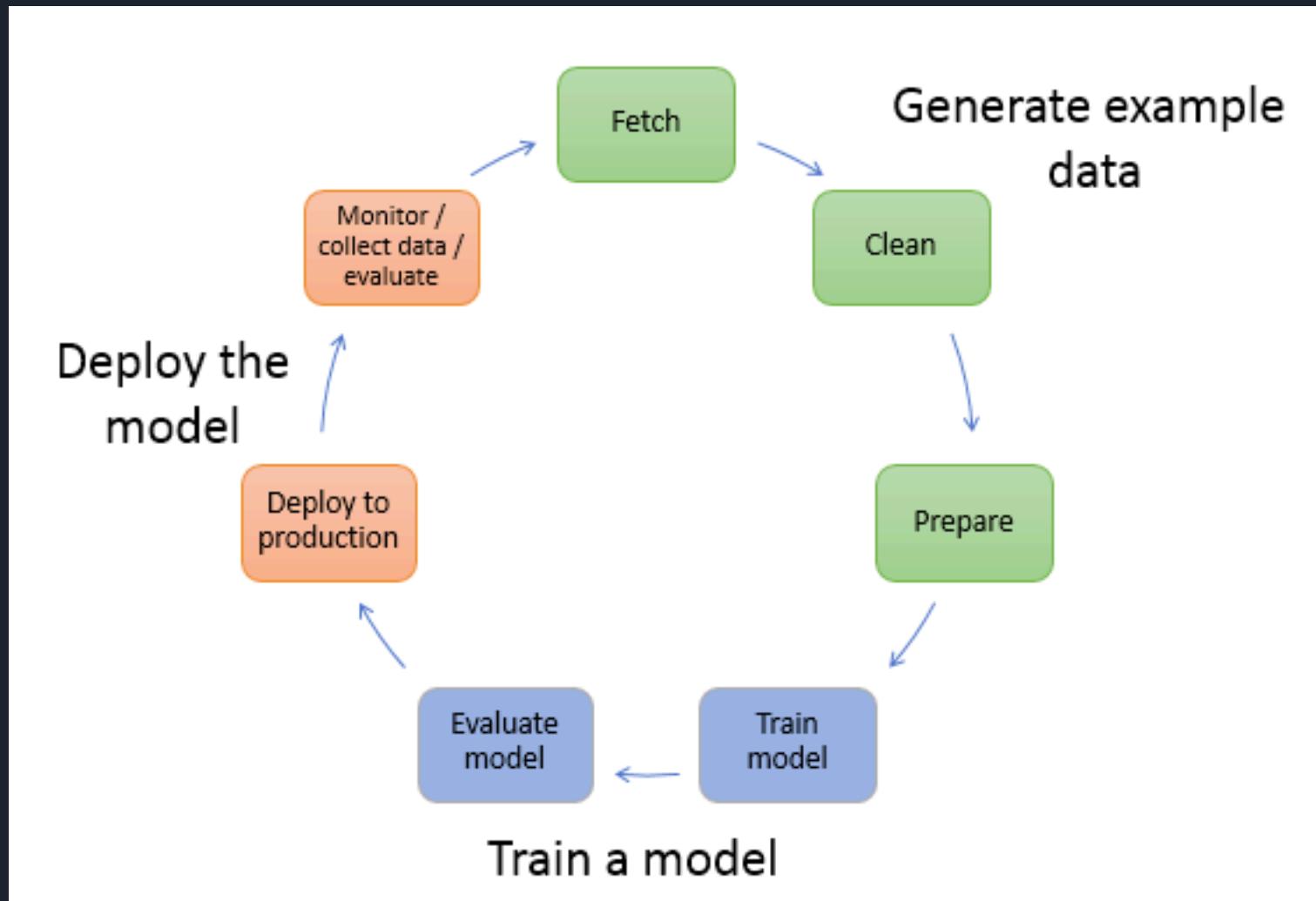




Amazon SageMaker

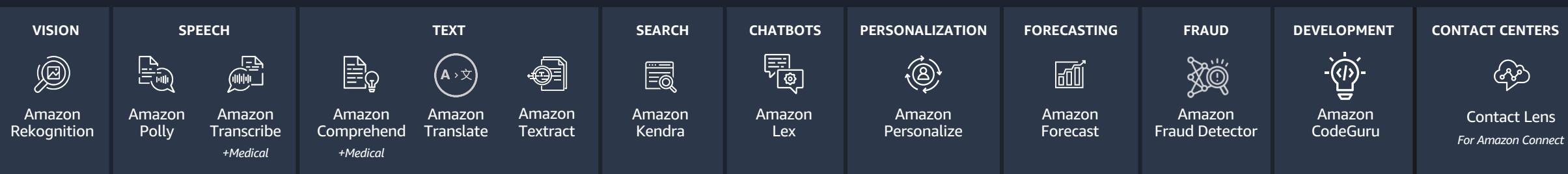
ML workflow for a machine learning model – most common



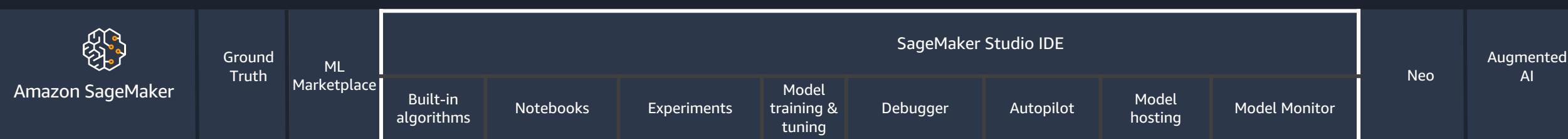
The AWS ML Stack

Broadest and most complete set of Machine Learning capabilities

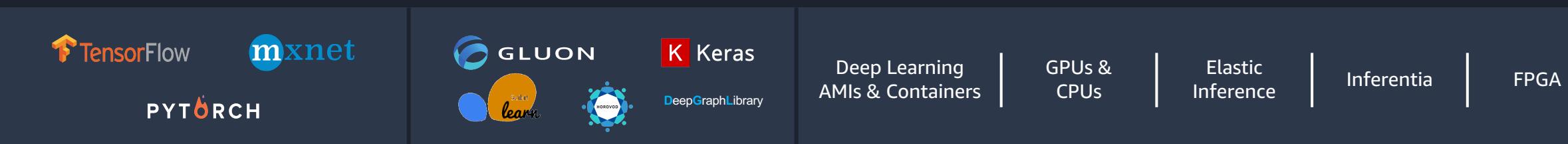
AI SERVICES



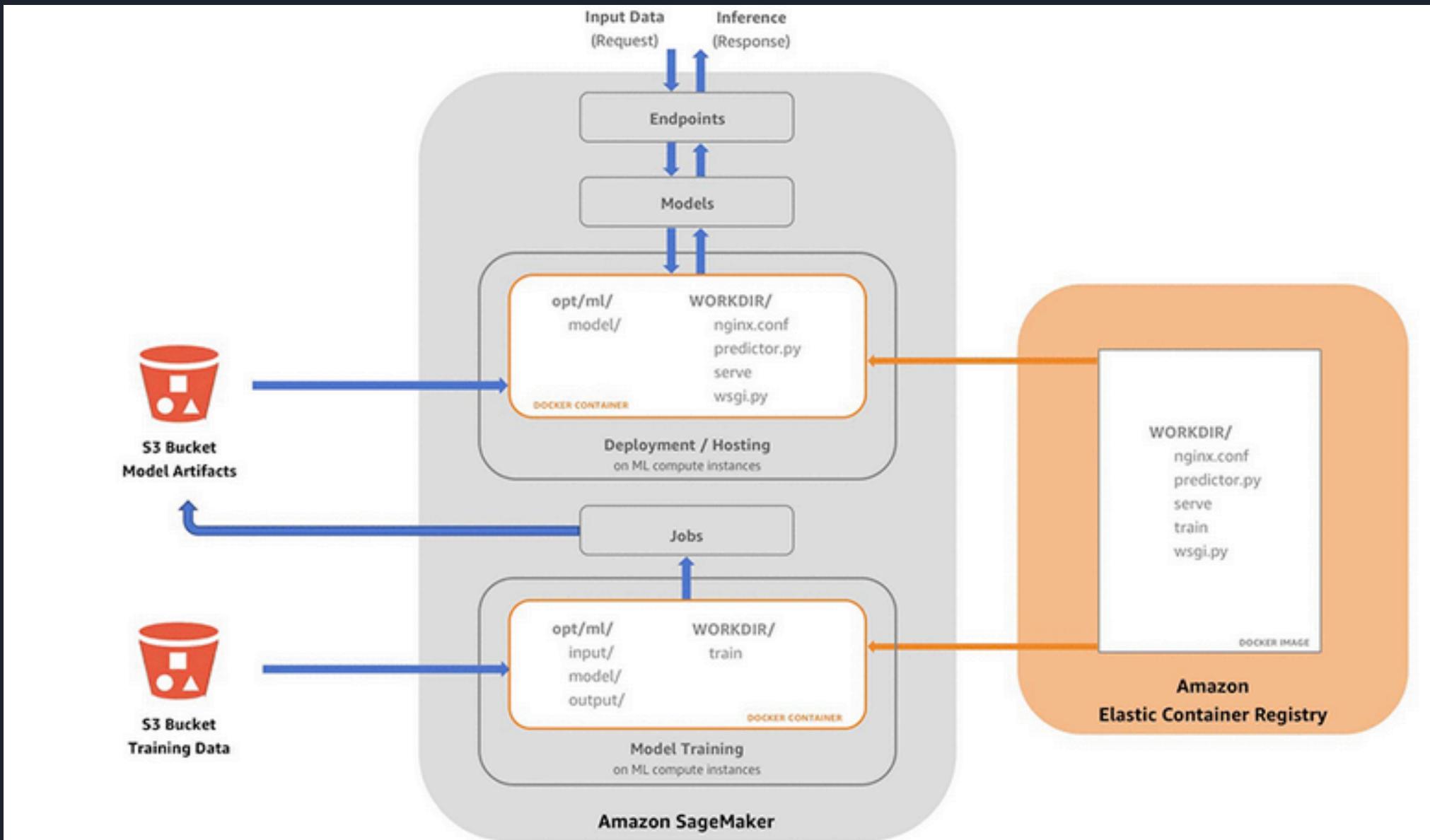
ML SERVICES



ML FRAMEWORKS & INFRASTRUCTURE



ML workflow for a machine learning model – most common



Amazon SageMaker helps you build, train, and deploy models

Prepare

Build

Train & Tune

Deploy & Manage

Automatically build and train models

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

One-click collaborative notebooks and built-in, high performance algorithms and models



Choose or build an ML algorithm

One-click training



Set up and manage environments for training

Debugging and optimization



Train, debug, and tune models

Visually track and compare experiments



Manage training runs

One-click deployment and autoscaling



Deploy model in production

Automatically spot concept drift



Monitor models

Add human review of predictions



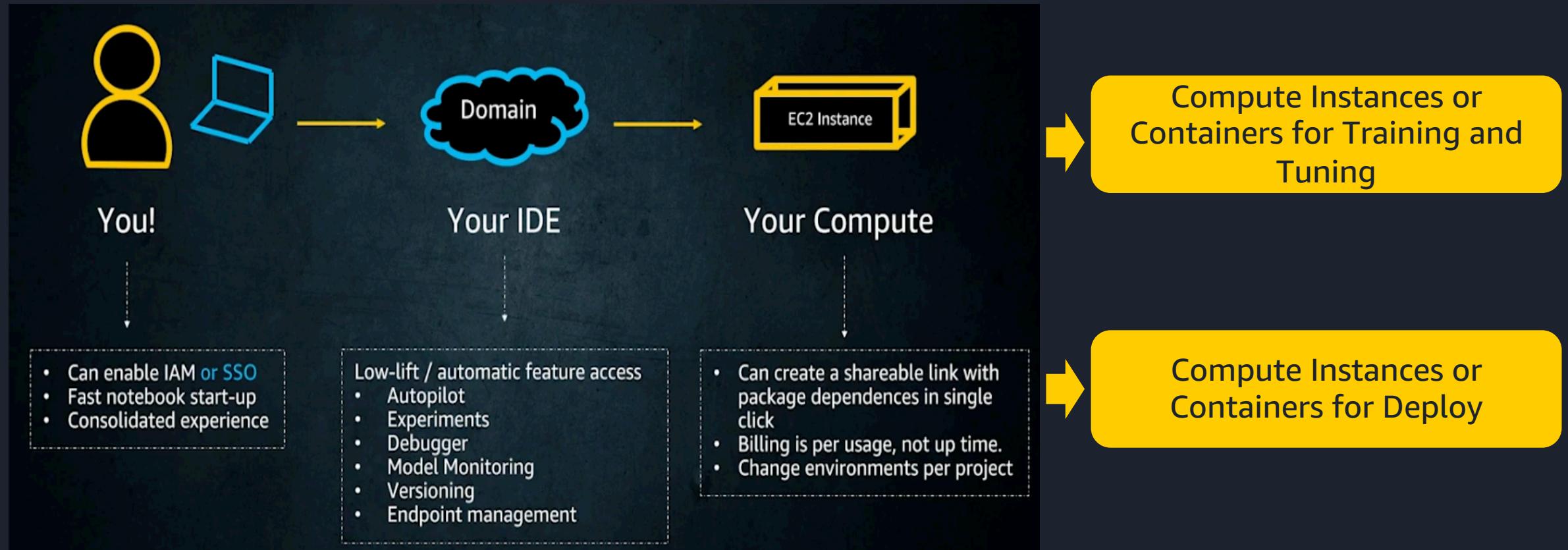
Validate predictions

Fully managed with auto-scaling for 75% less



Scale manage the production environment

Amazon SageMaker Studio



Amazon SageMaker Notebooks

Fast-start sharable notebooks



Easy access with
Single Sign-On (SSO)

Access your notebooks in
seconds



Fully managed
and secure

Administrators manage
access and permissions



Fast setup

Start your notebooks
without spinning up
compute resources



Easy collaboration

Share notebooks
with a single click



Flexible

Dial up or down
compute resources

Amazon SageMaker Autopilot

Automatic model creation 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

Use Amazon SageMaker Autopilot to create and review top performing regression and classification models

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help.
- Sidebar:** Shows a file tree under "/automl-preview/" with three entries: "bank-additional" (12 hours ago), "model" (12 hours ago), and "sagemaker_auto..." (3 hours ago).
- Top Bar:** Three tabs are open: "sagemaker_automl_direct_m;" (highlighted), "my-sagemaker-autopilot" (with a gear icon), and "Deploy model".
- Content Area:**
 - EXPERIMENT: MY-SAGEMAKER-AUTOPILOT:** Buttons for "Open candidate generation notebook" and "Open data exploration notebook".
 - Trials:** Tab selected, showing a table of trials. A "Deploy model" button is located at the bottom right of the table.
 - TRIALS:** Sub-section showing "0 row selected".
 - Table:** Displays completed trials with the following data:

Trial name	Status	Start time	End time	Objective
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9206119775772095
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9202479720115662
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9200050234794617
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9195190072059631
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9191550016403198
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9190340042114258
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9189119935035706
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825

Amazon SageMaker Processing

Analytics jobs for data processing and model evaluation



Fully managed

Achieve distributed processing for clusters



Custom processing

Bring your own script for feature engineering



Container support

Use SageMaker's built-in containers or bring your own



Security and compliance

Leverage SageMaker's security & compliance features



Automatic creation & termination

Your resources are created, configured, & terminated automatically

Amazon SageMaker Automatic Model Tuning

Automatically tune hyperparameters in your algorithms



Tuning at scale

Adjust thousands of
different combinations of
algorithm parameters



Automated

Uses ML to find the best
parameters



Faster

Eliminate days or weeks of
tedious manual work

Examples

Decision Trees

Tree depth

Max leaf nodes

Gamma

Eta

Lambda

Alpha

Neural Networks

Number of layers

Hidden layer width

Learning rate

Embedding dimensions

Dropout

Amazon SageMaker Experiments

Organize, track, and compare training experiments



Tracking at scale



Custom organization



Visualization



Metrics and logging



Fast Iteration

Track parameters & metrics
across experiments & users

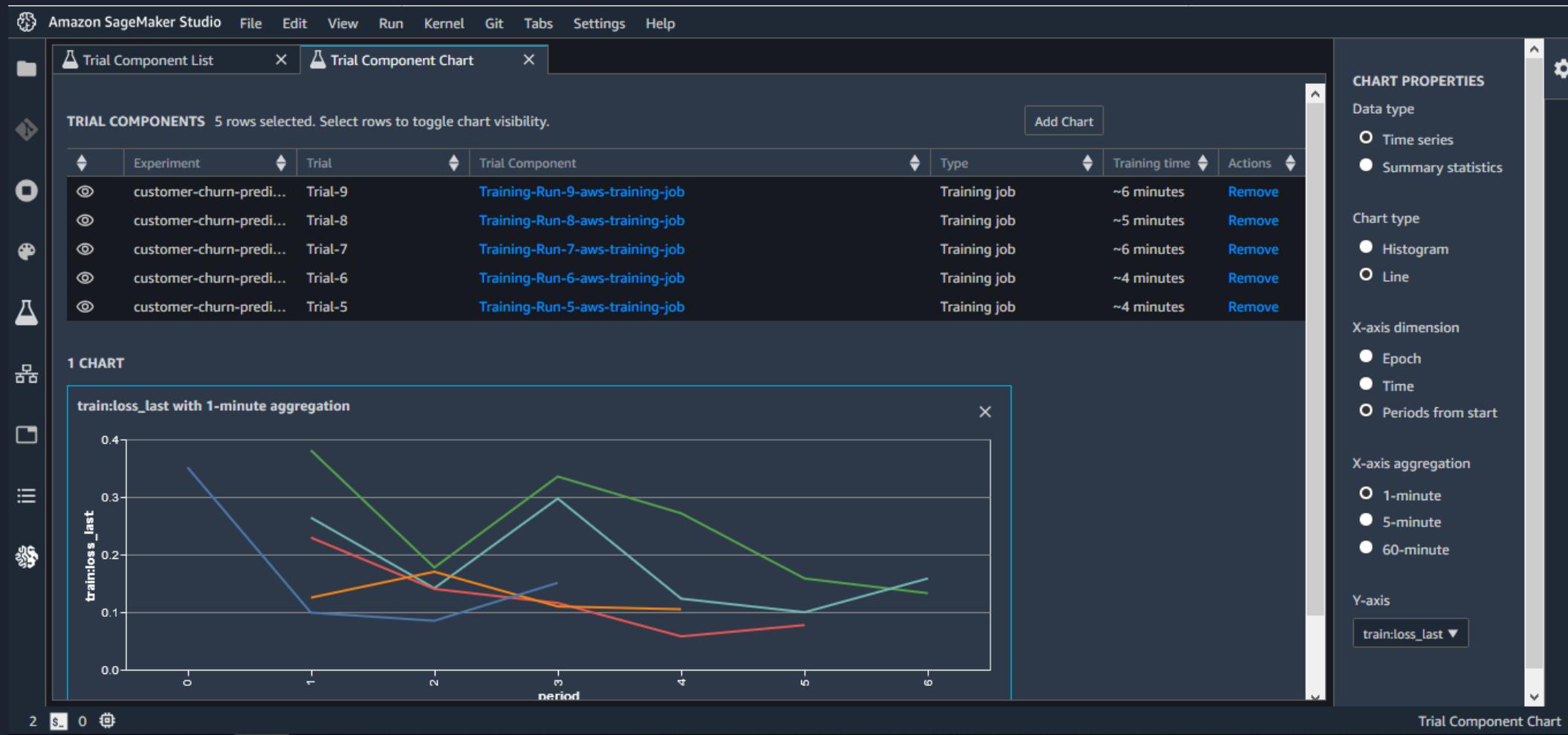
Organize experiments by
teams, goals, & hypotheses

Easily visualize experiments
and compare

Log custom metrics using
the Python SDK & APIs

Quickly go back & forth
& maintain high-quality

Use Amazon SageMaker Experiments to track and manage thousands of experiments



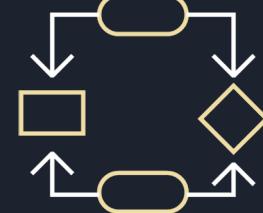
Amazon SageMaker Debugger

Analysis and debugging, explainability, and alert generation



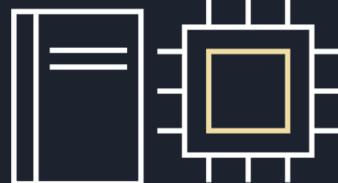
Relevant data capture

Data is automatically captured for analysis



Data analysis & debugging

Analyze & debug data with no code changes



Automatic error detection

Errors are automatically detected based on rules



Improved productivity with alerts

Take corrective action based on alerts



Visual analysis and debugging

Visually analyze & debug from SageMaker Studio

Use Amazon SageMaker Debugger to identify issues such as vanishing gradients

Amazon SageMaker Studio File Edit View Run Kernel Git Tabs Settings Help

SMDebugger-CloudWatch-Logs × Markdown git conda_tensorflow_p36

Using SageMaker Rules

In this example we'll demonstrate how to use SageMaker rules to be evaluated against your training. You can find the list of SageMaker rules and the configurations best suited for using them here.

We specify a few rules that check for overfitting, decrease in loss across epochs and for saturated activations.

```
[8]: estimator = TensorFlow(
    role=sagemaker.get_execution_role(),
    base_job_name='mnist-tensorflow-example',
    train_instance_count=1,
    train_instance_type='ml.p3.2xlarge',
    image_name='cpu_training_image',
    entry_point=entrypoint_script,
    framework_version='1.15',
    py_version='py3',
    train_max_run=3600,
    script_mode=True,
    sagemaker_session=sess,
    ## New parameter
    rules = [ Rule.sagemaker(rule_configs.vanishing_gradient()),
              Rule.custom(name='overfitting', # used to identify the rule
                          image_uri='759209512951.dkr.ecr.us-west-2.amazonaws.com',
                          instance_type='ml.c4.xlarge', # instance type to run the source
                          source=my_custom_rule.py, # path to the rule source file
                          rule_to_invoke='CustomGradientRule', # name of the class
                          volume_size_in_gb=400, # EBS volume size required to be
                          collections_to_save=[CollectionConfig(name='losses')]),
              rule_parameters={
                  "threshold": "20.0" # this will be used to initialize
              } ],
    hyperparameters = {'num_epochs' : 100 }
)
```

Note that Sagemaker-Debugger is only supported for py_version='py3' currently.

Let's start the training by calling `fit()` on the MXNet estimator

```
[9]: # After calling fit, SageMaker will spin off 1 training job and 1 rule job for you
# The rule evaluation status(es) will be visible in the training logs
# at regular intervals

estimator.fit(wait=False)
```

Result

Describe Trial Component

Experiment: Unassigned
Trial: Unassigned

Trial stages

mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-training-job	Status	Last modified	Rule name	Job ARN
Created 15 minutes ago	Issues Found	4 minutes ago	VanishingGradient	arn:aws:sagemaker:us-west-2:3
	Issues Found	4 minutes ago	Overfitting	arn:aws:sagemaker:us-west-2:3

Trial Component Chart

TRIAL COMPONENTS 1 rows selected. Select rows to toggle chart visibility.

Experiment	Trial	Trial Component	Type	Train
N/A	N/A	mnist-tensorflow-example-2019-12-02-09-52-13-126-aws-trainin...	Training job	~10

2 CHARTS

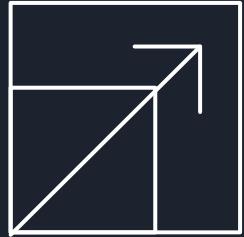
sparse_softmax_cross_entropy_loss/value:0_avg with 1-minute aggregation

period

trialComponentName
— mnist-tensorflow-example-2019-1...

Amazon SageMaker is fully managed

One click model deployment



Auto-scaling



Low latency and
high throughput



Bring your
own model



Python SDK



Deploy multiple
models on an
endpoint

Amazon SageMaker Model Monitor

Continuous monitoring of models in production



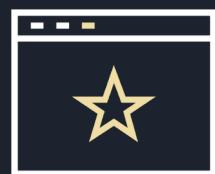
Automatic data collection

Data is automatically collected from your endpoints



Continuous Monitoring

Define a monitoring schedule and detect changes in quality against a pre-defined baseline



Flexibility with rules

Use built-in rules to detect data drift or write your own rules for custom analysis



Visual data analysis

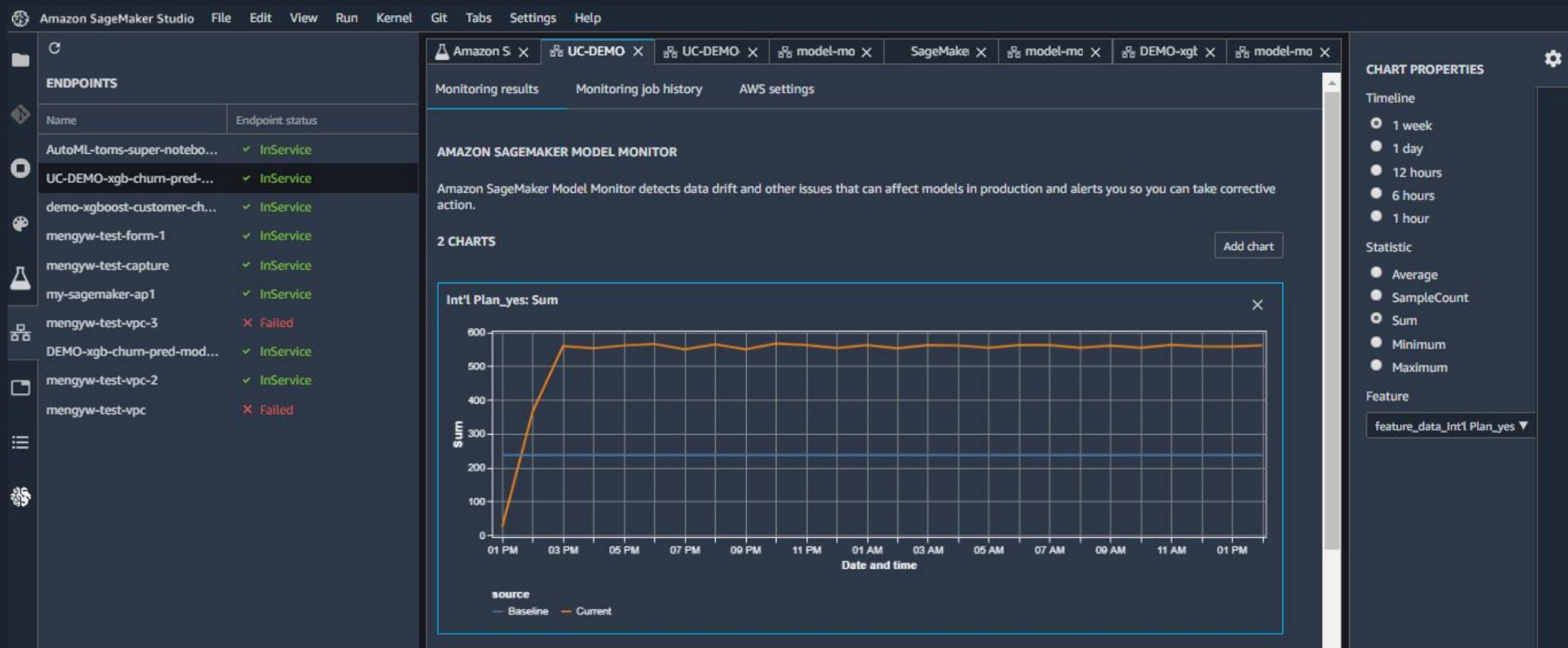
See monitoring results, data statistics, and violation reports in SageMaker Studio



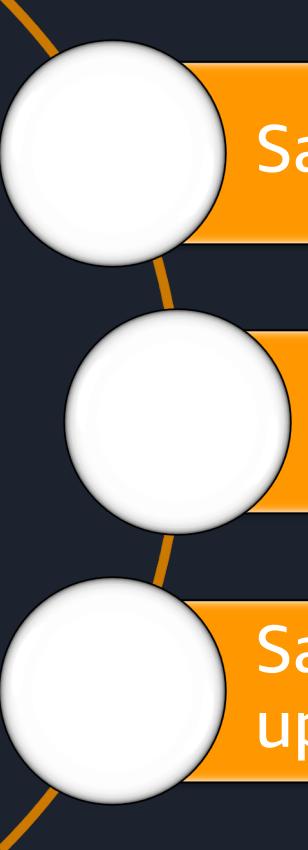
CloudWatch Integration

Automate corrective actions based on Amazon CloudWatch alerts

Use Amazon SageMaker Model Monitor to identify model drift and take action



SageMaker – New Features Released

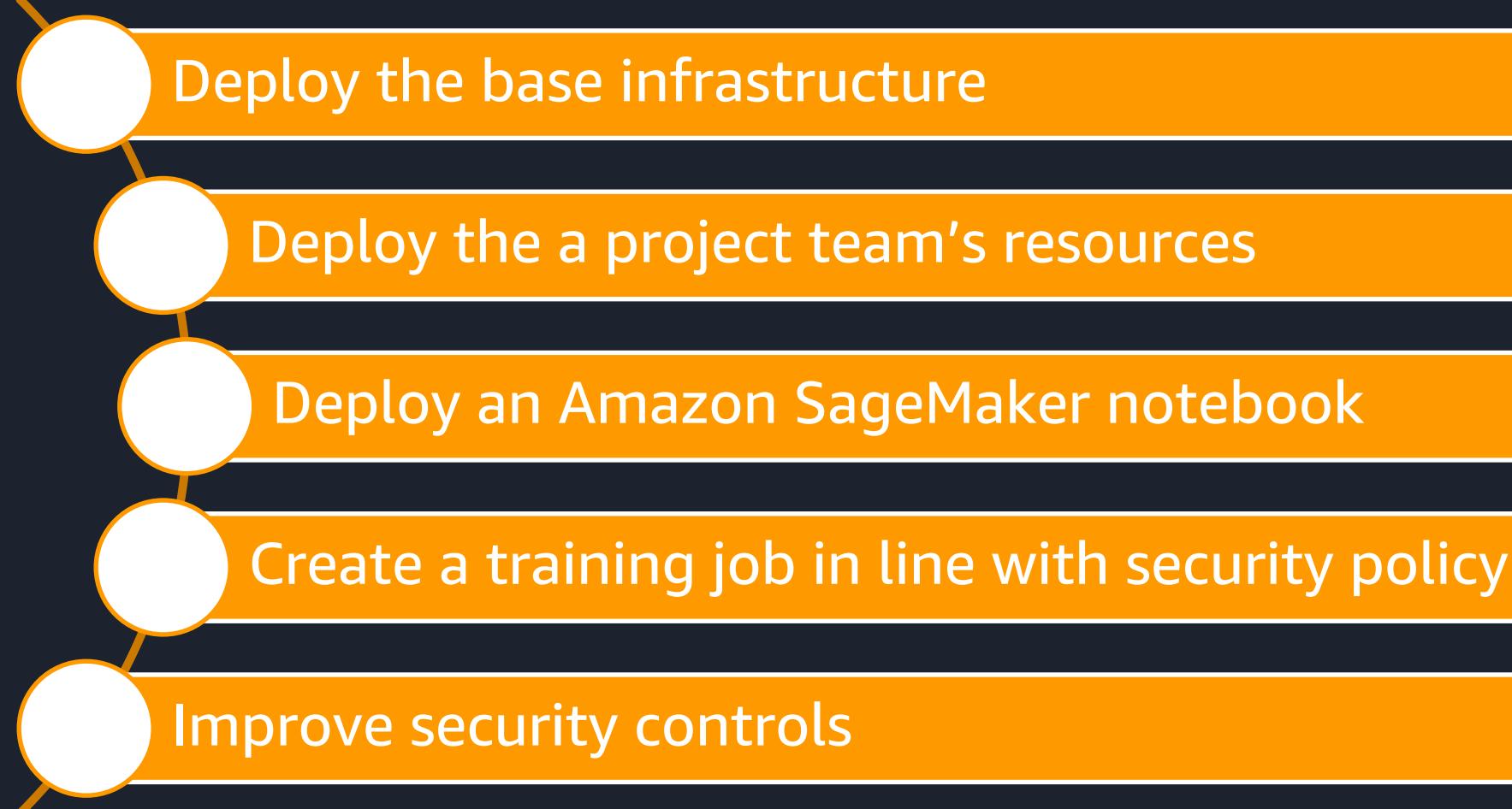


SageMaker Calrify - Detect bias in datasets and in models.

SageMaker Pipelines - First purpose-built CI/CD service for machine learning

SageMaker FeatureStore - purpose-built repository to store, update, retrieve, and share machine learning (ML) features

Personas & ML Operations..



Platform
Engineers



Data
Scientists



Project
Managers



Analyst Community
(Business/ Consumer/
Power Users)

SageMaker Customer success stories

Financial Services – Automated Due Diligence



BUSINESS CHALLENGE

- Automatic Due Diligence
- Support and automate the due diligence process on financial documentation through deep natural language processing



DATA

- Corpus of ~1000 documents similar to the target documents (unlabeled)
- Domain specific Wikipedia data (AWS sourced)
- 2,500 annotated pages (sections of interest highlighted using tagtag, an APN solution)



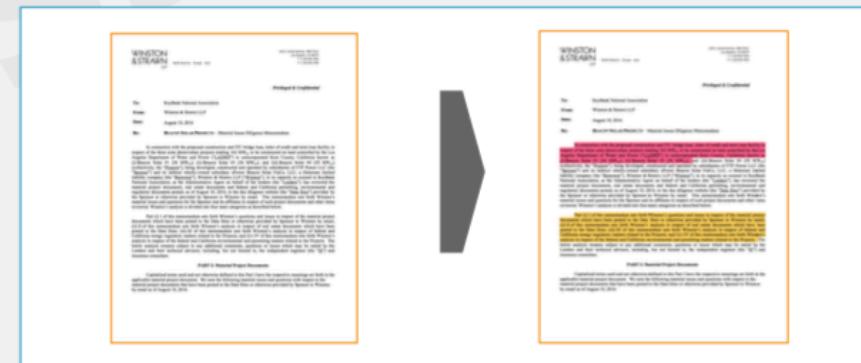
MACHINE LEARNING APPROACH

- Word embedding generation w/ Amazon SageMaker BlazingText
- Recurrent neural network sentence classifier custom built on Amazon SageMaker
- SageMaker integrated frontend based on tagtag to serve predictions natively in pdf and allow users to provide feedback



OUTCOMES

- Reduced document due diligence process by 50% (from 8h to 4h)
- End-to-end natural language processing (NLP) platform to
 - Annotate PDF documents
 - Train deep learning NLP models
 - Provide feedback on machine generated annotations in an easy-to-use user interface to improve the models over time
- Re:Invent session with customer



CONTACT

erikessl@, johbruba@, arkajyo@, jarvisle@

© 2020, Amazon Web Services, Inc. or its Affiliates.

PUBLIC REFERENCE

PRIVATE REFERENCE

LOGO ONLY REFERENCE





THOMSON REUTERS



PROBLEM

Thomson Reuters wanted to enable ML at scale

Data scientists needed secure access to data, while ensuring compliance with Thomson Reuters enterprise security policies

SOLUTION: SECURE CONTENT WORKSPACES

SCW provides access to data in compliance with internal security standards

Amazon SageMaker, enabled through SCW, provides a fully managed machine learning environment

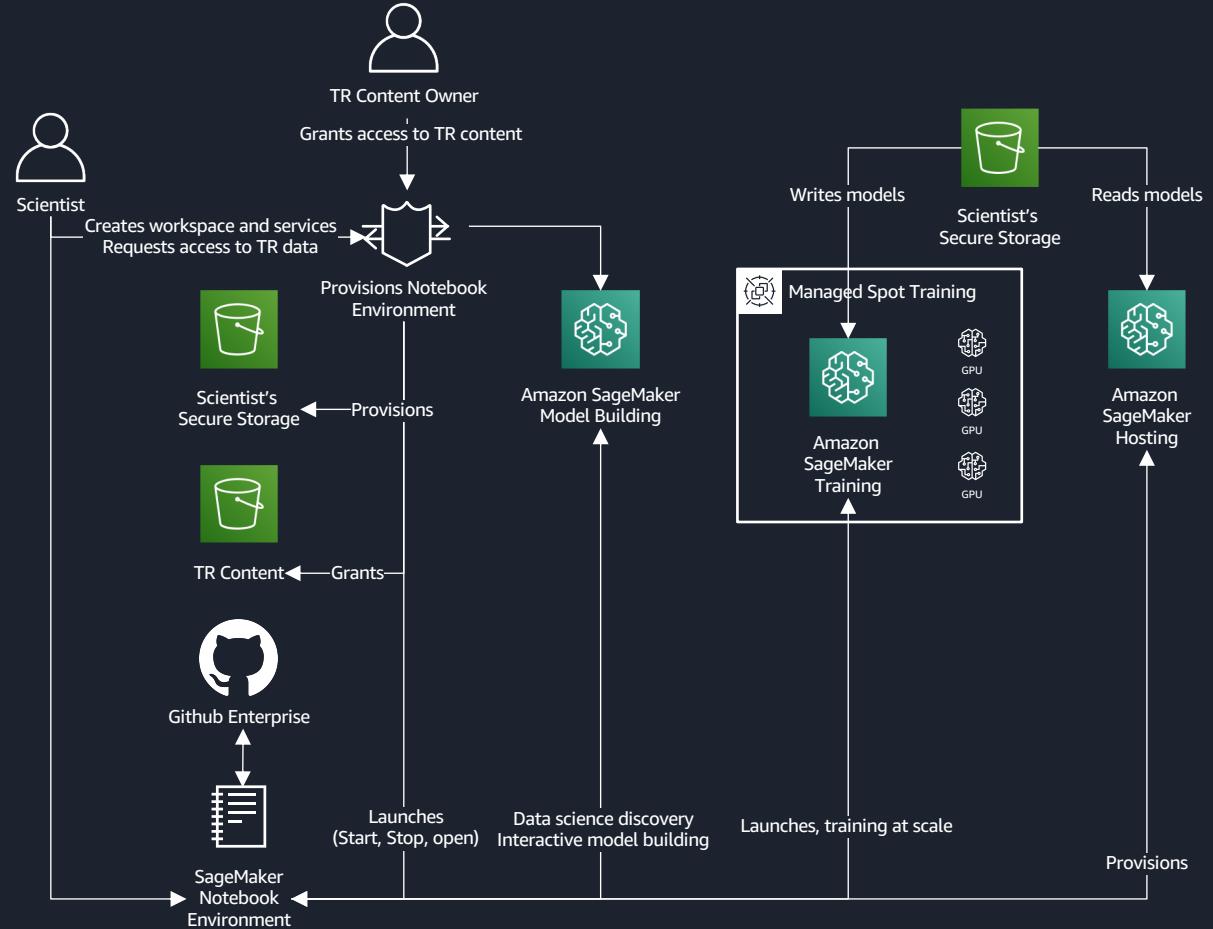
IMPACT

Secure experimentation at scale in compliance with TR policies
Intelligent + timely solutions for their customers

Thomson Reuters solution architecture

SCW with Amazon SageMaker allows ready data access to data with right permissions, security, user roles, etc.

Fully managed access to compute options: CPU, GPU, Spot instances



Demo