ContainerDefinition

PDF

Describes the container, as part of model definition.

Contents

ContainerHostname

This parameter is ignored for models that contain only a `PrimaryContainer`.

When a `ContainerDefinition` is part of an inference pipeline, the value of the parameter uniquely identifies the container for the purposes of logging and metrics. For information, see Use Logs and Metrics to Monitor an Inference Pipeline. If you don't specify a value for this parameter for a `ContainerDefinition` that is part of an inference pipeline, a unique name is automatically assigned based on the position of the `ContainerDefinition` in the pipeline. If you specify a value for the `ContainerHostName` for any `ContainerDefinition` that is part of an inference pipeline, you must specify a value for the `ContainerHostName` parameter of every `ContainerDefinition` in that pipeline.

Type: String

Length Constraints: Maximum length of 63.

Pattern: `^[a-zA-Z0-9](-*[a-zA-Z0-9]){0,62}`

Required: No

Environment

The environment variables to set in the Docker container.

The maximum length of each key and value in the `Environment` map is 1024 bytes. The maximum length of all keys and values in the map, combined, is 32 KB. If you pass multiple containers to a `CreateModel` request, then the maximum length of all of their maps, combined, is also 32 KB.

Type: String to string map

Map Entries: Maximum number of 100 items.

Key Length Constraints: Maximum length of 1024.

Key Pattern: `[a-zA-Z_][a-zA-Z0-9_]*`

Value Length Constraints: Maximum length of 1024.

Value Pattern: `[\S\s]*`

Required: No

Image

The path where inference code is stored. This can be either in Amazon EC2 Container Registry or in a Docker registry that is accessible from the same VPC that you configure for your endpoint. If you are using your own custom algorithm instead of an algorithm provided by SageMaker, the inference code must meet SageMaker requirements. SageMaker supports both `registry/repository[:tag]` and `registry/repository[@digest]` image path formats. For more information, see Using Your Own Algorithms with Amazon SageMaker.

> **Note**
>
> The model artifacts in an Amazon S3 bucket and the Docker image for inference container in Amazon EC2 Container Registry must be in the same region as the model or endpoint you are creating.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `[\S]+`

Required: No

ImageConfig

Specifies whether the model container is in Amazon ECR or a private Docker registry accessible from your Amazon Virtual Private Cloud (VPC). For information about storing containers in a private Docker registry, see Use a Private Docker Registry for Real-Time Inference Containers.

> **Note**
>
> The model artifacts in an Amazon S3 bucket and the Docker image for inference container in Amazon EC2 Container Registry must be in the same region as the model or endpoint you are creating.

Type: ImageConfig object

Required: No

InferenceSpecificationName

The inference specification name in the model package version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-zA-Z0-9](-*[a-zA-Z0-9]){0,62}$`

Required: No

Mode

Whether the container hosts a single model or multiple models.

Type: String

Valid Values: `SingleModel | MultiModel`

Required: No

ModelDataSource

Specifies the location of ML model data to deploy.

> **Note**
>
> Currently you cannot use `ModelDataSource` in conjunction with SageMaker batch transform, SageMaker serverless endpoints, SageMaker multi-model endpoints, and SageMaker Marketplace.

Type: ModelDataSource object

Required: No

ModelDataUrl

The S3 path where the model artifacts, which result from model training, are stored. This path must point to a single gzip compressed tar archive (.tar.gz suffix). The S3 path is required for SageMaker built-in algorithms, but not if you use your own algorithms. For more information on built-in algorithms, see Common Parameters.

> **Note**
>
> The model artifacts must be in an S3 bucket that is in the same region as the model or endpoint you are creating.

If you provide a value for this parameter, SageMaker uses AWS Security Token Service to download model artifacts from the S3 path you provide. AWS STS is activated in your AWS account by default. If you previously deactivated AWS STS for a region, you need to reactivate AWS STS for that region. For more information, see Activating and Deactivating AWS STS in an AWS Region in the *AWS Identity and Access Management User Guide*.

> **Important**
>
> If you use a built-in algorithm to create a model, SageMaker requires that you provide a S3 path

to the model artifacts in `ModelDataUrl`.

Type: String

Length Constraints: Maximum length of 1024.

Pattern: `^(https|s3)://([^/]+)/?(.*)$`

Required: No

ModelPackageName

The name or Amazon Resource Name (ARN) of the model package to use to create the model.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 176.

Pattern: `(arn:aws[a-z\-]*:sagemaker:[a-z0-9\-]*:[0-9]{12}:[a-z\-]*\/)?([a-zA-Z0-9]([a-zA-Z0-9-]){0,62})(?<!-)(\/[0-9]{1,5})?$`

Required: No

MultiModelConfig

Specifies additional configuration for multi-model endpoints.

Type: MultiModelConfig object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Did this page help you?

Yes No

Provide feedback

Next topic: ContextSource

Previous topic: ContainerConfig

Need help?

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