Use Docker containers to build models

PDF | RSS

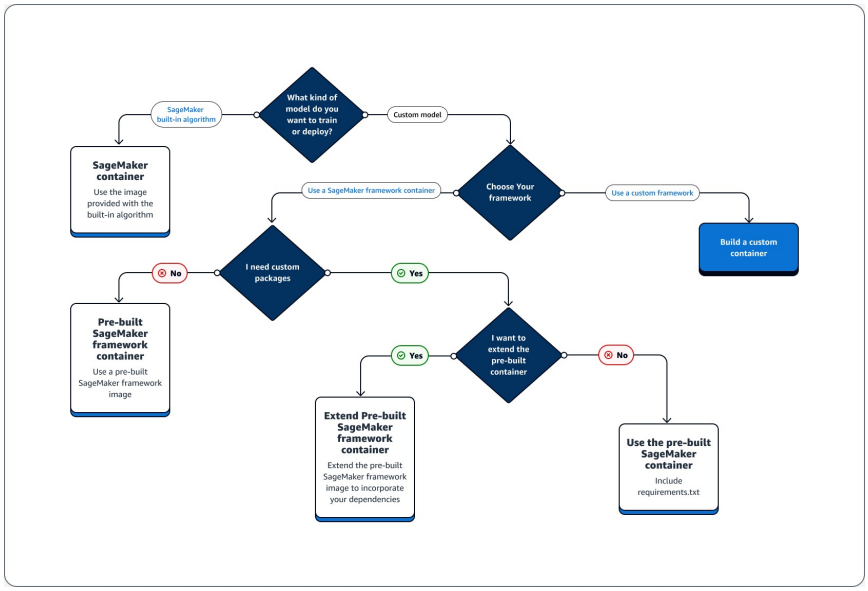
Amazon SageMaker makes extensive use of *Docker containers* for build and runtime tasks. SageMaker provides pre-built Docker images for its built-in algorithms and the supported deep learning frameworks used for training and inference. Using containers, you can train machine learning algorithms and deploy models quickly and reliably at any scale. The topics in this section show how to deploy these containers for your own use cases. For information about how to bring your own containers for use with Amazon SageMaker Studio Classic, see Bring your own SageMaker image.

Topics

Scenarios for Running Scripts, Training Algorithms, or Deploying Models with SageMaker

Amazon SageMaker always uses Docker containers when running scripts, training algorithms, and deploying models. Your level of engagement with containers depends on your use case.

The following decision tree illustrates three main scenarios: **Use cases for using pre-built Docker containers with SageMaker**; **Use cases for extending a pre-built Docker container**; **Use case for building your own container**.



Topics

- Use cases for using pre-built Docker containers with SageMaker
- Use cases for extending a pre-built Docker container
- Use case for building your own container

Use cases for using pre-built Docker containers with SageMaker

Consider the following use cases when using containers with SageMaker:

- **Pre-built SageMaker algorithm** – Use the image that comes with the built-in algorithm. See Use Amazon SageMaker Built-in Algorithms or Pre-trained Models for more information.

- **Custom model with pre-built SageMaker container** – If you train or deploy a custom model, but use a framework that has a pre-built SageMaker container including TensorFlow and PyTorch, choose one of the following options:
    - If you don't need a custom package, and the container already includes all required packages: Use the pre-built Docker image associated with your framework. For more information, see Use Pre-built SageMaker Docker images.
    - If you need a custom package installed into one of the pre-built containers: Confirm that the pre-built Docker image allows a requirements.txt file, or extend the pre-built container based on the following use cases.

Use cases for extending a pre-built Docker container

The following are use cases for extending a pre-built Docker container:

- **You can't import the dependencies** – Extend the pre-built Docker image associated with your framework. See Extend a Pre-built Container for more information.
- **You can't import the dependencies in the pre-built container and the pre-built container supports requirements.txt** – Add all the required dependencies in requirements.txt. The following frameworks support using requirements.txt.
    - TensorFlow
    - Chainer
    - Sci-kit learn
    - PyTorch
    - Apache MXNet

Use case for building your own container

If you build or train a custom model and require custom framework that does not have a pre-built image, build a custom container.

As an example use case of training and deploying a TensorFlow model, the following guide shows how to determine which option from the previous sections of **Use cases** fits to the case.

Assume that you have the following requirements for training and deploying a TensorFlow model.

- A TensorFlow model is a custom model.
- Because a TensorFlow model is going to be built in the TensorFlow framework, use the TensorFlow pre-built framework container to train and host the model.
- If you require custom packages in either your entrypoint script or inference script, either extend the pre-built container or use a requirements.txt file to install dependencies at runtime.

After you determine the type of container that you need, the following list provides details about the previously listed options.

- **Use a built-in SageMaker algorithm or framework**. For most use cases, you can use the built-in algorithms and frameworks without worrying about containers. You can train and deploy these algorithms from the SageMaker console, the AWS Command Line Interface (AWS CLI), a Python notebook, or the Amazon SageMaker Python SDK. You can do that by specifying the algorithm or framework version when creating your Estimator. The available built-in algorithms are itemized and described in the Use Amazon SageMaker Built-in Algorithms or Pre-trained Models topic. For more information about the available frameworks, see ML Frameworks and Languages. For an example of how to train and deploy a built-in algorithm using a Jupyter notebook running in a SageMaker notebook instance, see the Setting up Amazon SageMaker topic.
- **Use pre-built SageMaker container images**. Alternatively, you can use the built-in algorithms and frameworks using Docker containers. SageMaker provides containers for its built-in algorithms and pre-built Docker images for some of the most common machine learning frameworks, such as Apache MXNet, TensorFlow, PyTorch, and Chainer. For a full list of the available SageMaker Images, see Available

Deep Learning Containers Images. It also supports machine learning libraries such as scikit-learn and SparkML. If you use the Amazon SageMaker Python SDK, you can deploy the containers by passing the full container URI to their respective SageMaker SDK `Estimator` class. For the full list of deep learning frameworks that are currently supported by SageMaker, see Prebuilt SageMaker Docker Images for Deep Learning. For information about the scikit-learn and SparkML pre-built container images, see Prebuilt Amazon SageMaker Docker Images for Scikit-learn and Spark ML. For more information about using frameworks with the Amazon SageMaker Python SDK, see their respective topics in Machine Learning Frameworks and Languages.

- **Extend a pre-built SageMaker container image**. If you would like to extend a pre-built SageMaker algorithm or model Docker image, you can modify the SageMaker image to satisfy your needs. For an example, see Extending our PyTorch containers.
- **Adapt an existing container image**: If you would like to adapt a pre-existing container image to work with SageMaker, you must modify the Docker container to enable either the SageMaker Training or Inference toolkit. For an example that shows how to build your own containers to train and host an algorithm, see Bring Your Own R Algorithm.

Next topic: Docker Container Basics

Previous topic: Model Dashboard FAQ