Deploy models for inference

PDF | RSS

With Amazon SageMaker, you can deploy your machine learning (ML) models to make predictions, also known as *inference*. SageMaker provides a broad selection of ML infrastructure and model deployment options to help meet all your ML inference needs. It is a fully managed service and integrates with MLOps tools, so you can scale your model deployment, reduce inference costs, manage models more effectively in production, and reduce operational burden.

After you've built and trained a machine learning model, you can use SageMaker Inference to start getting predictions, or *inferences*, from your model. With SageMaker Inference, you can either set up an endpoint that returns inferences or run batch inferences from your model.

To get started with SageMaker Inference, see the following sections and review the Inference options to determine which feature best fits your use case.

You can refer to the Resources section for more troubleshooting and reference information, blogs and examples to help you get started, and common FAQs.

Topics

Before you begin

These topics assume that you have built and trained one or more machine learning models and are ready to deploy them. You don't need to train your model in SageMaker in order to deploy your model in SageMaker and get inferences. If you don't have your own model, you can also use SageMaker's built-in algorithms or pre-trained models.

If you are new to SageMaker and haven't picked out a model to deploy, work through the steps in the Get Started with Amazon SageMaker tutorial to familiarize yourself with an example of how SageMaker manages the data science process and how it handles model deployment. For more information about training a model, see Train Models.

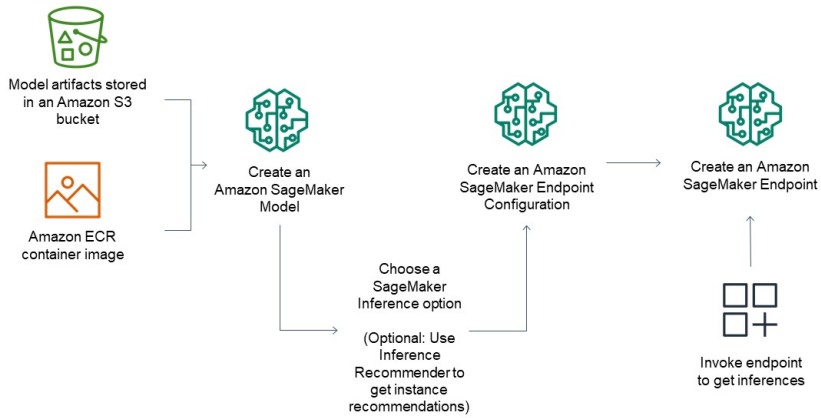
For additional information, reference, and examples, see the Resources.

Steps for model deployment

For inference endpoints, the general workflow consists of the following:

- Create a model in SageMaker Inference by pointing to model artifacts stored in Amazon S3 and a container image.
- Select an inference option. For more information, see Inference options.
- Create a SageMaker Inference endpoint configuration by choosing the instance type and number of instances you need behind the endpoint. You can use Amazon SageMaker Inference Recommender to get recommendations for instance types. For Serverless Inference, you only need to provide the memory configuration you need based on your model size.
- Create a SageMaker Inference endpoint.
- Invoke your endpoint to receive an inference as a response.

The following diagram shows the preceding workflow.



You can perform these actions using the AWS console, the AWS SDKs, the SageMaker Python SDK, AWS CloudFormation or the AWS CLI.

For batch inference with batch transform, point to your model artifacts and input data and create a batch inference job. Instead of hosting an endpoint for inference, SageMaker outputs your inferences to an Amazon S3 location of your choice.

Inference options

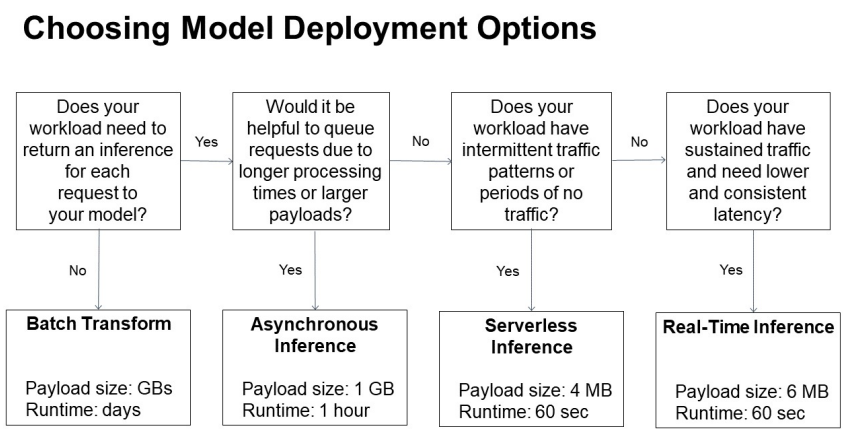
SageMaker provides multiple inference options so that you can pick the option that best suits your workload:

- Real-Time Inference: *Real-time inference* is ideal for online inferences that have low latency or high throughput requirements. Use real-time inference for a persistent and fully managed endpoint (REST API) that can handle sustained traffic, backed by the instance type of your choice. Real-time inference can support payload sizes up to 6 MB and processing times of 60 seconds.
- Serverless Inference: *Serverless inference* is ideal when you have intermittent or unpredictable traffic patterns. SageMaker manages all of the underlying infrastructure, so there's no need to manage instances or scaling policies. You pay only for what you use and not for idle time. It can support payload sizes up to 4 MB and processing times up

to 60 seconds.

- Batch Transform: *Batch transform* is suitable for offline processing when large amounts of data are available upfront and you don't need a persistent endpoint. You can also use batch transform for pre-processing datasets. It can support large datasets that are GBs in size and processing times of days.
- Asynchronous Inference: *Asynchronous inference* is ideal when you want to queue requests and have large payloads with long processing times. Asynchronous Inference can support payloads up to 1 GB and long processing times up to one hour. You can also scale down your endpoint to 0 when there are no requests to process.

The following diagram shows the preceding information in a flowchart and can help you choose the option that best fits your use case.



Advanced endpoint options

With real-time inference, you can further optimize for performance and cost with the following advanced inference options:

- If you have multiple models that use the same framework and can share a container, then use Host multiple models in one container behind one endpoint. This option helps

you optimize costs by improving endpoint utilization and reducing deployment overhead.

- If you have multiple models that use different frameworks and require their own containers, then use Host multiple models which use different containers behind one endpoint. With this option, you get many of the benefits of Multi-Model Endpoints and can deploy a variety of frameworks and models.
- If you want to host models with pre-processing and post-processing logic behind an endpoint, then use Serial Inference Pipelines. Inference pipelines are fully managed by SageMaker and provide lower latency because all of the containers are hosted on the same Amazon EC2 instances.

Bring your own model

To use an existing Docker container in SageMaker, see Adapting your own Docker container to work with SageMaker.

To create a new Docker container and receive more advanced guidance on how to run your own inference code, see the following links.

- To run your own inference code hosting services, see Use Your Own Inference Code with Hosting Services.
- To run your own inference code for batch inference, see Use Your Own Inference Code with Batch Transform.

Next steps

After you have an endpoint and understand the general inference workflow, you can use the following features within SageMaker Inference to improve your inference workflow.

Monitoring

To track your model over time through metrics such as model accuracy and drift, you can use Model Monitor. With Model Monitor, you can set alerts that notify you when there are deviations in your model's quality. To learn more, see the

Model Monitor documentation. To learn more about tools that can be used to monitor model deployments and events that change your endpoint, see Monitor Amazon SageMaker. For example, you can monitor your endpoint's health through metrics such as invocation errors and model latency using Amazon CloudWatch metrics. The SageMaker endpoint invocation metrics can provide you with valuable information about your endpoint's performance.

CI/CD for model deployment

To put together machine learning solutions in SageMaker, you can use SageMaker MLOps. You can use this feature to automate the steps in your machine learning workflow and practice CI/CD. You can use MLOps Project Templates to help with the setup and implementation of SageMaker MLOps projects. SageMaker also supports using your own third-party Git repo for creating a CI/CD system.

For your ML pipelines, use Model Registry to manage your model versions and the deployment and automation of your models.

Deployment guardrails

If you want to update your model while it's in production without impacting production, you can use deployment guardrails. Deployment guardrails are a set of model deployment options in SageMaker Inference to update your machine learning models in production. Using the fully managed deployment options, you can control the switch from the current model in production to a new one. Traffic shifting modes give you granular control over the traffic shifting process, and built-in safeguards like auto-rollbacks help you catch issues early on. To learn more about deployment guardrails, see the deployment guardrails documentation.

Inferentia

If you need to run large scale machine learning and deep learning applications for use cases such as image or speech recognition, natural language processing (NLP),

personalization, forecasting, or fraud detection, you can use an `Inf1` instance with a real-time endpoint.

`Inf1` instances are built to support machine learning inference applications and feature the AWS Inferentia chips. `Inf1` instances provide higher throughput and lower cost per inference than GPU-based instances.

To deploy a model on `Inf1` instances, compile your model with SageMaker Neo and choose an `Inf1` instance for your deployment option. To learn more, see Optimize model performance using SageMaker Neo.

Optimize model performance

SageMaker provides features to manage resources and optimize inference performance when deploying machine learning models. You can use SageMaker's built-in algorithms and pre-built models, as well as prebuilt Docker images, which are developed for machine learning. To train TensorFlow, Apache MXNet, PyTorch, ONNX, and XGBoost models once and optimize them to deploy on ARM, Intel, and Nvidia processors, see Optimize model performance using SageMaker Neo.

Autoscaling

If you have varying amounts of traffic to your endpoints, you might want to try autoscaling. For example, during peak hours, you might require more instances to process requests, but during periods of low traffic, you might want to reduce your use of computing resources. To dynamically adjust the number of instances provisioned in response to changes in your workload, see Automatically Scale Amazon SageMaker Models.

If you have unpredictable traffic patterns or don't want to set up scaling policies, you can also use Serverless Inference for an endpoint where SageMaker manages autoscaling for you. During periods of low traffic, SageMaker scales down your endpoint, and if traffic increases, then SageMaker scales your endpoint up. For more information, see the Serverless Inference documentation.

Did this page help you?

Yes | No

Provide feedback

Next topic: Model Deployment

Previous topic: Use Checkpoints

Need help?

- Try AWS re:Post
- Connect with an AWS IQ expert