AWS re:Invent

ENT 321

Build, Train, and Deploy Machine Learning for the Enterprise with Amazon SageMaker

Julien Simon
Principal Technical Evangelist, AI & Machine Learning
Amazon Web Services
@julsimon

AWS re:Invent

Agenda

- Welcome & housekeeping
- Slides: quick overview of Amazon SageMaker
- Labs

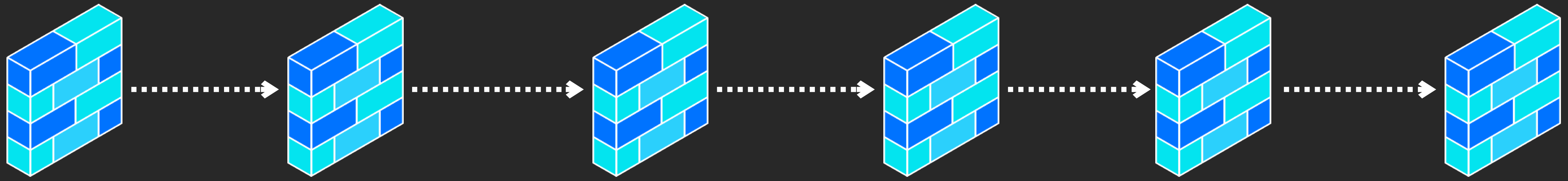
- What we'll cover today:
  - Loading data from Amazon S3
  - Training and deploying with built-in algorithms,
  - Finding optimal hyper parameters with Automatic Model Tuning,
  - Running HTTPS predictions and batch predictions,
  - Beyond built-in algorithms: a peek at Deep Learning.

Amazon SageMaker

AWS re:Invent

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

FREE TIER

Amazon SageMaker

Notebook instances

Built-in, high-performance algorithms

ALGORITHMS

K-Means Clustering
Principal Component Analysis
Neural Topic Modelling
Factorization Machines
Linear Learner

XGBoost
Latent Dirichlet Allocation
Image Classification
Seq2Seq,
And more!

FRAMEWORKS

Apache MXNet, Chainer
TensorFlow, PyTorch

Caffe2, CNTK, Torch

Set up and manage environments for training

Train and tune model (trial and error)

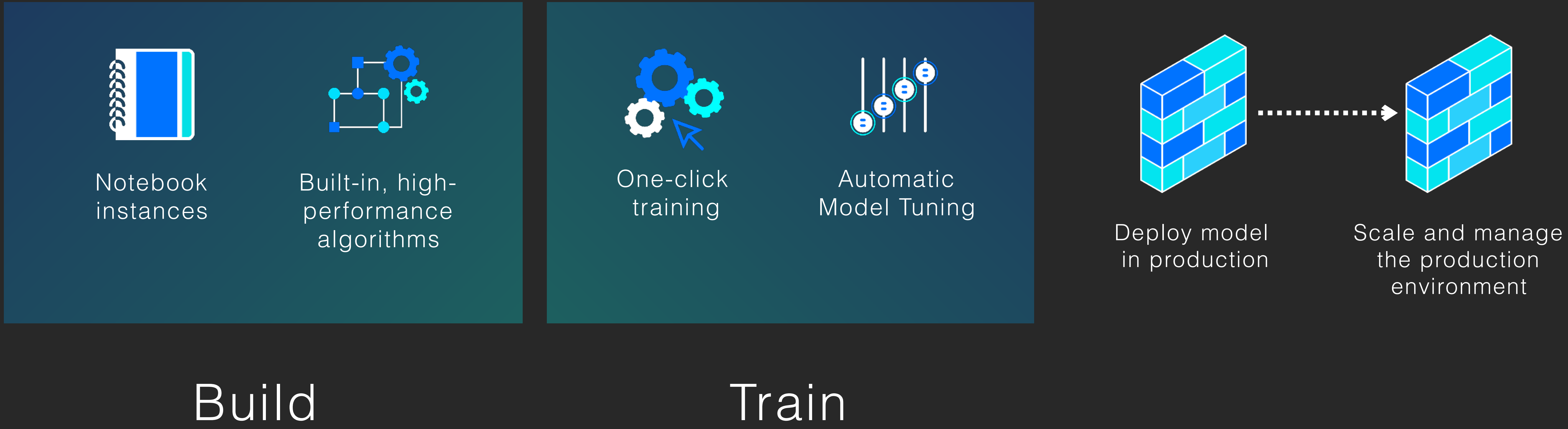
Deploy model in production

Scale and manage the production environment

Build

AWS re:Invent

Amazon SageMaker



Notebook instances

Built-in, high-performance algorithms

One-click training

Automatic Model Tuning

Deploy model in production

Scale and manage the production environment

Build

Train

AWS re:Invent

Amazon SageMaker

Notebook instances

Built-in, high-performance algorithms

Build

One-click training

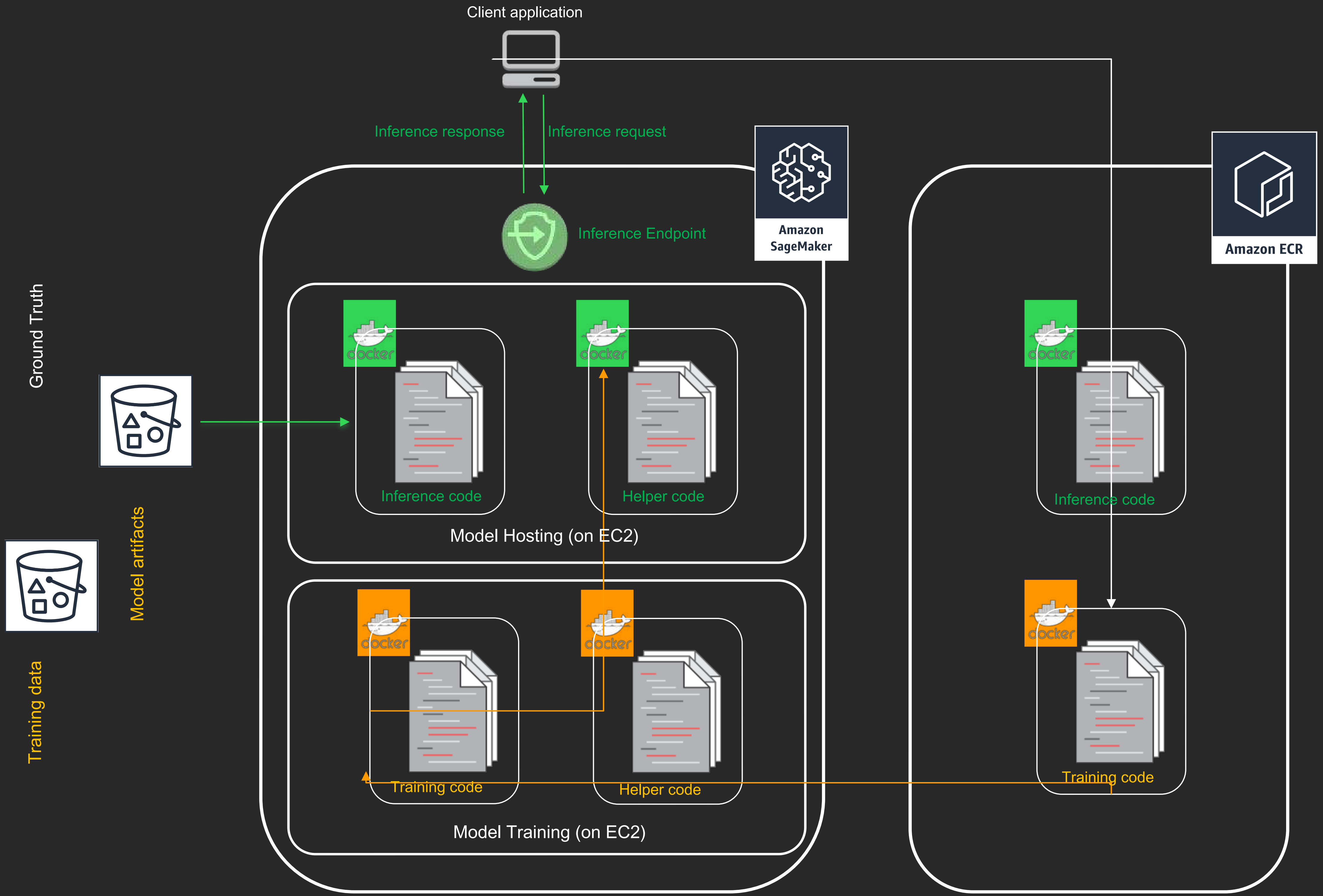
Automatic Model Tuning

Train

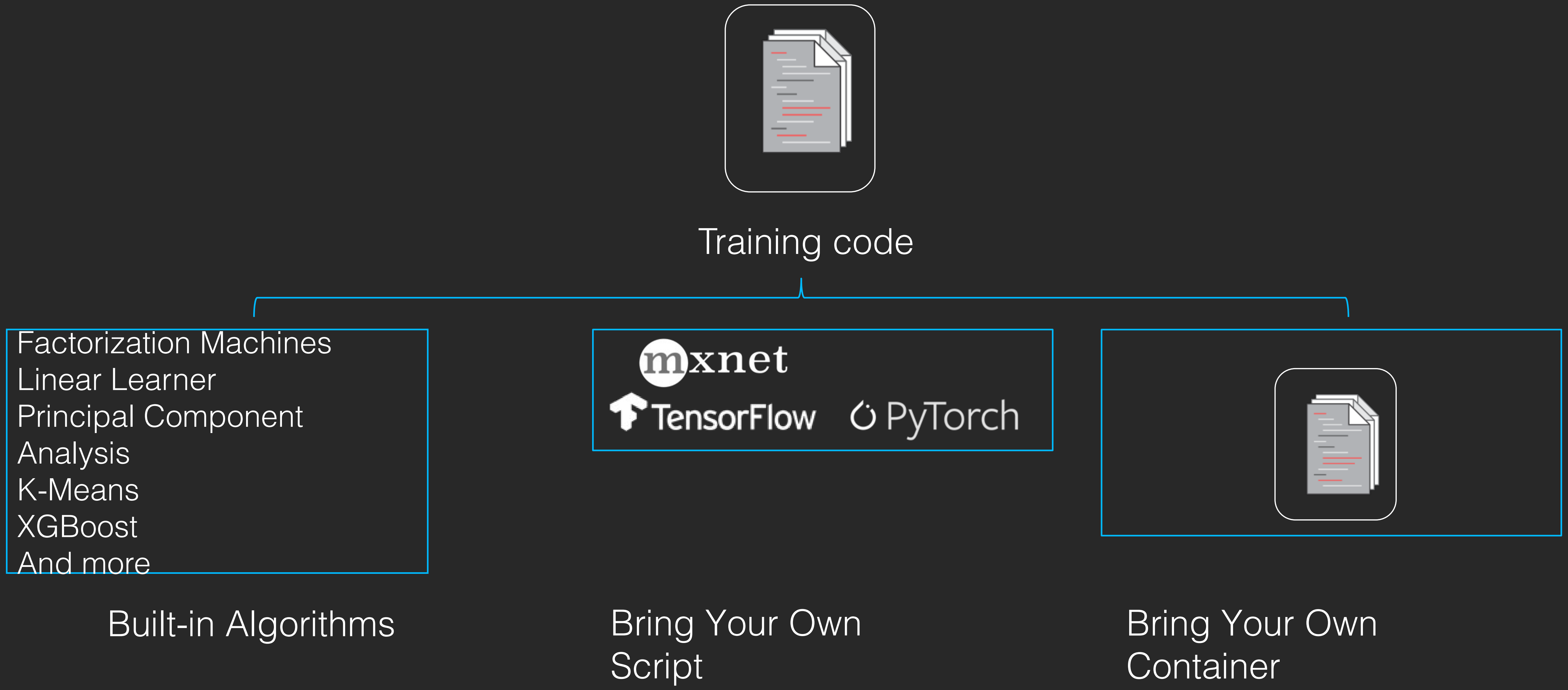
One-click deployment

Fully managed hosting with auto-scaling

Deploy


Client application
Inference response
Inference request
Inference Endpoint
Amazon SageMaker
Amazon ECR
Ground Truth
Model artifacts
Training data
Inference code
Helper code
Model Hosting (on EC2)
Training code
Helper code
Model Training (on EC2)
Inference code
Training code

Model options



Training code

Factorization Machines
Linear Learner
Principal Component
Analysis
K-Means
XGBoost
And more

Built-in Algorithms

mxnet
TensorFlow
PyTorch

Bring Your Own
Script

Bring Your Own
Container

The Amazon SageMaker SDK

- Python SDK orchestrating all Amazon SageMaker activity
  - Algorithm selection, training, deploying, hyper parameter optimization, etc.
  - There's also a Spark SDK (Python and Scala) which we won't cover today.

- High-level objects for:
  - Some built-in algos: Kmeans, PCA, etc.
  - Deep Learning libraries: TensorFlow, MXNet, PyTorch, Chainer.
  - Sagemaker.estimator.estimator for everything else.

https://github.com/aws/sagemaker-python-sdk
https://sagemaker.readthedocs.io/en/latest/

Built-in algorithms

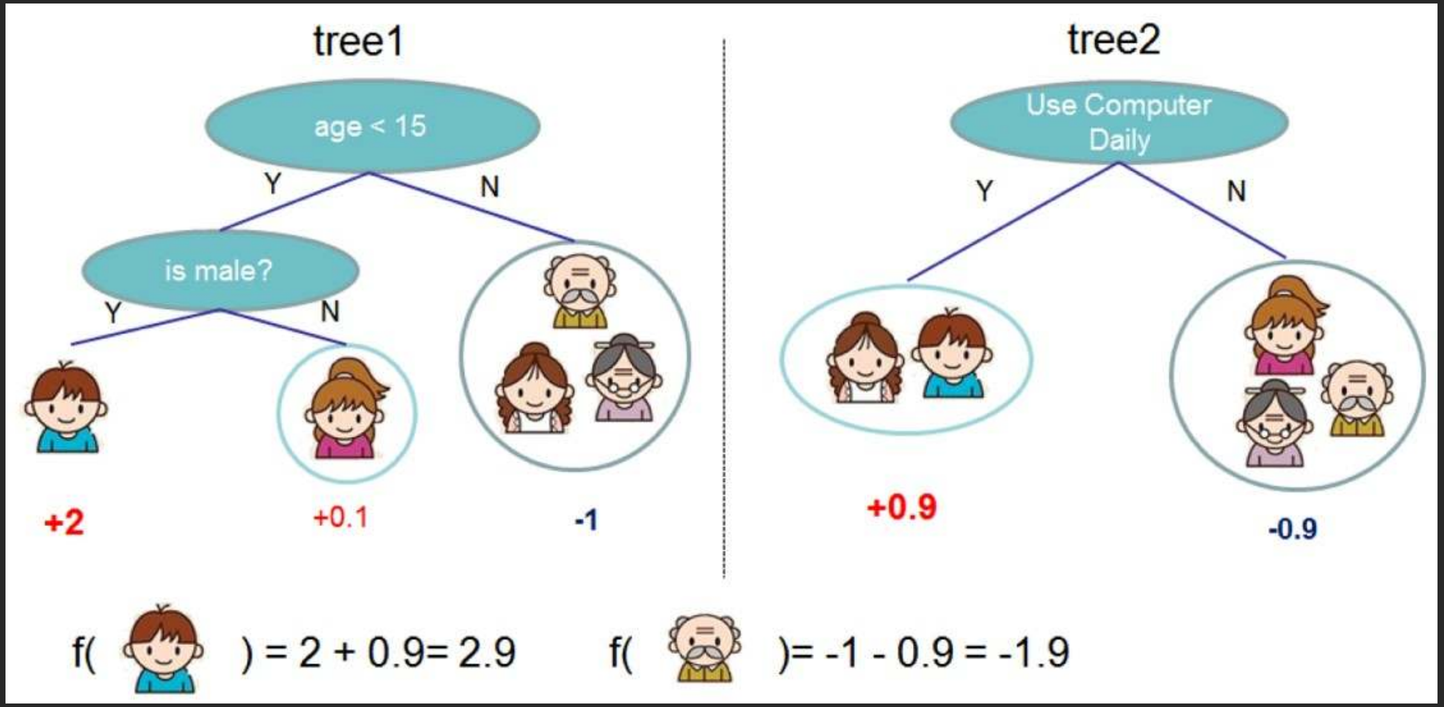
pink: supervised, blue: unsupervised

Linear Learner: regression, classification	**Image Classification**: Deep Learning (ResNet)
Factorization Machines: regression, classification, recommendation	**Object Detection**: Deep Learning (VGG or ResNet)
K-Nearest Neighbors: non-parametric regression and classification	**Neural Topic Model**: topic modeling
XGBoost: regression, classification, ranking https://github.com/dmlc/xgboost	**Latent Dirichlet Allocation**: topic modeling (mostly)
K-Means: clustering	**Blazing Text**: GPU-based Word2Vec, and text classification
Principal Component Analysis: dimensionality reduction	**Sequence to Sequence**: machine translation, speech to text and more
Random Cut Forest: anomaly detection	**DeepAR**: time-series forecasting (RNN)
Object2Vec: general-purpose embeddings	**IP Insights**: usage patterns for IP addresses

XGBoost

dmlc XGBoost

- Open Source project
- Popular tree-based algorithm for regression, classification and ranking
- Builds a collection of trees.
- Handles missing values and sparse data
- Supports distributed training
- Can work with data sets larger than RAM



https://github.com/dmlc/xgboost
https://xgboost.readthedocs.io/en/latest/
https://arxiv.org/abs/1603.02754

Loading training data from Amazon S3

- Two modes: File Mode and Pipe Mode.
  - *input_mode* parameter in *sagemaker.estimator.Estimator.*
- File Mode copies the data set to training instances.
  - You need to provision enough storage.
  - *S3DataSource* object.
  - *S3DataDistributionType* : *FullyReplicated* | *ShardedByS3Key*
  - Differerent data formats are supported: CSV, protobuf, JSON, libsvm (check algo docs!).
- Pipe Mode streams the data set to training instances.
  - This allows you to process infinitely-large data sets.
  - Training starts faster.
  - This mode is supported by some built-in algos as well as Tensorflow.
  - Your data set must be in recordIO-encoded protobuf format.

Walkthrough:

- AWS credits
- SageMaker console
- Notebook instance setup

AWS re:Invent

Labs

Labs

1. Training, deploying and predicting with XGBoost

2. Finding optimal hyper parameters with Automatic Model Tuning

3. Running HTTPS predictions and batch predictions,

4. Beyond built-in algorithms: a peek at TensorFlow.

Resources

AWS re:Invent

Resources

https://ml.aws

https://aws.amazon.com/sagemaker

https://github.com/awslabs/amazon-sagemaker-examples

https://github.com/aws/sagemaker-python-sdk

https://medium.com/@julsimon

Thank you!

Julien Simon
Principal Technical Evangelist, AI & Machine Learning
Amazon Web Services
@julsimon

AWS re:Invent