Lecture 18: Operations - Automation

AC215

Pavlos Protopapas
SEAS/ Harvard

Outline

1. Recap
2. Motivation
3. Automation

Outline

1. **Recap**
2. Motivation
3. Automation

Recap: MLOps - Tasks

Machine / Deep Learning:

- Data collection & exploration
- Model exploration & selection
- Training & evaluation
- Distillation & compression

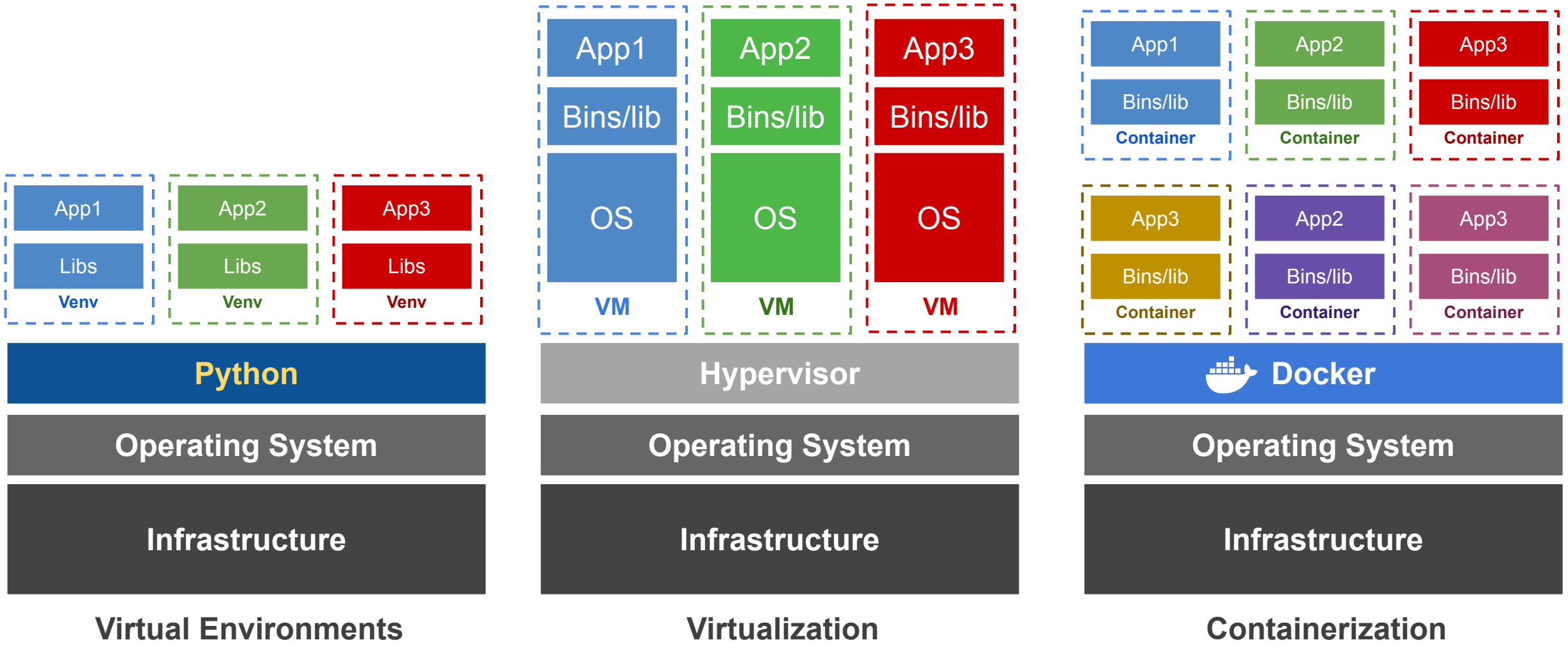
Application Development:

- APIs / Model serving
- ML integration
- Web & mobile apps
- Edge device apps
- Automation scripts

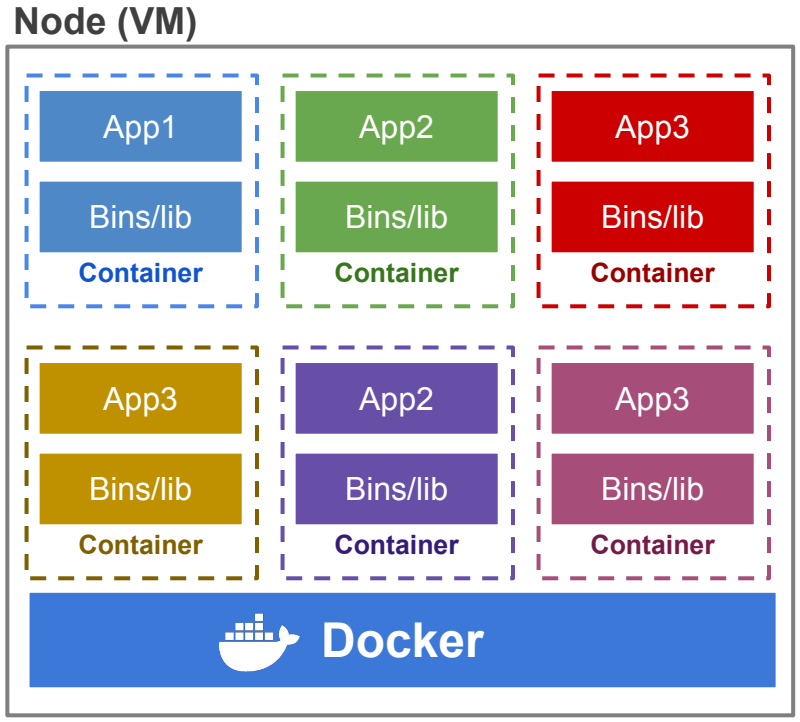
Operations:

- Provisioning and managing deployment servers, on-demand GPU servers
- Maintain 100% uptime of app / apis
- CI/CD: Continuous Integration / Deployment
- Continuous Data Collection / Model Training
- Model/data monitoring
- Model/data versioning
- ML Workflow Management

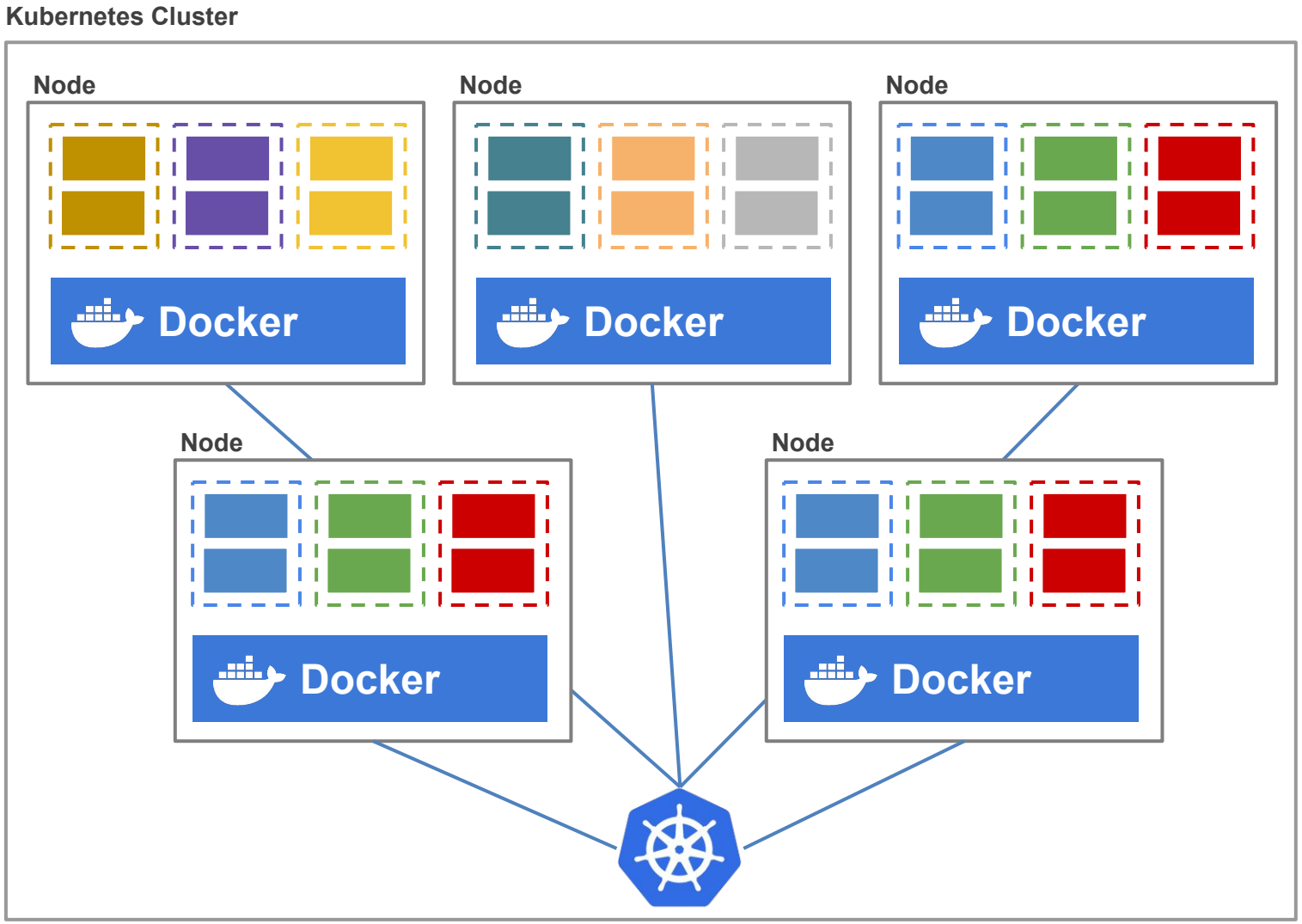
Recap: Environments vs Virtualization vs Containerization



Recap: Container vs Kubernetes Deployment



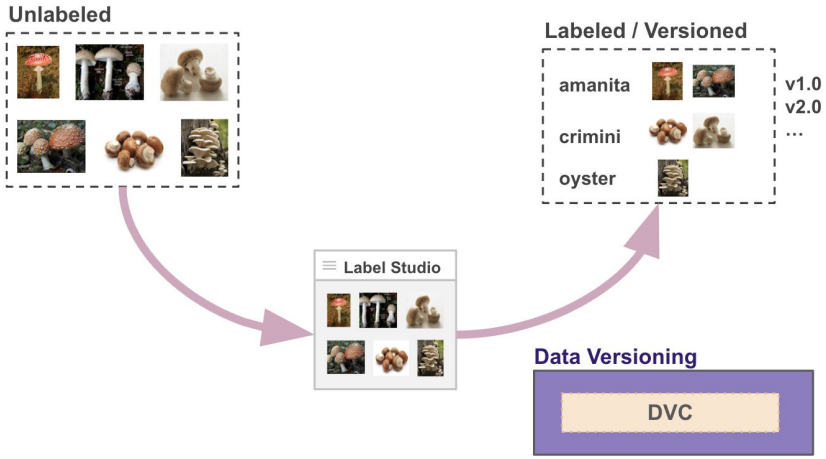
Container Deployment



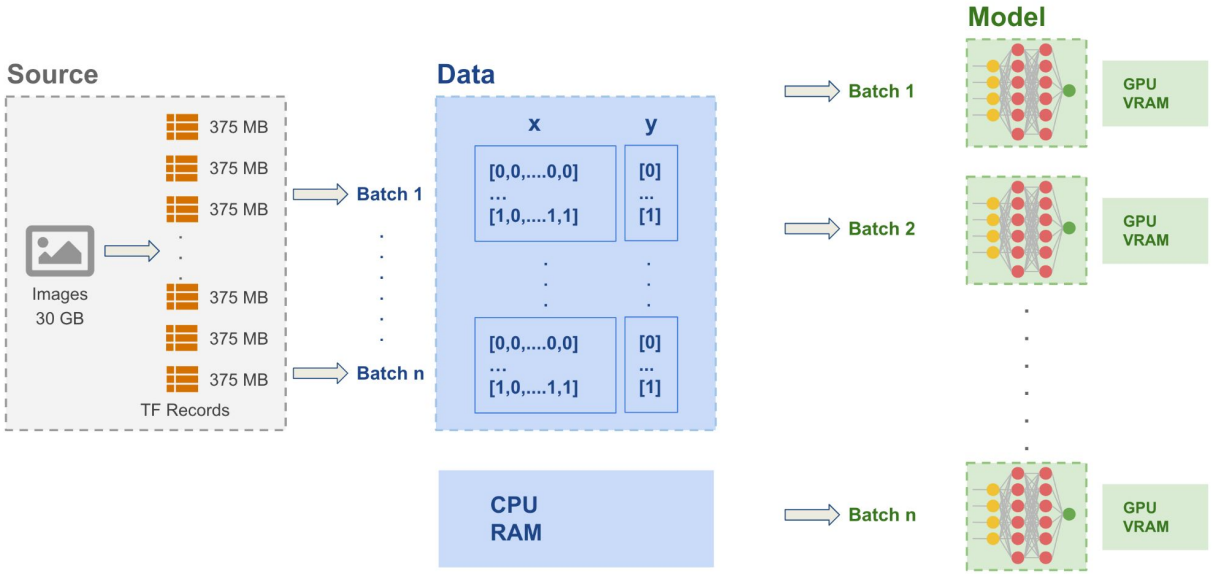
Kubernetes Deployment

Recap: Data

Data Labeling / Versioning



Data Parallelism



Data stored as TFRecords

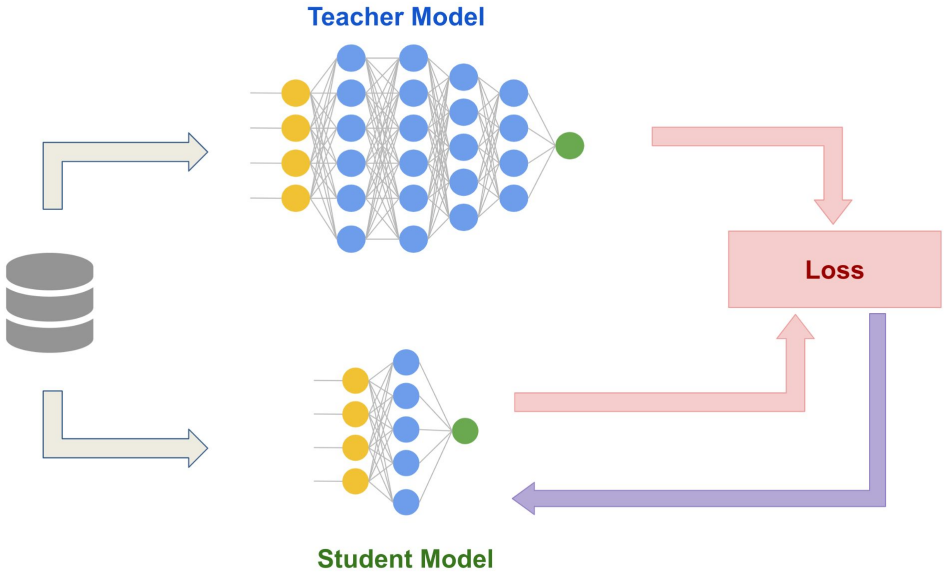
1 Training Server

Model trained on multiple GPUs

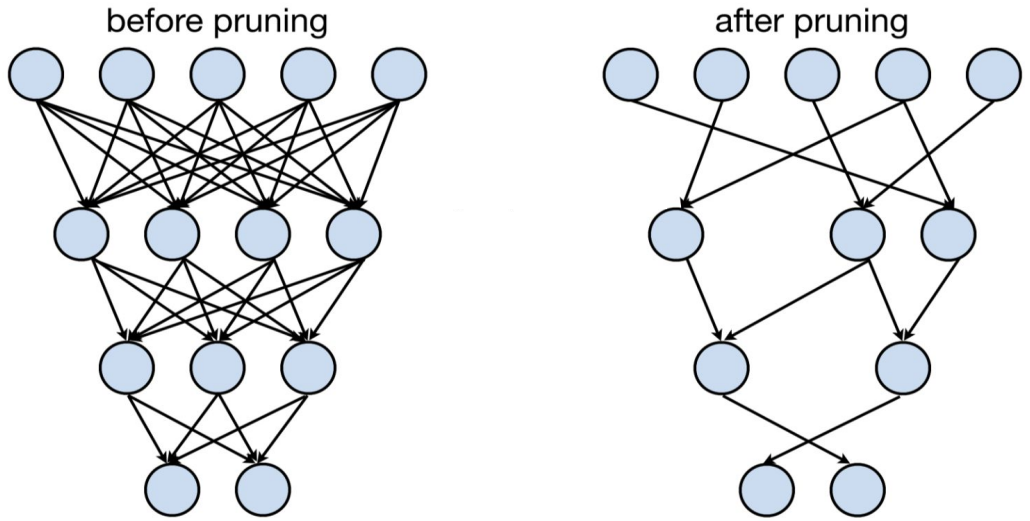
41

Recap: Model

Distillation

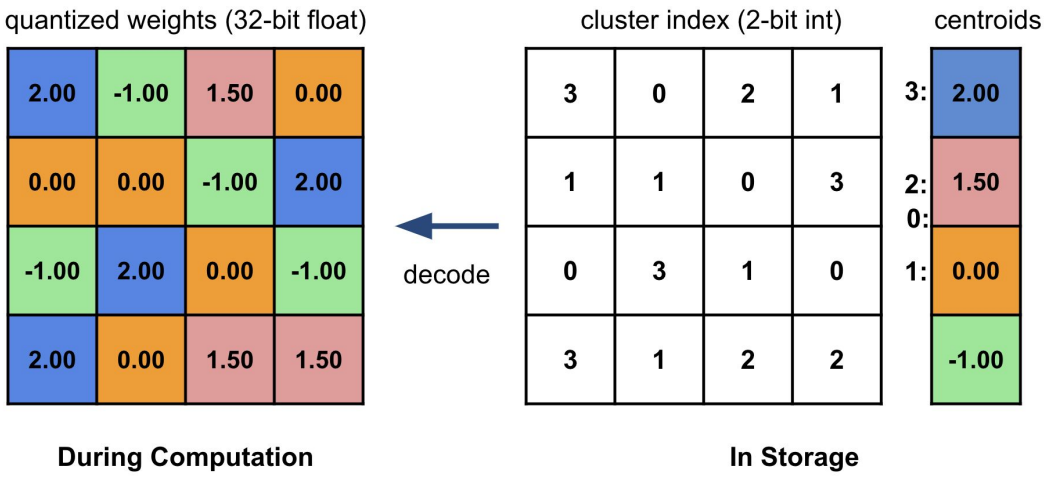


Pruning

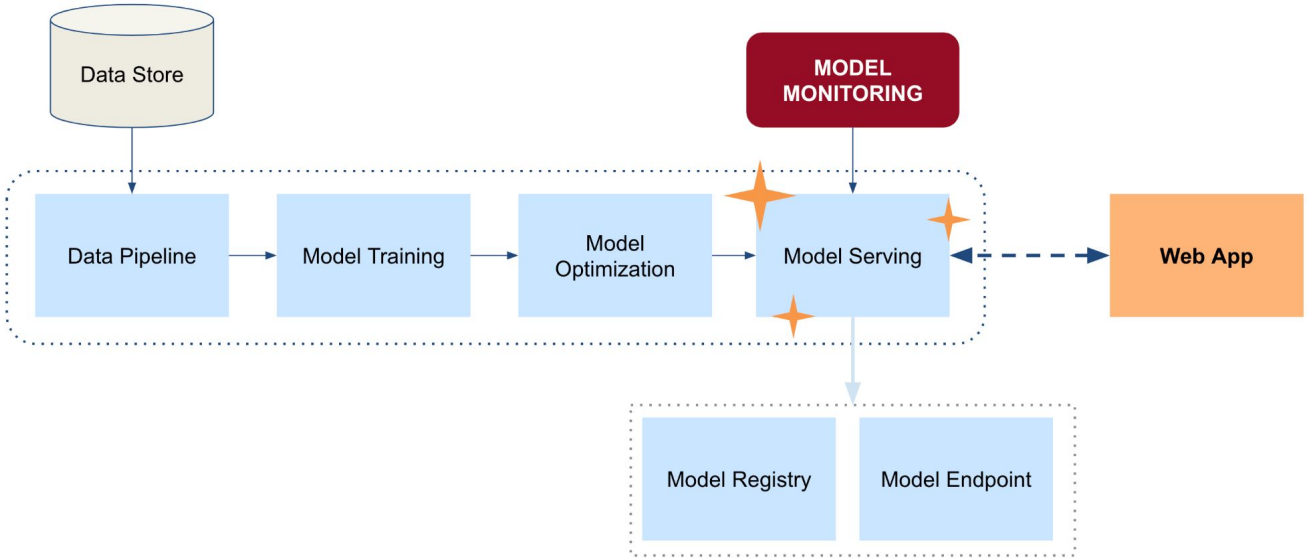


Recap: Model

Quantization

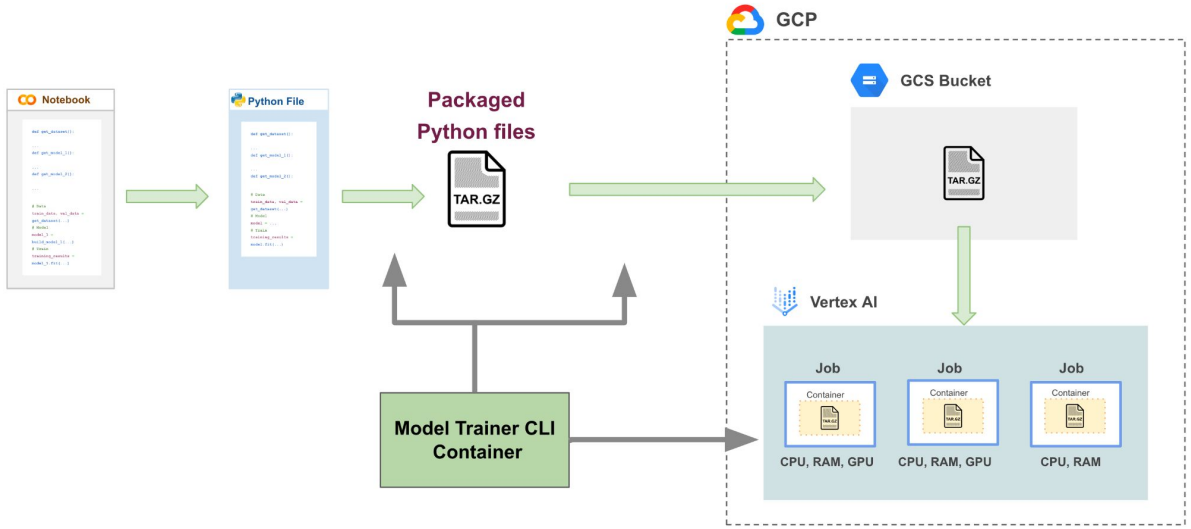


Model Monitoring

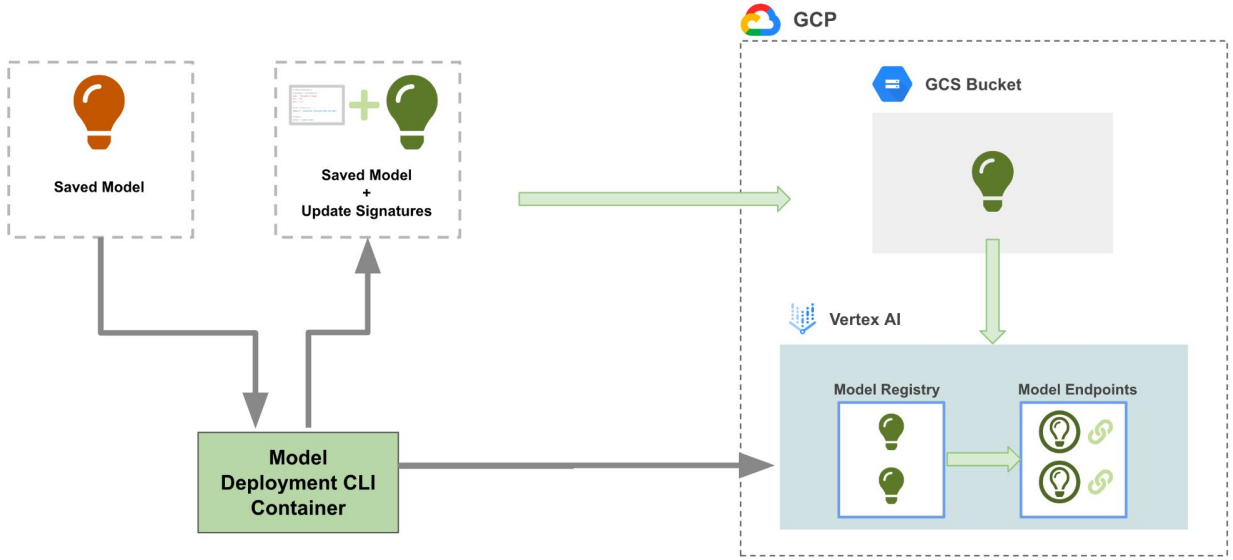


Recap: Model

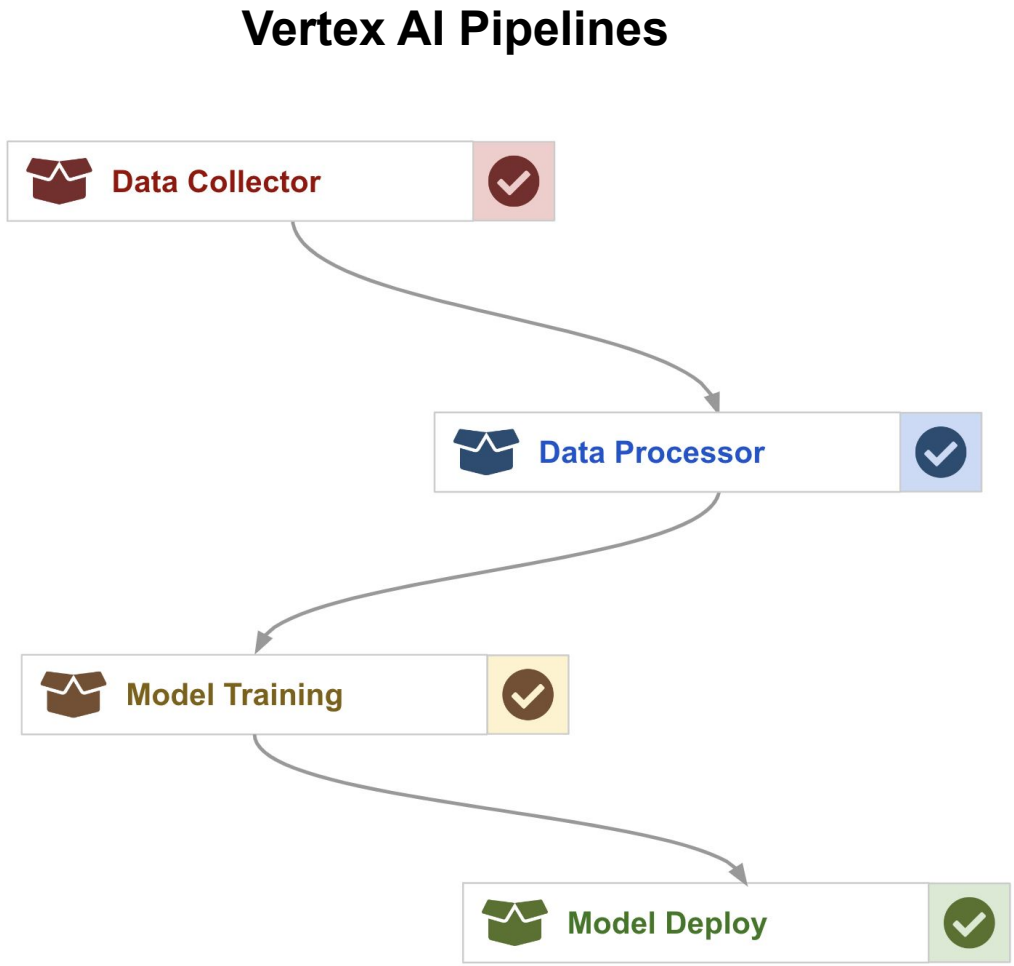
Serverless Training



Serverless Deployment

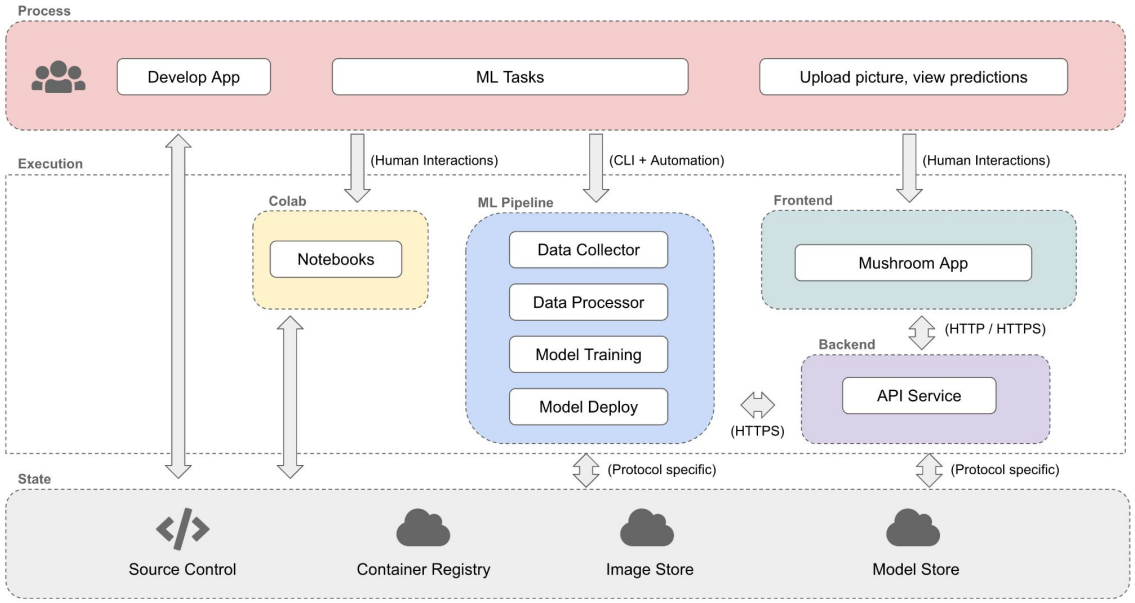


Recap: ML Workflow

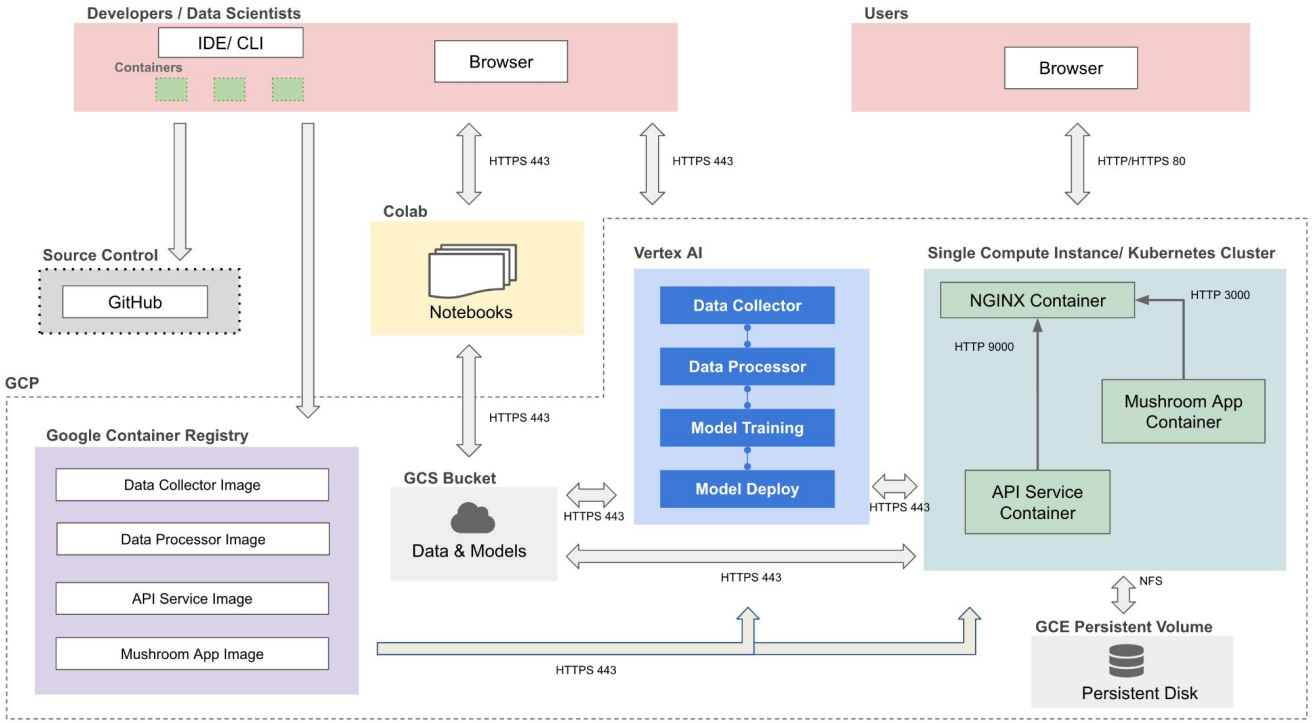


Recap: Design

Solution Architecture

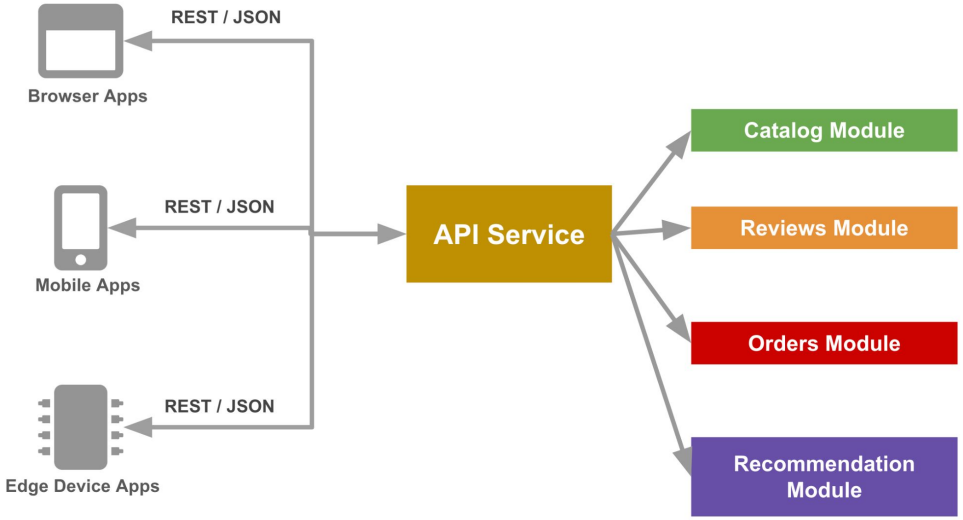


Technical Architecture



Recap: AI App

Backend APIs



Frontend

Recap: MLOps - Tech Stack

Deep Learning

Data:

Dask, TF Data
Label Studio, DVC

Framework:

TensorFlow
PyTorch

Training:

Google Colab
Vertex AI

Tracking:

W&B
Custom

Development

APIs:

FastAPI
Vertex AI Serving

Frontend:

HTML, Javascript
React

Workflow:

Kubeflow

IDE:

VS Code

Operations

Source Control:

GitHub

Containerization:

Docker

Cloud Provider:

Google Cloud Platform

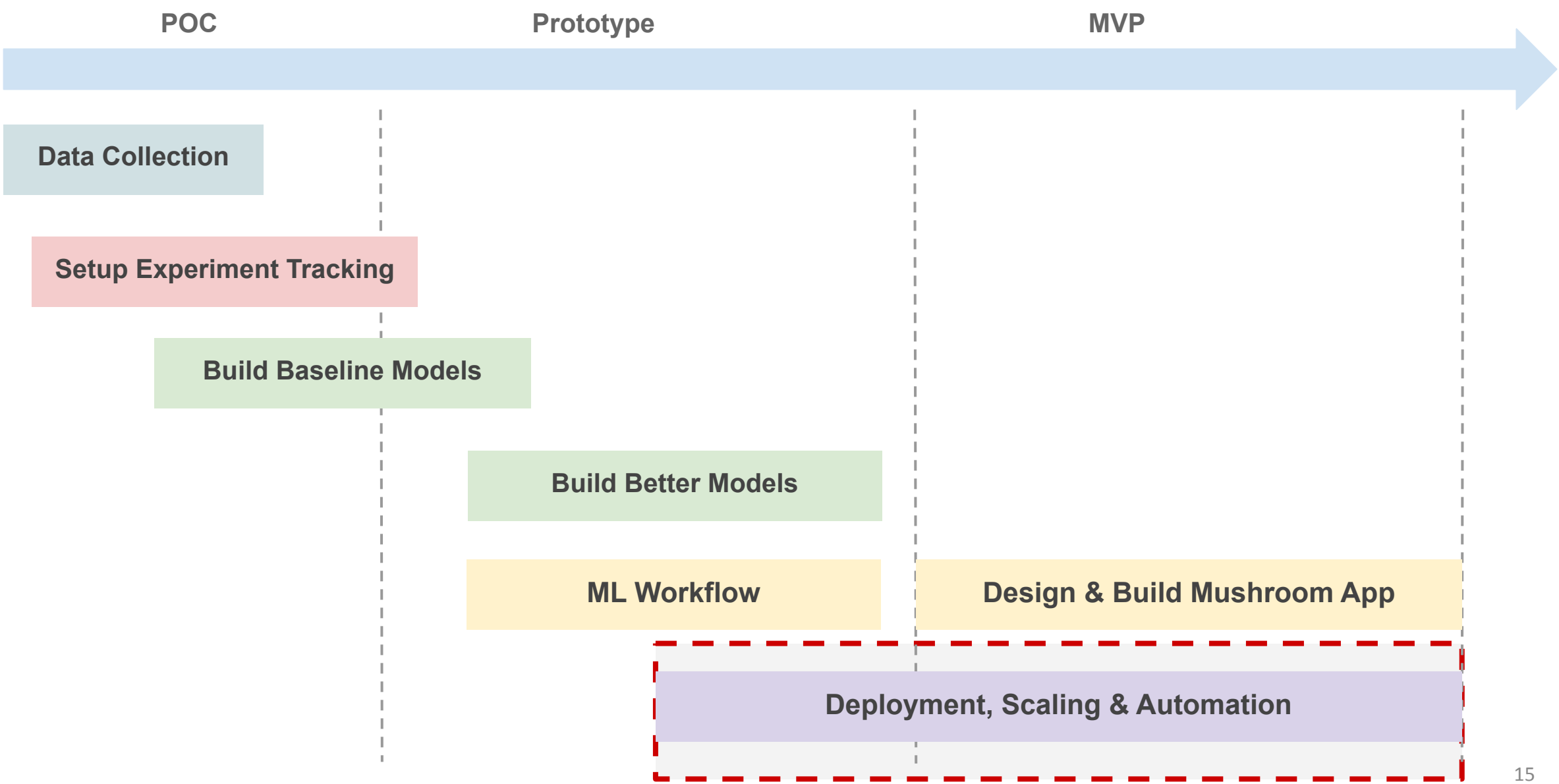
Continuous Integration/ Deployment:

Ansible, GitHub Actions

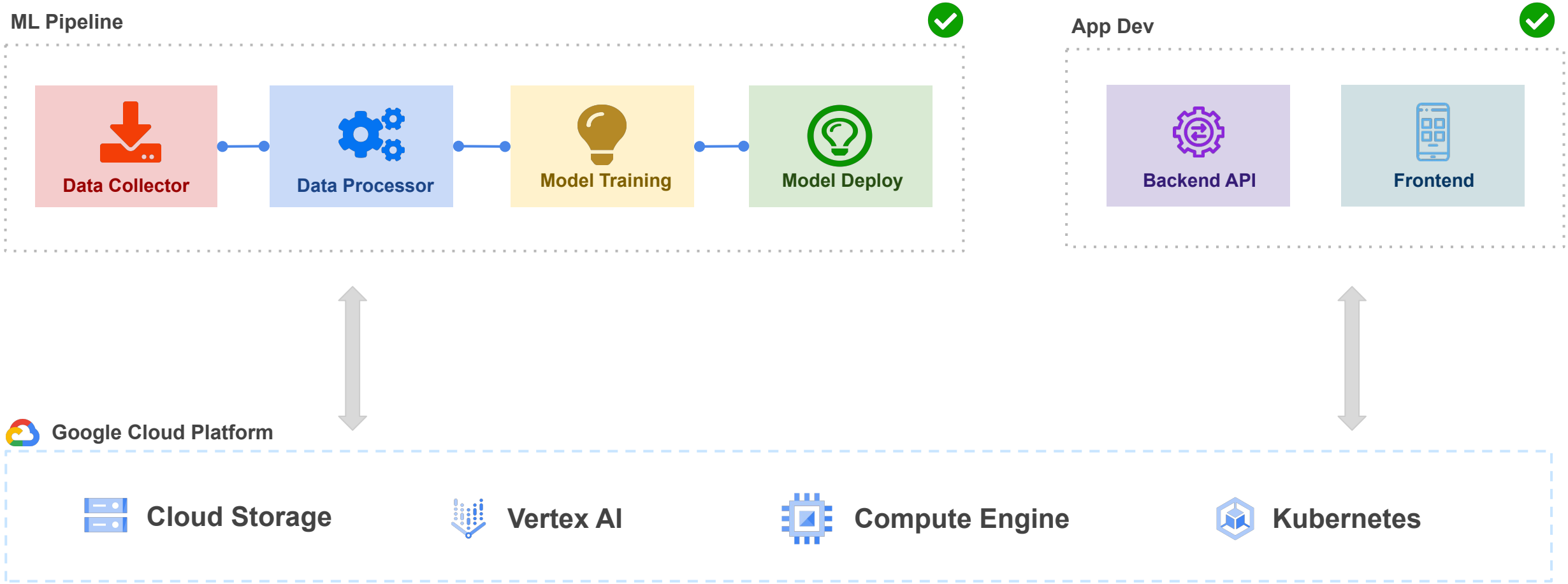
Scaling:

Kubernetes

Recap: Mushroom App Status



Recap: Mushroom App Development

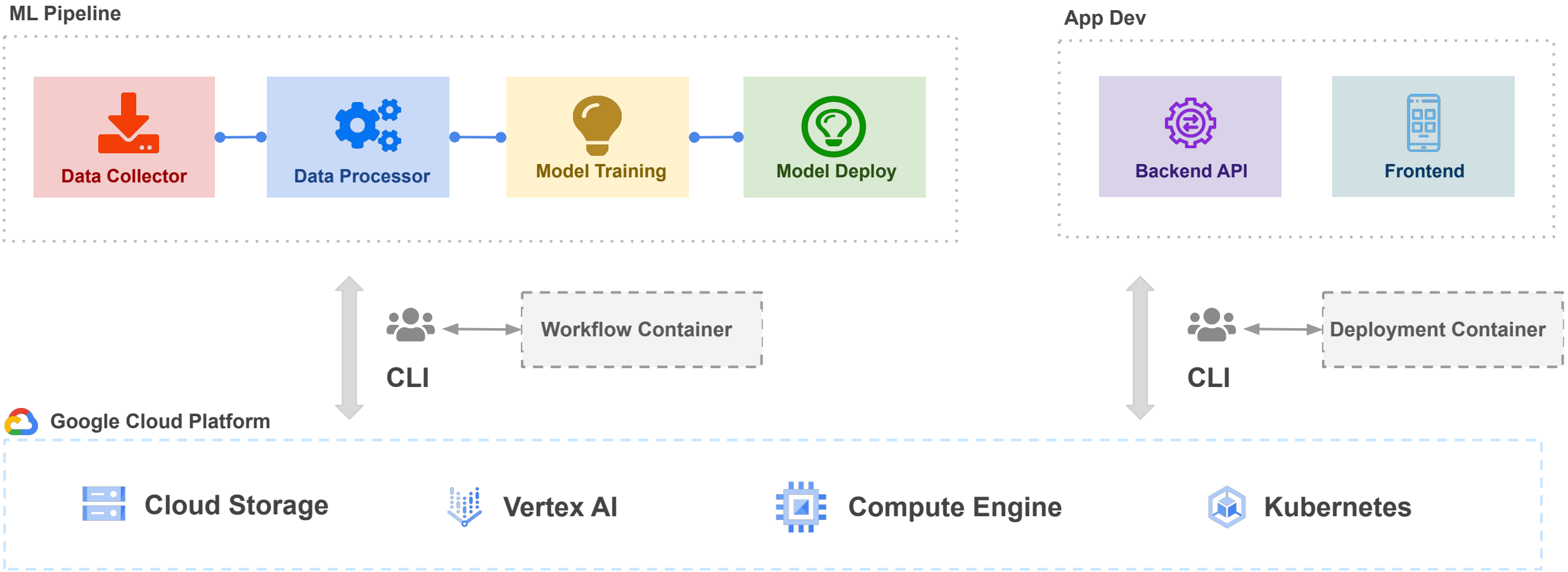


Outline

1. Recap
2. **Motivation**
3. Automation

Motivation: Mushroom App Development

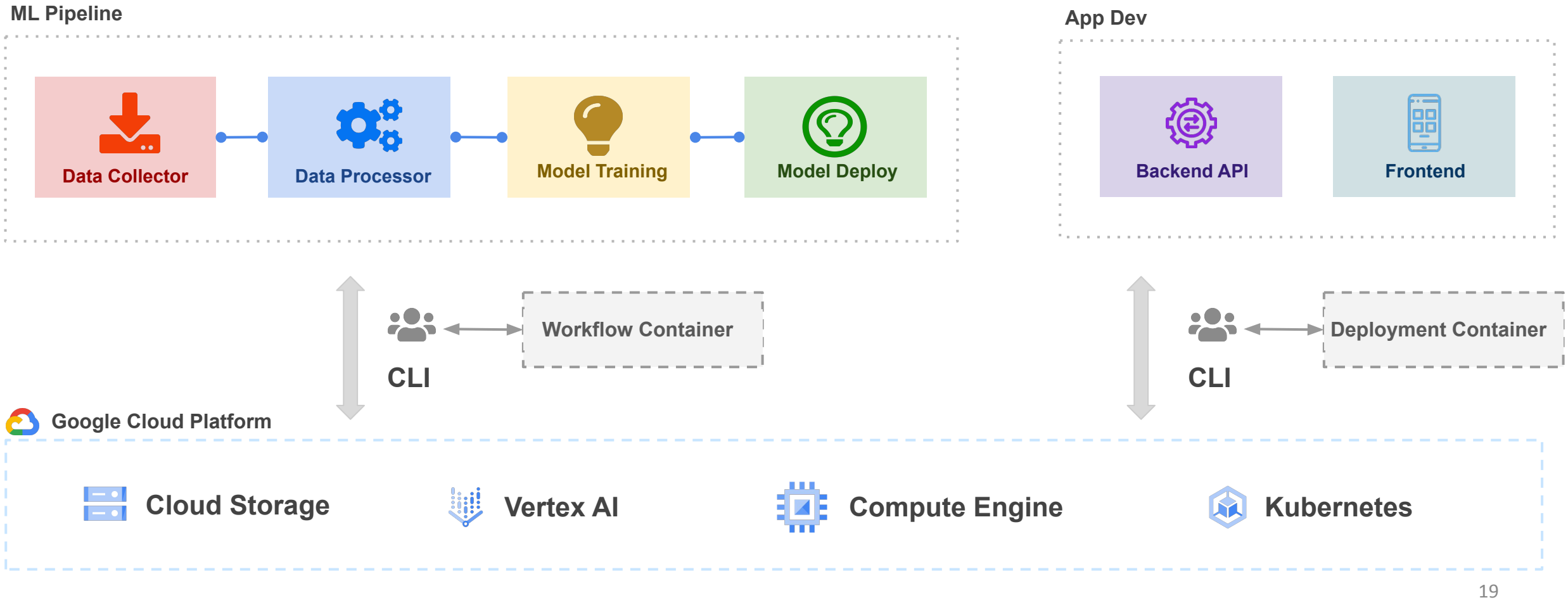
How did we run / deploy these components?



Motivation: Mushroom App Development

Can we Automate?

Continuous Integration (CI)
Continuous Delivery/Deployment (CD)



Outline

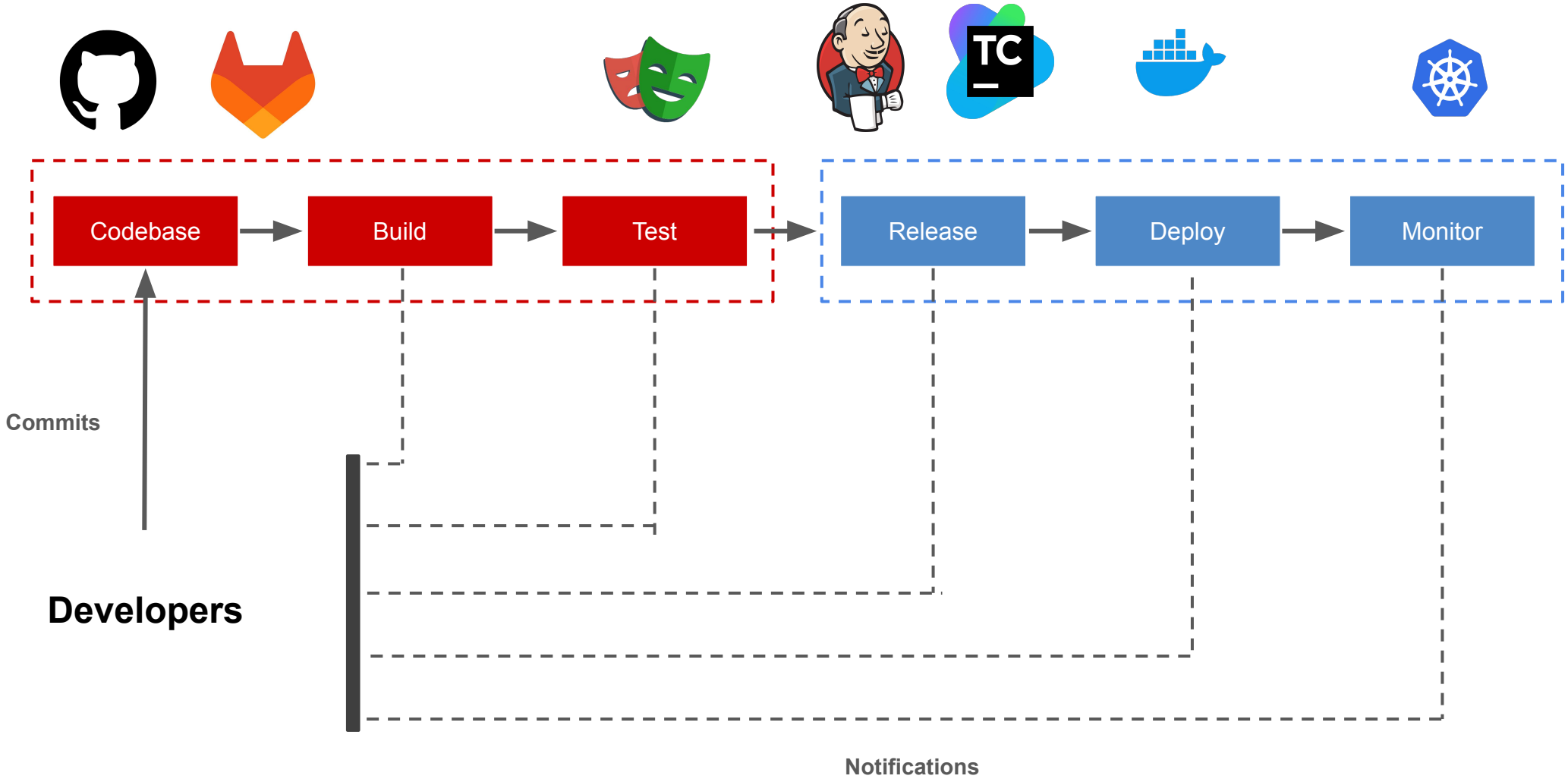
1. Recap
2. Motivation
3. **Automation**

Automation

We want to automate the following operations:

- Running data pipeline jobs
- Model training & deployment
- Frontend deployment
- Backend deployment
- Building & Pushing Docker containers

CI / CD


Codebase
Build
Test
Release
Deploy
Monitor
Commits
Developers
Notifications

Continuous Integration (CI)

Continuous Integration (CI): run a series of scripts automatically, anytime changes are pushed to do the following:

- Continuously Integrate our changes
- Automated tests - (not in this class :) CS107/AC207)
- Ensure coding standards
- Static code analysis

Continuous Delivery / Deployment (CD)

Continuous Delivery (CD): extension of CI to ensure software can be reliably released at any time

Continuous Deployment (CD): Is to take automation further by deploying code changes to production automatically.

CD (Continuous Delivery vs Deployment)

Delivery: every change is proven to be deployable at any time

Deployment: every change is actually deployed automatically

CI in ML

In AI/ML, CI involves regularly merging ML code, data, and models and automatically testing these integrations

- Quick detection of issues in data, code, or models

Challenges

- Balancing need for frequent integration with the computational demands of training and testing ML models.

CD in ML

In the context of AI/ML, CD involves the automated deployment of ML models into production environments.

- Streamlined processes allow for faster deployment of improved models.

Challenges

- Managing complex dependencies and configurations specific to ML models.
- Handling data drift and model degradation.

CI/CD Providers and Tools

Some of the common CI CD provider and tools:

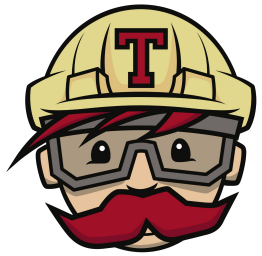
TeamCity

Github

GitLab

Jenkins

CircleCI



TravisCI

How to Implement CI CD

We already have a deployment container that can:

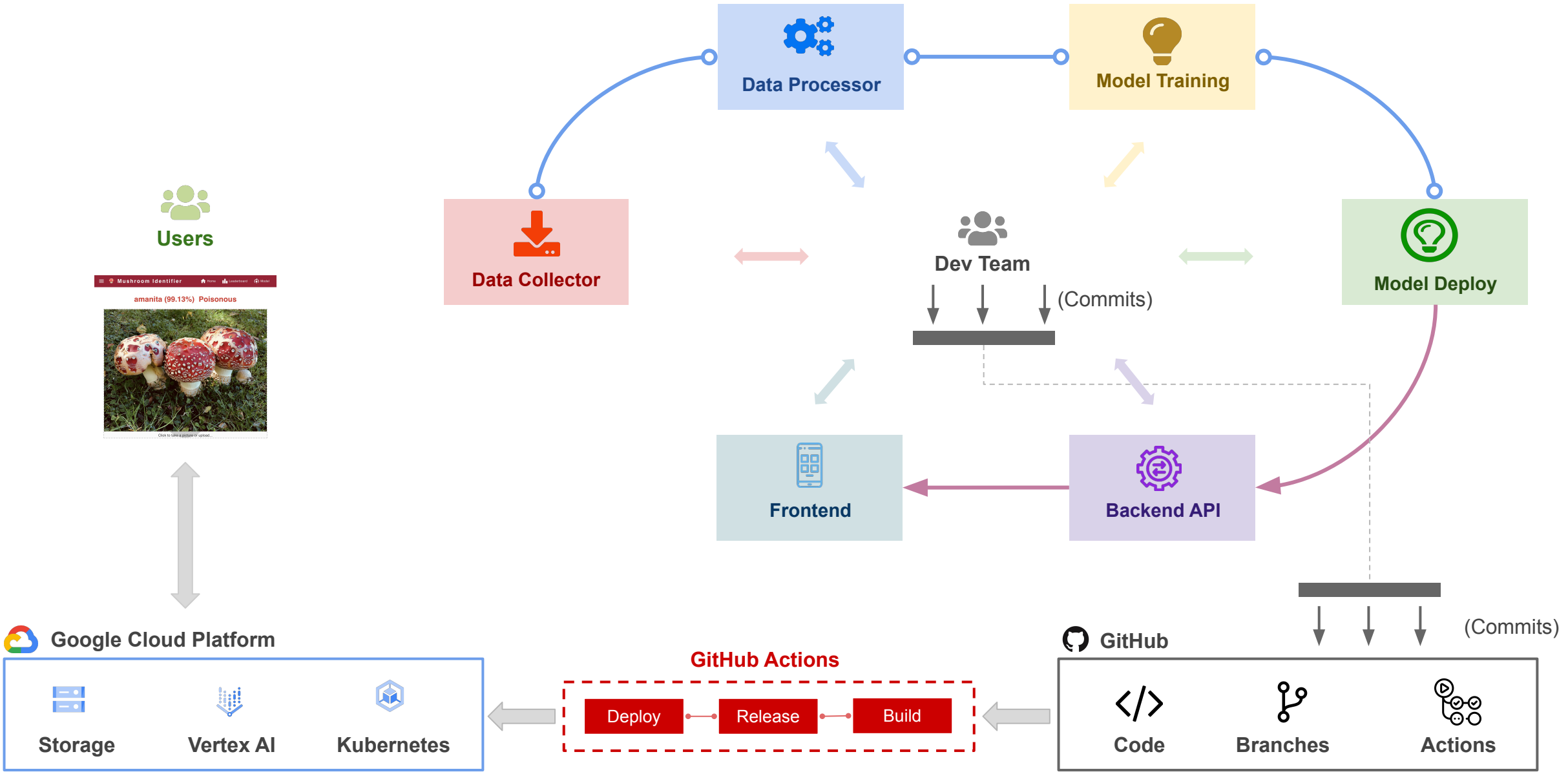
- Build Docker images.
- Run Vertex AI pipeline jobs.
- Deploy app to K8s cluster.

How to Implement CI CD

We can automate using GitHub Actions by:

- Monitoring code commits.
- Build & run **deployment container**.
- Invoke CLI in deployment container to:
  - Build & push docker images for release
  - Run Vertex AI jobs using new newly build images
  - Deploy newly build images of app to K8s cluster

Mushroom App: CI CD



Tutorial: Continuous Integration, Continuous Deployment

Steps to apply **CI / CD** on the mushroom app components:

- Make Deployment container callable.
- Create a Github / Workflow file defining deployment steps
- For detailed instructions, please refer to the following link
  - Mushroom App CICD. (https://github.com/dlops-io/mushroom-app-v4#mushroom-app---automation)
  - Mushroom App - GitHub Actions. (https://github.com/dlops-io/mushroom-app-v4/actions)

THANK YOU