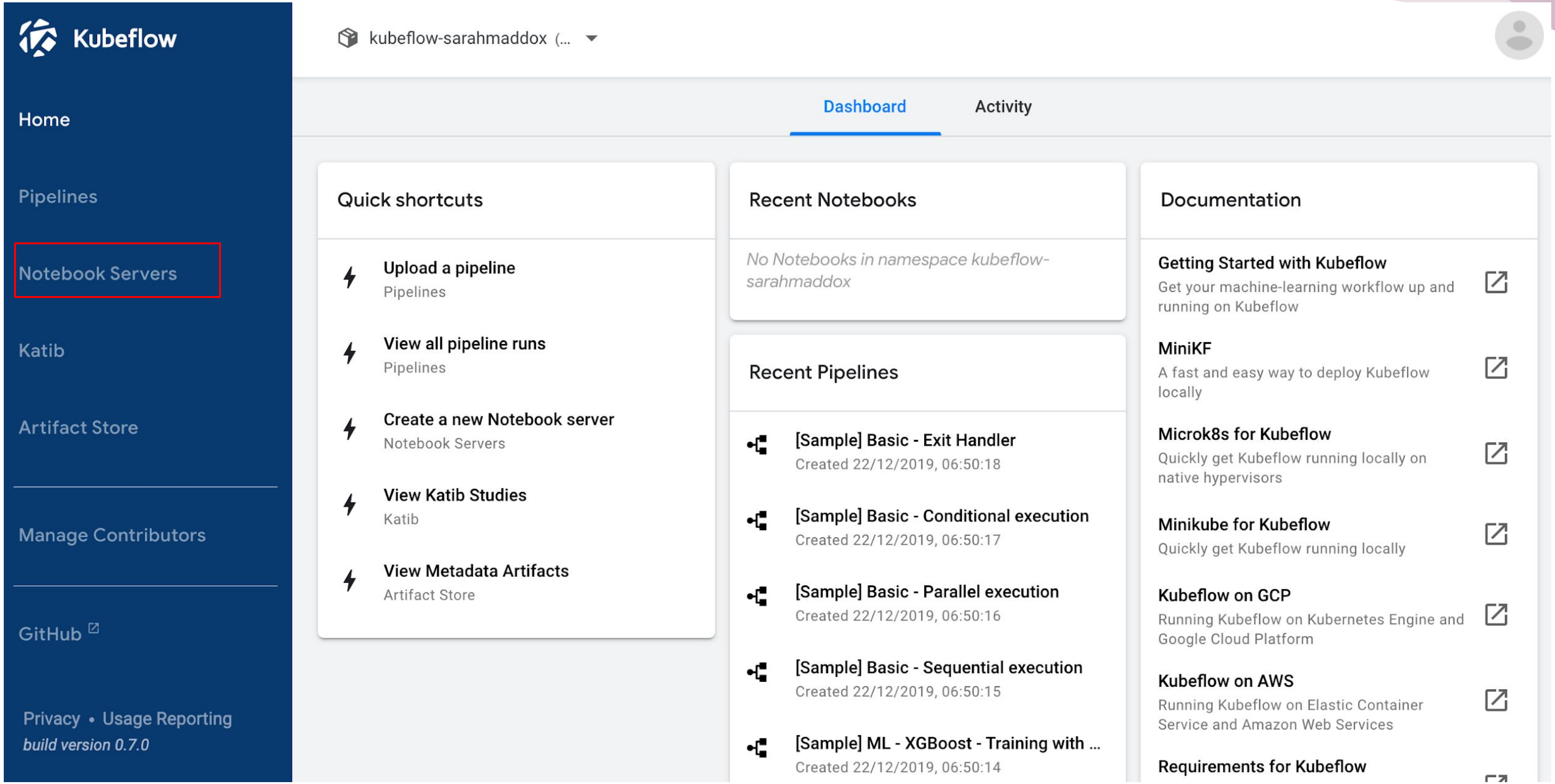
Jupyter Notebook on Microk8s Kubeflow

Jupyter Notebooks

- Appropriate open-source environment for code building, visualizations.
- Integrated as a core Kubeflow Component managed by the Notebook Server Controller.
- Allows standard/custom notebook images , role-based access control (RBAC), secrets and credentials to manage for teams.
- Kubeflow provides multiple notebook servers per kubeflow deployments, each having a single namespace that corresponds to a team or project name.

Jupyter Notebooks



Kubeflow

kubeflow-sarahmaddox (...

Home

Pipelines

Notebook Servers

Katib

Artifact Store

Manage Contributors

GitHub

Privacy • Usage Reporting

build version 0.7.0

Dashboard | Activity

Quick shortcuts

Upload a pipeline
Pipelines

View all pipeline runs
Pipelines

Create a new Notebook server
Notebook Servers

View Katib Studies
Katib

View Metadata Artifacts
Artifact Store

Recent Notebooks

No Notebooks in namespace kubeflow-sarahmaddox

Recent Pipelines

[Sample] Basic - Exit Handler
Created 22/12/2019, 06:50:18

[Sample] Basic - Conditional execution
Created 22/12/2019, 06:50:17

[Sample] Basic - Parallel execution
Created 22/12/2019, 06:50:16

[Sample] Basic - Sequential execution
Created 22/12/2019, 06:50:15

[Sample] ML - XGBoost - Training with ...
Created 22/12/2019, 06:50:14

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

Minikube for Kubeflow
Quickly get Kubeflow running locally

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

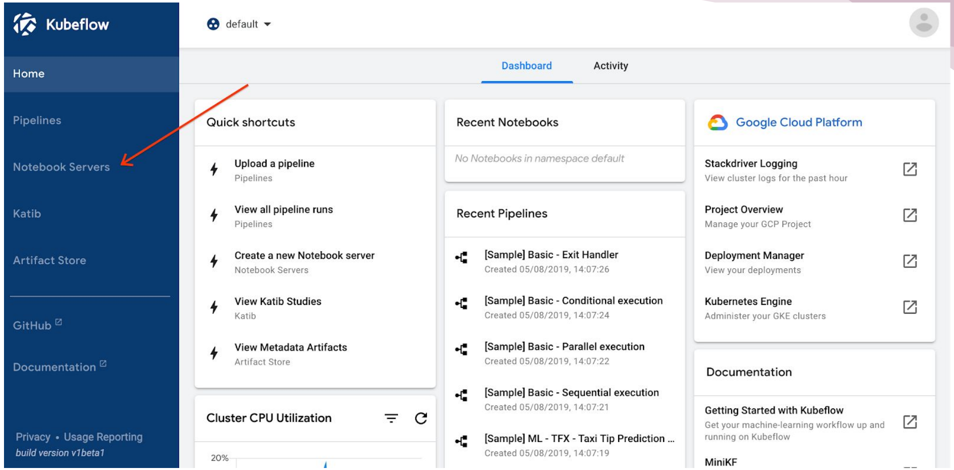
Set up your Notebook

Start by setting up a jupyter notebook through the Notebook Servers tab following the steps below:

1. Click **Notebook Servers** in the left-hand panel of the Kubeflow UI.

2. Click the **namespace** dropdown and choose the a that corresponds to your Kubeflow profile.

3. Click **new server** at the top right corner of the Notebook Servers page to create a notebook server.



Quick shortcuts

- Upload a pipeline — Pipelines
- View all pipeline runs — Pipelines
- Create a new Notebook server — Notebook Servers
- View Katib Studies — Katib
- View Metadata Artifacts — Artifact Store

Recent Notebooks

No Notebooks in namespace default

Recent Pipelines

- [Sample] Basic - Exit Handler — Created 05/08/2019, 14:07:26
- [Sample] Basic - Conditional execution — Created 05/08/2019, 14:07:24
- [Sample] Basic - Parallel execution — Created 05/08/2019, 14:07:22
- [Sample] Basic - Sequential execution — Created 05/08/2019, 14:07:21
- [Sample] ML - TFX - Taxi Tip Prediction ... — Created 05/08/2019, 14:07:19

Google Cloud Platform

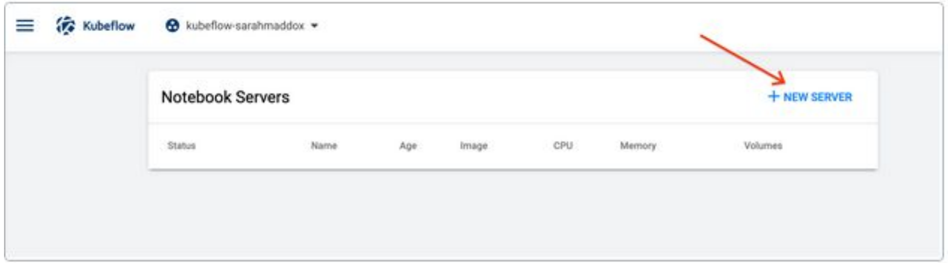
- Stackdriver Logging — View cluster logs for the past hour
- Project Overview — Manage your GCP Project
- Deployment Manager — View your deployments
- Kubernetes Engine — Administer your GKE clusters

Cluster CPU Utilization

20%

Documentation

- Getting Started with Kubeflow — Get your machine-learning workflow up and running on Kubeflow
- MiniKF



Notebook Servers

Status	Name	Age	Image	CPU	Memory	Volumes

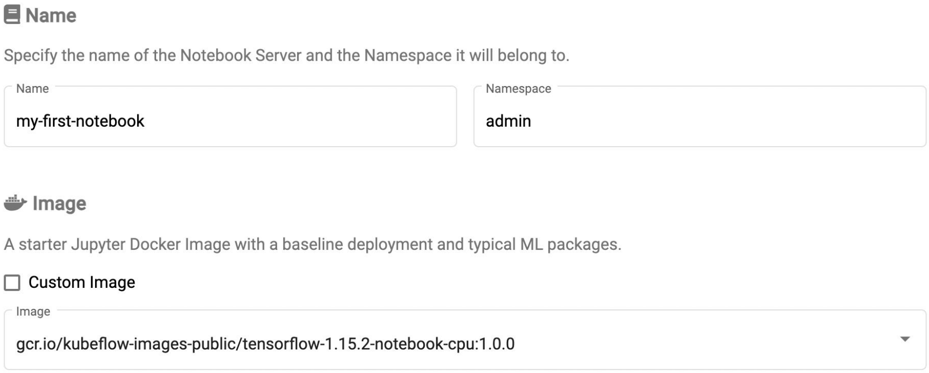
Set up your Notebook

4. Enter the details of your new server on the next page:

   a. Give a **name** of your choice to the notebook server, which must be in ***lowercase***

   b. The **namespace** is automatically updated by Kubeflow.

   c. Select a Docker image. Use the gcr.io/kubeflow-images-public/tensorflow-1.15.2-notebook-cpu:1.0.0 image for our example.



Name

Specify the name of the Notebook Server and the Namespace it will belong to.

Name: my-first-notebook

Namespace: admin

Image

A starter Jupyter Docker Image with a baseline deployment and typical ML packages.

□ Custom Image

Image: gcr.io/kubeflow-images-public/tensorflow-1.15.2-notebook-cpu:1.0.0

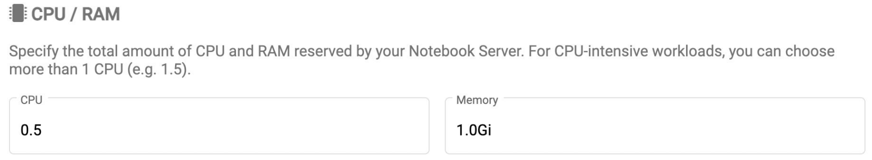
Set up your Notebook

6. Specify the total amount of **CPU** that your notebook server should reserve. **For this lab, 0.5 should be appropriate**

7. Specify the total amount of memory your notebook server should reserve.**For this lab, 1.0Gi should be appropriate**

8. Specify a **workspace volume** to hold your personal workspace for this notebook server. **For this lab, use the default name, size and mode given by kubeflow.**

9. Click **LAUNCH** and you should see a new Notebook server entry.

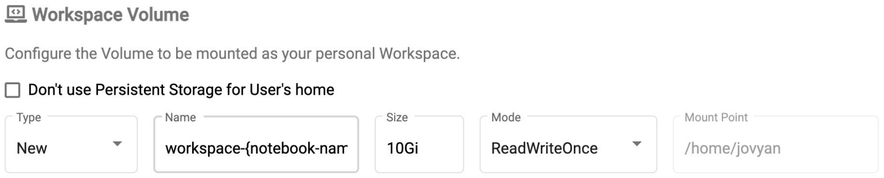


CPU / RAM

Specify the total amount of CPU and RAM reserved by your Notebook Server. For CPU-intensive workloads, you can choose more than 1 CPU (e.g. 1.5).

CPU: 0.5

Memory: 1.0Gi



Workspace Volume

Configure the Volume to be mounted as your personal Workspace.

□ Don't use Persistent Storage for User's home

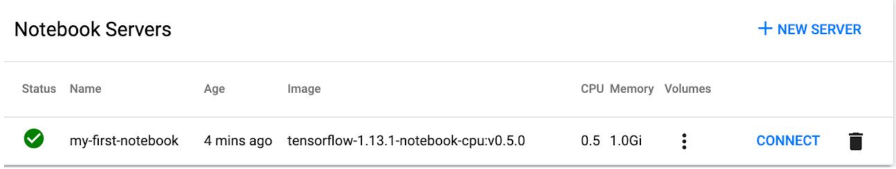
Type: New

Name: workspace-{notebook-nam

Size: 10Gi

Mode: ReadWriteOnce

Mount Point: /home/jovyan



Notebook Servers

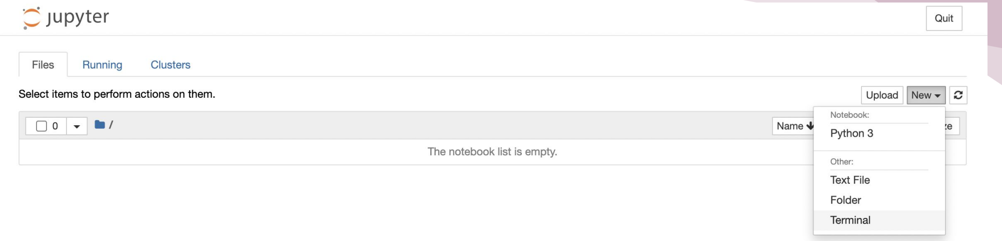
Status	Name	Age	Image	CPU	Memory	Volumes		
✓	my-first-notebook	4 mins ago	tensorflow-1.13.1-notebook-cpu:v0.5.0	0.5	1.0Gi	⋮	CONNECT	🗑

Clone the Repository

10. Connect to your notebook and open up the terminal.

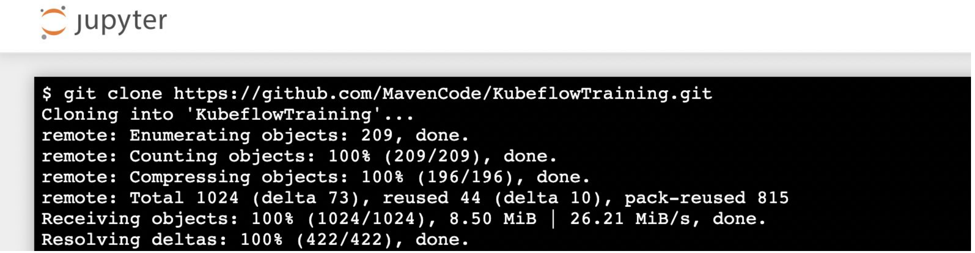
11. In the terminal run this code to clone the repository.

```
git clone
https://github.com/MavenCode/KubeflowTraining.git
```

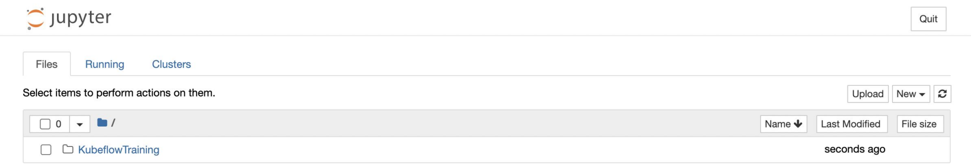


Select items to perform actions on them.

The notebook list is empty.



```
$ git clone https://github.com/MavenCode/KubeflowTraining.git
Cloning into 'KubeflowTraining'...
remote: Enumerating objects: 209, done.
remote: Counting objects: 100% (209/209), done.
remote: Compressing objects: 100% (196/196), done.
remote: Total 1024 (delta 73), reused 44 (delta 10), pack-reused 815
Receiving objects: 100% (1024/1024), 8.50 MiB | 26.21 MiB/s, done.
Resolving deltas: 100% (422/422), done.
```



Select items to perform actions on them.

Name	Last Modified
KubeflowTraining	seconds ago