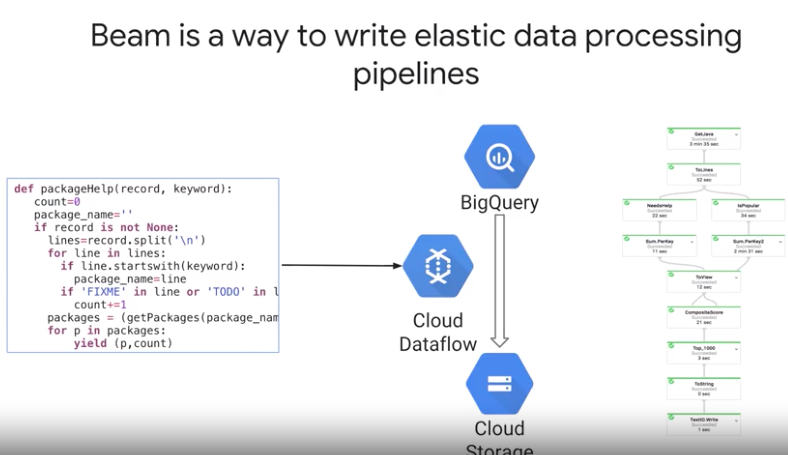
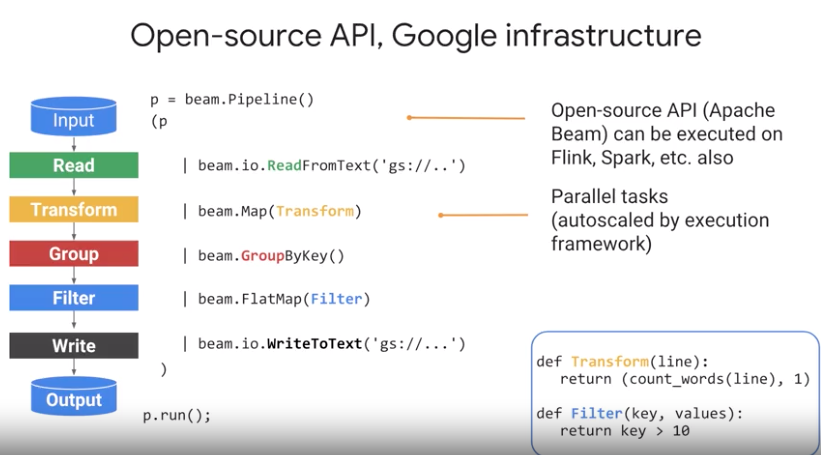
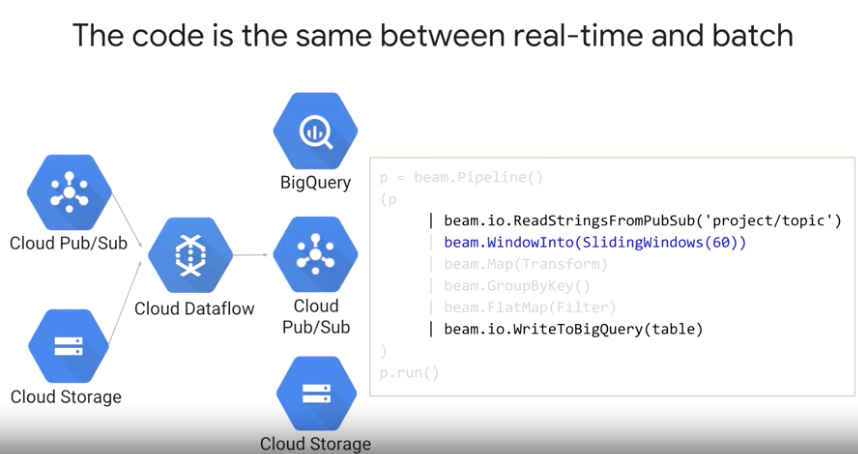
## Apache Beam

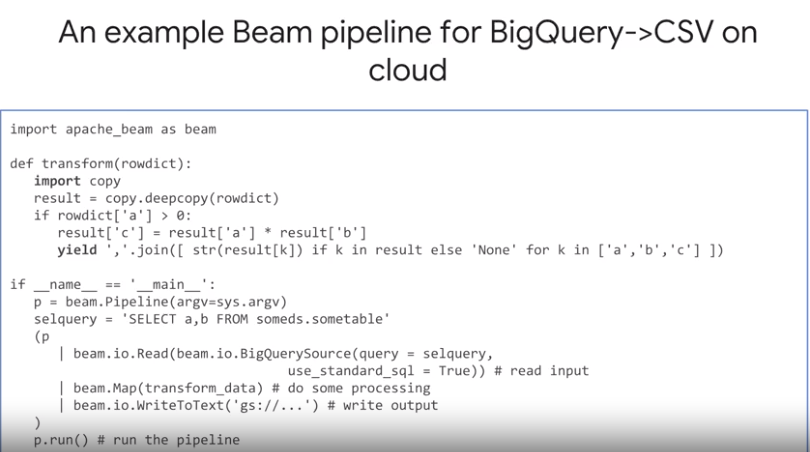
creating our data processing pipeline, training our model and ML engine, which will subsequently allow us to easily serve our model to end-users via a rest API. Then, finally to deploy an app, an app engine that will allow end-users to neatly consume our predictions

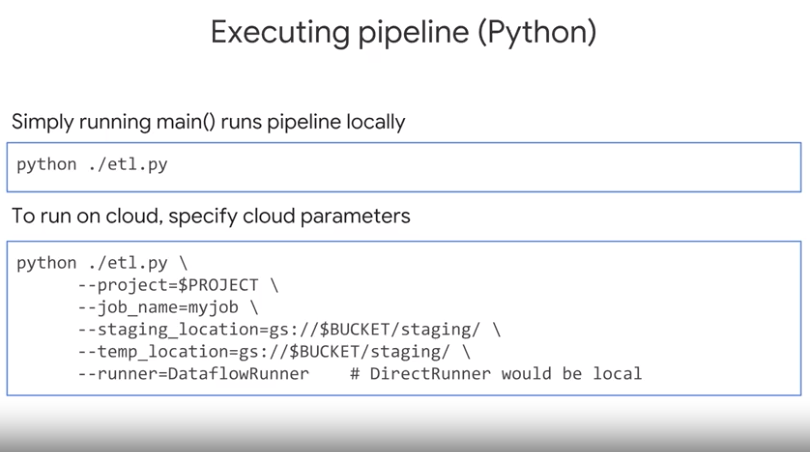
Cloud Dataflow are: it allows us to process and transform large amounts of data in parallel and it supports both streaming and batch jobs. Firstly, there's Apache Beam which is a unified model for defining both batch and streaming data parallel processing pipelines, as well as a set of language specific SDKs for constructing pipelines and runners for executing them on distributed processing backends. Then, there's Dataflow which executes the code you wrote using the Apache Beam API.

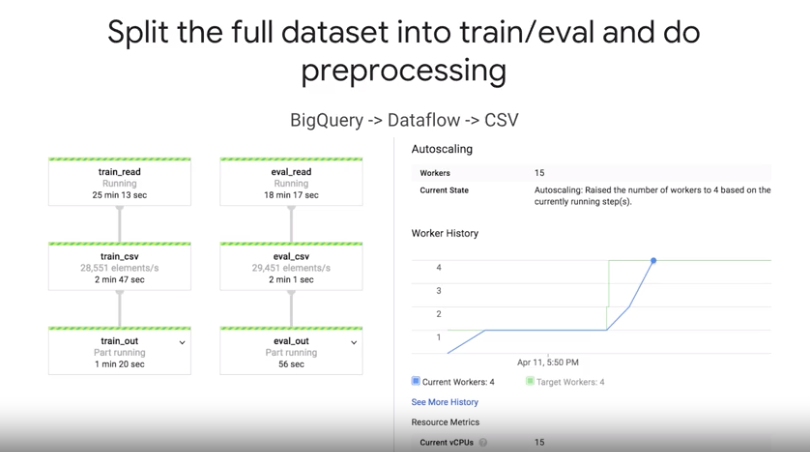


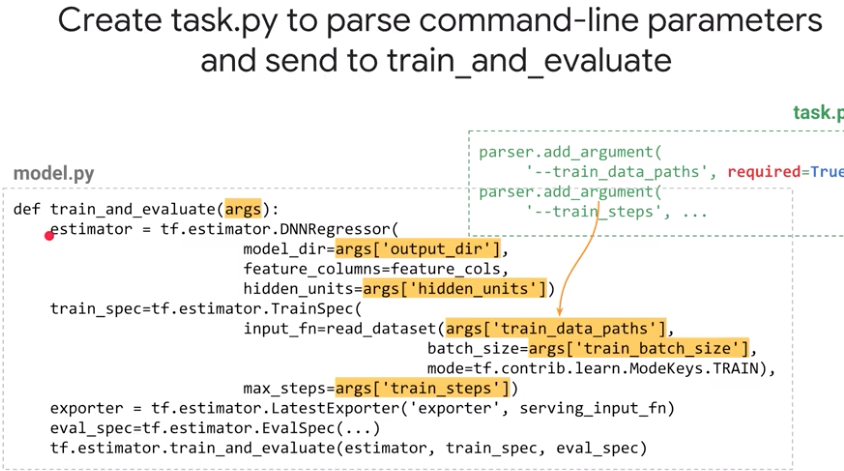


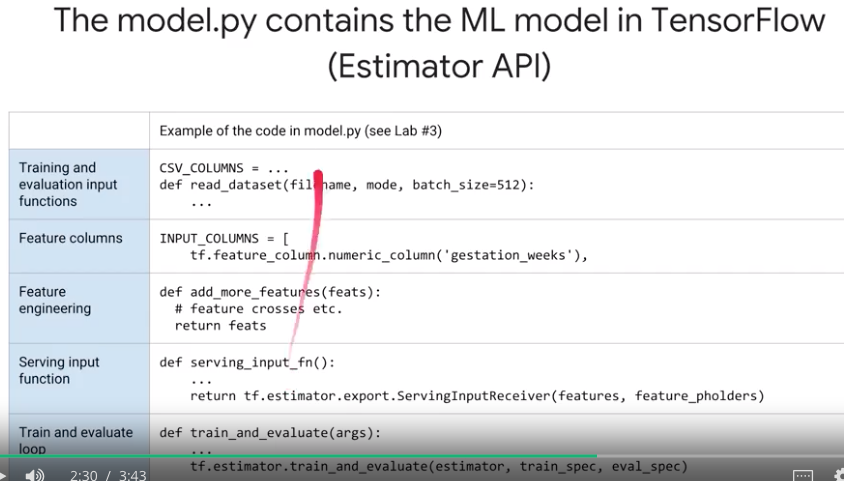


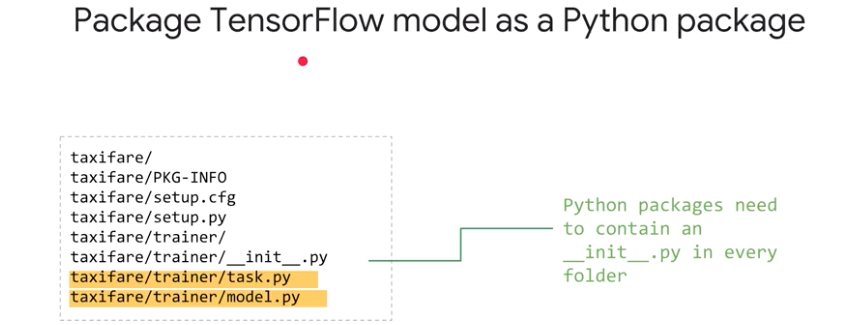


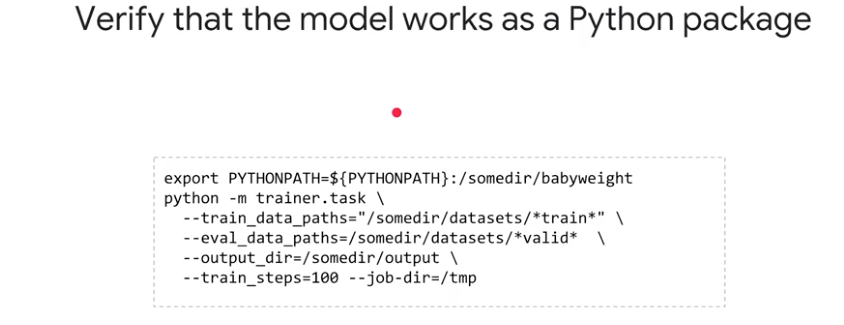


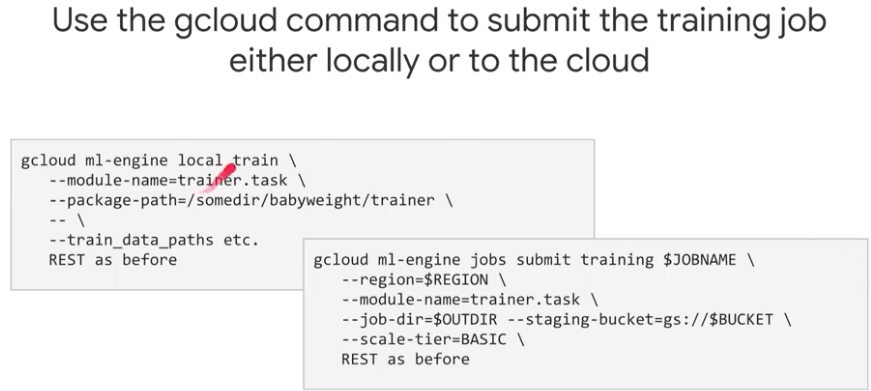


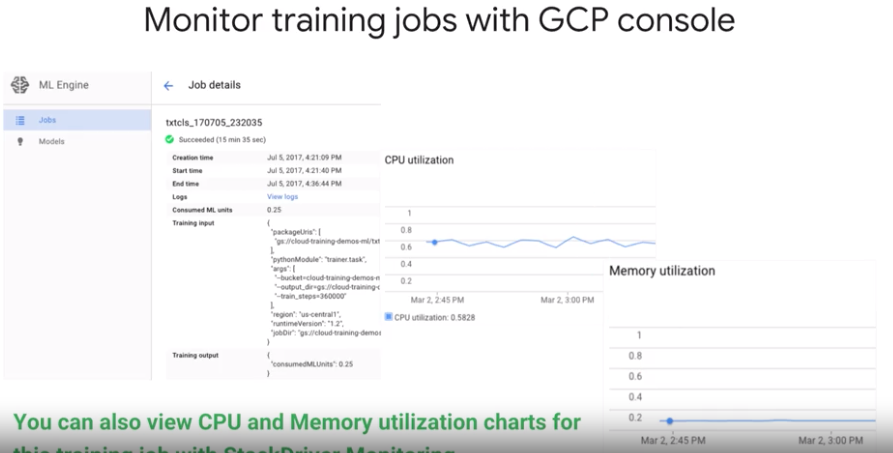




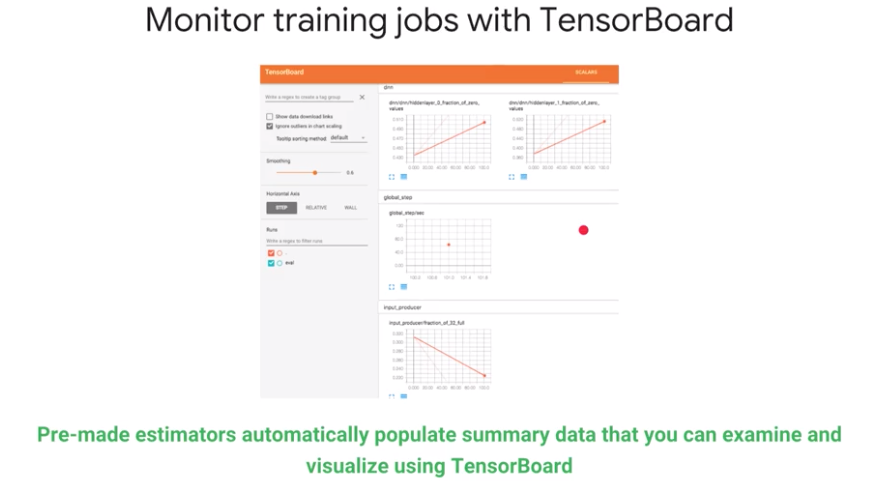


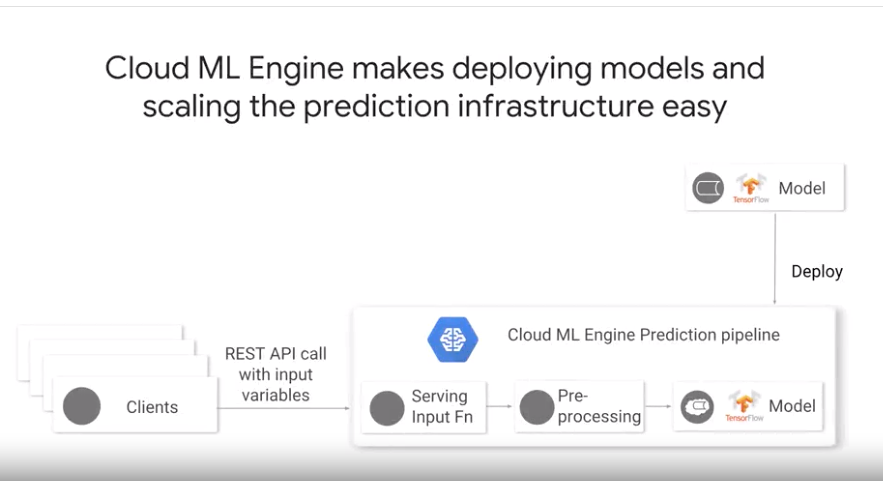


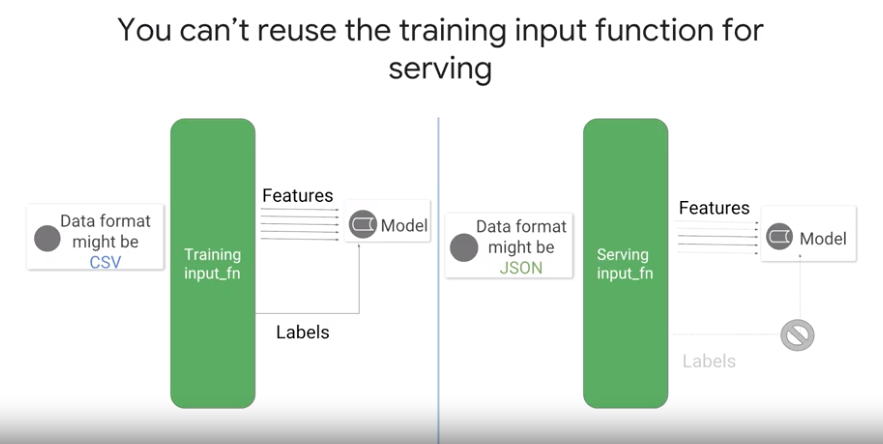


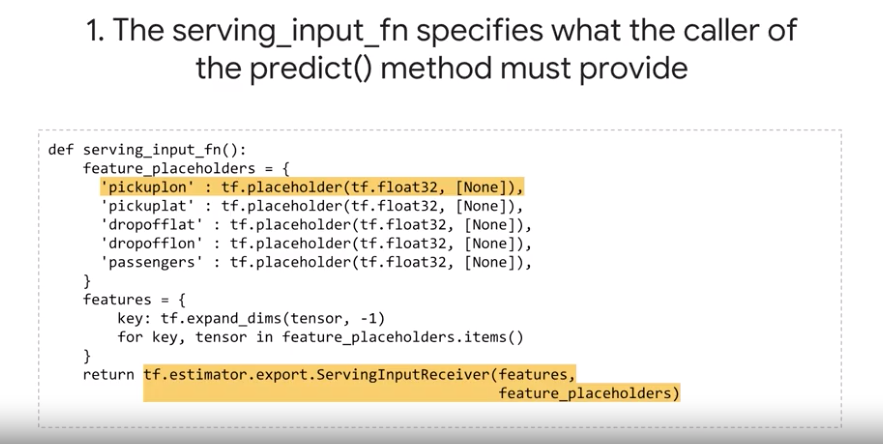


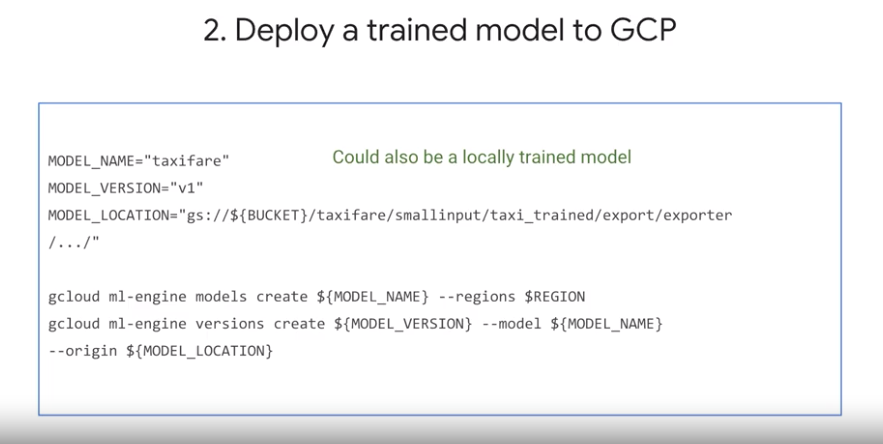
Machine Learning performance using Tensor Board.

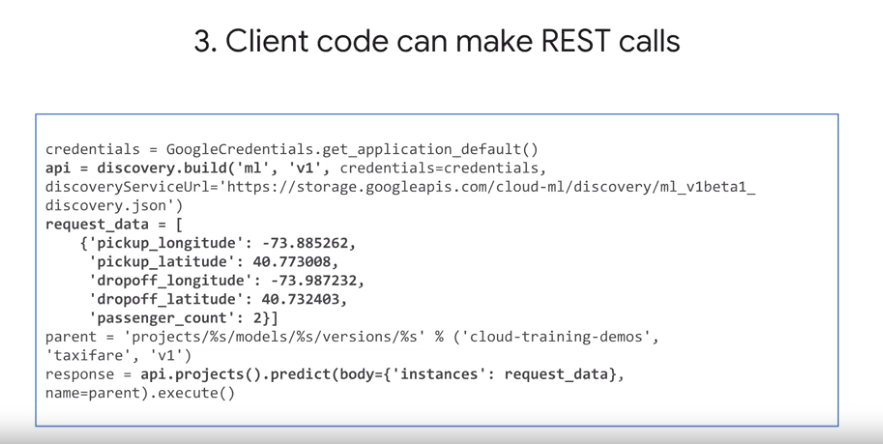












**Introduction**

[Kubeflow](https://www.kubeflow.org/) is a machine learning toolkit for [Kubernetes](https://kubernetes.io/). The project is dedicated to making **deployments** of machine learning (ML) workflows on Kubernetes simple, portable, and scalable. The goal is to provide a straightforward way to deploy best-of-breed open-source systems for ML to diverse infrastructures.

A Kubeflow deployment is:

* **Portable** - Works on any Kubernetes cluster, whether it lives on Google Cloud Platform (GCP), on-premise, or across providers.
* **Scalable** - Can utilize fluctuating resources and is only constrained by the number of resources allocated to the Kubernetes cluster.
* **Composable** - Enhanced with service workers to work offline or on low-quality networks

Kubeflow will let you organize loosely-coupled microservices as a single unit and deploy them to a variety of locations, whether that's a laptop or the cloud. This codelab will walk you through creating your own Kubeflow deployment.

What you'll build

In this lab, you're going to build a web app that summarizes GitHub issues using a trained model. Upon completion, your infrastructure will contain:

* A GKE cluster with standard Kubeflow and Seldon Core installations
* A training job that uses Tensorflow to generate a Keras model
* A serving container that provides predictions
* A UI that uses the trained model to provide summarizations for GitHub issues

What you'll learn

* How to install [Kubeflow](https://github.com/kubeflow/kubeflow)
* How to run training using the [Tensorflow](https://www.tensorflow.org/) job server to generate a [Keras](https://keras.io/) model
* How to serve a trained model with [Seldon Core](https://github.com/SeldonIO/seldon-core)
* How to generate and use predictions from a trained model