



Kubeflow

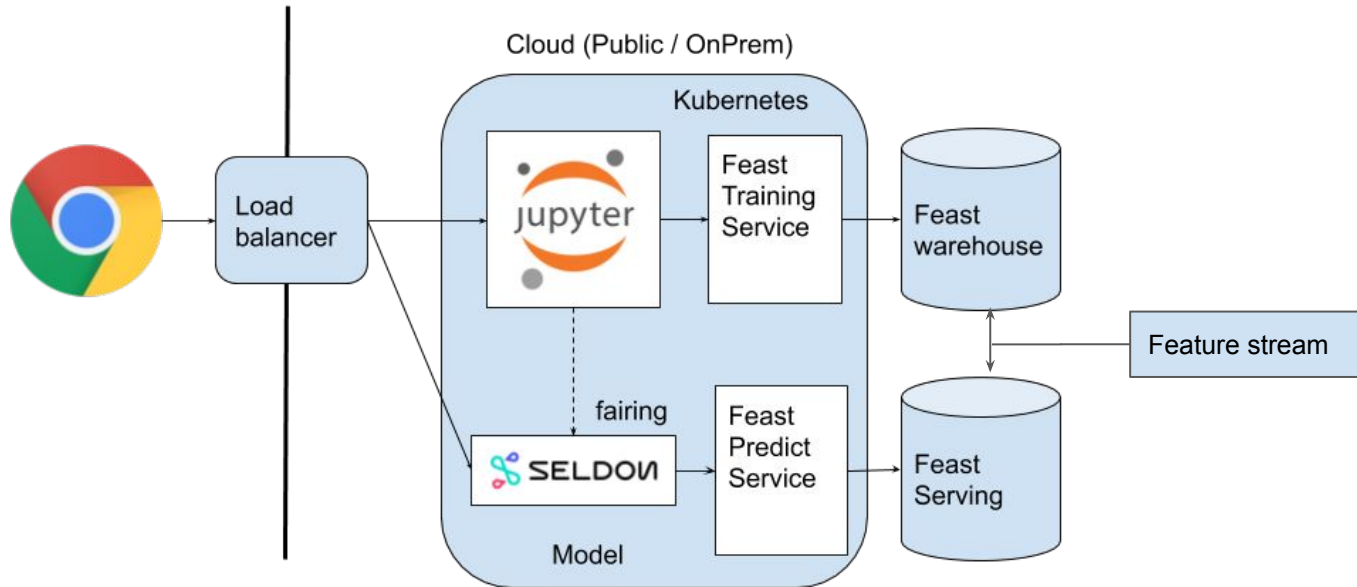
Moving People and Products with ML on Kubeflow

Jeremy Lewi (jlewi@) Google
Willem Pienaar GOJEK
2019-05-23



Takeaway Message

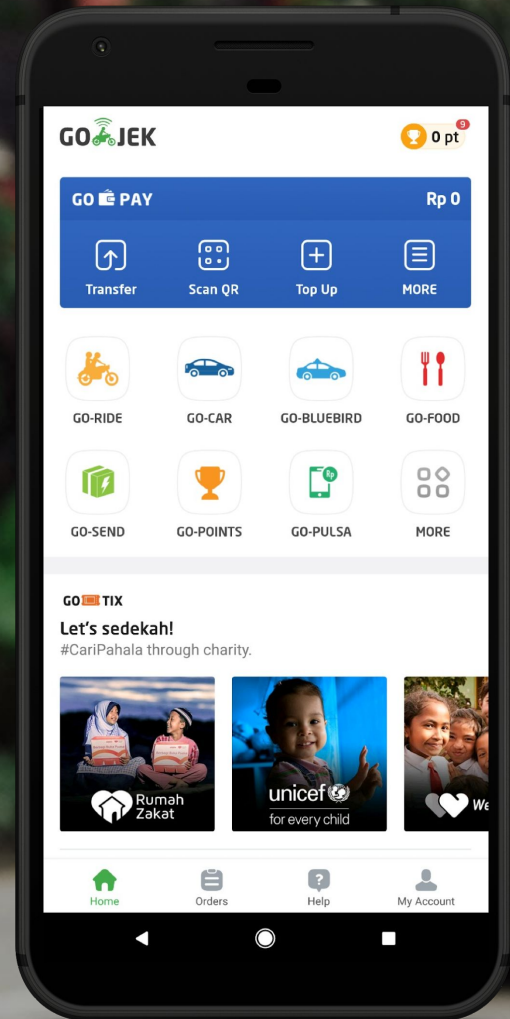
1. Kubernetes + Kubeflow is a really good platform for ML
2. Feast (Feature Store) + Kubeflow lets data scientists rapidly iterate on models



Agenda

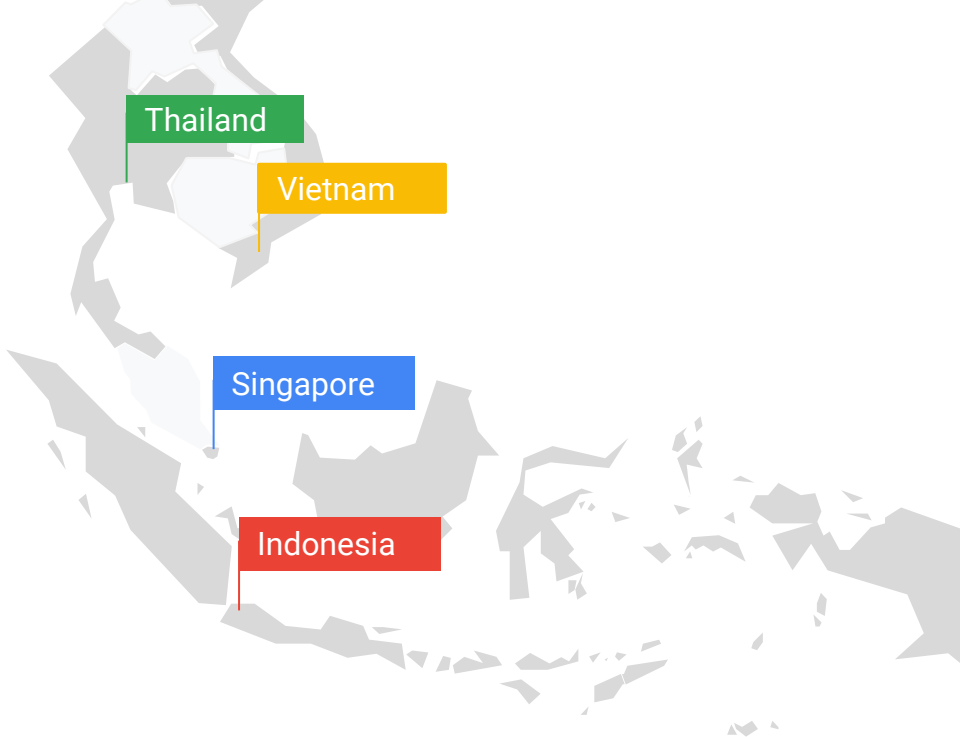
- How K8s and Kubeflow empower companies like GOJEK to build ML platforms
- Demo - Using Feast + Kubeflow to build & deploy from notebooks
- What is Kubeflow





Our scale

Operating in 4 countries throughout Southeast Asia



125m+
app downloads



+400k
merchants



4
countries



2m+
drivers

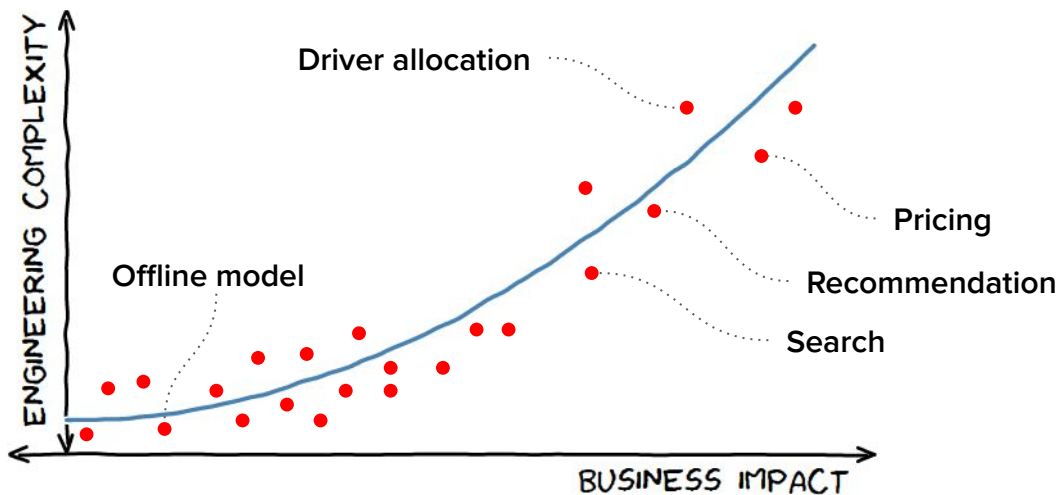


100m+
monthly bookings

Our Data



ML PROJECTS



Data science requirements

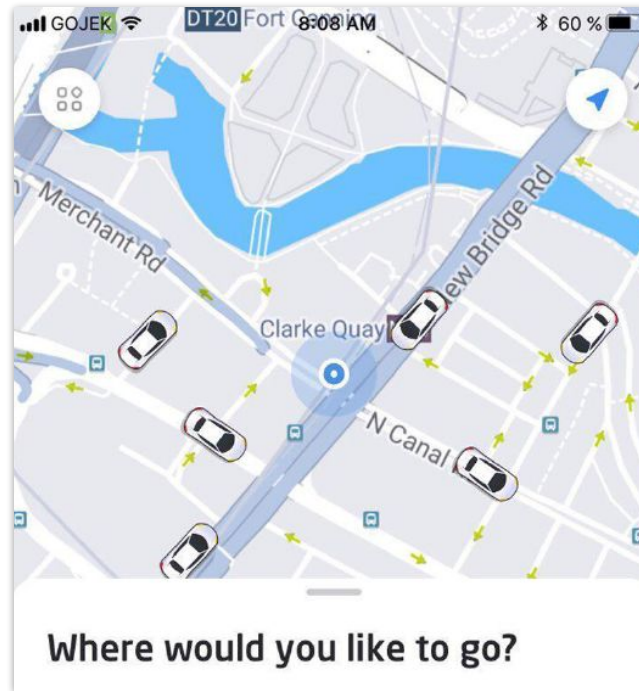
- High abstraction
- Rapid iteration and experimentation
- Customizable workflows

Engineering requirements

- Integrates with existing systems (requires escape hatches)
- Able to operate at scale
- Easy to maintain and debug
- Easy to extend and build on

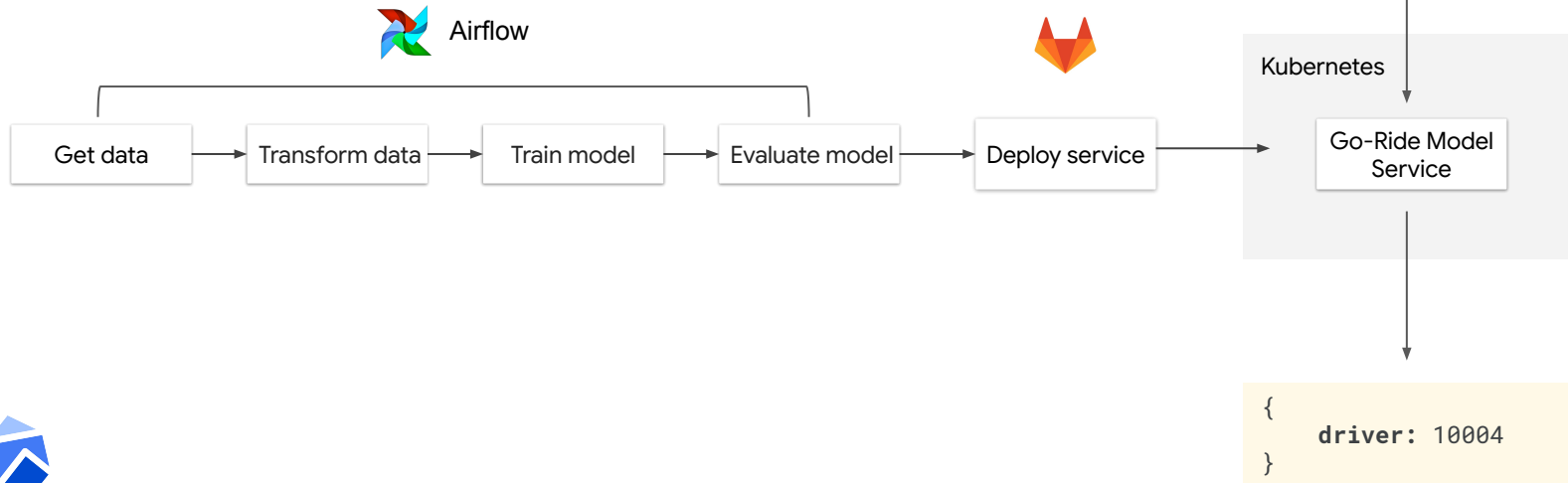


Finding the right driver



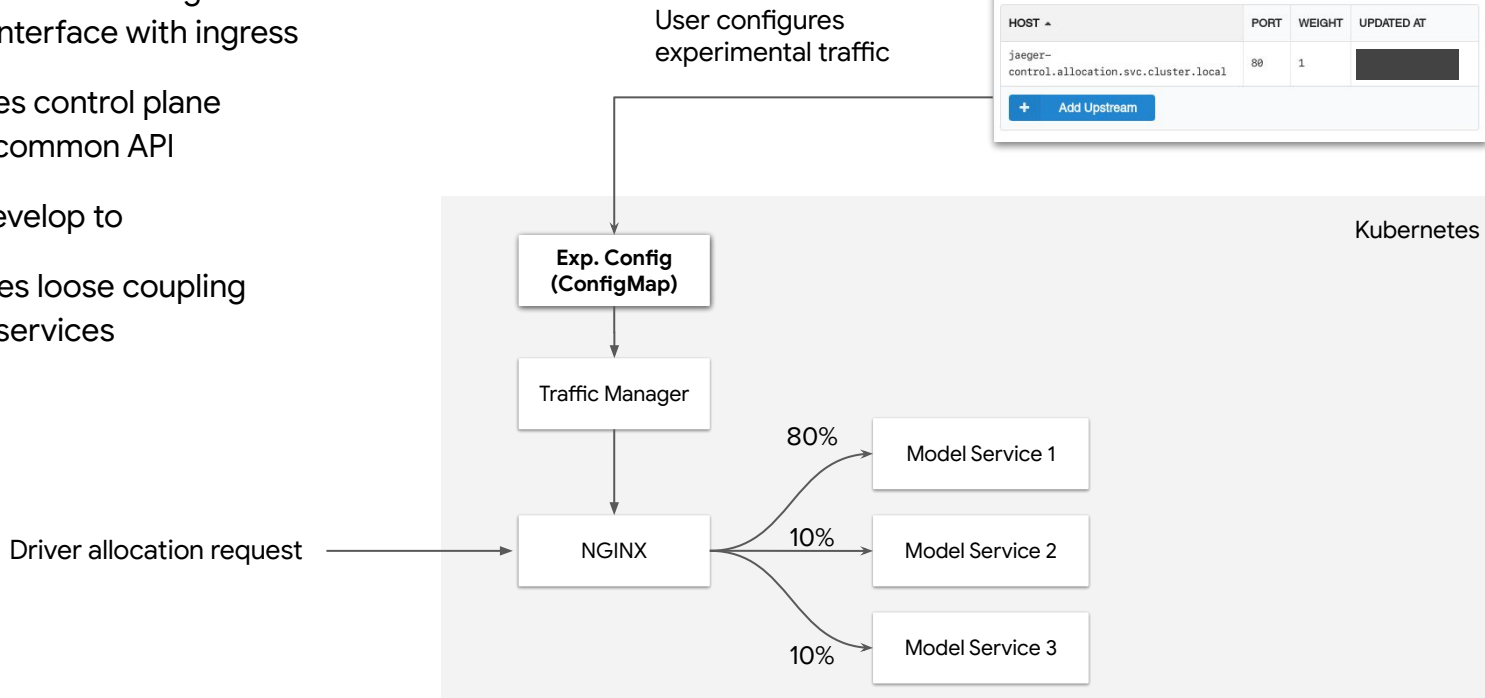
Let's use Kubernetes

- Dependency management
- Container orchestration
- Process isolation



ML on K8s: Experimentation

- Experimentation manager required interface with ingress
- Kubernetes control plane provides common API
- Easy to develop to
- Encourages loose coupling between services



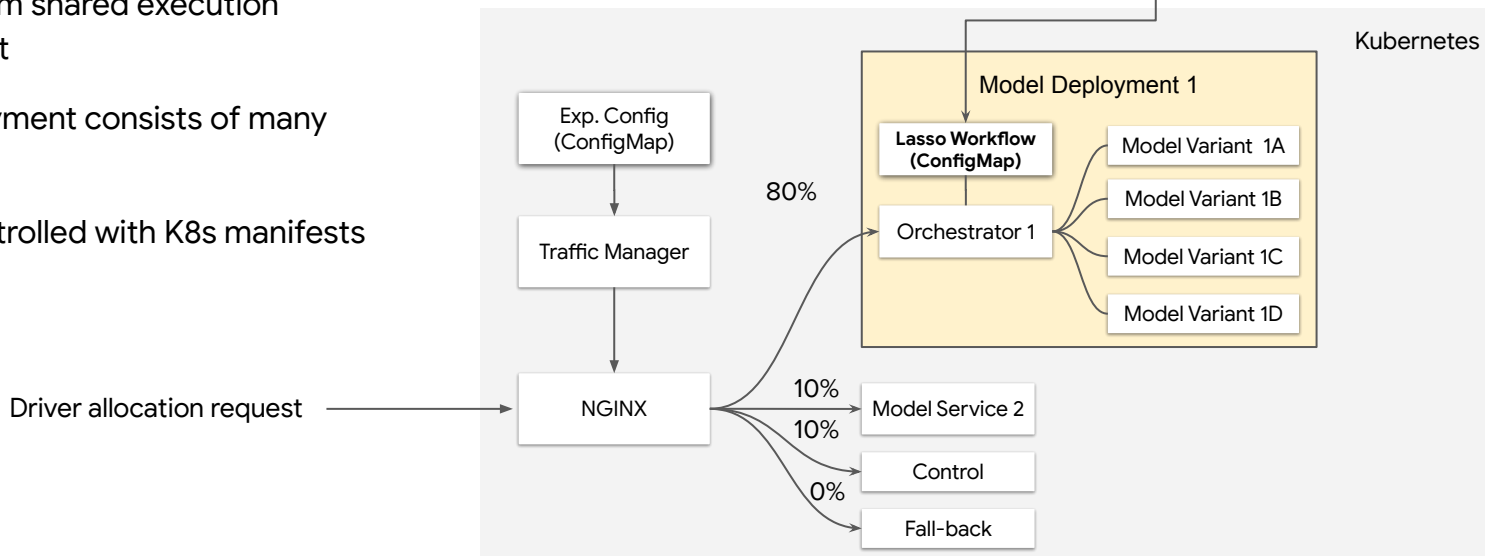
ML on K8s: Orchestration

- Lasso orchestrates services through workflow configurations (see also Seldon)
- Benefits from shared execution environment
- Each deployment consists of many models
- Version controlled with K8s manifests

Lasso workflow
YAML

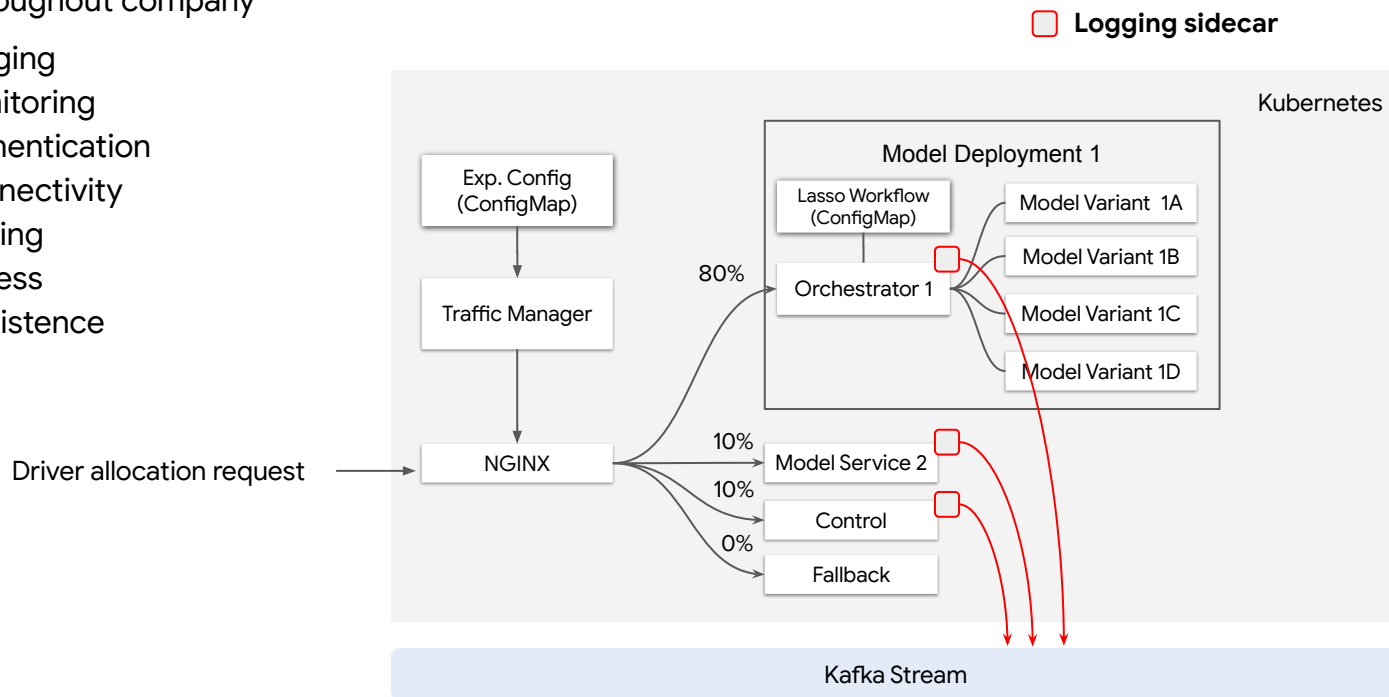
```
apiVersion: v1
name: ja-
endpoint: /ja-
timeout: 140

tasks:
  control:
    type: http
    inputs:
      address: http://ja- control.allocation.svc.cluster.local/predict
      timeout: 140
      retries: 0
      method: POST
      body: '{{ .request }}'
    output:
      path: response.body
      write: ignoreIfExists
```



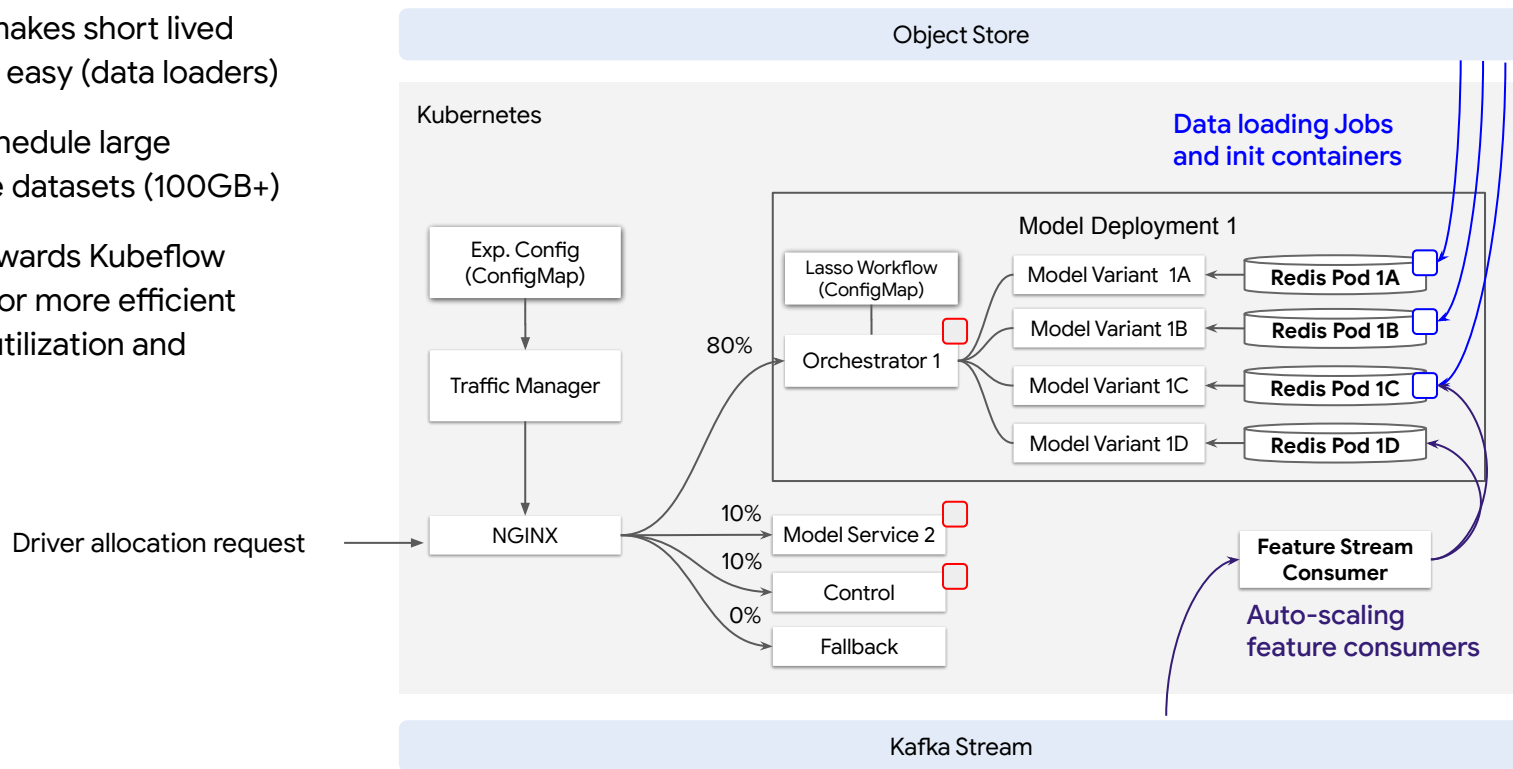
ML on K8s: Economies of scale

- Leverage components and tooling throughout company
 - Logging
 - Monitoring
 - Authentication
 - Connectivity
 - Tracing
 - Access
 - Persistence

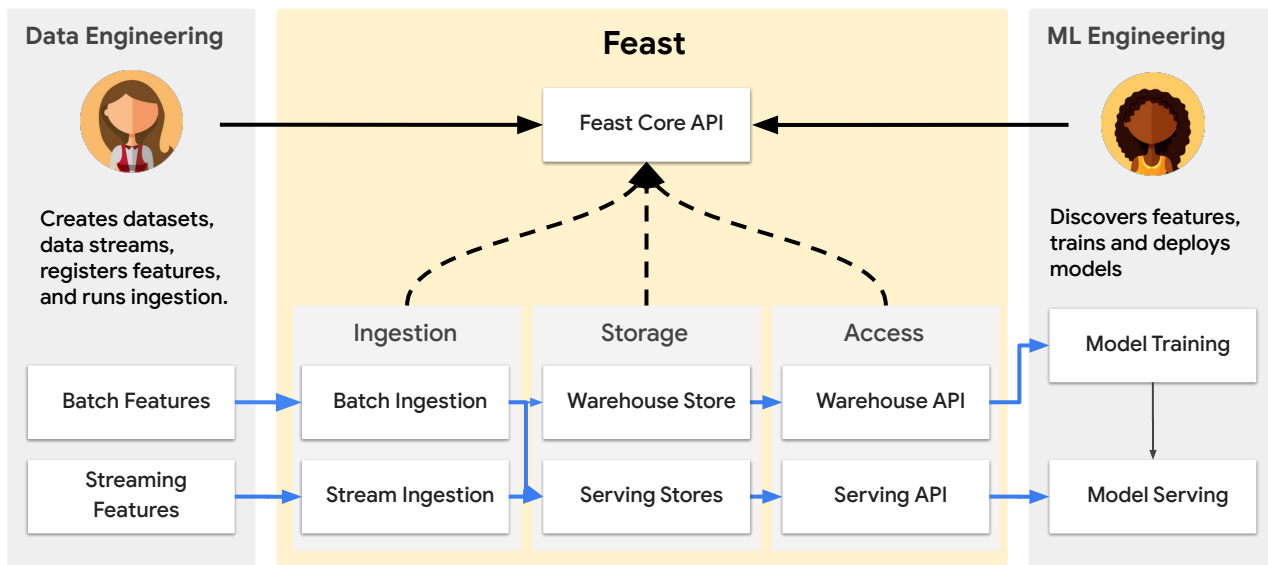


ML on K8s: Workloads

- Jobs API makes short lived processes easy (data loaders)
- Able to schedule large immutable datasets (100GB+)
- Moving towards Kubeflow Pipelines for more efficient resource utilization and tracking



ML on K8s: Feature Store (Feast)

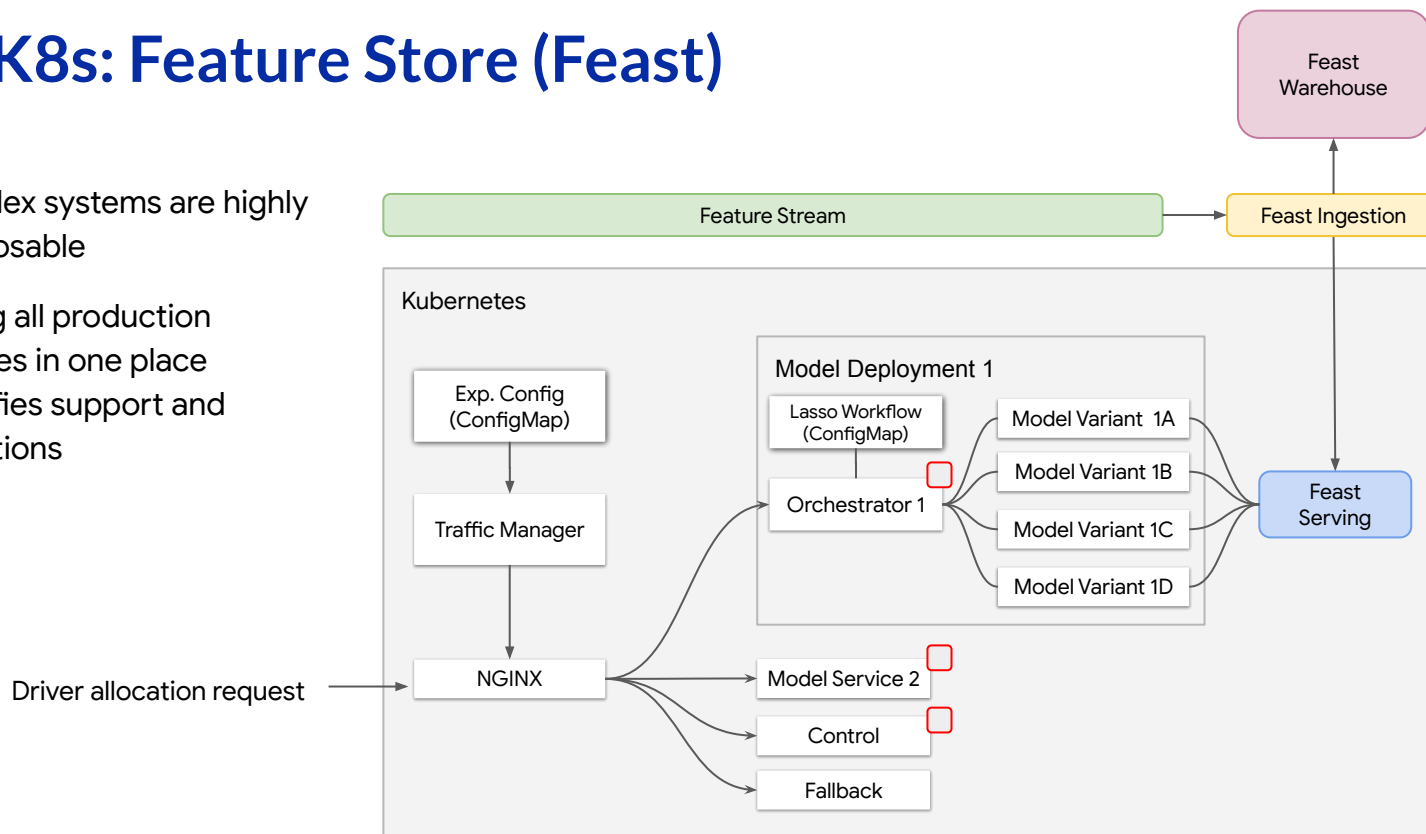


- Encourages feature discovery and reuse through centralization
- Prevents training-serving skew
- Provide scalable storage of feature data for serving and training



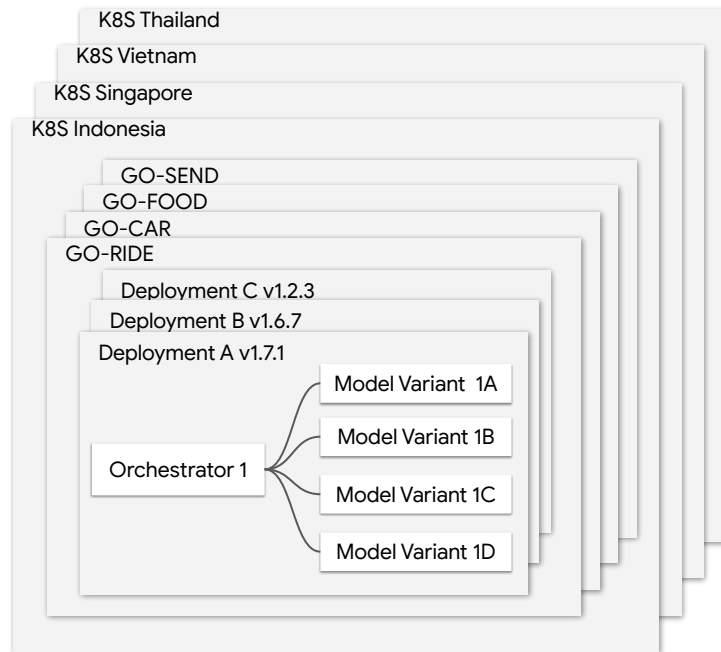
ML on K8s: Feature Store (Feast)

- Complex systems are highly composable
- Having all production services in one place simplifies support and operations



ML on K8s: Rapid expansion

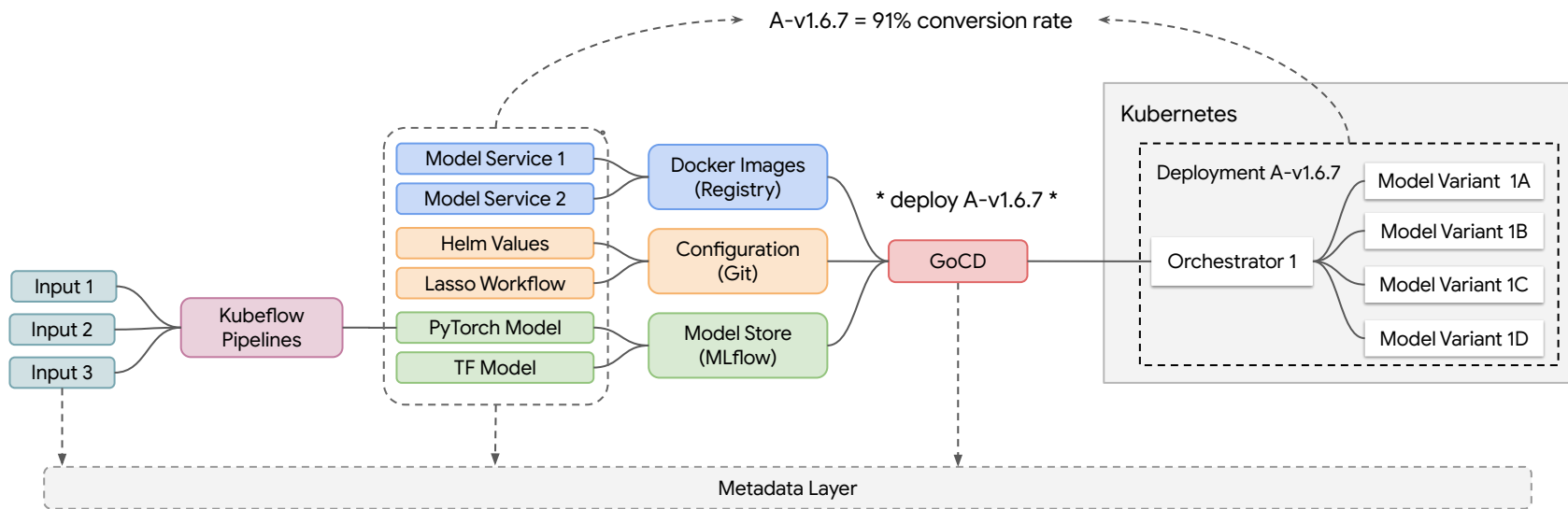
- Scaling to new markets required very large service-to-engineer ratio
- GitOps approach allowed us to increase our leverage and expand to new markets
- Terraform for all infrastructure (even Helm deployments)
- CD for model deployments



<http://bit.ly/gojek-olympus> (by Ravi Suhag)

ML on K8s + Kubeflow: Traceability

- Generalize & parameterize complete ML life cycle
- Track artifact combination as deployment version
- Measure version against experiment



ML on K8s: The good parts

- Large ecosystem of ML frameworks built on top of Kubernetes
- Consistent API simplifies developing complex systems
- Workloads benefit from intelligent scheduling, resource utilization, and dependency management.
- Single production environment simplifies operations
- GitOps allows for high leverage and portability
- Artifact based versioning and tracking allows for traceable experiments



ML on K8s: The rough parts

- Multi tenancy
- Stateful systems aren't there yet
- Leaky abstractions (CRDs / annotations exposed to users)



ML on K8s: What's next

- Simplify the end-to-end user experience
- Metadata tracking (Kubeflow)
- Istio integration





Kelsey Hightower 

@kelseyhightower



Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.

10:04 PM · Nov 27, 2017

237 Retweets **676** Likes



**Kubeflow is an open,
Kubernetes native
platform for ML**



**Make it Easy for Everyone
to **Develop, Deploy** and **Manage**
Portable, Distributed ML
on Kubernetes**



Demo setup

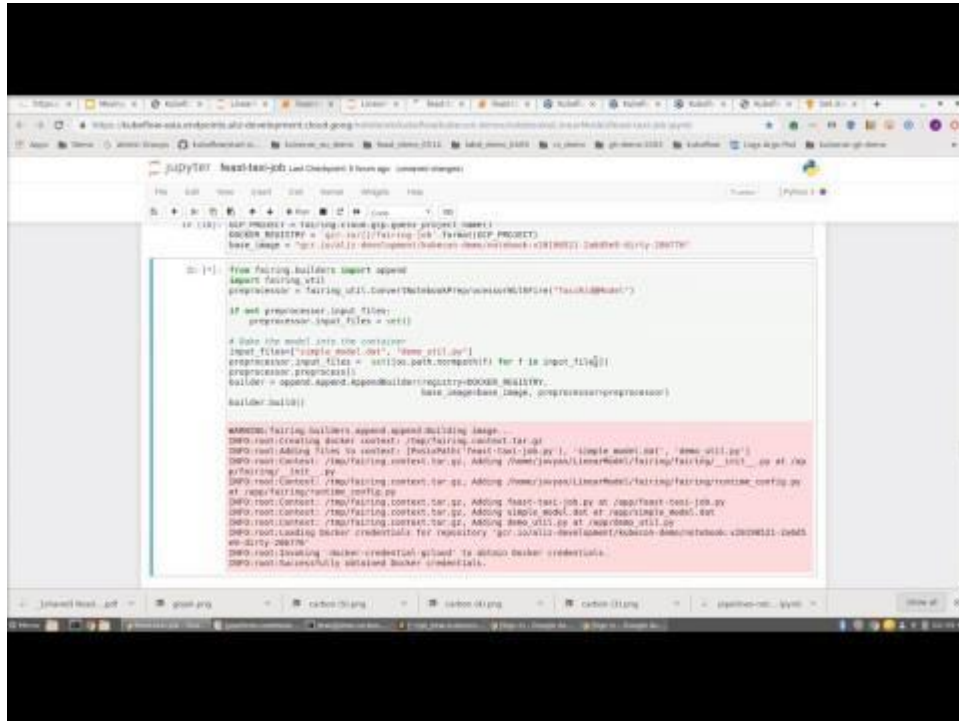


Using Feast and Kubeflow to build and deploy models

- Fetch data from feast for training
- Develop models in a notebook
- Deploy models on K8s
- Fetch data at serving time for inference



Video of Demo



```
from fairing.builders import opendocker
import logging

logger = logging.getLogger('fairing_builder')

if not processor.input_files:
    processor.input_files = []

# Take the model into the container
input_files = ['sample_model.dat', 'demo_util.py']
processor.input_files = set(input_files)
processor.process_files()
builder = opendocker.Builder(
    base_image=base_image, processor=processor)
builder.build()

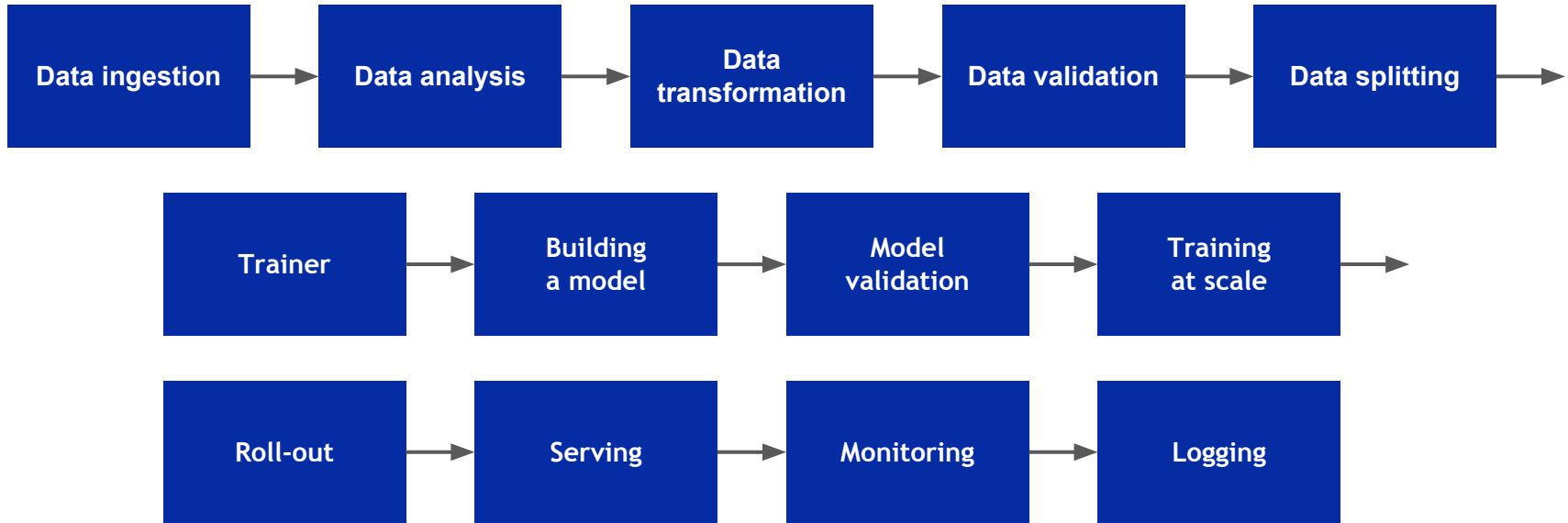
#INFO:fairing.builders.opendocker:Building image -
INFO:root:Creating docker context: /tmp/fairing-context.tar.gz
INFO:root:Adding files to context: ['sample_model.dat', 'demo_util.py']
INFO:root:Context: /tmp/fairing-context.tar.gz, Adding /home/jupyter/linear-model/fairing/fairing_init.py at /x
x/fairing_init.py
INFO:root:Context: /tmp/fairing-context.tar.gz, Adding /home/jupyter/linear-model/fairing/fairing_runtime_config.py
at /tmp/fairing-runtime_config.py
INFO:root:Context: /tmp/fairing-context.tar.gz, Adding /tmp/fairing-context.tar.gz at /tmp/fairing-context.tar.gz
INFO:root:Context: /tmp/fairing-context.tar.gz, Adding /tmp/fairing-context.tar.gz at /tmp/fairing-context.tar.gz
INFO:root:Loading Docker credentials for repository 'gcr.io/google-containers'
INFO:root:Loading Docker credentials from 'gcr.io/google-containers'
INFO:root:Successfully obtained Docker credentials.
```



Kubeflow



ML Development Workflow



**Kubeflow makes it easy to
run these steps on
Kubernetes**



Core Tenets

- Kubeflow makes it easy to run ML applications on Kubernetes
 - e.g. notebooks, HP tuning, pipelines, model servers, etc...
- **Composable** - Use the libraries/frameworks of your choice
- **Scalable** - number of users & workload size
- **Portable** - on prem, public cloud, local



Hyperparameter Tuning

The screenshot shows the 'Create StudyJob' interface in the Kubeflow UI. On the left, there is a form with fields for 'Study Name', 'Owner', 'Optimization Type', 'Optimization Goal', 'Objective Value Name', 'Metrics (space separated)', and 'Request Count'. On the right, the 'Generated StudyJob YAML' is displayed as a code block. The YAML includes fields for 'apiVersion', 'kind', 'metadata', 'spec', 'studyName', 'owner', 'optimizationType', 'objectiveValueName', 'optimizationGoal', 'metricsNames', 'parameterConfigs', 'requestCount', 'suggestionSpec', 'workerSpec', 'metricsCollectorSpec', and 'templatePath'.

Pipelines

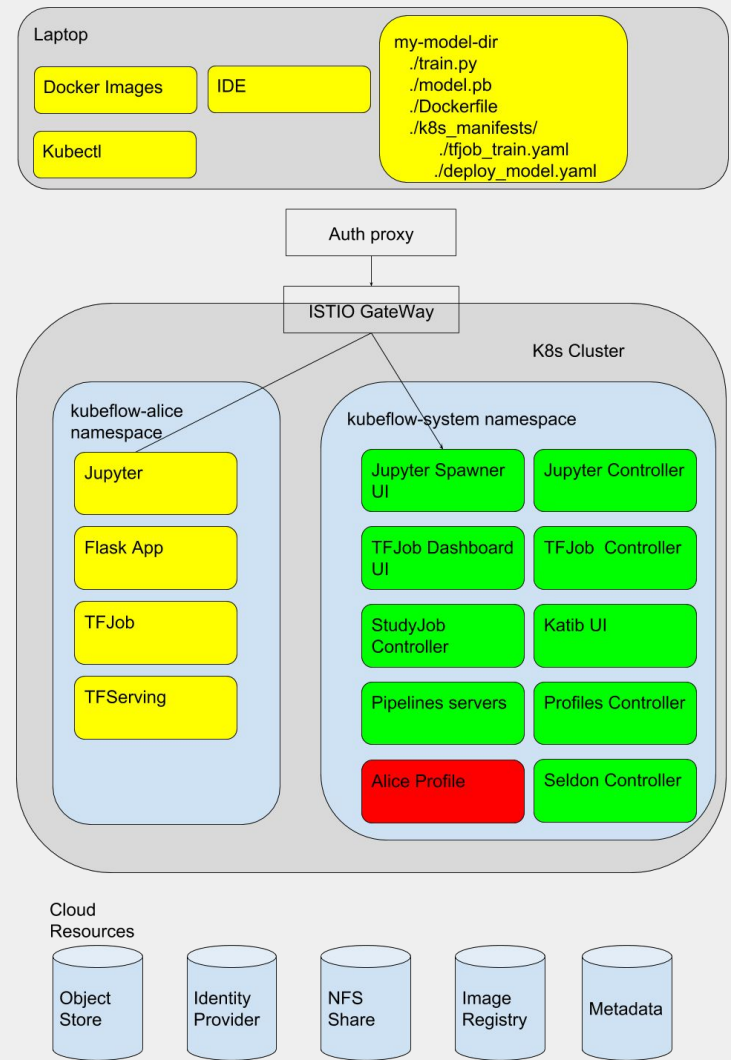
The screenshot shows the 'Pipelines' interface in the Kubeflow UI. It features a sidebar with navigation options like 'Pipelines', 'Experiments', and 'Archive'. The main area displays a table of pipeline definitions with columns for 'Pipeline name', 'Description', and 'Uploaded on'. The table lists several sample pipelines, including 'Basic - Condition', 'Basic - Exit Handler', 'Basic - Immediate Value', 'Basic - Parallel Join', 'Basic - Sequential', 'ML - TFX - Taxi Tip Predict...', and 'ML - XGBoost - Training with...'. Each row includes a checkbox, the pipeline name, a brief description, and the upload date and time.

Kubeflow Architecture



System Diagram

- Kubeflow is a collection of microservices
 - Will use ISTIO for service mesh in 0.6
- Users/teams consume Kubeflow in their own namespace



OSS Momentum!



New PRs Last 28 Days



Unique PR Authors Last 28 Days



Goal: low bar; high ceiling

- Day 0 focus on model development
 - Use UIs to launch notebooks
 - Python SDK (fairing) for training / deploying models
- Day 0 start with the infrastructure (Kubernetes, ISTIO, etc...) that you can ride into production
- Day N leverage K8s to scale
 - Use the same infrastructure as non-ML applications
 - Build a single infrastructure team



The screenshot displays the Kubeflow Notebook Servers management interface. At the top, the Kubeflow logo and 'Namespace kubeflow' are visible. Below this, a search bar shows 'Select Namespace kubeflow'. The main content area is titled 'Notebook Servers' and includes a '+ NEW SERVER' button. A table lists the server details:

Status	Name	Created	Image	CPU	Memory	Volumes	Actions
✓	jlewi	1 day ago	tensorflow-1.13.1-notebook-cpu	32	32Gi		CONNECT

Below the table is a 'New Notebook Server' form with the following sections:

- Name:** Specify the name of the Notebook Server and the Namespace it will belong to. Notebook Server's Name: kubeflow
- Image:** A starter Jupyter Docker image with a baseline deployment and typical ML packages. Standard (Selected) Custom. Image: gcr.io/kubeflow-images-public/tensorflow-1.13.1-notebook-cpu:v5.0
- CPU:** Specify the total amount of CPU reserved by your Notebook Server. For CPU-intensive workloads, you can choose more than 1 CPU (e.g. 1.5). CPU: 0.5
- Memory:** Specify the total amount of RAM reserved by your Notebook Server (e.g. 2 GB). Memory: 1.0GB
- Workspace Volume:** Configure the Volume to be mounted on your personal Workspace. For example, to create an empty Workspace, use notebook-workspace-1b. /home/jupyter/Reader/1ed63c
- Data Volumes:** Configure the Volumes to be mounted on your Datasets. For example, to create an empty Data Volume, use data-volume-1. /home/jupyter/data-volume-1. Reader/1ed63c
- Extra Resources:** Reserve additional resources. For example, to reserve 2 GPUs, use 'nvidia.com/gpu': 2

At the bottom of the form are 'SHOW' and 'CANCEL' buttons.

Call To Action

- Install Kubeflow - <https://www.kubeflow.org/docs/started/>
- Install Feast - <https://github.com/gojek/feast/blob/master/docs/install.md>
- Try Fairing + Feast:
 - <https://github.com/gabrielwen/LinearModel>



Kubeflow Talks ([bit.ly/kf calendar](https://bit.ly/kf_calendar))

- **Tutorial Introduction to Pipelines** - *Tuesday May 21 14:00-15:25*; Michelle Casbon, Dan Sanche, Dan Anghel & Michal Zylinski Google (<https://sched.co/MPgr>)
- **Kubeflow BOF** - *Tuesday May 21 15:55-16:30*; David Aronchick, Microsoft & Yaron Haviv, Iguazio (<https://sched.co/PiUF>)
- **Toward Kubeflow 1.0, Bringing a Cloud Native Platform for ML to Kubernetes** - *Wednesday May 22 11:55 - 12:30*; David Aronchick, Microsoft & Jeremy Lewi Google (<https://sched.co/MPax>)
- **Building Cross-Cloud ML Pipelines with Kubeflow with Spark & TensorFlow** - *Wednesday May 22 14:00 - 14:35*; Holden Karau, Google & Trevor Grant, IBM (<https://sched.co/MPaZ>)
- **Managing Machine Learning Pipelines In Production with Kubeflow with Devops** - *Wednesday May 22 14:40-14:35* - David Aronchick, Microsoft (<https://sched.co/MPaZ>)
- **Large Scale Distributed Deep Learning with Kubernetes Operators** - *Wed May 22 15:55 - 16:30*; Yuan Tang, Ant Financial & Yong Tang MobileIron (<https://sched.co/MPaT>)
- **Moving People and Products with Machine Learning on Kubeflow** - *Thursday May 23 14:00 -14:35*; Jeremy Lewi, Google & Willem Pienaar, GO-JEK (<https://sched.co/MPac>)





Kubeflow

Thank You

www.kubeflow.org

github.com/gojek/feast

github.com/gabrielwen/LinearModel