



Easily deploy models for the best performance and cost using Amazon SageMaker

Santosh Bhavani

Sr. Product Manager—Technical,
Amazon SageMaker

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



Hosting ML models on SageMaker

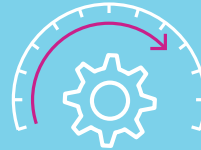


Easily deploy and manage models

Set up an endpoint in minutes to get predictions

Infrastructure management, patching, and built-in updates

Collect metrics and logs for your endpoints in Amazon CloudWatch



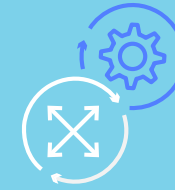
Best price-performance tradeoffs

99.99% service availability SLA

70+ SageMaker ML instances

Autoscaling based on traffic

Deploy multiple (10K+) models on an endpoint for cost savings



Integrated MLOps

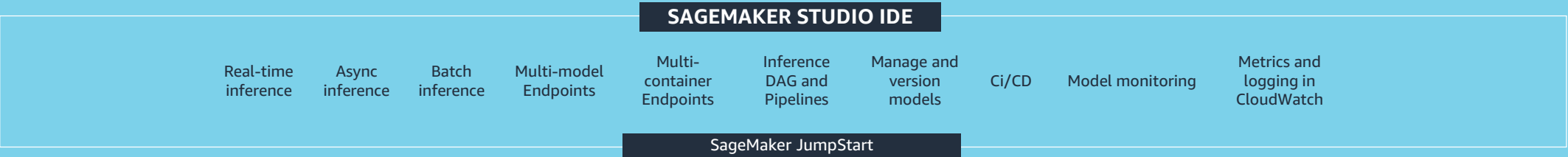
CI/CD: SageMaker Pipelines and Projects

Model Registry: Catalog models, versioning, approval workflows

Model Monitor: Alerts on data and model drift

SageMaker Inference ML stack

Amazon SageMaker



FRAMEWORKS



MODEL SERVERS

AWS Deep Learning Containers	TensorFlow Serving	TorchServe	NVIDIA Triton Inference Server	AWS Multi Model Server (MMS)
------------------------------	--------------------	------------	--------------------------------	------------------------------

ML COMPUTE INSTANCES

CPU	GPU	Inferentia	Graviton (ARM)
-----	-----	------------	----------------

ACCELERATORS

SageMaker Neo	NVIDIA TensorRT/cuDNN	Intel oneDNN	ARM Compute Library
---------------	-----------------------	--------------	---------------------

Optimizing inference takes skills, time, and effort



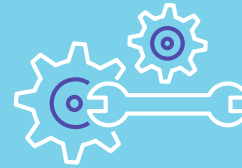
70+ ML instance types

Selecting the right instance type based on resource requirements of the ML model and data payloads



Systems for ML

Selecting the right instance size, container parameters, and auto-scaling properties to maximize performance



Model tuning

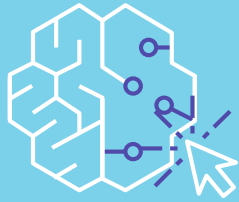
Using ML frameworks with converters, compilers, and kernel libraries specific to different instance types and hardware vendors



Manual benchmarking

Performance and load testing to validate latency and throughput requirements are met and costs are within budget

Introducing SageMaker Inference Recommender



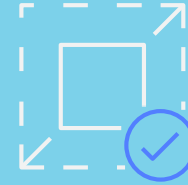
Instance recommendations

Instance type recommendation for initial deployments



Load tests

Run extensive load tests that include production requirements—throughput, latency



Endpoint recommendations

Get endpoint configuration settings that meet your production requirements

Designed for ML Engineers and Data Scientists to reduce time to get models into production

Get started with Inference Recommender

1



Container image

2



Model artifacts and
sample payload

3



Model metadata



Model Registry



Inference Recommender



Get initial instance
recommendations

Specify performance requirements
and instance types for a custom load
test

View and compare performance and
cost across different endpoint
configurations



**Deploy
your model**



Instance recommendations



Python SDK

Get instance type recommendations for your ML models right from your Jupyter Notebook



Integrated with Model Registry

Store your model metadata and get instance type recommendations for all your registered models



Review recommendations

Review key performance metrics from Studio and deploy your model in a few clicks

Get an instance recommendation in minutes

Model version

Version 4

pyt-cpu-models-1632186494

Status
Approved

Model group
pyt-cpu-models-16321...

Update status

Activity

Metrics

Inference Recommender

Load test

Settings

Instance recommendations for getting started

Get initial instance recommendations that deliver the best price performance based on your model and sample payloads. Deploy to one of the recommended instance types or run a custom load test.

EC2	Est. cost/hour	MaximumInvocations	ModelLatency	
ml.inf1.xlarge	\$0.05	1100	23.5 ms	Create endpoint
ml.g4dn.8xlarge	\$0.15	1100	23.5 ms	Create endpoint
ml.c5.9xlarge	\$0.18	1100	23.5 ms	Create endpoint
ml.g4dn.2xlarge	\$0.27	1100	23.5 ms	Create endpoint
ml.c5.9xlarge	\$0.29	1100	23.5 ms	Create endpoint

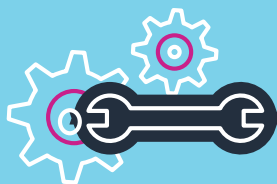


Load tests



Customize your load tests

Customize your load tests by providing production requirements (throughput and latency), traffic pattern, and instance types



Tune your model and container

Fine-tune your model, model server and containers by sweeping through different environment variable values (e.g., number of threads)



Review performance metrics

Review latency, throughput, and cost across different endpoint configurations or get detailed metrics from CloudWatch

Run custom load tests across instance types

Inference Recommender Jobs

Inference Recommender helps customers determine the EC2 instance types and initial count, inference container parameter tuning and model optimizations that provide the best inference performance at the lowest cost.

Create

Job Name	Status	Created
torchvision-yolo4-2021-03-26...	Complete	15 days ago
torchvision-yolo4-2021-03-26...	Complete	20 days ago
torchvision-yolo4-2021-03-26...	Complete	21 days ago

Create inference recommender job

Easily compare the performance of a model across various instance types such as CPU, GPU and Inferentia. To get started, select a model, provide performance requirements such as latency and throughput, upload a sample payload, and finally select and configure instance types for load testing. [Learn More](#)

✓ Model selection ✓ Job settings ● Instance selection

Selected instances

Instances for benchmarking
Select all instances and set environment variables for load testing.

+ Add instances to test

EC2	Price per hour		
m1.xlarge	\$0.05	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.15	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.18	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.27	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.29	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.25	<button>Env. variables</button>	<button>Delete</button>
m1.xlarge	\$0.64	<button>Env. variables</button>	<button>Delete</button>

Additional settings - optional

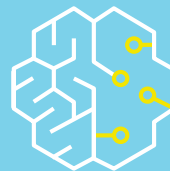
Max number of tests ?

Max parallel tests ?

Back Submit



Endpoint recommendations



Get instance type and count

Provides both instance type and initial instance count that can support your production requirements



Optimize your model and container

Recommends model optimizations and container parameter settings to improve performance



Deploy to production

Integrated with Studio—easy to compare endpoint configurations and create an endpoint in a few clicks

Review endpoint recommendations

Results

Details

Deployment goals & recommendations

Deployment goal importance

Select the dropdowns below to adjust deployment goal importance.

Cost

Moderate importance

Latency

Moderate importance

Throughput

Moderate importance

SageMaker recommendation

ml.inf1.xlarge

Create endpoint

Estimated Cost

\$0.19 / hour

ModelLatency

2.41s

MaximumInvocations

32.5

Instance count

1

\$0.0022 / inference

All runs

Search column name to filter runs

EC2	Instances	Est. cost/hour	Cost/inference	MaximumInvocations	ModelLatency
ml.inf1.xlarge	1	\$12.34	\$0.0022	32.5	28.1ms
ml.m5.8xlarge	2	\$68.93	\$0.0022	46	28.1ms
ml.c5.9xlarge	1	\$103.26	\$0.0022	62	28.1ms
ml.g4dn.2xlarge	1	\$106.72	\$0.0022	32	28.1ms
ml.g4dn.8xlarge	3	\$107.62	\$0.0022	12	28.1ms

Deploy ML models into production faster



Loka

Loka is a Silicon Valley full-stack consultancy accelerating AIML, DevOps, and Big Data projects for Fintech and HCLS customers. Since 2004, they've helped startups funded by leading VCs nail their designs, builds, and deadlines.

"At Loka, part of our job is to make sure our customers have ML environments that are performant and scalable, yet cost effective. Between optimizing models, tuning servers, and testing instance types for customer deployments, we spend a huge amount of time and energy making sure we make the right choices. With Inference Recommender, our ML Engineers are able to get an ML model deployed to production within minutes from any location."

—Bobby Mukherjee, CEO at Loka

Improve data scientist productivity



Holmusk

Holmusk is a Singapore-based data science and health technology company that aims to reverse chronic disease and behavioral health issues. Holmusk launched its FoodDX app to help people improve their diet and health.

"Our food image recognition algorithms need low latency to ensure our users get the right diet recommendations at the right time. Using Inference Recommender, we can easily conduct load tests across different instances and determine an instance configuration within hours to reduce our compute costs significantly while maintaining latency requirements. This is a huge productivity win for our team and lets our ML scientists focus on creating algorithms to help people live healthier lives rather than managing infrastructure."

—Subra, CTO at Holmusk

Boost ML model performance

Eko

Eko, a cardiopulmonary digital health company, is elevating the way clinicians detect and monitor heart and lung disease with its innovative suite of digital tools, patient and provider software, and AI-powered analysis.

“To provide real-time disease detection, every second matters. With the ML model optimizations suggested by SageMaker Inference Recommender, we could speed up our model predictions by 20%.”

—Daniel Barbosa,
ML Engineer at Eko

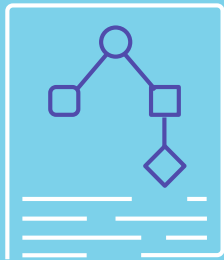


Demo



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

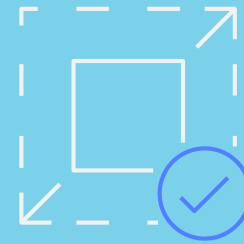
Inference Recommender



**Instance
recommendations**



Load tests



**Endpoint
recommendations**

Designed for ML Engineers and Data Scientists to reduce time to get models into production

Inference Recommender pricing and availability

General
availability

Available in
all commercial
regions where
SageMaker is
available except KIX

Accessible via
Studio, AWS SDK for
Python (Boto3),
AWS CLI

Service is free, but
you pay for instance
usage during testing



Thank you!