Continuous Delivery of ML models

Machine Learning Immersion Day

Module 4

Agenda

09:00 Welcome and Introductions
09:30 Data Science Lifecycle
10:15 Lab 1: Understanding and preparing data with S3, Glue and Athena
11:00 Finnair Data Platform
11:30 Lunch
12:15 Model training, testing and deploying with Sagemaker
13:00 Lab 2: Train, test and deploy your first model with Sagemaker
14:30 Bring your own model
15:00 Break
15:15 Continuous Delivery of ML models
15:45 Lab 3: Continuous Delivery of ML models to Amazon SageMaker
16:30 Wrap Up

DevOps

What is DevOps?

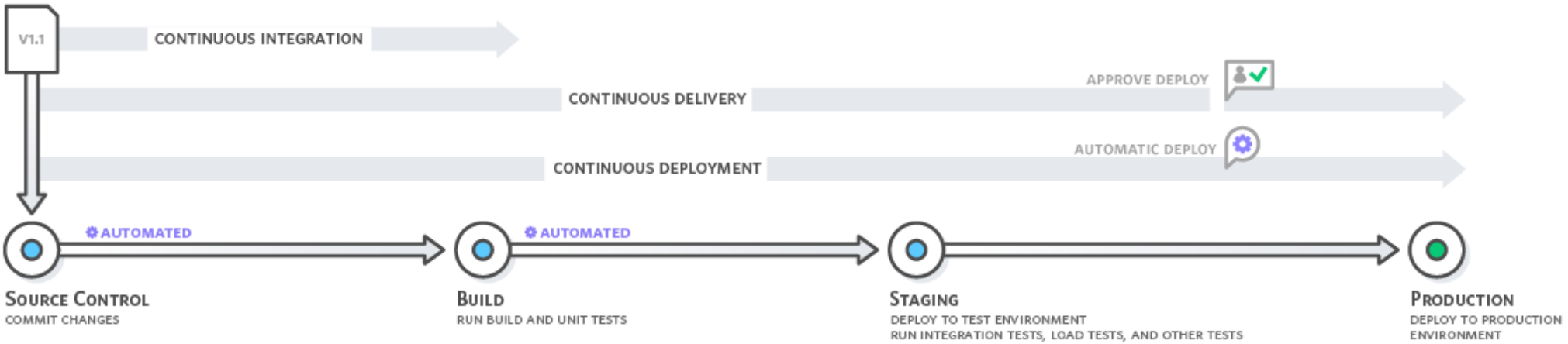
- Cultural philosophies
- Practices
- Tools

DevOps Culture

- Dev & Ops coming together
    - No more “silos”
- Shared responsibility
- Ownership
- Visibility and communication

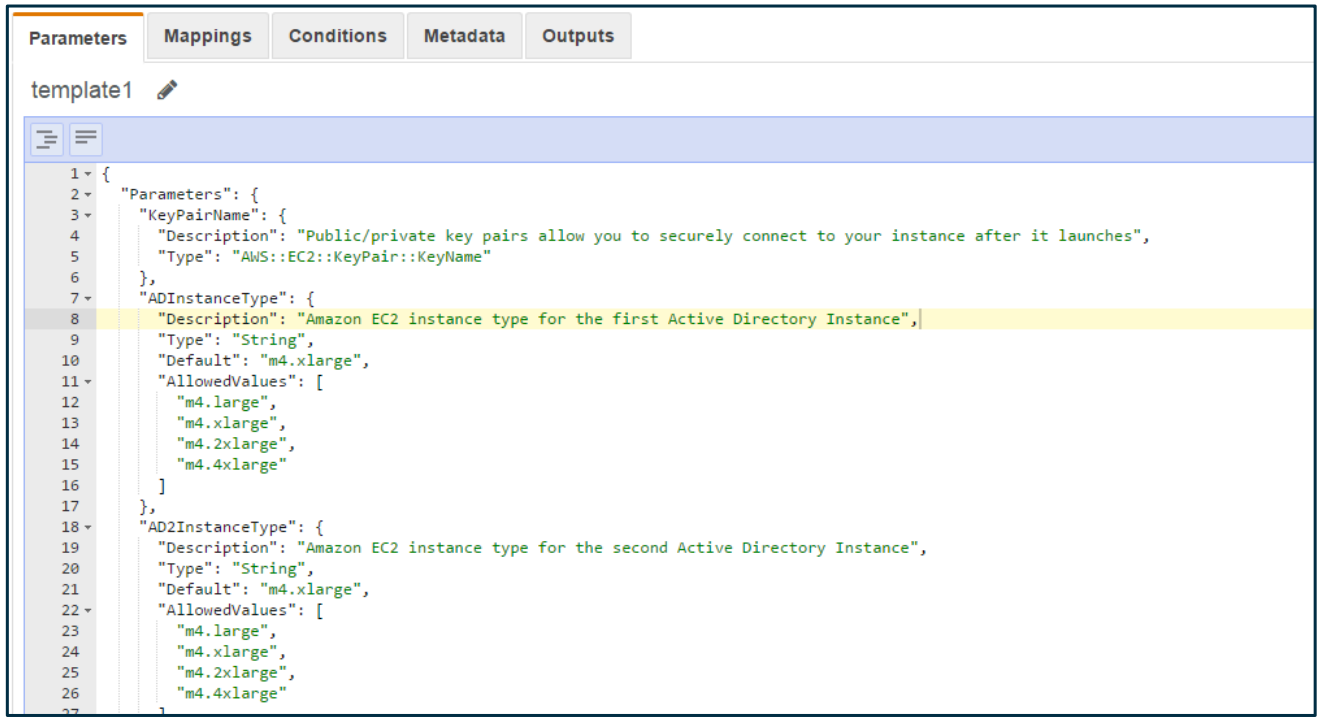
DevOps Practices

- Continuous Integration
- Continuous Delivery & Deployment



DevOps Practices

- Infrastructure as Code
  - Model your AWS resources using code



DevOps Practices

- Monitoring and Logging
  - Track and analyze metrics and logs
  - Understand real-time performance of infrastructure and application

Benefits of DevOps

Improved Collaboration

Rapid Delivery

Reliability

Security

Scale

Speed

aws

Operating SageMaker in Production

aws

Creating endpoints

Easy deployment to production REST API

Scalable, high throughput, and high reliability

aws

Creating endpoints

Model

```
aws sagemaker create-model
  --model-name model1
  --primary-container ‘{“Image”: “123.dkr.ecr.amazonaws.com/algo”,
                        “ModelDataUrl”: “s3://bkt/model1.tar.gz”}’
  --execution-role-arn arn:aws:iam::123:role/me
```


aws

Creating endpoints

Model

```
aws sagemaker create-model
  --model-name model1
  --primary-container '{"Image": "123.dkr.ecr.amazonaws.com/algo",
                        "ModelDataUrl": "s3://bkt/model1.tar.gz"}'
  --execution-role-arn arn:aws:iam::123:role/me
```

Endpoint configuration

```
aws sagemaker create-endpoint-config
  --endpoint-config-name model1-config
  --production-variants '{"InitialInstanceCount": 2,
                          "InstanceType": "ml.m4.xlarge",
                          "InitialVariantWeight": 1,
                          "ModelName": "model1",
                          "VariantName": "AllTraffic"}'
```

Creating endpoints

Model

```
aws sagemaker create-model
  --model-name model1
  --primary-container '{"Image": "123.dkr.ecr.amazonaws.com/algo",
                        "ModelDataUrl": "s3://bkt/model1.tar.gz"}'
  --execution-role-arn arn:aws:iam::123:role/me
```

Endpoint configuration

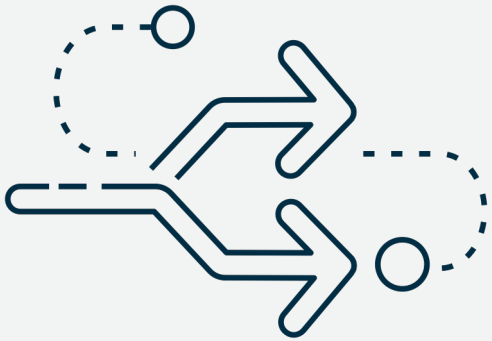
```
aws sagemaker create-endpoint-config
  --endpoint-config-name model1-config
  --production-variants '{"InitialInstanceCount": 2,
                          "InstanceType": "ml.m4.xlarge",
                          "InitialVariantWeight": 1,
                          "ModelName": "model1",
                          "VariantName": "AllTraffic"}'
```

Endpoint

```
aws sagemaker create-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name model1-config
```


aws

Updating endpoints



Blue-green deployments means no scheduled downtime

Deploy one or more models behind the same endpoint

Updating endpoints

New model

```
aws sagemaker create-model
  --model-name model2
  --primary-container ‘{“Image”: “123.dkr.ecr.amazonaws.com/algo”,
                        “ModelDataUrl”: “s3://bkt/model2.tar.gz”}’
  --execution-role-arn arn:aws:iam::123:role/me
```

Updating endpoints

New model

```
aws sagemaker create-model
  --model-name model2
  --primary-container '{"Image": "123.dkr.ecr.amazonaws.com/algo",
                        "ModelDataUrl": "s3://bkt/model2.tar.gz"}'
  --execution-role-arn arn:aws:iam::123:role/me
```

New endpoint configuration

```
aws sagemaker create-endpoint-config
  --endpoint-config-name model2-config
  --production-variants '{"InitialInstanceCount": 2,
                          "InstanceType": "ml.m4.xlarge",
                          "InitialVariantWeight": 1,
                          "ModelName": "model2",
                          "VariantName": "AllTraffic"}'
```


aws

Updating endpoints

New model

```
aws sagemaker create-model
  --model-name model2
  --primary-container '{"Image": "123.dkr.ecr.amazonaws.com/algo",
                        "ModelDataUrl": "s3://bkt/model2.tar.gz"}
  --execution-role-arn arn:aws:iam::123:role/me
```

New endpoint configuration

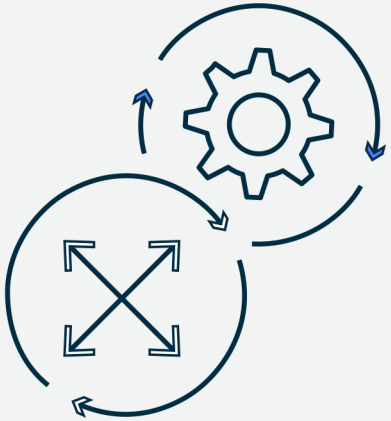
```
aws sagemaker create-endpoint-config
  --endpoint-config-name model2-config
  --production-variants '{"InitialInstanceCount": 2,
                          "InstanceType": "ml.m4.xlarge",
                          "InitialVariantWeight": 1,
                          "ModelName": "model2",
                          "VariantName": "AllTraffic"}'
```

Same endpoint

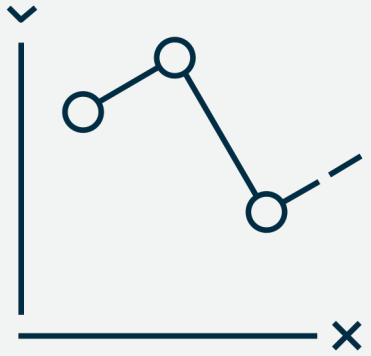
```
aws sagemaker update-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name model2-config
```


aws

Reduced risk deployments



Incrementally retrain models with new data



Try new models and improved algorithms

Reduced risk deployments

Two model endpoint configuration

```
aws sagemaker create-endpoint-config
  --endpoint-config-name both-models-config
  --production-variants '[{"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 95,
                           "ModelName": "model1",
                           "VariantName": "model1-traffic"},
                          {"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 5,
                           "ModelName": "model2",
                           "VariantName": "model2-traffic"}]'
```


aws

Reduced risk deployments

Two model endpoint configuration

```
aws sagemaker create-endpoint-config
  --endpoint-config-name both-models-config
  --production-variants '[{"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 95,
                           "ModelName": "model1",
                           "VariantName": "model1-traffic"},
                          {"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 5,
                           "ModelName": "model2",
                           "VariantName": "model2-traffic"}]'
```

Same endpoint

```
aws sagemaker update-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name both-models-config
```


aws

Reduced risk deployments

Two model endpoint configuration

```
aws sagemaker create-endpoint-config
  --endpoint-config-name both-models-config
  --production-variants '[{"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 95,
                           "ModelName": "model1",
                           "VariantName": "model1-traffic"},
                          {"InitialInstanceCount": 2,
                           "InstanceType": "ml.m4.xlarge",
                           "InitialVariantWeight": 5,
                           "ModelName": "model2",
                           "VariantName": "model2-traffic"}]'
```

Same endpoint

```
aws sagemaker update-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name both-models-config
```

Swap

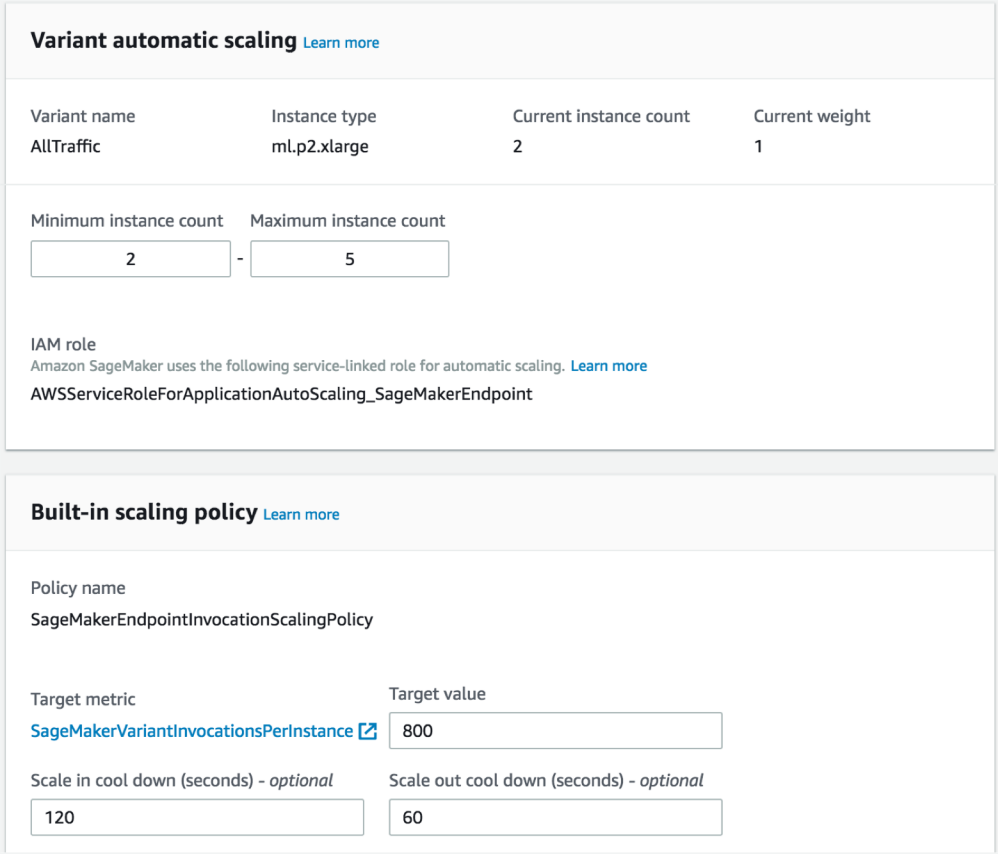
```
aws sagemaker update-endpoint-weights-and-capacities
  --endpoint-name my-endpoint
  --desired-weights-and-capacities '{"VariantName": "model1",
                                     "DesiredWeight": 5}'
```


aws

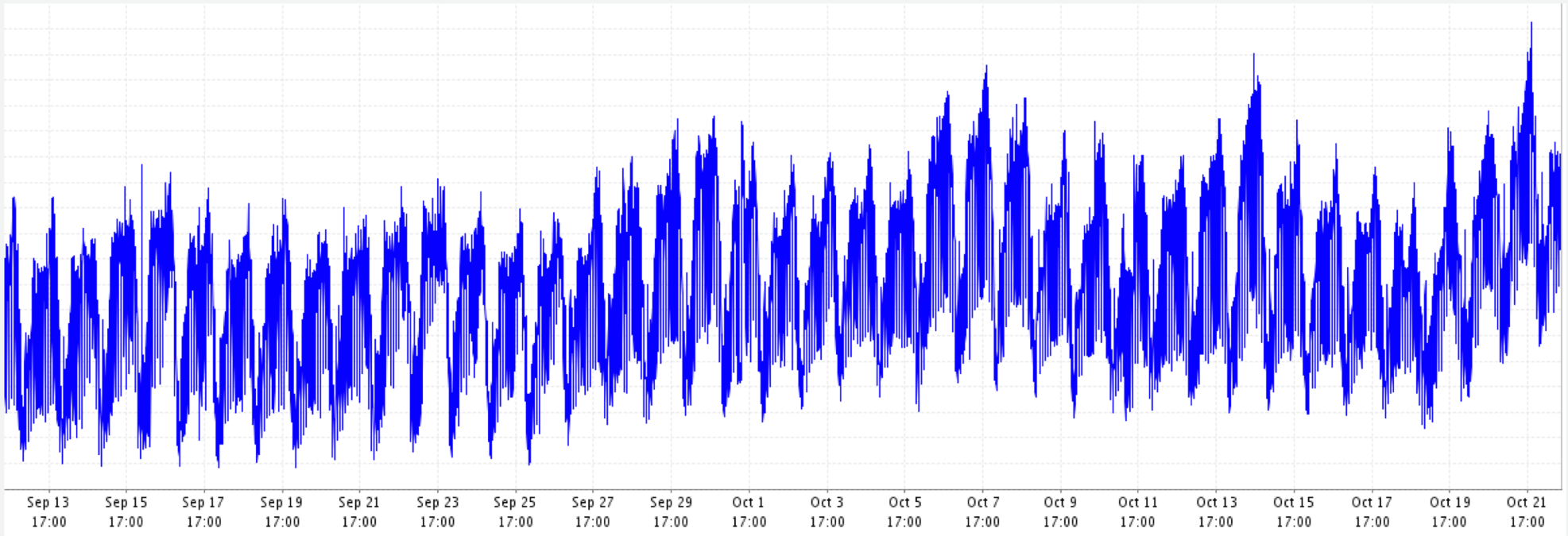
Automatically scaling endpoints

SageMaker console settings:

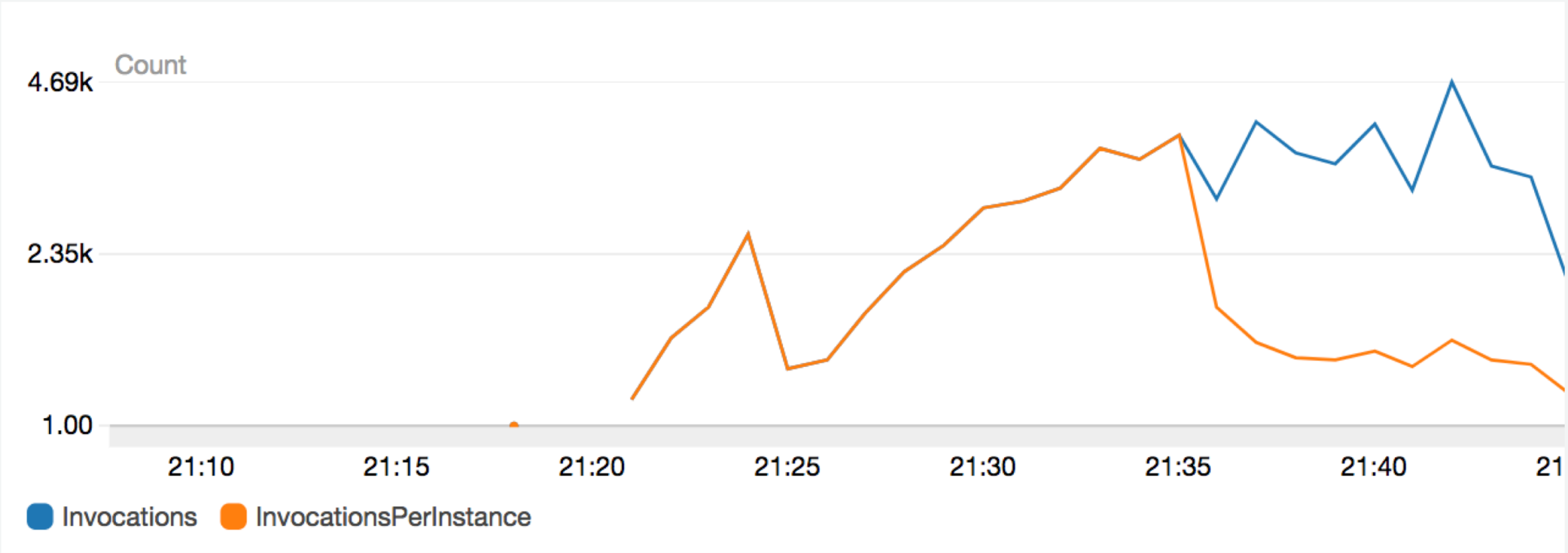
- Min and max instances
- Target invocations per instance
- Scaling cool downs



Why automatic scaling?

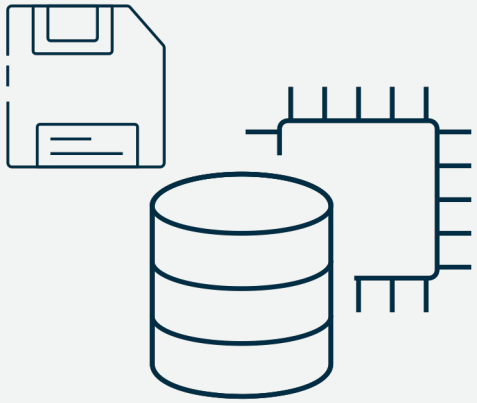


Automatic scaling in action



aws

Scaling criteria



Algorithms have different memory, CPU, or GPU requirements

Autoscale based on endpoint instance's CloudWatch Metrics

Creating an automatic scaling policy

Variant

```
aws application-autoscaling register-scalable-target
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --min-capacity 2
  --max-capacity 5
```


aws

Creating an automatic scaling policy

Variant

```
aws application-autoscaling register-scalable-target
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --min-capacity 2
  --max-capacity 5
```

Policy

```
aws application-autoscaling put-scaling-policy
  --policy-name model2-scaling
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --policy-type TargetTrackingScaling
  --target-tracking-scaling-policy-configuration
    '{"TargetValue": 50,
      "CustomizedMetricSpecification":
        {"MetricName": "CPUUtilization",
         "Namespace": "/aws/sagemaker/Endpoints",
         "Dimensions":
           [{"Name": "EndpointName", "Value": "my-endpoint"},
            {"Name": "VariantName","Value": "model2"}],
        "Statistic": "Average",
        "Unit": "Percent"}}'
```


aws

Creating an automatic scaling policy

Variant

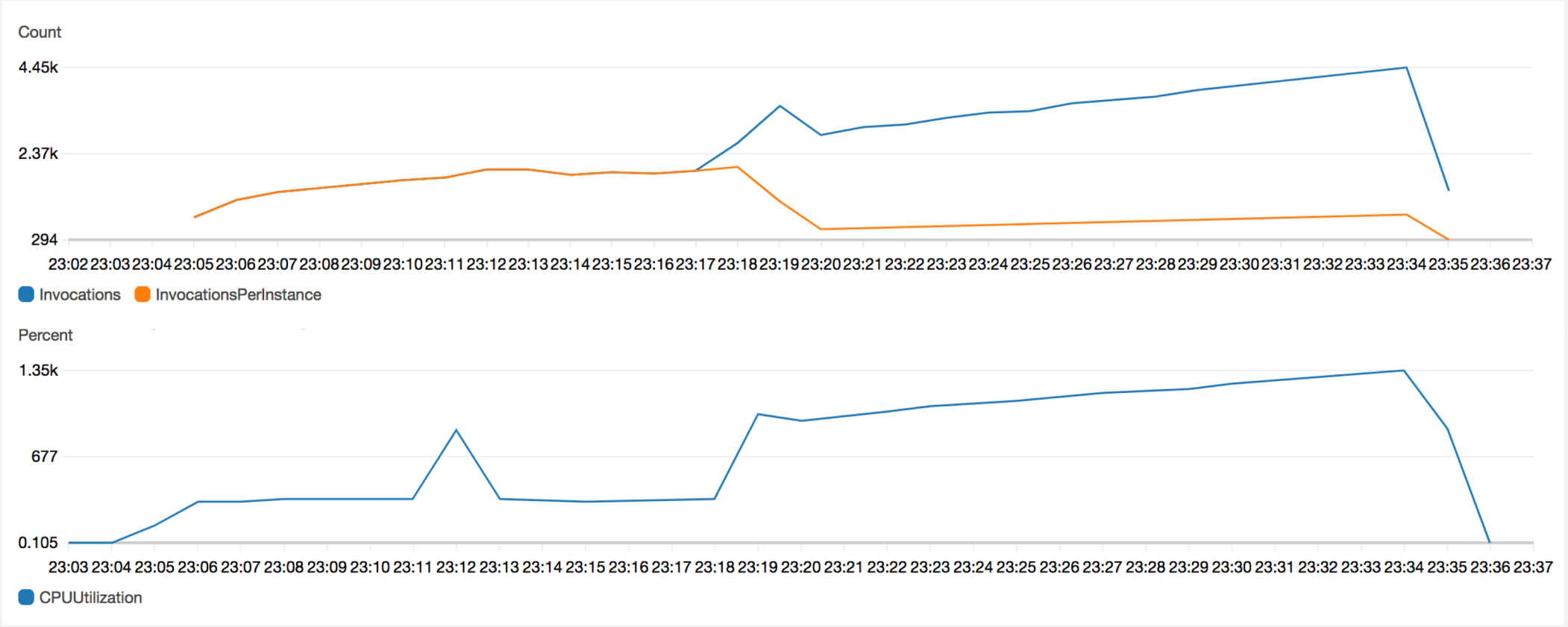
```
aws application-autoscaling register-scalable-target
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --min-capacity 2
  --max-capacity 5
```

Policy

```
aws application-autoscaling put-scaling-policy
  --policy-name model2-scaling
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --policy-type TargetTrackingScaling
  --target-tracking-scaling-policy-configuration
    '{"TargetValue": 50,
      "CustomizedMetricSpecification":
        {"MetricName": "CPUUtilization",
         "Namespace": "/aws/sagemaker/Endpoints",
         "Dimensions":
           [{"Name": "EndpointName", "Value": "my-endpoint"},
            {"Name": "VariantName","Value": "model2"}],
        "Statistic": "Average",
        "Unit": "Percent"}}'
```


aws

Scale by utilization

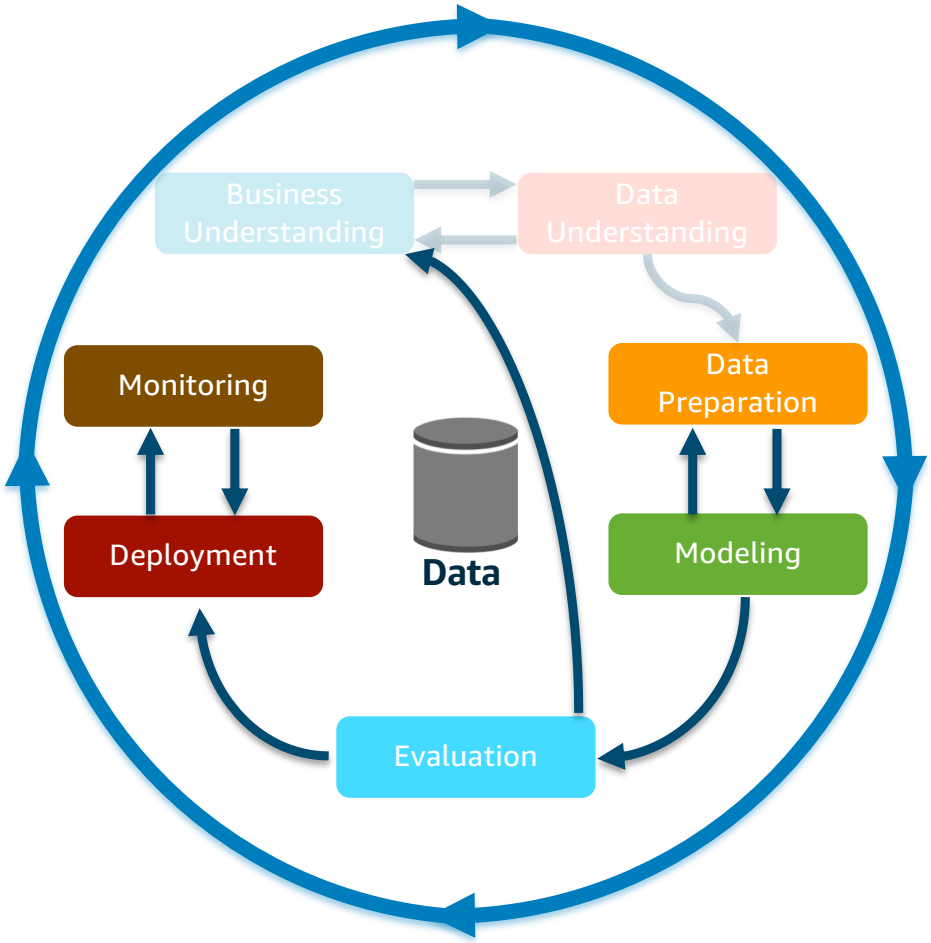


aws

DevOps and ML

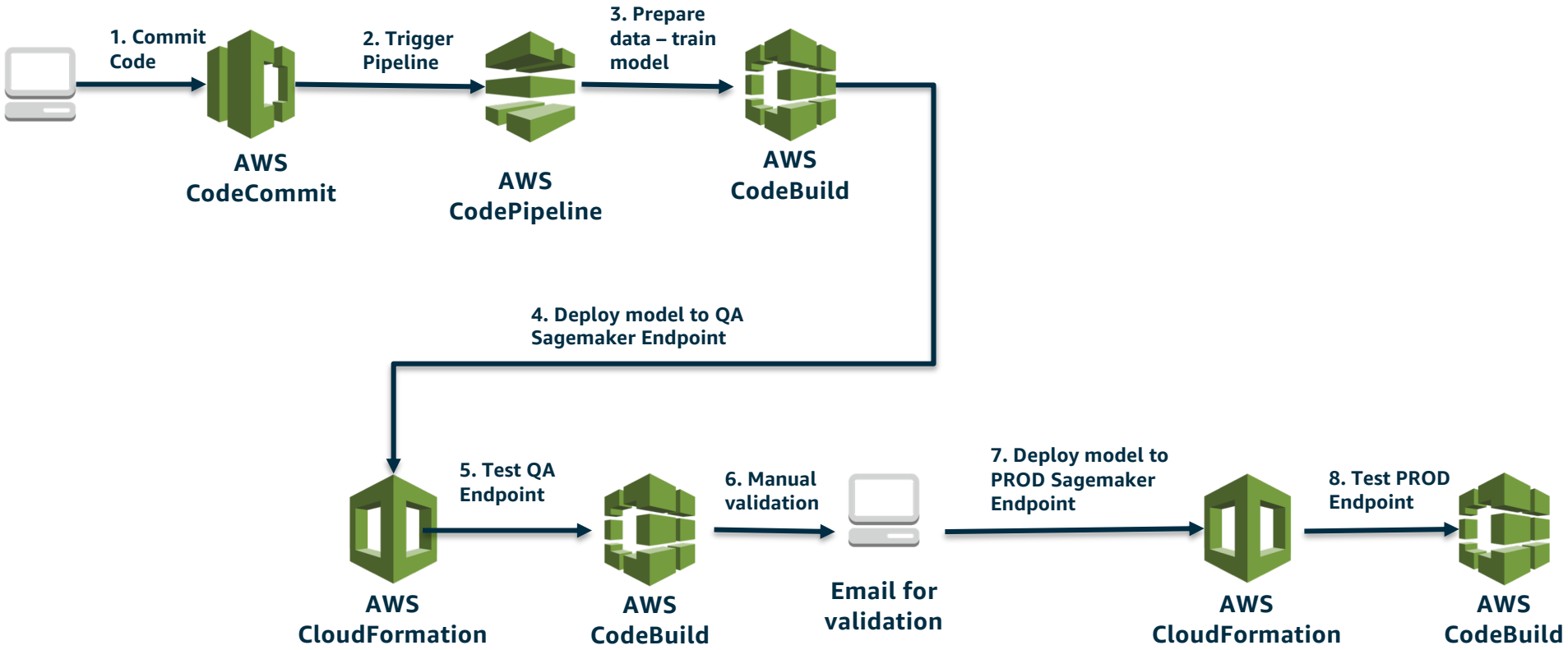
Why Continuous Delivery?

- Roll out improved ML models as quickly as possible
- Predictable and reproducible environments
- Immutable containers and models. They will run the same in every environment
- Fast feedback



aws

Continuous Delivery example



CodeBuild – Prepare Data & Train Model

```
BuildSpec: !Sub |
  version: 0.2
  phases:
    install:
      commands:
        - echo "Installing boto3 and sagemaker"
        - pip3 install boto3
        - pip3 install sagemaker
    build:
      commands:
        - echo "Running prepdata.py"
        - python3 Source/prepdata.py "${DataBucket}" "${GlueCatalogDatabase}"
        - echo "Running train.py"
        - python3 Source/train.py "${SagemakerRole.Arn}" "${DataBucket}" "${AWS::StackName}" $CODEBUILD_RESOLVED_SOURCE_VERSION
    post_build:
      commands:
        - echo "Cleaning. We leave ua.test for later predictor tests"
        - rm -f u.data
        - rm -f ua.base*
  artifacts:
    files:
      - '**/*'
```


aws

CloudFormation – Deploy Model to Endpoint

```
Resources:
  Model:
    Type: "AWS::SageMaker::Model"
    Properties:
      ModelName: !Sub ${Environment}-${ParentStackName}-${CommitID}-${Timestamp}
      ExecutionRoleArn: !Sub ${SageMakerRole}
      PrimaryContainer:
        ModelDataUrl: !Sub ${ModelData}
        Image: !Sub ${ContainerImage}
  Endpoint:
    Type: "AWS::SageMaker::Endpoint"
    DependsOn: EndpointConfig
    Properties:
      EndpointName: !Sub ${Environment}-${ParentStackName}-${CommitID}-${Timestamp}
      EndpointConfigName: !GetAtt EndpointConfig.EndpointConfigName
  EndpointConfig:
    Type: "AWS::SageMaker::EndpointConfig"
    DependsOn: Model
    Properties:
        EndpointConfigName: !Sub ${Environment}-${ParentStackName}-${CommitID}-${Timestamp}
        ProductionVariants:
        - ModelName: !GetAtt Model.ModelName
          VariantName: AllTraffic
          InitialInstanceCount: 1
          InstanceType: ml.t2.medium
          InitialVariantWeight: 1
```


aws

CodeBuild – Test Endpoint

```
BuildSpec: !Sub |
  version: 0.2
  phases:
    install:
      commands:
        - echo "Installing boto3 and sagemaker"
        - pip3 install boto3
        - pip3 install sagemaker
    build:
      commands:
        - echo "Running test.py"
        - python3 Source/test.py "qa-${AWS::StackName}" "prepdata_result.json" "CloudFormation/configuration_qa.json"
```

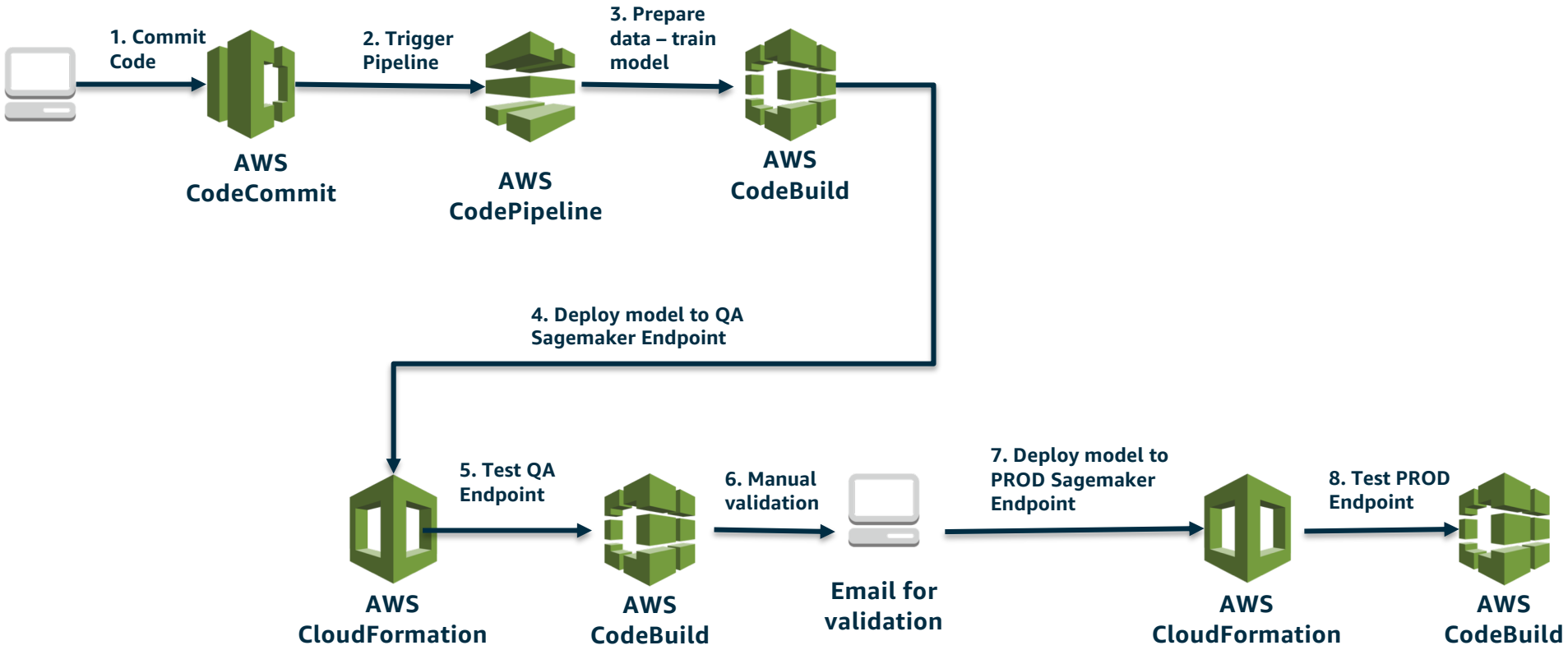

aws

CodeBuild – Test Endpoint

```
fm_predictor = sagemaker.predictor.RealTimePredictor(endpoint_name,
                                                     serializer=fm_serializer,
                                                     deserializer=json_deserializer,
                                                     content_type='application/json',
                                                     sagemaker_session=sagemaker.Session())

nb_predictions = 10
offset = 1000

# Run some predictions
result = fm_predictor.predict(X_test[offset:offset+nb_predictions].toarray())
pprint.pprint(result["predictions"])
pprint.pprint(Y_test[offset:offset+nb_predictions])

# Compare predictions to labelled test data to check for match rate
matches = 0
for index in list(range(nb_predictions)):
    offset_index = offset + index
    if int(result["predictions"][index]["predicted_label"]) == int(Y_test[offset_index]):
        matches = matches + 1

match_rate = matches / nb_predictions
print("Match Rate: %s" % (match_rate))

# If match rate is not 80% we throw an error that will break the codepipeline test stage
assert match_rate >= 0.80
```


aws

Continuous Delivery example



Lab 3: Continuous Delivery of ML models to Amazon SageMaker

aws

Questions?