AWS re:Invent

CON306

Building machine-learning infrastructure on Amazon EKS with Kubeflow

Yaniv Donenfeld
Sr. BDM, Container Services
AWS

Jean Marie Ferdegue
Director, Platform Engineering, Babylon Health

Jeremie Vallee
AI Infrastructure Lead
Babylon Health

AWS re:Invent

aws

Agenda

AWS

- The AWS ML stack
- Machine learning - Why Kubernetes?
- Kubeflow and Kubeflow pipelines
- Common requirements
- Making AWS a first-class citizen of Kubeflow

Babylon Health

- The mission
- Challenges
- Our solution
- Next steps

The AWS ML stack

Broadest and deepest set of capabilities

AI services

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

ML services

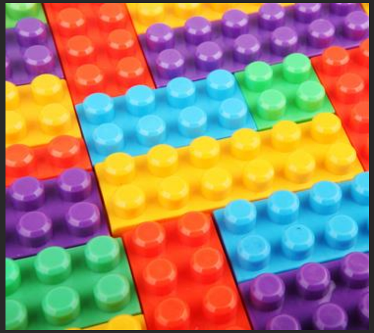
Amazon SageMaker	Ground Truth	Notebooks	Algorithms + Marketplace	Reinforcement learning	Training	Optimization	Deployment	Hosting

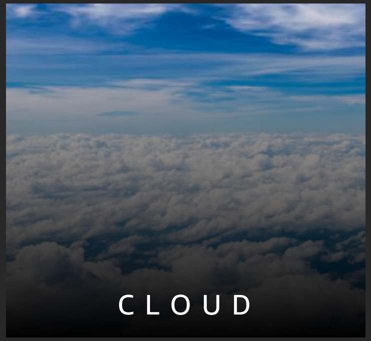
ML frameworks + infrastructure

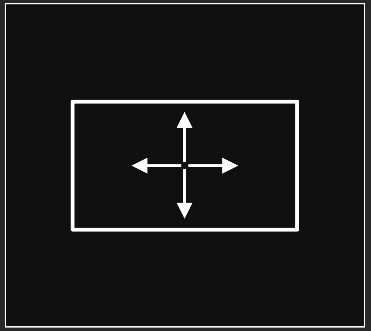
FRAMEWORKS	INTERFACES
TensorFlow, mxnet, PYTORCH	GLUON, Keras

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

Why machine learning on Kubernetes?







Composability

Portability

Scalability

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

Amazon EKS: run Kubernetes in cloud

Managed Kubernetes control plane, attach data plane

Native upstream Kubernetes experience

Platform for enterprises to run production-grade workloads

Integrates with additional AWS services

Which user are you?

ML practitioner

ML ops team member

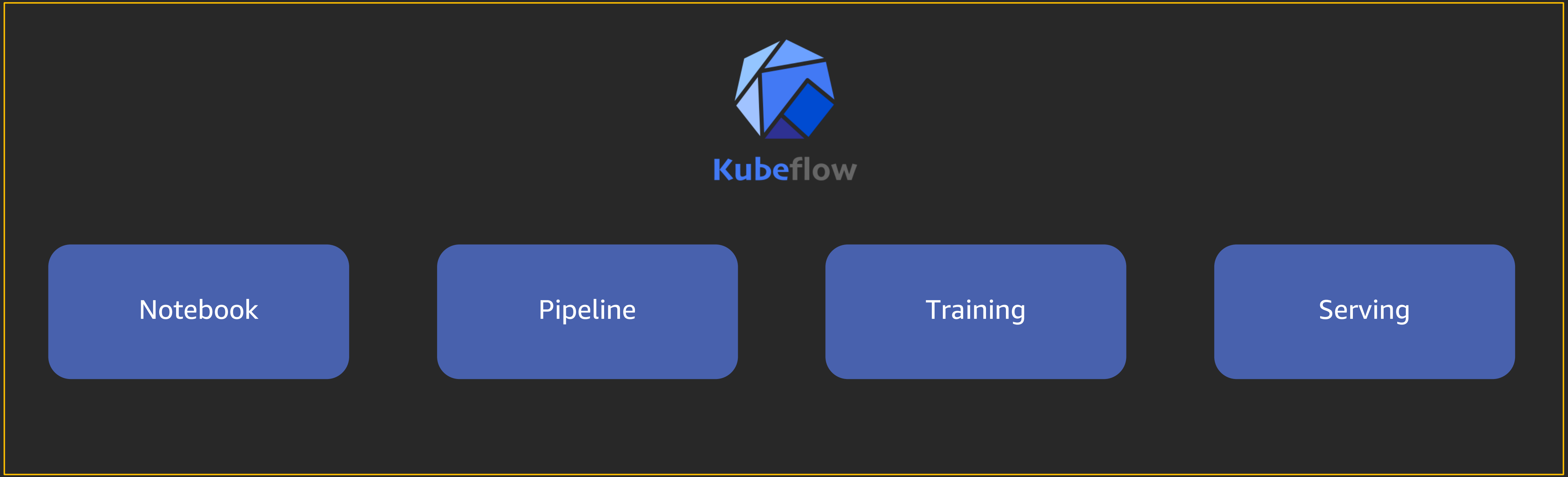
ML practitioners would like to have…

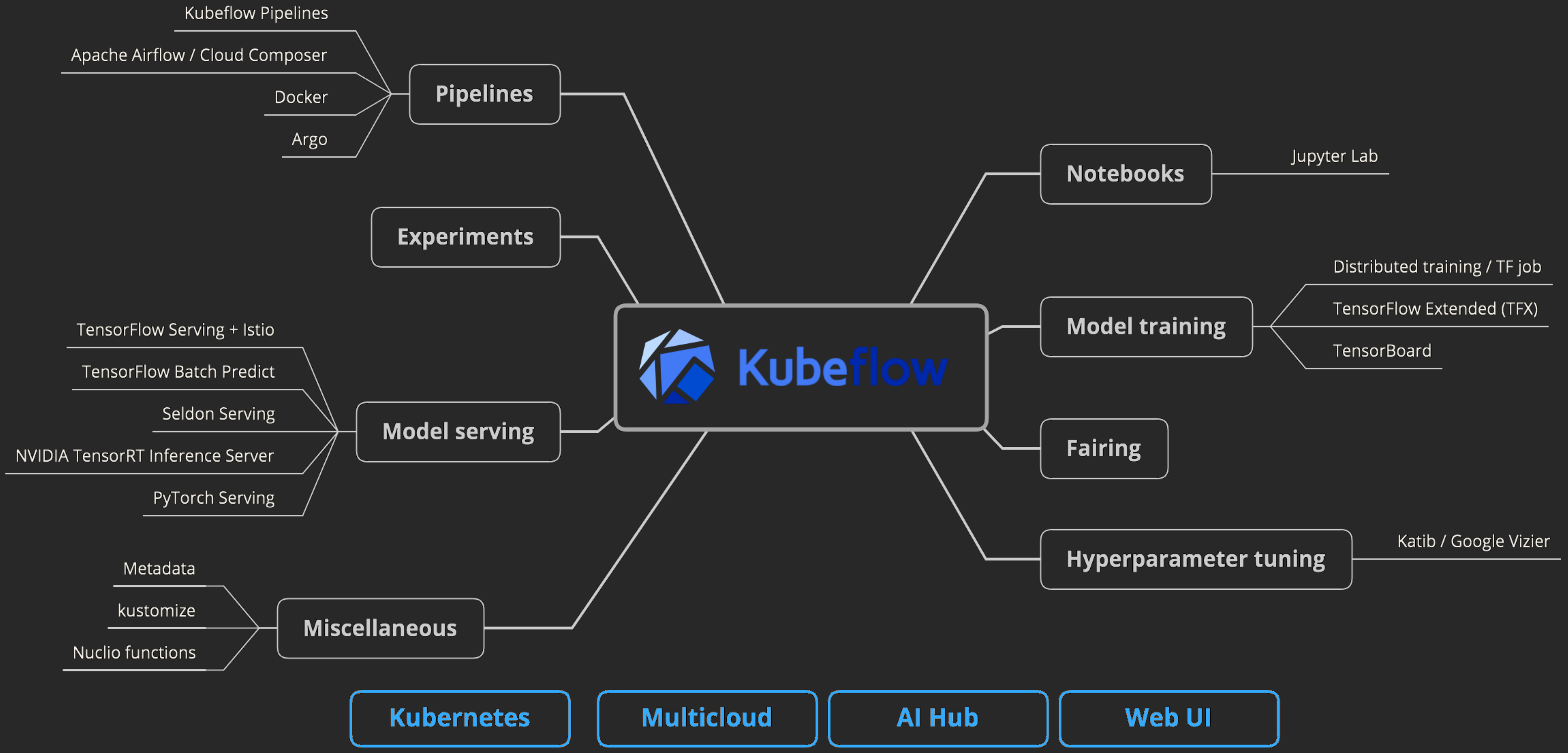
- Infrastructure abstraction
- Sharing and collaboration
- End-to-end workflow (build, train, test, deploy)
- Experiment management
- Hyperparameter tuning/optimization

ML ops teams would need to implement…

- An end to end platform
- Infrastructure abstraction
- Authentication and authorization support (multi-tenant access)
- Resource and quota management

Introducing Kubeflow





Version 1.1 20190807 @MichalBrys

Prototyping

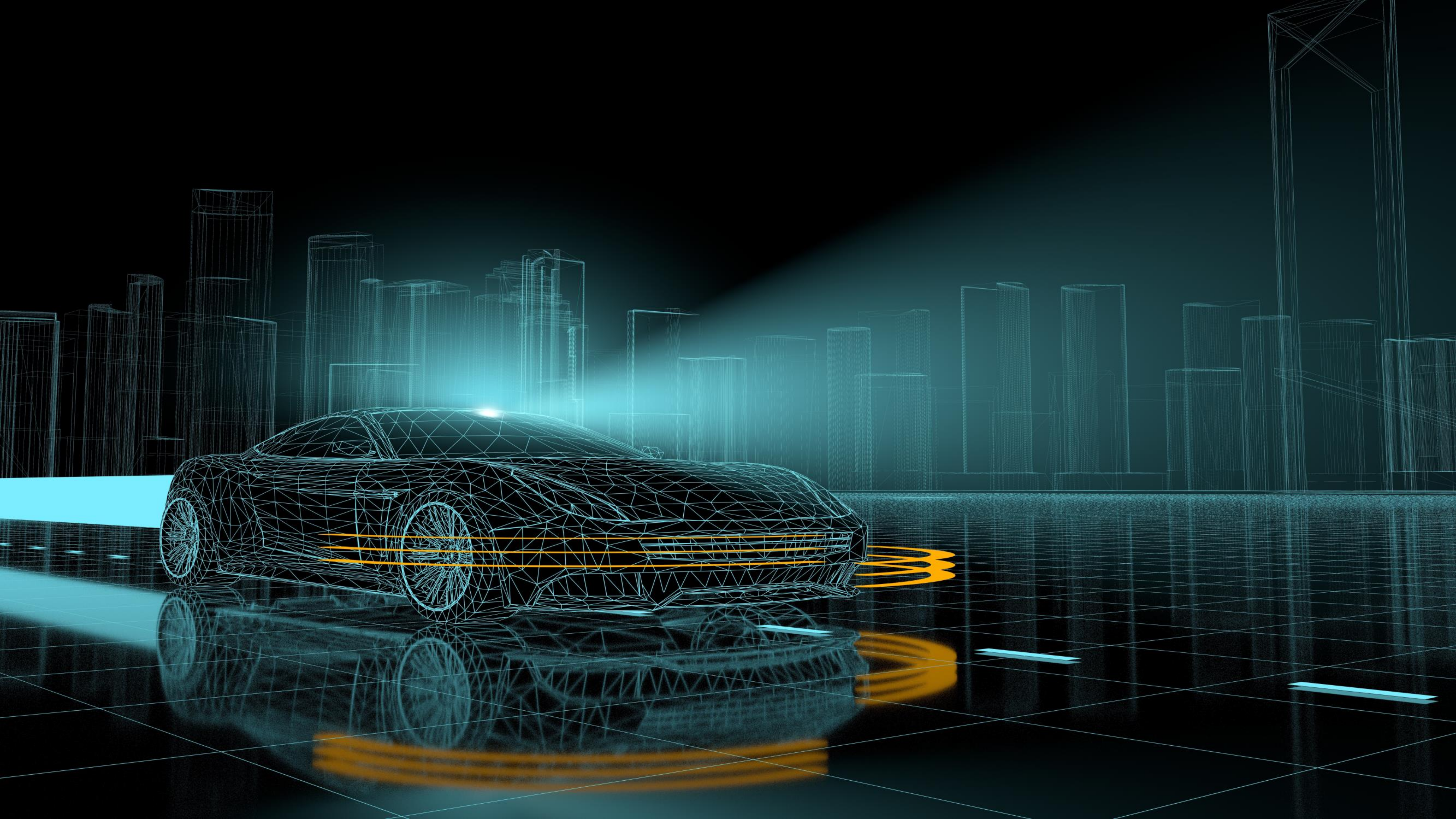
Jupyter / JupyterHub

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

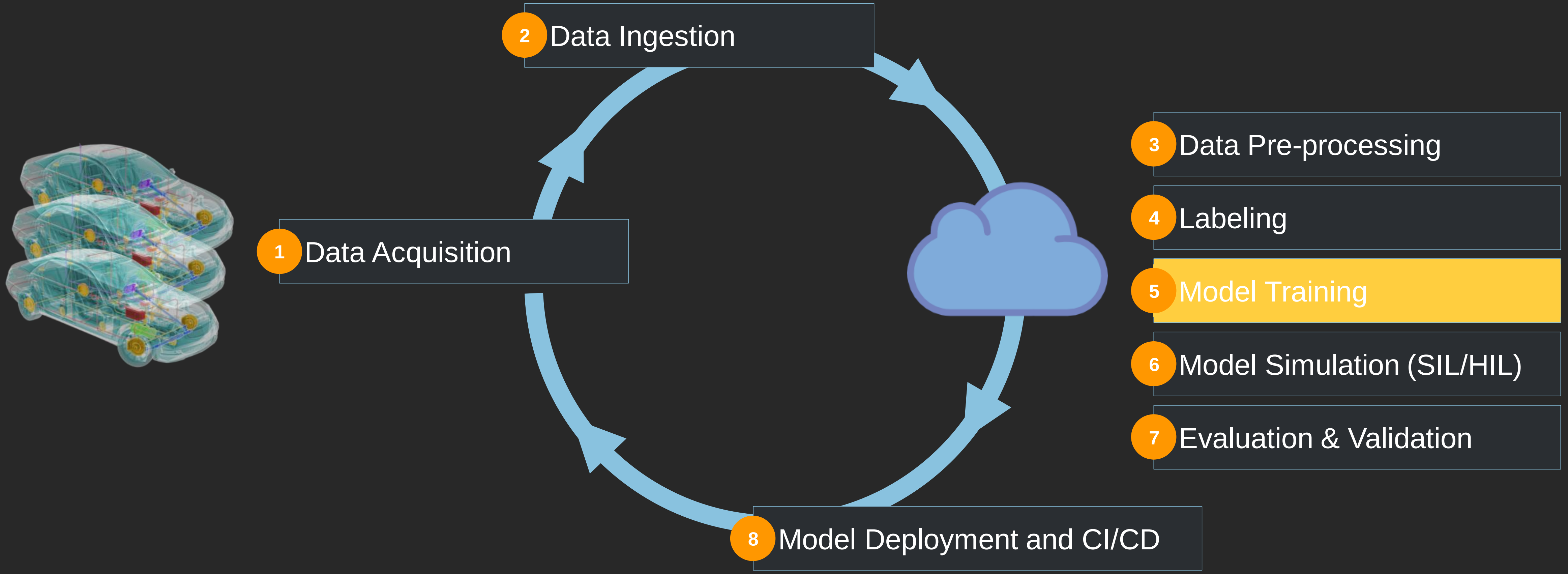
👉 EFS for reusing training data and results

👉 Built-in AWS CLI and ECR support

Training



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

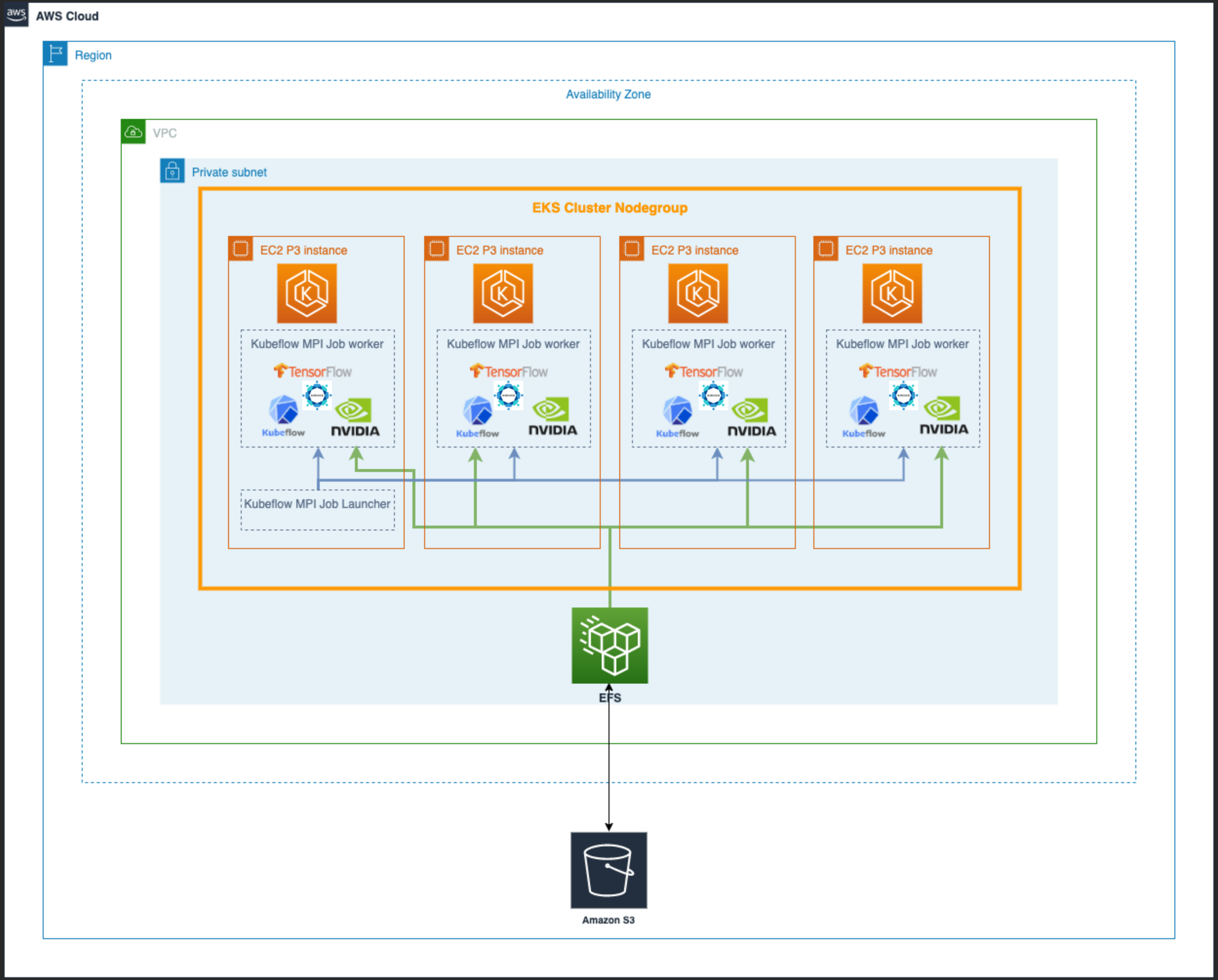
Distributed Training Challenges

- Single GPU code ➔ multiple
- Dataset Copying time
- Dataset Sharing and Reuse

Horovod + MPIJob

Use FSx Lustre / EFS

👉 Built-in CSI driver with S3 integration


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
- Dataset Copying time
- Dataset Sharing and Reuse

Horovod + MPIJob

Use FSx Lustre / EFS

Built-in CSI driver with S3 integration

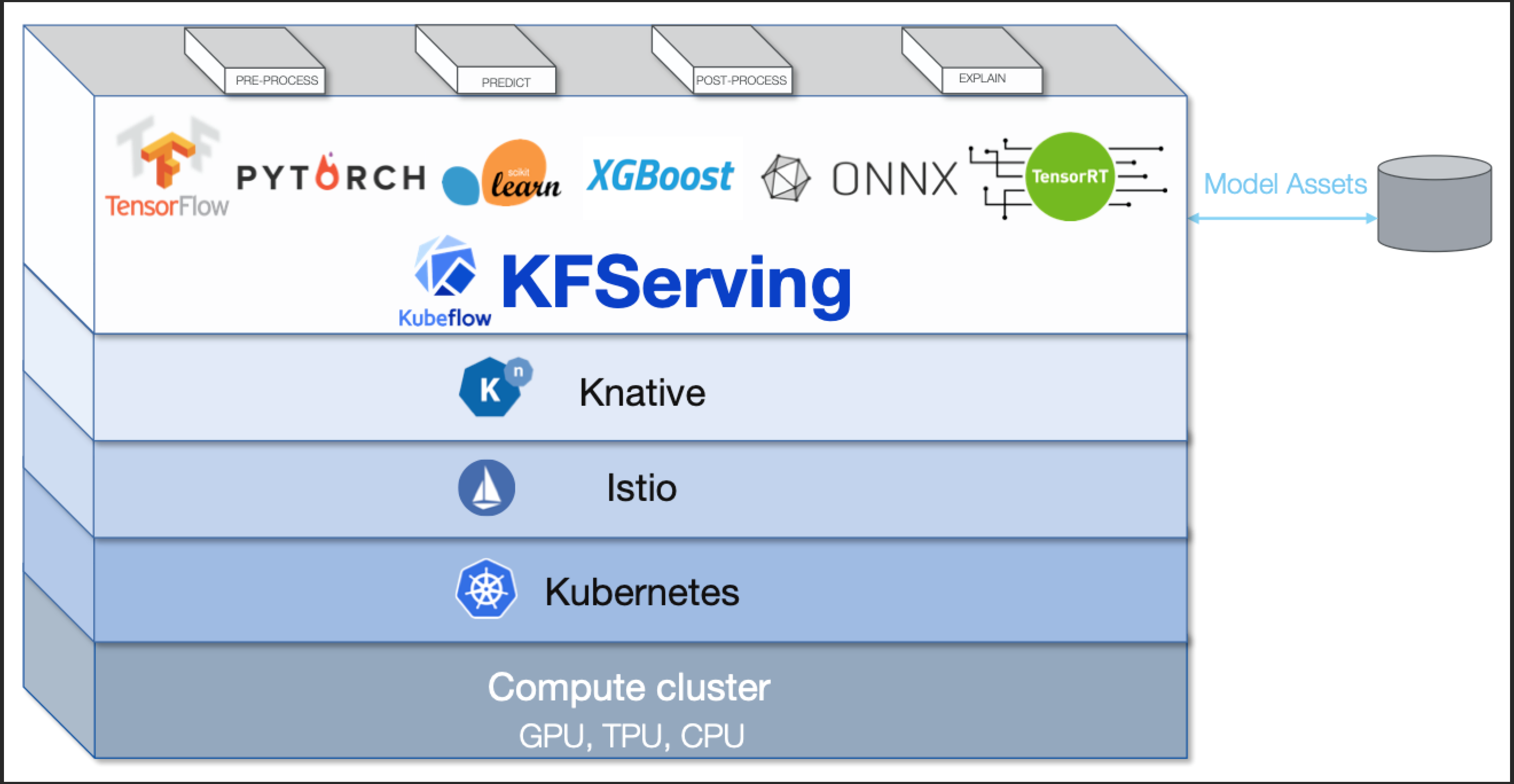
Want to run Distributed Training on EKS?

Distributed TensorFlow training using Kubeflow on Amazon EKS

Ajay Vohra
Principal SA - Vision/AI/ML

Inference

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"
```

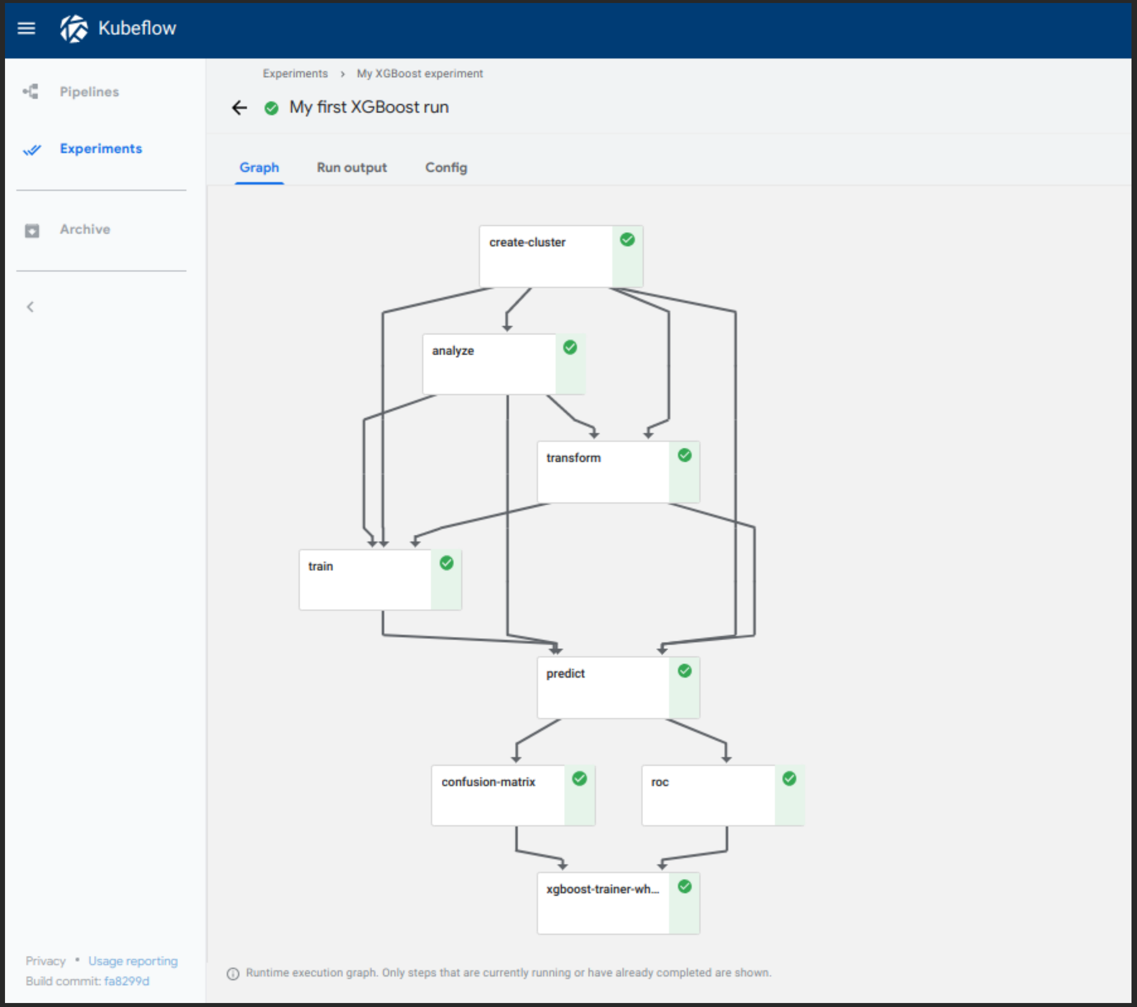
```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "flowers-sample"
spec:
  default:
    tensorflow:
      storageUri: "gs://kfserving-samples/models/tensorflow/flowers"
```

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

End-to-end ML

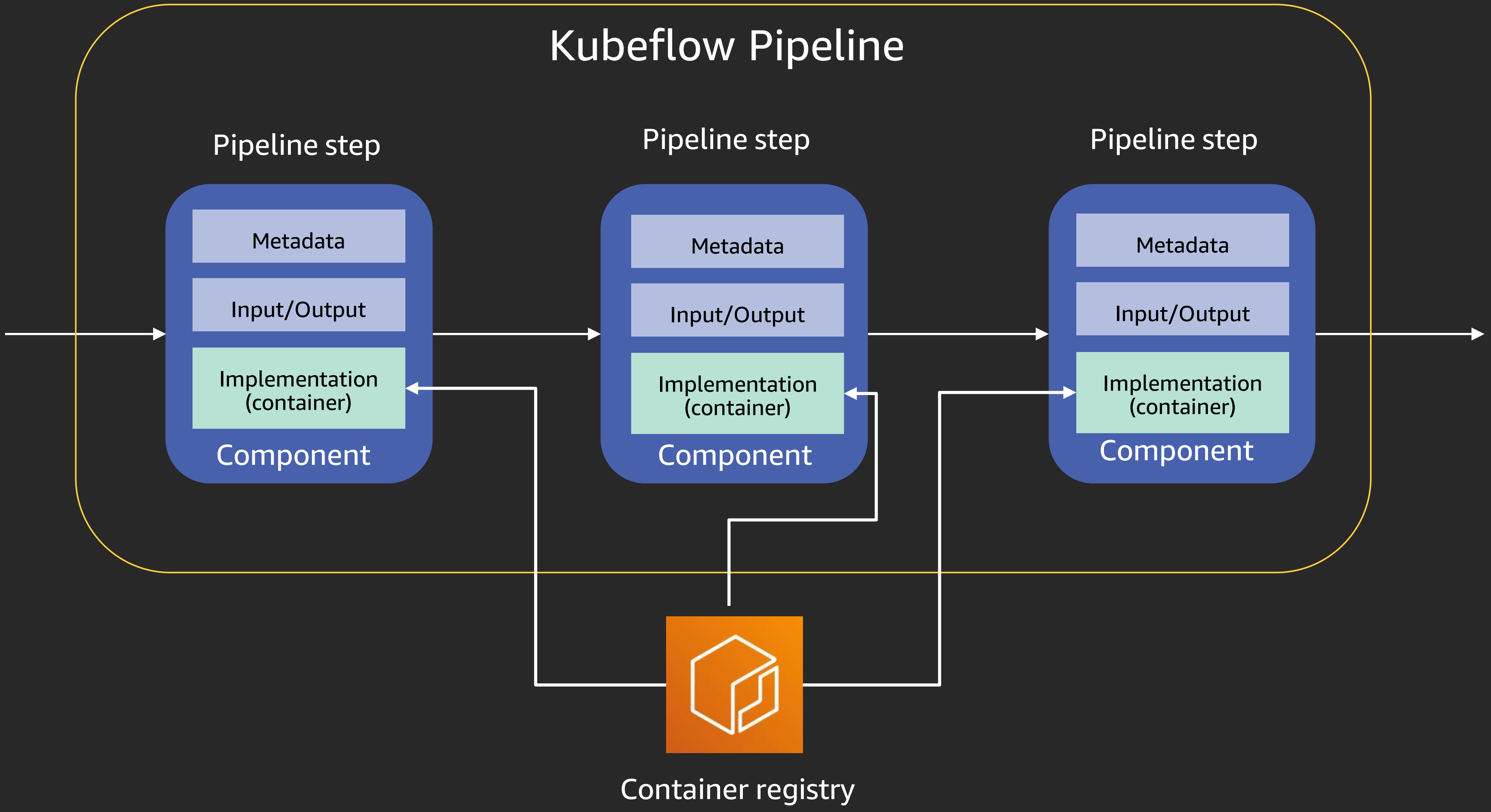
Kubeflow Pipelines

- 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

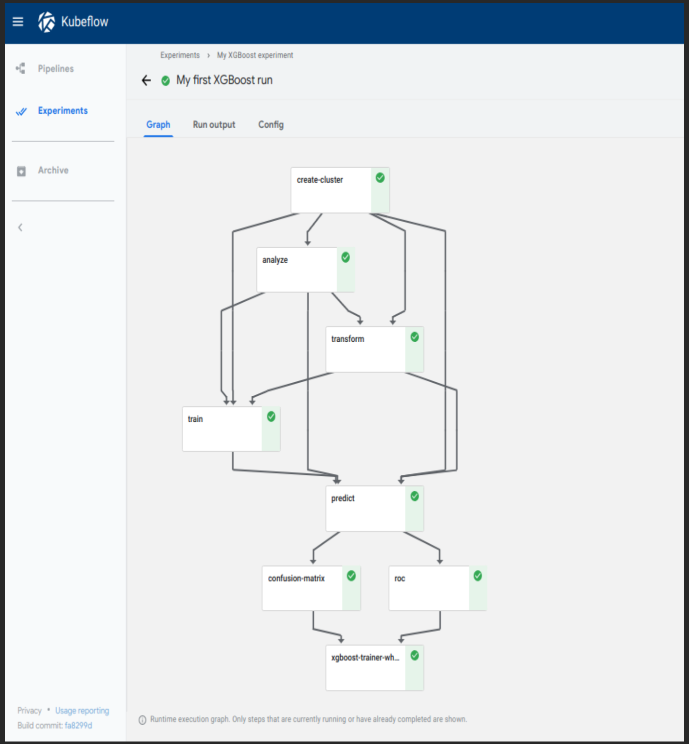


Runtime execution graph. Only steps that are currently running or have already completed are shown.

Pipeline component



Creating a pipeline



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')
```

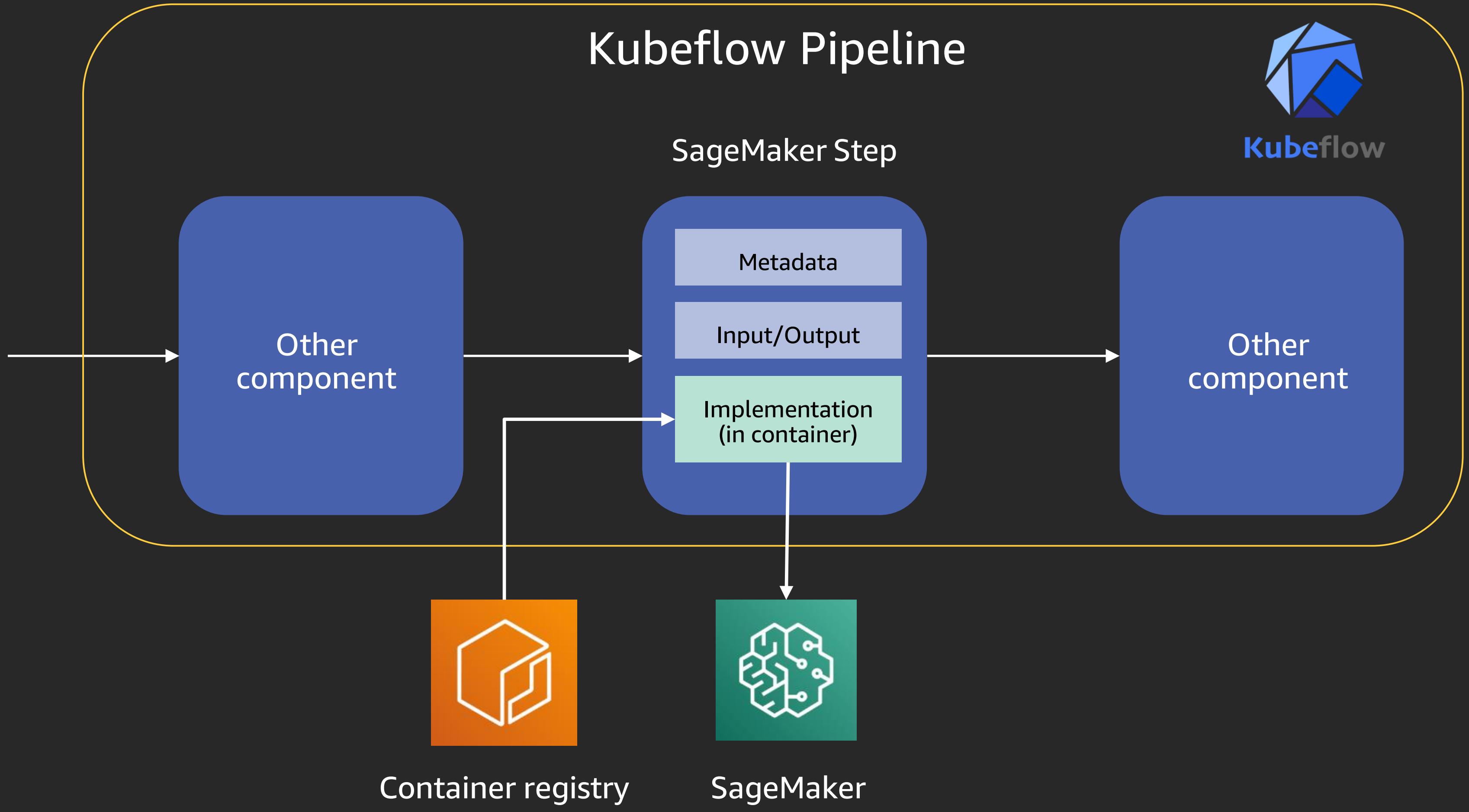
Leveraging AWS Innovations through Kubeflow

Do-It-Yourself

+

Managed Service

Running SageMaker – pipeline component



Supported components

- Training
- Model generation
- Hyperparameter tuning
- Model deployment
- Batch transform

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

Related breakouts

[AIM326-R1] Implement ML workflows with Kubernetes and Amazon SageMaker

12/4/19 (Wednesday) 4:00 PM - Bellagio, Monet 1

What’s next?

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
  - …
- **Beta**
  - Katib for hyper-parameter tuning
  - Fairing SDK to facilitate use of notebooks for build-train-deploy
  - 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

Call to Action

re:Invent workshop

OPN401-R1 - [REPEAT 1] Machine learning with Kubeflow on AWS

12/5/19 (Thursday) 3:15 PM - MGM, Level 1, Grand Ballroom 120

Online workshop

https://eksworkshop.com/kubeflow/

Community

Join the kubeflow#aws Slack channel:

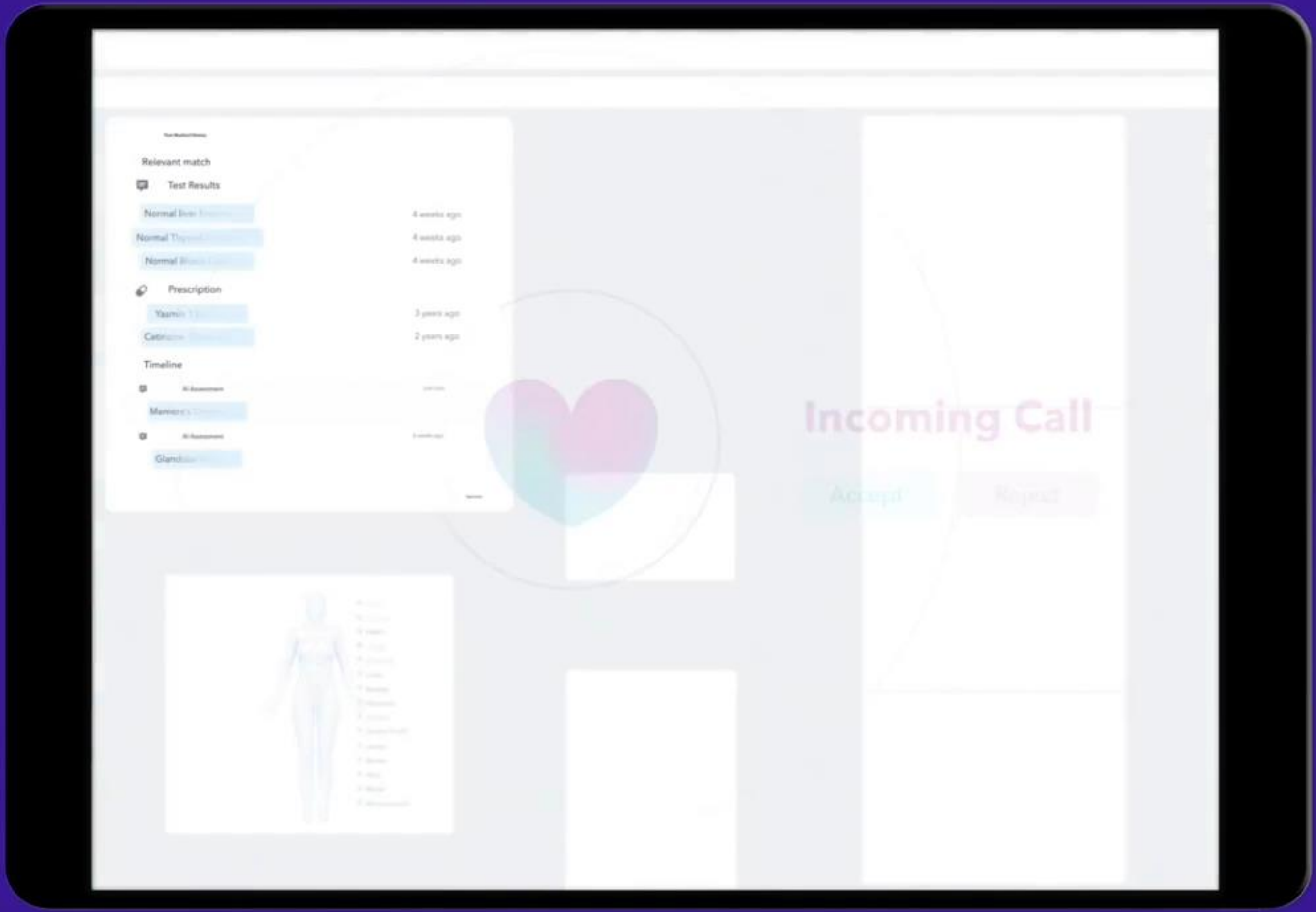
Babylon

Babylon

The mission

babylon

We believe it is possible to put an **accessible** and **affordable** health service in the hands of every person on Earth


Relevant match

Test Results

Normal liver [illegible]	4 weeks ago
Normal Thyroid [illegible]	4 weeks ago
Normal Blood [illegible]	4 weeks ago

Prescription

Yasmin [illegible]	3 years ago
Catrin[illegible]	2 years ago

Timeline

Mamm[illegible]

Gland[illegible]

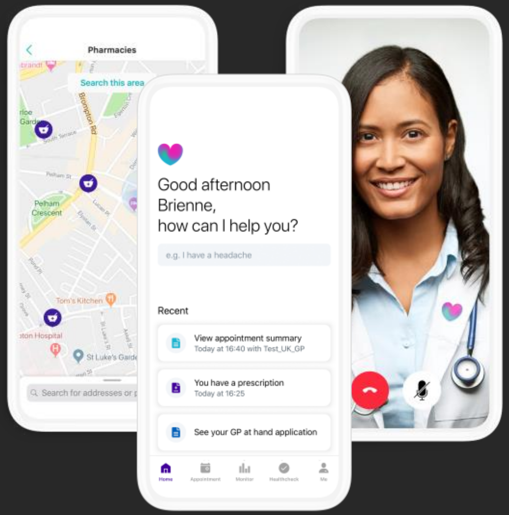
Babylon

Challenges

AWS re:Invent

aws

Our challenges



Global products

Global company

Global platform

Our challenges

300+	500+	15+	5
Microservices	Deployments per day on average	EKS clusters	AWS Regions

Our challenges

- Providing a safe and secure environment for our research teams
- Data locality
- Training and managing AI models at a global scale
- Improving overall engineering efficiency

Babylon

Our solution

AI platform: The big picture

Providing our teams with a single interface to experiment, train, tune, validate, and track their AI models

AI platform

Secure

Treat it as a production platform

Scalable

Pay for what you use, and scale as much as you need

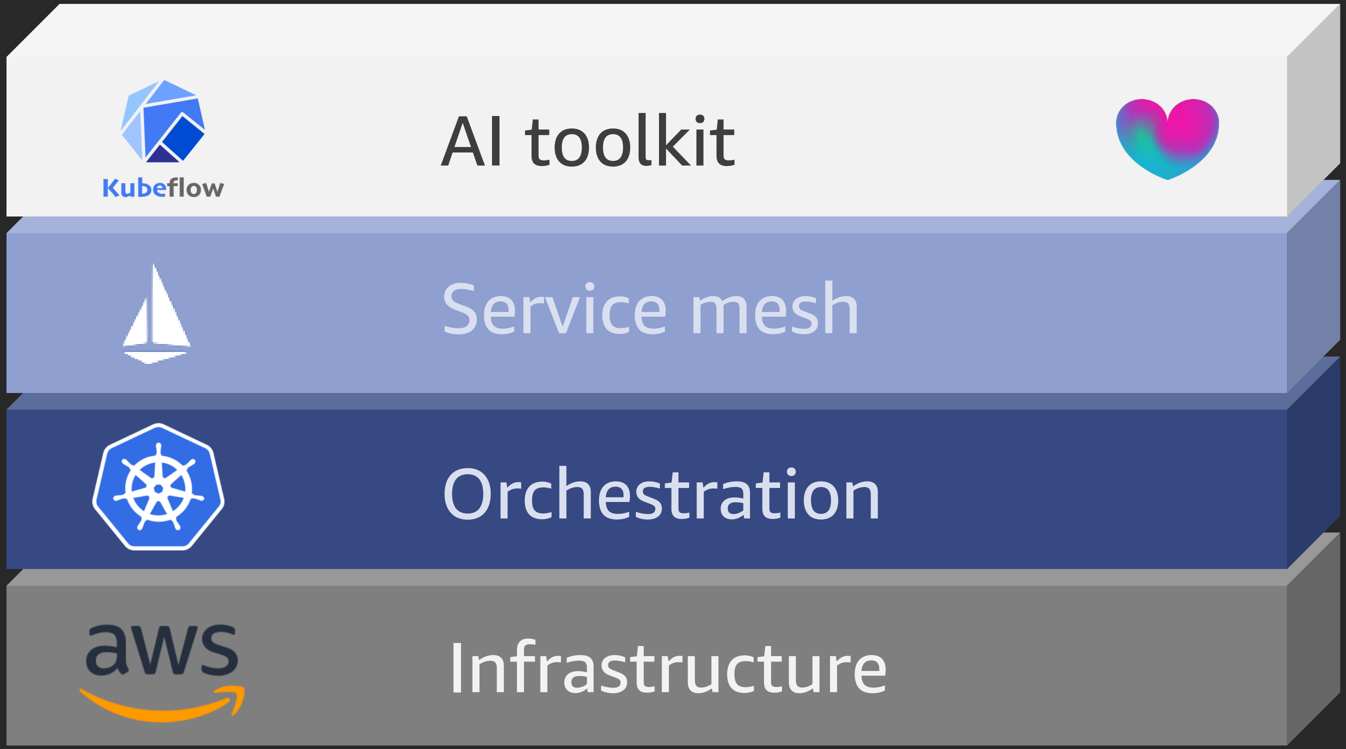
Flexible

It must be able to run any tool or service our users need

Global

We need to deploy it anywhere we operate

AI platform


Kubeflow
AI toolkit
Service mesh
Orchestration
aws
Infrastructure

AI platform

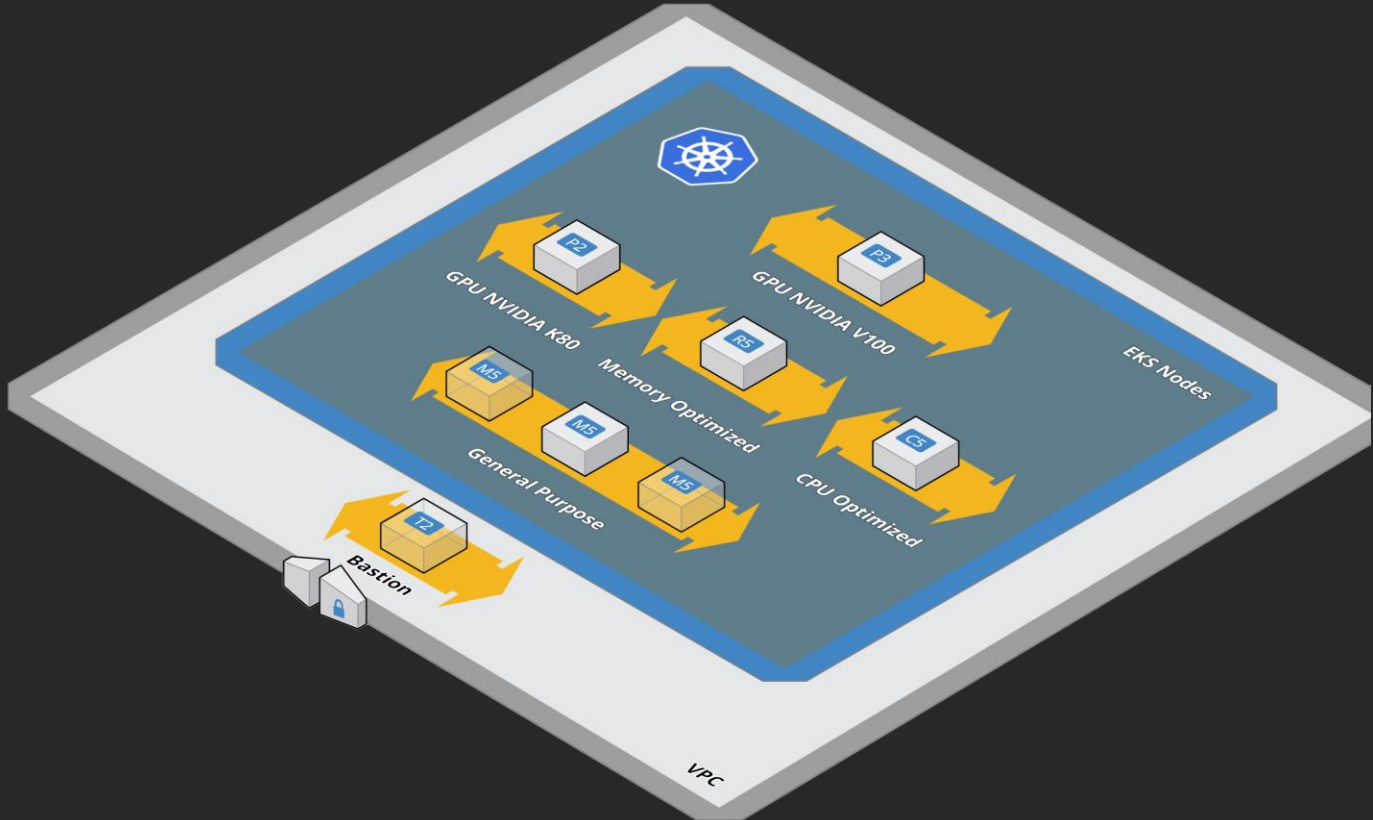
- EKS infrastructure
- GPUs
- Networking

- Kubeflow deployment
- Project structure
- Extensibility

- Global platform
- Deployments

AI platform: *EKS infrastructure*

- Make everything private
  - EKS API
  - EKS nodes
- Prepare for different workloads
  - High CPU, high memory, GPUs
  - Spot instances for better pricing
- Encrypt by default
  - EBS volumes
  - AMIs (root volumes)



AI platform: *GPUs on EKS*

- NVidia device plugin *DaemonSet*
- Register nodes with
  - Taint: `nvidia.com/gpu=true`
  - Label: GPU type and brand

    nvidia-tesla-k80

```
# Add a toleration, allowing the pod to run on GPU nodes
tolerations:
  - key: "nvidia.com/gpu"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"

# Use nodeSelector to choose the GPU type
nodeSelector:
  accelerator: nvidia-tesla-k80 # or nvidia-tesla-v100
```

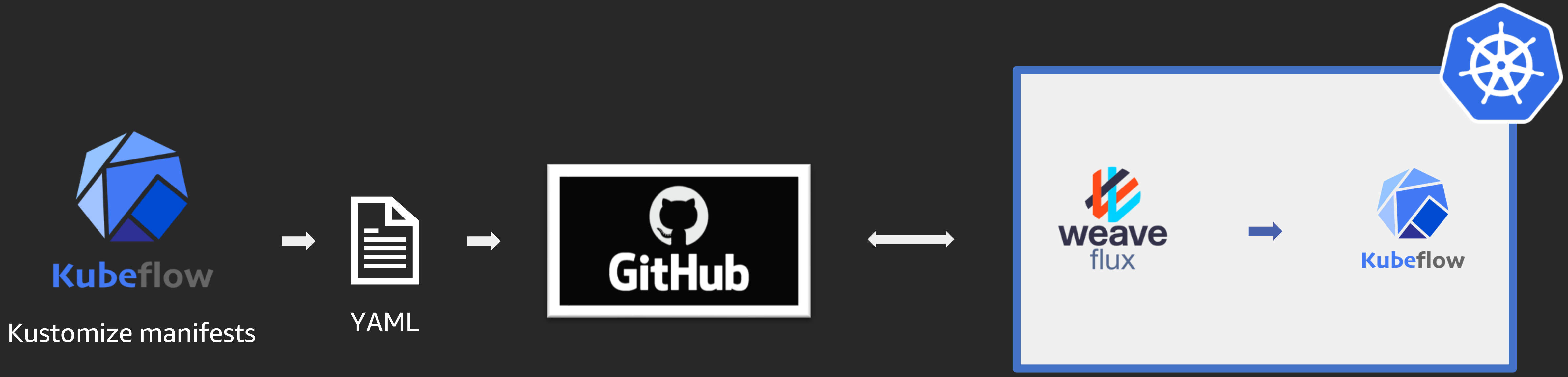
AI platform: *Networking*

- Virtual Private Cloud (VPC)
  - Span over at least 3 Availability Zones (AZ) … if you can!
  - 1 *Auto Scaling Group* per AZ per instance type
  - Use NAT gateways
- Kubernetes
  - Zero-trust policy
  - Mutual TLS
  - Service role-based access control
  - Service mesh (Istio)

AI platform: *Kubernetes and Kubeflow*

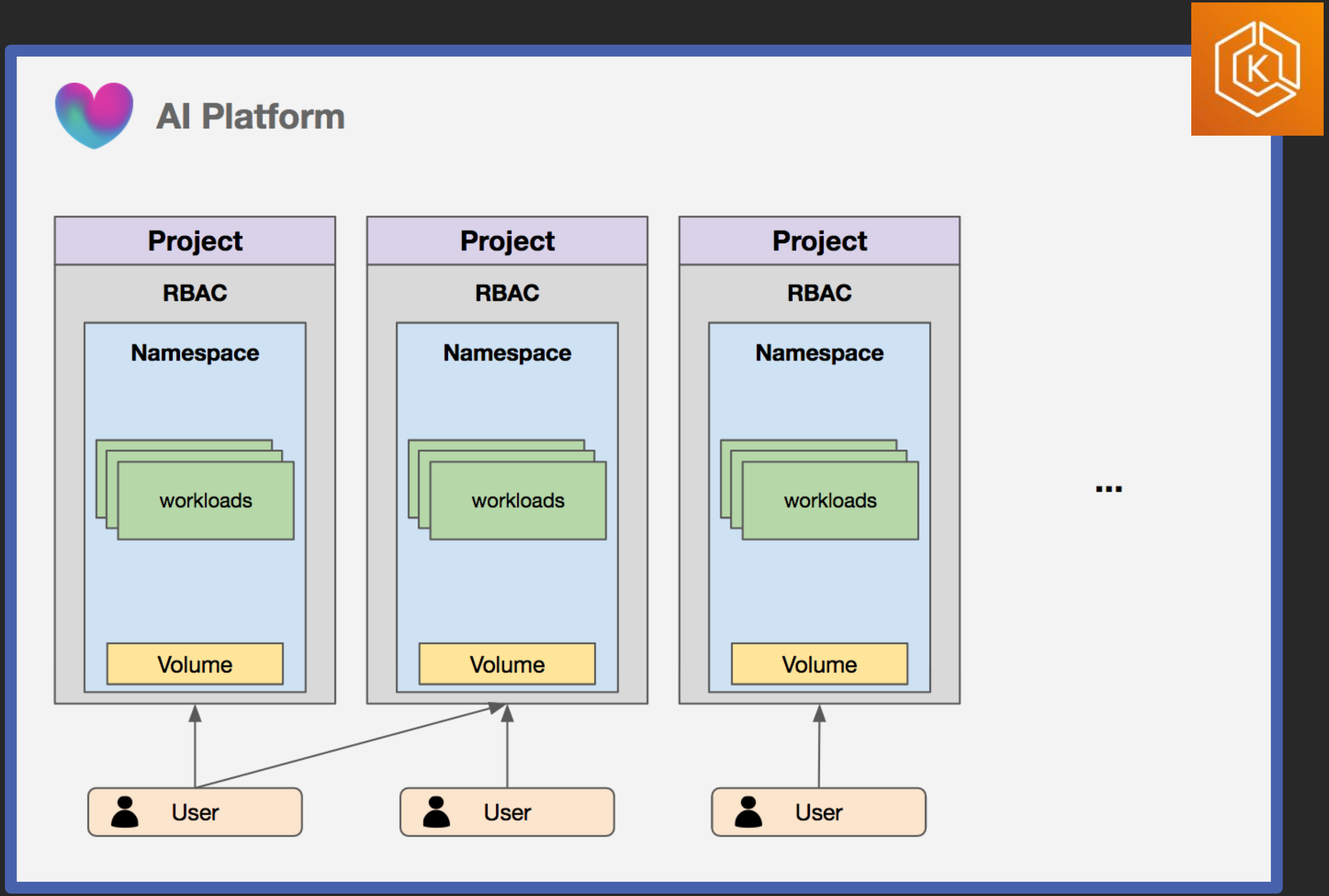
- Kubernetes
  - Perfect for scheduling and running any workloads
  - Massive scale
- Kubeflow
  - ML toolkit for Kubernetes
  - Modular: use and deploy only what you need
  - Multi-user: plug in your enterprise OIDC
  - Open-source: deploy it anywhere you have Kubernetes
  - We mostly use Jupyter Notebooks, TensorFlow jobs, hyperparameter tuning

AI platform: Deploying *Kubeflow*



AI platform: *Project isolation and collaboration*

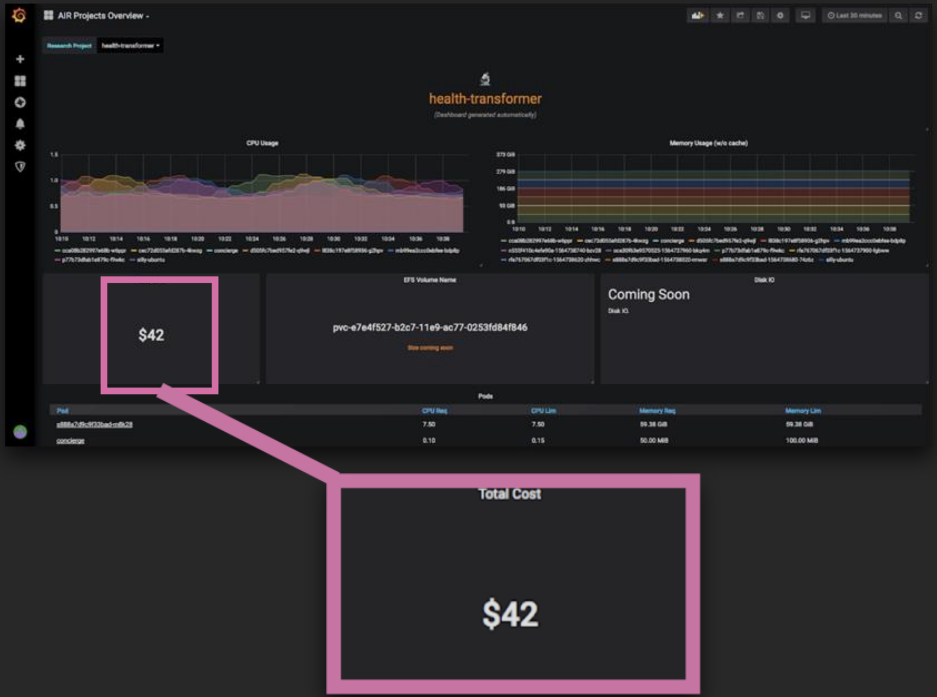
- Project *CRD*
  - Isolated namespace
  - RBAC rules for user management
  - Encrypted **EFS** partition
  - Quotas management
- Need faster storage?
  - EBS volumes
  - FSx for Lustre



AI Platform: *Workload Monitoring*

- Monitor Projects *(CPU, RAM, etc)*
- Monitor Cluster State *(MLOps)*
- Collect metrics from jobs
- Automated Dashboards via ConfigMaps

Prometheus Grafana



health-transformer

(Dashboard generated automatically)

$42

pvc-e7e4f527-b2c7-11e9-ac77-0253fd84f846

Size coming soon

Pods

Pod	CPU Req	CPU Lim	Memory Req	Memory Lim
s888a7d9c9f33bad-m8k28	7.50	7.50	59.38 GiB	59.38 GiB
concierge	0.10	0.15	50.00 MiB	100.00 MiB

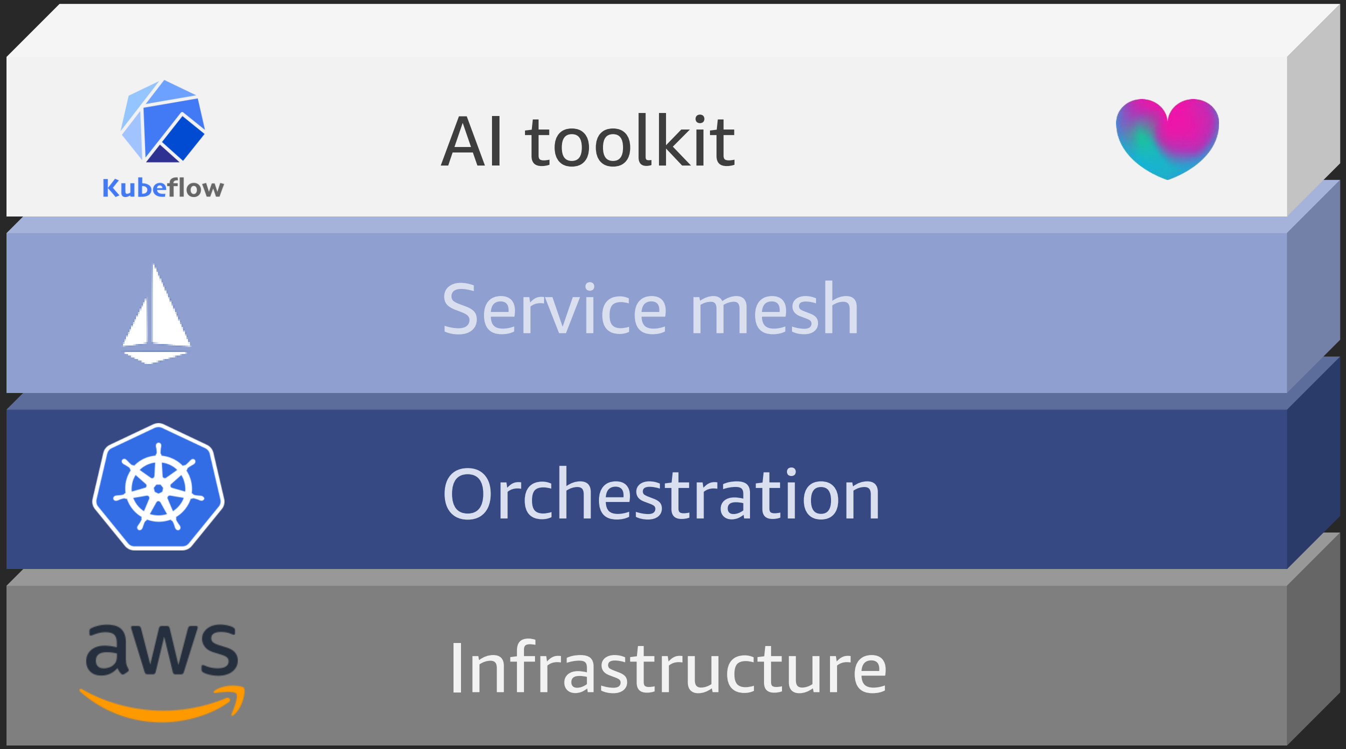
… And monitor cost

AI platform: *Extensibility of services*

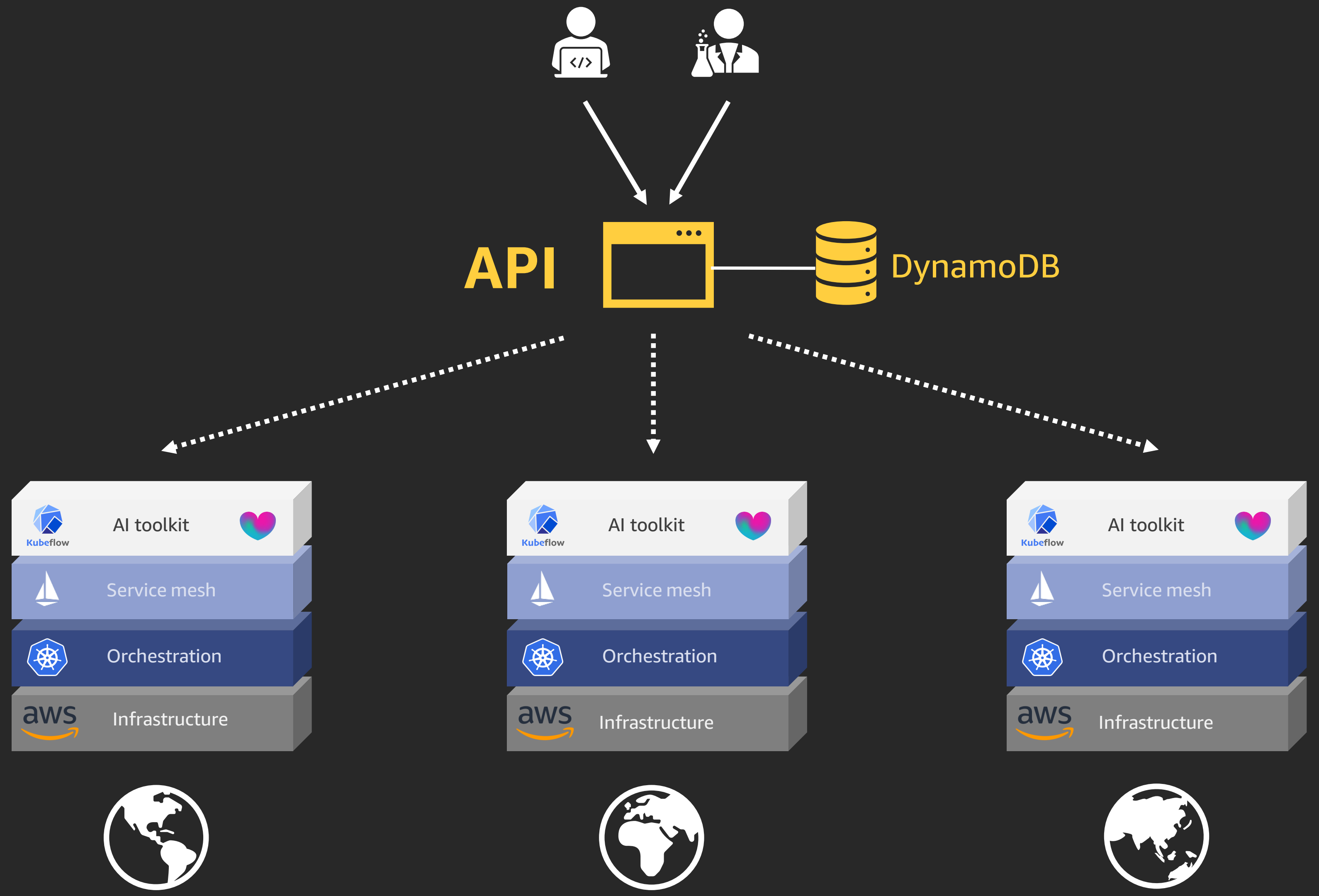
- Modularity of Kubernetes and Kubeflow
  - Gives us ability to add new tools ***fast***
  - Ex: deploying ***Argo*** for workflow management
- Use case:
  - Migrating our model Clinical Validation pipeline to Argo on EKS

Code optimization + Parallelization of compute → From 10 hours to **20 minutes**

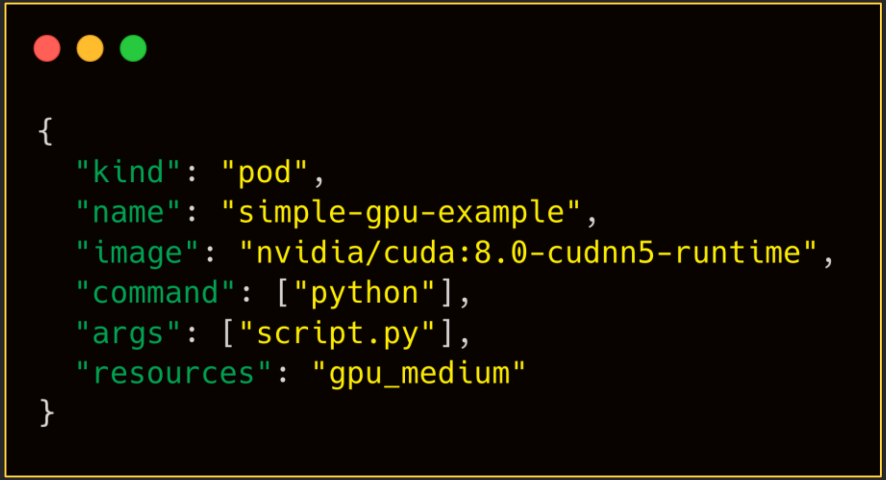
AI platform: *Single* cluster



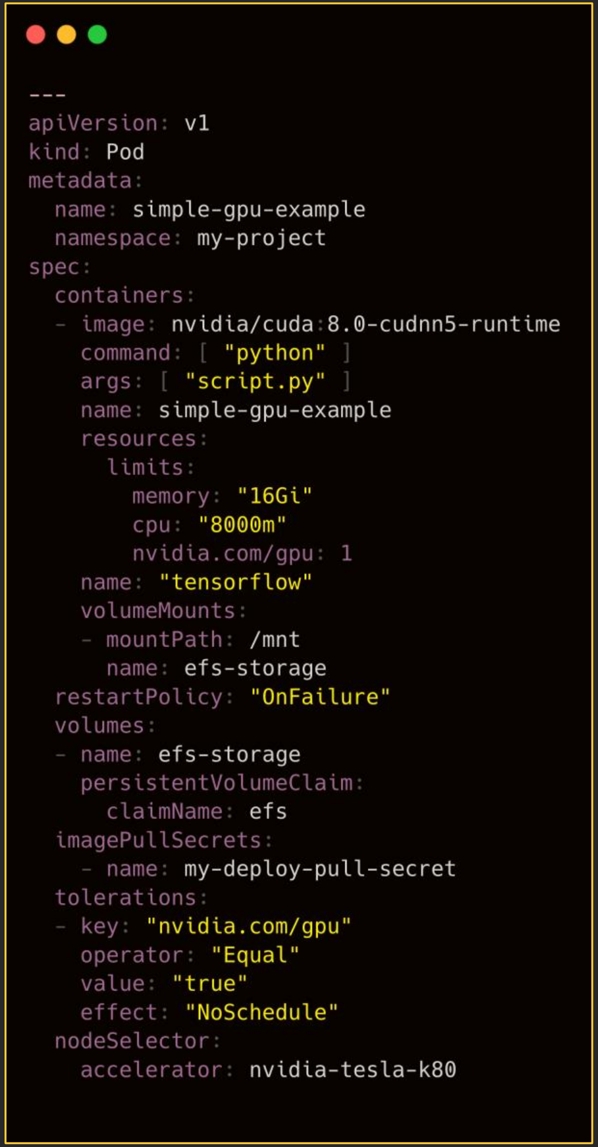
AI platform: *Global footprint*



API



```json
{
  "kind": "pod",
  "name": "simple-gpu-example",
  "image": "nvidia/cuda:8.0-cudnn5-runtime",
  "command": ["python"],
  "args": ["script.py"],
  "resources": "gpu_medium"
}
```



```yaml
---
apiVersion: v1
kind: Pod
metadata:
  name: simple-gpu-example
  namespace: my-project
spec:
  containers:
  - image: nvidia/cuda:8.0-cudnn5-runtime
    command: [ "python" ]
    args: [ "script.py" ]
    name: simple-gpu-example
    resources:
      limits:
        memory: "16Gi"
        cpu: "8000m"
        nvidia.com/gpu: 1
    name: "tensorflow"
    volumeMounts:
    - mountPath: /mnt
      name: efs-storage
  restartPolicy: "OnFailure"
  volumes:
  - name: efs-storage
    persistentVolumeClaim:
      claimName: efs
  imagePullSecrets:
    - name: my-deploy-pull-secret
  tolerations:
  - key: "nvidia.com/gpu"
    operator: "Equal"
    value: "true"
    effect: "NoSchedule"
  nodeSelector:
    accelerator: nvidia-tesla-k80
```

Babylon platform: *Global deployments*

- Deploying models and services in multiple regions
- Secure SDLC-aware deployment tool: *Shipcat* (in Rust)
- Per microservice tracking of:
  - Compliance and regulations
  - Engineering documents
  - Data management

Babylon

Next steps

AWS re:Invent

aws

Next steps

- Providing a common model serving framework at Babylon
- Better metadata tracking for our models
- Integration with Kubeflow pipelines
- Improving Docker user experience for researchers

Thank you!

Yaniv Donenfeld
donenfel@amazon.com

Jean-Marie Ferdegue
jeanmarie.ferdegue@babylon
health.com

Jeremie Vallee
jeremie.vallee@babylon
health.com

Please complete the session survey in the mobile app.