Reduce model deployment times with Amazon SageMaker Inference Recommender

Michael Lin

Senior Solutions Architect
AWS

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 trade-offs

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

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

Model tuning

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

Systems for ML

Selecting the right instance size, container parameters, and autoscaling properties to maximize performance

Manual benchmarking

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

Introducing SageMaker Inference Recommender

FIRST PERFORMANCE TESTING SERVICE FOR MACHINE LEARNING

Automate testing and optimizing model performance to help select an endpoint that delivers the best performance at the lowest cost

aws

Inference Recommender

FEATURES

Instance recommendations

Instance type recommendation for initial deployments

Load tests

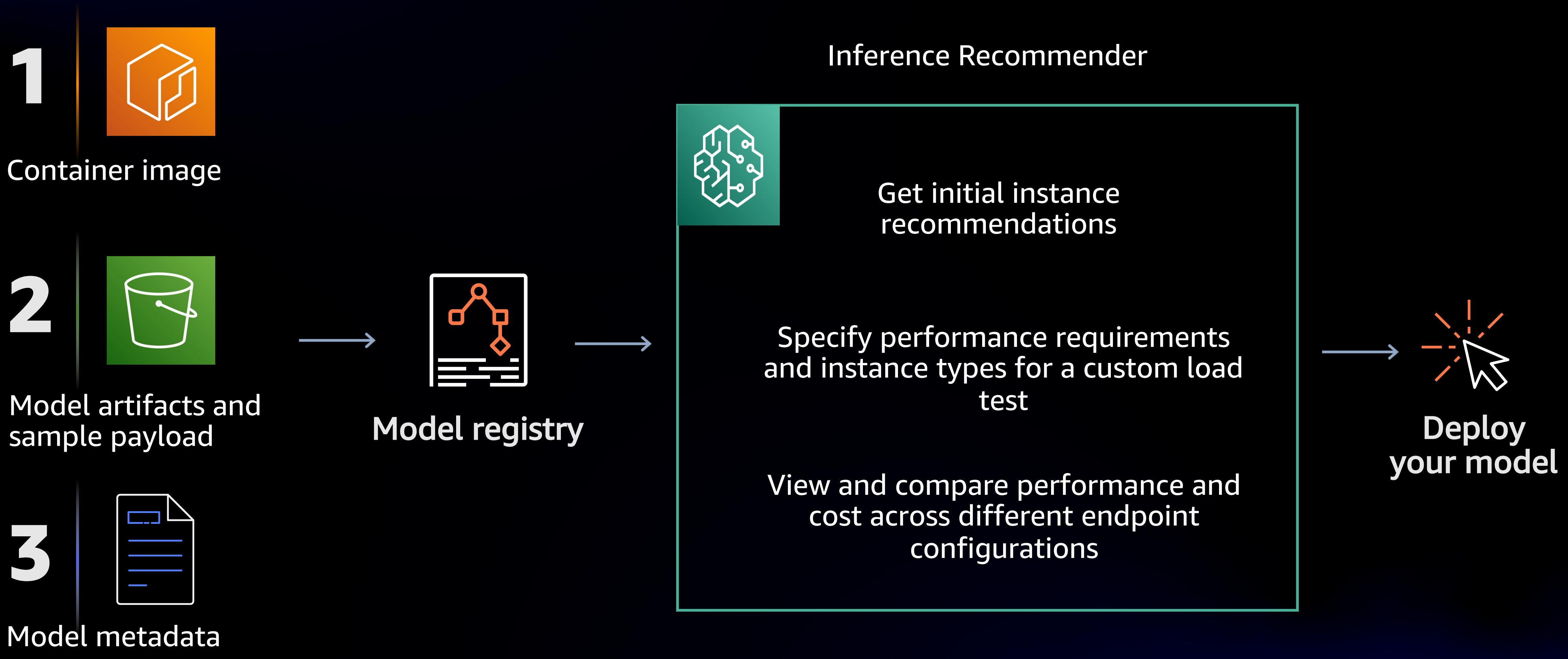
Run extensive load tests that include production requirements – throughput, latency

Endpoint recommendations

Get endpoint configuration settings that meet your production requirements

Designed for MLOps 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

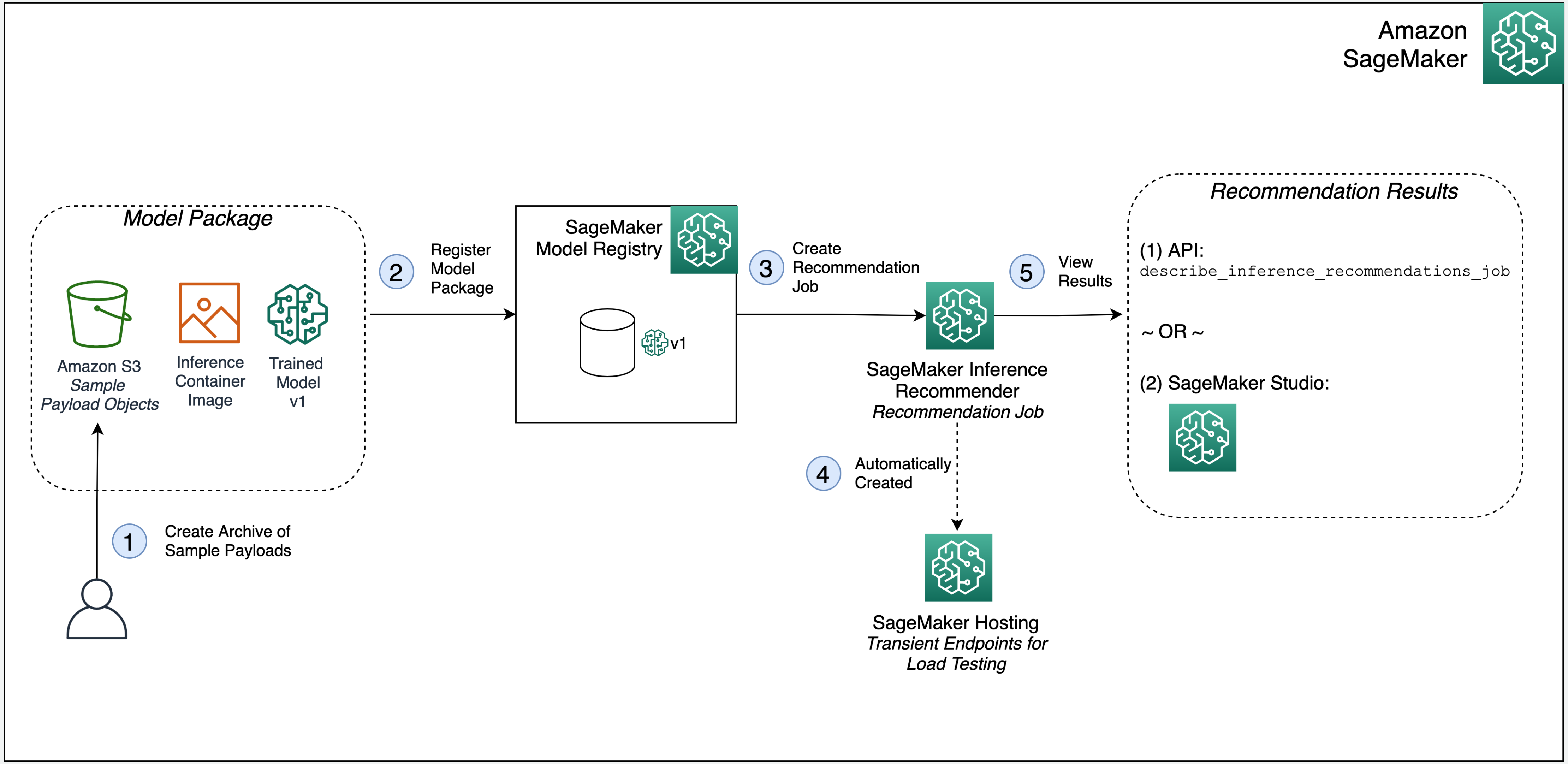
Get initial instance recommendations

Specify performance requirements and instance types for a custom load test

View and compare performance and cost across different endpoint configurations

Deploy your model

Activity diagram & reference architecture



aws machine learning

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 SageMaker Studio and deploy your model in a few clicks

aws

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

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 SageMaker Studio – easy to compare endpoint configurations and create an endpoint in a few clicks

User Experience Overview

AWS Python SDK (boto3)

CLI

SageMaker Studio

Output Results

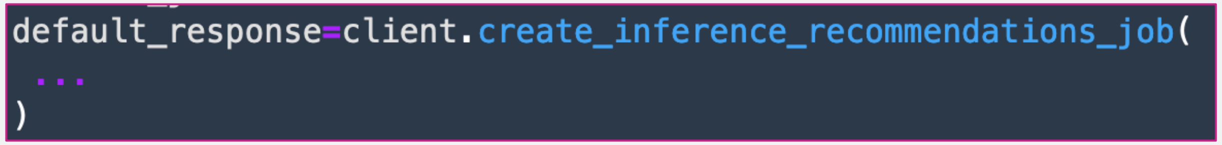
Recommended instance type(s)

Maximum throughput & latency

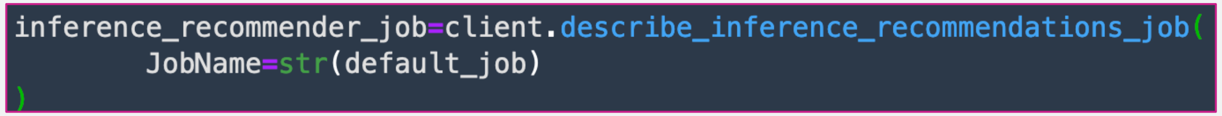
Price-per-inference calculation

Link to documentation:
https://docs.aws.amazon.com/sagemaker/latest/dg/inference-recommender-instance-recommendation.html

Boto3 →

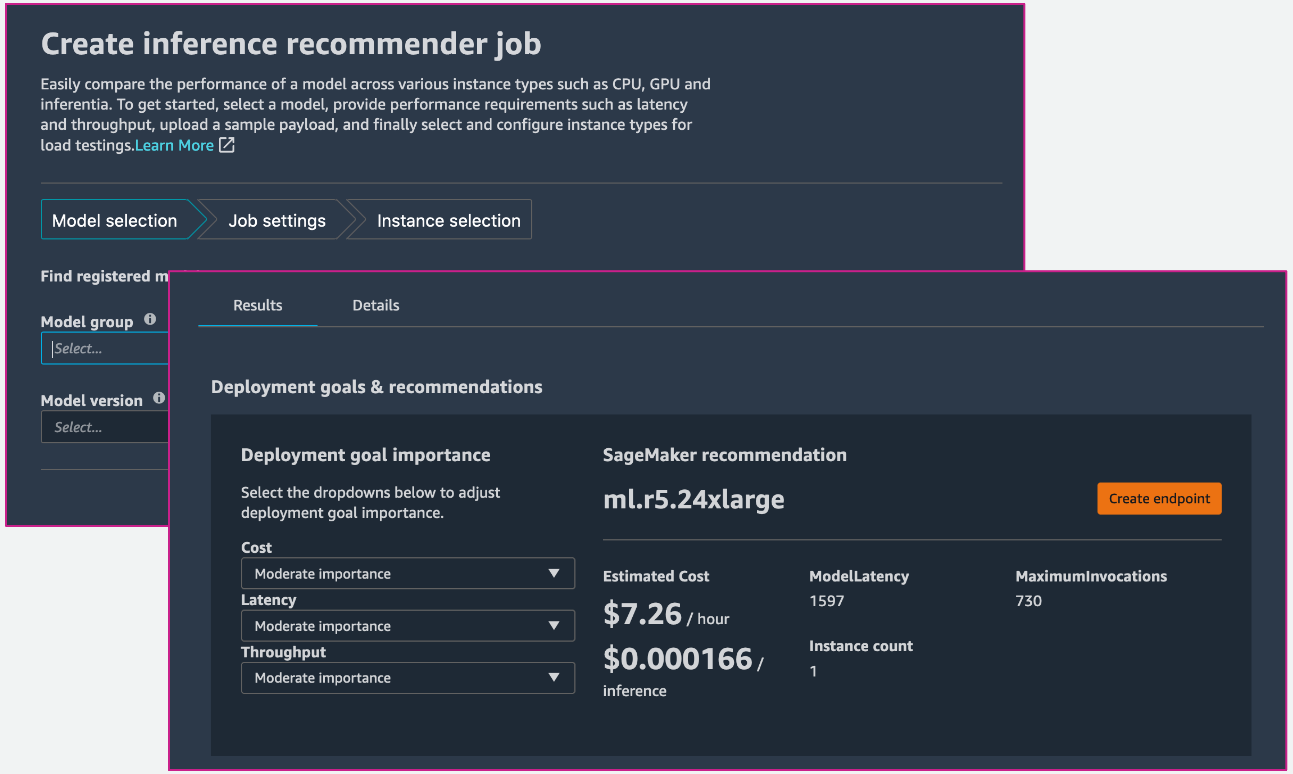


```
default_response=client.create_inference_recommendations_job(
  ...
)
```



```
inference_recommender_job=client.describe_inference_recommendations_job(
        JobName=str(default_job)
)
```

Studio →



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 testings.Learn More

Find registered m

Model group: Select...

Model version: Select...

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.r5.24xlarge

Estimated Cost: $7.26 / hour; $0.000166 / inference

ModelLatency: 1597

Instance count: 1

MaximumInvocations: 730

Best Practices

for Inference Recommender

Use for instance right sizing before deployment

Utilize inference recommender for loading testing to right-size instances prior to deployment.

Use advanced recommendation jobs to conduct custom load tests

While default jobs will give baseline recommendations, advanced recommendation jobs will improve the accuracy of your recommendations.

Use Inference Recommender to estimate hosting costs

Utilize inference recommender for a more accurate estimate of SageMaker hosting costs.

Implement SageMaker Model Registry as part of your model build workflow

Standardize your model build workflows to register candidate models for deployment into Model Registry for easy integration with Inference Recommender.

Thank you!

Michael Lin

linmicht@amazon.com