

Creating a machine learning pipeline

Product or feature is covered by the [Pre-GA Offerings Terms \(/terms/service-terms#1\)](#) of the Google Cloud Platform of Service. Pre-GA products and features may have limited support, and changes to pre-GA products and features may not be compatible with other pre-GA versions. For more information, see the [launch stage descriptions \(/docs/ai-platform-pipelines#product-launch-stages\)](#).

Before you can run your machine learning (ML) process on AI Platform Pipelines, you must first define your process as a pipeline. You can orchestrate your ML process as a pipeline using TensorFlow Extended (TFX) or the Kubeflow Pipelines SDK.

This document provides guidance for choosing the best option for building your pipeline, and resources for getting started.

- If you are orchestrating a process that trains a TensorFlow model, [use TFX to build your pipeline \(#tfx-pipeline\)](#).
- If you are orchestrating a process that trains a model using frameworks such as PyTorch, XGBoost, and scikit-learn, [use the Kubeflow Pipelines SDK to build your pipeline \(#kfp-sdk\)](#).

Building pipelines using the TFX SDK

TFX is an open source project that you can use to define your ML workflow as a pipeline. Currently, TFX components can only train TensorFlow based models. TFX provides components that you can use to ingest and transform data, train and evaluate a model, deploy a trained model for inference, etc. By using the TFX SDK, you can compose a pipeline for your ML process from TFX components.

To get started building pipelines with TFX pipeline templates:

- Follow the tutorial about [TFX pipelines on Google Cloud](#) (<https://www.tensorflow.org/tfx/tutorials/tfx/cloud-ai-platform-pipelines>).
- Read the [TFX User Guide](#) (<https://www.tensorflow.org/tfx/guide>) to learn more about TFX concepts and components.

Building pipelines using the Kubeflow Pipelines SDK

The Kubeflow Pipelines SDK is an open source SDK that you can use to build complex custom ML pipelines based on containers. You can reuse pre-built components or build custom pipeline components using the Kubeflow Pipelines SDK. At a high level, you build components and pipelines by:

1. Developing the code for each step in your workflow using your preferred language and tools
2. Creating a Docker container image for each step's code
3. Using Python to define your pipeline using the Kubeflow Pipelines SDK

To get started building pipelines with the Kubeflow Pipelines SDK:

- Read the [Introduction to the Kubeflow Pipelines SDK](https://www.kubeflow.org/docs/pipelines/sdk/sdk-overview/) (<https://www.kubeflow.org/docs/pipelines/sdk/sdk-overview/>).
- Learn more about Kubeflow pipelines by [exploring the Kubeflow Pipelines samples](https://www.kubeflow.org/docs/pipelines/tutorials/build-pipeline/) (<https://www.kubeflow.org/docs/pipelines/tutorials/build-pipeline/>).
- Reuse pre-built components by [exploring the Kubeflow pipeline components on GitHub](https://github.com/kubeflow/pipelines/tree/master/components) (<https://github.com/kubeflow/pipelines/tree/master/components>).
- Search for [pre-built components and pipelines on AI Hub](https://aihub.cloud.google.com/u/0/s?category=pipeline) (<https://aihub.cloud.google.com/u/0/s?category=pipeline>).

What's next

- Learn how to [run your ML pipelines](/ai-platform/pipelines/docs/run-pipeline/) (/ai-platform/pipelines/docs/run-pipeline).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-06-30 UTC.