Achieve high performance and cost-effective model deployment

Emily Webber
Sr Machine Learning Specialist SA
Amazon Web Services

Today’s agenda

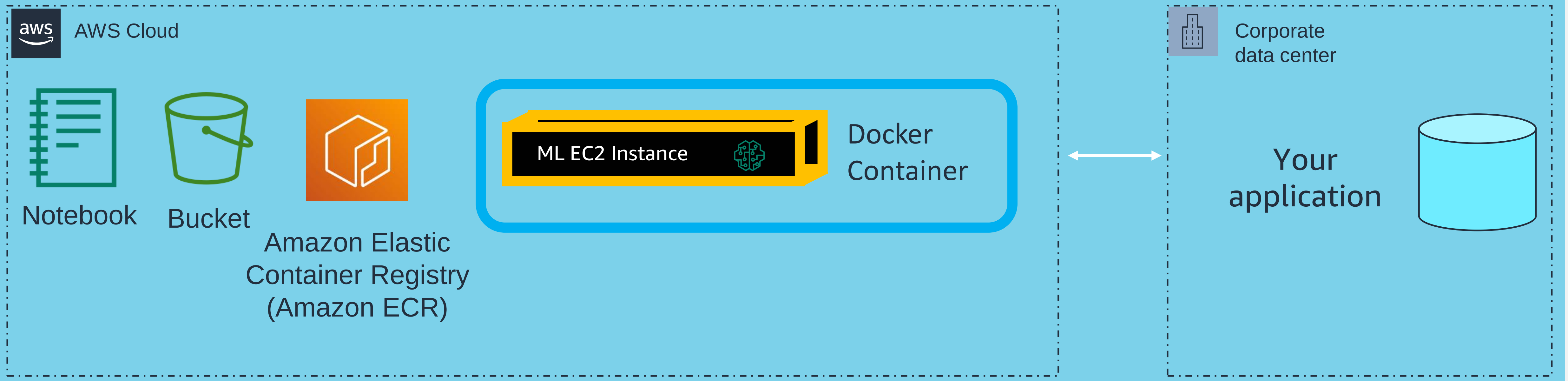
- Deploy ML models on Amazon SageMaker
- Pick the right deployment solution
- Perfect your deployment
- Deploy thousands of models with SageMaker multi-model endpoints
- Demo

So you want to host a model on the cloud

Model

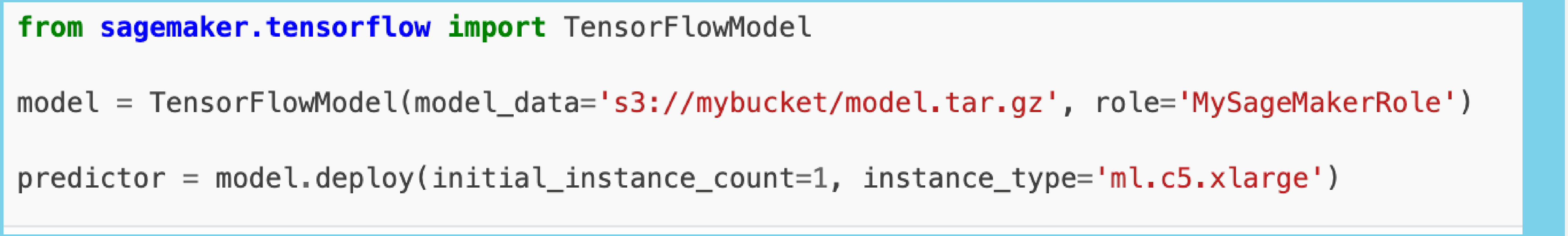
- model.tar.gz
- inference.py

```
model = Model(ecr_image_uri, s3_model_data)
predictor = model.deploy(1, ‘ml.g5.xlarge’)
result = predictor.predict(payload)
```



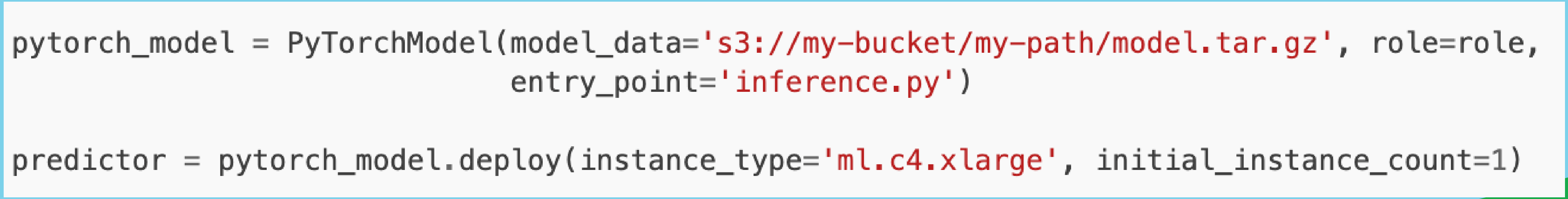
Amazon SageMaker deploys your model and inference code onto real-time endpoints with low latency

Bring your own pretrained models to host on SageMaker using script mode



```
from sagemaker.tensorflow import TensorFlowModel

model = TensorFlowModel(model_data='s3://mybucket/model.tar.gz', role='MySageMakerRole')

predictor = model.deploy(initial_instance_count=1, instance_type='ml.c5.xlarge')
```

Use our built-in inference script with deep learning containers, or bring your own script



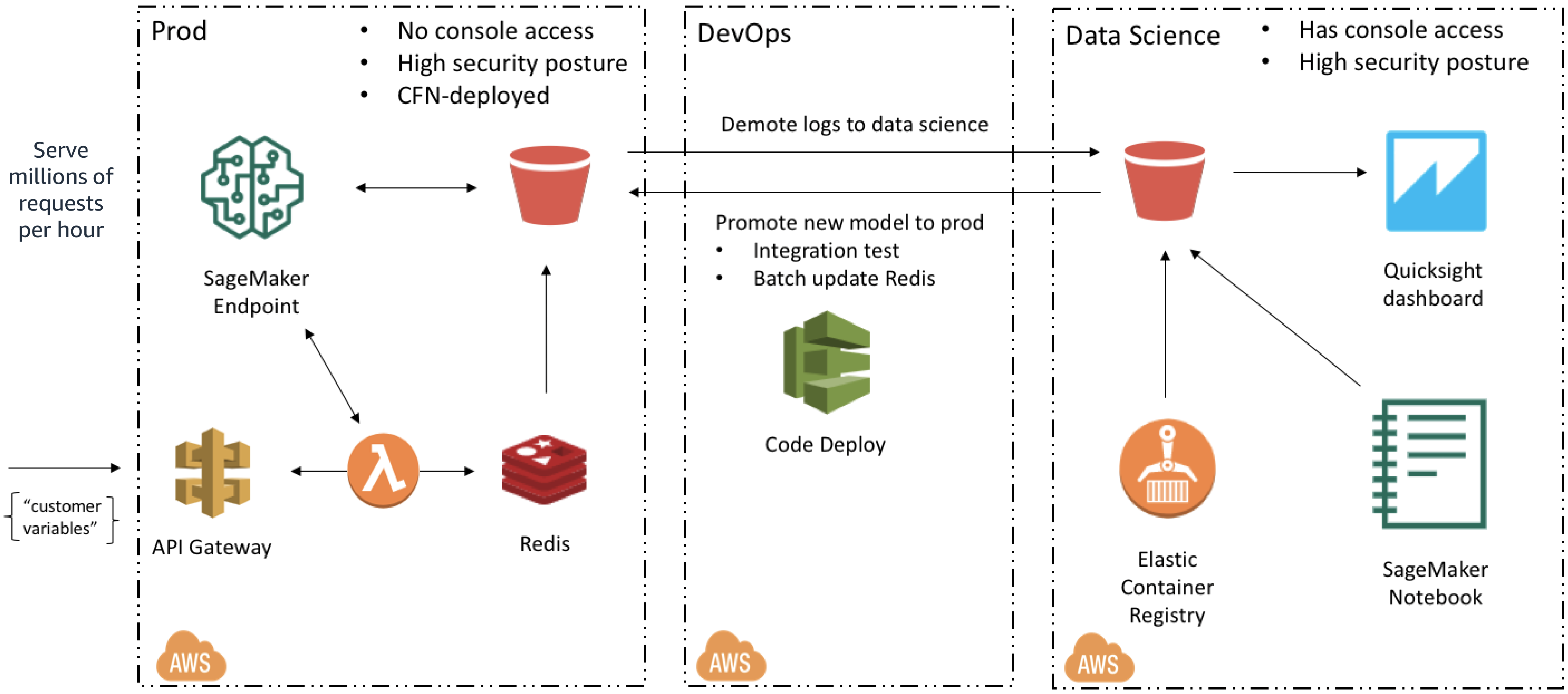
```
pytorch_model = PyTorchModel(model_data='s3://my-bucket/my-path/model.tar.gz', role=role,
                             entry_point='inference.py')

predictor = pytorch_model.deploy(instance_type='ml.c4.xlarge', initial_instance_count=1)
```

Deploy any open source model on SageMaker

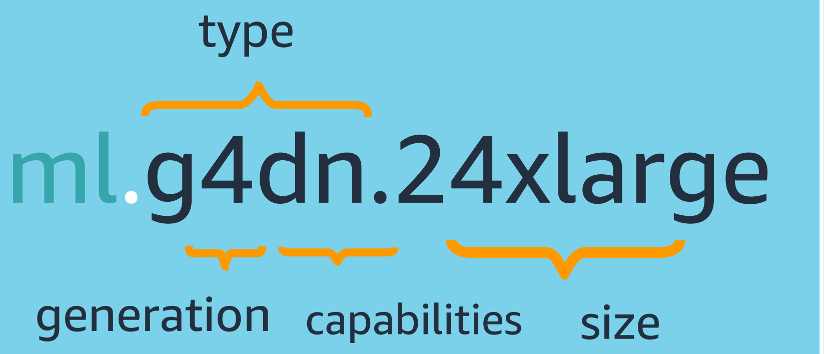
AutoGluon	Chainer	DMLC XGBoost	H20.ai	C++ Language	NVIDIA	Julia Language
RayRLlib	DeepSpeed	DataRobot	Docker	Go Language	Apache Airflow	AWS ML Marketplace
Deeplearning4j	Apache SparkML	Databricks	Deep Graph Library	PyCharm	Kubernetes	AWS IOT Greengrass

Amazon SageMaker supports the leading machine learning frameworks, toolkits, and programming languages

Leverage MLOps and account isolation at scale



Pick the right ML instance type and size



70+ options across 22 regions

General Purpose

	.large	.2xlarge	.12xlarge	.24xlarge
t2	2 vCPU 4 GiB	**8 vCPU** **32 GiB**		
m5	2 vCPU 8 GiB	8 vCPU 32 GiB	48 vCPU 192 GiB	96 vCPU 384 GiB
m5d	2 vCPU 8 GiB	8 vCPU 32 GiB	48 vCPU 192 GiB	96 vCPU 384 GiB

Compute Optimized

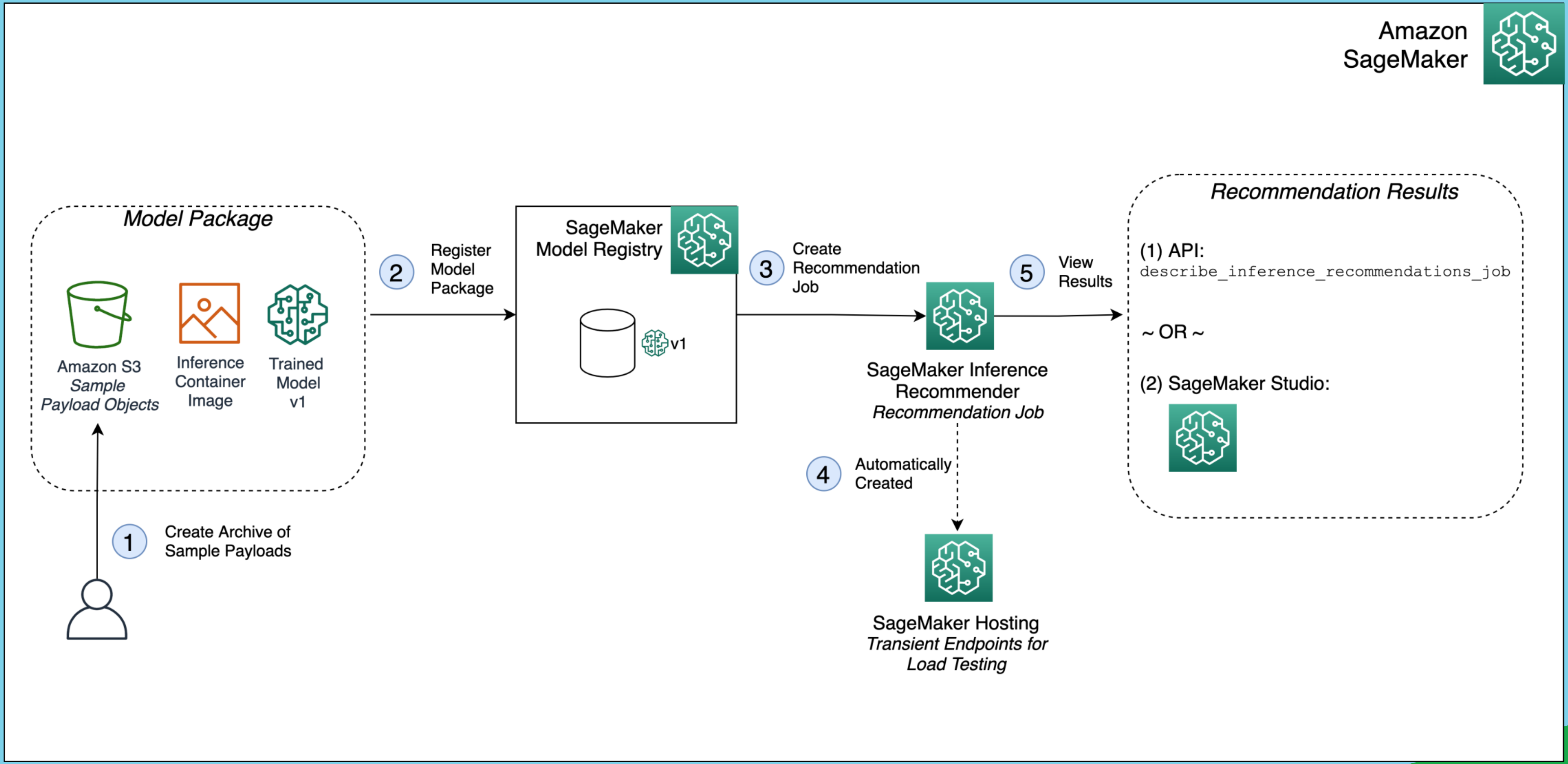
			.9xlarge	.18xlarge
c5	2 vCPU 4 GiB	4 vCPU 8 GiB	36 vCPU 72 GiB	72 vCPU 144 GiB
c5d	2 vCPU 4 GiB	8 vCPU 16 GiB	36 vCPU 72 GiB	72 vCPU 144 GiB

Accelerated Computing

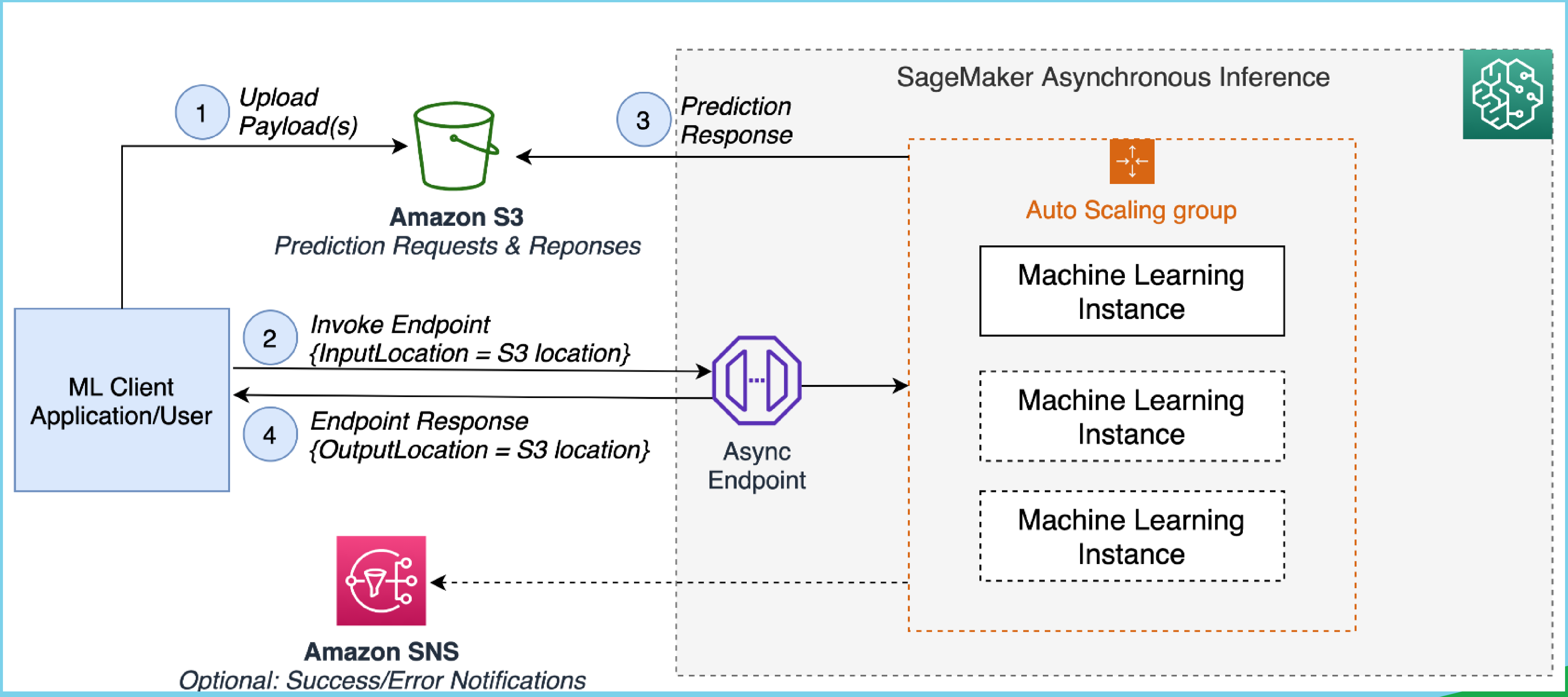
			.16xlarge	.24xlarge
p3		8 vCPU 61 GiB	64 vCPU 488 GiB	
g4dn	4 vCPU 16 GiB	8 vCPU 32 GiB	64 vCPU 258 GiB	
inf1	4 vCPU 8 GiB	8 vCPU 16 GiB		96 vCPU 192 GiB

Announcing SageMaker Inference Recommender

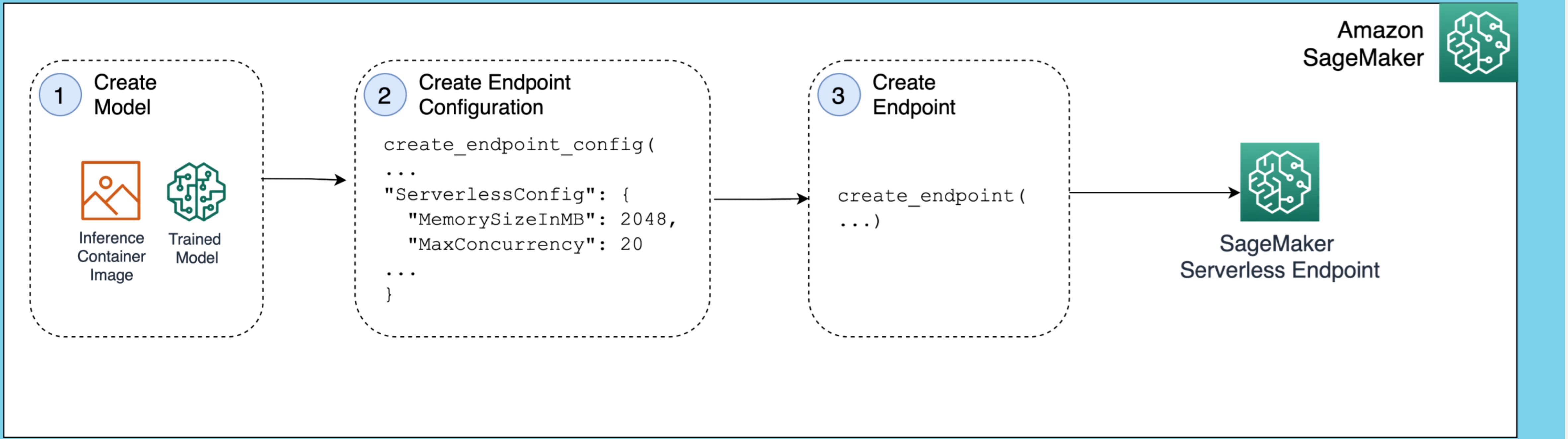
re:Invent 2021



Asynchronous inference with SageMaker


SageMaker Asynchronous Inference
1
Upload Payload(s)
Amazon S3
Prediction Requests & Reponses
3
Prediction Response
Auto Scaling group
Machine Learning Instance
Machine Learning Instance
Machine Learning Instance
ML Client Application/User
2
Invoke Endpoint {InputLocation = S3 location}
4
Endpoint Response {OutputLocation = S3 location}
Async Endpoint
Amazon SNS
Optional: Success/Error Notifications

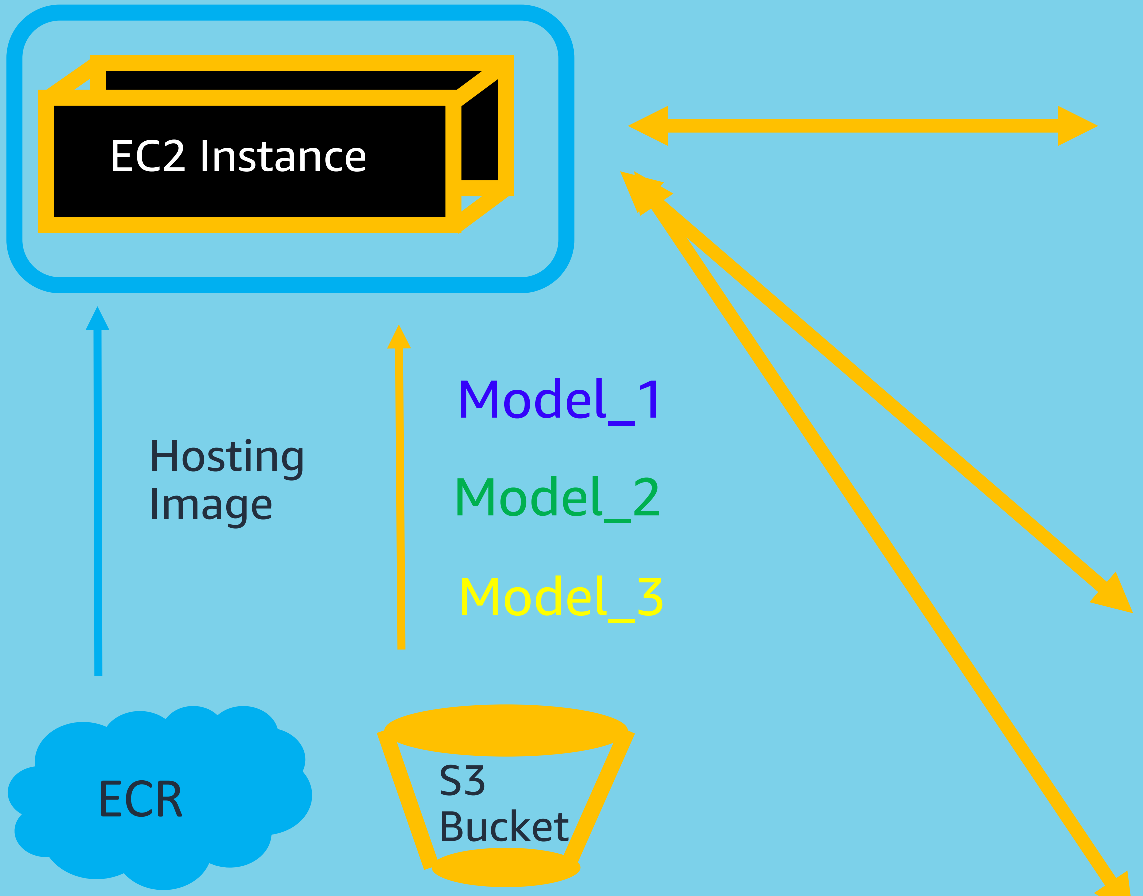
Serverless inference with SageMaker


Amazon SageMaker
1 Create Model
Inference Container Image
Trained Model
2 Create Endpoint Configuration
3 Create Endpoint
SageMaker Serverless Endpoint

```
create_endpoint_config(
...
"ServerlessConfig": {
  "MemorySizeInMB": 2048,
  "MaxConcurrency": 20
...
}
```

```
create_endpoint(
...)
```

Serve thousands of models with Multi-Model Endpoints



EC2 Instance

Hosting Image

Model_1

Model_2

Model_3

ECR

S3 Bucket

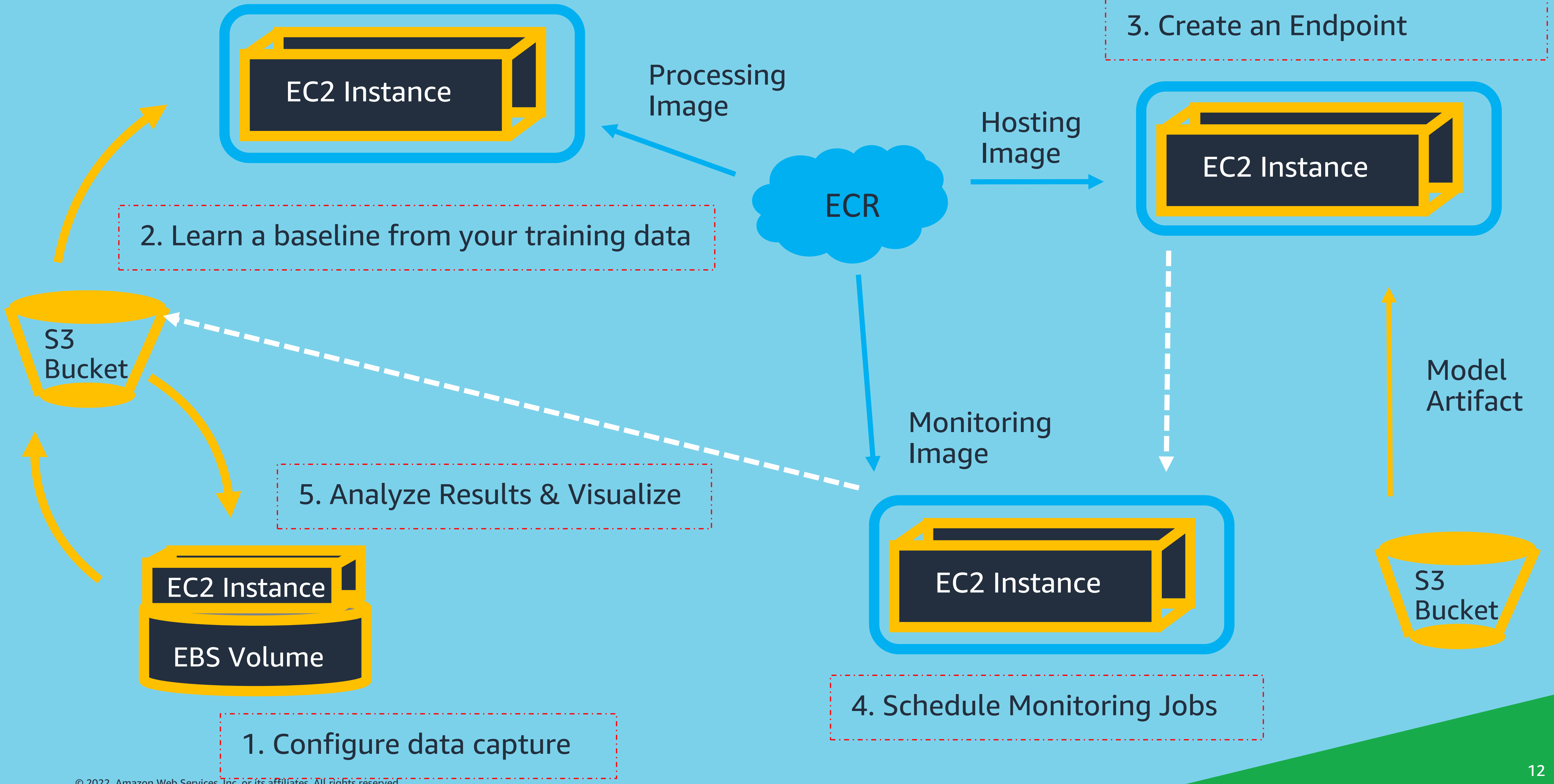
```
client.invoke_endpoint(
            EndpointName = ‘my-endpoint’,
            ContentType = ‘text/csv’
            TargetModel = ‘model_1.tar.gz’,
            Body = body)
```

```
client.invoke_endpoint(
            TargetModel = ‘model_2.tar.gz’)
```

```
client.invoke_endpoint(
            TargetModel = ‘model_3.tar.gz’)
```

Invoke the *relative S3 path* for your model, and SageMaker handles the rest

SageMaker Model Monitor



1. Configure data capture

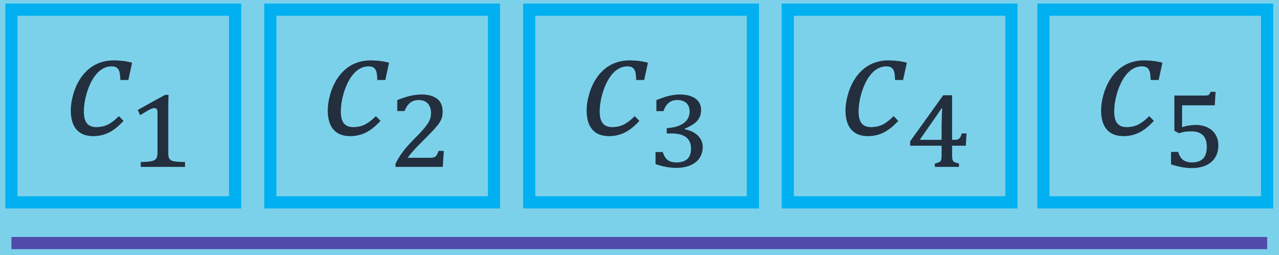
2. Learn a baseline from your training data

3. Create an Endpoint

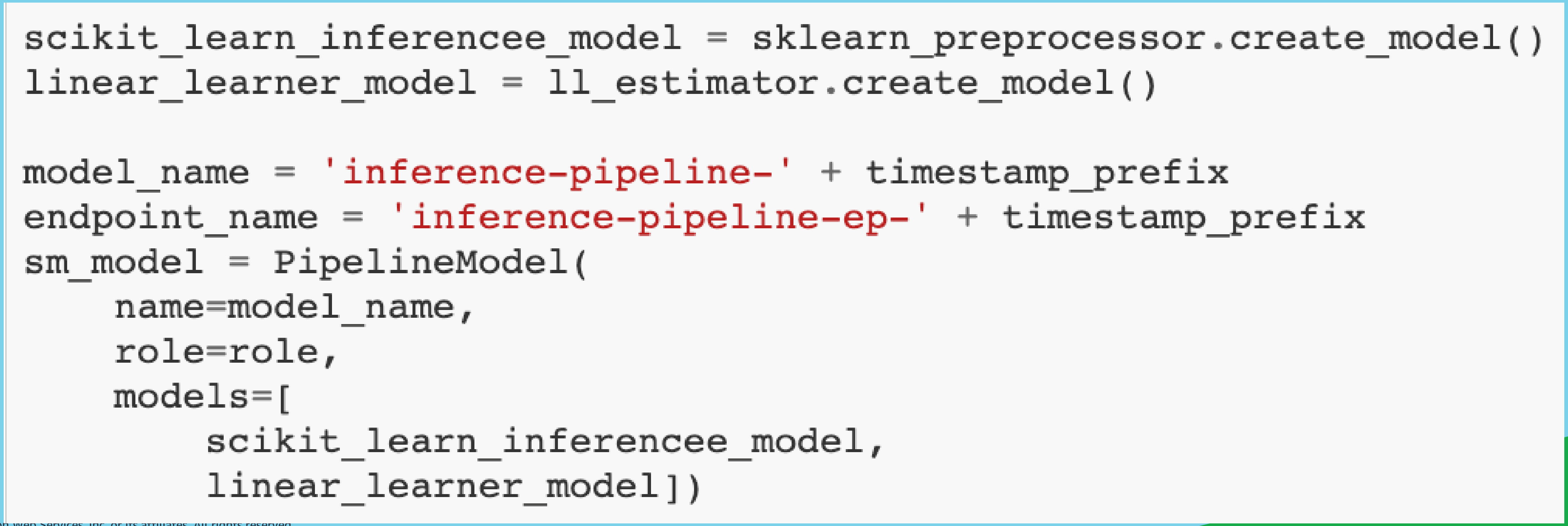
4. Schedule Monitoring Jobs

5. Analyze Results & Visualize

Create inference pipelines with SageMaker Pipeline Models



Create up to 5 containers, and connect these by creating a *Pipeline Model.*
Run this on both endpoints and batch transform. This is how Autopilot deploys!



```
scikit_learn_inferencee_model = sklearn_preprocessor.create_model()
linear_learner_model = ll_estimator.create_model()

model_name = 'inference-pipeline-' + timestamp_prefix
endpoint_name = 'inference-pipeline-ep-' + timestamp_prefix
sm_model = PipelineModel(
    name=model_name,
    role=role,
    models=[
        scikit_learn_inferencee_model,
        linear_learner_model])
```

Drive down cost per inference with accelerators

M5 M5a M6g C5 C5a C6g R5 R5a R6 R6g

Training + Inference

P3d P4d

Training

Inf1 G4dn

Inference

AWS Inferentia

45% lower cost performance
30% higher throughput
Runs billions of Alexa inferences 25% faster

AWS Trainium

Most teraflops of any ML instance in the cloud
Same Neuron SDK as Inferentia
Available in 2022 as EC2 and SageMaker instances

Fastest training times on Mask-RCNN and T5-3B

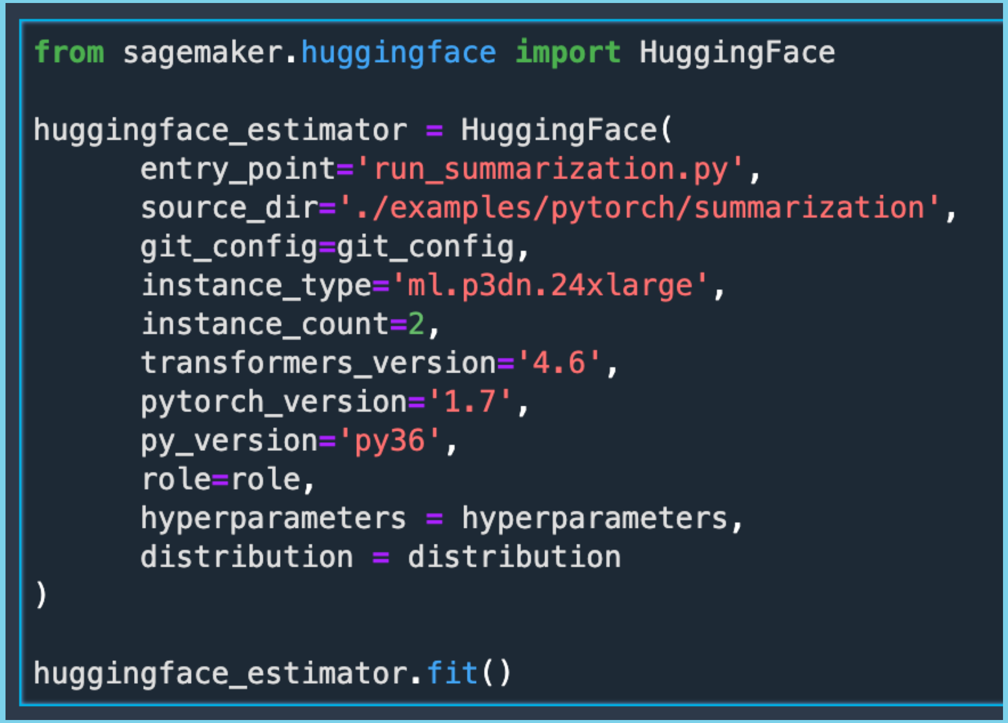
TensorFlow

Mask-RCNN
From 28 min to
6 min, 13 sec

PyTorch

T5-3B
From weeks to
5.9 days

Train and deploy 26k+ Hugging Face models on SageMaker

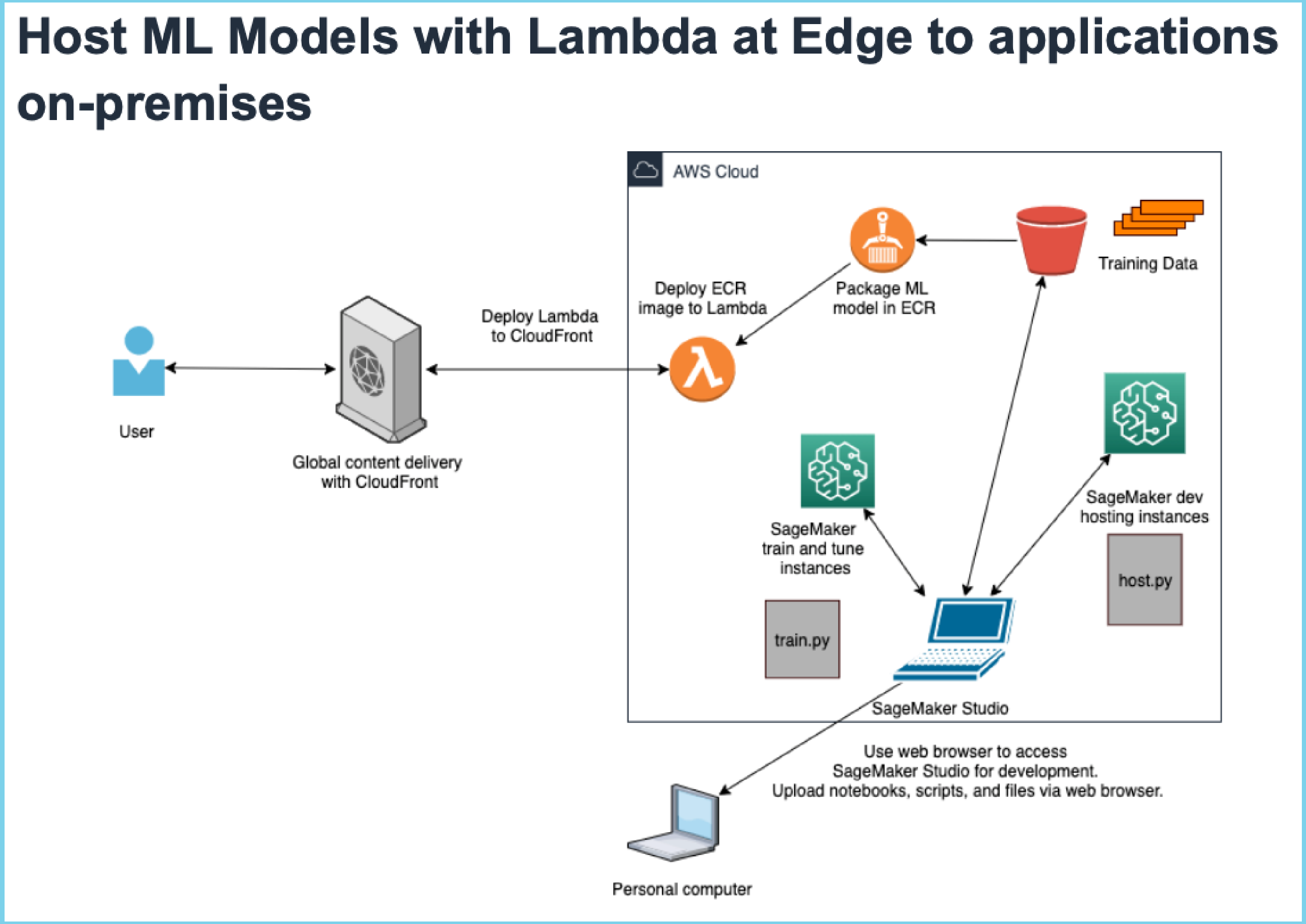


```
from sagemaker.huggingface import HuggingFace

huggingface_estimator = HuggingFace(
      entry_point='run_summarization.py',
      source_dir='./examples/pytorch/summarization',
      git_config=git_config,
      instance_type='ml.p3dn.24xlarge',
      instance_count=2,
      transformers_version='4.6',
      pytorch_version='1.7',
      py_version='py36',
      role=role,
      hyperparameters = hyperparameters,
      distribution = distribution
)

huggingface_estimator.fit()
```

- ✓ Built-in integration with SageMaker distributed libraries
- ✓ Default to latest versions of frameworks
- ✓ Customize with any packages / software you prefer
- ✓ Config generation for hundreds of supported models

huggingface / **transformers**

Hybrid ML patterns for deployment



Host ML Models with Lambda at Edge to applications on-premises

AWS Cloud
Training Data
Package ML model in ECR
Deploy ECR image to Lambda
Deploy Lambda to CloudFront
User
Global content delivery with CloudFront
SageMaker dev hosting instances
host.py
SageMaker train and tune instances
train.py
SageMaker Studio
Use web browser to access SageMaker Studio for development. Upload notebooks, scripts, and files via web browser.
Personal computer

https://tinyurl.com/44cmaw67

Demo

Thank you!