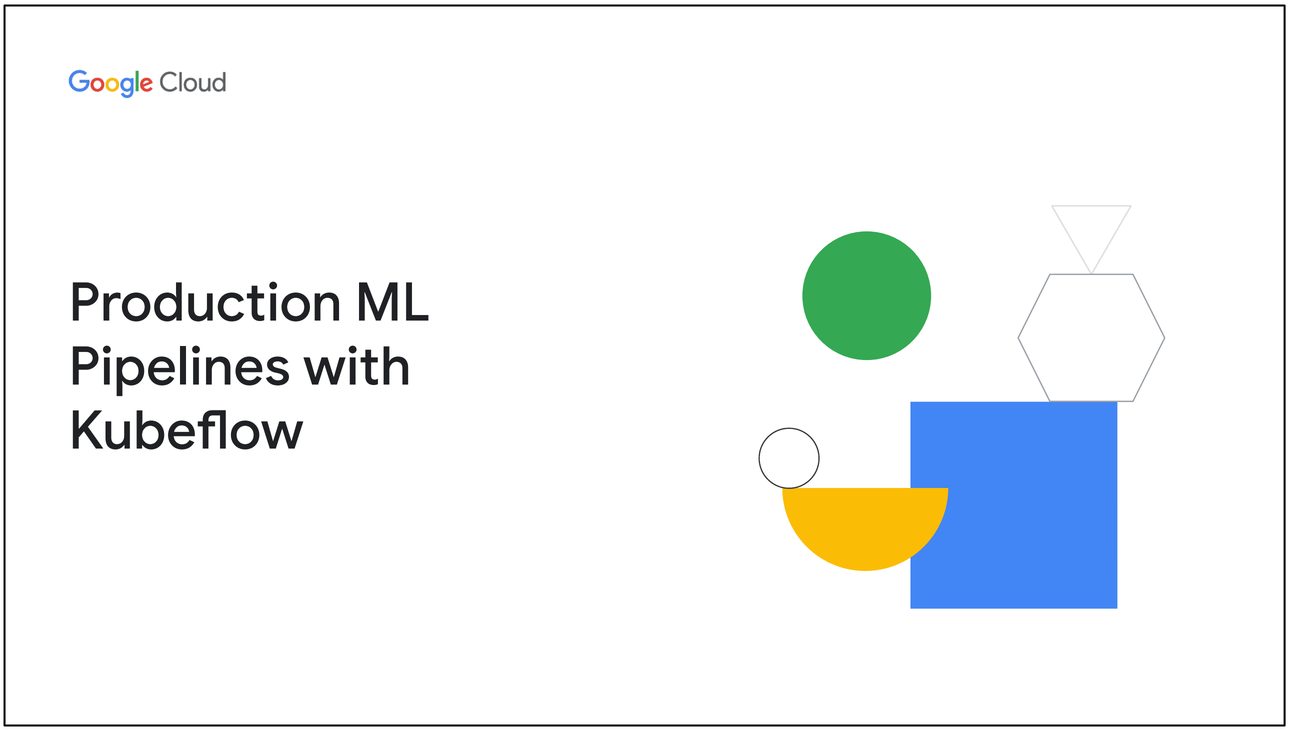
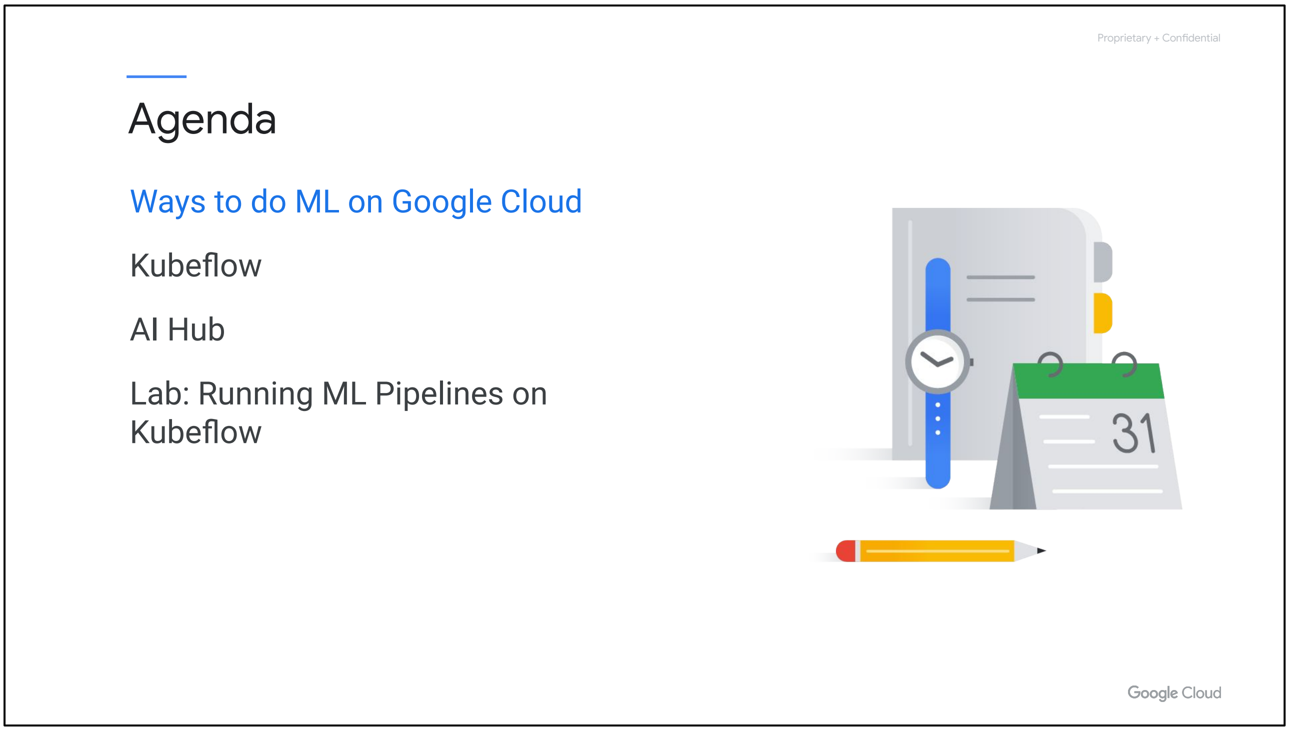


Google Cloud

Production ML Pipelines with Kubeflow



Proprietary + Confidential

Agenda

Ways to do ML on Google Cloud

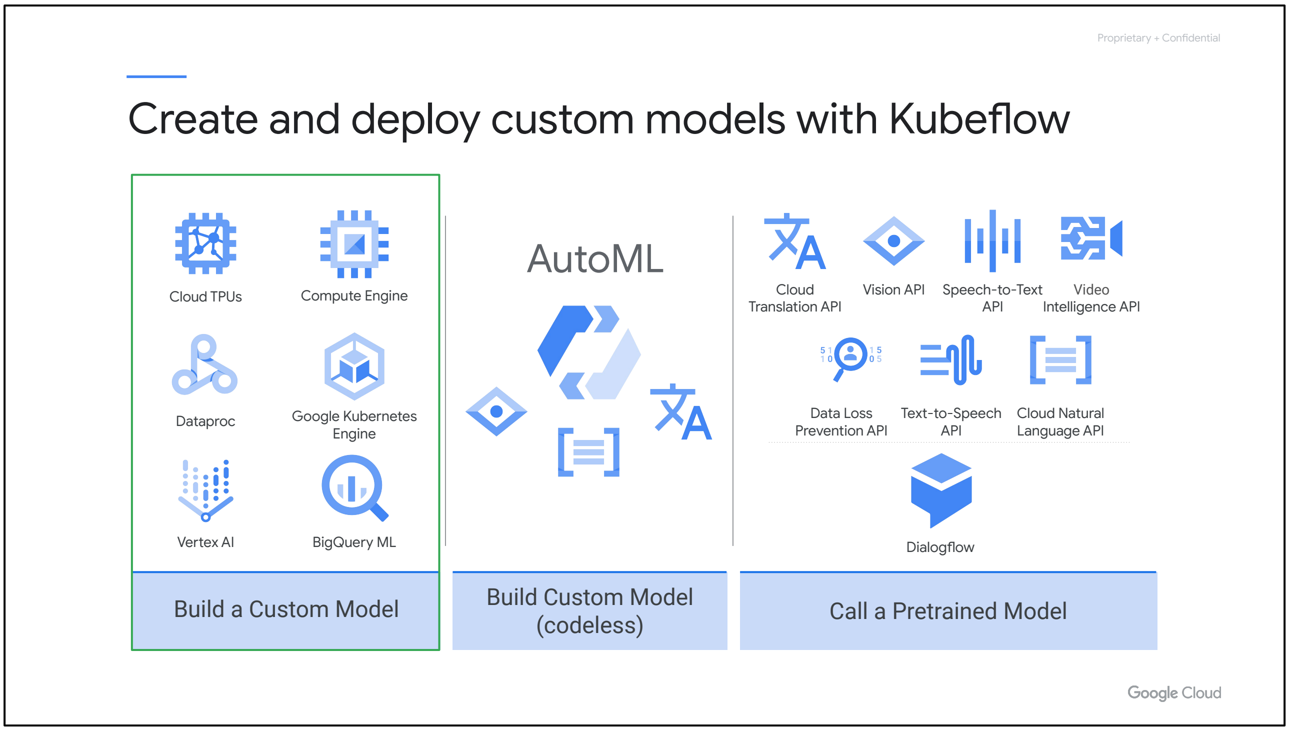
Kubeflow

AI Hub

Lab: Running ML Pipelines on Kubeflow

Google Cloud

In a previous module, we leveraged pre-trained ML APIs to process natural text. These are great options for seeing if your use case can just use a model that's already created and trained on Google's data. But, you may want a more tailored model trained on your own data. For that we will need a custom model. Let's talk about the different ways of building custom models.

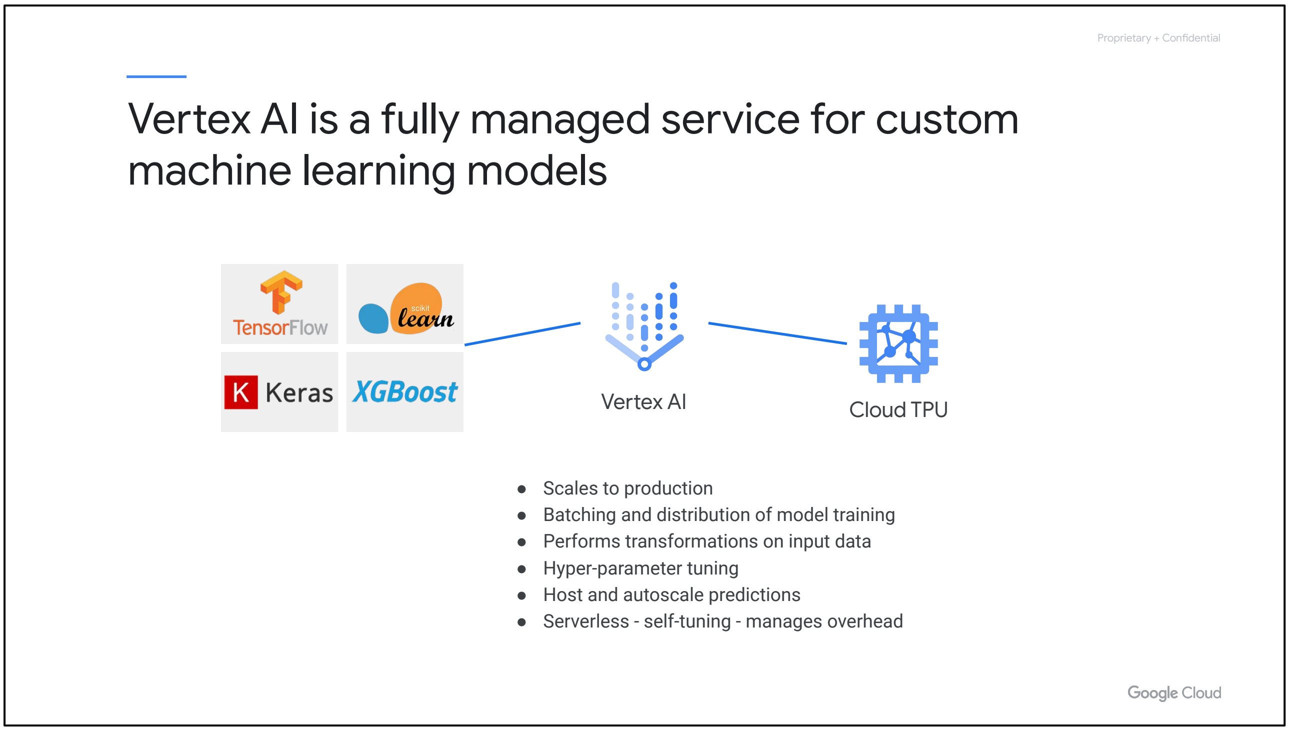


Proprietary + Confidential

Create and deploy custom models with Kubeflow

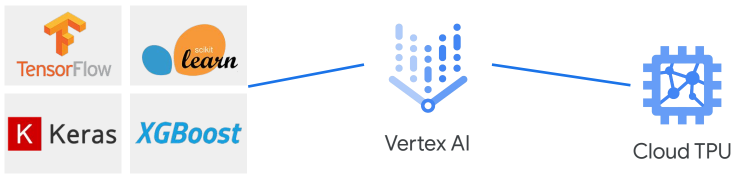
Google Cloud

As we covered previously, here are the three ways you can do machine learning on Google Cloud. We've already looked at pretrained models on the right. Now, we're going to visit the other side of the spectrum and build your own custom model and productionalize it on Google Cloud. There are a few ways of doing custom model development, training, and serving.



Proprietary + Confidential

Vertex AI 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

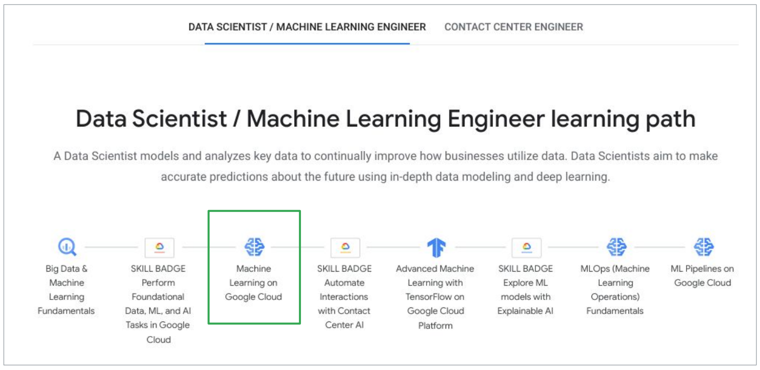
What is Vertex AI exactly? It's a fully managed service for custom machine learning models, both training and serving predictions. It can scale from the experimentation stage all the way to production. You can also, using features of TensorFlow, include transformations on input data and perform hyperparameter tuning to choose the best model for your case. You can deploy your models to Vertex AI to serve predictions, which will autoscale to the demands of your clients.

Vertex AI also supports Kubeflow which is Google's open source framework for building ML pipelines -- and you'll have a lab on this later.

Essentially, Vertex AI is the engine behind doing machine learning at scale on Google Cloud. A data scientist can train and deploy production models from Notebooks with just a few commands.

Proprietary + Confidential

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



DATA SCIENTIST / MACHINE LEARNING ENGINEER   CONTACT CENTER ENGINEER

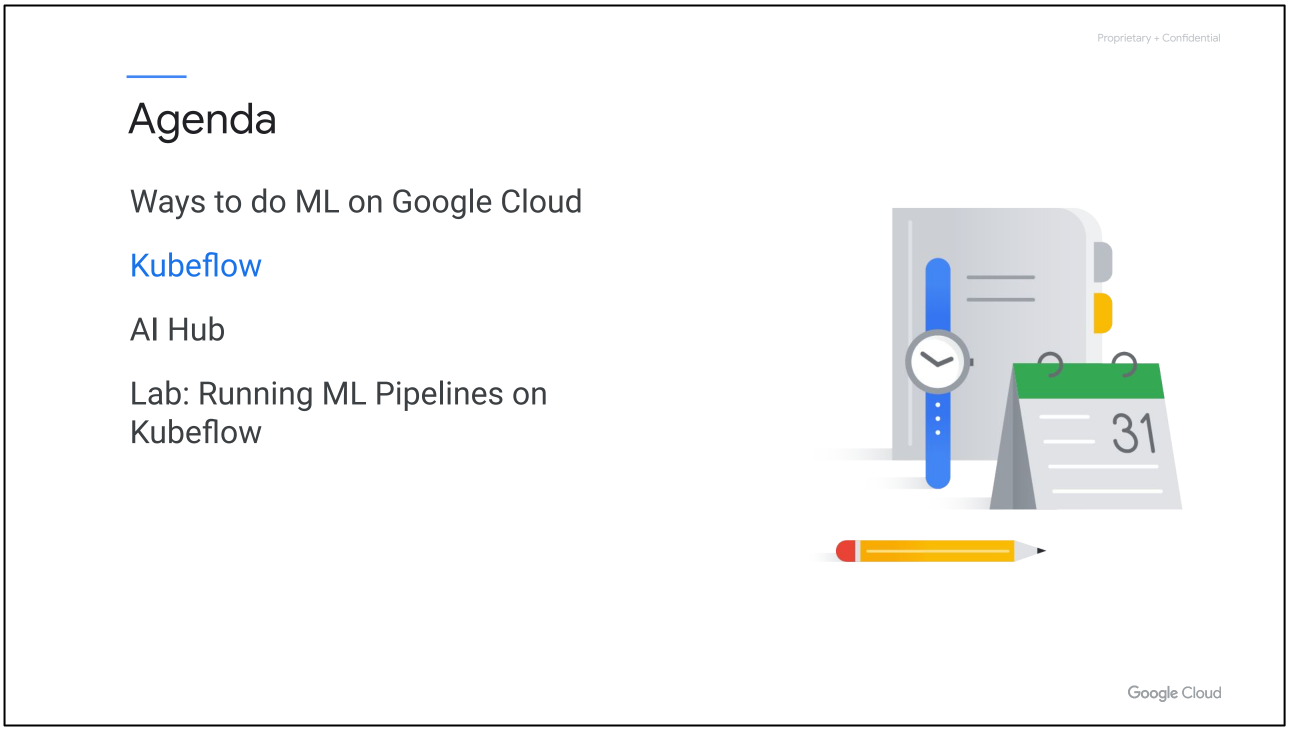
Data Scientist / Machine Learning Engineer learning path

A Data Scientist models and analyzes key data to continually improve how businesses utilize data. Data Scientists aim to make accurate predictions about the future using in-depth data modeling and deep learning.

- Big Data & Machine Learning Fundamentals
- SKILL BADGE Perform Foundational Data, ML, and AI Tasks in Google Cloud
- Machine Learning on Google Cloud
- SKILL BADGE Automate Interactions with Contact Center AI
- Advanced Machine Learning with TensorFlow on Google Cloud Platform
- SKILL BADGE Explore ML models with Explainable AI
- MLOps (Machine Learning Operations) Fundamentals
- ML Pipelines on Google Cloud

Google Cloud Training - Machine Learning and AI

Since we're using Vertex AI and Kubeflow, we will often be thinking about using TensorFlow models. However, this isn't the course to dive into the details of TensorFlow. You can learn more about this in the Machine Learning on Google Cloud course, which is path of the Machine Learning and AI learning path for Data Scientists and Machine Learning Engineers linked here.



Proprietary + Confidential

Agenda

Ways to do ML on Google Cloud

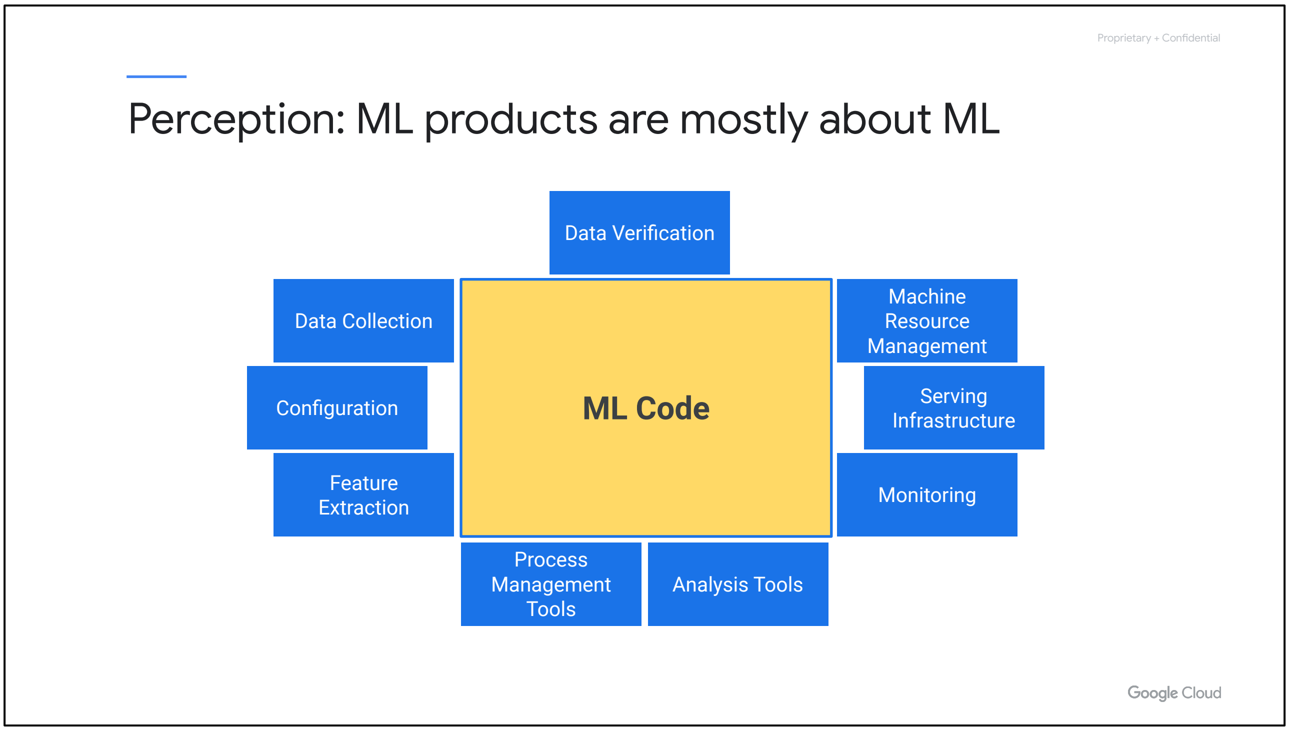
Kubeflow

AI Hub

Lab: Running ML Pipelines on Kubeflow

Google Cloud

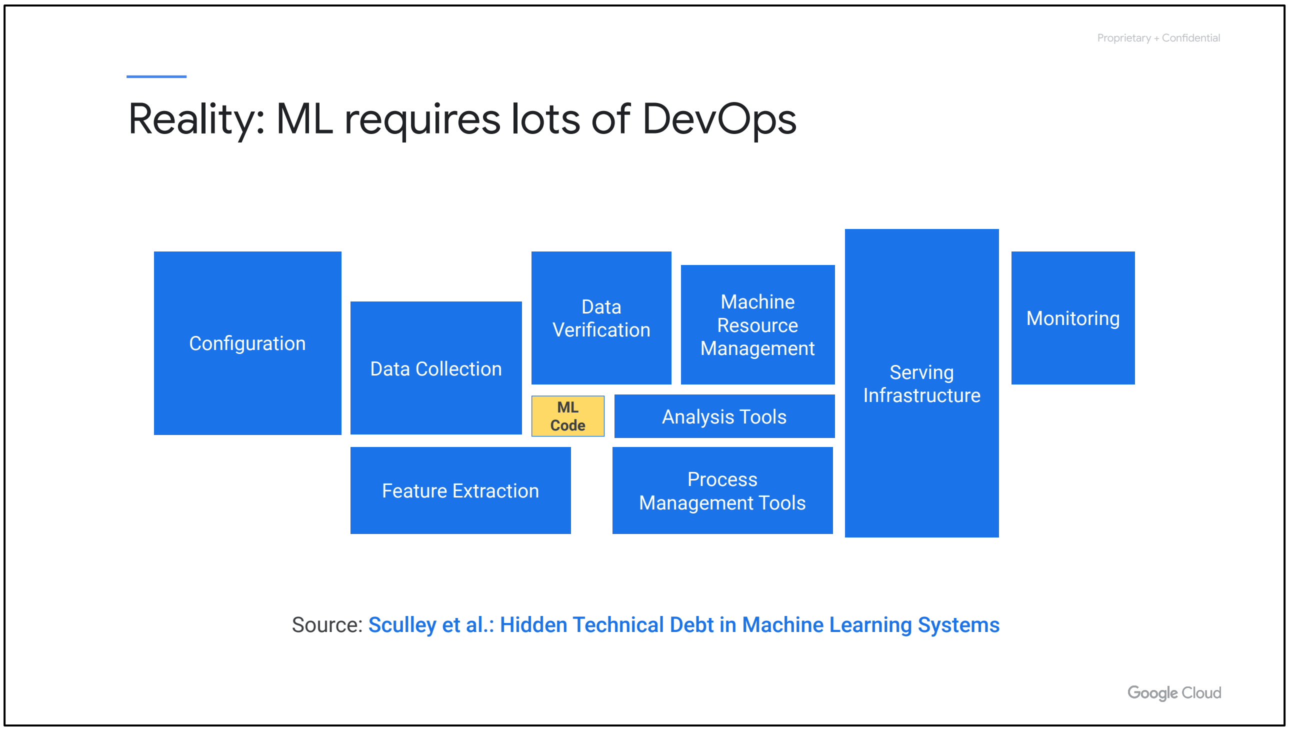
Where do Data Engineers come into the picture? Don’t forget Data Engineers build data pipelines, and machine learning pipelines are no different. If we want to have a flexible pipeline for all stages of machine learning, Kubeflow is a great option.


Proprietary + Confidential

Perception: ML products are mostly about ML

Google Cloud

Many people think that machine learning products are all about the code that ML scientists write locally on their machines. Does this code ensure the data going into it is clean? Can the code auto-scale to clients who want to use it for serving predictions? What if we have to re-train the model, does it go off-line at that point?



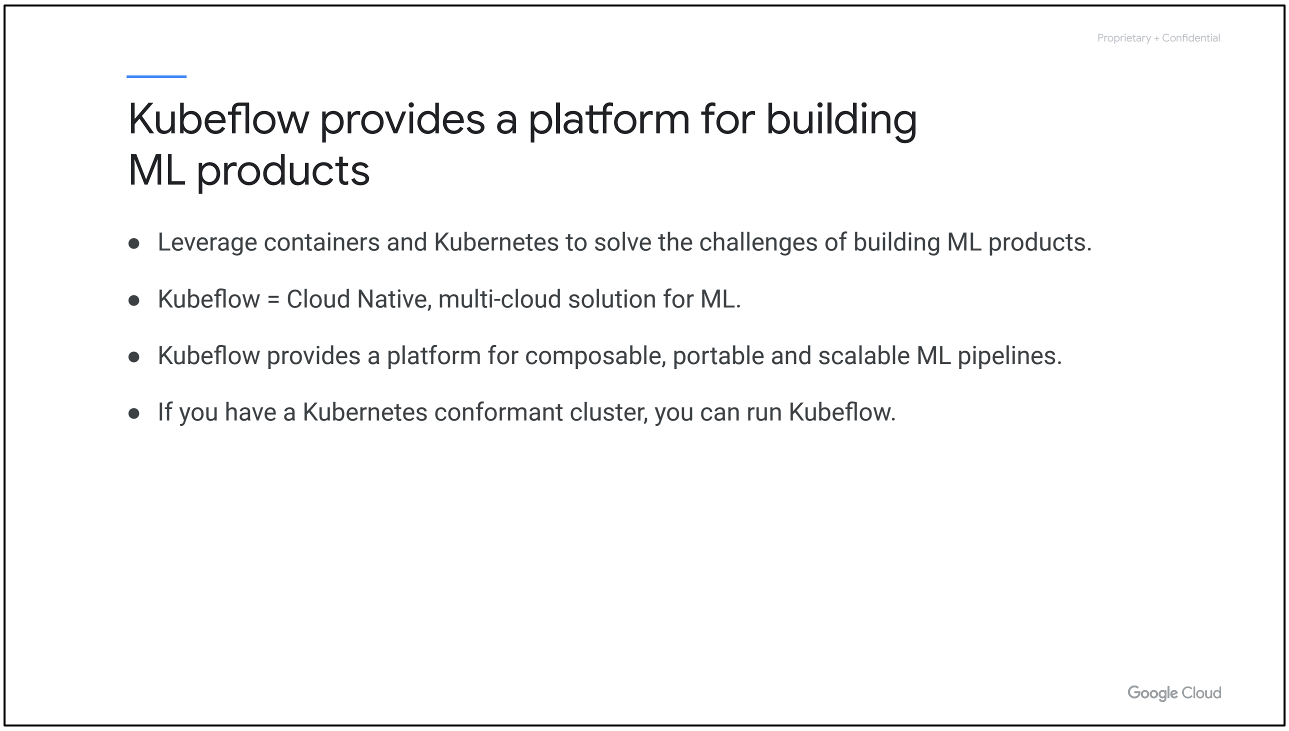
Proprietary + Confidential

Reality: ML requires lots of DevOps

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

Google Cloud

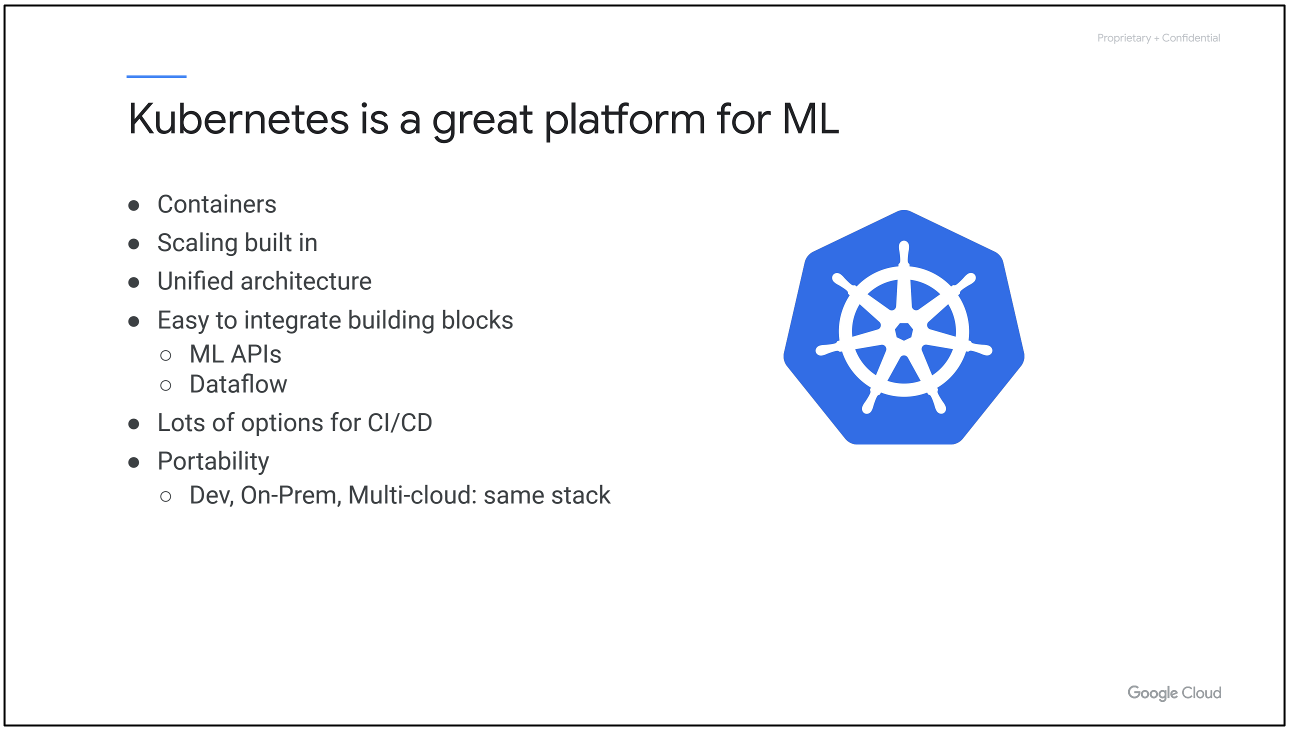
The truth is, production machine learning systems are large, complicated, distributed systems. There's a lot of DevOps involved for things like monitoring, and process management tools. Google started building Kubeflow to tackle these DevOps challenges using Kubernetes and containers.


Proprietary + Confidential

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



Proprietary + Confidential

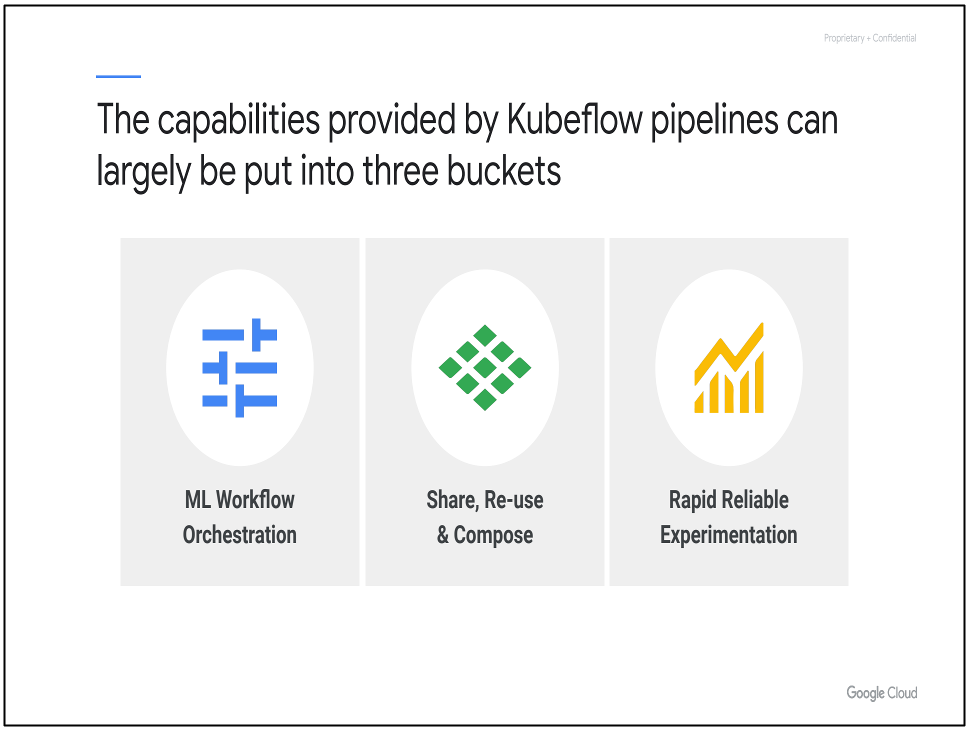
Kubernetes is a great platform for ML

- 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

Google Cloud

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 environments. On Google Cloud, you can easily deploy Kubeflow on Google Kubernetes Engine.



Proprietary + Confidential

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

ML Workflow Orchestration

Share, Re-use & Compose

Rapid Reliable Experimentation

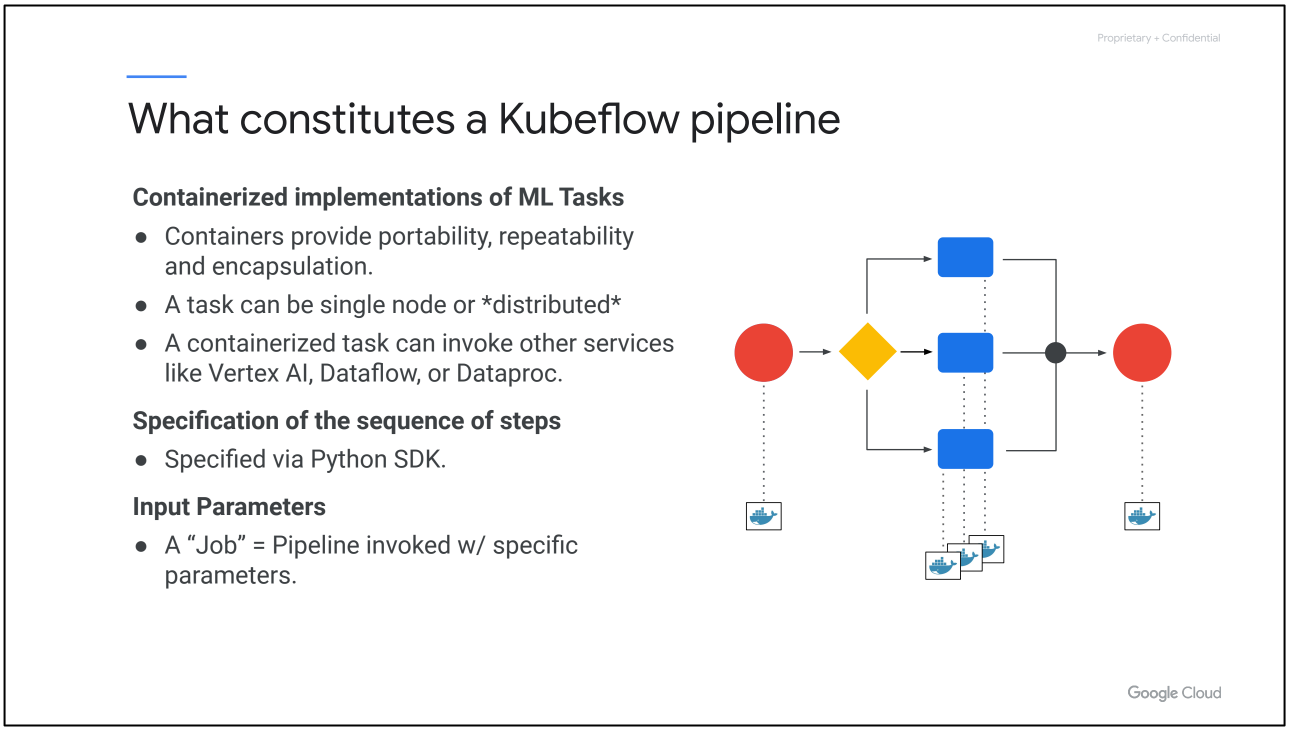
Google Cloud

One option to help manage the overhead of productionalizing ML pipelines is to use Kubeflow

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

- ML Workflow Orchestration
- Share, Re-use & Compose
- Rapid Reliable Experimentation

You can think of the benefits as similar to those of Cloud Composer but better tailored for ML workloads. Let’s see what a pipeline looks like.



Proprietary + Confidential

What constitutes a Kubeflow pipeline

Containerized implementations of ML Tasks

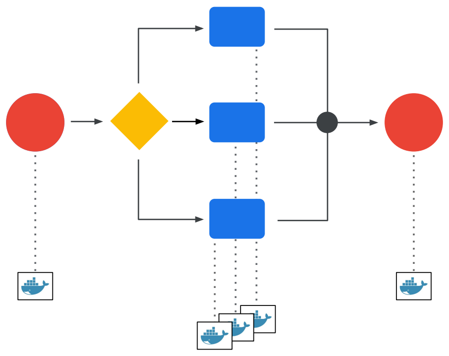
- Containers provide portability, repeatability and encapsulation.
- A task can be single node or *distributed*
- A containerized task can invoke other services like Vertex AI, Dataflow, or Dataproc.

Specification of the sequence of steps

- Specified via Python SDK.

Input Parameters

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

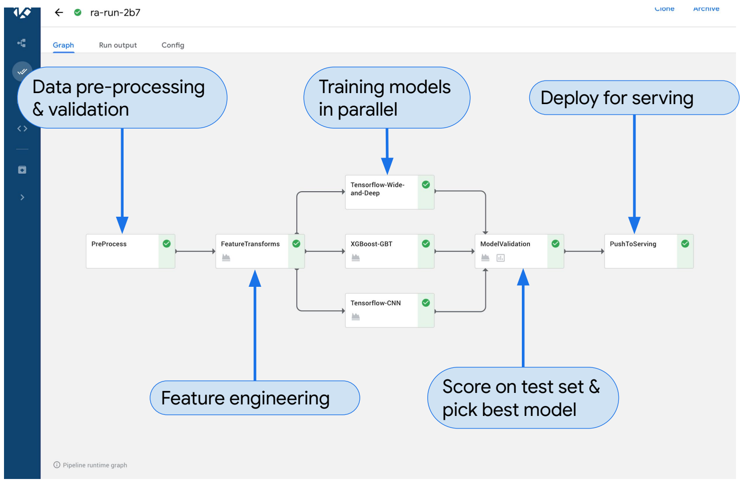


Google Cloud

Let me provide a peek under the hood.

Proprietary + Confidential

Visual depiction of pipeline topology



Google Cloud

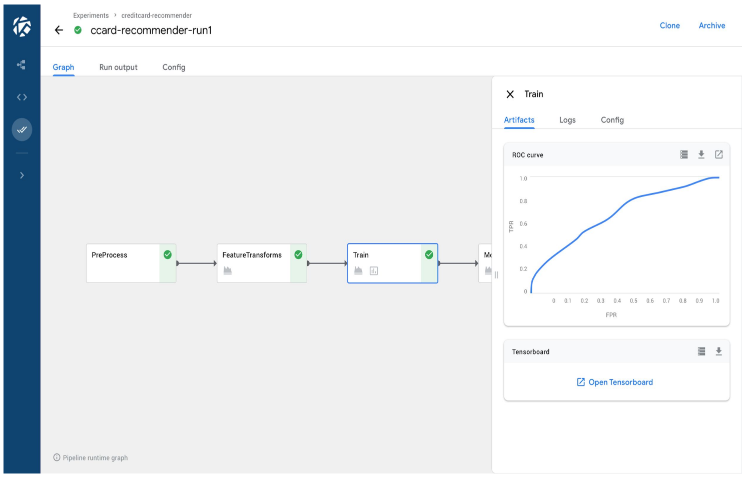
To make things more concrete let's look at a screenshot of an illustrative workflow that was run on Kubeflow Pipelines. This is just an illustrative workflow and users can author and run many different kinds of workflow topologies with different code and tools in the various steps of the workflow.

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

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

Proprietary + Confidential

Rich visualization of metrics

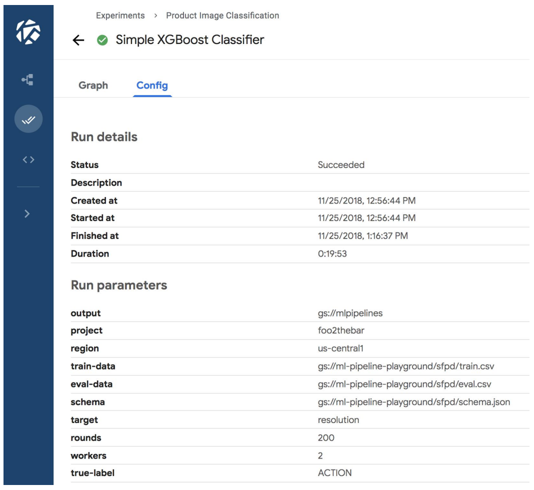


Google Cloud

For each step of the workflow, you have rich ML specific information at your fingertips. 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.

Proprietary + Confidential

View all configs, inputs and outputs


Experiments › Product Image Classification
Simple XGBoost Classifier
Graph
Config

Run details

Status	Succeeded
Description	
Created at	11/25/2018, 12:56:44 PM
Started at	11/25/2018, 12:56:44 PM
Finished at	11/25/2018, 1:16:37 PM
Duration	0:19:53

Run parameters

output	gs://mlpipelines
project	foo2thebar
region	us-central1
train-data	gs://ml-pipeline-playground/sfpd/train.csv
eval-data	gs://ml-pipeline-playground/sfpd/eval.csv
schema	gs://ml-pipeline-playground/sfpd/schema.json
target	resolution
rounds	200
workers	2
true-label	ACTION

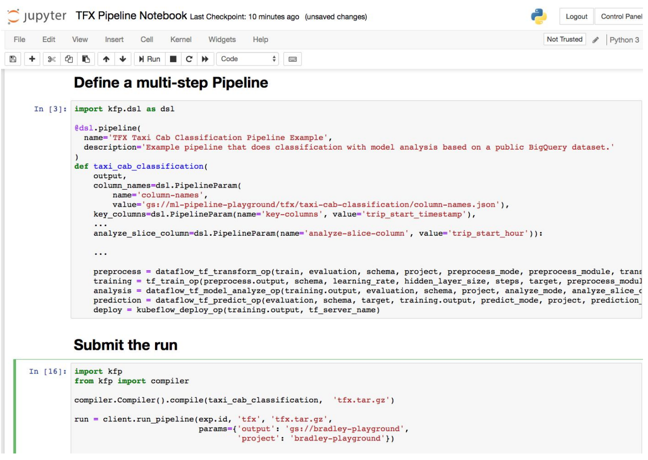
Google Cloud

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

Here you can quickly see how long the model training took, where the trained model is, and what data was used for training and evaluation.

Proprietary + Confidential

Author pipelines with an intuitive Python SDK



Define a multi-step Pipeline

```
In [3]: import kfp.dsl as dsl

        @dsl.pipeline(
          name='TFX Taxi Cab Classification Pipeline Example',
          description='Example pipeline that does classification with model analysis based on a public BigQuery dataset.'
        )
        def taxi_cab_classification(
            output,
            column_names=dsl.PipelineParam(
                name='column-names',
                value='gs://ml-pipeline-playground/tfx/taxi-cab-classification/column-names.json'),
            key_columns=dsl.PipelineParam(name='key-columns', value='trip_start_timestamp'),
            ...
            analyze_slice_column=dsl.PipelineParam(name='analyze-slice-column', value='trip_start_hour')):

            ...

            preprocess = dataflow_tf_transform_op(train, evaluation, schema, project, preprocess_mode, preprocess_module, trans
            training = tf_train_op(preprocess.output, schema, learning_rate, hidden_layer_size, steps, target, preprocess_modul
            analysis = dataflow_tf_model_analyze_op(training.output, evaluation, schema, project, analyze_mode, analyze_slice_c
            prediction = dataflow_tf_predict_op(evaluation, schema, target, training.output, predict_mode, project, prediction_
            deploy = kubeflow_deploy_op(training.output, tf_server_name)
```

Submit the run

```
In [16]: import kfp
         from kfp import compiler

         compiler.Compiler().compile(taxi_cab_classification,  'tfx.tar.gz')

         run = client.run_pipeline(exp.id, 'tfx', 'tfx.tar.gz',
                                   params={'output': 'gs://bradley-playground',
                                           'project': 'bradley-playground'})
```

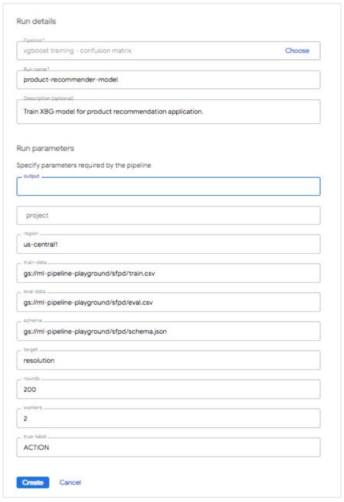
Google Cloud

You can define the ML workflow using Kubeflow's Python SDK. By defining the workflow we mean specifying each step's inputs and outputs and how the various steps are connected. The topology of the workflow is implicitly defined by connecting the outputs of an upstream step to the inputs of a downstream step. You can also define looping constructs as well as conditional steps.

Proprietary + Confidential

Package and share pipelines as zip files

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


Run details

Pipeline*	xgboost training - confusion matrix
Run name*	product-recommender-model
Description (optional)	Train XBG model for product recommendation application.

Run parameters

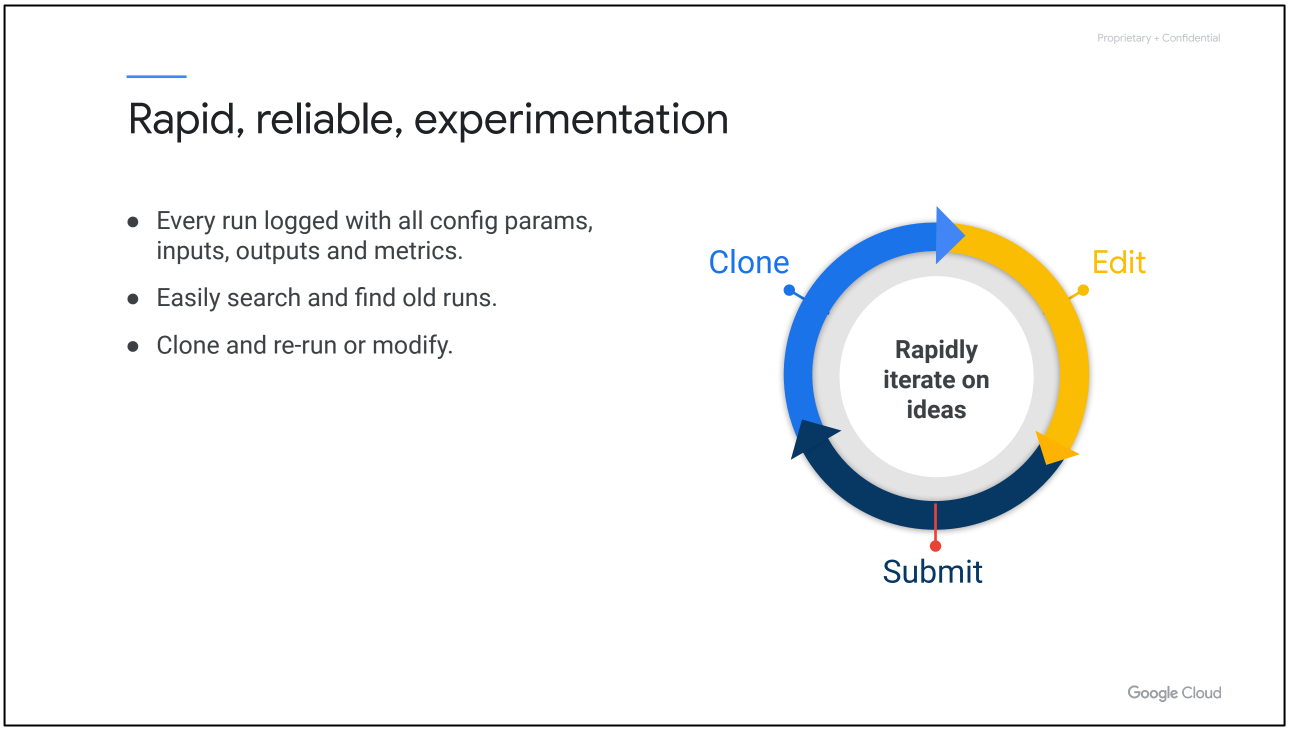
Specify parameters required by the pipeline

output	
project	
region	us-central1
train-data	gs://ml-pipeline-playground/sfpd/train.csv
eval-data	gs://ml-pipeline-playground/sfpd/eval.csv
schema	gs://ml-pipeline-playground/sfpd/schema.json
target	resolution
rounds	200
workers	2
true-label	ACTION

Google Cloud

Another nice Kubeflow feature is the ability to package pipeline components. This adds an element of portability since you can then move your ML pipelines even between cloud providers.

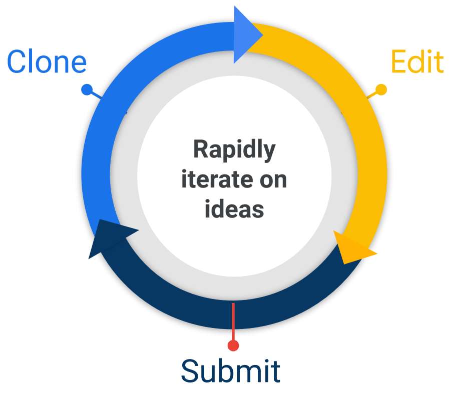
Kubeflow pipelines separate the work for different parts of the pipeline to enable people to specialize. For example, a ML engineer can focus on feature engineering and the other parts of creating the model such as hyperparameter tuning. The ML engineer's solutions can then be bundled up and used by a data engineer as part of a data engineering solution. The solution can then appear as a service used by a data analyst to derive business insights.



Proprietary + Confidential

Rapid, reliable, experimentation

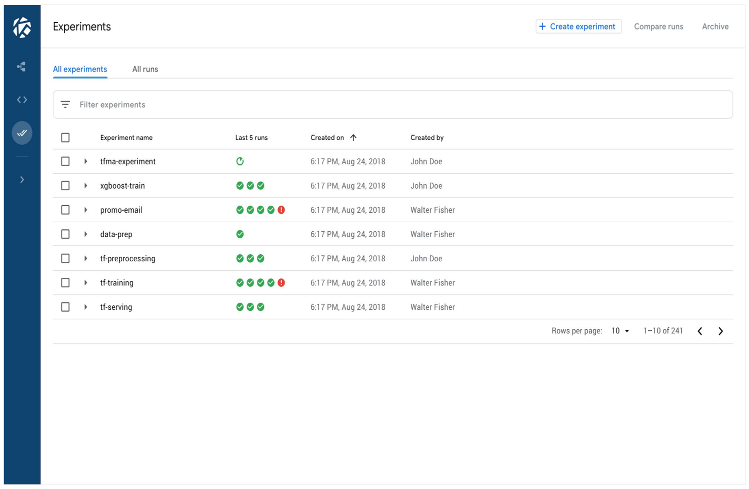
- Every run logged with all config params, inputs, outputs and metrics.
- Easily search and find old runs.
- Clone and re-run or modify.



Google Cloud

Proprietary + Confidential

View all current and past runs in one place



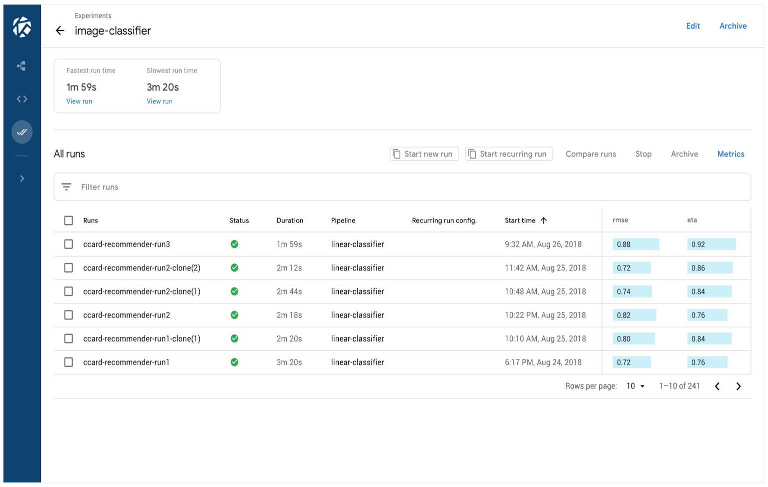
Experiment name	Last 5 runs	Created on	Created by
tfma-experiment		6:17 PM, Aug 24, 2018	John Doe
xgboost-train	✓✓✓	6:17 PM, Aug 24, 2018	John Doe
promo-email	✓✓✓✓!	6:17 PM, Aug 24, 2018	Walter Fisher
data-prep	✓	6:17 PM, Aug 24, 2018	Walter Fisher
tf-preprocessing	✓✓✓	6:17 PM, Aug 24, 2018	John Doe
tf-training	✓✓✓✓!	6:17 PM, Aug 24, 2018	Walter Fisher
tf-serving	✓✓✓	6:17 PM, Aug 24, 2018	Walter Fisher

Rows per page: 10 1–10 of 241

Google Cloud

Proprietary + Confidential

Easy comparison and analysis of runs



Experiments

image-classifier

Fastest run time 1m 59s View run

Slowest run time 3m 20s View run

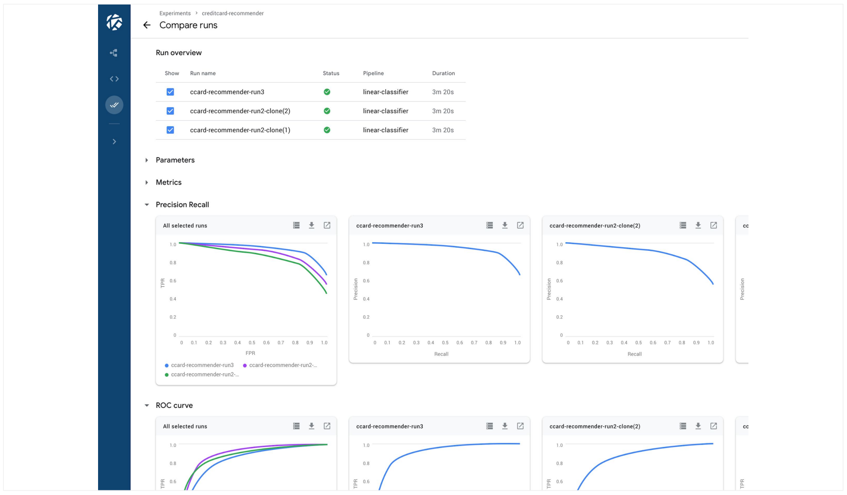
All runs

Runs	Status	Duration	Pipeline	Recurring run config.	Start time ↑	rmse	eta
ccard-recommender-run3	✓	1m 59s	linear-classifier		9:32 AM, Aug 26, 2018	0.88	0.92
ccard-recommender-run2-clone(2)	✓	2m 12s	linear-classifier		11:42 AM, Aug 25, 2018	0.72	0.86
ccard-recommender-run2-clone(1)	✓	2m 44s	linear-classifier		10:48 AM, Aug 25, 2018	0.74	0.84
ccard-recommender-run2	✓	2m 18s	linear-classifier		10:22 PM, Aug 25, 2018	0.82	0.76
ccard-recommender-run1-clone(1)	✓	2m 20s	linear-classifier		10:10 AM, Aug 25, 2018	0.80	0.84
ccard-recommender-run1	✓	3m 20s	linear-classifier		6:17 PM, Aug 24, 2018	0.72	0.76

Rows per page: 10 1–10 of 241

Proprietary + Confidential

Easy comparison and analysis of runs


Experiments › creditcard-recommender

Compare runs

Run overview

Show	Run name	Status	Pipeline	Duration
✓	ccard-recommender-run3	✓	linear-classifier	3m 20s
✓	ccard-recommender-run2-clone(2)	✓	linear-classifier	3m 20s
✓	ccard-recommender-run2-clone(1)	✓	linear-classifier	3m 20s

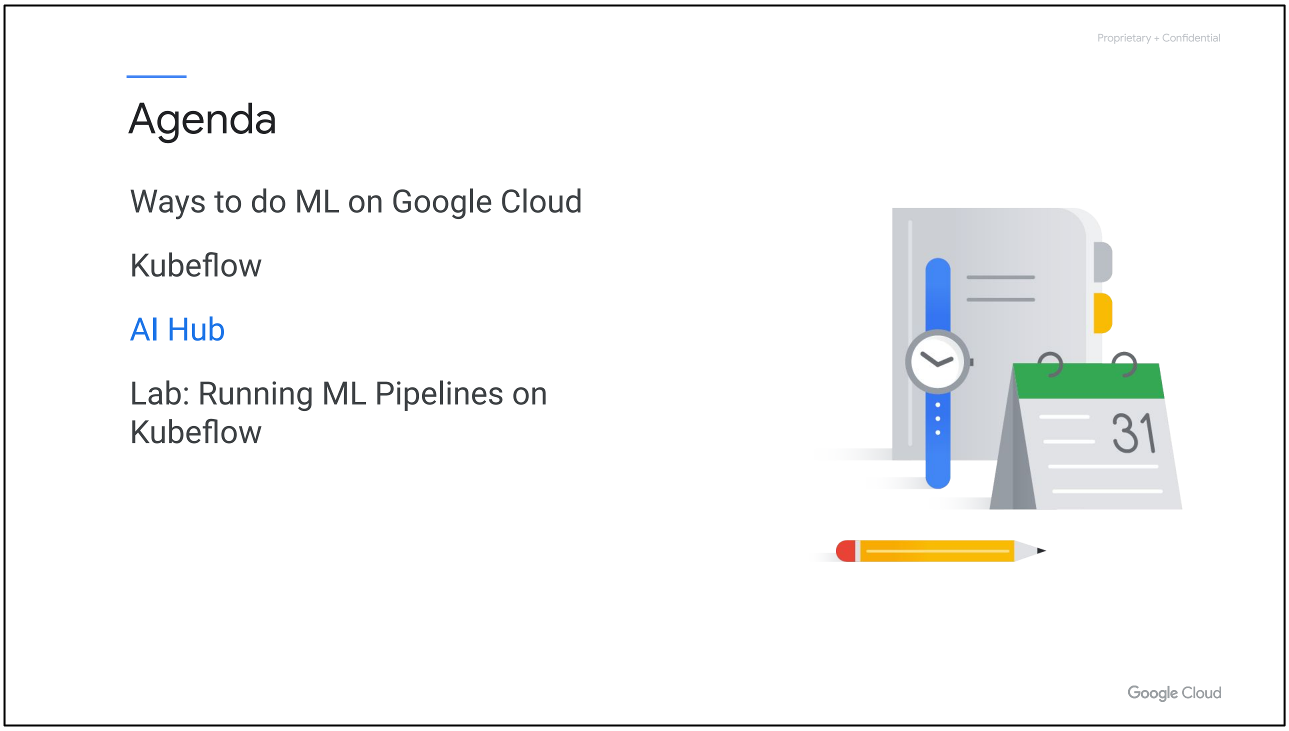
Parameters

Metrics

Precision Recall

Google Cloud

Kubeflow makes it easy to run a number of ML experiments at the same time. For example, if you're doing hyperparameter optimization, you can easily deploy a number of different training instances with different hyperparameter sets. Kubeflow's run overview makes it easy to hone in on the techniques or parameters generating the best results. You can quickly identify what worked and what did not work.



Proprietary + Confidential

Agenda

Ways to do ML on Google Cloud

Kubeflow

AI Hub

Lab: Running ML Pipelines on Kubeflow

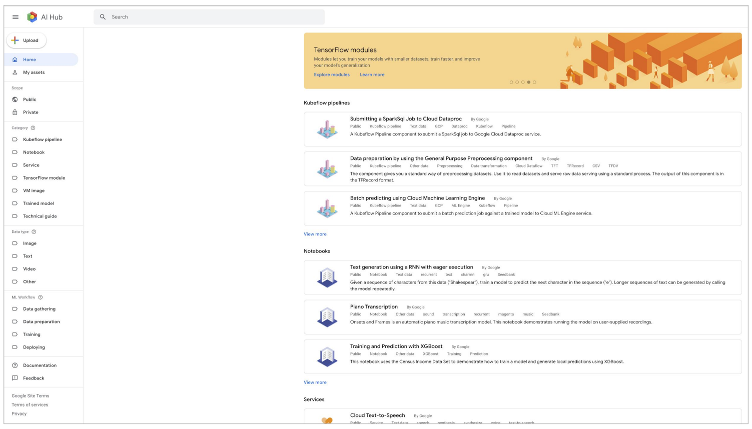
Google Cloud

We mentioned that Kubeflow pipelines can be packaged and shared with other users. This leads us to a discussion of AI Hub.

Proprietary + Confidential

AI Hub is a repository for AI assets

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



TensorFlow modules

Modules let you train your models with smaller datasets, train faster, and improve your model's generalization

Kubeflow pipelines

Submitting a SparkSql Job to Cloud Dataproc By Google
Public Kubeflow pipeline Text data GCP Dataproc Kubeflow Pipeline
A Kubeflow Pipeline component to submit a SparkSql job to Google Cloud Dataproc service.

Data preparation by using the General Purpose Preprocessing component By Google
Public Kubeflow pipeline Other data Preprocessing Data transformation Cloud Dataflow TFT TFRecord CSV TFDV
The component gives you a standard way of preprocessing datasets. Use it to read datasets and serve raw data serving using a standard process. The output of this component is in the TFRecord format.

Batch predicting using Cloud Machine Learning Engine By Google
Public Kubeflow pipeline Text data GCP ML Engine Kubeflow Pipeline
A Kubeflow Pipeline component to submit a batch prediction job against a trained model to Cloud ML Engine service.

Notebooks

Text generation using a RNN with eager execution By Google
Public Notebook Text data recurrent text charrnn gru Seedbank
Given a sequence of characters from this data ("Shakespear"), train a model to predict the next character in the sequence ("e"). Longer sequences of text can be generated by calling the model repeatedly.

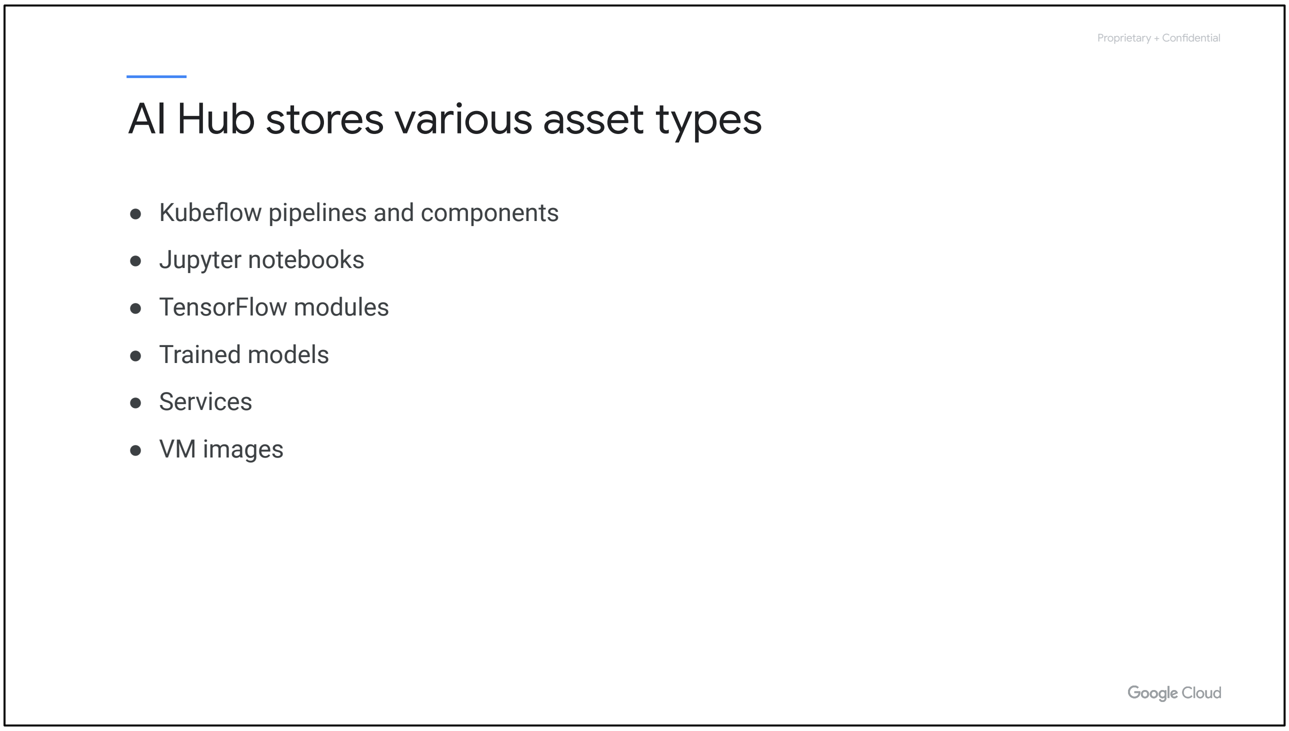
Piano Transcription By Google
Public Notebook Other data sound transcription recurrent magenta music Seedbank
Onsets and Frames is an automatic piano music transcription model. This notebook demonstrates running the model on user-supplied recordings.

Training and Prediction with XGBoost By Google
Public Notebook Other data XGBoost Training Prediction
This notebook uses the Census Income Data Set to demonstrate how to train a model and generate local predictions using XGBoost.

Services

Cloud Text-to-Speech By Google
Public Service Text data speech synthesis synthesize voice text-to-speech

AI Hub is a repository for ML components. Don’t reinvent the wheel! Avoid building some component when someone else has already built it, and most likely, has already optimized it. You can find and deploy not just containerized applications for machine learning, but full ML pipelines on AI hub.


Proprietary + Confidential

AI Hub stores various asset types

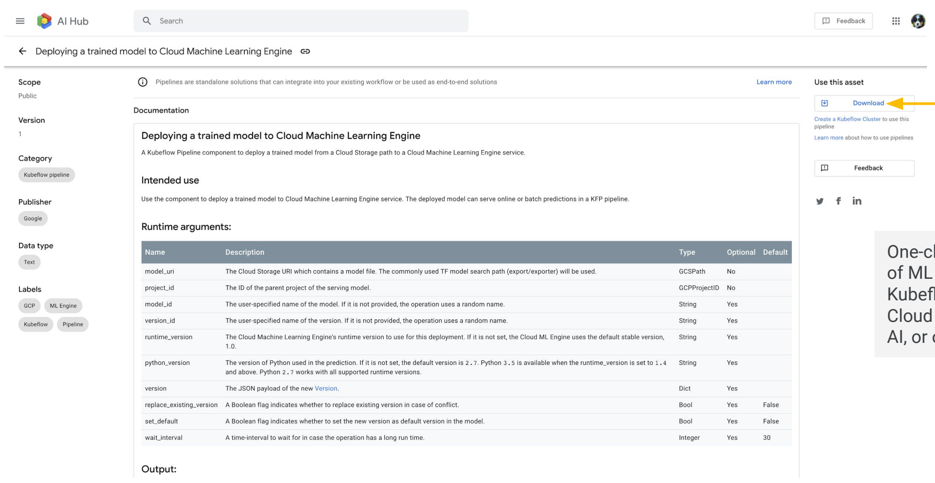
- Kubeflow pipelines and components
- Jupyter notebooks
- TensorFlow modules
- Trained models
- Services
- VM images

Google Cloud

What asset types can we find on AI Hub? Among the assets stored on AI Hub are entire Kubeflow pipelines, Jupyter notebooks, TensorFlow modules, fully trained models, services and VM images.

Proprietary + Confidential

This is what a typical asset looks like



AI Hub

Search

Feedback

Deploying a trained model to Cloud Machine Learning Engine

Scope
Public

Version
1

Category
Kubeflow pipeline

Publisher
Google

Data type
Text

Labels
GCP ML Engine Kubeflow Pipeline

Pipelines are standalone solutions that can integrate into your existing workflow or be used as end-to-end solutions Learn more

Documentation

Deploying a trained model to Cloud Machine Learning Engine

A Kubeflow Pipeline component to deploy a trained model from a Cloud Storage path to a Cloud Machine Learning Engine service.

Intended use

Use the component to deploy a trained model to Cloud Machine Learning Engine service. The deployed model can serve online or batch predictions in a KFP pipeline.

Runtime arguments:

Name	Description	Type	Optional	Default
model_uri	The Cloud Storage URI which contains a model file. The commonly used TF model search path (export/exporter) will be used.	GCSPath	No	
project_id	The ID of the parent project of the serving model.	GCPProjectID	No	
model_id	The user-specified name of the model. If it is not provided, the operation uses a random name.	String	Yes	
version_id	The user-specified name of the version. If it is not provided, the operation uses a random name.	String	Yes	
runtime_version	The Cloud Machine Learning Engine's runtime version to use for this deployment. If it is not set, the Cloud ML Engine uses the default stable version, 1.0.	String	Yes	
python_version	The version of Python used in the prediction. If it is not set, the default version is `2.7`. Python `3.5` is available when the runtime_version is set to `1.4` and above. Python `2.7` works with all supported runtime versions.	String	Yes	
version	The JSON payload of the new Version.	Dict	Yes	
replace_existing_version	A Boolean flag indicates whether to replace existing version in case of conflict.	Bool	Yes	False
set_default	A Boolean flag indicates whether to set the new version as default version in the model.	Bool	Yes	False
wait_interval	A time-interval to wait for in case the operation has a long run time.	Integer	Yes	30

Output:

Use this asset

Download

Create a Kubeflow Cluster to use this pipeline

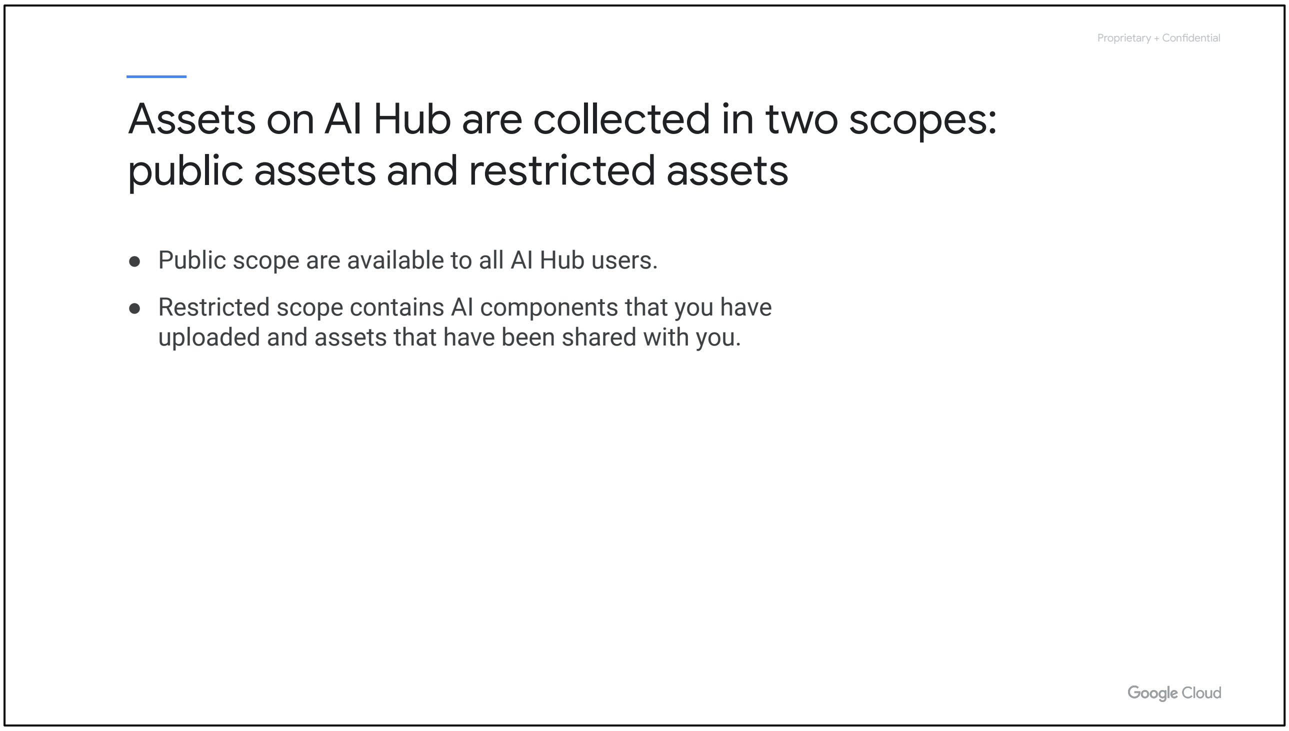
Learn more about how to use pipelines

Feedback

One-click deployment of ML pipelines via Kubeflow on Google Cloud as platform for AI, or on premise.

Google Cloud

Here you see what a typical asset looks like. You can see information about the pipeline, such as inputs and outputs, and download options.


Proprietary + Confidential

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.

Google Cloud

The assets on AI Hub are collected into two scopes: public assets and restricted assets. Public assets are available to all AI Hub users. Restricted scope assets contains AI components you have uploaded and those that have been shared with you. For example, you could have assets only available to people within your organization or teams.

Proprietary + Confidential

Agenda

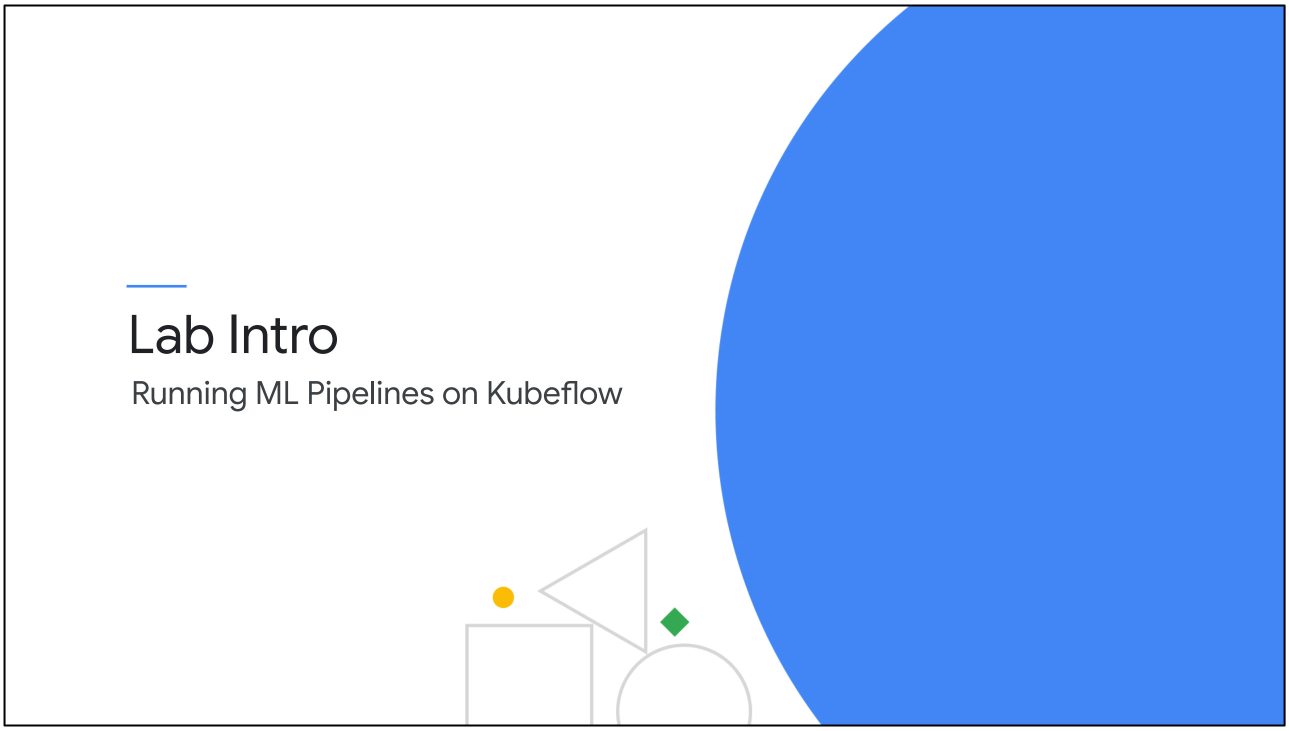
Ways to do ML on Google Cloud

Kubeflow

AI Hub

Lab: Running ML Pipelines on Kubeflow

Google Cloud



Lab Intro

Running ML Pipelines on Kubeflow

In this lab you learn how to install and use Kubeflow Pipelines. The objectives of the lab are for you to:

- 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.

Proprietary + Confidential

Module summary

- Use ML on Google Cloud using either:
    - Vertex AI (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.