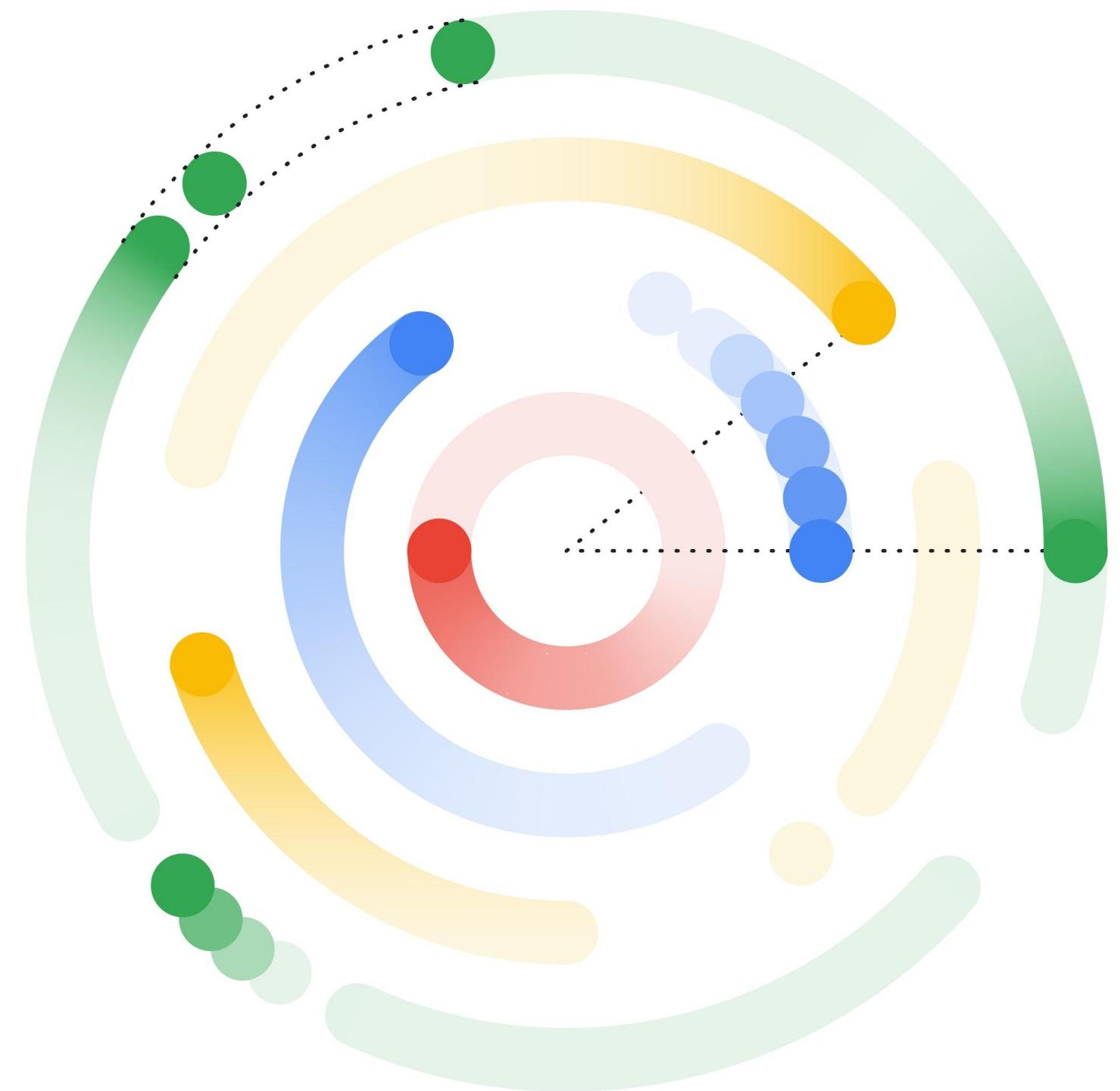
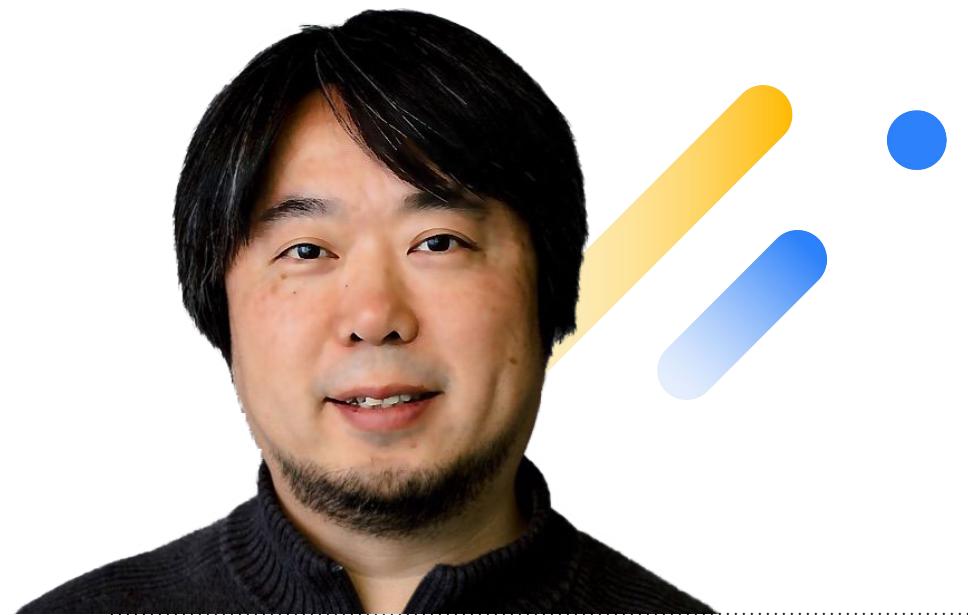


# Model Monitoring

**Google Cloud Applied ML Summit**  
Solving for the future.

06/10/21





**Kaz Sato**  
Developer Advocate  
Google Cloud



**Marc Cohen**  
Developer Advocate  
Google Cloud

# Table of Contents

**MLOps at Google**

01



**MLOps with Vertex AI**

02

**Model Monitoring**

03

**Demo**

04

01

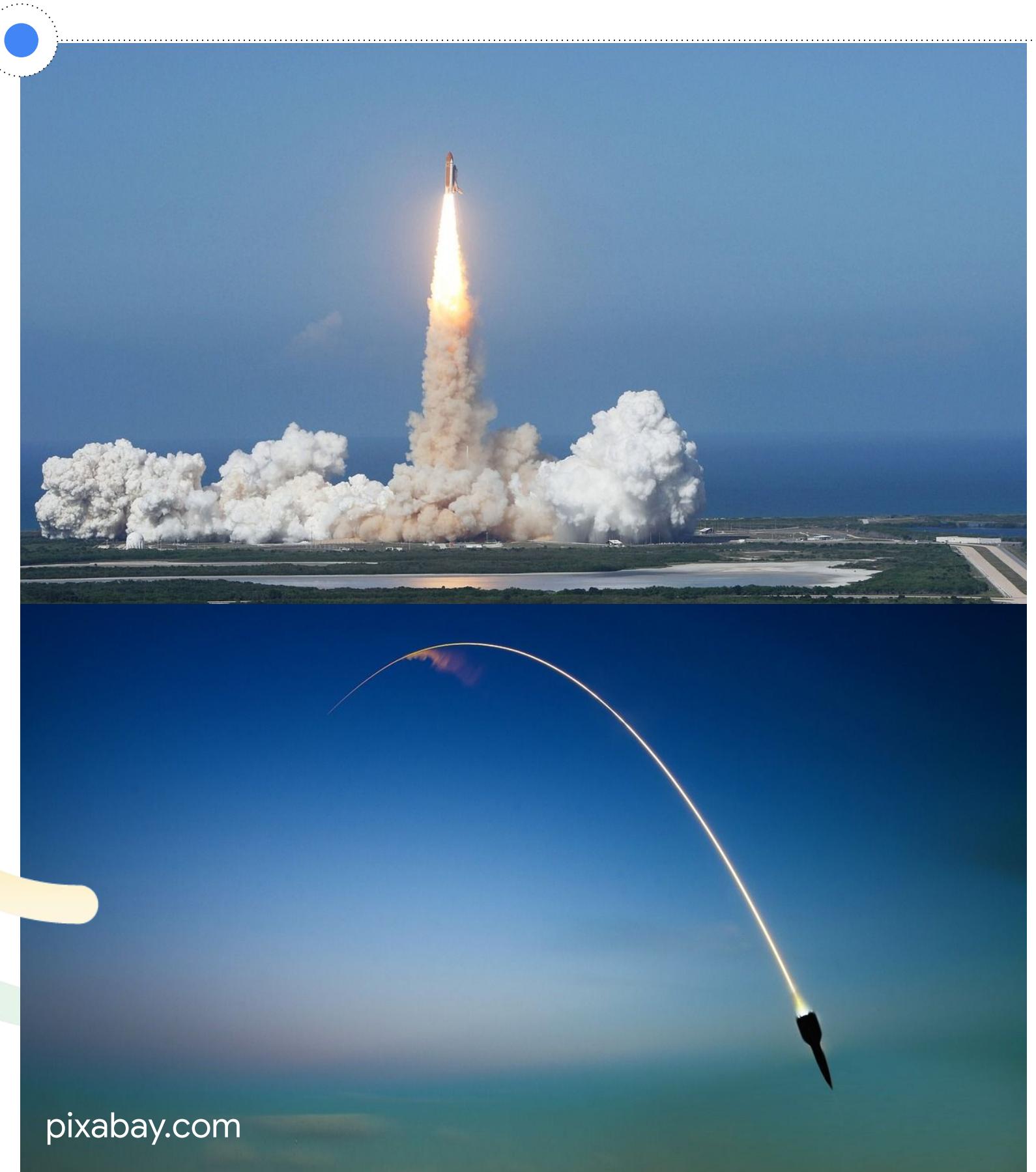
# MLOps at Google



# Launching is easy, Operating is hard.

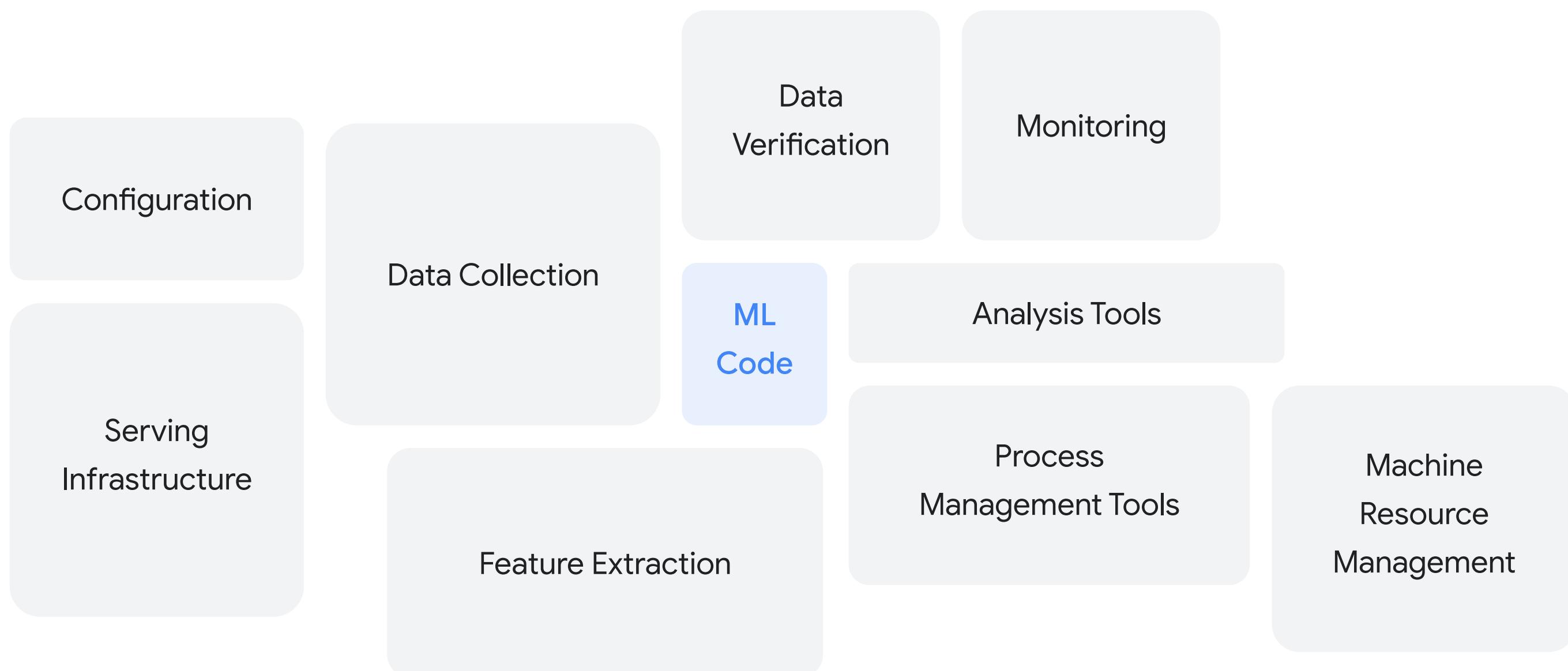
“

The real problems with a  
ML system will be found  
while you are continuously  
operating it for the long term”

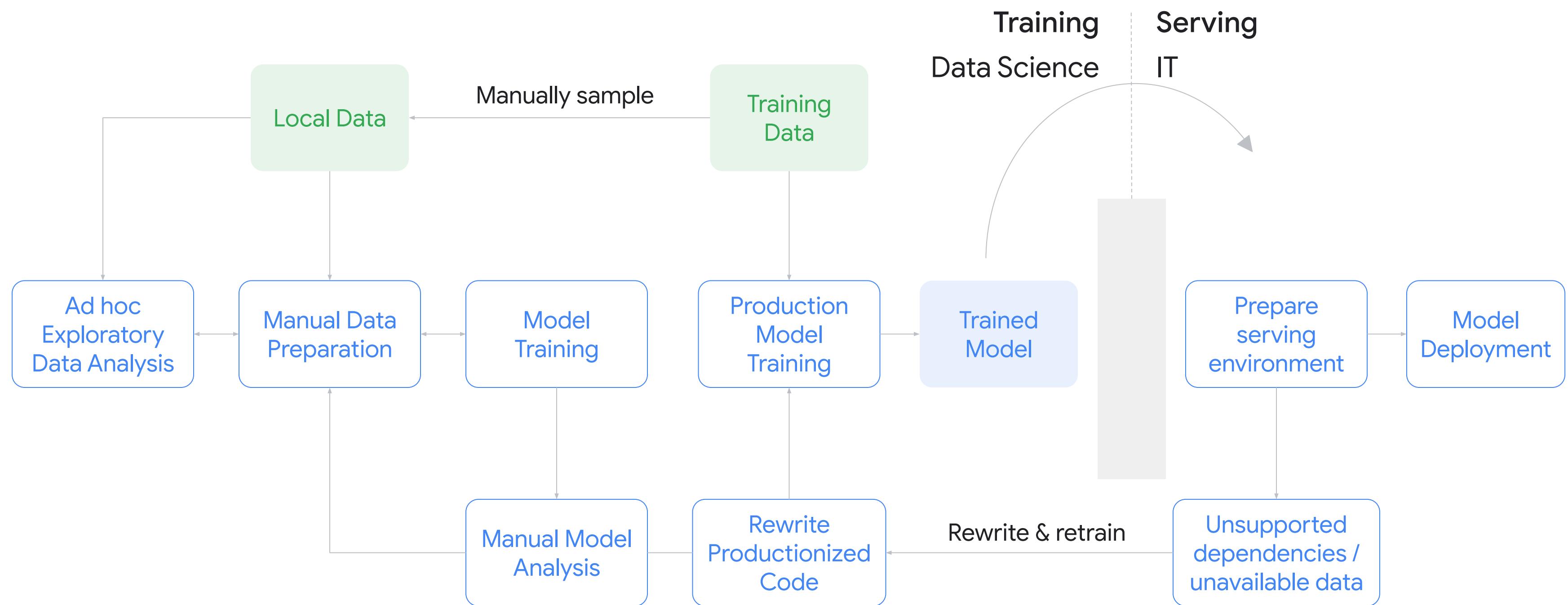


[pixabay.com](https://pixabay.com)

Google Cloud



# What's happening today: Data Science and IT (Ops) are isolated



# MLOps

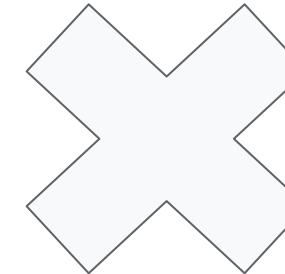
---

An ML engineering culture and practice that  
aims at **unifying** ML system development  
(Dev) and ML system operation (Ops)

# The Challenges on Productionizing ML

## ML challenges

- Governance of data, features, models, pipelines and experiments
- Continuous training and deployment
- Training-serving skew
- Data validation
- Model analysis
- Fairness and Explainability



## Production system challenges

- Scalability
- Availability
- Portability
- Reproducibility
- Modularity
- Monitoring and Alerting
- Security
- Hosted or Serverless

# The History of Productionizing ML at Google

Continuous Training for Production ML in the TFX Platform. OpML (2019).

Slice Finder: Automated Data Slicing for Model Validation. ICDE (2019).

Data Validation for Machine Learning. SysML (2019).

TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. KDD (2017).

Data Management Challenges in Production Machine Learning. SIGMOD (2017).

Rules of Machine Learning: Best Practices for ML Engineering. Google AI Web (2017).

Machine Learning: The High Interest Credit Card of Technical Debt. NeurIPS (2015).

Hidden Technical Debt in Machine Learning Systems. NIPS (2015).



KDD 2017 Applied Data Science Paper

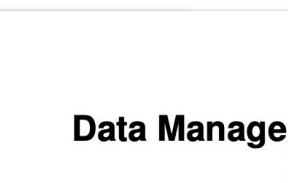
## TFX: A TensorFlow-Based Production-Scale Machine Learning Platform

Denis Baylor, Eric Breck, Heng-Tze Chen, Salem Haykal, Mustafa Ispir, Clemens Mewald, Akshay Narayan, Steven Euijong Whang, Martin Zinkevich

D. Sculley  
Todd Phillips  
{dsculley, tdp}@google.com

Machine learning systems are deployed quickly. This is as coming for free, markably easy to do when applying machine learning specifically where possible. The loops, undeclared dependencies, and lack of visibility make it difficult to understand what's happening in the system. This leads to duplicated effort and fragile system-wide technical debt.

We present TensorFlow Extended (TFX), a general-purpose machine learning platform at Google. By integrating the aforementioned components into one platform, we were able to standardize on a common set of components, simplify the platform configuration, and reduce time to production from the order of months to days, providing platform stability that minimizes deployment risk.



## Data Management Challenges in Production Machine Learning

Neoklis Polyzotis, Sudip Roy, Martin Zinkevich

{npolyzotis, srujan, zinkevich}@google.com

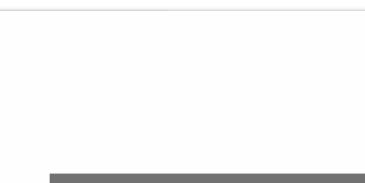
### ABSTRACT

Creating and maintaining a platform for reliably deploying machine learning models require coordination of many components—a learner for models based on training data, modules for analyzing both data as well as models, and finally for serving models in production. This becomes challenging when data changes over time and needs to be produced continuously. Unfortunately, coordination is often done ad hoc using glue code scripts developed by individual teams for specific leading to duplicated effort and fragile system-wide technical debt.

We present TensorFlow Extended (TFX), a general-purpose machine learning platform at Google. By integrating the aforementioned components into one platform, we were able to standardize on a common set of components, simplify the platform configuration, and reduce time to production from the order of months to days, providing platform stability that minimizes deployment risk.

### 1. INTRODUCTION

It is hard to overemphasize the importance of learning in modern computing. More and more organizations adopt machine learning as a tool to gain competitive advantage. However, the workflows and underlying systems for machine learning (ML) in production systems come in different forms, which makes it difficult to understand what's happening in the system. This leads to duplicated effort and fragile system-wide technical debt.



## Continuous Training for Production ML in the TFX Platform

Denis Baylor, Eric Breck, Heng-Tze Chen, Salem Haykal, Mustafa Ispir, Clemens Mewald, Akshay Narayan, Steven Euijong Whang, Martin Zinkevich

Rose Liu, Clemens Mewald, Akshay Narayan, Steven Euijong Whang, Martin Zinkevich

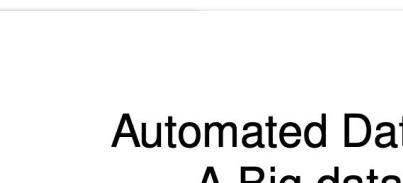
{dsculley, tdp}@google.com

### Abstract

Large organizations rely increasingly on continuous training pipelines in order to keep machine-learned models up-to-date with respect to data. In this scenario, disruptions in the pipeline can increase model staleness and degrade the quality of downstream services that depend on these models. In this paper we describe the operation of continuous pipelines in the Tensorflow Extended (TFX) that we developed and deployed at Google. We present main mechanisms in TFX to support this type of production and the lessons learned from the deployment internally at Google.

### 1 Introduction

Machine learning systems [1] are becoming prevalent thanks to a vast number of success stories. However, the data tools for interpreting aging models have not caught up yet, and many challenges exist to improve our model understanding [2]. One such key problem is to understand why a model performs poorly on certain parts of the data referred to as a *slice*.



## Automated Data Slicing for Model Validation

Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, Martin Zinkevich

{yechung, tkraska, npolyzotis, srujan, euijw, zinkevich}@google.com

**Abstract**—As machine learning systems become more complex and widely used, the need for automated data slicing has increased. However, current data tools are still primitive and lack the ability to slice data effectively. We focus on the particular problem of slicing large datasets, which is an important problem in model validation because it allows users to analyze the model performance on specific subsets of data. Our goal is to find interpretable slices that are both accurate and efficient. We propose Slice Finder, which uses a combination of machine learning and statistical methods to identify slices that are both accurate and efficient. Applications include diagnosing model failure and improving model performance. This research is part of a larger trend towards automated data slicing, which is crucial for the success of machine learning systems.

**Index Terms**—data slicing, model validation, model interpretation



## DATA VALIDATION FOR MACHINE LEARNING

Eric Breck<sup>1</sup> Neoklis Polyzotis<sup>1</sup> Sudip Roy<sup>1</sup> Steven Euijong Whang<sup>2</sup> Martin Zinkevich<sup>1</sup>

### ABSTRACT

Machine learning is a powerful tool for gleaning knowledge from massive amounts of data. While a great deal of machine learning research has focused on improving the accuracy and efficiency of training and inference algorithms, there is less attention in the equally important problem of monitoring the quality of data fed to machine learning. The importance of this problem is hard to dispute: errors in the input data can nullify any benefits on speed and accuracy for training and inference. This argument points to a data-centric approach to machine learning that treats training and serving data as an important production asset, on par with the algorithm and infrastructure used for learning.

In this paper, we tackle this problem and present a data validation system that is designed to detect anomalies specifically in data fed into machine learning pipelines. This system is deployed in production as an integral part of TFX(Baylor et al., 2017) – an end-to-end machine learning platform at Google. It is used by hundreds of product teams use it to continuously monitor and validate several petabytes of production data per day. We faced several challenges in developing our system, most notably around the ability of ML pipelines to soldier on in the face of unexpected patterns, schema-free data, or training/serving skew. We discuss these challenges, the techniques we used to address them, and the various design choices that we made in implementing the system. Finally, we present evidence from the system’s deployment in production that illustrate the tangible benefits of data validation in the context of ML: early detection of errors, model-quality wins from using better data, savings in engineering hours to debug problems, and a shift towards data-centric workflows in model development.

02



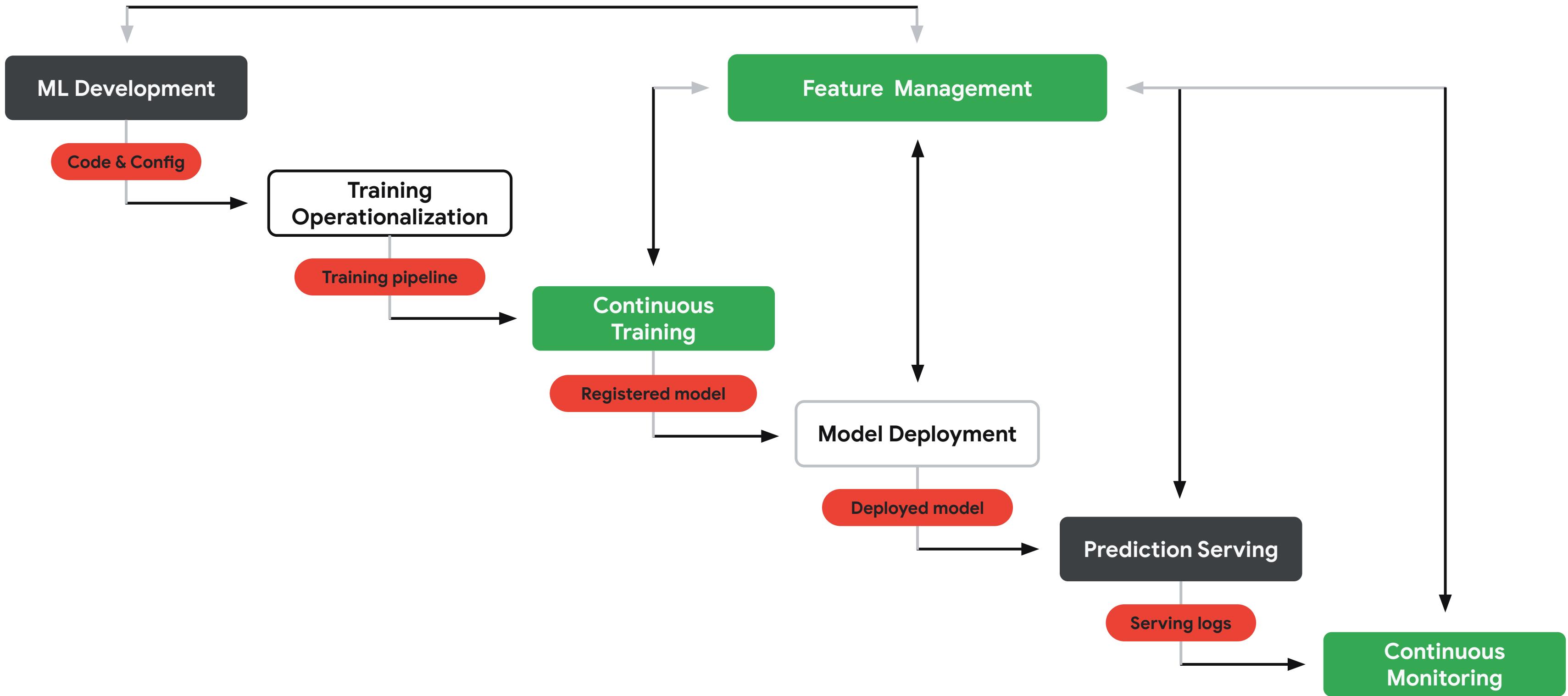
# MLOps with Vertex AI



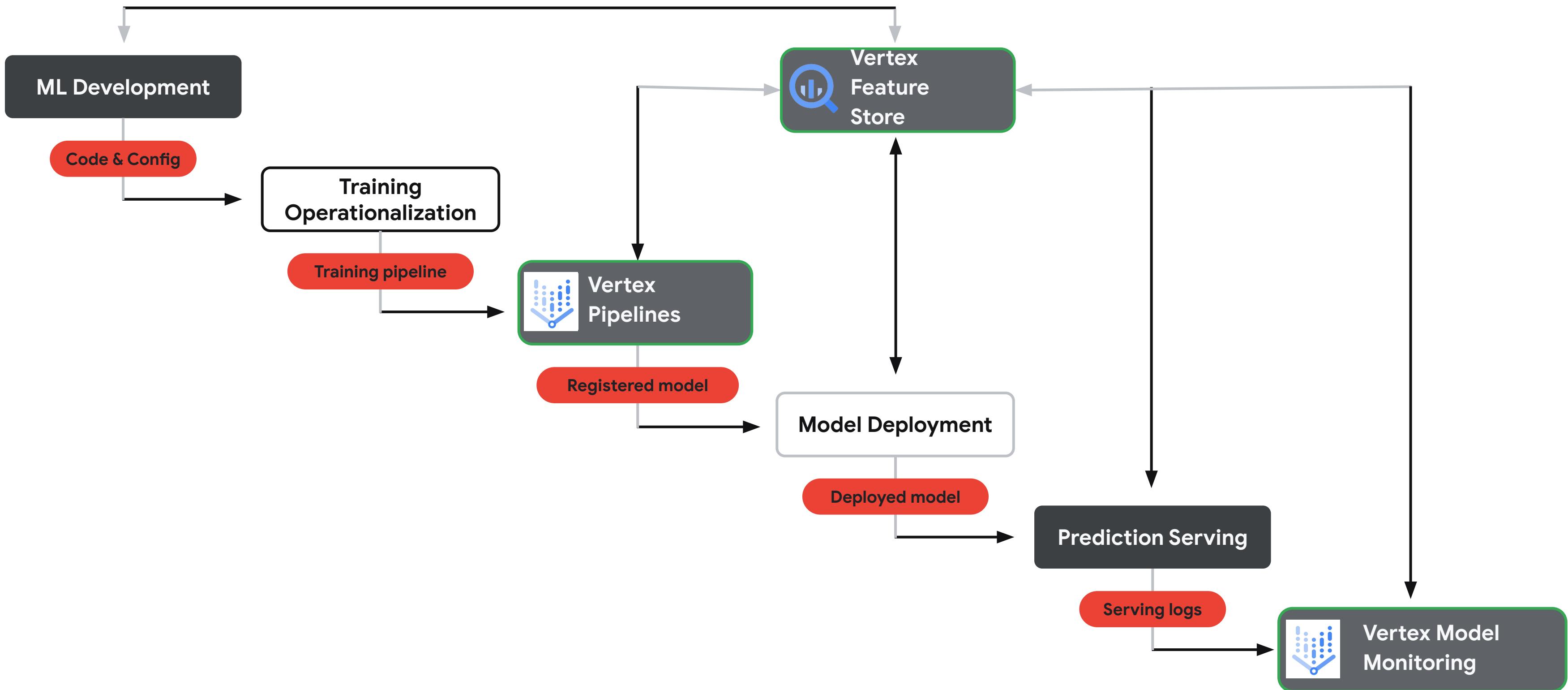
Vertex AI

Vertex AI is a **managed ML platform** for practitioners to accelerate **experiments** and **deploy** AI models.

# MLOps challenges in ML lifecycle



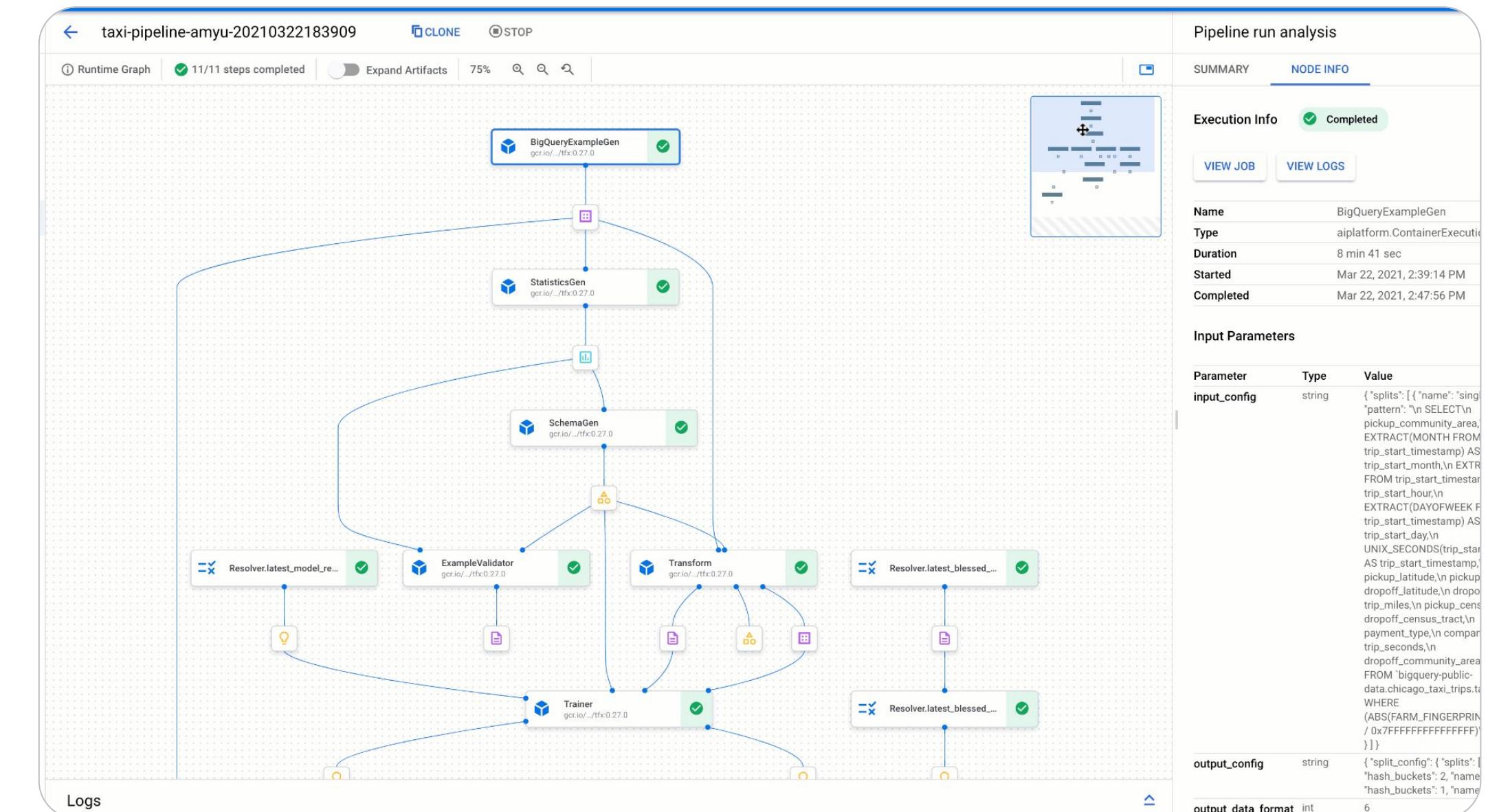
# MLOps challenges with Vertex AI



# Vertex Pipelines

Build pipelines with familiar  
open-source **Kubeflow Pipelines**

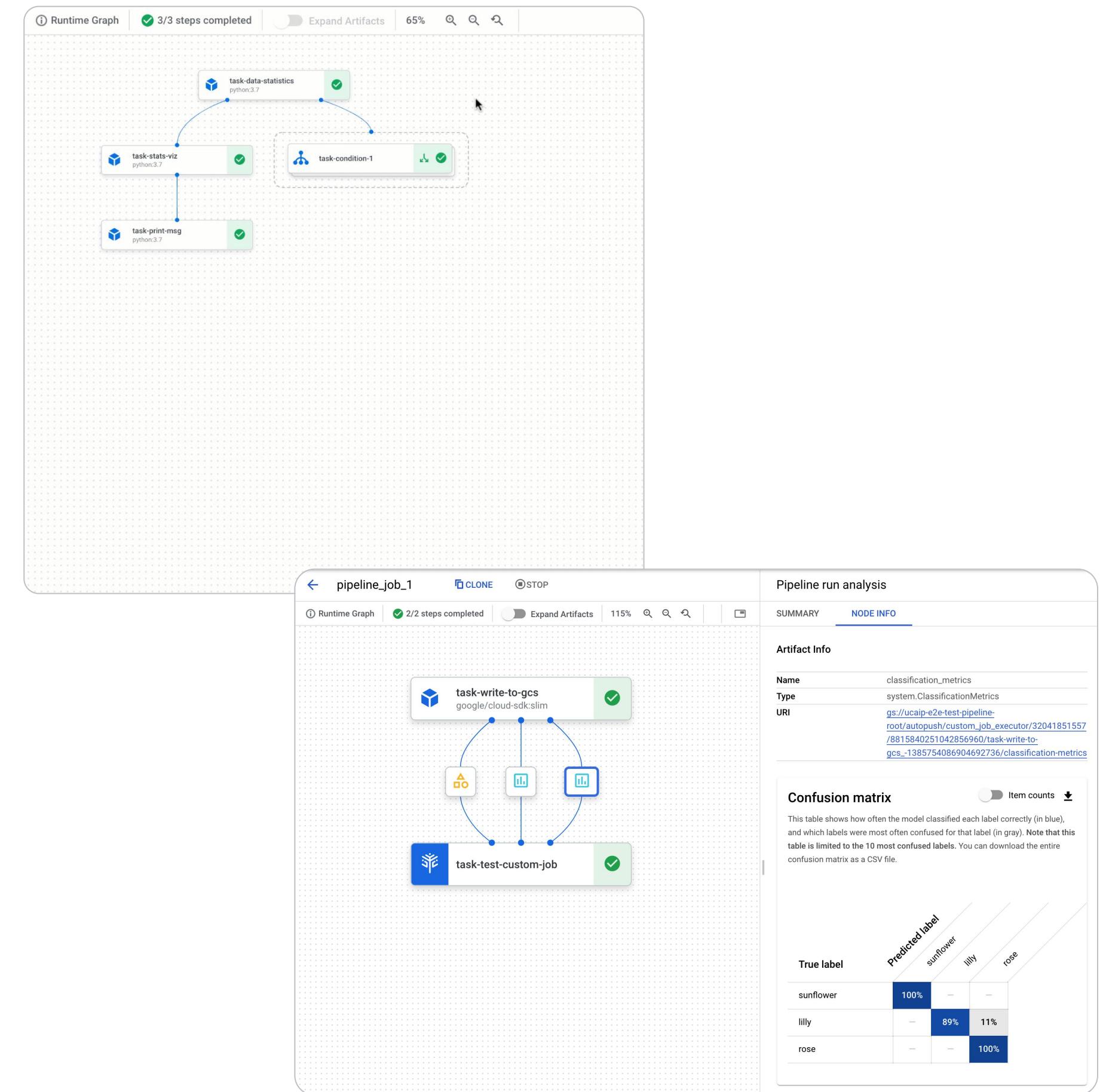
**Serverless**, automated, scalable:  
pay only for what you use



# Vertex Pipelines

Define **pipelines** with your **components** for data ingestion, preprocessing, training, validation and deployment

Track **artifacts** and **lineage** of data, feature, model and experiment metrics across your ML workflow with metadata store

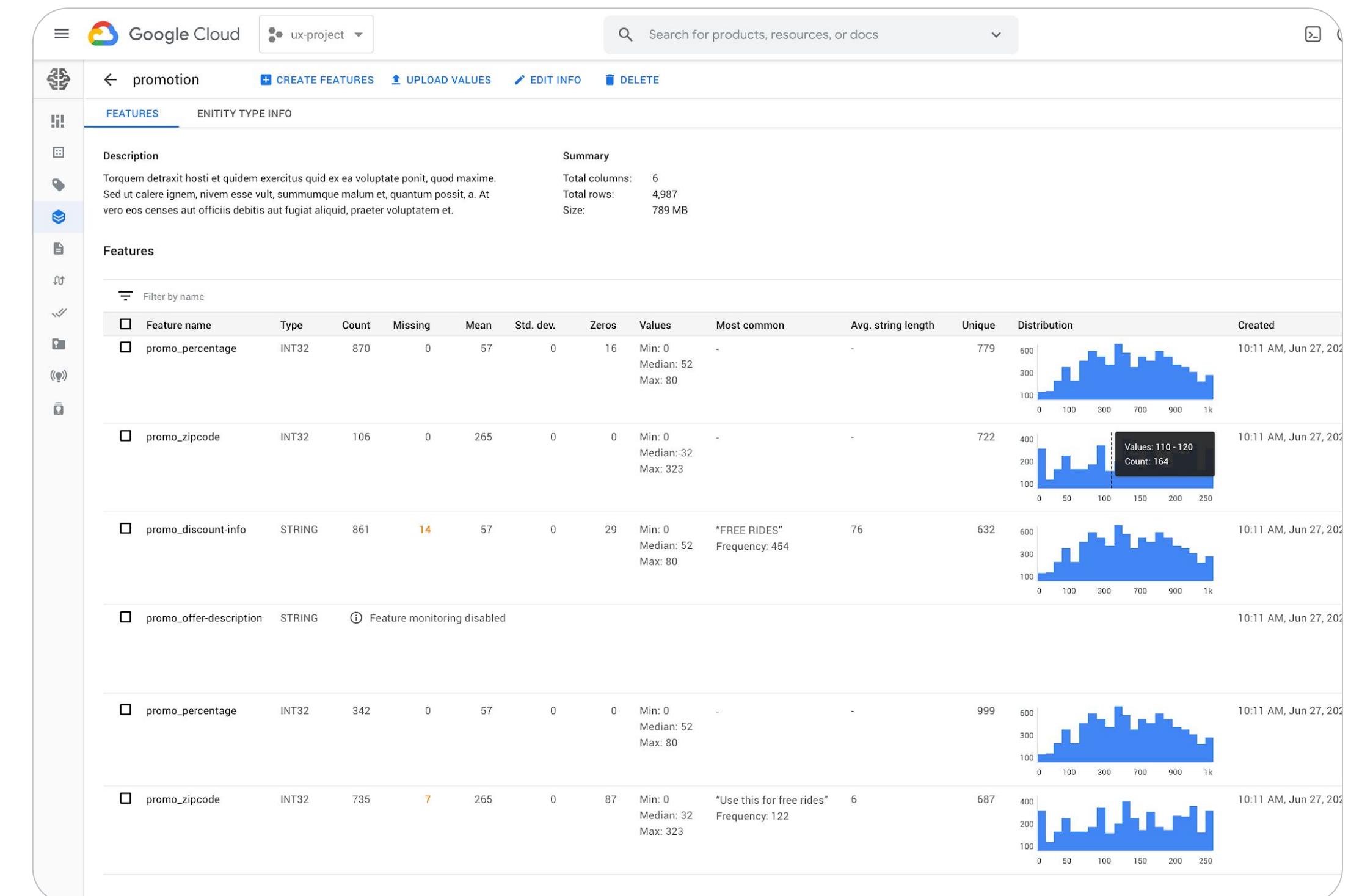


# Vertex Feature Store

Share and reuse ML **features** across use cases

Serve the features at scale with low latency

Alleviate training serving skew



03

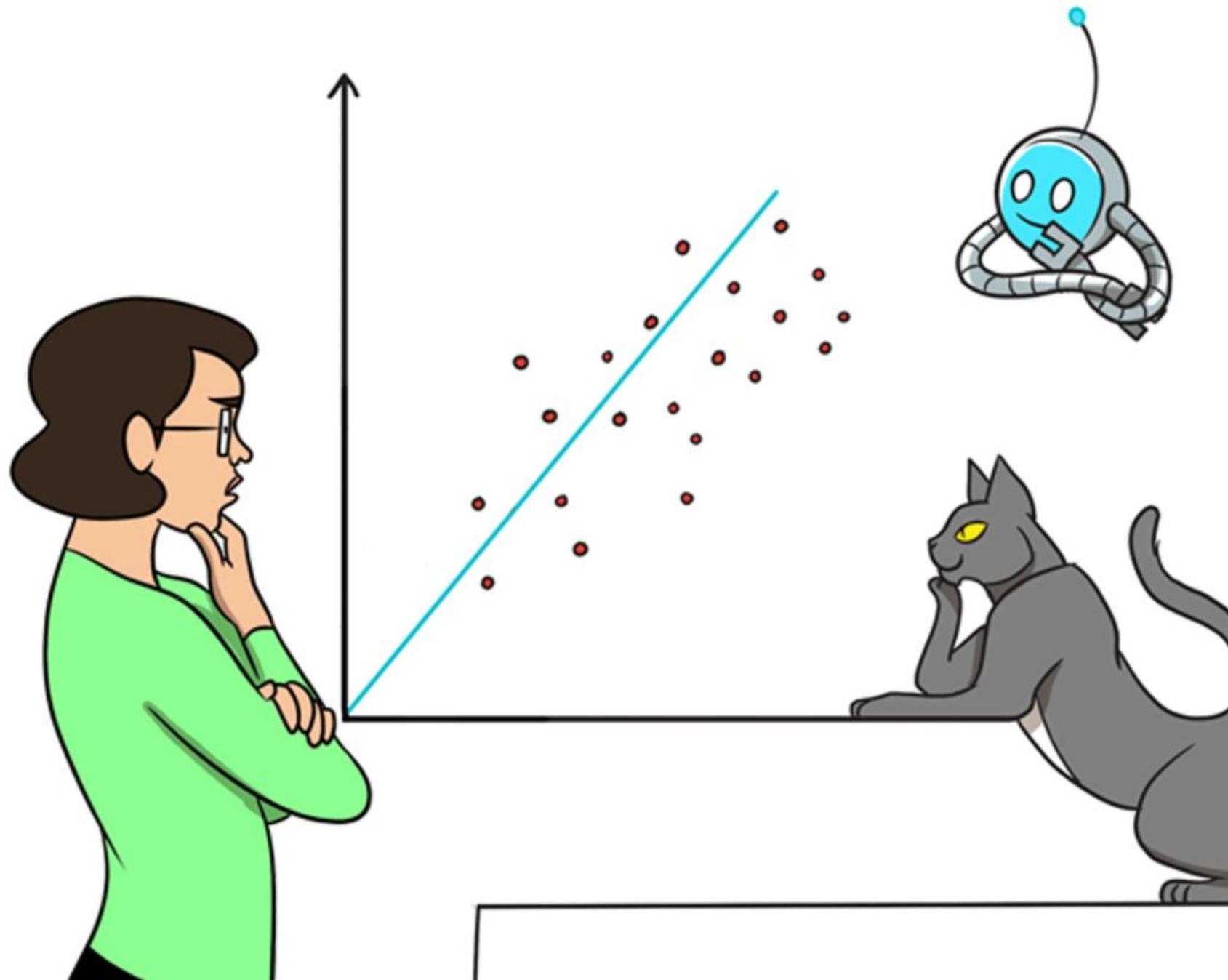
# Model Monitoring



# Why my model doesn't work well?

Any models **get stale**, as they are trained with data in the past

After the deployment, you would see **degradation** of performance and UX



Google

Google Cloud

# The Silent Killer: Skew and Drift

**The important lesson learned at Google:  
Skew and Drift are the silent killers of your  
ML models**

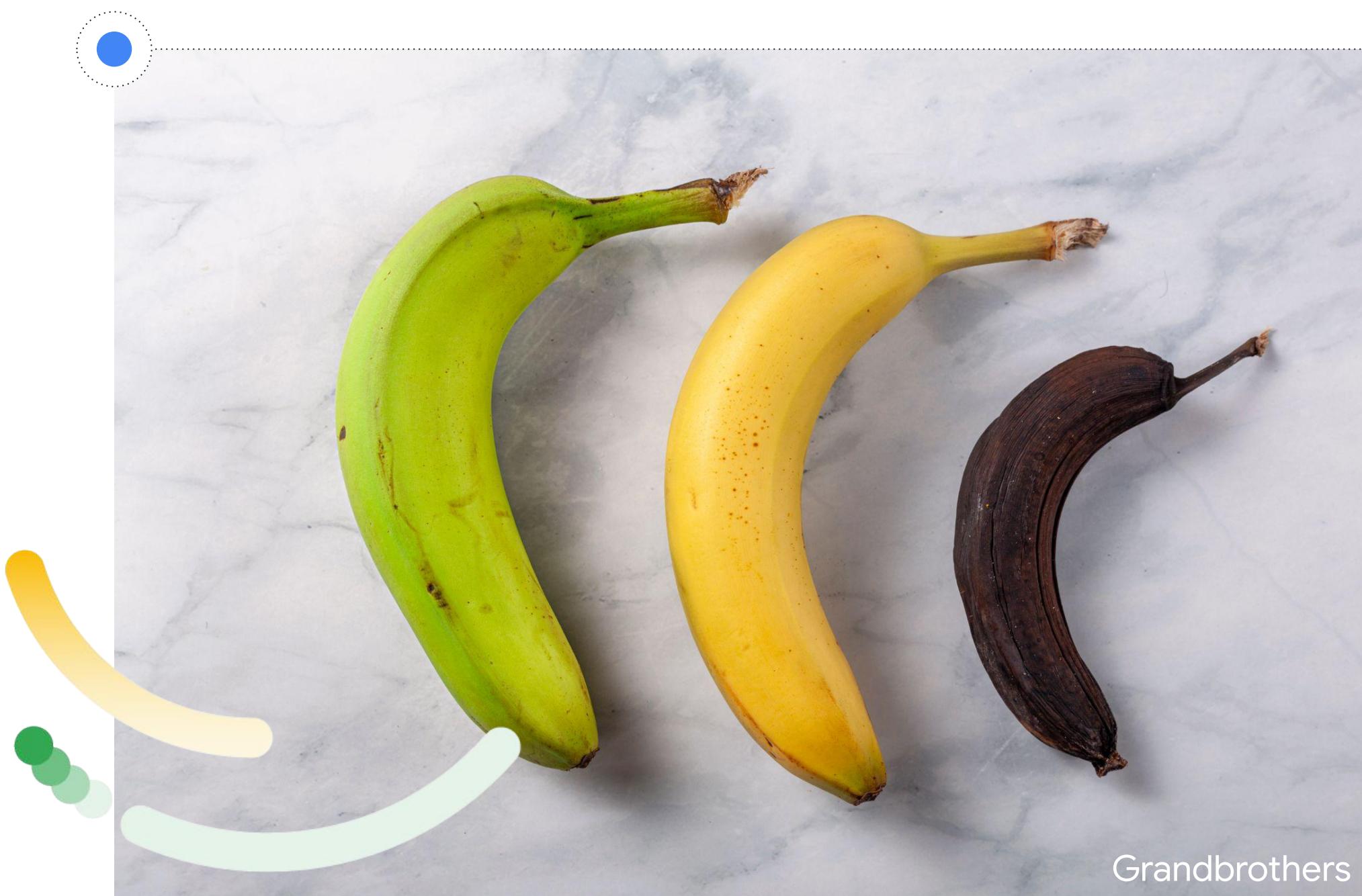
## **Training-Serving Skew:**

Feature at training: green banana

Feature at serving: yellow banana

## **Prediction Drift:**

Feature at serving: changing from green to  
yellow



Grandbrothers

Google Cloud

“



**We want the user to treat data errors  
with the same rigor and care that they  
deal with bugs in code”**

Google Play app install rate improved 2% after introducing data validation, finding stale table

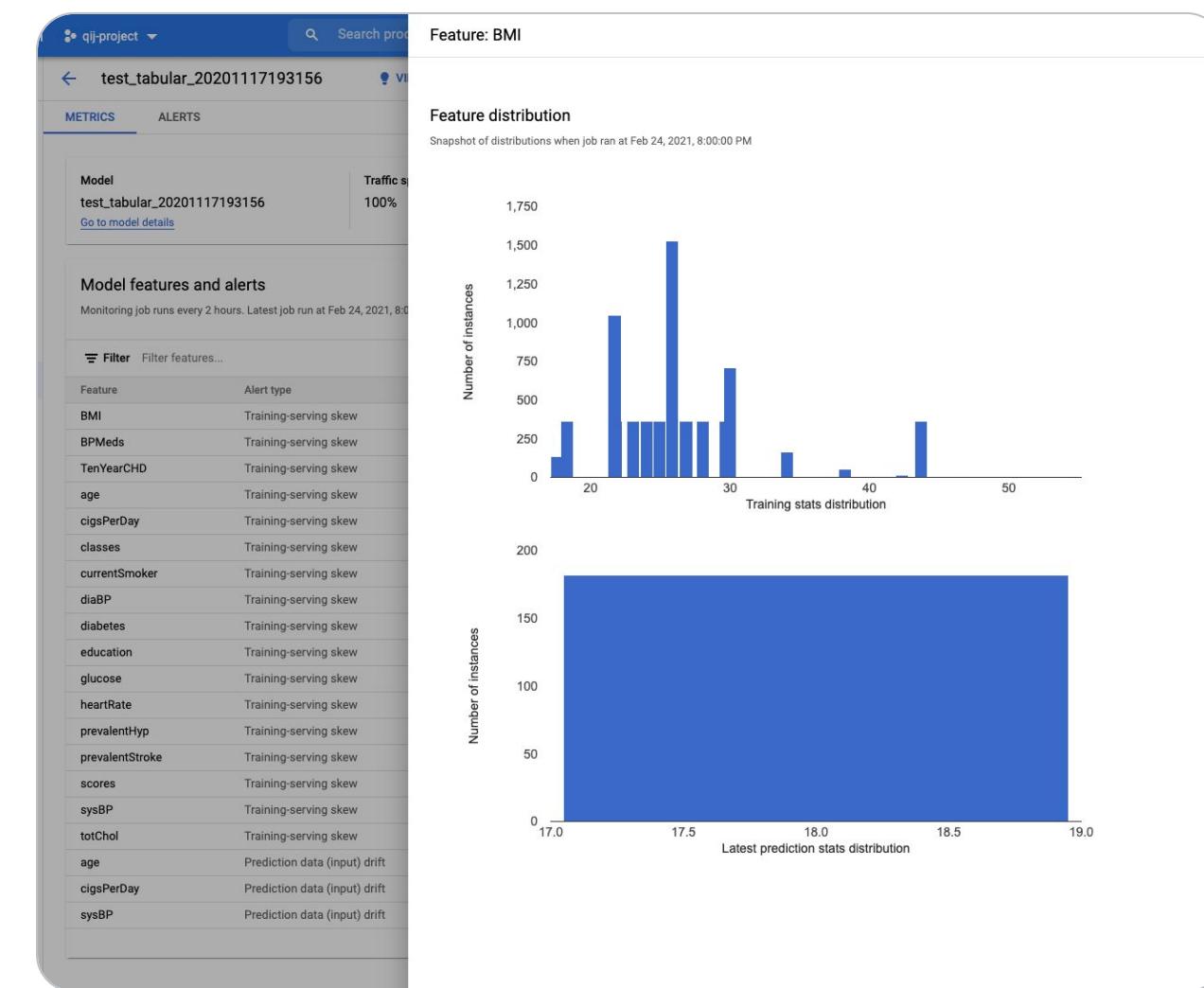
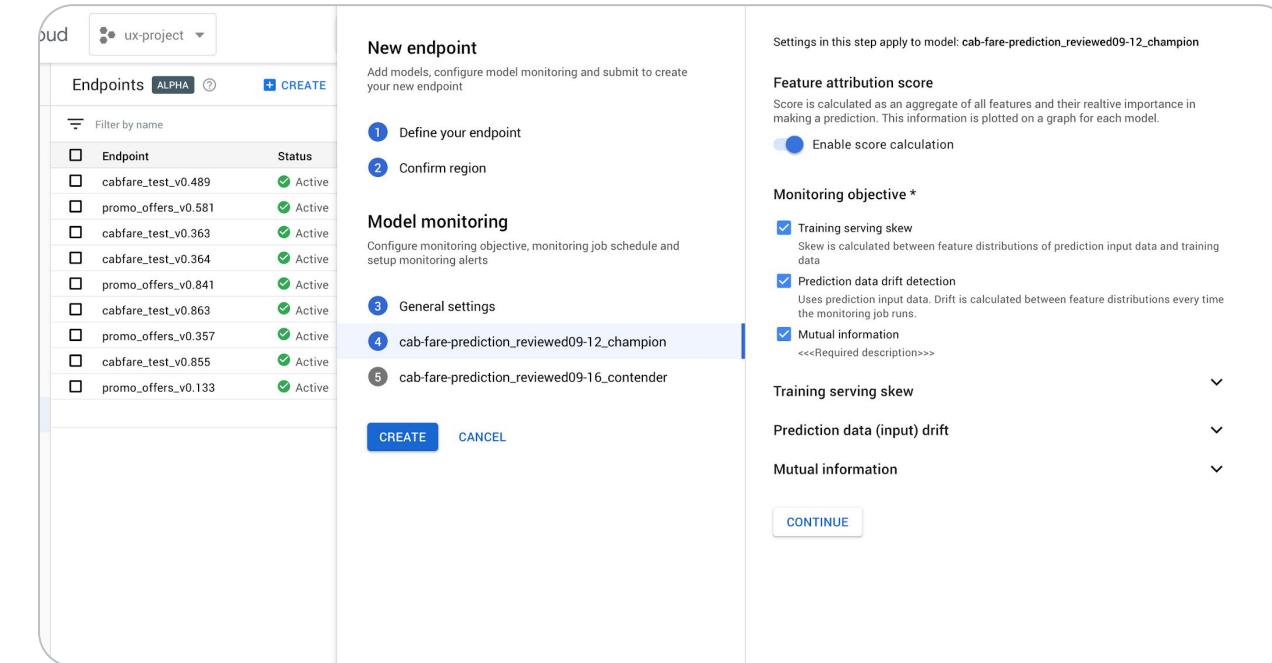
- [TFX paper](#)

# Vertex Model Monitoring

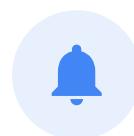
Automatically alert your data scientists and ML engineers when model performance changes

Detect drift and training-serving skew

Provides confidence in model reliability



# Easy and proactive monitoring of model performance



## Monitor and alert

For model's predictive performance



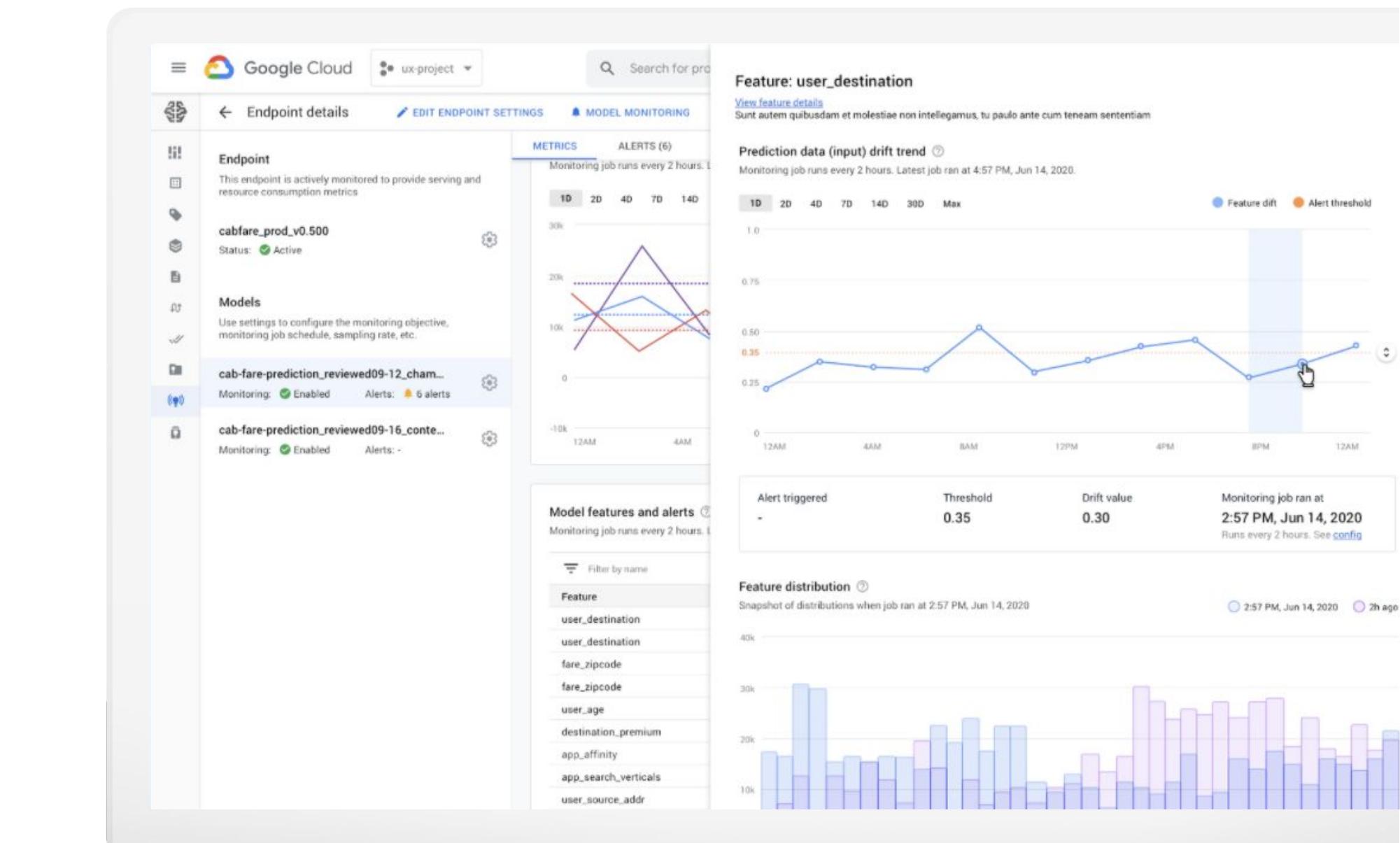
## Diagnose

Identify the cause for the deviation



## Update Model

Trigger model re-training pipeline



# Drift and skew detection



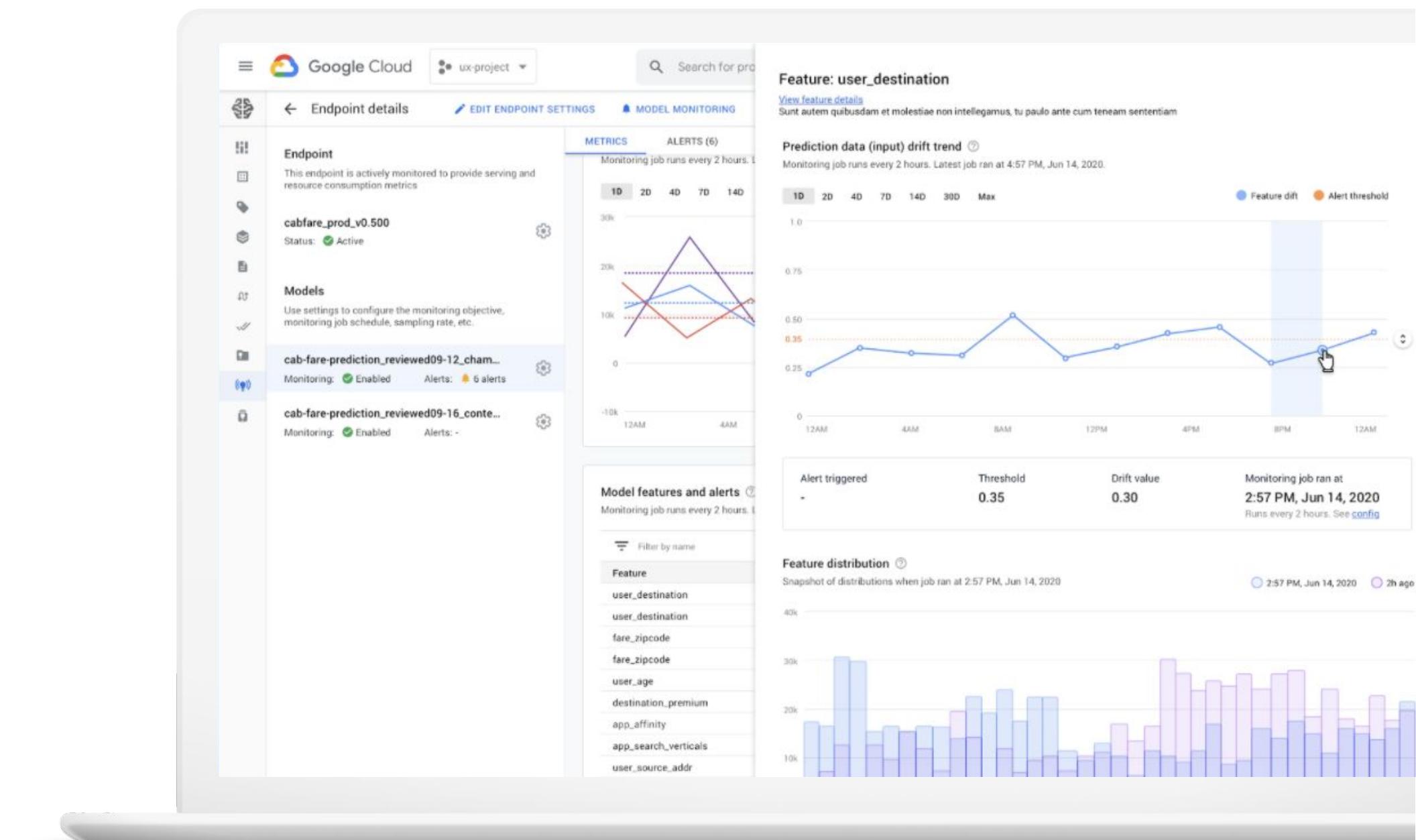
Visualize drift over time



Easily compare latest feature distributions with baseline



Easily set or update alerting thresholds



# How to detect Skew and Drift

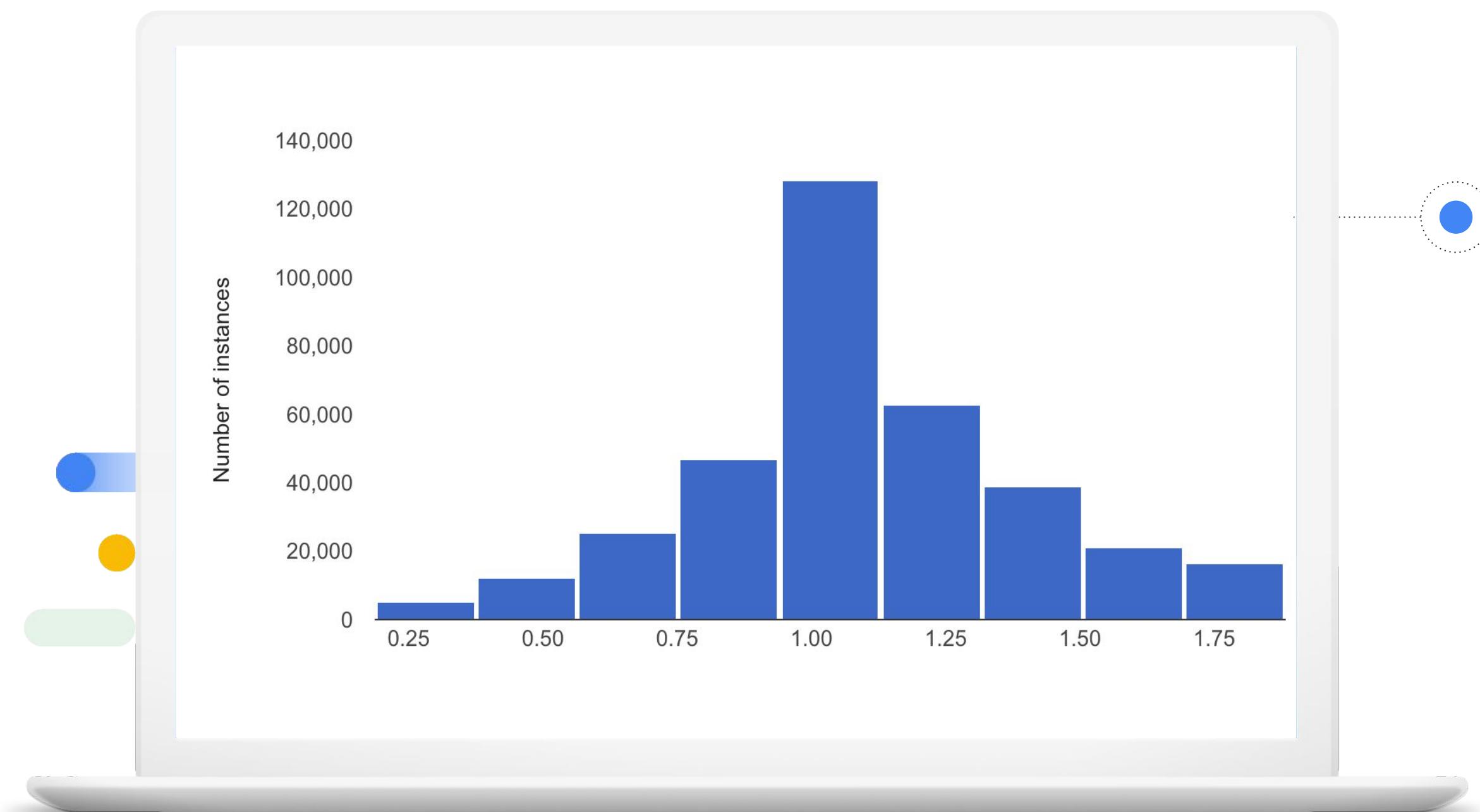
**For Skew:** set the baseline with stats of the training data

**For Drift:** set the baseline with the stats of the serving data in the past

**Measure:**

JS divergence  
(for numerical features)

L-inf distance  
(for categorical features)



04

# Demo





## Endpoints

[+ CREATE ENDPOINT](#)



Endpoints are machine learning models made available for online prediction requests.



Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results.



To create an endpoint, you need at least one machine learning model



### Region

us-central1 (Iowa)



[Filter](#) Filter endpoints



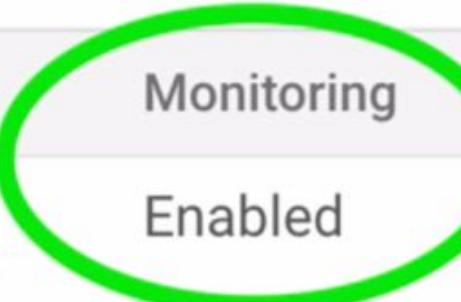
<input type="checkbox"/>	<input checked="" type="radio"/>	Name	ID	Models	Region	Monitoring	Most recent alerts	Last updated
--------------------------	----------------------------------	------	----	--------	--------	------------	--------------------	--------------



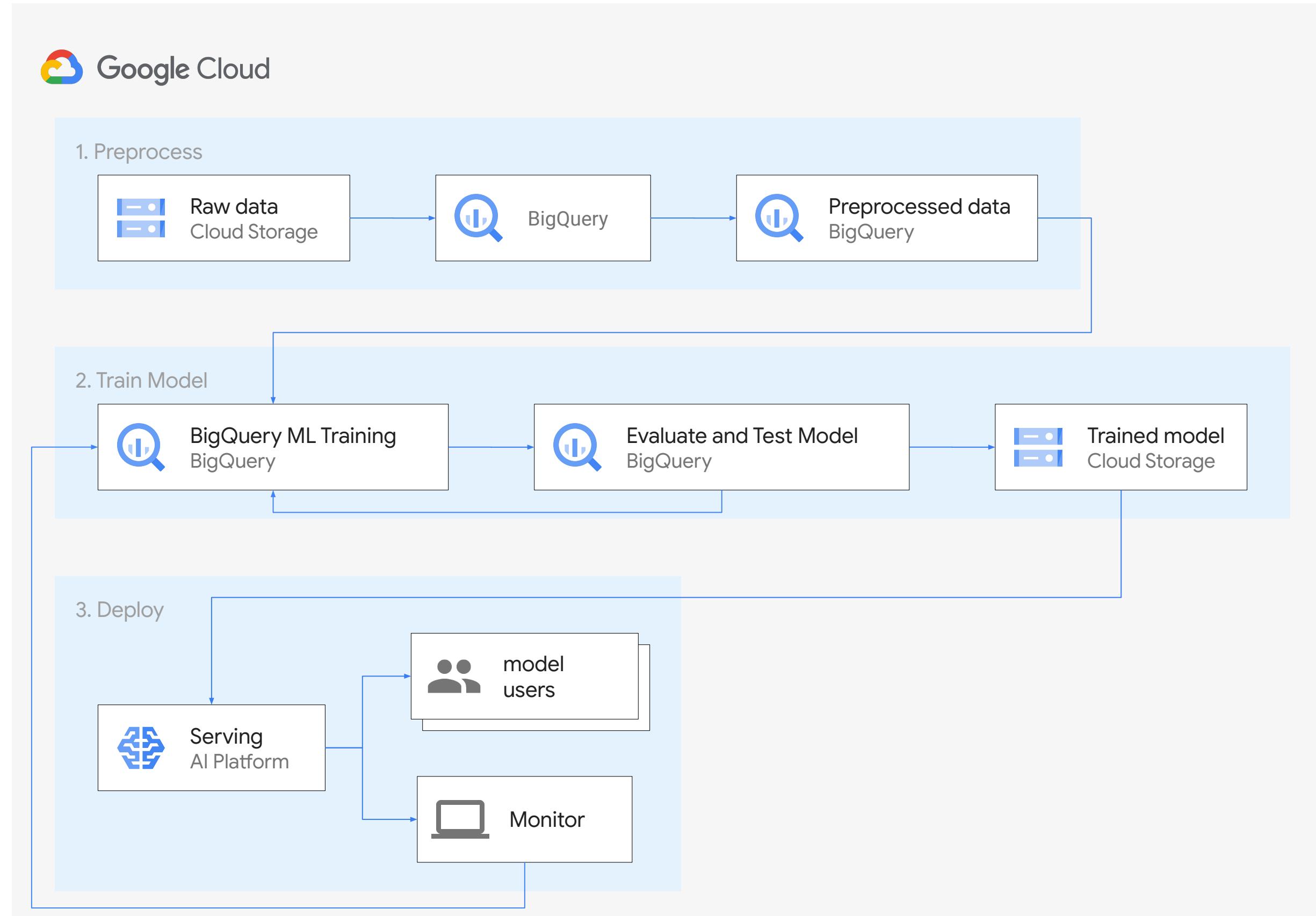
<input type="checkbox"/>	<input checked="" type="radio"/>	churn2	8124159476241530880	1	us-central1	Enabled	0 alerts	May 12, 2021,
--------------------------	----------------------------------	--------	---------------------	---	-------------	---------	----------	---------------



1:18:55 PM



## Model Monitoring Demo



05

# Next Steps





# Thank you.