Amazon SageMaker & Algorithms

Mahesh Viswanathan, PhD

Principal Solutions Architect – AI/ML Specialist &

Snr Manager, Specialists Team – AI/ML, Big Data, IoT, Robotics

Amazon Web Services

Agenda

- Amazon AI Services
- Amazon SageMaker Overview
- Algorithms in SageMaker + demo
- SageMaker Autopilot (“AutoML”) + demo
- Using SageMaker for Regression + demo
- SageMaker BYOM, BYOC + demo
- Time series forecasting using Amazon Forecast + demo
- Stretch goal: AWS Step Functions, AWS Lambda, Amazon API Gateway

How we work with customers

- SAs and Specialist SAs
  - Understand your business needs
  - Strategize on high-impact requirements
  - Deliver workshops and training (use cases, new services)
  - Proof-of-concepts
  - Collaboration on architecture, development, MLOps 🔄
- In addition, AWS offers:
  - Prototyping team (6-week engagements)
  - ML Solutions Lab (for multi-week projects)
  - Professional Services (weeks- to months-long projects)

Amazon AI Services

The AWS Machine Learning Stack

Broadest and most complete set of Machine Learning capabilities

AI SERVICES

- **VISION**: Amazon Rekognition
- **SPEECH**: Amazon Polly; Amazon Transcribe *+Medical*
- **TEXT**: Amazon Comprehend *+Medical*; Amazon Translate; Amazon Textract
- **SEARCH**: Amazon Kendra
- **CHATBOTS**: Amazon Lex
- **PERSONALIZATION**: Amazon Personalize
- **FORECASTING**: Amazon Forecast
- **FRAUD**: Amazon Fraud Detector
- **DEVELOPMENT**: Amazon CodeGuru
- **CONTACT CENTERS**: Contact Lens *For Amazon Connect*

ML SERVICES

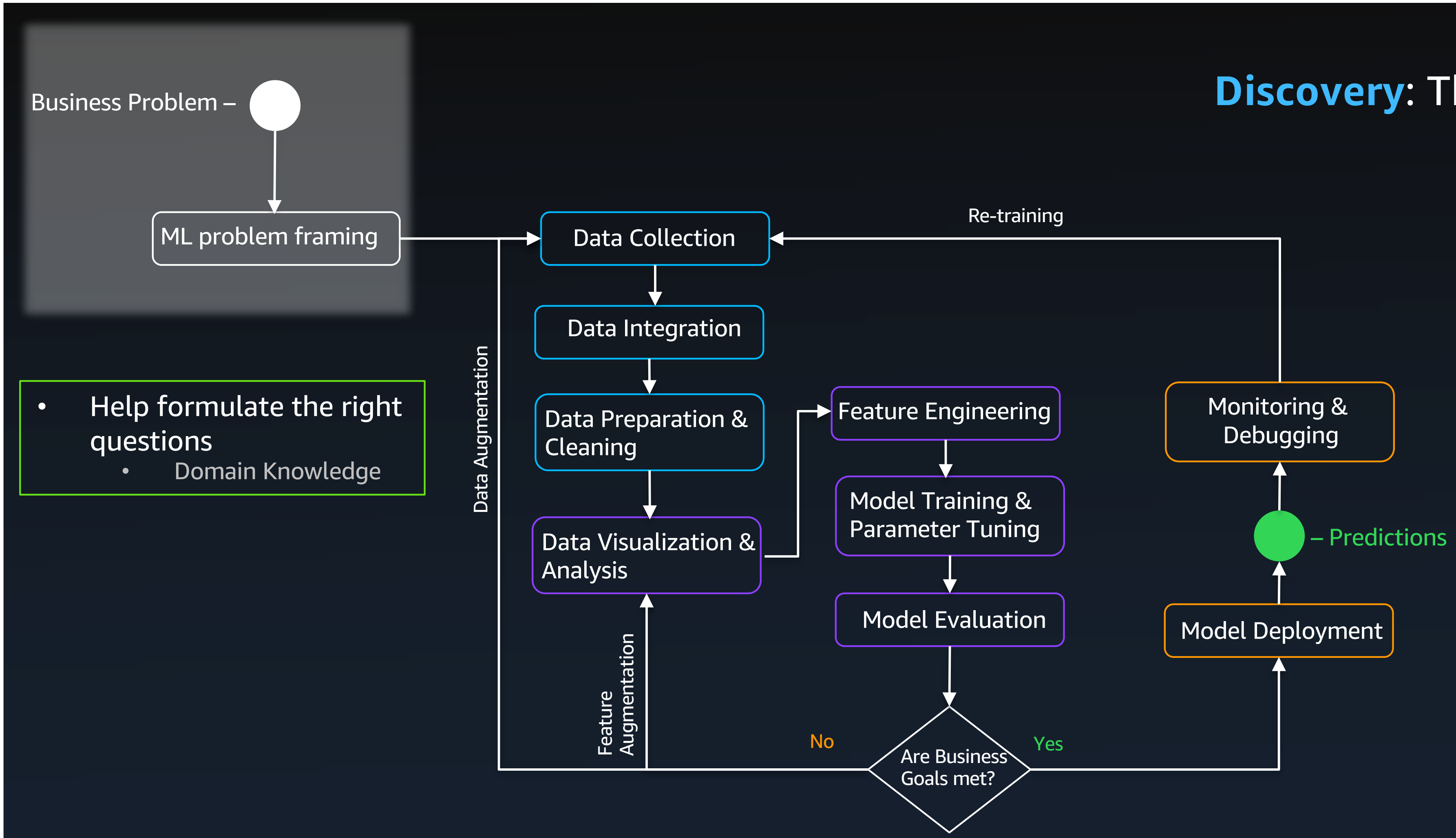
- Amazon SageMaker
- Ground Truth
- AWS Marketplace for ML
- SageMaker Studio IDE
  - Built-in algorithms
  - Notebooks
  - Experiments
  - Processing
  - Model training & tuning
  - Debugger
  - Autopilot
  - Model hosting
  - Model Monitor
- Neo
- Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

- TensorFlow
- mxnet
- PYTORCH
- GLUON
- Keras
- learn
- HOROVOD
- DeepGraphLibrary
- Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

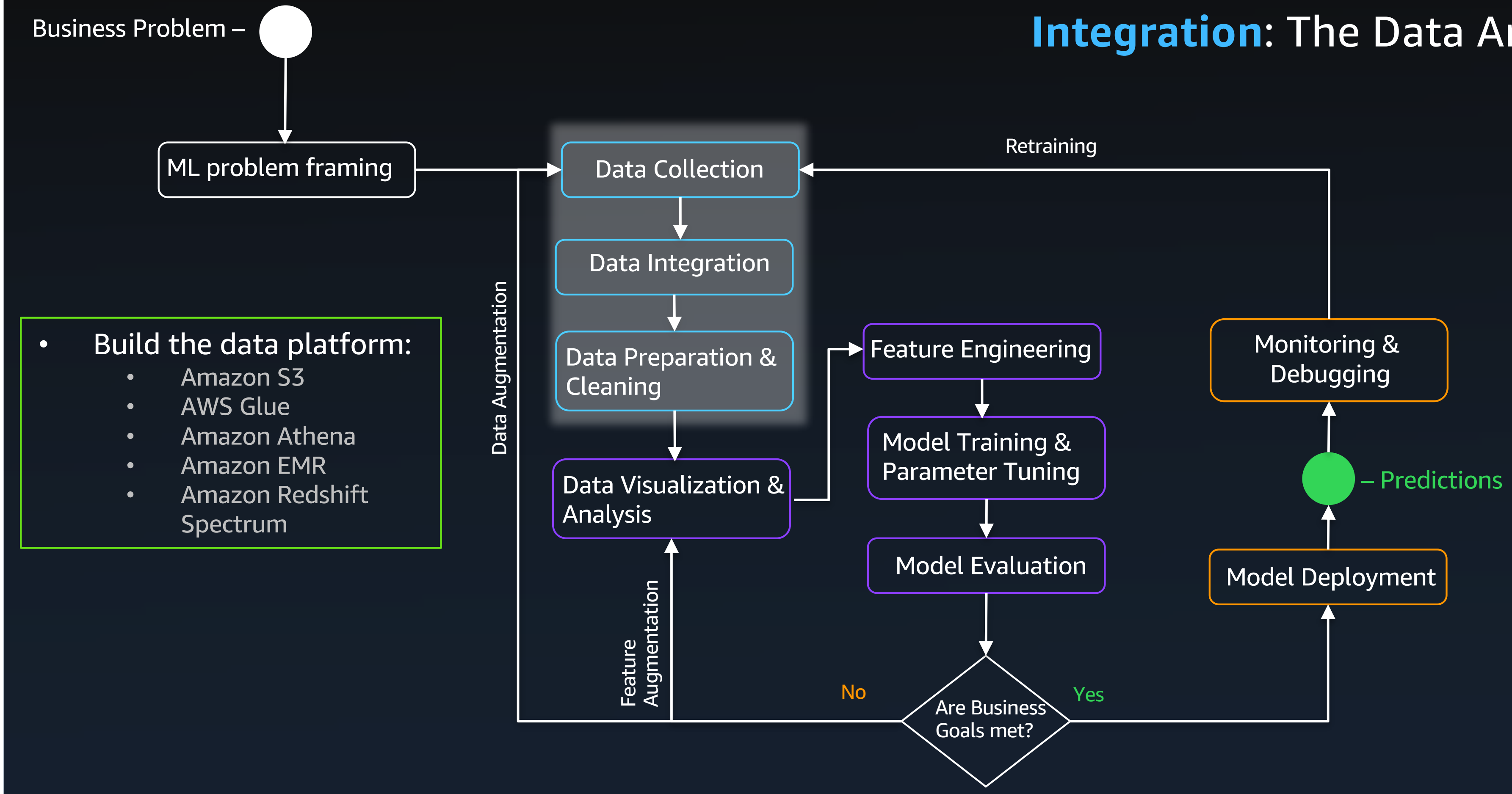
aws machine learning

Discovery: The Analysts



- Help formulate the right questions
  - Domain Knowledge

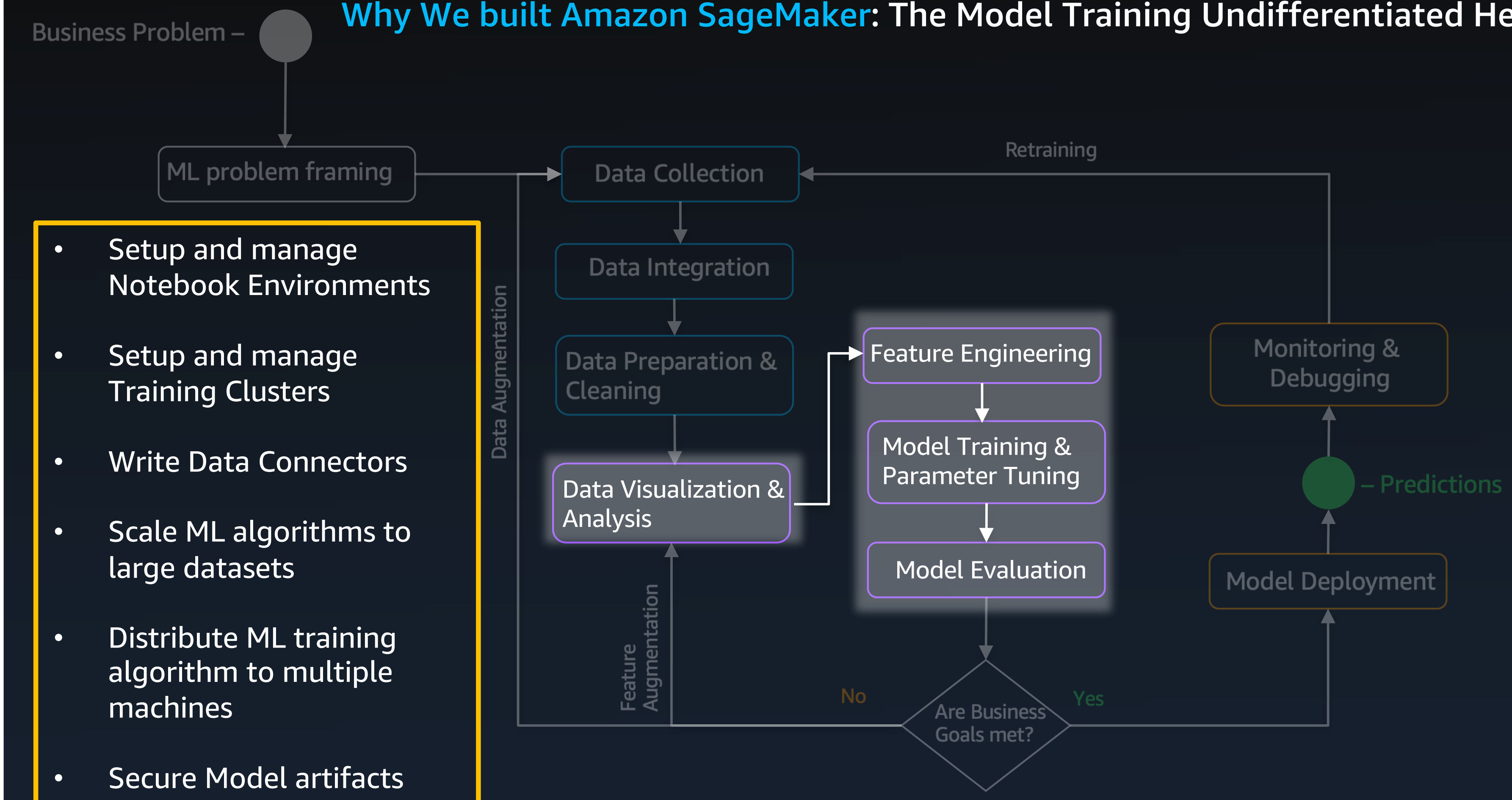
Integration: The Data Architecture



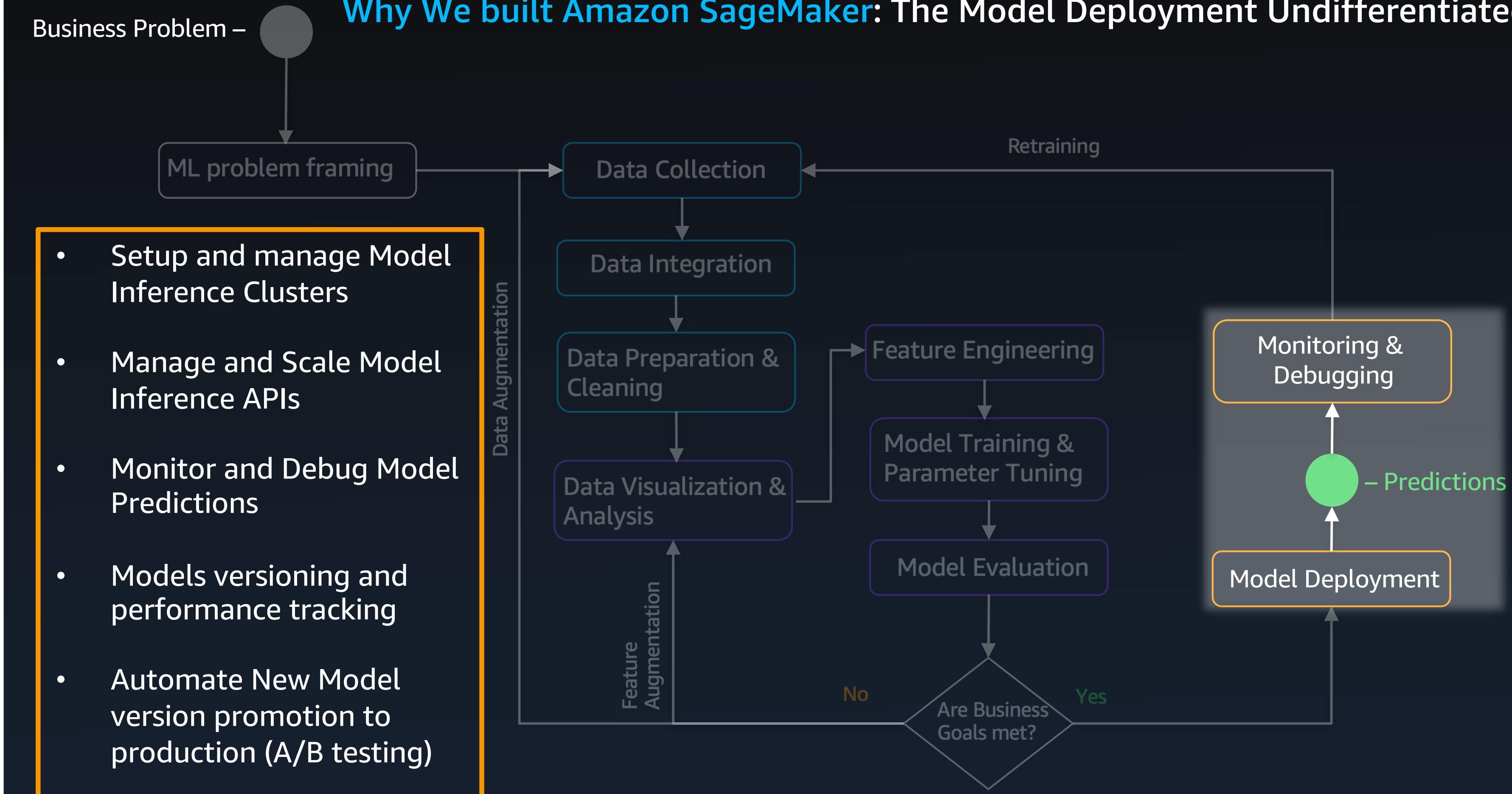
- Build the data platform:
  - Amazon S3
  - AWS Glue
  - Amazon Athena
  - Amazon EMR
  - Amazon Redshift Spectrum

Why We built Amazon SageMaker: The Model Training Undifferentiated Heavy Lifting

- Setup and manage Notebook Environments
- Setup and manage Training Clusters
- Write Data Connectors
- Scale ML algorithms to large datasets
- Distribute ML training algorithm to multiple machines
- Secure Model artifacts



Why We built Amazon SageMaker: The Model Deployment Undifferentiated Heavy Lifting



- Setup and manage Model Inference Clusters
- Manage and Scale Model Inference APIs
- Monitor and Debug Model Predictions
- Models versioning and performance tracking
- Automate New Model version promotion to production (A/B testing)

The AWS Machine Learning Stack

Broadest and most complete set of Machine Learning capabilities

AI SERVICES

VISION	SPEECH		TEXT			SEARCH	CHATBOTS	PERSONALIZATION	FORECASTING	FRAUD	DEVELOPMENT	CONTACT CENTERS
Amazon Rekognition	Amazon Polly	Amazon Transcribe *+Medical*	Amazon Comprehend *+Medical*	Amazon Translate	Amazon Textract	Amazon Kendra	Amazon Lex	Amazon Personalize	Amazon Forecast	Amazon Fraud Detector	Amazon CodeGuru	Contact Lens *For Amazon Connect*

ML SERVICES

Amazon SageMaker

Ground Truth

AWS Marketplace for ML

SageMaker Studio IDE

Built-in algorithms | Notebooks | Experiments | Processing | Model training & tuning | Debugger | Autopilot | Model hosting | Model Monitor

Neo

Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow

mxnet

PYTORCH

GLUON

Keras

Scikit learn

HOROVOD

DeepGraphLibrary

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

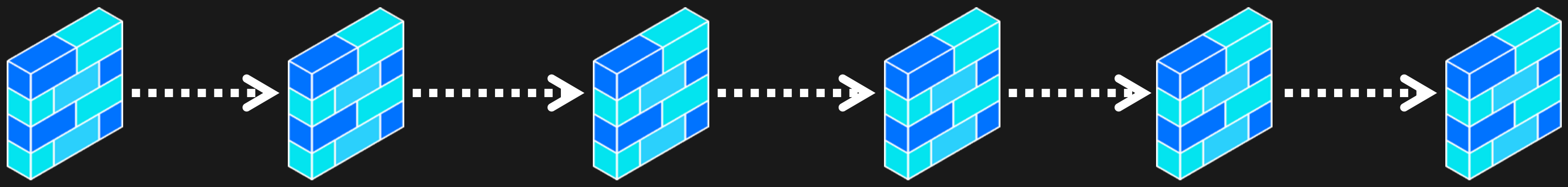
aws machine learning

Amazon SageMaker

A **fully managed service** that enables **data scientists** and **developers** to quickly and easily **build** machine-learning based models **into production**.

Amazon SageMaker

Easily build, train, and deploy machine learning models



Collect and prepare training data

Choose and optimize your ML algorithm

Set up and manage environments for training

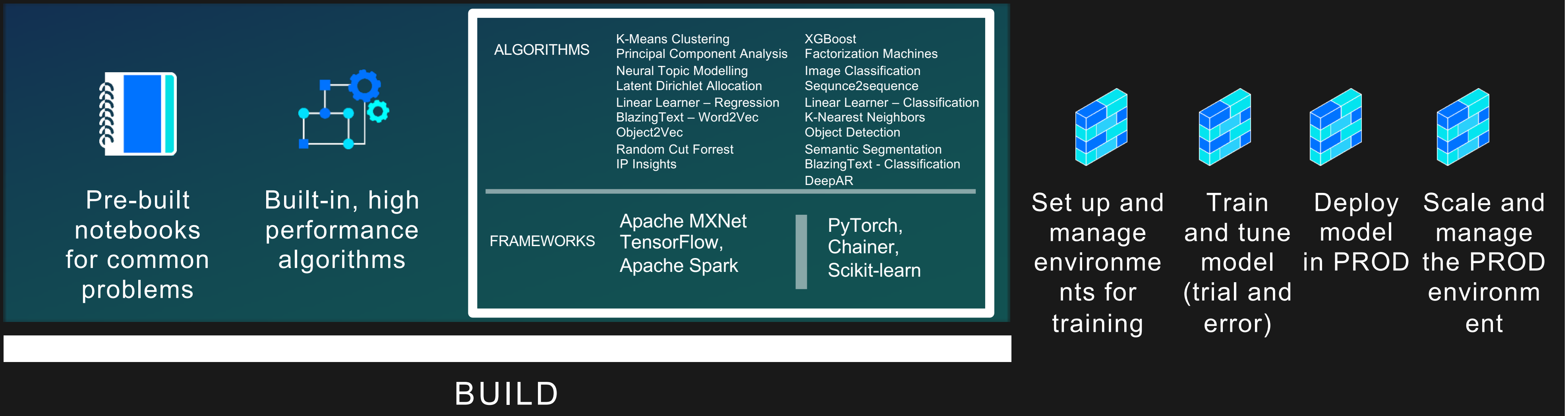
Train and tune model (trial and error)

Deploy model in production

Scale and manage the production environment

https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html
https://sagemaker.readthedocs.io/en/stable/
https://docs.aws.amazon.com/sagemaker/latest/dg/r-guide.html

Amazon SageMaker



Pre-built notebooks for common problems

Built-in, high performance algorithms

ALGORITHMS

K-Means Clustering
Principal Component Analysis
Neural Topic Modelling
Latent Dirichlet Allocation
Linear Learner – Regression
BlazingText – Word2Vec
Object2Vec
Random Cut Forrest
IP Insights

XGBoost
Factorization Machines
Image Classification
Sequnce2sequence
Linear Learner – Classification
K-Nearest Neighbors
Object Detection
Semantic Segmentation
BlazingText - Classification
DeepAR

FRAMEWORKS

Apache MXNet
TensorFlow,
Apache Spark

PyTorch,
Chainer,
Scikit-learn

Set up and manage environments for training

Train and tune model (trial and error)

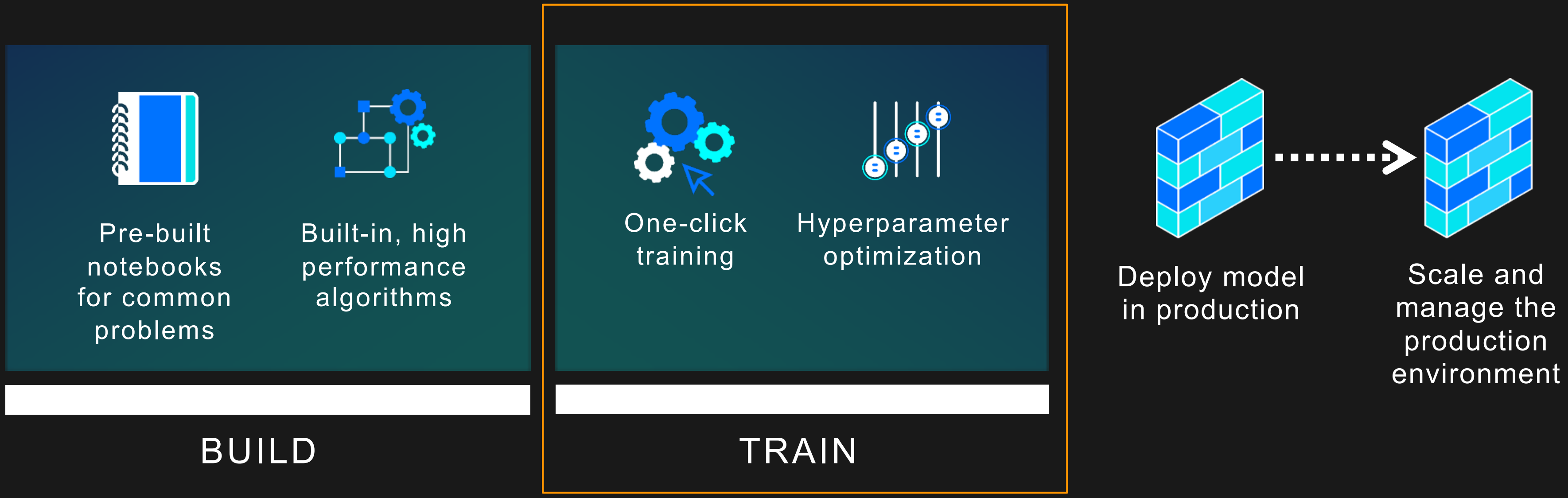
Deploy model in PROD

Scale and manage the PROD environment

BUILD

https://docs.aws.amazon.com/sagemaker/latest/dg/frameworks.html

Amazon SageMaker



Pre-built notebooks for common problems

Built-in, high performance algorithms

BUILD

One-click training

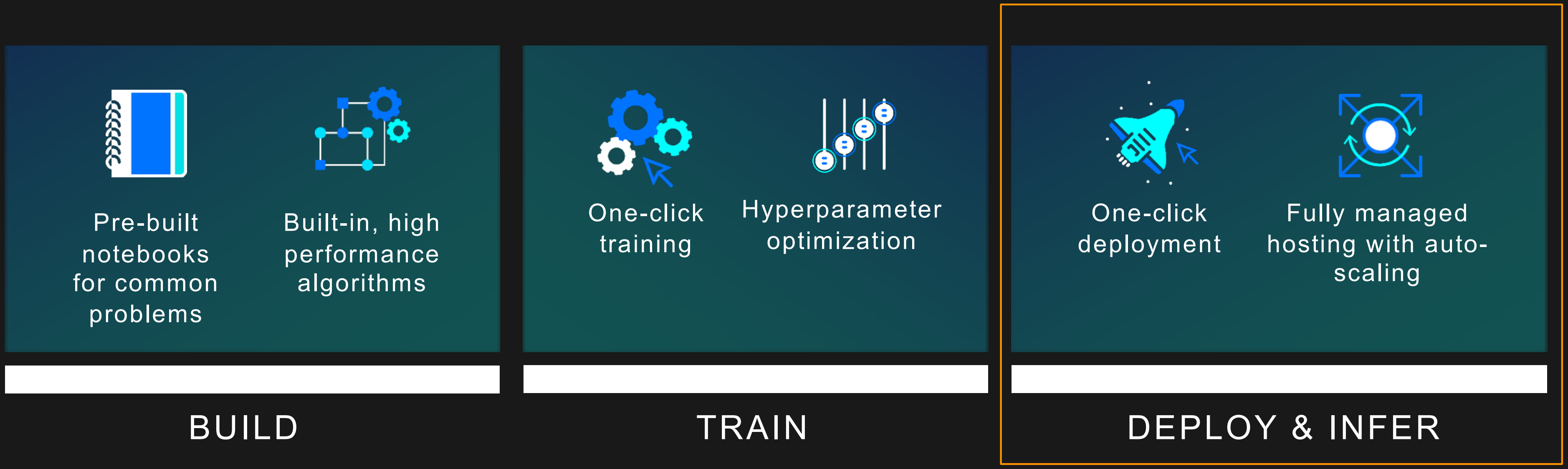
Hyperparameter optimization

TRAIN

Deploy model in production

Scale and manage the production environment

Amazon SageMaker



Pre-built notebooks for common problems

Built-in, high performance algorithms

One-click training

Hyperparameter optimization

One-click deployment

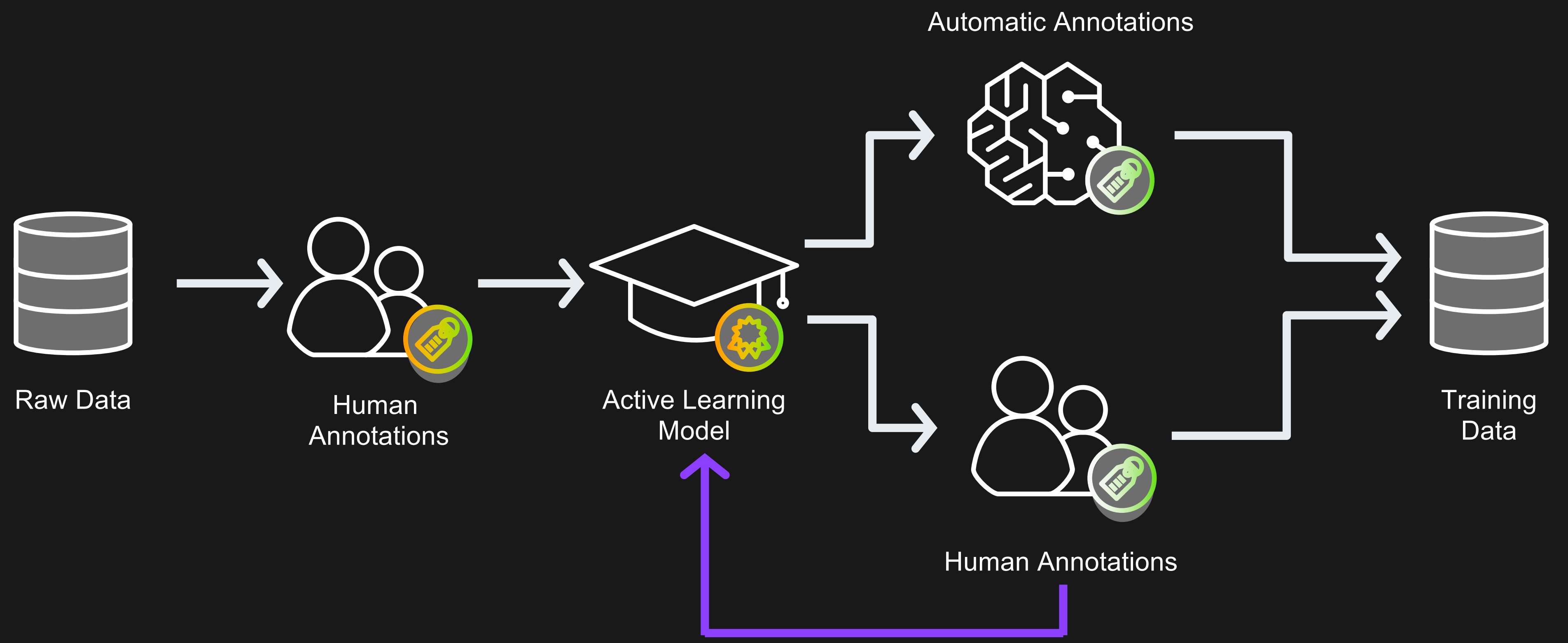
Fully managed hosting with auto-scaling

BUILD

TRAIN

DEPLOY & INFER

Amazon SageMaker Ground Truth: Build highly accurate training datasets and reduce data labeling costs



Amazon Elastic Inference: Add GPU acceleration to any Amazon EC2 instance for faster inference at much lower cost

Lower inference costs

Match capacity
to demand

Available between 1 to 32
TFLOPS per accelerator

KEY FEATURES

Integrated with
Amazon EC2 and
Amazon SageMaker

Support for TensorFlow,
Apache MXNet -
PyTorch coming soon

Single and
mixed-precision
operations

Amazon SageMaker Neo:

Train once, run anywhere with 2x the performance

Get accuracy and performance

Automatic optimization

mxnet

TensorFlow

PYTORCH

Broad framework support

Qualcomm

intel cadence

NVIDIA

XILINX arm

Broad hardware support

KEY FEATURES

Open-source device runtime and compiler,
1/10th the size of original frameworks

AWS Marketplace for Machine Learning

ML algorithms and models available instantly

Browse or search AWS Marketplace

Subscribe in a single click

Available in Amazon SageMaker

KEY FEATURES

SELLERS

- Automatic labeling via machine learning
- IP protection
- Automated billing and metering

BUYERS

- Broad selection of paid, free, and open-source algorithms and models
- Data protection
- Discoverable on your AWS bill

Algorithms in SageMaker

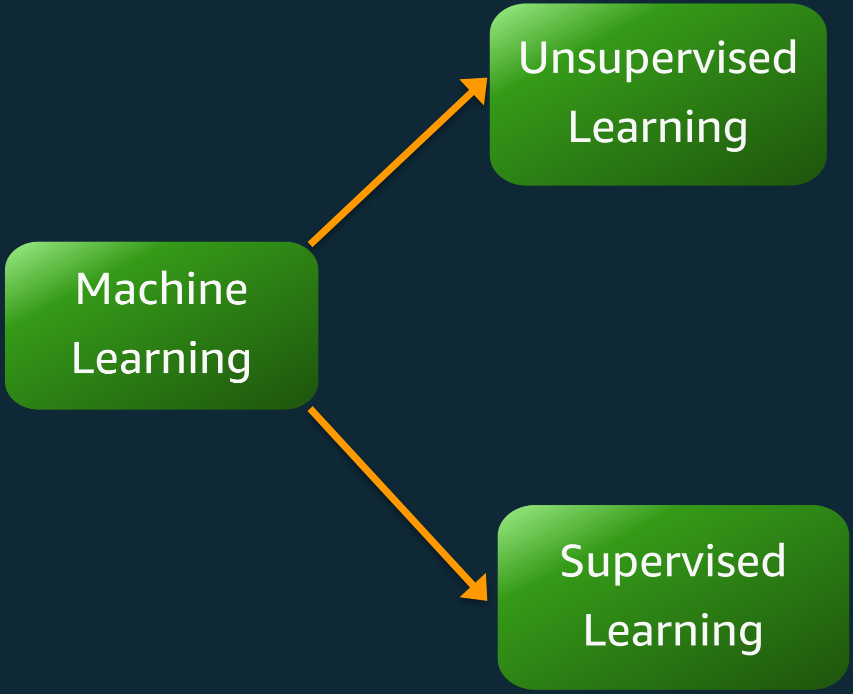
Common enterprise use cases

Use Case	Approach
Healthcare	develop better processes for diagnosis
Financial Services	prevent fraud, know when to trade, and identify high-risk profiles
Retail	capture, analyze, and use customer shopping data to personalize the shopping experience.
Automotive	improve operations, marketing, and customer experience, as well as quality control vehicle parts.
Government	mine data from multiple sources in order to increase efficiency, save money, detect fraud, and protect against identity theft
Oil & Gas	accurate modeling, optimizing drilling operations, predictive maintenance, subsurface characterization, predicting energy purchasing markets
Manufacturing	automation, quality control, supply chain efficiency, smart factory

SageMaker Built-in Algorithms

Reinforcement Learning

SageMaker RL



- K-Means Algorithm
- Principal Component Analysis
- Latent Dirichlet Allocation (LDA)
- Neural Topic Model
- Random Cut Forest
- IP Insights
- BlazingText* – Word2vec, Object2vec

- Linear Learner
- XGBoost Algorithm
- Factorization Machines
- Image Classification
- Sequence2Sequence
- DeepAR Forecasting
- K Nearest-Neighbors
- Object detection
- Semantic segmentation
- BlazingText* – text or document classification

*Semi-supervised

https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html

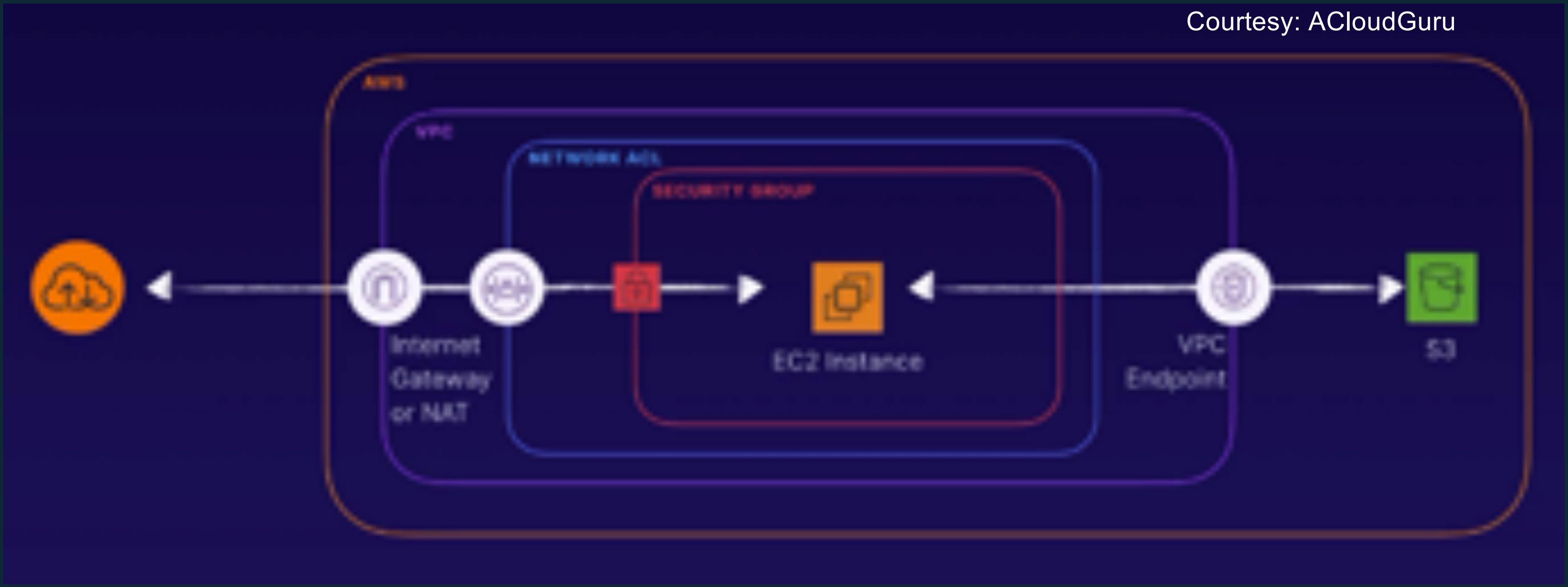
SageMaker Setup & Console demo

SageMaker Setup

- Notebook instance
  https://docs.aws.amazon.com/sagemaker/latest/dg/howitworks-create-ws.html
- IAM role
- S3 bucket
- SageMaker SDK
  - To train, deploy, and validate a model,
    - SageMaker Python SDK or
    - AWS SDK for Python (Boto 3)
  - SageMaker Python SDK abstracts several implementation details, and is easy to use
  - Recommended for first-time users
  - https://sagemaker.readthedocs.io/en/stable/

SageMaker Security

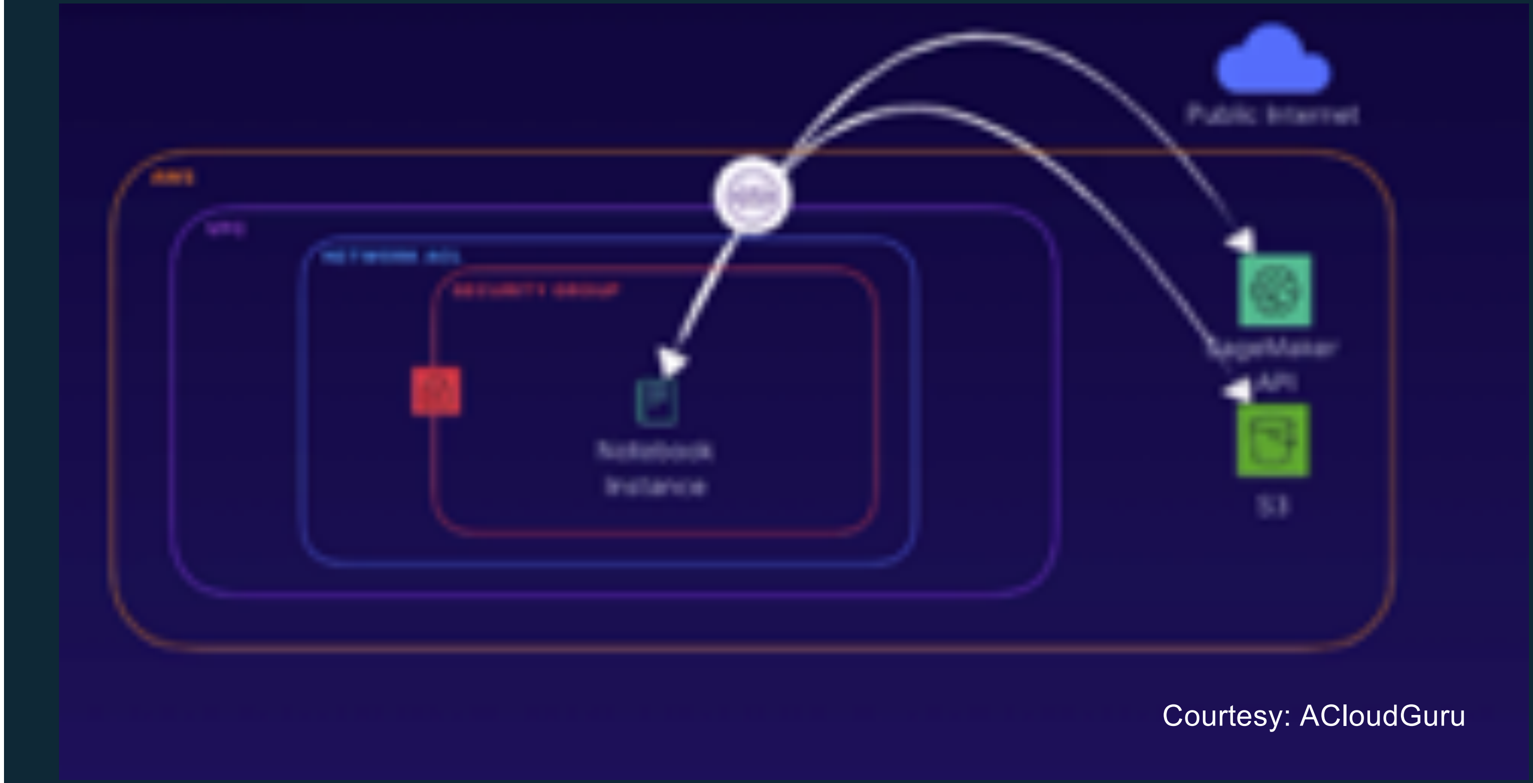
- Visibility, access control, authentication, and encryption



Courtesy: ACloudGuru

https://docs.aws.amazon.com/sagemaker/latest/dg/security.html
https://docs.aws.amazon.com/sagemaker/latest/dg/appendix-notebook-and-internet-access.html

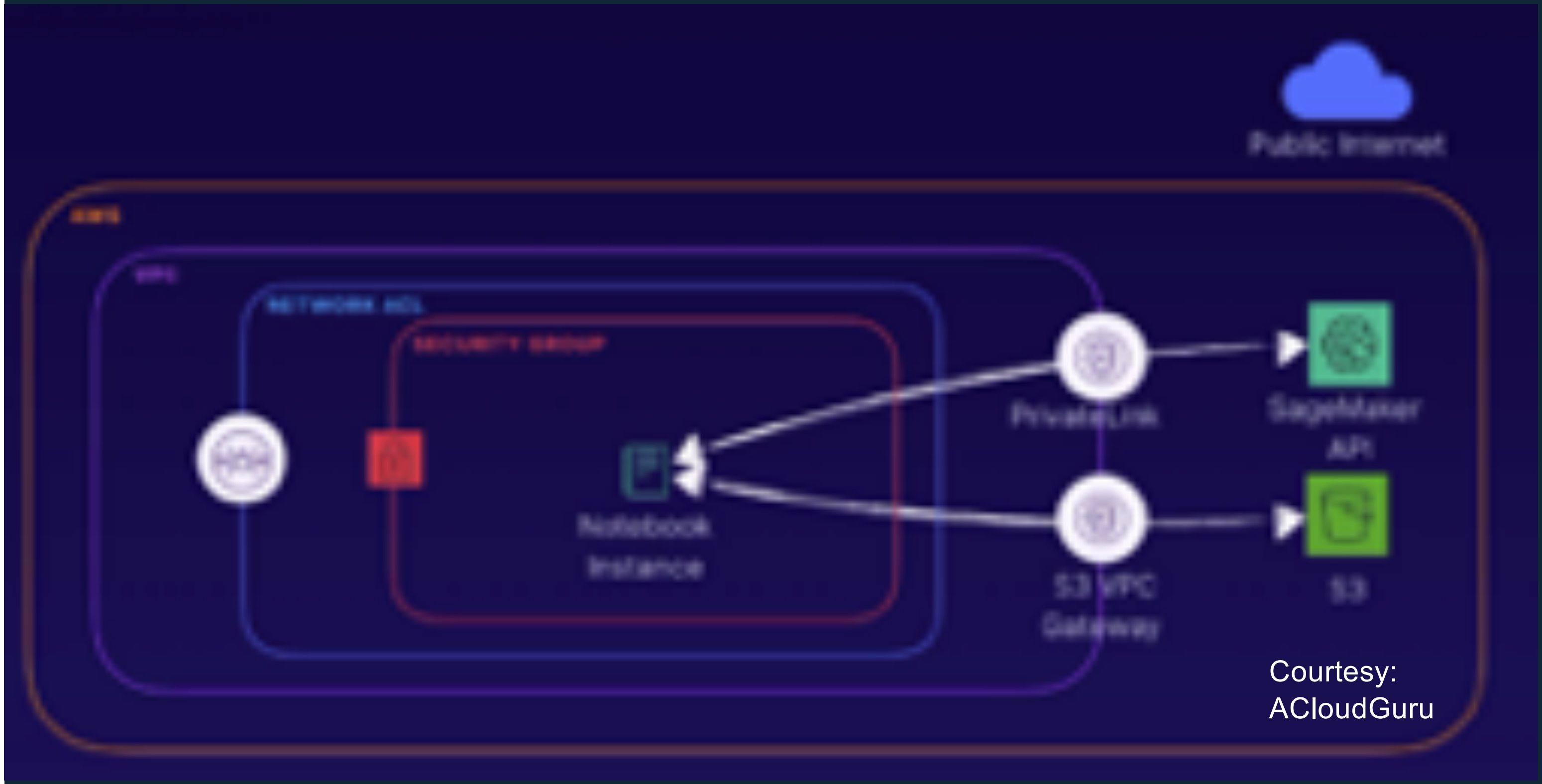
Notebooks are internet enabled by default



Courtesy: ACloudGuru

https://docs.aws.amazon.com/sagemaker/latest/dg/mkt-algo-model-internet-free.html

With VPC Endpoints …



Courtesy: ACloudGuru

- Disable internet access and also permit selected access via NAT GWY, Routes or SGs
- One user/notebook
- Notebooks allow root access
- KMS keys to encrypt data at rest

https://docs.aws.amazon.com/sagemaker/latest/dg/notebook-interface-endpoint.html

Amazon SageMaker Autopilot

Automatic model creation for tabular data with full visibility & control

Quick to start

Provide your data in a tabular form & specify target prediction

Automatic model creation

Get ML models with feature engineering & model tuning automatically done

Visibility & control

Get notebooks for your models with source code

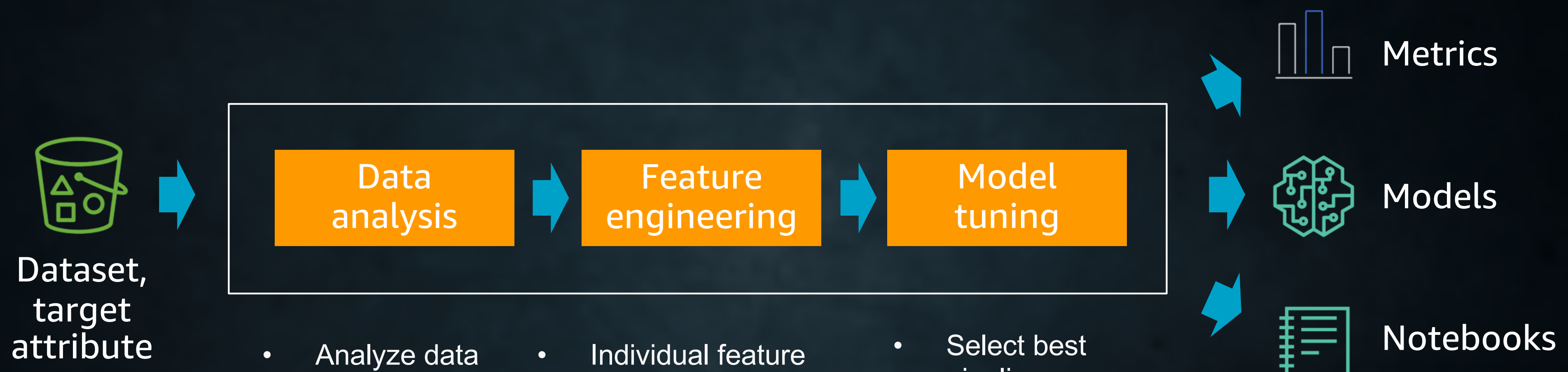
Recommendations & Optimization

Get a leaderboard & continue to improve your model

- Classification, binary and multi-class
- Regression

- https://docs.aws.amazon.com/sagemaker/latest/dg/autopilot-automate-model-development.html

How SageMaker AutoPilot Works



Dataset, target attribute

Data analysis

Feature engineering

Model tuning

Metrics

Models

Notebooks

- Analyze data
- Pipelines
- Data splits

- Individual feature transforms
- Aggregate feature transforms

- Select best pipeline
- Select optimal HPs

https://github.com/awslabs/amazon-sagemaker-examples/tree/master/autopilot

SageMaker Autopilot from the Studio

- only S3 location and target variable required
- optional control points:
  - dry-run vs complete mode
  - setting problem type
  - security settings
- API level control points:
  - number of candidate models to build
  - maximum time to take
  - model evaluation metric (accuracy, F1, RMSE)

Autopilot

Fully runnable model candidate notebook:

- data transformers
- featurization techniques applied

- override points:
  - algorithms considered
  - evaluation metric
  - hyper-parameter ranges
  - model search strategy
  - instances used

What Autopilot does well

- AutoML for classification and regression learning
- Exhibits model transparency and extensibility
    - White box approach
- Data analysis → data properties → feature engineering candidate generation → multiple candidate pipelines
- XGBoost and linear-learner algorithms
    - Scalable, can run distributed for large datasets (5GB)
    - More algorithms to come
    - Up to 10 different candidate pipelines are run in parallel
- Handles both numerical and text data (will featurize text with TF-IDF, etc.

SageMaker Autopilot

Demo

Specific Use Cases

- Regression
  - Two options
    - Autopilot
    - Built-in SageMaker algorithms
- Time series forecasting
  - Two options
    - Amazon Forecast
    - Built-in SageMaker algorithm

How to use SageMaker Algorithms for Regression

- Multiple SageMaker algorithms, e.g., Linear Learner and XGBoost
  - For regression (sales forecasting, predicting delivery times)
    - Set hyperparameter *predictor_type = regressor* (for linear-learner)
    - Set hyperparameter *objective = reg:linear* (for XGBoost)
  - For classification (ad-click prediction, customer churn)
    - Set hyperparameter *predictor_type = binary_classifier* (for linear-learner)
    - Set hyperparameter *objective = reg:logistic* (for XGBoost)
  - Ensembling, aka using both

Regression Example

- Using XGBoost
- https://github.com/awslabs/amazon-sagemaker-examples/blob/master/introduction_to_amazon_algorithms/xgboost_abalone/xgboost_abalone.ipynb
- Console demo

Model Deployment – BYO Algorithms or Models

- SageMaker uses Docker containers for build and runtime tasks
- Put scripts, algorithms, and inference code of your MLmodels into containers
- Package your training code, inference code
- Four options:
  1. Use a built-in algorithm
  2. Use pre-built container images that supports Deep Learning frameworks
  3. Extend a pre-built container image (e.g., PyTorch)
  4. Build your own custom container image

https://docs.aws.amazon.com/sagemaker/latest/dg/your-algorithms.html

Bring-Your-Own-Code Inferencing (BYOM)

- Train your own model
- Model file name must satisfy this RE pattern
- Model file has to be tar-zipped
- Upload ***your*** model to S3
- Import model into hosting (scikit-learn XGBoost model is compatible with SageMaker XGBoost container, other gradient boosted tree models are not)
- Create end-point configuration with model name (now in S3)
- Create end-point
- Run inferencing
- https://docs.aws.amazon.com/sagemaker/latest/dg/your-algorithms-inference-code.html
- Example notebooks: BYOM XGBoost , BYOM K-Means

Bring-Your-Own-Code Training and Hosting

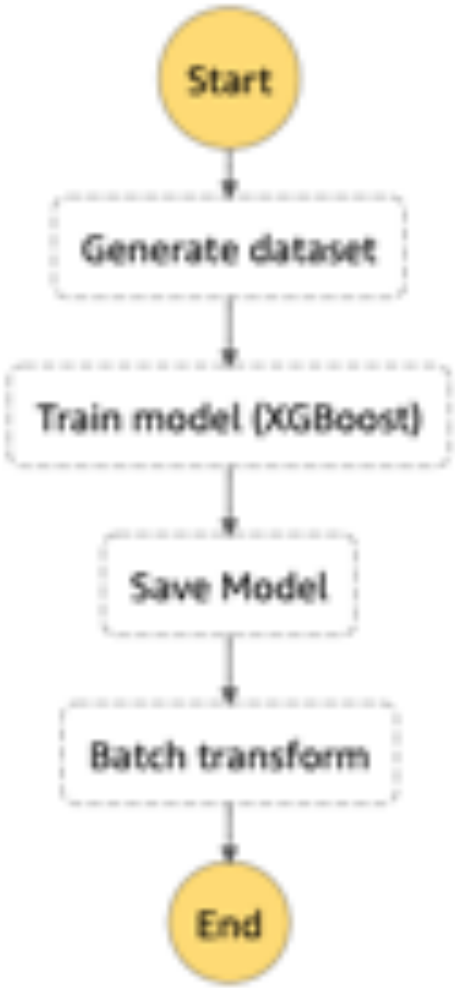
- Package your own algorithm for training and deployment
- Bring any code to SageMaker regardless of programming language, environment, framework, etc.
- Why build your own container?
  - Complex algorithm
  - Special additions to framework
- No need to provide your container for common frameworks
  - Provide code that implements your algorithm
- Add additional permissions: *AmazonEC2ContainerRegistryFullAccess*
- Build the image files (Docker)
- One Docker image for training and hosting or two separate
- How to: https://github.com/aws/sagemaker-training-toolkit
- Example notebooks: BYOM Scikit , BYO R , BYO Host Multiple Models , BYOC TF

Using your custom algorithm in SageMaker

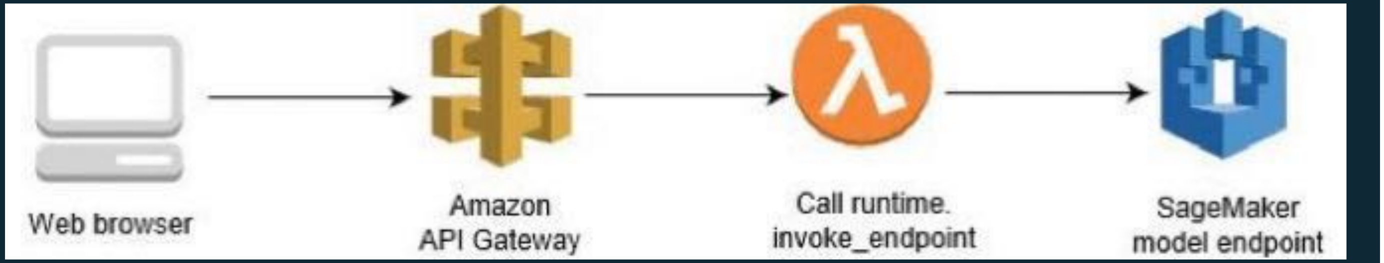
- Define Docker image as described earlier
- Register with SageMaker image registry (ECR)
- Create code entry points as described earlier
- Pass image to SageMaker estimator function
- Fit the model
- Deploy the model for real-time prediction or Batch
- Run inference
- BYOC TF

Using SageMaker with AWS Step Functions

- Using AWS Step Functions to manage batch training:
- https://docs.aws.amazon.com/step-functions/latest/dg/sample-train-model.html
- Notebook: https://github.com/juliensimon/amazon-sagemaker-examples/blob/master/step-functions-data-science-sdk/machine_learning_workflow_abalone/machine_learning_workflow_abalone.ipynb
- https://www.youtube.com/watch?v=0kMdOi69tjQ



Calling SageMaker Endpoints using Amazon API Gateway



- How do we use a hosted SageMaker model?
- Create a SageMaker endpoint → Call using SageMaker run-time API
    - You need infrastructure to host that invocation code
- Can we make this independent of infrastructure? Yes
- Use Lambda to invoke that endpoint (SageMaker API is embedded as Lambda function): https://docs.aws.amazon.com/lambda/latest/dg/welcome.html
- Call Lambda from an API Gateway (https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html)

Additional Workbooks to try
and
Documentation Links

Other useful links: AWS development

- SageMaker Python SDK: https://github.com/aws/sagemaker-python-sdk#sagemaker-python-sdk-overview
- AWS Python SDK: https://aws.amazon.com/sdk-for-python/
- Boto3 for SageMaker: https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/sagemaker.html

SageMaker Notebooks Doclinks

1. Create an S3 bucket: https://docs.aws.amazon.com/sagemaker/latest/dg/gs-config-permissions.html
2. Create a SageMaker notebook instance: https://docs.aws.amazon.com/sagemaker/latest/dg/gs-setup-working-env.html
3. Customize a notebook instance (optional): https://docs.aws.amazon.com/sagemaker/latest/dg/notebook-lifecycle-config.html
4. Additional exercises (for homework) https://docs.aws.amazon.com/sagemaker/latest/dg/ex1.html

SageMaker Operations - Doclinks

1. Monitor and visualize: https://aws.amazon.com/blogs/machine-learning/easily-monitor-and-visualize-metrics-while-training-models-on-amazon-sagemaker/
2. Using common workflows for cloud-based development: https://aws.amazon.com/blogs/machine-learning/how-to-use-common-workflows-on-amazon-sagemaker-notebook-instances/
3. Invoke the model as an endpoint using API Gateway and Lambda: https://aws.amazon.com/blogs/machine-learning/call-an-amazon-sagemaker-model-endpoint-using-amazon-api-gateway-and-aws-lambda/