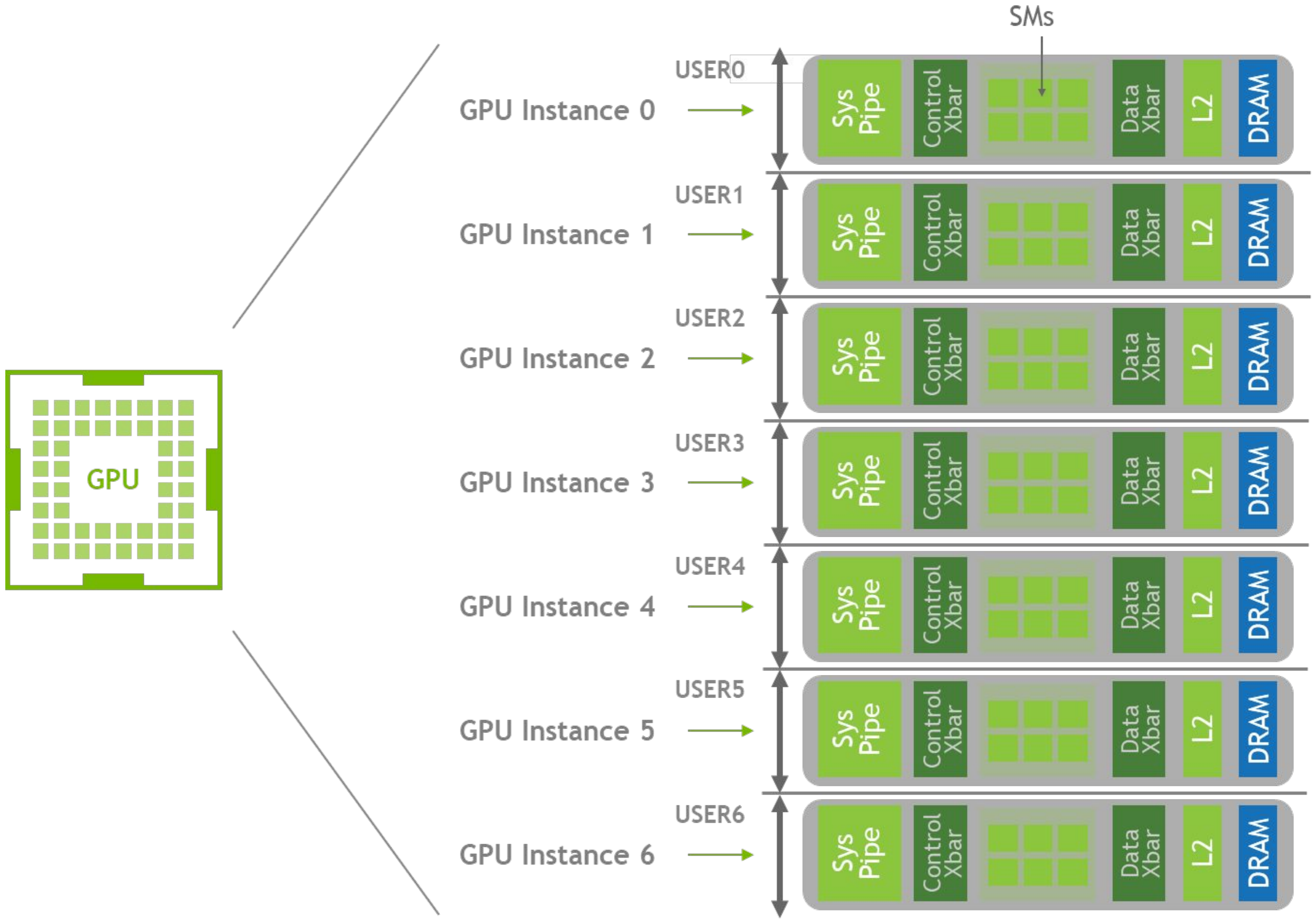


Create multiple “GPU Instances” on a single GPU:
Dedicated SM, Memory, L2 cache, Bandwidth for hardware QoS & isolation

Simultaneous Workload Execution With Guaranteed Quality Of Service:
All GPU instances run in parallel with predictable throughput & latency

WHAT ARE MULTI-INSTANCE GPUs?

Slices of a full GPU with dedicated memory and compute resources



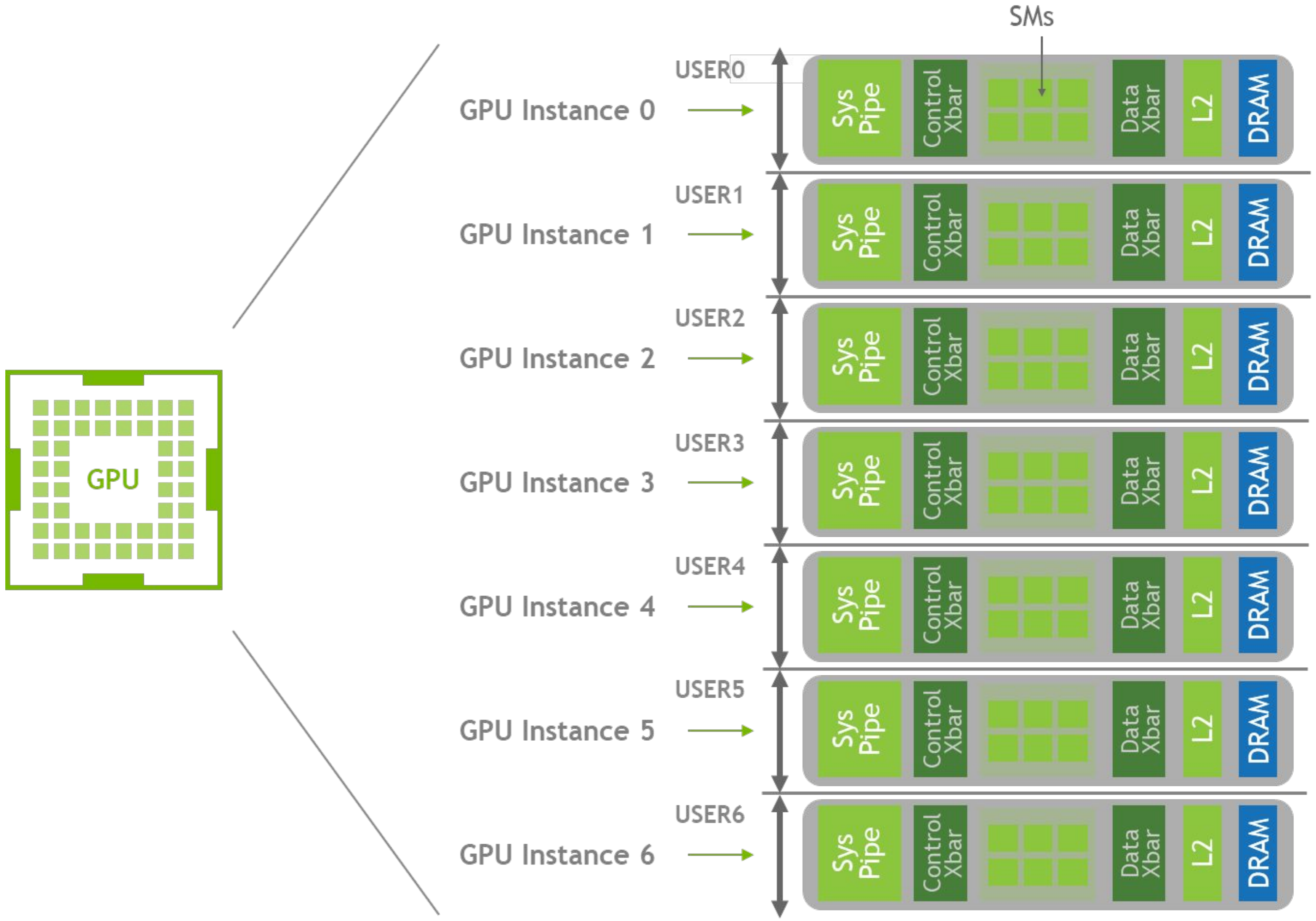
Create multiple “GPU Instances” on a single GPU:
Dedicated SM, Memory, L2 cache, Bandwidth for hardware QoS & isolation

Simultaneous Workload Execution With Guaranteed Quality Of Service:
All GPU instances run in parallel with predictable throughput & latency

Right Sized GPU Allocation:
Different sized GPU instances based on target workloads

WHAT ARE MULTI-INSTANCE GPUs?

Slices of a full GPU with dedicated memory and compute resources



Create multiple "GPU Instances" on a single GPU:
Dedicated SM, Memory, L2 cache, Bandwidth for hardware QoS & isolation

Simultaneous Workload Execution With Guaranteed Quality Of Service:
All GPU instances run in parallel with predictable throughput & latency

Right Sized GPU Allocation:
Different sized GPU instances based on target workloads

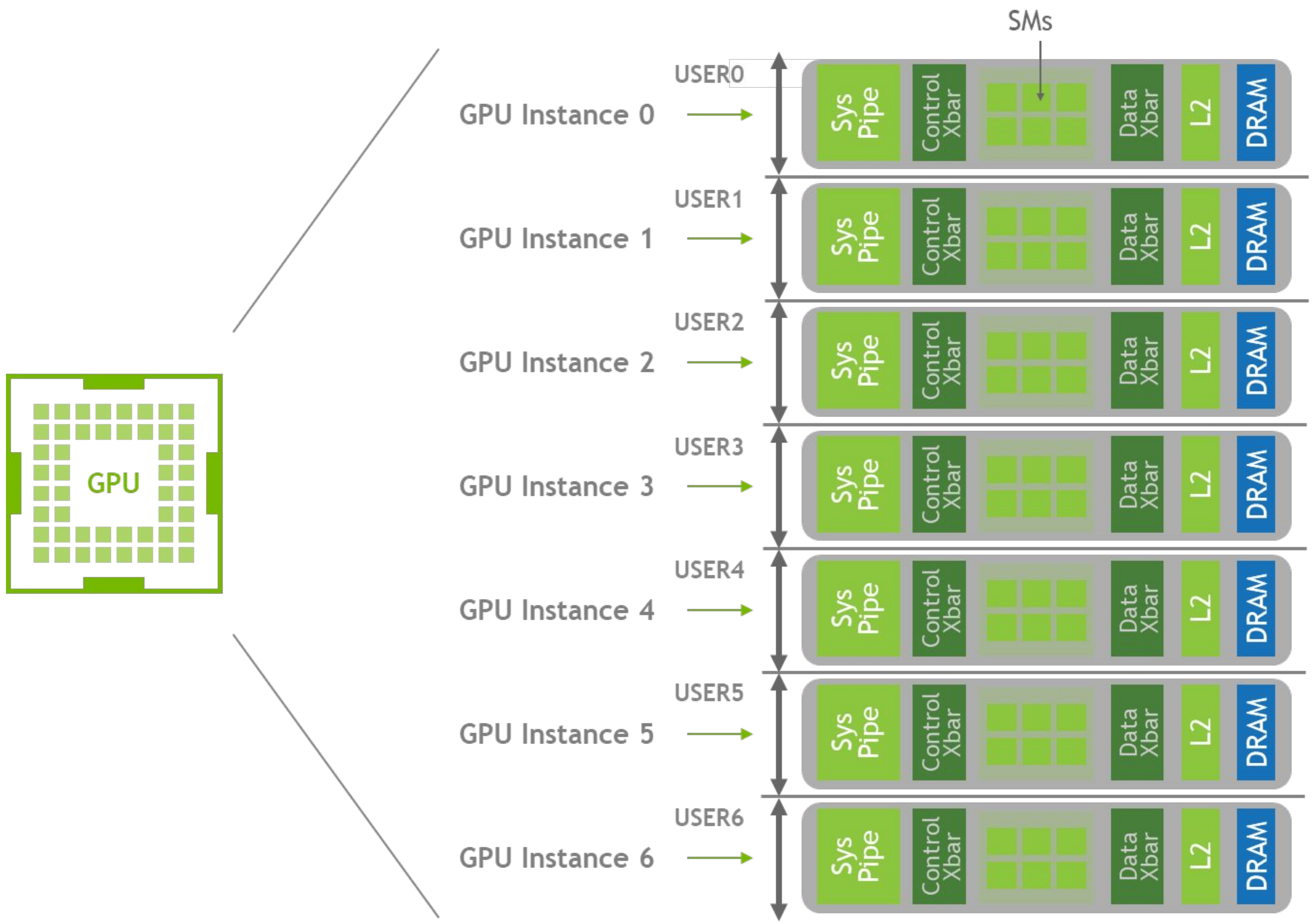
Diverse Deployment Environments:
Supported with Bare metal, Container and Virtualized Env.

OUTLINE

- Multi-Instance GPUs (MIG)
- GPUs and Containers
- GPUs and Kubernetes
- MIG in Containers
- MIG in Kubernetes
- System-Level Interface for MIG
- Best-Practices for Provisioning MIG
- Putting it all together — Demo

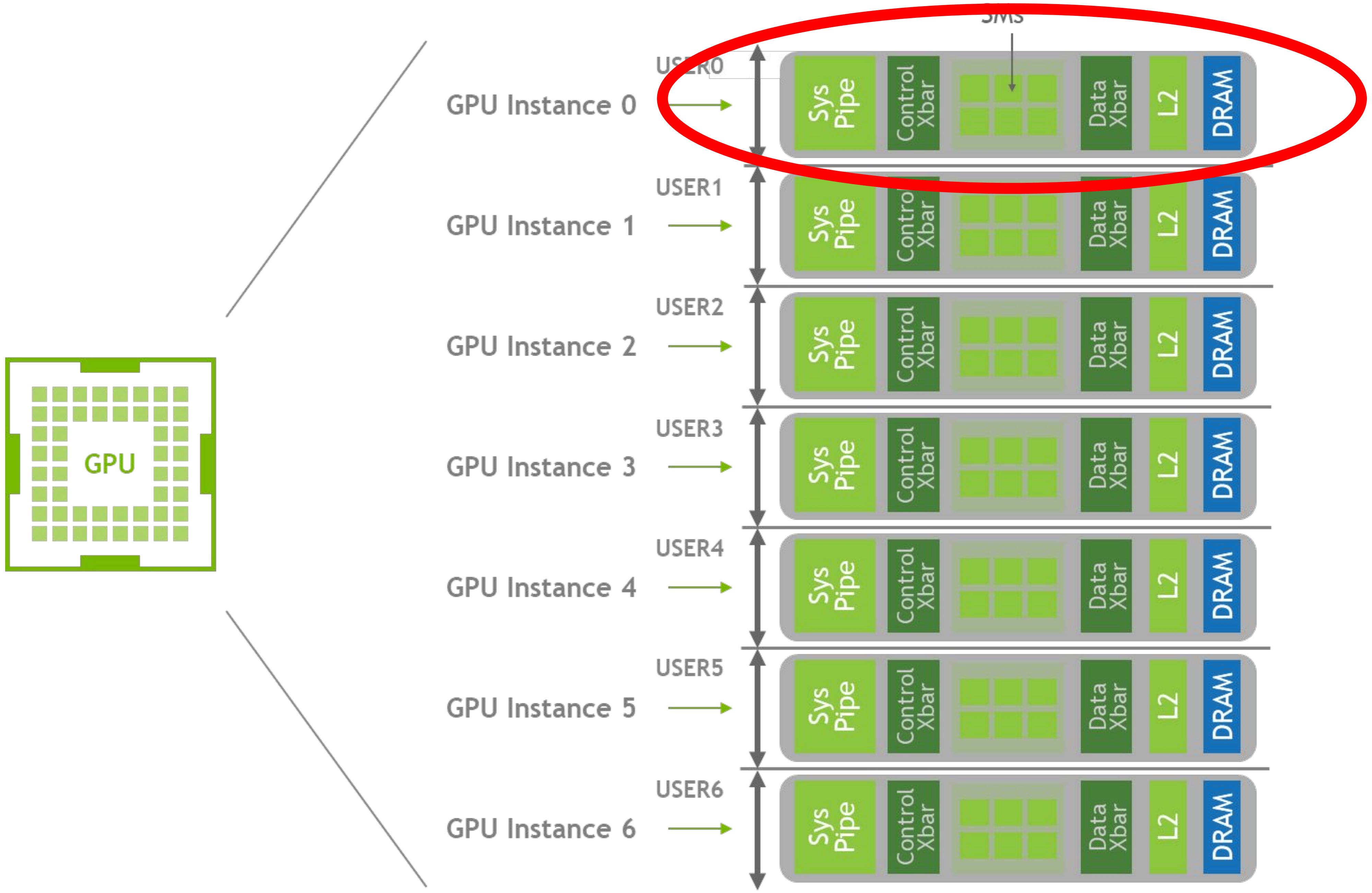
MULTI-INSTANCE GPUs (MIG)

Use Cases



MULTI-INSTANCE GPUs (MIG)

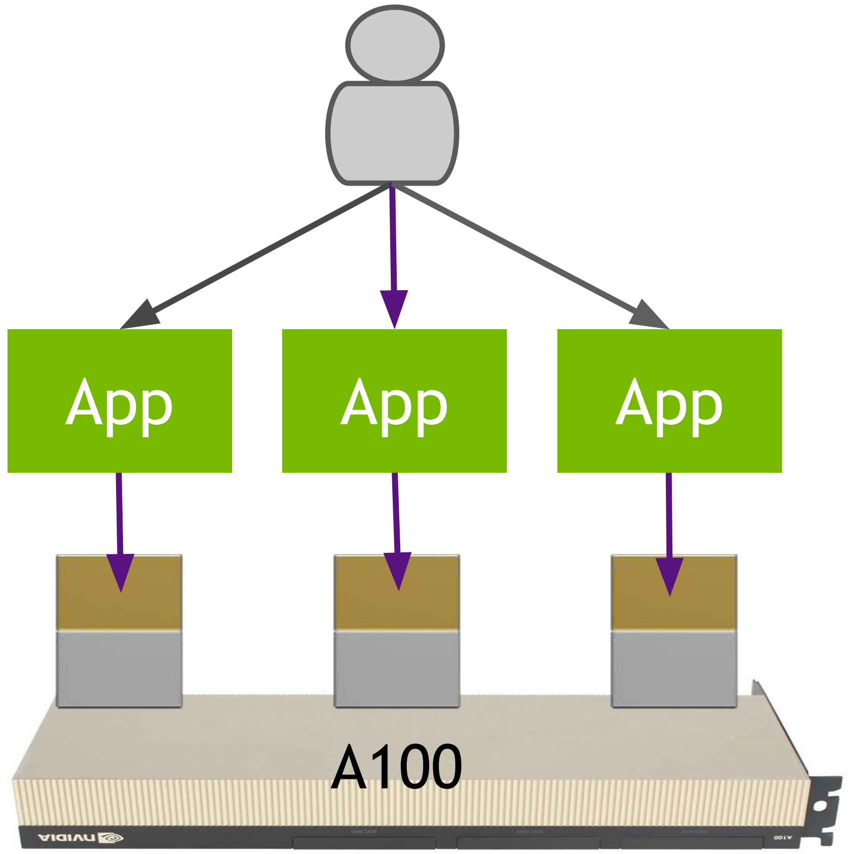
Use Cases



MULTI-INSTANCE GPUs (MIG)

Use Cases

Single User → Multiple Apps

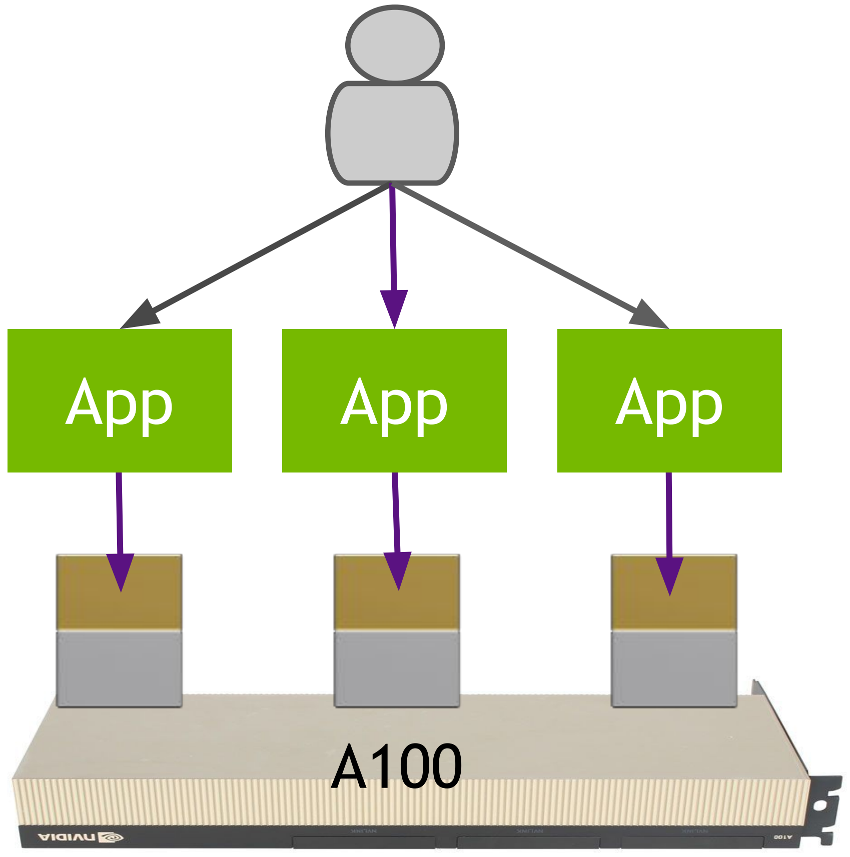


E.g. *Multiple inference jobs*

MULTI-INSTANCE GPUs (MIG)

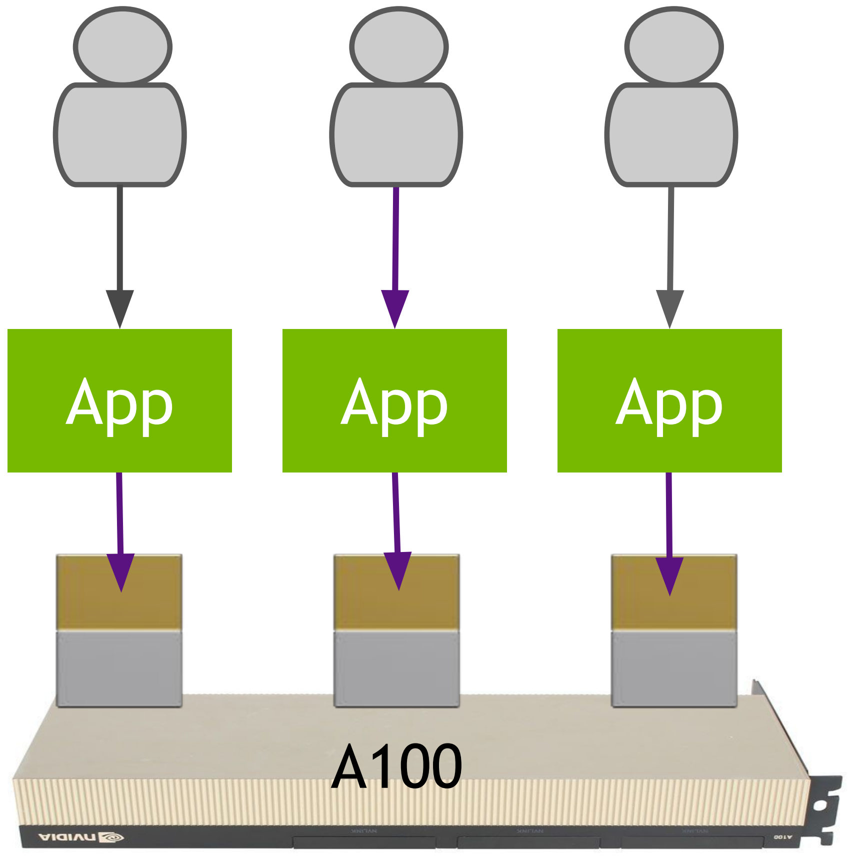
Use Cases

Single User → Multiple Apps



E.g. *Multiple inference jobs*

Single Tenant → Multi-User

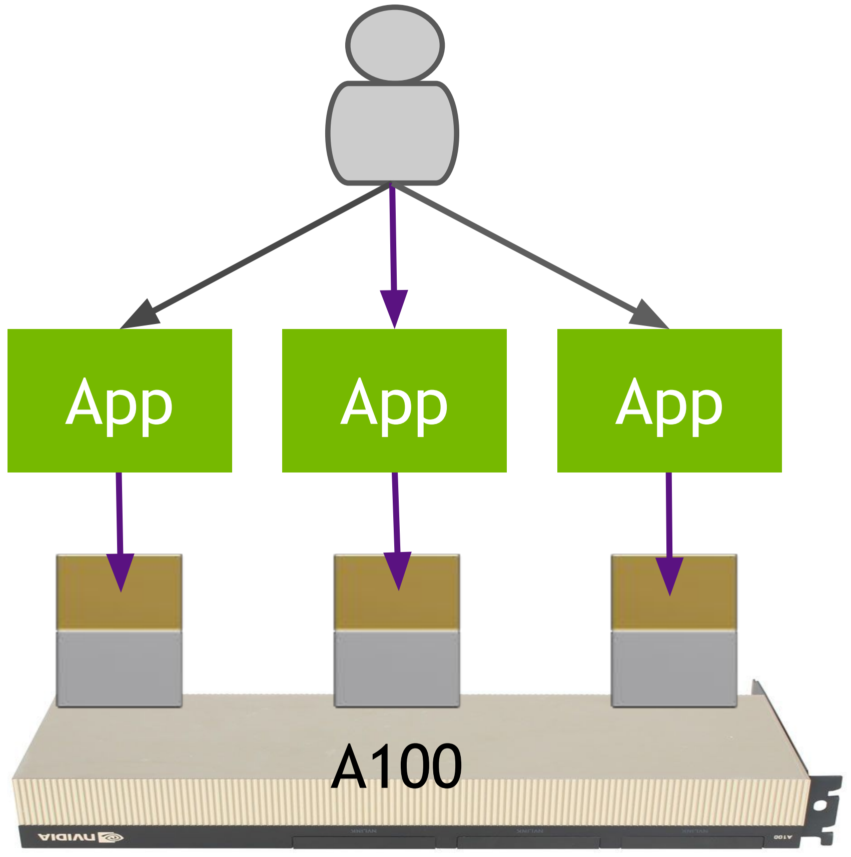


E.g. *Jupyter notebooks for model exploration*

MULTI-INSTANCE GPUs (MIG)

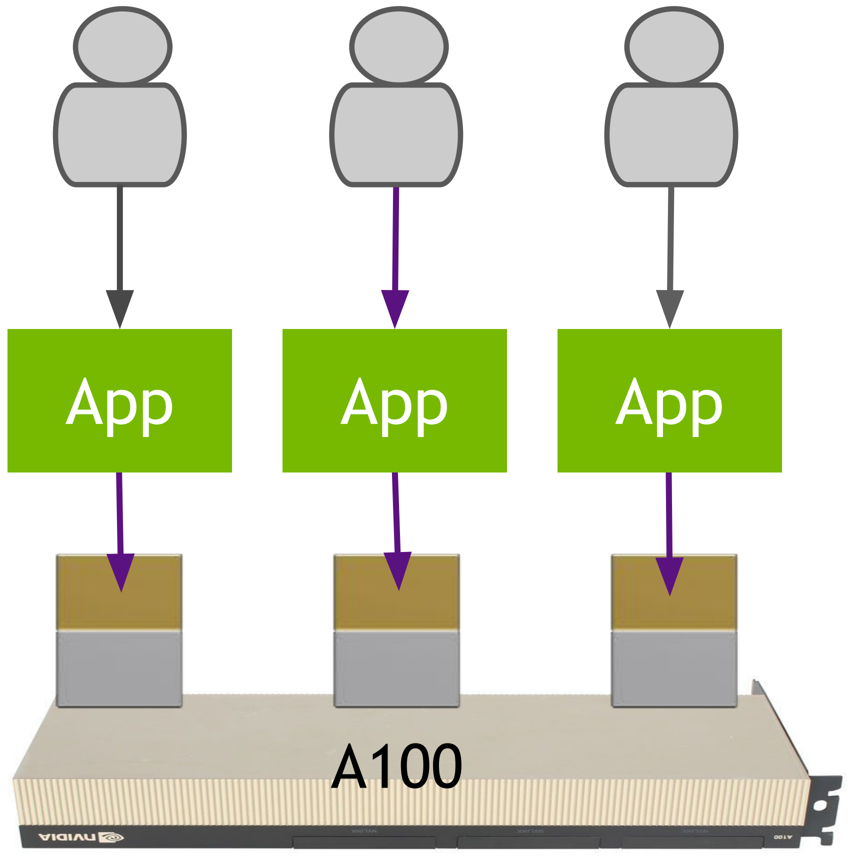
Use Cases

Single User → Multiple Apps



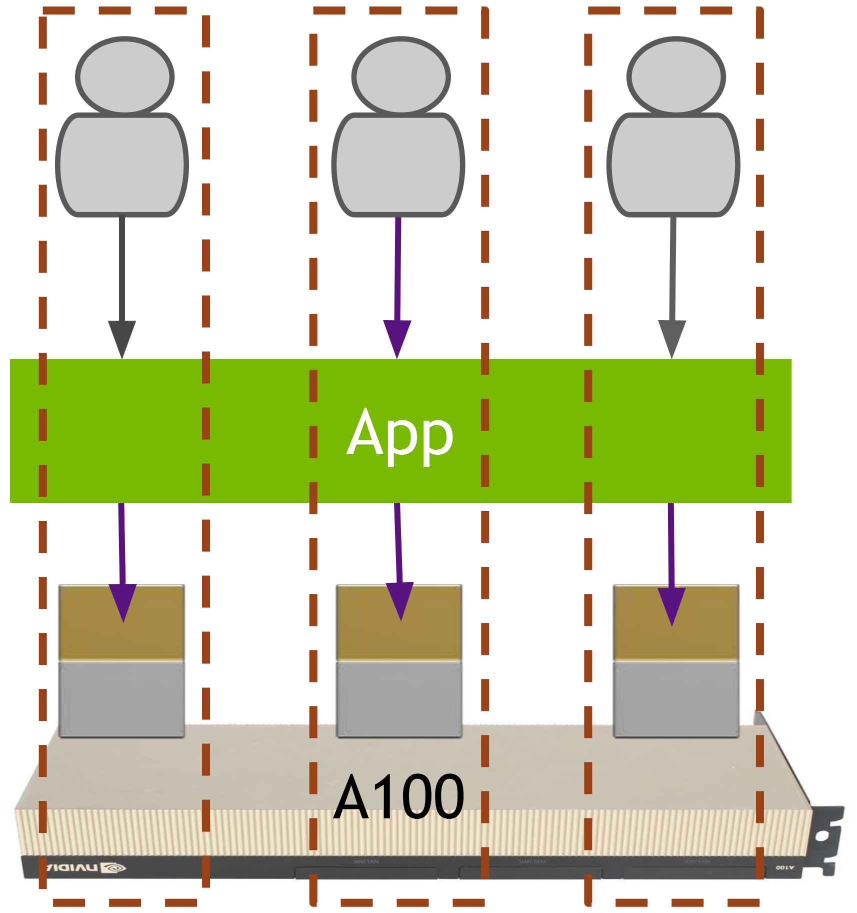
E.g. *Multiple inference jobs*

Single Tenant → Multi-User



E.g. *Jupyter notebooks for model exploration*

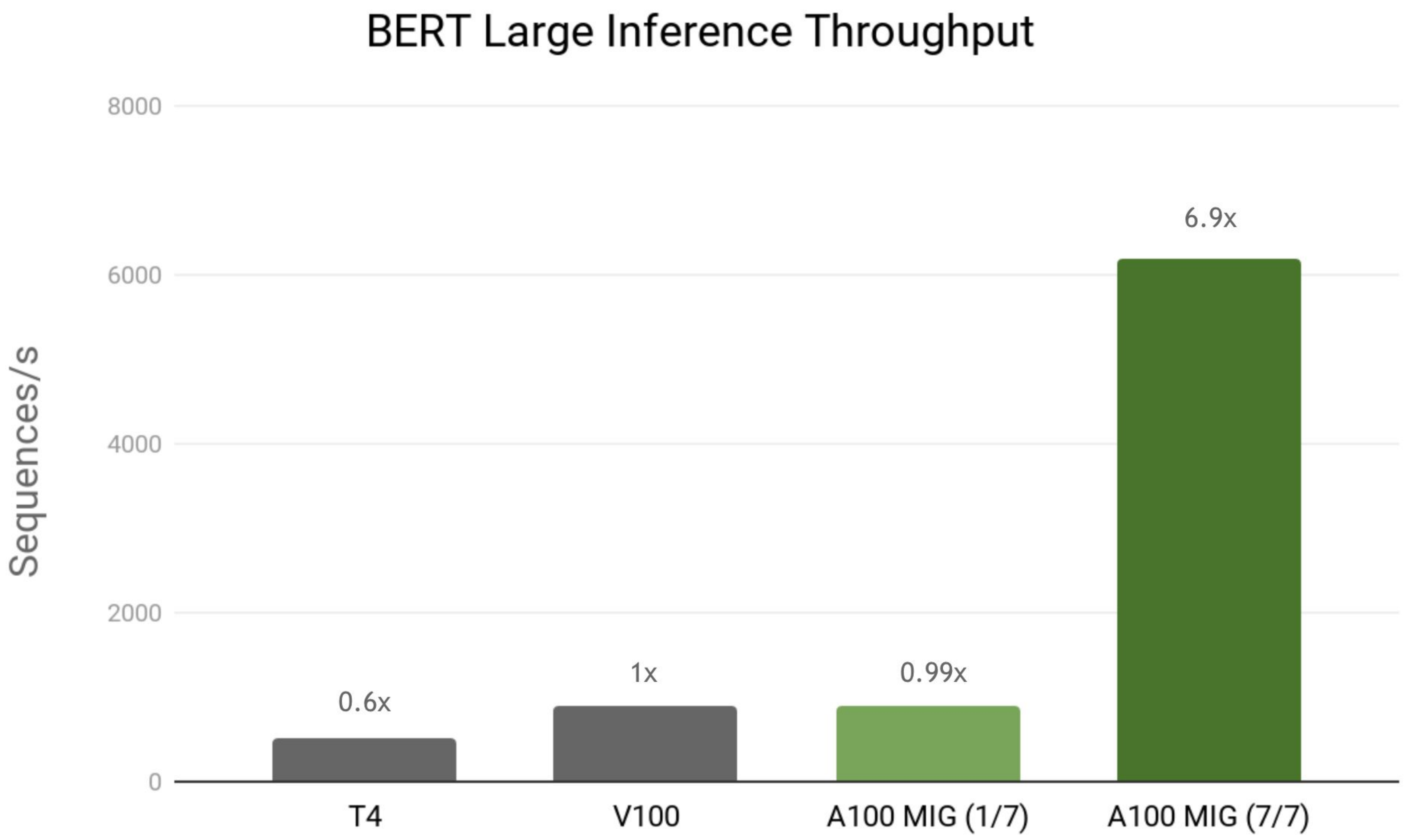
Multi-Tenant → Multi-User



E.g. *Managed Cloud Services*

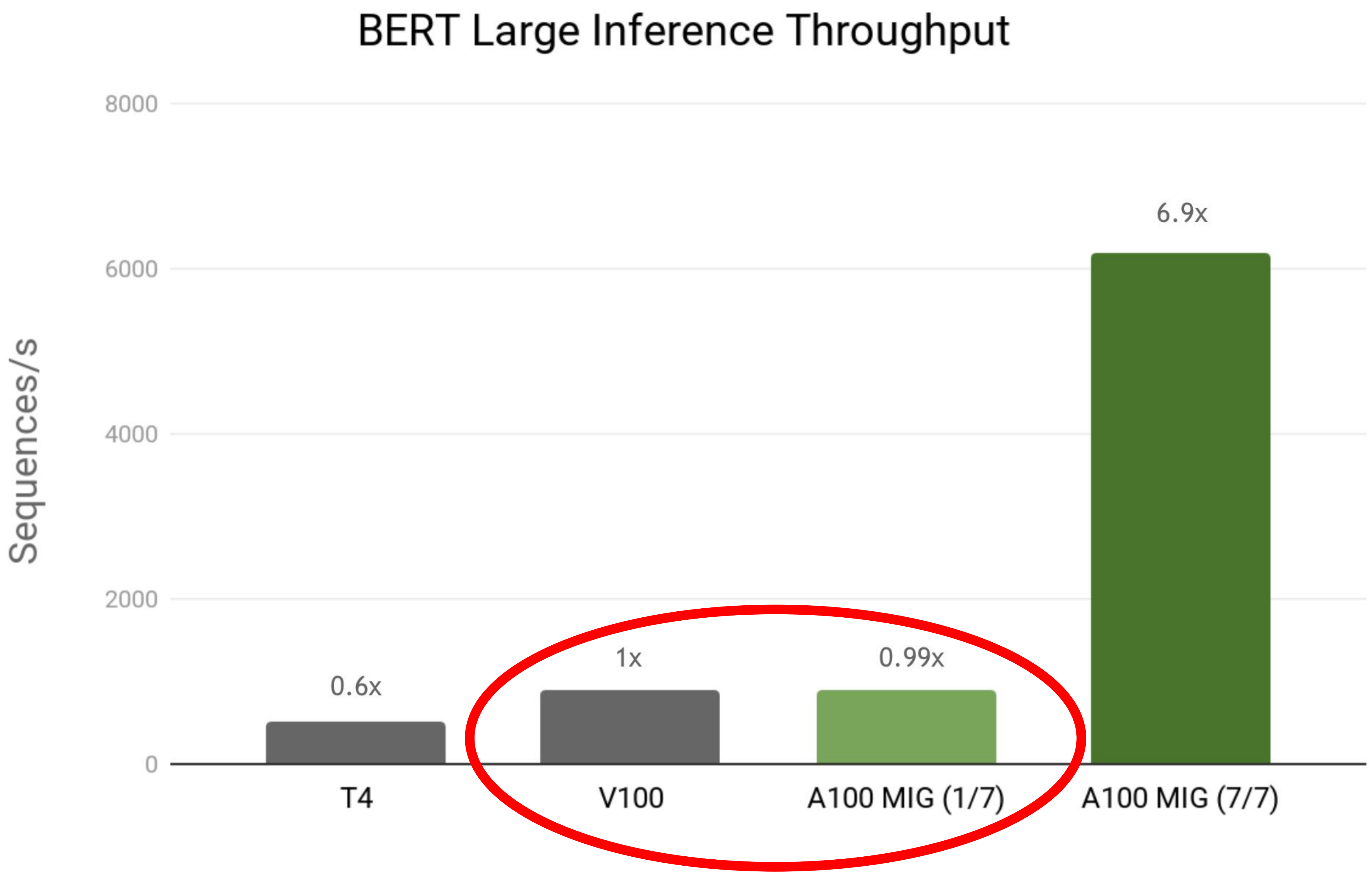
MULTI-INSTANCE GPUs (MIG)

Relative Performance



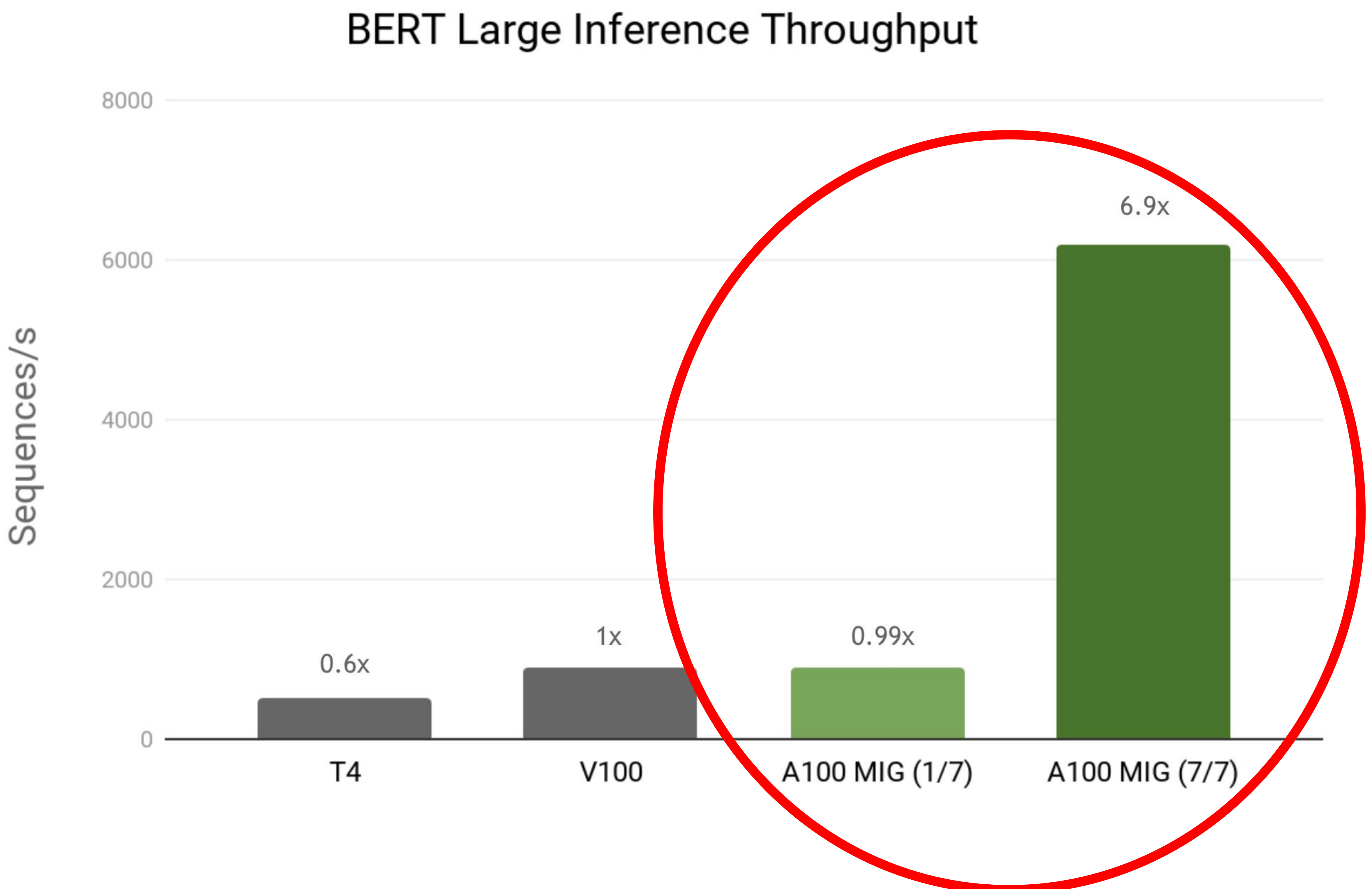
MULTI-INSTANCE GPUs (MIG)

Relative Performance



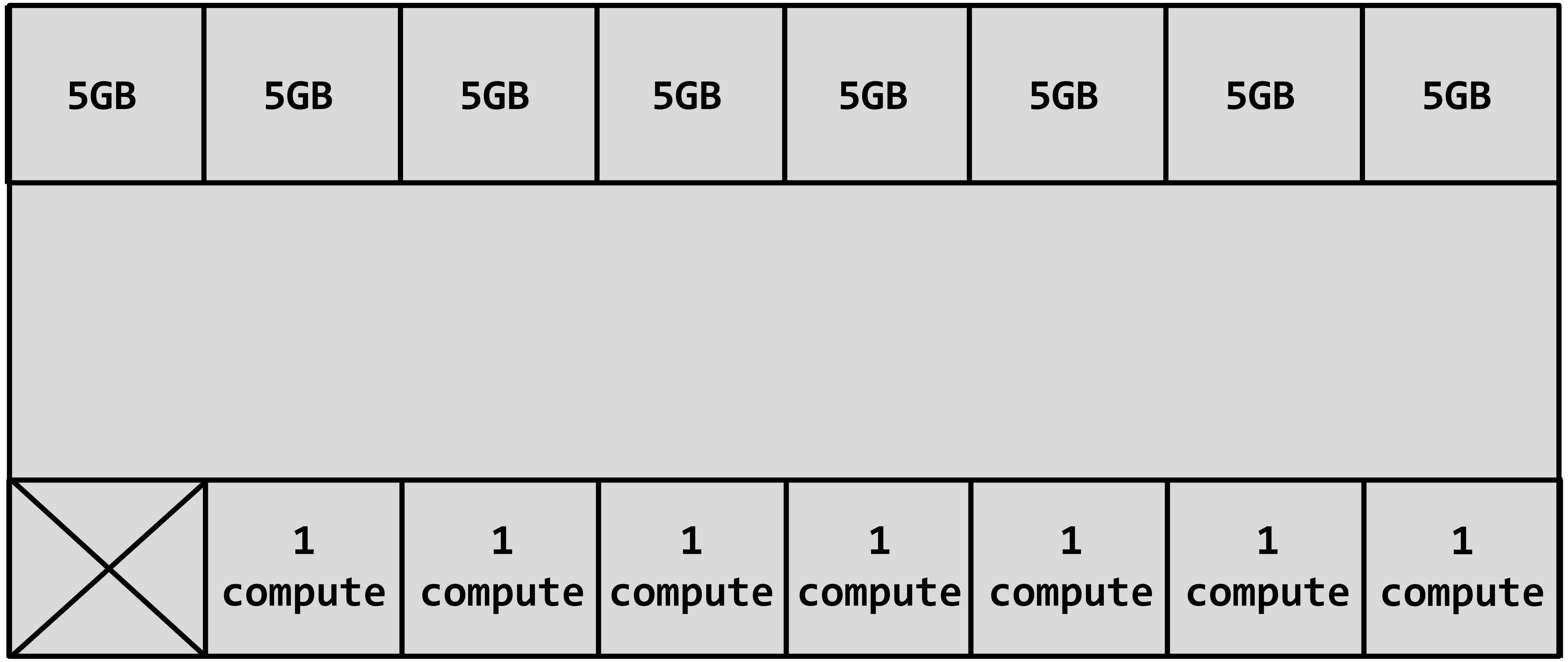
MULTI-INSTANCE GPUs (MIG)

Relative Performance



MULTI-INSTANCE GPUs (MIG)

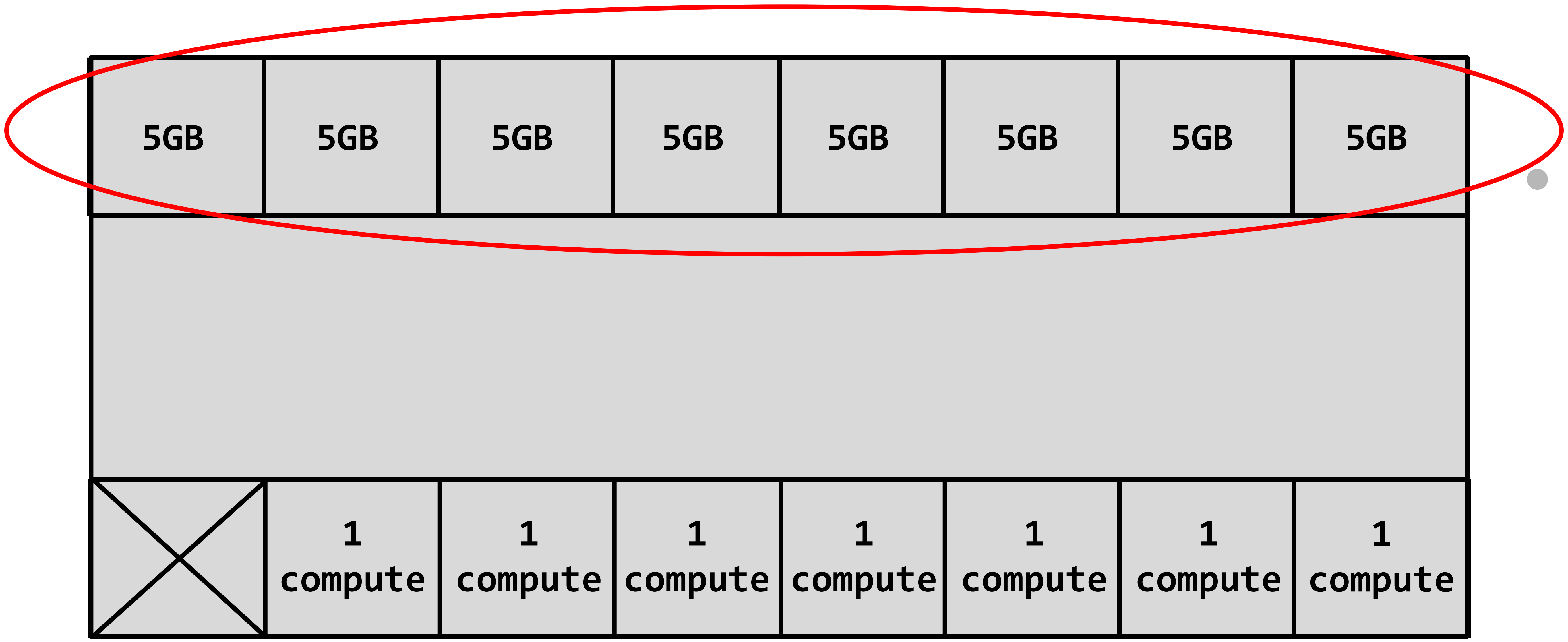
GPU Instances, Compute Instances, and MIG Devices



NVIDIA A100 (40GB)

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



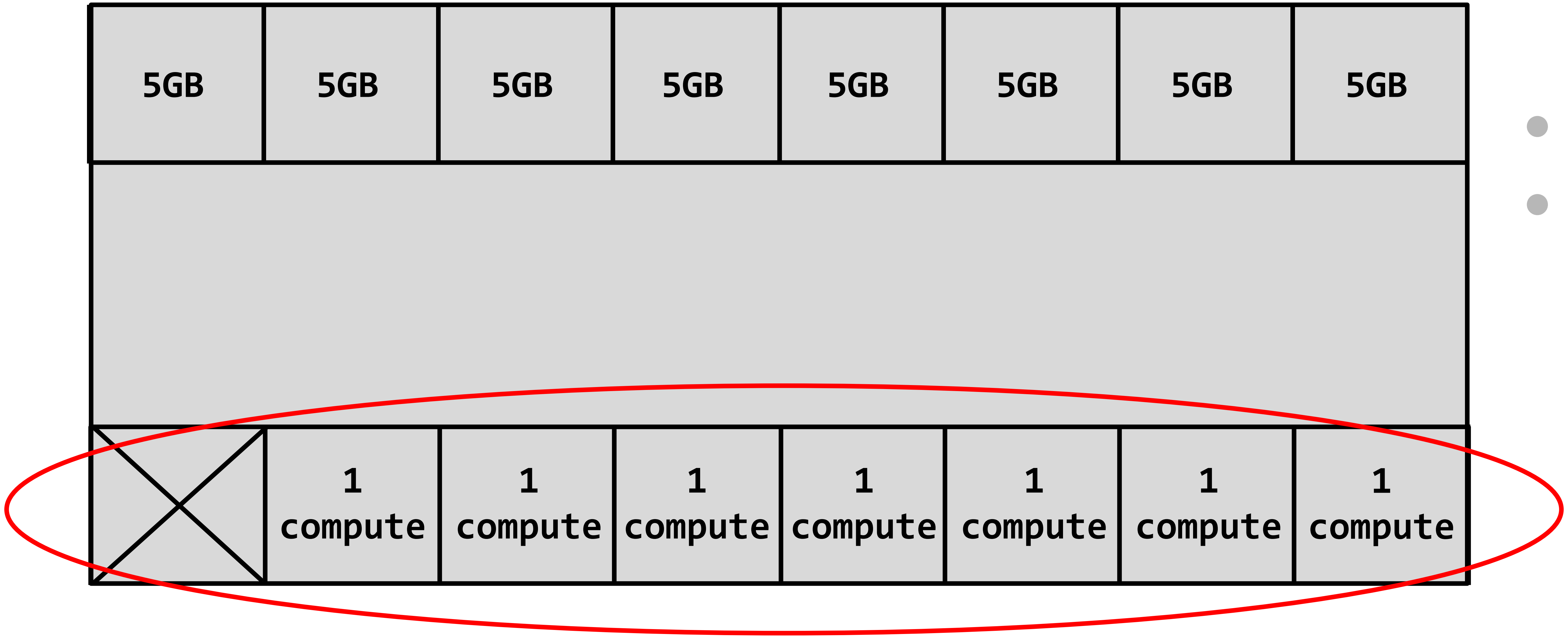
5GB	5GB	5GB	5GB	5GB	5GB	5GB	5GB
☒	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

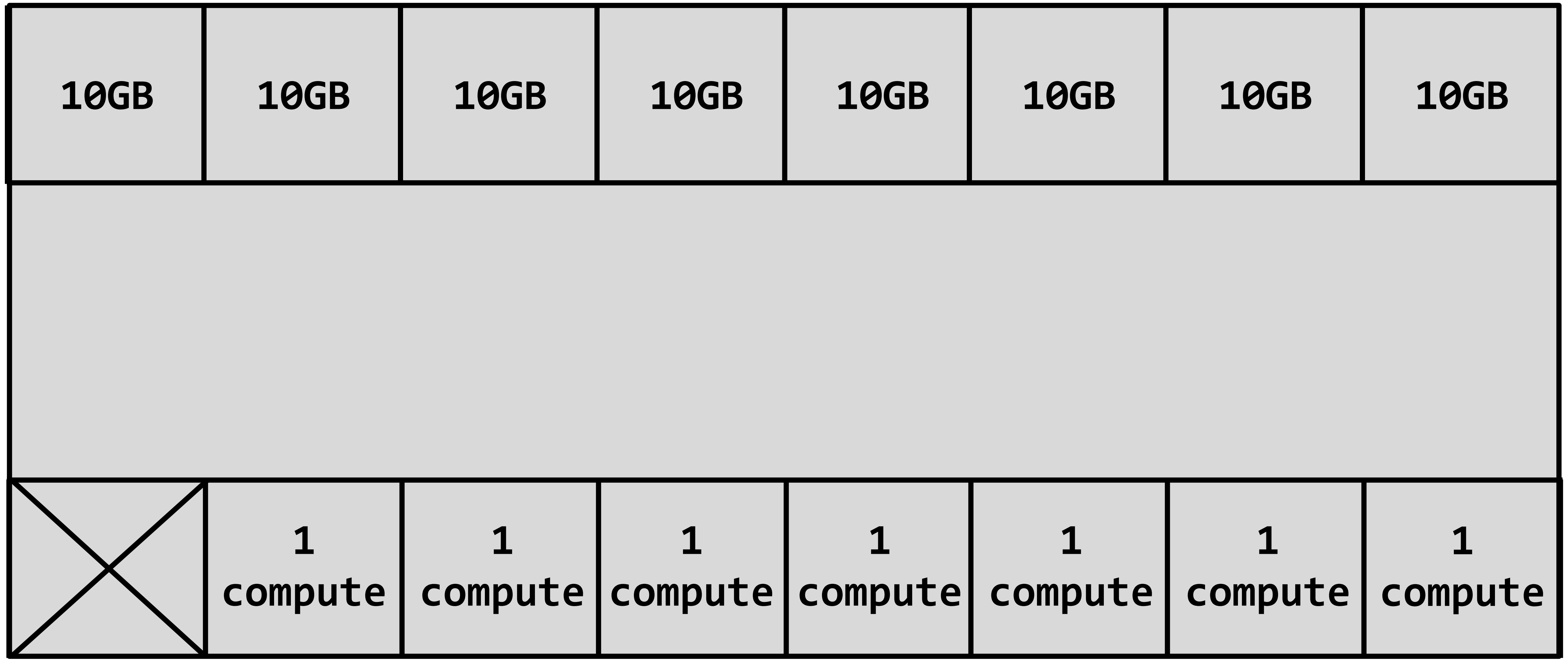


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

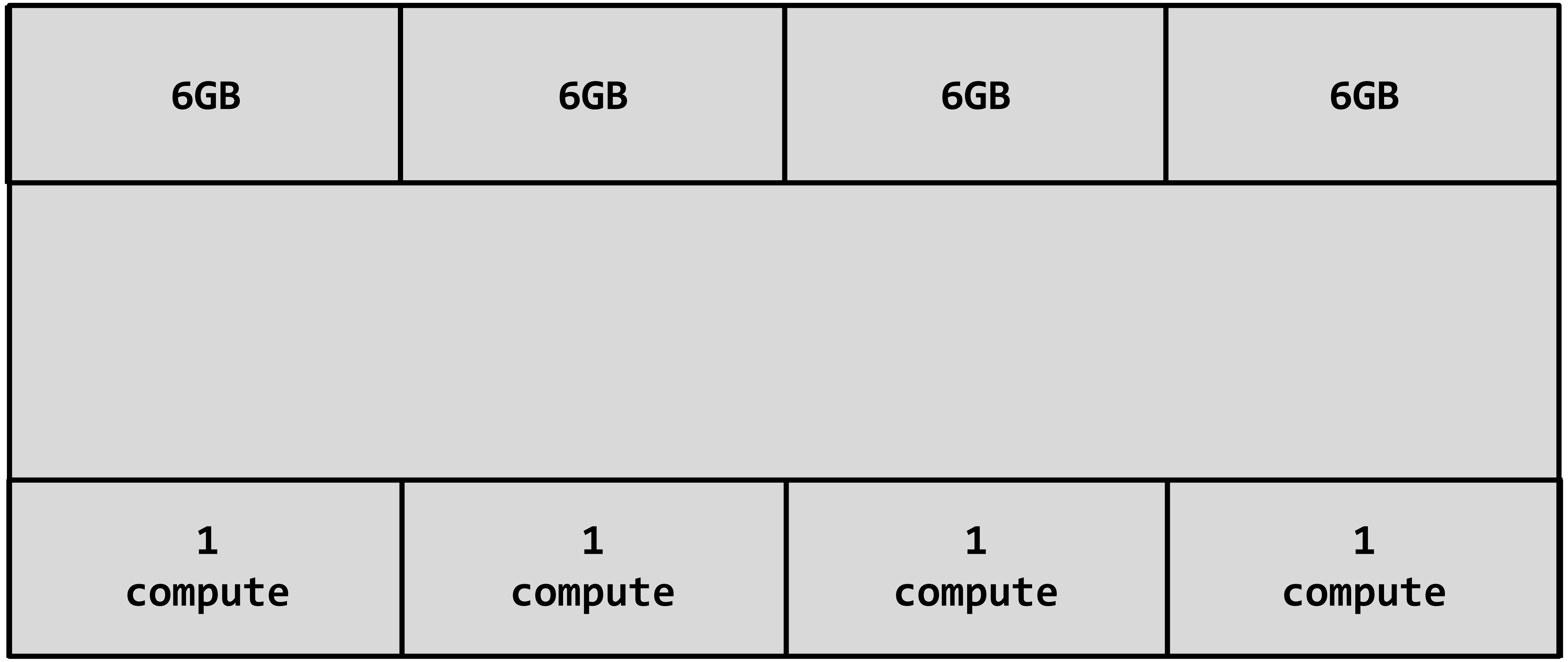


NVIDIA A100 (80GB)

- 8 x 10GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



6GB

6GB

6GB

6GB

1 compute

1 compute

1 compute

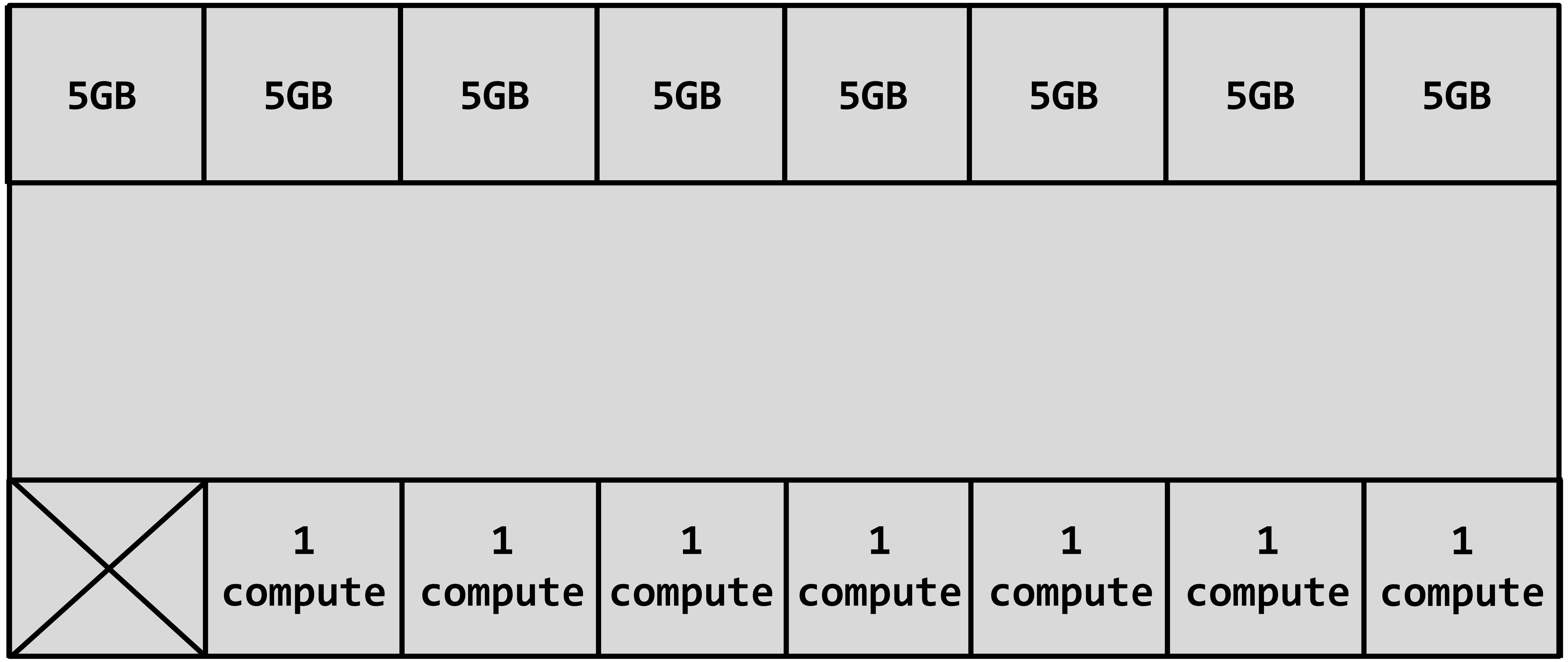
1 compute

NVIDIA A30 (24GB)

- 4 x 6GB Memory Slices
- 4 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



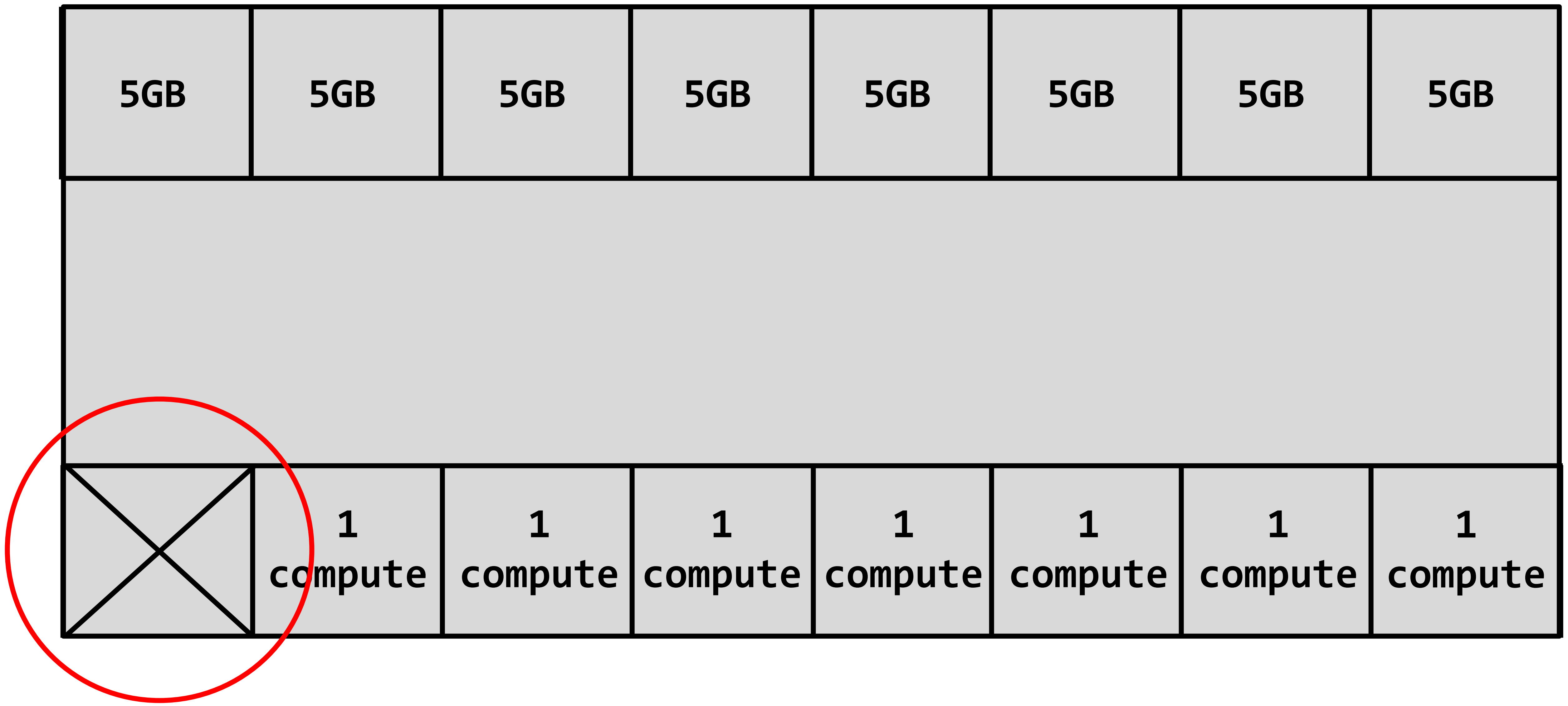
5GB	5GB	5GB	5GB	5GB	5GB	5GB	5GB
☒	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

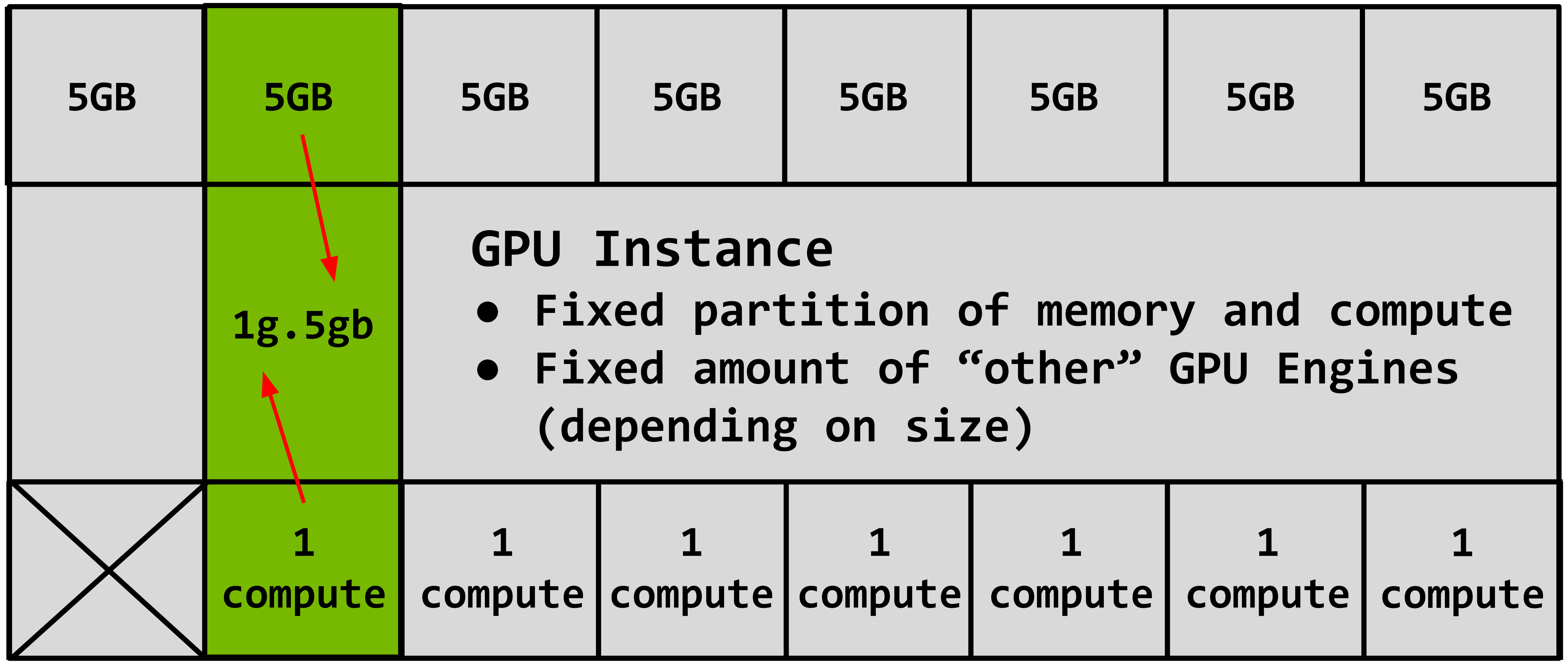


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

1g.5gb

GPU Instance

- Fixed partition of memory and compute
- Fixed amount of “other” GPU Engines (depending on size)

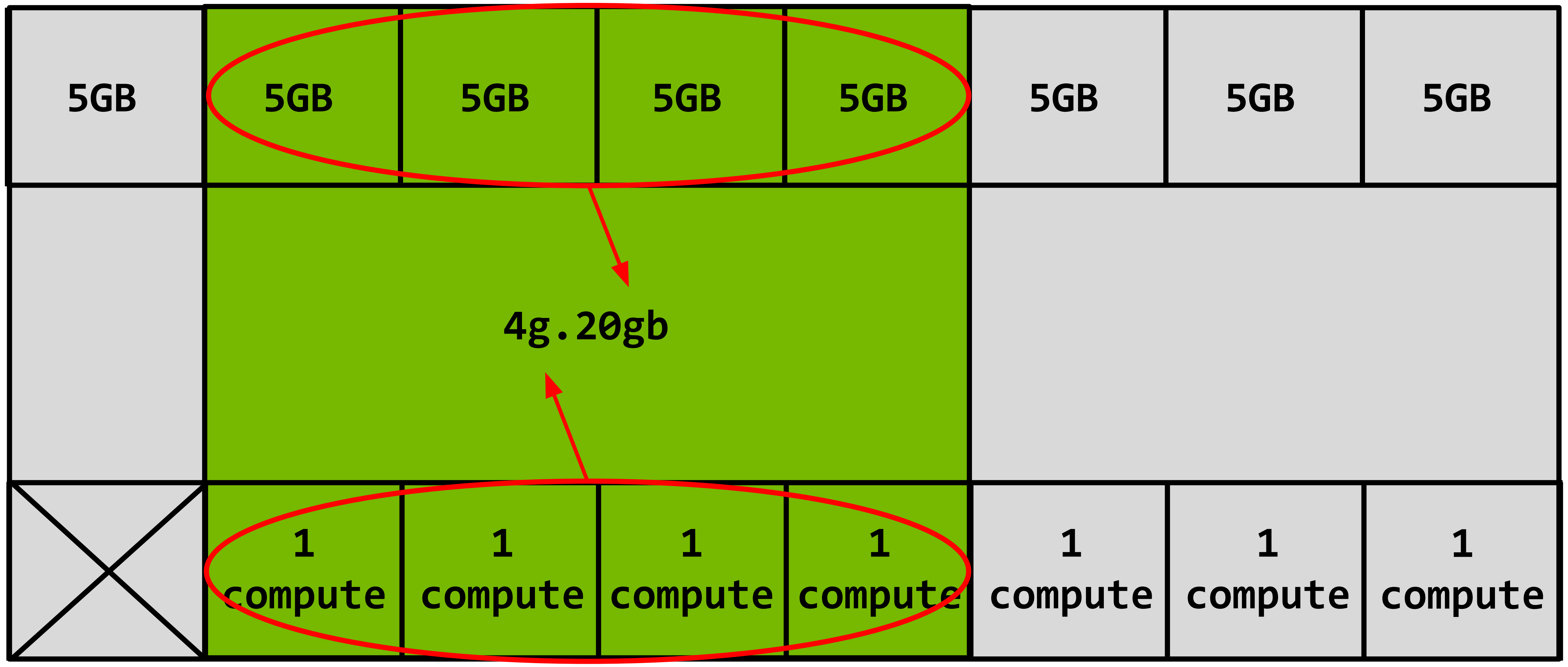
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

4g.20gb

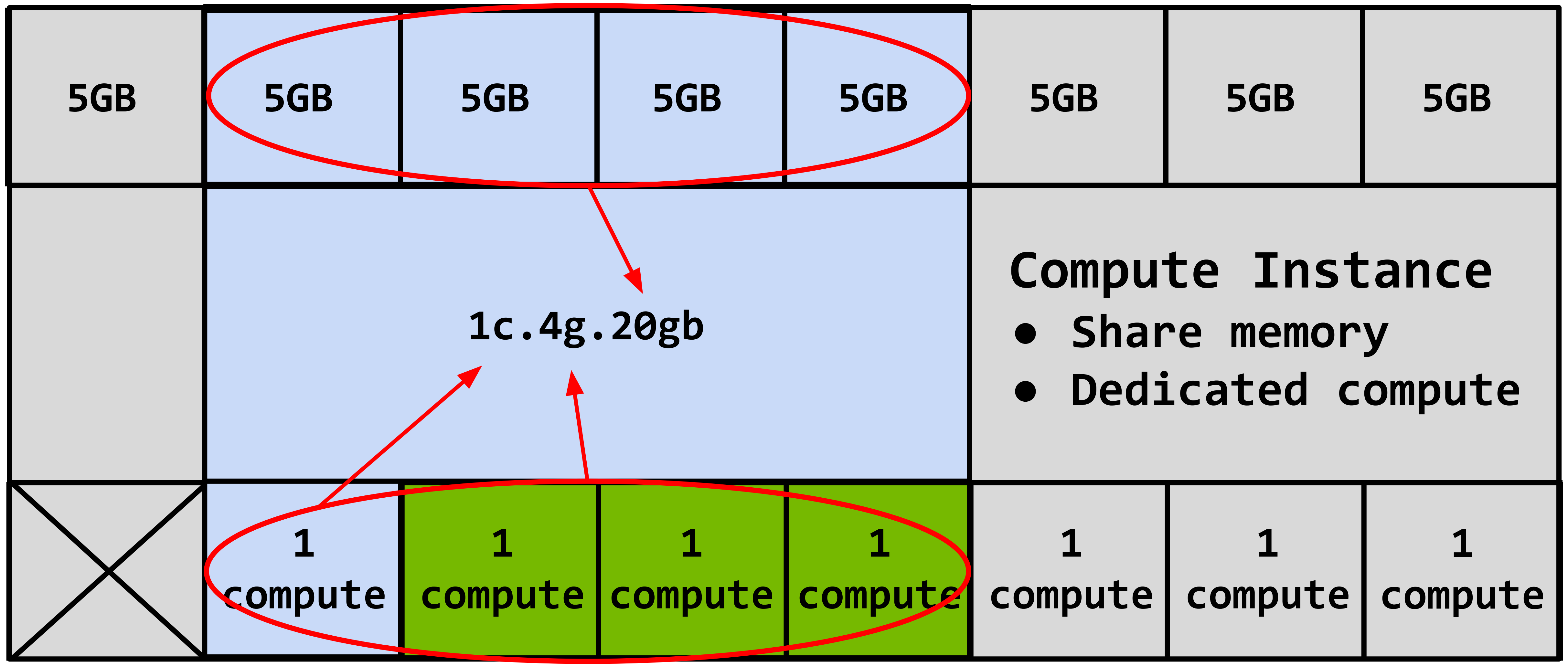
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

1c.4g.20gb

Compute Instance

- Share memory
- Dedicated compute

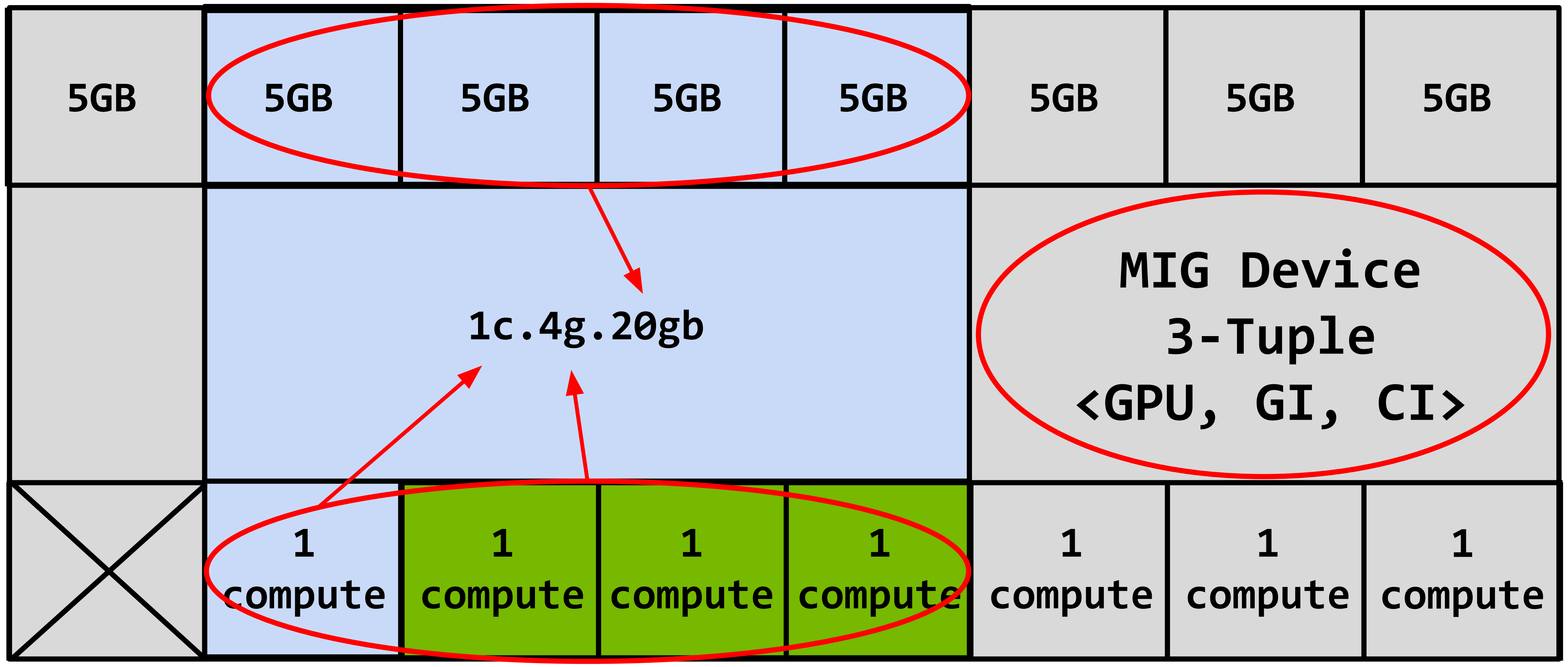
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

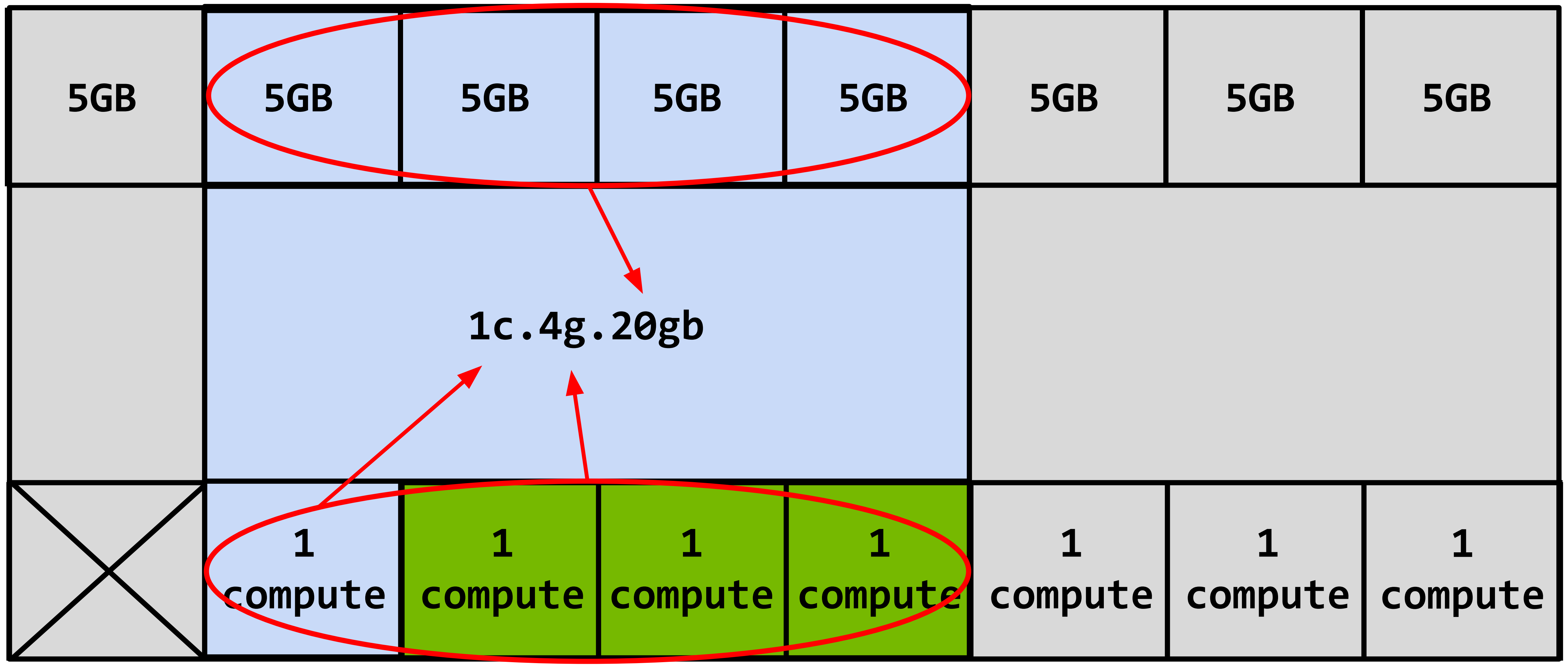


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

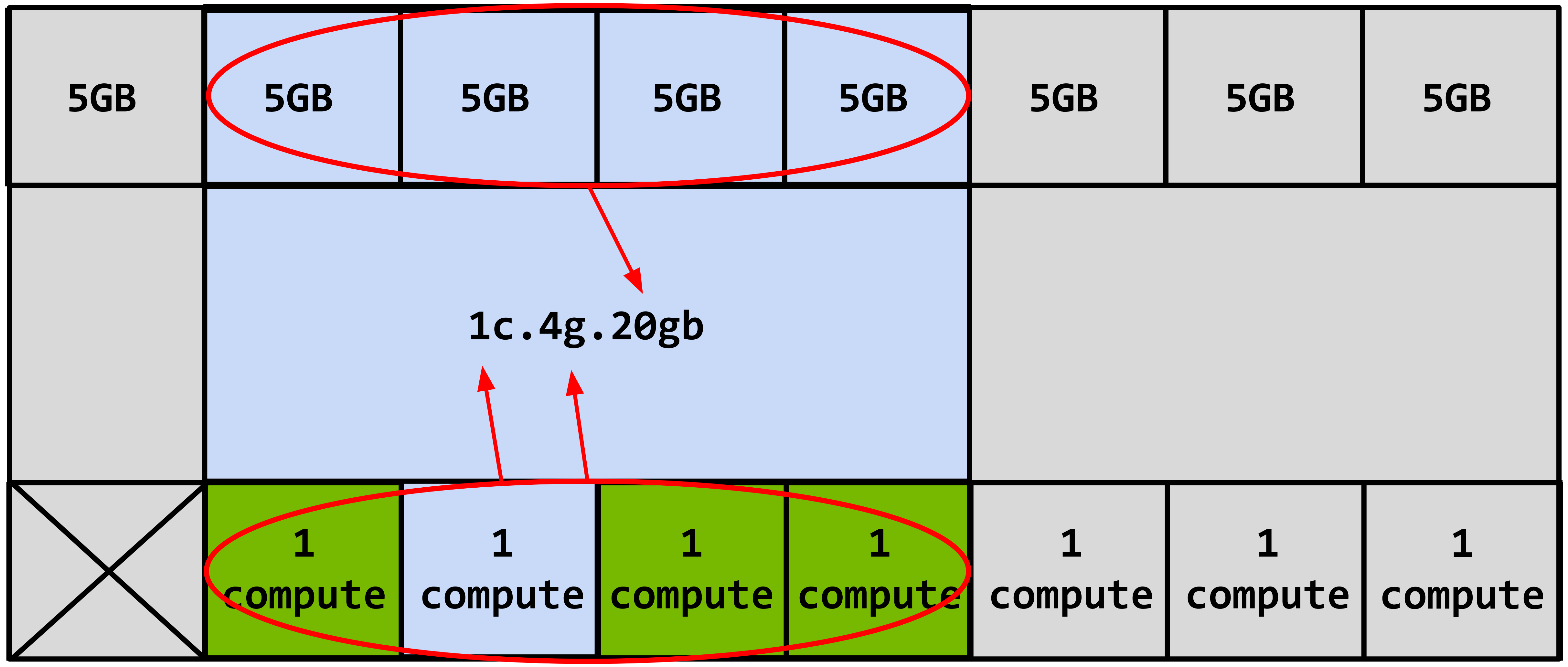


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

1c.4g.20gb

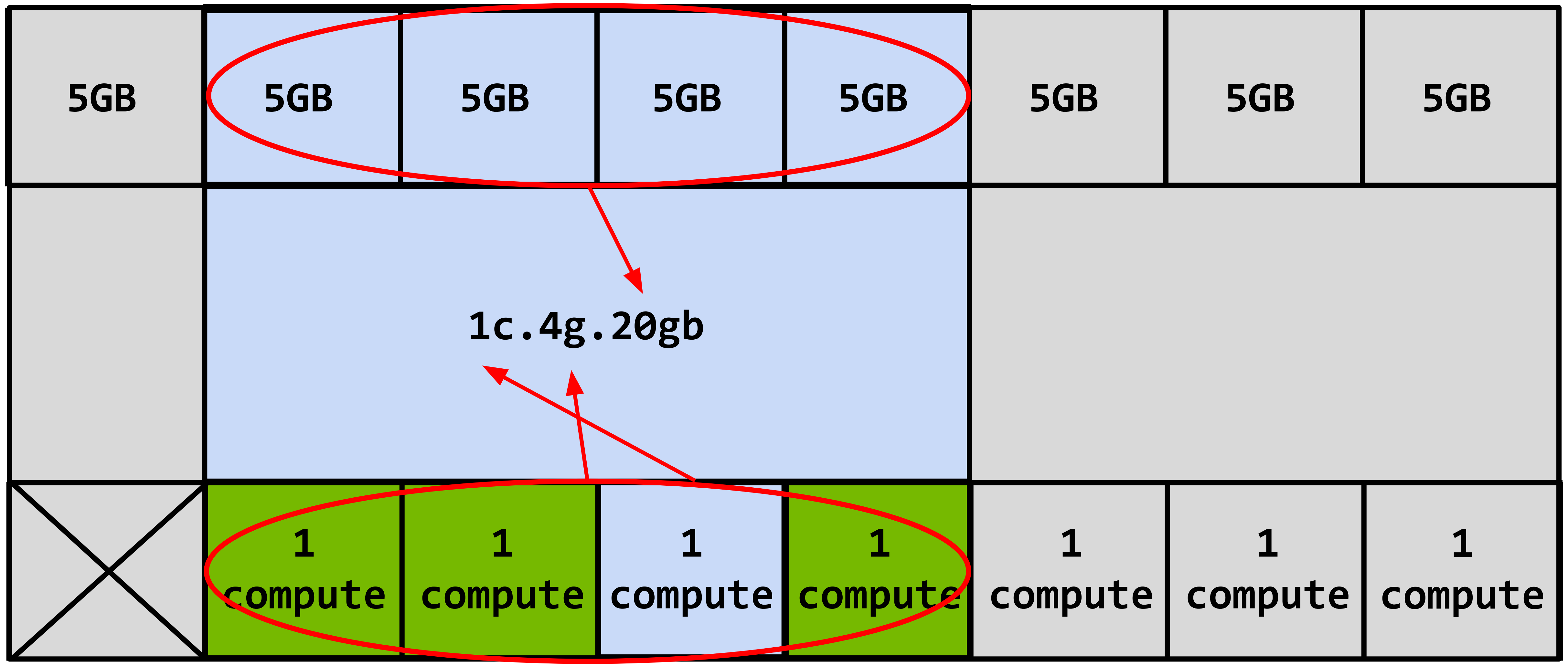
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

1c.4g.20gb

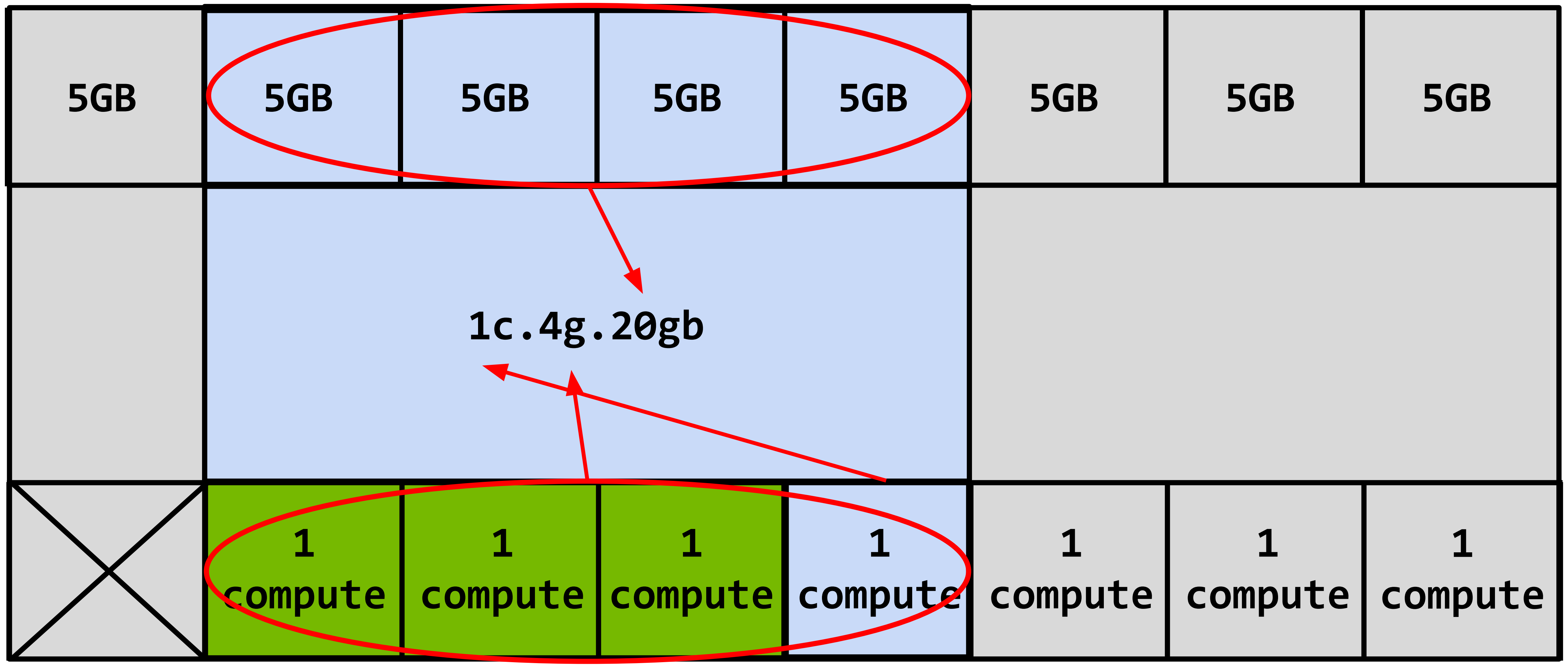
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

1c.4g.20gb

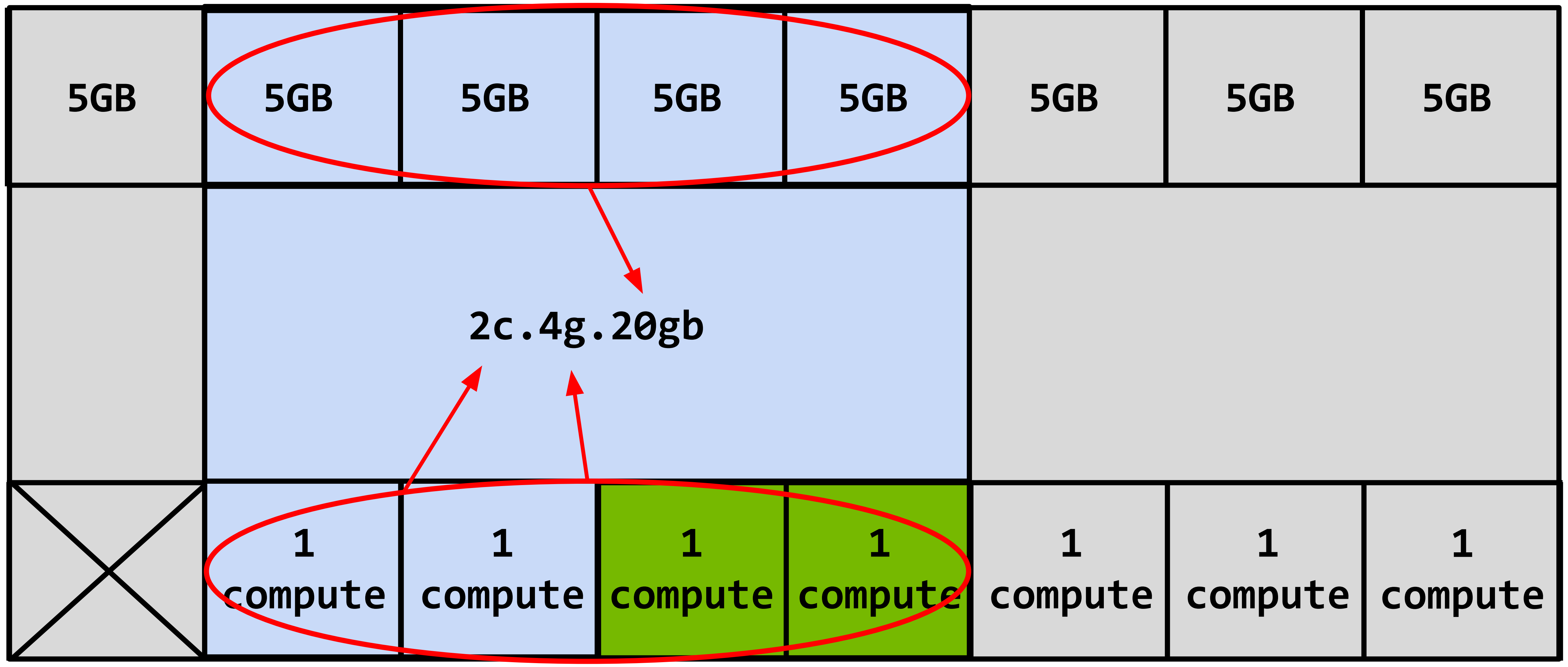
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

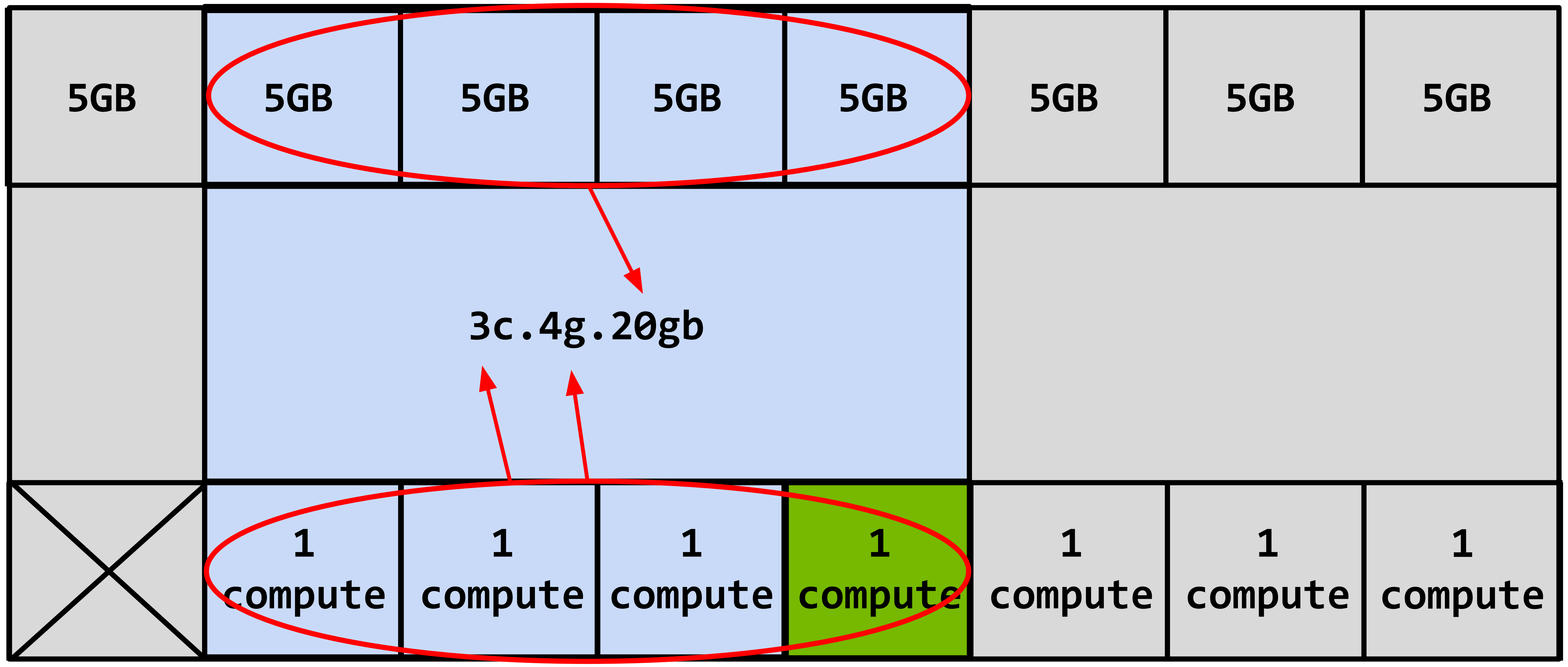


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

3c.4g.20gb

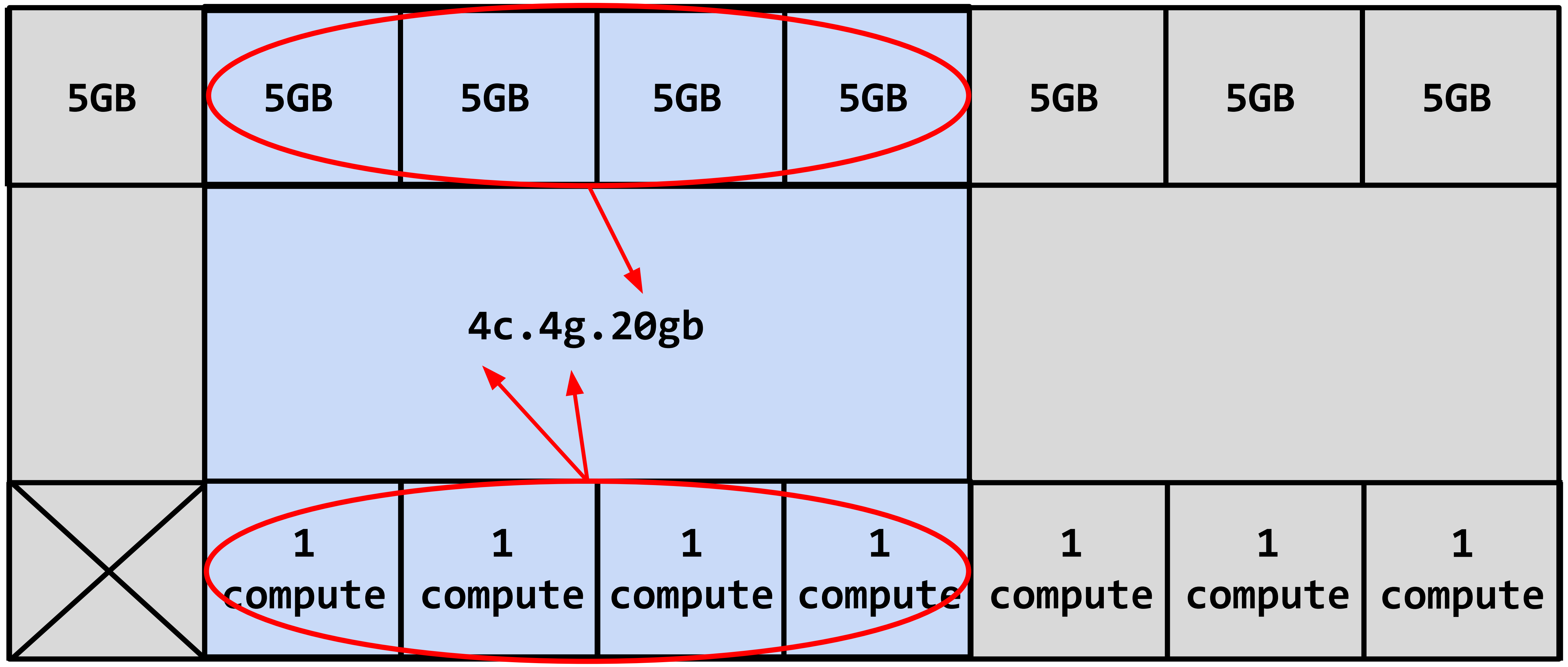
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

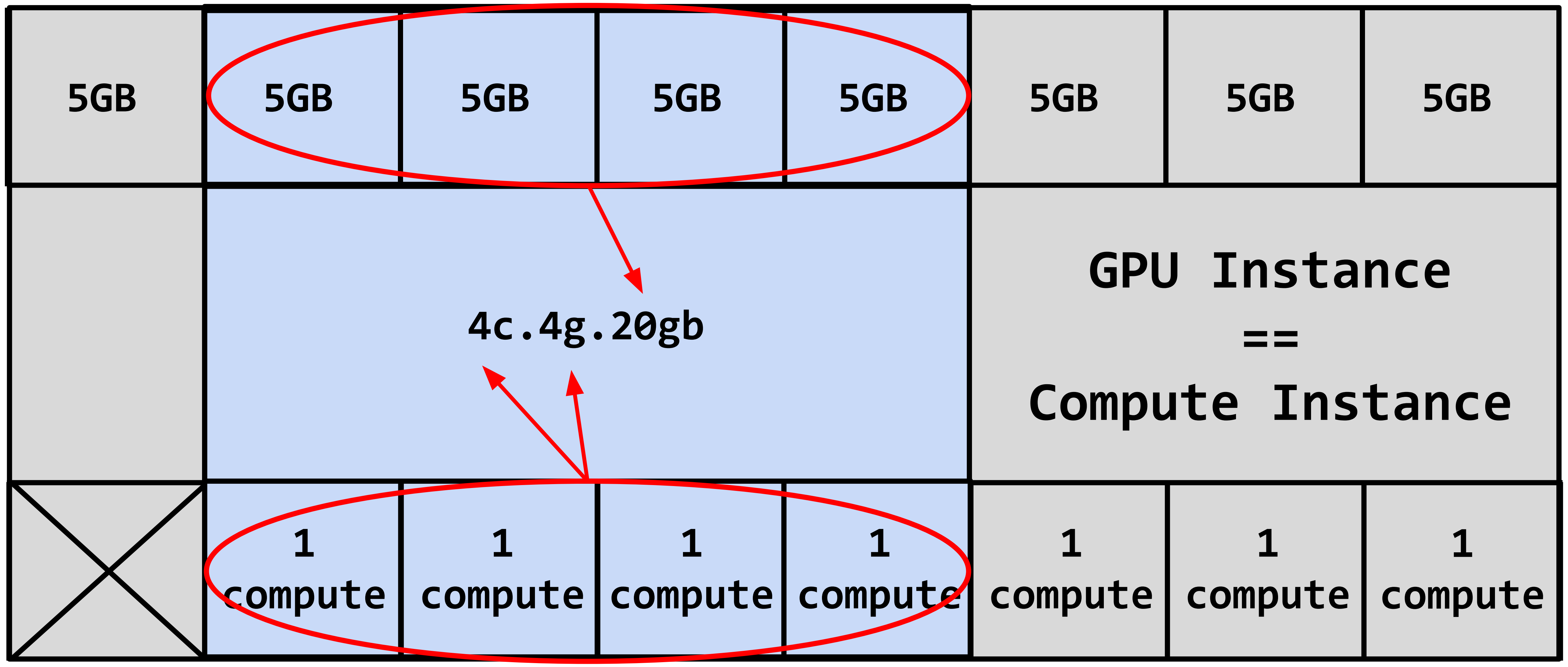


NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

4c.4g.20gb

GPU Instance
==
Compute Instance

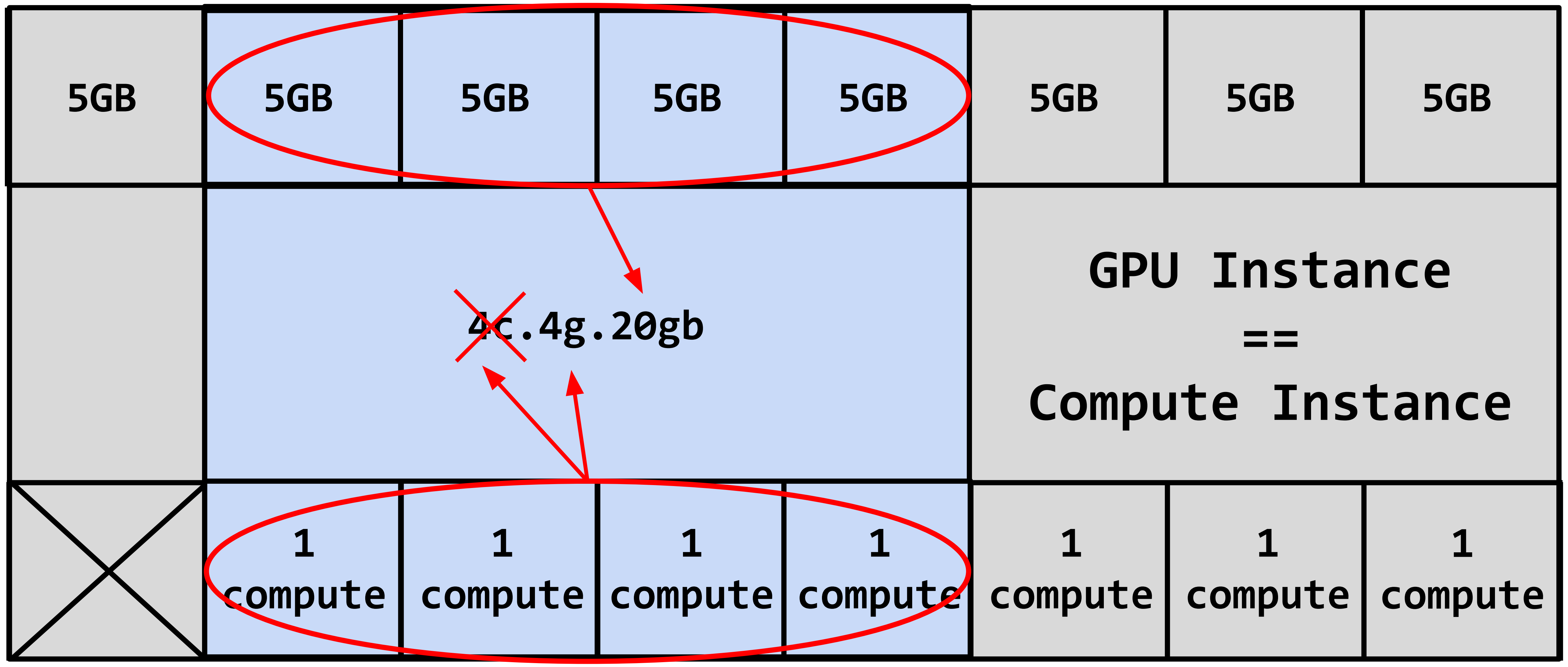
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

~~4c~~.4g.20gb

GPU Instance
==
Compute Instance

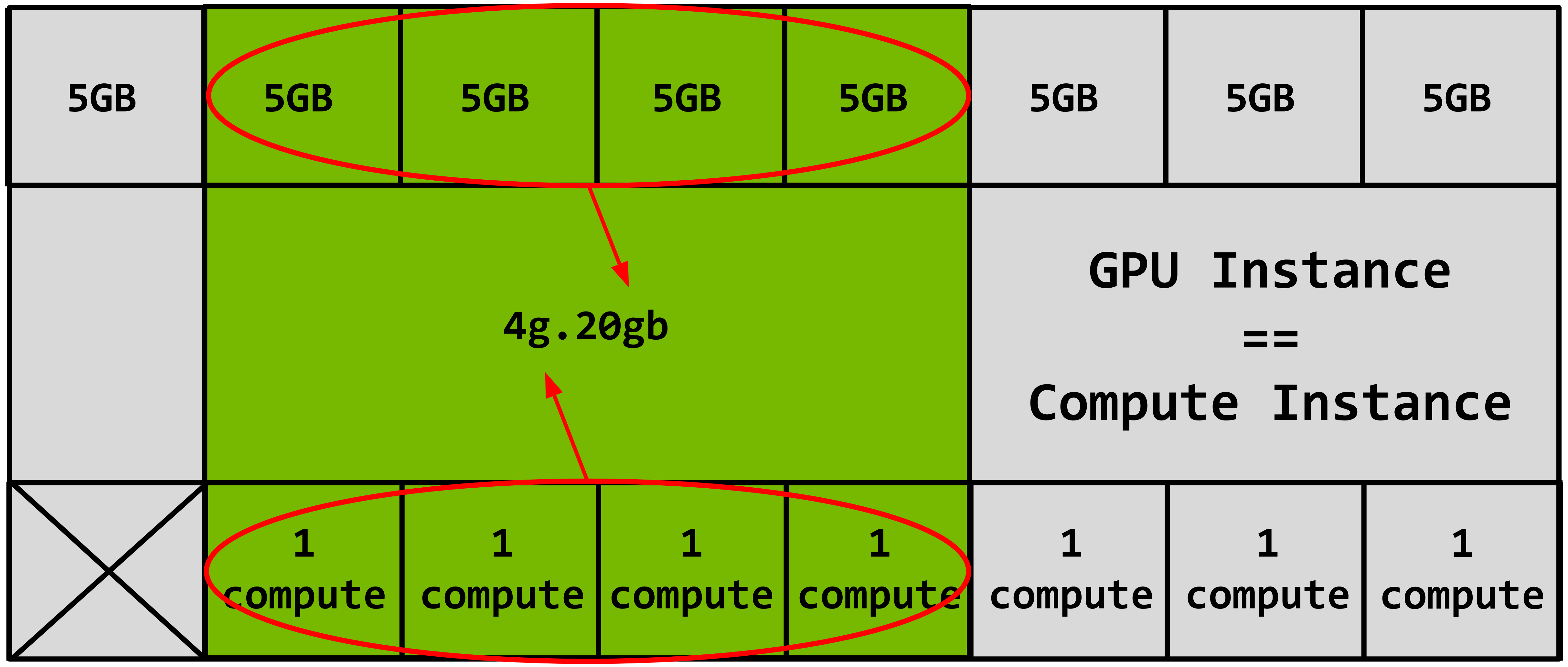
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

4g.20gb

GPU Instance
==
Compute Instance

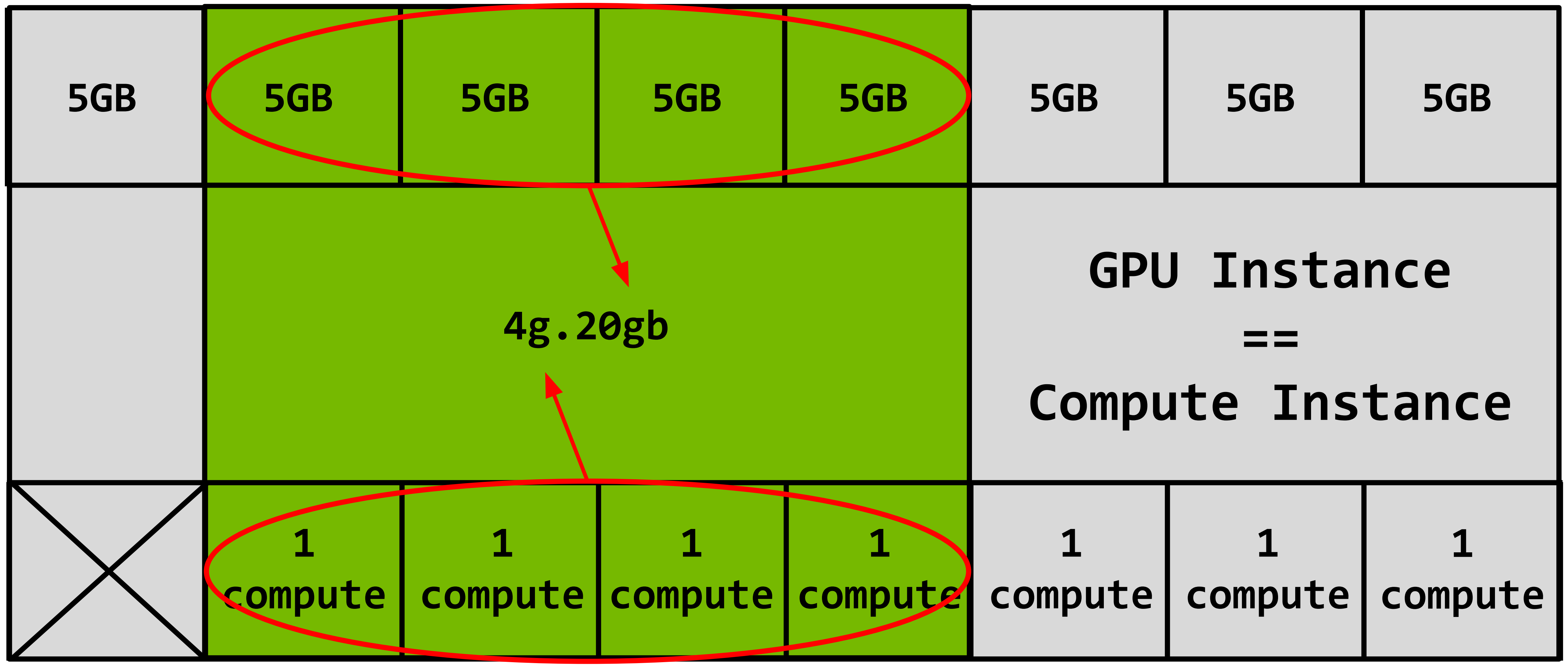
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



5GB 5GB 5GB 5GB 5GB 5GB 5GB 5GB

4g.20gb

GPU Instance
==
Compute Instance

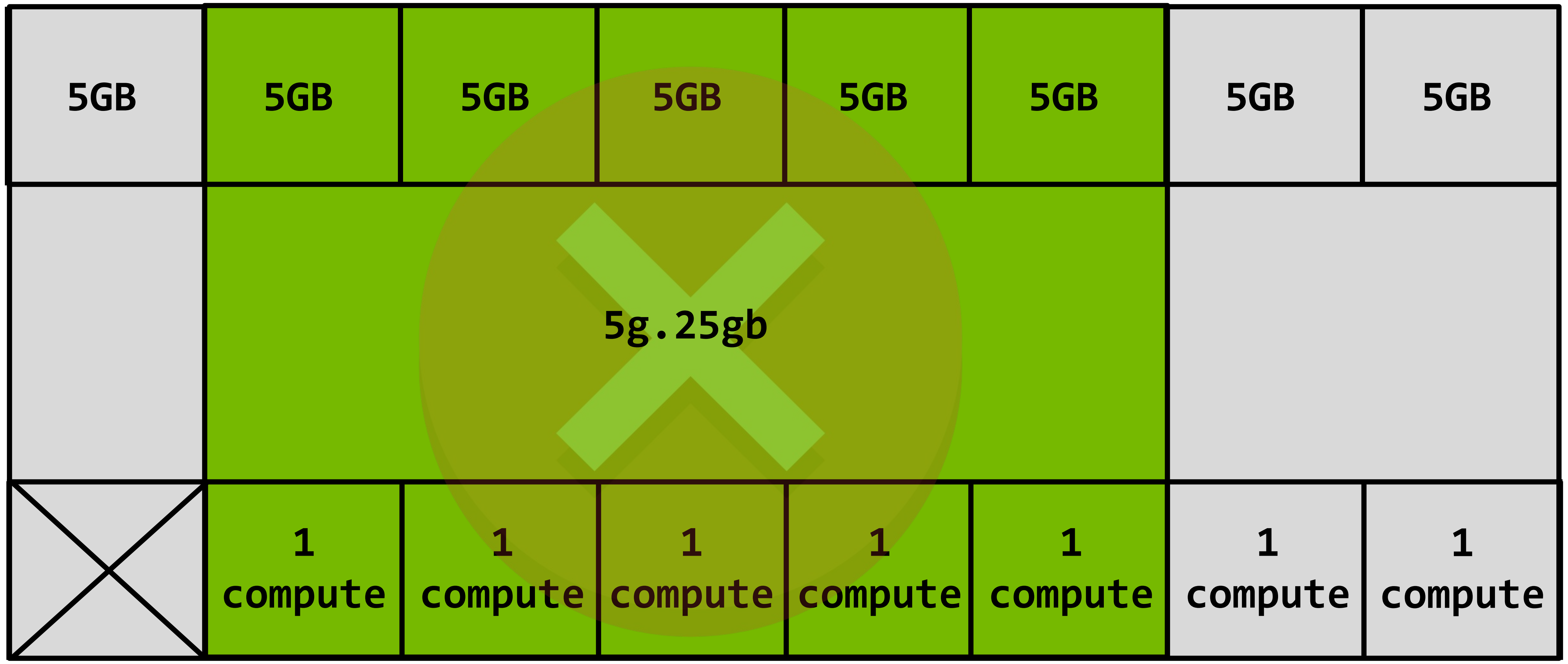
1 compute 1 compute 1 compute 1 compute 1 compute 1 compute 1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



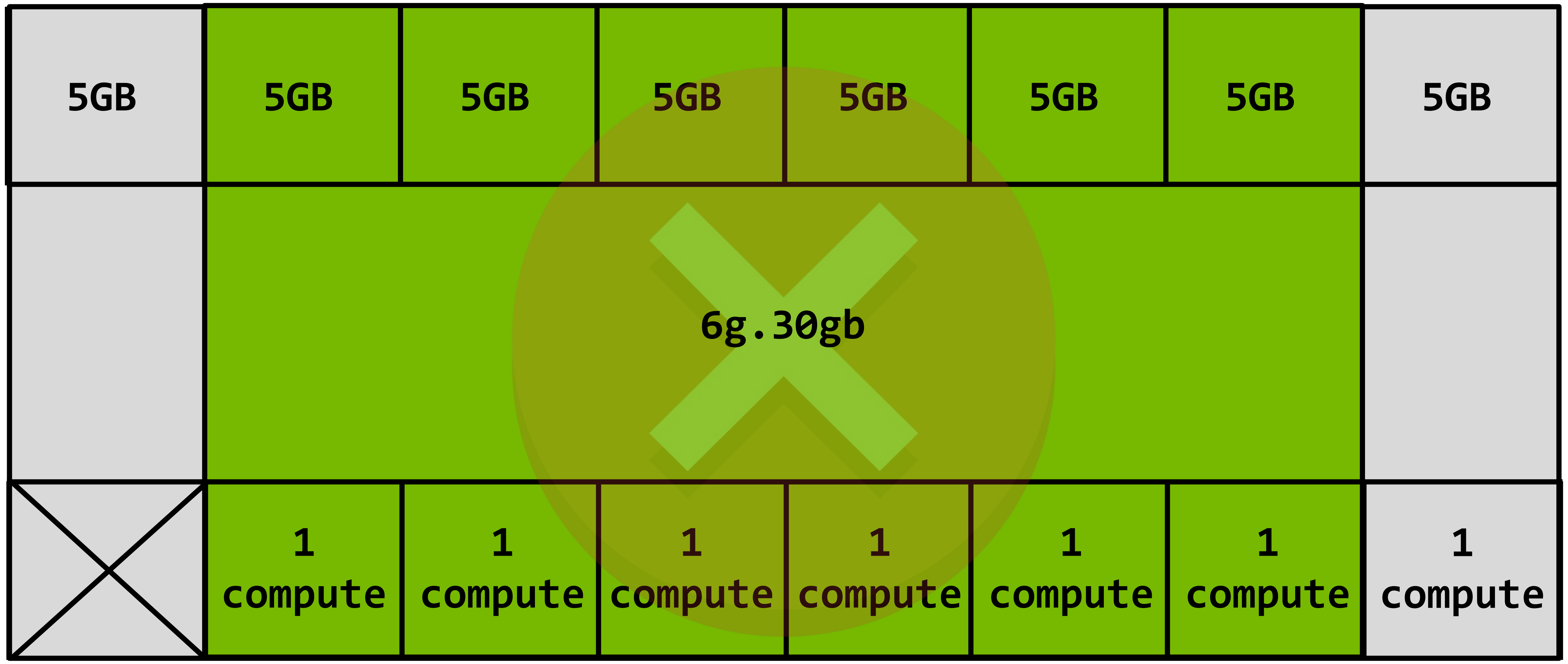
5GB	5GB	5GB	5GB	5GB	5GB	5GB	5GB
	5g.25gb						
☒	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



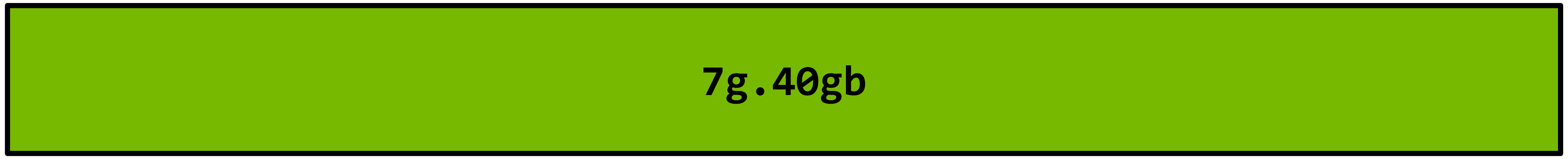
5GB	5GB	5GB	5GB	5GB	5GB	5GB	5GB
	6g.30gb						
☒	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute	1 compute

NVIDIA A100 (40GB)

- 8 x 5GB Memory Slices
- 7 Compute Slices

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

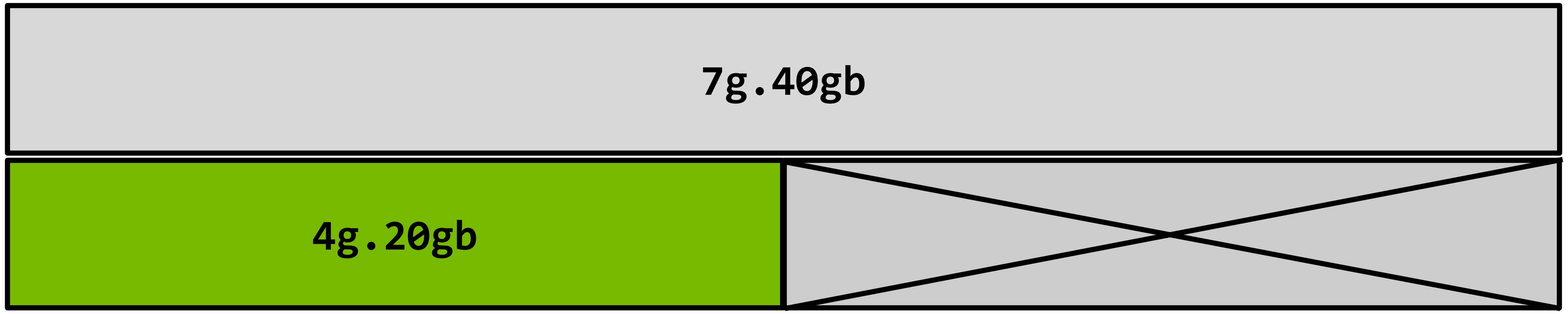


7g.40gb

- `1 x 7g.40gb`

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



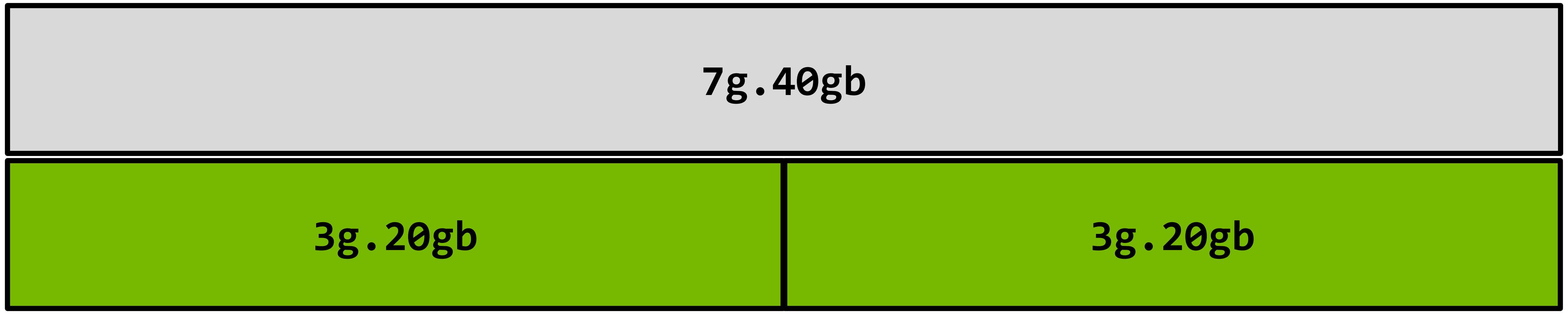
- 1 x 7g.40gb

  or

- 1 x 4g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



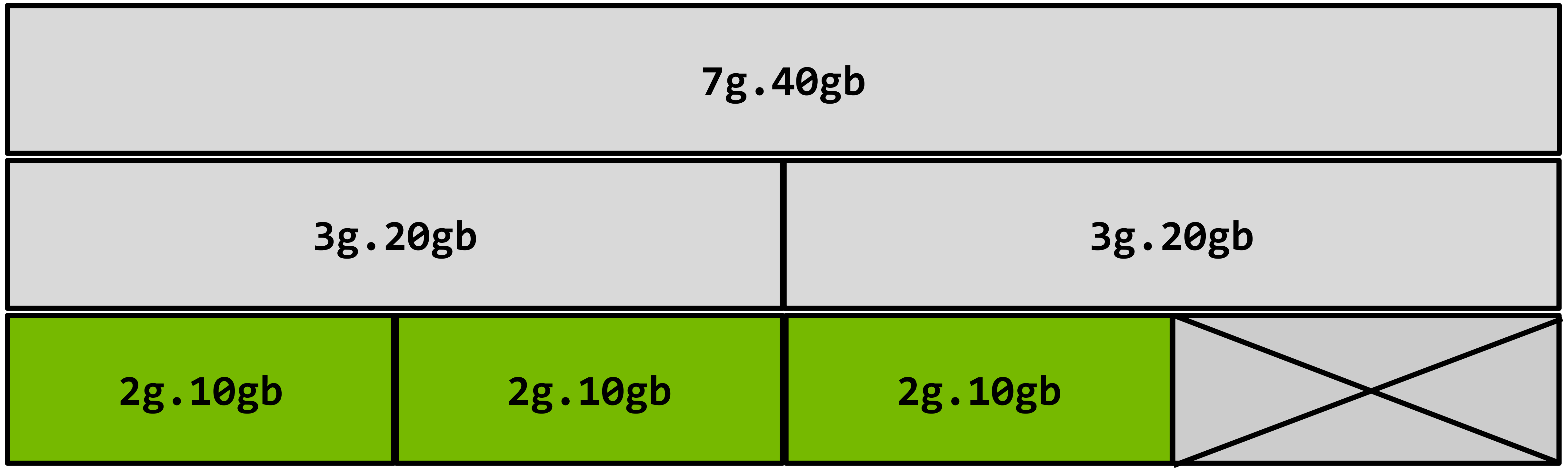
- 1 x 7g.40gb

  or

- 2 x 3g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



- 1 x 7g.40gb

or

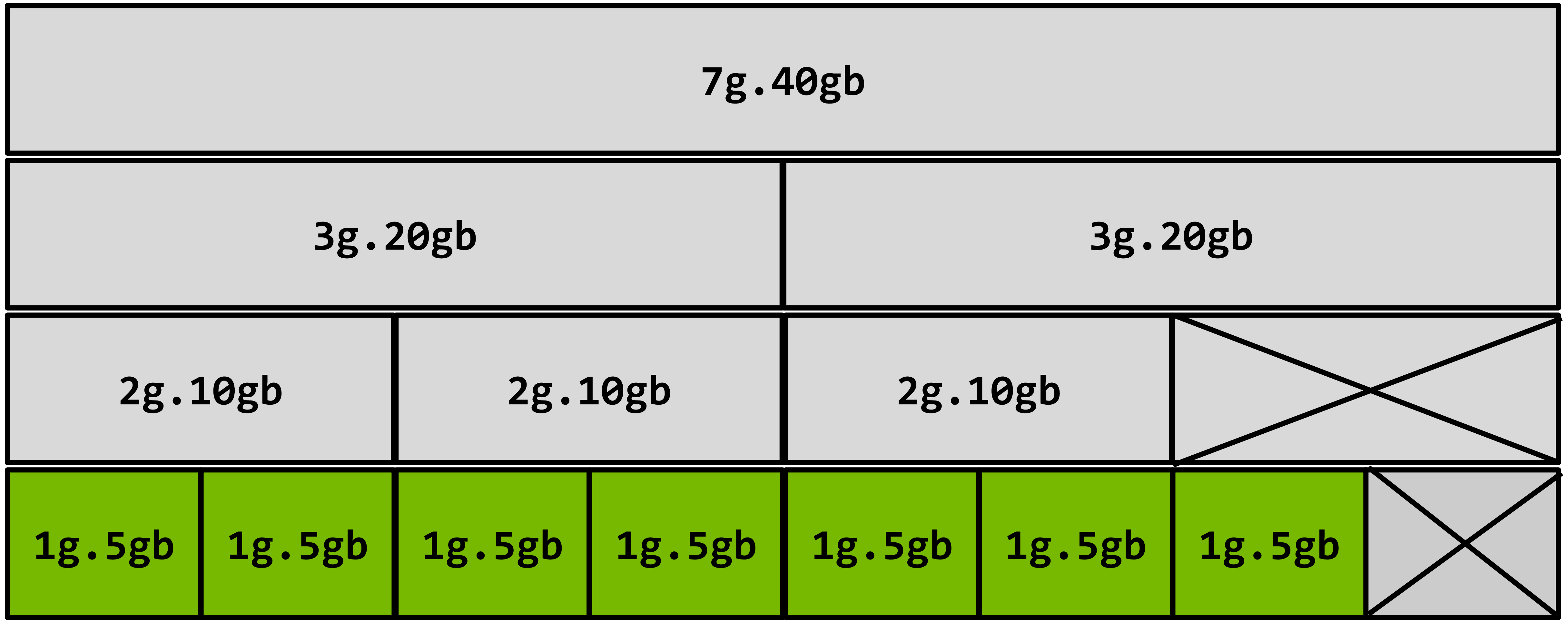
- 2 x 3g.20gb

or

- 3 x 2g.10gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



7g.40gb

3g.20gb 3g.20gb

2g.10gb 2g.10gb 2g.10gb

1g.5gb 1g.5gb 1g.5gb 1g.5gb 1g.5gb 1g.5gb 1g.5gb

- 1 x 7g.40gb

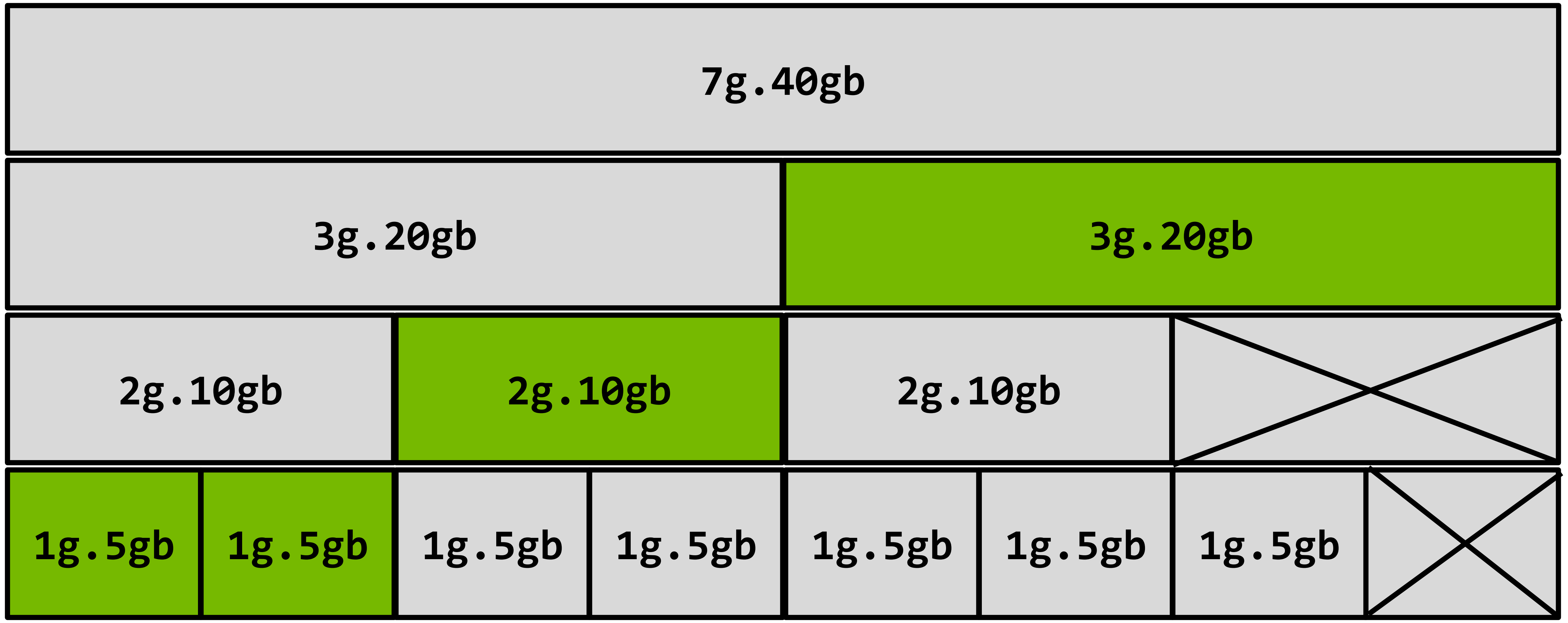
  or
- 2 x 3g.20gb

  or
- 3 x 2g.10gb

  or
- 7 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



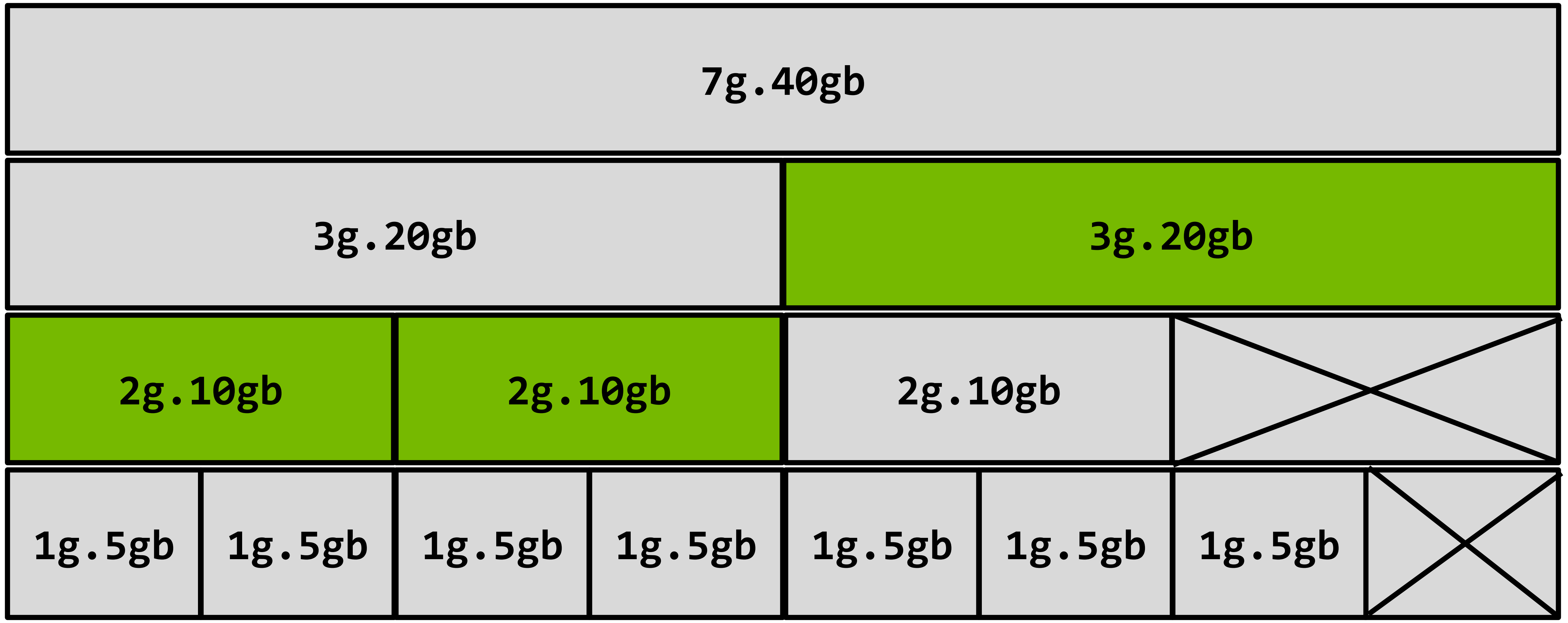
- 1 x 3g.20gb

  and
- 1 x 2g.10gb

  and
- 2 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



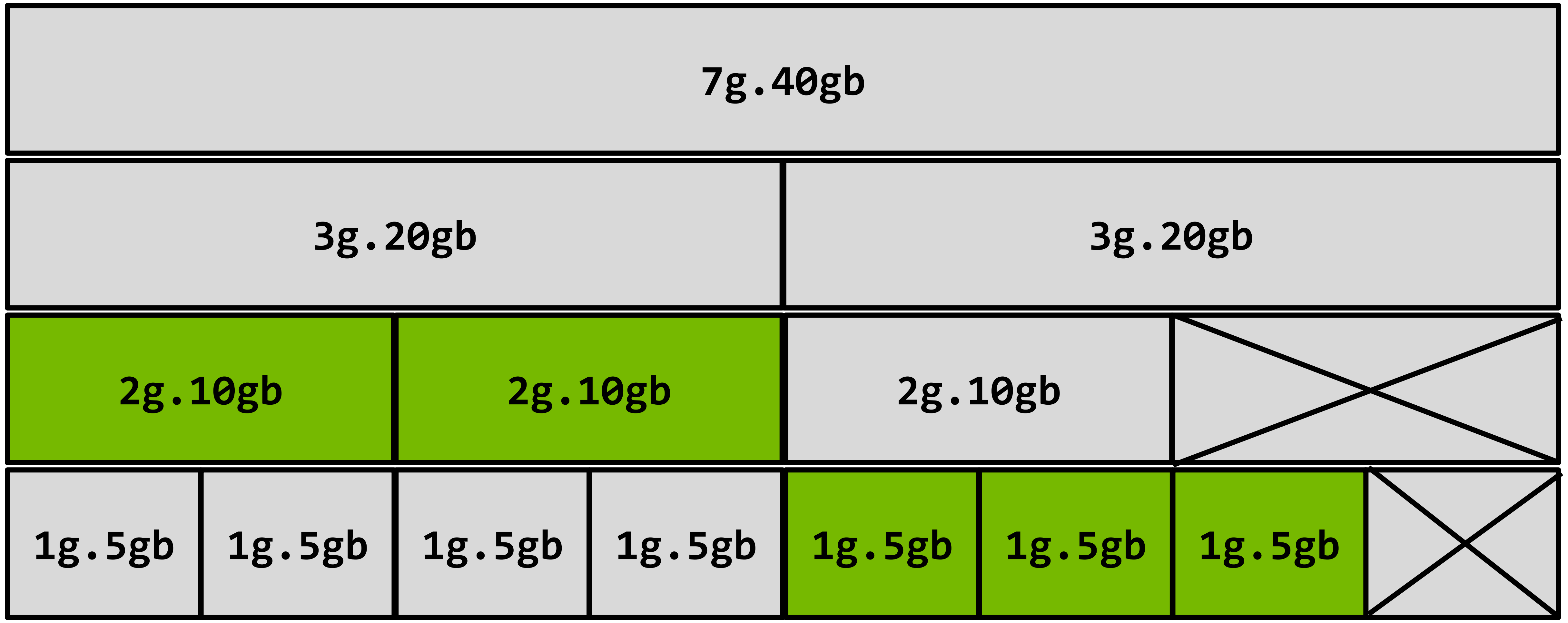
- 1 x 3g.20gb

  and

- 2 x 2g.10gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



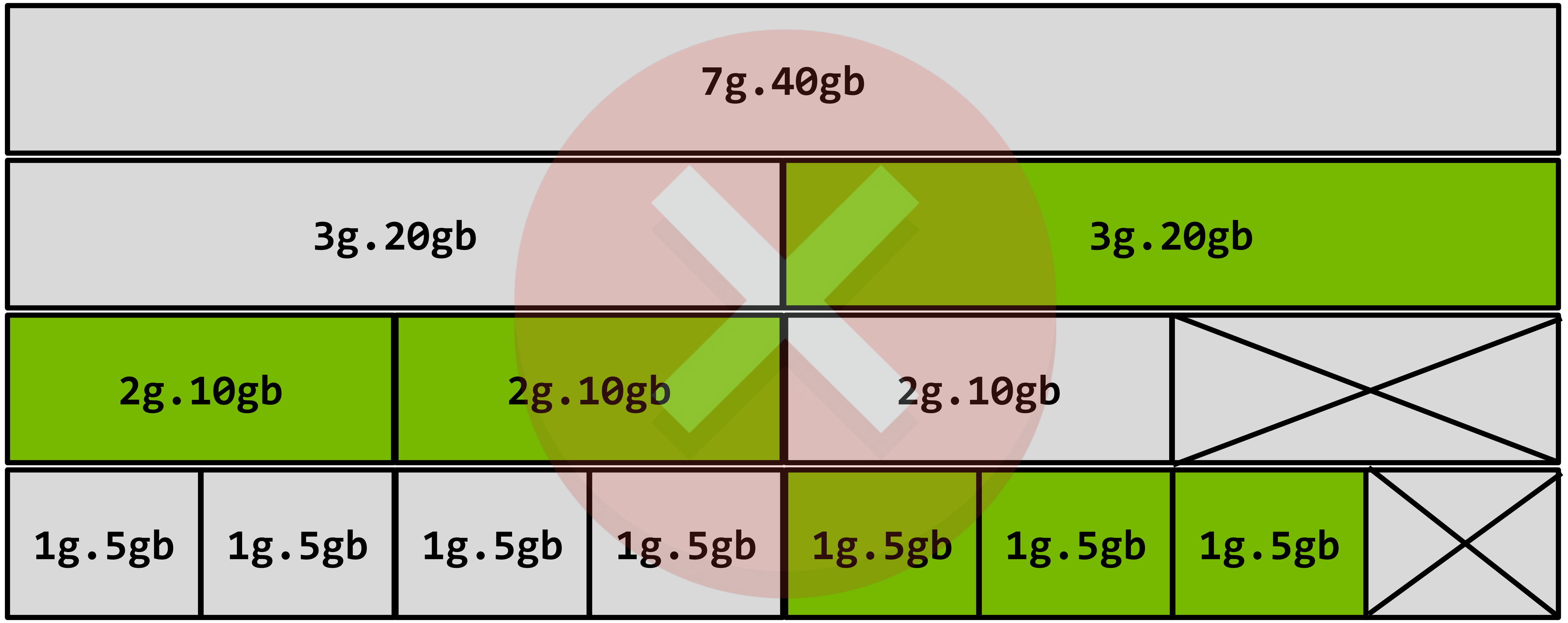
- 2 x 2g.10gb

and

- 3 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



7g.40gb

3g.20gb | 3g.20gb

2g.10gb | 2g.10gb | 2g.10gb

1g.5gb | 1g.5gb | 1g.5gb | 1g.5gb | 1g.5gb | 1g.5gb | 1g.5gb

- 1 x 3g.30gb

  and

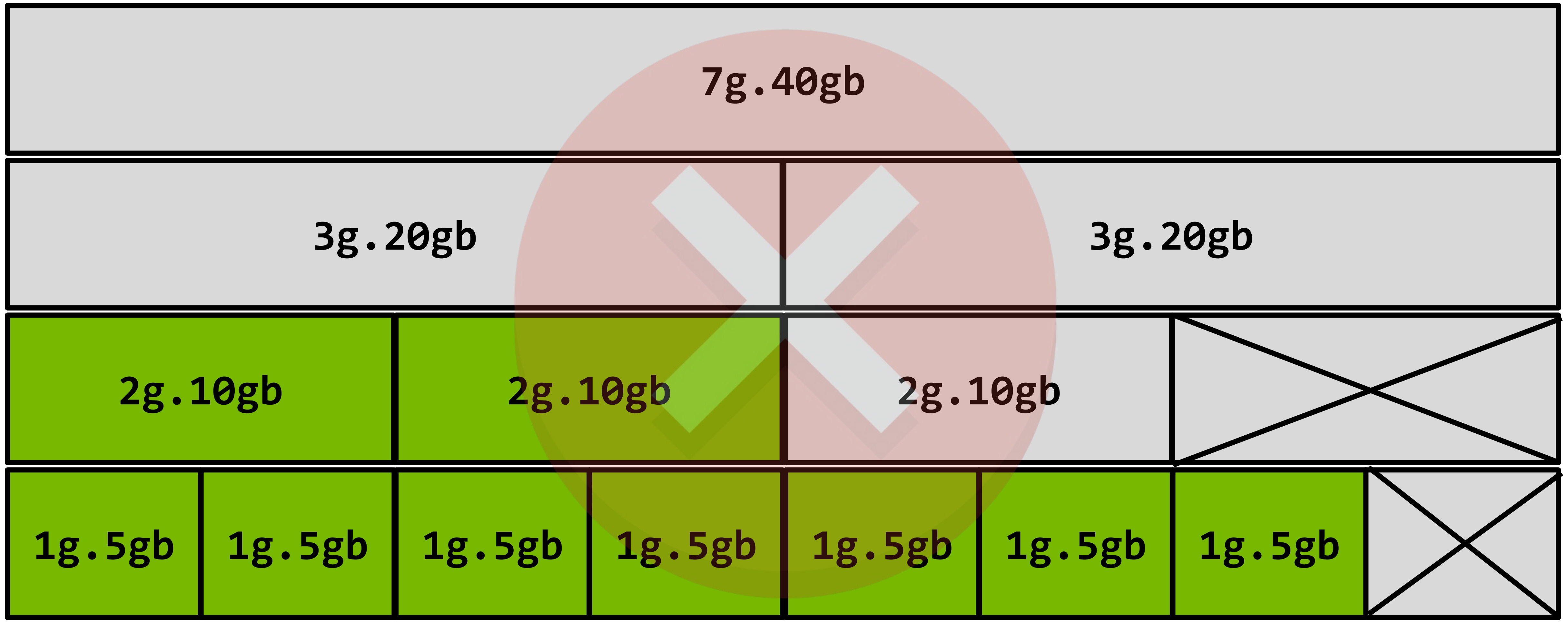
- 2 x 2g.10gb

  and

- 3 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



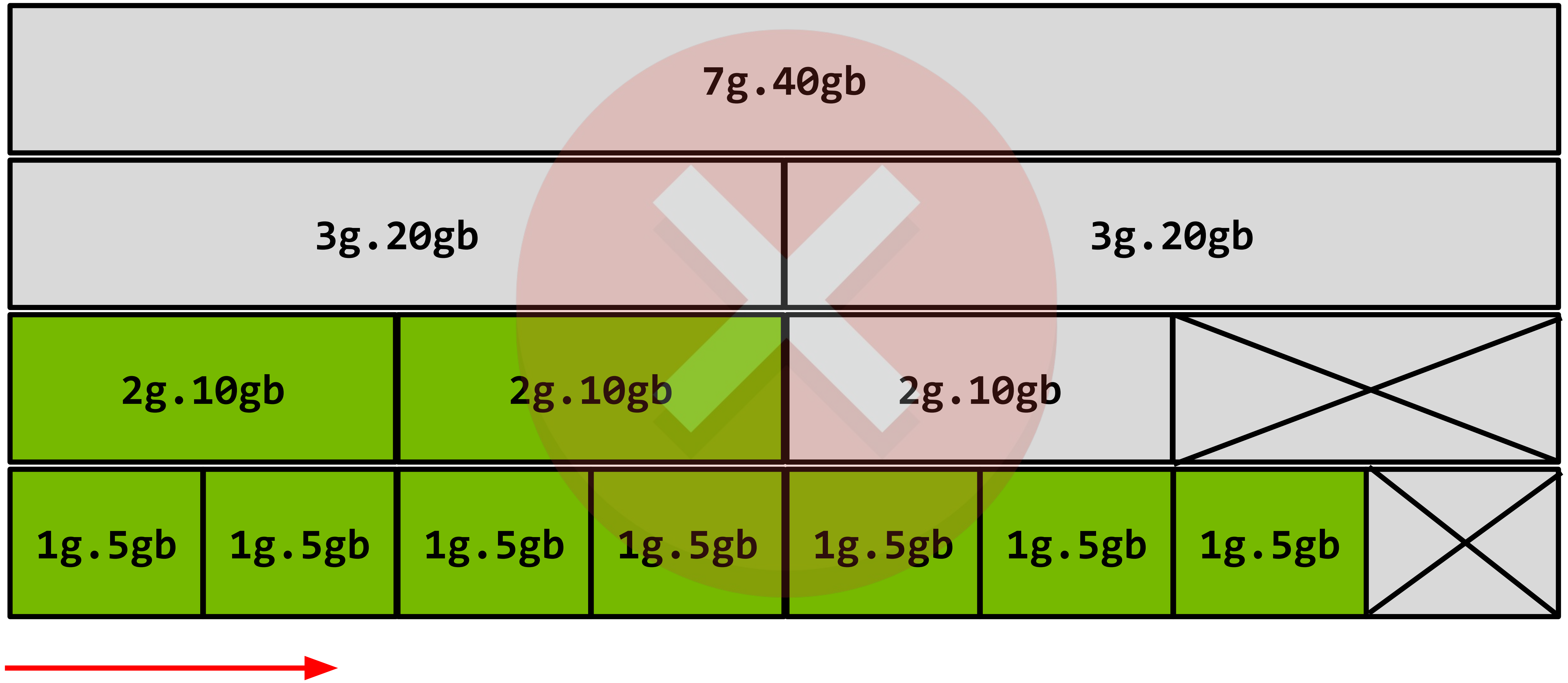
- 2 x 2g.10gb

and

- 7 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



No Overlapping Verticals

- 2 x 2g.10gb

and

- 7 x 1g.5gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

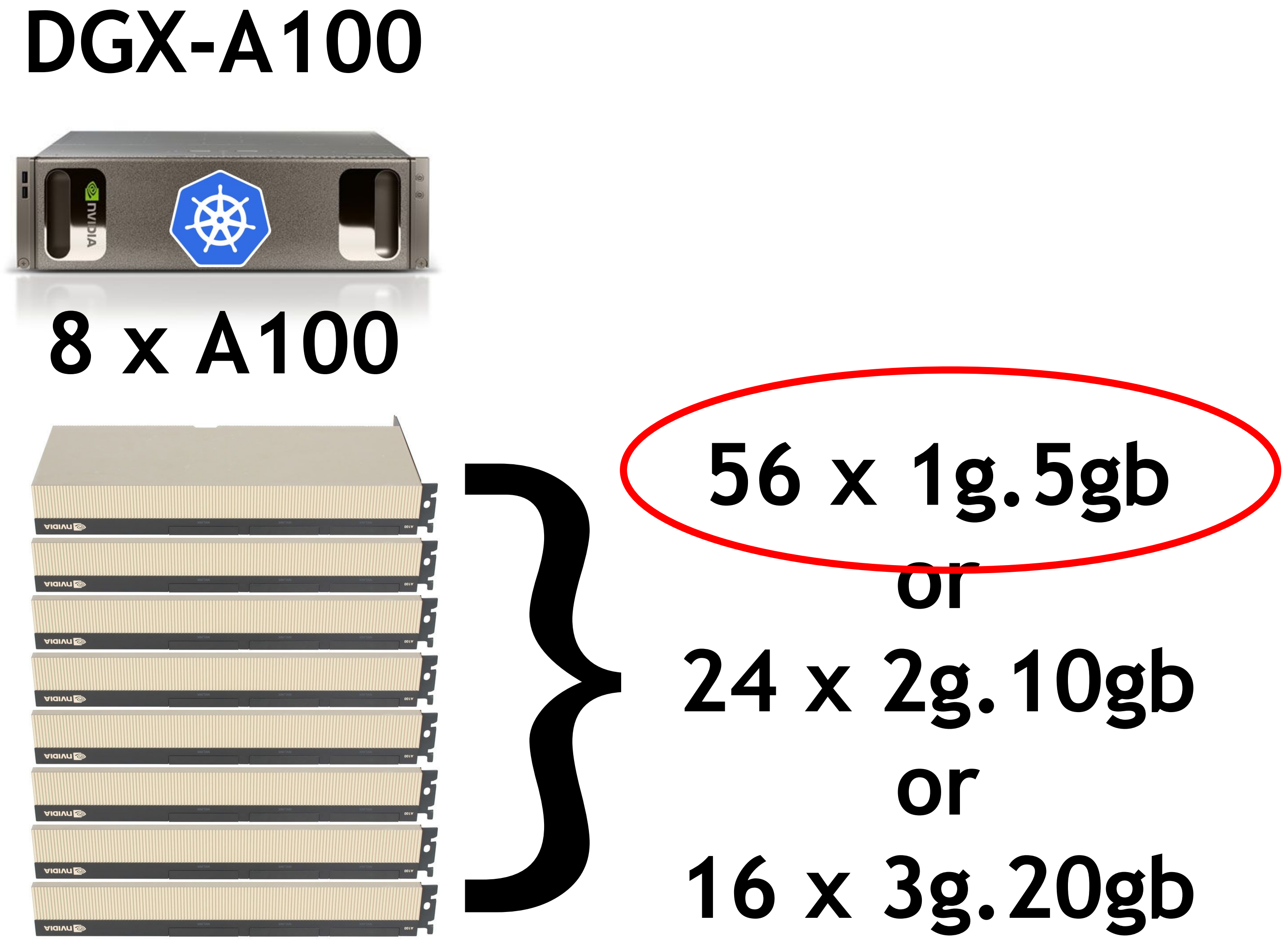
DGX-A100

8 x A100

56 x 1g.5gb
or
24 x 2g.10gb
or
16 x 3g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices



DGX-A100

8 x A100

56 x 1g.5gb
or
24 x 2g.10gb
or
16 x 3g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

DGX-A100

8 x A100

56 x 1g.5gb

or

24 x 2g.10gb

or

16 x 3g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

DGX-A100

8 x A100

56 x 1g.5gb

or

24 x 2g.10gb

or

[✓ 16 x 3g.20gb]

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

DGX-A100

8 x A100

16 x 1g.5gb
and
8 x 2g.10gb
and
8 x 3g.20gb

MULTI-INSTANCE GPUs (MIG)

GPU Instances, Compute Instances, and MIG Devices

DGX-A100

8 x A100

4 x Full A100s

and

8 x 1g.5gb

and

4 x 2g.10gb

and

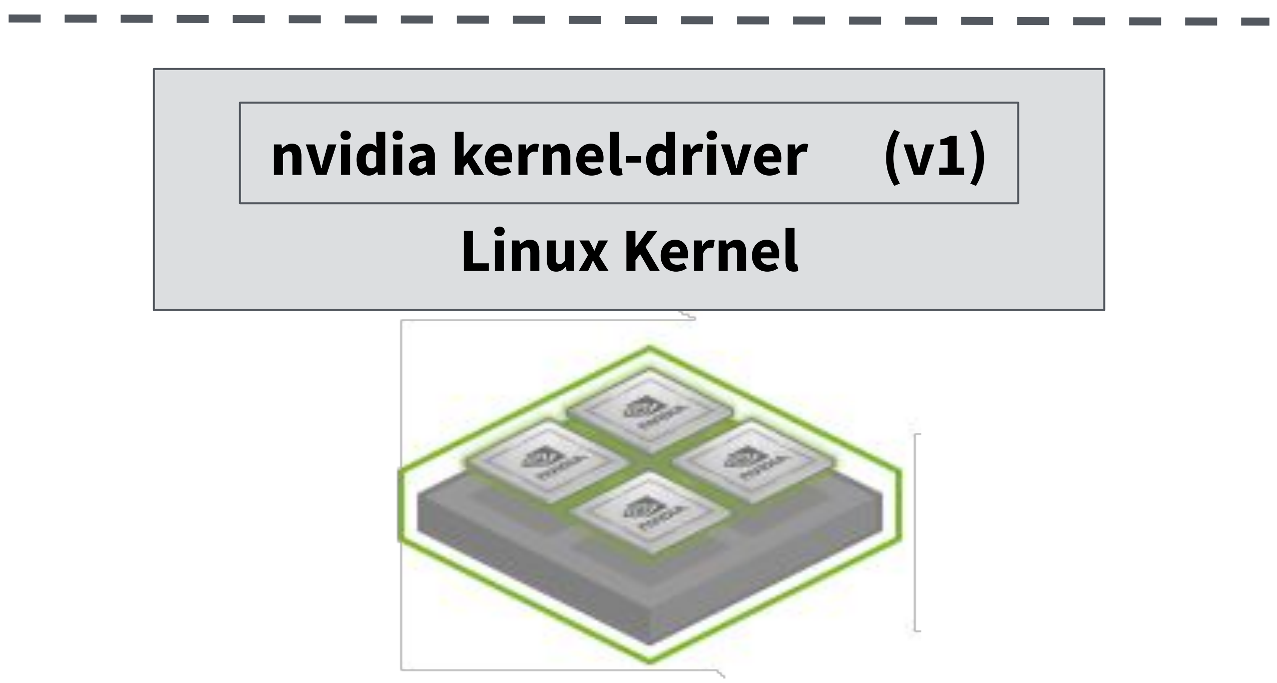
4 x 3g.20gb

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

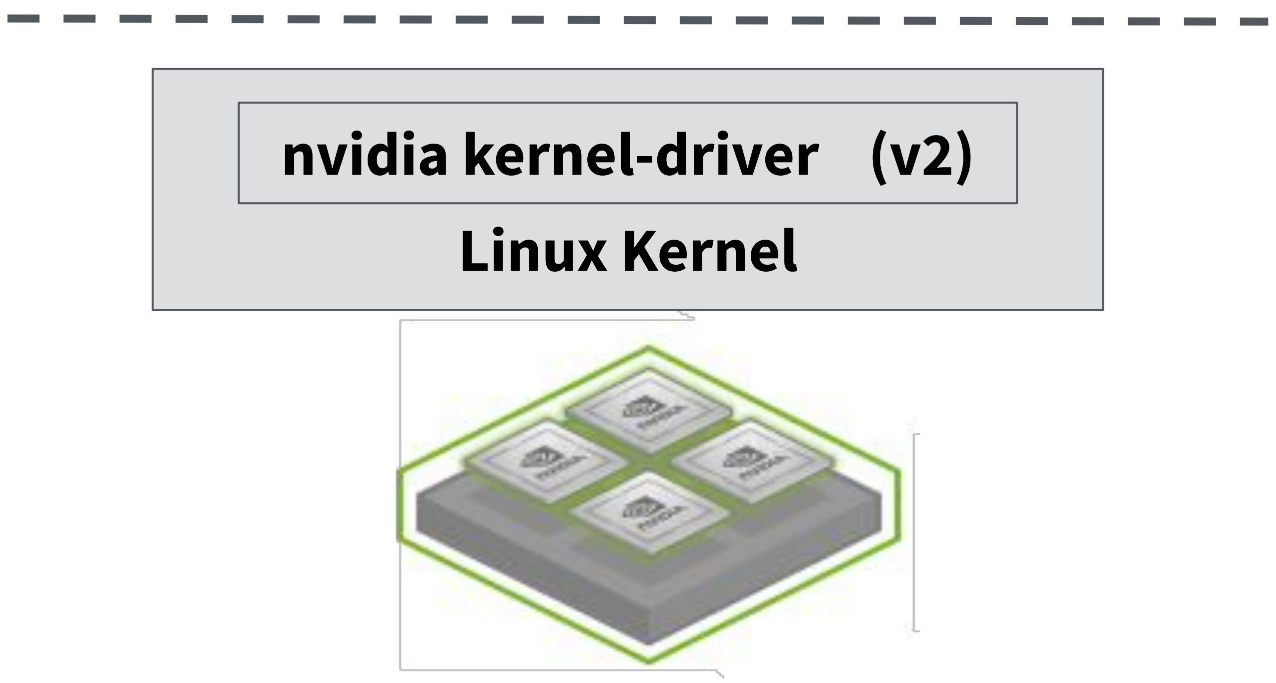
GPUs AND CONTAINERS

The NVIDIA Container Toolkit



nvidia kernel-driver (v1)

Linux Kernel

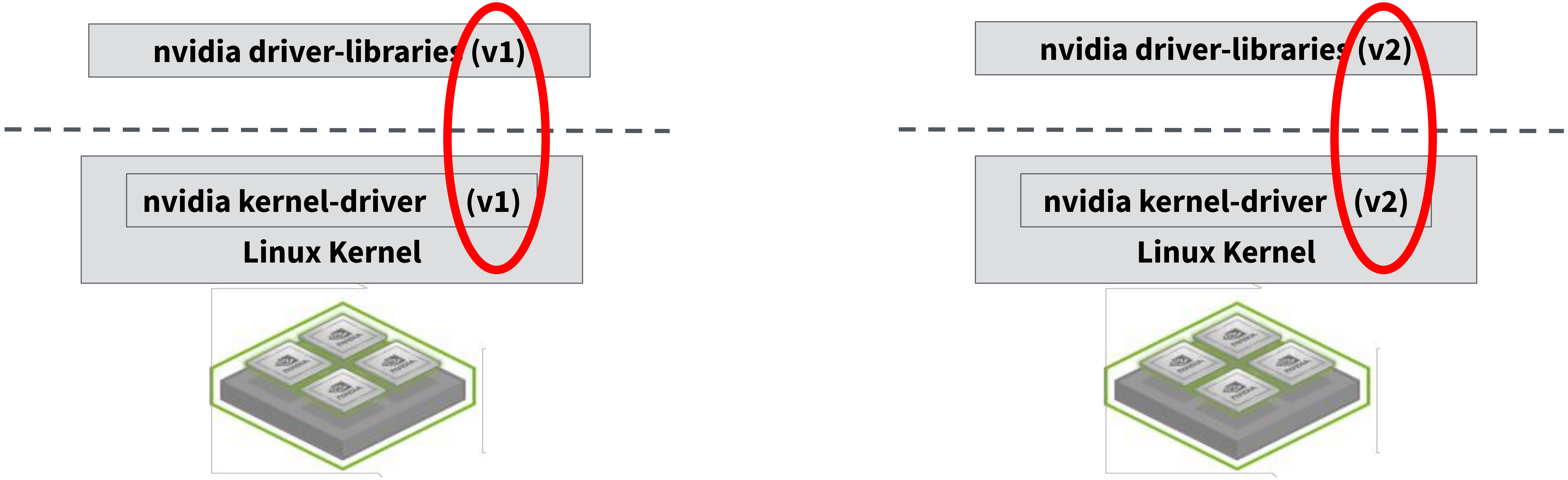


nvidia kernel-driver (v2)

Linux Kernel

GPUs AND CONTAINERS

The NVIDIA Container Toolkit



nvidia driver-libraries (v1)

nvidia kernel-driver (v1)

Linux Kernel

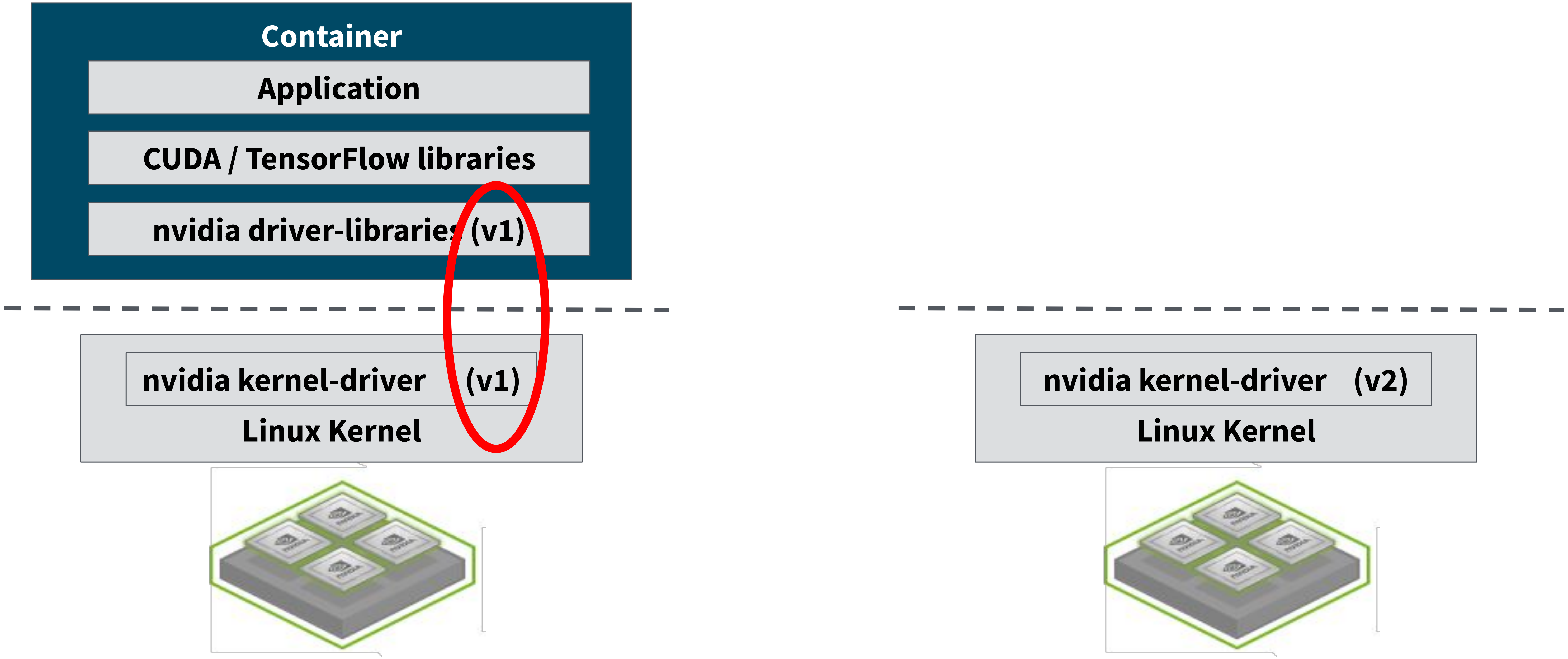
nvidia driver-libraries (v2)

nvidia kernel-driver (v2)

Linux Kernel

GPUs AND CONTAINERS

The NVIDIA Container Toolkit



Container

Application

CUDA / TensorFlow libraries

nvidia driver-libraries (v1)

nvidia kernel-driver (v1)

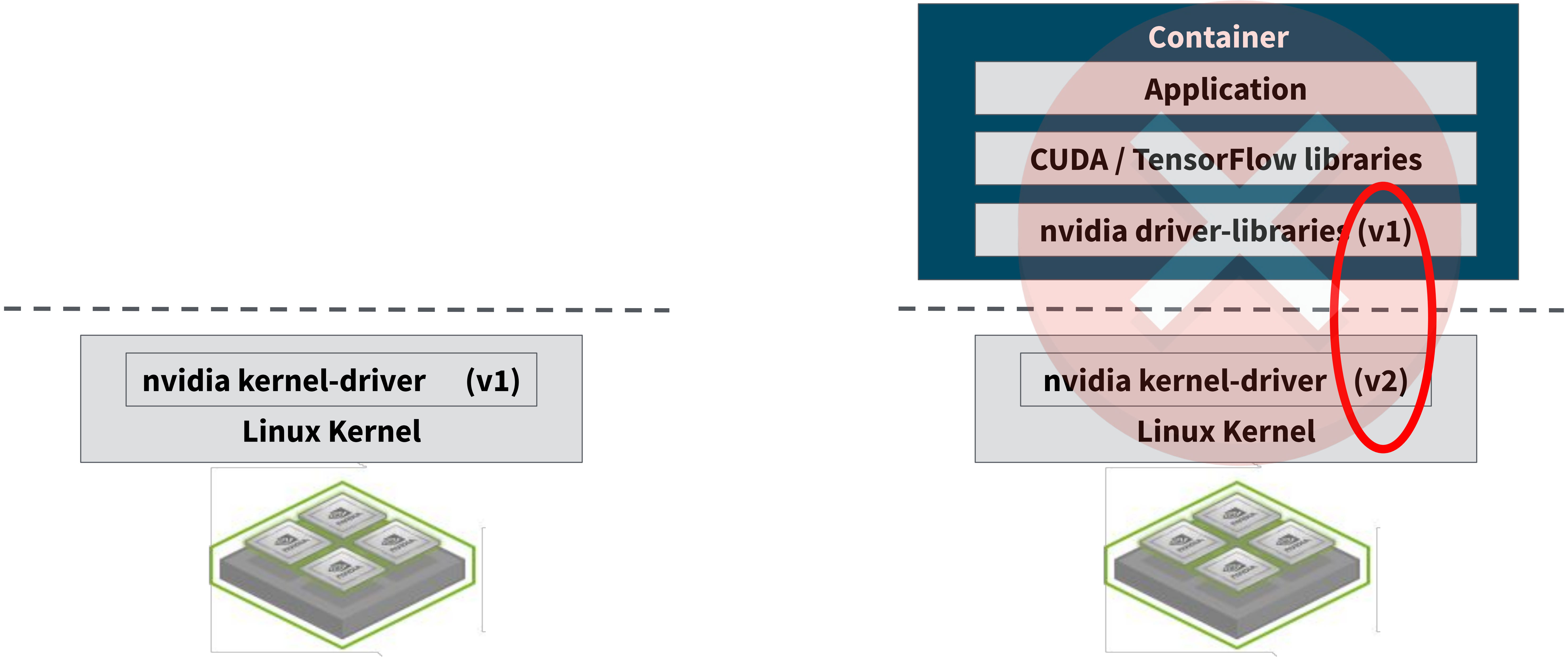
Linux Kernel

nvidia kernel-driver (v2)

Linux Kernel

GPUs AND CONTAINERS

The NVIDIA Container Toolkit



Container

Application

CUDA / TensorFlow libraries

nvidia driver-libraries (v1)

nvidia kernel-driver (v1)

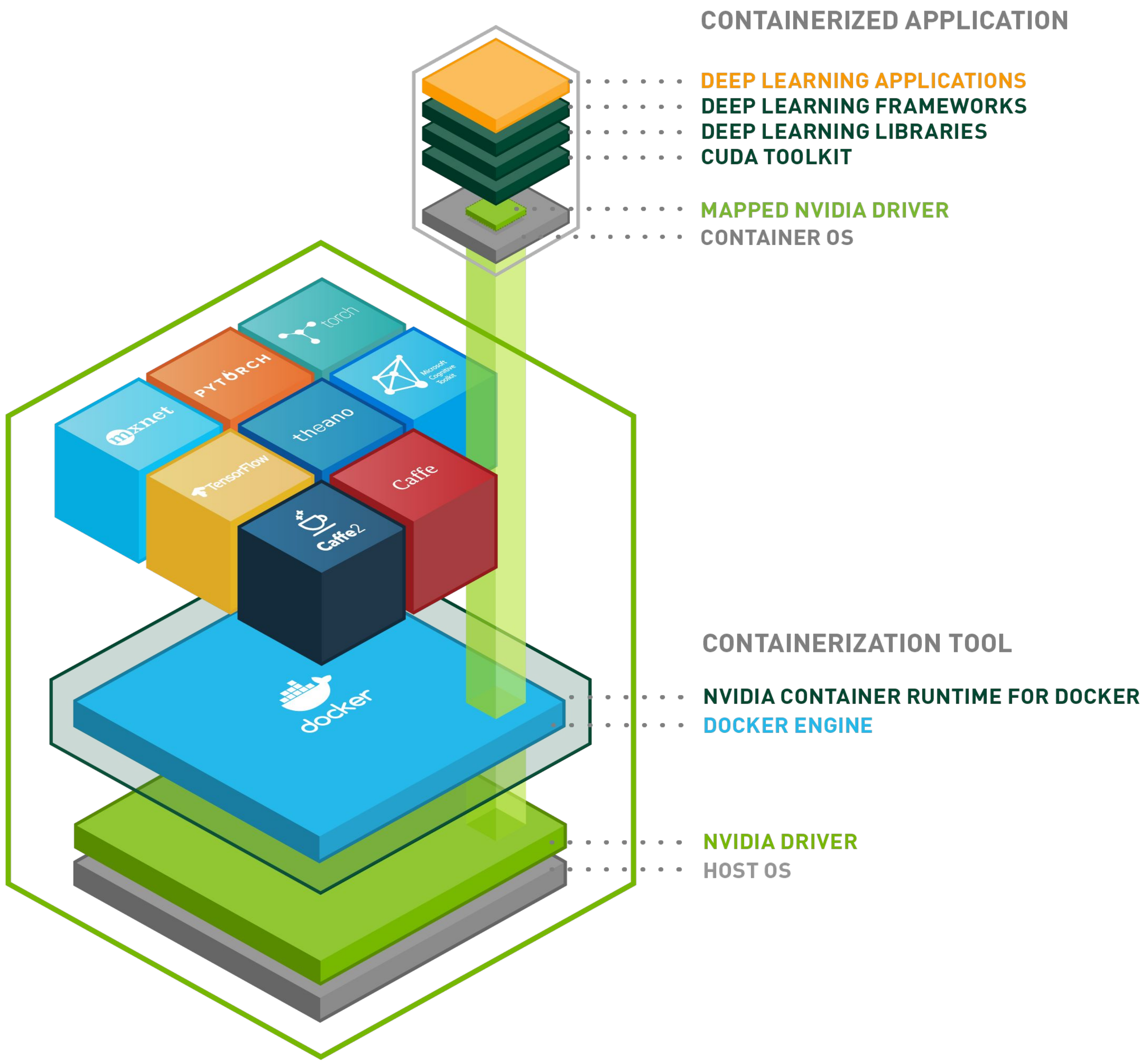
Linux Kernel

nvidia kernel-driver (v2)

Linux Kernel

GPUs AND CONTAINERS

The NVIDIA Container Toolkit



CONTAINERIZED APPLICATION

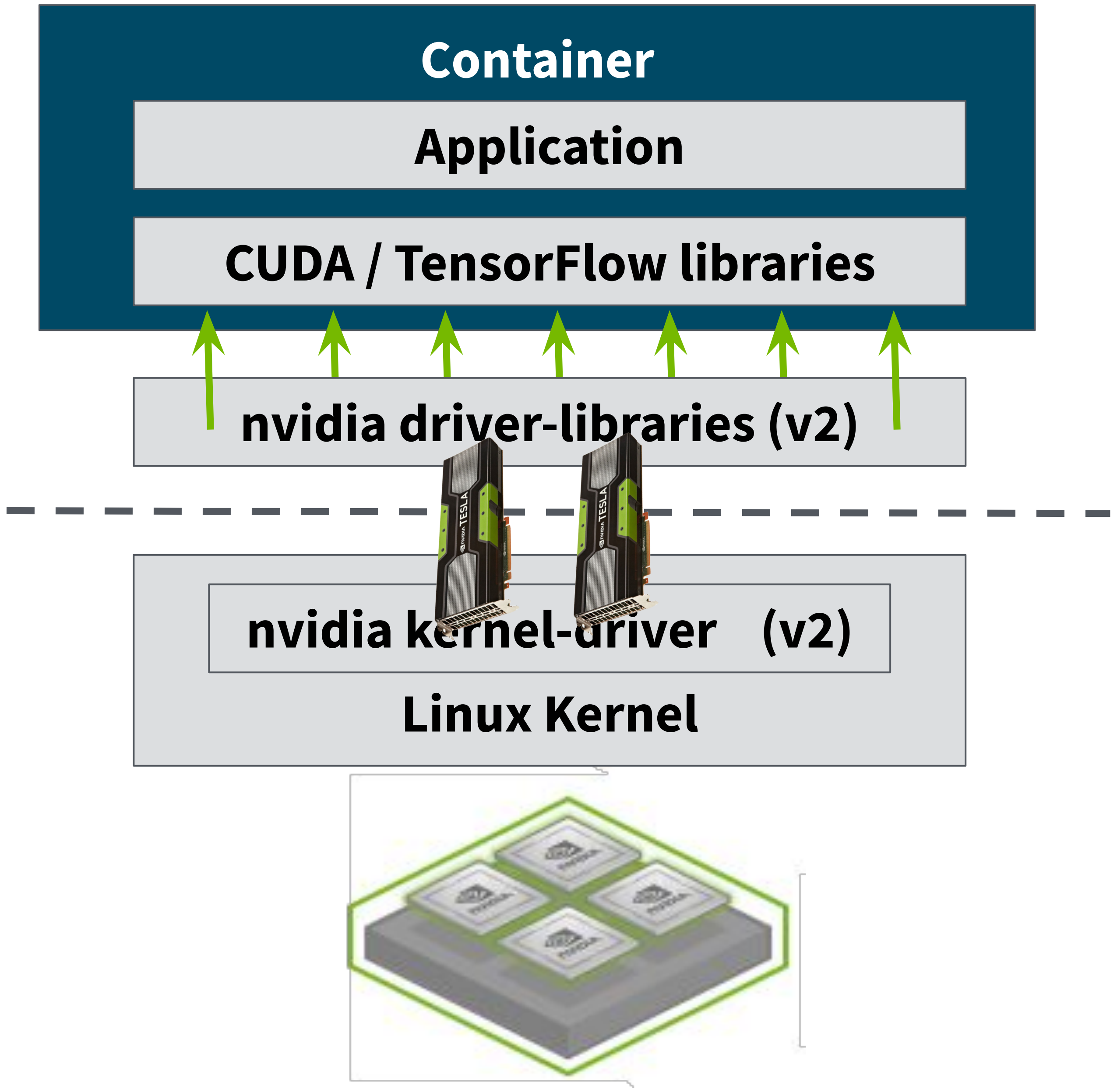
DEEP LEARNING APPLICATIONS
DEEP LEARNING FRAMEWORKS
DEEP LEARNING LIBRARIES
CUDA TOOLKIT

MAPPED NVIDIA DRIVER
CONTAINER OS

CONTAINERIZATION TOOL

NVIDIA CONTAINER RUNTIME FOR DOCKER
DOCKER ENGINE

NVIDIA DRIVER
HOST OS



Container

Application

CUDA / TensorFlow libraries

nvidia driver-libraries (v2)

nvidia kernel-driver (v2)

Linux Kernel

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

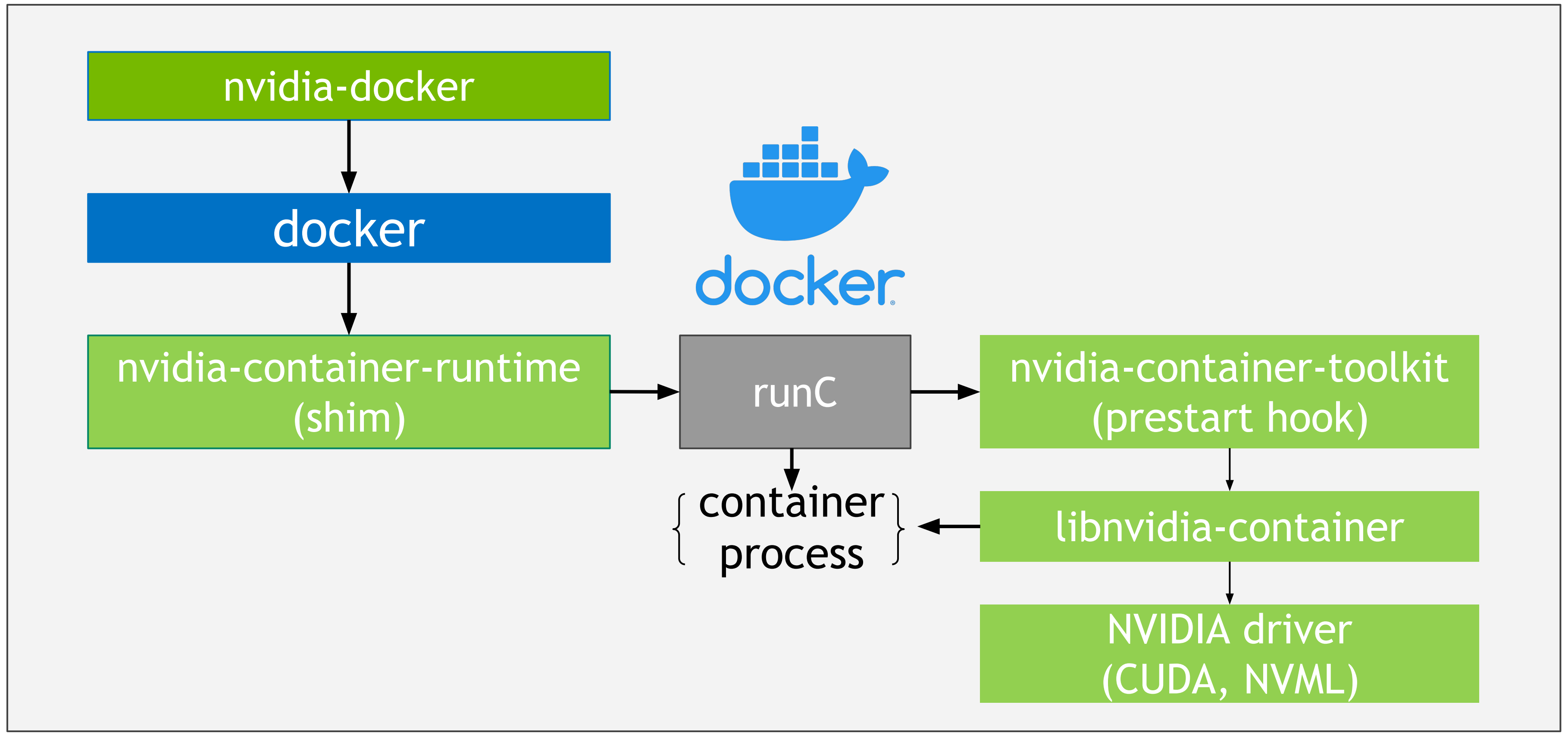
```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

- nvidia-docker
- nvidia-container-runtime
- nvidia-container-toolkit
- libnvidia-container



nvidia-docker

docker

nvidia-container-runtime (shim)

runC

nvidia-container-toolkit (prestart hook)

container process

libnvidia-container

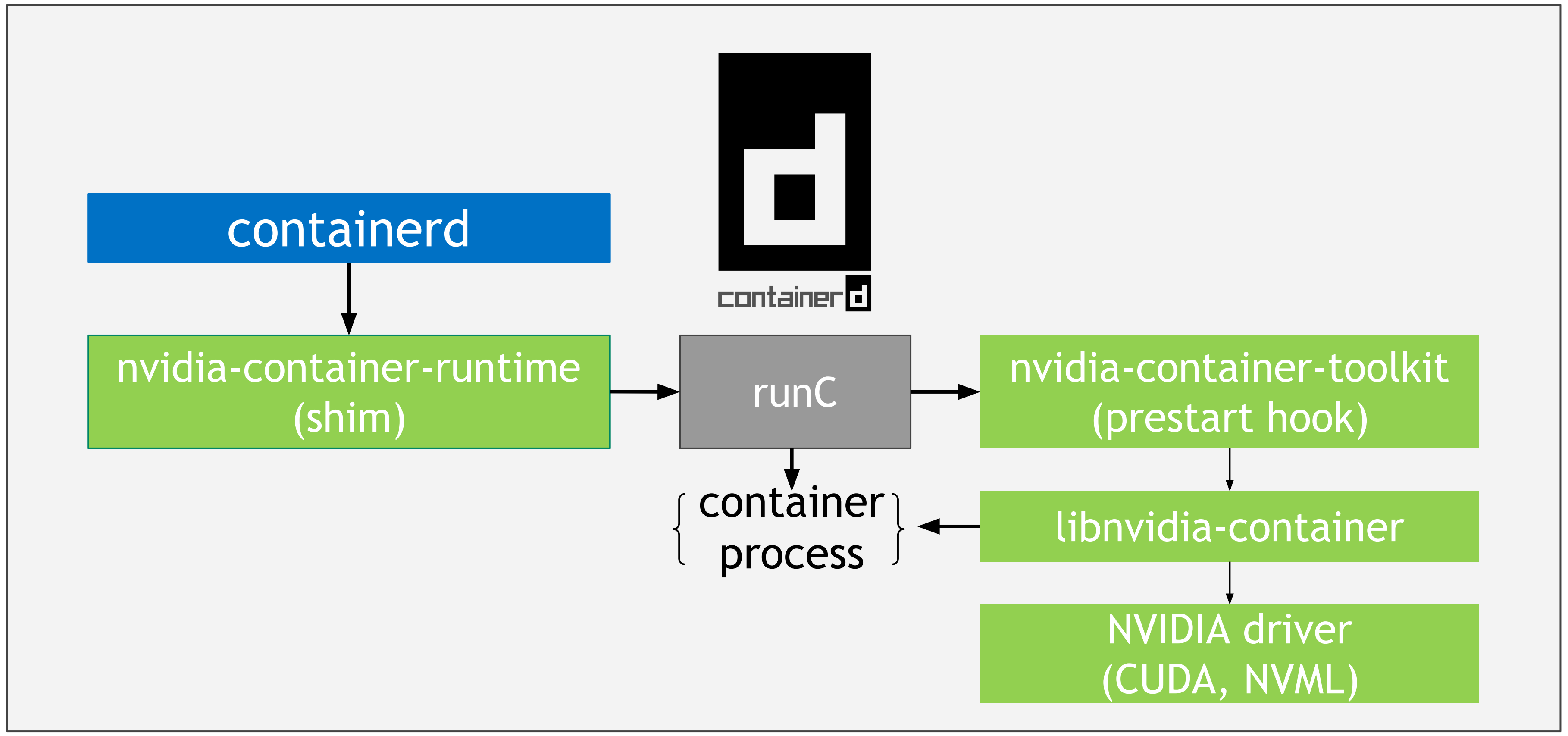
NVIDIA driver (CUDA, NVML)

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

- `nvidia-docker`
- `nvidia-container-runtime`
- `nvidia-container-toolkit`
- `libnvidia-container`

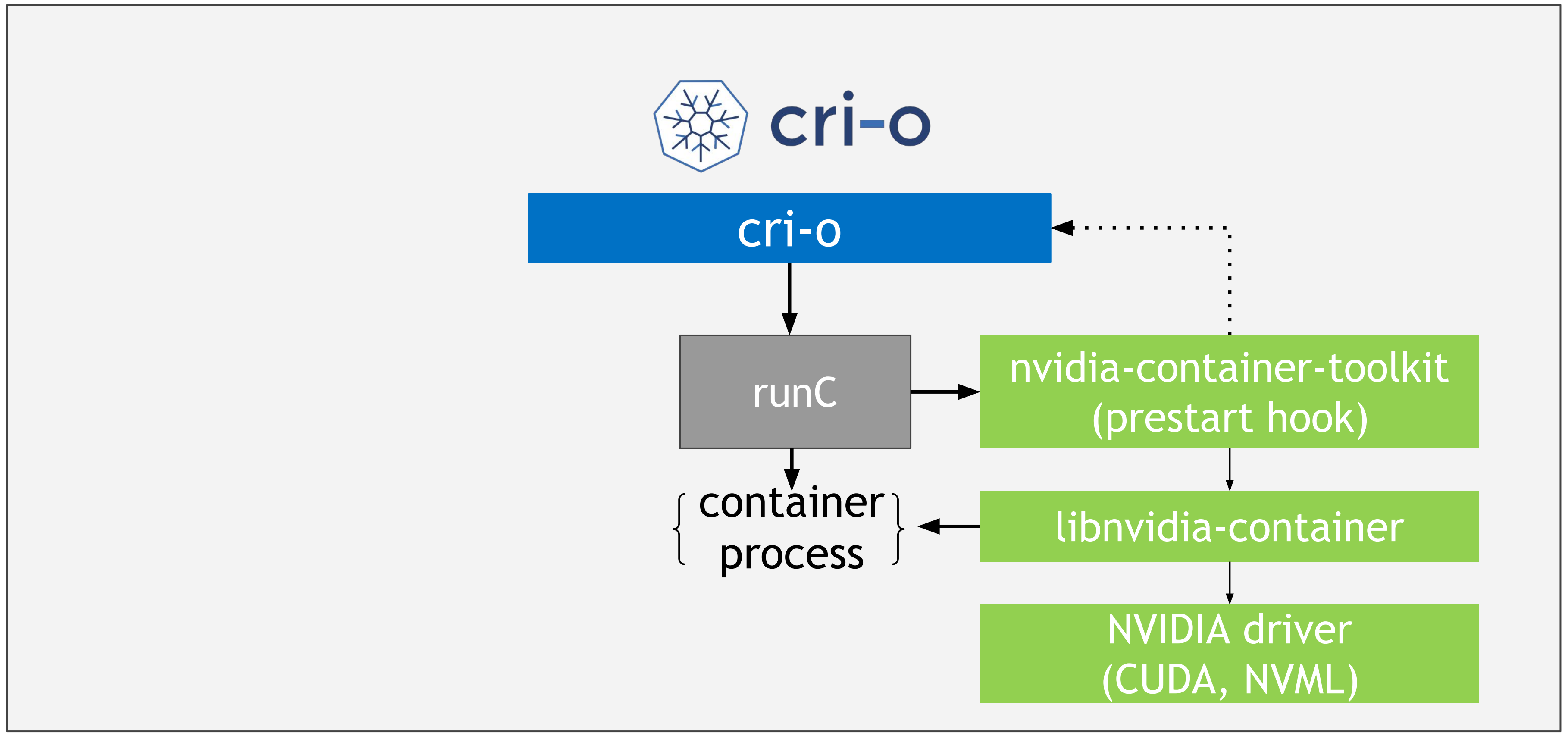

containerd
containerd
nvidia-container-runtime (shim)
runC
nvidia-container-toolkit (prestart hook)
container process
libnvidia-container
NVIDIA driver (CUDA, NVML)

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

- **`nvidia-docker`**
- **`nvidia-container-runtime`**
- **`nvidia-container-toolkit`**
- **`libnvidia-container`**

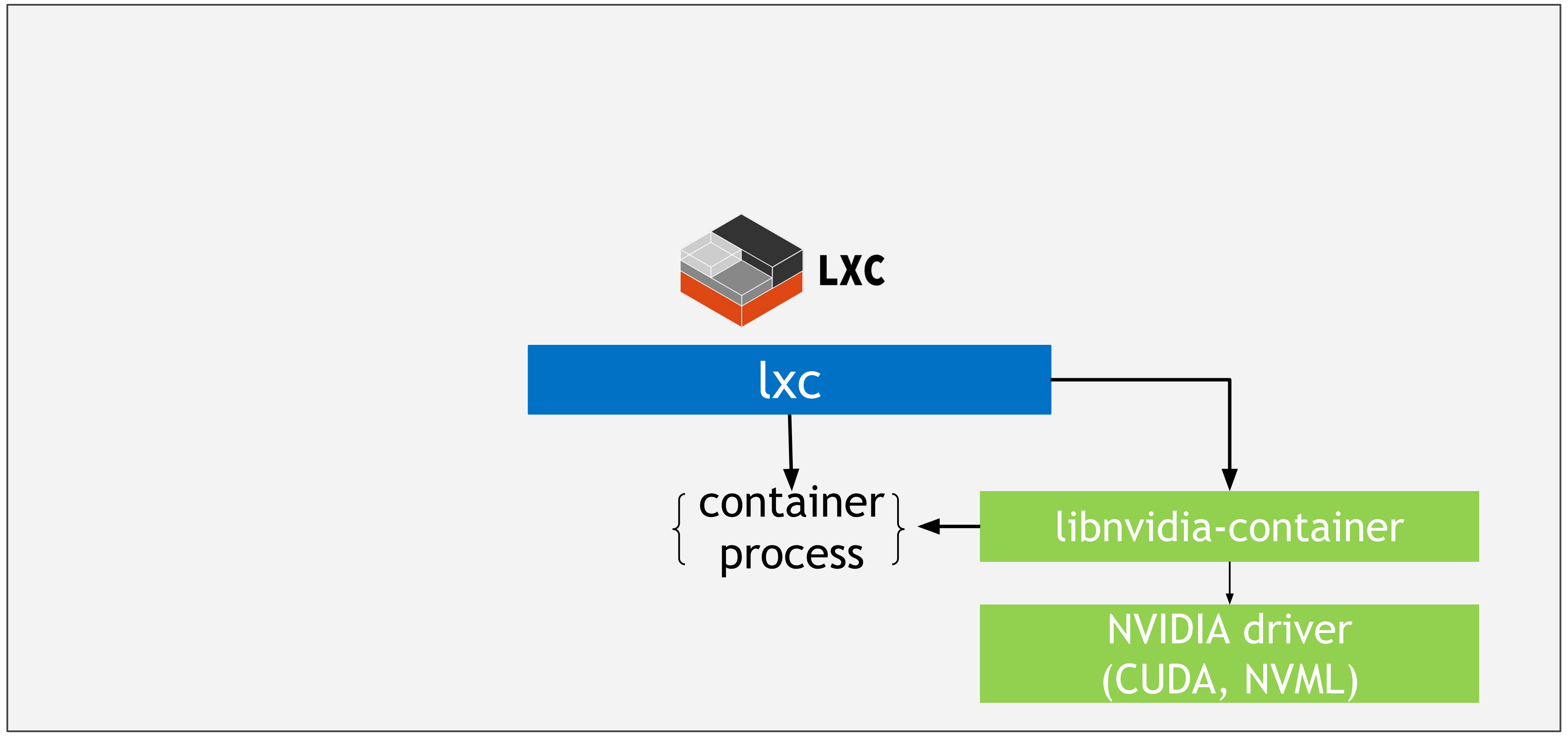


GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

- **nvidia-docker**
- **nvidia-container-runtime**
- **nvidia-container-toolkit**
- **libnvidia-container**



LXC

lxc

container process

libnvidia-container

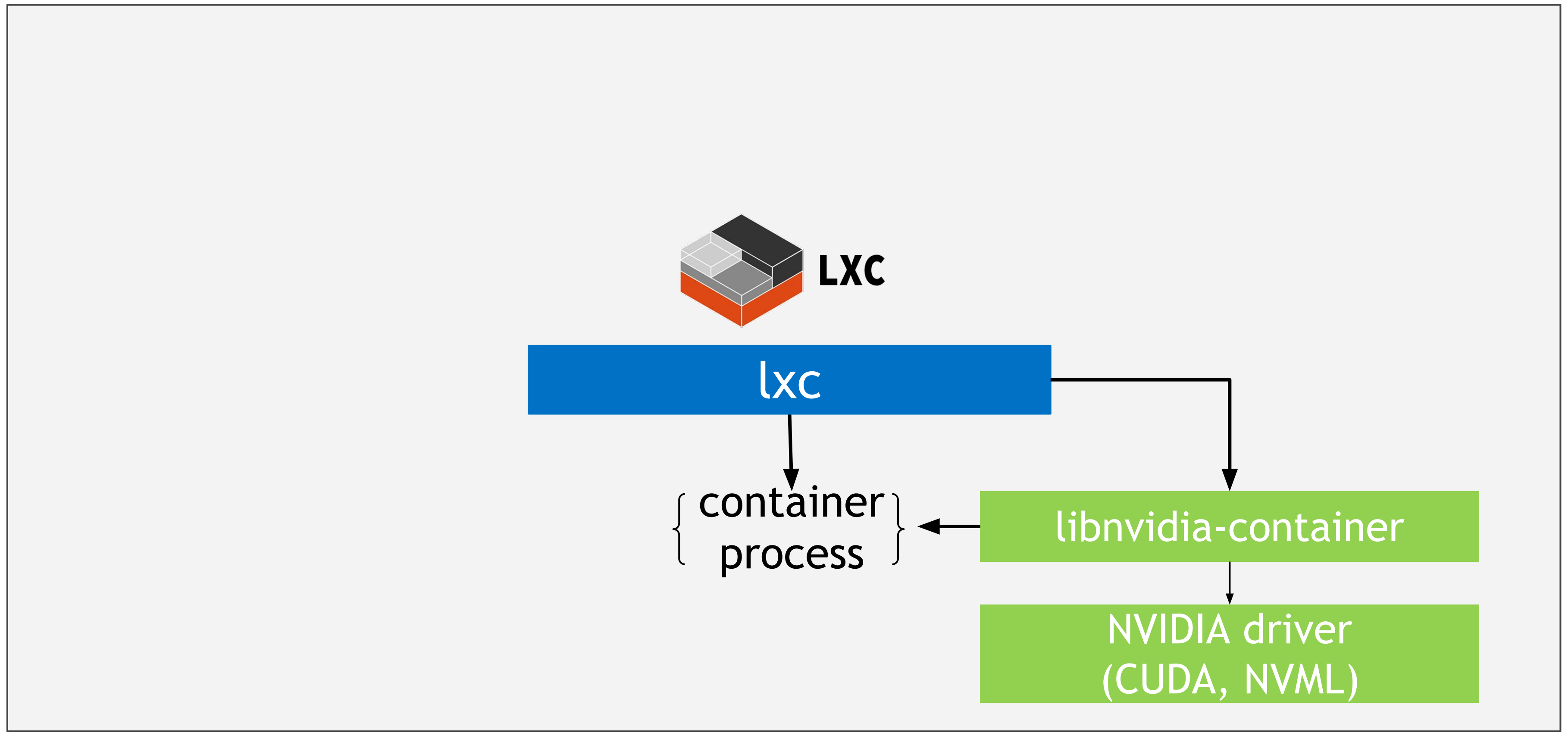
NVIDIA driver (CUDA, NVML)

GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

- **nvidia-docker**
- **nvidia-container-runtime**
- **nvidia-container-toolkit**
- **libnvidia-container**



LXC

lxc

container process

libnvidia-container

NVIDIA driver (CUDA, NVML)

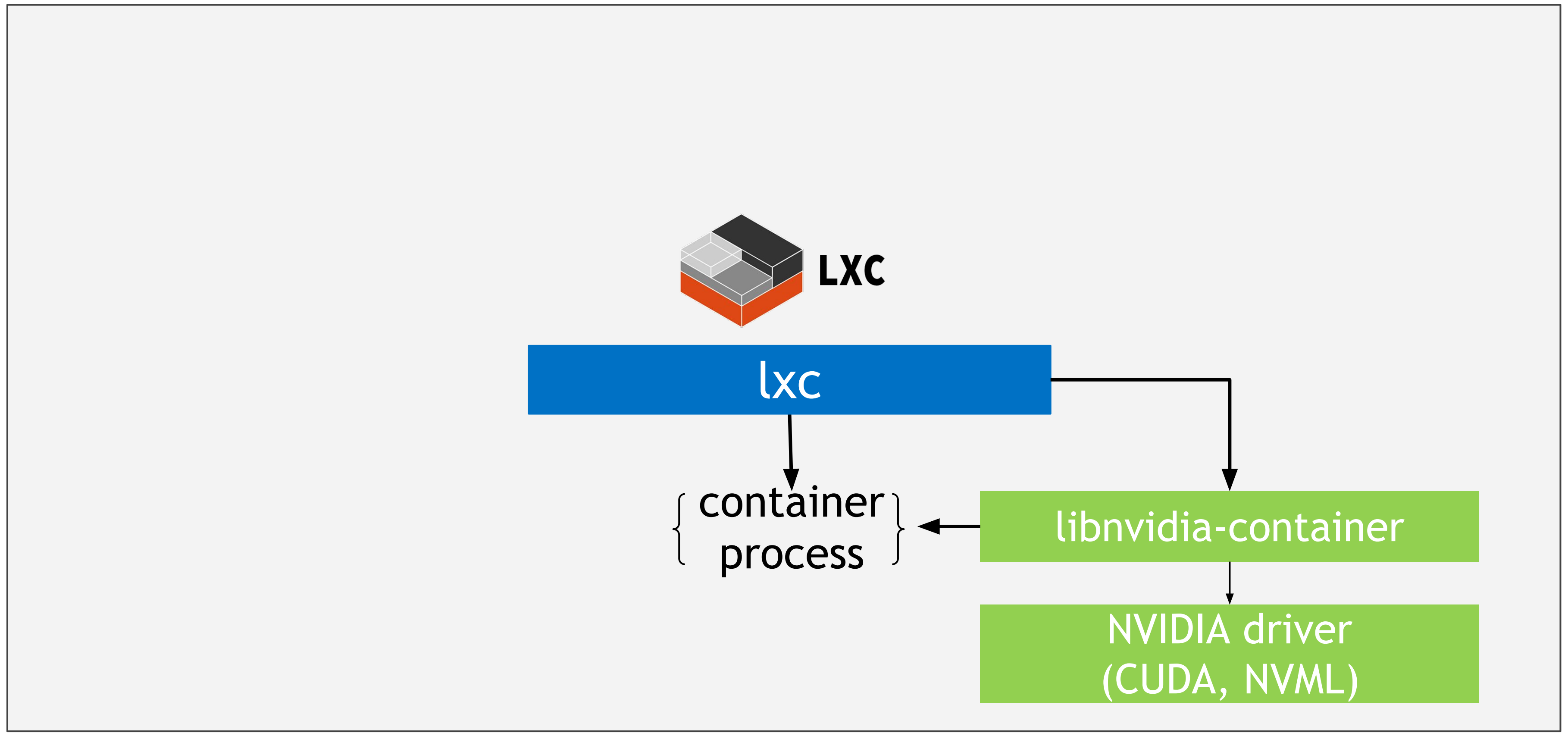
GPUs AND CONTAINERS

The NVIDIA Container Toolkit

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

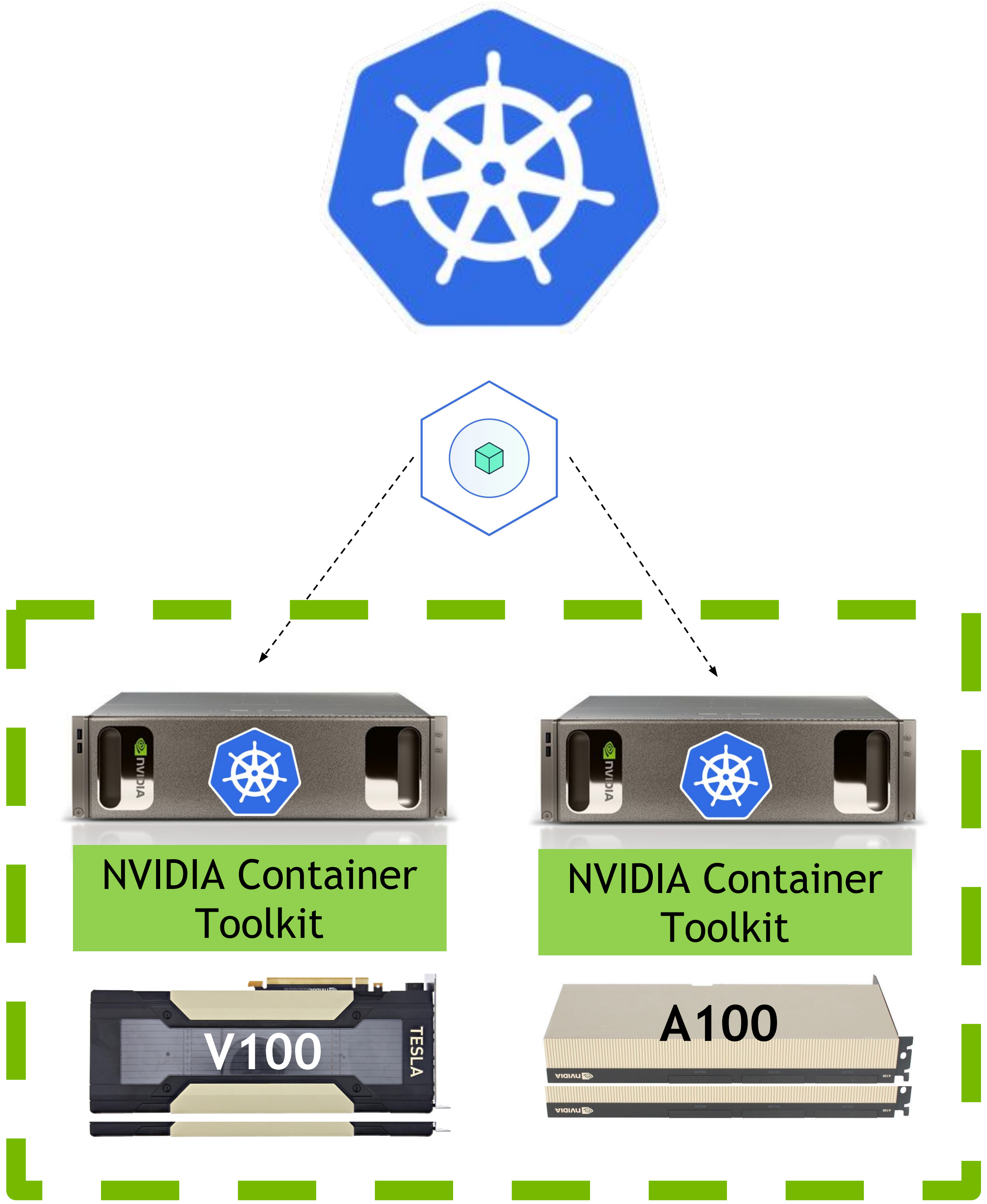
- `nvidia-docker`
- `nvidia-container-runtime`
- `nvidia-container-toolkit`
- `libnvidia-container`

Majority of code for MIG support
in containers added here
v1.3.0+



GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster



GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster

```
$ cat /etc/docker/daemon.json
{
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
     "path": "/usr/bin/nvidia-container-runtime",
     "runtimeArgs": []
    }
   }
  }
}
```

GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster

```
$ cat /etc/containerd/config.toml
[plugins]
  [plugins."io.containerd.grpc.v1.cri"]
    [plugins."io.containerd.grpc.v1.cri".containerd]
      default_runtime_name = "nvidia"

      [plugins."io.containerd.grpc.v1.cri".containerd.runtimes]
        [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia]
          privileged_without_host_devices = false
          runtime_engine = ""
          runtime_root = ""
          runtime_type = "io.containerd.runc.v2"
          [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.nvidia.options]
            BinaryName = "/usr/bin/nvidia-container-runtime"
```

containerd

GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster

```
$ cat /usr/share/containers/oci/hooks.d/oci-nvidia-hook.json
{
    "version": "1.0.0",
    "hook": {
        "path": "/usr/bin/nvidia-container-toolkit",
        "args": ["nvidia-container-toolkit", "prestart"]
    },
    "when": {
        "always": true,
        "commands": [".*"]
    },
    "stages": ["prestart"]
}
```

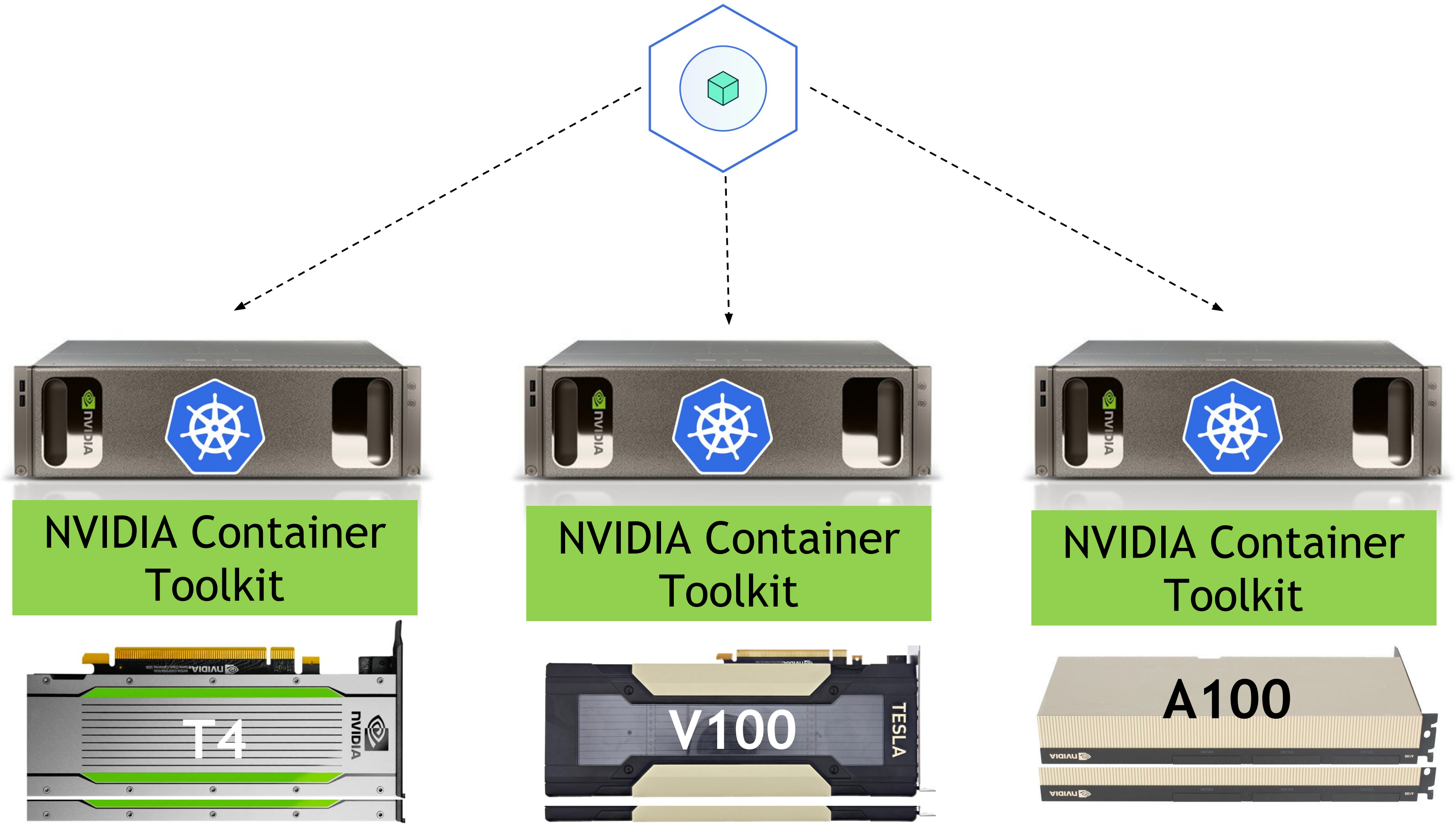
cri-o

GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 4
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
    nvidia.com/cuda.runtime: 11.0
    nvidia.com/cuda.driver: 450.51.06
```

- `k8s-device-plugin`
- `gpu-feature-discovery`

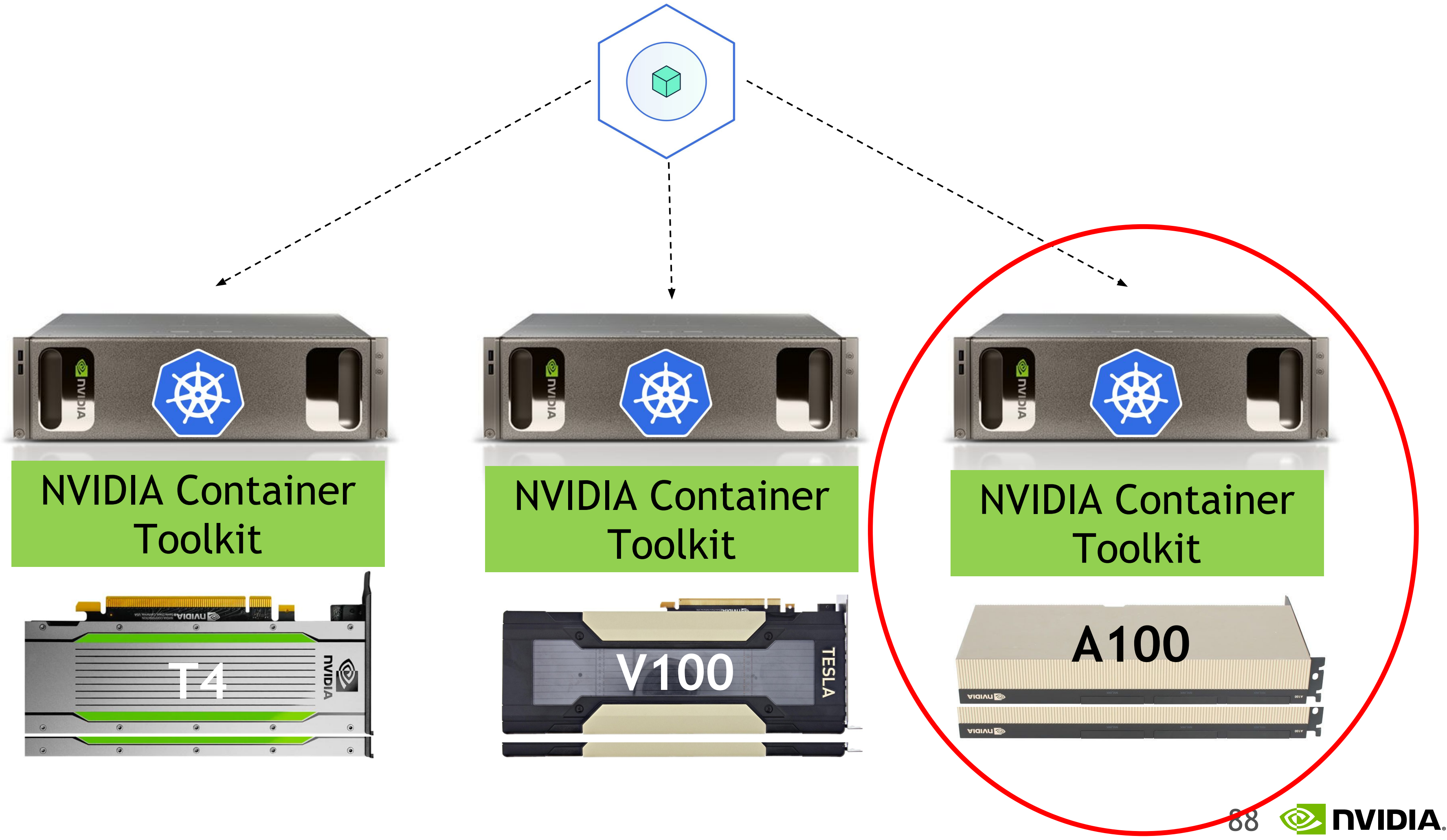


GPUs AND KUBERNETES

Allocate GPUs to pods in a Kubernetes Cluster

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 4
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
    nvidia.com/cuda.runtime: 11.0
    nvidia.com/cuda.driver: 450.51.06
```

- `k8s-device-plugin`
- `gpu-feature-discovery`



NVIDIA Container Toolkit

NVIDIA Container Toolkit

NVIDIA Container Toolkit

MIG IN CONTAINERS AND KUBERNETES

Injecting a MIG Device Into a Container

MIG IN CONTAINERS AND KUBERNETES

Injecting a MIG Device Into a Container

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

MIG IN CONTAINERS AND KUBERNETES

Injecting a MIG Device Into a Container

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

```
$ docker run \
  --gpus '"device=0:0,0:1,1:0,1:1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/4/0)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/4/0)
```

MIG IN CONTAINERS AND KUBERNETES

Injecting a MIG Device Into a Container

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

```
$ docker run \
  --gpus '"device=0:0,0:1,1:0,1:1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/4/0)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/4/0)
```

MIG IN CONTAINERS AND KUBERNETES

Injecting a MIG Device Into a Container

```
$ docker run \
  --gpus '"device=0,1"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
```

```
$ docker run \
  --gpus '"device=0:0,0:1,1:0,MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/4/0"' \
  nvidia/cuda:11.1-base nvidia-smi -L

GPU 0: A100-SXM4-40GB (UUID: GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-238f350c-0ed9-09cf-9945-fc0649ef02aa/4/0)
GPU 1: A100-SXM4-40GB (UUID: GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/3/0)
  MIG 1g.5gb Device 1: (UUID: MIG-GPU-dc36c15c-b7f1-cf33-f79a-bd01c0de7125/4/0)
```

MIG IN CONTAINERS AND KUBERNETES

Allocate MIG Devices to pods in a Kubernetes Cluster

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

MIG IN CONTAINERS AND KUBERNETES

Allocate MIG Devices to pods in a Kubernetes Cluster

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/mig-1g.5gb: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

MIG IN CONTAINERS AND KUBERNETES

Allocate MIG Devices to pods in a Kubernetes Cluster

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/mig-1g.5gb: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

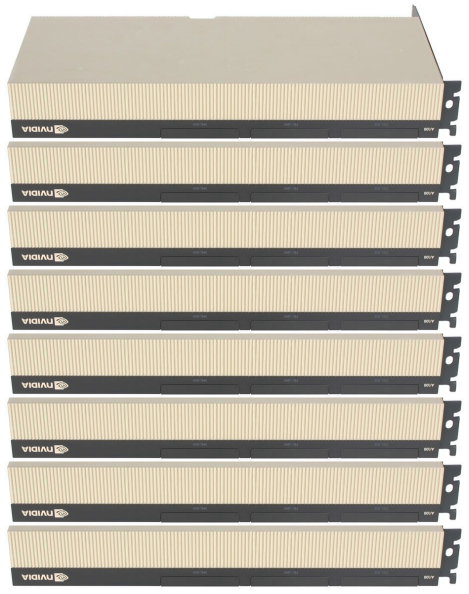
Mixed Strategy

MIG IN CONTAINERS AND KUBERNETES

Allocate MIG Devices to pods in a Kubernetes Cluster

DGX-A100

8 x A100



4 x Full A100s
and
8 x 1g.5gb
and
4 x 2g.10gb
and
4 x 3g.20gb

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/mig-1g.5gb:  1
          nvidia.com/mig-2g.10gb: 1
          nvidia.com/mig-3g.20gb: 1
          nvidia.com/gpu:         1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB
```

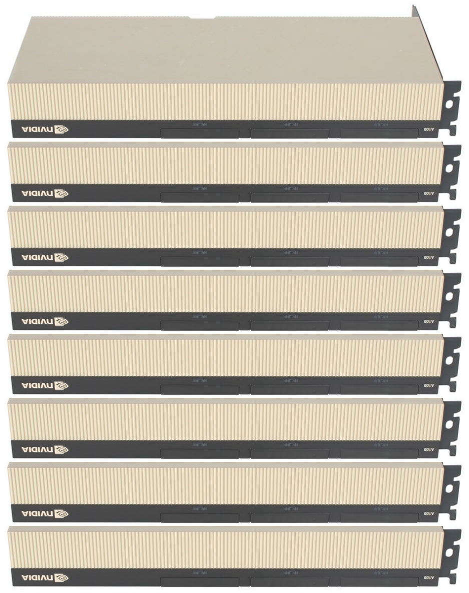
Mixed Strategy

MIG IN CONTAINERS AND KUBERNETES

Allocate MIG Devices to pods in a Kubernetes Cluster

DGX-A100

8 x A100



56 x 1g.5gb
or
24 x 2g.10gb
or
16 x 3g.20gb

```
apiVersion: v1
kind: Pod
metadata:
  name: gpu-example
spec:
  containers:
    - name: gpu-example
      image: nvidia/cuda
      resources:
        limits:
          nvidia.com/gpu: 1
  nodeSelector:
    nvidia.com/gpu.product: A100-PCIE-40GB MIG 1g.5gb
```

Single Strategy

MIG IN CONTAINERS AND KUBERNETES

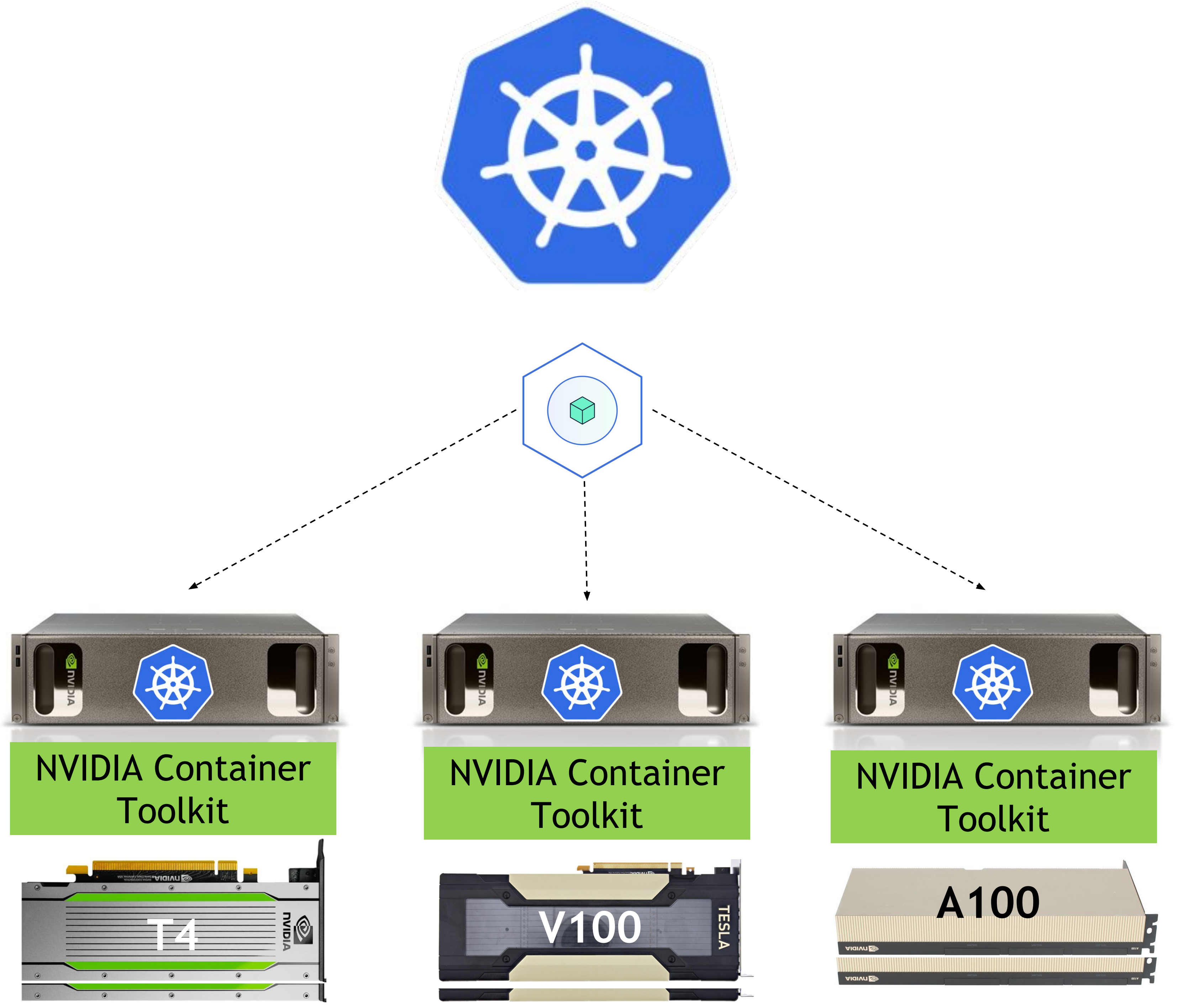
Allocate MIG Devices to pods in a Kubernetes Cluster

NVIDIA Container Toolkit

`nvidia-docker`	`v2.5.0+`
`nvidia-container-runtime`	`v3.4.2+`
`nvidia-container-toolkit`	`v1.4.2+`
`libnvidia-container`	`v1.3.3+`

Kubernetes

`k8s-device-plugin`	`v0.8.2+`
`gpu-feature-discovery`	`v0.4.1+`



SYSTEM LEVEL INTERFACE FOR MIG

Injecting GPUs and MIG devices into a container

SYSTEM LEVEL INTERFACE FOR MIG

Injecting GPUs and MIG devices into a container

For full GPUs:

Inject all of:

```
/dev
├── nvidiactl
├── nvidia-modeset
├── nvidia-uvm
└── nvidia-uvm-tools
```

Selectively inject for isolation:

```
/dev
└── nvidia0
└── nvidia1
└── nvidia2
```

SYSTEM LEVEL INTERFACE FOR MIG

Injecting GPUs and MIG devices into a container

For full GPUs:

Inject all of:

```
/dev
├── nvidiactl
├── nvidia-modeset
├── nvidia-uvm
└── nvidia-uvm-tools
```

Selectively inject for isolation:

```
/dev
└── nvidia0
└── nvidia1
└── nvidia2
```

Limit **/proc** view of:

```
/proc/driver/nvidia/gpus
├── 0000:07:00.0
├── 0000:0f:00.0
├── 0000:47:00.0
└── ...
```

Folders represent **PCIeBusID** of each full GPU whose device node is injected

SYSTEM LEVEL INTERFACE FOR MIG

Injecting GPUs and MIG devices into a container

For MIG devices:

Inject all of:

```
/dev
├── nvidiactl
├── nvidia-modeset
├── nvidia-uvm
└── nvidia-uvm-tools
```

Selectively inject for isolation:

```
/dev
├── nvidia0          ⎫
└── nvidia-caps      ⎬  MIG Device
    ├── nvidia-cap15 ⎪   3-Tuple
    └── nvidia-cap16 ⎭ <GPU, GI, CI>
```

SYSTEM LEVEL INTERFACE FOR MIG

Injecting GPUs and MIG devices into a container

For MIG devices:

Inject all of:

```
/dev
├── nvidiactl
├── nvidia-modeset
├── nvidia-uvm
└── nvidia-uvm-tools
```

Selectively inject for isolation:

```
/dev
├── nvidia0
└── nvidia-caps
    ├── nvidia-cap15
    └── nvidia-cap16
```

MIG Device
3-Tuple
<GPU, GI, CI>

SYSTEM LEVEL INTERFACE FOR MIG

nvidia-capabilities

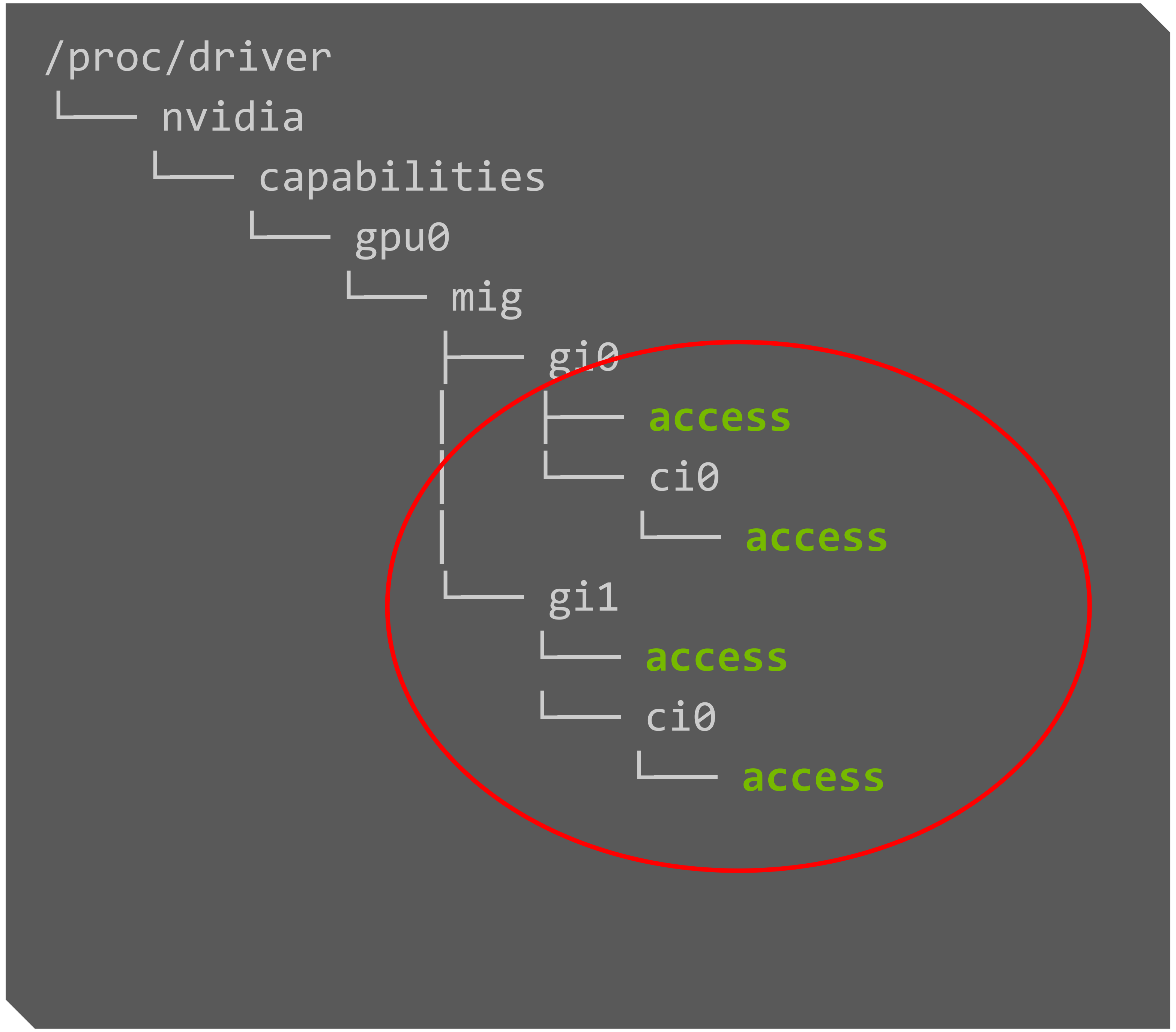
Set of **`nvidia-capabilities`** in **`/proc`**

Point to **`/dev/nvidia-caps`** for access control

SYSTEM LEVEL INTERFACE FOR MIG

nvidia-capabilities

Set of **`nvidia-capabilities`** in **`/proc`**



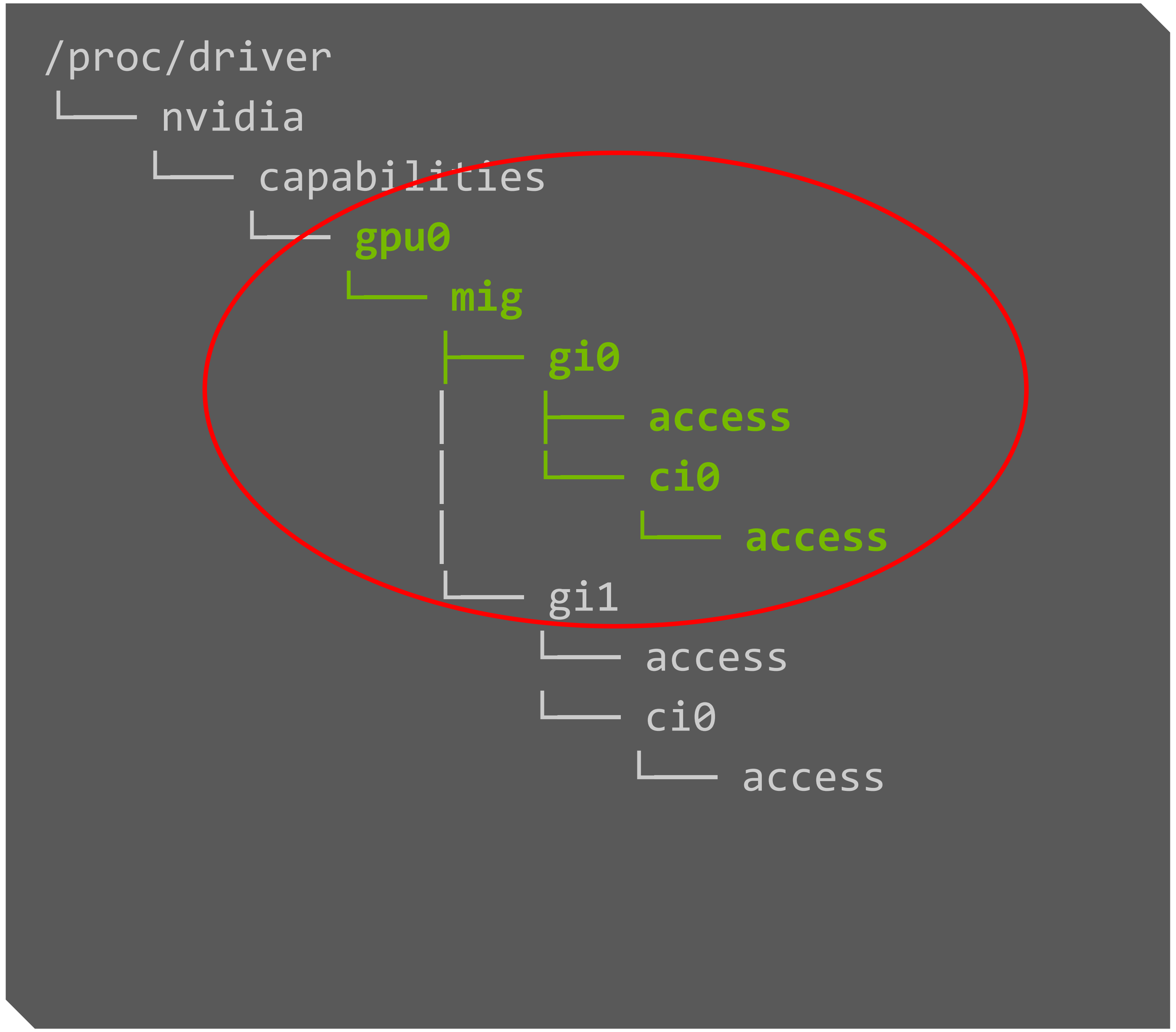
```
/proc/driver
└── nvidia
    └── capabilities
        └── gpu0
            └── mig
                ├── gi0
                │   ├── access
                │   └── ci0
                │       └── access
                └── gi1
                    └── access
                    └── ci0
                        └── access
```

Point to **`/dev/nvidia-caps`** for access control

SYSTEM LEVEL INTERFACE FOR MIG

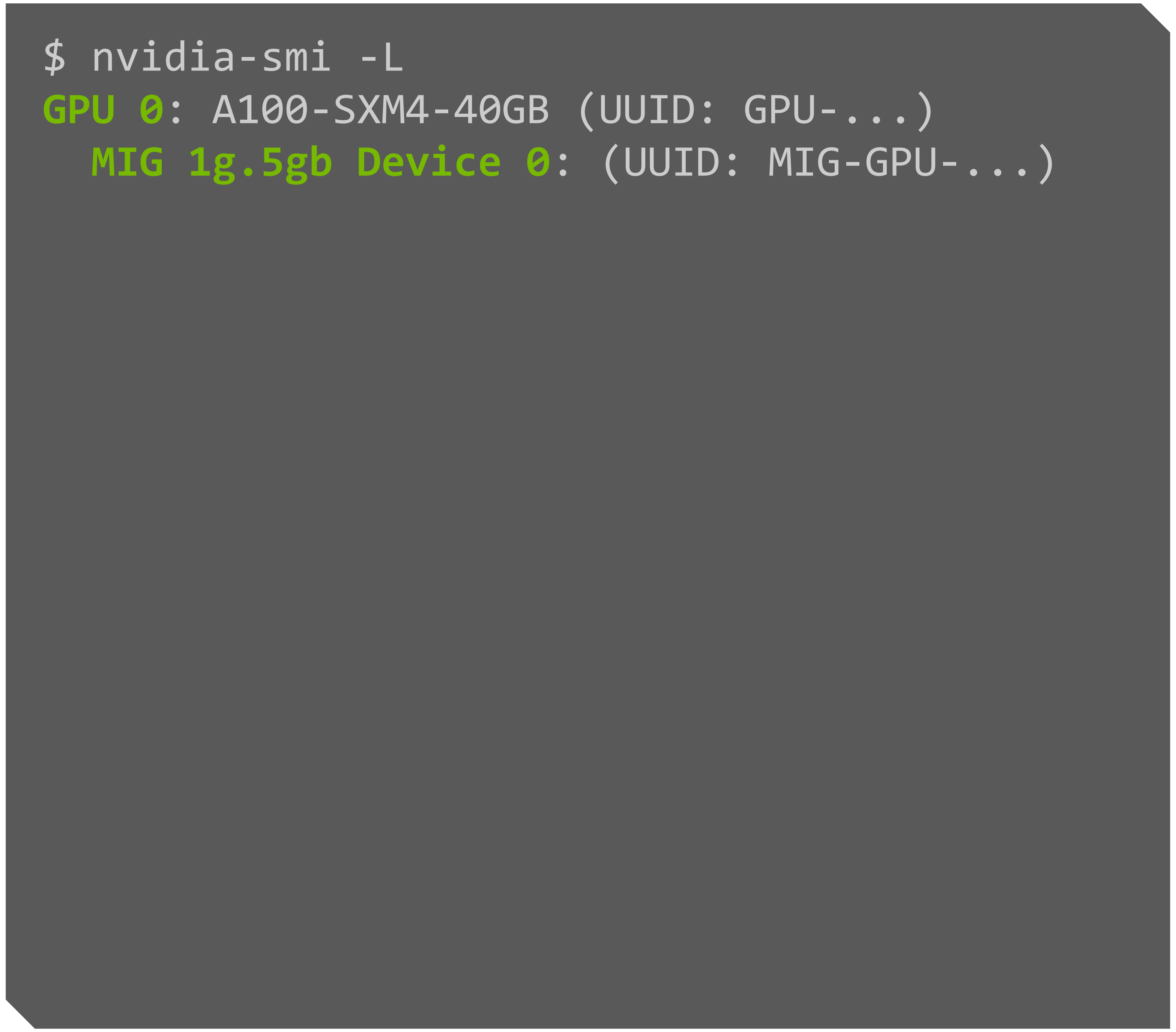
nvidia-capabilities

Set of **nvidia-capabilities** in **/proc**



```
/proc/driver
└── nvidia
    └── capabilities
        └── gpu0
            └── mig
                ├── gi0
                │   ├── access
                │   └── ci0
                │       └── access
                └── gi1
                    └── access
                    └── ci0
                        └── access
```

Point to **/dev/nvidia-caps** for access control



```
$ nvidia-smi -L
GPU 0: A100-SXM4-40GB (UUID: GPU-...)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-...)
```

SYSTEM LEVEL INTERFACE FOR MIG

nvidia-capabilities

Set of **nvidia-capabilities** in **/proc**

```
/proc/driver
 └── nvidia
     └── capabilities
         └── gpu0
             └── mig
                 ├── gi0
                 │   ├── access
                 │   └── ci0
                 │       └── access
                 └── gi1
                     └── access
                     └── ci0
                         └── access
```

Point to **/dev/nvidia-caps** for access control

```
$ nvidia-smi -L
GPU 0: A100-SXM4-40GB (UUID: GPU-...)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-...)

$ cat /proc/.../gpu0/mig/gi0/access
DeviceFileMinor: 15
```

SYSTEM LEVEL INTERFACE FOR MIG

nvidia-capabilities

Set of **`nvidia-capabilities`** in **`/proc`**

```
/proc/driver
└── nvidia
    └── capabilities
        └── gpu0
            └── mig
                ├── gi0
                │   ├── access
                │   └── ci0
                │       └── access
                └── gi1
                    └── access
                    └── ci0
                        └── access
```

Point to **`/dev/nvidia-caps`** for access control

```
$ nvidia-smi -L
GPU 0: A100-SXM4-40GB (UUID: GPU-...)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-...)

$ cat /proc/.../gpu0/mig/gi0/access
DeviceFileMinor: 15


$ cat /proc/.../gpu0/mig/gi0/ci0/access
DeviceFileMinor: 16
```

SYSTEM LEVEL INTERFACE FOR MIG

nvidia-capabilities

Set of **`nvidia-capabilities`** in **`/proc`**

```
/proc/driver
 └── nvidia
      └── capabilities
           └── gpu0
                └── mig
                     ├── gi0
                     │    ├── access
                     │    └── ci0
                     │         └── access
                     └── gi1
                          └── access
                          └── ci0
                               └── access
```

Point to **`/dev/nvidia-caps`** for access control

```
$ nvidia-smi -L
GPU 0: A100-SXM4-40GB (UUID: GPU-...)
  MIG 1g.5gb Device 0: (UUID: MIG-GPU-...)

$ cat /proc/.../gpu0/mig/gi0/access
DeviceFileMinor: 15


$ cat /proc/.../gpu0/mig/gi0/ci0/access
DeviceFileMinor: 16


$ ls /dev/nvidia-caps
cr---------  1 root root 238, 15 nvidia-cap15
cr--r--r--  1 root root 238, 16 nvidia-cap16
```

SYSTEM LEVEL INTERFACE FOR MIG

Limiting the view of /proc/driver/nvidia/capabilities

For MIG devices:

Additionally limit **/proc** view of:

```
/proc/driver/nvidia/capabilities
└── gpu0
    └── mig
        ├── gi0
        ├── access
        └── ci0
            └── access
```

Access files point to **nvidia-caps** whose devices nodes are injected

CHALLENGES WITH MIG PARTITIONING

How do I create a MIG Device in the first place?

CHALLENGES WITH MIG PARTITIONING

How do I create a MIG Device in the first place?

Two distinct workflows when configuring a GPU for use with MIG

- Enabling MIG mode on a GPU in the first place
- Configuring MIG devices on a GPU that is already in MIG mode

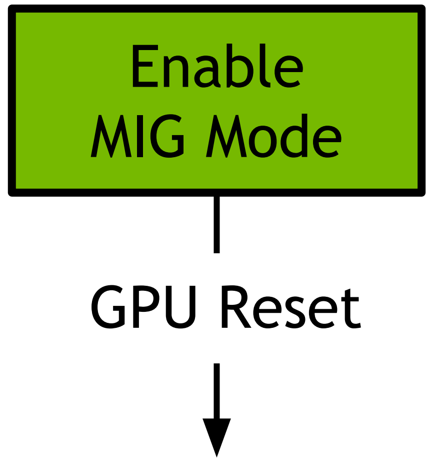
```
# Enable MIG mode
nvidia-smi -mig 1

# Create 7 x 1g.5gb MIG devices
nvidia-smi mig -cgi 19,19,19,19,19,19,19 -C
```

CHALLENGES WITH MIG PARTITIONING

How do I create a MIG Device in the first place?

Enabling MIG mode on a GPU

- Complete all running GPU workloads
- Disconnect all driver clients from the GPU
- Enable MIG mode on the GPU
- Perform a GPU reset



Challenges:

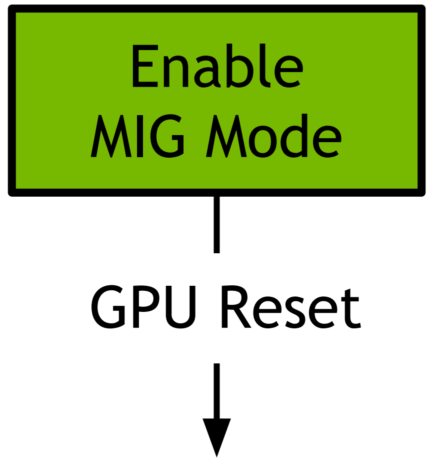
- Hard to enumerate all driver clients and reconnect them after the reset
- Requires a ***full node reboot*** under GPU pass-through virtualization

CHALLENGES WITH MIG PARTITIONING

How do I create a MIG Device in the first place?

Enabling MIG mode on a GPU

- Complete all running GPU workloads
- Disconnect all driver clients from the GPU
- Enable MIG mode on the GPU
- Perform a GPU reset



Very Heavy-weight operation

Should be done very infrequently

Challenges:

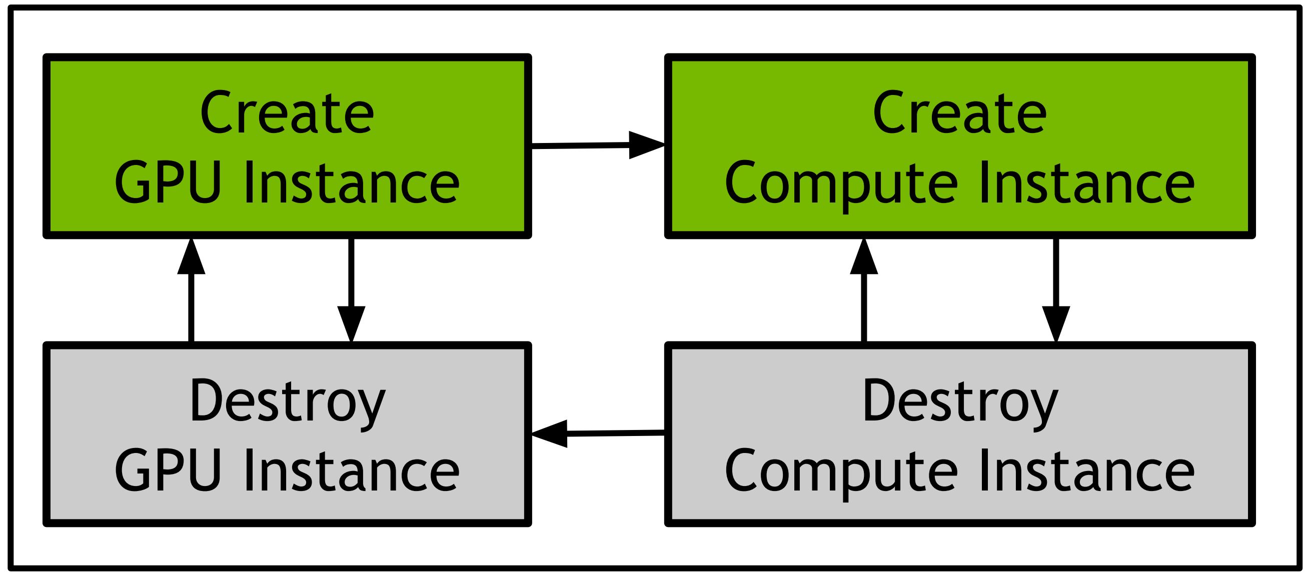
- Hard to enumerate all driver clients and reconnect them after the reset
- Requires a ***full node reboot*** under GPU pass-through virtualization

CHALLENGES WITH MIG PARTITIONING

How do I create a MIG Device in the first place?

Configuring MIG devices on a GPU

- Complete all running GPU workloads
- Perform device reconfiguration



Challenges:

- The order in which MIG devices are created matters
- Need to restart any components caching previous MIG device state
- Does not persist configuration across a GPU reset (or a node reboot)
- Hard to manage MIG device state across a cluster of machines

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

```
version: v1
mig-configs:
  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2
```

- Declaratively define the set of *possible* MIG partitions you want to create on GPUs throughout your cluster

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

```
version: v1
mig-configs:
  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2
```

- Declaratively define the set of *possible* MIG partitions you want to create on GPUs throughout your cluster

- Make this file available to all of their nodes (as an actual file or a **`configMap`**)

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

```
version: v1
mig-configs:
  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2
```

- Declaratively define the set of *possible* MIG partitions you want to create on GPUs throughout your cluster

- Make this file available to all of their nodes (as an actual file or a **`configMap`**)

- Use **`nvidia-mig-parted`** to apply one of these configurations on a given node

```
$ nvidia-mig-parted apply -f config.yaml -c all-2g.10gb
```

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

```
version: v1
mig-configs:
  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2
```

`systemd` service wrapper

- Persists MIG configurations across node reboots
- Applies MIG mode changes *without* the NVIDIA driver loaded
- Automatically handles start/stop of GPU clients across configuration
  https://github.com/NVIDIA/mig-parted/tree/master/deployments/systemd

MIG PARTITION EDITOR

https://github.com/NVIDIA/mig-parted

```
version: v1
mig-configs:
  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2
```

`systemd` service wrapper

- Persists MIG configurations across node reboots
- Applies MIG mode changes *without* the NVIDIA driver loaded
- Automatically handles start/stop of GPU clients across configuration

  https://github.com/NVIDIA/mig-parted/tree/master/deployments/systemd

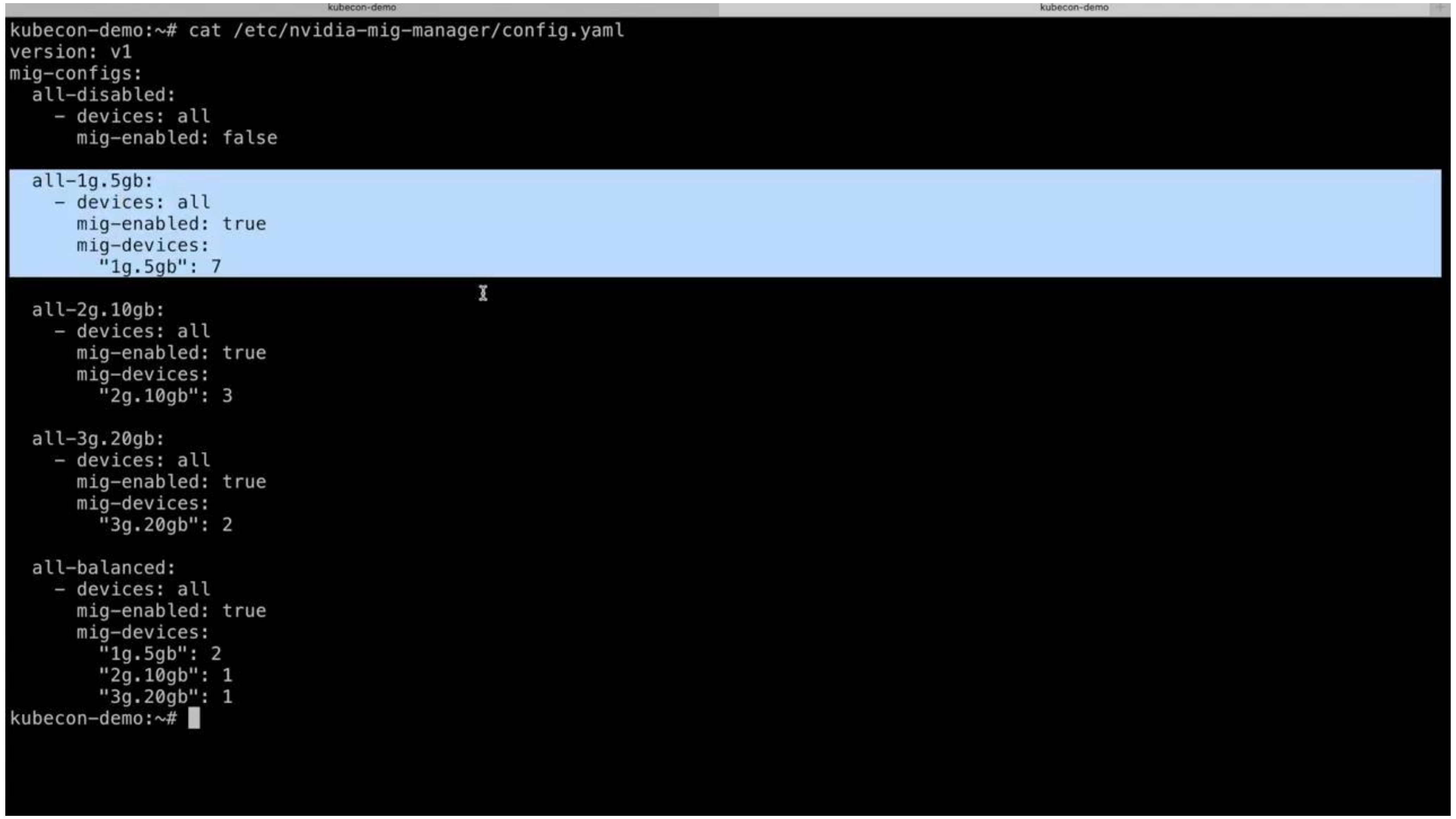
`kubernetes` service wrapper (coming soon)

- Provides similar functionality as the `systemd` service wrapper
- But integrated as a Kubernetes service instead
- Will become part of the GPU Operator

PUTTING IT ALL TOGETHER

Demo

1. Show full GPUs
2. Run **`mig-parted`**
3. Show mixed strategy
4. Show single strategy



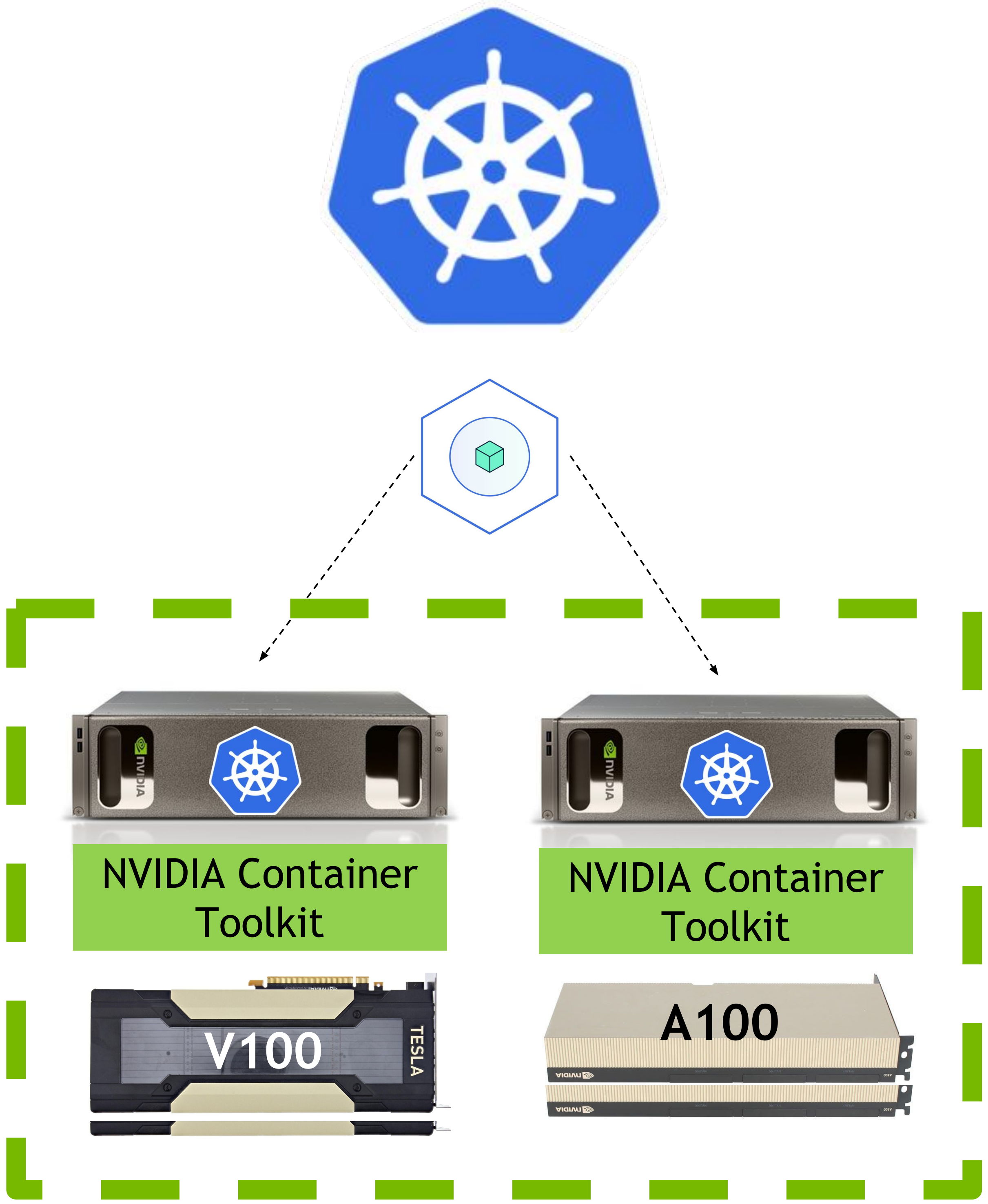
```
kubecon-demo:~# cat /etc/nvidia-mig-manager/config.yaml
version: v1
mig-configs:
  all-disabled:
    - devices: all
      mig-enabled: false

  all-1g.5gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 7

  all-2g.10gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "2g.10gb": 3

  all-3g.20gb:
    - devices: all
      mig-enabled: true
      mig-devices:
        "3g.20gb": 2

  all-balanced:
    - devices: all
      mig-enabled: true
      mig-devices:
        "1g.5gb": 2
        "2g.10gb": 1
        "3g.20gb": 1
kubecon-demo:~#
```

SUMMARY AND CONCLUSION

https://docs.nvidia.com/datacenter/cloud-native/kubernetes/mig-k8s.html

- MIG provides hardware support for sharing GPUs
- Support exists in both standalone containers and Kubernetes
- The **`nvidia-mig-parted`** tool simplifies partitioning MIG on a node

Integrated support on EKS, GKE, and AKS coming very soon



NVIDIA Container Toolkit

NVIDIA Container Toolkit

NVIDIA®