Vertex Pipelines

1. Product Overview

Vertex AI Platform Pipelines makes it easy to build cost-effective, scalable, auditable Machine Learning (ML) Pipelines.

Vertex AI Platform Pipelines - Managed (Managed Pipelines) makes it easier for you to run your ML Pipelines in a scalable and cost-effective way, while offering you 'no lock-in' flexibility. You can build your pipelines in Python using the following:

- The Kubeflow Pipelines SDK, which you can use to run ML pipelines on Kubeflow Pipelines (KFP), an open source project.
- TensorFlow Extended (TFX), the same technology that Google uses to build its ML systems.

You can then execute your pipelines in a serverless manner on Google Cloud. Managed Pipelines adds and removes resources as your pipeline executes so you don't have to worry about scale and only pay for what you use. You can also take those pipelines, authored in TFX or KFP, and execute them on-premises using Kubeflow Pipelines. Or you can run these pipelines on Google Cloud using Kubernetes Engine and AI Platform Pipelines - Hosted. On Managed Pipelines, the metadata from each pipeline run is logged to a centralized ML Metadata store which is queryable for analysis and audit.

2. Introduction to Managed Pipelines

Managed Pipelines helps you manage and run your machine learning workflows. Building Machine Learning models require complex multi-step workflows. When building a model, you may have to clean and transform data, create features, train multiple models, and evaluate those models. Managing and executing these workflows can be difficult -- especially if you want to ensure that they are run in a reproducible, auditable, cost-effective, and scalable way.

Vertex AI Platform Pipelines helps you solve this problem by letting you define your workflows as a *Kubeflow pipeline* or *TensorFlow Extended* (TFX) pipelines and managing the execution of that pipeline for you. Kubeflow Pipelines is an open source ML orchestration platform that enables reusable end-to-end ML workflow on Kubernetes. TensorFlow Extended is an ML pipeline authoring framework that draws upon years of Google's experiences building some of the most sophisticated ML systems. While it's tightly integrated with TensorFlow, it also lets you build pipelines using any ML framework.

2.1 What is an ML Pipeline?

A pipeline is a way of modeling a workflow as a set of connected steps called *components*. Each component takes as inputs the outputs of previous steps, performs some additional computations, and produces outputs that can be utilized by future components. For example, a simple ML pipeline might do the following:

1. Load a dataset from a comma-separated value file.
2. Analyze the dataset to identify and remove outliers.
3. Split the cleaned dataset into a training and evaluation dataset.
4. Train a model on the training dataset.
5. Evaluate the model against the evaluation dataset.

In this example, each step (except for the first one) takes as input the output of a previous step; and produces an output that can be utilized by a subsequent step.

The idea of a pipeline is a common one in the world of data processing and machine learning. Most Data Scientists begin an ML project with a single Jupyter notebook or Python script that carries out an end-to-end workflow. However, as these projects mature, many data scientists find that it's easier to manage the workflow if it is broken up into smaller independent steps with well defined interfaces. Each workflow step can be written, tested, packaged, and executed separately; further these steps can be shared and reused in other pipelines. AI Pipelines are fundamentally about modularity: individual components that do one thing well.

2.2 Why should I use Managed Pipelines?

Using Managed Pipelines to execute an ML workflow makes it easier to:

1. Collaborate and reuse code
2. Automate the training and deployment of models
3. Keep track of how models were produced
4. Optimize the use of compute resources

Pipelines make it simpler for Data Scientists to collaborate and reuse code. Since pipelines consist of individual components, teams can build and share the code for individual components. KFP and TFX make it easy to build your own components and come with a set of pre-built pipelines, components, and libraries. With TensorFlow Extended (TFX), you can use the same code used by Google to build its ML pipelines; or where your problems differ you can build your own components and share them.

When ML models are being used consistently for critical business problems, auditability and traceability become key concerns. AI Platform Pipelines helps you with built-in artifact tracking— letting you see which specific execution of a pipeline produced a particular model.

Finally, Managed Pipelines ensures that your pipeline is executed in a scalable and cost-effective manner. Managed Pipelines service allocates compute resources as you need them. You only pay for the resources you use.

2.3 Which Google AI Pipelines technology should I use?

Google Cloud has a suite of products that are designed to help you build and execute ML workflows in a reproducible, scalable, auditable, and cost-effective way.

- Kubeflow Pipelines (KFP): An open source platform that lets you run ML pipelines built using either Kubeflow SDK or TensorFlow Extended and execute those pipelines on any Kubernetes cluster. GCP offers an assisted deployment of the KFP OSS stack through Marketplace, using AI Platform Pipelines Hosted.

- Managed AI Platform Pipelines: A fully managed serverless platform in GCP for executing ML Pipelines built using KFP SDK or TensorFlow Extended.

Managed Pipelines and Kubeflow Pipelines are distinct products. They were built with different goals in mind, on different stacks.

From a user perspective, the primary differences are:
- Managed Pipelines relies on Cloud Storage, rather than persistent volumes to handle all of your pipelines storage needs.
- Managed Pipelines does not support Kubernetes style secrets management

- Kubeflow Pipelines is a better fit for those :
  - Who have a hybrid environment where pipelines need to execute on-premises as well in the Cloud
  - Who have standardized their infrastructure on Kubeflow Pipelines or Kubernetes.
  - Who need to control and manage Kubernetes resources to achieve a specific goal not currently available using Managed Pipelines.

- AI Platform Pipelines (Hosted) is a better fit for those :
  - Who prefer Kubeflow Pipelines but want to run it on GKE.

2.4 How do I use Managed Pipelines ?

There are some tools that you need to be familiar with to use Managed Pipelines:

- Python-based SDK to author ML Pipelines. There are currently two supported SDK options:
  - **KFP SDK**
  - **TFX SDK**
- **Google Cloud Console** - Google Cloud Console provides you with a rich user interface for:
  a. Submitting new pipelines that have been authored and compiled with TFX.
  b. Creating new runs of existing pipelines

c. Monitoring the execution of pipeline runs

1. Build your pipelines using KFP SDK or TFX SDK;
2. Compile your pipeline into a JSON file ;
3. Use the provided Client SDK or Google Cloud Console to upload and run the pipelines; and
4. Use the Google Cloud Console to monitor the execution of your pipeline.

In the how-to section below, we walk you through each of these activities.

2.5 Which SDK should I use?

We recommend using the following criteria to decide which SDK to use:

- If you are an existing Kubeflow Pipelines user, use KFP SDK to author pipelines for Managed Pipelines
- If you are an existing TFX user, you can stick with TFX SDK to author your pipelines
- If you are a new user, we recommend you get started with the intro KFP notebook below.

3 Concepts

There are several key concepts that we refer to in this user guide. In this section, we provide brief descriptions of these concepts. These concepts are common across both KFP and TFX. You can read more about KFP Pipelines and TFX Pipelines using the following links:

- KFP Pipelines Documentation
- TFX Pipelines Documentation

3.1 Artifacts and Metadata

The outputs of steps in a pipeline are called **artifacts.** Subsequent steps in your ML workflow may use these artifacts as inputs. Artifacts must have associated metadata, which defines the type and properties of the artifact.

Managed Pipelines automatically stores information about your pipeline's artifacts in a queryable Managed Metadata store which can be used to review, analyze, debug and audit your pipeline runs. Please see the Managed Metadata User Guide for an in-depth description of the service, which includes an SDK for recording and retrieving ML Metadata from both pipelines and notebooks.

You can learn more about how ML Metadata is used in pipelines in the public documentation.

3.2 Parameters

Parameters let you change the behavior of the code through configuration instead of code. Parameter values can be specified when you create the pipeline using SDK or Google Cloud Console. Parameter values are stored as Execution properties that can be later queried.

You can learn more about parameters in TFX and KFP documentations.

3.3 Components

Pipeline components are self-contained sets of code that perform one step in a pipeline's workflow, such as data preprocessing, data transformation, model training, etc. Components are composed of a set of input and output artifacts, a set of parameters, and the code to implement the component's functionality.

3.3.1 KFP Components

KFP allows users to containerize any program and use it as a component in their ML Pipeline. KFP components can be authored from pre-existing containers, or by writing simple Python functions that the KFP SDK will containerize for the user. For further information on how to create KFP components, please refer to the open-source documentation.

Also see this section for current limitations of as yet unsupported KFP features.

3.3.2 TFX Components

TFX provides several standard components that you can use in your pipelines. If these components do not meet your needs, you can build custom components. Learn more about TFX Components and TFX Pipelines.

TFX ships with a rich set of components that cover common steps in machine learning workflows:

- ExampleGen -- Ingest and optionally split the input datasets into a training and validation set.
- StatisticsGen - Calculates statistics for the dataset
- SchemaGen - Creates a data schema
- ExampleValidator - looks for anomalies and missing values
- Transform - performs feature engineering
- Trainer - Trains the model
- Evaluator -analyzes the trained model to determine the quality of your model
- Pusher - deploys the model on a serving infrastructure.

You can learn more about the components that ship with TFX in the TFX Documentation.

If you do not use TensorFlow, you can still use TFX to build your pipelines. The simplest approach is to use TFX's Container-based Component feature which lets you utilize any container as a component in a TFX pipeline.

3.4 Pipeline

A pipeline is a definition of an ML workflow where each step in the workflow is executed by a *component*. Each component within a pipeline can use *artifacts or parameters* produced by other components as part of its inputs.

Managed Pipelines determine the execution sequence of a pipeline's components, in part, by creating a directed acyclic graph of the artifact dependencies. You can also define task-based dependencies between components. With a task-based dependency, you indicate that while component **A** does not depend on the output of component **B**, component **A** must be executed after component **B**. Artifact and parameter-based dependencies are recommended for most cases since they help to track the lineage of artifacts that the pipeline produces.

In addition to components. Pipelines may have a set of *parameters,* such as: dataset location, training hyper parameters, or output locations. Pipelines can be run multiple times with different input parameters. You may want to re-run a particular pipeline on a different dataset; or with different hyperparameters.

When you run a pipeline, the system executes each step of the pipeline in the most efficient way possible. Currently, most components are executed as containers on AI Platform Training.