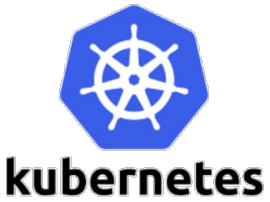


Kubernetes for Scientists

Examples drawn from AI

Fifth National Research Platform (5NRP) Workshop

March 19, 2024

Presented by Mahidhar Tatineni and Dmitry Mishin
University of California San Diego – San Diego Supercomputer Center

Ref: Tutorial developed based on updates on prior presentations at PEARC and SC conferences by Igor Sfiligoi, Dima Mishin and Mahidhar Tatineni

Overview of Tutorial

- Kubernetes Basics
  - Background on containers, orchestration of containers
  - Driving Kubernetes with kubectl
  - Basic examples with YAML description
  - Hands On
- AI Examples with Hands On
  - Training
  - Inference
  - Retrieval Augmented Generation (RAG)

Overview of Tutorial

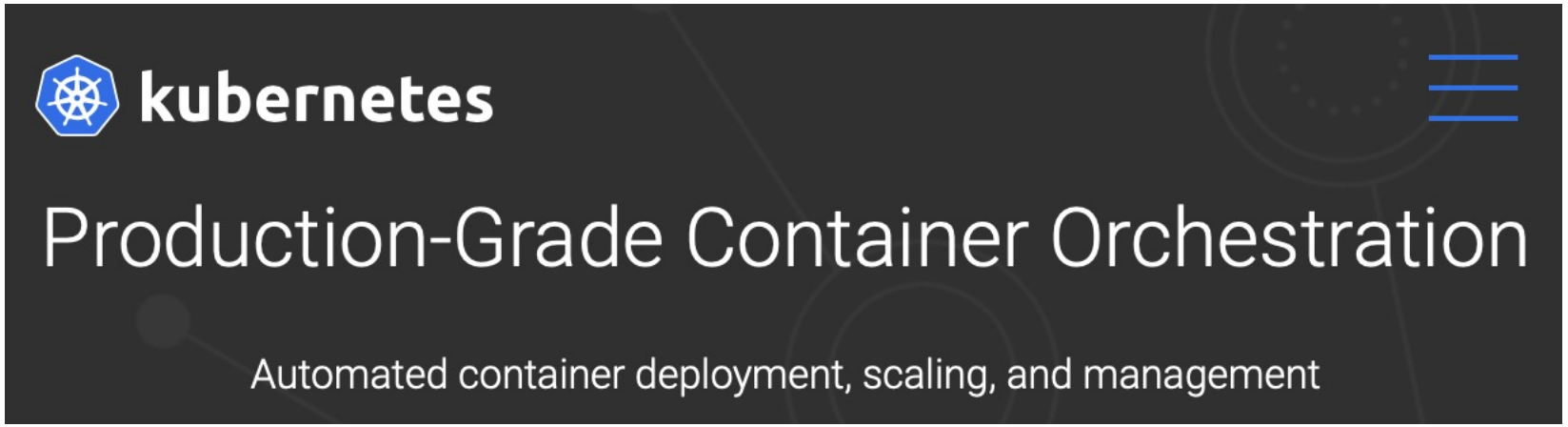
- Kubernetes Basics
  - Background on containers, orchestration of containers
  - Driving Kubernetes with kubectl
  - Basic examples with YAML description
  - Hands On
- AI Examples with Hands On
  - Training
  - Inference
  - Retrieval Augmented Generation (RAG)

A containerized world

Containers are becoming the norm	• Although many runtimes exist
Helps with code portability	• Also more efficient than VMs
Just remember containers are stateless	• If state needed, must be held outside

Container Orchestration

- Once you have many containers on many nodes, you need something to manage the whole
  - This is usually referred to as **Orchestration**



Attribution: https://kubernetes.io

Kubernetes or K8S

Originally created by Google

- Now maintained by Cloud Native Computing Foundation https://kubernetes.io

Open source

- With very large and active development community

Can be deployed anywhere

- Available in HPC centers (e.g. at SDSC)
- Also at all major Clouds (GCP, AWS, Azure)

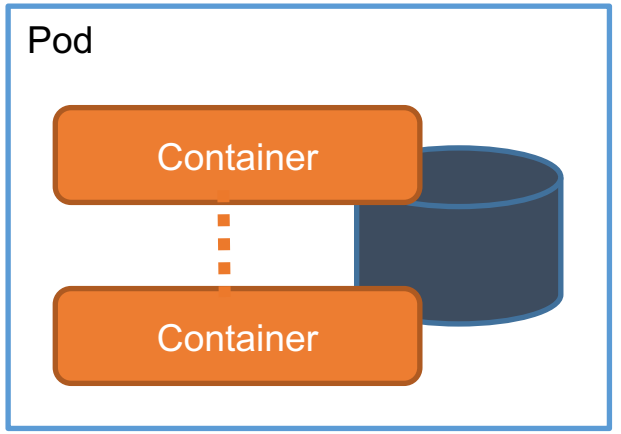
Packing containers into pods

The smallest concept in K8S is actually the Pod

https://kubernetes.io/docs/concepts/workloads/pods/pod/

A Pod is a set of containers

- Having a single Container in a Pod OK



Containers within a Pod are guaranteed to run alongside

- And can share a local storage area

Container image

Each container must pick a container image to use

- Each container can pick its own (typically, no defaults)
- You can mix and match in multi-container pods

Images are externally hosted

- By default, they are loaded from DockerHub
- But you can provide an arbitrary URL, too

Pod scheduling

Kubernetes comes with a reasonable scheduler

Will match Pods to available resources

- Nearly instantons, if free compute resources available
- Else, pod will wait in line until some other pod terminates

Packing Pods into batch Jobs

A Job will make sure the pod completes (container exits with 0 exit code)

- Can retry the job up to N times

Handles pod and container failures

- e.g. if node goes offline, the job will restart elsewhere (up to backoff limit)

Facilitates parallel execution

- Including making sure all iterations succeed

Great for scientific compute

https://kubernetes.io/docs/concepts/workloads/controllers/job/

Driving Kubernetes

No submit nodes

Cloud-native philosophy

- Any device can be used to control K8S
- Typically, your **laptop** is all you need

No shared storage areas

- You can submit from one device and monitor it through another
- **No local state on your laptop**
- **Requires explicit data movement**

kubectl most used tool

- A simple static binary
  - Available for all major platforms (Linux, MacOS, Windows)
  - Detailed download instructions (**use curl**) at https://kubernetes.io/docs/tasks/tools/install-kubectl/
- Just install it on your laptop
  - Can be used over WiFi/WAN
  - Uses a cluster-specific config file in $KUBECONFIG
    - On Linux and MacOS, if not set, defaults to ~/.kube/config
    - On Windows, it defaults to %USERPROFILE%\.kube\config

https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/

Interacting with Kubernetes

kubectl most used options

- `kubectl create -f <filename>` - Create new object (e.g. a job)
- `kubectl get <type> -n <namespace>` - Query existing objects
- `kubectl edit <type> -n <namespace> <id>` - Edit existing object
- `kubectl delete -f <filename>` - Delete existing object
- `kubectl apply -f <filename>` - Create or update an object

https://kubernetes.io/docs/reference/kubectl/

YAML Everywhere

Most interactions with Kubernetes will involve YAML documents

- Both for creating/configuring Pods and Jobs
- And for querying their (detailed) status

YAML is quite easy to use

- Describes itself as "a human friendly markup language"
- Uses Python-like indentation to indicate nesting

https://en.wikipedia.org/wiki/YAML

A simple pod YAML

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod-123
spec:
  containers:
  - name: mypod
    image: ubuntu:22.04
    resources:
      limits:
        memory: 100Mi
        cpu: 100m
      requests:
        memory: 100Mi
        cpu: 100m
    command: ["sh", "-c", "sleep 7200"]
```

A simple pod YAML


```
apiVersion: v1
kind: Pod
metadata:
  name: mypod-123
spec:
  containers:
  - name: mypod
    image: ubuntu:22.04
    resources:
      limits:
        memory: 100Mi
        cpu: 100m
      requests:
        memory: 100Mi
        cpu: 100m
    command: ["sh", "-c", ”sleep 7200"]
```

A simple pod YAML

Container image to use

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod-123
spec:
  containers:
  - name: mypod
    image: ubuntu:22.04
    resources:
      limits:
        memory: 100Mi
        cpu: 100m
      requests:
        memory: 100Mi
        cpu: 100m
    command: ["sh", "-c", "sleep 7200"]
```

A simple pod YAML


```
apiVersion: v1
kind: Pod
metadata:
  name: mypod-123
spec:
  containers:
  - name: mypod
    image: ubuntu:22.04
    resources:
      limits:
        memory: 100Mi
        cpu: 100m
      requests:
        memory: 100Mi
        cpu: 100m
    command: ["sh", "-c", "sleep 7200"]
```

Ready to submit your first container

After you have the YAML, it is trivial

- `vim mypod-123.yaml`
- `kubectl create -f mypod-123.yaml` # Create the pod

More than just pod launching

List your pods

- Another kubectl command
  `kubectl get pods`
  - Using the
    `–o wide`
    option provides good balance between detail and readability

Interactive access to pods

List your requests

Check progress

Log into running pods

- Useful both for true interactive pods as well as for debugging `kubectl exec`
- By default just runs a command, but can be made interactive with `-it -- /bin/bash`

Putting the info so far together

The lifetime of the simple interactive pod

- `vim mypod-123.yaml`
- `kubectl create -f mypod-123.yaml` # Create the pod
- `kubectl get pods –o wide` # Check if the pod is running yet
- `kubectl exec –it mypod-123 -- /bin/bash` # Log into the node
- `kubectl delete –f mypod-123.yaml` # Delete the pod

Fetching the output

Stdout and stderr can be accessed at any time

- `kubectl logs <pod name>`

If you used persistent storage, it can stay there

Or you can explicitly copy it out

- Pick you favorite (non-K8S) tool (e.g. S3, Globus, scp)

Hands On!

Follow steps in:

https://github.com/mahidhar/5nrp_k8s_tutorial/blob/main/Basic_hands_on.md

Overview of Tutorial

- Kubernetes Basics
  - Background on containers, orchestration of containers
  - Driving Kubernetes with kubectl
  - Basic examples with YAML description
  - Hands On
- AI Examples with Hands On
  - Training
  - Inference
  - Retrieval Augmented Generation (RAG)

AI Training Example with PyTorch

```
containers:
- name: mypod
  image: nvcr.io/nvidia/pytorch:23.07-py3
  resources:
    limits:
      memory: 32Gi
      cpu: 4
      nvidia.com/gpu: 1
    requests:
      memory: 32Gi
      cpu: 4
      nvidia.com/gpu: 1
  command: ["/bin/bash", "-c"]
  args:
  - >-
      cd /scratch;
      wget https://raw.githubusercontent.com/pytorch/examples/main/mnist/main.py;
      nvidia-smi;
      python3 main.py;
  volumeMounts:
          - name: scratch
            mountPath: /scratch
```

Using PyTorch container from NVIDIA. Choose one that is compatible with our driver

Downloading the MNIST example python code from github.

Run the python code

AI Training Example with PyTorch

```
|=========================================+======================+======================|
|   0  NVIDIA A10                     Off| 00000000:81:00.0 Off |                    0 |
|  0%   29C    P8              15W / 150W|      0MiB / 23028MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
:|  No running processes found                                                           |
+-----------------------------------------------------------------------------------------+
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-images-idx3-ubyte.gz to ../data/MNIST/raw/train-images-idx3-ubyte.gz
^M  0%|          | 0/9912422 [00:00<?, ?it/s]^M 36%|███▌      | 3538944/9912422 [00:00<00:00, 35362915.52it/s]^M100%|██████████| 9912422/
9912422 [00:00<00:00, 78095812.00it/s]
Extracting ../data/MNIST/raw/train-images-idx3-ubyte.gz to ../data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/train-labels-idx1-ubyte.gz to ../data/MNIST/raw/train-labels-idx1-ubyte.gz
^M  0%|          | 0/28881 [00:00<?, ?it/s]^M100%|██████████| 28881/28881 [00:00<00:00, 198258091.36it/s]
Extracting ../data/MNIST/raw/train-labels-idx1-ubyte.gz to ../data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-images-idx3-ubyte.gz to ../data/MNIST/raw/t10k-images-idx3-ubyte.gz
^M  0%|          | 0/1648877 [00:00<?, ?it/s]^M100%|██████████| 1648877/1648877 [00:00<00:00, 17900446.99it/s]
Extracting ../data/MNIST/raw/t10k-images-idx3-ubyte.gz to ../data/MNIST/raw

Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz
Downloading http://yann.lecun.com/exdb/mnist/t10k-labels-idx1-ubyte.gz to ../data/MNIST/raw/t10k-labels-idx1-ubyte.gz
^M  0%|          | 0/4542 [00:00<?, ?it/s]^M100%|██████████| 4542/4542 [00:00<00:00, 46127188.30it/s]
Extracting ../data/MNIST/raw/t10k-labels-idx1-ubyte.gz to ../data/MNIST/raw

Train Epoch: 1 [0/60000 (0%)]   Loss: 2.283596
Train Epoch: 1 [640/60000 (1%)] Loss: 1.259783
Train Epoch: 1 [1280/60000 (2%)]        Loss: 0.920904
```

Interactive Jobs: Jupyterhub

- https://jupyterhub-west.nrp-nautilus.io
- Template setup to use CILogon
- Choice of prebuilt images for environment - can choose PyTorch or TensorFlow images for example
- Choose resources: number of cores, gpus, and amount of memory

Interactive Jobs: Jupyterhub

Server Options

/home/jovyan is persistent volume, 5GB by default. Make sure you don't fill it up - jupyter won't start next time. You can ask admins to increase the size.
The storage is created in West ceph pool by default. You can ask admins to move it to a different region.

Available resources page

Contact admins in Matrix.

Region

Any

GPUs

0

Cores

1

RAM, GB

8

GPU type

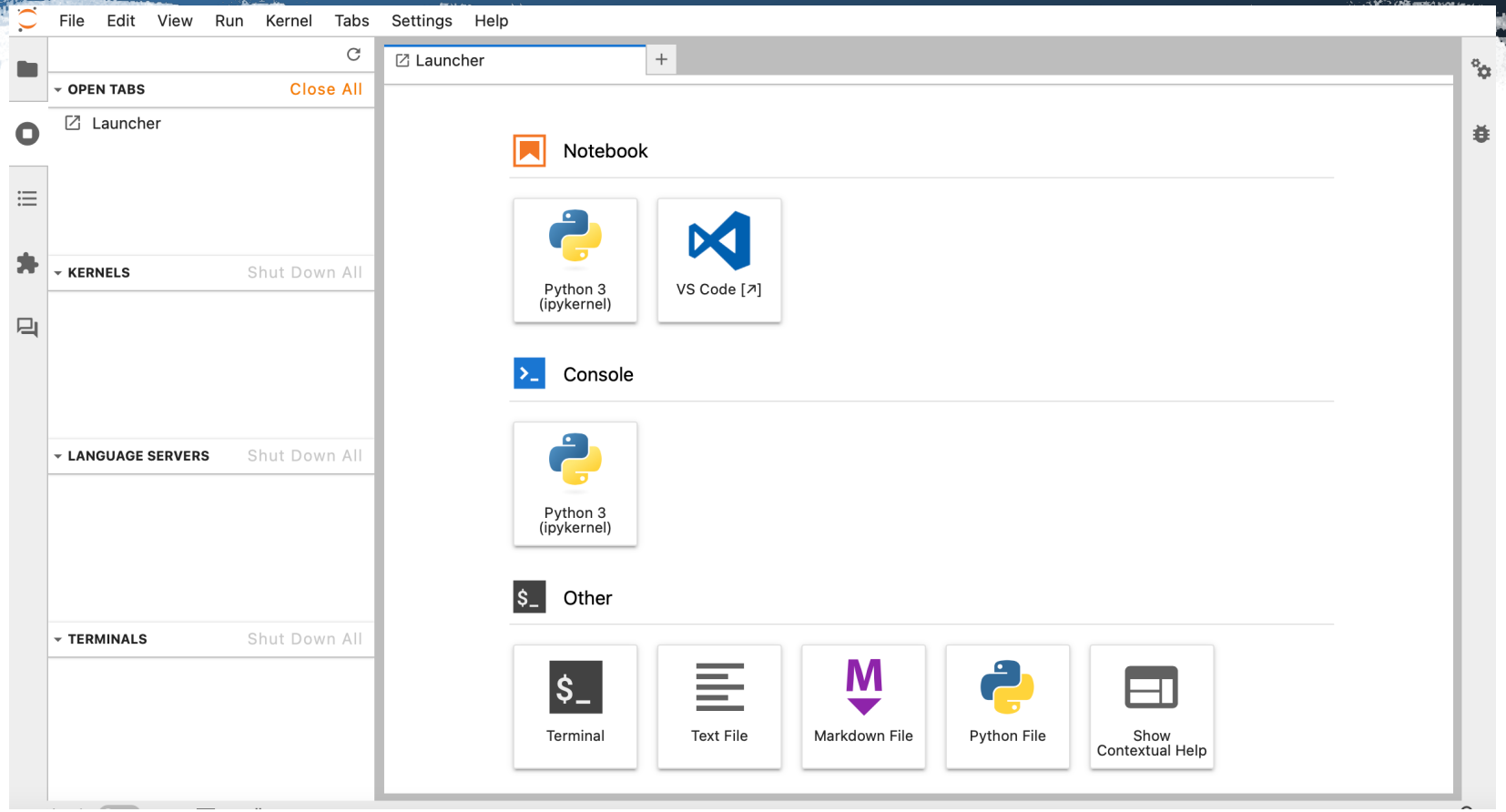
Any

/dev/shm for pytorch

Image

- Stack Minimal
- Stack Scipy
- Stack R
- Stack Julia
- Stack Tensorflow
- Stack Datascience (scipy, Julia, R)
- Stack Pyspark
- Stack All Spark
- NRP Stack Tensorflow + (only default tag, v1.3.1)
- NRP Stack Tensorflow +, other tags
- NRP Desktop GUI
- NRP Desktop GUI + Relion
- NRP Desktop GUI + PRISM
- NRP R Studio Server
- NRP Matlab
- NRP OSGEO

Interactive Jobs: Jupyterhub



Inference Example

```
  - name: mypod
    image: ghcr.io/huggingface/text-generation-inference:1.4
    resources:
      limits:
        memory: 16Gi
        cpu: 4
        nvidia.com/gpu: 1
      requests:
        memory: 16Gi
        cpu: 4
        nvidia.com/gpu: 1
    command: ["/bin/bash", "-c"]
    args:
    - >-
        cd /scratch;
        export HOME=/scratch;
        timeout 1h text-generation-launcher --model-id mistralai/Mistral-7B-Inst
ruct-v0.2;
    volumeMounts:
            - name: scratch
              mountPath: /scratch
            - name: shm
              mountPath: /dev/shm
```

Docker image from huggingface

Launching with timeout so that the pod completes and is not running forever!

Inference Example

- Detailed on Text Generation Interface at:

https://huggingface.co/docs/text-generation-inference/quicktour
https://huggingface.co/docs/text-generation-inference/basic_tutorials/consuming_tgi

- Simple test for today:

```
from huggingface_hub import InferenceClient
client = InferenceClient(model="http://0.0.0.0:80")
for token in client.text_generation("Who made cat videos?", max_new_tokens=24, stream=True): print (token)
```

Reference: https://github.com/huggingface/text-generation-inference?tab=readme-ov-file#run-a-model

Inference Example: Sample Output

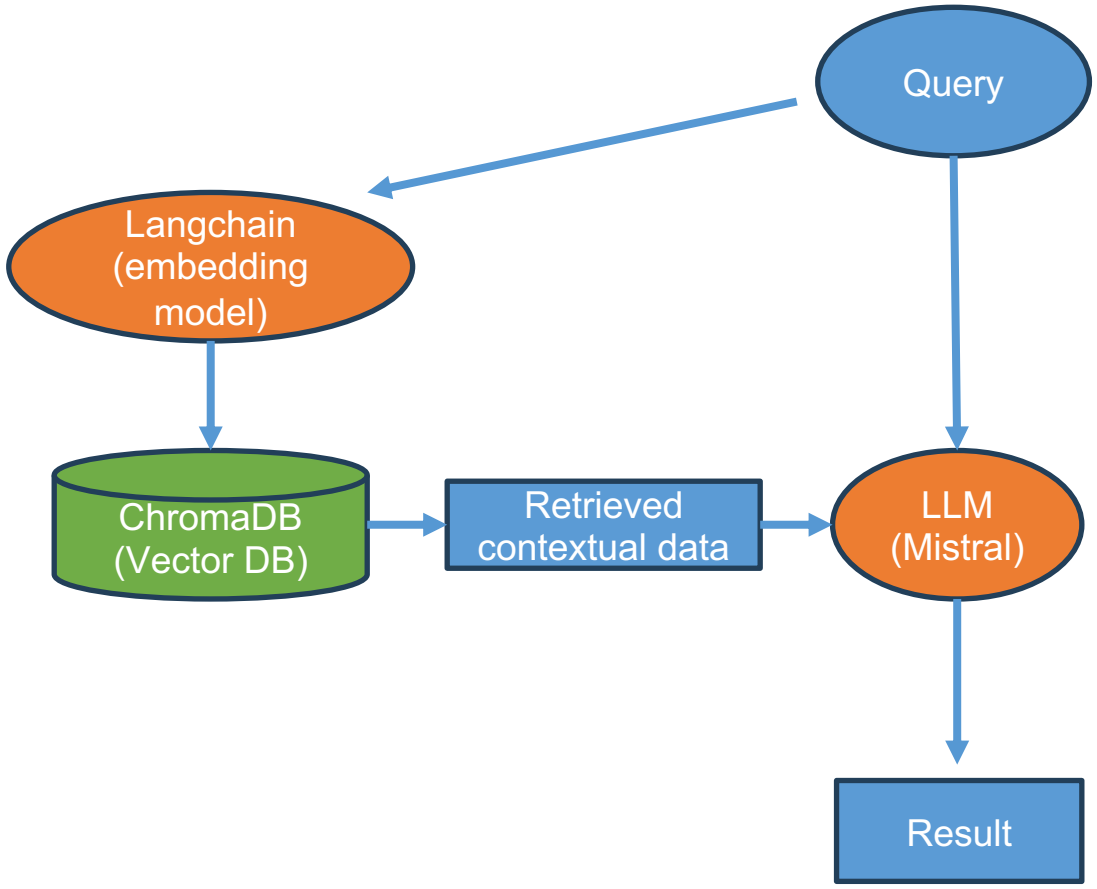
```
Python 3.10.13 | packaged by conda-forge | (main, Dec 23 2023, 15:36:39) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from huggingface_hub import InferenceClient
>>> client = InferenceClient(model="http://0.0.0.0:80")
>>> for token in client.text_generation("Who made cat videos?", max_new_tokens=24, stream=True): print (token
)
...



Cat
 videos
 are
 typically
 made
 by
 individuals
 or
 organizations
 who
 have
 access
 to
 cats
 and
 the
 equipment
 to
 record
 and
 edit
 videos
>>>
```

Retrieval-Augmented Generation (RAG)

- Improve accuracy and reliability of generative AI models by supplementing with contextual facts from additional external sources.
- LLMs are only aware of data from their training set and might give general answers.
- Will likely not have answers for specific queries beyond the initial dataset (for example something internal to an organization).
- RAG incorporates info from external data sources with the trained LLM model to improve the responses.



RAG Example

- Uses Ollama docker image to run language models locally
- Example will use “mistral” model
- We will augment by reading a manual on homing pigeons from the US Army (published 1945, downloaded from Project Gutenberg)
- We use Sentence transformer embeddings from langchain and chromadb to help augment the model
- We then query the RAG setup for info on feeding pigeons!

RAG Example

- Start the pod:
  kubectl apply -f ollama-rag.yaml
- Verify the installs are done and book is downloaded:
  kubectl logs ollama-username
- Start up Ollama server and pull mistral model:
  kubectl exec -it ollama-username -- /bin/bash
  cd /scratch
  nohup ollama serve&
  ollama pull mistral
- Download the test script and run it:
  wget https://raw.githubusercontent.com/mahidhar/5nrp_k8s_tutorial/main/test.py
  python3 -i test.py
- Within interactive python session run:
  rag.invoke("What do you feed pigeons ?")
  rag.invoke("Do tame pigeons have better plumage ?")
  rag.invoke("What affects pigeon plumage ?")

Hands On!

Follow steps in:

https://github.com/mahidhar/5nrp_k8s_tutorial/blob/main/AI-Examples.md

Storage Options

Ephemeral storage

(work area while the pod is running)

Storage inside the container image

- All areas inside the container are writable (typically)
- You can write data straight into the directories provided by the image

Ephemeral partition

- Sometimes you need a larger and faster partition
- Kubernetes allows for an explicit ephemeral mount
- Known as an emptyDir volume

RAM disk

- As with all Linux systems, RAM disk is mounted in all containers
- But (typically) by default limited to 64M
- Must explicitly request a larger one (memory-based emptyDir)

https://kubernetes.io/docs/concepts/storage/volumes/#emptydir

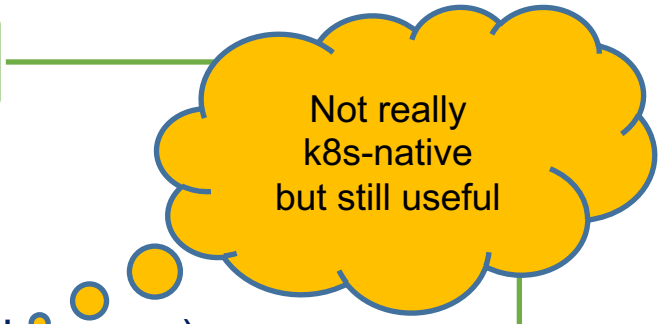
Using external storage

External storage essential for persistency

- Remember, ephemeral storage is gone once the pod is gone
- Most applications will need some persistency

Kubernetes provides several hooks at Pod launch time

- Remote filesystem (e.g. NFS, CEPH)
- Block storage (seen as a block device in the pod)
- Local storage, typically ephemeral but can be persistent
- Direct access to external services (e.g. S3, HTTP/WebDAV, Globus, scp)



https://kubernetes.io/docs/concepts/storage/volumes/
https://kubernetes-csi.github.io/docs/

Mounting storage

Pick the volume to mount

- You may be able to create it at Pod creation type
- But most persistent storage pre-created as Persistent Volume Claims (PVC)

Mount it inside the container

- Any directory path will work
- Whatever works for you

Example PVC creation yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: vol-mahidhar
spec:
  storageClassName: rook-cephfs
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

Example PVC mount yaml

```
apiVersion: batch/v1
kind: Job
metadata:
  name: s3-mahidhar
spec:
  completionMode: Indexed
  completions: 10
  parallelism: 10
  ttlSecondsAfterFinished: 1800
  template:
    spec:
      restartPolicy: OnFailure
      containers:
      - name: mypod
        image: rockylinux:8
        resources:
           limits:
             memory: 100Mi
             cpu: 0.1
           requests:
             memory: 100Mi
             cpu: 0.1
        command: ["sh", "-c", "let s=2*$JOB_COMPLETION_INDEX; d=`date +%s`; date; sleep $s; (echo Job
$JOB_COMPLETION_INDEX; ls -l /mnt/mylogs/)  > /mnt/mylogs/log.$d.$JOB_COMPLETION_INDEX; sleep 1000"]
        volumeMounts:
        - name: mydata
          mountPath: /mnt/mylogs
      volumes:
      - name: mydata
        persistentVolumeClaim:
          claimName: vol-mahidhar
```

Hands On!

Follow steps in:

https://github.com/mahidhar/5nrp_k8s_tutorial/blob/main/Storage.md

Storage on Nautilus cluster:

https://docs.nationalresearchplatform.org/userdocs/storage/intro/

Acknowledgements

This work was partially funded by
US National Science Foundation (NSF) awards
OAC-2112167, OAC-1826967, OAC-1541349,
OAC-2030508 and CNS-1730158.