

# Kubeflow *as-a-service* on HPC clusters – first experience

Mohsin Ahmed Shaikh<sup>1</sup>, Nasr Hassanien<sup>2</sup>, Islam Elmas<sup>2</sup>, Saber Feki<sup>1</sup>

1: KAUST Supercomputing Lab, 2: Brightskies Inc

Correspondance: [mohsin.shaikh@kaust.edu.sa](mailto:mohsin.shaikh@kaust.edu.sa)

# KAUST Supercomputing Core Lab

- Shaheen 2 (Flagship)
  - Cray XC40: 6174 nodes (Intel Haswell 32 cores 128GB)
- Ibex cluster:
  - Heterogeneous CPU (Intel/AMD): > 22K cores (250GB-3TB memory)
  - Heterogeneous GPU (NVIDIA): > 600 GPUs
- Shaheen 3 (Flagship refresh – Coming Soon)
  - HPE Cray EX4000: CPUs + GPUs

**SLURM Everywhere**

# User Personas – by jobtypes

- SLURM batch jobs
  - Large scale HPC (CFD, CompChem, Earth Sci, GeoSci, Bioscience, Datascience)
  - Singleton, arrays, dependency, burst buffer
  - Client/Server (Dask, Ray)
- SLURM Interactive jobs
  - Datascience on GPUs – model development
  - Jupyter, VS Code

# Motivation

- Ease-of-use for Interactive computing
- Enable resilient + reproducible workflows
- Cross-platform portability -- workstation/cloud/HPC cluster

# Kubeflow

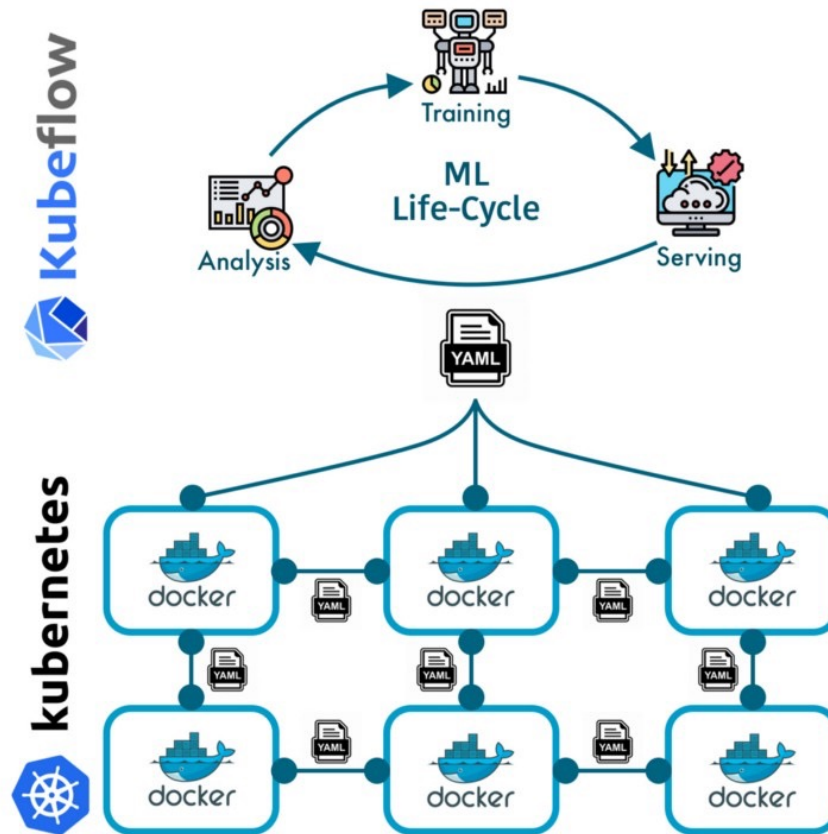
# Kubeflow

Kubeflow is a framework for developing data science models and workflows

Depends on Kubernetes and abstract its use through a GUI

Kubeflow *as a service* provides:

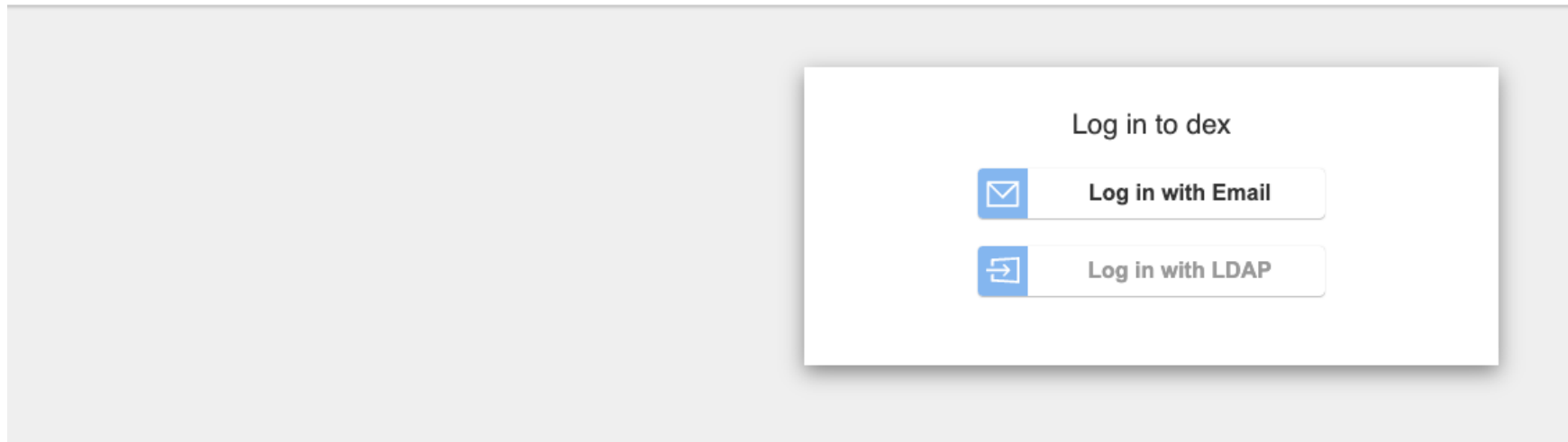
- Interactive notebooks
- Kubeflow pipelines – create reproducible workflows
- Katib – hyperparameter optimization experiments
- Kserve – model serving platform



# Functionality tested so far ...

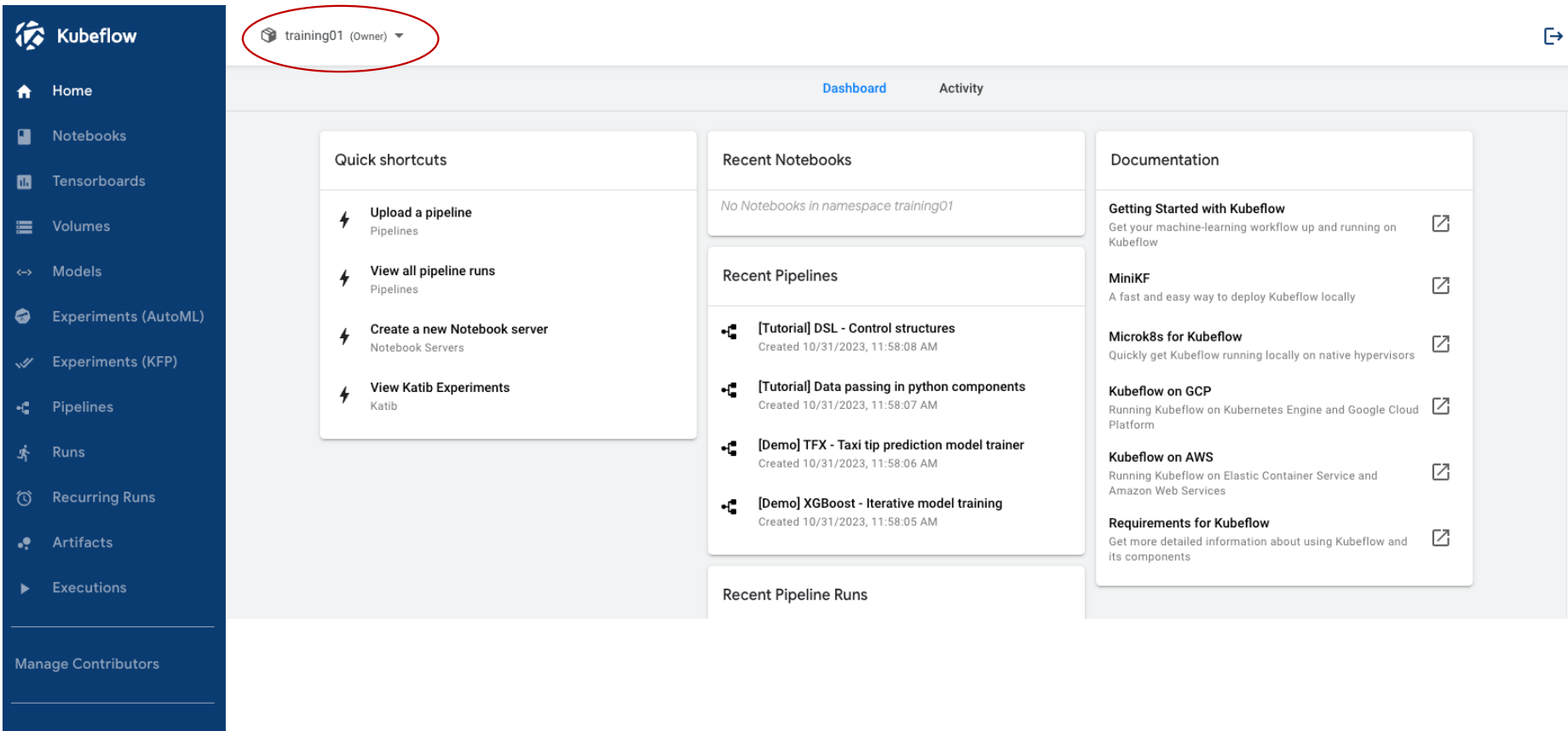
- Interactive computing
- Distributed training

# Kubeflow Authentication





# Main dashboard



The screenshot shows the Kubeflow Main dashboard. On the left is a dark blue sidebar with the Kubeflow logo and a list of navigation items: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, Executions, and Manage Contributors. At the top of the main content area, there is a header bar with the namespace 'training01 (Owner)' circled in red, and tabs for 'Dashboard' and 'Activity'. The dashboard is divided into several sections: 'Quick shortcuts' with links to 'Upload a pipeline', 'View all pipeline runs', 'Create a new Notebook server', and 'View Katib Experiments'; 'Recent Notebooks' showing 'No Notebooks in namespace training01'; 'Recent Pipelines' listing four items with their creation times; 'Recent Pipeline Runs' (partially visible); and 'Documentation' with links to 'Getting Started with Kubeflow', 'MiniKF', 'Microk8s for Kubeflow', 'Kubeflow on GCP', 'Kubeflow on AWS', and 'Requirements for Kubeflow'.

**Kubeflow**

training01 (Owner)

Dashboard Activity

**Quick shortcuts**

- ⚡ Upload a pipeline  
Pipelines
- ⚡ View all pipeline runs  
Pipelines
- ⚡ Create a new Notebook server  
Notebook Servers
- ⚡ View Katib Experiments  
Katib

**Recent Notebooks**

No Notebooks in namespace training01

**Recent Pipelines**

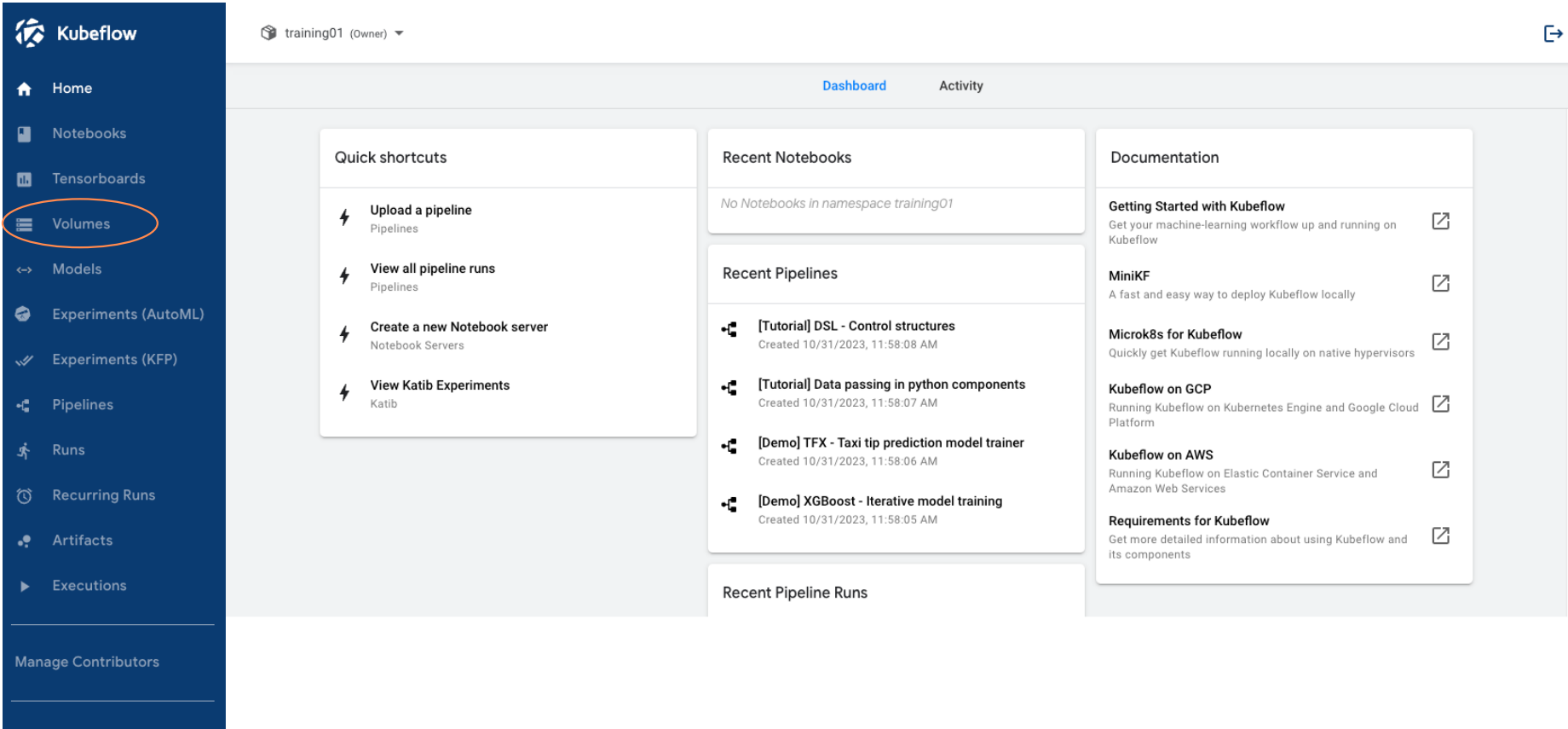
- 🔧 [Tutorial] DSL - Control structures  
Created 10/31/2023, 11:58:08 AM
- 🔧 [Tutorial] Data passing in python components  
Created 10/31/2023, 11:58:07 AM
- 🔧 [Demo] TFX - Taxi tip prediction model trainer  
Created 10/31/2023, 11:58:06 AM
- 🔧 [Demo] XGBoost - Iterative model training  
Created 10/31/2023, 11:58:05 AM

**Recent Pipeline Runs**

**Documentation**

- Getting Started with Kubeflow**  
Get your machine-learning workflow up and running on Kubeflow
- MiniKF**  
A fast and easy way to deploy Kubeflow locally
- Microk8s for Kubeflow**  
Quickly get Kubeflow running locally on native hypervisors
- Kubeflow on GCP**  
Running Kubeflow on Kubernetes Engine and Google Cloud Platform
- Kubeflow on AWS**  
Running Kubeflow on Elastic Container Service and Amazon Web Services
- Requirements for Kubeflow**  
Get more detailed information about using Kubeflow and its components

# Persistent Volume -- attachable



The image shows the Kubeflow dashboard interface. On the left is a dark blue sidebar with the Kubeflow logo and a list of navigation items: Home, Notebooks, Tensorboards, Volumes (highlighted with an orange circle), Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, Executions, and Manage Contributors. The main content area is titled 'training01 (Owner)' and has tabs for 'Dashboard' and 'Activity'. It features several panels: 'Quick shortcuts' with links to upload a pipeline, view all pipeline runs, create a new notebook server, and view Katib experiments; 'Recent Notebooks' showing 'No Notebooks in namespace training01'; 'Recent Pipelines' listing four tutorials and demos with their creation times; 'Recent Pipeline Runs'; and 'Documentation' with links to getting started, MiniKF, Microk8s, Kubeflow on GCP, Kubeflow on AWS, and requirements.

Kubeflow

training01 (Owner)

Dashboard Activity

Quick shortcuts

- ⚡ Upload a pipeline  
Pipelines
- ⚡ View all pipeline runs  
Pipelines
- ⚡ Create a new Notebook server  
Notebook Servers
- ⚡ View Katib Experiments  
Katib

Recent Notebooks

No Notebooks in namespace training01

Recent Pipelines

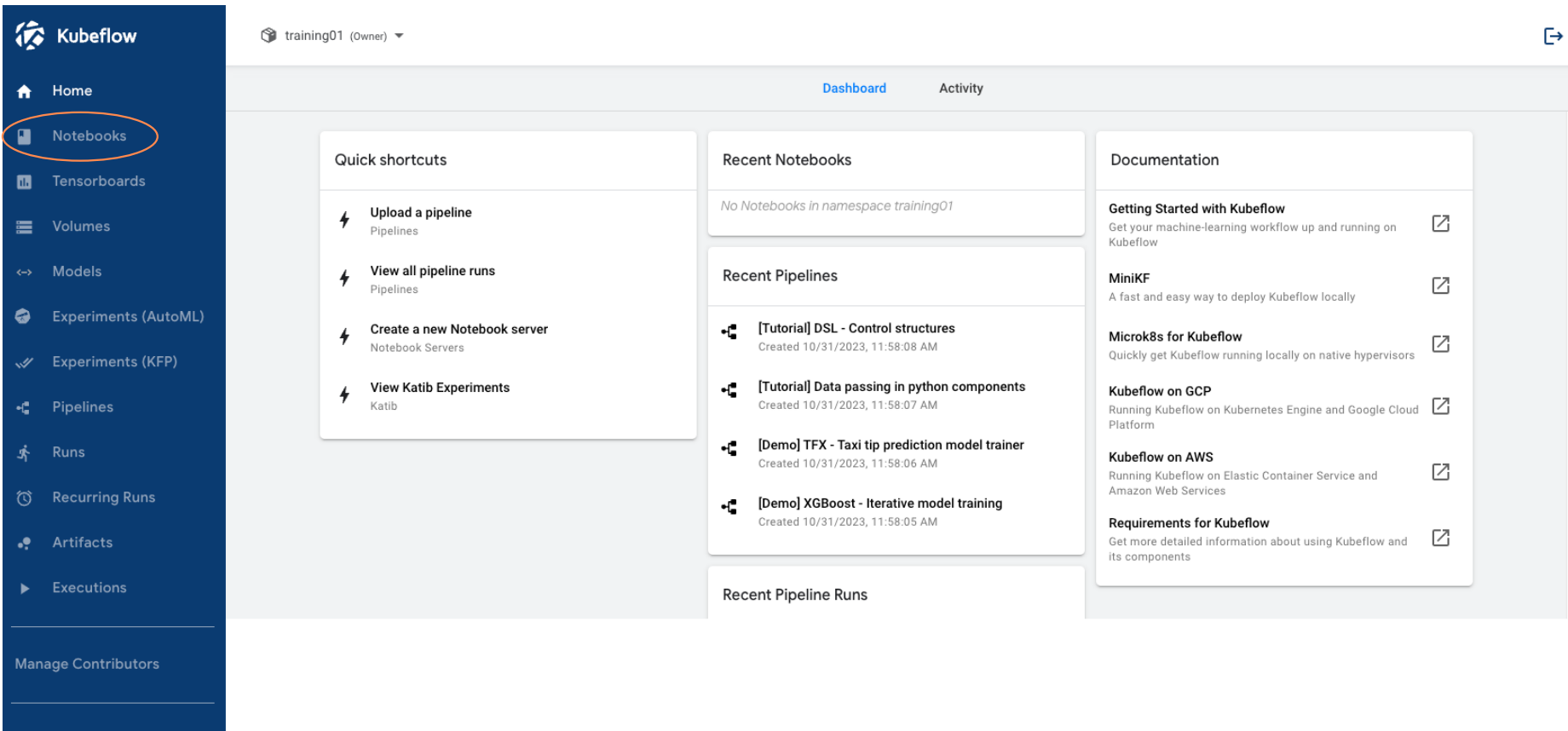
- 🔧 [Tutorial] DSL - Control structures  
Created 10/31/2023, 11:58:08 AM
- 🔧 [Tutorial] Data passing in python components  
Created 10/31/2023, 11:58:07 AM
- 🔧 [Demo] TFX - Taxi tip prediction model trainer  
Created 10/31/2023, 11:58:06 AM
- 🔧 [Demo] XGBoost - Iterative model training  
Created 10/31/2023, 11:58:05 AM

Recent Pipeline Runs

Documentation

- Getting Started with Kubeflow  
Get your machine-learning workflow up and running on Kubeflow
- MiniKF  
A fast and easy way to deploy Kubeflow locally
- Microk8s for Kubeflow  
Quickly get Kubeflow running locally on native hypervisors
- Kubeflow on GCP  
Running Kubeflow on Kubernetes Engine and Google Cloud Platform
- Kubeflow on AWS  
Running Kubeflow on Elastic Container Service and Amazon Web Services
- Requirements for Kubeflow  
Get more detailed information about using Kubeflow and its components

# Creating Notebook



The image shows the Kubeflow dashboard interface. On the left is a dark blue sidebar with the Kubeflow logo and a list of navigation items: Home, Notebooks (highlighted with an orange circle), Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, Executions, and Manage Contributors. The main content area is light gray and displays the 'training01 (Owner)' namespace. It features a top navigation bar with 'Dashboard' and 'Activity' tabs. The dashboard is organized into several sections: 'Quick shortcuts' with links to upload pipelines, view pipeline runs, create a new notebook server, and view Katib experiments; 'Recent Notebooks' showing a message that no notebooks are present; 'Recent Pipelines' listing four recent pipeline runs with their titles and creation times; and 'Documentation' with links to getting started guides, MiniKF, Microk8s, Kubeflow on GCP, Kubeflow on AWS, and requirements. A 'Recent Pipeline Runs' section is partially visible at the bottom.

Kubeflow

training01 (Owner)

Dashboard Activity

Quick shortcuts

- ⚡ Upload a pipeline  
Pipelines
- ⚡ View all pipeline runs  
Pipelines
- ⚡ Create a new Notebook server  
Notebook Servers
- ⚡ View Katib Experiments  
Katib

Recent Notebooks

No Notebooks in namespace training01

Recent Pipelines


- 🔧 [Tutorial] DSL - Control structures  
Created 10/31/2023, 11:58:08 AM
- 🔧 [Tutorial] Data passing in python components  
Created 10/31/2023, 11:58:07 AM
- 🔧 [Demo] TFX - Taxi tip prediction model trainer  
Created 10/31/2023, 11:58:06 AM
- 🔧 [Demo] XGBoost - Iterative model training  
Created 10/31/2023, 11:58:05 AM

Recent Pipeline Runs

Documentation

- Getting Started with Kubeflow**  
Get your machine-learning workflow up and running on Kubeflow
- MiniKF**  
A fast and easy way to deploy Kubeflow locally
- Microk8s for Kubeflow**  
Quickly get Kubeflow running locally on native hypervisors
- Kubeflow on GCP**  
Running Kubeflow on Kubernetes Engine and Google Cloud Platform
- Kubeflow on AWS**  
Running Kubeflow on Elastic Container Service and Amazon Web Services
- Requirements for Kubeflow**  
Get more detailed information about using Kubeflow and its components

# Creating Notebook

 **Kubeflow**

- Home
- Notebooks**
- Tensorboards
- Volumes
- Models
- Experiments (AutoML)
- Experiments (KFP)
- Pipelines
- Runs
- Recurring Runs
- Artifacts
- Executions

 training01 (Owner) ▼

## ← New notebook

### Name

Name

Namespace

### Docker Image

☒ Custom Image

jupyterlab	1	2
------------	---	---

Custom Image


### ▼ Advanced Options

### CPU / RAM

Requested CPUs

Requested memory in Gi

# Creating Notebook

 **Kubeflow**

[Home](#)  
[Notebooks](#)  
[Tensorboards](#)  
[Volumes](#)  
[Models](#)  
[Experiments \(AutoML\)](#)  
[Experiments \(KFP\)](#)  
[Pipelines](#)  
[Runs](#)  
[Recurring Runs](#)  
[Artifacts](#)  
[Executions](#)

training01 (Owner) ▼

[← New notebook](#)

Additional volumes that will be mounted in your Notebook

[+ Add new volume](#) [+ Attach existing volume](#)

**Configurations**

Configurations  
Reference datasets and examples ▼

**Affinity / Tolerations**

Affinity Config  
Volta ▼


Tolerations Group  
None ▼


**Miscellaneous Settings**

☒ Enable Shared Memory

[LAUNCH](#) [CANCEL](#)

# Creating Notebook








 File Edit View Run Kernel Tabs Settings Help


 Filter files by name

/ kube-poc-examples / interactive\_usecase /

Name	Last Modified
dask	2 hours ago
logs	an hour ago
cifa10_pro...	2 hours ago
cifa10_vani...	an hour ago
cifa10_with...	2 hours ago
img_classi...	seconds ago
pytorch-en...	2 hours ago

img\_classifier\_ddp.ipynb

Code

PyTorch 1.12.1

```

os.environ['MASTER_PORT']='12355'
os.environ['DATA_DIR']='reference/CV/tiny-imagenet-200/'
os.environ['SLURM_JOBID']='100'
args=nb_args()

[*]: # Comment when using as python script
import multiprocessing as mp
num_processes = args.gpus
# NOTE: this is required for the `fork` method to work
net.share_memory()

processes = []
for rank in range(num_processes):
    p = mp.Process(target=main, args=(net,args))
    p.start()
    processes.append(p)
for p in processes:
    p.join()
  
```

/opt/conda/envs/pytorch-env/lib/python3.9/site-packages/torch/utils/data/dataloader.py:478: UserWarning: This DataLoader will create 10 worker processes in total. Our suggested max number of worker in current system is 2, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.

warnings.warn(\_create\_warning\_msg(

/opt/conda/envs/pytorch-env/lib/python3.9/site-packages/torch/utils/data/dataloader.py:478: UserWarning: This DataLoader will create 10 worker processes in total. Our suggested max number of worker in current system is 2, which is smaller than what this DataLoader is going to create. Please be aware that excessive worker creation might get DataLoader running slow or even freeze, lower the worker number to avoid potential slowness/freeze if necessary.

warnings.warn(\_create\_warning\_msg(

/opt/conda/envs/pytorch-env/lib/python3.9/site-packages/torch/cuda/memory.py:271: FutureWarning: torch.cuda.reset\_max\_memory\_allocated now calls torch.cuda.reset\_peak\_memory\_stats, which resets all/ peak memory stats.





warnings.warn(

/opt/conda/envs/pytorch-env/lib/python3.9/site-packages/torch/cuda/memory.py:271: FutureWarning: torch.cuda.reset\_max\_memory\_allocated now calls torch.cuda.reset\_peak\_memory\_stats, which resets all/ peak memory stats.

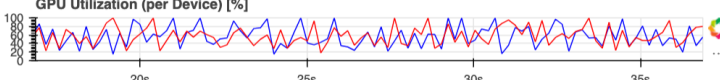

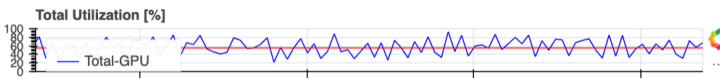
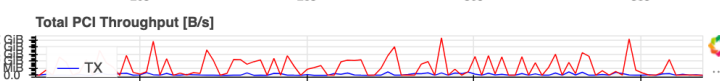
warnings.warn(

Starting training on GPU 1 of 2 -- Starting training on GPU 0 of 2 --

Machine Resources

GPU Resources

Simple

4 3

PyTorch 1.12.1 | Idle

Saving started

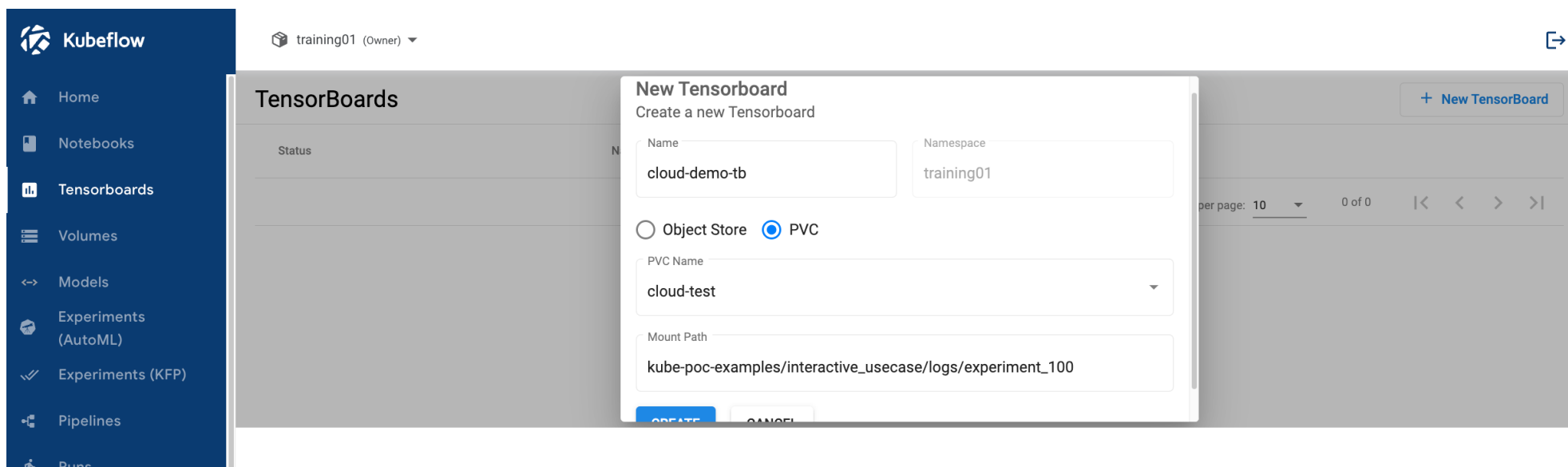
Mode: Command

Ln 1, Col 1

img\_classifier\_ddp.ipynb

1

# Tensorboard service



The screenshot displays the Kubeflow dashboard with a sidebar on the left containing navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, and Runs. The main content area is titled 'TensorBoards' and shows a table with a 'Status' column. A modal window titled 'New Tensorboard' is open, prompting the user to 'Create a new Tensorboard'. The modal contains the following fields and options:

- Name:** cloud-demo-tb
- Namespace:** training01
- Storage Type:** ☐ Object Store, ☒ PVC
- PVC Name:** cloud-test
- Mount Path:** kube-poc-examples/interactive\_usecase/logs/experiment\_100

At the bottom of the modal are 'CREATE' and 'CANCEL' buttons. In the background, a '+ New TensorBoard' button is visible in the top right corner of the TensorBoards section.

# Distributed training

## Training Operator -- PytorchJob



```

apiVersion: "kubeflow.org/v1"
kind: "PyTorchJob"
metadata:
  name: "DDP"
  namespace: training01
spec:
  pytorchReplicaSpecs:
    Worker:
      replicas: 1
      restartPolicy: Never
      template:
        metadata:
          annotations:
            sidecar.istio.io/inject: "false"
        spec:
          affinity:
            nodeAffinity:
              requiredDuringSchedulingIgnoredDuringExecution:
                nodeSelectorTerms:
                  - matchExpressions:
                      - key: nvidia.com/gpu.product
                        operator: In
                        values:
                          - "Tesla-V100-SXM2-32GB"

```

## containers:

```

- name: pytorch
  image: docker.io/mshaikh/kubeflow-demo:kf-dist-torch-vanilla
  imagePullPolicy: IfNotPresent
  command: [ "torchrun",
    "--nnodes", "1",
    "--nproc_per_node", "4",
    "--node_rank", "0",
    "/workspace/mnist/src/ddp.py",
    "--batch-size", "32",
    "--num-worker", "8",
    "--epochs", "4",
    "--lr", "0.001" ]
  env:
    - name: 'NCCL_DEBUG'
      value: 'INFO'
    - name: 'DATA_DIR'
      value: '/data/tiny-imagenet-200'
    - name: 'OMP_NUM_THREADS'
      value: '1'
  resources:
    limits:
      cpu: 8
      memory: '200Gi'
      nvidia.com/gpu: 4
  volumeMounts:
    - name: dshm
      mountPath: /dev/shm
  volumes:
    - emptyDir:
        medium: Memory
      name: dshm

```

+ NVIDIA-SMI 535.86.10			Driver Version: 535.86.10			CUDA Version: 12.2		
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC	
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.	MIG M.
0	Tesla V100-SXM2-32GB		Off	00000000:61:00.0	Off		0	
N/A	62C	P0	190W / 300W	5071MiB / 32768MiB		93%	Default	N/A
1	Tesla V100-SXM2-32GB		Off	00000000:62:00.0	Off		0	
N/A	61C	P0	226W / 300W	5031MiB / 32768MiB		94%	Default	N/A
2	Tesla V100-SXM2-32GB		Off	00000000:89:00.0	Off		0	
N/A	61C	P0	246W / 300W	5031MiB / 32768MiB		94%	Default	N/A
3	Tesla V100-SXM2-32GB		Off	00000000:8A:00.0	Off		0	
N/A	66C	P0	279W / 300W	5039MiB / 32768MiB		95%	Default	N/A

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	
	ID	ID				Usage	
0	N/A	N/A	2745778	C	/opt/conda/bin/python	5066MiB	
1	N/A	N/A	2745779	C	/opt/conda/bin/python	5026MiB	
2	N/A	N/A	2745780	C	/opt/conda/bin/python	5026MiB	
3	N/A	N/A	2745781	C	/opt/conda/bin/python	5034MiB	

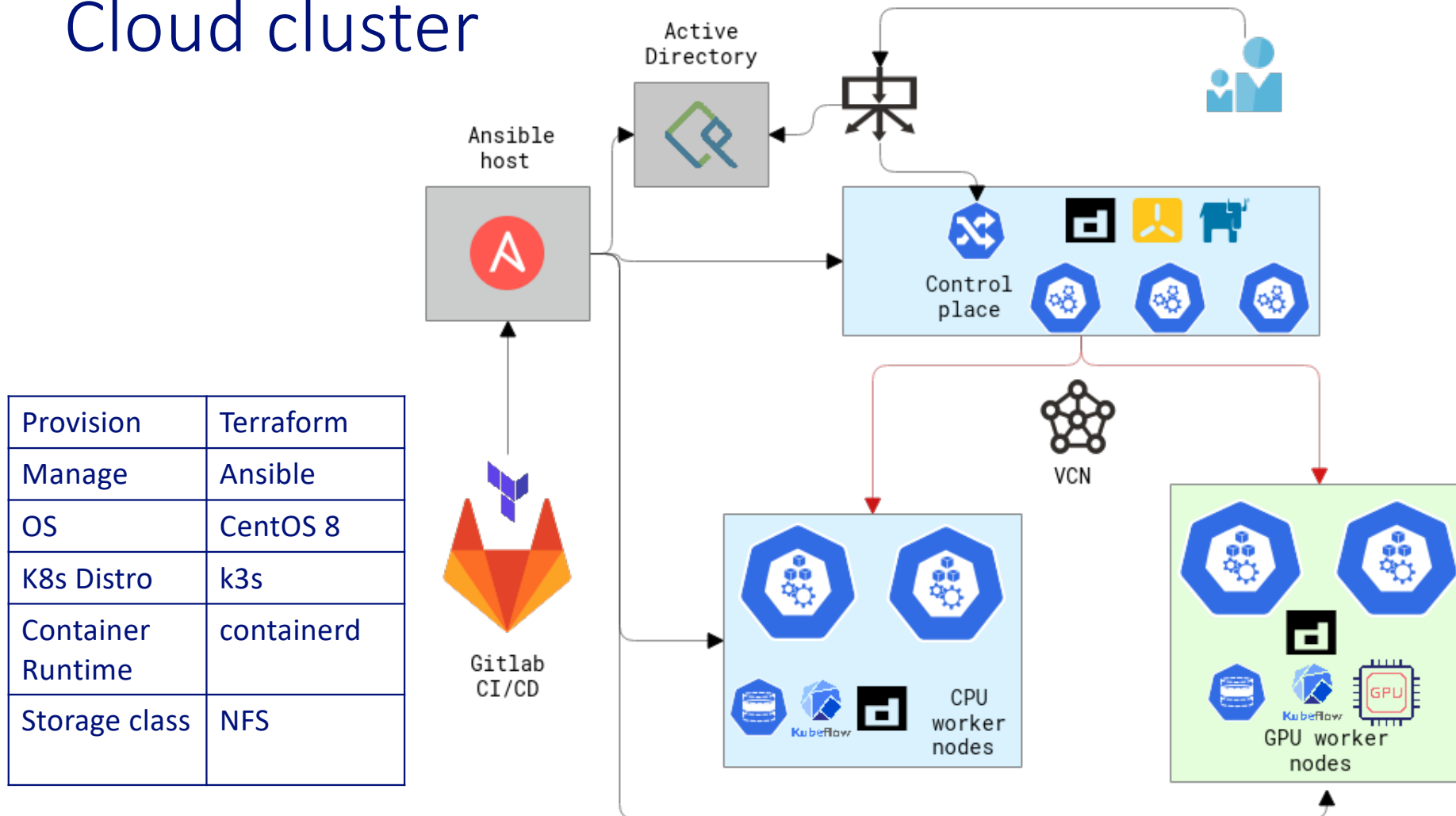
# SLURM vs Kubeflow

- PyTorch DDP training
  - Dataset: TinyImageNet200
  - Model: ResNet50
  - Nodes: 1
  - GPUs per node: 4 V100 SXM2 (32GB)
- Training for 20 epochs
  - Batch size: 256

SLURM		Kubeflow training opeator	
Time to solution(s)	Accuracy	Time to solution(s)	Accuracy
1939.232	48%	1881.21	48%

# Cluster Infrastructure

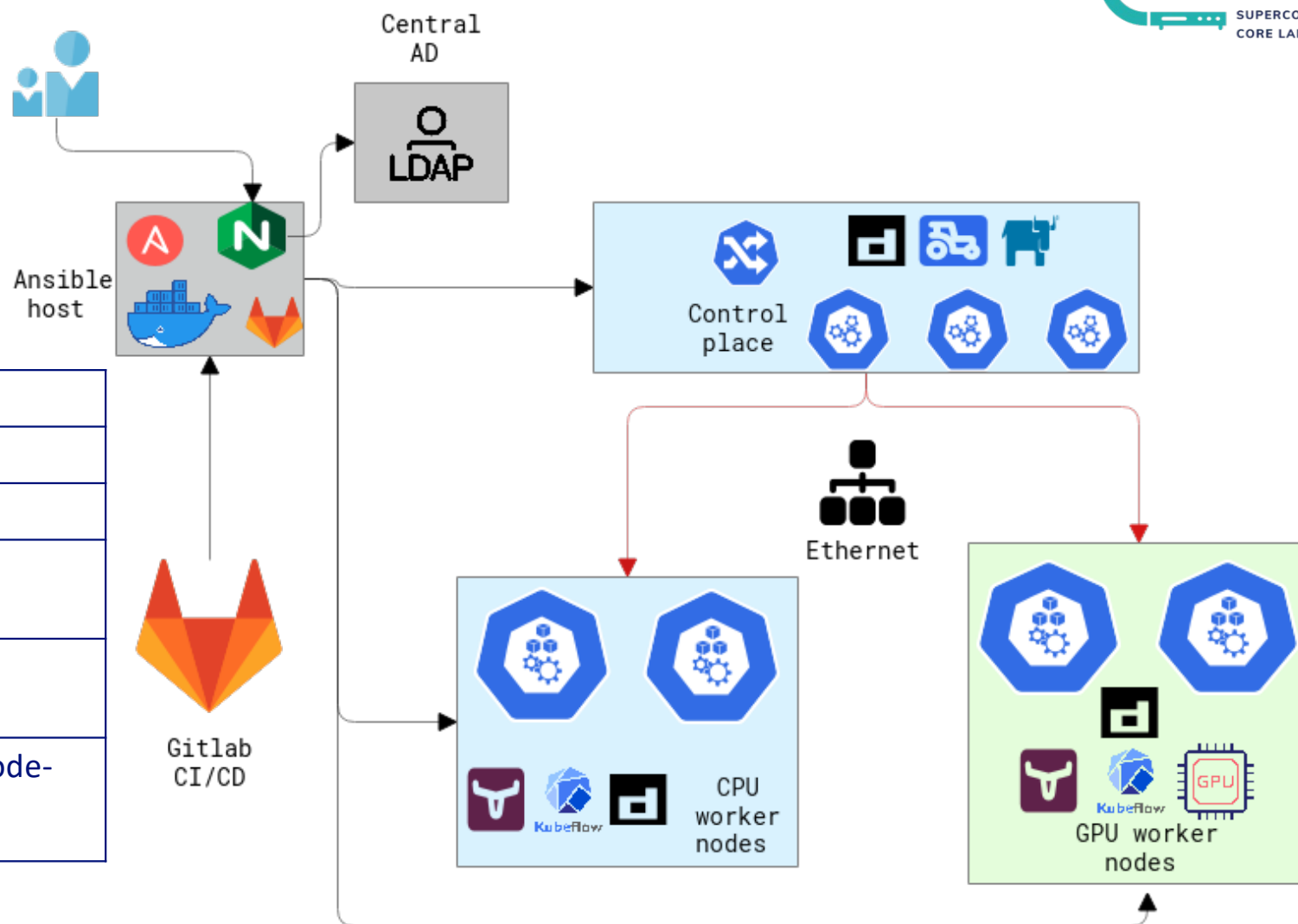
# Cloud cluster



Provision	Terraform
Manage	Ansible
OS	CentOS 8
K8s Distro	k3s
Container Runtime	containerd
Storage class	NFS

# On-Prem cluster

Provision	None
Manage	Ansible
OS	Rocky Linux 9.2
K8s Distro	RKE2
Container Runtime	containerd
Storage class	Longhorn on node-local storage



# Lessons Learnt

- Master in HA
- K3s vs RKE2 (mysql vs etcd)
- DiskPressure – CrashLoopBackOff
- Multi-node cluster is important for PoC
- NVIDIA GPU operator vs Device plugin
- Node feature discovery add-on
- Appropriate Container Network Interface (Ethernet + IB + .....)

# Lessons Learnt

- Identity management in a container
  - Per user image vs injecting credentials via Configmaps
- The `kubectl` CLI use in Jupyter session – restricted to namespace
- Customize base images form Kubeflow
  - <https://www.kubeflow.org/docs/components/notebooks/container-images/>
- The Kubeflow UI is useful, until there is a problem. `kubectl` access is needed to investigate what went wrong. That's a user support ticket for HPC centers.
- The `kubectl describe` output and `kubectl logs` are not always helpful
- For high throughput workflows (with SPMD), `restartPolicy=OnFailure` provides resiliency



# Thank you

Questions and contact:  
[mohsin.shaikh@kaust.edu.sa](mailto:mohsin.shaikh@kaust.edu.sa)