Building ML environments for regulatory customers

Suraj Muraleedharan, AWS Professional Services

The reach of ML is growing

INCREASED SPENDING

By 2024, global spending on artificial intelligence will reach $110 billion

—IDC

FROM PILOTING TO OPERATIONALISING

By the end of 2024, 75% of enterprises will shift from piloting to operationalising AI

—Gartner

AI TRANSFORMATION

57% said that AI would transform their organisation in the next three years

—Deloitte

Transform customer experience

Improve business operations

Better and faster decision-making

Innovation

More than one hundred thousand customers use AWS for machine learning

AUTODESK · BAYER · Capital One · Cerner · CONDÉ NAST · Domino's · infor

INTERCOM · intuit · INVISTA · Jack in the box · KOCH INDUSTRIES INC · Lenovo · LexisNexis · Liberty Mutual INSURANCE · lyft · mercado libre

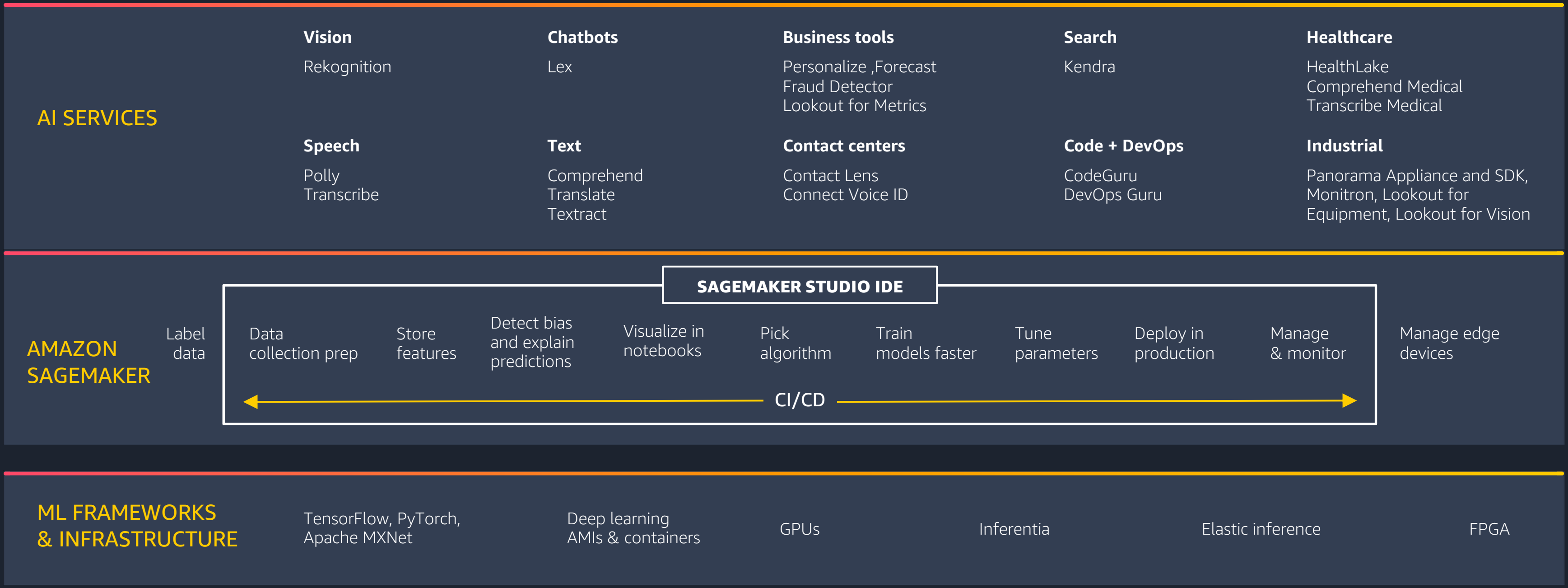
move · NASCAR · NFL · nerdwallet · NOVARTIS · NuData Security mastercard · OakNorth · PELOTON · Philips HealthSuite

Propeller · ProQuest · pwc · Roche · Root Insurance · salesforce · sallie mae · SIEMENS · slack · Slice

snapdocs · StanleyBlack&Decker · SUBWAY · THOMSON REUTERS · tinder · T · TRANSFIX · twilio · VEOLIA ENVIRONNEMENT · verizon

Machine Learning on AWS

The AWS ML stack



AI SERVICES

Vision
Rekognition

Chatbots
Lex

Business tools
Personalize ,Forecast
Fraud Detector
Lookout for Metrics

Search
Kendra

Healthcare
HealthLake
Comprehend Medical
Transcribe Medical

Speech
Polly
Transcribe

Text
Comprehend
Translate
Textract

Contact centers
Contact Lens
Connect Voice ID

Code + DevOps
CodeGuru
DevOps Guru

Industrial
Panorama Appliance and SDK, Monitron, Lookout for Equipment, Lookout for Vision

AMAZON SAGEMAKER

Label data

SAGEMAKER STUDIO IDE

Data collection prep

Store features

Detect bias and explain predictions

Visualize in notebooks

Pick algorithm

Train models faster

Tune parameters

Deploy in production

Manage & monitor

CI/CD

Manage edge devices

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow, PyTorch, Apache MXNet

Deep learning AMIs & containers

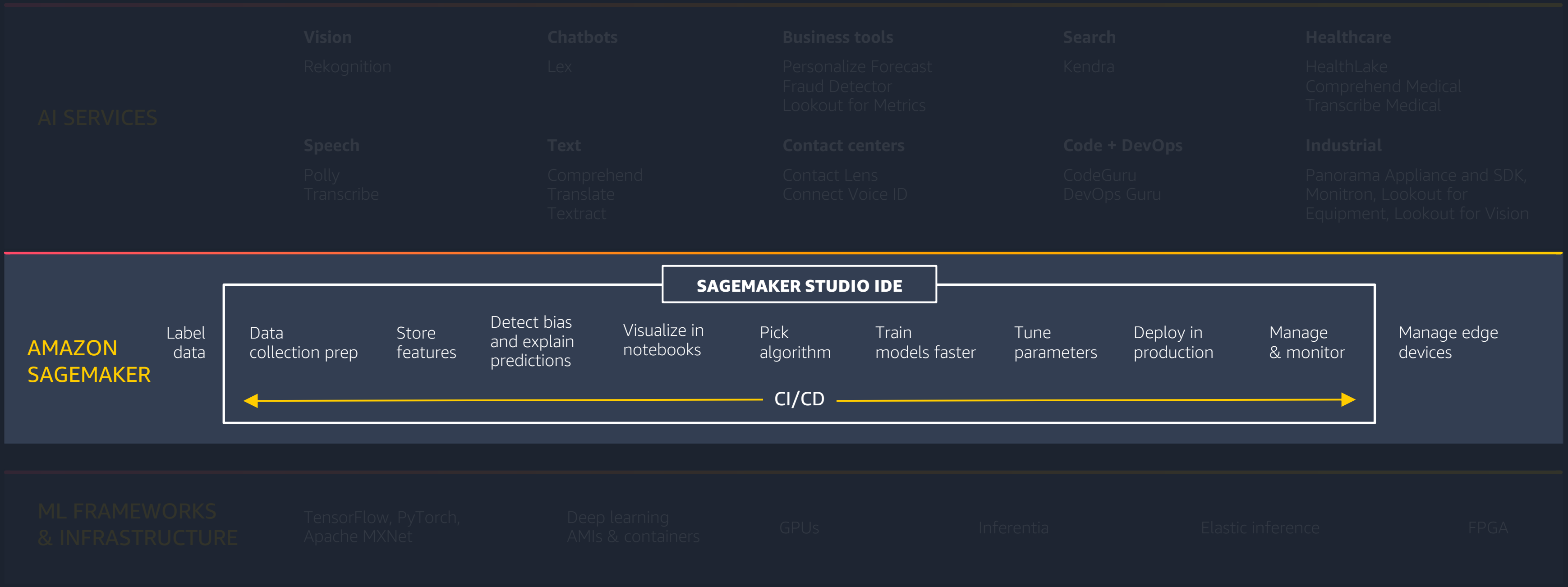
GPUs

Inferentia

Elastic inference

FPGA

The AWS ML stack



AI SERVICES

Vision	Chatbots	Business tools	Search	Healthcare
Rekognition	Lex	Personalize Forecast Fraud Detector Lookout for Metrics	Kendra	HealthLake Comprehend Medical Transcribe Medical
Speech	**Text**	**Contact centers**	**Code + DevOps**	**Industrial**
Polly Transcribe	Comprehend Translate Textract	Contact Lens Connect Voice ID	CodeGuru DevOps Guru	Panorama Appliance and SDK, Monitron, Lookout for Equipment, Lookout for Vision

AMAZON SAGEMAKER

Label data

SAGEMAKER STUDIO IDE

Data collection prep

Store features

Detect bias and explain predictions

Visualize in notebooks

Pick algorithm

Train models faster

Tune parameters

Deploy in production

Manage & monitor

CI/CD

Manage edge devices

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow, PyTorch, Apache MXNet

Deep learning AMIs & containers

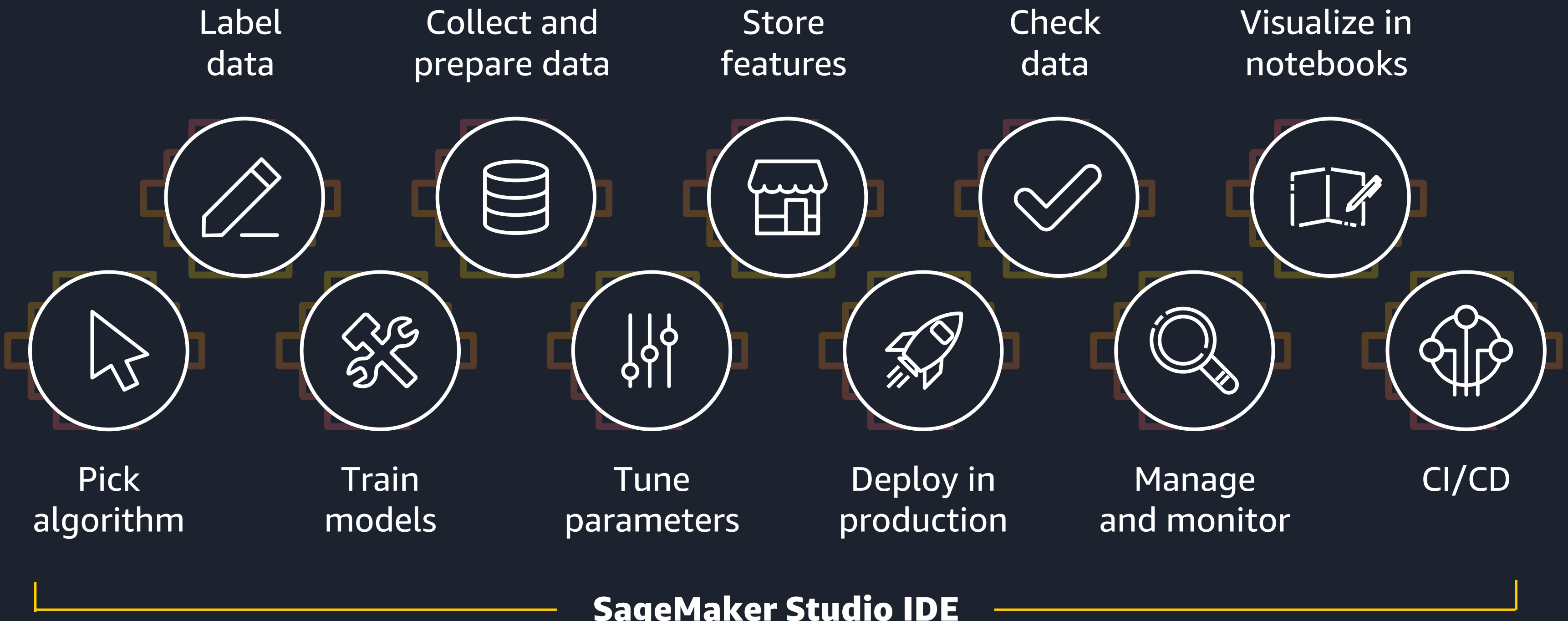
GPUs

Inferentia

Elastic inference

FPGA

Amazon SageMaker: Built to make ML **more accessible**



Label data

Collect and prepare data

Store features

Check data

Visualize in notebooks

Pick algorithm

Train models

Tune parameters

Deploy in production

Manage and monitor

CI/CD

SageMaker Studio IDE

INTEGRATED WORKBENCH

IDE designed specifically for ML, data preparation, experiment management, and pipelines

MANAGED INFRASTRUCTURE

Designed for ultra low latency and high throughput; automatic scaling, and distributed training

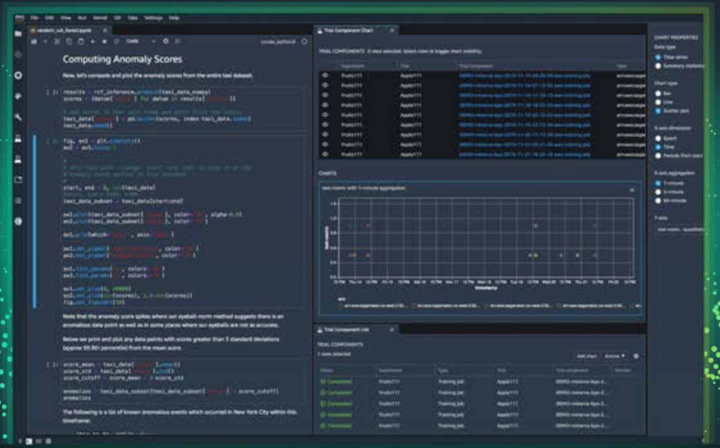
MANAGED TOOLING

Purpose-built from the ground up to work together incl. Autopilot, collaboration, notebooks, experiments, debugger, and model monitor

https://aws.amazon.com/sagemaker

Amazon SageMaker

Most complete, end-to-end ML service



Computing Anomaly Scores

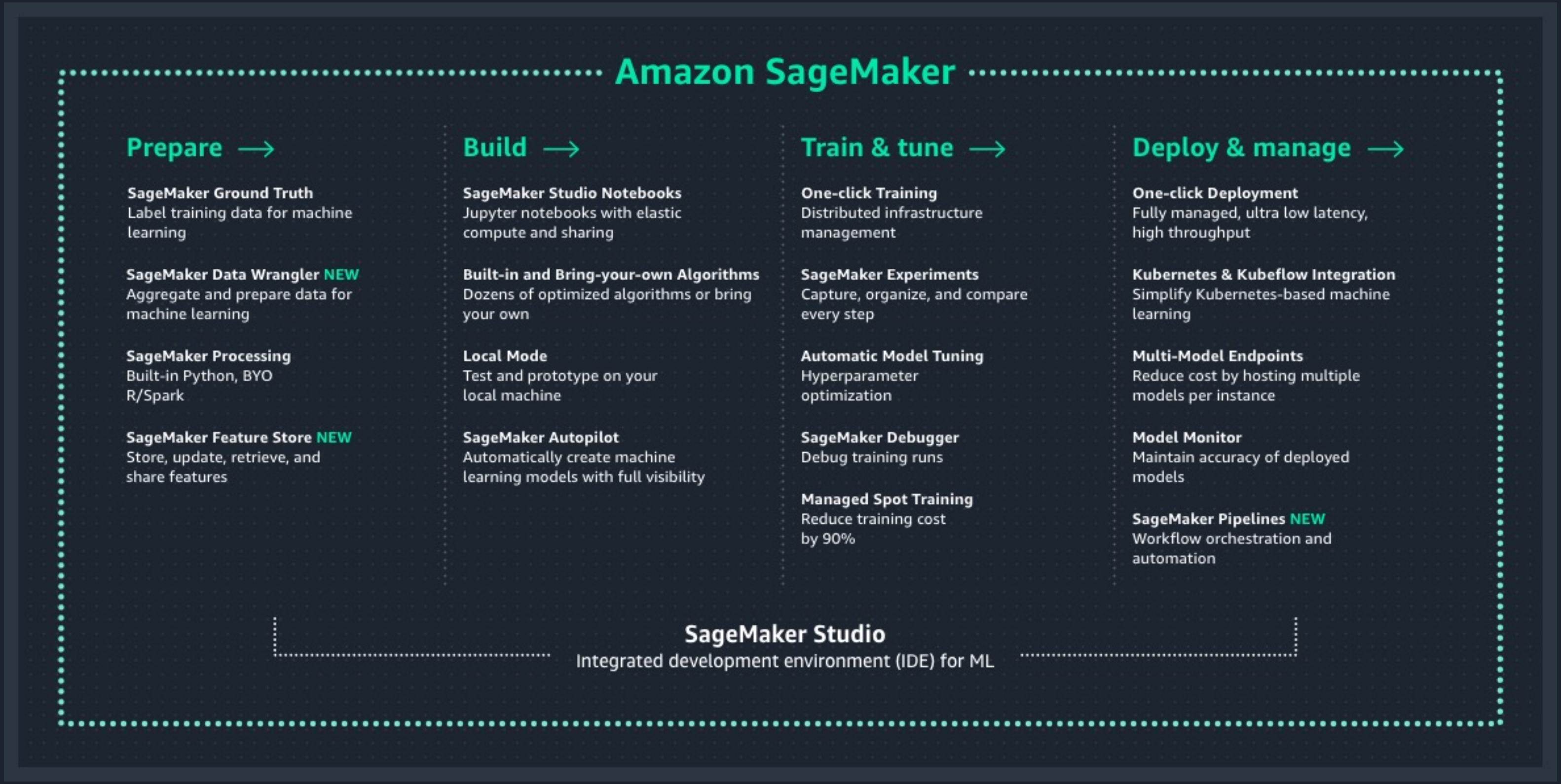
Now, let's compute and plot the anomaly scores from the entire taxi dataset.

Note that the anomaly score spikes where our eyeball-norm method suggests there is an anomalous data point as well as in some places where our eyeballs are not as accurate.

Below we print and plot any data points with scores greater than 3 standard deviations (approx 99.9th percentile) from the mean score.

The following is a list of known anomalous events which occurred in New York City within this timeframe.

Amazon SageMaker Overview



Amazon SageMaker

Prepare →

SageMaker Ground Truth
Label training data for machine learning

SageMaker Data Wrangler NEW
Aggregate and prepare data for machine learning

SageMaker Processing
Built-in Python, BYO R/Spark

SageMaker Feature Store NEW
Store, update, retrieve, and share features

Build →

SageMaker Studio Notebooks
Jupyter notebooks with elastic compute and sharing

Built-in and Bring-your-own Algorithms
Dozens of optimized algorithms or bring your own

Local Mode
Test and prototype on your local machine

SageMaker Autopilot
Automatically create machine learning models with full visibility

Train & tune →

One-click Training
Distributed infrastructure management

SageMaker Experiments
Capture, organize, and compare every step

Automatic Model Tuning
Hyperparameter optimization

SageMaker Debugger
Debug training runs

Managed Spot Training
Reduce training cost by 90%

Deploy & manage →

One-click Deployment
Fully managed, ultra low latency, high throughput

Kubernetes & Kubeflow Integration
Simplify Kubernetes-based machine learning

Multi-Model Endpoints
Reduce cost by hosting multiple models per instance

Model Monitor
Maintain accuracy of deployed models

SageMaker Pipelines NEW
Workflow orchestration and automation

SageMaker Studio
Integrated development environment (IDE) for ML

Enabling ML for customers

What did our customers want?

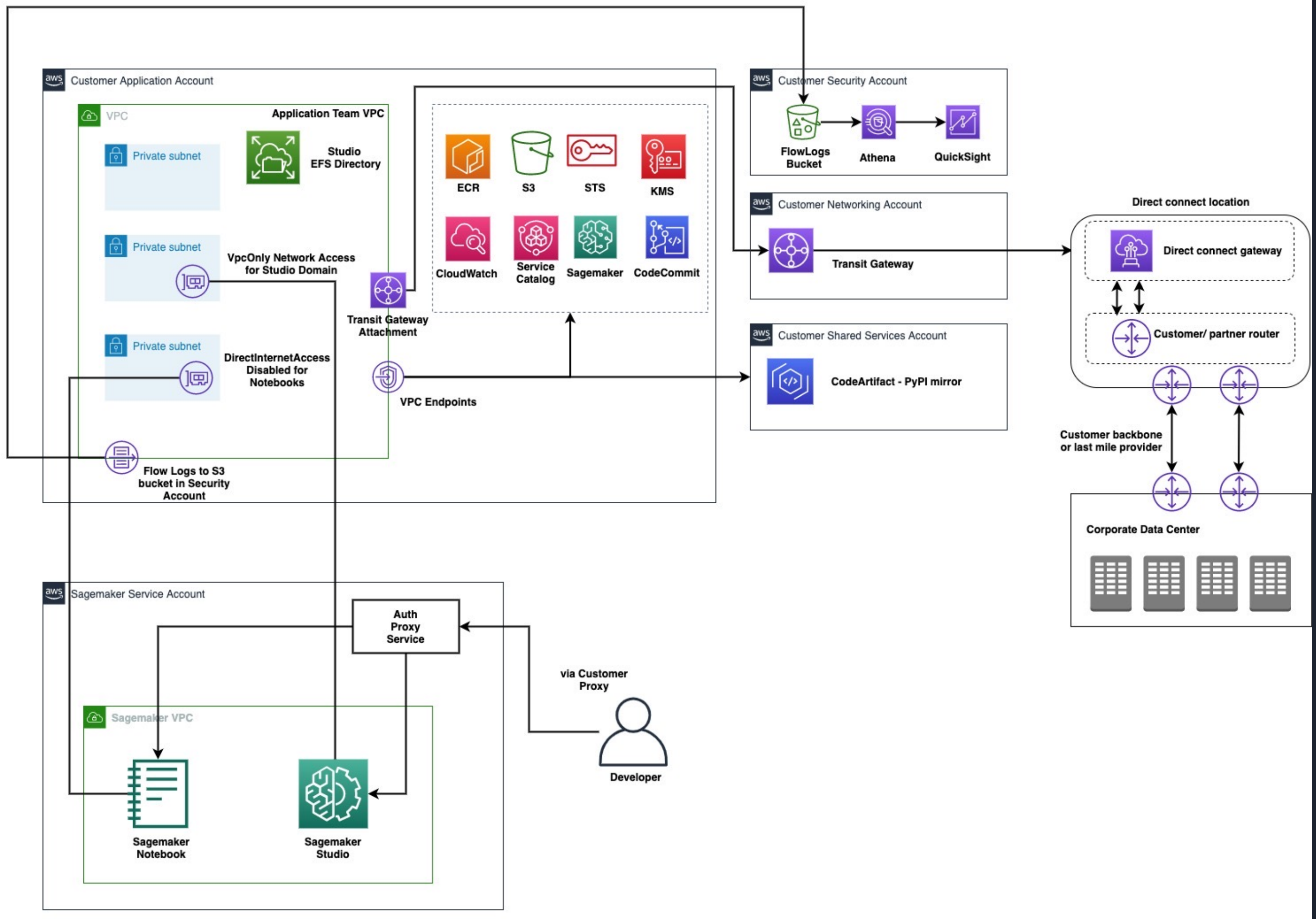
Customers asked for a solution that would enable business data scientists to deliver secure machine learning-based solutions that are trained on highly sensitive company and customer data.

What are the requirements?

- No internet connectivity in AWS accounts
- Self-service model for provisioning AWS ML resources
- Centralised governance and guardrails for the infrastructure
- Observability of the solution

Target Architecture

- Multi-account structure leveraging AWS Organizations
- Private VPC network and all traffic should be over VPC endpoints
- PyPI mirror using AWS Code Artifact
- AWS Service Catalog for provisioning resources
- Amazon CloudWatch for Observability
- Transit Gateway for network connectivity to corporate data center


Customer Application Account
VPC
Application Team VPC
Private subnet
Studio EFS Directory
Private subnet
VpcOnly Network Access for Studio Domain
Private subnet
DirectInternetAccess Disabled for Notebooks
Transit Gateway Attachment
VPC Endpoints
Flow Logs to S3 bucket in Security Account
ECR
S3
STS
KMS
CloudWatch
Service Catalog
Sagemaker
CodeCommit
Customer Security Account
FlowLogs Bucket
Athena
QuickSight
Customer Networking Account
Transit Gateway
Customer Shared Services Account
CodeArtifact - PyPI mirror
Direct connect location
Direct connect gateway
Customer/ partner router
Customer backbone or last mile provider
Corporate Data Center
Sagemaker Service Account
Auth Proxy Service
Sagemaker VPC
Sagemaker Notebook
Sagemaker Studio
via Customer Proxy
Developer

Simplifying provisioning using AWS Service Catalog

Security

Curation
Compliance
Standardisation

Speed

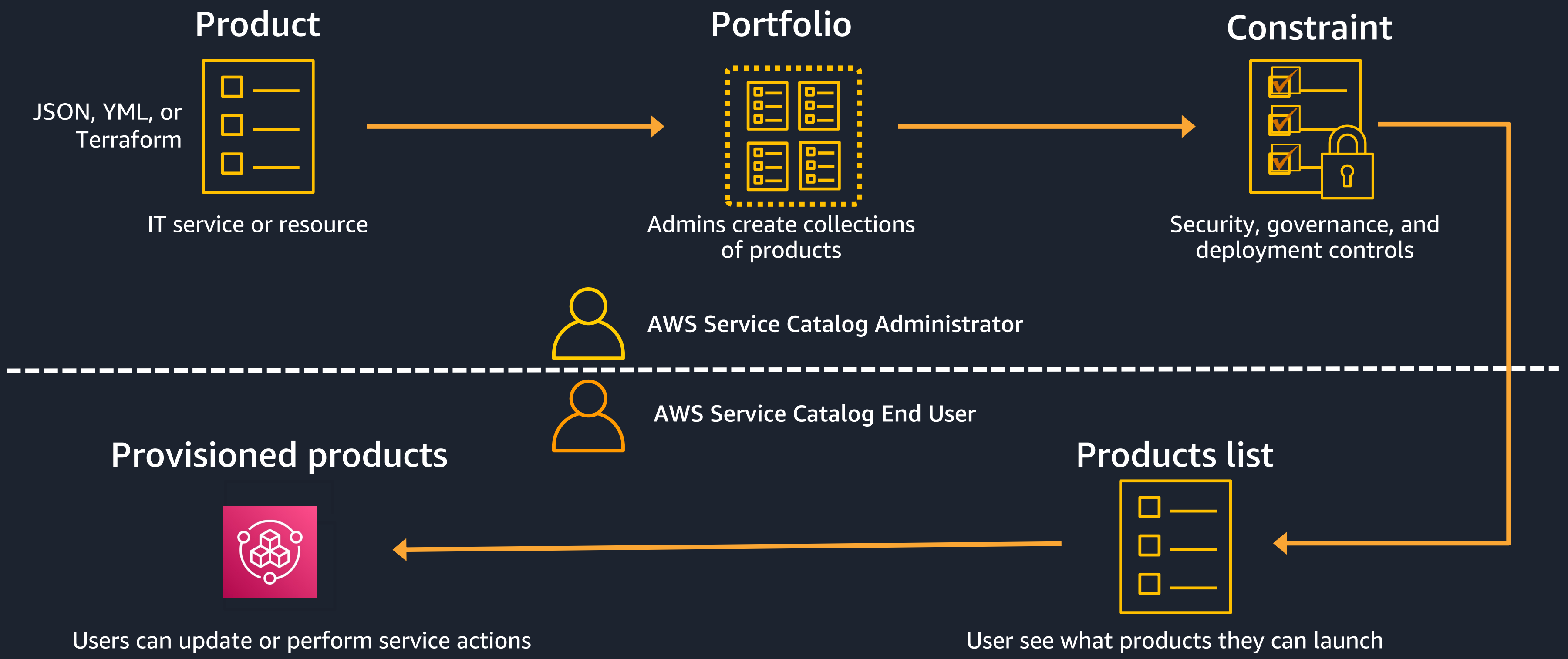
Agility
Self-Service
Time to market

Organisations

Service catalogs enable organisations to deploy and manage infrastructure and applications that reflect the organisation’s security and operational policies

End Users

A few terms to note



Product

JSON, YML, or
Terraform

IT service or resource

Portfolio

Admins create collections
of products

Constraint

Security, governance, and
deployment controls

AWS Service Catalog Administrator

AWS Service Catalog End User

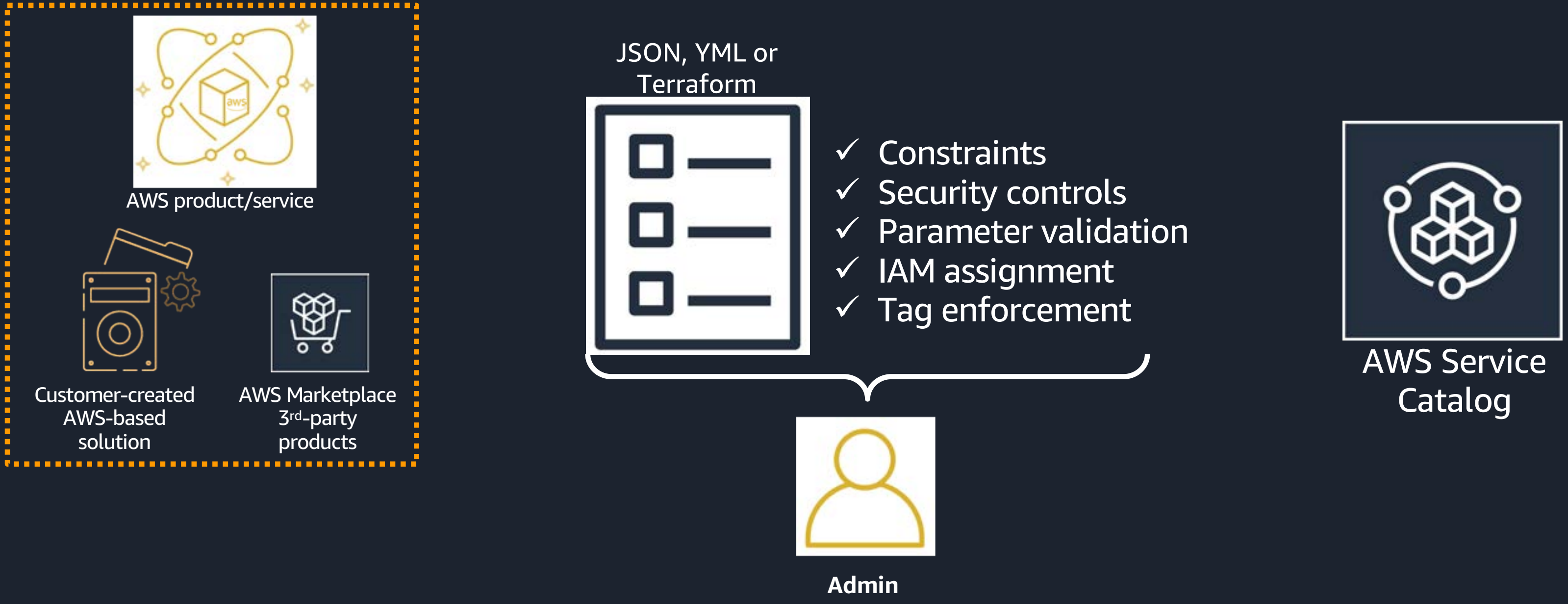
Provisioned products

Users can update or perform service actions

Products list

User see what products they can launch

Self-service with preconfigured compliance



AWS product/service

Customer-created AWS-based solution

AWS Marketplace 3rd-party products

JSON, YML or Terraform

- ✓ Constraints
- ✓ Security controls
- ✓ Parameter validation
- ✓ IAM assignment
- ✓ Tag enforcement

Admin

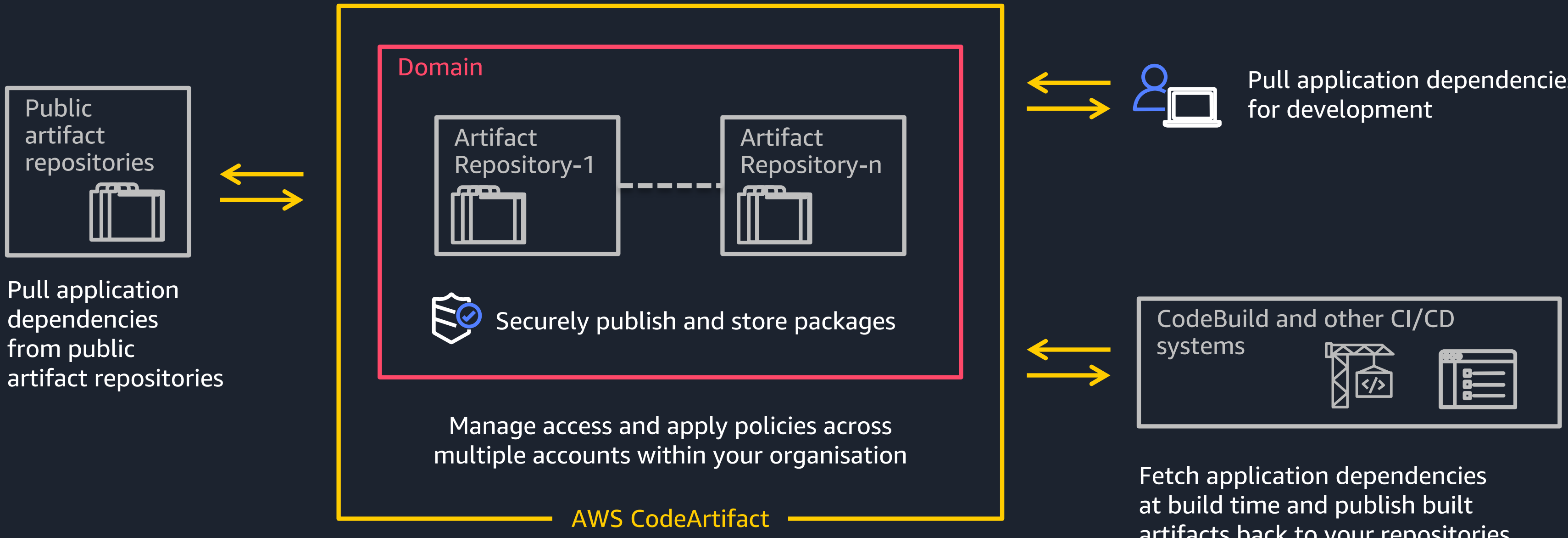
AWS Service Catalog

Standardises best practices

What is AWS CodeArtifact?

- Fully-managed artifact repository service
- Supports NPM, Maven, Python, NuGet package formats
- Currently works with Maven, Gradle, npm, yarn, twine, and pip
- Pay as you go, no upfront license fees

AWS CodeArtifact overview



Public artifact repositories

Pull application dependencies from public artifact repositories

Domain

Artifact Repository-1

Artifact Repository-n

Securely publish and store packages

Manage access and apply policies across multiple accounts within your organisation

AWS CodeArtifact

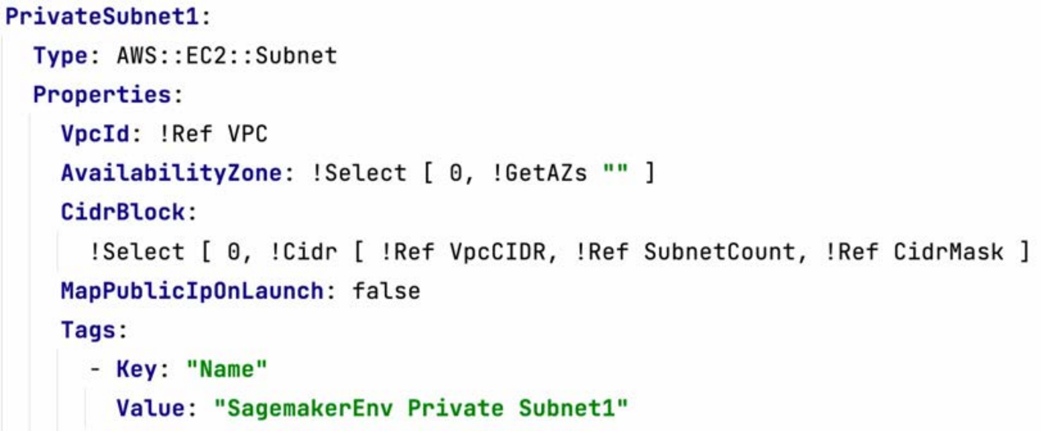
Pull application dependencies for development

CodeBuild and other CI/CD systems

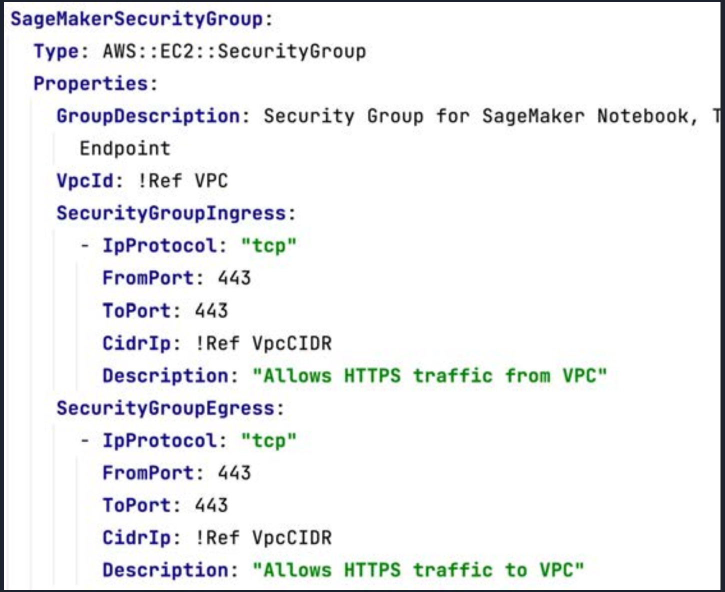
Fetch application dependencies at build time and publish built artifacts back to your repositories

Build infrastructure using AWS CloudFormation

Private VPC networking

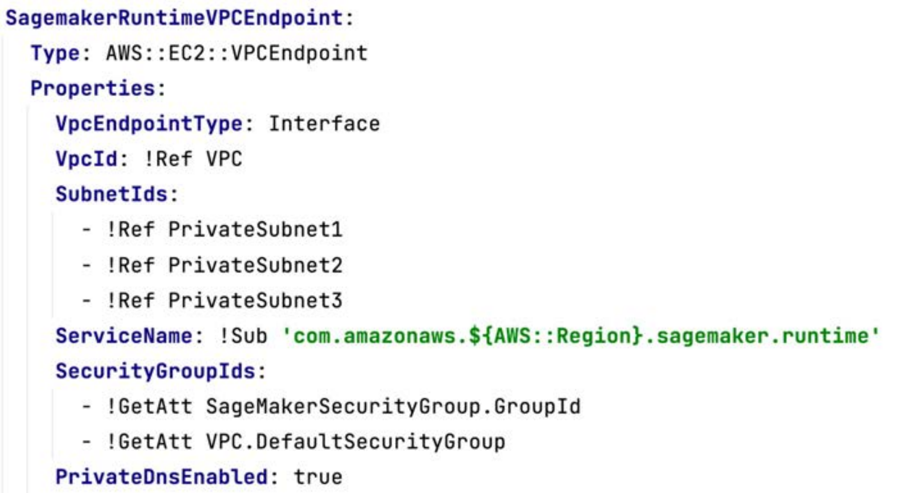


```yaml
PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [ 0, !GetAZs "" ]
    CidrBlock:
      !Select [ 0, !Cidr [ !Ref VpcCIDR, !Ref SubnetCount, !Ref CidrMask ]
    MapPublicIpOnLaunch: false
    Tags:
      - Key: "Name"
        Value: "SagemakerEnv Private Subnet1"
```

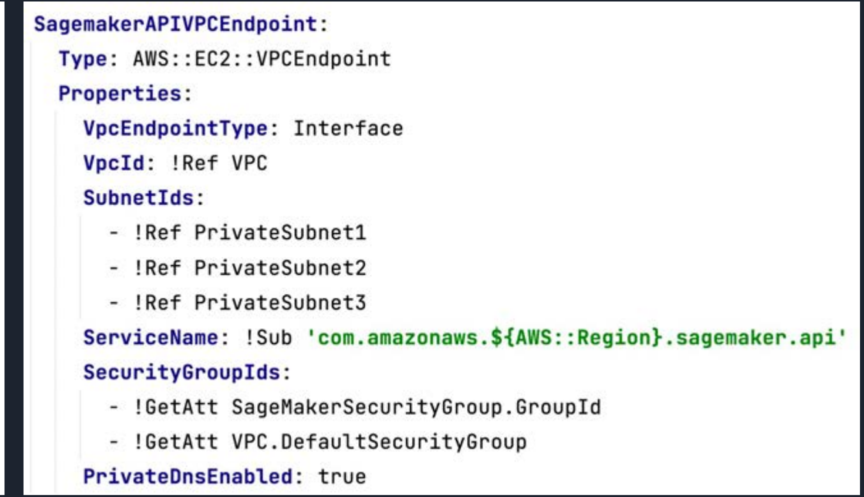


```yaml
SageMakerSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security Group for SageMaker Notebook, T
      Endpoint
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: "tcp"
        FromPort: 443
        ToPort: 443
        CidrIp: !Ref VpcCIDR
        Description: "Allows HTTPS traffic from VPC"
    SecurityGroupEgress:
      - IpProtocol: "tcp"
        FromPort: 443
        ToPort: 443
        CidrIp: !Ref VpcCIDR
        Description: "Allows HTTPS traffic to VPC"
```

Enable VPC endpoints

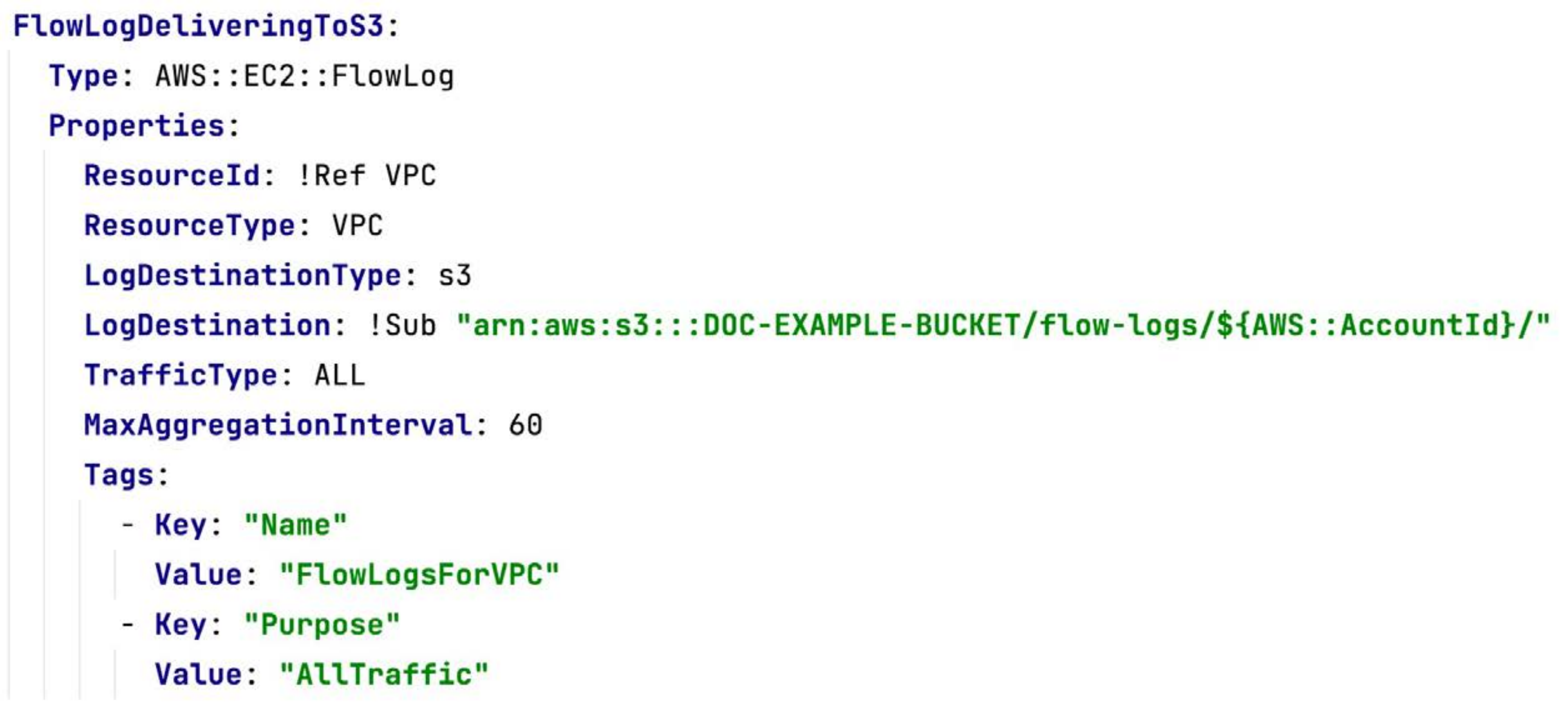


```
SagemakerRuntimeVPCEndpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    VpcEndpointType: Interface
    VpcId: !Ref VPC
    SubnetIds:
      - !Ref PrivateSubnet1
      - !Ref PrivateSubnet2
      - !Ref PrivateSubnet3
    ServiceName: !Sub 'com.amazonaws.${AWS::Region}.sagemaker.runtime'
    SecurityGroupIds:
      - !GetAtt SageMakerSecurityGroup.GroupId
      - !GetAtt VPC.DefaultSecurityGroup
    PrivateDnsEnabled: true
```



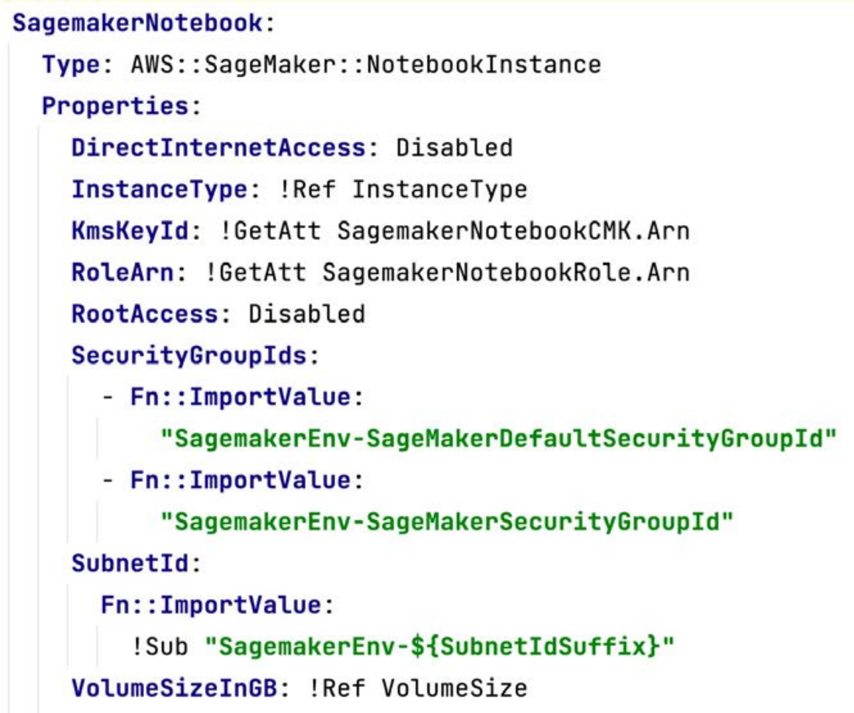
```
SagemakerAPIVPCEndpoint:
  Type: AWS::EC2::VPCEndpoint
  Properties:
    VpcEndpointType: Interface
    VpcId: !Ref VPC
    SubnetIds:
      - !Ref PrivateSubnet1
      - !Ref PrivateSubnet2
      - !Ref PrivateSubnet3
    ServiceName: !Sub 'com.amazonaws.${AWS::Region}.sagemaker.api'
    SecurityGroupIds:
      - !GetAtt SageMakerSecurityGroup.GroupId
      - !GetAtt VPC.DefaultSecurityGroup
    PrivateDnsEnabled: true
```

Enable VPC flow logs

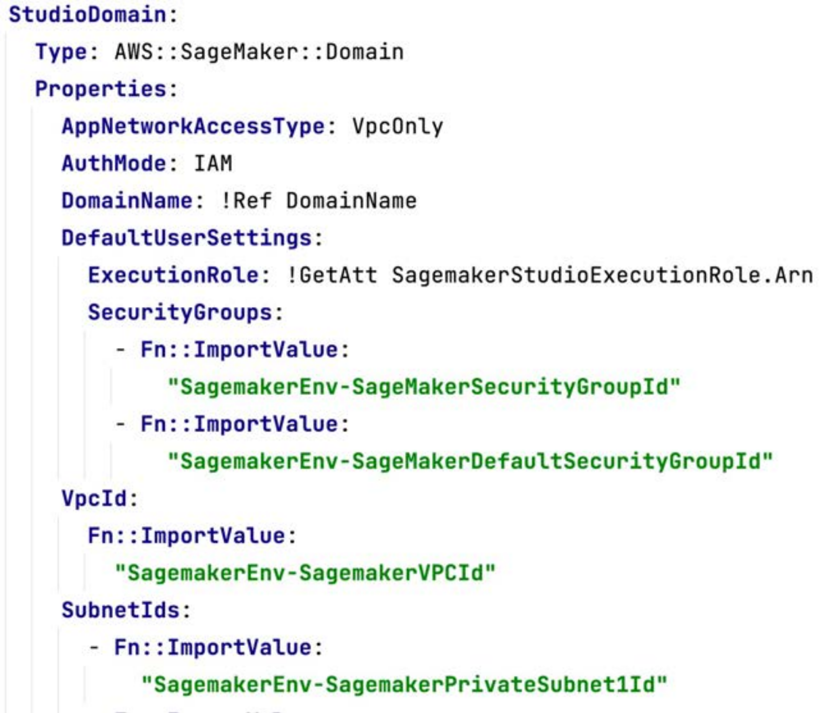


```
FlowLogDeliveringToS3:
  Type: AWS::EC2::FlowLog
  Properties:
    ResourceId: !Ref VPC
    ResourceType: VPC
    LogDestinationType: s3
    LogDestination: !Sub "arn:aws:s3:::DOC-EXAMPLE-BUCKET/flow-logs/${AWS::AccountId}/"
    TrafficType: ALL
    MaxAggregationInterval: 60
    Tags:
      - Key: "Name"
        Value: "FlowLogsForVPC"
      - Key: "Purpose"
        Value: "AllTraffic"
```

Amazon Sagemaker studio and notebook

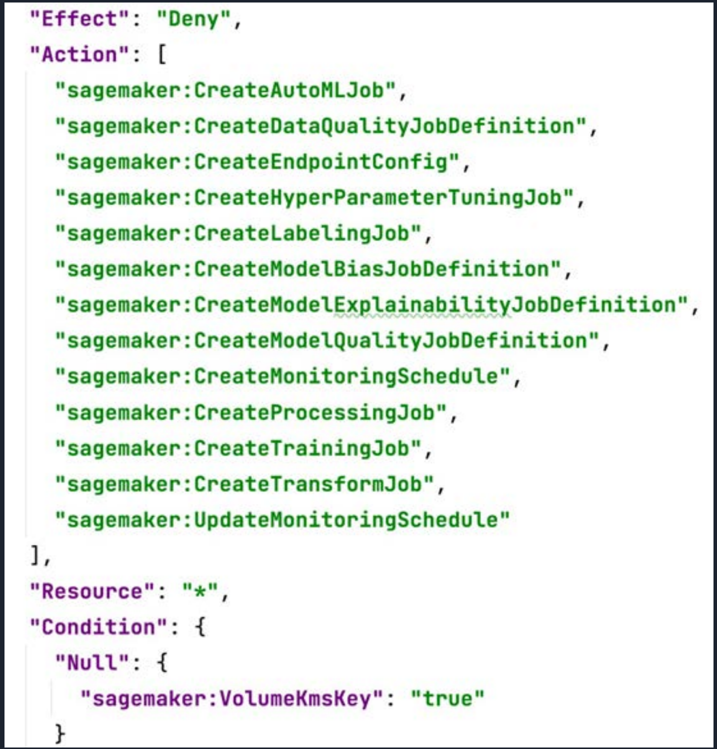


```yaml
SagemakerNotebook:
  Type: AWS::SageMaker::NotebookInstance
  Properties:
    DirectInternetAccess: Disabled
    InstanceType: !Ref InstanceType
    KmsKeyId: !GetAtt SagemakerNotebookCMK.Arn
    RoleArn: !GetAtt SagemakerNotebookRole.Arn
    RootAccess: Disabled
    SecurityGroupIds:
      - Fn::ImportValue:
          "SagemakerEnv-SageMakerDefaultSecurityGroupId"
      - Fn::ImportValue:
          "SagemakerEnv-SageMakerSecurityGroupId"
    SubnetId:
      Fn::ImportValue:
        !Sub "SagemakerEnv-${SubnetIdSuffix}"
    VolumeSizeInGB: !Ref VolumeSize
```

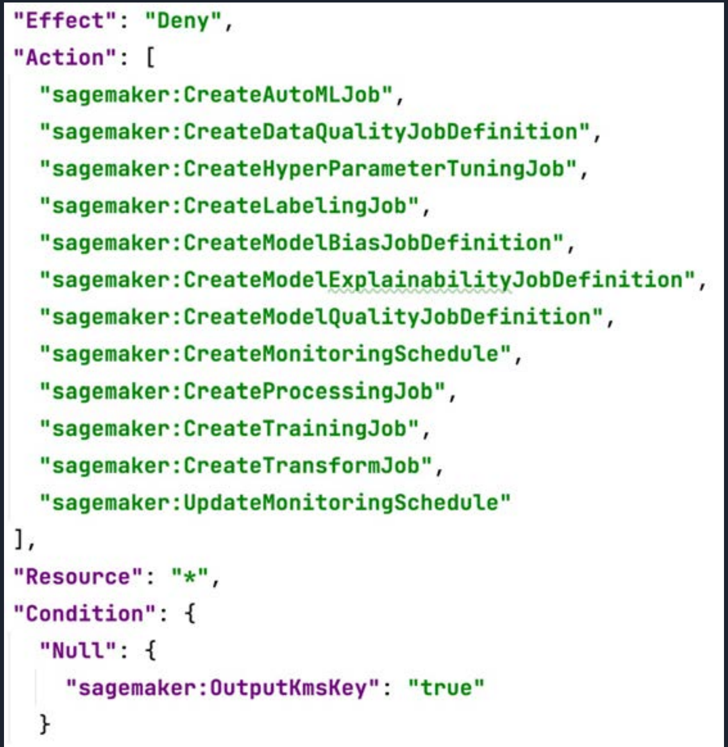


```yaml
StudioDomain:
  Type: AWS::SageMaker::Domain
  Properties:
    AppNetworkAccessType: VpcOnly
    AuthMode: IAM
    DomainName: !Ref DomainName
    DefaultUserSettings:
      ExecutionRole: !GetAtt SagemakerStudioExecutionRole.Arn
      SecurityGroups:
        - Fn::ImportValue:
            "SagemakerEnv-SageMakerSecurityGroupId"
        - Fn::ImportValue:
            "SagemakerEnv-SageMakerDefaultSecurityGroupId"
    VpcId:
      Fn::ImportValue:
        "SagemakerEnv-SagemakerVPCId"
    SubnetIds:
      - Fn::ImportValue:
          "SagemakerEnv-SagemakerPrivateSubnet1Id"
```

Service control policies for data

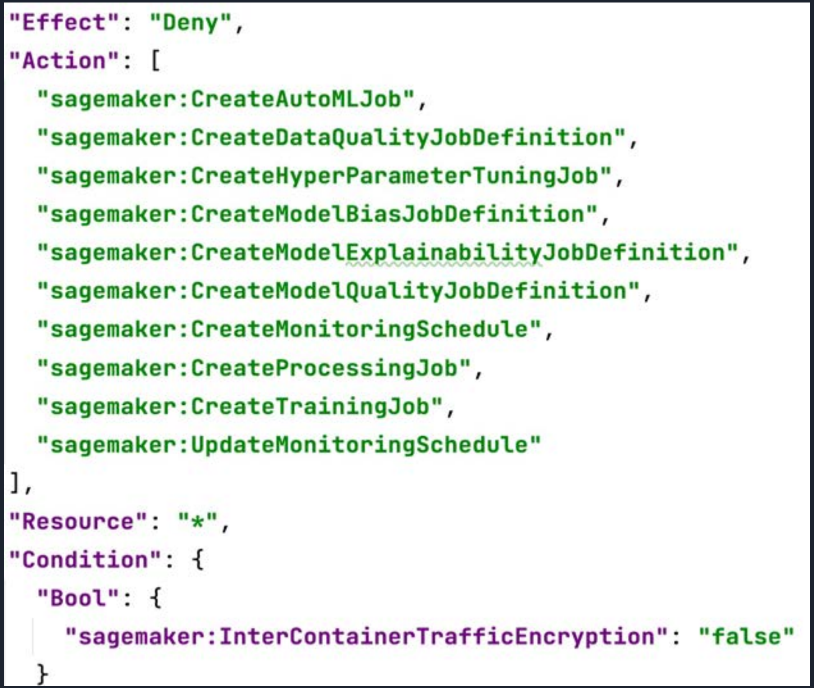


```
"Effect": "Deny",
"Action": [
  "sagemaker:CreateAutoMLJob",
  "sagemaker:CreateDataQualityJobDefinition",
  "sagemaker:CreateEndpointConfig",
  "sagemaker:CreateHyperParameterTuningJob",
  "sagemaker:CreateLabelingJob",
  "sagemaker:CreateModelBiasJobDefinition",
  "sagemaker:CreateModelExplainabilityJobDefinition",
  "sagemaker:CreateModelQualityJobDefinition",
  "sagemaker:CreateMonitoringSchedule",
  "sagemaker:CreateProcessingJob",
  "sagemaker:CreateTrainingJob",
  "sagemaker:CreateTransformJob",
  "sagemaker:UpdateMonitoringSchedule"
],
"Resource": "*",
"Condition": {
  "Null": {
    "sagemaker:VolumeKmsKey": "true"
  }
```

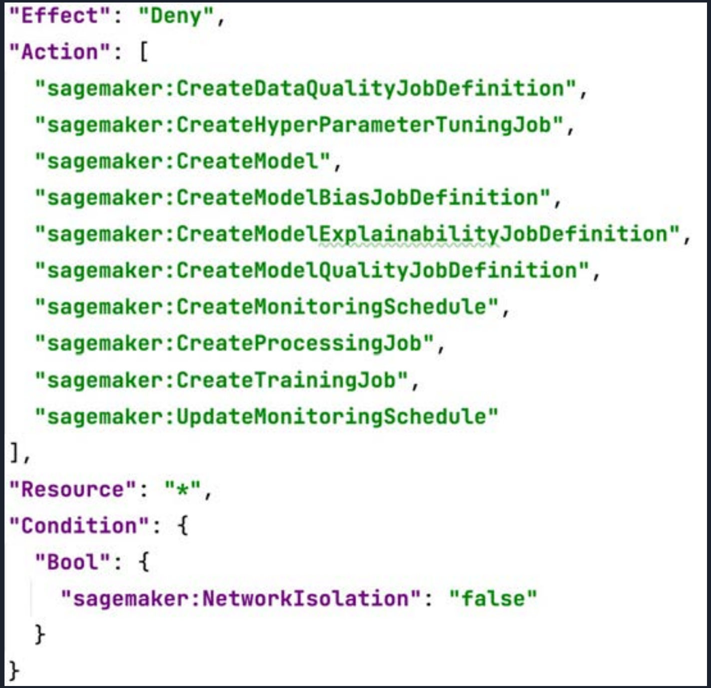


```
"Effect": "Deny",
"Action": [
  "sagemaker:CreateAutoMLJob",
  "sagemaker:CreateDataQualityJobDefinition",
  "sagemaker:CreateHyperParameterTuningJob",
  "sagemaker:CreateLabelingJob",
  "sagemaker:CreateModelBiasJobDefinition",
  "sagemaker:CreateModelExplainabilityJobDefinition",
  "sagemaker:CreateModelQualityJobDefinition",
  "sagemaker:CreateMonitoringSchedule",
  "sagemaker:CreateProcessingJob",
  "sagemaker:CreateTrainingJob",
  "sagemaker:CreateTransformJob",
  "sagemaker:UpdateMonitoringSchedule"
],
"Resource": "*",
"Condition": {
  "Null": {
    "sagemaker:OutputKmsKey": "true"
  }
```

Service control policies for traffic and network



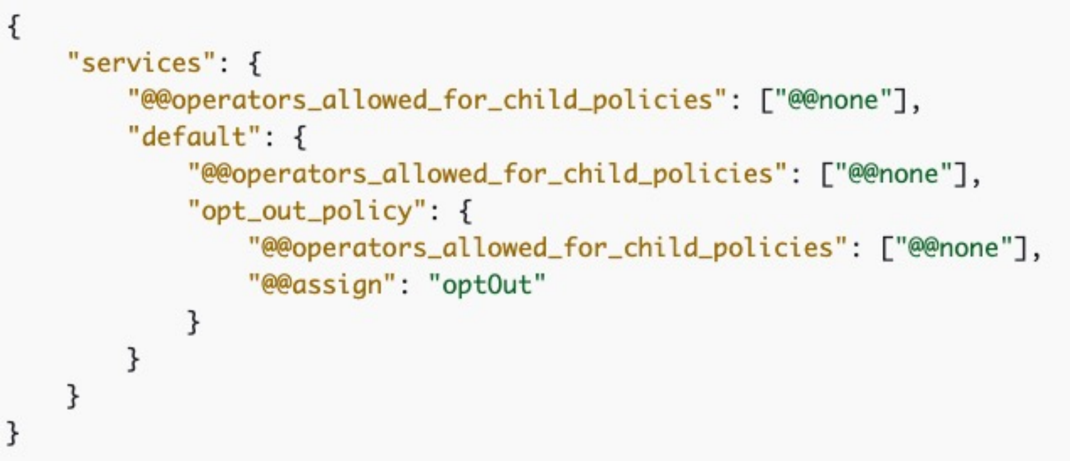
```
"Effect": "Deny",
"Action": [
  "sagemaker:CreateAutoMLJob",
  "sagemaker:CreateDataQualityJobDefinition",
  "sagemaker:CreateHyperParameterTuningJob",
  "sagemaker:CreateModelBiasJobDefinition",
  "sagemaker:CreateModelExplainabilityJobDefinition",
  "sagemaker:CreateModelQualityJobDefinition",
  "sagemaker:CreateMonitoringSchedule",
  "sagemaker:CreateProcessingJob",
  "sagemaker:CreateTrainingJob",
  "sagemaker:UpdateMonitoringSchedule"
],
"Resource": "*",
"Condition": {
  "Bool": {
    "sagemaker:InterContainerTrafficEncryption": "false"
  }
```



```
"Effect": "Deny",
"Action": [
  "sagemaker:CreateDataQualityJobDefinition",
  "sagemaker:CreateHyperParameterTuningJob",
  "sagemaker:CreateModel",
  "sagemaker:CreateModelBiasJobDefinition",
  "sagemaker:CreateModelExplainabilityJobDefinition",
  "sagemaker:CreateModelQualityJobDefinition",
  "sagemaker:CreateMonitoringSchedule",
  "sagemaker:CreateProcessingJob",
  "sagemaker:CreateTrainingJob",
  "sagemaker:UpdateMonitoringSchedule"
],
"Resource": "*",
"Condition": {
  "Bool": {
    "sagemaker:NetworkIsolation": "false"
  }
}
```

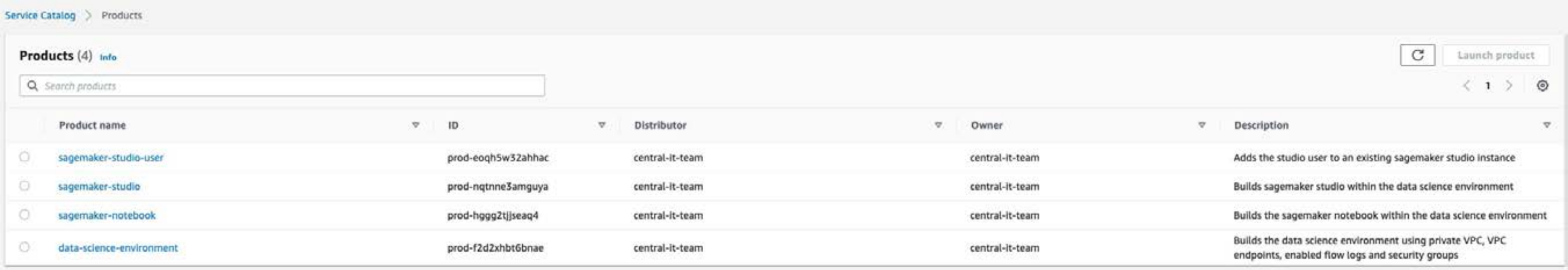
AI services opt-out policies

Certain AWS artificial intelligence (AI) services, may store and use customer content processed by those services for the development and continuous improvement of Amazon AI services and technologies. As an AWS customer, you can choose to opt out of having your content stored or used for service improvements.



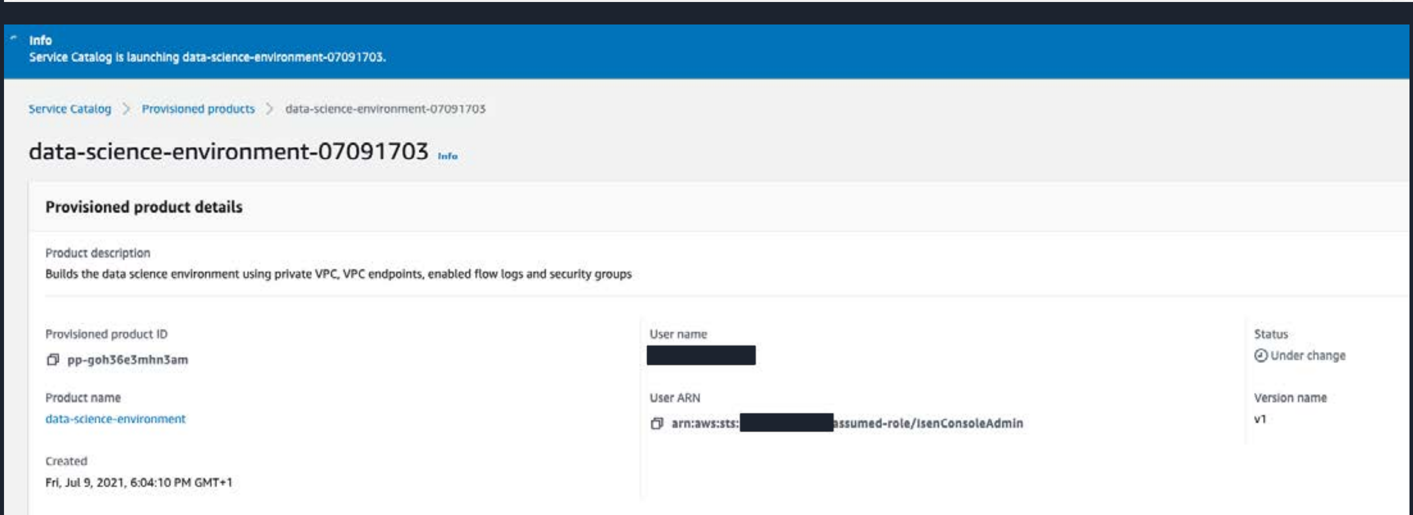
```
{
    "services": {
        "@@operators_allowed_for_child_policies": ["@@none"],
        "default": {
            "@@operators_allowed_for_child_policies": ["@@none"],
            "opt_out_policy": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "@@assign": "optOut"
            }
        }
    }
}
```

Provision the products using AWS Service Catalog



Products (4) Info

	Product name	ID	Distributor	Owner	Description
○	sagemaker-studio-user	prod-eoqh5w32ahhac	central-it-team	central-it-team	Adds the studio user to an existing sagemaker studio instance
○	sagemaker-studio	prod-nqtnne3amguya	central-it-team	central-it-team	Builds sagemaker studio within the data science environment
○	sagemaker-notebook	prod-hggg2tjjseaq4	central-it-team	central-it-team	Builds the sagemaker notebook within the data science environment
○	data-science-environment	prod-f2d2xhbt6bnae	central-it-team	central-it-team	Builds the data science environment using private VPC, VPC endpoints, enabled flow logs and security groups



Info
Service Catalog is launching data-science-environment-07091703.

Service Catalog > Provisioned products > data-science-environment-07091703

data-science-environment-07091703 Info

Provisioned product details

Product description
Builds the data science environment using private VPC, VPC endpoints, enabled flow logs and security groups

Provisioned product ID
pp-goh36e3mhn3am

User name
[redacted]

Status
Under change

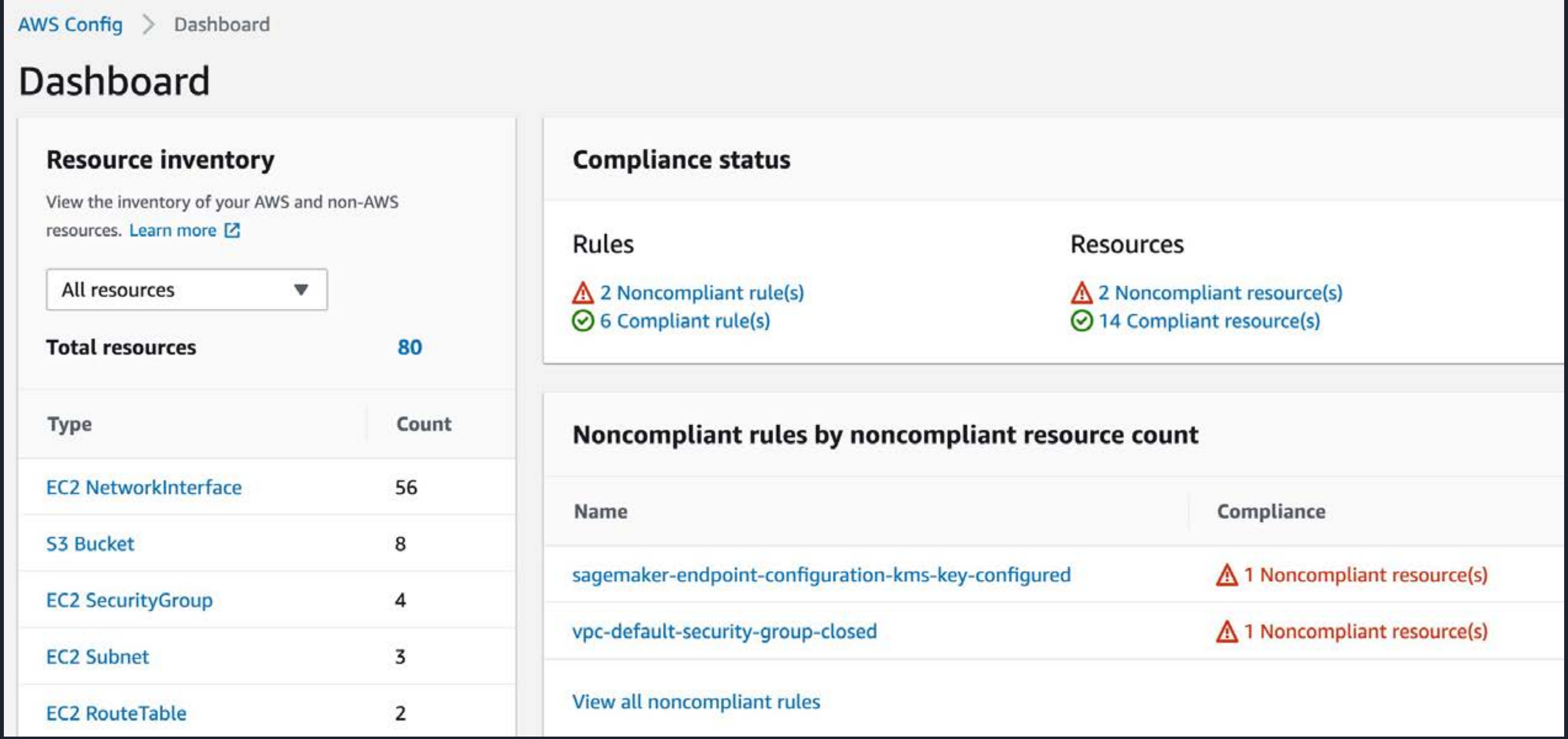
Product name
data-science-environment

User ARN
arn:aws:sts:[redacted]assumed-role/IsenConsoleAdmin

Version name
v1

Created
Fri, Jul 9, 2021, 6:04:10 PM GMT+1

AWS Config – detective controls



AWS Config > Dashboard

Dashboard

Resource inventory

View the inventory of your AWS and non-AWS resources. Learn more

All resources ▼

Total resources 80

Type	Count
EC2 NetworkInterface	56
S3 Bucket	8
EC2 SecurityGroup	4
EC2 Subnet	3
EC2 RouteTable	2

Compliance status

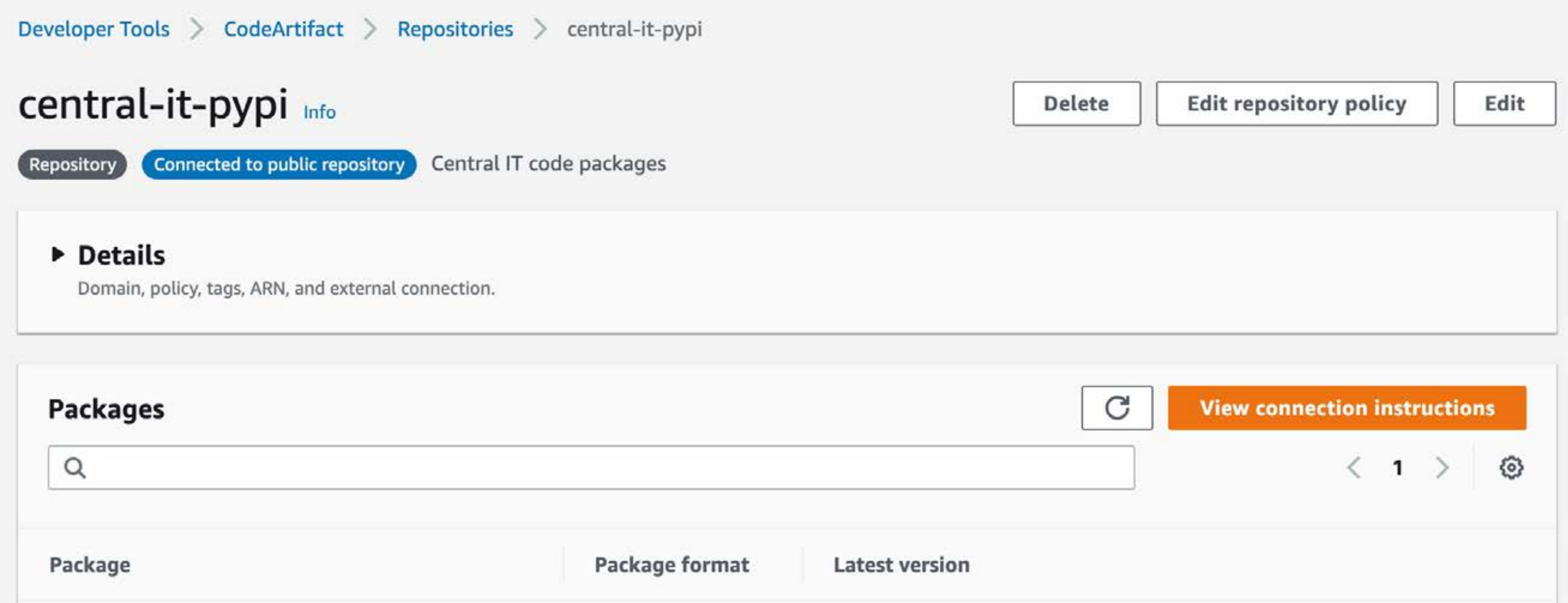
Rules	Resources
⚠ 2 Noncompliant rule(s)	⚠ 2 Noncompliant resource(s)
✓ 6 Compliant rule(s)	✓ 14 Compliant resource(s)

Noncompliant rules by noncompliant resource count

Name	Compliance
sagemaker-endpoint-configuration-kms-key-configured	⚠ 1 Noncompliant resource(s)
vpc-default-security-group-closed	⚠ 1 Noncompliant resource(s)

View all noncompliant rules

Centralised governance for pip dependencies


Developer Tools > CodeArtifact > Repositories > central-it-pypi
Delete
Edit repository policy
Edit
View connection instructions
1

central-it-pypi Info

Repository | Connected to public repository | Central IT code packages

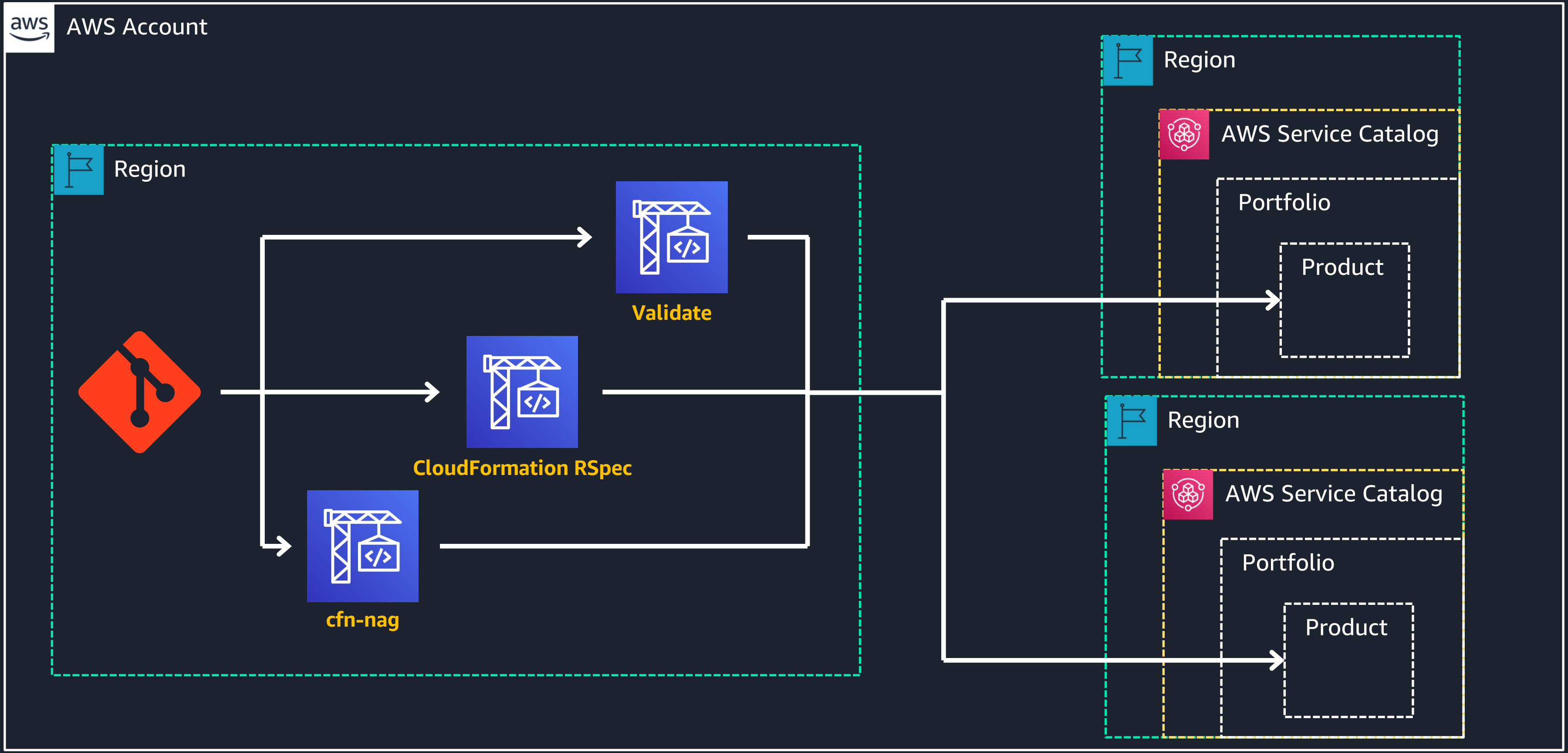
▶ Details

Domain, policy, tags, ARN, and external connection.

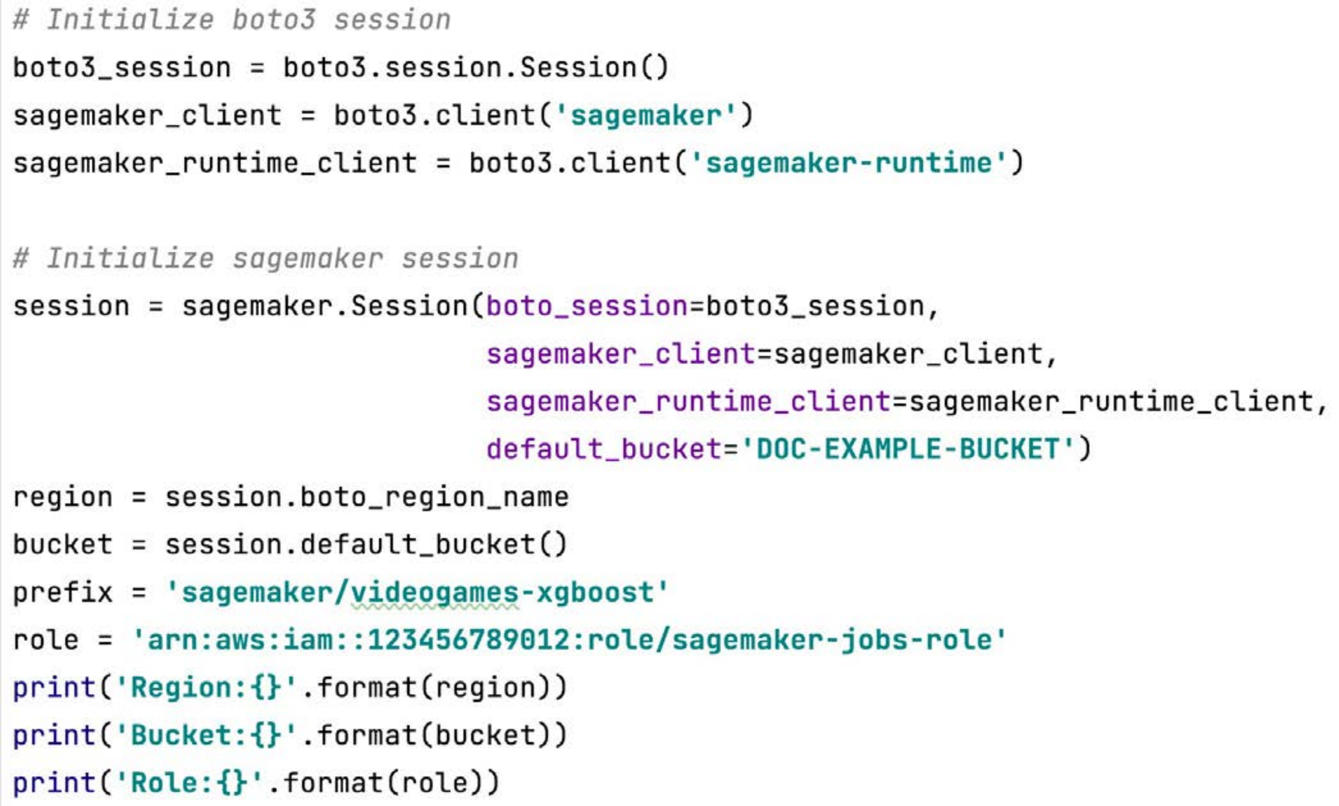
Packages

Package	Package format	Latest version

AWS CodePipelines using AWS Service Catalog tools



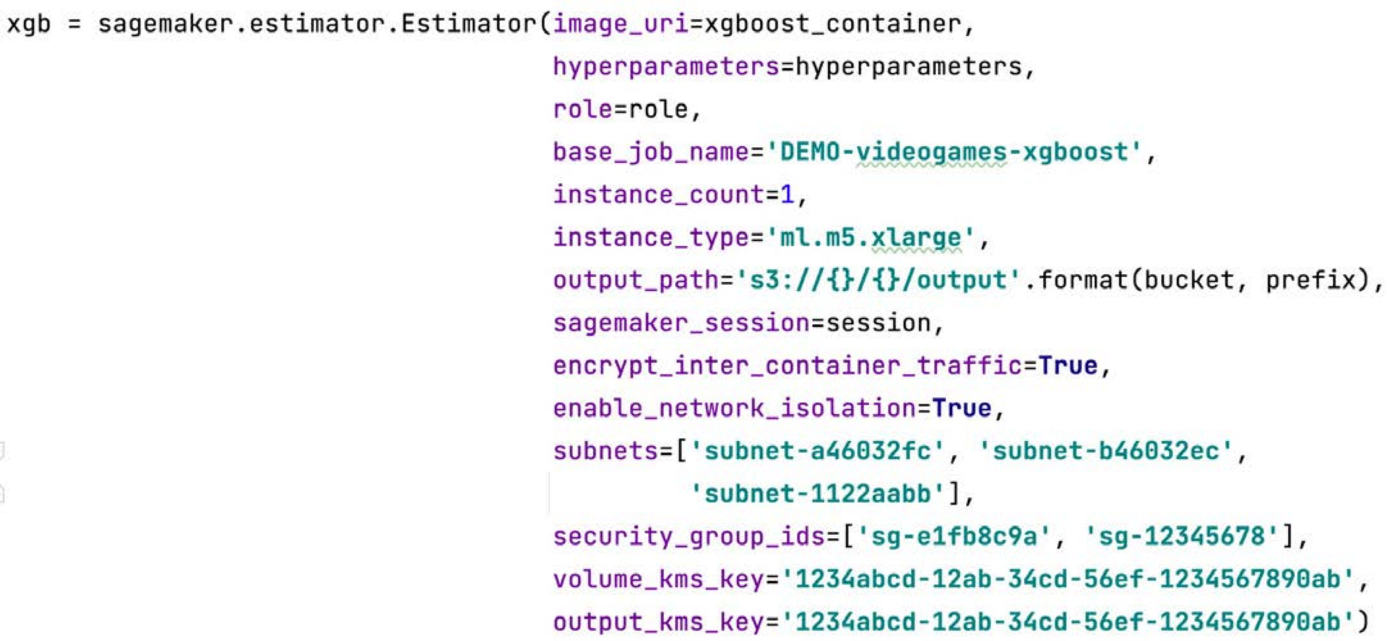
Writing a Jupyter notebook



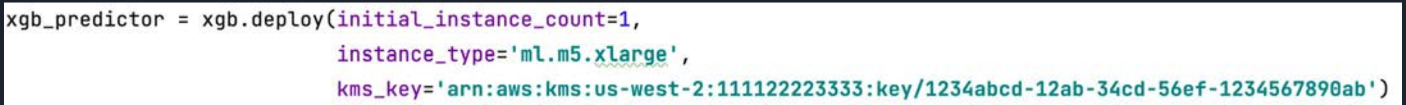
```
# Initialize boto3 session
boto3_session = boto3.session.Session()
sagemaker_client = boto3.client('sagemaker')
sagemaker_runtime_client = boto3.client('sagemaker-runtime')

# Initialize sagemaker session
session = sagemaker.Session(boto_session=boto3_session,
                            sagemaker_client=sagemaker_client,
                            sagemaker_runtime_client=sagemaker_runtime_client,
                            default_bucket='DOC-EXAMPLE-BUCKET')
region = session.boto_region_name
bucket = session.default_bucket()
prefix = 'sagemaker/videogames-xgboost'
role = 'arn:aws:iam::123456789012:role/sagemaker-jobs-role'
print('Region:{}'.format(region))
print('Bucket:{}'.format(bucket))
print('Role:{}'.format(role))
```

Build the estimator, train and deploy model

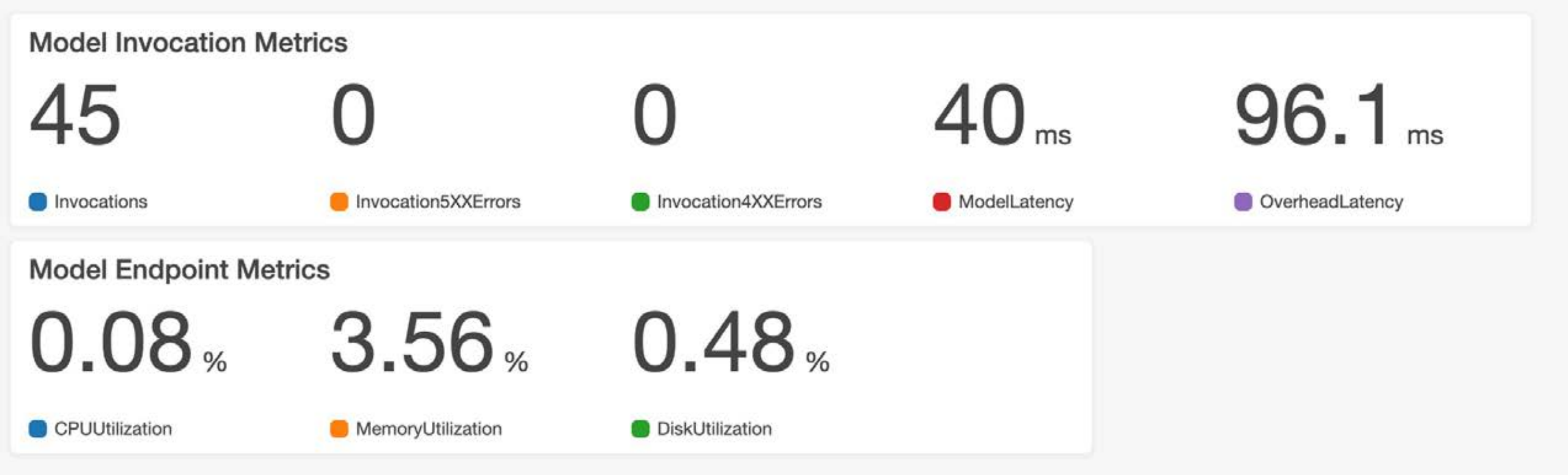


```
xgb = sagemaker.estimator.Estimator(image_uri=xgboost_container,
                                    hyperparameters=hyperparameters,
                                    role=role,
                                    base_job_name='DEMO-videogames-xgboost',
                                    instance_count=1,
                                    instance_type='ml.m5.xlarge',
                                    output_path='s3://{}/{}/output'.format(bucket, prefix),
                                    sagemaker_session=session,
                                    encrypt_inter_container_traffic=True,
                                    enable_network_isolation=True,
                                    subnets=['subnet-a46032fc', 'subnet-b46032ec',
                                             'subnet-1122aabb'],
                                    security_group_ids=['sg-e1fb8c9a', 'sg-12345678'],
                                    volume_kms_key='1234abcd-12ab-34cd-56ef-1234567890ab',
                                    output_kms_key='1234abcd-12ab-34cd-56ef-1234567890ab')
```



```
xgb_predictor = xgb.deploy(initial_instance_count=1,
                           instance_type='ml.m5.xlarge',
                           kms_key='arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab')
```

Monitor the deployed models



Model Invocation Metrics

45

🟦 Invocations

0

🟧 Invocation5XXErrors

0

🟩 Invocation4XXErrors

40 ms

🟥 ModelLatency

96.1 ms

🟪 OverheadLatency

Model Endpoint Metrics

0.08 %

🟦 CPUUtilization

3.56 %

🟧 MemoryUtilization

0.48 %

🟩 DiskUtilization

What did we learn?

- Use the multi-account org structure to improve security and segregation of responsibilities
- Use SCPs and IAM policies to setup the preventative guardrails
- Leverage AWS Config for the detective controls
- Provide application teams autonomy via self-service products with AWS Service Catalog

References

- Service Catalog Tools - https://service-catalog-tools-workshop.com/
- Amazon Sagemaker - https://sagemaker-workshop.com/
- GitHub examples - https://github.com/aws/amazon-sagemaker-examples
- Whitepaper - https://d1.awsstatic.com/whitepapers/machine-learning-in-financial-services-on-aws.pdf

Thank you!