DEV DAY

Build, train and deploy Machine Learning models at scale

Julien Simon
Global Evangelist, AI & Machine Learning
@julsimon

aws

Our mission

Put Machine Learning in the hands
of every developer and data scientist

The Amazon ML Stack: Broadest & Deepest Set of Capabilities

AI SERVICES

Vision: REKOGNITION IMAGE, REKOGNITION VIDEO, TEXTRACT (NEW)

Speech: POLLY, TRANSCRIBE

Language: TRANSLATE, COMPREHEND, COMPREHEND MEDICAL (NEW)

Chatbots: LEX

Forecasting: FORECAST (NEW)

Recommendations: PERSONALIZE (NEW)

ML SERVICES

AMAZON SAGEMAKER

BUILD
- Pre-built algorithms & notebooks (NEW)
- Data labeling (GROUND TRUTH) (NEW)
- Algorithms & models (AWS MARKETPLACE)

TRAIN
- One-click model training & tuning
- Optimization (NEO) (NEW)
- Models without training data (REINFORCEMENT LEARNING) (NEW)

DEPLOY
- One-click deployment & hosting

ML FRAMEWORKS & INFRASTRUCTURE

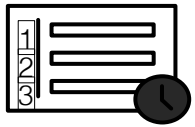
Frameworks: TensorFlow, mxnet, PYTORCH

Interfaces: GLUON, Keras

Infrastructure: EC2 P3 & P3dn, EC2 C5, FPGAs, GREENGRASS, ELASTIC INFERENCE (NEW)

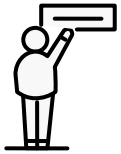
Amazon SageMaker:
Build, Train, and Deploy ML Models at Scale

Collect and prepare training data



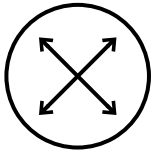
Choose and optimize your ML algorithm

Set up and manage environments for training



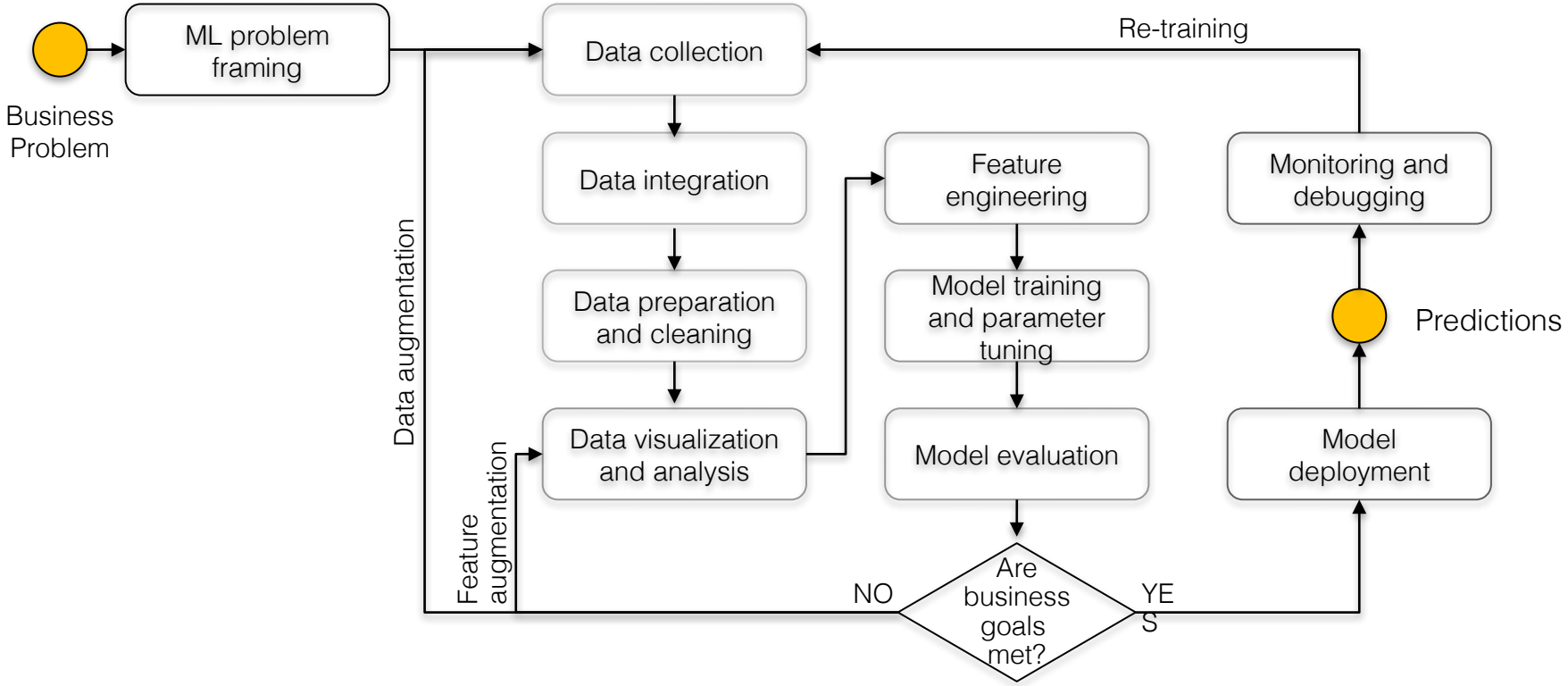
Train and Tune ML Models

Deploy models in production

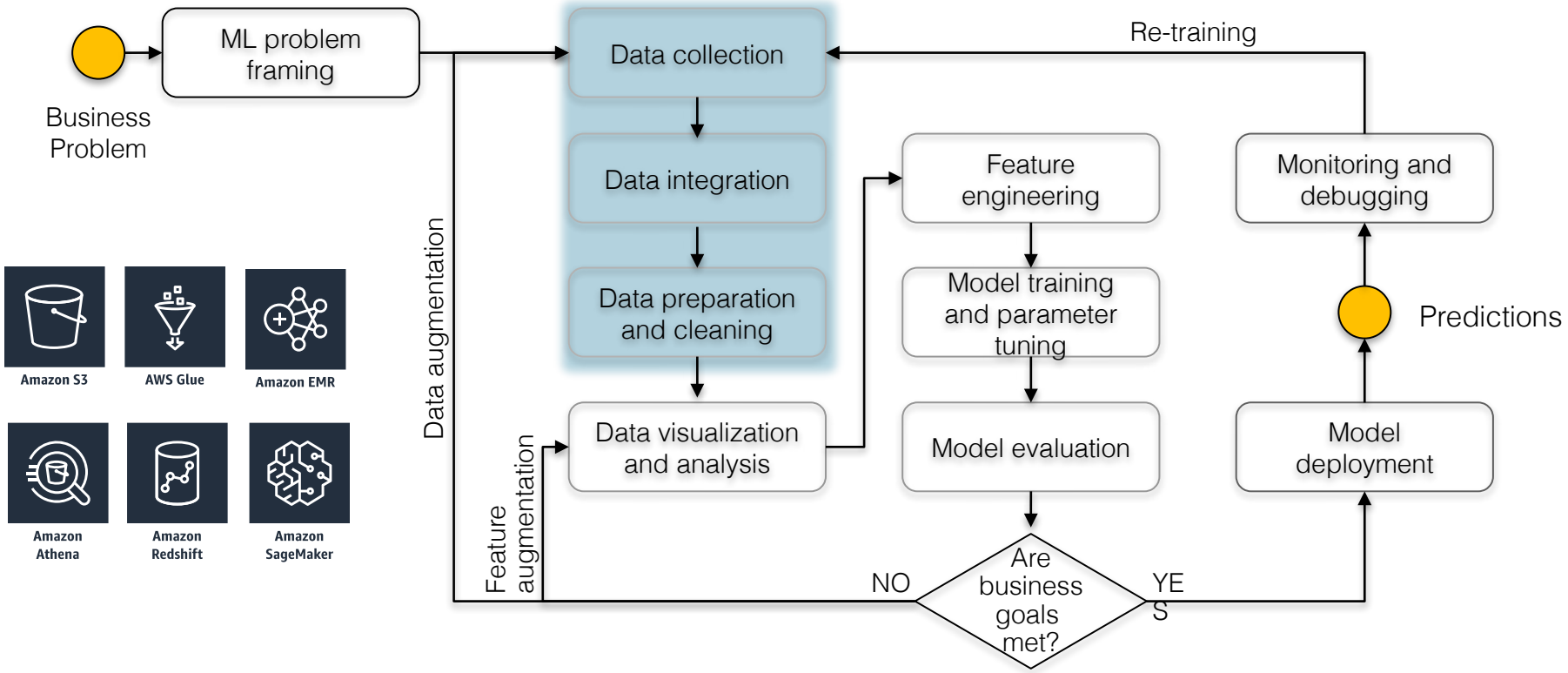


Scale and manage the production environment

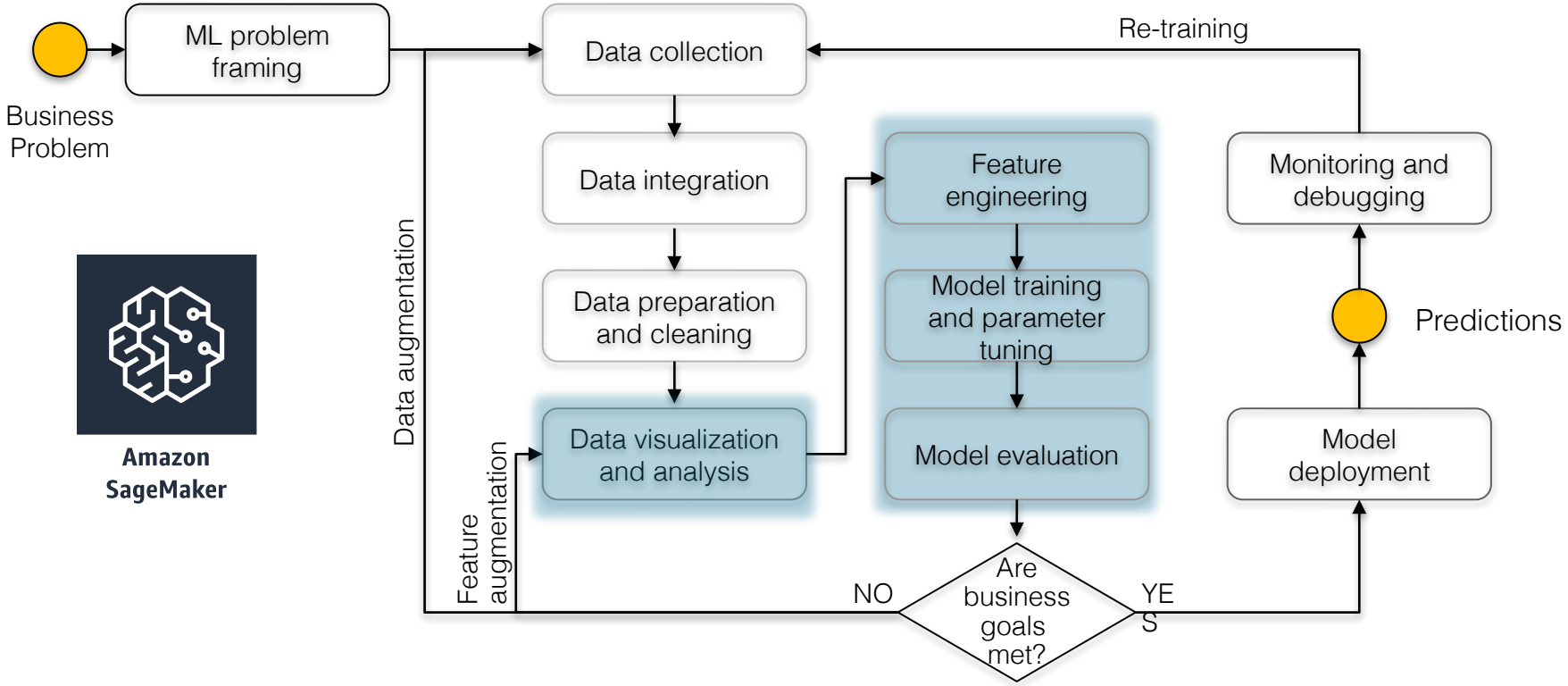
Machine learning cycle



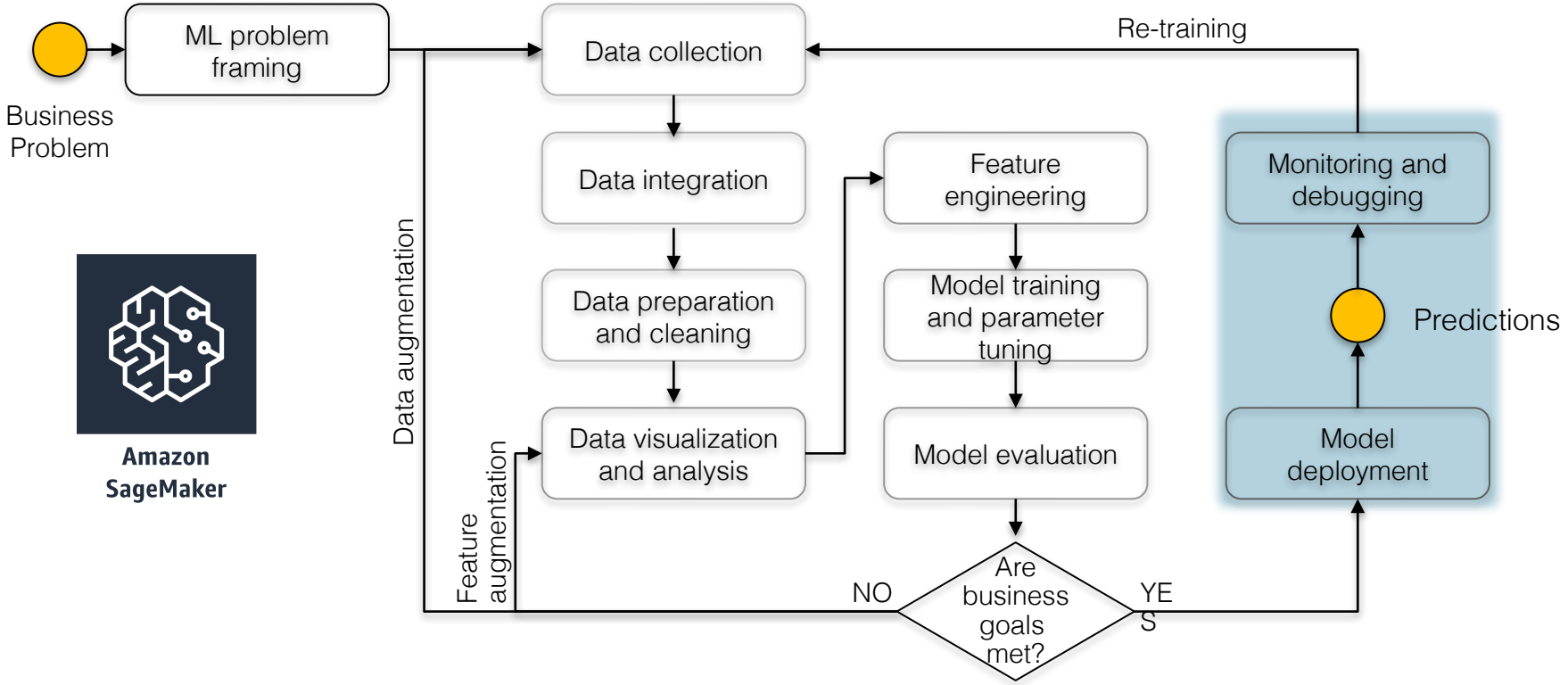
Manage data on AWS



Build and train models using SageMaker



Deploy models using SageMaker



Amazon SageMaker

Build

Pre-built notebooks for common problems

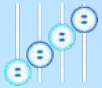
Built-in, high-performance algorithms and frameworks

Git integration
Elastic inference

New built-in algorithms
scikit-learn environment
Model marketplace
Search

Train

One-click training



Hyperparameter optimization

P3DN, C5N
TensorFlow on 256 GPUs
Resume HPO tuning job

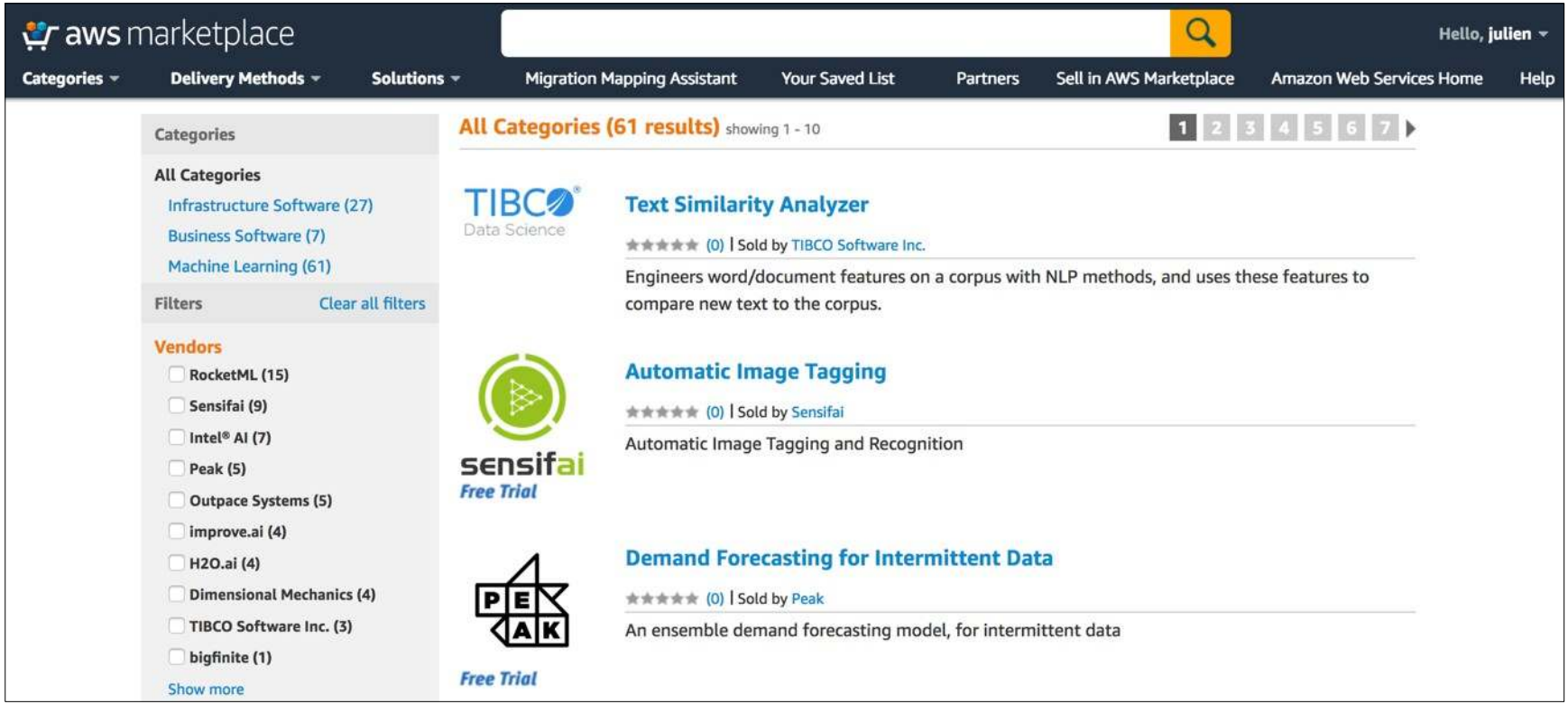
Deploy

One-click deployment

Fully managed hosting with auto-scaling

Model compilation
Elastic inference
Inference pipelines

Machine Learning Marketplace



DEV DAY

Working with Amazon SageMaker

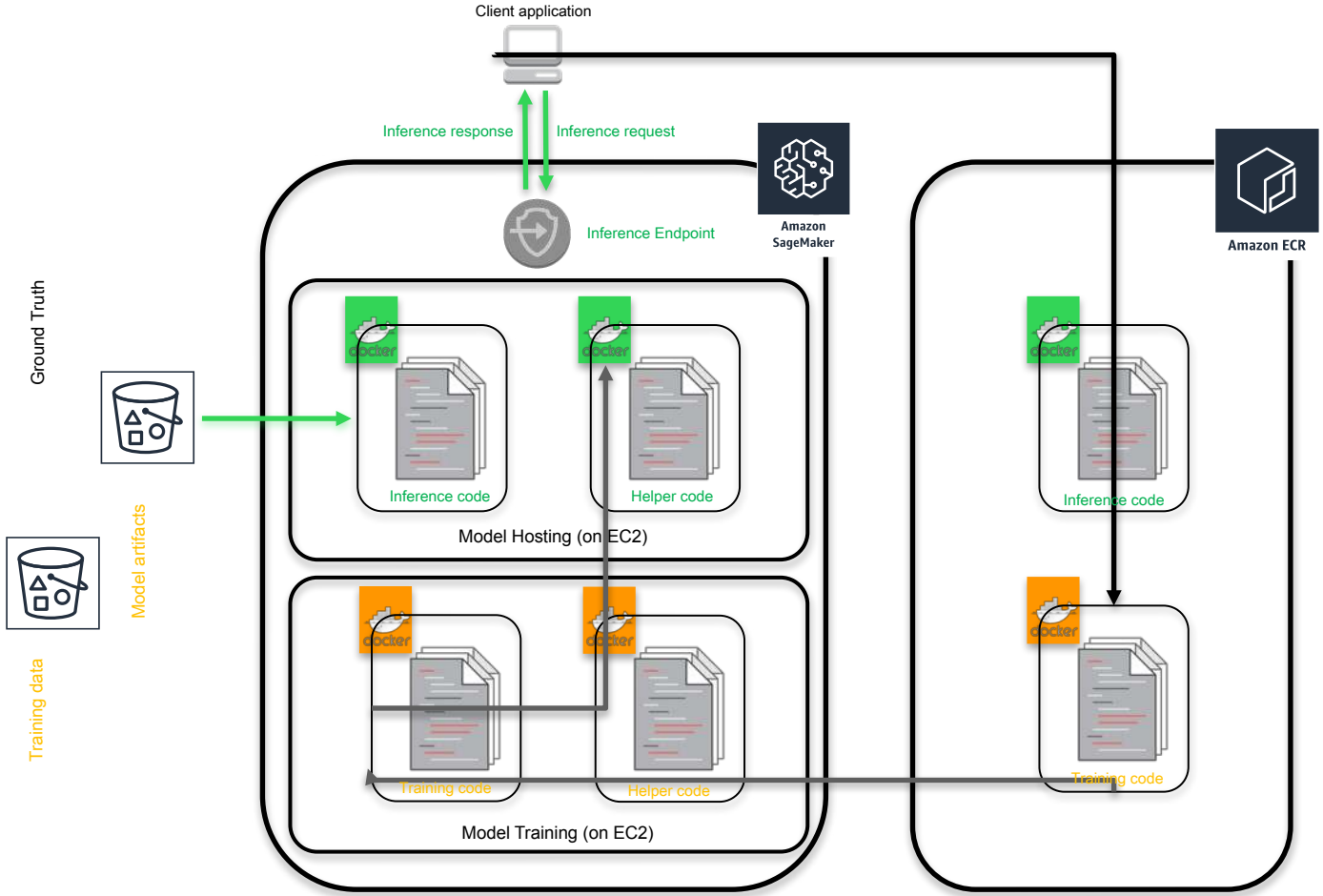
aws

The Amazon SageMaker API

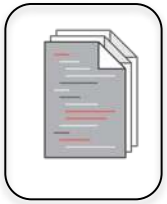
- Python SDK orchestrating all Amazon SageMaker activity
  - High-level objects for algorithm selection, training, deploying, automatic model tuning, etc.
  - Spark SDK (Python & Scala)

- AWS CLI: '*aws sagemaker*'

- AWS SDK: boto3, etc.


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)
Inference code
Training code
Helper code
Model Training (on EC2)
Training code

Model options



Training code

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

Built-in Algorithms

mxnet Chainer TensorFlow PyTorch scikit learn

Bring Your Own Script

Bring Your Own Container

DEV DAY

Built-in algorithms

aws

Built-in algorithms

orange: supervised, yellow: unsupervised

Linear Learner: regression, classification	**Image Classification**: Deep Learning (ResNet)
Factorization Machines: regression, classification, recommendation	**Object Detection (SSD)**: 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 embedding	**IP Insights**: usage patterns for IP addresses
Semantic Segmentation: Deep Learning	

Demo:
Image classification with Caltech-256
+ model tuning

https://gitlab.com/juliensimon/dlnotebooks/sagemaker/

Blazing Text

BlazingText: Scaling and Accelerating Word2Vec using Multiple GPUs

Saurabh Gupta
Amazon Web Services
gsaur@amazon.com

Vineet Khare
Amazon Web Services
vkhare@amazon.com

https://dl.acm.org/citation.cfm?id=3146354

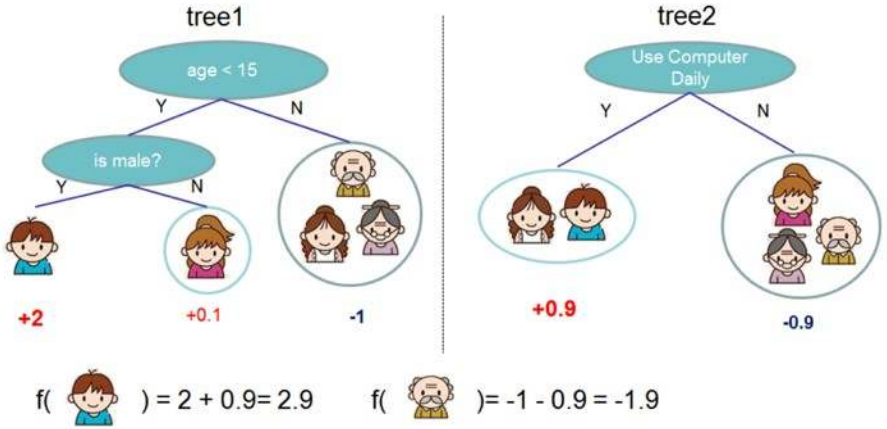
Demo:
Text Classification with BlazingText

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

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

Demo: XGBoost in depth

AWS re:Invent 2018 workshop

https://gitlab.com/juliensimon/ent321

DEV DAY

Built-in frameworks

Demo:
Keras + model tuning+ Elastic Inference

https://gitlab.com/juliensimon/dlnotebooks/keras/

Demo:
Sentiment analysis with Apache MXNet

https://github.com/awslabs/amazon-sagemaker-examples/blob/master/sagemaker-python-sdk/mxnet_sentiment_analysis_with_gluon.ipynb

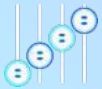
Amazon SageMaker

Pre-built notebooks for common problems

Built-in, high-performance algorithms and frameworks

Build

One-click training



Hyperparameter optimization

Train

One-click deployment

Fully managed hosting with auto-scaling

Deploy

FREE TIER

Score card

Flame war in 3, 2, 1…

	EC2	ECS / EKS	SageMaker
Infrastructure effort	Maximal	Some (Docker tools)	None
ML setup effort	Some (DL AMI)	Some (DL containers)	Minimal
CI/CD integration	No change	No change	Some (SDK, Step Functions)
Build models	DIY	DIY	17 built-in algorithms
Train models (at scale)	DIY	DIY (Docker tools)	2 LOCs
Deploy models (at scale)	DIY (model servers)	DIY (Docker tools)	1 LOCs
Scale/HA inference	DIY (Auto Scaling, LB)	DIY (Services, pods, etc.)	Built-in
Optimize costs	DIY (Spot, RIs, automation)	DIY (Spot, RIs, automation)	On-demand training, Auto Scaling for inference
Security	DIY (IAM, VPC, KMS)	DIY (IAM, VPC, KMS)	API parameters
Personal opinion	Small scale only, unless you have strong DevOps skills and enjoy exercising them.	Reasonable choice if you're a Docker shop and know how to use the rich Docker ecosystem. If not, I'd think twice: Docker isn't an ML platform.	Learn it in a few hours, forget about servers, focus 100% on ML, enjoy goodies like pipe mode, distributed training, HPO, inference pipelines and more.

Getting started

http://aws.amazon.com/free

https://ml.aws
https://aws.amazon.com/sagemaker

https://github.com/aws/sagemaker-python-sdk
https://github.com/aws/sagemaker-spark
https://github.com/awslabs/amazon-sagemaker-examples
https://gitlab.com/juliensimon/ent321

https://medium.com/@julsimon
https://gitlab.com/juliensimon/dlnotebooks

DEV DAY

Thank you!

Julien Simon
Global Evangelist, AI & Machine Learning
@julsimon
https://medium.com/@julsimon

aws