
KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

BUILDING FOR THE ROAD AHEAD

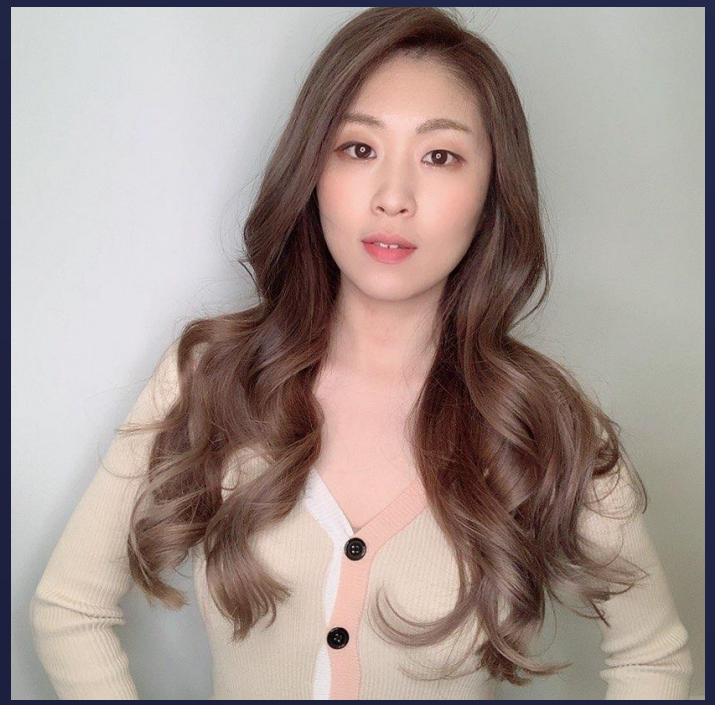
DETROIT 2022

Machine Learning Using Various GPU Technologies with Kubeflow

Jihye Choi, SAMSUNG SDS

Machine Learning Using Various GPU Technologies with Kubeflow

KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022



Jihye Choi
Cloud Architect
SAMSUNG SDS

October 24-28, 2022

Samsung Cloud Platform

- Compute
- Storage
- Database
- Container
- Networking
- Security
- Application Service
- DevOps Tools
- Analytics
- AI Service
- Management

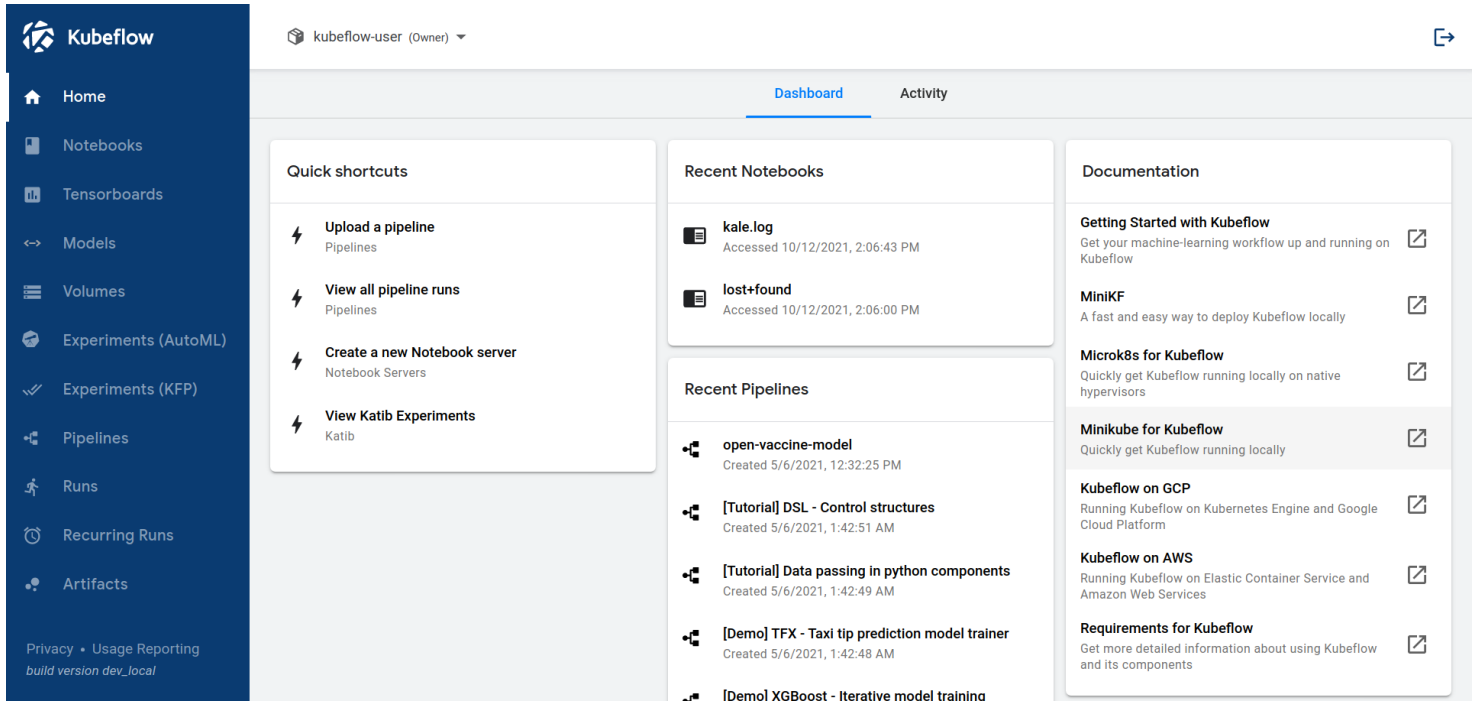
Kubeflow

Kubernetes-based Machine Learning Platform

SAMSUNG SDS

What is Kubeflow?

Kubeflow



kubernetes

jupyter

PyTorch

Istio

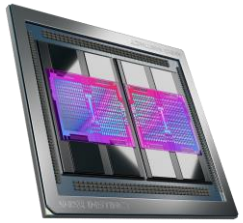
TensorFlow

mxnet

TensorRT

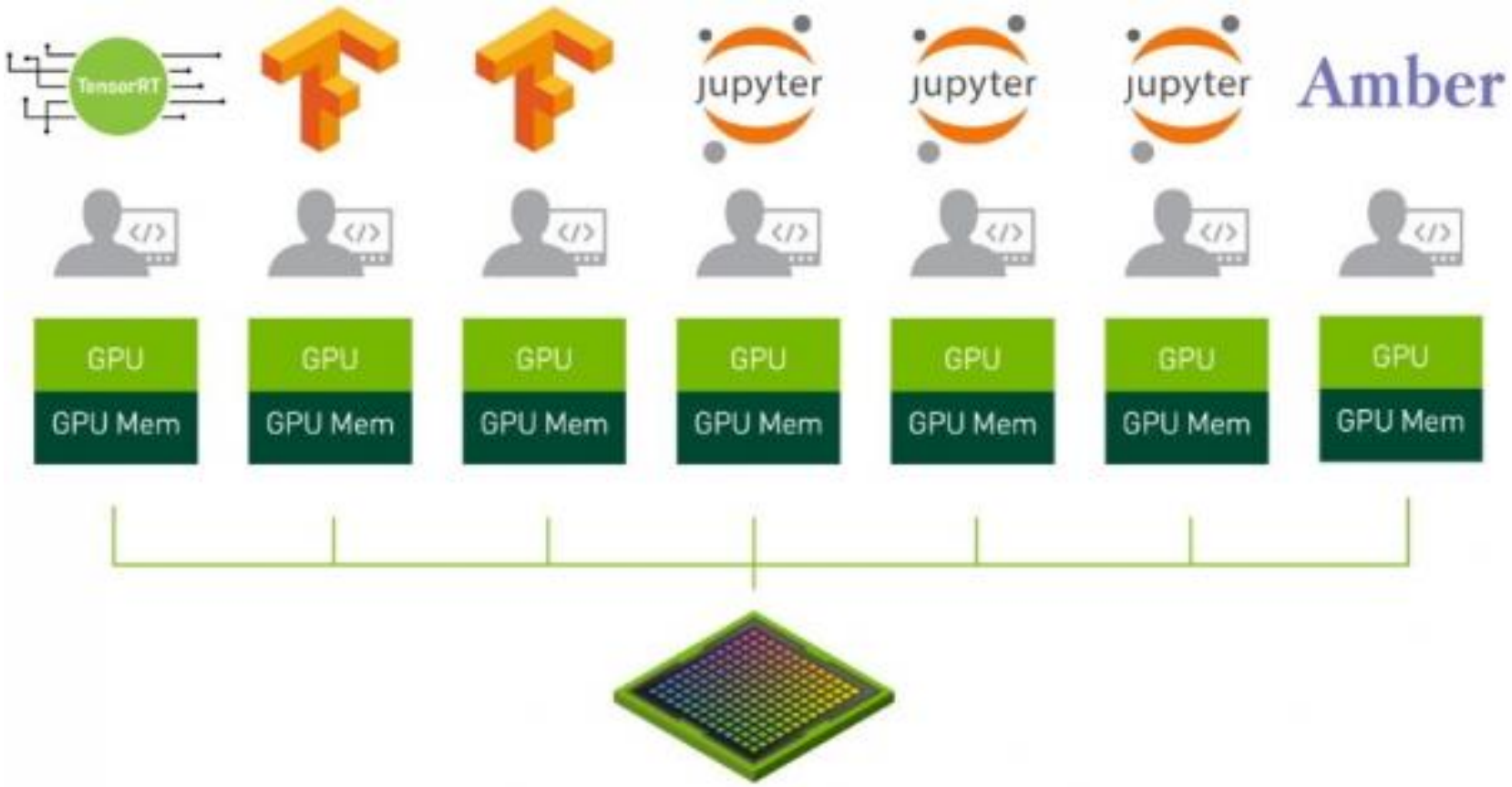
Case Study 1 : Multi-Instance GPU

GPUs in **Data Center** VS **Professional Desktop**

	NVIDIA H100	NVIDIA A100	AMD MI250	NVIDIA RTX
				
GPU Memory	80GB	80GB	128GB	4GB - 48GB
Memory Bandwidth	2.0TB/s	1,935 GB/s	3,276 GB/s	80 GB/s – 768 GB/s
Interconnect	PCIe Gen5	PCIe Gen4	PCIe Gen4	PCIe Gen3 PCIe Gen4
	GPUs in Data Center			**GPUs in Professional Desktop**

Case Study 1 : Multi-Instance GPU

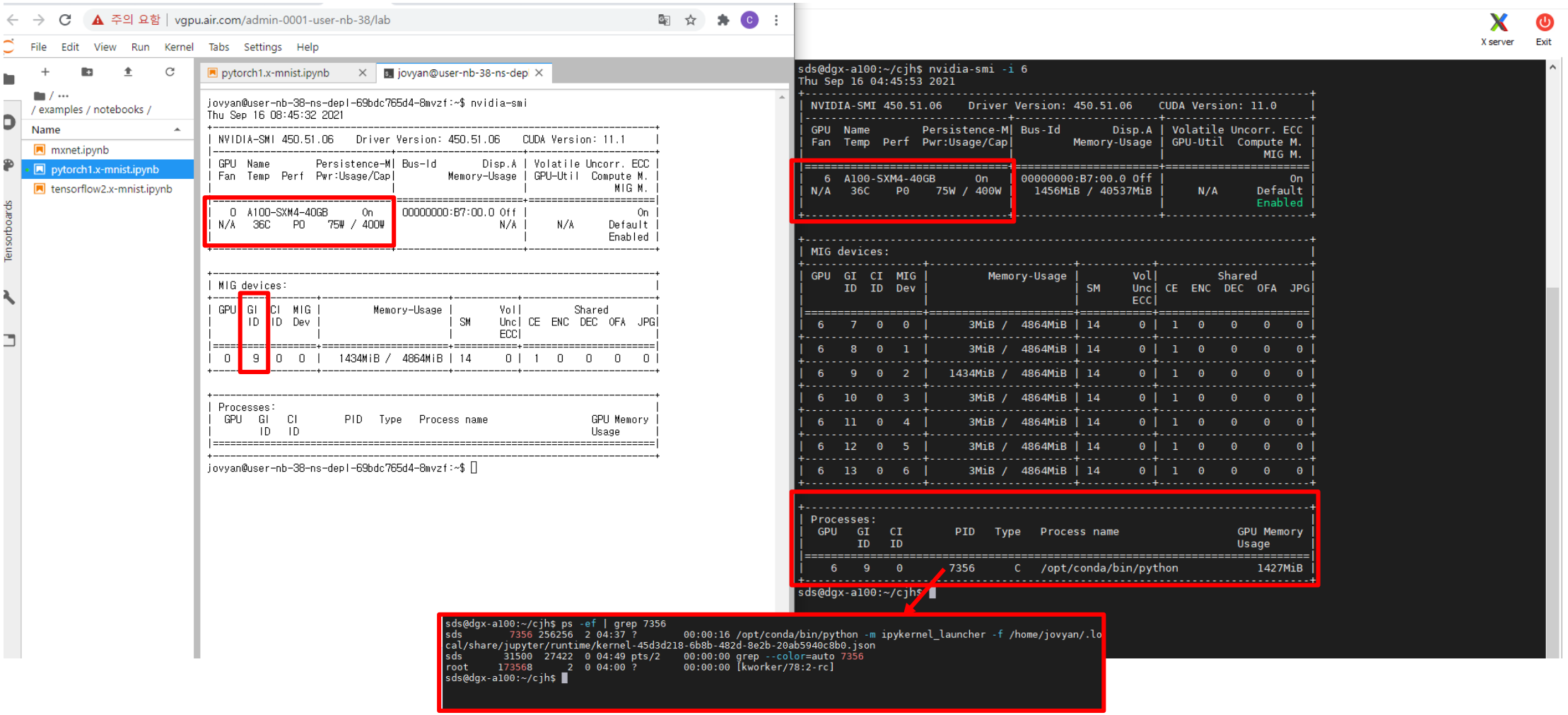
What is Multi-Instance GPU (MIG)?



Source : https://blogs.nvidia.com/blog/2020/05/14/multi-instance-gpus/

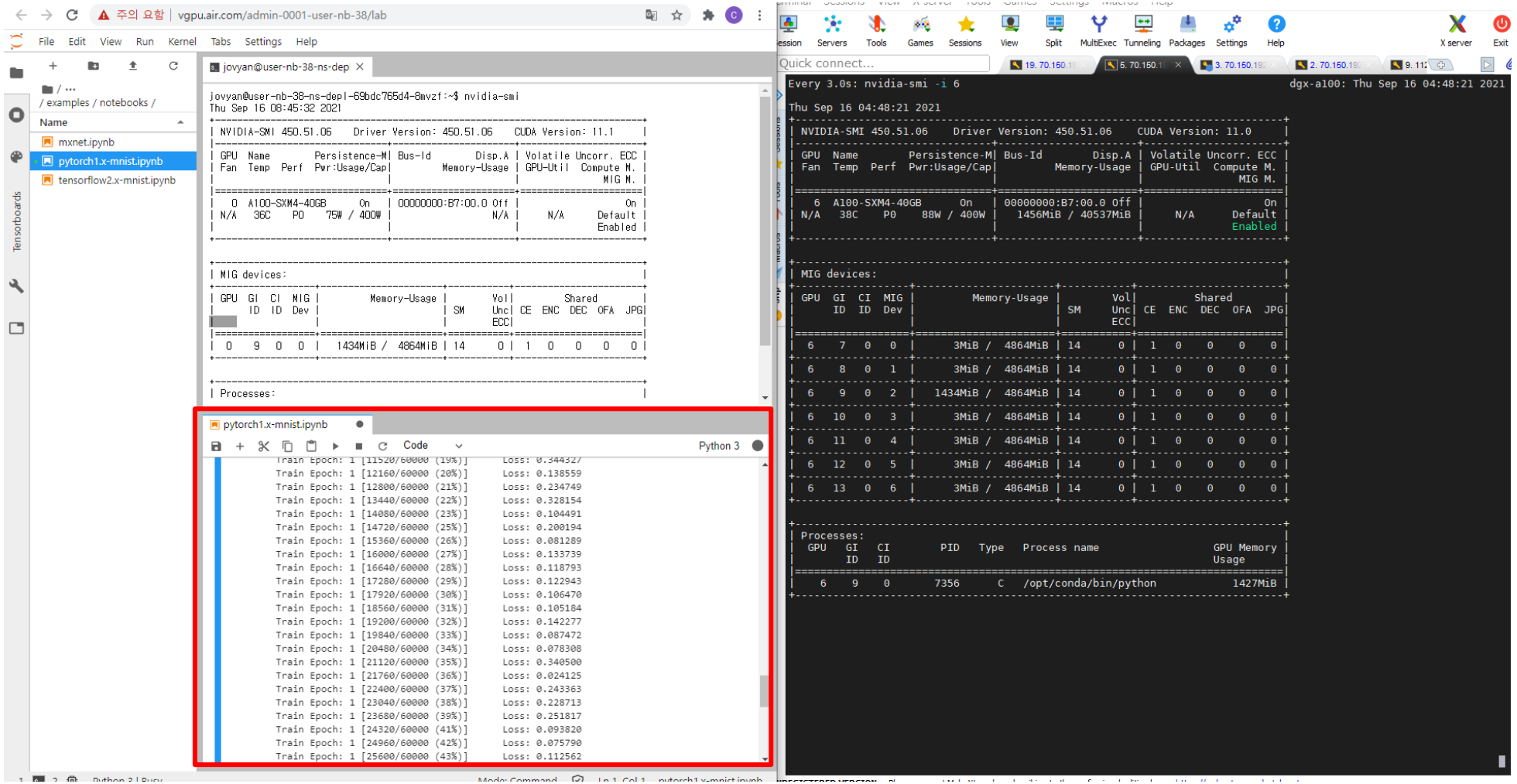
Case Study 1 : Multi-Instance GPU

1. Does it work well on Kubeflow?



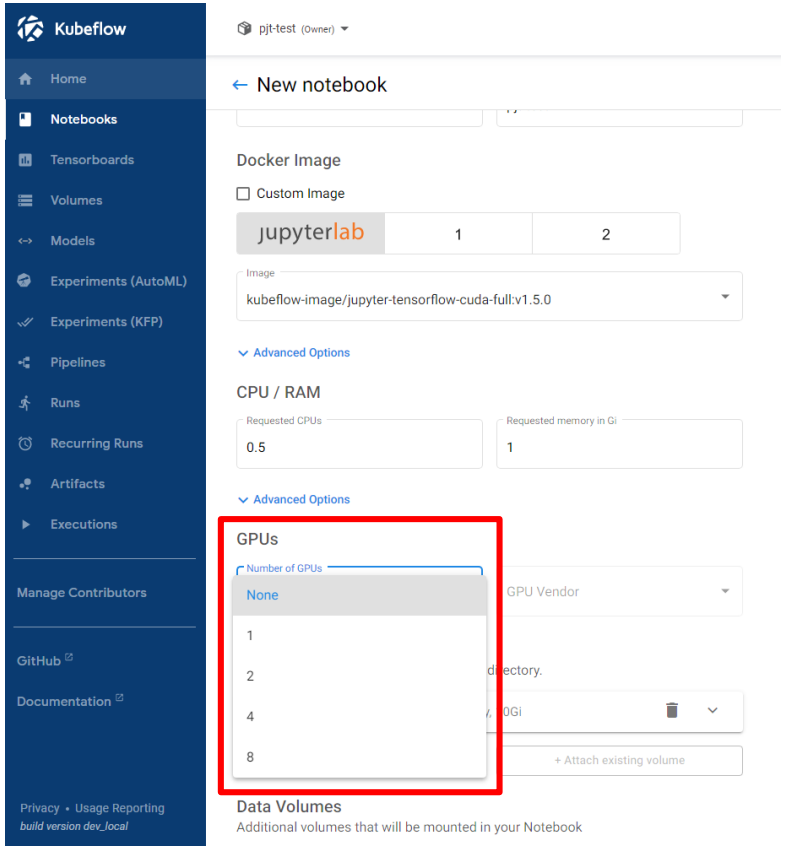
Case Study 1 : Multi-Instance GPU

1. Does it work well on Kubeflow?

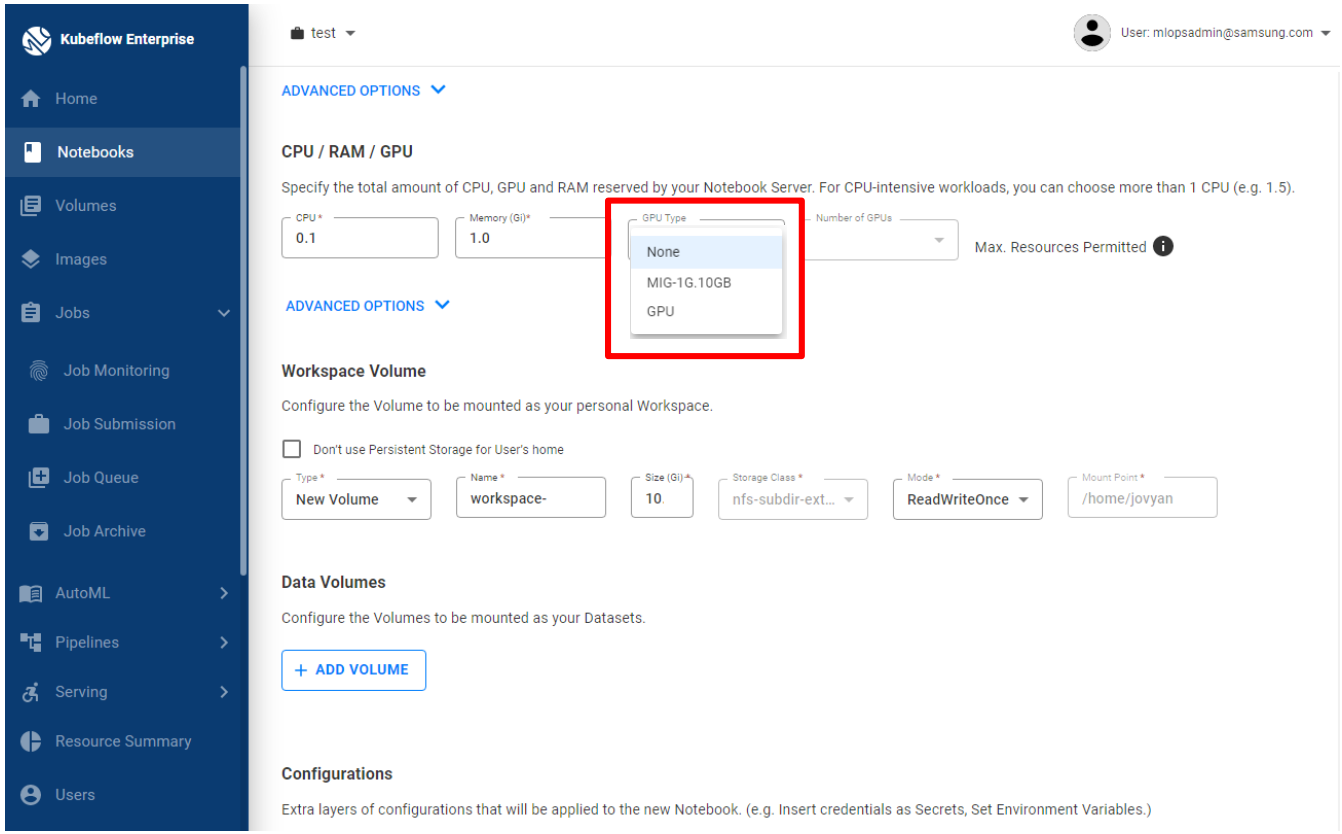


Case Study 1 : Multi-Instance GPU

1. Does it work well on Kubeflow?



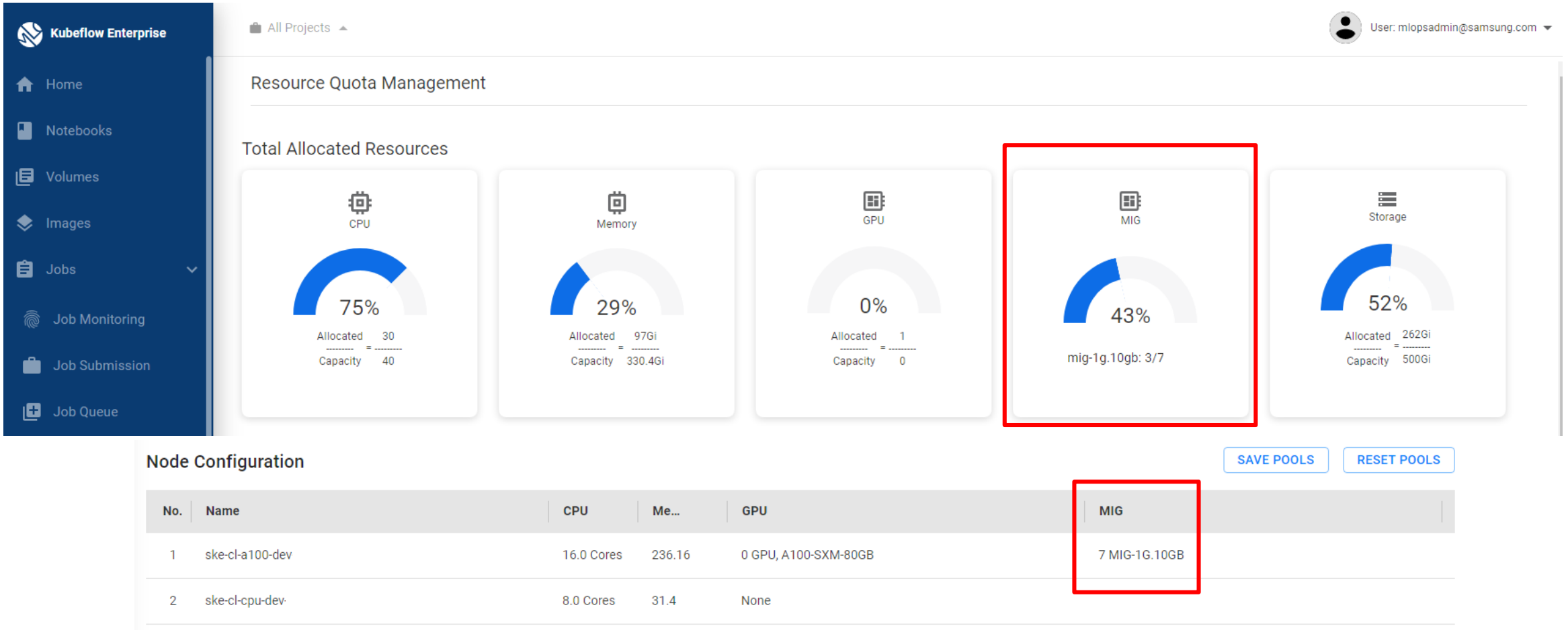
Open Source Kubeflow



SCP Kubeflow Service

Case Study 1 : Multi-Instance GPU

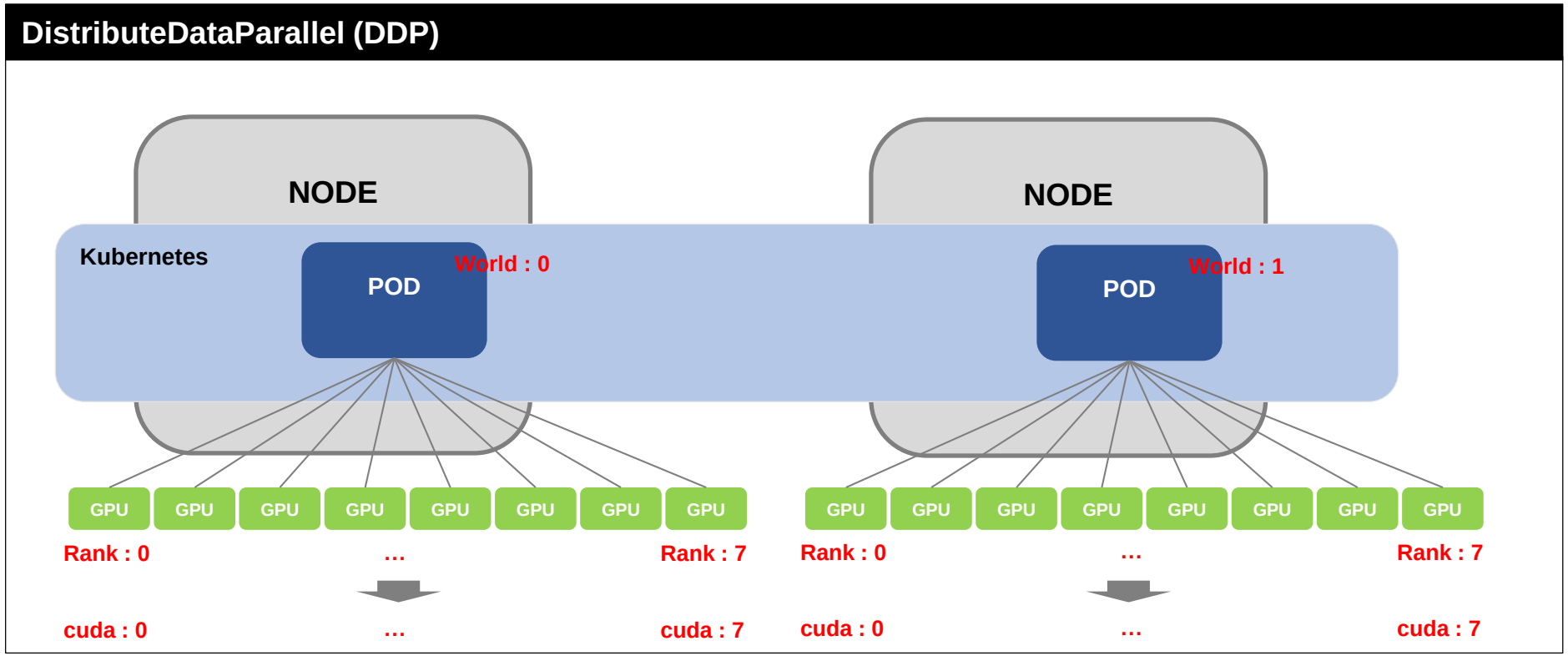
1. Does it work well on Kubeflow?



SCP Kubeflow Service

Case Study 1 : Multi-Instance GPU

2. Is Distributed Training Feasible? (only one MIG device can be assigned per POD)




```
device = torch.device("cuda:{}".format(rank))
```

- World Size: Total Number of Process
- Rank : Name of Process

Case Study 1 : Multi-Instance GPU

2. Is Distributed Training feasible? (only one MIG device can be assigned per POD)

```yaml
kind: PyTorchJob
metadata:
  name: pytorch-albert-4gpu
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      template:
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/gpu: 4
    Worker:
      replicas: 1
      restartPolicy: OnFailure
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/gpu: 4
```

Total 8 GPUs

<4 GPUs per POD, Total 2 PODs>

```
Every 2.0s: nvidia-smi

Wed Oct 20 09:58:54 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.57.02    Driver Version: 470.57.02    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| 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 A100-SXM...  Off  | 00000000:07:00.0 Off |                    0 |
| N/A   30C    P0    63W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-SXM...  Off  | 00000000:0B:00.0 Off |                    0 |
| N/A   30C    P0    65W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   2  NVIDIA A100-SXM...  Off  | 00000000:48:00.0 Off |                    0 |
| N/A   30C    P0    66W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   3  NVIDIA A100-SXM...  Off  | 00000000:4C:00.0 Off |                    0 |
| N/A   31C    P0    69W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   4  NVIDIA A100-SXM...  Off  | 00000000:88:00.0 Off |                    0 |
| N/A   29C    P0    66W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   5  NVIDIA A100-SXM...  Off  | 00000000:8B:00.0 Off |                    0 |
| N/A   30C    P0    67W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   6  NVIDIA A100-SXM...  Off  | 00000000:C8:00.0 Off |                    0 |
| N/A   30C    P0    68W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   7  NVIDIA A100-SXM...  Off  | 00000000:CB:00.0 Off |                    0 |
| N/A   30C    P0    66W / 400W |   1012MiB / 81251MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A   4044499      C   /opt/conda/bin/python            1009MiB |
|    1   N/A  N/A   4044500      C   /opt/conda/bin/python            1009MiB |
|    2   N/A  N/A   4044501      C   /opt/conda/bin/python            1009MiB |
|    3   N/A  N/A   4044502      C   /opt/conda/bin/python            1009MiB |
|    4   N/A  N/A   4041834      C   /opt/conda/bin/python            1009MiB |
|    5   N/A  N/A   4041840      C   /opt/conda/bin/python            1009MiB |
|    6   N/A  N/A   4041842      C   /opt/conda/bin/python            1009MiB |
|    7   N/A  N/A   4041843      C   /opt/conda/bin/python            1009MiB |
+-----------------------------------------------------------------------------+
```

Case Study 1 : Multi-Instance GPU

2. Is Distributed Training feasible? (only one MIG device can be assigned per POD)

Case Study 1 : Multi-Instance GPU

2. Is Distributed Training feasible? (only one MIG device can be assigned per POD)

```yaml
kind: PyTorchJob
metadata:
  name: pytorch-albert-mig2
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      template:
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/mig-1g.10gb: 2
    Worker:
      replicas: 1
      restartPolicy: OnFailure
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/mig-1g.10gb: 2
```

Total 4 MIGs

<2 MIGs per POD, Total 2 PODs>

```
root@ac-a100-10:~/cjh# kubectl logs -f pytorch-albert-mig2-master-0
World Size: : 2
Number of Gpus : 2
Node Rank : 0
Master Addr : localhost
Master Port : 23456
Params :  --do_train --fp16 --train_batch_size 1 --gradient_accumulation_steps 1 --num_train
b_disable 1 --nccl_ib_gdr_level 0
*****************************************
Setting OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid y
 for optimal performance in your application as needed.
*****************************************
============================================================
Train Batch Size:              1
Gradient Acculumation Steps:   1
NCCL Debug:           INFO
NCCL_DEBUG_SUBSYS:    INIT
NCCL_IB_DISABLE:      1
NCCL_IB_GDR_LEVEL:    0
Model Name:           albert
Config File:          /workspace/src/bert/albert_config_large_with_dropout.json
============================================================
Traceback (most recent call last):
  File "/workspace/src/bert/benchmark.py", line 2250, in <module>
    main()
  File "/workspace/src/bert/benchmark.py", line 2207, in main
    torch.cuda.set_device(args.local_rank)
  File "/opt/conda/lib/python3.6/site-packages/torch/cuda/__init__.py", line 263, in set_dev
    torch._C._cuda_setDevice(device)
RuntimeError: CUDA error: invalid device ordinal
============================================================
```

Error!

Case Study 1 : Multi-Instance GPU

2. Is Distributed Training feasible? (only one MIG device can be assigned per POD)

```yaml
kind: PyTorchJob
metadata:
  name: pytorch-albert-mig2
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      template:
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/mig-1g.10gb: 1
    Worker:
      replicas: 1
      restartPolicy: OnFailure
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          containers:
            image:
            name: pytorch
            resources:
              limits:
                nvidia.com/mig-1g.10gb: 1
```

Total 2 MIGs

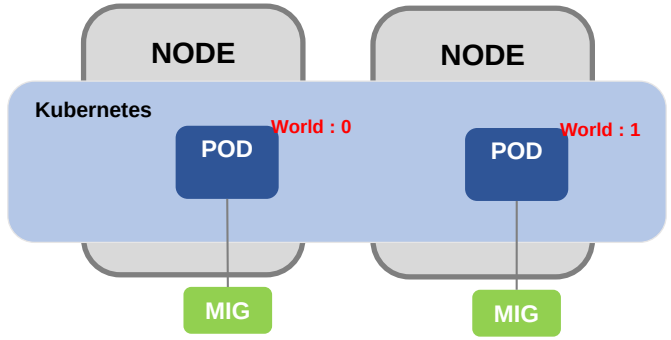
<1 MIGs per POD, Total 2 PODs>

```
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0    8    0    3595546      C   /opt/conda/bin/python             585MiB |
|    5   13    0    3594150      C   /opt/conda/bin/python             585MiB |
```

```
Init Process
[__main__] device: cuda:0 n_gpu: 1, distributed training: True, 16-bits training: True
Train Start
WARNING: Output directory /workspace/shared-dir/bert/results/squad ... 281181_2e-05
korquard_config.pt', 'albert_korquard_model_final.pt']
LOADING CHECKPOINT
LOADED CHECKPOINT
to device
Selected optimization level O2:  FP16 training with FP32 batchnorm and FP32 master weights.
```

Running!

DistributeDataParallel (DDP)



Case Study 1 : Multi-Instance GPU

3. Performance

□ Model

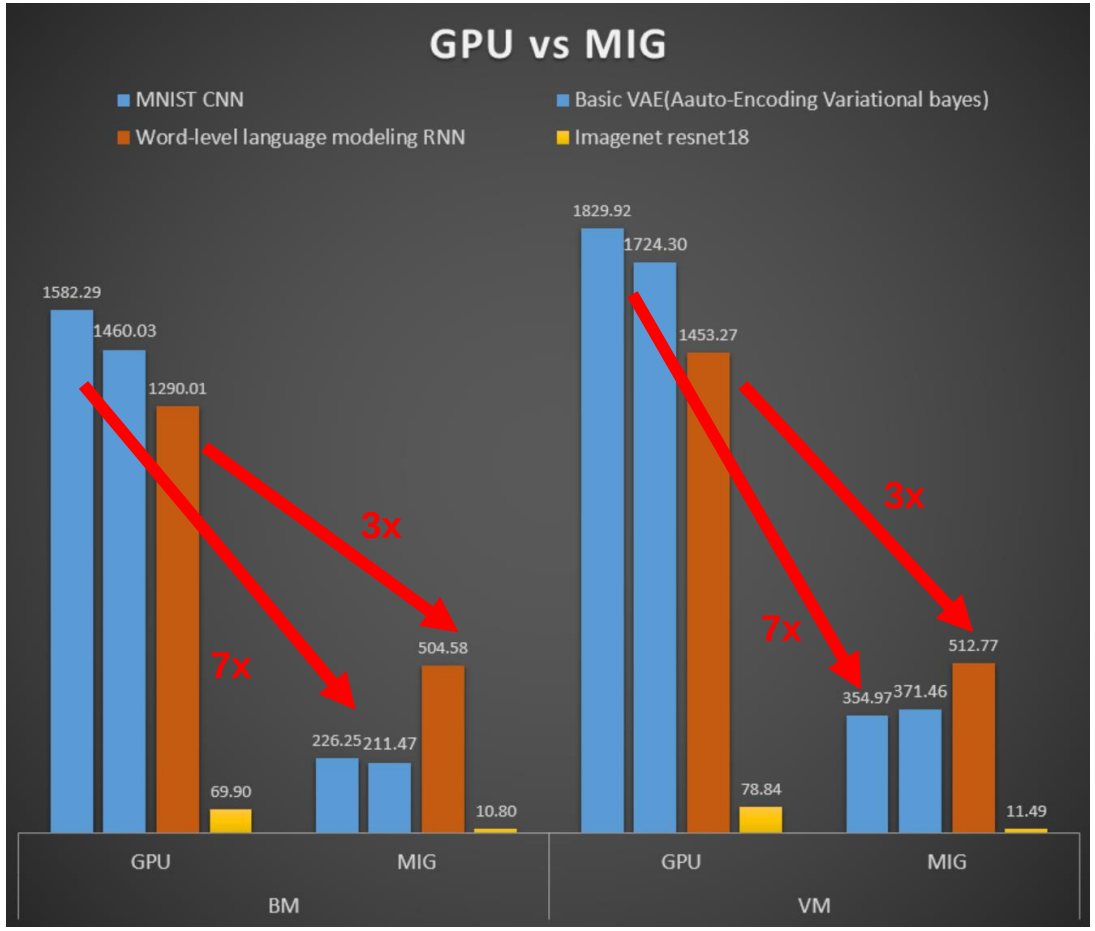
Dataset	MNIST	Basic VAE	ImageNet	World-Level Languge
ML Models	CNN		Resnet18	RNN
ML Frameworks	PyTorch			

□ How to Test

- GPU : Sequentially 7 times on 1 GPU
- MIG : Parallelly 1 time on 7 MIGs (1g.10gb)

□ Result

- Comparing **Total Execution Time**
  . CNN : 500-700%
  . Resnet18 : 700%
  . RNN : 300%

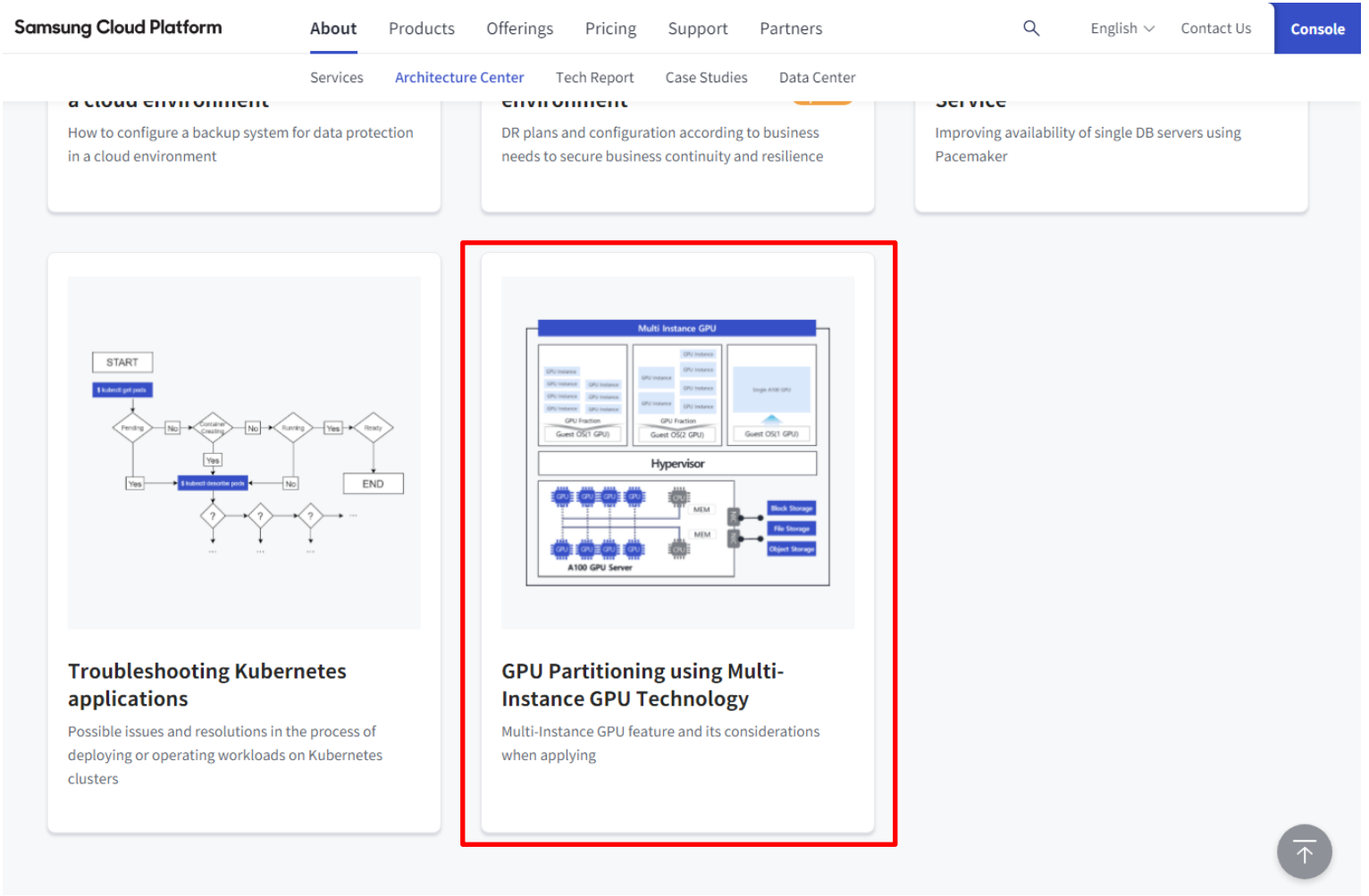


Case Study 1 : Multi-Instance GPU

KubeCon CloudNativeCon North America 2022

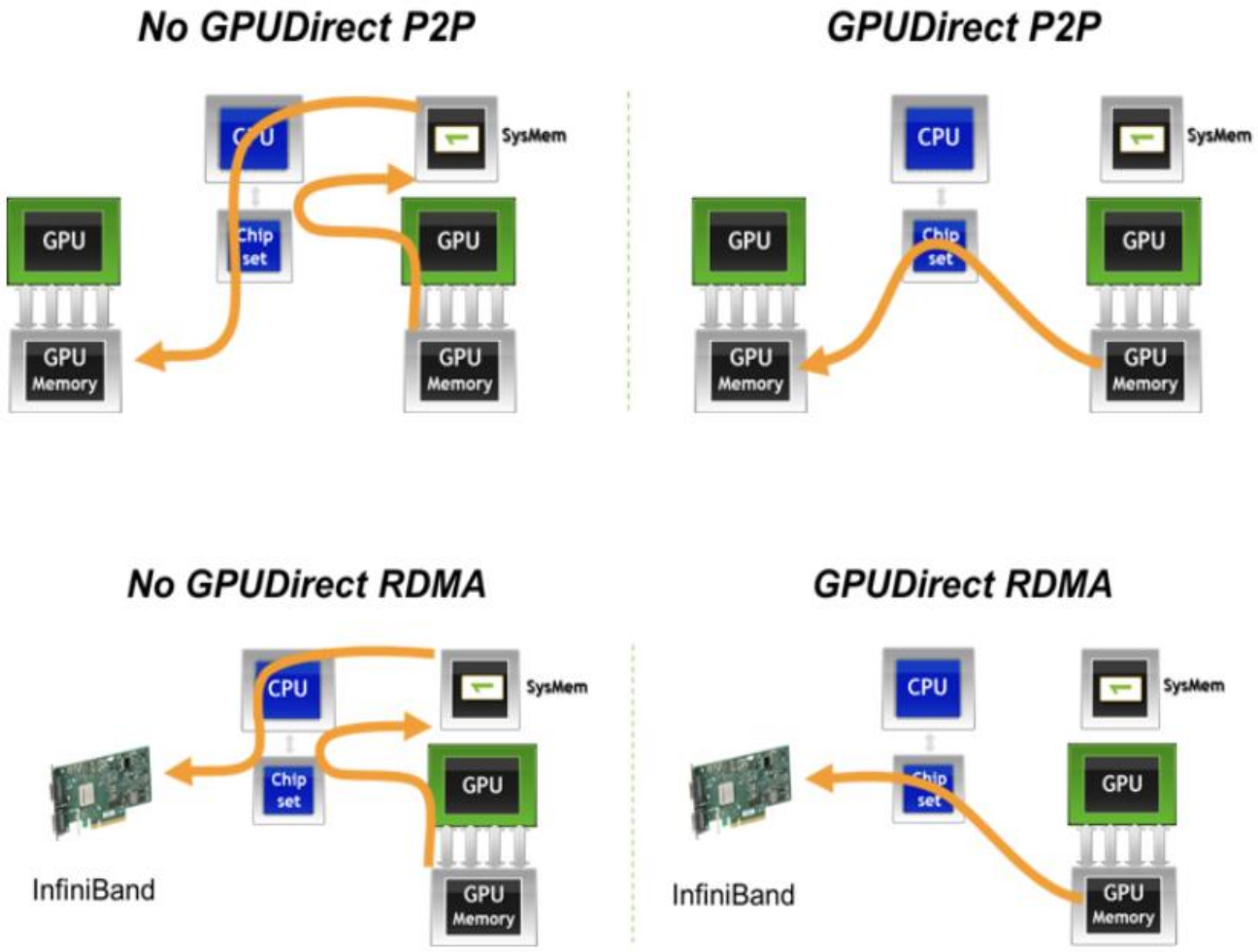
https://cloud.samsungsds.com/

Architecture Center > Technical Guides



Case Study 2 : GPUDirect RDMA

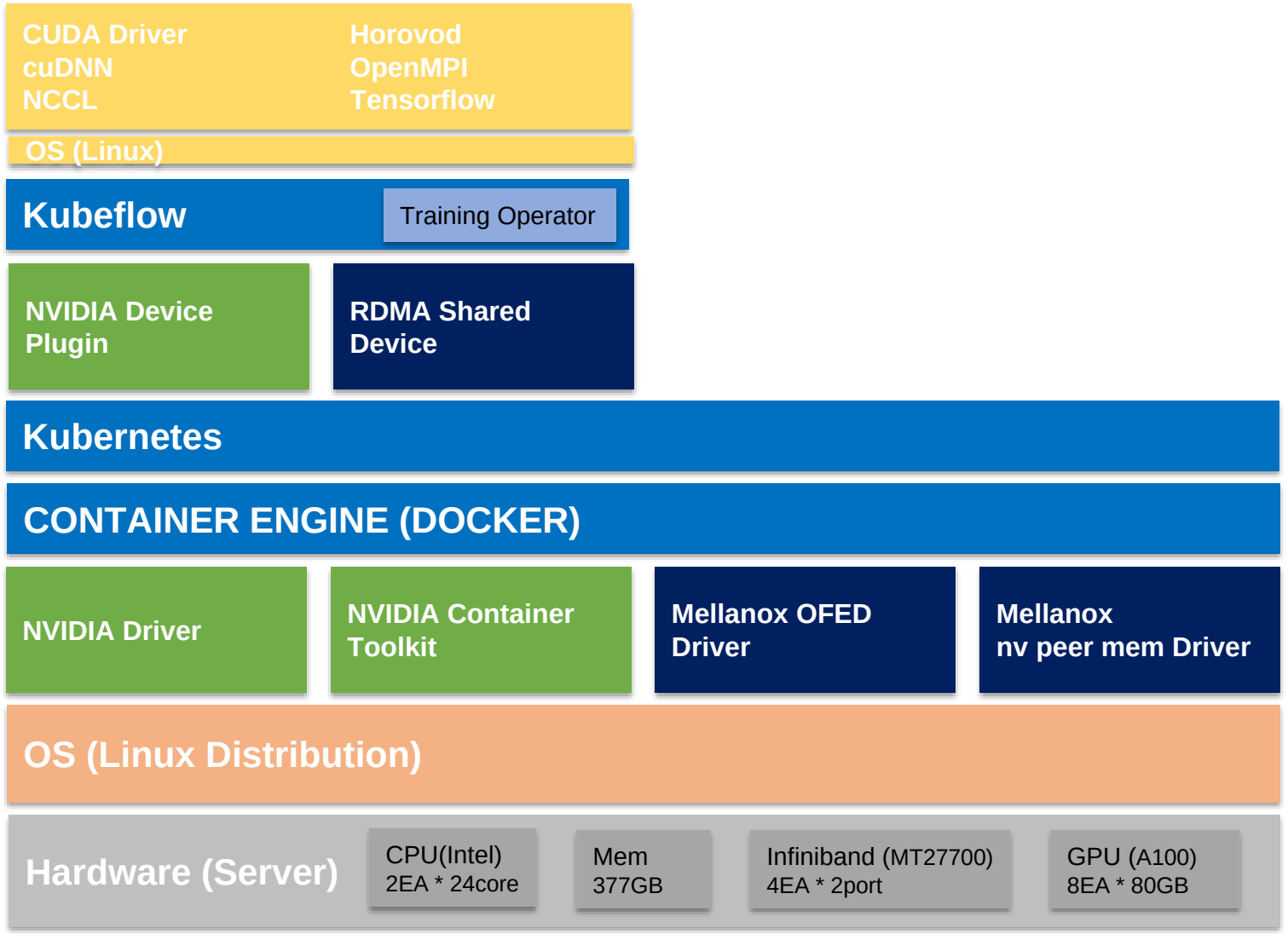
What is GPUDirect RDMA(Remote Direct Memory Access)?



Source :https://developer.nvidia.com/blog/introduction-cuda-aware-mpi/

Case Study 2 : GPUDirect RDMA

Software Architecture



Case Study 2 : GPUDirect RDMA

Examples

```yaml
apiVersion: "kubeflow.org/v1"
kind: "PyTorchJob"
metadata:
  name: "pytorch-flower-rdma"
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      restartPolicy: OnFailure
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          containers:
            - name: pytorch
              image:
              command: ["python","/workspace/jovyan/src/launch.py","--dist","2“, "--nccl_ib_disable","0","--nccl_ib_gdr_level","1"]
              # Comment out the below resources to use the CPU.
              resources:
                limits:
                  nvidia.com/gpu: 8
                  rdma/hca_shared_devices_a: "1"
              securityContext:
                capabilities:
                  add: [ "IPC_LOCK" ]
```

OS ENV

Kubernetes Resource
& SecurityContext

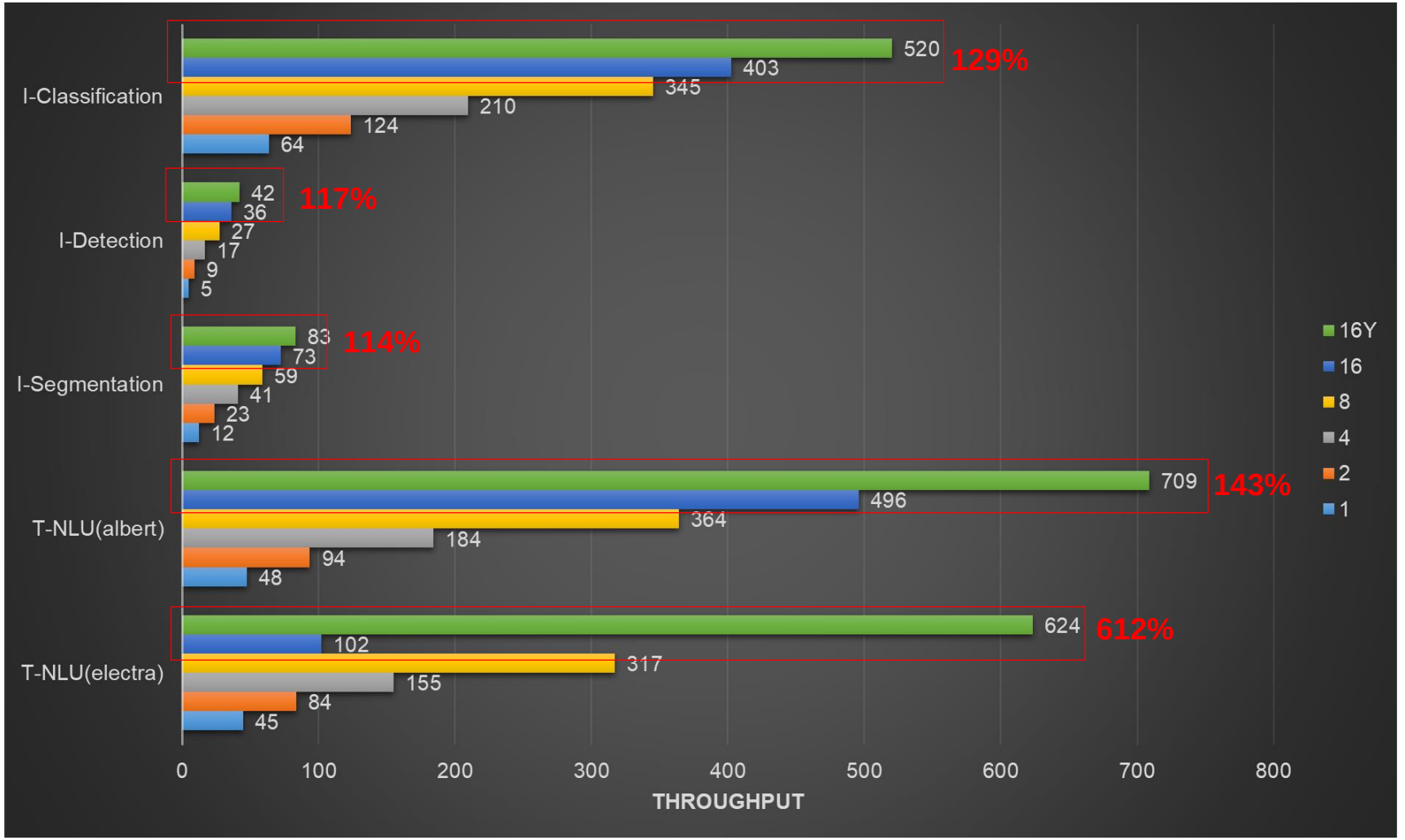
Case Study 2 : GPUDirect RDMA

Examples

```
World Size:          2
Number of Gpus:      8
Dist World Size:     16 (World Size x Number of Gpus)
Node Rank:           0
NCCL_IB_DISABLE:     0
NCCL_IB_GDR_LEVEL:   1
(중략)
INFO: =============   start   =============
INFO: ## model_type             : segmentation
INFO: ## model_name             : unet
INFO: ## model_input    : [512,512,3]
INFO: ## gpu_name               : A100 Graphics Device
INFO: ## gpus                   : 16
INFO: ## epochs                 : 5
INFO: ## batch_size             : 128
pytorch-flower-rdma-master-0:149:149 [0] NCCL INFO Bootstrap : Using [0]eth0:40.244.64.19<0>
pytorch-flower-rdma-master-0:149:149 [0] NCCL INFO NET/Plugin : No plugin found (libnccl-net.so), using internal implementation
pytorch-flower-rdma-master-0:149:149 [0] NCCL INFO NCCL_IB_DISABLE set by environment to 0.
pytorch-flower-rdma-master-0:149:149 [0] NCCL INFO NET/IB : Using [0]mlx5_0:1/IB [1]mlx5_2:1/IB [2]mlx5_5:1/RoCE [3]mlx5_7:1/IB ; OOB
eth0:40.244.64.19<0>
pytorch-flower-rdma-master-0:149:149 [0] NCCL INFO Using network IB
NCCL version 2.7.8+cuda11.1
(중략)
pytorch-flower-rdma-master-0:154:299 [5] NCCL INFO GPU Direct RDMA Enabled for GPU 8b000 / HCA 3 (distance 3 <= 3), read 0
pytorch-flower-rdma-master-0:153:300 [4] NCCL INFO Channel 01 : 4[88000] -> 3[4c000] via P2P/IPC/read
pytorch-flower-rdma-master-0:156:296 [7] NCCL INFO GPU Direct RDMA Enabled for GPU cb000 / HCA 1 (distance 3 <= 3), read 1
pytorch-flower-rdma-master-0:154:299 [5] NCCL INFO Channel 01 : 10[48000] -> 5[8b000] [receive] via NET/IB/3/GDRDMA
pytorch-flower-rdma-master-0:156:296 [7] NCCL INFO Channel 02 : 7[cb000] -> 12[88000] [send] via NET/IB/1/GDRDMA
(중략)
INFO: =============   end   =============
```

Case Study 2 : GPUDirect RDMA

Performance



※ throughput = total_examples / exec_time

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Please scan the QR Code above to
leave feedback on this session