AWS re:Invent

CON413-R

Move your Machine Learning workloads on Amazon Elastic Kubernetes Service (EKS)

Jiaxin Shan
Software Engineer
AWS

Mike Stefaniak
Product Manager
AWS

Arun Gupta
Open Source Technologist
AWS

Machine Learning Stack on EKS

ML Frameworks + Infrastructure

FRAMEWORKS			INTERFACES	
TensorFlow	mxnet	PYTORCH	GLUON	Keras

INFRASTRUCTURE						
EC2 P3 & P3dn	EC2 G4	EC2 C5	FPGAs	FSx Lustre	EFA	Inferentia

Machine Learning Stack on EKS

Container Platform

- **Amazon EKS**
- Optimized GPU AMI
- Auto Scaling
- Deep Learning Container
- FSx CSI Plugin

ML Frameworks + Infrastructure

FRAMEWORKS

TensorFlow
mxnet
PYTORCH

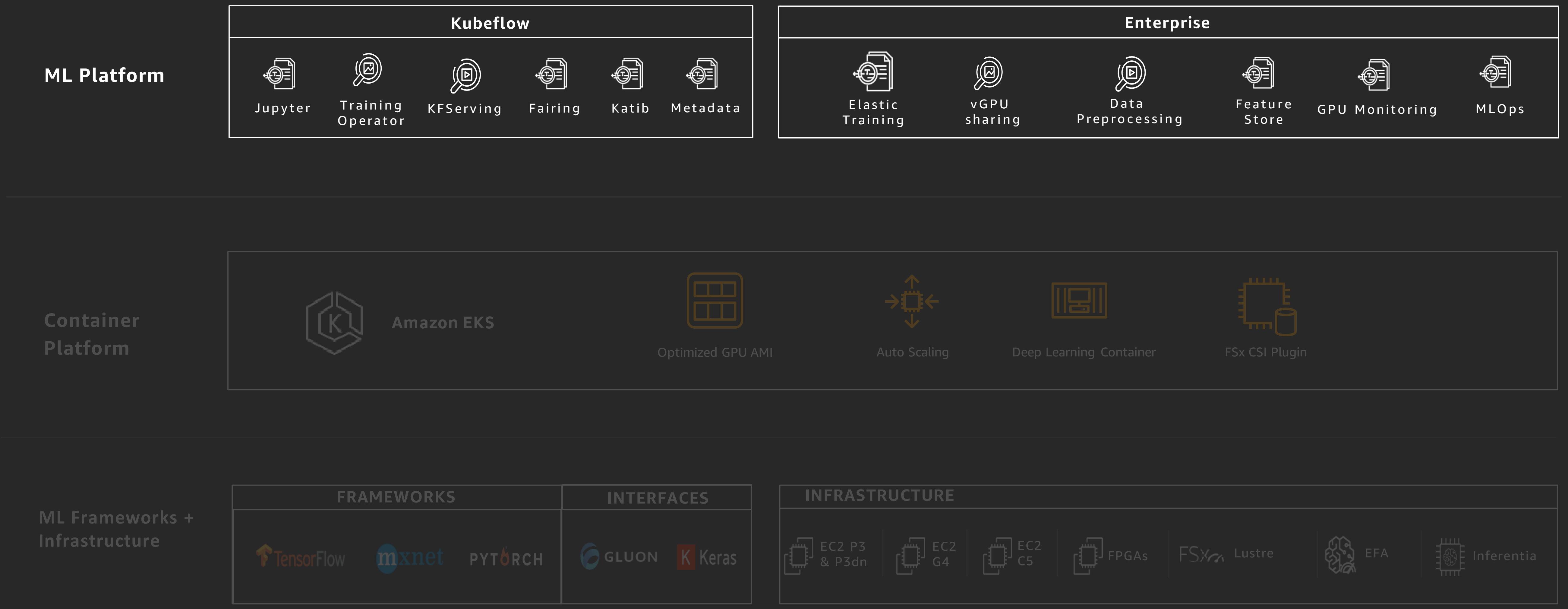
INTERFACES

GLUON
Keras

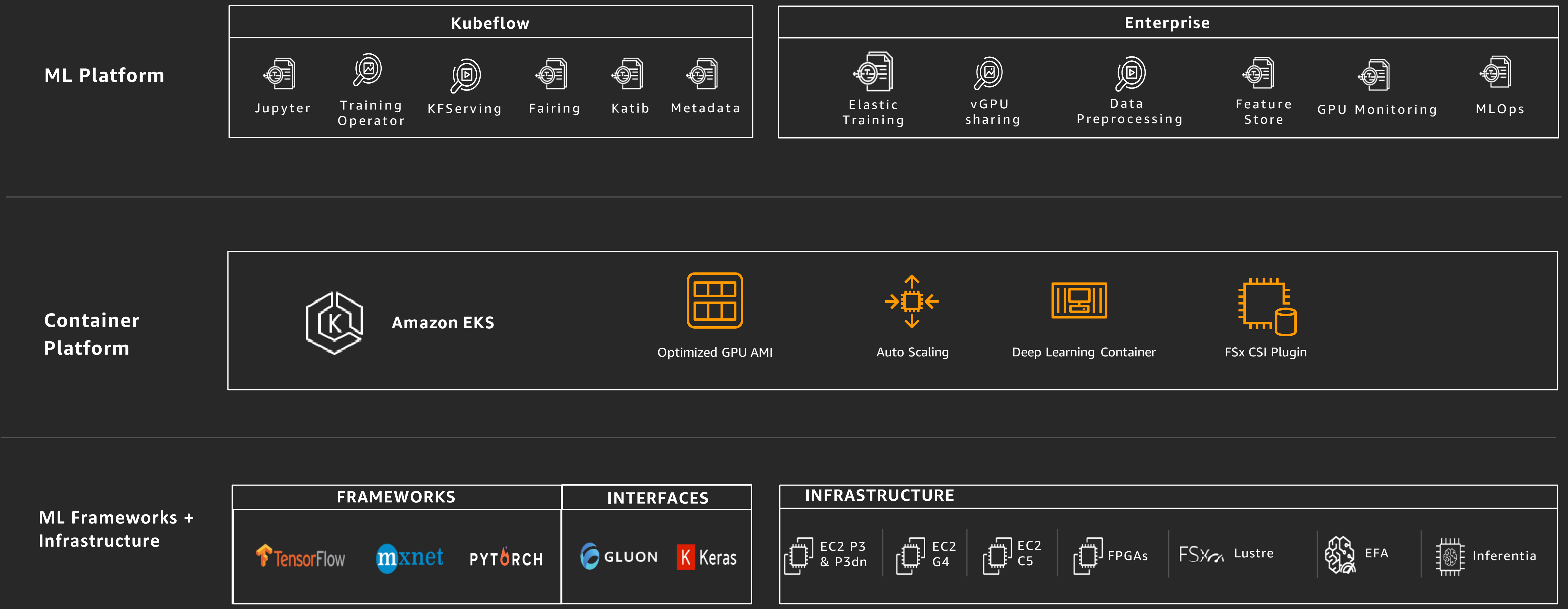
INFRASTRUCTURE

EC2 P3 & P3dn
EC2 G4
EC2 C5
FPGAs
FSx Lustre
EFA
Inferentia

Machine Learning Stack on EKS

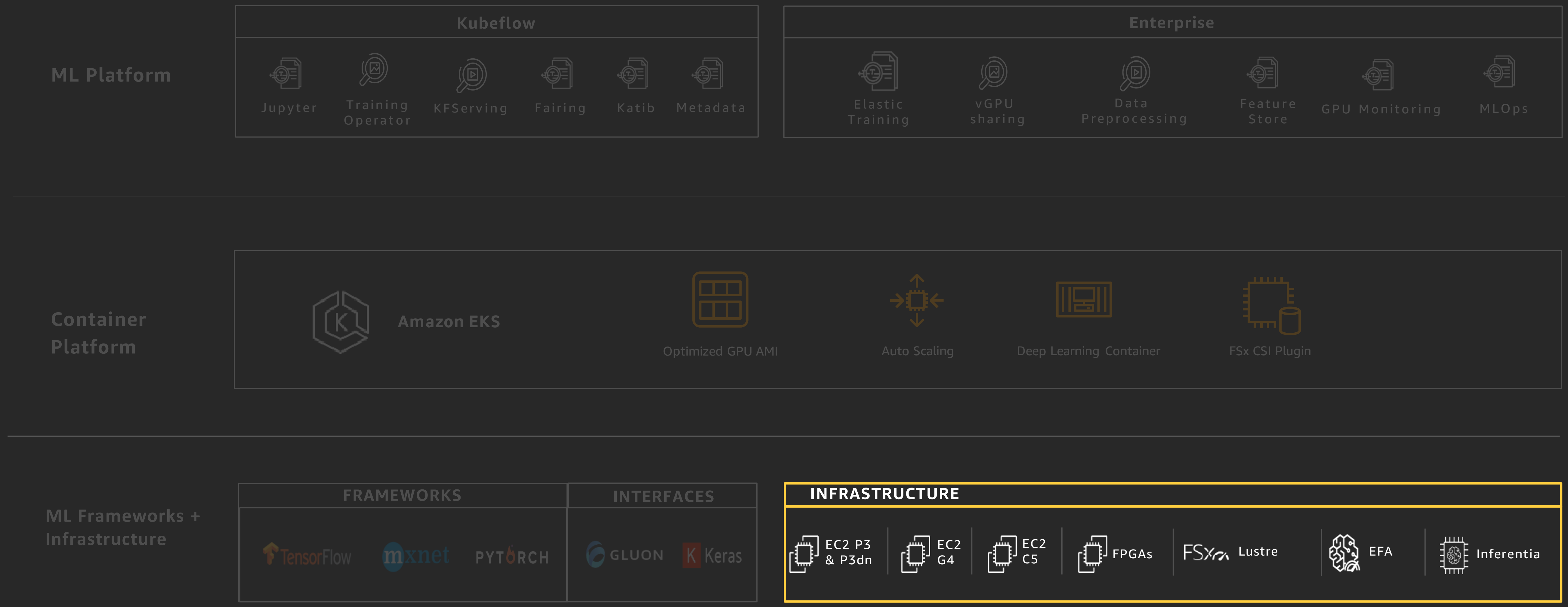


Machine Learning Stack on EKS



Infrastructure

Machine Learning Stack on EKS



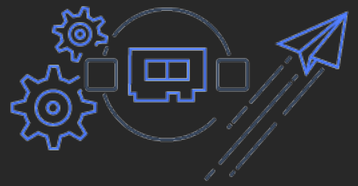
ML Infrastructure supported by EKS

Compute

P3 G4

Accelerated Instance Types

Networking



Elastic Fabric Adapter

Storage

Amazon Elastic File System

Amazon FSx for Lustre

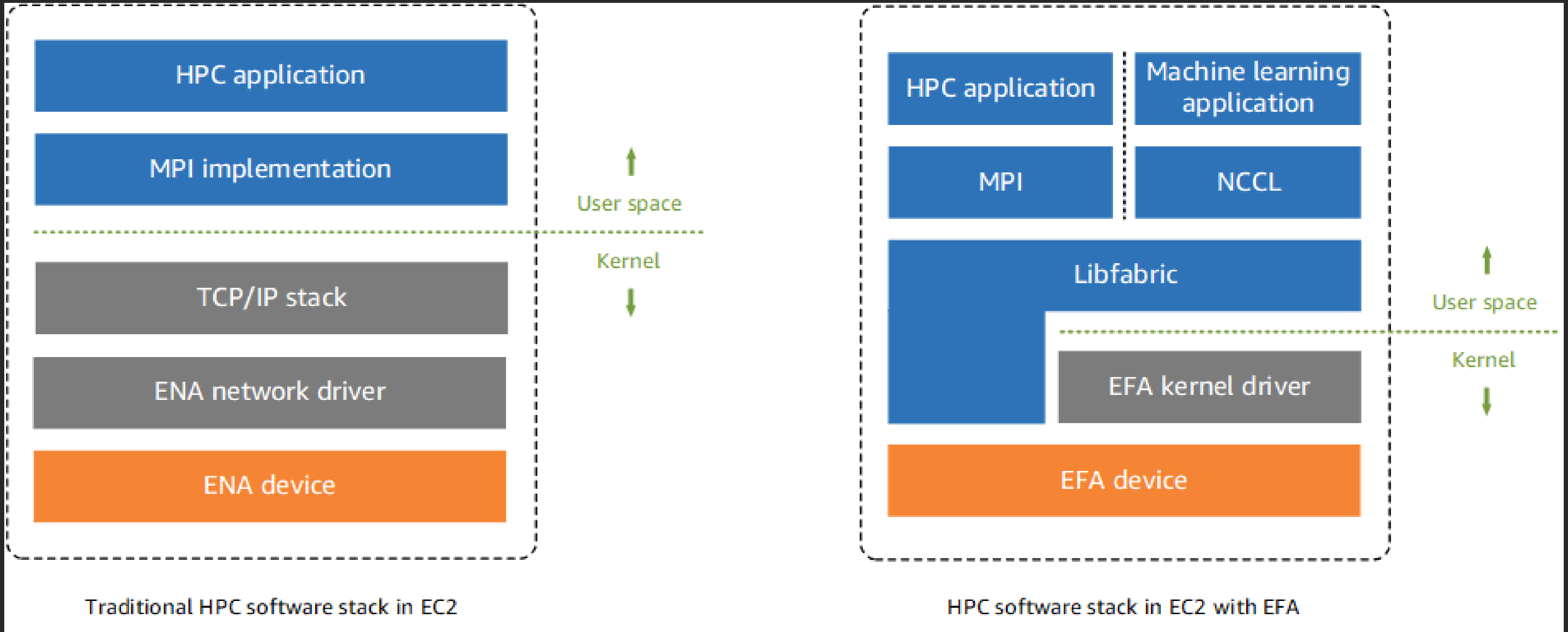
Inference

AWS Inferentia (EKS support coming soon)

Elastic Fabric Adapter

Use MPI or NCCL to interface with Libfabric API (bypass OS kernel)

Reduce overhead and enable HPC applications to run more efficiently



Distributed Storage

Amazon Elastic
File System

Amazon FSx
for Lustre

Cloud-native, shared NFS storage solution

Mount shared filesystem into pods

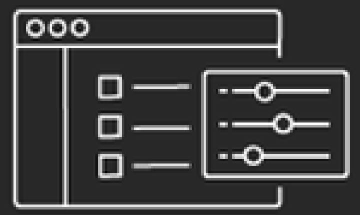
Share datasets or models across teams

File system optimized for ML and HPC workloads

Native integration with S3 for stored training data

Read/write data up to hundreds of GB/sec of throughput and millions of IOPS

AWS Inferentia (EKS support coming soon)



Built for high throughput, low latency applications

High performance alternative to a GPU

Order of magnitude cost reduction

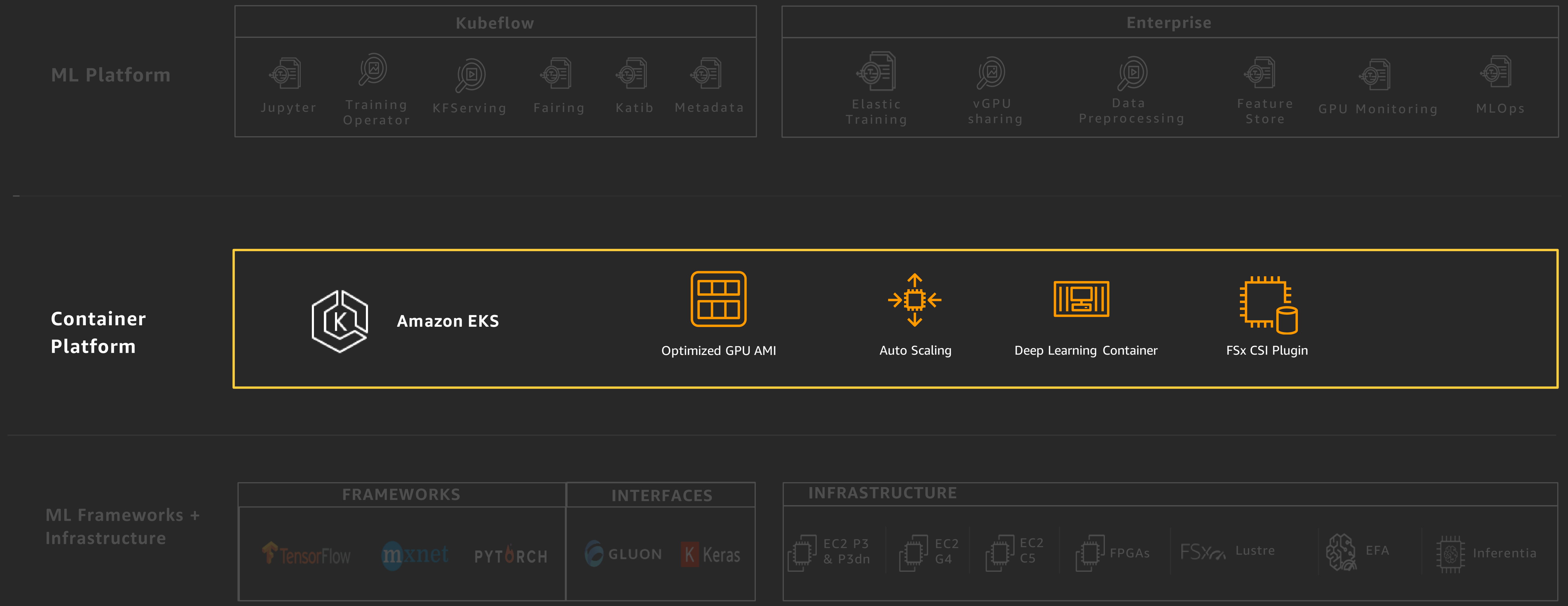
Each chip provides hundreds of TOPS (tera operations per second) of inference throughput to allow complex models to make fast predictions.

For even more performance, multiple Inferentia chips can be used together to drive thousands of TOPS of throughput

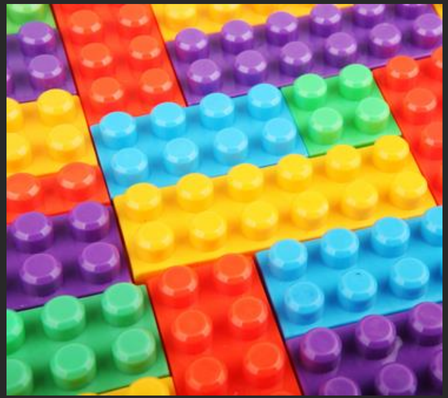
Supports all major frameworks used in the deep learning community including TensorFlow, Apache MXNet, and PyTorch, as well as models that use the ONNX format

Kubernetes

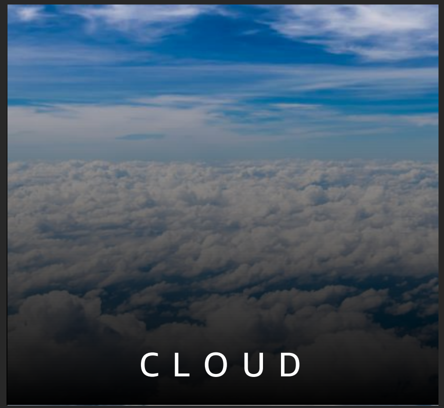
Machine Learning Stack on EKS



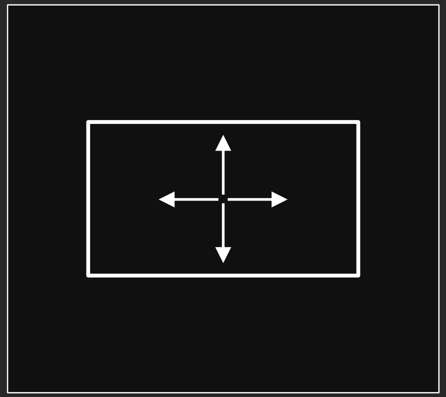
Why Machine Learning on Kubernetes?



Composability



Portability



Scalability

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

Amazon EKS-Optimized GPU AMI

Built on top of the standard Amazon EKS-Optimized AMI

Includes packages to support Amazon P2/P3/G3/G4 instances

- NVIDIA drivers
- nvidia-docker2 package
- nvidia-container-runtime (as default runtime)

GPU Clock Optimization

AWS Deep Learning Containers

Optimized and customizable containers for deep learning environments

Pre-packaged
Docker container
images
fully configured
and validated

Best performance
and scalability
without tuning

Works with
Amazon EKS,
Amazon ECS,
and Amazon EC2

KEY FEATURES

Customizable
container images

Support for
TensorFlow mxnet
PYTORCH HOROVOD

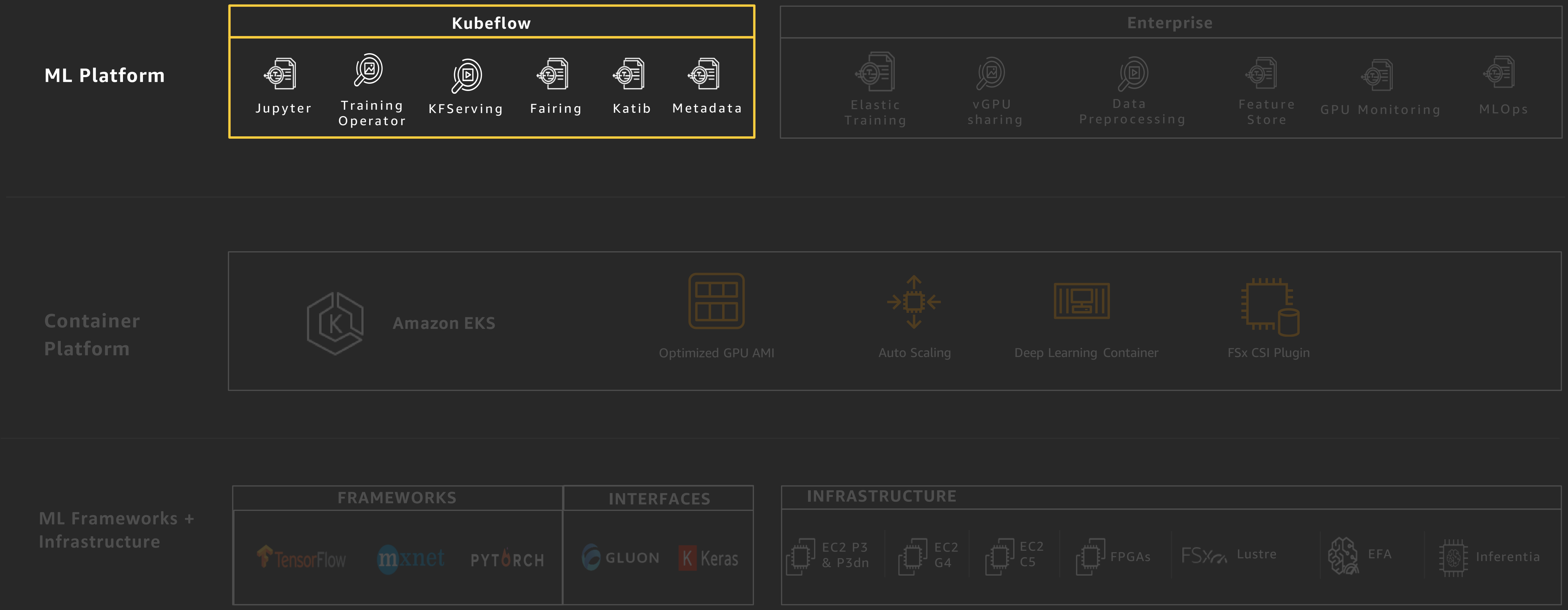
Single and multi-node
training and inference

Cluster Autoscaler Improvements

- Add GPU Support
  - autoscaler#1584 Move GPULabel and GPUTypes to cloud provider –> GPU autoscaling supported for AWS
  - autoscaler#1589 Consider GPU utilization in scaling down –> GPU scale down performance optimization
- Prevent CA from removing a node with ML training job running
  - Annotate job ”cluster-autoscaler.kubernetes.io/safe-to-evict”: “false”
- Recommended to create GPU node group per AZ
  - Improve network communication performance
  - Prevent ASG rebalancing

Kubeflow

Machine Learning Stack on EKS

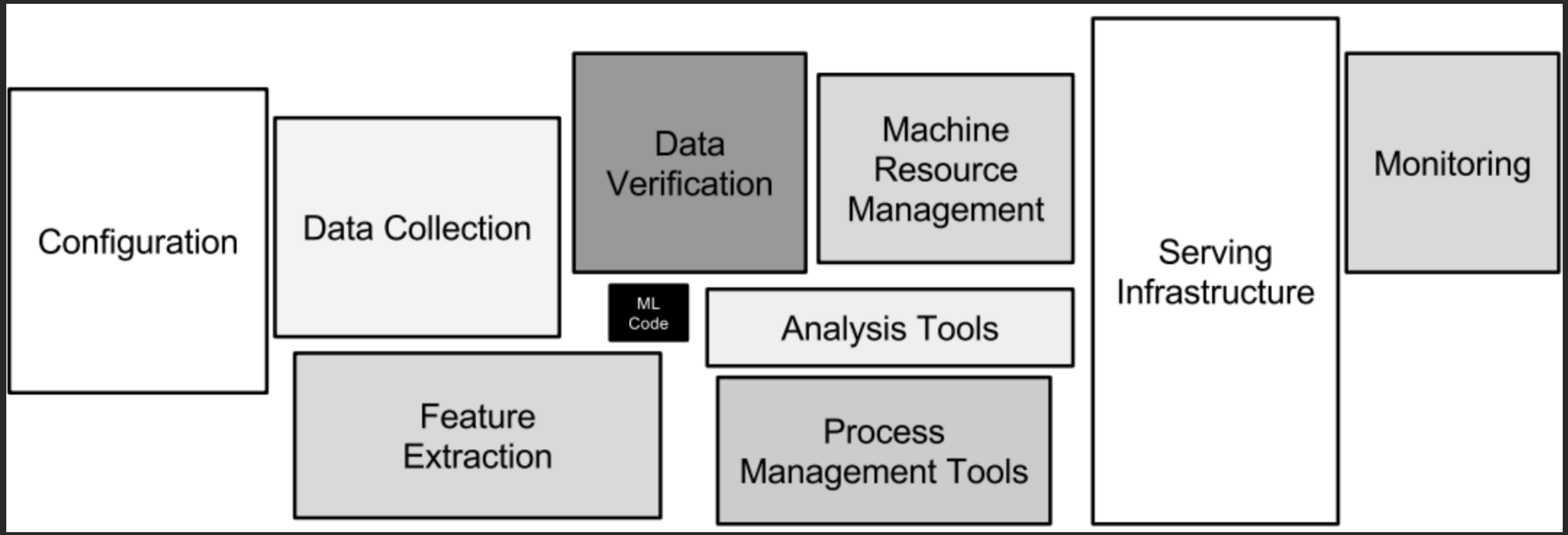


What is Kubeflow

Containerized machine learning platform

Makes it easy to develop, deploy, and manage portable, scalable end-to-end ML workflows on k8s

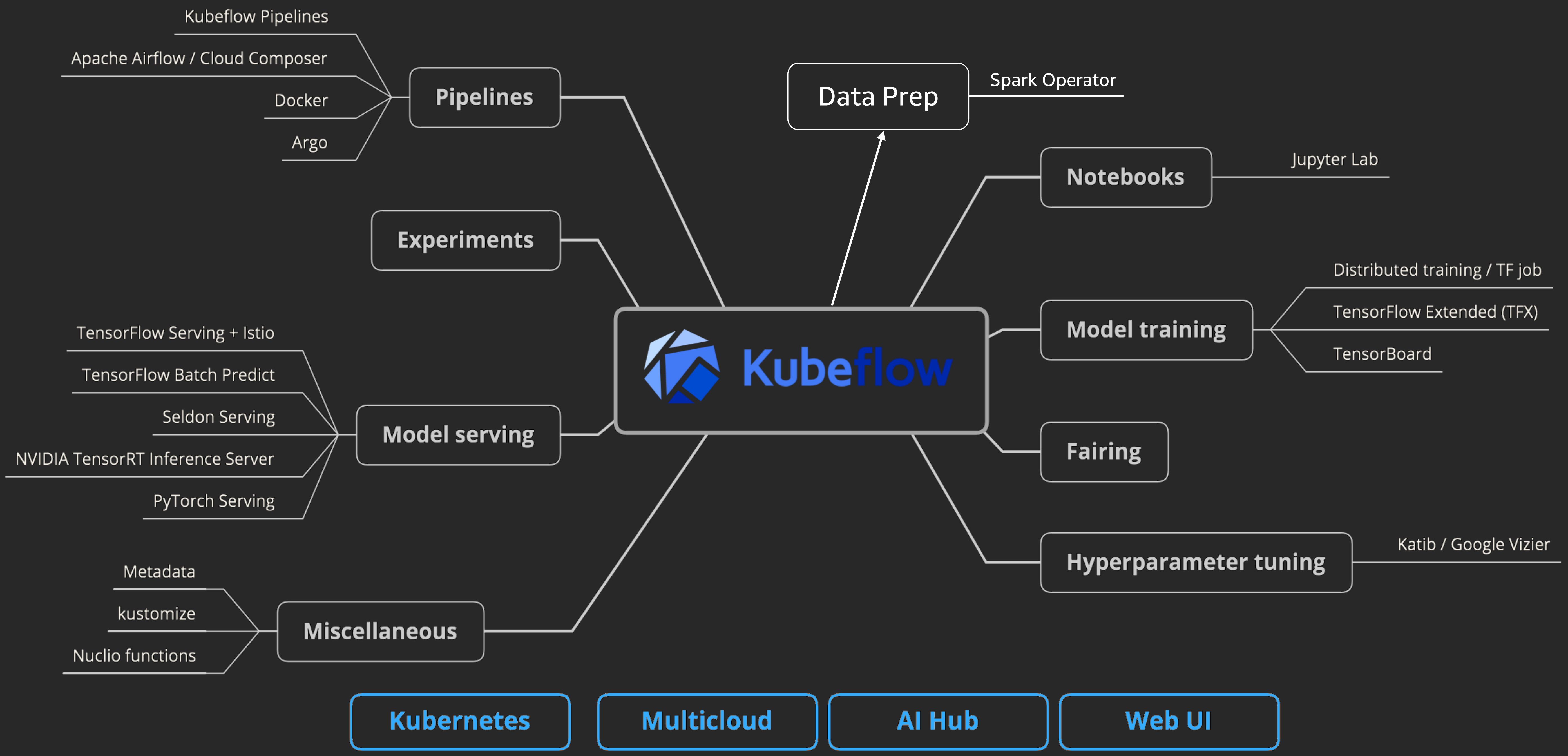
“Toolkit” – loosely coupled tools and blueprints for ML



End to End ML workflow – ML code is only a small component

https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

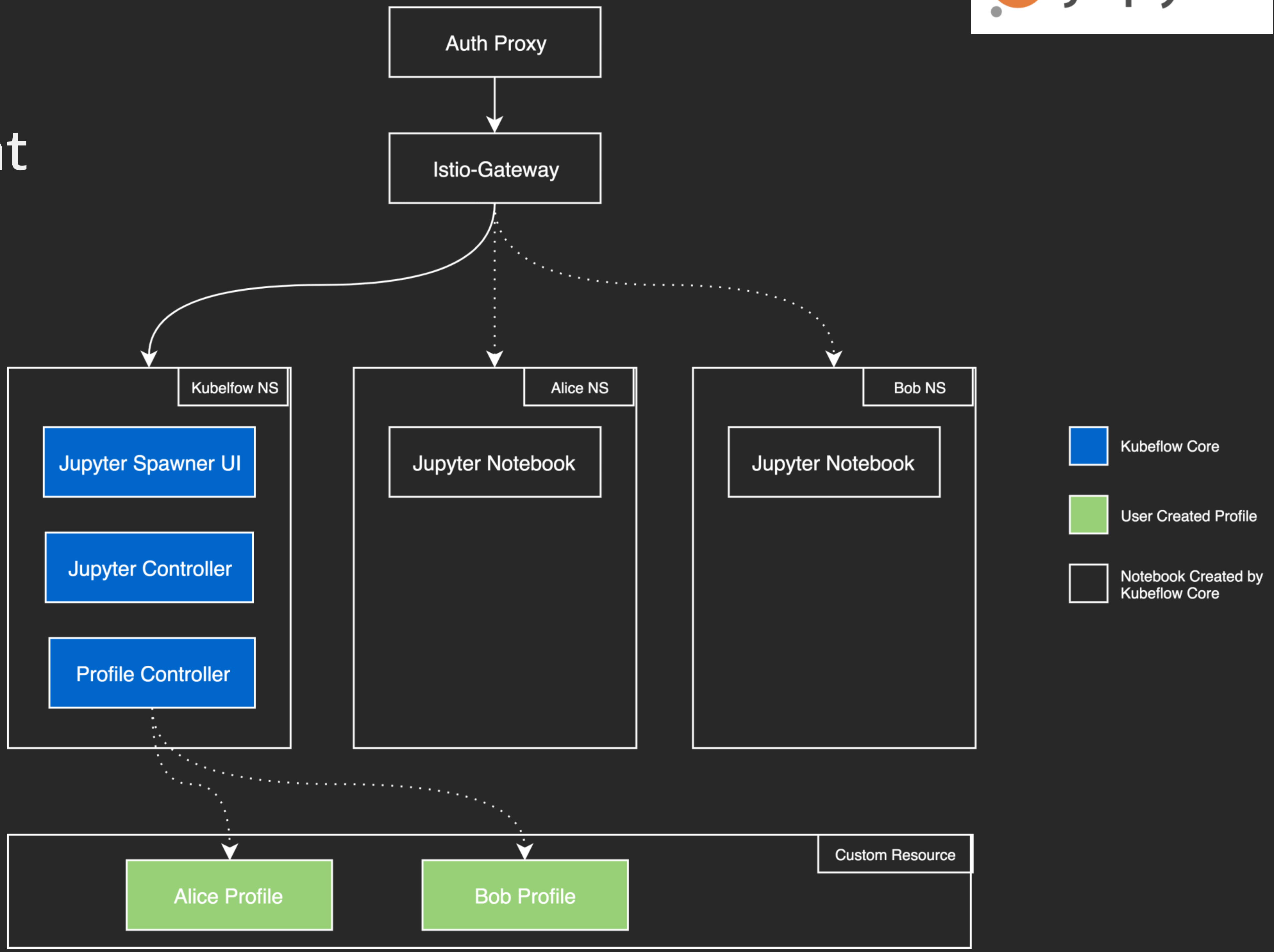
Kubeflow Components



Jupyter Notebook

Create and share docs that contain live code, equations, visualizations, and narrative text

- UI to manage notebooks
- Integrate with RBAC/IAM
- Ingress/Service Mesh



Fairing

Python SDK to build, train and deploy ML models

- Easily package ML training jobs
- Train ML models from notebook to k8s
- Streamline the model development process

Setup Kubeflow Fairing for training and prediction

```
from kubeflow import fairing
from kubeflow.fairing import TrainJob
from kubeflow.fairing.backends import KubeflowAWSBackend


from kubeflow import fairing

FAIRING_BACKEND = 'KubeflowAWSBackend'

AWS_ACCOUNT_ID = fairing.cloud.aws.guess_account_id()
AWS_REGION = 'us-west-2'
DOCKER_REGISTRY = '{}.dkr.ecr.{}.amazonaws.com'.format(AWS_ACCOUNT_ID, AWS_REGION)
S3_BUCKET = 'kubeflow-pipeline-data'
```

```
import importlib

if FAIRING_BACKEND == 'KubeflowAWSBackend':
    from kubeflow.fairing.builders.cluster.s3_context import S3ContextSource
    BuildContext = S3ContextSource(
        aws_account=AWS_ACCOUNT_ID, region=AWS_REGION,
        bucket_name=S3_BUCKET
    )

BackendClass = getattr(importlib.import_module('kubeflow.fairing.backends'), FAIRING_BACKEND)
```

Train an XGBoost model remotely on Kubeflow

```
from kubeflow.fairing import TrainJob
train_job = TrainJob(HousingServe, input_files=['ames_dataset/train.csv', "requirements.txt"],
                     docker_registry=DOCKER_REGISTRY,
                     backend=BackendClass(build_context_source=BuildContext))
train_job.submit()
```

Deploy the trained model to Kubeflow for prediction

```
from kubeflow.fairing import PredictionEndpoint
endpoint = PredictionEndpoint(HousingServe, input_files=['trained_ames_model.dat', "requirements.t
xt"],
                              docker_registry=DOCKER_REGISTRY,
                              service_type='ClusterIP',
                              backend=BackendClass(build_context_source=BuildContext))
endpoint.create()
```

https://github.com/aws-samples/eks-kubeflow-workshop/blob/master/notebooks/02_Fairing/02_06_fairing_e2e.ipynb

Katib – Hyperparameter Tuning

Hyperparameter are parameters external to the model to control the training, e.g. learning rate, batch size, epochs

Tuning finds a set of hyperparameters that optimizes an objective function, e.g. find the optimal batch size and learning rate to maximize prediction accuracy

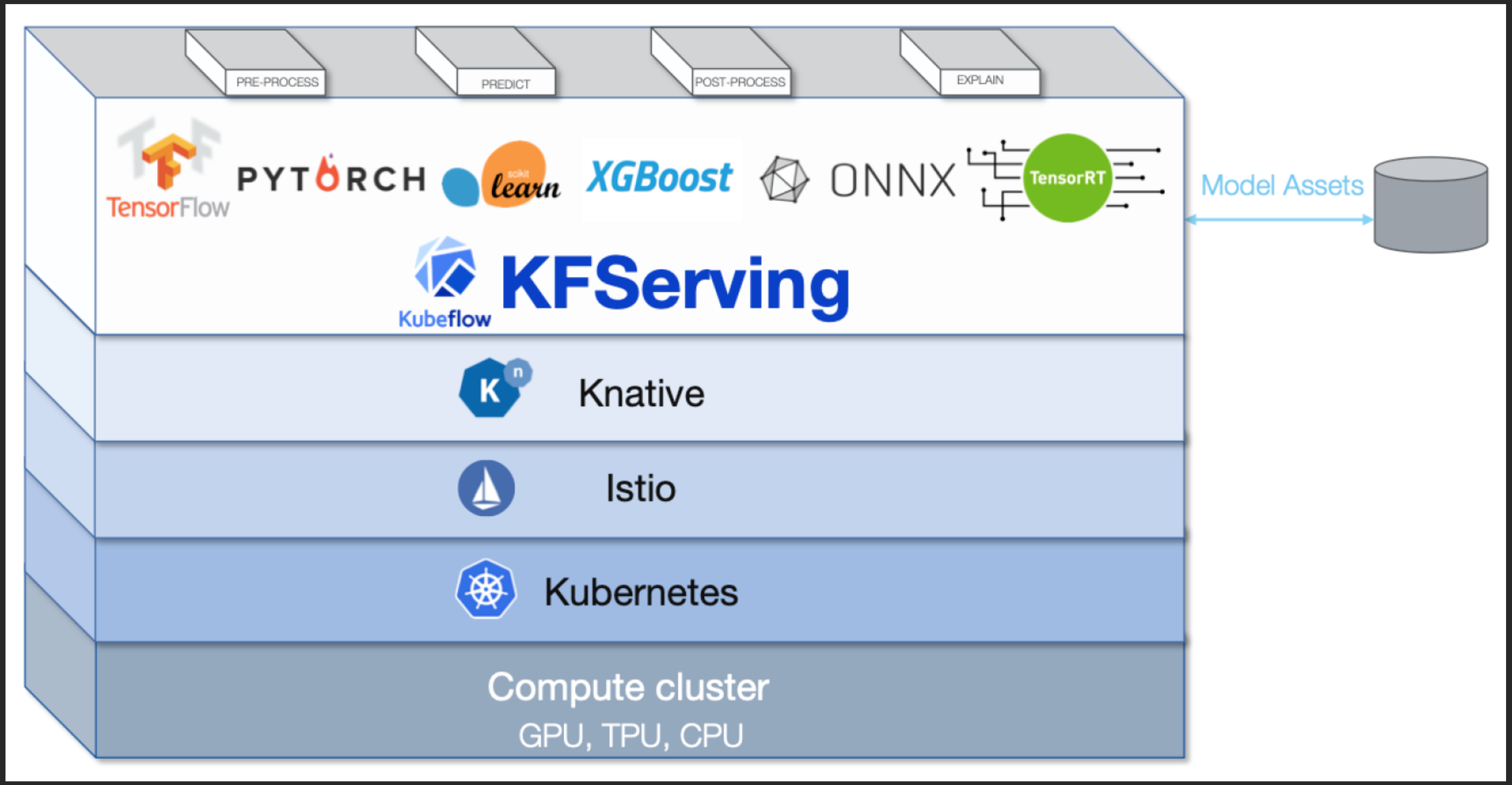
Hyperparameters

```
  parameters:
    - name: --lr
      parameterType: double
      feasibleSpace:
        min: "0.01"
        max: "0.03"
    - name: --num-layers
      parameterType: int
      feasibleSpace:
        min: "2"
        max: "5"
    - name: --optimizer
      parameterType: categorical
      feasibleSpace:
        list:
        - sgd
        - adam
        - ftrl
```

trialName	Validation-accuracy	accuracy	--lr	--num-layers	--optimizer
random-experiment-rfwwbnsd	0.974920	0.984844	0.013831565266960293	4	sgd
random-experiment-vxgwlgqq	0.113854	0.116646	0.024225789898529138	4	ftrl
random-experiment-wclrwlcq	0.979697	0.998437	0.021916171239020756	4	sgd
random-experiment-7lsc4pwb	0.113854	0.115312	0.024163810384272653	5	ftrl
random-experiment-86vv9vgv	0.963475	0.971562	0.02943228249244735	3	adam
random-experiment-jh884cxz	0.981091	0.999219	0.022372025623908262	2	sgd
random-experiment-sgtwhrgz	0.980693	0.997969	0.016641686851083654	4	sgd
random-experiment-c6vvz6dv	0.980792	0.998906	0.0264125850165842	3	sgd
random-experiment-vqs2xmfj	0.113854	0.105313	0.026629394628228185	4	ftrl
random-experiment-bv8lsh2m	0.980195	0.999375	0.021769570793012488	2	sgd
random-experiment-7vbnqc7z	0.113854	0.102188	0.025079750575740783	4	ftrl
random-experiment-kwj9drmg	0.979498	0.995469	0.014985919312945063	4	sgd

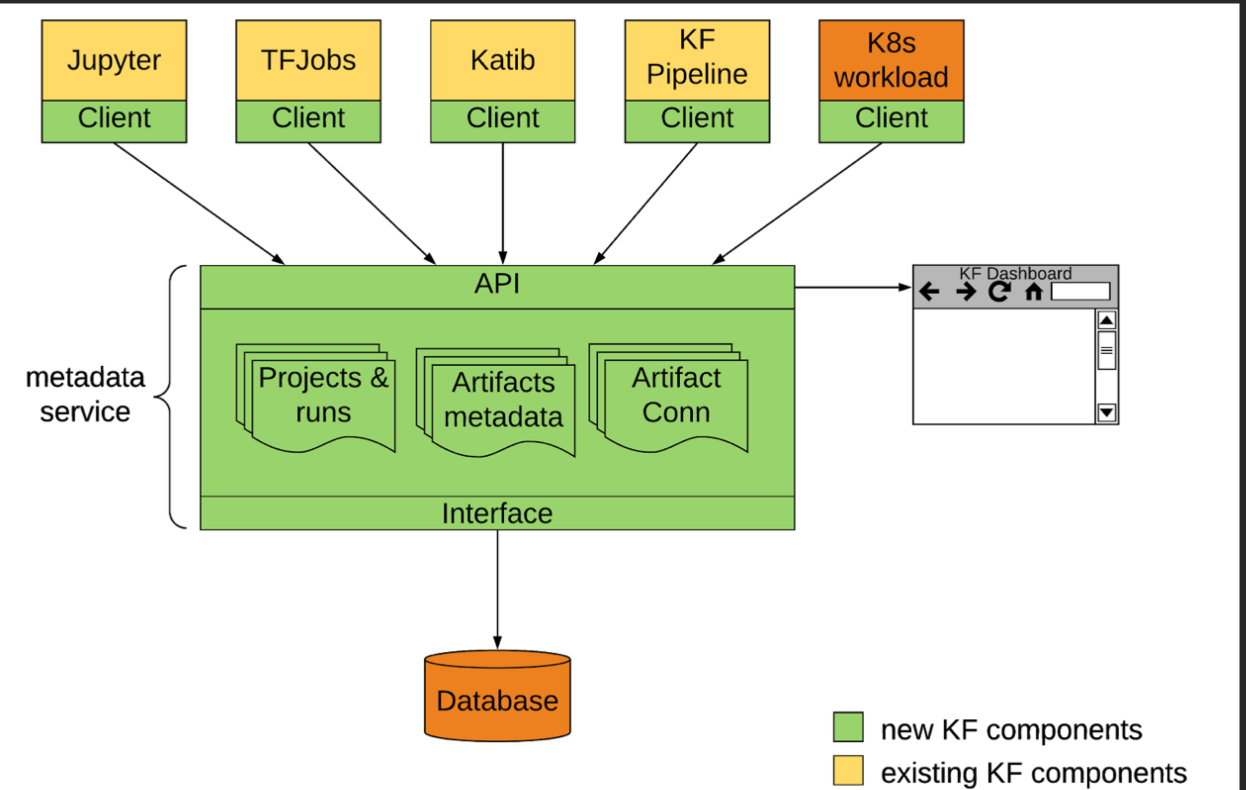
KFServing: Model serving and management

- Provides a Kubernetes CRD for serving ML models on arbitrary frameworks
- Encapsulates the complexity of autoscaling, networking and server configuration to bring features like scale to zero, transformations, and canary rollouts to your deployments
- Enables a simple, pluggable, and complete story for your production ML inference server by providing prediction, pre-processing, post-processing and explainability



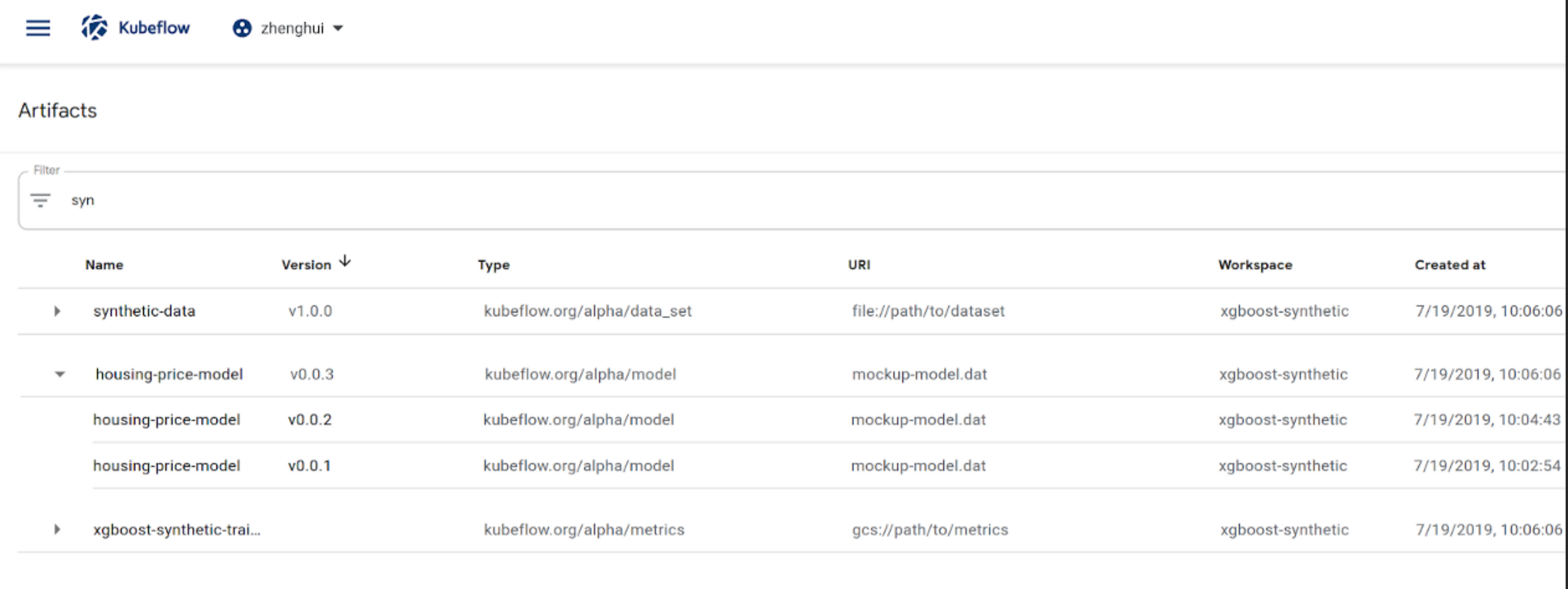
Metadata – Model Tracking

- Metadata schema to track artifacts related to execution contexts
- Metadata API for storing and retrieving metadata
- Client libraries for end-users to interact with the Metadata service from their Notebooks or Pipelines code



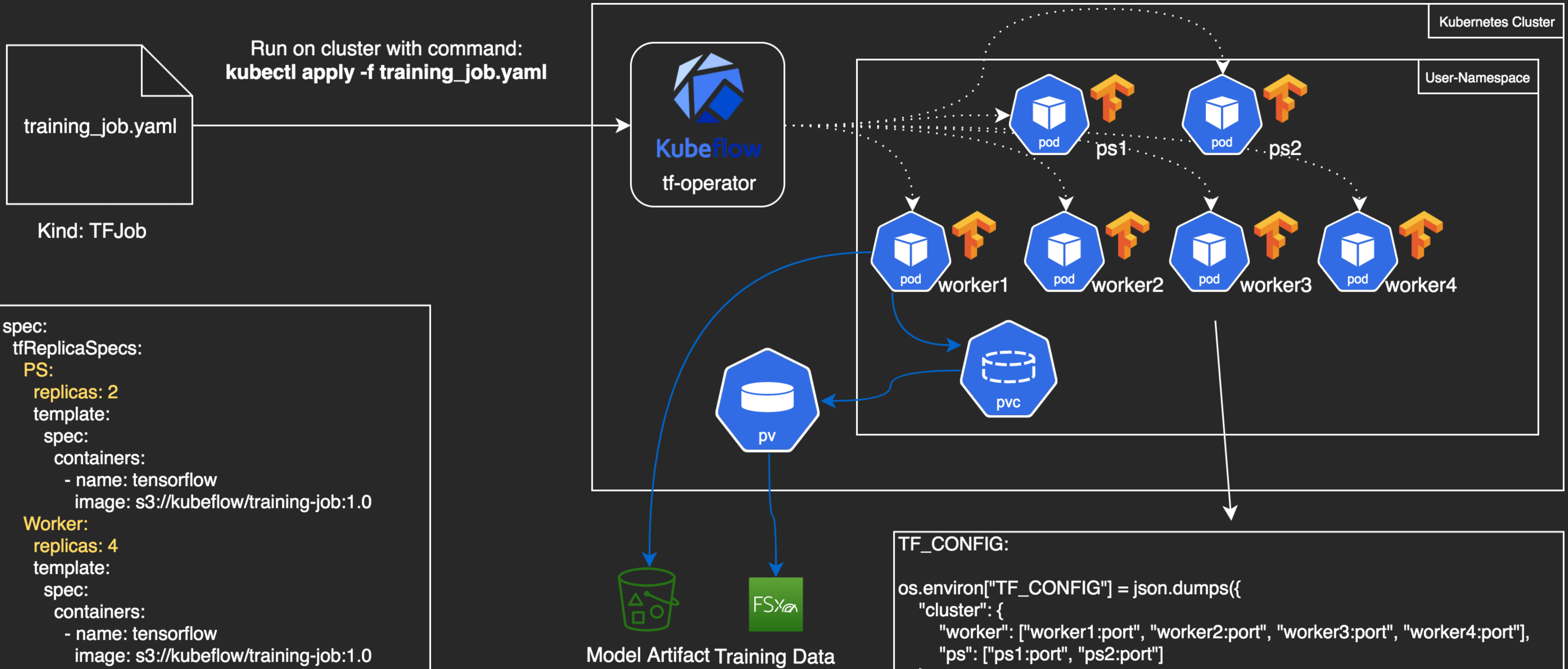
Kubeflow ML Metadata Architecture

```python
def create_execution(self):
    workspace = metadata.Workspace(
    # Connect to metadata-service in namesapce kubeflow in k8s cluster.
    backend_url_prefix="metadata-service.kubeflow:8080",
    name="xgboost-synthetic",
    description="workspace for xgboost-synthetic artifacts and executions")

    r = metadata.Run(
        workspace=workspace,
        name="xgboost-synthetic-faring-run" + datetime.utcnow().isoformat("T"),
        description="a notebook run")

    return metadata.Execution(
        name = "execution" + datetime.utcnow().isoformat("T"),
        workspace=workspace,
        run=r,
        description="execution for training xgboost-synthetic")
```



Name	Version	Type	URI	Workspace	Created at
synthetic-data	v1.0.0	kubeflow.org/alpha/data_set	file://path/to/dataset	xgboost-synthetic	7/19/2019, 10:06:06
housing-price-model	v0.0.3	kubeflow.org/alpha/model	mockup-model.dat	xgboost-synthetic	7/19/2019, 10:06:06
housing-price-model	v0.0.2	kubeflow.org/alpha/model	mockup-model.dat	xgboost-synthetic	7/19/2019, 10:04:43
housing-price-model	v0.0.1	kubeflow.org/alpha/model	mockup-model.dat	xgboost-synthetic	7/19/2019, 10:02:54
xgboost-synthetic-trai...		kubeflow.org/alpha/metrics	gcs://path/to/metrics	xgboost-synthetic	7/19/2019, 10:06:06

Distributed Training

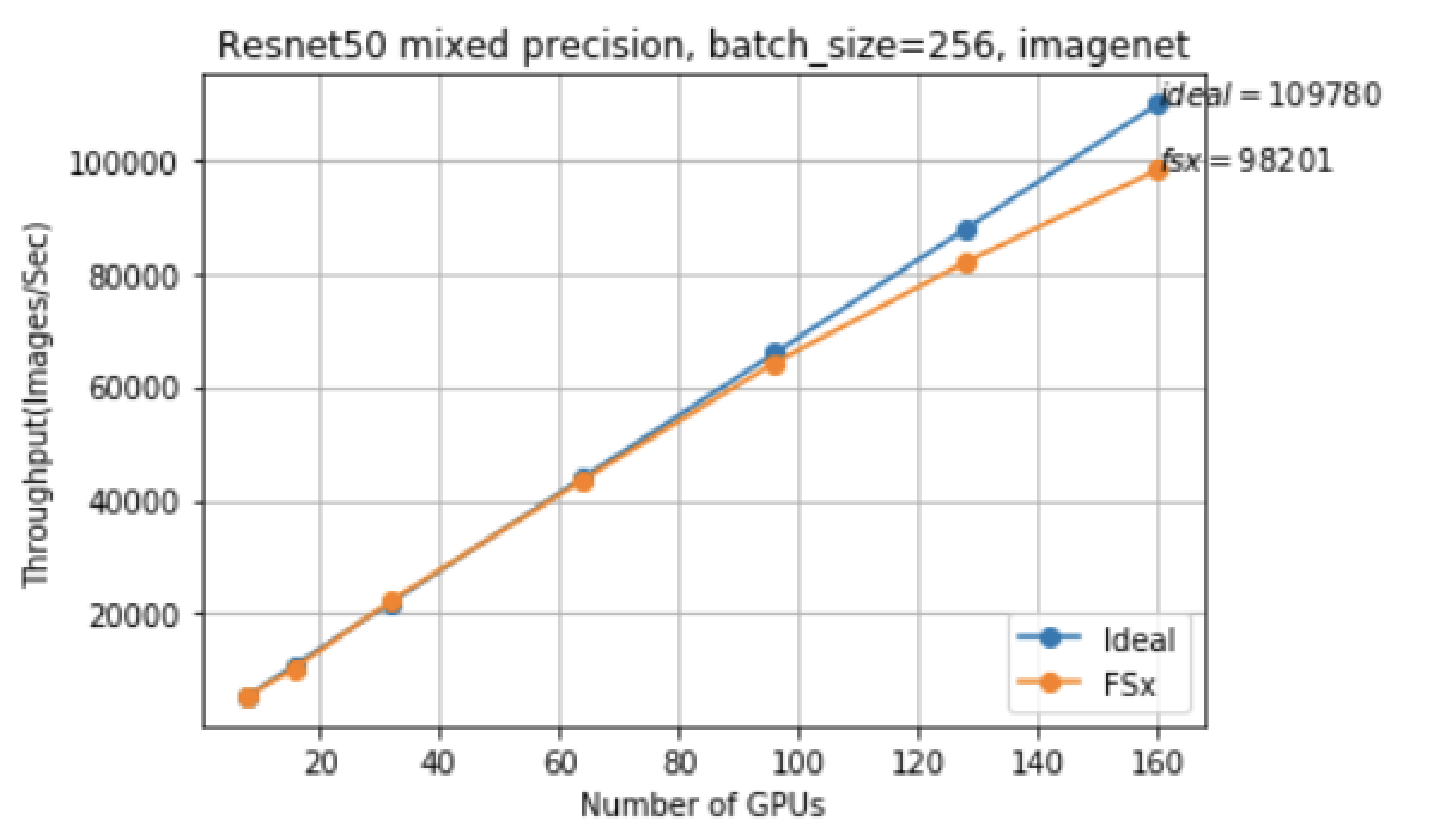



```yaml
spec:
  tfReplicaSpecs:
    PS:
      replicas: 2
      template:
        spec:
          containers:
            - name: tensorflow
              image: s3://kubeflow/training-job:1.0
    Worker:
      replicas: 4
      template:
        spec:
          containers:
            - name: tensorflow
              image: s3://kubeflow/training-job:1.0
```

TF_CONFIG:

```python
os.environ["TF_CONFIG"] = json.dumps({
    "cluster": {
        "worker": ["worker1:port", "worker2:port", "worker3:port", "worker4:port"],
        "ps": ["ps1:port", "ps2:port"]
    },
   "task": {"type": "worker", "index": 3}
})
```

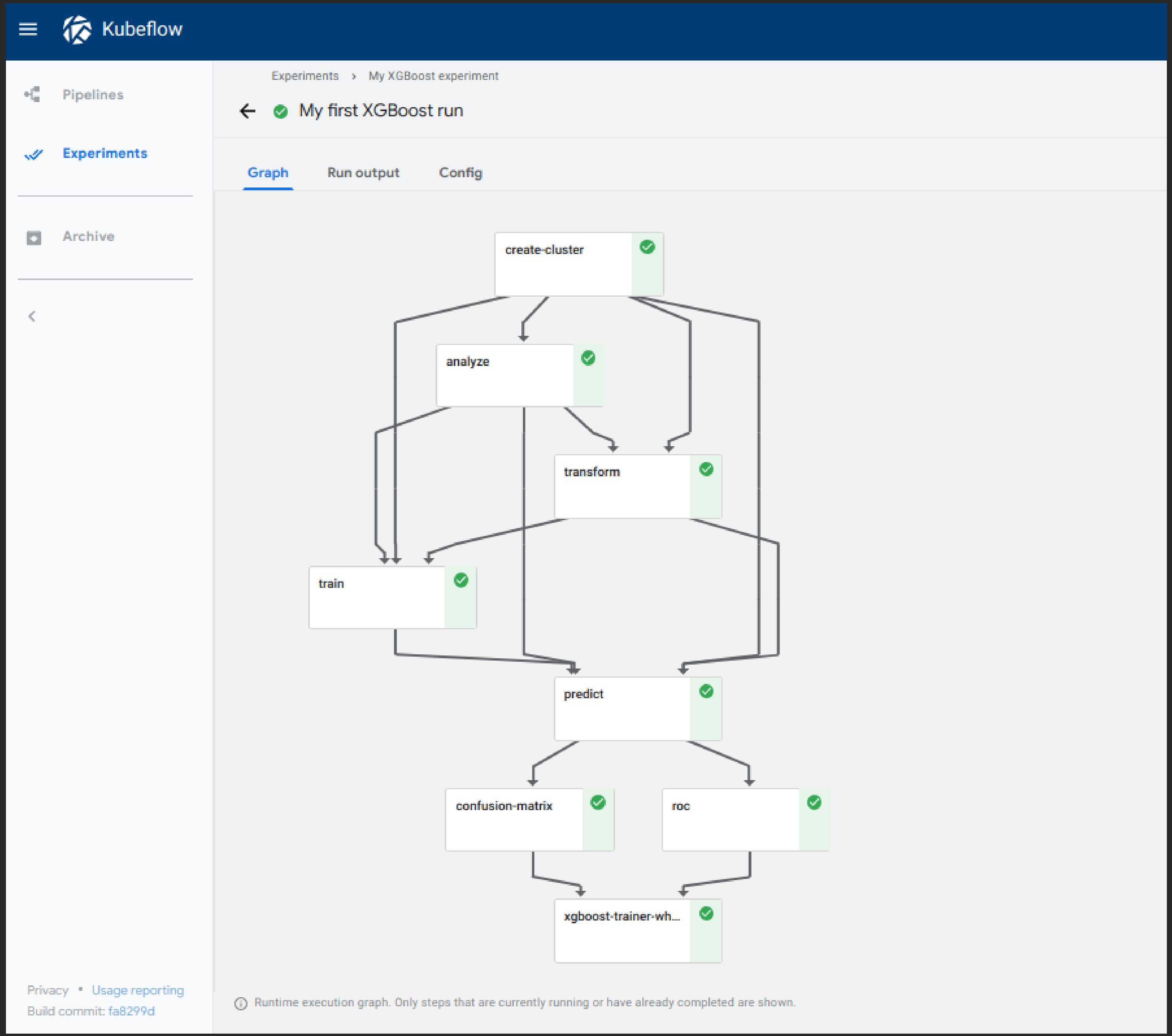
Best Practices for Optimizing Distributed Deep Learning Performance on Amazon EKS



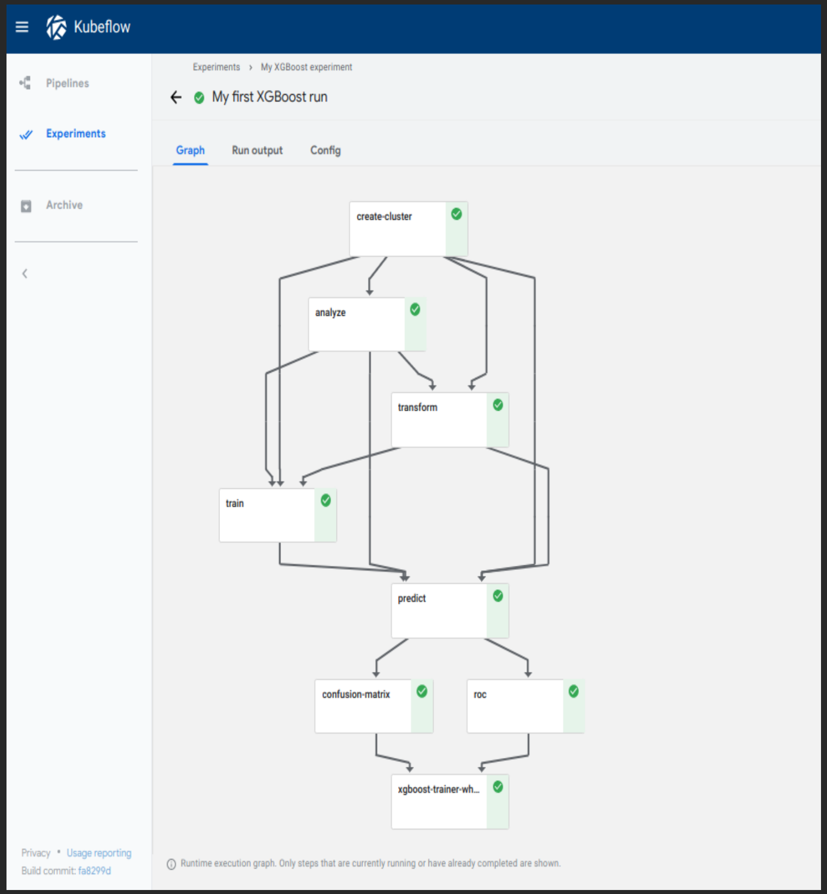
https://aws.amazon.com/blogs/opensource/optimizing-distributed-deep-learning-performance-amazon-eks/

Pipelines – Machine Learning Job Orchestrator

- Compose, deploy, and manage end-to-end ML workflows
  - End-to-end orchestration
  - Easy, rapid, and reliable experimentation
  - Easy re-use
- Built using Pipelines SDK
  - `kfp.compiler`, `kfp.components`, `kfp.Client`
- Uses Argo under the hood to orchestrate resources



Creating Kubeflow Pipeline Components



Pipeline decorator

Pipeline function

Pipeline component

Compile pipeline

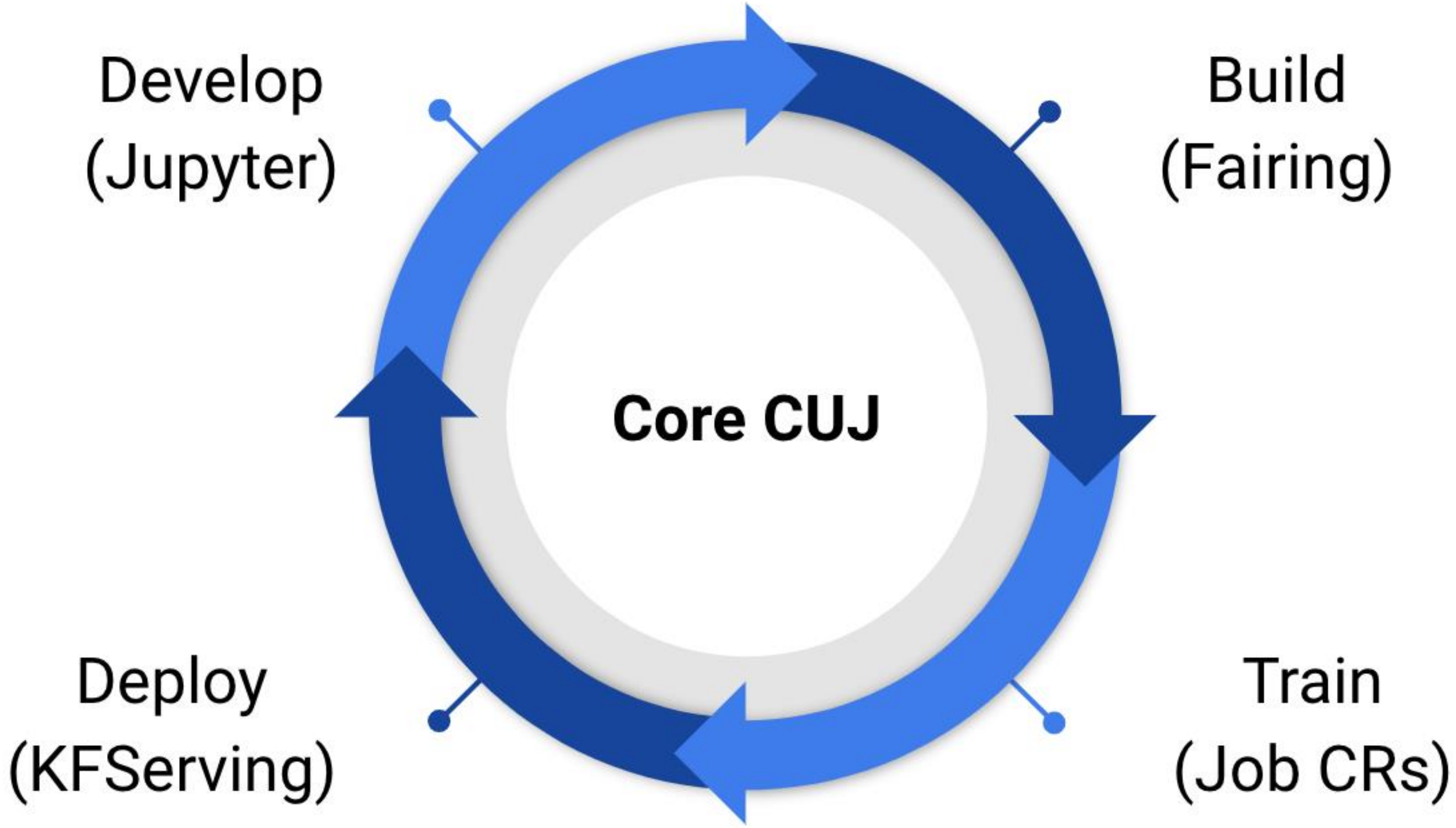
```
@dsl.pipeline(
  name='Sample Trainer',
  description=''
)

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')
```

Making Kubeflow a first class citizen on AWS

- Centralized and unified Kubernetes cluster logs in Amazon CloudWatch
- External traffic and authentication management with ALB Ingress Controller
- TLS and authentication with AWS Certificate Manager and AWS Cognito
- In-built FSx CSI driver w/S3 data repository integration to optimize training performance
- Elastic File System integration for common data sharing in JupyterHub
- Easier and customizable Kubeflow installation with kfctl and Kustomize support
- Kubeflow Pipeline integration with AWS Services – Amazon EMR, Athena, SageMaker
- Add ECR integration to Kubeflow Fairing
- Jupyter Notebook images with AWS CLI installed and ECR support
- Auto detect GPU worker nodes and install NVIDIA device plugin

https://www.kubeflow.org/docs/aws/

Kubeflow 1.0 Arriving January 2020



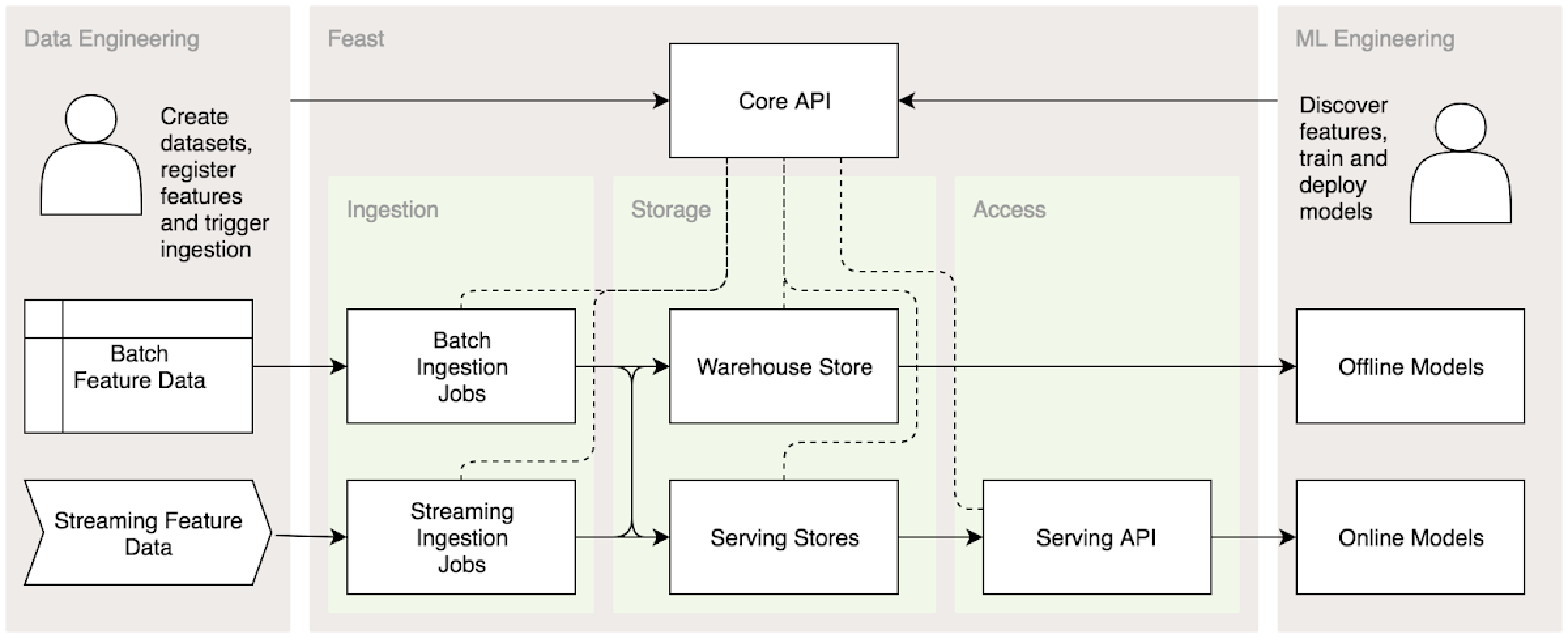
http://bit.ly/kf_roadmap

AWS Kubeflow Roadmap

Kubeflow v1.0 - Theme: Enterprise Readiness

- E2E examples and increased documentation on Kubeflow site
- Upstream testing for Kubeflow on AWS
- Support DIY K8S on AWS
- IAM Roles for Service Accounts integration with Jupyter notebooks
- Support for managed contributors

Feature store - Feast



- Discoverability and reuse of features
- Standardization of features
- Access to features for training and serving
- Consistency between training and serving

Enterprise

Machine Learning Stack on EKS

ML Platform

Kubeflow

- Jupyter
- Training Operator
- KFServing
- Fairing
- Katib
- Metadata

Enterprise

- Elastic Training
- vGPU sharing
- Data Preprocessing
- Feature Store
- GPU Monitoring
- MLOps

Container Platform

- Amazon EKS
- Optimized GPU AMI
- Auto Scaling
- Deep Learning Container
- FSx CSI Plugin

ML Frameworks + Infrastructure

FRAMEWORKS

- TensorFlow
- mxnet
- PYTORCH

INTERFACES

- GLUON
- Keras

INFRASTRUCTURE

- EC2 P3 & P3dn
- EC2 G4
- EC2 C5
- FPGAs
- FSx Lustre
- EFA
- Inferentia

Future Ideas

- Elastic Training
- Virtual GPU Device Plugin
- GPU Monitoring
- MLOps

Elastic Training

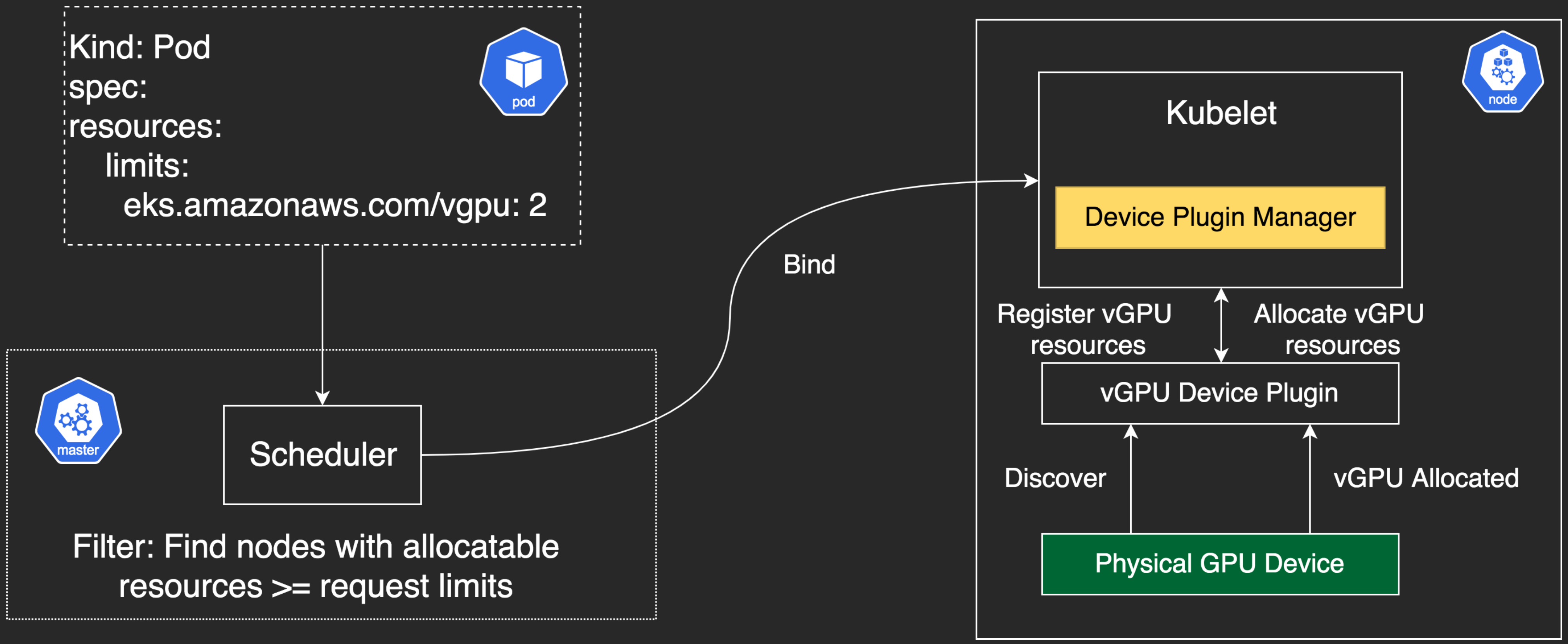
Fault Tolerant

- Enable Job Priority & Preemption
- Unlock SLA Critical Jobs to run on Spot instances

Elastic Scheduling

- Improve the GPU utilization rate

Virtual GPU for ML inference



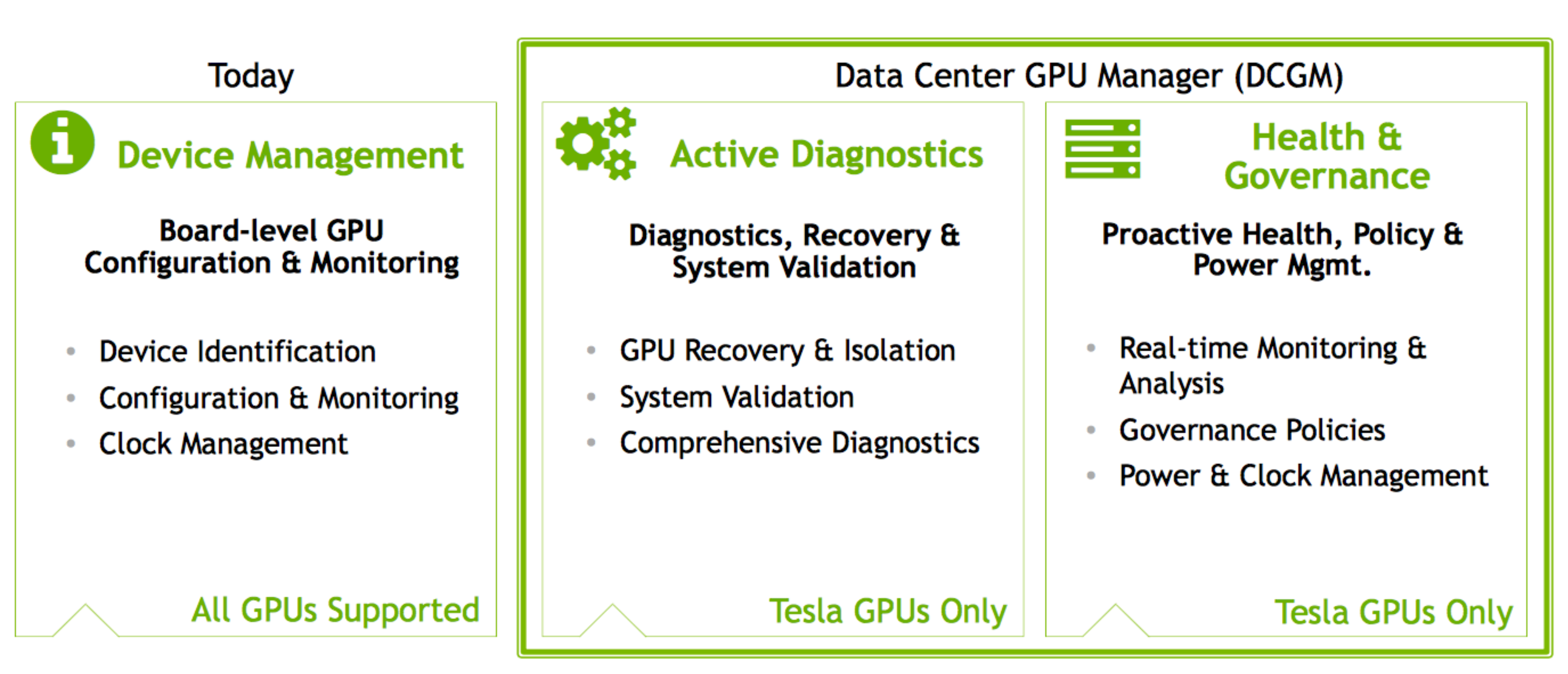
- Map physical GPU to virtual GPUs in vGPU Device Plugin

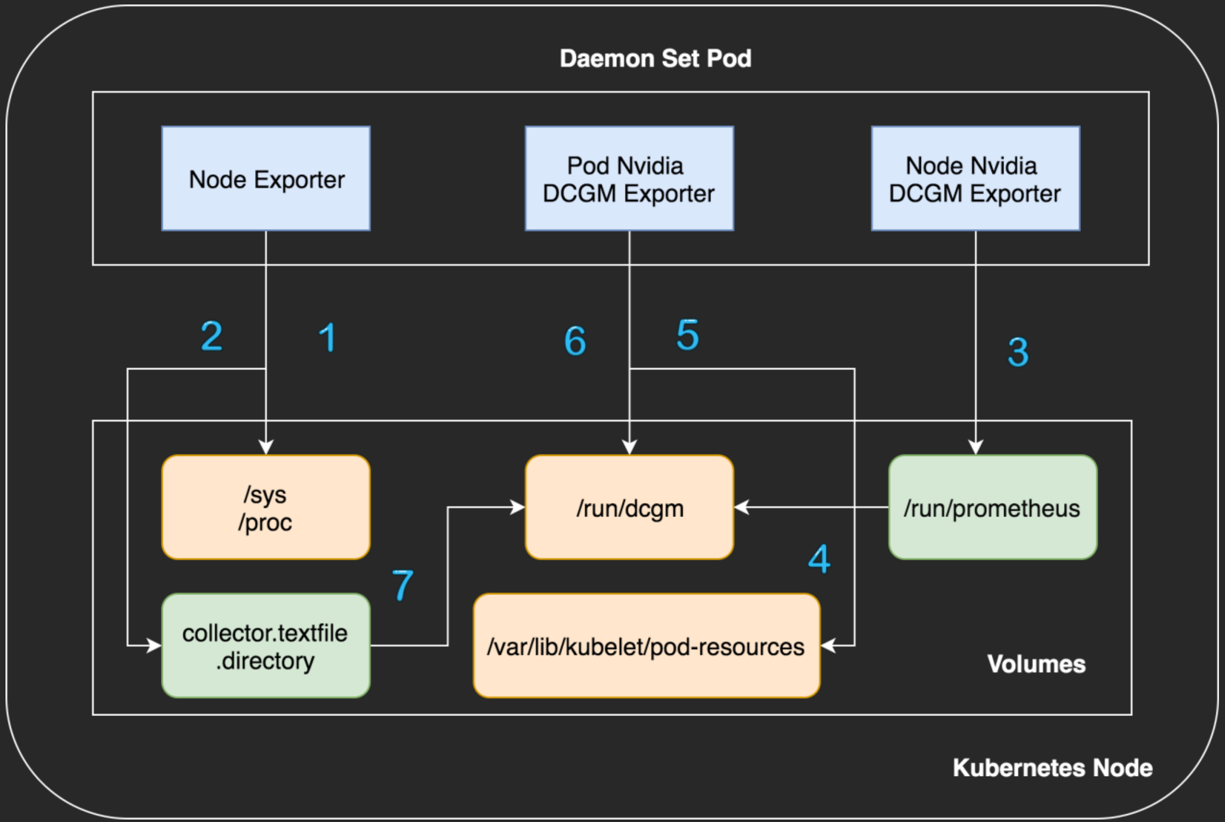
  e.g. Physical GPU = 100 vGPUs
- vGPU Device Plugin reports discovered extended resources to API Server
- Kubernetes Scheduler schedule pods request Custom Resource e.g eks.amazonaws.com/vgpu
- vGPU Device Plugin allocate Physical GPU

GPU Monitoring

Monitoring Stack

- Nvidia Management Library (NVML)
- Data Center GPU Manager (DCGM)
- CAdvisor accelerator metrics (CAdvisor)





MLOps

Data Scientists

Hybrid, integrated, cloud-based dev env

Version control (scripts & artifacts)

DevOps

Seamless deployment of hybrid pipelines

Trigger-based scheduling & orchestration of runs

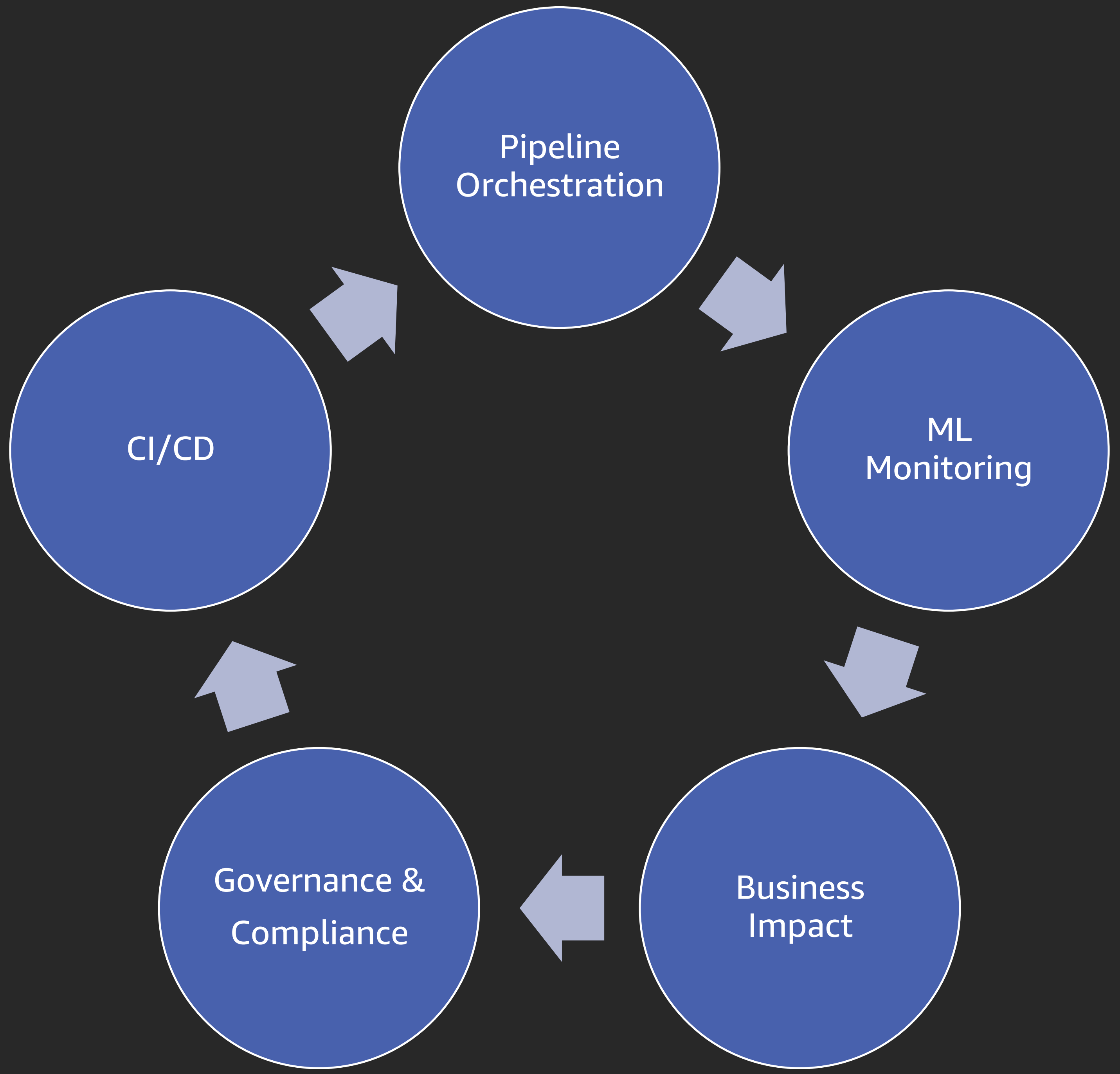
Monitoring & dashboard

Version control (runs & pipelines)

Business Analyst

Ensure Compliance

Explain-ability



Goal: Graduate ML into a first class citizen of software development

Thank you!

Jiaxin Shan
Software Engineer
AWS

Mike Stefaniak
Product Manager
AWS

Arun Gupta
Open Source Technologist
AWS

Please complete the session survey in the mobile app.

AWS re:Invent

aws