Host models along with pre-processing logic as serial inference pipeline behind one endpoint

PDF (/pdfs/sagemaker/latest/dg/sagemaker-dg.pdf#inference-pipelines) | **RSS (amazon-sagemaker-release-notes.rss)**

An *inference pipeline* is a Amazon SageMaker model that is composed of a linear sequence of two to fifteen containers that process requests for inferences on data. You use an inference pipeline to define and deploy any combination of pretrained SageMaker built-in algorithms and your own custom algorithms packaged in Docker containers. You can use an inference pipeline to combine preprocessing, predictions, and post-processing data science tasks. Inference pipelines are fully managed.

You can add SageMaker Spark ML Serving and scikit-learn containers that reuse the data transformers developed for training models. The entire assembled inference pipeline can be considered as a SageMaker model that you can use to make either real-time predictions or to process batch transforms directly without any external preprocessing.

Within an inference pipeline model, SageMaker handles invocations as a sequence of HTTP requests. The first container in the pipeline handles the initial request, then the intermediate response is sent as a request to the second container, and so on, for each container in the pipeline. SageMaker returns the final response to the client.

When you deploy the pipeline model, SageMaker installs and runs all of the containers on each Amazon

Elastic Compute Cloud (Amazon EC2) instance in the endpoint or transform job. Feature processing and inferences run with low latency because the containers are co-located on the same EC2 instances. You define the containers for a pipeline model using the `CreateModel` (https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_CreateModel.html) operation or from the console. Instead of setting one `PrimaryContainer`, you use the `Containers` parameter. to set the containers that make up the pipeline You also specify the order in which the containers are executed.

A pipeline model is immutable, but you can update an inference pipeline by deploying a new one using the `UpdateEndpoint` (https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_UpdateEndpoint.html) operation. This modularity supports greater flexibility during experimentation.

For information on how to create an inference pipeline with the SageMaker model registry, see Register and Deploy Models with Model Registry (./model-registry.html) .

There are no additional costs for using this feature. You pay only for the instances running on an endpoint.

Topics

Sample Notebooks for Inference Pipelines

For an example that shows how to create and deploy inference pipelines, see the Inference Pipeline with Scikit-learn and Linear Learner (https://github.com/aws/amazon-sagemaker-examples/tree/main/sagemaker-python-sdk/scikit_learn_inference_pipeline) sample notebook. For instructions on creating and accessing Jupyter notebook instances that you can use to run the example in SageMaker, see Amazon SageMaker Notebook Instances (./nbi.html) .

To see a list of all the SageMaker samples, after creating and opening a notebook instance, choose the **SageMaker Examples** tab. There are three inference pipeline notebooks. The first two inference pipeline notebooks just described are located in the `advanced_functionality` folder and the third notebook is in the `sagemaker-python-sdk` folder. To open a notebook, choose its **Use** tab, then choose **Create copy**.

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.