

Jupyter Notebook

Jupyter Notebooks

- Appropriate open-source environment for code building, visualizations.
- Integrated into Kubeflow by connecting to the Notebook Server.
- 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

The image shows the Kubeflow dashboard interface. On the left is a dark blue sidebar with navigation links: Home, Pipelines, Notebook Servers (highlighted with a red box), Katib, Artifact Store, Manage Contributors, and GitHub. At the bottom of the sidebar are links for Privacy and Usage Reporting, and the build version 0.7.0. The main content area has a top bar with the Kubeflow logo, the namespace 'kubeflow-sarahmaddox', and tabs for Dashboard and Activity. The Dashboard tab is active. It contains three columns of widgets. The first column, 'Quick shortcuts', lists actions like 'Upload a pipeline', 'View all pipeline runs', 'Create a new Notebook server', 'View Katib Studies', and 'View Metadata Artifacts'. The second column, 'Recent Notebooks', shows a message that no notebooks are present in the namespace. The third column, 'Recent Pipelines', lists several sample pipelines with their creation times. A fourth column, 'Documentation', provides links to various guides like 'Getting Started with Kubeflow', 'MiniKF', 'Microk8s for Kubeflow', 'Minikube for Kubeflow', 'Kubeflow on GCP', 'Kubeflow on AWS', and 'Requirements for Kubeflow'.

Kubeflow

Home

Pipelines

Notebook Servers

Katib

Artifact Store

Manage Contributors

GitHub

Privacy • Usage Reporting
build version 0.7.0

kubeflow-sarahmaddox (...)

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

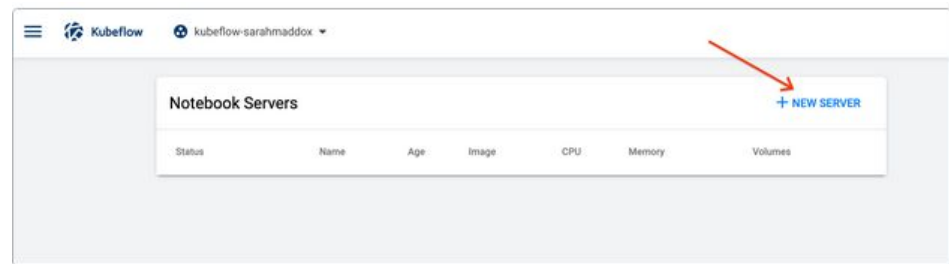
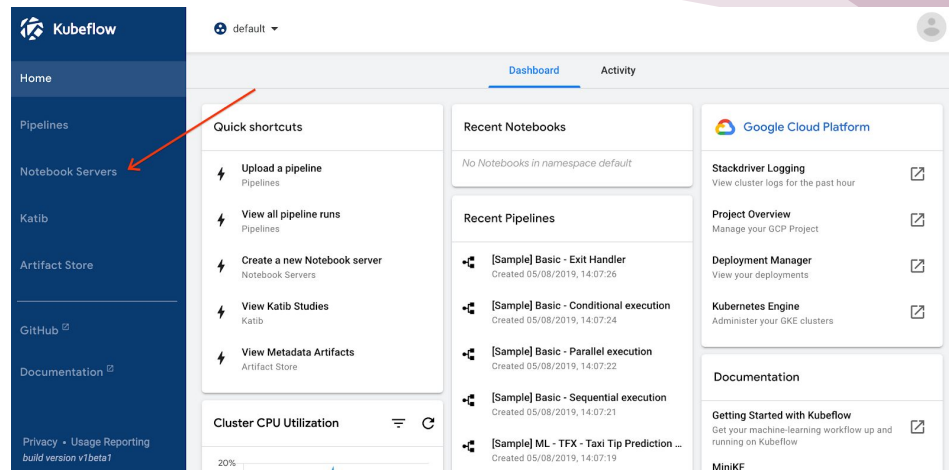
Documentation

- 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 one 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.



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	Namespace
my-first-notebook	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 labs use 0.5**

7. Specify the total amount of memory your notebook server should reserve. **For this labs use 1.0Gi**

8. Specify a **workspace volume** to hold your personal workspace for this notebook server. The name is automatically updated by kubeflow, leave the size as 10Gi and the mode at ReadWriteOnce.

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

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-narr}	Size 10Gi	Mode ReadWriteOnce	Mount Point /home/jovyan
-------------	-----------------------------------	--------------	-----------------------	-----------------------------

Notebook Servers

[+ NEW SERVER](#)

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 the following to clone the repository.

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

