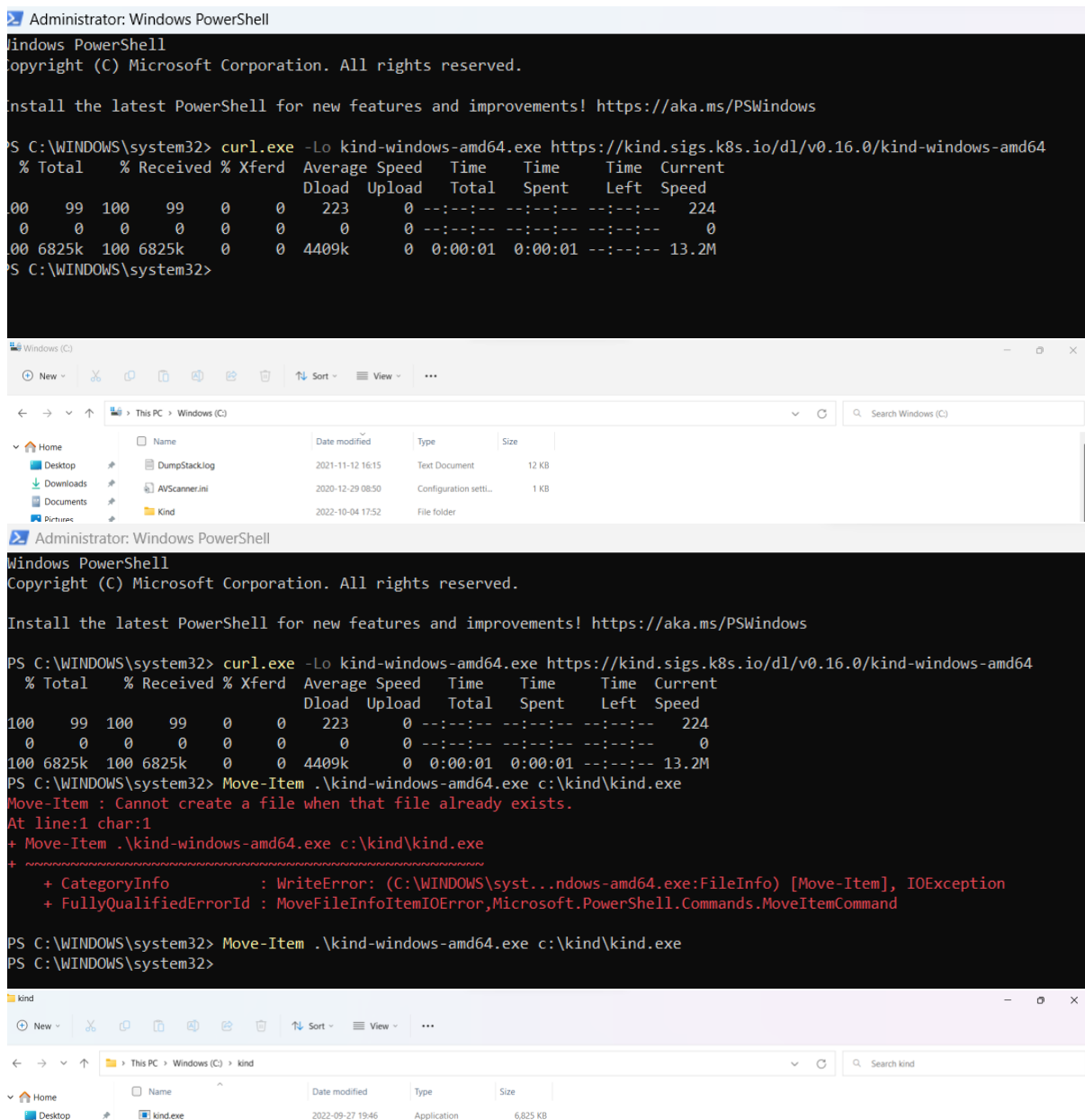


Kubeflow Setup on Local Host

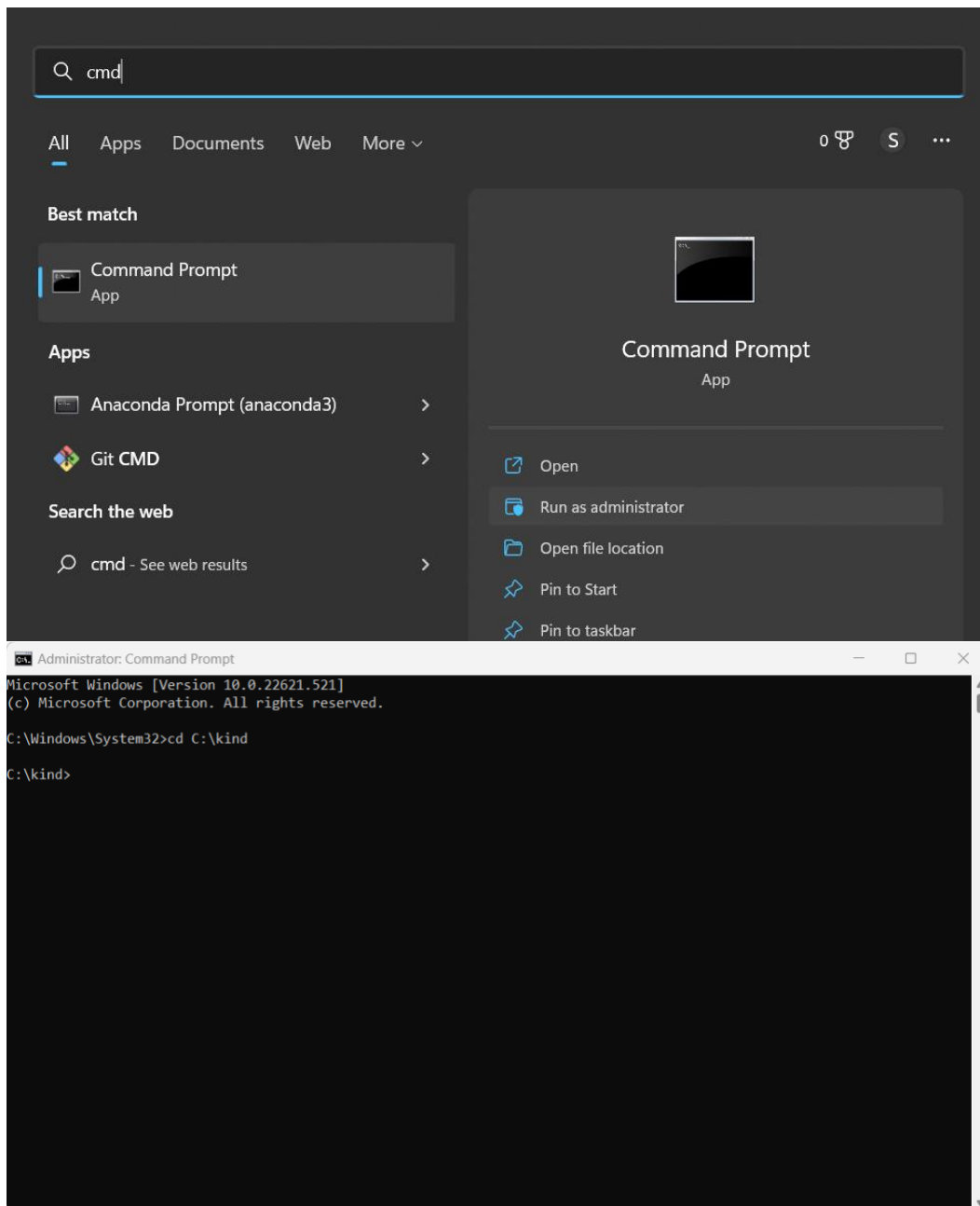
- Install kind on command prompt and follow the below mentioned commands (can also be seen in the screenshots given below):
 - o `curl.exe -Lo kind-windows-amd64.exe https://kind.sigs.k8s.io/dl/v0.16.0/kind-windows-amd64`
 - o `curl.exe -Lo kind-windows-amd64.exe https://kind.sigs.k8s.io/dl/v0.16.0/kind-windows-amd64`
 - o `Move-Item .\kind-windows-amd64.exe c:\kind\kind.exe`



- Install Docker Desktop on Windows and follow the link given:
<https://docs.docker.com/desktop/install/windows-install/>



- Open the command prompt



- Change your directory to where you have installed the kind (kind's directory) and create clusters.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.521]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>cd C:\kind

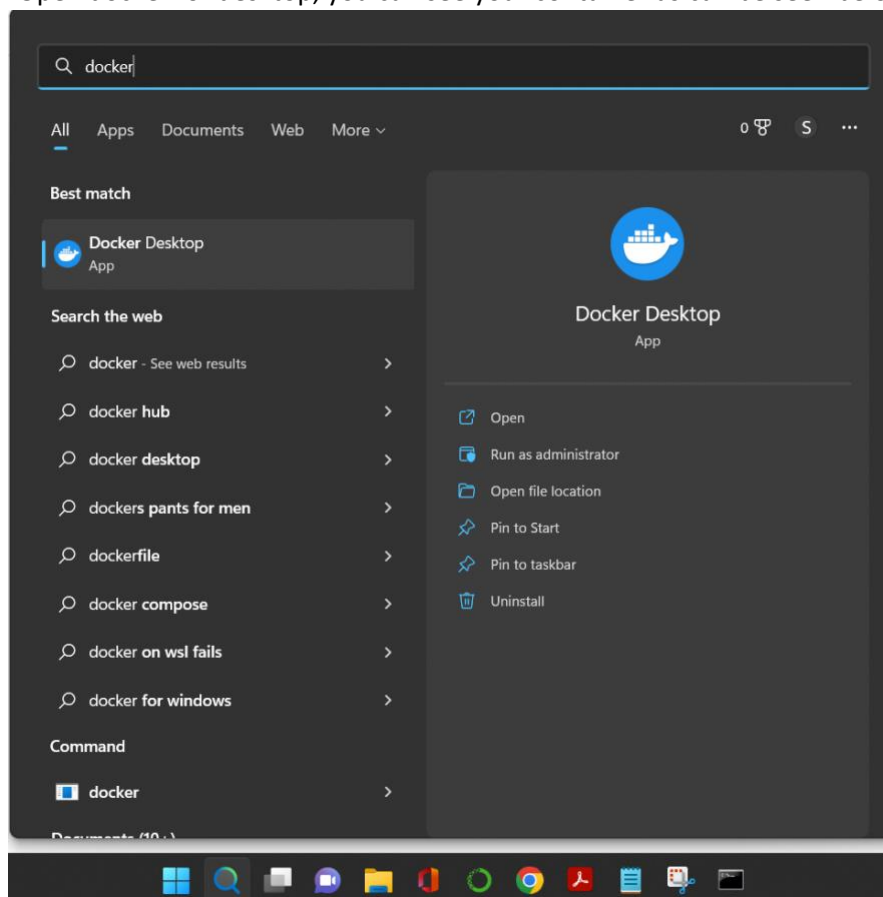
C:\kind>kind create cluster
Creating cluster "kind" ...
  • Ensuring node image (kindest/node:v1.25.2) ...
  ✓ Ensuring node image (kindest/node:v1.25.2) ...
  • Preparing nodes ...
  ✓ Preparing nodes ...
  • Writing configuration ...
  ✓ Writing configuration ...
  • Starting control-plane ...
  ✓ Starting control-plane ...
  • Installing CNI ...
  ✓ Installing CNI ...
  • Installing StorageClass ...
  ✓ Installing StorageClass ...
Set kubectl context to "kind-kind"
You can now use your cluster with:

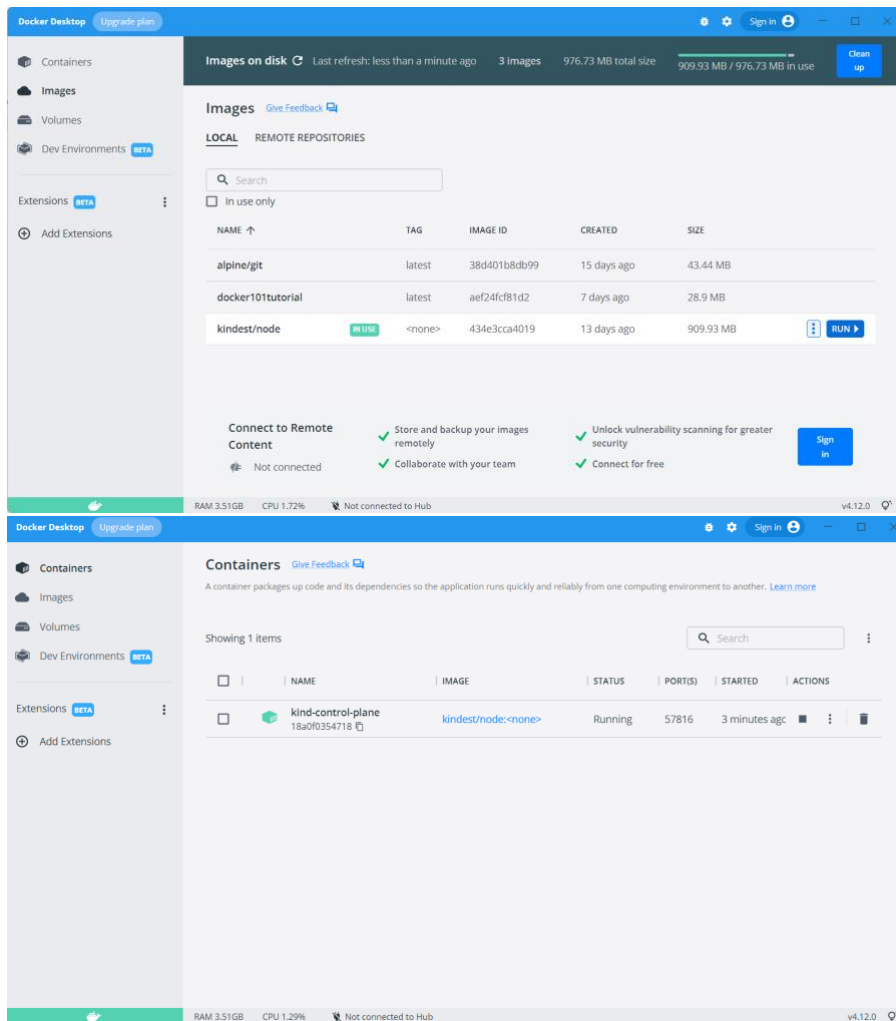
kubectl cluster-info --context kind-kind

Thanks for using kind! 😊

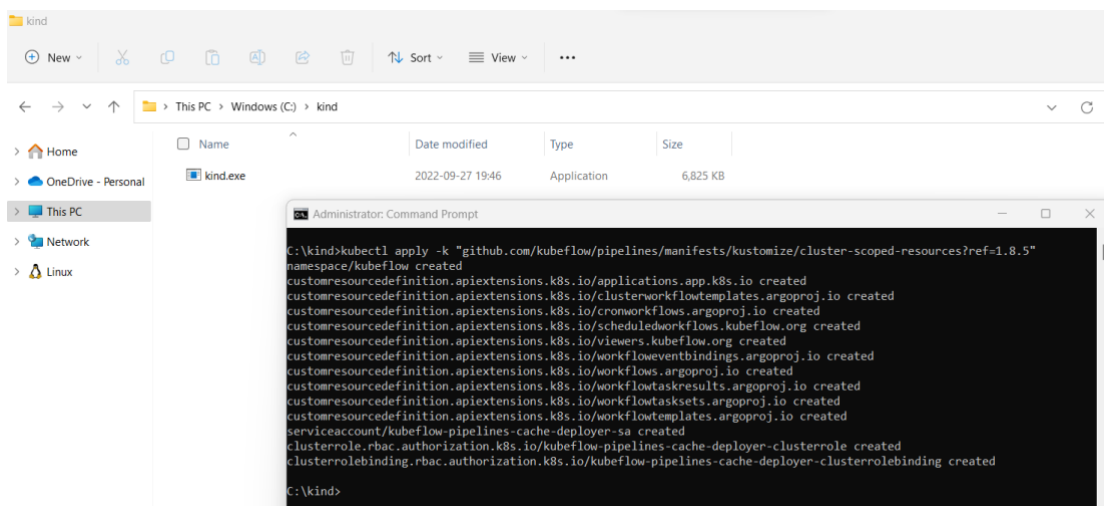
C:\kind>
```

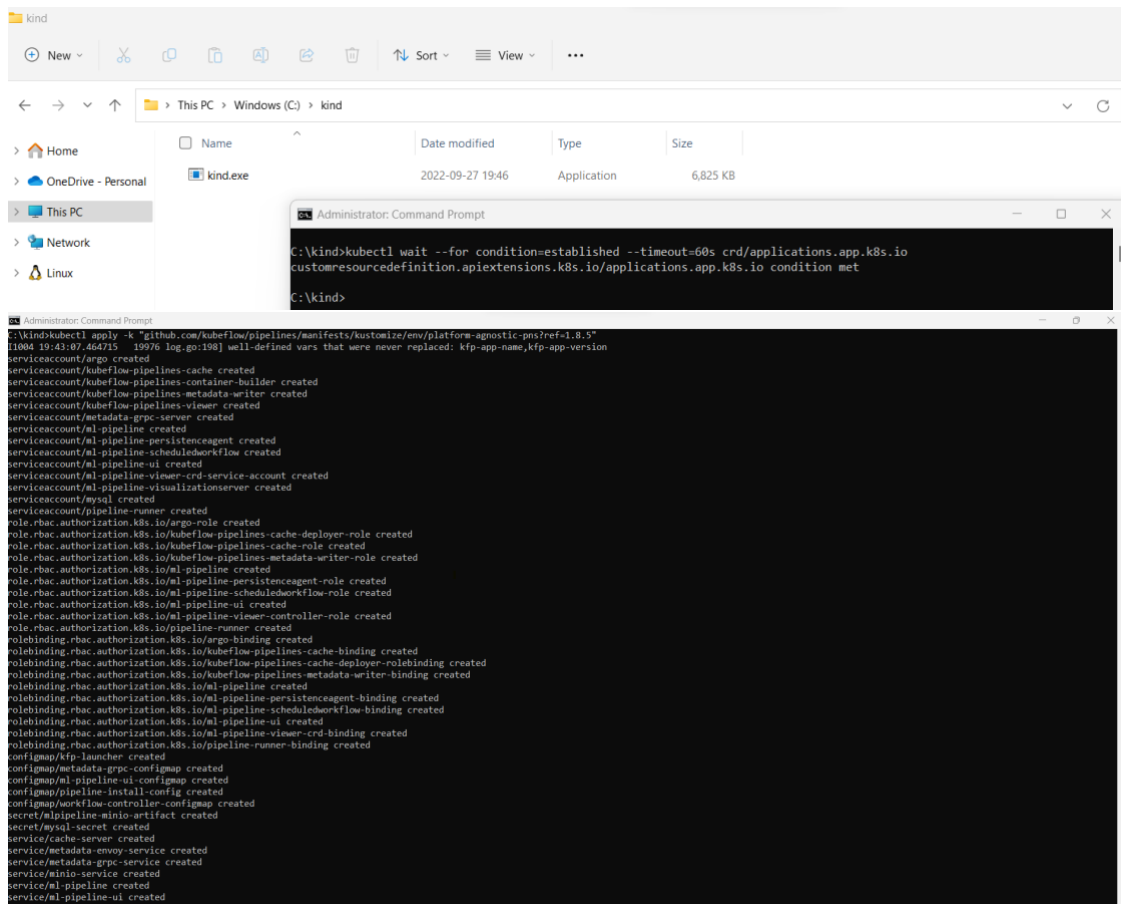
-Open docker for desktop, you can see your container as can be seen below:



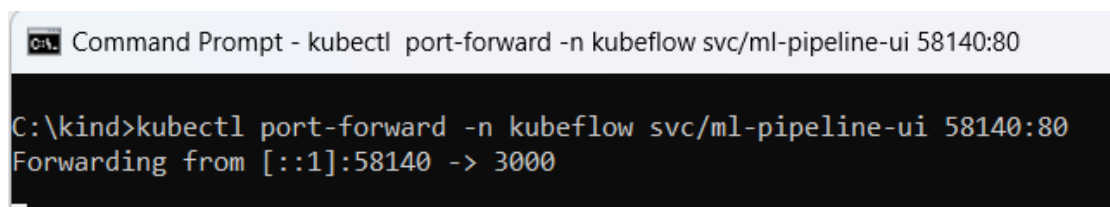


- Execute the following commands to deploy the Kubeflow Pipelines as illustrated below:
 - o `kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/cluster-scoped-resources?ref=1.8.5"`
 - o `kubectl wait --for condition=established --timeout=60s crd/applications.app.k8s.io`
 - o `kubectl apply -k "github.com/kubeflow/pipelines/manifests/kustomize/env/platform-agnostic-pns?ref=1.8.5"`

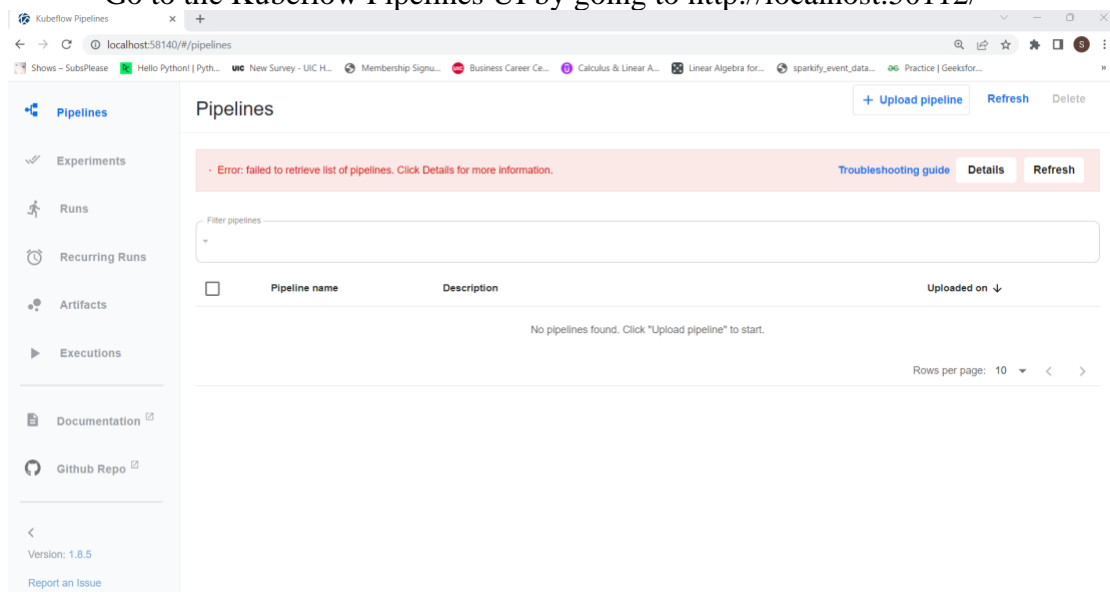




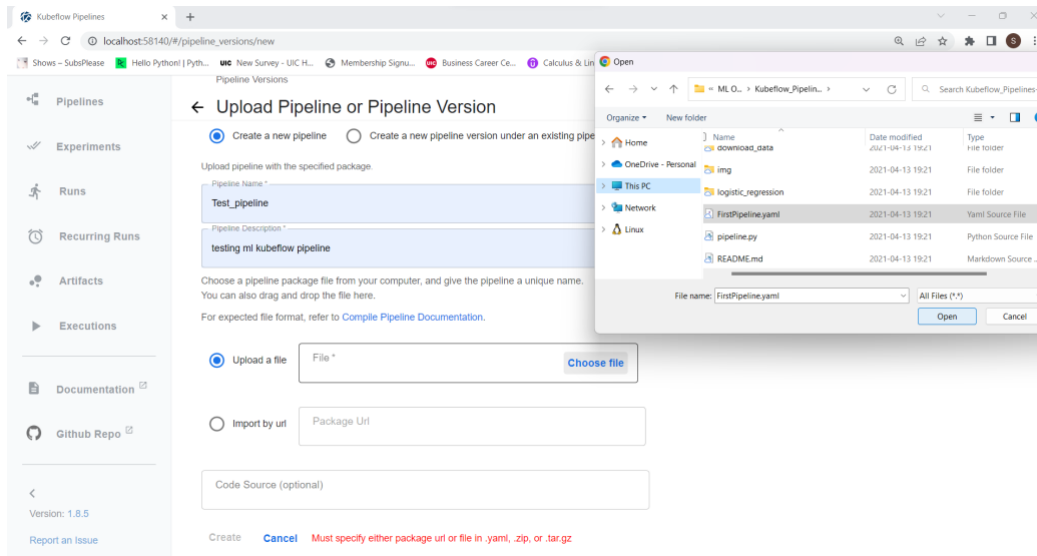
- By port-forwarding, confirm that the Kubeflow Pipelines UI is reachable:
 - o `kubectl port-forward -n kubeflow svc/ml-pipeline-ui 50112:80`



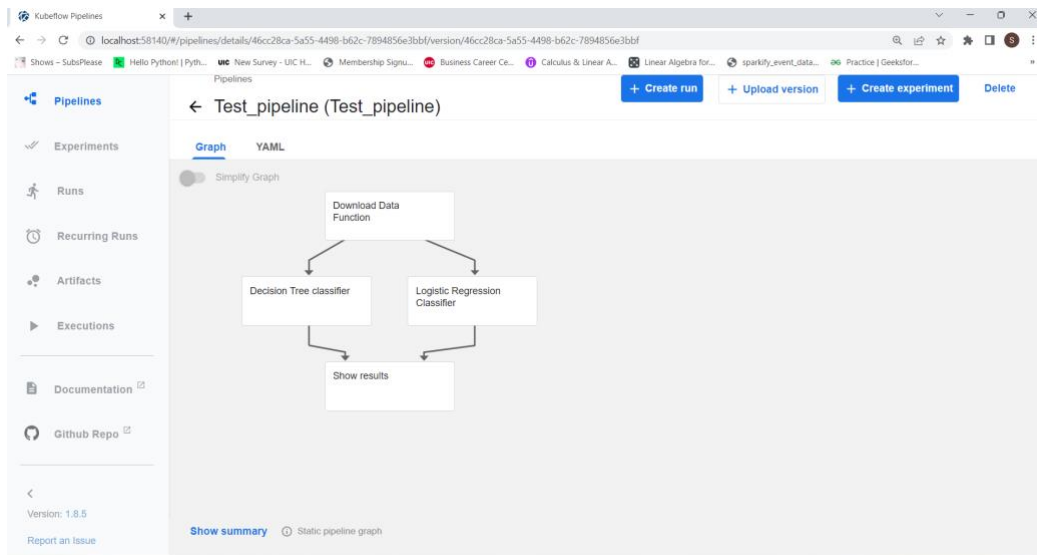
- Go to the Kubeflow Pipelines UI by going to <http://localhost:50112/>



- Creating a new pipeline taking .yaml file:



- - Testing the pipeline



The screenshot shows the 'Runs' page in the Kubeflow Pipelines UI. The 'Active' tab is selected, showing a table of runs. The table has columns: Run name, Status, Duration, Experiment, Pipeline Version, Recurring, and Start time. One run is listed: 'Test_Run' with status 'Running' and start time '10/4/2022, 8:48:04 PM'.

Run name	Status	Duration	Experiment	Pipeline Version	Recurring	Start time
Test_Run	Running	-	Default	Test_pipeline	-	10/4/2022, 8:48:04 PM

- Test Run (Finding the Accuracy)

The screenshot displays the Kubeflow Pipelines interface for a pipeline named **Test_Run**. The pipeline graph shows the following steps:

- Download Data Function** (Completed)
- Decision Tree classifier** (Completed)
- Logistic Regression Classifier** (Completed)
- Show results** (Currently running)

The **Show results** step is highlighted, and its output is displayed in a detailed view on the right. The output includes input parameters, input artifacts, output parameters, output artifacts, and main logs.

Input parameters:

Parameter Name	Value
decision-tree-classifier-Accuracy	0.8859649122807017
logistic-regression-classifier-Accuracy	0.9385964912280702

Input artifacts:

Output parameters:

Output artifacts:

main-logs:

```
minio://mlpipeline/artifacts/first-pipeline-bwpgj-2967029873/022/10/05/first-pipeline-bwpgj-2967029873/main.log
Decision tree (accuracy): 0.8859649122807017
Logistic regression (accuracy): 0.9385964912280702
```