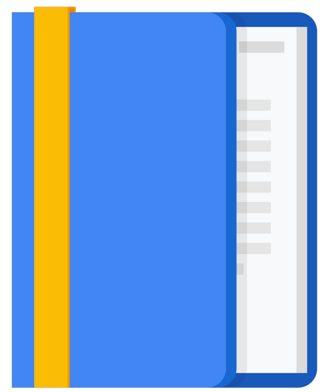
Production ML Pipelines with Kubeflow

Agenda

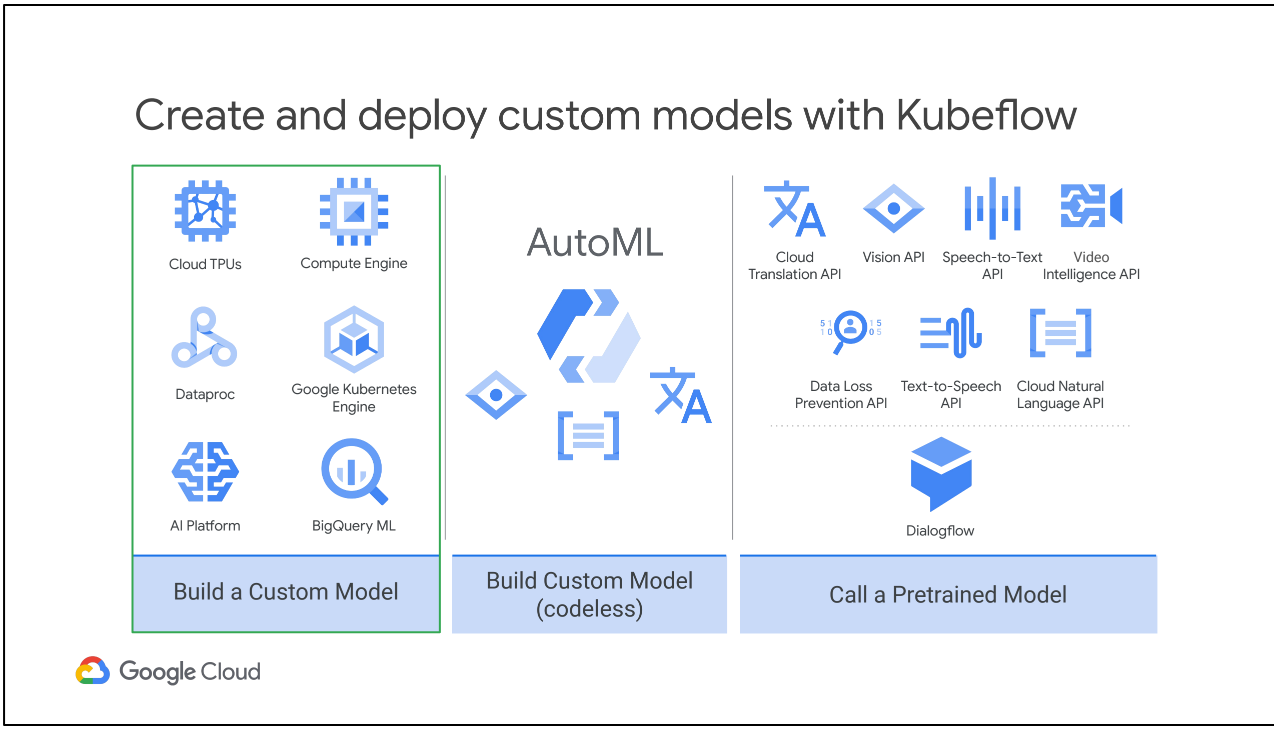
Ways to do ML on GCP

Kubeflow

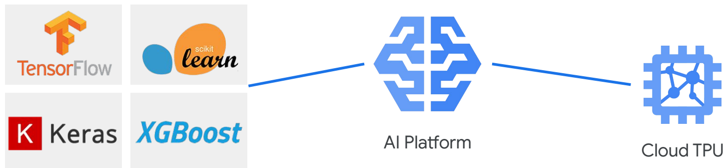
AI Hub



Google Cloud


Create and deploy custom models with Kubeflow
Cloud TPUs
Compute Engine
Dataproc
Google Kubernetes Engine
AI Platform
BigQuery ML
Build a Custom Model
AutoML
Build Custom Model (codeless)
Cloud Translation API
Vision API
Speech-to-Text API
Video Intelligence API
Data Loss Prevention API
Text-to-Speech API
Cloud Natural Language API
Dialogflow
Call a Pretrained Model
Google Cloud

AI Platform is a fully managed service for custom machine learning models



- Scales to production
- Batching and distribution of model training
- Performs transformations on input data
- Hyper-parameter tuning
- Host and autoscale predictions
- Serverless - self-tuning - manages overhead

Google Cloud

In this course, we don't cover writing TensorFlow models, only ways to operationalize them

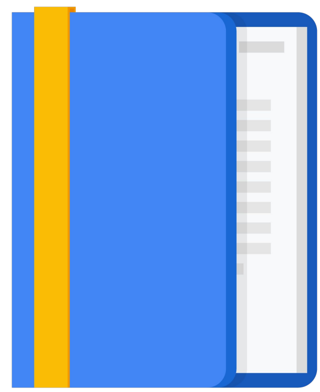
Google Cloud Training - Machine Learning and AI

Agenda

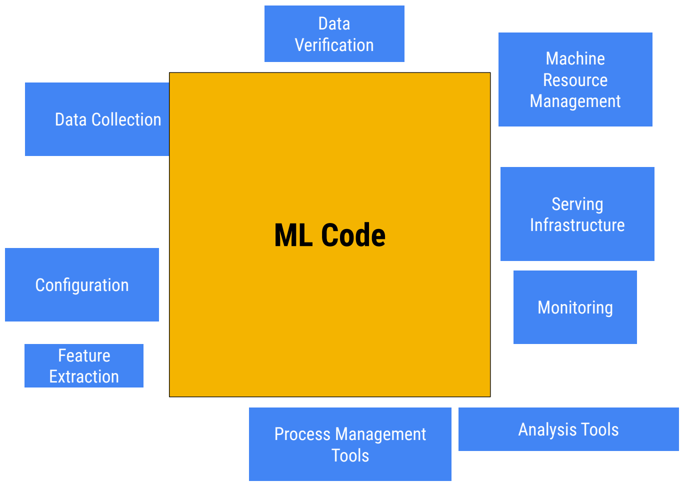
Ways to do ML on GCP

Kubeflow

AI Hub

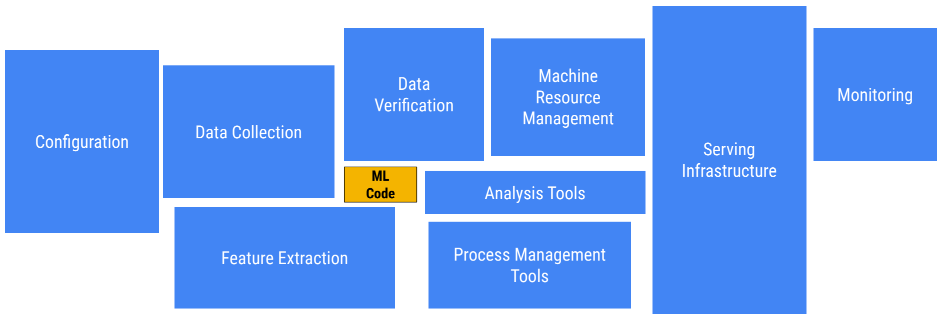


Perception: ML products are mostly about ML



Google Cloud

Reality: ML Requires lots of DevOps



Source: Sculley et al.: Hidden Technical Debt in Machine Learning Systems

ML systems are large, complicated distributed systems.
We started building Kubeflow to tackle these devops challenges using Kubernetes and containers

Kubeflow provides a platform for building ML products

- Leverage containers and Kubernetes to solve the challenges of building ML products
- Kubeflow = Cloud Native, multi-cloud solution for ML.
- Kubeflow provides a platform for composable, portable and scalable ML pipelines
- If you have a Kubernetes conformant cluster, you can run Kubeflow

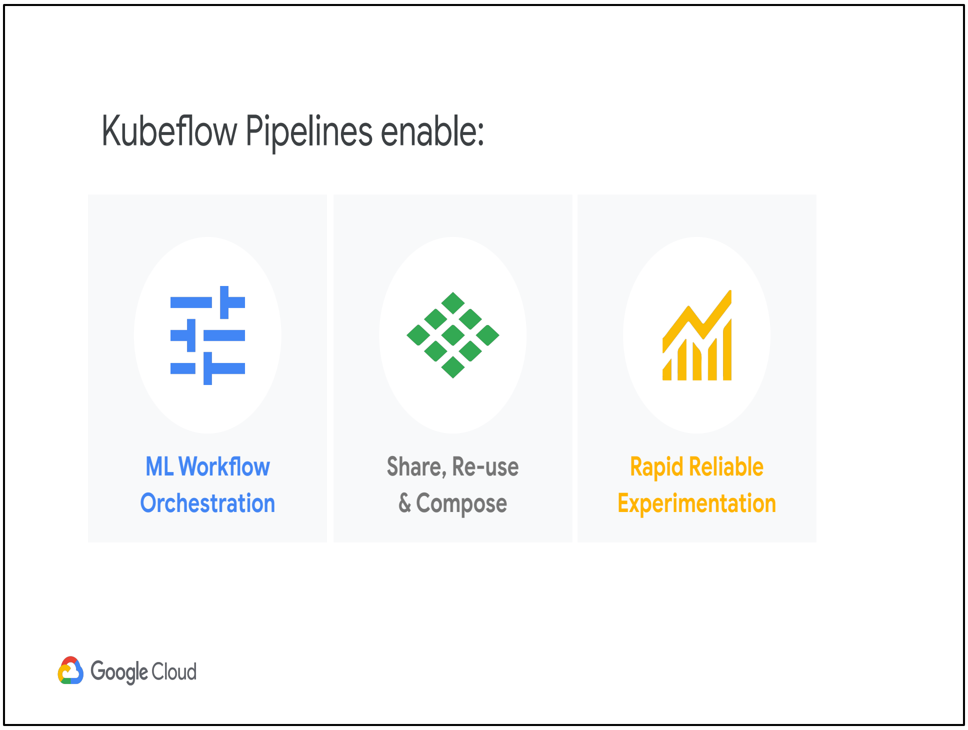
Google Cloud

Kubernetes is a great platform for ML

- Containers
- Scaling built in
- Unified architecture
- Easy to integrate building blocks
  - ML APIs
  - Dataflow
- Lots of options for CI/CD
- Portability
  - Dev, On-Prem, Multi-cloud: same stack

Before we dive into the specifics of Kubeflow Pipelines, I should provide additional context.

Kubeflow Pipelines is a part of the open source project Kubeflow.
Kubeflow is a platform that provides the tools and scalable services required to develop and deploy ML workloads, all the way from distributed training, to scalable serving, to Notebooks w/ JupyterHub and workflow orchestration and much more. Kubeflow services are built on top on Kubernetes. Kubernetes provides scalability and hybrid protability. You can run Kubeflow anywhere you can run a Kubernetes cluster, and thus applications built on Kubeflow are portable across clouds and on-premise environmetns. On GCP, You can easily deploy Kubeflow on Google Kubernetes Engine.



Ok now lets dive into Kubeflow Pipelines.

The capabilities provided by Kubeflow pipelines can largely be put into three buckets:

What constitutes a Kubeflow Pipeline

Containerized implementations of ML Tasks

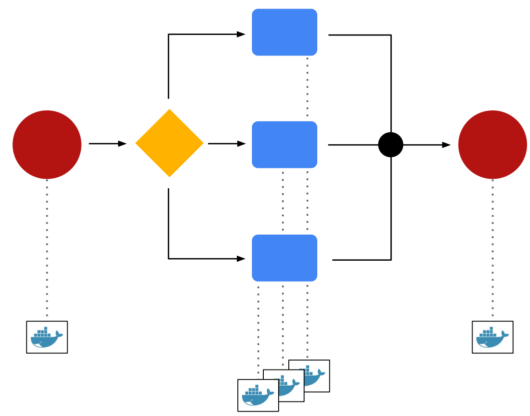
- Containers provide portability, repeatability and encapsulation
- A task can be single node or *distributed*
- A containerized task can invoke other services like CMLE, Dataflow or Dataproc

Specification of the sequence of steps

- Specified via Python SDK

Input Parameters

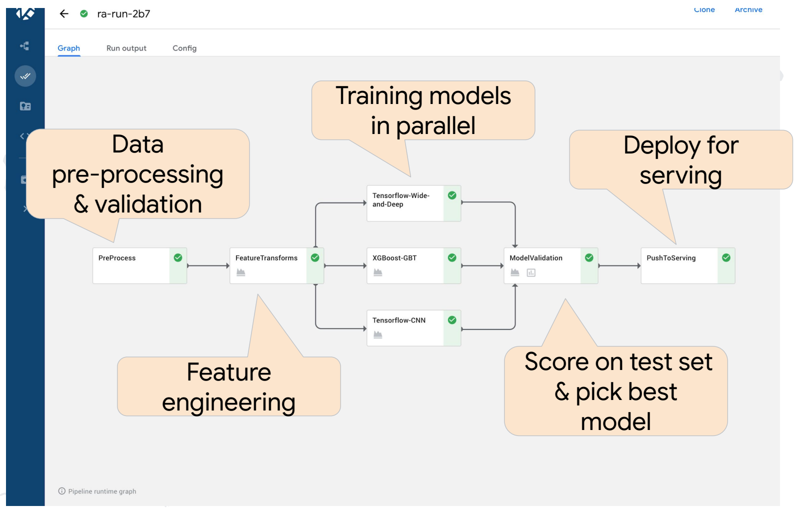
- A “Job” = Pipeline invoked w/ specific parameters



Google Cloud

Let me provide a peek under the hood:

Visual depiction of pipeline topology



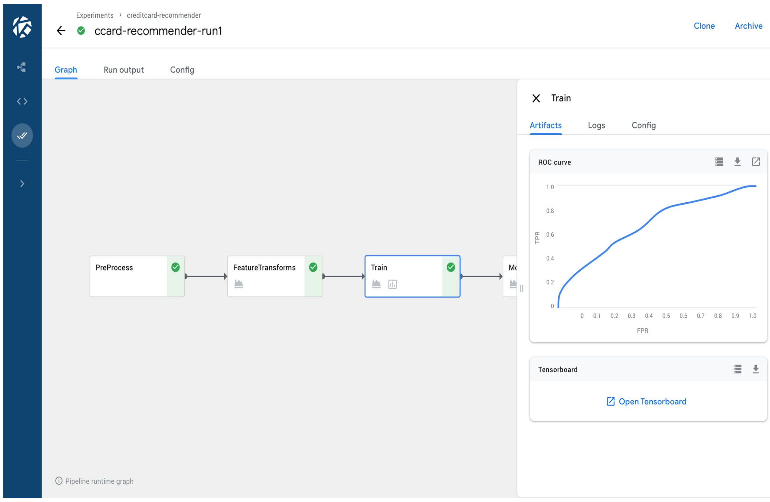
Google Cloud

To make things more concrete lets look at a screenshot of an illustrative workflow that was run on Kubeflow Pipelines.
This is just an illustrative workflow and users can author and run many different kinds of workflow topologies with different code&tools in the various steps of the workflow

For each workflow that is run on Kubeflow PIpelines, you get a rich visual depticton of the topology so that you know what was executed as part of the workflow

In this workflow we start with a data preprocessign and validation step
Followed by feature engineeirng
Following by a fork where we traing many different kinds of models
The models that are trained are then analyzed and compared on a test data set
Finally, if an improved model is produced, it is deployed to a serving endpoint

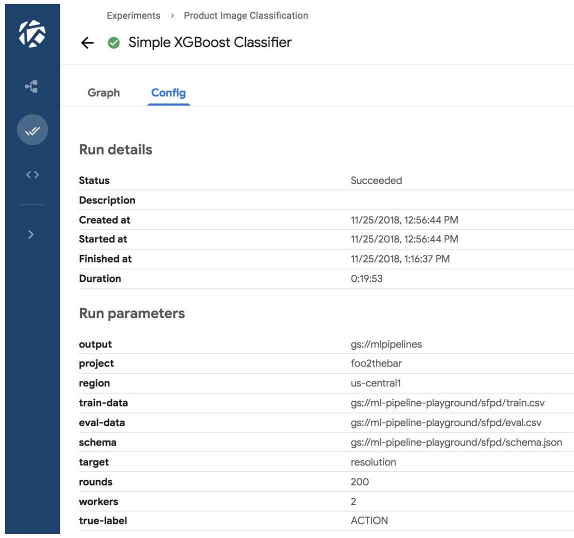
Rich visualization of metrics



Google Cloud

For each step of the workflow, you have rich ML specific information at your finger tips. Just click on a step and visualize relevant metrics produced by that step, such as an ROC curve for example.
If you did model training and produced the rich metadata that can be visualized with TensorBoard, that is just a click away.

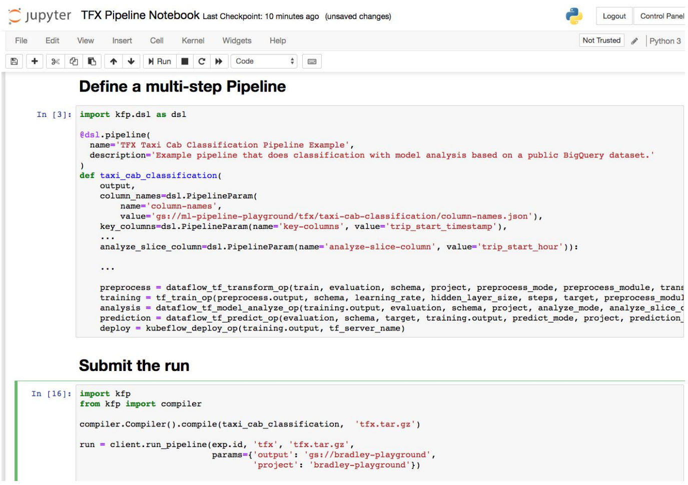
View all configs, inputs and outputs



Google Cloud

For each step of the workflow you can see the precise configuration parameters, and inputs and outputs.
Thus, for a model trained with Kubeflow PIpelines, you never have to wonder, how exactly did I create this model

Author pipelines with an intuitive Python SDK



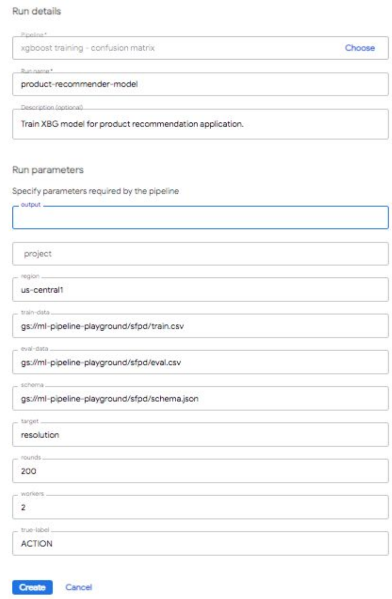
Google Cloud

Lingua franca of ML practitioners

Python SDK to define the workflow i.e. define every step of teh workflow, the inputs and output, and also define how the various steps are connected. The topology of the workflow is implicitly defined by connecting the various steps i.e. connecting the outputs of an upstream step to the inputs of a downstream step.
You can also define looping constrtucts as well as conditional steps.

Package & share pipelines as zip files

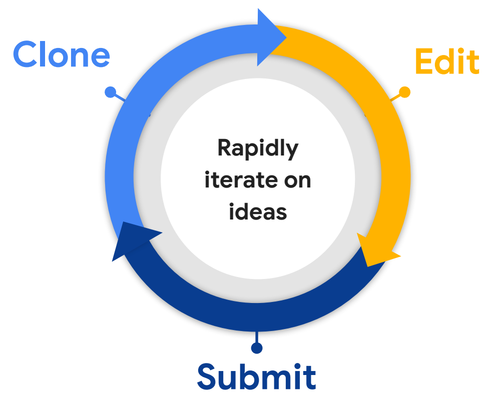
- Upload and execute pipelines via UI (in addition to API/SDK)
- Pipeline steps can be authored as reusable components



Google Cloud

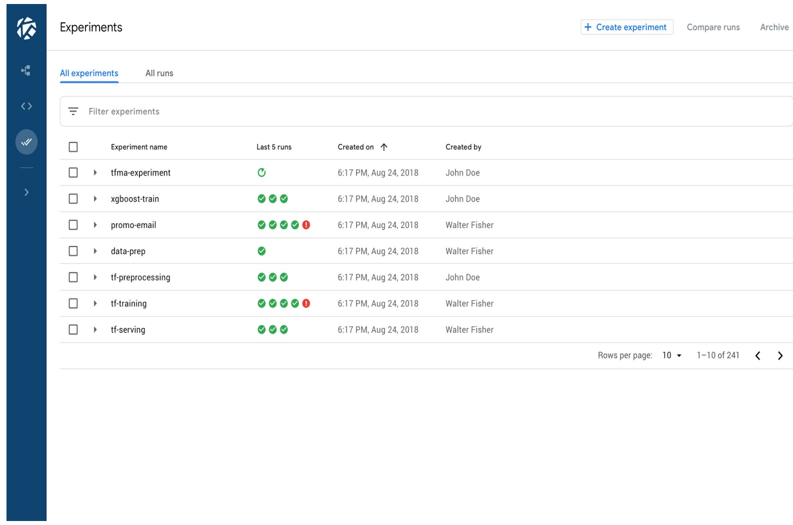
Rapid, Reliable, Experimentation

- Every run logged with all config params, inputs, outputs & metrics
- Easily search and find old runs
- Clone and re-run or modify



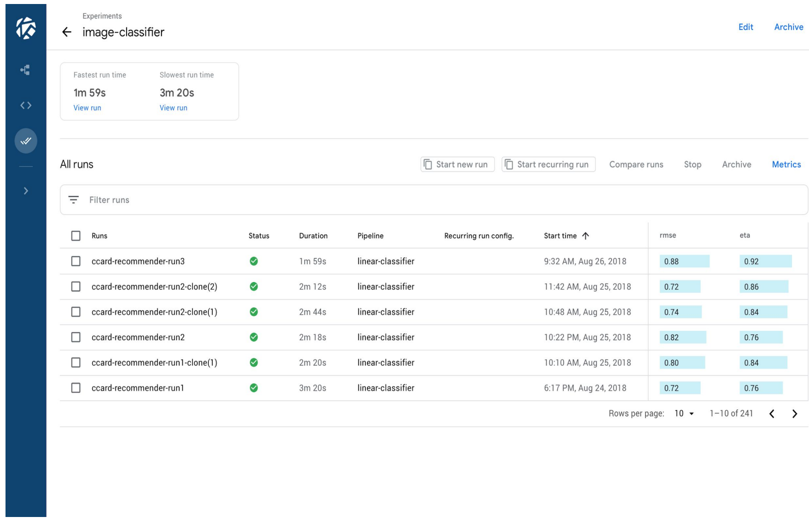
Google Cloud

View all current and past runs in one place



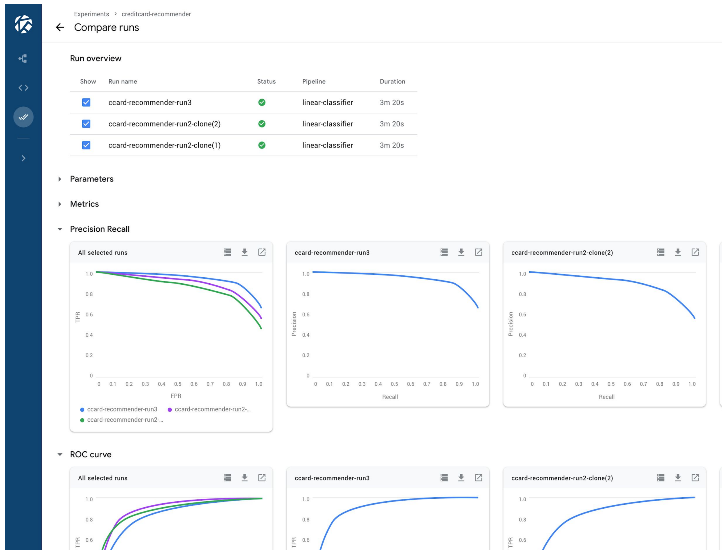
Google Cloud

Easy comparison and analysis of runs



Google Cloud

Easy comparison and analysis of runs



Google Cloud

Hone in on the technique or parameter that was different

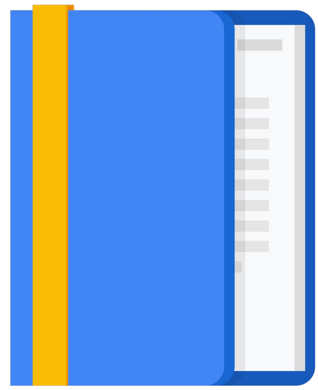
Quickly identify what worked and what did not work

Agenda

Ways to do ML on GCP

Kubeflow

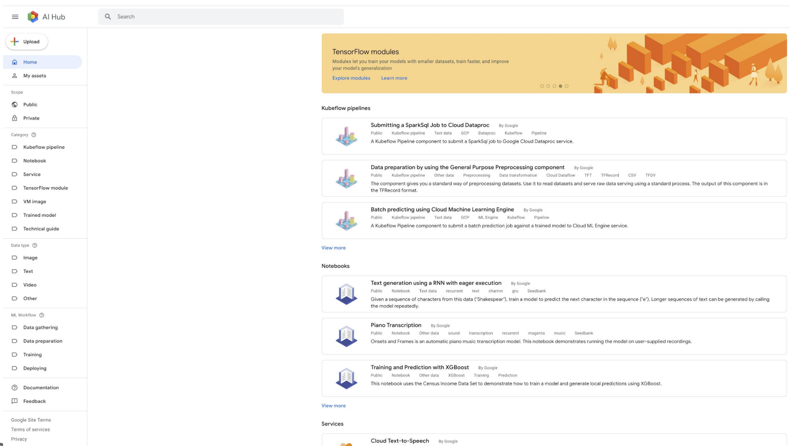
AI Hub



Google Cloud

AI Hub is a repository for AI assets

- Don’t reinvent the wheel! Find and deploy ML pipelines



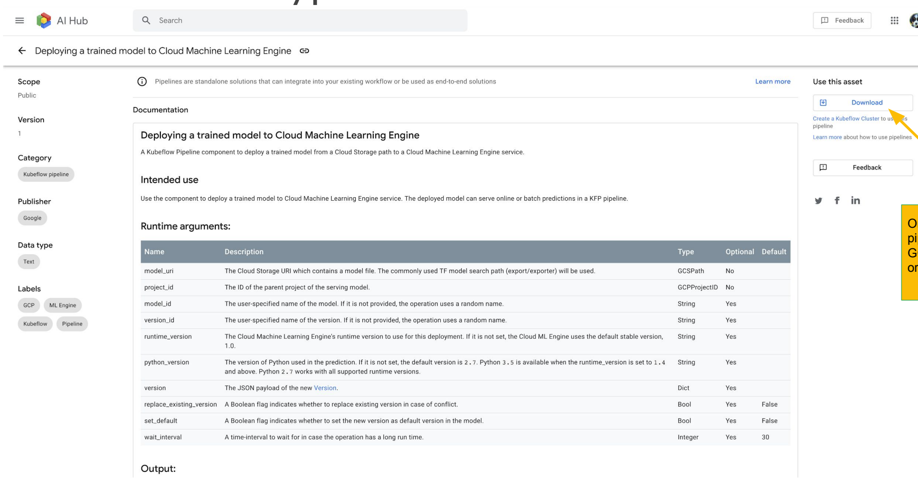
Google Cloud

AI Hub stores various asset types

- Kubeflow pipelines and components
- Jupyter notebooks
- TensorFlow modules
- Trained models
- Services
- VM images

Google Cloud

This is what a typical asset looks like...



Google Cloud

Assets on AI Hub are collected in two scopes: public assets and restricted assets

- Public scope are available to all AI Hub users
- Restricted scope contains AI components that you have uploaded and assets that have been shared with you

Lab

Running ML Pipelines on Kubeflow

Objectives

- Create a Kubernetes cluster and configure AI Platform pipelines
- Launch the pipelines dashboard
- Create and run an experiment from an example end-to-end ML Pipeline
- Examine and verify the output of each step
- Inspect the pipeline graph, various metrics, logs, charts and parameters

In this lab you learn how to install and use Kubeflow Pipelines.

Once Kubeflow Pipelines are installed, you create and run an experiment end-to-end ML Pipeline.

When the pipeline is complete you will examine the pipeline graph, metrics, logs and parameters.

Module Summary

- Use ML on GCP using either
  - AI Platform (your model, your data)
  - AutoML (our models, your data)
  - Perception API (our models, our data)
- Use Kubeflow to deploy end-to-end ML pipelines
- Don’t reinvent the wheel for your ML pipeline! Leverage pipelines on AI Hub

Google Cloud