
AWS
re:Invent

AIM315

Accelerating autonomous vehicle solutions with SageMaker, featuring Lyft

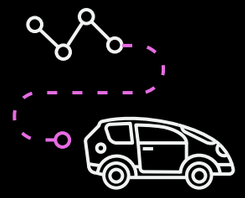
Alex Bain
Machine Learning Engineer
Lyft Level 5

Nitin Wagh
Principal ML Specialist
AWS

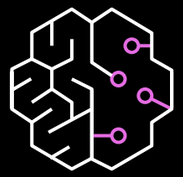
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda



Challenges in AV development

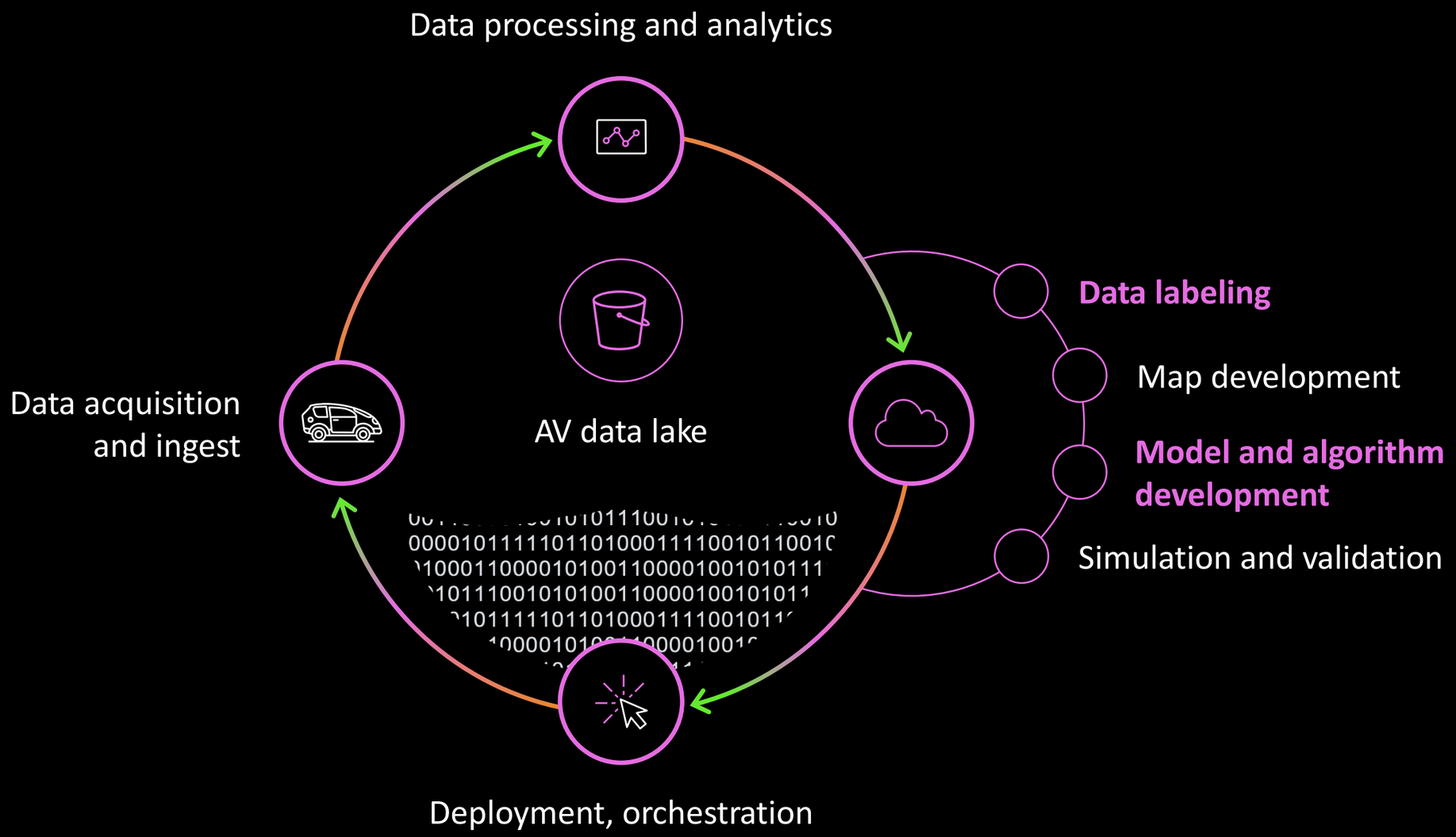


Machine learning for AV development

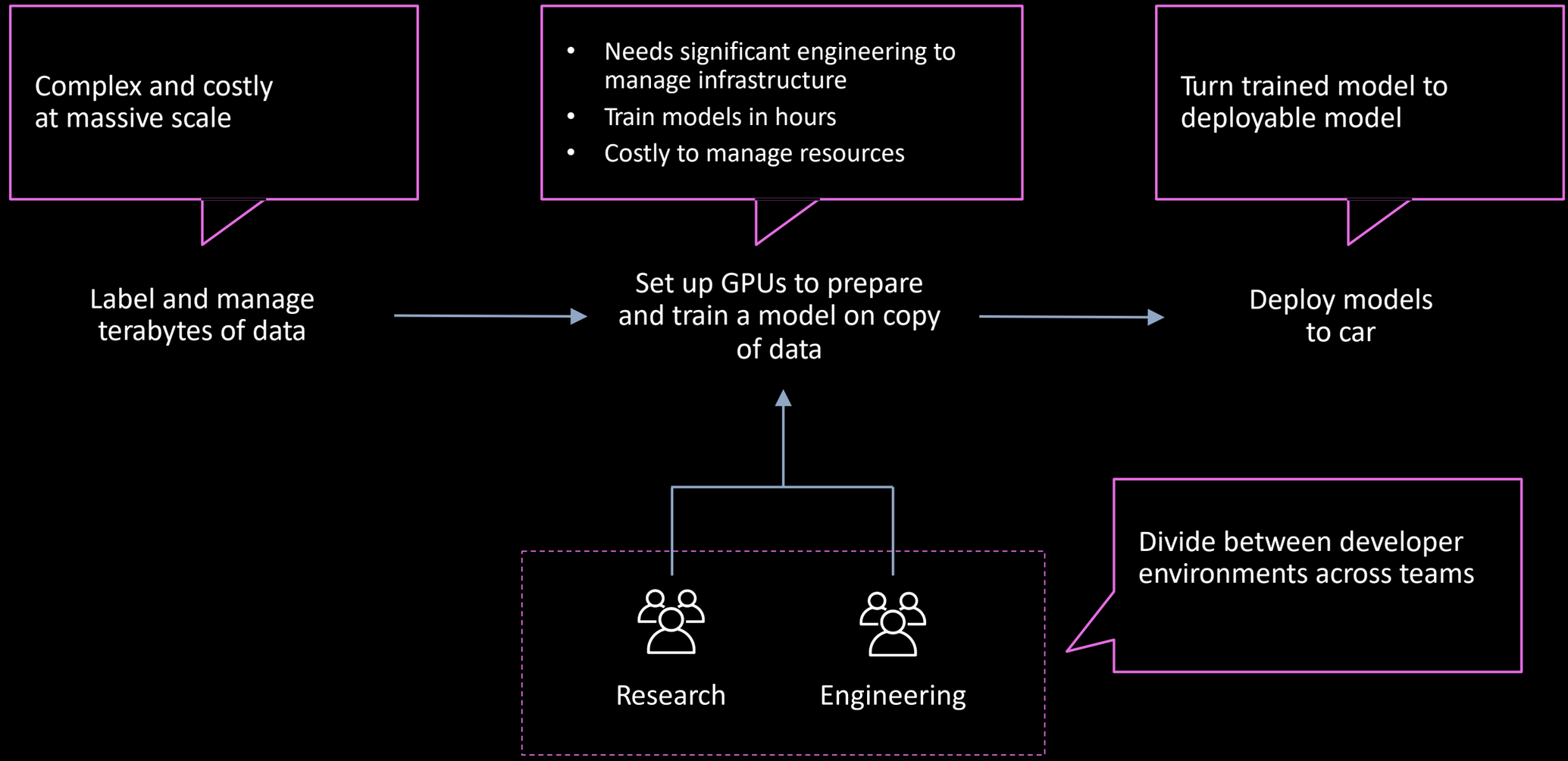
Lyft Level 5 story

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

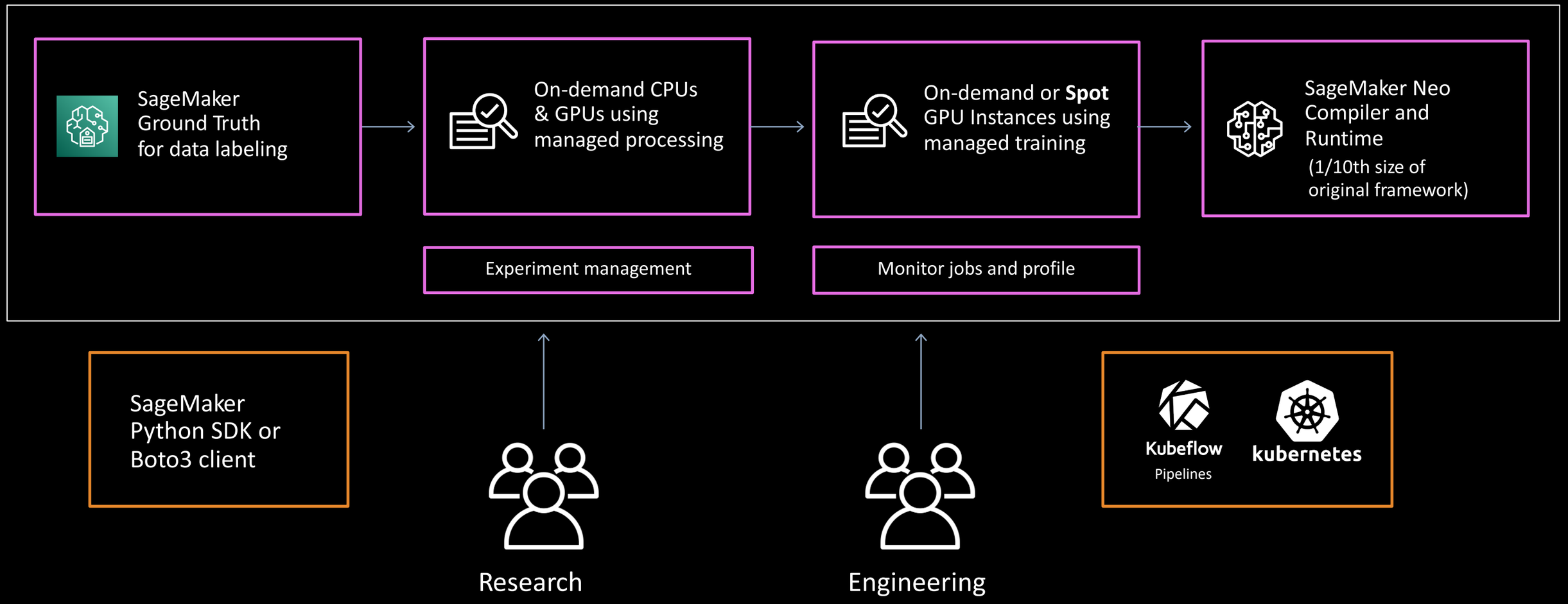
Autonomous development lifecycle on AWS



Challenges in ML development for AV



Using Amazon SageMaker to manage ML development for AV



SageMaker Ground Truth to easily label training data at scale

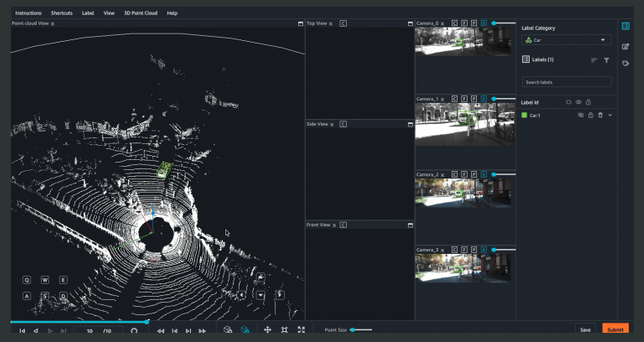
Fully managed and secure

30+ labeling workflows

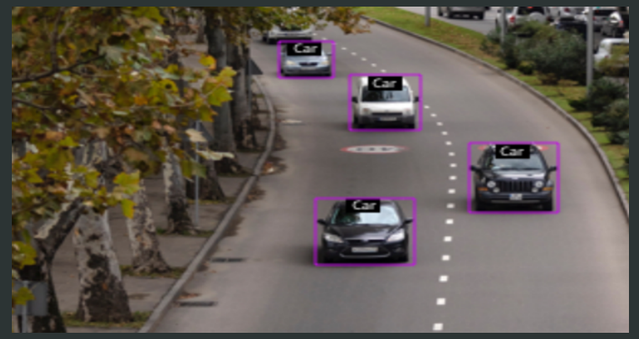
Assistive tooling, auto-labeling, consensus

Your choice of workforce

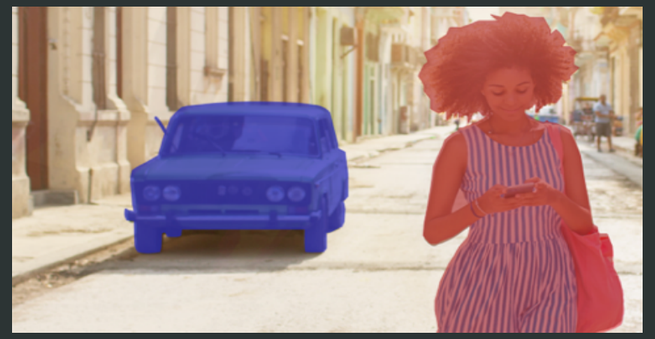
Pay as you go, volume discounts



3D Point Cloud

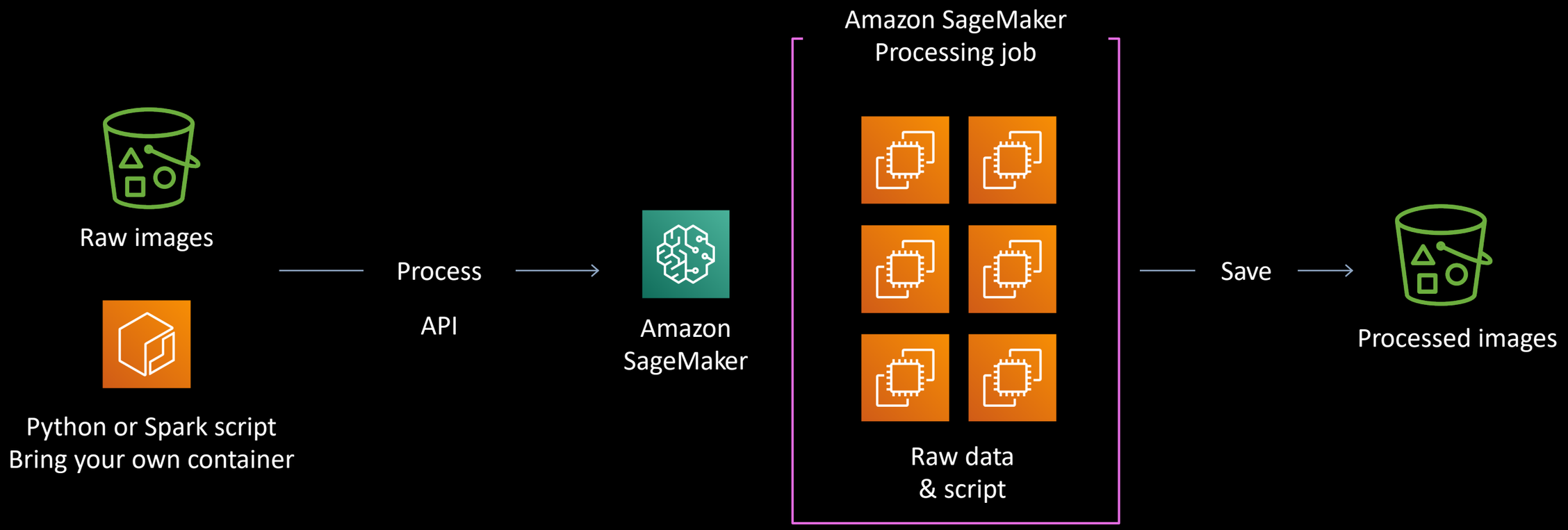


Video



Image

Pre-process large datasets using SageMaker Processing



SageMaker Processing one-click API helps prepare massive datasets

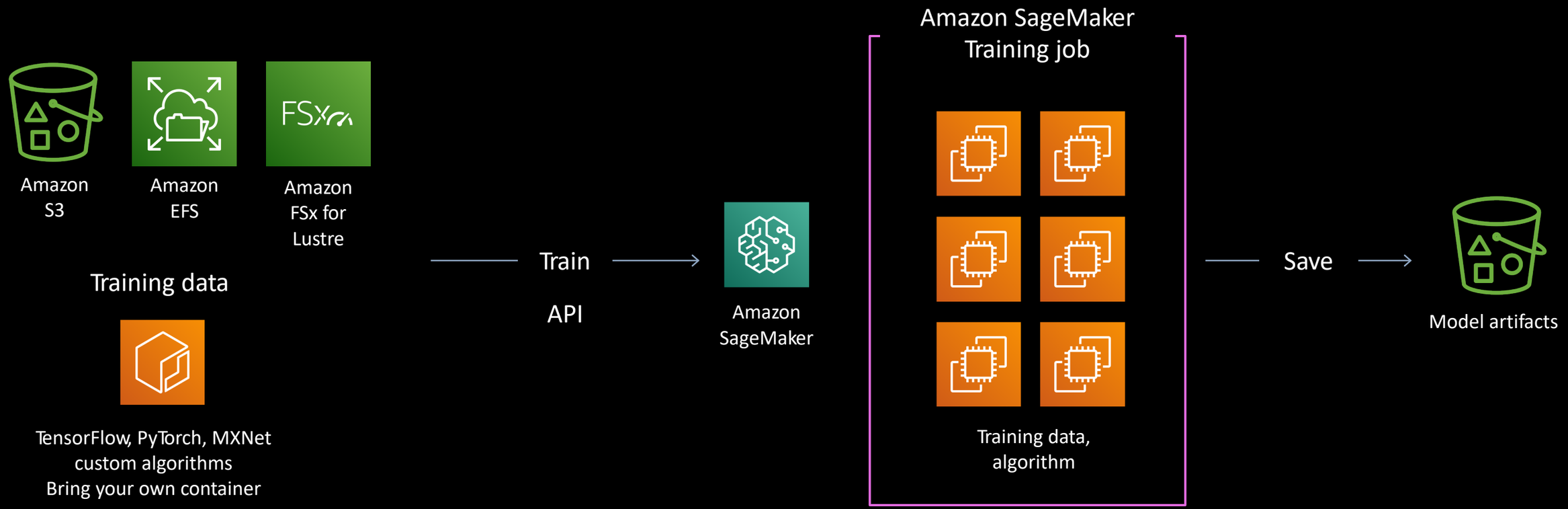
```
from sagemaker.processing import Processor,
                ProcessingInput, ProcessingOutput

processor = Processor(image_uri='<your_ecr_image_uri>',
                                 instance_count=10,
                instance_type="ml.p3.2xlarge",
                                 ...)

processor.run(inputs=<input path>,
          outputs=<output path>))
```

- Distributed Python or Spark
- Bring your own container or script
- Data sharding
- On-demand instances
- Amazon S3 for storage

Fully managed training using on-demand GPUs



Using Amazon SageMaker for training ML models with one-click API

```
rules = [ Rule.sagemaker(rule_configs.vanishing_gradient()) ]

estimator = PyTorch(entry_point=detectron_v2_script,
                    base_job_name='aerial_image_detection',
                    role=role,
                    train_instance_count=10,
                    metric_definitions= metric_definitions,
                    train_instance_type='ml.p3.16xlarge',
                    hyperparameters=hyperparameters,
                    train_use_spot_instances=True,
                    framework_version='1.6.0',
                    py_version='py3',
                    rules = rules,
                    … )
estimator.fit('training':f"s3://{bucket}/train-coco"},
                    experiment_config={
                "TrialName": d2_trial.trial_name,
                "TrialComponentDisplayName": "Training"})
```

Monitor the job

Your training script

Distributed training

Choice of GPU

Use Spot Instance (up to70% savings)

S3 as data source

Track experiments

Amazon SageMaker + Lyft Level 5

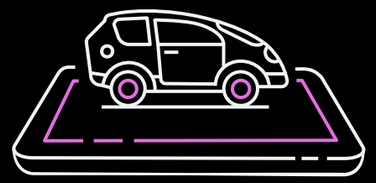
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Story of Lyft Level 5



About Lyft Level 5



Level 5 is the autonomous vehicle division at Lyft

Formed in 2018 to build an AV ridesharing service

Now testing our AVs in San Francisco (but not open to the public)

AV development at Lyft Level 5

ML powers the most important elements of our AVs

Perception, planning, and prediction on the AV all run with ML models

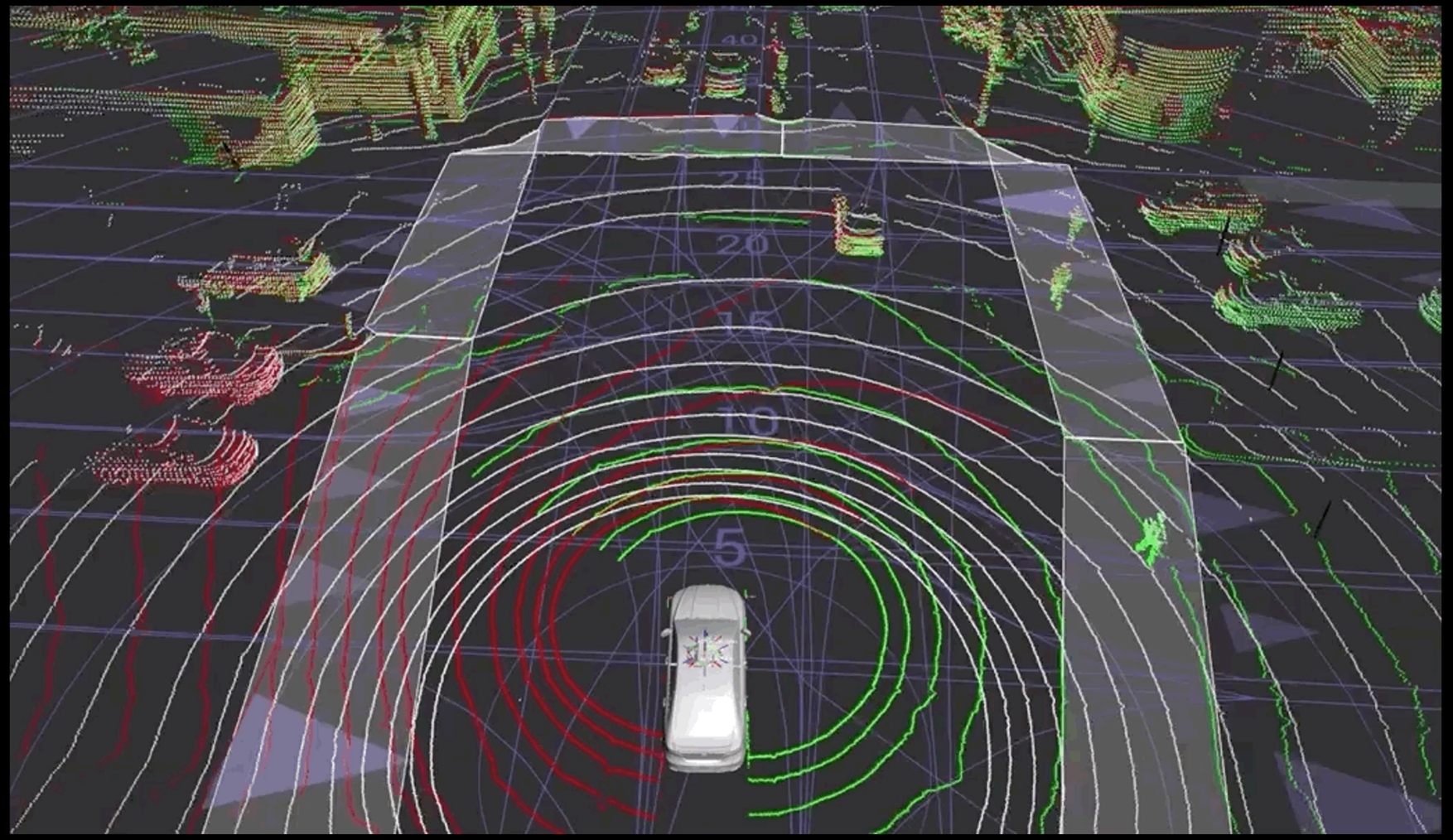




Lyft Level 5 ML in action

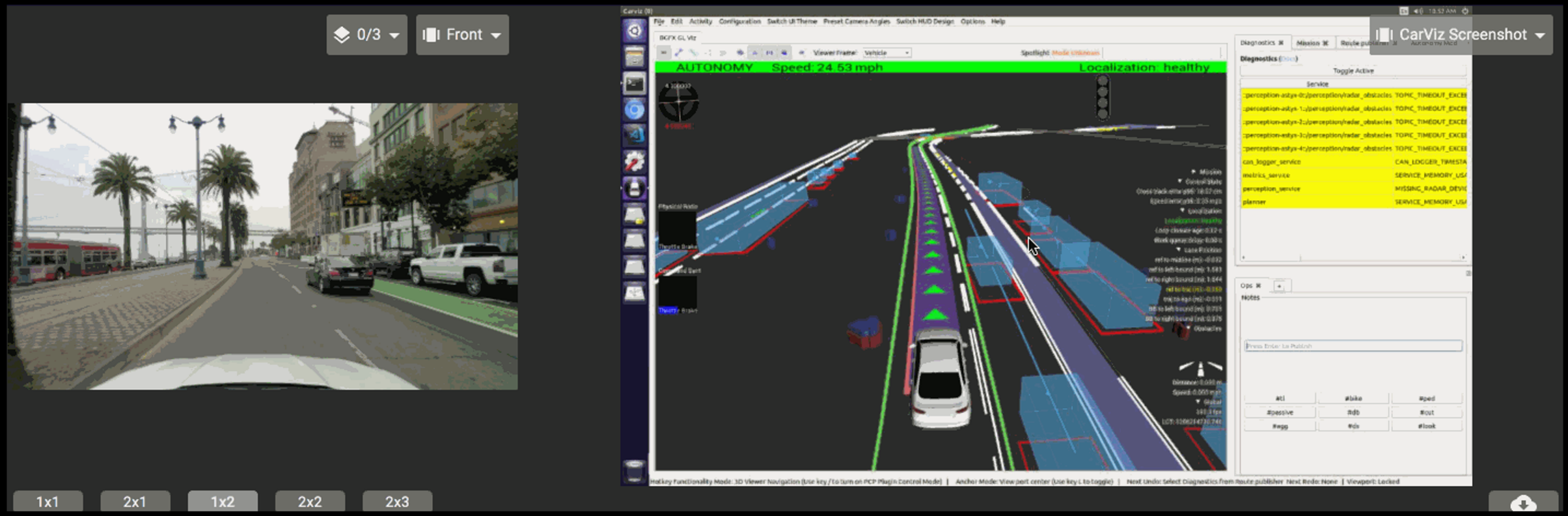
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3D LiDAR sensor visualization



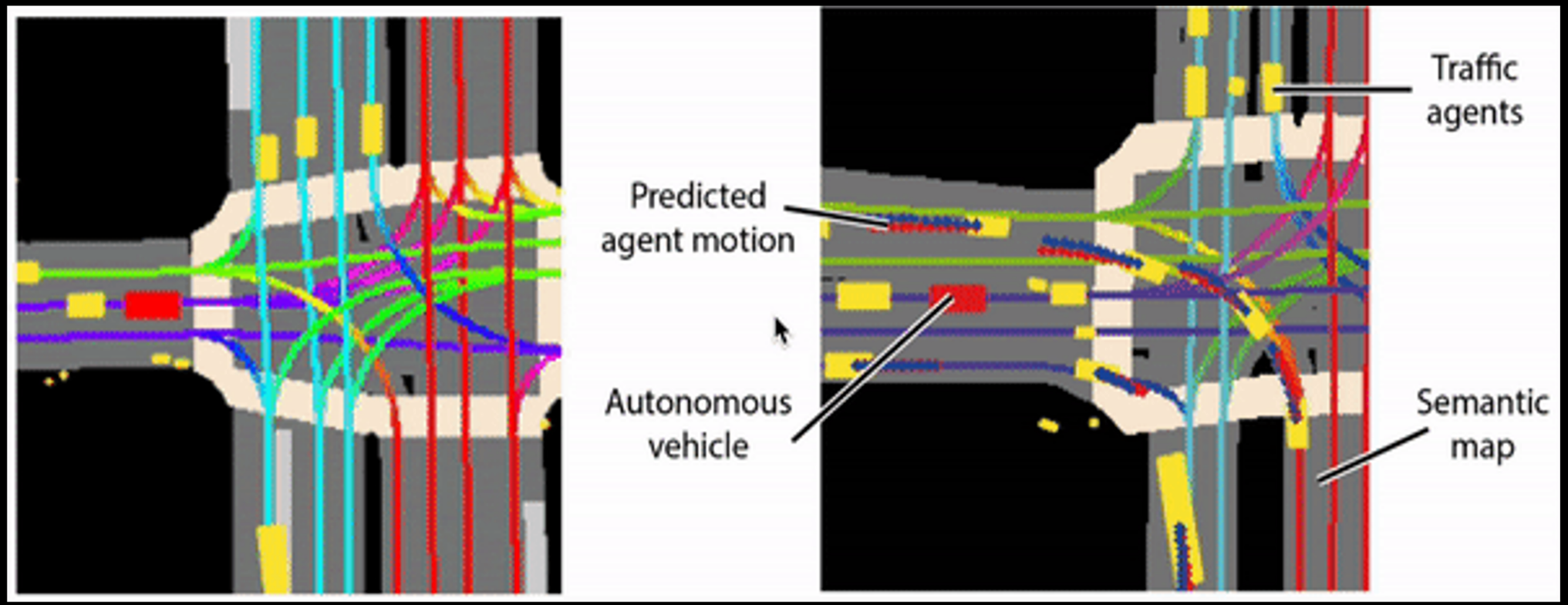
Perception in the world

Left: Our AV in operation in SF | **Right:** AV visualization



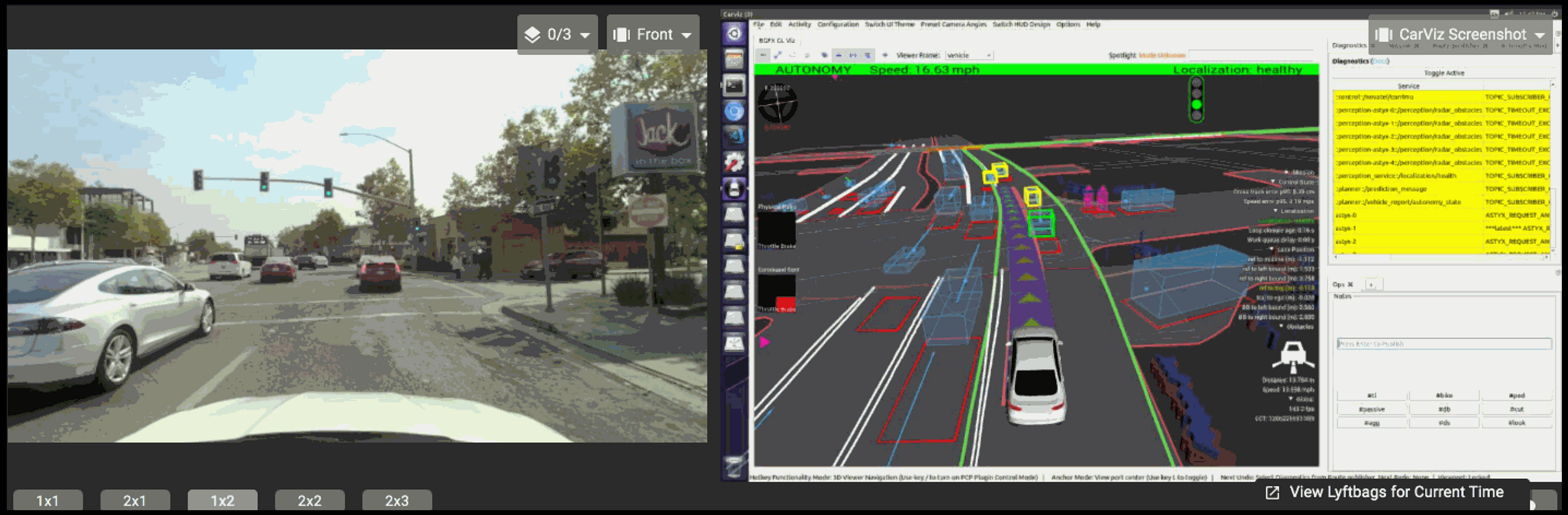
Predicting motion of agents in the world

Red = our car | **Yellow** = predicted paths of other agents overlaid on lanes



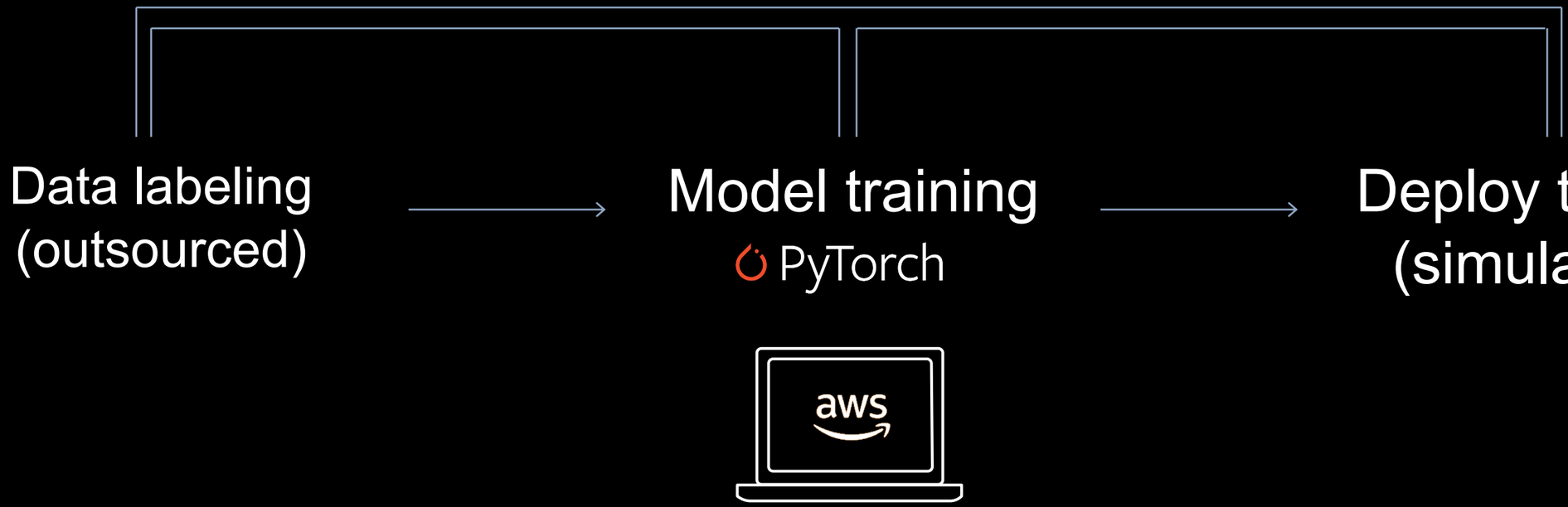
Planning for our AVs

Left: Deciding to nudge around a stopped car | **Right:** AV visualization

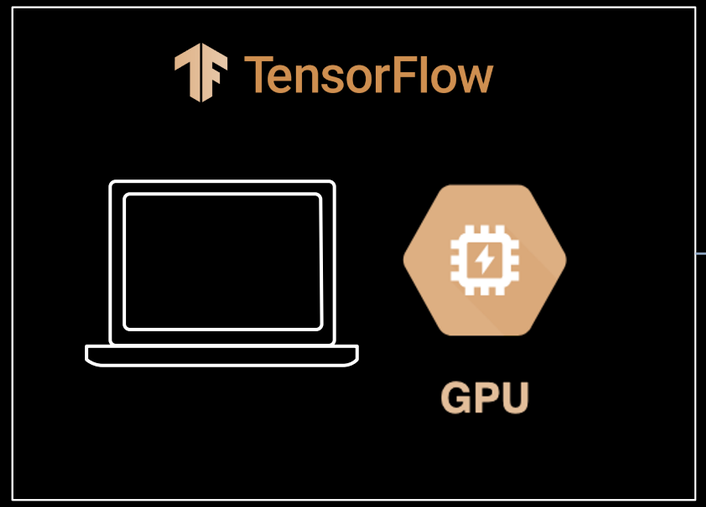


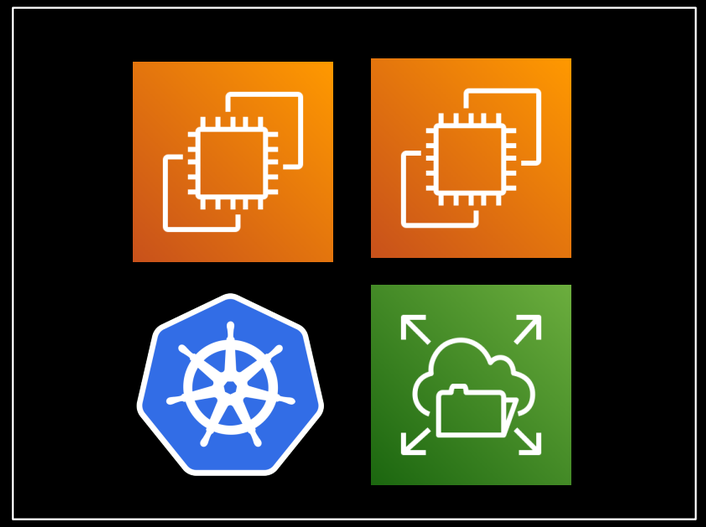
ML development workflow

A day in the life of our ML team



Our ML journey





Constraint by Compute and Storage

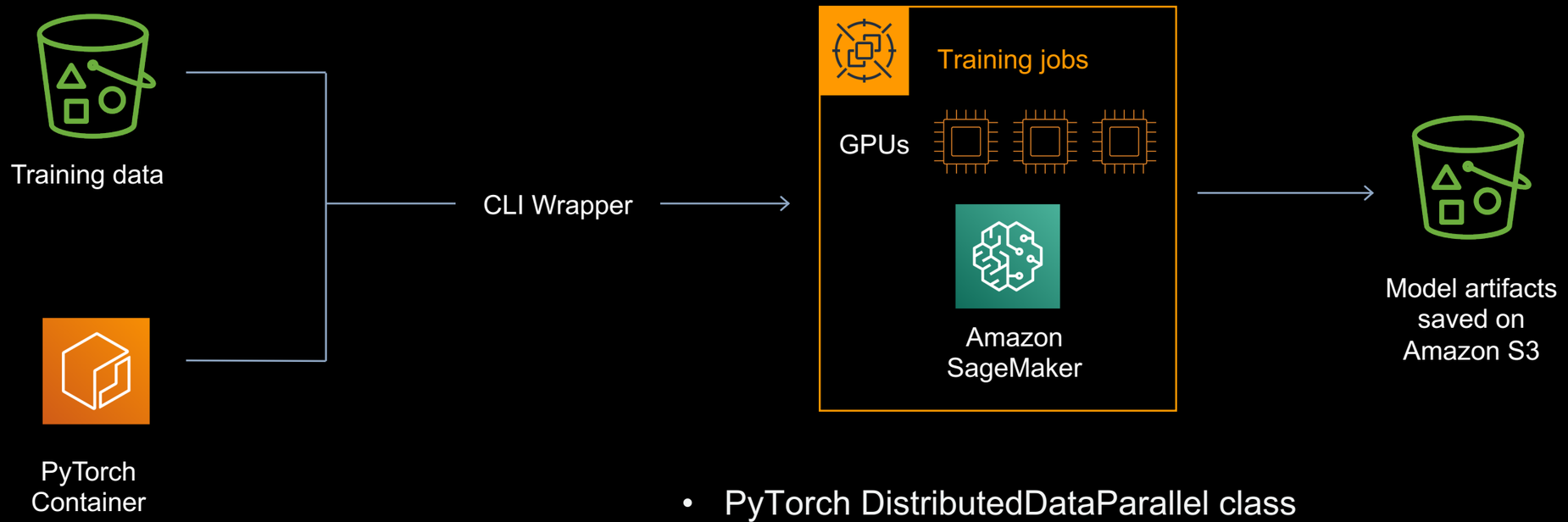
Expensive to operate and support

Data loading became hard as data size grew

Training took days (8 GPUs)

TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc.

Our journey with Amazon SageMaker



- PyTorch DistributedDataParallel class
- Distributed training for up to 16 instances
- P3.16xlarge and P3dn.24xlarge
- Managed Spot training

What happened



We massively accelerated training time and reduced overhead

- About one infrastructure engineer to support >100 ML engineers using SageMaker
- Improved data efficiency + distributed training → 10x improvement in training times
- Lower support costs + only get billed for training time → much lower costs overall

Modern distributed training for deep learning

The latest hardcore knowledge for the best deep distributed training

Productivity / velocity of PyTorch community → TensorFlow right now

Ignore Horovod, just use PyTorch DistributedDataParallel

- All Horovod all-reduce innovations incorporated into PyTorch (and TensorFlow 2.x)
- Data partitioning / sampling also built into PyTorch
- Disregard OpenMPI to start, just use NCCL as communications layer for <16 node training

Can start with Amazon SageMaker PyTorch SDK and combine with PyTorch Lightning framework for even more productivity

Lessons

- Importance of prioritizing the efficiency of our users.
  10x improvement in training time → huge improvement in ML in 2020 compared to 2019

- Huge cost win from the fact that SageMaker is a managed service
  - Kubernetes cluster downtime → now gone
  - Much lower support overhead since “it just works” (mostly)
  - Only billed for training time

- Unexpected wins:
  - Access to latest NVIDIA GPU instance types
  - We run on Amazon SageMaker Spot Instances to reduce costs further

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.