aws machine learning

Distributed Training using PyTorch with Kubeflow on AWS and Amazon SageMaker

Kanwaljit Khurmi
Sr. AI/ML Solutions Architect
AWS

Amit Kalawat
Principal Solutions Architect
AWS

Workshop Steps

1. Step 0: Provision AWS account and run automated Terraform script to create AWS EKS cluster and Kubeflow on AWS

2. Discuss why distributed machine learning training

3. Discuss AWS AI/ML stack for building scalable machine learning platforms

4. Discuss AWS Kubeflow Distribution and Kubeflow Pipelines

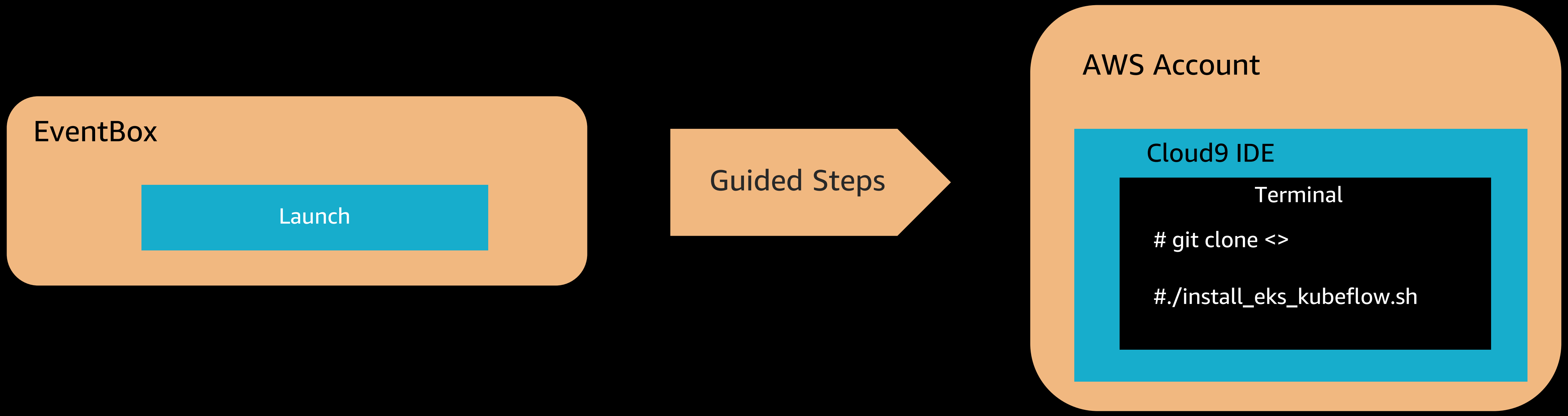
5. Step 1: Create our first Kubeflow Pipeline running XGBoost training

6. Discuss Kubeflow on AWS integration with Amazon SageMaker

7. Step 2: Demo to create pipeline for running distributed training on Amazon SageMaker

8. Wrap up with QnA

Step 0 – Provision AWS account and log on to AWS Cloud9 to run Terraform script



EventBox

Launch

Guided Steps

AWS Account

Cloud9 IDE

Terminal

```
# git clone <>

#./install_eks_kubeflow.sh
```

Let’s launch the workshop and walk through the steps to set up Kubeflow on AWS

Enter to begin the lab: https://bit.ly/dtw-eks-kubeflow

Refer to workshop instructions : https://bit.ly/dtw-kubeflow-instructions

Kubeflow on AWS – Deployment Automation Options

Kustomize

A standalone tool to customize Kubernetes objects through a kustomization file

Helm

The package manager for Kubernetes. Find share and use software built for Kubernetes

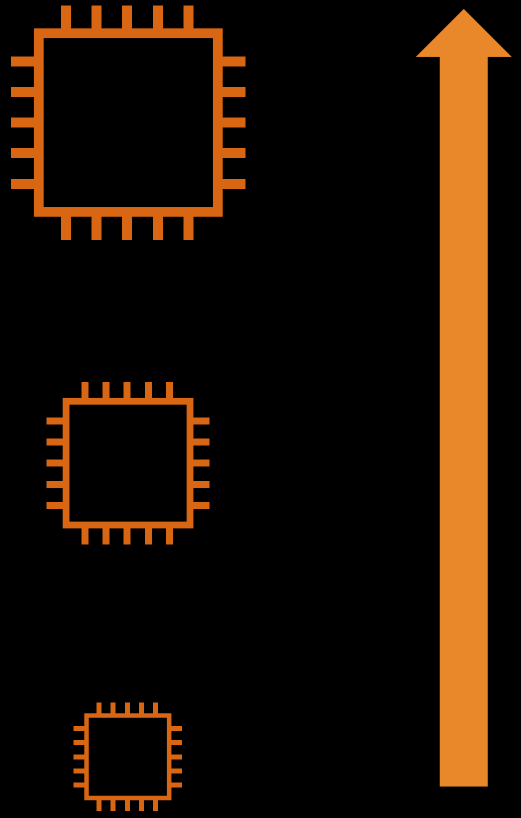
Terraform

An open-source, infrastructure as code, software tool

Why distributed training

Why scale ? Scaling DL to get faster results

Scaling-up



GPU instance size

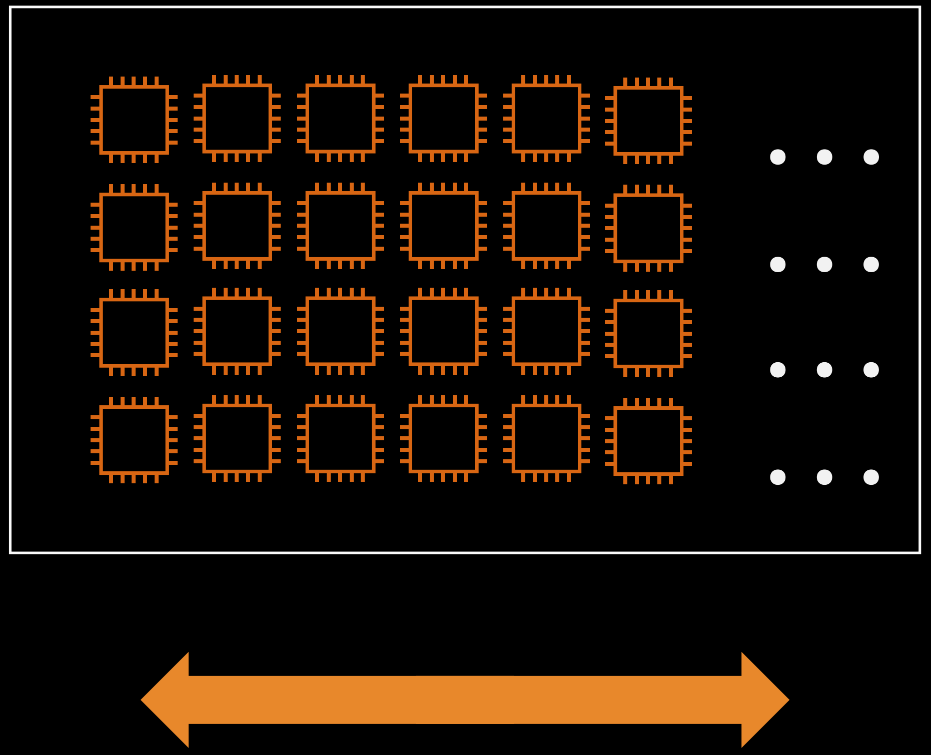
GPU memory

CPU cores

System memory

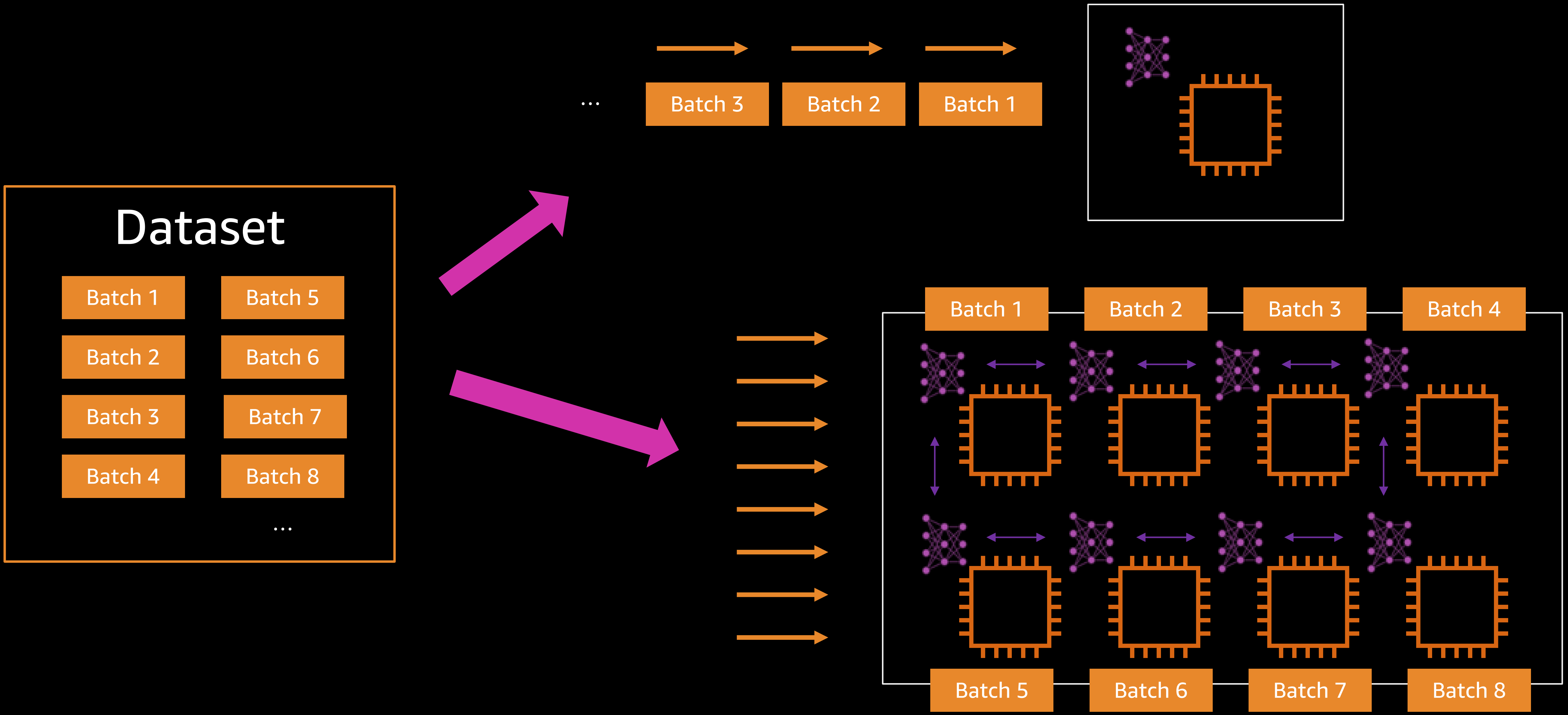
...

Scaling-out



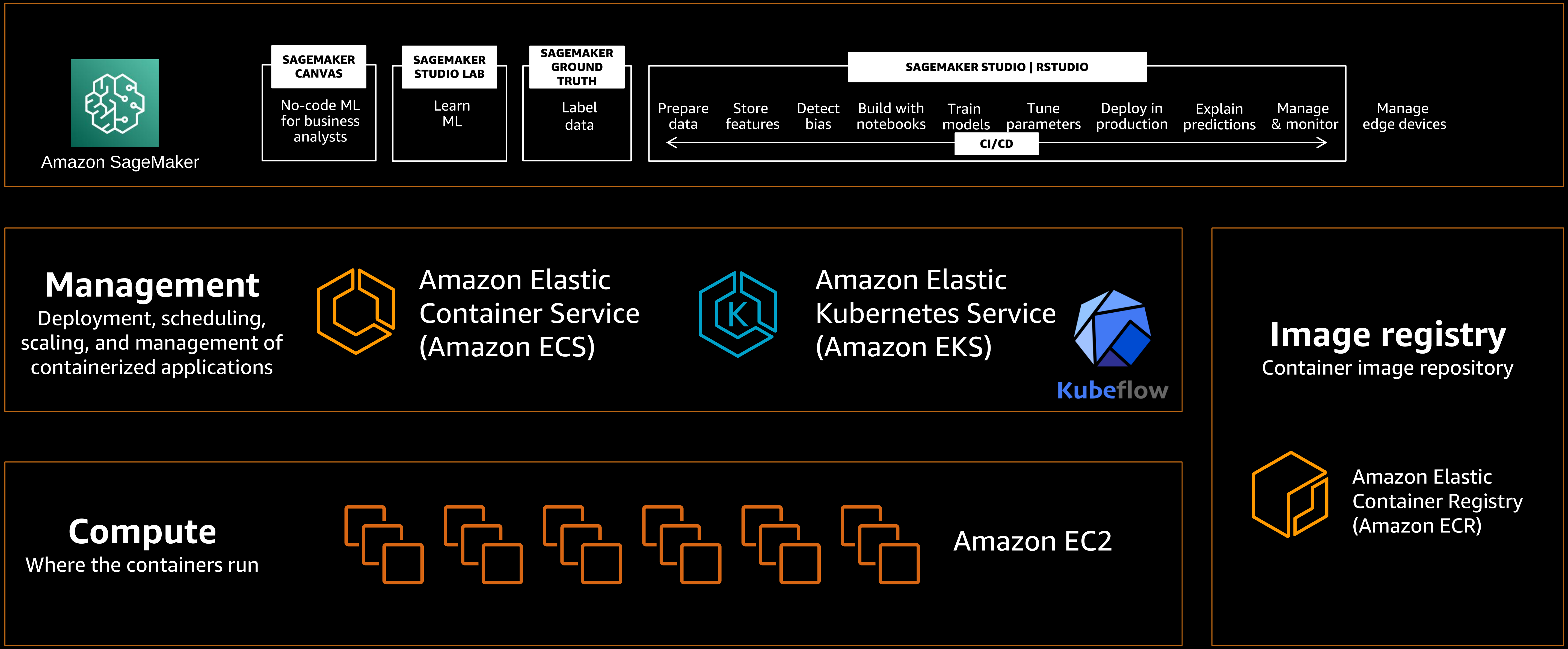
10s, 100s and 1000s GPU-accelerated instances

Training faster with distributing training



Effectively training on a large batch → speeding up training

ML services for scaling ML environments



Amazon SageMaker

SAGEMAKER CANVAS
No-code ML for business analysts

SAGEMAKER STUDIO LAB
Learn ML

SAGEMAKER GROUND TRUTH
Label data

SAGEMAKER STUDIO | RSTUDIO
Prepare data · Store features · Detect bias · Build with notebooks · Train models · Tune parameters · Deploy in production · Explain predictions · Manage & monitor
CI/CD

Manage edge devices

Management
Deployment, scheduling, scaling, and management of containerized applications

Amazon Elastic Container Service (Amazon ECS)

Amazon Elastic Kubernetes Service (Amazon EKS)

Kubeflow

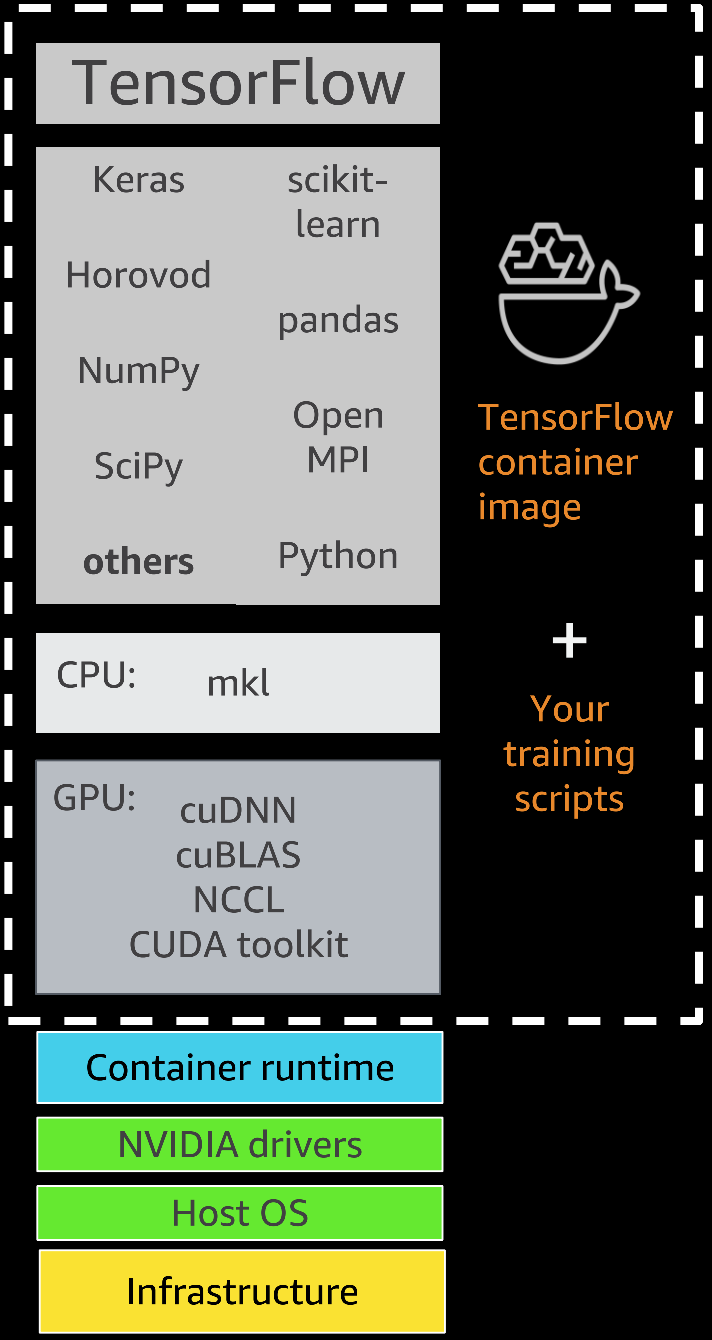
Compute
Where the containers run

Amazon EC2

Image registry
Container image repository

Amazon Elastic Container Registry (Amazon ECR)

Why machine learning with containers



TensorFlow

Keras

scikit-learn

Horovod

pandas

NumPy

Open MPI

SciPy

others

Python

CPU: mkl

GPU: cuDNN cuBLAS NCCL CUDA toolkit

TensorFlow container image

+

Your training scripts

Container runtime

NVIDIA drivers

Host OS

Infrastructure

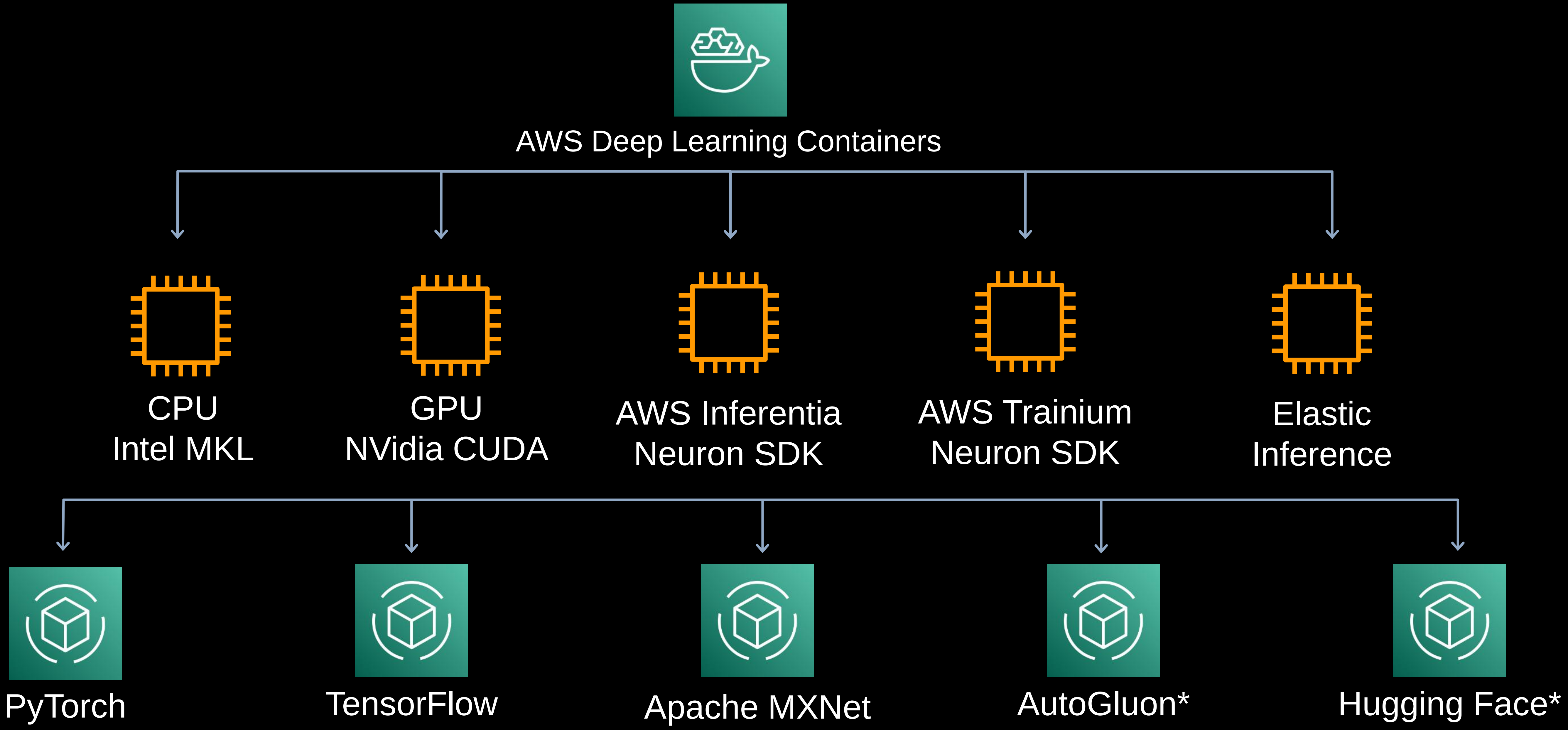
Packages

- Training code
- Dependencies
- Configurations

ML environments that are

- Lightweight
- Portable
- Scalable
- Consistent

AWS Deep learning containers



AWS Deep Learning Containers

CPU
Intel MKL

GPU
NVidia CUDA

AWS Inferentia
Neuron SDK

AWS Trainium
Neuron SDK

Elastic
Inference

PyTorch

TensorFlow

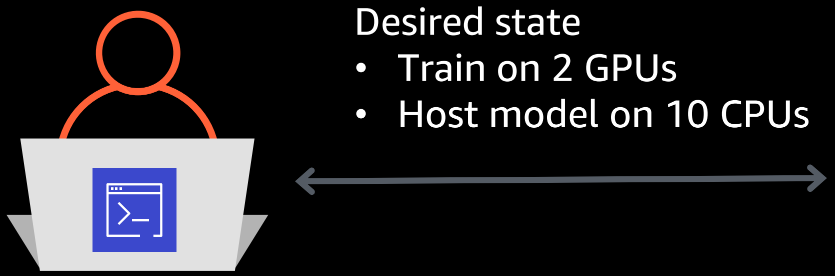
Apache MXNet

AutoGluon*

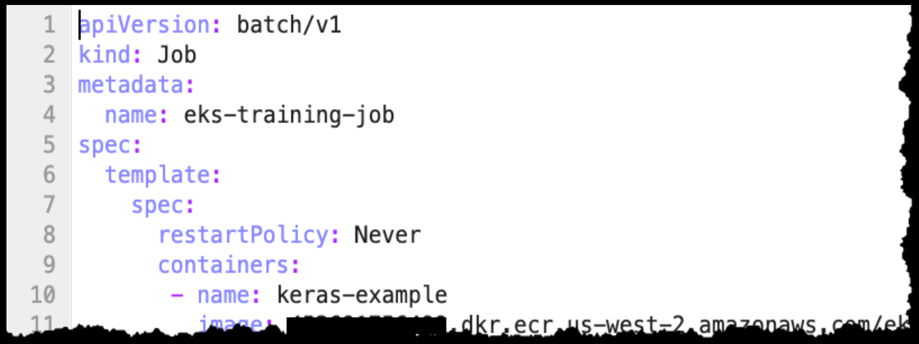
Hugging Face*

** Different inference & training containers*

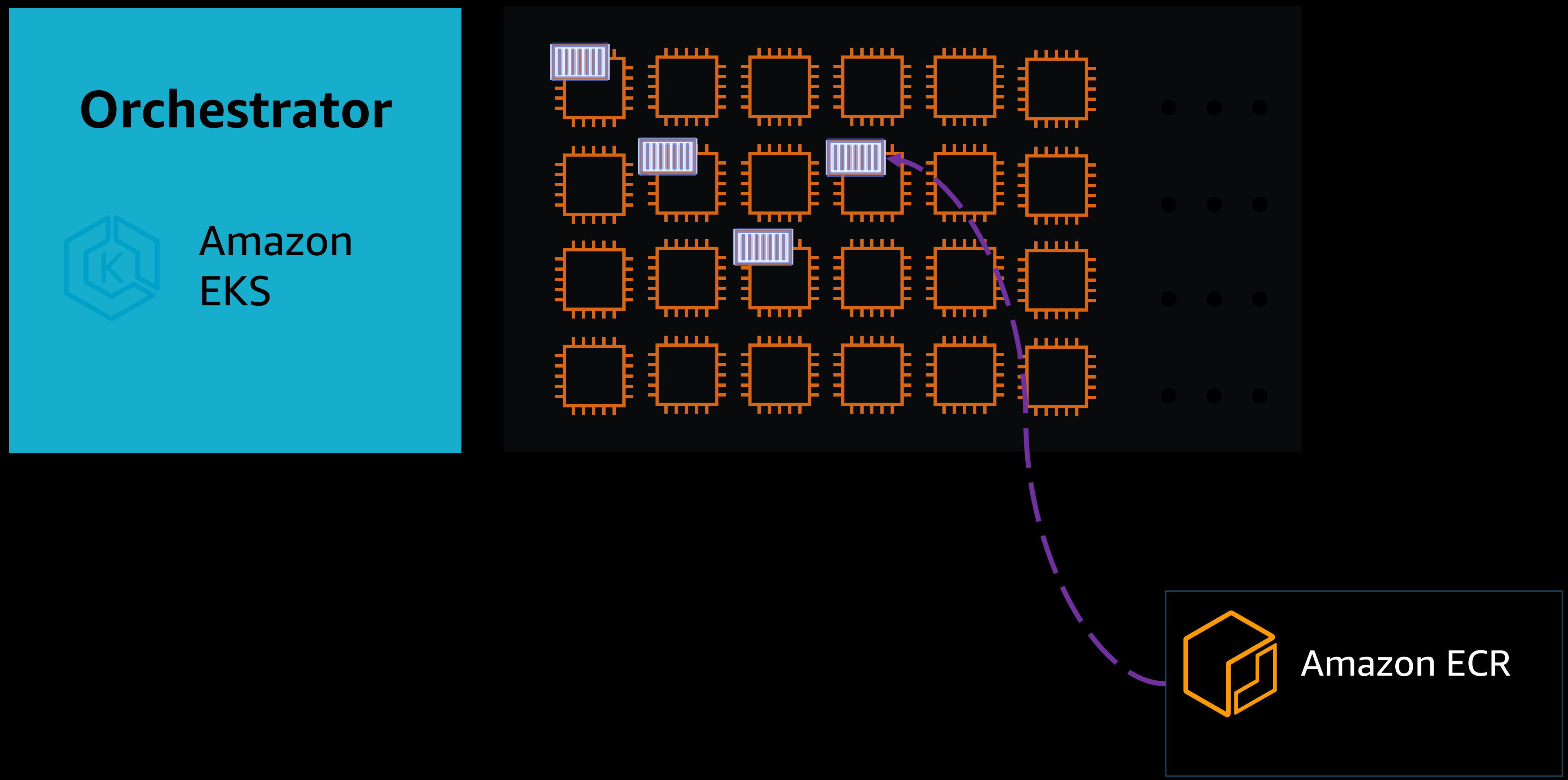
Kubernetes for machine learning



Desired state
- Train on 2 GPUs
- Host model on 10 CPUs



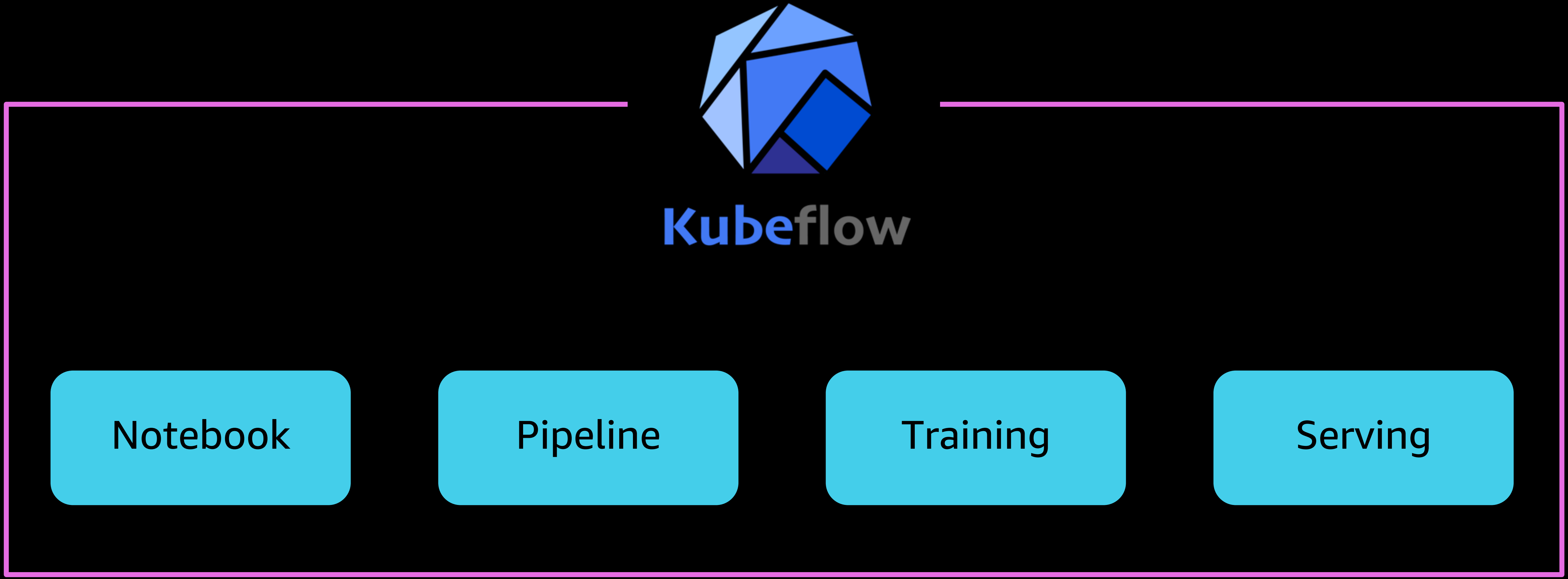
```
apiVersion: batch/v1
kind: Job
metadata:
  name: eks-training-job
spec:
  template:
    spec:
      restartPolicy: Never
      containers:
       - name: keras-example
         image: [redacted].dkr.ecr.us-west-2.amazonaws.com/ek
```



Kubernetes for ML

- ✓ Not data scientist friendly
- ✓ No ML experiment and workflow management
- ✓ Operational overheads in managing worker nodes

Introducing Kubeflow



Kubeflow

Notebook

Pipeline

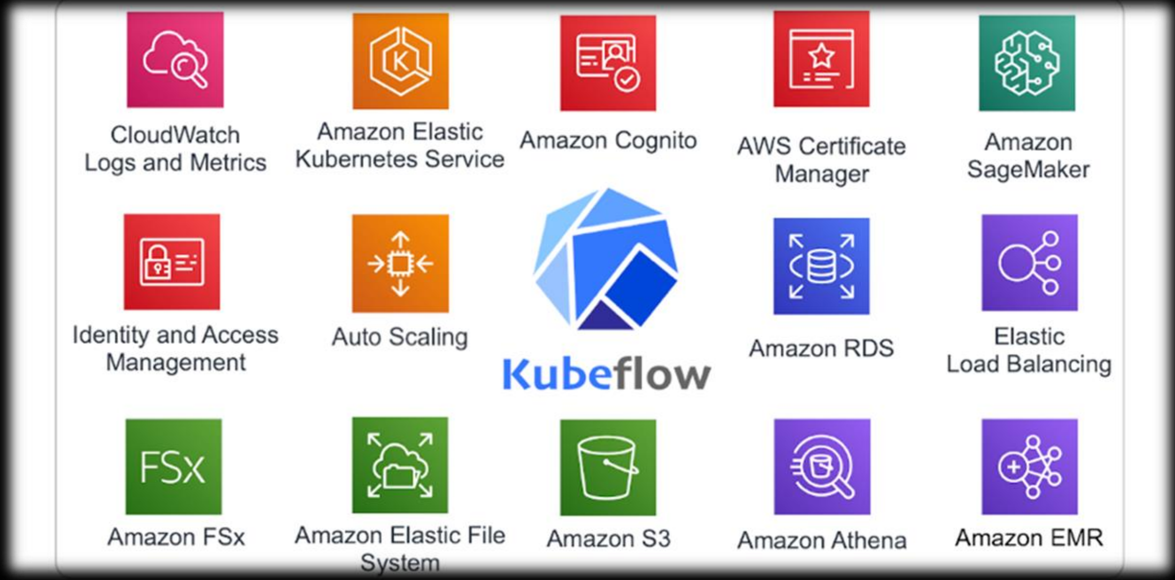
Training

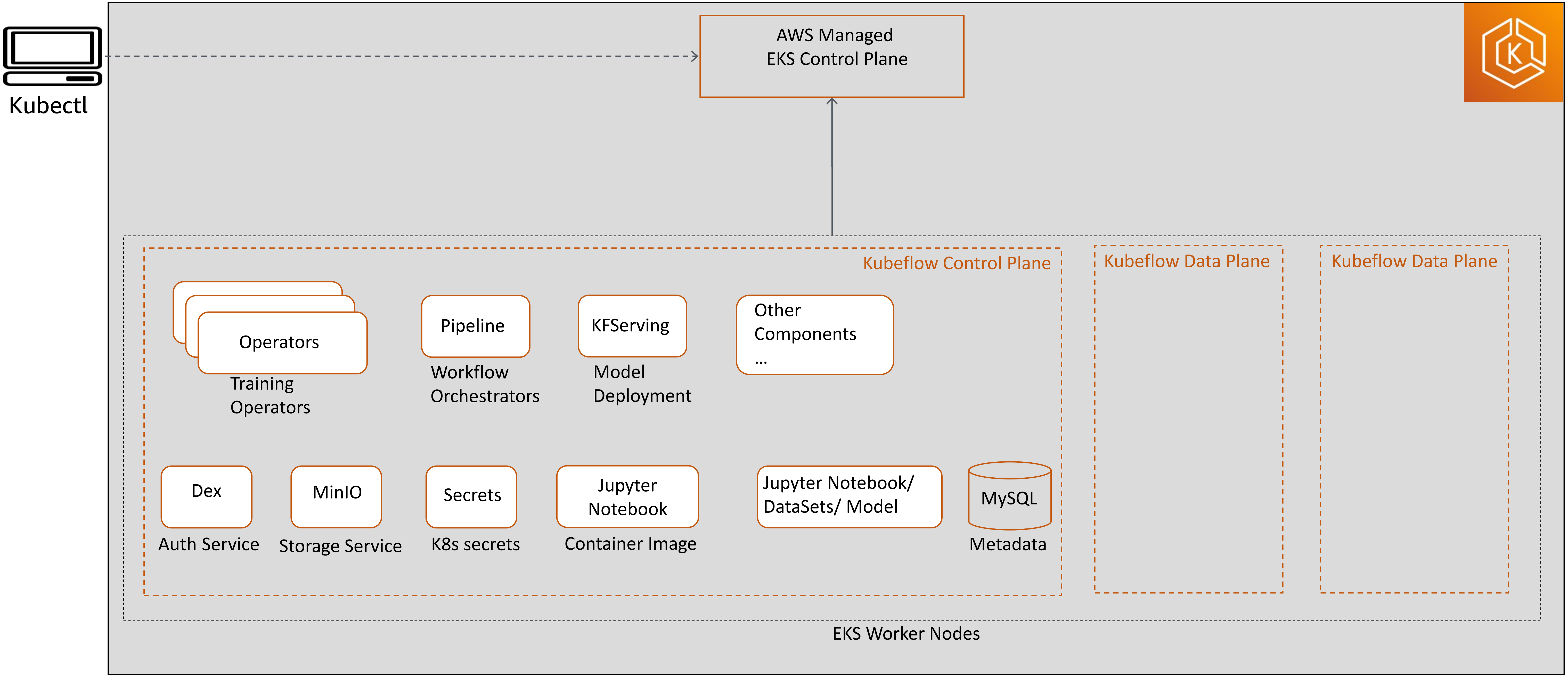
Serving

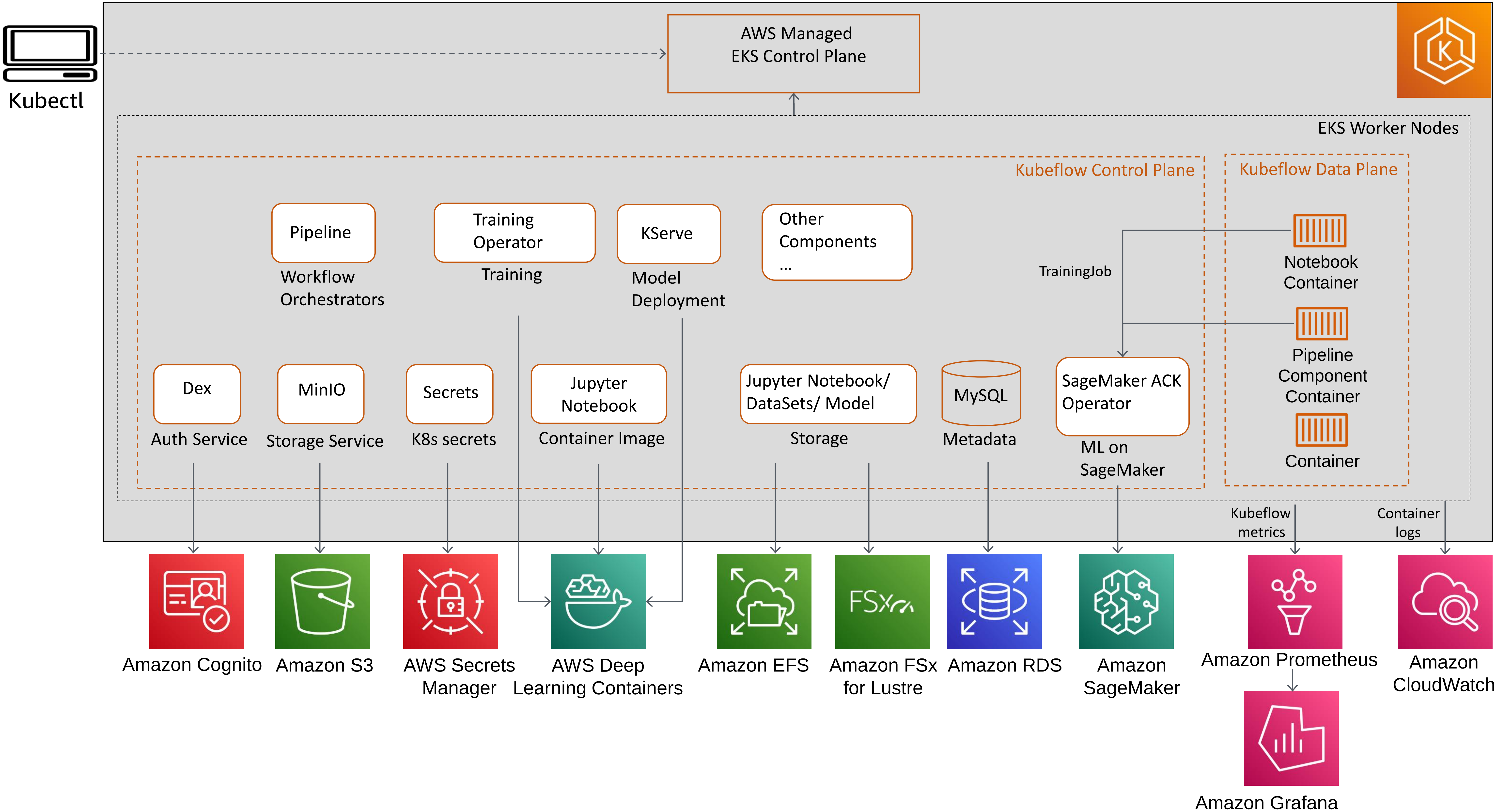
Kubeflow is the machine learning (ML)
toolkit for Kubernetes

Leverage AWS innovations through Kubeflow on AWS

Do-it-yourself + Managed service




Kubectl
AWS Managed
EKS Control Plane
Kubeflow Control Plane
Operators
Training
Operators
Pipeline
Workflow
Orchestrators
KFServing
Model
Deployment
Other
Components
…
Dex
Auth Service
MinIO
Storage Service
Secrets
K8s secrets
Jupyter
Notebook
Container Image
Jupyter Notebook/
DataSets/ Model
MySQL
Metadata
Kubeflow Data Plane
Kubeflow Data Plane
EKS Worker Nodes

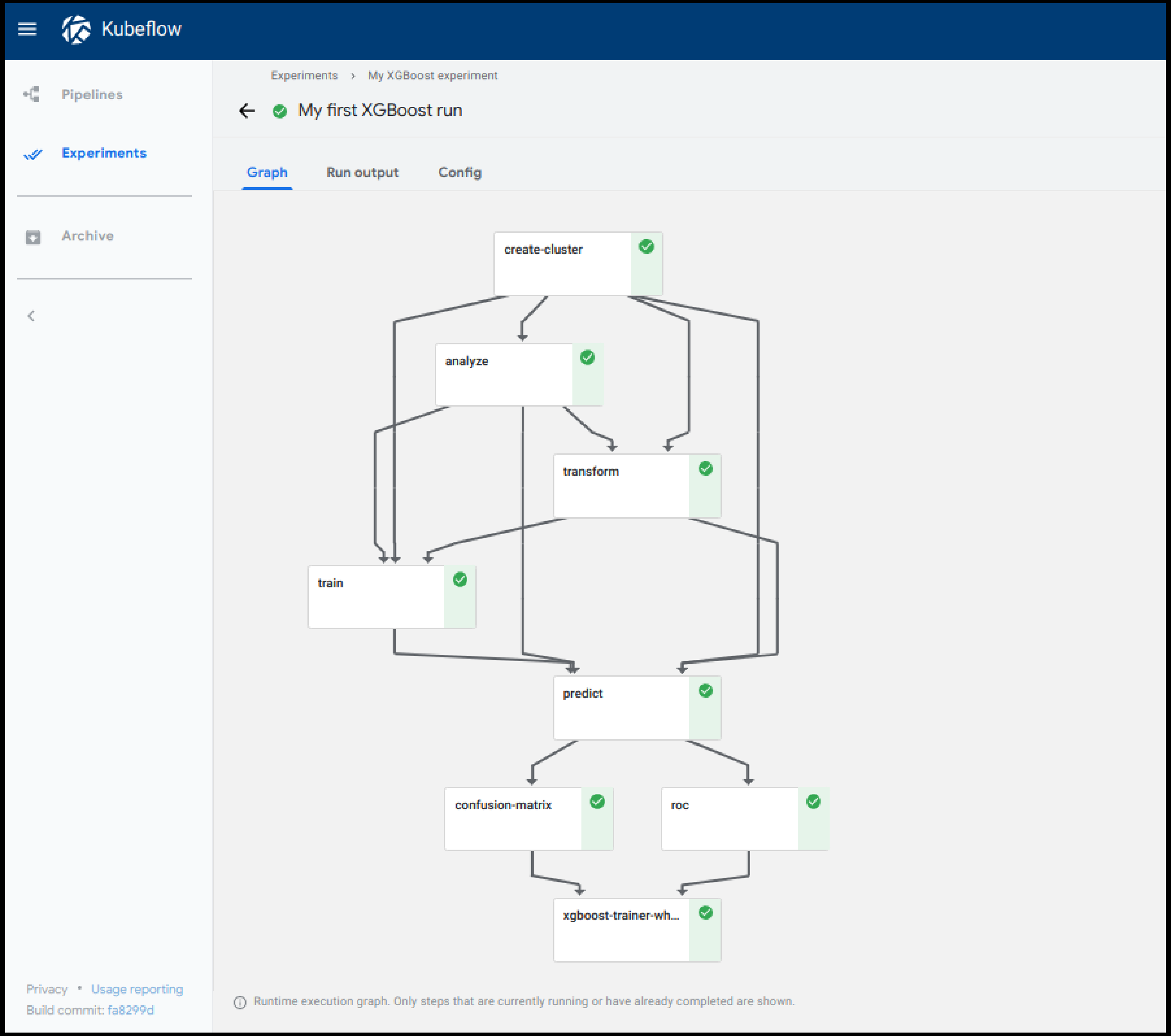

Kubectl
AWS Managed
EKS Control Plane
EKS Worker Nodes
Kubeflow Control Plane
Kubeflow Data Plane
Pipeline
Workflow
Orchestrators
Training
Operator
Training
KServe
Model
Deployment
Other
Components
...
TrainingJob
Notebook
Container
Pipeline
Component
Container
Container
Dex
Auth Service
MinIO
Storage Service
Secrets
K8s secrets
Jupyter
Notebook
Container Image
Jupyter Notebook/
DataSets/ Model
Storage
MySQL
Metadata
SageMaker ACK
Operator
ML on
SageMaker
Kubeflow
metrics
Container
logs
Amazon Cognito
Amazon S3
AWS Secrets
Manager
AWS Deep
Learning Containers
Amazon EFS
FSx
Amazon FSx
for Lustre
Amazon RDS
Amazon
SageMaker
Amazon Prometheus
Amazon
CloudWatch
Amazon Grafana

Kubeflow Pipelines

- A user interface (UI) for managing and tracking experiments, jobs, and runs

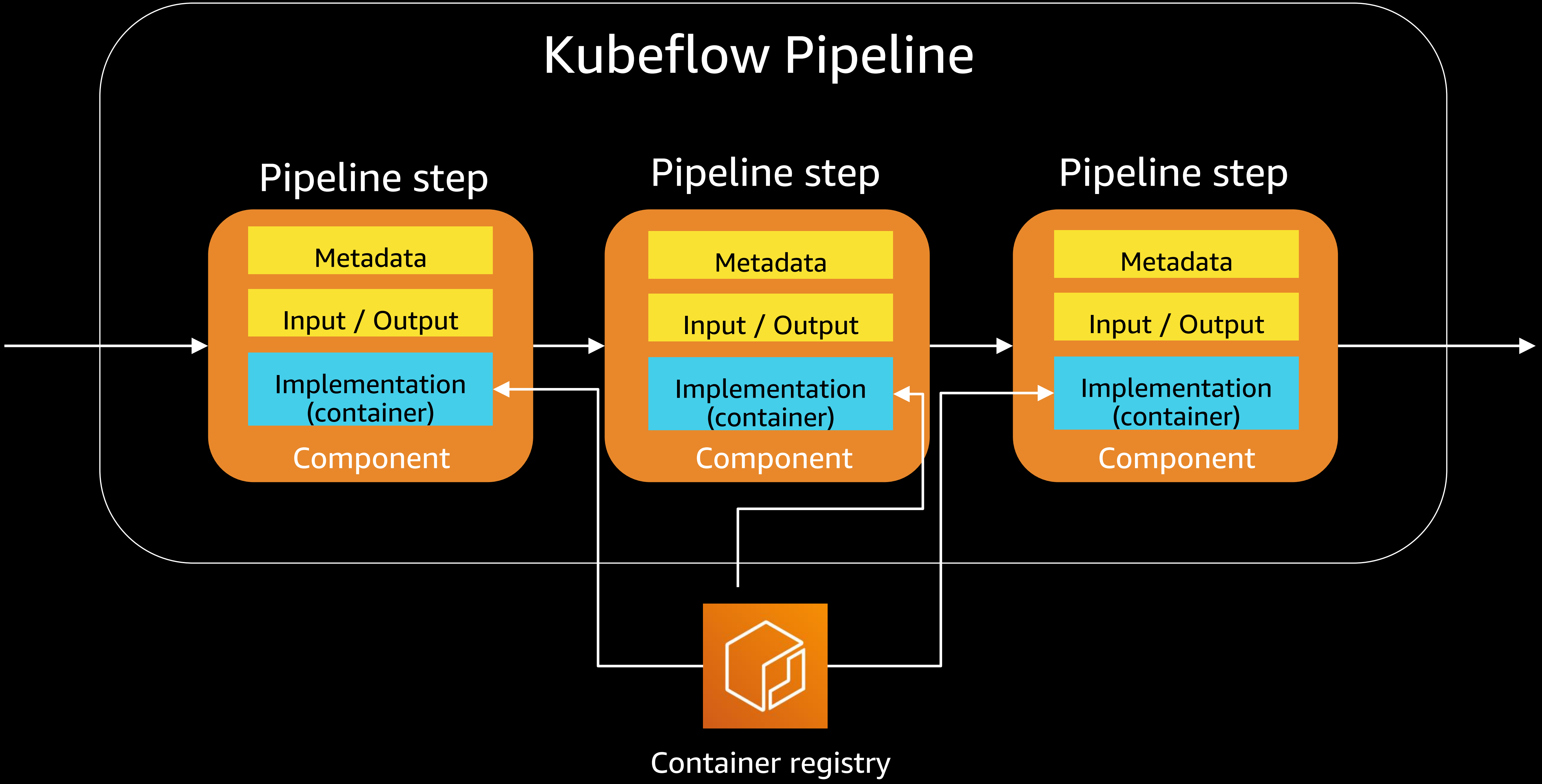
- An engine for scheduling multi-step ML workflows

- A software development kit (SDK) for defining and manipulating pipelines and components

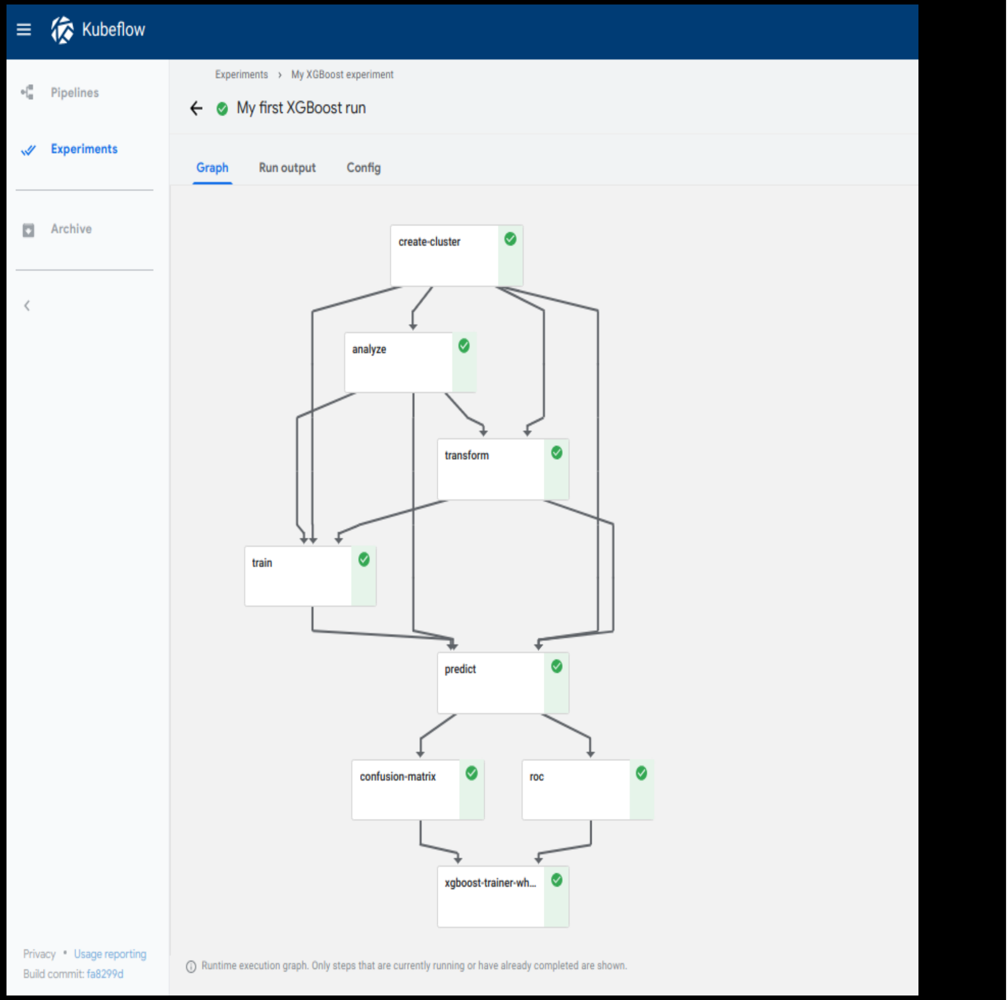


Runtime execution graph. Only steps that are currently running or have already completed are shown.

Pipeline component



Creating a pipeline (2)



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

Pipeline decorator

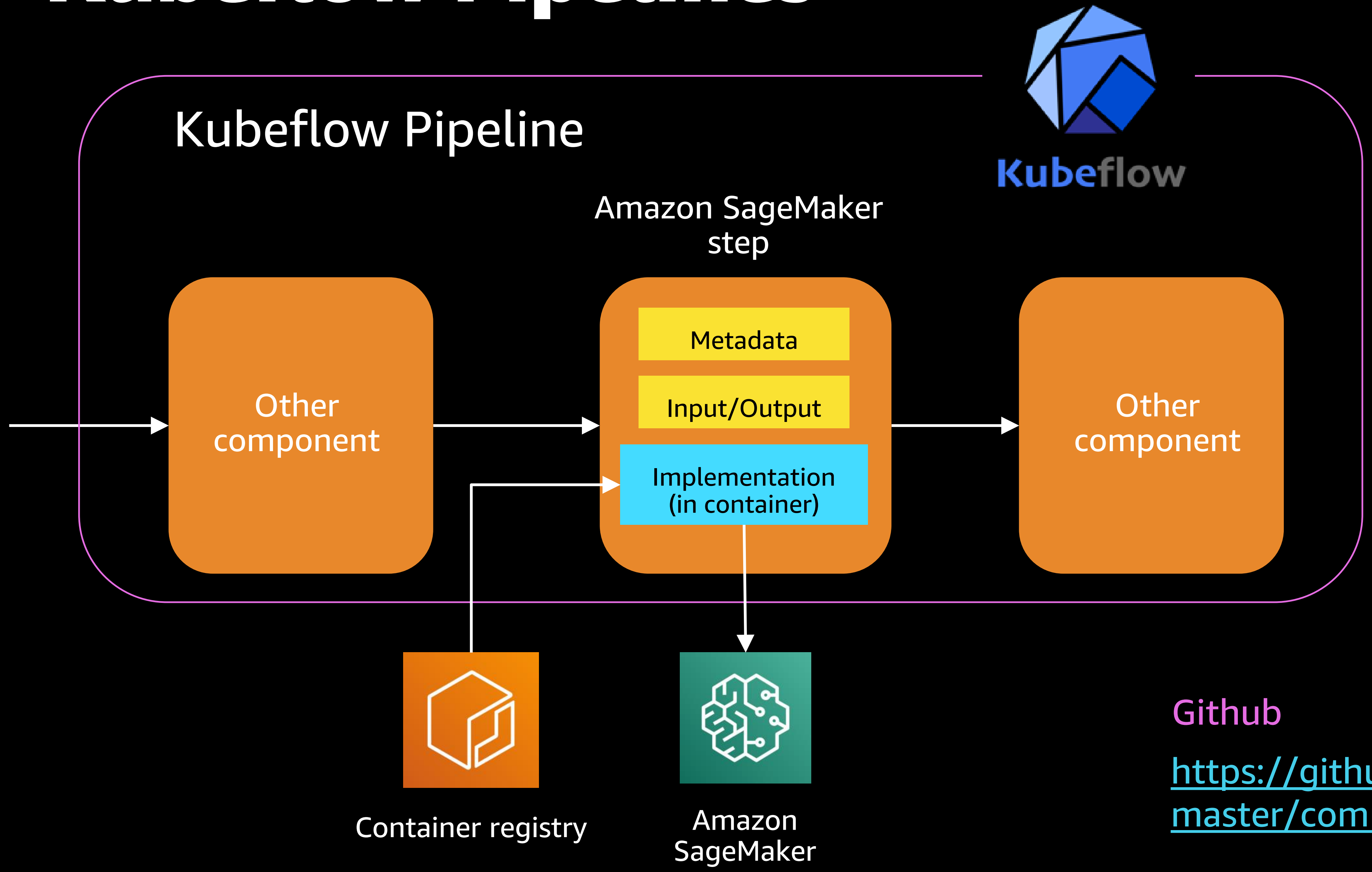
Pipeline function

Pipeline component

Compile pipeline

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

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', ...)

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

Amazon SageMaker components for Kubeflow Pipelines



Supported components

- Model training
- Hyperparameter tuning
- Processing
- Model deployment
- Batch transform
- Amazon SageMaker Ground Truth

Github

https://github.com/kubeflow/pipelines/tree/master/components/aws/sagemaker

Check on Amazon EKS and Kubeflow on AWS deployment status

Step 1.0: Demo to create pipeline for running distributed training on Amazon SageMaker

Step 1.1: Demo to invoke Amazon SageMaker endpoint

Step 2 : Create our first Kubeflow Pipeline running XGBoost training

Takeaways

- Kubeflow provides end-to-end ML capabilities on Kubernetes
- Kubeflow on AWS makes it easy to deploy Kubeflow and provides easy integration with highly optimized, cloud-native, enterprise-ready AWS services.
- Using Kubeflow and PyTorch, we demonstrated a Kubeflow pipeline for distributed model training which runs fully managed infrastructure.

References

- AWS: https://aws.amazon.com
- EKS: https://aws.amazon.com/eks
- Kubeflow: https://kubeflow.org
- Blog (Kubeflow on AWS): https://go.aws/3p7CGCo
- Manifests (Kubeflow on AWS): https://awslabs.github.io/kubeflow-manifests/
- aws-do-kubeflow project: https://bit.ly/aws-do-kubeflow
- PyTorch: https://pytorch.org/
- Distributed Training Workshop with PyTorch and Kubeflow: https://bit.ly/dtw-kubeflow

Credits

Arindam Paul
Shashank Murthy
Tatsuo Azeyanagi
Suraj Kota
Sree Arasanagatta

Thank you!

Kanwaljit Khurmi
Sr. AI/ML Solutions Architect
AWS
https://www.linkedin.com/in/kanwaljitkhurmi/

Amit Kalawat
Principal Solutions Architect
AWS
https://www.linkedin.com/in/amitkalawat/