MLOps Pipeline with Kubeflow & MLflow in Minikube

Introduction

In this project we are setting up miikube in local machine and creating a cluster in minikube. MLFlow and Kubeflow we are setting up in the same minikube cluster.

We are creating a ML model to do the predictions and give us necessary metrics.

Kubeflow is used to setup that pipeline and tracking of output we are doing in MLFlow.

Prerequisites

Few software installation we have to do first:

1. Python. (Link)
2. Install pip (Link)
3. Docker Desktop. (Link)
4. Minikube. (Link)

Setup

1. After installing Docker Desktop: Go to Settings -> Kubernetes -> Check *'Enable Kubernetes'* and *'Show System Containers'*

2. After installing Minikube: Open Windows Poershell in Administrator mode and run this command:

```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri
'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
```

3. Install the Kubeflow Pipeline SDK:

```
pip install kfp
```

4. Pull the MLFLow image from docker:

```
docker pull ghcr.io/mlflow/mlflow
```

5. Installing MLFlow and Kubeflow

Here we have two options to install MLFlow and Kubeflow in minikube cluster.

OPTION 1: (Install them individually)

- **MLFLOW:**
  - Download my .yaml file that i have put in github (Link)

  - Run these commands in Windows Powershell:

```
minikube start
kubectl create namespace mlflow
kubectl create -f mlflow.yaml

```

- **KUBEFLOW**:
  - Open the Windows Powershell and run these commands

```
minikube start

set PIPELINE_VERSION=1.8.5

kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/cluster-scoped-resources?ref=$PIPELINE_VERSION"

kubectl wait --for condition=established --timeout=60s crd/applications.app.k8s.io
kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/env/platform-agnostic-pns?ref=$PIPELINE_VERSION"
```

- It will take 30-40 mins to complete, you can check the container getting created here:
  minikube kubectl -- get pods -A

- ***NOTE:*** *if you see Error or CrashLoopBackOff dont worry it will get fixed automatically.*

OPTION 2: (Install them in a one go)

- Download my .yaml file that i have put in github (Link)

- Run these commands in Windows Powershell:

```
minikube start

kubectl create namespace mlflow

kubectl create -f Kubeflow_and_MLFlow.yaml
```

6. Installing Addons in minikube:

   You can add some important addons in minikube cluster by running these commands after starting minikube in Windows Powershell:

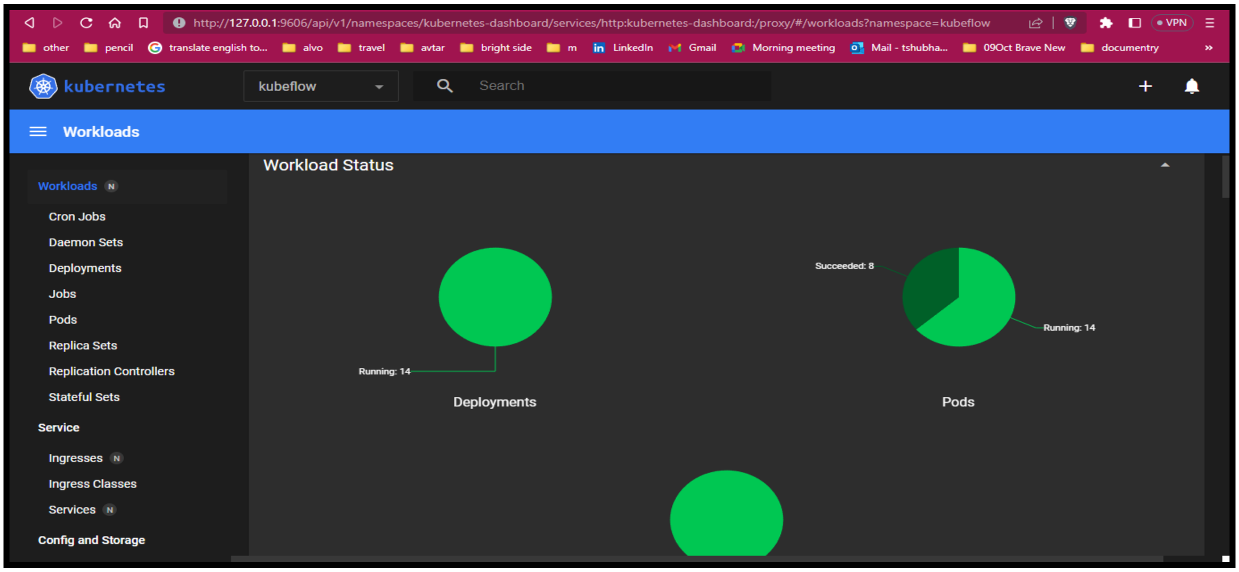
```
minikube addons enable metrics-server

minikube addons enable ingress
```

7. Now all the configs and installation is done, we can check the dashboards by running these commands:

- Minikube Dashboard:

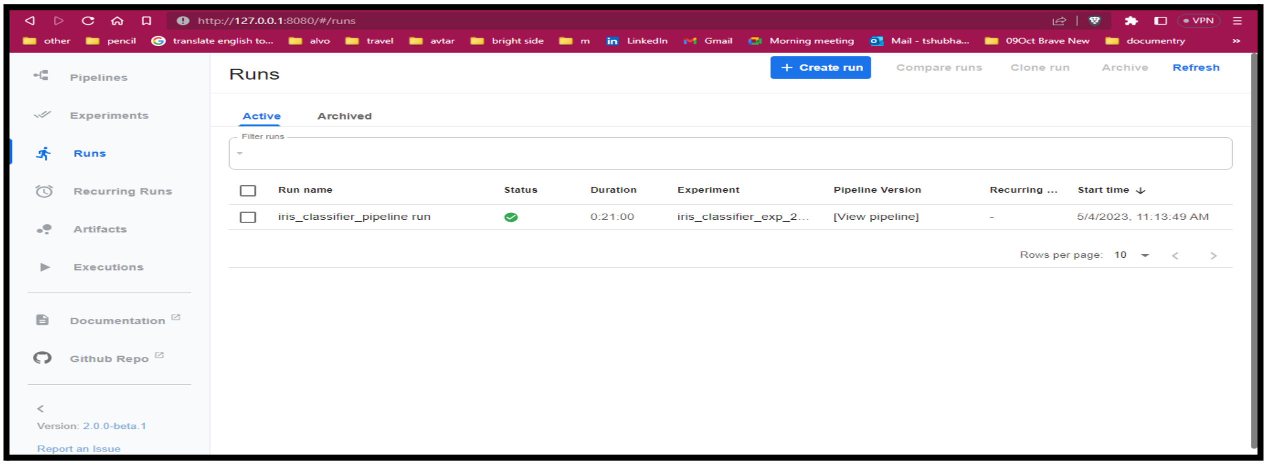
```
minikube dashboard
```



- Kubeflow Dashboard:

```
minikube tunnel
kubectl port-forward -n kubeflow svc/ml-pipeline-ui 8080:80
```

Now go to http://127.0.0.1:8080/



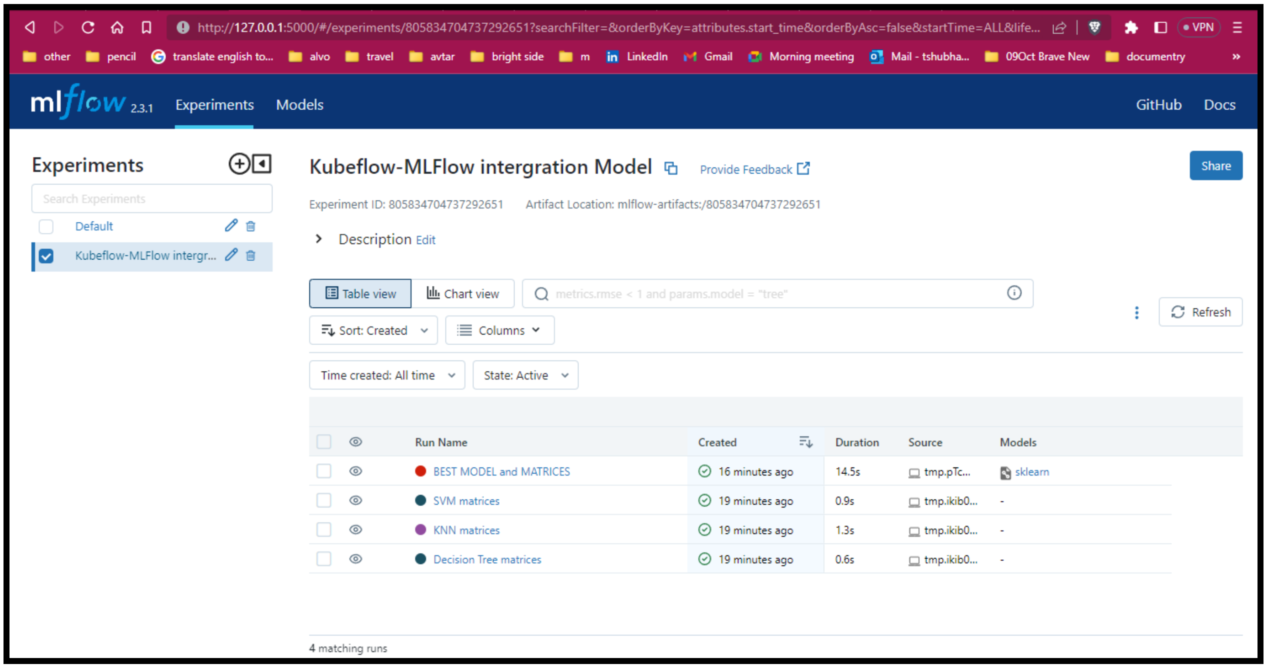
□	Run name	Status	Duration	Experiment	Pipeline Version	Recurring ...	Start time ↓
□	iris_classifier_pipeline run	✓	0:21:00	iris_classifier_exp_2...	[View pipeline]	-	5/4/2023, 11:13:49 AM

Rows per page: 10

- MLFlow Dashboard:

```
minikube tunnel
kubectl port-forward -n mlflow service/mlflow-service 5000:80
```

Now go to http://127.0.0.1:5000/
(if you have already started the tunnel then no need to start again)



http://127.0.0.1:5000/#/experiments/805834704737292651?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&life...

Kubeflow-MLFlow intergration Model

Experiment ID: 805834704737292651 Artifact Location: mlflow-artifacts:/805834704737292651

Description Edit

Run Name	Created	Duration	Source	Models
BEST MODEL and MATRICES	16 minutes ago	14.5s	tmp.pTc...	sklearn
SVM matrices	19 minutes ago	0.9s	tmp.ikib0...	-
KNN matrices	19 minutes ago	1.3s	tmp.ikib0...	-
Decision Tree matrices	19 minutes ago	0.6s	tmp.ikib0...	-

4 matching runs

- **Minikube tunnel:** This command creates a network tunnel between the local machine and the Minikube cluster, allowing the cluster services to be accessible from the local network.

- **Kubectl port-forward:** It forwards traffic from a local port on the machine to a specific port on a Kubernetes service.
  E.g. *kubectl port-forward -n kubeflow svc/ml-pipeline-ui 8080:80* Here it is forwarding traffic from port 8080 on the local machine to port 80 on the ml-pipeline-ui service in the kubeflow namespace.

- By running these commands, wecan access the Kubeflow and MLFlow user interface running inside the Minikube cluster through a web browser on our local machine.

Pipeline Creation

Now our configs and installations ins done with minikube, now we can create our pipeline. We have two options to do it.

OPTION 1: (By running the Python code)

- Go to this link, I have given .ipynb and .py file of my code, directly run the code and it will start the pipeline creation that we can check in Kubeflow UI and to track the metrics that also i have handelled in my python code, that you can check in MLFlow UI (Link)

OPTION 2:

- I have provided .yaml in my github, You download it and go to Kubeflow UI -> Pipelines -> Upload pipeline -> upload file and select that .yaml file.

- It will create the pipeline and you can run directly in UI

❖ ***NOTE:*** *In Python code if you see, to make connection between MLFlow and Kubeflow so that metrics we could be able to see in MLFlow that are generating during Kubeflow pipeline, I have used this special IP: http://host.docker.internal:5000/ because minikube is running inside docker container only.*

Tracking Logs and Metrics

- When pipeline is getting created and when it completes 'Metrices' component then you can see in MLFlow UI that metrics are visible there.

- Also best model and its metrix we can see in MLFlow UI

Troubleshooting

Some common issues that i faced:

1. While starting the minikbe if you see this or similar message:

```
X Exiting due to MK_USAGE: Docker Desktop has only 3864MB memory but you specified 4096MB
```

Then run this command to start the minikube:

```
monikube start --memory=3000
```

2. If you run this command you can see all the pods running or not running. Sometimes new container creation takes time so we have to wait

```
minikube kubectl -- get pods -A
```

3. Due to heavy load in system post-forward command get breaks and some container shows ‘CrashLoopBackOff’ status if we run this command

```
minikube kubectl -- get pods -A
```

At that time just wait for sometime, it will get resolved automatically and then re-run post-forward command

4. If your port is already busy then post-forward will not work, so first check and free the port.

Conclusion

By using Kubeflow and MLflow in Minikube, you can create a scalable and flexible pipeline for managing your machine learning workflows. Kubeflow simplifies the process of creating and managing machine learning pipelines, while MLflow makes it easier to track and manage experiments

So if you follow these steps, you will be able to setup everything easily. This is my github repo that has all the necessary files.