

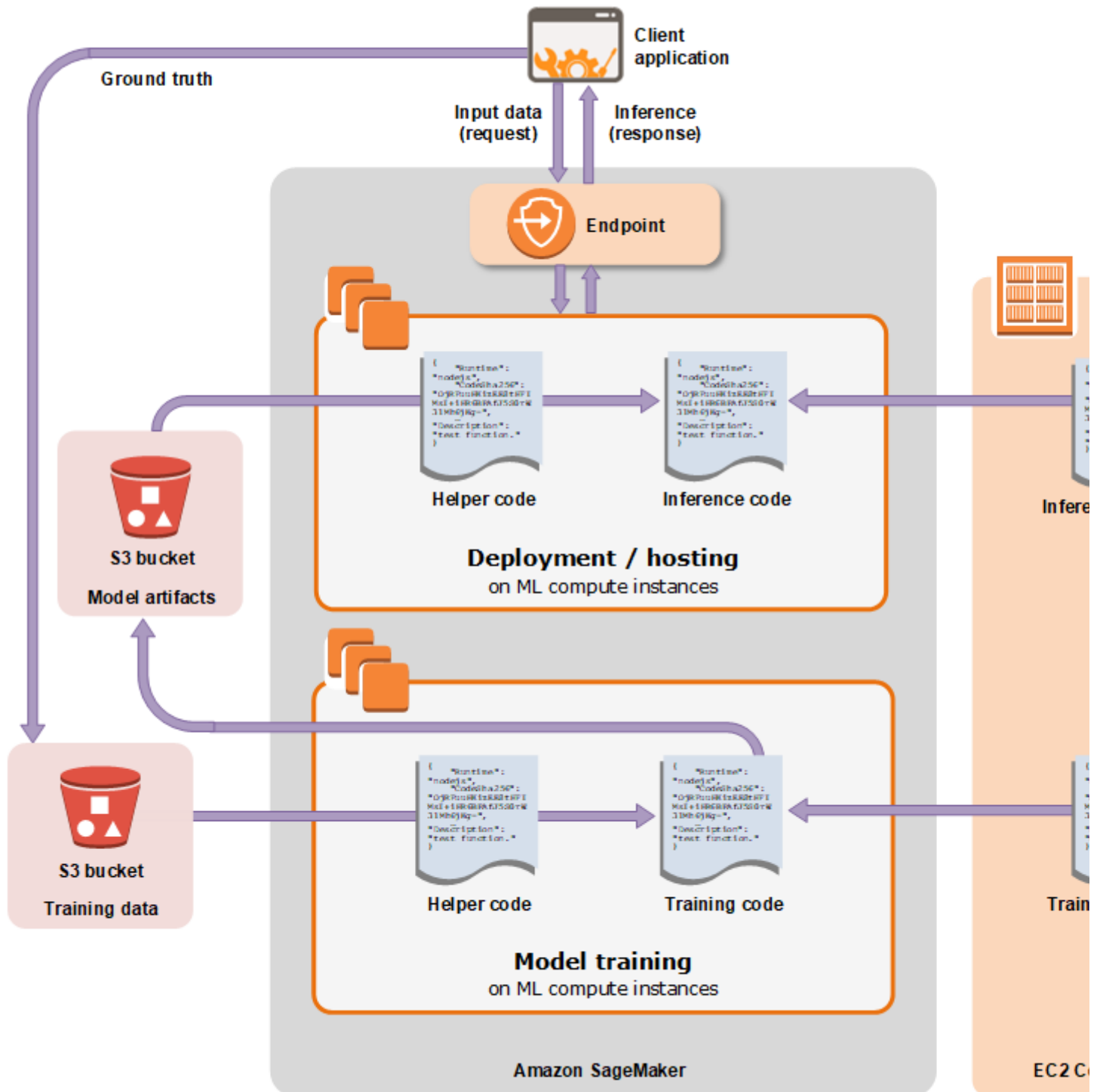
Train a Model with Amazon SageMaker

PDF (sagemaker-dg.pdf#how-it-works-training)

Kindle (<https://www.amazon.com/dp/B07JVSBS9J>)

RSS (amazon-sagemaker-release-notes.rss)

The following diagram shows how you train and deploy a model with Amazon SageMaker:



The area labeled SageMaker highlights the two components of SageMaker: model training and model deployment.

To train a model in SageMaker, you create a training job. The training job includes the following information:

- The URL of the Amazon Simple Storage Service (Amazon S3) bucket where you've stored the training data.
- The compute resources that you want SageMaker to use for model training. Compute resources are ML compute instances that are managed by SageMaker.

- The URL of the S3 bucket where you want to store the output of the job.
- The Amazon Elastic Container Registry path where the training code is stored. For more information, see [Docker Registry Paths and Example Code \(./sagemaker-algo-docker-registry-paths.html\)](#) .

You have the following options for a training algorithm:

- **Use an algorithm provided by SageMaker**—SageMaker provides training algorithms. If one of these meets your needs, it's a great out-of-the-box solution for quick model training. For a list of algorithms provided by SageMaker, see [Use Amazon SageMaker Built-in Algorithms \(./algorithms.html\)](#) . To try an exercise that uses an algorithm provided by SageMaker, see [Get Started with Amazon SageMaker \(./gs.html\)](#) .
- **Use SageMaker Debugger**—to inspect training parameters and data throughout the training process when working with the TensorFlow, PyTorch, and Apache MXNet learning frameworks or the XGBoost algorithm. Debugger automatically detects and alerts users to commonly occurring errors such as parameter values getting too large or small. For more information about using Debugger, see [Amazon SageMaker Debugger \(./train-debugger.html\)](#) . Debugger sample notebooks are available at [Amazon SageMaker Debugger Samples](https://github.com/aws-labs/amazon-sagemaker-examples/tree/master/sagemaker-debugger) [\[https://github.com/aws-labs/amazon-sagemaker-examples/tree/master/sagemaker-debugger\]](https://github.com/aws-labs/amazon-sagemaker-examples/tree/master/sagemaker-debugger) .
- **Use Apache Spark with SageMaker**—SageMaker provides a library that you can use in Apache Spark to train models with SageMaker. Using the library provided by SageMaker is similar to using Apache Spark MLlib. For more information, see [Use Apache Spark with Amazon SageMaker \(./apache-spark.html\)](#) .
- **Submit custom code to train with deep learning frameworks**—You can submit custom Python code that uses TensorFlow, PyTorch, or Apache MXNet for model training. For more information, see [Use TensorFlow with Amazon SageMaker \(./tf.html\)](#) , [Use PyTorch with Amazon SageMaker \(./pytorch.html\)](#) , and [Use Apache MXNet with Amazon SageMaker \(./mxnet.html\)](#) .
- **Use your own custom algorithms**—Put your code together as a Docker image and specify the registry path of the image in a SageMaker CreateTrainingJob API call. For more information, see [Using Docker containers with SageMaker \(./docker-containers.html\)](#) .
- **Use an algorithm that you subscribe to from AWS Marketplace**—For information, see [Find and Subscribe to Algorithms and Model Packages on AWS Marketplace \(./sagemaker-mkt-find-subscribe.html\)](#) .

After you create the training job, SageMaker launches the ML compute instances and uses the training code and the training dataset to train the model. It saves the resulting model artifacts and other output in the S3 bucket you specified for that purpose.

You can create a training job with the SageMaker console or the API. For information about creating a training job with the API, see the [CreateTrainingJob \(https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_CreateTrainingJob.html\)](#) API.

When you create a training job with the API, SageMaker replicates the entire dataset on ML compute instances by default. To make SageMaker replicate a subset of the data on each ML compute instance, you must set the `S3DataDistributionType` field to `ShardedByS3Key`. You can set this field using the low-level SDK. For more information, see `S3DataDistributionType` in [S3DataSource \(https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_S3DataSource.html\)](#) .

Important

To prevent your algorithm container from contending for memory, we reserve memory for our SageMaker critical system processes on your ML compute instances and therefore you cannot expect to see all the memory for your instance type.

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.