

Kubeflow - Introduction

- [Overview](#)
- [Stages](#)
- [Creating a Notebook](#)
- [Components](#)
 - [KFServing](#)
 - [Jupyter Notebooks](#)
 - [Kubeflow Pipelines](#)
 - [Hyperparameter Tuning with Katib](#)

Overview

The Kubeflow project is dedicated to making deployments of machine learning (ML) workflows on Kubernetes simple, portable and scalable. The goal is not to recreate other services, but to provide a straightforward way to deploy best-of-breed open-source systems for ML to diverse infrastructures. Anywhere you are running Kubernetes, you should be able to run Kubeflow.



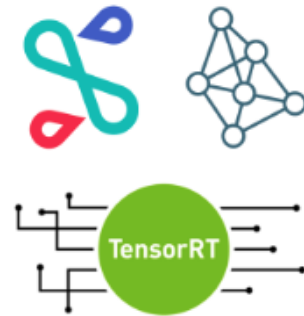
Notebooks

Kubeflow includes services to create and manage interactive [Jupyter notebooks](#). You can customize your notebook deployment and your compute resources to suit your data science needs. Experiment with your workflows locally, then deploy them to a cloud when you're ready.



TensorFlow model training

Kubeflow provides a custom [TensorFlow training job operator](#) that you can use to train your ML model. In particular, Kubeflow's job operator can handle distributed TensorFlow training jobs. Configure the training controller to use CPUs or GPUs and to suit various cluster sizes.



Model serving

Kubeflow supports a [TensorFlow Serving](#) container to export trained TensorFlow models to Kubernetes. Kubeflow is also integrated with [Seldon Core](#), an open source platform for deploying machine learning models on Kubernetes, and [NVIDIA Triton Inference Server](#) for maximized GPU utilization when deploying ML/DL models at scale.



Pipelines

Kubeflow Pipelines is a comprehensive solution for deploying and managing end-to-end ML workflows. Use Kubeflow Pipelines for rapid and reliable experimentation. You can schedule and compare runs, and examine detailed reports on each run.



Multi-framework

Our development plans extend beyond TensorFlow. We're working hard to extend the support of **PyTorch**, **Apache MXNet**, **MPI**, **XGBoost**, **Chainer**, and more. We also integrate with **Istio** and **Ambassador** for ingress, **Nucleo** as a fast multi-purpose serverless framework, and **Pachyderm** for managing your data science pipelines.

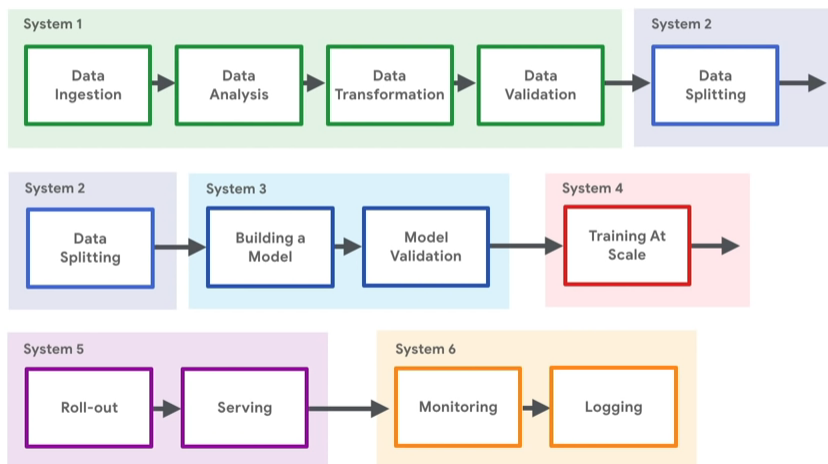


Community

We are an open and welcoming community of software developers, data scientists, and organizations! Join our **Slack workspace** for help with any issues you may face, and read **more about the community**.

Stages

Each ML Stage is an Independent System



Defining ML Workflows including:

- Managing data
- Running notebooks
- Training Models
- Serving Models

Creating a Notebook

From the Notebook menu, create a new notebook server.



Components

Central Dashboard

The central user interface (UI) in Kubeflow

Notebook Servers

Using Jupyter notebooks in Kubeflow

Kubeflow Pipelines

A powerful platform for building end to end ML workflows. Pipelines allow you to build a set of steps to handle everything from collecting data to serving your model.

KFServing

Kubeflow model deployment and serving toolkit

Katib

Katib is a Kubernetes-native project for automated machine learning (AutoML). Katib supports hyperparameter tuning, early stopping and neural architecture search (NAS).

Training Operators

Training of ML models in Kubeflow through operators such as Tensorflow and Pytorch.

Multi-Tenancy

Multi-user isolation and identity access management (IAM)

KFServing

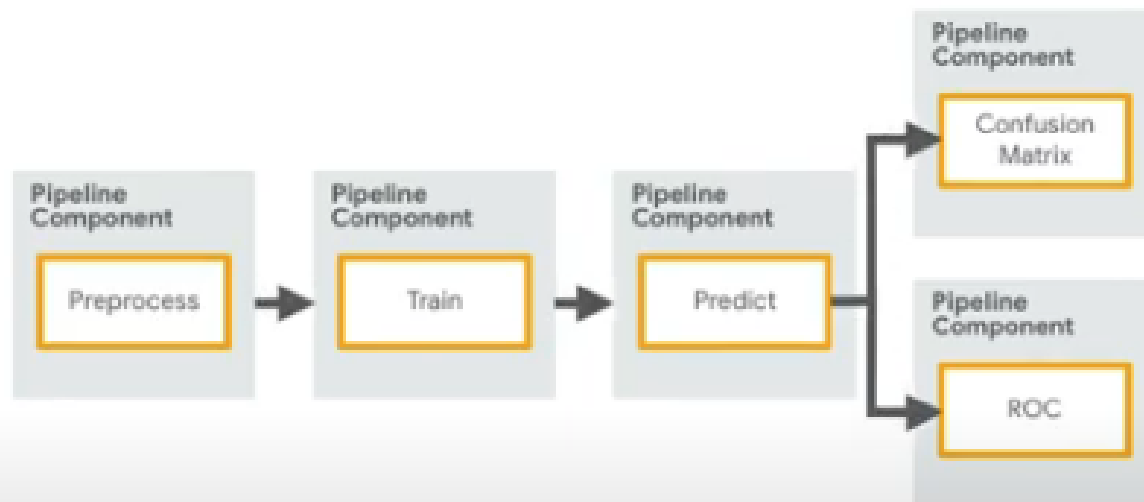
The video player displays a presentation slide titled "Kubeflow Components". The slide features a diagram of the Kubeflow architecture stack. At the top, there are four boxes labeled "Pre-Process", "Predict", "Post-Process", and "Explain". Below these are logos for "TensorFlow", "PyTorch", "XGBoost", and "ONNX". The next layer is "KFServing" with the "Kubeflow" logo. Below that are "Knative", "Istio", and "Kubernetes". The bottom layer is "Compute cluster GPU, TPU, CPU". To the right of the stack is a "Model Assets" database icon. A presenter, a woman with long brown hair wearing a blue t-shirt, is standing in front of the slide. The video player interface includes a progress bar at 3:21 / 3:52 and various control icons.

Jupyter Notebooks

Defacto standard for data scientists for performing rapid data analysis.

Kubeflow Pipelines

Example pipeline



A pipeline component is one step in the workflow that performs one specific task.

Takes inputs and produces outputs.

```
In [14]: import kfp.dsl as dsl

def my_pipeline_step(step_name, param1, param2, ...):
    return dsl.ContainerOp(
        name = step_name,
        image = '<path to my container image>',
        arguments = [
            '--param1', param1,
            '--param2', param2,
            ...
        ],
        file_outputs = {
            'output1': '/output1.txt',
            'output2': '/output2.json',
            ...
        }
    )
```

Hyperparameter Tuning with Katib

Katib is a Kubernetes-native project for automated machine learning (AutoML). Katib supports hyperparameter tuning, early stopping and neural architecture search (NAS).

Reference	URL
Kubeflow 101 Videos	https://www.youtube.com/playlist?list=PLlivdWyY5sqLS4lN75RPDEyBgTro_YX7x