Machine Learning with Kubernetes

Yaniv Donenfeld
AI/ML Solutions
Container Services, AWS

Jiaxin Shan
Software Development Engineer
Container Services, AWS

“Cloud has removed so many of the barriers to experimenting and innovating with AI that even risk-adverse businesses are making it part of their strategies.”

- Yaniv Donenfeld, just now.

40% of digital transformation initiatives supported by AI in 2019 *—IDC 2018*

Our mission at AWS

Put machine learning in the hands of every developer

The AWS ML Stack

Broadest and deepest set of capabilities

AI Services

VISION			SPEECH		LANGUAGE		CHATBOTS	FORECASTING	RECOMMENDATIONS
REKOGNITION IMAGE	REKOGNITION VIDEO	TEXTRACT	POLLY	TRANSCRIBE	TRANSLATE	COMPREHEND & COMPREHEND MEDICAL	LEX	FORECAST	PERSONALIZE

ML Services

Amazon SageMaker	Ground Truth	Notebooks	Algorithms + Marketplace	Reinforcement Learning	Training	Optimization	Deployment	Hosting

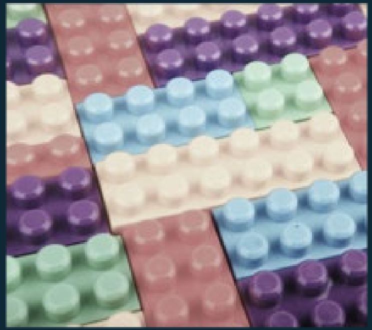
ML Frameworks + Infrastructure

FRAMEWORKS	INTERFACES
TensorFlow mxnet PYTORCH	GLUON Keras

INFRASTRUCTURE								
EC2 P3 & P3DN	EC2 G4 EC2 C5	FPGAS	DL CONTAINERS & AMIs	ELASTIC CONTAINER SERVICE	ELASTIC KUBERNETES SERVICE	GREENGRASS	ELASTIC INFERENCE	INFERENTIA

aws

Why Machine Learning on Kubernetes?



Composability

Portability

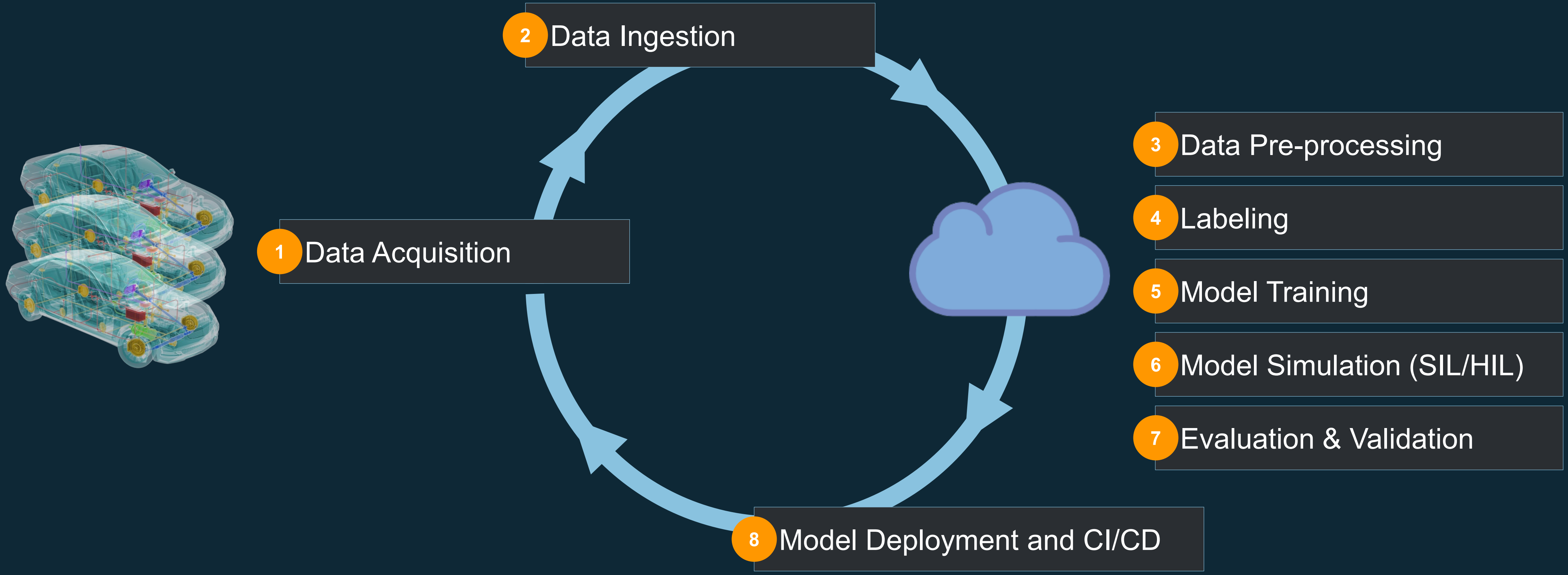
Scalability

http://www.shutterstock.com/gallery-635827p1.html

Use Case #1: Large Scale ML

Autonomous Vehicles Workloads

Typical Autonomous Vehicle Development Workflow



1 Data Acquisition

2 Data Ingestion

3 Data Pre-processing

4 Labeling

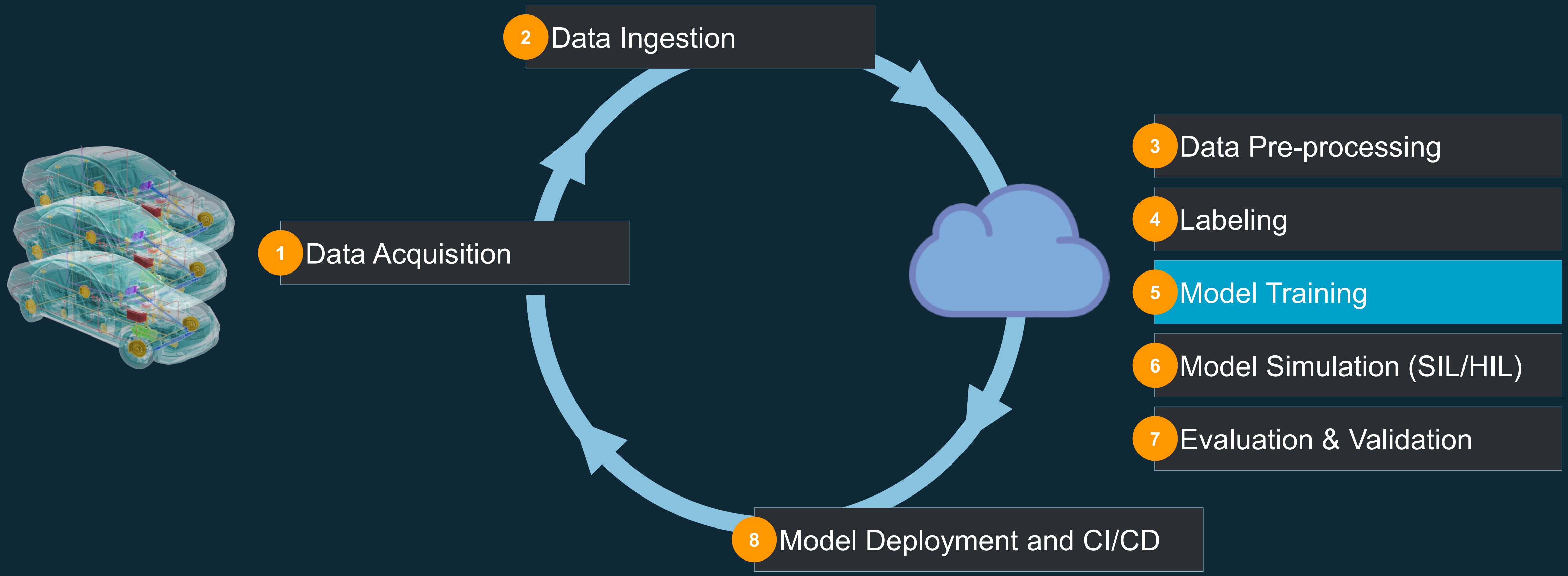
5 Model Training

6 Model Simulation (SIL/HIL)

7 Evaluation & Validation

8 Model Deployment and CI/CD

Typical Autonomous Vehicle Development Workflow



1 Data Acquisition

2 Data Ingestion

3 Data Pre-processing

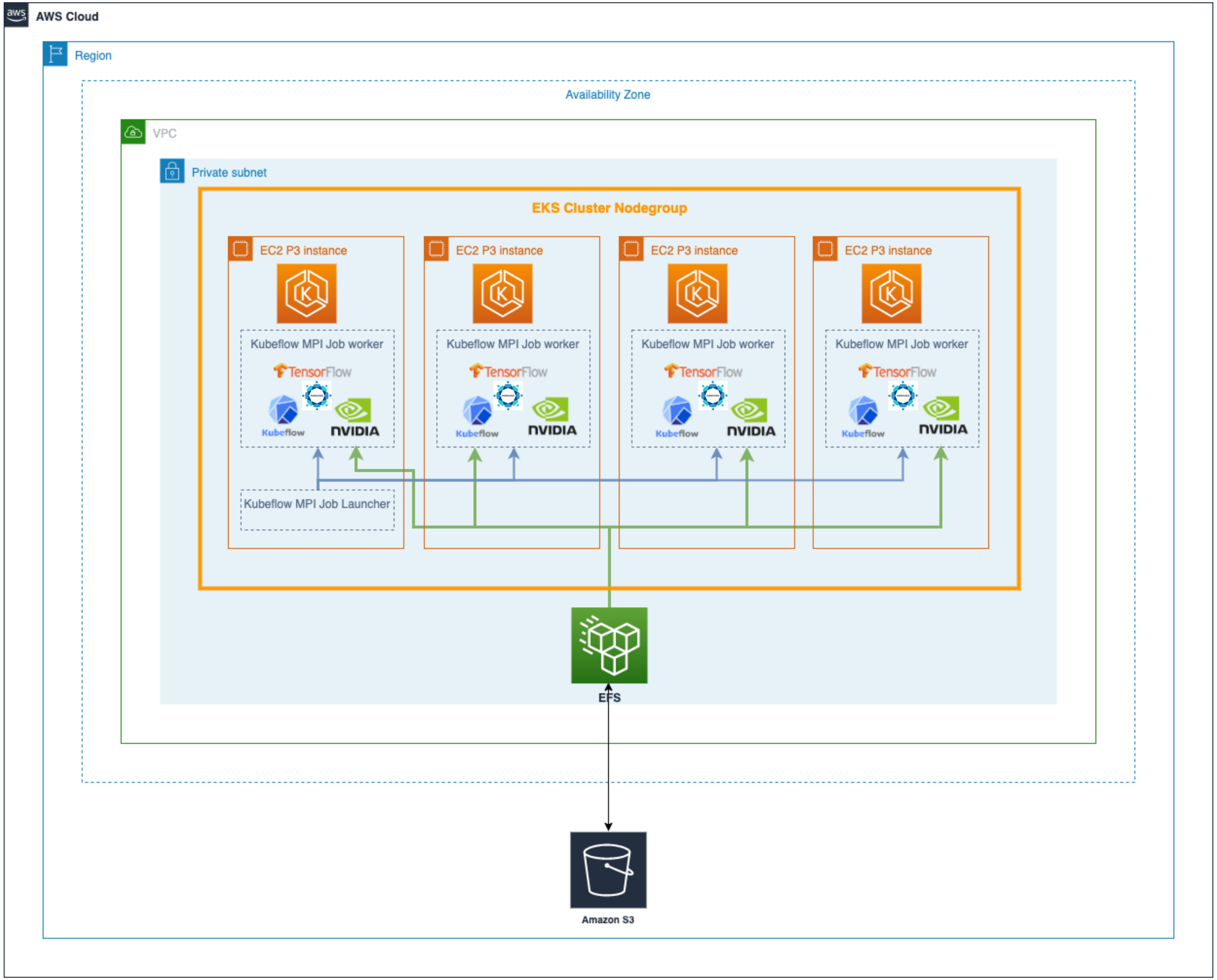
4 Labeling

5 Model Training

6 Model Simulation (SIL/HIL)

7 Evaluation & Validation

8 Model Deployment and CI/CD


AWS Cloud
Region
Availability Zone
VPC
Private subnet
EKS Cluster Nodegroup
EC2 P3 instance
EC2 P3 instance
EC2 P3 instance
EC2 P3 instance
Kubeflow MPI Job worker
Kubeflow MPI Job worker
Kubeflow MPI Job worker
Kubeflow MPI Job worker
TensorFlow
Kubeflow
NVIDIA
Kubeflow MPI Job Launcher
EFS
Amazon S3

- Distributed Training Challenges

- Single GPU code → multiple

Horovod + MPIJob (or TFJob)

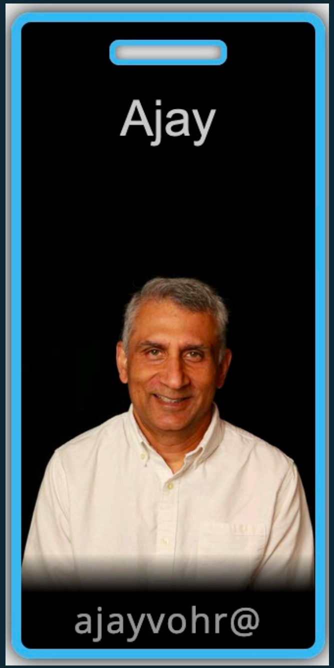
- Dataset Copying time
- Dataset Sharing and Reuse

Use FSx Lustre / EFS

👉 Built-in CSI driver with S3 integration

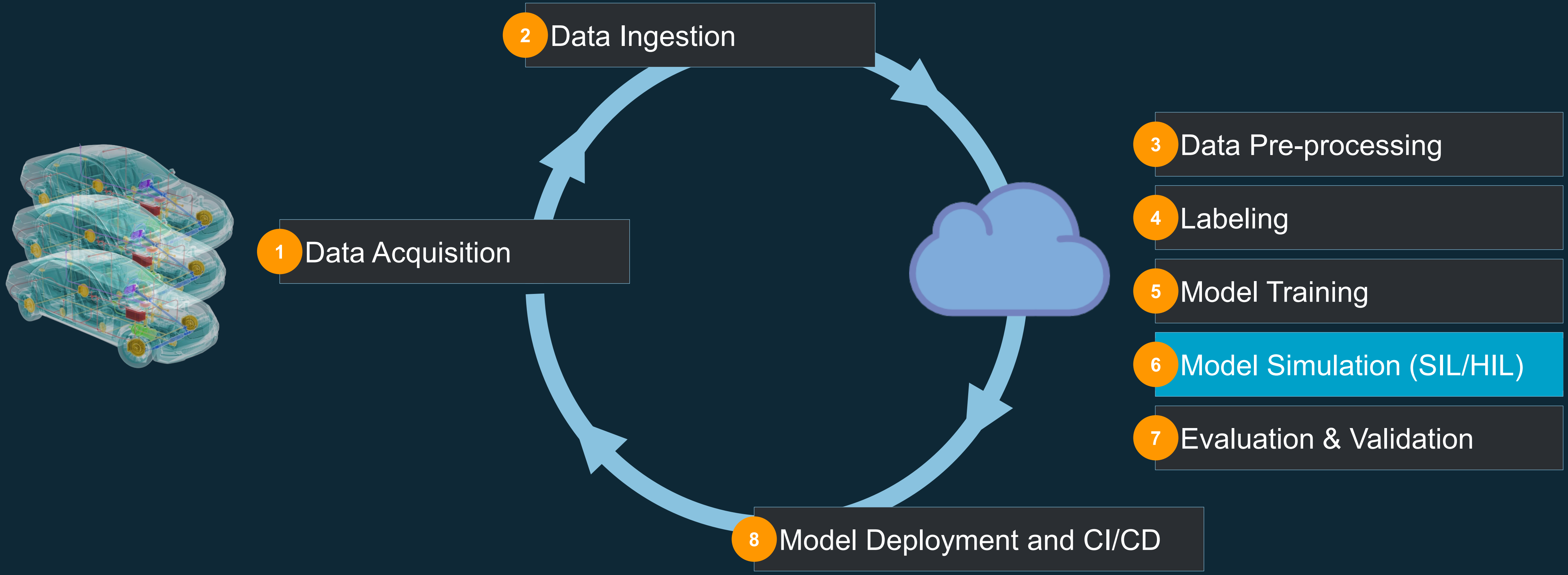
Want to Run Distributed Training on EKS?

Distributed TensorFlow training using Kubeflow on Amazon EKS



Ajay Vohra
Principal SA - Vision/AI/ML

Typical Autonomous Vehicle Development Workflow



1 Data Acquisition

2 Data Ingestion

3 Data Pre-processing

4 Labeling

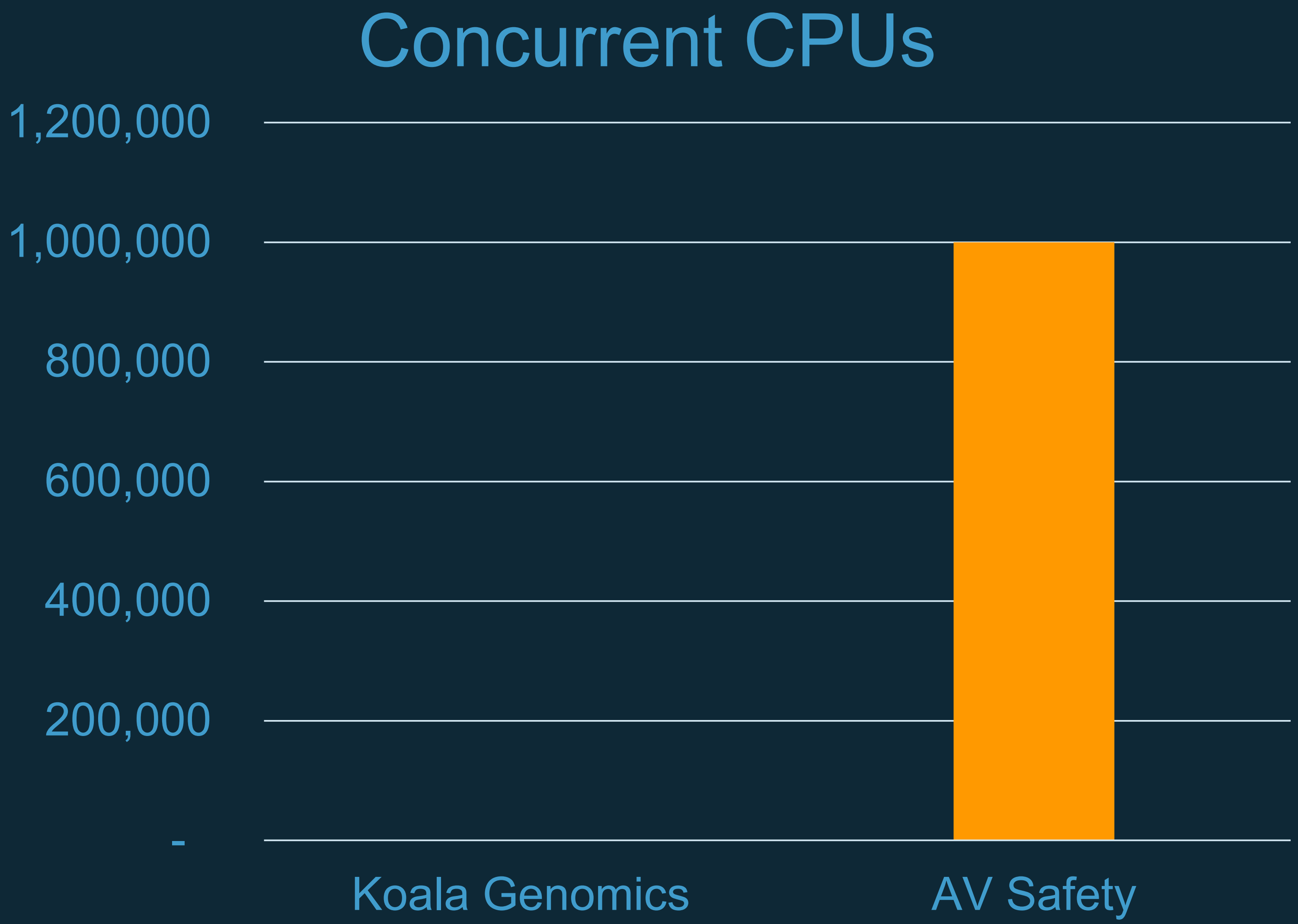
5 Model Training

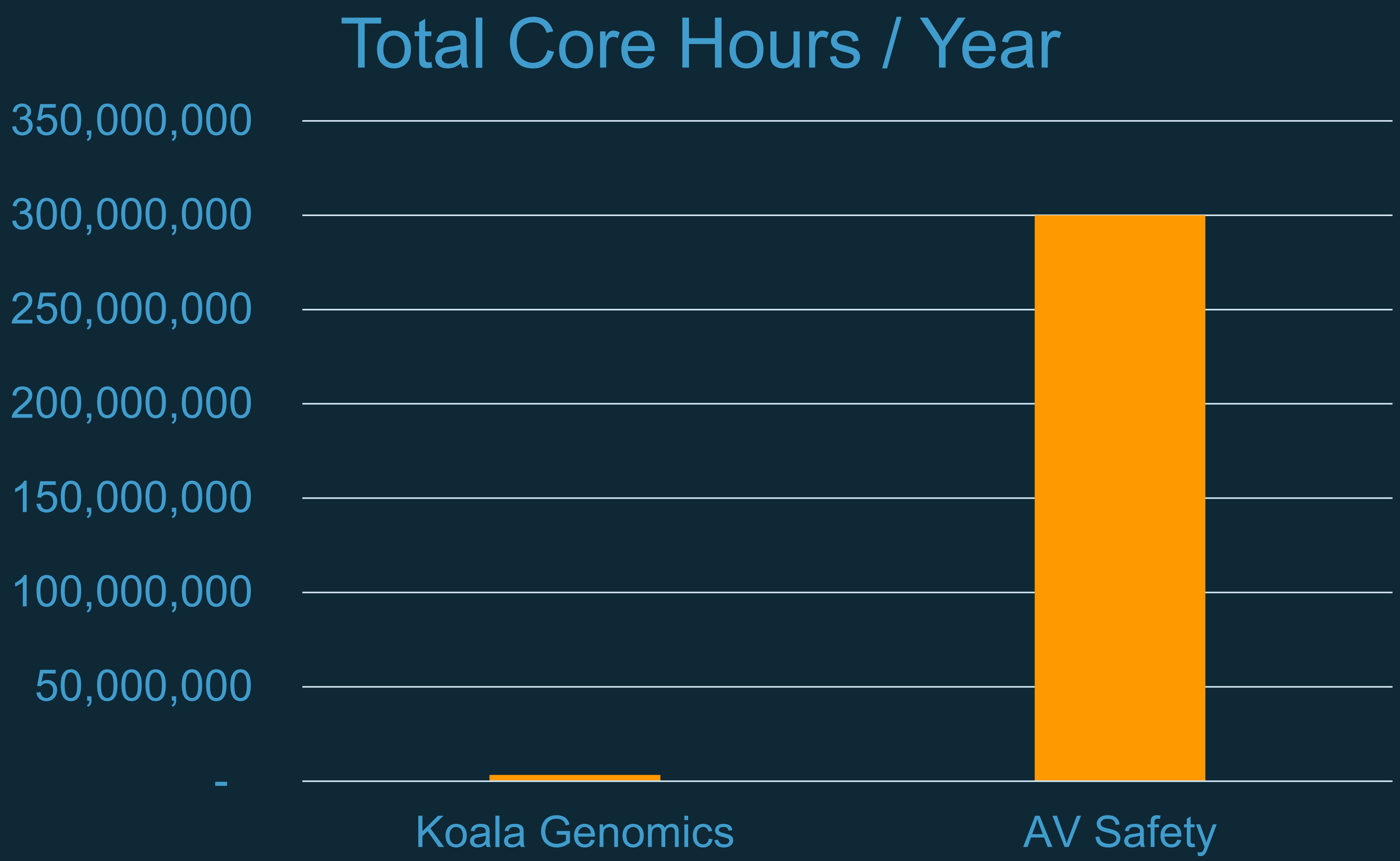
6 Model Simulation (SIL/HIL)

7 Evaluation & Validation

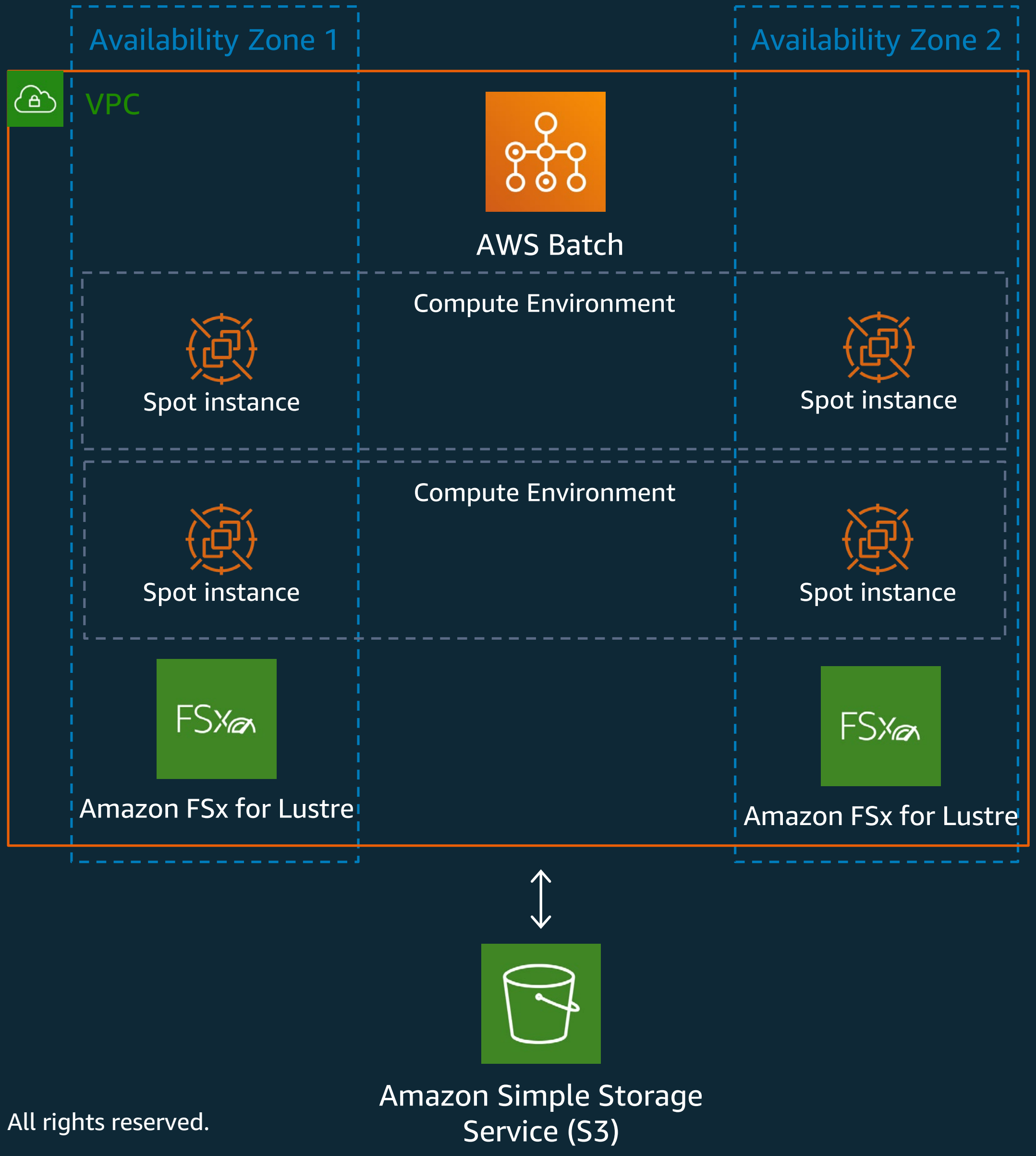
8 Model Deployment and CI/CD

Can you run my workload?

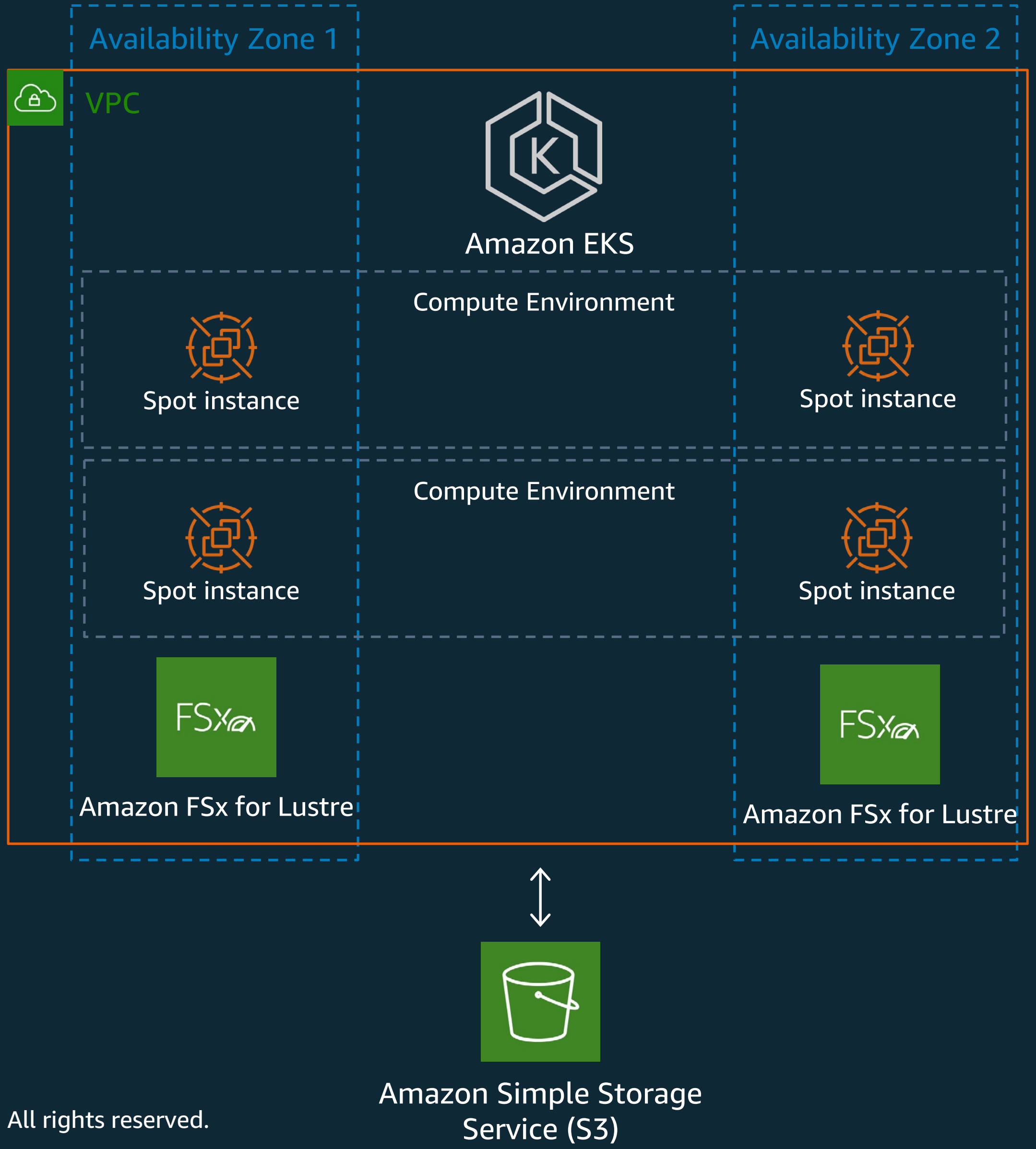

Concurrent CPUs
1,200,000
1,000,000
800,000
600,000
400,000
200,000
-
Koala Genomics
AV Safety


Total Core Hours / Year
350,000,000
300,000,000
250,000,000
200,000,000
150,000,000
100,000,000
50,000,000
-
Koala Genomics
AV Safety

Simulations Architecture



Simulations Architecture



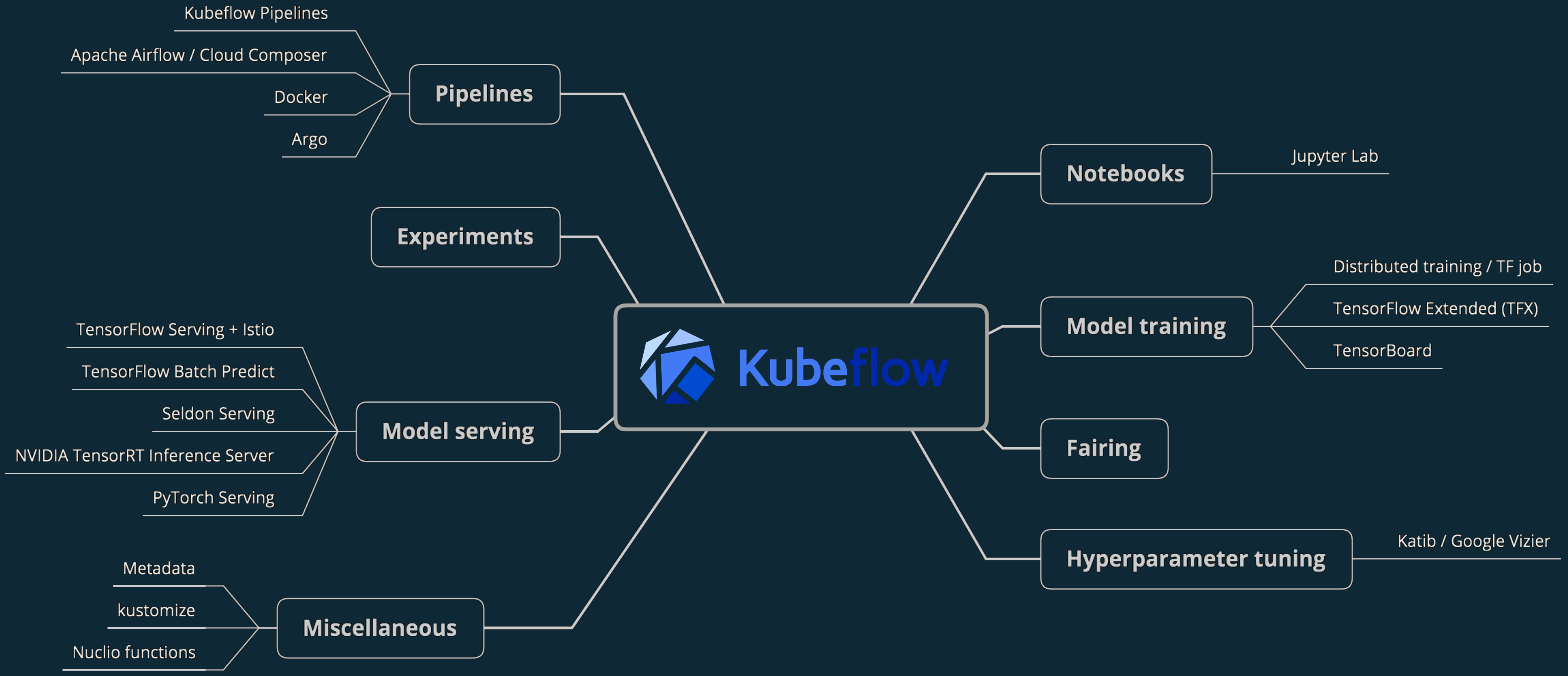
TOP500 – Top 10 Supercomputers in June 2019

Rank / Name	Rmax / Rpeak (Petaflops)
1. Summit	148.600 / 200.795
2. Sierra	94.640 / 125.712
3. Sunway Tahihu Light	93.015 / 125.436
4. Tianhe-2A	61.445 / 100.679
5. Frontera	23.516 / 38.746
6. Piz Daint	21.230 / 27.154
7. Trinity	20.159 / 41.461
8. AI Bridging Cloud Infrastructure	19.880 / 32.577
9. SuperMUCNG	19.477 / 26.874
10. Lassen	18.200 / 23.047

We’re helping our customers run at Supercomputer Scale, targeting the equivalent of **one of the Top 10 largest supercomputers** in the world.

Use Case #2: ML Development Platform

Kubeflow



- Kubeflow
  - Pipelines
    - Kubeflow Pipelines
    - Apache Airflow / Cloud Composer
    - Docker
    - Argo
  - Experiments
  - Model serving
    - TensorFlow Serving + Istio
    - TensorFlow Batch Predict
    - Seldon Serving
    - NVIDIA TensorRT Inference Server
    - PyTorch Serving
  - Miscellaneous
    - Metadata
    - kustomize
    - Nuclio functions
  - Notebooks
    - Jupyter Lab
  - Model training
    - Distributed training / TF job
    - TensorFlow Extended (TFX)
    - TensorBoard
  - Fairing
  - Hyperparameter tuning
    - Katib / Google Vizier

Kubernetes **Multicloud** **AI Hub** **Web UI**

Version 1.1 20190807 @MichalBrys

Jupyter Notebook / JupyterHub

- Build, deploy, and train ML models
- Live code, equations, visualizations, and narrative text
- 40+ programming languages
- Sharing and collaboration

👉 EFS for reusing training data and results

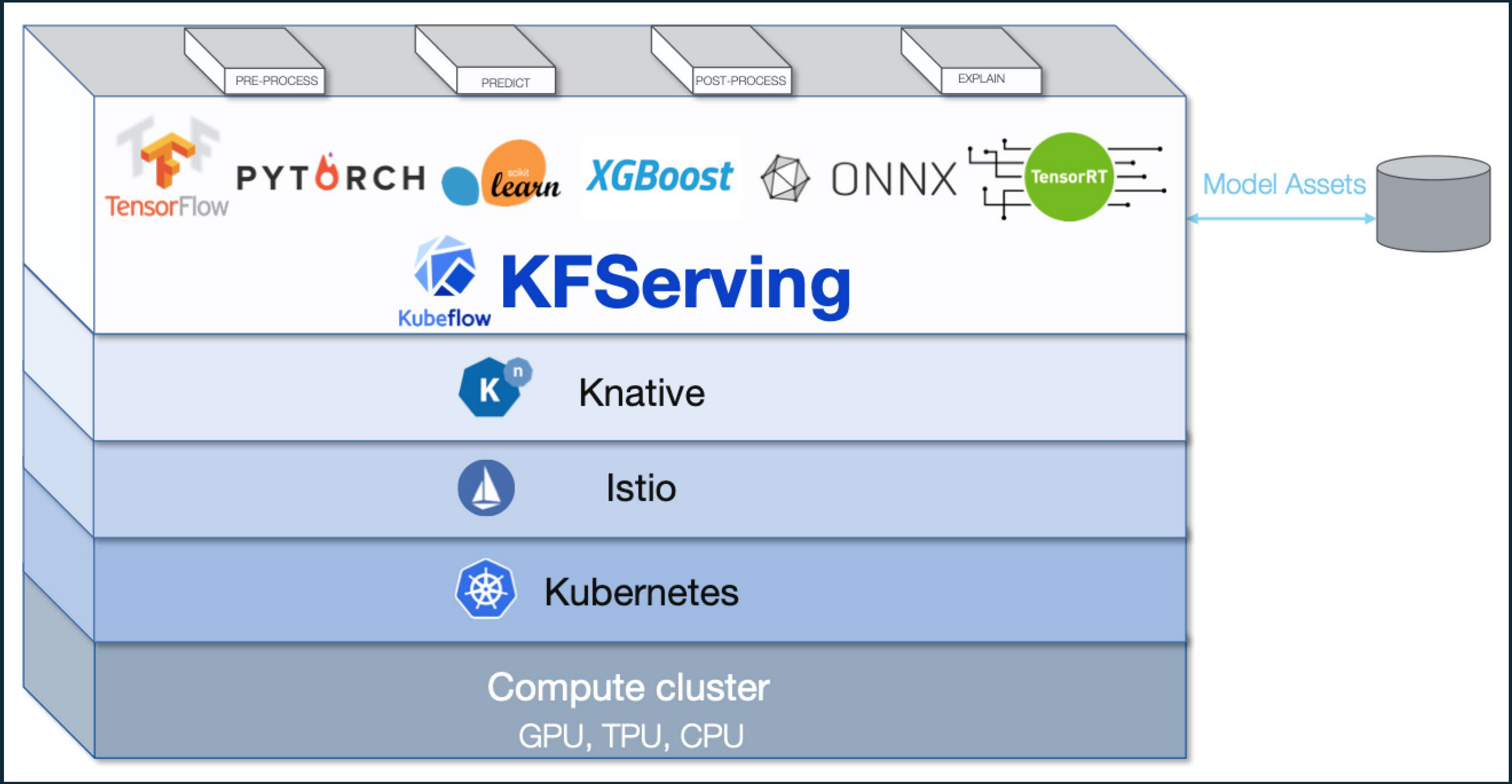
👉 Built-in AWS CLI and ECR support

Kubeflow KFServing

- Simple and pluggable platform for ML inference
- Intuitive and consistent experience
- Serving models on arbitrary frameworks
- e.g. TensorFlow, XGBoost, SciKitLearn
- Encapsulates GPU auto-scaling, canary rollouts

Credits @ellis-bigelow (Kubeflow slack)

Kubeflow KFServing



Pluggable Interface

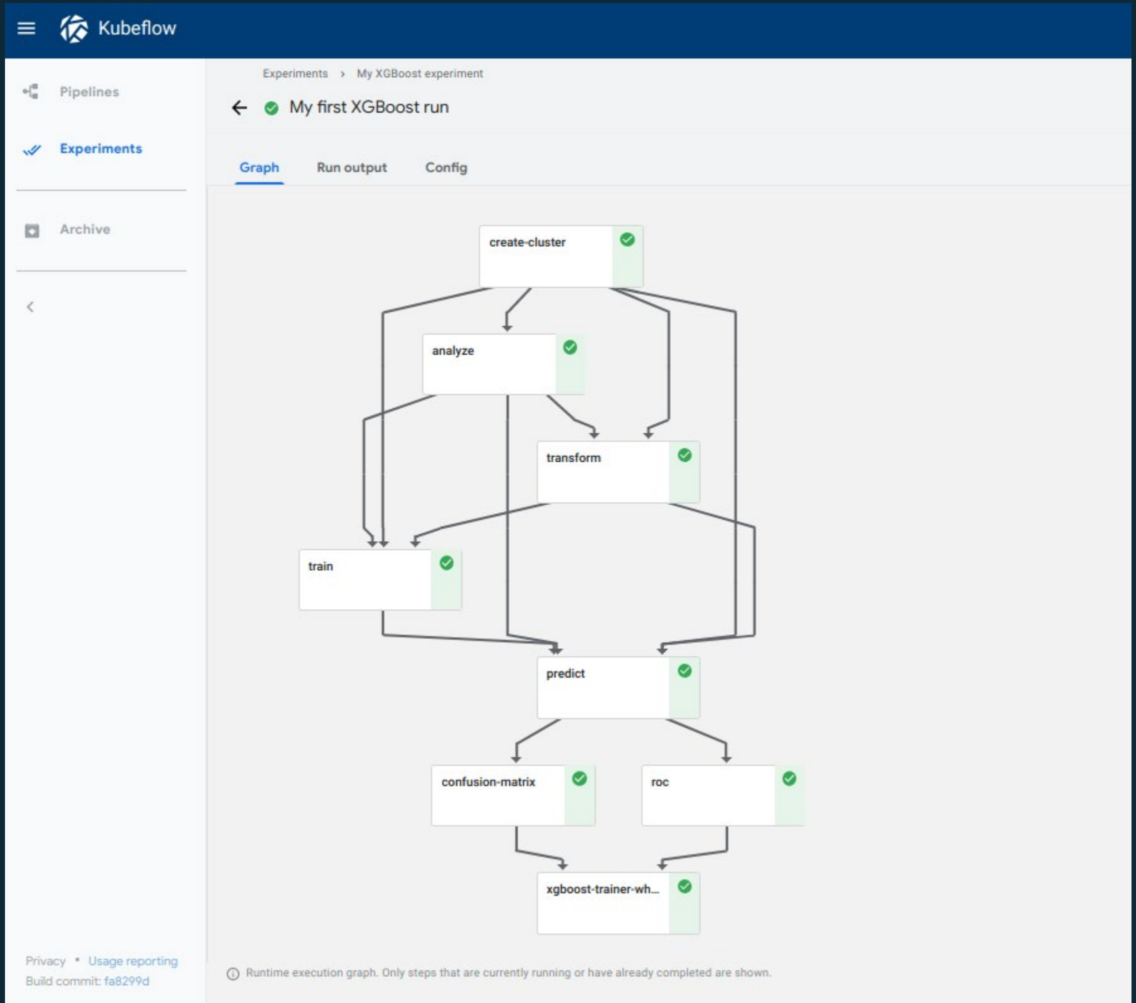
```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "sklearn-iris"
spec:
  default:
    sklearn:
      storageUri: "gs://kfserving-samples/models/sklearn/iris"
```

```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "flowers-sample"
spec:
  default:
    tensorflow:
      storageUri: "gs://kfserving-samples/models/tensorflow/flowers"
```

```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "KFService"
metadata:
  name: "pytorch-cifar10"
spec:
  default:
    pytorch:
      storageUri: "gs://kfserving-samples/models/pytorch/cifar10"
      modelClassName: "Net"
```

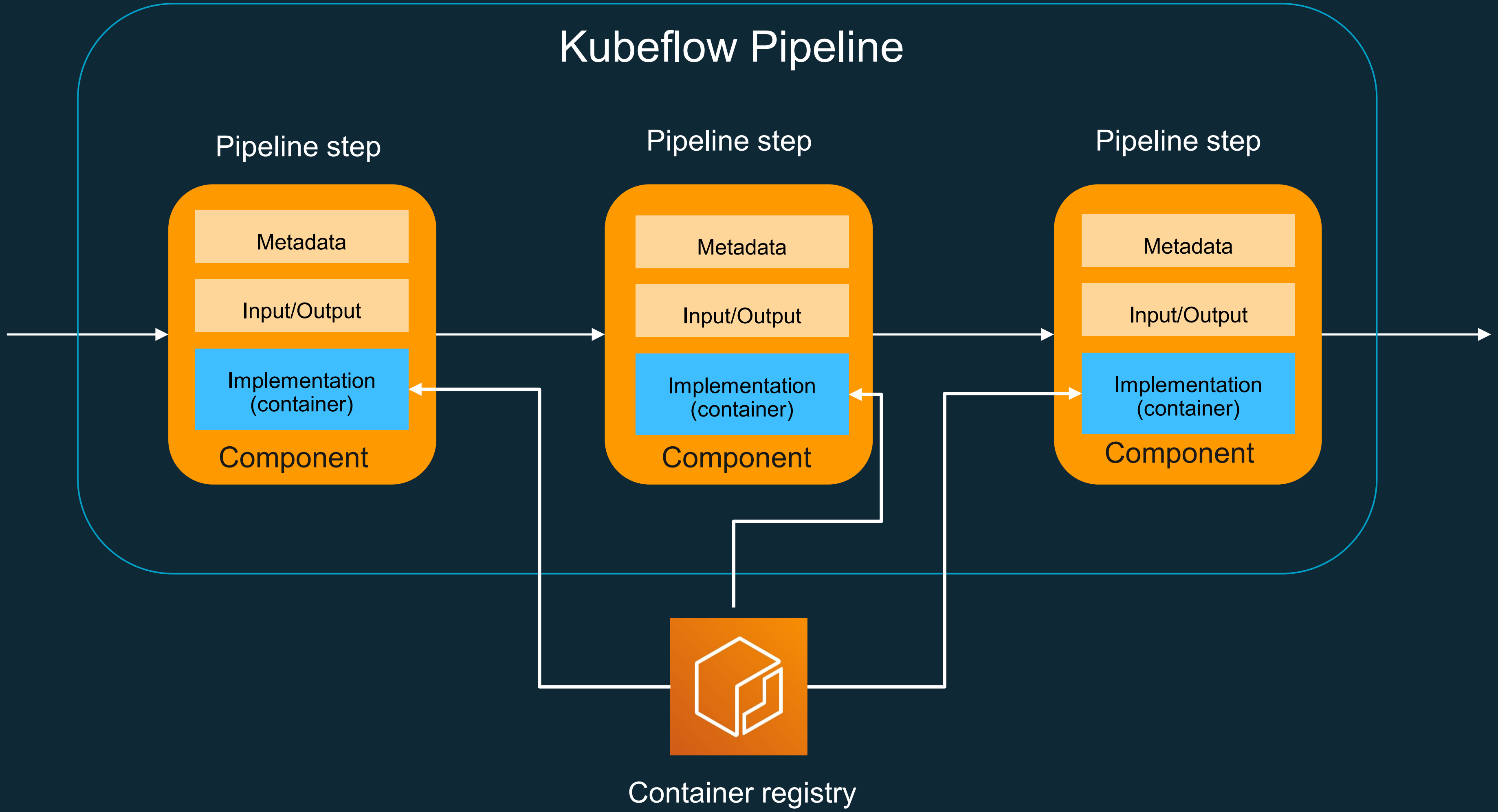
Kubeflow Pipelines

- A user interface (UI) for managing and tracking experiments, jobs, and runs.
- An engine for scheduling multi-step ML workflows.
- An SDK for defining and manipulating pipelines and components.

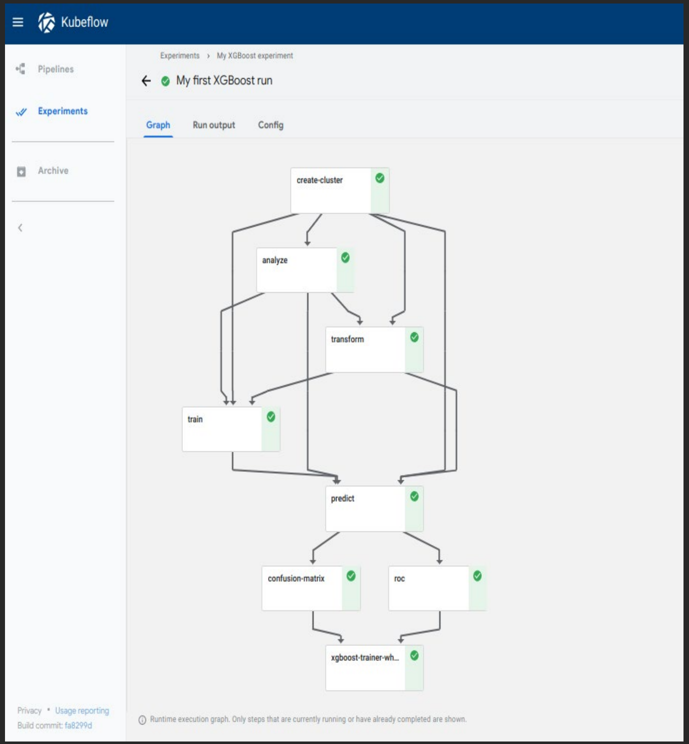


Runtime execution graph. Only steps that are currently running or have already completed are shown.

Kubeflow Pipelines Component



Creating a pipeline



Runtime execution graph. Only steps that are currently running or have already completed are shown.

Pipeline decorator

```
@dsl.pipeline(
  name='Sample Trainer',
  description=’’
)
```

Pipeline function

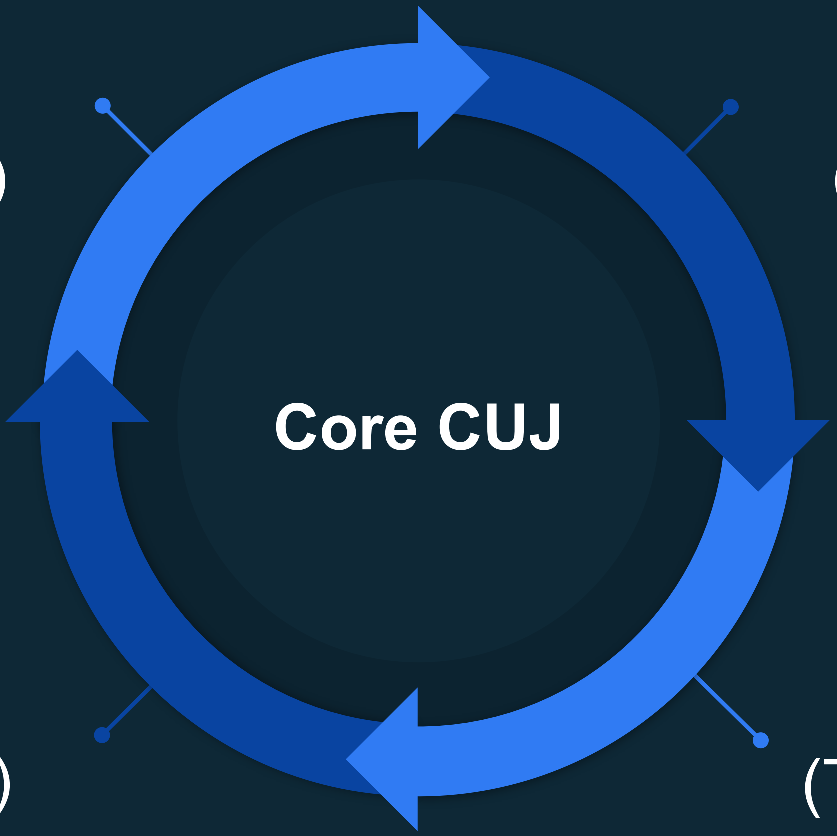
Pipeline component

```
def sample_train_pipeline(... ):

    create_cluster_op = CreateClusterOp('create-cluster', ...)

    analyze_op = AnalyzeOp('analyze', ...)

    transform_op = TransformOp('transform', ...)

    train_op = TrainerOp('train', ...)

    predict_op = PredictOp('predict', ...)

    confusion_matrix_op = ConfusionMatrixOp('confusion-matrix', ...)

    roc_op = RocOp('roc', ...)
```

Compile pipeline

```
kfp.compiler.Compiler().compile(sample_train_pipeline , 'my-pipeline.zip’)
```

Kubeflow 1.0 Arriving January 2020



Develop
(Jupyter)

Build
(Fairing)

Train
(TFJob and
PyTorch)

Deploy
(KFServing)

http://bit.ly/kf_roadmap

Kubeflow 1.0 – Main components

- Graduating 1.0
  - kfctl for deployment and upgrades
  - TFJob and PyTorch for distributed training (already 1.0)
  - Jupyter notebook controller and web app
  - Profile controller and UI for multiuser management
- Beta
  - Katib for hyper-parameter tuning
  - Fairing SDK to facilite use of notebooks for build-train-deploy
  - Metadata SDK, UI, and backend
  - KFServing for model deployment and inference

Kubeflow 1.0 – AWS Support

- Multi user support
  - Kubeflow pipelines
  - Managed contributors
- IAM Roles for Service Accounts integration with notebooks

Want to Dive Deeper on Kubeflow?

Now

2:30PM Kubeflow Workshop

(Workshop Room Harborside)

Later

https://eksworkshop.com/kubeflow/

Join the kubeflow#aws Slack channel !

DEMO