# Table of Contents

Welcome ............................................................................................................................................. 1  
Actions .................................................................................................................................................. 2  
  CreateCluster .................................................................................................................................. 3  
    Request Syntax ............................................................................................................................... 3  
    Request Parameters ....................................................................................................................... 3  
    Response Syntax ............................................................................................................................ 3  
    Response Elements ........................................................................................................................ 4  
    Errors ........................................................................................................................................... 4  
    Example ........................................................................................................................................ 4  
    See Also ....................................................................................................................................... 5  
  CreateService .................................................................................................................................. 6  
    Request Syntax ............................................................................................................................... 6  
    Request Parameters ....................................................................................................................... 7  
    Response Syntax ............................................................................................................................ 10  
    Response Elements ....................................................................................................................... 11  
    Errors ........................................................................................................................................... 11  
    Example ........................................................................................................................................ 12  
    See Also ....................................................................................................................................... 13  
  DeleteAttributes ............................................................................................................................ 14  
    Request Syntax ............................................................................................................................... 14  
    Request Parameters ....................................................................................................................... 14  
    Response Syntax ............................................................................................................................ 14  
    Response Elements ....................................................................................................................... 15  
    Errors ........................................................................................................................................... 15  
    Example ........................................................................................................................................ 15  
    See Also ....................................................................................................................................... 16  
  DeleteCluster .................................................................................................................................. 17  
    Request Syntax ............................................................................................................................... 17  
    Request Parameters ....................................................................................................................... 17  
    Response Syntax ............................................................................................................................ 17  
    Response Elements ....................................................................................................................... 17  
    Errors ........................................................................................................................................... 18  
    Example ........................................................................................................................................ 18  
    See Also ....................................................................................................................................... 19  
  DeleteService .................................................................................................................................. 21  
    Request Syntax ............................................................................................................................... 21  
    Request Parameters ....................................................................................................................... 21  
    Response Syntax ............................................................................................................................ 21  
    Response Elements ....................................................................................................................... 23  
    Errors ........................................................................................................................................... 23  
    Example ........................................................................................................................................ 23  
    See Also ....................................................................................................................................... 25  
  DeregisterContainerInstance ........................................................................................................... 26  
    Request Syntax ............................................................................................................................... 26  
    Request Parameters ....................................................................................................................... 26  
    Response Syntax ............................................................................................................................ 27  
    Response Elements ....................................................................................................................... 28  
    Errors ........................................................................................................................................... 28  
    Example ........................................................................................................................................ 28  
    See Also ....................................................................................................................................... 31  
  DeregisterTaskDefinition ................................................................................................................ 32  
    Request Syntax ............................................................................................................................... 32  
    Request Parameters ....................................................................................................................... 32  
    Response Syntax ............................................................................................................................ 32  
    Response Elements ....................................................................................................................... 32  
    Errors ........................................................................................................................................... 32  
    Example ........................................................................................................................................ 32  
    See Also ....................................................................................................................................... 33
### Amazon EC2 Container Service API Reference

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>See Also</td>
<td>182</td>
</tr>
<tr>
<td>Device</td>
<td>183</td>
</tr>
<tr>
<td>Contents</td>
<td>183</td>
</tr>
<tr>
<td>See Also</td>
<td>183</td>
</tr>
<tr>
<td>Failure</td>
<td>184</td>
</tr>
<tr>
<td>Contents</td>
<td>184</td>
</tr>
<tr>
<td>See Also</td>
<td>184</td>
</tr>
<tr>
<td>HostEntry</td>
<td>185</td>
</tr>
<tr>
<td>Contents</td>
<td>185</td>
</tr>
<tr>
<td>See Also</td>
<td>185</td>
</tr>
<tr>
<td>HostVolumeProperties</td>
<td>186</td>
</tr>
<tr>
<td>Contents</td>
<td>186</td>
</tr>
<tr>
<td>See Also</td>
<td>186</td>
</tr>
<tr>
<td>KernelCapabilities</td>
<td>187</td>
</tr>
<tr>
<td>Contents</td>
<td>187</td>
</tr>
<tr>
<td>See Also</td>
<td>187</td>
</tr>
<tr>
<td>KeyValuePair</td>
<td>189</td>
</tr>
<tr>
<td>Contents</td>
<td>189</td>
</tr>
<tr>
<td>See Also</td>
<td>189</td>
</tr>
<tr>
<td>LinuxParameters</td>
<td>190</td>
</tr>
<tr>
<td>Contents</td>
<td>190</td>
</tr>
<tr>
<td>See Also</td>
<td>190</td>
</tr>
<tr>
<td>LoadBalancer</td>
<td>191</td>
</tr>
<tr>
<td>Contents</td>
<td>191</td>
</tr>
<tr>
<td>See Also</td>
<td>191</td>
</tr>
<tr>
<td>LogConfiguration</td>
<td>192</td>
</tr>
<tr>
<td>Contents</td>
<td>192</td>
</tr>
<tr>
<td>See Also</td>
<td>192</td>
</tr>
<tr>
<td>MountPoint</td>
<td>193</td>
</tr>
<tr>
<td>Contents</td>
<td>193</td>
</tr>
<tr>
<td>See Also</td>
<td>193</td>
</tr>
<tr>
<td>NetworkBinding</td>
<td>194</td>
</tr>
<tr>
<td>Contents</td>
<td>194</td>
</tr>
<tr>
<td>See Also</td>
<td>194</td>
</tr>
<tr>
<td>NetworkConfiguration</td>
<td>195</td>
</tr>
<tr>
<td>Contents</td>
<td>195</td>
</tr>
<tr>
<td>See Also</td>
<td>195</td>
</tr>
<tr>
<td>NetworkInterface</td>
<td>196</td>
</tr>
<tr>
<td>Contents</td>
<td>196</td>
</tr>
<tr>
<td>See Also</td>
<td>196</td>
</tr>
<tr>
<td>PlacementConstraint</td>
<td>197</td>
</tr>
<tr>
<td>Contents</td>
<td>197</td>
</tr>
<tr>
<td>See Also</td>
<td>197</td>
</tr>
<tr>
<td>PlacementStrategy</td>
<td>198</td>
</tr>
<tr>
<td>Contents</td>
<td>198</td>
</tr>
<tr>
<td>See Also</td>
<td>198</td>
</tr>
<tr>
<td>PortMapping</td>
<td>199</td>
</tr>
<tr>
<td>Contents</td>
<td>199</td>
</tr>
<tr>
<td>See Also</td>
<td>200</td>
</tr>
<tr>
<td>Resource</td>
<td>201</td>
</tr>
<tr>
<td>Contents</td>
<td>201</td>
</tr>
<tr>
<td>See Also</td>
<td>201</td>
</tr>
<tr>
<td>Service</td>
<td>203</td>
</tr>
<tr>
<td>Contents</td>
<td>203</td>
</tr>
<tr>
<td>See Also</td>
<td>205</td>
</tr>
<tr>
<td>ServiceEvent</td>
<td>206</td>
</tr>
<tr>
<td>Contents</td>
<td>206</td>
</tr>
</tbody>
</table>

**API Version 2014-11-13**

viii
<table>
<thead>
<tr>
<th>Contents</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>See Also</td>
<td>206</td>
</tr>
<tr>
<td>Task</td>
<td>207</td>
</tr>
<tr>
<td>Contents</td>
<td>207</td>
</tr>
<tr>
<td>See Also</td>
<td>210</td>
</tr>
<tr>
<td>TaskDefinition</td>
<td>211</td>
</tr>
<tr>
<td>Contents</td>
<td>211</td>
</tr>
<tr>
<td>See Also</td>
<td>214</td>
</tr>
<tr>
<td>TaskDefinitionPlacementConstraint</td>
<td>215</td>
</tr>
<tr>
<td>Contents</td>
<td>215</td>
</tr>
<tr>
<td>See Also</td>
<td>215</td>
</tr>
<tr>
<td>TaskOverride</td>
<td>216</td>
</tr>
<tr>
<td>Contents</td>
<td>216</td>
</tr>
<tr>
<td>See Also</td>
<td>216</td>
</tr>
<tr>
<td>Ulimit</td>
<td>217</td>
</tr>
<tr>
<td>Contents</td>
<td>217</td>
</tr>
<tr>
<td>See Also</td>
<td>217</td>
</tr>
<tr>
<td>VersionInfo</td>
<td>218</td>
</tr>
<tr>
<td>Contents</td>
<td>218</td>
</tr>
<tr>
<td>See Also</td>
<td>218</td>
</tr>
<tr>
<td>Volume</td>
<td>219</td>
</tr>
<tr>
<td>Contents</td>
<td>219</td>
</tr>
<tr>
<td>See Also</td>
<td>219</td>
</tr>
<tr>
<td>VolumeFrom</td>
<td>220</td>
</tr>
<tr>
<td>Contents</td>
<td>220</td>
</tr>
<tr>
<td>See Also</td>
<td>220</td>
</tr>
<tr>
<td>Common Parameters</td>
<td>221</td>
</tr>
<tr>
<td>Common Errors</td>
<td>223</td>
</tr>
</tbody>
</table>
Welcome

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage Docker containers on a cluster. You can host your cluster on a serverless infrastructure that is managed by Amazon ECS by launching your services or tasks using the Fargate launch type. For more control, you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) instances that you manage by using the EC2 launch type. For more information about launch types, see Amazon ECS Launch Types.

Amazon ECS lets you launch and stop container-based applications with simple API calls, allows you to get the state of your cluster from a centralized service, and gives you access to many familiar Amazon EC2 features.

You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs, isolation policies, and availability requirements. Amazon ECS eliminates the need for you to operate your own cluster management and configuration management systems or worry about scaling your management infrastructure.

This document was last published on December 19, 2017.
Actions

The following actions are supported:

- CreateCluster (p. 3)
- CreateService (p. 6)
- DeleteAttributes (p. 14)
- DeleteCluster (p. 17)
- DeleteService (p. 21)
- DeregisterContainerInstance (p. 26)
- DeregisterTaskDefinition (p. 32)
- DescribeClusters (p. 37)
- DescribeContainerInstances (p. 41)
- DescribeServices (p. 47)
- DescribeTaskDefinition (p. 52)
- DescribeTasks (p. 57)
- DiscoverPollEndpoint (p. 62)
- ListAttributes (p. 64)
- ListClusters (p. 68)
- ListContainerInstances (p. 71)
- ListServices (p. 75)
- ListTaskDefinitionFamilies (p. 79)
- ListTaskDefinitions (p. 84)
- ListTasks (p. 88)
- PutAttributes (p. 93)
- RegisterContainerInstance (p. 97)
- RegisterTaskDefinition (p. 101)
- RunTask (p. 111)
- StartTask (p. 119)
- StopTask (p. 126)
- SubmitContainerStateChange (p. 131)
- SubmitTaskStateChange (p. 134)
- UpdateContainerAgent (p. 137)
- UpdateContainerInstancesState (p. 142)
- UpdateService (p. 149)
CreateCluster

Creates a new Amazon ECS cluster. By default, your account receives a default cluster when you launch your first container instance. However, you can create your own cluster with a unique name with the CreateCluster action.

**Note**
When you call the CreateCluster (p. 3) API operation, Amazon ECS attempts to create the service-linked role for your account so that required resources in other AWS services can be managed on your behalf. However, if the IAM user that makes the call does not have permissions to create the service-linked role, it is not created. For more information, see Using Service-Linked Roles for Amazon ECS in the Amazon Elastic Container Service Developer Guide.

**Request Syntax**

```json
{
    "clusterName": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**clusterName (p. 3)**

The name of your cluster. If you do not specify a name for your cluster, you create a cluster named default. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Type: String

Required: No

**Response Syntax**

```json
{
    "cluster": {
        "activeServicesCount": number,
        "clusterArn": "string",
        "clusterName": "string",
        "pendingTasksCount": number,
        "registeredContainerInstancesCount": number,
        "runningTasksCount": number,
        "statistics": [
            {
                "name": "string",
                "value": "string"
            }
        ],
        "status": "string"
    }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

cluster (p. 3)

The full description of your new cluster.

Type: Cluster (p. 162) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request creates a cluster called My-cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 29
```
X-Amz-Target: AmazonEC2ContainerServiceV20141113.CreateCluster
X-Amz-Date: 20150429T163840Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "clusterName": "My-cluster"
}

Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 16:38:41 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 209
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "cluster": {
    "activeServicesCount": 0,
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/My-cluster",
    "clusterName": "My-cluster",
    "pendingTasksCount": 0,
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "status": "ACTIVE"
  }
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
CreateService

Runs and maintains a desired number of tasks from a specified task definition. If the number of tasks running in a service drops below `desiredCount`, Amazon ECS spawns another copy of the task in the specified cluster. To update an existing service, see `UpdateService (p. 149)`.

In addition to maintaining the desired count of tasks in your service, you can optionally run your service behind a load balancer. The load balancer distributes traffic across the tasks that are associated with the service. For more information, see Service Load Balancing in the Amazon Elastic Container Service Developer Guide.

You can optionally specify a deployment configuration for your service. During a deployment, the service scheduler uses the `minimumHealthyPercent` and `maximumPercent` parameters to determine the deployment strategy. The deployment is triggered by changing the task definition or the desired count of a service with an `UpdateService (p. 149)` operation.

The `minimumHealthyPercent` represents a lower limit on the number of your service's tasks that must remain in the `RUNNING` state during a deployment, as a percentage of the `desiredCount` (rounded up to the nearest integer). This parameter enables you to deploy without using additional cluster capacity. For example, if your service has a `desiredCount` of four tasks and a `minimumHealthyPercent` of 50%, the scheduler can stop two existing tasks to free up cluster capacity before starting two new tasks. Tasks for services that do not use a load balancer are considered healthy if they are in the `RUNNING` state.

Tasks for services that do use a load balancer are considered healthy if they are in the `RUNNING` state and the container instance they are hosted on is reported as healthy by the load balancer. The default value for `minimumHealthyPercent` is 50% in the console and 100% for the AWS CLI, the AWS SDKs, and the APIs.

The `maximumPercent` parameter represents an upper limit on the number of your service's tasks that are allowed in the `RUNNING` or `PENDING` state during a deployment, as a percentage of the `desiredCount` (rounded down to the nearest integer). This parameter enables you to define the deployment batch size. For example, if your service has a `desiredCount` of four tasks and a `maximumPercent` value of 200%, the scheduler can start four new tasks before stopping the four older tasks (provided that the cluster resources required to do this are available). The default value for `maximumPercent` is 200%.

When the service scheduler launches new tasks, it determines task placement in your cluster using the following logic:

- Determine which of the container instances in your cluster can support your service's task definition (for example, they have the required CPU, memory, ports, and container instance attributes).
- By default, the service scheduler attempts to balance tasks across Availability Zones in this manner (although you can choose a different placement strategy) with the `placementStrategy` parameter:
  - Sort the valid container instances by the fewest number of running tasks for this service in the same Availability Zone as the instance. For example, if zone A has one running service task and zones B and C each have zero, valid container instances in either zone B or C are considered optimal for placement.
  - Place the new service task on a valid container instance in an optimal Availability Zone (based on the previous steps), favoring container instances with the fewest number of running tasks for this service.

Request Syntax

```json
{
  "clientToken": "string",

  "placementStrategy": "eks: Spread",

  "desiredCount": 3
}
```

API Version 2014-11-13
6
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

clientToken (p. 6)

Unique, case-sensitive identifier you provide to ensure the idempotency of the request. Up to 32 ASCII characters are allowed.

Type: String

Required: No

cluster (p. 6)

The short name or full Amazon Resource Name (ARN) of the cluster on which to run your service. If you do not specify a cluster, the default cluster is assumed.

Type: String
Required: No

deploymentConfiguration (p. 6)

Optional deployment parameters that control how many tasks run during the deployment and the
ordering of stopping and starting tasks.

Type: DeploymentConfiguration (p. 182) object

Required: No
desiredCount (p. 6)

The number of instantiations of the specified task definition to place and keep running on your
cluster.

Type: Integer

Required: Yes

launchType (p. 6)

The launch type on which to run your service.

Type: String

Valid Values: EC2 | FARGATE

Required: No

loadBalancers (p. 6)

A load balancer object representing the load balancer to use with your service. Currently, you are
limited to one load balancer or target group per service. After you create a service, the load balancer
name or target group ARN, container name, and container port specified in the service definition are
immutable.

For Classic Load Balancers, this object must contain the load balancer name, the container name (as
it appears in a container definition), and the container port to access from the load balancer. When a
task from this service is placed on a container instance, the container instance is registered with the
load balancer specified here.

For Application Load Balancers and Network Load Balancers, this object must contain the load
balancer target group ARN, the container name (as it appears in a container definition), and the
container port to access from the load balancer. When a task from this service is placed on a
container instance, the container instance and port combination is registered as a target in the
target group specified here.

Type: Array of LoadBalancer (p. 191) objects

Required: No

networkConfiguration (p. 6)

The network configuration for the service. This parameter is required for task definitions that use
the awsvpc network mode to receive their own Elastic Network Interface, and it is not supported for
other network modes. For more information, see Task Networking in the Amazon Elastic Container
Service Developer Guide.

Type: NetworkConfiguration (p. 195) object

Required: No
placementConstraints (p. 6)

An array of placement constraint objects to use for tasks in your service. You can specify a maximum of 10 constraints per task (this limit includes constraints in the task definition and those specified at run time).

Type: Array of PlacementConstraint (p. 197) objects

Required: No

placementStrategy (p. 6)

The placement strategy objects to use for tasks in your service. You can specify a maximum of five strategy rules per service.

Type: Array of PlacementStrategy (p. 198) objects

Required: No

platformVersion (p. 6)

The platform version on which to run your service. If one is not specified, the latest version is used by default.

Type: String

Required: No

taskDefinition (p. 6)

The family and revision (family:revision) or full ARN of the task definition to run in your service. If a revision is not specified, the latest ACTIVE revision is used.

Required: Yes
Type: String
Required: Yes

Response Syntax

{  
"service": {  
"clusterArn": "string",
"createdAt": number,
"deploymentConfiguration": {  
"maximumPercent": number,
"minimumHealthyPercent": number
 },
"deployments": [  
 {  
"createdAt": number,
"desiredCount": number,
"id": "string",
"launchType": "string",
"networkConfiguration": {  
"awsvpcConfiguration": {  
"assignPublicIp": "string",
"securityGroups": [ "string" ],
"subnets": [ "string" ]
 } },
"pendingCount": number,
"platformVersion": "string",
"runningCount": number,
"status": "string",
"taskDefinition": "string",
"updatedAt": number
  } ],
"desiredCount": number,
"events": [  
 {  
"createdAt": number,
"id": "string",
"message": "string"
 } ],
"launchType": "string",
"loadBalancers": [  
 {  
"containerName": "string",
"containerPort": number,
"loadBalancerName": "string",
"targetGroupArn": "string"
  } ],
"networkConfiguration": {  
"awsvpcConfiguration": {  
"assignPublicIp": "string",
"securityGroups": [ "string" ],
"subnets": [ "string" ]
 } },
"pendingCount": number,
"placementConstraints": [  
 {  
"expression": "string",
"key": "string",
"value": "string"
  } ]
}

API Version 2014-11-13
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

service (p. 10)

The full description of your service following the create call.

Type: Service (p. 203) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

AccessDeniedException

You do not have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400
Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example API request creates a service in your default region called ecs-simple-service. The service uses the ecs-demo task definition and it maintains 10 instantiations of that task.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 87
X-Amz-Target: AmazonEC2ContainerServiceV20141113.CreateService
X-Amz-Date: 20150429T170125Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
   "serviceName": "ecs-simple-service",
   "taskDefinition": "ecs-demo",
   "desiredCount": 10
}
```

Sample Response

```
HTTP/1.1 200 OK
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteAttributes

Deletes one or more custom attributes from an Amazon ECS resource.

Request Syntax

```json
{
    "attributes": [
        {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
        }
    ],
    "cluster": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

attributes (p. 14)

The attributes to delete from your resource. You can specify up to 10 attributes per request. For custom attributes, specify the attribute name and target ID, but do not specify the value. If you specify the target ID using the short form, you must also specify the target type.

Type: Array of Attribute (p. 160) objects

Required: Yes

cluster (p. 14)

The short name or full Amazon Resource Name (ARN) of the cluster that contains the resource to delete attributes. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

Response Syntax

```json
{
    "attributes": [
        {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
        }
    ]
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

attributes (p. 14)

A list of attribute objects that were successfully deleted from your resource.

Type: Array of Attribute (p. 160) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

TargetNotFoundException

The specified target could not be found. You can view your available container instances with ListContainerInstances (p. 71). Amazon ECS container instances are cluster-specific and region-specific.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example deletes an attribute with the name stack from a container instance.

Sample Request

```
POST / HTTP/1.1
Host: madison.us-west-2.amazonaws.com
Accept-Encoding: identity
```
Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Thu, 22 Dec 2016 19:38:51 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 158
Connection: keep-alive
x-amzn-RequestId: 445193ca-c87e-11e6-86db-1bd3d928caf

{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-
      df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteCluster

Deletes the specified cluster. You must deregister all container instances from this cluster before you may delete it. You can list the container instances in a cluster with ListContainerInstances (p. 71) and deregister them with DeregisterContainerInstance (p. 26).

Request Syntax

```
{
  "cluster": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

* **cluster** (p. 17)
  
  The short name or full Amazon Resource Name (ARN) of the cluster to delete.
  
  Type: String
  
  Required: Yes

Response Syntax

```
{
  "cluster": {
    "activeServicesCount": number,
    "clusterArn": "string",
    "clusterName": "string",
    "pendingTasksCount": number,
    "registeredContainerInstancesCount": number,
    "runningTasksCount": number,
    "statistics": [
      {
        "name": "string",
        "value": "string"
      }
    ],
    "status": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

* **cluster** (p. 17)
  
  The full description of the deleted cluster.
Type: Cluster (p. 162) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterContainsContainerInstancesException

You cannot delete a cluster that has registered container instances. You must first deregister the container instances before you can delete the cluster. For more information, see DeregisterContainerInstance (p. 26).

HTTP Status Code: 400

ClusterContainsServicesException

You cannot delete a cluster that contains services. You must first update the service to reduce its desired task count to 0 and then delete the service. For more information, see UpdateService (p. 149) and DeleteService (p. 21).

HTTP Status Code: 400

ClusterContainsTasksException

You cannot delete a cluster that has active tasks.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request deletes the cluster called My-cluster.

Sample Request

```plaintext
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 25
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeleteCluster
X-Amz-Date: 20150429T170952Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "My-cluster"
}
```

Sample Response

```plaintext
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 17:09:54 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 211
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "cluster": {
        "activeServicesCount": 0,
        "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/My-cluster",
        "clusterName": "My-cluster",
        "pendingTasksCount": 0,
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "status": "INACTIVE"
    }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
DeleteService

Deletes a specified service within a cluster. You can delete a service if you have no running tasks in it and the desired task count is zero. If the service is actively maintaining tasks, you cannot delete it, and you must update the service to a desired task count of zero. For more information, see UpdateService (p. 149).

**Note**

When you delete a service, if there are still running tasks that require cleanup, the service status moves from **ACTIVE** to **DRAINING**, and the service is no longer visible in the console or in ListServices (p. 75) API operations. After the tasks have stopped, then the service status moves from **DRAINING** to **INACTIVE**. Services in the **DRAINING** or **INACTIVE** status can still be viewed with DescribeServices (p. 47) API operations. However, in the future, INACTIVE services may be cleaned up and purged from Amazon ECS record keeping, and DescribeServices (p. 47) API operations on those services return a ServiceNotFoundException error.

**Request Syntax**

```json
{
    "cluster": "string",
    "service": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 21)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service to delete. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**service (p. 21)**

The name of the service to delete.

Type: String

Required: Yes

**Response Syntax**

```json
{
    "service": {
        "clusterArn": "string",
        "createdAt": number,
        "deploymentConfiguration": {
            "maximumPercent": number,
        }
    }
}
```
"minimumHealthyPercent": number,
},
"deployments": [
{
"createdAt": number,
"desiredCount": number,
"id": "string",
"launchType": "string",
"networkConfiguration": {
"awsvpcConfiguration": {
 "assignPublicIp": "string",
 "securityGroups": [ "string" ],
 "subnets": [ "string" ]
}
},
"pendingCount": number,
"platformVersion": "string",
"runningCount": number,
"status": "string",
"taskDefinition": "string",
"updatedAt": number
}
],
"desiredCount": number,
"events": [
{
"createdAt": number,
"id": "string",
"message": "string"
}
],
"launchType": "string",
"loadBalancers": [
{
"containerName": "string",
"containerPort": number,
"loadBalancerName": "string",
"targetGroupArn": "string"
}
],
"networkConfiguration": {
"awsvpcConfiguration": {
 "assignPublicIp": "string",
 "securityGroups": [ "string" ],
 "subnets": [ "string" ]
}
},
"pendingCount": number,
"placementConstraints": [
{
 "expression": "string",
 "type": "string"
}
],
"placementStrategy": [
{
 "field": "string",
 "type": "string"
}
],
"platformVersion": "string",
"roleArn": "string",
"runningCount": number,
"serviceArn": "string",
"serviceName": "string",
"status": "string"
"taskDefinition": "string"
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**service (p. 21)**

The full description of the deleted service.

Type: Service (p. 203) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

**ClientException**

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

**ClusterNotFoundException**

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

**ServiceNotFoundException**

The specified service could not be found. You can view your available services with ListServices (p. 75). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don’t have to sign requests yourself.

**Example**

This example API request deletes the test service from the default cluster.

**Sample Request**

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 19
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeleteService
X-Amz-Date: 20150429T172539Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "service": "test"
}
```

**Sample Response**

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 17:25:40 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 13590
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deploymentConfiguration": {
      "maximumPercent": 200,
      "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "createdAt": 1430320735.285,
        "desiredCount": 0,
        "id": "ecs-svc/9223370606534040511",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/sleep360:27",
        "updatedAt": 1430320735.285
      }
    ],
    "desiredCount": 0,
    "events": [],
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/test",
    "serviceName": "test",
    "status": "DRAINING",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/sleep360:27"
  }
}
```

API Version 2014-11-13
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeregisterContainerInstance

Deregisters an Amazon ECS container instance from the specified cluster. This instance is no longer available to run tasks.

If you intend to use the container instance for some other purpose after deregistration, you should stop all of the tasks running on the container instance before deregistration. That prevents any orphaned tasks from consuming resources.

Deregistering a container instance removes the instance from a cluster, but it does not terminate the EC2 instance; if you are finished using the instance, be sure to terminate it in the Amazon EC2 console to stop billing.

**Note**

If you terminate a running container instance, Amazon ECS automatically deregisters the instance from your cluster (stopped container instances or instances with disconnected agents are not automatically deregistered when terminated).

**Request Syntax**

```
{
  "cluster": "string",
  "containerInstance": "string",
  "force": boolean
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

**cluster (p. 26)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instance to deregister. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**containerInstance (p. 26)**

The container instance ID or full ARN of the container instance to deregister. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the `container-instance` namespace, and then the container instance ID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_ID`.

Type: String

Required: Yes

**force (p. 26)**

Forces the deregistration of the container instance. If you have tasks running on the container instance when you deregister it with the `force` option, these tasks remain running until you terminate the instance or the tasks stop through some other means, but they are orphaned (no...
longer monitored or accounted for by Amazon ECS). If an orphaned task on your container instance is part of an Amazon ECS service, then the service scheduler starts another copy of that task, on a different container instance if possible.

Any containers in orphaned service tasks that are registered with a Classic Load Balancer or an Application Load Balancer target group are deregistered. They begin connection draining according to the settings on the load balancer or target group.

Type: Boolean

Required: No

Response Syntax

```json
{
  "containerInstance": {
    "agentConnected": boolean,
    "agentUpdateStatus": "string",
    "attachments": [
      {
        "details": [
          {
            "name": "string",
            "value": "string"
          }
        ],
        "id": "string",
        "status": "string",
        "type": "string"
      }
    ],
    "attributes": [
      {
        "name": "string",
        "targetId": "string",
        "targetType": "string",
        "value": "string"
      }
    ],
    "containerInstanceArn": "string",
    "ec2InstanceId": "string",
    "pendingTasksCount": number,
    "registeredAt": number,
    "registeredResources": [
      {
        "doubleValue": number,
        "integerValue": number,
        "longValue": number,
        "name": "string",
        "stringSetValue": [ "string" ],
        "type": "string"
      }
    ],
    "remainingResources": [
      {
        "doubleValue": number,
        "integerValue": number,
        "longValue": number,
        "name": "string",
        "stringSetValue": [ "string" ],
        "type": "string"
      }
    }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**containerInstance (p. 27)**

The container instance that was deregistered.

Type: `ContainerInstance (p. 174)` object

Errors

For information about the errors that are common to all actions, see [Common Errors (p. 223)].

**ClientException**

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

**ClusterNotFoundException**

The specified cluster could not be found. You can view your available clusters with [ListClusters (p. 68)](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cluster_tasks.html). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (**AUTHPARAMS**) must be replaced with an AWS Signature Version 4 signature. For more information, see [Signature Version 4 Signing Process](https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC-Get-Started.html) in the *AWS General Reference*.  

---

API Version 2014-11-13

28
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request deregisters a container instance with the ID `f4292606-fbed-4b53-833b-92cad7c687c2` in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 61
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeregisterContainerInstance
X-Amz-Date: 20151001T191224Z
User-Agent: aws-cli/1.8.7 Python/2.7.9 Darwin/14.5.0
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "containerInstance": "c9c9a6f2-8766-464b-8805-9c57b9368fb0"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 01 Oct 2015 19:12:25 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1613
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "containerInstance": {
        "agentConnected": true,
        "attributes": [
            {
                "name": "com.amazonaws.ecs.capability.privileged-container"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
            }
        ],
        "containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-instance/c9c9a6f2-8766-464b-8805-9c57b9368fb0",
        "ec2InstanceId": "i-0c3826c9",
```

API Version 2014-11-13
"pendingTasksCount": 0,
"registeredResources": [
  {
    "doubleValue": 0,
    "integerValue": 1024,
    "longValue": 0,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 995,
    "longValue": 0,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS",
    "stringSetValue": ["22",
                       "2376",
                       "2375",
                       "51678"],
    "type": "STRINGSET"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
],
"remainingResources": [
  {
    "doubleValue": 0,
    "integerValue": 1024,
    "longValue": 0,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 995,
    "longValue": 0,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS",
    "stringSetValue": ["22",
                       "2376",
                       "2375",
                       "51678"],
    "type": "STRINGSET"
  }
],
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeregisterTaskDefinition

Deregisters the specified task definition by family and revision. Upon deregistration, the task definition is marked as INACTIVE. Existing tasks and services that reference an INACTIVE task definition continue to run without disruption. Existing services that reference an INACTIVE task definition can still scale up or down by modifying the service's desired count.

You cannot use an INACTIVE task definition to run new tasks or create new services, and you cannot update an existing service to reference an INACTIVE task definition (although there may be up to a 10-minute window following deregistration where these restrictions have not yet taken effect).

Note
At this time, INACTIVE task definitions remain discoverable in your account indefinitely; however, this behavior is subject to change in the future, so you should not rely on INACTIVE task definitions persisting beyond the lifecycle of any associated tasks and services.

Request Syntax

```
{
  "taskDefinition": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

taskDefinition (p. 32)

The family and revision (family:revision) or full Amazon Resource Name (ARN) of the task definition to deregister. You must specify a revision.

Type: String
Required: Yes

Response Syntax

```
{
  "taskDefinition": {
    "compatibilities": [ "string" ],
    "containerDefinitions": [ {
      "command": [ "string" ],
      "cpu": number,
      "disableNetworking": boolean,
      "dnsSearchDomains": [ "string" ],
      "dnsServers": [ "string" ],
      "dockerLabels": { "string" : "string" },
      "dockerSecurityOptions": [ "string" ],
      "entryPoint": [ "string" ],
      "environment": [ "string"
```
{  
  "name": "string",
  "value": "string"
}
],
"essential": boolean,
"extraHosts": [
  {
    "hostname": "string",
    "ipAddress": "string"
  }
],
"hostname": "string",
"image": "string",
"links": [ "string" ],
"linuxParameters": {
  "capabilities": {
    "add": [ "string" ],
    "drop": [ "string" ]
  },
  "devices": [
    {
      "containerPath": "string",
      "hostPath": "string",
      "permissions": [ "string" ]
    }
  ],
  "initProcessEnabled": boolean
},
"logConfiguration": {
  "logDriver": "string",
  "options": {
    "string": "string"
  }
},
"memory": number,
"memoryReservation": number,
"mountPoints": [
  {
    "containerPath": "string",
    "readOnly": boolean,
    "sourceVolume": "string"
  }
],
"name": "string",
"portMappings": [
  {
    "containerPort": number,
    "hostPort": number,
    "protocol": "string"
  }
],
"privileged": boolean,
"readonlyRootFilesystem": boolean,
"ulimits": [
  {
    "hardLimit": number,
    "name": "string",
    "softLimit": number
  }
],
"user": "string",
"volumesFrom": [
  {
    "readOnly": boolean,
    "sourceContainer": "string"
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**taskDefinition (p. 32)**

The full description of the deregistered task.

Type: TaskDefinition (p. 211) object

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 223).

**ClientException**

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.
HTTP Status Code: 400
**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400
**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

---

**Example**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see [Signature Version 4 Signing Process](https://aws.amazon.com/documentation/generalref/) in the _AWS General Reference_.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the [AWS Command Line Interface (AWS CLI)](https://aws.amazon.com/cli/) or one of the [AWS SDKs](https://aws.amazon.com/sdk/) to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

**Example**

The following example request deregisters the first revision of the `cpu-wave` task definition family (cpu-wave:1). In the resulting output, the task definition status becomes **INACTIVE**.

**Sample Request**

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 35
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeregisterTaskDefinition
X-Amz-Date: 20150429T184806Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "taskDefinition": "cpu-wave:1"
}
```

**Sample Response**

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 12 Jun 2015 23:07:39 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 491
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "taskDefinition": {
    "containerDefinitions": [
      {
        "command": [
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeClusters

Describes one or more of your clusters.

Request Syntax

```
{
   "clusters": [ "string" ],
   "include": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

clusters (p. 37)

A list of up to 100 cluster names or full cluster Amazon Resource Name (ARN) entries. If you do not specify a cluster, the default cluster is assumed.

Type: Array of strings

Required: No

include (p. 37)

Additional information about your clusters to be separated by launch type, including:

- runningEC2TasksCount
- runningFargateTasksCount
- pendingEC2TasksCount
- pendingFargateTasksCount
- activeEC2ServiceCount
- activeFargateServiceCount
- drainingEC2ServiceCount
- drainingFargateServiceCount

Type: Array of strings

Valid Values: STATISTICS

Required: No

Response Syntax

```
{
   "clusters": [
      {
         "activeServicesCount": number,
         "clusterArn": "string",
         ...
      }
   ]
}
```

API Version 2014-11-13
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

clusters (p. 37)

The list of clusters.

Type: Array of Cluster (p. 162) objects

failures (p. 37)

Any failures associated with the call.

Type: Array of Failure (p. 184) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.
HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request provides descriptive information about the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 25
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeClusters
X-Amz-Date: 20150429T185014Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "clusters": [
    "default"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 18:50:14 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 220
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "clusters": [
    {
      "activeServicesCount": 1,
      "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
      "clusterName": "default",
      "pendingTasksCount": 0,
      "registeredContainerInstancesCount": 0,
      "runningTasksCount": 0,
      "status": "ACTIVE"
    }
  ],
  "failures": []
}
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeContainerInstances

Describes Amazon Elastic Container Service container instances. Returns metadata about registered and remaining resources on each container instance requested.

Request Syntax

```json
{
  "cluster": "string",
  "containerInstances": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 41)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instances to describe. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**containerInstances (p. 41)**

A list of container instance IDs or full ARN entries.

Type: Array of strings

Required: Yes

Response Syntax

```json
{
  "containerInstances": [
    {
      "agentConnected": boolean,
      "agentUpdateStatus": "string",
      "attachments": [
        {
          "details": [
            {
              "name": "string",
              "value": "string"
            }
          ],
          "id": "string",
          "status": "string",
          "type": "string"
        }
      ],
      "attributes": [
      
    }
  
```

API Version 2014-11-13

41
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**containerInstances (p. 41)**

The list of container instances.

Type: Array of ContainerInstance (p. 174) objects

**failures (p. 41)**

Any failures associated with the call.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents ([AUTHPARAMS]) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request provides descriptive information about a container instance with an ID of f9cc75bb-0c94-46b9-bf6d-49d320bc1551 in the default cluster.

Sample Request

POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 64
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeContainerInstances
X-Amz-Date: 20160520T171518Z
User-Agent: aws-cli/1.10.30 Python/2.7.11 Darwin/15.4.0 botocore/1.4.17
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "containerInstances": [
        "f9cc75bb-0c94-46b9-bf6d-49d320bc1551"
    ]
}

Sample Response
{
    "containerInstances": [
        {
            "agentConnected": true,
            "attributes": [
                {
                    "name": "com.amazonaws.ecs.capability.privileged-container"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
                },
                {
                    "name": "com.amazonaws.ecs.capability.ecr-auth"
                }
            ],
            "ec2InstanceId": "i-042f39dc",
            "pendingTasksCount": 0,
            "registeredResources": [
                {
                    "doubleValue": 0,
                    "integerValue": 1024,
                    "longValue": 0,
                    "name": "CPU",
                    "type": "INTEGER"
                },
                {
                    "doubleValue": 0,
                    "integerValue": 995,
                    "longValue": 0,
                    "name": "MEMORY",
                    "type": "INTEGER"
                }
            ]
        }
    ]
}
Example

```
{
  "doubleValue": 0,
  "integerValue": 0,
  "longValue": 0,
  "name": "PORTS",
  "stringSetValue": [
    "22",
    "2376",
    "2375",
    "51678"
  ],
  "type": "STRINGSET"
},
{
  "doubleValue": 0,
  "integerValue": 0,
  "longValue": 0,
  "name": "PORTS_UDP",
  "stringSetValue": [],
  "type": "STRINGSET"
],
"remainingResources": [
  {
    "doubleValue": 0,
    "integerValue": 1024,
    "longValue": 0,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 995,
    "longValue": 0,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS",
    "stringSetValue": [
      "22",
      "2376",
      "2375",
      "51678"
    ],
    "type": "STRINGSET"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
],
"runningTasksCount": 0,
"status": "ACTIVE",
"version": 850,
"versionInfo": {
  "agentHash": "0931217",
  "agentVersion": "1.9.0",

API Version 2014-11-13
45
```
"dockerVersion": "DockerVersion: 1.9.1"
]
],
"failures": []
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeServices

Describes the specified services running in your cluster.

Request Syntax

```json
{
    "cluster": "string",
    "services": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 47)**

The short name or full Amazon Resource Name (ARN) the cluster that hosts the service to describe. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**services (p. 47)**

A list of services to describe. You may specify up to 10 services to describe in a single operation.

Type: Array of strings

Required: Yes

Response Syntax

```json
{
    "failures": [
        {
            "arn": "string",
            "reason": "string"
        }
    ],
    "services": [
        {
            "clusterArn": "string",
            "createdAt": number,
            "deploymentConfiguration": {
                "maximumPercent": number,
                "minimumHealthyPercent": number
            },
            "deployments": [
                {
                    "createdAt": number,
                    "desiredCount": number,
                    "id": "string",
                    "launchType": "string",
                    "networkConfiguration": {
```
"awsvpcConfiguration": {  
  "assignPublicIp": "string",  
  "securityGroups": [ "string" ],  
  "subnets": [ "string" ]  
},  
"pendingCount": number,  
"platformVersion": "string",  
"runningCount": number,  
"status": "string",  
"taskDefinition": "string",  
"updatedAt": number  
},  
"desiredCount": number,  
"events": [  
  {  
    "createdAt": number,  
    "id": "string",  
    "message": "string"  
  }  
],  
"launchType": "string",  
"loadBalancers": [  
  {  
    "containerName": "string",  
    "containerPort": number,  
    "loadBalancerName": "string",  
    "targetGroupArn": "string"  
  }  
],  
"networkConfiguration": {  
  "awsvpcConfiguration": {  
    "assignPublicIp": "string",  
    "securityGroups": [ "string" ],  
    "subnets": [ "string" ]  
  }  
},  
"pendingCount": number,  
"placementConstraints": [  
  {  
    "expression": "string",  
    "type": "string"  
  }  
],  
"placementStrategy": [  
  {  
    "field": "string",  
    "type": "string"  
  }  
],  
"platformVersion": "string",  
"roleArn": "string",  
"runningCount": number,  
"serviceArn": "string",  
"serviceName": "string",  
"status": "string",  
"taskDefinition": "string"  
}  
]  
}
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

failures (p. 47)

Any failures associated with the call.

Type: Array of Failure (p. 184) objects

services (p. 47)

The list of services described.

Type: Array of Service (p. 203) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.
Example

This example request provides a full description of the bunker_buster service in the telemetry cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 55
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeServices
X-Amz-Date: 20150528T163859Z
User-Agent: aws-cli/1.7.30 Python/2.7.9 Darwin/14.3.0
Authorization: AUTHPARAMS

{
  "services": [
    "bunker-buster"
  ],
  "cluster": "telemetry"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:02:59 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 2449
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "failures": [],
  "services": [
    {
      "deploymentConfiguration": {
        "maximumPercent": 200,
        "minimumHealthyPercent": 100
      },
      "deployments": [
        {
          "createdAt": 1432829320.611,
          "desiredCount": 4,
          "id": "ecs-svc/9223706040254555196",
          "pendingCount": 0,
          "runningCount": 4,
          "status": "PRIMARY",
          "updatedAt": 1432829320.611
        }
      ],
      "desiredCount": 4,
      "events": [],
      "loadBalancers": [],
      "pendingCount": 0,
      "runningCount": 4,
    }
  ]
}
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
Describes a task definition. You can specify a family and revision to find information about a specific task definition, or you can simply specify the family to find the latest ACTIVE revision in that family.

**Note**
You can only describe INACTIVE task definitions while an active task or service references them.

**Request Syntax**

```
{
  "taskDefinition": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**taskDefinition (p. 52)**

The family for the latest ACTIVE revision, family and revision (family:revision) for a specific revision in the family, or full Amazon Resource Name (ARN) of the task definition to describe.

Type: String

Required: Yes

**Response Syntax**

```
{
  "taskDefinition": {
    "compatibilities": [ "string" ],
    "containerDefinitions": [
      {
        "command": [ "string" ],
        "cpu": number,
        "disableNetworking": boolean,
        "dnsSearchDomains": [ "string" ],
        "dnsServers": [ "string" ],
        "dockerLabels": {
          "string": "string"
        },
        "dockerSecurityOptions": [ "string" ],
        "entryPoint": [ "string" ],
        "environment": [
          {
            "name": "string",
            "value": "string"
          }
        ],
        "essential": boolean,
        "extraHosts": [
          {
            "hostname": "string"
          }
        ]
      }
    ],
    "version": number
  }
}
```

API Version 2014-11-13
"ipAddress": "string",
"hostname": "string",
"image": "string",
"linuxParameters": {
"capabilities": {
"add": [ "string" ],
"drop": [ "string" ]
},
"devices": [
{"containerPath": "string",
"hostPath": "string",
"permissions": [ "string" ]
}
],
"initProcessEnabled": boolean
},
"logConfiguration": {
"logDriver": "string",
"options": {
"string": "string"
}
},
"memory": number,
"memoryReservation": number,
"mountPoints": [
{"containerPath": "string",
"readOnly": boolean,
"sourceVolume": "string"
}
],
"name": "string",
"portMappings": [
{"containerPort": number,
"hostPort": number,
"protocol": "string"
}
],
"privileged": boolean,
"readonlyRootFilesystem": boolean,
"ulimits": [
{"hardLimit": number,
"name": "string",
"softLimit": number
}
],
"user": "string",
"volumesFrom": [
{"readOnly": boolean,
"sourceContainer": "string"
}
],
"workingDirectory": "string"
],
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"memory": "string"
"networkMode": "string",
"placementConstraints": [  
  {    "expression": "string",
      "type": "string"
  }
],
"requiresAttributes": [  
  {    "name": "string",
      "targetId": "string",
      "targetType": "string",
      "value": "string"
  }
],
"requiresCompatibilities": [ "string" ],
"revision": number,
"status": "string",
"taskDefinitionArn": "string",
"taskRoleArn": "string",
"volumes": [  
  {    "host": {    "sourcePath": "string"
  },    "name": "string"
  }
]
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

taskDefinition (p. 52)

The full task definition description.

Type: TaskDefinition (p. 211) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400
ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request provides descriptive information about the 10th revision of a task definition in the hello_world family.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 36
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeTaskDefinition
X-Amz-Date: 20150429T190902Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "taskDefinition": "hello_world:10"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:09:03 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 574
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "taskDefinition": {
    "containerDefinitions": [
      {
        "cpu": 10,
        "environment": [],
        "essential": true,
        "image": "wordpress",
        "links": [
          "mysql"
        ],
        "memory": 500,
      }
    ]
  }
}
```
"mountPoints": [],
"name": "wordpress",
"portMappings": [
{
  "containerPort": 80,
  "hostPort": 80
}
],
"volumesFrom": []
},
{
"cpu": 10,
"environment": [
{
  "name": "MYSQL_ROOT_PASSWORD",
  "value": "password"
}
],
"essential": true,
"image": "mysql",
"memory": 500,
"mountPoints": [],
"name": "mysql",
"portMappings": [],
"volumesFrom": []
}
],
"family": "hello_world",
"revision": 10,
"taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:10",
"volumes": []
}
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeTasks

Describes a specified task or tasks.

Request Syntax

```
{
  "cluster": "string",
  "tasks": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 57)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task to describe. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**tasks (p. 57)**

A list of up to 100 task IDs or full ARN entries.

Type: Array of strings

Required: Yes

Response Syntax

```
{
  "failures": [ 
    { 
      "arn": "string",
      "reason": "string"
    } 
  ],
  "tasks": [ 
    { 
      "attachments": [ 
        { 
          "details": [ 
            { 
              "name": "string",
              "value": "string"
            } 
          ],
          "id": "string",
          "status": "string",
          "type": "string"
        }
      ]
    }
  ]
}
```

API Version 2014-11-13
"clusterArn": "string",
"connectivity": "string",
"connectivityAt": number,
"containerInstanceArn": "string",
"containers": [
{
"containerArn": "string",
"exitCode": number,
"lastStatus": "string",
"name": "string",
"networkBindings": [
{
"bindIP": "string",
"containerPort": number,
"hostPort": number,
"protocol": "string"
}
],
"networkInterfaces": [
{
"attachmentId": "string",
"ipv6Address": "string",
"privateIpv4Address": "string"
}
],
"reason": "string",
"taskArn": "string"
}
],
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
"containerOverrides": [
{
"command": [ "string" ],
"cpu": number,
"environment": [
{
"name": "string",
"value": "string"
}
],
"memory": number,
"memoryReservation": number,
"name": "string"
}
],
"executionRoleArn": "string",
"taskRoleArn": "string"
},
"platformVersion": "string",
"pullStartedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "string",
"stoppedAt": number,
"stoppedReason": "string",
"stoppingAt": number,
"taskArn": "string"
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

failures (p. 57)

Any failures associated with the call.

Type: Array of Failure (p. 184) objects
tasks (p. 57)

The list of tasks.

Type: Array of Task (p. 207) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don’t have to sign requests yourself.

Example

This example request provides descriptive information about a task with an ID of 1dc5c17a-422b-4dc4-b493-371970c6c4d6 in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 51
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeTasks
X-Amz-Date: 20161121T214915Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "tasks": [
    "1dc5c17a-422b-4dc4-b493-371970c6c4d6"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 21:49:16 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1238
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "failures": [],
  "tasks": [
    {
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6",
      "overrides": {
        "containerOverrides": [
          {
            "name": "simple-app"
          },
          {
            "name": "busybox"
          }
        ]
      },
      "lastStatus": "RUNNING",
      "createdAt": 1476822811.295,
      "version": 0,
      "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
      "startedAt": 147682833.998,
    }
  ]
}
```
"desiredStatus": "RUNNING",  
"startedBy": "ecs-svc/9223370560003250756",  
"containers": [
  {
    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/4df26bb4-f057-467b-a079-961675296e64",  
    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6",  
    "lastStatus": "RUNNING",  
    "name": "simple-app",  
    "networkBindings": [
      {
        "protocol": "tcp",  
        "bindIP": "0.0.0.0",  
        "containerPort": 80,  
        "hostPort": 32774
      }
    ]
  },
  {
    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e09064f7-7361-4c87-8ab9-8d073bbdbcb9",  
    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6",  
    "lastStatus": "RUNNING",  
    "name": "busybox",  
    "networkBindings": []
  }
]

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DiscoverPollEndpoint

**Note**
This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

Returns an endpoint for the Amazon ECS agent to poll for updates.

**Request Syntax**

```json
{
   "cluster": "string",
   "containerInstance": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

`cluster (p. 62)`

The short name or full Amazon Resource Name (ARN) of the cluster that the container instance belongs to.

Type: String

Required: No

`containerInstance (p. 62)`

The container instance ID or full ARN of the container instance. The ARN contains the `arn:aws:ecs` namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the container-instance namespace, and then the container instance ID. For example, `arn:aws:ecs:region:aws_account_id:container-instance/container_instance_ID`.

Type: String

Required: No

**Response Syntax**

```json
{
   "endpoint": "string",
   "telemetryEndpoint": "string"
}
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.
endpoint (p. 62)

The endpoint for the Amazon ECS agent to poll.

Type: String
telemetryEndpoint (p. 62)

The telemetry endpoint for the Amazon ECS agent.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListAttributes

Lists the attributes for Amazon ECS resources within a specified target type and cluster. When you specify a target type and cluster, ListAttributes returns a list of attribute objects, one for each attribute on each resource. You can filter the list of results to a single attribute name to only return results that have that name. You can also filter the results by attribute name and value, for example, to see which container instances in a cluster are running a Linux AMI (ecs.os-type=linux).

Request Syntax

```
{
  "attributeName": "string",
  "attributeValue": "string",
  "cluster": "string",
  "maxResults": number,
  "nextToken": "string",
  "targetType": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

attributeName (p. 64)

The name of the attribute with which to filter the results.

Type: String

Required: No

attributeValue (p. 64)

The value of the attribute with which to filter results. You must also specify an attribute name to use this parameter.

Type: String

Required: No

cluster (p. 64)

The short name or full Amazon Resource Name (ARN) of the cluster to list attributes. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

maxResults (p. 64)

The maximum number of cluster results returned by ListAttributes in paginated output. When this parameter is used, ListAttributes only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListAttributes request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListAttributes returns up to 100 results and a nextToken value if applicable.
Response Syntax

Type: Integer
Required: No

**nextToken (p. 64)**

The `nextToken` value returned from a previous paginated `ListAttributes` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value.

**Note**
This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String
Required: No

**targetType (p. 64)**

The type of the target with which to list attributes.

Type: String
Valid Values: `container-instance`
Required: Yes

**Response Syntax**

```
{
  "attributes": [
    {
      "name": "string",
      "targetId": "string",
      "targetType": "string",
      "value": "string"
    }
  ],
  "nextToken": "string"
}
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**attributes (p. 65)**

A list of attribute objects that meet the criteria of the request.

Type: Array of `Attribute (p. 160)` objects

**nextToken (p. 65)**

The `nextToken` value to include in a future `ListAttributes` request. When the results of a `ListAttributes` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String
Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example lists the attributes for container instances that have the stack=production attribute in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: madison.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 122
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListAttributes
X-Amz-Date: 20161222T181559Z
User-Agent: aws-cli/1.11.30 Python/2.7.12 Darwin/16.3.0 botocore/1.4.87
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "cluster": "default",
  "attributeName": "stack",
  "attributeValue": "production",
  "targetType": "container-instance"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 22 Dec 2016 18:16:00 GMT
Content-Type: application/x-amz-json-1.1
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListClusters

Returns a list of existing clusters.

Request Syntax

```json
{
   "maxResults": number,
   "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

maxResults (p. 68)

The maximum number of cluster results returned by ListClusters in paginated output. When this parameter is used, ListClusters only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListClusters request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListClusters returns up to 100 results and a nextToken value if applicable.

Type: Integer

Required: No

nextToken (p. 68)

The nextToken value returned from a previous paginated ListClusters request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value.

**Note**

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No

Response Syntax

```json
{
   "clusterArns": [ "string" ],
   "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

**clusterArns (p. 68)**

The list of full Amazon Resource Name (ARN) entries for each cluster associated with your account.

Type: Array of strings

**nextToken (p. 68)**

The `nextToken` value to include in a future `ListClusters` request. When the results of a `ListClusters` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

### Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

**ClientException**

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

### Example

In the following example or examples, the Authorization header contents (`AUTHPARAMS`) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

**Example**

This example request lists the clusters for your account.

**Sample Request**

```bash
POST / HTTP/1.1
```
Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 17:06:21 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 126
Connection: keep-alive
x-amzn-Request-Id: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "clusterArns": [
    "arn:aws:ecs:us-east-1:012345678910:cluster/My-cluster",
    "arn:aws:ecs:us-east-1:012345678910:cluster/default"
  ]
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListContainerInstances

Returns a list of container instances in a specified cluster. You can filter the results of a
ListContainerInstances operation with cluster query language statements inside the filter
parameter. For more information, see Cluster Query Language in the Amazon Elastic Container Service
Developer Guide.

Request Syntax

```
{
  "cluster": "string",
  "filter": "string",
  "maxResults": number,
  "nextToken": "string",
  "status": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common
Parameters (p. 221).

The request accepts the following data in JSON format.

cluster (p. 71)

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container
instances to list. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

filter (p. 71)

You can filter the results of a ListContainerInstances operation with cluster query language
statements. For more information, see Cluster Query Language in the Amazon Elastic Container
Service Developer Guide.

Type: String

Required: No

maxResults (p. 71)

The maximum number of container instance results returned by ListContainerInstances
in paginated output. When this parameter is used, ListContainerInstances only returns
maxResults results in a single page along with a nextToken response element. The remaining
results of the initial request can be seen by sending another ListContainerInstances request
with the returned nextToken value. This value can be between 1 and 100. If this parameter is
not used, then ListContainerInstances returns up to 100 results and a nextToken value if
applicable.

Type: Integer

Required: No
nextToken (p. 71)

The `nextToken` value returned from a previous paginated ListContainerInstances request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value.

**Note**

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No

status (p. 71)

Filters the container instances by status. For example, if you specify the `DRAINING` status, the results include only container instances that have been set to `DRAINING` using `UpdateContainerInstancesState` (p. 142). If you do not specify this parameter, the default is to include container instances set to `ACTIVE` and `DRAINING`.

Type: String

Valid Values: ACTIVE | DRAINING

Required: No

**Response Syntax**

```json
{
    "containerInstanceArns": [ "string" ],
    "nextToken": "string"
}
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

containerInstanceArns (p. 72)

The list of container instances with full ARN entries for each container instance associated with the specified cluster.

Type: Array of strings

nextToken (p. 72)

The `nextToken` value to include in a future ListContainerInstances request. When the results of a ListContainerInstances request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 223).
ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request lists the container instances in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListContainerInstances
X-Amz-Date: 20150429T175306Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListServices

Lists the services that are running in a specified cluster.

Request Syntax

```
{
  "cluster": "string",
  "launchType": "string",
  "maxResults": number,
  "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 75)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the services to list. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**launchType (p. 75)**

The launch type for services you want to list.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**maxResults (p. 75)**

The maximum number of service results returned by ListServices in paginated output. When this parameter is used, ListServices only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListServices request with the returned nextToken value. This value can be between 1 and 10. If this parameter is not used, then ListServices returns up to 10 results and a nextToken value if applicable.

Type: Integer

Required: No

**nextToken (p. 75)**

The nextToken value returned from a previous paginated ListServices request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value.

**Note**

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.
**Response Syntax**

```json
{
    "nextToken": "string",
    "serviceArns": [ "string" ]
}
```

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

- **nextToken (p. 76)**
  - The `nextToken` value to include in a future `ListServices` request. When the results of a `ListServices` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.
  - Type: String

- **serviceArns (p. 76)**
  - The list of full ARN entries for each service associated with the specified cluster.
  - Type: Array of strings

**Errors**

For information about the errors that are common to all actions, see [Common Errors (p. 223)](#).

- **ClientException**
  - These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.
  - HTTP Status Code: 400

- **ClusterNotFoundException**
  - The specified cluster could not be found. You can view your available clusters with `ListClusters (p. 68)`. Amazon ECS clusters are region-specific.
  - HTTP Status Code: 400

- **InvalidParameterException**
  - The specified parameter is invalid. Review the available parameters for the API request.
  - HTTP Status Code: 400

- **ServerException**
  - These errors are usually caused by a server issue.
HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don’t have to sign requests yourself.

Example

This example request lists the services in the default cluster.

Sample Request

```plaintext
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListServices
X-Amz-Date: 20150429T191342Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

Sample Response

```plaintext
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:13:42 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 138
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "serviceArns": [
        "arn:aws:ecs:us-east-1:012345678910:service/hello_world",
        "arn:aws:ecs:us-east-1:012345678910:service/ecs-simple-service"
    ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
ListTaskDefinitionFamilies

Returns a list of task definition families that are registered to your account (which may include task definition families that no longer have any ACTIVE task definition revisions).

You can filter out task definition families that do not contain any ACTIVE task definition revisions by setting the status parameter to ACTIVE. You can also filter the results with the familyPrefix parameter.

Request Syntax

```
{
   "familyPrefix": "string",
   "maxResults": number,
   "nextToken": "string",
   "status": "String"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

familyPrefix (p. 79)

The familyPrefix is a string that is used to filter the results of ListTaskDefinitionFamilies. If you specify a familyPrefix, only task definition family names that begin with the familyPrefix string are returned.

Type: String

Required: No

maxResults (p. 79)

The maximum number of task definition family results returned by ListTaskDefinitionFamilies in paginated output. When this parameter is used, ListTaskDefinitions only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListTaskDefinitionFamilies request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListTaskDefinitionFamilies returns up to 100 results and a nextToken value if applicable.

Type: Integer

Required: No

nextToken (p. 79)

The nextToken value returned from a previous paginated ListTaskDefinitionFamilies request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value.

Note

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.
Status (p. 79)

The task definition family status with which to filter the ListTaskDefinitionFamilies results. By default, both ACTIVE and INACTIVE task definition families are listed. If this parameter is set to ACTIVE, only task definition families that have an ACTIVE task definition revision are returned. If this parameter is set to INACTIVE, only task definition families that do not have any ACTIVE task definition revisions are returned. If you paginate the resulting output, be sure to keep the status value constant in each subsequent request.

Type: String

Valid Values: ACTIVE | INACTIVE | ALL

Response Syntax

```json
{
    "families": [ "string" ],
    "nextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

families (p. 80)

The list of task definition family names that match the ListTaskDefinitionFamilies request.

Type: Array of strings

nextToken (p. 80)

The nextToken value to include in a future ListTaskDefinitionFamilies request. When the results of a ListTaskDefinitionFamilies request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400
InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don’t have to sign requests yourself.

Example

This example request lists all of the task definition families in your account in the current region.

Sample Request

```plaintext
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTaskDefinitionFamilies
X-Amz-Date: 20150429T191650Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

Sample Response

```plaintext
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:16:51 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 270
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "families": [
        "console-sample-app",
        "ecs-demo",
        "ecs-private",
        "hello_world",
        "hpcc",
        "hpcc-t2-medium",
        "image-dedupe",
    ]
`
```
Example

This example request lists all of the task definition families in your account in the current region that begin with hpcc.

Sample Request

POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 24
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTaskDefinitionFamilies
X-Amz-Date: 20150429T191825Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "familyPrefix": "hpcc"
}

Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:18:25 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 38
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "families": [
    "hpcc",
    "hpcc-t2-medium"
  ]
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
ListTaskDefinitions

Returns a list of task definitions that are registered to your account. You can filter the results by family name with the familyPrefix parameter or by status with the status parameter.

Request Syntax

```json
{
    "familyPrefix": "string",
    "maxResults": number,
    "nextToken": "string",
    "sort": "string",
    "status": "String"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

familyPrefix (p. 84)

The full family name with which to filter the ListTaskDefinitions results. Specifying a familyPrefix limits the listed task definitions to task definition revisions that belong to that family.

Type: String

Required: No

maxResults (p. 84)

The maximum number of task definition results returned by ListTaskDefinitions in paginated output. When this parameter is used, ListTaskDefinitions only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListTaskDefinitions request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListTaskDefinitions returns up to 100 results and a nextToken value if applicable.

Type: Integer

Required: No

nextToken (p. 84)

The nextToken value returned from a previous paginated ListTaskDefinitions request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value.

Note

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No
sort (p. 84)

The order in which to sort the results. Valid values are ASC and DESC. By default (ASC), task definitions are listed lexicographically by family name and in ascending numerical order by revision so that the newest task definitions in a family are listed last. Setting this parameter to DESC reverses the sort order on family name and revision so that the newest task definitions in a family are listed first.

Type: String

Valid Values: ASC | DESC

Required: No

status (p. 84)

The task definition status with which to filter the ListTaskDefinitions results. By default, only ACTIVE task definitions are listed. By setting this parameter to INACTIVE, you can view task definitions that are INACTIVE as long as an active task or service still references them. If you paginate the resulting output, be sure to keep the status value constant in each subsequent request.

Type: String

Valid Values: ACTIVE | INACTIVE

Required: No

Response Syntax

{
   "nextToken": "string",
   "taskDefinitionArns": [ "string" ]
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 85)

The nextToken value to include in a future ListTaskDefinitions request. When the results of a ListTaskDefinitions request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

taskDefinitionArns (p. 85)

The list of task definition Amazon Resource Name (ARN) entries for the ListTaskDefinitions request.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).
ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerError

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request lists all of the task definitions in the hello_world family.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 31
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTaskDefinitions
X-Amz-Date: 20150429T192041Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "familyPrefix": "hello_world"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:20:41 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 695
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListTasks

Returns a list of tasks for a specified cluster. You can filter the results by family name, by a particular container instance, or by the desired status of the task with the family, containerInstance, and desiredStatus parameters.

Recently stopped tasks might appear in the returned results. Currently, stopped tasks appear in the returned results for at least one hour.

Request Syntax

```
{
  "cluster": "string",
  "containerInstance": "string",
  "desiredStatus": "string",
  "family": "string",
  "launchType": "string",
  "maxResults": number,
  "nextToken": "string",
  "serviceName": "string",
  "startedBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

cluster (p. 88)

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the tasks to list. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

containerInstance (p. 88)

The container instance ID or full ARN of the container instance with which to filter the ListTasks results. Specifying a containerInstance limits the results to tasks that belong to that container instance.

Type: String

Required: No

desiredStatus (p. 88)

The task desired status with which to filter the ListTasks results. Specifying a desiredStatus of STOPPED limits the results to tasks that Amazon ECS has set the desired status to STOPPED, which can be useful for debugging tasks that are not starting properly or have died or finished. The default status filter is RUNNING, which shows tasks that Amazon ECS has set the desired status to RUNNING.

Note

Although you can filter results based on a desired status of PENDING, this does not return any results because Amazon ECS never sets the desired status of a task to that value (only a task’s lastStatus may have a value of PENDING).
Type: String

Valid Values: RUNNING | PENDING | STOPPED

Required: No

family (p. 88)

The name of the family with which to filter the ListTasks results. Specifying a family limits the results to tasks that belong to that family.

Type: String

Required: No

launchType (p. 88)

The launch type for services you want to list.

Type: String

Valid Values: EC2 | FARGATE

Required: No

maxResults (p. 88)

The maximum number of task results returned by ListTasks in paginated output. When this parameter is used, ListTasks only returns maxResults results in a single page along with a nextToken response element. The remaining results of the initial request can be seen by sending another ListTasks request with the returned nextToken value. This value can be between 1 and 100. If this parameter is not used, then ListTasks returns up to 100 results and a nextToken value if applicable.

Type: Integer

Required: No

nextToken (p. 88)

The nextToken value returned from a previous paginated ListTasks request where maxResults was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the nextToken value.

Note

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No

serviceName (p. 88)

The name of the service with which to filter the ListTasks results. Specifying a serviceName limits the results to tasks that belong to that service.

Type: String

Required: No

startedBy (p. 88)

The startedBy value with which to filter the task results. Specifying a startedBy value limits the results to tasks that were started with that value.

Type: String
Response Syntax

```
{
  "nextToken": "string",
  "taskArns": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

nextToken (p. 90)

The `nextToken` value to include in a future `ListTasks` request. When the results of a `ListTasks` request exceed `maxResults`, this value can be used to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

taskArns (p. 90)

The list of task ARN entries for the `ListTasks` request.

Type: Array of strings

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with `ListClusters` (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500
ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 75). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request lists all of the tasks in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTasks
X-Amz-Date: 20150429T192615Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:26:16 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 330
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
{
    "taskArns": [
        "arn:aws:ecs:us-east-1:012345678910:task/0b69d5c0-d655-4695-98cd-5d2d526d9d5a",
        "arn:aws:ecs:us-east-1:012345678910:task/51a01bdf-d00e-487e-ab14-7645330b6207",
        "arn:aws:ecs:us-east-1:012345678910:task/b0b28bb8-2be3-4810-b52b-88df129d893c",
        "arn:aws:ecs:us-east-1:012345678910:task/c09f0188-7f87-4b0f-bfc3-16296622b6fe"
    ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutAttributes

Create or update an attribute on an Amazon ECS resource. If the attribute does not exist, it is created. If the attribute exists, its value is replaced with the specified value. To delete an attribute, use DeleteAttributes (p. 14). For more information, see Attributes in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```
{
  "attributes": [
    {
      "name": "string",
      "targetId": "string",
      "targetType": "string",
      "value": "string"
    }
  ],
  "cluster": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

attributes (p. 93)

The attributes to apply to your resource. You can specify up to 10 custom attributes per resource. You can specify up to 10 attributes in a single call.

Type: Array of Attribute (p. 160) objects

Required: Yes

cluster (p. 93)

The short name or full Amazon Resource Name (ARN) of the cluster that contains the resource to apply attributes. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

Response Syntax

```
{
  "attributes": [
    {
      "name": "string",
      "targetId": "string",
      "targetType": "string",
      "value": "string"
    }
  ]
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

attributes (p. 93)

The attributes applied to your resource.

Type: Array of Attribute (p. 160) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

AttributeLimitExceededException

You can apply up to 10 custom attributes per resource. You can view the attributes of a resource with ListAttributes (p. 64). You can remove existing attributes on a resource with DeleteAttributes (p. 14).

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

TargetNotFoundException

The specified target could not be found. You can view your available container instances with ListContainerInstances (p. 71). Amazon ECS container instances are cluster-specific and region-specific.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.
Example

This example applies an attribute with the name `stack` and the value `production` to a container instance.

Sample Request

```
POST / HTTP/1.1
Host: madison.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 192
X-Amz-Target: AmazonEC2ContainerServiceV20141113.PutAttributes
X-Amz-Date: 20161222T180005Z
User-Agent: aws-cli/1.11.30 Python/2.7.12 Darwin/16.3.0 botocore/1.4.87
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "default-gamma",
    "attributes": [
        {
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
            "name": "stack",
            "value": "production"
        }
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 22 Dec 2016 18:00:06 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 158
Connection: keep-alive
x-amzn-RequestId: 7835c1be-c870-11e6-a3b0-295902c79de2

{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
See Also

- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
RegisterContainerInstance

**Note**
This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

Registers an EC2 instance into the specified cluster. This instance becomes available to place containers on.

**Request Syntax**

```json
{
    "attributes": [
        {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
        }
    ],
    "cluster": "string",
    "containerInstanceArn": "string",
    "instanceIdentityDocument": "string",
    "instanceIdentityDocumentSignature": "string",
    "totalResources": [
        {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": ["string"],
            "type": "string"
        }
    ],
    "versionInfo": {
        "agentHash": "string",
        "agentVersion": "string",
        "dockerVersion": "string"
    }
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**attributes (p. 97)**

The container instance attributes that this container instance supports.

Type: Array of Attribute (p. 160) objects

Required: No

**cluster (p. 97)**

The short name or full Amazon Resource Name (ARN) of the cluster with which to register your container instance. If you do not specify a cluster, the default cluster is assumed.
Type: String
Required: No
**containerInstanceArn (p. 97)**
The ARN of the container instance (if it was previously registered).
Type: String
Required: No
**instanceIdentityDocument (p. 97)**
The instance identity document for the EC2 instance to register. This document can be found by running the following command from the instance:

```
```
Type: String
Required: No
**instanceIdentityDocumentSignature (p. 97)**
The instance identity document signature for the EC2 instance to register. This signature can be found by running the following command from the instance:

```
```
Type: String
Required: No
**totalResources (p. 97)**
The resources available on the instance.
Type: Array of Resource (p. 201) objects
Required: No
**versionInfo (p. 97)**
The version information for the Amazon ECS container agent and Docker daemon running on the container instance.
Type: VersionInfo (p. 218) object
Required: No

**Response Syntax**

```
{
   "containerInstance": {
      "agentConnected": boolean,
      "agentUpdateStatus": "string",
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
         }
      ],
      "id": "string",
      "status": "string",
   }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

containerInstance (p. 98)

The container instance that was registered.

Type: ContainerInstance (p. 174) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).
**ClientException**

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

**See Also**

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
RegisterTaskDefinition

Registers a new task definition from the supplied family and containerDefinitions. Optionally, you can add data volumes to your containers with the volumes parameter. For more information about task definition parameters and defaults, see Amazon ECS Task Definitions in the Amazon Elastic Container Service Developer Guide.

You can specify an IAM role for your task with the taskRoleArn parameter. When you specify an IAM role for a task, its containers can then use the latest versions of the AWS CLI or SDKs to make API requests to the AWS services that are specified in the IAM policy associated with the role. For more information, see IAM Roles for Tasks in the Amazon Elastic Container Service Developer Guide.

You can specify a Docker networking mode for the containers in your task definition with the networkMode parameter. The available network modes correspond to those described in Network settings in the Docker run reference. If you specify the awsvpc network mode, the task is allocated an Elastic Network Interface, and you must specify a NetworkConfiguration (p. 195) when you create a service or run a task with the task definition. For more information, see Task Networking in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
  "containerDefinitions": [
    {
      "command": [ "string" ],
      "cpu": number,
      "disableNetworking": boolean,
      "dnsSearchDomains": [ "string" ],
      "dnsServers": [ "string" ],
      "dockerLabels": {
        "string": "string"
      },
      "dockerSecurityOptions": [ "string" ],
      "entryPoint": [ "string" ],
      "environment": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "essential": boolean,
      "extraHosts": [
        {
          "hostname": "string",
          "ipAddress": "string"
        }
      ],
      "hostname": "string",
      "image": "string",
      "links": [ "string" ],
      "linuxParameters": {
        "capabilities": {
          "add": [ "string" ],
          "drop": [ "string" ]
        },
        "devices": [
          {
            "containerPath": "string",
            "hostPath": "string",
            "permissions": [ "string" ]
          }
        ]
      }
    }
  ]
}
```
{
    "initProcessEnabled": boolean
},
"logConfiguration": {
    "logDriver": "string",
    "options": {
        "string": "string"
    }
},
"memory": number,
"memoryReservation": number,
"mountPoints": [
    {
        "containerPath": "string",
        "readOnly": boolean,
        "sourceVolume": "string"
    }
],
"name": "string",
"portMappings": [
    {
        "containerPort": number,
        "hostPort": number,
        "protocol": "string"
    }
],
"privileged": boolean,
"readonlyRootFilesystem": boolean,
"ulimits": [
    {
        "hardLimit": number,
        "name": "string",
        "softLimit": number
    }
],
"user": "string",
"volumesFrom": [
    {
        "readOnly": boolean,
        "sourceContainer": "string"
    }
],
"workingDirectory": "string"
},
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"memory": "string",
"networkMode": "string",
"placementConstraints": [
    {
        "expression": "string",
        "type": "string"
    }
],
"requiresCompatibilities": ["string"],
"taskRoleArn": "string",
"volumes": [
    {
        "host": {
            "sourcePath": "string"
        },
        "name": "string"
    }
]
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

containerDefinitions (p. 101)

A list of container definitions in JSON format that describe the different containers that make up your task.

Type: Array of ContainerDefinition (p. 166) objects

Required: Yes

cpu (p. 101)

The number of cpu units used by the task. If using the EC2 launch type, this field is optional and any value can be used.

Note
Task-level CPU and memory parameters are ignored for Windows containers. We recommend specifying container-level resources for Windows containers.

If you are using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the memory parameter:

- 256 (.25 vCPU) - Available memory values: 0.5GB, 1GB, 2GB
- 512 (.5 vCPU) - Available memory values: 1GB, 2GB, 3GB, 4GB
- 1024 (1 vCPU) - Available memory values: 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB
- 2048 (2 vCPU) - Available memory values: Between 4GB and 16GB in 1GB increments
- 4096 (4 vCPU) - Available memory values: Between 8GB and 30GB in 1GB increments

Type: String

Required: No

executionRoleArn (p. 101)

The Amazon Resource Name (ARN) of the task execution role that the Amazon ECS container agent and the Docker daemon can assume.

Type: String

Required: No

family (p. 101)

You must specify a family for a task definition, which allows you to track multiple versions of the same task definition. The family is used as a name for your task definition. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

Type: String

Required: Yes

memory (p. 101)

The amount (in MiB) of memory used by the task. If using the EC2 launch type, this field is optional and any value can be used.
Note
Task-level CPU and memory parameters are ignored for Windows containers. We recommend specifying container-level resources for Windows containers.

If you are using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the cpu parameter:

- 0.5GB, 1GB, 2GB - Available cpu values: 256 (.25 vCPU)
- 1GB, 2GB, 3GB, 4GB - Available cpu values: 512 (.5 vCPU)
- 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB - Available cpu values: 1024 (1 vCPU)
- Between 4GB and 16GB in 1GB increments - Available cpu values: 2048 (2 vCPU)
- Between 8GB and 30GB in 1GB increments - Available cpu values: 4096 (4 vCPU)

Type: String
Required: No

networkMode (p. 101)

The Docker networking mode to use for the containers in the task. The valid values are none, bridge, awsvpc, and host. The default Docker network mode is bridge. If using the Fargate launch type, the awsvpc network mode is required. If using the EC2 launch type, any network mode can be used. If the network mode is set to none, you can't specify port mappings in your container definitions, and the task's containers do not have external connectivity. The host and awsvpc network modes offer the highest networking performance for containers because they use the EC2 network stack instead of the virtualized network stack provided by the bridge mode.

With the host and awsvpc network modes, exposed container ports are mapped directly to the corresponding host port (for the host network mode) or the attached elastic network interface port (for the awsvpc network mode), so you cannot take advantage of dynamic host port mappings.

If the network mode is awsvpc, the task is allocated an Elastic Network Interface, and you must specify a NetworkConfiguration (p. 195) when you create a service or run a task with the task definition. For more information, see Task Networking in the Amazon Elastic Container Service Developer Guide.

If the network mode is host, you can't run multiple instantiations of the same task on a single container instance when port mappings are used.

Docker for Windows uses different network modes than Docker for Linux. When you register a task definition with Windows containers, you must not specify a network mode.

For more information, see Network settings in the Docker run reference.

Type: String
Valid Values: bridge | host | awsvpc | none
Required: No

placementConstraints (p. 101)

An array of placement constraint objects to use for the task. You can specify a maximum of 10 constraints per task (this limit includes constraints in the task definition and those specified at run time).

Type: Array of TaskDefinitionPlacementConstraint (p. 215) objects
Required: No

requiresCompatibilities (p. 101)

The launch type required by the task. If no value is specified, it defaults to EC2.
Response Syntax

```
{
    "taskDefinition": {
        "compatibilities": [ "string" ],
        "containerDefinitions": [
            {
                "command": [ "string" ],
                "cpu": number,
                "disableNetworking": boolean,
                "dnsSearchDomains": [ "string" ],
                "dnsServers": [ "string" ],
                "dockerLabels": {
                    "string": "string"
                },
                "dockerSecurityOptions": [ "string" ],
                "entryPoint": [ "string" ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "essential": boolean,
                "extraHosts": [
                    {
                        "hostname": "string",
                        "ipAddress": "string"
                    }
                ],
                "hostname": "string",
                "image": "string",
                "links": [ "string" ],
                "linuxParameters": {
                    "capabilities": {
                        "add": [ "string" ],
                        "drop": [ "string" ]
                    },
                    "devices": [ ...
```

{
    "containerPath": "string",
    "hostPath": "string",
    "permissions": [ "string" ]
},
    "initProcessEnabled": boolean
},
    "logConfiguration": {
        "logDriver": "string",
        "options": {
            "string": "string"
        }
    },
    "memory": number,
    "memoryReservation": number,
    "mountPoints": [ {
        "containerPath": "string",
        "readOnly": boolean,
        "sourceVolume": "string"
    } ],
    "name": "string",
    "portMappings": [ {
        "containerPort": number,
        "hostPort": number,
        "protocol": "string"
    } ],
    "privileged": boolean,
    "readonlyRootFilesystem": boolean,
    "ulimits": [ {
        "hardLimit": number,
        "name": "string",
        "softLimit": number
    } ],
    "user": "string",
    "volumesFrom": [ {
        "readOnly": boolean,
        "sourceContainer": "string"
    } ],
    "workingDirectory": "string"
},
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"memory": "string",
"networkMode": "string",
"placementConstraints": [ {
    "expression": "string",
    "type": "string"
} ],
"requiresAttributes": [ {
    "name": "string",
    "targetId": "string",
    "targetType": "string",
    "value": "string"
} ]}
"value": "string"

"requiresCompatibilities": [ "string" ],
"revision": number,
"status": "string",
"taskDefinitionArn": "string",
"taskRoleArn": "string",
"volumes": [
  {
    "host": {
      "sourcePath": "string"
    },
    "name": "string"
  }
]

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

taskDefinition (p. 105)

The full description of the registered task definition.

Type: TaskDefinition (p. 211) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don’t have to sign requests yourself.

**Example**

This example request registers a task definition in the `hello_world` family with the `host` networking mode.

**Sample Request**

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 486
X-Amz-Target: AmazonEC2ContainerServiceV20141113.RegisterTaskDefinition
X-Amz-Date: 20150429T193109Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "networkMode": "host",
  "containerDefinitions": [
    {
      "name": "wordpress",
      "links": [
        "mysql"
      ],
      "image": "wordpress",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80
        }
      ],
      "memory": 500,
      "cpu": 10
    },
    {
      "name": "mysql",
      "image": "mysql",
      "cpu": 10,
      "environment": [
        {
          "name": "MYSQL_ROOT_PASSWORD",
          "value": "password"
        }
      ],
      "memory": 500,
      "essential": true
    }
  ],
  "family": "hello_world"
}
```

**Sample Response**

```
HTTP/1.1 200 OK
Server: Server
```
Amazon EC2 Container Service API Reference

See Also

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

API Version 2014-11-13

109
See Also

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
RunTask

Starts a new task using the specified task definition.

You can allow Amazon ECS to place tasks for you, or you can customize how Amazon ECS places tasks using placement constraints and placement strategies. For more information, see Scheduling Tasks in the Amazon Elastic Container Service Developer Guide.

Alternatively, you can use StartTask (p. 119) to use your own scheduler or place tasks manually on specific container instances.

Request Syntax

```json
{
    "cluster": "string",
    "count": number,
    "group": "string",
    "launchType": "string",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
        }
    },
    "overrides": {
        "containerOverrides": [
            {
                "command": [ "string" ],
                "cpu": number,
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "memory": number,
                "memoryReservation": number,
                "name": "string"
            }
        ],
        "executionRoleArn": "string",
        "taskRoleArn": "string"
    },
    "placementConstraints": [
        {
            "expression": "string",
            "type": "string"
        }
    ],
    "placementStrategy": [
        {
            "field": "string",
            "type": "string"
        }
    ],
    "platformVersion": "string",
    "startedBy": "string",
    "taskDefinition": "string"
}
```
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 111)**

The short name or full Amazon Resource Name (ARN) of the cluster on which to run your task. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**count (p. 111)**

The number of instantiations of the specified task to place on your cluster. You can specify up to 10 tasks per call.

Type: Integer

Required: No

**group (p. 111)**

The name of the task group to associate with the task. The default value is the family name of the task definition (for example, family:my-family-name).

Type: String

Required: No

**launchType (p. 111)**

The launch type on which to run your task.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**networkConfiguration (p. 111)**

The network configuration for the task. This parameter is required for task definitions that use the awsvpc network mode to receive their own Elastic Network Interface, and it is not supported for other network modes. For more information, see Task Networking in the Amazon Elastic Container Service Developer Guide.

Type: NetworkConfiguration (p. 195) object

Required: No

**overrides (p. 111)**

A list of container overrides in JSON format that specify the name of a container in the specified task definition and the overrides it should receive. You can override the default command for a container (that is specified in the task definition or Docker image) with a command override. You can also override existing environment variables (that are specified in the task definition or Docker image) on a container or add new environment variables to it with an environment override.
Note
A total of 8192 characters are allowed for overrides. This limit includes the JSON formatting characters of the override structure.

Type: TaskOverride (p. 216) object

Required: No

**placementConstraints (p. 111)**

An array of placement constraint objects to use for the task. You can specify up to 10 constraints per task (including constraints in the task definition and those specified at run time).

Type: Array of PlacementConstraint (p. 197) objects

Required: No

**placementStrategy (p. 111)**

The placement strategy objects to use for the task. You can specify a maximum of five strategy rules per task.

Type: Array of PlacementStrategy (p. 198) objects

Required: No

**platformVersion (p. 111)**

The platform version on which to run your task. If one is not specified, the latest version is used by default.

Type: String

Required: No

**startedBy (p. 111)**

An optional tag specified when a task is started. For example if you automatically trigger a task to run a batch process job, you could apply a unique identifier for that job to your task with the startedBy parameter. You can then identify which tasks belong to that job by filtering the results of a ListTasks (p. 88) call with the startedBy value. Up to 36 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

If a task is started by an Amazon ECS service, then the startedBy parameter contains the deployment ID of the service that starts it.

Type: String

Required: No

**taskDefinition (p. 111)**

The family and revision(family:revision) or full ARN of the task definition to run. If a revision is not specified, the latest ACTIVE revision is used.

Type: String

Required: Yes

**Response Syntax**

```json
{
    ...
}```
"failures": [    
    
    ],
"tasks": [    
    
    ],
"clusterArn": "string",
"connectivity": "string",
"connectivityAt": number,
"containerInstanceArn": "string",
"containers": [    
    
    ],
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
    
    },
"containerOverrides": [    
    
    ],
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
    
    },
"containerOverrides": [    
    
    ],
"cpu": "string",
"createdAt": number,
"value": "string"
},
"memory": number,
"memoryReservation": number,
"name": "string"
},
"executionRoleArn": "string",
"taskRoleArn": "string"
},
"platformVersion": "string",
"pullStartedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "string",
"stoppedAt": number,
"stoppedReason": "string",
"stoppingAt": number,
"taskArn": "string",
"taskDefinitionArn": "string",
"version": number
]
}]

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**failures (p. 113)**

Any failures associated with the call.

Type: Array of [Failure (p. 184)] objects

**tasks (p. 113)**

A full description of the tasks that were run. The tasks that were successfully placed on your cluster are described here.

Type: Array of [Task (p. 207)] objects

Errors

For information about the errors that are common to all actions, see [Common Errors (p. 223)].

**AccessDeniedException**

You do not have authorization to perform the requested action.

HTTP Status Code: 400

**BlockedException**

Your AWS account has been blocked. [Contact AWS Customer Support](#) for more information.

HTTP Status Code: 400
ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version does not satisfy the task definition's required capabilities.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version does not exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task is not supported in this region.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request runs the latest ACTIVE revision of the hello_world task definition family in the default cluster.
Sample Request

POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 45
X-Amz-Target: AmazonEC2ContainerServiceV20141113.RunTask
X-Amz-Date: 20161121T215740Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Authorization: AUTHPARAMS

{
    "count": 1,
    "taskDefinition": "hello_world"
}

Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 21:57:40 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1025
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{  
    "failures": [],  
    "tasks": [  
        {  
            "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",  
            "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-instance/4c543eed-f83f-47da-b1d8-3d23f1da4c64",  
            "containers": [  
                {  
                    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e76594d4-27e1-4c74-98b5-46a6435eb769",  
                    "lastStatus": "PENDING",  
                    "name": "wordpress",  
                    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/fdf2c302-468c-4e55-b884-5331d816e7fb"  
                },  
                {  
                    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/b19106ea-4fa8-4f1d-9767-96922c82b070",  
                    "lastStatus": "PENDING",  
                    "name": "mysql",  
                    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/fdf2c302-468c-4e55-b884-5331d816e7fb"  
                }  
            ],  
            "createdAt": 1479765460.842,  
            "desiredStatus": "RUNNING",  
            "lastStatus": "PENDING",  
            "overrides": [  
                {  
                    "name": "wordpress"  
                },  
                {  
                    "name": "mysql"  
                }  
            ]  
        }  
    ]
}
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
StartTask

Starts a new task from the specified task definition on the specified container instance or instances.

Alternatively, you can use RunTask (p. 111) to place tasks for you. For more information, see Scheduling Tasks in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```
{
    "cluster": "string",
    "containerInstances": [ "string" ],
    "group": "string",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
        }
    },
    "overrides": {
        "containerOverrides": [
            {
                "command": [ "string" ],
                "cpu": number,
                "environment": [ {
                    "name": "string",
                    "value": "string"
                } ],
                "memory": number,
                "memoryReservation": number,
                "name": "string"
            }
        ],
        "executionRoleArn": "string",
        "taskRoleArn": "string"
    },
    "startedBy": "string",
    "taskDefinition": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 119)**

The short name or full Amazon Resource Name (ARN) of the cluster on which to start your task. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No
containerInstances (p. 119)
The container instance IDs or full ARN entries for the container instances on which you would like to place your task. You can specify up to 10 container instances.

Type: Array of strings
Required: Yes

group (p. 119)
The name of the task group to associate with the task. The default value is the family name of the task definition (for example, family:my-family-name).

Type: String
Required: No

networkConfiguration (p. 119)
The VPC subnet and security group configuration for tasks that receive their own Elastic Network Interface by using the `awsvpc` networking mode.

Type: NetworkConfiguration (p. 195) object
Required: No

overrides (p. 119)
A list of container overrides in JSON format that specify the name of a container in the specified task definition and the overrides it should receive. You can override the default command for a container (that is specified in the task definition or Docker image) with a command override. You can also override existing environment variables (that are specified in the task definition or Docker image) on a container or add new environment variables to it with an environment override.

**Note**
A total of 8192 characters are allowed for overrides. This limit includes the JSON formatting characters of the override structure.

Type: TaskOverride (p. 216) object
Required: No

startedBy (p. 119)
An optional tag specified when a task is started. For example if you automatically trigger a task to run a batch process job, you could apply a unique identifier for that job to your task with the `startedBy` parameter. You can then identify which tasks belong to that job by filtering the results of a ListTasks (p. 88) call with the `startedBy` value. Up to 36 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.

If a task is started by an Amazon ECS service, then the `startedBy` parameter contains the deployment ID of the service that starts it.

Type: String
Required: No

taskDefinition (p. 119)
The family and revision (family:revision) or full ARN of the task definition to start. If a revision is not specified, the latest ACTIVE revision is used.
Type: String
Required: Yes

Response Syntax

```json
{
   "failures": [
      {
         "arn": "string",
         "reason": "string"
      }
   ],
   "tasks": [
      {
         "attachments": [
            {
               "details": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "id": "string",
               "status": "string",
               "type": "string"
            }
         ],
         "clusterArn": "string",
         "connectivity": "string",
         "connectivityAt": number,
         "containerInstanceArn": "string",
         "containers": [
            {
               "containerArn": "string",
               "exitCode": number,
               "lastStatus": "string",
               "name": "string",
               "networkBindings": [
                  {
                     "bindIP": "string",
                     "containerPort": number,
                     "hostPort": number,
                     "protocol": "string"
                  }
               ],
               "networkInterfaces": [
                  {
                     "attachmentId": "string",
                     "ipv6Address": "string",
                     "privateI Pv4Address": "string"
                  }
               ],
               "reason": "string",
               "taskArn": "string"
            }
         ],
         "cpu": "string",
         "createdAt": number,
         "desiredStatus": "string",
         "executionStoppedAt": number,
         "group": "string",
         "lastStatus": "string"
      }
   }
}
```
"launchType": "string",
"memory": "string",
"overrides": {
  "containerOverrides": [
    {
      "command": [ "string" ],
      "cpu": number,
      "environment": [ {
        "name": "string",
        "value": "string"
      } ],
      "memory": number,
      "memoryReservation": number,
      "name": "string"
    }
  ],
  "executionRoleArn": "string",
  "taskRoleArn": "string"
},
"platformVersion": "string",
"pullStartedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "string",
"stoppedAt": number,
"stoppedReason": "string",
"stoppingAt": number,
"taskArn": "string",
"taskDefinitionArn": "string",
"version": number
}
]

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

failures (p. 121)

Any failures associated with the call.

Type: Array of Failure (p. 184) objects

tasks (p. 121)

A full description of the tasks that were started. Each task that was successfully placed on your container instances is described.

Type: Array of Task (p. 207) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).
ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request starts the latest ACTIVE revision of the hello_world task definition family in the default cluster on the container instance with the ID 4c543eed-f83f-47da-b1d8-3d23f1da4c64.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 97
X-Amz-Target: AmazonEC2ContainerServiceV20141113.StartTask
X-Amz-Date: 20161121T220032Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "containerInstances": [
    "4c543eed-f83f-47da-b1d8-3d23f1da4c64"
  ],
  "taskDefinition": "hello_world"
}
```
Sample Response

HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 22:00:32 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1025
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "failures": [],
    "tasks": [
        {
            "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-instance/4c543eed-f83f-47da-b1d8-3d23f1da4c64",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e76594d4-27e1-4c74-98b5-46a6435eb769",
                    "lastStatus": "PENDING",
                    "name": "wordpress",
                    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/fdf2c302-468c-4e55-b884-5331d816e7fb"
                },
                {
                    "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/b19106ea-4fa8-4f1d-9767-96922c82b070",
                    "lastStatus": "PENDING",
                    "name": "mysql",
                    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/fdf2c302-468c-4e55-b884-5331d816e7fb"
                }
            ],
            "createdAt": 1479765460.842,
            "desiredStatus": "RUNNING",
            "lastStatus": "PENDING",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "wordpress"
                    },
                    {
                        "name": "mysql"
                    }
                ],
                "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/fdf2c302-468c-4e55-b884-5331d816e7fb",
                "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:6",
                "version": 1
            }
        }
    ]
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
StopTask

Stops a running task.

When StopTask (p. 126) is called on a task, the equivalent of docker stop is issued to the containers running in the task. This results in a SIGTERM and a default 30-second timeout, after which SIGHUP is sent and the containers are forcibly stopped. If the container handles the SIGTERM gracefully and exits within 30 seconds from receiving it, no SIGHUP is sent.

**Note**
The default 30-second timeout can be configured on the Amazon ECS container agent with the ECS_CONTAINER_STOP_TIMEOUT variable. For more information, see Amazon ECS Container Agent Configuration in the Amazon Elastic Container Service Developer Guide.

### Request Syntax

```
{
  "cluster": "string",
  "reason": "string",
  "task": "string"
}
```

### Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 126)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task to stop. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**reason (p. 126)**

An optional message specified when a task is stopped. For example, if you are using a custom scheduler, you can use this parameter to specify the reason for stopping the task here, and the message appears in subsequent DescribeTasks (p. 57) API operations on this task. Up to 255 characters are allowed in this message.

Type: String

Required: No

**task (p. 126)**

The task ID or full ARN entry of the task to stop.

Type: String

Required: Yes
Response Syntax

```json
{
   "task": {
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "clusterArn": "string",
      "connectivity": "string",
      "connectivityAt": number,
      "containerInstanceArn": "string",
      "containers": [
         {
            "containerArn": "string",
            "exitCode": number,
            "lastStatus": "string",
            "name": "string",
            "networkBindings": [
               {
                  "bindIP": "string",
                  "containerPort": number,
                  "hostPort": number,
                  "protocol": "string"
               }
            ],
            "networkInterfaces": [
               {
                  "attachmentId": "string",
                  "ipv6Address": "string",
                  "privateIpv4Address": "string"
               }
            ],
            "reason": "string",
            "taskArn": "string"
         }
      ],
      "cpu": "string",
      "createdAt": number,
      "desiredStatus": "string",
      "executionStoppedAt": number,
      "group": "string",
      "lastStatus": "string",
      "launchType": "string",
      "memory": "string",
      "overrides": {
         "containerOverrides": [
            {
               "command": [ "string" ],
               "cpu": number,
               "environment": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ]
            }
         ]
      }
   }
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

task (p. 127)
The task that was stopped.

Type: Task (p. 207) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.
HTTP Status Code: 500

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request stops a task with the ID a126249b-b7e4-4b06-9d8f-1b56e75a99b5 in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 88
X-Amz-Target: AmazonEC2ContainerServiceV20141113.StopTask
X-Amz-Date: 20161121T220318Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
  "task": "1dc5c17a-422b-4dc4-b493-371970c6c4d6"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 22:03:18 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1260
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
{
  "task": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "containers": [
      {
        "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/4df26bb4-f057-467b-a079-961675296e64",
        "lastStatus": "RUNNING",
        "name": "simple-app",
        "networkBindings": [
          {
            "bindIP": "0.0.0.0",
            "containerPort": 80,
          }
        ]
      }
    ]
  }
```
"hostPort": 32774,
  "protocol": "tcp"
},
"taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6"
},
{ "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e0906f7-7361-4c87-8ab9-8d073bbdbcb9",
  "lastStatus": "RUNNING",
  "name": "busybox",
  "networkBindings": [],
  "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6"
}
],
"createdAt": 1476822811.295,
"desiredStatus": "STOPPED",
"lastStatus": "RUNNING",
"overrides": {
  "containerOverrides": [
    { "name": "simple-app" },
    { "name": "busybox" }
  ]
},
"startedAt": 1476822833.998,
"startedBy": "ecs-svc/9223370560032507596",
"stoppedReason": "Task stopped by user",
"taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6",
"version": 0
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
SubmitContainerStateChange

**Note**
This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a container changed states.

**Request Syntax**

```json
{
   "cluster": "string",
   "containerName": "string",
   "exitCode": number,
   "networkBindings": [
       {
           "bindIP": "string",
           "containerPort": number,
           "hostPort": number,
           "protocol": "string"
       }
   ],
   "reason": "string",
   "status": "string",
   "task": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 131)**

The short name or full ARN of the cluster that hosts the container.

Type: String

Required: No

**containerName (p. 131)**

The name of the container.

Type: String

Required: No

**exitCode (p. 131)**

The exit code returned for the state change request.

Type: Integer

Required: No

**networkBindings (p. 131)**

The network bindings of the container.
Type: Array of NetworkBinding (p. 194) objects

Required: No

reason (p. 131)
The reason for the state change request.
Type: String
Required: No

status (p. 131)
The status of the state change request.
Type: String
Required: No

task (p. 131)
The task ID or full Amazon Resource Name (ARN) of the task that hosts the container.
Type: String
Required: No

Response Syntax

{  
  "acknowledgment": "string"
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

acknowledgment (p. 132)
Acknowledgement of the state change.
Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

AccessDeniedException
You do not have authorization to perform the requested action.

HTTP Status Code: 400

ClientException
These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.
HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

**See Also**

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
SubmitTaskStateChange

**Note**
This action is only used by the Amazon ECS agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a task changed states.

**Request Syntax**

```json
{
  "attachments": [
    {
      "attachmentArn": "string",
      "status": "string"
    }
  ],
  "cluster": "string",
  "containers": [
    {
      "containerName": "string",
      "exitCode": number,
      "networkBindings": [
        {
          "bindIP": "string",
          "containerPort": number,
          "hostPort": number,
          "protocol": "string"
        }
      ],
      "reason": "string",
      "status": "string"
    }
  ],
  "executionStoppedAt": number,
  "pullStartedAt": number,
  "pullStoppedAt": number,
  "reason": "string",
  "status": "string",
  "task": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**attachments (p. 134)**

Any attachments associated with the state change request.

Type: Array of AttachmentStateChange (p. 159) objects

Required: No

**cluster (p. 134)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task.
containers (p. 134)
Any containers associated with the state change request.
Type: Array of ContainerStateChange (p. 179) objects
Required: No

executionStoppedAt (p. 134)
The Unix timestamp for when the task execution stopped.
Type: Timestamp
Required: No

pullStartedAt (p. 134)
The Unix time stamp for when the container image pull began.
Type: Timestamp
Required: No

pullStoppedAt (p. 134)
The Unix time stamp for when the container image pull completed.
Type: Timestamp
Required: No

reason (p. 134)
The reason for the state change request.
Type: String
Required: No

status (p. 134)
The status of the state change request.
Type: String
Required: No

task (p. 134)
The task ID or full ARN of the task in the state change request.
Type: String
Required: No

Response Syntax

```json
{
  "acknowledgment": "string"
}
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

acknowledgment (p. 135)

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

AccessDeniedException

You do not have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UpdateContainerAgent

Updates the Amazon ECS container agent on a specified container instance. Updating the Amazon ECS container agent does not interrupt running tasks or services on the container instance. The process for updating the agent differs depending on whether your container instance was launched with the Amazon ECS-optimized AMI or another operating system.

UpdateContainerAgent requires the Amazon ECS-optimized AMI or Amazon Linux with the `ecs-init` service installed and running. For help updating the Amazon ECS container agent on other operating systems, see Manually Updating the Amazon ECS Container Agent in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
    "cluster": "string",
    "containerInstance": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 137)**

The short name or full Amazon Resource Name (ARN) of the cluster that your container instance is running on. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**containerInstance (p. 137)**

The container instance ID or full ARN entries for the container instance on which you would like to update the Amazon ECS container agent.

Type: String

Required: Yes

Response Syntax

```json
{
    "containerInstance": {
        "agentConnected": boolean,
        "agentUpdateStatus": "string",
        "attachments": [
            {
                "details": [
                    {
                    
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

containerInstance (p. 137)

The container instance for which the container agent was updated.

Type: ContainerInstance (p. 174) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

MissingVersionException

Amazon ECS is unable to determine the current version of the Amazon ECS container agent on the container instance and does not have enough information to proceed with an update. This could be because the agent running on the container instance is an older or custom version that does not use our version information.

HTTP Status Code: 400

NoUpdateAvailableException

There is no update available for this Amazon ECS container agent. This could be because the agent is already running the latest version, or it is so old that there is no update path to the current version.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UpdateInProgressException

There is already a current Amazon ECS container agent update in progress on the specified container instance. If the container agent becomes disconnected while it is in a transitional stage, such as PENDING or STAGING, the update process can get stuck in that state. However, when the agent reconnects, it resumes where it stopped previously.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

**Example**

This example updates the container agent version for the container instance with the ID 53ac7152-dcd1-4102-81f5-208962864132 in the update cluster.

**Sample Request**

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 82
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateContainerAgent
X-Amz-Date: 20150528T152756Z
User-Agent: aws-cli/1.7.30 Python/2.7.9 Darwin/14.3.0
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
{
    "cluster": "update",
    "containerInstance": "53ac7152-dcd1-4102-81f5-208962864132"
}
```

**Sample Response**

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 28 May 2015 15:27:54 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1033
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "containerInstance": {
        "agentConnected": true,
        "agentUpdateStatus": "PENDING",
        ...
        "versionInfo": {
            "agentHash": "4023248",
            "agentVersion": "1.0.0",
            "dockerVersion": "DockerVersion: 1.5.0"
        }
    }
}
```

**See Also**

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
UpdateContainerInstancesState

Modifies the status of an Amazon ECS container instance.

You can change the status of a container instance to DRAINING to manually remove an instance from a cluster, for example to perform system updates, update the Docker daemon, or scale down the cluster size.

When you set a container instance to DRAINING, Amazon ECS prevents new tasks from being scheduled for placement on the container instance and replacement service tasks are started on other container instances in the cluster if the resources are available. Service tasks on the container instance that are in the PENDING state are stopped immediately.

Service tasks on the container instance that are in the RUNNING state are stopped and replaced according to the service’s deployment configuration parameters, minimumHealthyPercent and maximumPercent. You can change the deployment configuration of your service using UpdateService (p. 149).

- If minimumHealthyPercent is below 100%, the scheduler can ignore desiredCount temporarily during task replacement. For example, desiredCount is four tasks, a minimum of 50% allows the scheduler to stop two existing tasks before starting two new tasks. If the minimum is 100%, the service scheduler can’t remove existing tasks until the replacement tasks are considered healthy. Tasks for services that do not use a load balancer are considered healthy if they are in the RUNNING state. Tasks for services that use a load balancer are considered healthy if they are in the RUNNING state and the container instance they are hosted on is reported as healthy by the load balancer.
- The maximumPercent parameter represents an upper limit on the number of running tasks during task replacement, which enables you to define the replacement batch size. For example, if desiredCount of four tasks, a maximum of 200% starts four new tasks before stopping the four tasks to be drained (provided that the cluster resources required to do this are available). If the maximum is 100%, then replacement tasks can’t start until the draining tasks have stopped.

Any PENDING or RUNNING tasks that do not belong to a service are not affected; you must wait for them to finish or stop them manually.

A container instance has completed draining when it has no more RUNNING tasks. You can verify this using ListTasks (p. 88).

When you set a container instance to ACTIVE, the Amazon ECS scheduler can begin scheduling tasks on the instance again.

Request Syntax

```json
{
  "cluster": "string",
  "containerInstances": [ "string" ],
  "status": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.
cluster (p. 142)  
The short name or full Amazon Resource Name (ARN) of the cluster that hosts the container instance to update. If you do not specify a cluster, the default cluster is assumed.

Type: String
Required: No

containerInstances (p. 142)  
A list of container instance IDs or full ARN entries.

Type: Array of strings
Required: Yes

status (p. 142)  
The container instance state with which to update the container instance.

Type: String
Valid Values: ACTIVE | DRAINING
Required: Yes

Response Syntax

```json
{
   "containerInstances": [
      {
         "agentConnected": boolean,
         "agentUpdateStatus": "string",
         "attachments": [
            {
               "details": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "id": "string",
               "status": "string",
               "type": "string"
            }
         ],
         "attributes": [
            {
               "name": "string",
               "targetId": "string",
               "targetType": "string",
               "value": "string"
            }
         ],
         "containerInstanceArn": "string",
         "ec2InstanceId": "string",
         "pendingTasksCount": number,
         "registeredAt": number,
         "registeredResources": [
            {
               "doubleValue": number,
               "integerValue": number,
            }
         ]
      }
   ]
}
```

API Version 2014-11-13
Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

containerInstances (p. 143)

The list of container instances.

Type: Array of ContainerInstance (p. 174) objects

failures (p. 143)

Any failures associated with the call.

Type: Array of Failure (p. 184) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn’t have permissions to use the action or resource, or specifying an identifier that is not valid.
HTTP Status Code: 400

**ClusterNotFoundException**

The specified cluster could not be found. You can view your available clusters with 

HTTP Status Code: 400

**InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

**ServerException**

These errors are usually caused by a server issue.

HTTP Status Code: 500

---

**Example**

In the following example or examples, the Authorization header contents (**AUTHPARAMS**) must be replaced with an AWS Signature Version 4 signature. For more information, see [Signature Version 4 Signing Process](https://docs.aws.amazon.com/general/latest/gr/signature_version_4.html) in the **AWS General Reference**.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the [AWS Command Line Interface (AWS CLI)](https://docs.aws.amazon.com/cli/latest/index.html) or one of the [AWS SDKs](https://docs.aws.amazon.com/sdk-for-python/latest/), these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

---

**Example**

This example sets a container instance in the **default** cluster with the ID **1c3be8ed-df30-47b4-8f1e-6e68ebd01f34** to the **DRAINING** status so that it cannot receive tasks for placement.

**Sample Request**

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 114
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateContainerInstancesState
X-Amz-Date: 20161220T221142Z
User-Agent: aws-cli/1.11.31 Python/2.7.12 Darwin/16.3.0 botocore/1.4.88
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "status": "DRAINING",
  "cluster": "default",
  "containerInstances": [
    "1c3be8ed-df30-47b4-8f1e-6e68ebd01f34"
  ]
}
```

**Sample Response**

```
HTTP/1.1 200 OK
```
{
  "containerInstances": [
    {
      "agentConnected": true,
      "attributes": [
        {
          "name": "ecs.availability-zone",
          "value": "us-west-2b"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
        },
        {
          "name": "ecs.instance-type",
          "value": "c4.xlarge"
        },
        {
          "name": "ecs.ami-id",
          "value": "ami-a2ca61c2"
        },
        {
          "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
        },
        {
          "name": "com.amazonaws.ecs.capability.privileged-container"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
        },
        {
          "name": "com.amazonaws.ecs.capability.ecr-auth"
        },
        {
          "name": "ecs.os-type",
          "value": "linux"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
        },
        {
          "name": "com.amazonaws.ecs.capability.task-iam-role"
        }
    }
  ]
}
{
  "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
},
"containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-instance/1c3be8ed-df30-47b4-8f1e-6e88ebd01f34",
"ec2InstanceId": "i-05d99c76955727ec6",
"pendingTasksCount": 0,
"registeredResources": [
  {
    "doubleValue": 0,
    "integerValue": 4096,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 7482,
    "longValue": 0,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS",
    "stringSetValue": [
      "22",
      "2376",
      "2375",
      "51678",
      "51679"
    ],
    "type": "STRINGSET"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
],
"remainingResources": [
  {
    "doubleValue": 0,
    "integerValue": 4096,
    "longValue": 0,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 7482,
    "longValue": 0,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS",
    "stringSetValue": [],
    "type": "STRINGSET"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "longValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
]
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
UpdateService

Modifies the desired count, deployment configuration, network configuration, or task definition used in a service.

You can add to or subtract from the number of instantiations of a task definition in a service by specifying the cluster that the service is running in and a new `desiredCount` parameter.

You can use UpdateService (p. 149) to modify your task definition and deploy a new version of your service.

You can also update the deployment configuration of a service. When a deployment is triggered by updating the task definition of a service, the service scheduler uses the deployment configuration parameters, `minimumHealthyPercent` and `maximumPercent`, to determine the deployment strategy.

- If `minimumHealthyPercent` is below 100%, the scheduler can ignore `desiredCount` temporarily during a deployment. For example, if `desiredCount` is four tasks, a minimum of 50% allows the scheduler to stop two existing tasks before starting two new tasks. Tasks for services that do not use a load balancer are considered healthy if they are in the `RUNNING` state. Tasks for services that use a load balancer are considered healthy if they are in the `RUNNING` state and the container instance they are hosted on is reported as healthy by the load balancer.

- The `maximumPercent` parameter represents an upper limit on the number of running tasks during a deployment, which enables you to define the deployment batch size. For example, if `desiredCount` is four tasks, a maximum of 200% starts four new tasks before stopping the four older tasks (provided that the cluster resources required to do this are available).

When UpdateService (p. 149) stops a task during a deployment, the equivalent of `docker stop` is issued to the containers running in the task. This results in a `SIGTERM` and a 30-second timeout, after which `SIGKILL` is sent and the containers are forcibly stopped. If the container handles the `SIGTERM` gracefully and exits within 30 seconds from receiving it, no `SIGKILL` is sent.

When the service scheduler launches new tasks, it determines task placement in your cluster with the following logic:

- Determine which of the container instances in your cluster can support your service's task definition (for example, they have the required CPU, memory, ports, and container instance attributes).
- By default, the service scheduler attempts to balance tasks across Availability Zones in this manner (although you can choose a different placement strategy):
  - Sort the valid container instances by the fewest number of running tasks for this service in the same Availability Zone as the instance. For example, if zone A has one running service task and zones B and C each have zero, valid container instances in either zone B or C are considered optimal for placement.
  - Place the new service task on a valid container instance in an optimal Availability Zone (based on the previous steps), favoring container instances with the fewest number of running tasks for this service.

When the service scheduler stops running tasks, it attempts to maintain balance across the Availability Zones in your cluster using the following logic:

- Sort the container instances by the largest number of running tasks for this service in the same Availability Zone as the instance. For example, if zone A has one running service task and zones B and C each have two, container instances in either zone B or C are considered optimal for termination.
- Stop the task on a container instance in an optimal Availability Zone (based on the previous steps), favoring container instances with the largest number of running tasks for this service.
Request Syntax

```
{
  "cluster": "string",
  "deploymentConfiguration": {
    "maximumPercent": number,
    "minimumHealthyPercent": number
  },
  "desiredCount": number,
  "forceNewDeployment": boolean,
  "networkConfiguration": {
    "awsVpcConfiguration": {
      "assignPublicIp": "string",
      "securityGroups": [ "string" ],
      "subnets": [ "string" ]
    }
  },
  "platformVersion": "string",
  "service": "string",
  "taskDefinition": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 221).

The request accepts the following data in JSON format.

**cluster (p. 150)**

The short name or full Amazon Resource Name (ARN) of the cluster that your service is running on. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

**deploymentConfiguration (p. 150)**

Optional deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.

Type: DeploymentConfiguration (p. 182) object

Required: No

**desiredCount (p. 150)**

The number of instantiations of the task to place and keep running in your service.

Type: Integer

Required: No

**forceNewDeployment (p. 150)**

Whether or not to force a new deployment of the service.

Type: Boolean

Required: No
networkConfiguration (p. 150)

The network configuration for the service. This parameter is required for task definitions that use the awsvpc network mode to receive their own Elastic Network Interface, and it is not supported for other network modes. For more information, see Task Networking in the Amazon Elastic Container Service Developer Guide.

Note
Updating a service to add a subnet to a list of existing subnets does not trigger a service deployment. For example, if your network configuration change is to keep the existing subnets and simply add another subnet to the network configuration, this does not trigger a new service deployment.

Type: NetworkConfiguration (p. 195) object

Required: No

platformVersion (p. 150)

The platform version you want to update your service to run.

Type: String

Required: No

service (p. 150)

The name of the service to update.

Type: String

Required: Yes

taskDefinition (p. 150)

The family and revision (family:revision) or full ARN of the task definition to run in your service. If a revision is not specified, the latest ACTIVE revision is used. If you modify the task definition with UpdateService, Amazon ECS spawns a task with the new version of the task definition and then stops an old task after the new version is running.

Type: String

Required: No

**Response Syntax**

```json
{
"service": {
  "clusterArn": "string",
  "createdAt": number,
  "deploymentConfiguration": {
    "maximumPercent": number,
    "minimumHealthyPercent": number
  },
  "deployments": [
  {
    "createdAt": number,
    "desiredCount": number,
    "id": "string",
    "launchType": "string",
    "networkConfiguration": {
```
Response Elements

If the action is successful, the service sends back an HTTP 200 response.
The following data is returned in JSON format by the service.

service (p. 151)

The full description of your service following the update call.

Type: Service (p. 203) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 223).

AccessDeniedException

You do not have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action, such as using an action or resource on behalf of a user that doesn't have permissions to use the action or resource, or specifying an identifier that is not valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 68). Amazon ECS clusters are region-specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version does not satisfy the task definition's required capabilities.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version does not exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service is not active. You can't update a service that is inactive. If you have previously deleted a service, you can re-create it with CreateService (p. 6).

HTTP Status Code: 400
ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 75). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Example

In the following example or examples, the Authorization header contents ([AUTHPARAMS]) must be replaced with an AWS Signature Version 4 signature. For more information, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to create them manually. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, these tools automatically sign the requests for you, with the access key that you specify when you configure the tools. When you use these tools, you don't have to sign requests yourself.

Example

This example request updates the hello_world service to a desired count of 3.

Sample Request

```plaintext
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 45
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateService
X-Amz-Date: 20150429T194543Z
Content-Type: application/x-amz-json-1.1
Authorization: [AUTHPARAMS]

{
  "service": "hello_world",
  "desiredCount": 3
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:45:43 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 13376
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deploymentConfiguration": {
      "maximumPercent": 200,
      "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "createdAt": 1430333711.033,
        "desiredCount": 3
      }
    ]
  }
```

API Version 2014-11-13
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
Data Types

The Amazon EC2 Container Service API contains several data types that various actions use. This section describes each data type in detail.

**Note**
The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- Attachment (p. 158)
- AttachmentStateChange (p. 159)
- Attribute (p. 160)
- AwsVpcConfiguration (p. 161)
- Cluster (p. 162)
- Container (p. 164)
- ContainerDefinition (p. 166)
- ContainerInstance (p. 174)
- ContainerOverride (p. 177)
- ContainerStateChange (p. 179)
- Deployment (p. 180)
- DeploymentConfiguration (p. 182)
- Device (p. 183)
- Failure (p. 184)
- HostEntry (p. 185)
- HostVolumeProperties (p. 186)
- KernelCapabilities (p. 187)
- KeyValuePair (p. 189)
- LinuxParameters (p. 190)
- LoadBalancer (p. 191)
- LogConfiguration (p. 192)
- MountPoint (p. 193)
- NetworkBinding (p. 194)
- NetworkConfiguration (p. 195)
- NetworkInterface (p. 196)
- PlacementConstraint (p. 197)
- PlacementStrategy (p. 198)
- PortMapping (p. 199)
- Resource (p. 201)
- Service (p. 203)
- ServiceEvent (p. 206)
- Task (p. 207)
- TaskDefinition (p. 211)
- TaskDefinitionPlacementConstraint (p. 215)
- TaskOverride (p. 216)
• Ulimit (p. 217)
• VersionInfo (p. 218)
• Volume (p. 219)
• VolumeFrom (p. 220)
Attachment

An object representing a container instance or task attachment.

Contents

details

Details of the attachment. For Elastic Network Interfaces, this includes the network interface ID, the MAC address, the subnet ID, and the private IPv4 address.

Type: Array of KeyValuePair (p. 189) objects

Required: No

id

The unique identifier for the attachment.

Type: String

Required: No

status

The status of the attachment. Valid values are PRECREATED, CREATED, ATTACHING, ATTACHED, DETACHING, DETACHED, and DELETED.

Type: String

Required: No

type

The type of the attachment, such as an ElasticNetworkInterface.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AttachmentStateChange

An object representing a change in state for a task attachment.

Contents

attachmentArn

The Amazon Resource Name (ARN) of the attachment.

Type: String

Required: Yes

status

The status of the attachment.

Type: String

Required: Yes

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
**Attribute**

An attribute is a name-value pair associated with an Amazon ECS object. Attributes enable you to extend the Amazon ECS data model by adding custom metadata to your resources. For more information, see **Attributes** in the *Amazon Elastic Container Service Developer Guide*.

**Contents**

**name**

The name of the attribute. Up to 128 letters (uppercase and lowercase), numbers, hyphens, underscores, and periods are allowed.

Type: String

Required: Yes

**targetId**

The ID of the target. You can specify the short form ID for a resource or the full Amazon Resource Name (ARN).

Type: String

Required: No

**targetType**

The type of the target with which to attach the attribute. This parameter is required if you use the short form ID for a resource instead of the full ARN.

Type: String

Valid Values: container-instance

Required: No

**value**

The value of the attribute. Up to 128 letters (uppercase and lowercase), numbers, hyphens, underscores, periods, at signs (@), forward slashes, colons, and spaces are allowed.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
AwsVpcConfiguration

An object representing the networking details for a task or service.

Contents

assignPublicIp

Specifies whether or not the task's elastic network interface receives a public IP address.

Type: String

Valid Values: ENABLED | DISABLED

Required: No

securityGroups

The security groups associated with the task or service. If you do not specify a security group, the default security group for the VPC is used.

Type: Array of strings

Required: No

subnets

The subnets associated with the task or service.

Type: Array of strings

Required: Yes

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Cluster

A regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service, but you may also create other clusters. Clusters may contain more than one instance type simultaneously.

Contents

activeServicesCount

The number of services that are running on the cluster in an ACTIVE state. You can view these services with ListServices (p. 75).

Type: Integer
Required: No

clusterArn

The Amazon Resource Name (ARN) that identifies the cluster. The ARN contains the arn:aws:ecs namespace, followed by the region of the cluster, the AWS account ID of the cluster owner, the cluster namespace, and then the cluster name. For example, arn:aws:ecs:region:012345678910:cluster/test..

Type: String
Required: No

clusterName

A user-generated string that you use to identify your cluster.

Type: String
Required: No

pendingTasksCount

The number of tasks in the cluster that are in the PENDING state.

Type: Integer
Required: No

registeredContainerInstancesCount

The number of container instances registered into the cluster.

Type: Integer
Required: No

runningTasksCount

The number of tasks in the cluster that are in the RUNNING state.

Type: Integer
Required: No

statistics

Additional information about your clusters that are separated by launch type, including:
- runningEC2TasksCount
- RunningFargateTasksCount
- pendingEC2TasksCount
- pendingFargateTasksCount
- activeEC2ServiceCount
- activeFargateServiceCount
- drainingEC2ServiceCount
- drainingFargateServiceCount

Type: Array of KeyValuePair (p. 189) objects

Required: No

status

The status of the cluster. The valid values are ACTIVE or INACTIVE. ACTIVE indicates that you can register container instances with the cluster and the associated instances can accept tasks.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Container

A Docker container that is part of a task.

Contents

countainerArn

The Amazon Resource Name (ARN) of the container.

Type: String

Required: No

exitCode

The exit code returned from the container.

Type: Integer

Required: No

lastStatus

The last known status of the container.

Type: String

Required: No

name

The name of the container.

Type: String

Required: No

networkBindings

The network bindings associated with the container.

Type: Array of NetworkBinding (p. 194) objects

Required: No

networkInterfaces

The network interfaces associated with the container.

Type: Array of NetworkInterface (p. 196) objects

Required: No

reason

A short (255 max characters) human-readable string to provide additional details about a running or stopped container.

Type: String

Required: No
taskArn

The ARN of the task.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ContainerDefinition

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

Contents

command

The command that is passed to the container. This parameter maps to \texttt{Cmd} in the \texttt{Create a container} section of the Docker Remote API and the \texttt{COMMAND} parameter to \texttt{docker run}. For more information, see \url{https://docs.docker.com/engine/reference/builder/#cmd}.

Type: Array of strings

Required: No

cpu

The number of \texttt{cpu} units reserved for the container. This parameter maps to \texttt{CpuShares} in the \texttt{Create a container} section of the Docker Remote API and the \texttt{--cpu-shares} option to \texttt{docker run}.

This field is optional for tasks using the Fargate launch type, and the only requirement is that the total amount of CPU reserved for all containers within a task be lower than the task-level \texttt{cpu} value.

\textbf{Note}

You can determine the number of CPU units that are available per EC2 instance type by multiplying the vCPUs listed for that instance type on the Amazon EC2 Instances detail page by 1,024.

For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

Linux containers share unallocated CPU units with other containers on the container instance with the same ratio as their allocated amount. For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that is the only task running on the container instance, that container could use the full 1,024 CPU unit share at any given time. However, if you launched another copy of the same task on that container instance, each task would be guaranteed a minimum of 512 CPU units when needed, and each container could float to higher CPU usage if the other container was not using it, but if both tasks were 100% active all of the time, they would be limited to 512 CPU units.

On Linux container instances, the Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see \texttt{CPU share constraint} in the Docker documentation. The minimum valid CPU share value that the Linux kernel will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- \textbf{Agent versions less than or equal to 1.1.0}: Null and zero CPU values are passed to Docker as 0, which Docker then converts to 1,024 CPU shares. CPU values of 1 are passed to Docker as 1, which the Linux kernel converts to 2 CPU shares.

- \textbf{Agent versions greater than or equal to 1.2.0}: Null, zero, and CPU values of 1 are passed to Docker as 2.
On Windows container instances, the CPU limit is enforced as an absolute limit, or a quota. Windows containers only have access to the specified amount of CPU that is described in the task definition.

Type: Integer
Required: No

disableNetworking

When this parameter is true, networking is disabled within the container. This parameter maps to NetworkDisabled in the Create a container section of the Docker Remote API.

Note
This parameter is not supported for Windows containers.

Type: Boolean
Required: No

dnsSearchDomains

A list of DNS search domains that are presented to the container. This parameter maps to DnsSearch in the Create a container section of the Docker Remote API and the --dns-search option to docker run.

Note
This parameter is not supported for Windows containers.

Type: Array of strings
Required: No

dnsServers

A list of DNS servers that are presented to the container. This parameter maps to Dns in the Create a container section of the Docker Remote API and the --dns option to docker run.

Note
This parameter is not supported for Windows containers.

Type: Array of strings
Required: No

dockerLabels

A key/value map of labels to add to the container. This parameter maps to Labels in the Create a container section of the Docker Remote API and the --label option to docker run. This parameter requires version 1.18 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: sudo docker version | grep "Server API version"

Type: String to string map
Required: No

dockerSecurityOptions

A list of strings to provide custom labels for SELinux and AppArmor multi-level security systems. This field is not valid for containers in tasks using the Fargate launch type.

This parameter maps to SecurityOpt in the Create a container section of the Docker Remote API and the --security-opt option to docker run.
Note
The Amazon ECS container agent running on a container instance must register with
the ECS_SELINUX_CAPABLE=true or ECS_APPARMOR_CAPABLE=true environment
variables before containers placed on that instance can use these security options. For
more information, see Amazon ECS Container Agent Configuration in the Amazon Elastic
Container Service Developer Guide.

This parameter is not supported for Windows containers.

Type: Array of strings

Required: No

entryPoint

Important
Early versions of the Amazon ECS container agent do not properly handle entryPoint
parameters. If you have problems using entryPoint, update your container agent or enter
your commands and arguments as command array items instead.

The entry point that is passed to the container. This parameter maps to Entrypoint in the Create a
container section of the Docker Remote API and the --entrypoint option to docker run. For more
information, see https://docs.docker.com/engine/reference/builder/#entrypoint.

Type: Array of strings

Required: No

environment

The environment variables to pass to a container. This parameter maps to Env in the Create a
container section of the Docker Remote API and the --env option to docker run.

Important
We do not recommend using plaintext environment variables for sensitive information, such
as credential data.

Type: Array of KeyValuePair (p. 189) objects

Required: No

essential

If the essential parameter of a container is marked as true, and that container fails or stops for
any reason, all other containers that are part of the task are stopped. If the essential parameter
of a container is marked as false, then its failure does not affect the rest of the containers in a task.
If this parameter is omitted, a container is assumed to be essential.

All tasks must have at least one essential container. If you have an application that is composed
of multiple containers, you should group containers that are used for a common purpose into
components, and separate the different components into multiple task definitions. For more
information, see Application Architecture in the Amazon Elastic Container Service Developer Guide.

Type: Boolean

Required: No

extraHosts

A list of hostnames and IP address mappings to append to the /etc/hosts file on the container. If
using the Fargate launch type, this may be used to list non-Fargate hosts you want the container to
talk to. This parameter maps to ExtraHosts in the Create a container section of the Docker Remote
API and the --add-host option to docker run.

Note
This parameter is not supported for Windows containers.
Type: Array of HostEntry (p. 185) objects

Required: No

hostname

The hostname to use for your container. This parameter maps to Hostname in the Create a container section of the Docker Remote API and the --hostname option to docker run.

Type: String

Required: No

image

The image used to start a container. This string is passed directly to the Docker daemon. Images in the Docker Hub registry are available by default. Other repositories are specified with either repository-url/image:tag or repository-url/image@digest. Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed. This parameter maps to Image in the Create a container section of the Docker Remote API and the IMAGE parameter of docker run.

- Images in Amazon ECR repositories can be specified by either using the full registry/repository:tag or registry/repository@digest. For example, 012345678910.dkr.ecr.<region-name>.amazonaws.com/<repository-name>:latest or 012345678910.dkr.ecr.<region-name>.amazonaws.com/<repository-name>@sha256:94afd1f2e64d908bc90dbca0035a5b567EXAMPLE.

- Images in official repositories on Docker Hub use a single name (for example, ubuntu or mongo).

- Images in other repositories on Docker Hub are qualified with an organization name (for example, amazon/amazon-ecs-agent).

- Images in other online repositories are qualified further by a domain name (for example, quay.io/assemblyline/ubuntu).

Type: String

Required: No

links

The link parameter allows containers to communicate with each other without the need for port mappings. Only supported if the network mode of a task definition is set to bridge. The name:internalName construct is analogous to name:alias in Docker links. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. For more information about linking Docker containers, go to https://docs.docker.com/engine/userguide/networking/default_network/dockerlinks/. This parameter maps to Links in the Create a container section of the Docker Remote API and the --link option to docker run.

Note

This parameter is not supported for Windows containers.

Important

Containers that are collocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

Type: Array of strings

Required: No

linuxParameters

Linux-specific modifications that are applied to the container, such as Linux KernelCapabilities (p. 187).
Note
This parameter is not supported for Windows containers or tasks using the Fargate launch type.

Type: LinuxParameters (p. 190) object

Required: No

logConfiguration

The log configuration specification for the container.

If using the Fargate launch type, the only supported value is awslogs.

This parameter maps to LogConfig in the Create a container section of the Docker Remote API and
the --log-driver option to docker run. By default, containers use the same logging driver that
the Docker daemon uses; however the container may use a different logging driver than the Docker
daemon by specifying a log driver with this parameter in the container definition. To use a different
logging driver for a container, the log system must be configured properly on the container instance
(or on a different log server for remote logging options). For more information on the options for
different supported log drivers, see Configure logging drivers in the Docker documentation.

Note
Amazon ECS currently supports a subset of the logging drivers available to the Docker
daemon (shown in the LogConfiguration (p. 192) data type). Additional log drivers may be
available in future releases of the Amazon ECS container agent.

This parameter requires version 1.18 of the Docker Remote API or greater on your container
instance. To check the Docker Remote API version on your container instance, log in to your
container instance and run the following command: sudo docker version | grep "Server
API version"

Note
The Amazon ECS container agent running on a container instance must register the
logging drivers available on that instance with the ECS_AVAILABLE_LOGGING_DRIVERS
environment variable before containers placed on that instance can use these log
configuration options. For more information, see Amazon ECS Container Agent
Configuration in the Amazon Elastic Container Service Developer Guide.

Type: LogConfiguration (p. 192) object

Required: No

memory

The hard limit (in MiB) of memory to present to the container. If your container attempts to exceed
the memory specified here, the container is killed. This parameter maps to Memory in the Create a
container section of the Docker Remote API and the --memory option to docker run.

If your containers will be part of a task using the Fargate launch type, this field is optional and the
only requirement is that the total amount of memory reserved for all containers within a task be
lower than the task memory value.

For containers that will be part of a task using the EC2 launch type, you must specify a non-zero
integer for one or both of memory or memoryReservation in container definitions. If you specify
both, memory must be greater than memoryReservation. If you specify memoryReservation,
then that value is subtracted from the available memory resources for the container instance on
which the container is placed; otherwise, the value of memory is used.

The Docker daemon reserves a minimum of 4 MiB of memory for a container, so you should not
specify fewer than 4 MiB of memory for your containers.
memoryReservation

The soft limit (in MiB) of memory to reserve for the container. When system memory is under heavy contention, Docker attempts to keep the container memory to this soft limit; however, your container can consume more memory when it needs to, up to either the hard limit specified with the memory parameter (if applicable), or all of the available memory on the container instance, whichever comes first. This parameter maps to MemoryReservation in the Create a container section of the Docker Remote API and the --memory-reservation option to docker run.

You must specify a non-zero integer for one or both of memory or memoryReservation in container definitions. If you specify both, memory must be greater than memoryReservation. If you specify memoryReservation, then that value is subtracted from the available memory resources for the container instance on which the container is placed; otherwise, the value of memory is used.

For example, if your container normally uses 128 MiB of memory, but occasionally bursts to 256 MiB of memory for short periods of time, you can set a memoryReservation of 128 MiB, and a memory hard limit of 300 MiB. This configuration would allow the container to only reserve 128 MiB of memory from the remaining resources on the container instance, but also allow the container to consume more memory resources when needed.

The Docker daemon reserves a minimum of 4 MiB of memory for a container, so you should not specify fewer than 4 MiB of memory for your containers.

mountPoints

The mount points for data volumes in your container.

This parameter maps to Volumes in the Create a container section of the Docker Remote API and the --volume option to docker run.

Windows containers can mount whole directories on the same drive as $env:ProgramData. Windows containers cannot mount directories on a different drive, and mount point cannot be across drives.

Type: Array of MountPoint (p. 193) objects

name

The name of a container. If you are linking multiple containers together in a task definition, the name of one container can be entered in the links of another container to connect the containers. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. This parameter maps to name in the Create a container section of the Docker Remote API and the --name option to docker run.

Type: String

portMappings

The list of port mappings for the container. Port mappings allow containers to access ports on the host container instance to send or receive traffic.
For task definitions that use the `awsvpc` network mode, you should only specify the `containerPort`. The `hostPort` can be left blank or it must be the same value as the `containerPort`.

Port mappings on Windows use the `NetNAT` gateway address rather than `localhost`. There is no loopback for port mappings on Windows, so you cannot access a container’s mapped port from the host itself.

This parameter maps to `PortBindings` in the Create a container section of the Docker Remote API and the `--publish` option to `docker run`. If the network mode of a task definition is set to `none`, then you can’t specify port mappings. If the network mode of a task definition is set to `host`, then host ports must either be undefined or they must match the container port in the port mapping.

**Note**
After a task reaches the `RUNNING` status, manual and automatic host and container port assignments are visible in the Network Bindings section of a container description for a selected task in the Amazon ECS console, or the `networkBindings` section of `DescribeTasks` responses.

Type: Array of `PortMapping` objects

Required: No

`privileged`

When this parameter is true, the container is given elevated privileges on the host container instance (similar to the `root` user). This parameter maps to `Privileged` in the Create a container section of the Docker Remote API and the `--privileged` option to `docker run`.

**Note**
This parameter is not supported for Windows containers or tasks using the Fargate launch type.

Type: Boolean

Required: No

`readonlyRootFilesystem`

When this parameter is true, the container is given read-only access to its root file system. This parameter maps to `ReadonlyRootfs` in the Create a container section of the Docker Remote API and the `--read-only` option to `docker run`.

**Note**
This parameter is not supported for Windows containers.

Type: Boolean

Required: No

`ulimits`

A list of `ulimits` to set in the container. This parameter maps to `Ulimits` in the Create a container section of the Docker Remote API and the `--ulimit` option to `docker run`. Valid naming values are displayed in the `Ulimit` data type. This parameter requires version 1.18 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version | grep "Server API version"`

**Note**
This parameter is not supported for Windows containers.

Type: Array of `Ulimit` objects
Required: No

**user**

The user name to use inside the container. This parameter maps to `User` in the Create a container section of the Docker Remote API and the `--user` option to `docker run`.

*Note*

This parameter is not supported for Windows containers.

Type: String

Required: No

**volumesFrom**

Data volumes to mount from another container. This parameter maps to `VolumesFrom` in the Create a container section of the Docker Remote API and the `--volumes-from` option to `docker run`.

Type: Array of `VolumeFrom` (p. 220) objects

Required: No

**workingDirectory**

The working directory in which to run commands inside the container. This parameter maps to `WorkingDir` in the Create a container section of the Docker Remote API and the `--workdir` option to `docker run`.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ContainerInstance

An EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

Contents

agentConnected

This parameter returns true if the agent is connected to Amazon ECS. Registered instances with an agent that may be unhealthy or stopped return false. Instances without a connected agent can't accept placement requests.

Type: Boolean
Required: No

agentUpdateStatus

The status of the most recent agent update. If an update has never been requested, this value is NULL.

Type: String
Valid Values: PENDING | STAGING | STAGED | UPDATING | UPDATED | FAILED
Required: No

attachments

The Elastic Network Interfaces associated with the container instance.

Type: Array of Attachment (p. 158) objects
Required: No

attributes

The attributes set for the container instance, either by the Amazon ECS container agent at instance registration or manually with the PutAttributes (p. 93) operation.

Type: Array of Attribute (p. 160) objects
Required: No

containerInstanceArn

The Amazon Resource Name (ARN) of the container instance. The ARN contains the arn:aws:ecs namespace, followed by the region of the container instance, the AWS account ID of the container instance owner, the container-instance namespace, and then the container instance ID. For example, arn:aws:ecs:region:aws_account_id:container-instance/container_instance_ID.

Type: String
Required: No

c2InstanceId

The EC2 instance ID of the container instance.

Type: String
Required: No
pendingTasksCount

The number of tasks on the container instance that are in the PENDING status.

Type: Integer
Required: No

registeredAt

The Unix time stamp for when the container instance was registered.

Type: Timestamp
Required: No

registeredResources

For most resource types, this parameter describes the registered resources on the container instance that are in use by current tasks. For port resource types, this parameter describes the ports that were reserved by the Amazon ECS container agent when it registered the container instance with Amazon ECS.

Type: Array of Resource (p. 201) objects
Required: No

remainingResources

For most resource types, this parameter describes the remaining resources of the container instance that are available for new tasks. For port resource types, this parameter describes the ports that are reserved by the Amazon ECS container agent and any containers that have reserved port mappings; any port that is not specified here is available for new tasks.

Type: Array of Resource (p. 201) objects
Required: No

runningTasksCount

The number of tasks on the container instance that are in the RUNNING status.

Type: Integer
Required: No

status

The status of the container instance. The valid values are ACTIVE, INACTIVE, or DRAINING. ACTIVE indicates that the container instance can accept tasks. DRAINING indicates that new tasks are not placed on the container instance and any service tasks running on the container instance are removed if possible. For more information, see Container Instance Draining in the Amazon Elastic Container Service Developer Guide.

Type: String
Required: No

version

The version counter for the container instance. Every time a container instance experiences a change that triggers a CloudWatch event, the version counter is incremented. If you are replicating your Amazon ECS container instance state with CloudWatch Events, you can compare the version of a container instance reported by the Amazon ECS APIs with the version reported in CloudWatch.
Events for the container instance (inside the detail object) to verify that the version in your event stream is current.

Type: Long
Required: No

**versionInfo**

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

Type: VersionInfo (p. 218) 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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ContainerOverride

The overrides that should be sent to a container.

Contents

command

The command to send to the container that overrides the default command from the Docker image or the task definition. You must also specify a container name.

Type: Array of strings

Required: No

cpu

The number of cpu units reserved for the container, instead of the default value from the task definition. You must also specify a container name.

Type: Integer

Required: No

environment

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition. You must also specify a container name.

Type: Array of KeyValuePair (p. 189) objects

Required: No

memory

The hard limit (in MiB) of memory to present to the container, instead of the default value from the task definition. If your container attempts to exceed the memory specified here, the container is killed. You must also specify a container name.

Type: Integer

Required: No

memoryReservation

The soft limit (in MiB) of memory to reserve for the container, instead of the default value from the task definition. You must also specify a container name.

Type: Integer

Required: No

name

The name of the container that receives the override. This parameter is required if any override is specified.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ContainerStateChange

An object representing a change in state for a container.

Contents

containerName

The name of the container.

Type: String

Required: No

exitCode

The exit code for the container, if the state change is a result of the container exiting.

Type: Integer

Required: No

networkBindings

Any network bindings associated with the container.

Type: Array of NetworkBinding (p. 194) objects

Required: No

reason

The reason for the state change.

Type: String

Required: No

status

The status of the container.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Deployment

The details of an Amazon ECS service deployment.

Contents

createdAt

The Unix time stamp for when the service was created.

Type: Timestamp

Required: No

desiredCount

The most recent desired count of tasks that was specified for the service to deploy or maintain.

Type: Integer

Required: No

id

The ID of the deployment.

Type: String

Required: No

launchType

The launch type on which your service is running.

Type: String

Valid Values: EC2 | FARGATE

Required: No

networkConfiguration

The VPC subnet and security group configuration for tasks that receive their own Elastic Network Interface by using the awsvpc networking mode.

Type: NetworkConfiguration (p. 195) object

Required: No

pendingCount

The number of tasks in the deployment that are in the PENDING status.

Type: Integer

Required: No

platformVersion

The platform version on which your service is running.

Type: String

Required: No
runningCount

The number of tasks in the deployment that are in the RUNNING status.

Type: Integer

Required: No

status

The status of the deployment. Valid values are PRIMARY (for the most recent deployment), ACTIVE (for previous deployments that still have tasks running, but are being replaced with the PRIMARY deployment), and INACTIVE (for deployments that have been completely replaced).

Type: String

Required: No

taskDefinition

The most recent task definition that was specified for the service to use.

Type: String

Required: No

updatedAt

The Unix time stamp for when the service was last updated.

Type: Timestamp

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
DeploymentConfiguration

Optional deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.

Contents

maximumPercent

The upper limit (as a percentage of the service's desiredCount) of the number of tasks that are allowed in the RUNNING or PENDING state in a service during a deployment. The maximum number of tasks during a deployment is the desiredCount multiplied by maximumPercent/100, rounded down to the nearest integer value.

Type: Integer
Required: No

minimumHealthyPercent

The lower limit (as a percentage of the service's desiredCount) of the number of running tasks that must remain in the RUNNING state in a service during a deployment. The minimum number of healthy tasks during a deployment is the desiredCount multiplied by minimumHealthyPercent/100, rounded up to the nearest integer value.

Type: Integer
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Device

An object representing a container instance host device.

Contents

containerPath

The path inside the container at which to expose the host device.

Type: String
Required: No

hostPath

The path for the device on the host container instance.

Type: String
Required: Yes

permissions

The explicit permissions to provide to the container for the device. By default, the container has permissions for read, write, and mknod for the device.

Type: Array of strings
Valid Values: read | write | mknod
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Failure

A failed resource.

Contents

arn

The Amazon Resource Name (ARN) of the failed resource.

Type: String

Required: No

reason

The reason for the failure.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
HostEntry

Hostnames and IP address entries that are added to the `/etc/hosts` file of a container via the `extraHosts` parameter of its `ContainerDefinition` (p. 166).

Contents

hostname

The hostname to use in the `/etc/hosts` entry.

Type: String

Required: Yes

ipAddress

The IP address to use in the `/etc/hosts` entry.

Type: String

Required: Yes

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
HostVolumeProperties

Details on a container instance host volume.

Contents

sourcePath

The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you. If the host parameter contains a sourcePath file location, then the data volume persists at the specified location on the host container instance until you delete it manually. If the sourcePath value does not exist on the host container instance, the Docker daemon creates it. If the location does exist, the contents of the source path folder are exported.

If you are using the Fargate launch type, the sourcePath parameter is not supported.

Type: String
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
KernelCapabilities

The Linux capabilities for the container that are added to or dropped from the default configuration provided by Docker. For more information on the default capabilities and the non-default available capabilities, see Runtime privilege and Linux capabilities in the Docker run reference. For more detailed information on these Linux capabilities, see the capabilities(7) Linux manual page.

Contents

add

The Linux capabilities for the container that have been added to the default configuration provided by Docker. This parameter maps to CapAdd in the Create a container section of the Docker Remote API and the --cap-add option to docker run.

Valid values: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Type: Array of strings

Required: No

drop

The Linux capabilities for the container that have been removed from the default configuration provided by Docker. This parameter maps to CapDrop in the Create a container section of the Docker Remote API and the --cap-drop option to docker run.

Valid values: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Type: Array of strings

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Key::ValuePair

A key and value pair object.

**Contents**

**name**

The name of the key value pair. For environment variables, this is the name of the environment variable.

Type: String

Required: No

**value**

The value of the key value pair. For environment variables, this is the value of the environment variable.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LinuxParameters

Linux-specific options that are applied to the container, such as Linux KernelCapabilities (p. 187).

Contents

capabilities

The Linux capabilities for the container that are added to or dropped from the default configuration provided by Docker.

Type: KernelCapabilities (p. 187) object

Required: No
devices

Any host devices to expose to the container. This parameter maps to Devices in the Create a container section of the Docker Remote API and the --device option to docker run.

Type: Array of Device (p. 183) objects

Required: No
initProcessEnabled

Run an init process inside the container that forwards signals and reaps processes. This parameter maps to the --init option to docker run. This parameter requires version 1.25 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: sudo docker version | grep "Server API version"

Type: Boolean

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LoadBalancer

Details on a load balancer that is used with a service.

Contents

containerName

The name of the container (as it appears in a container definition) to associate with the load balancer.

Type: String

Required: No

containerPort

The port on the container to associate with the load balancer. This port must correspond to a containerPort in the service's task definition. Your container instances must allow ingress traffic on the hostPort of the port mapping.

Type: Integer

Required: No

loadBalancerName

The name of a load balancer.

Type: String

Required: No

targetGroupArn

The full Amazon Resource Name (ARN) of the Elastic Load Balancing target group associated with a service.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
LogConfiguration

Log configuration options to send to a custom log driver for the container.

Contents

logDriver

The log driver to use for the container. The valid values listed for this parameter are log drivers that
the Amazon ECS container agent can communicate with by default. If using the Fargate launch type,
the only supported value is awslogs. For more information about using the awslogs driver, see
Using the awslogs Log Driver in the Amazon Elastic Container Service Developer Guide.

Note
If you have a custom driver that is not listed above that you would like to work with the
Amazon ECS container agent, you can fork the Amazon ECS container agent project that is
available on GitHub and customize it to work with that driver. We encourage you to submit
pull requests for changes that you would like to have included. However, Amazon Web
Services does not currently support running modified copies of this software.

This parameter requires version 1.18 of the Docker Remote API or greater on your container
instance. To check the Docker Remote API version on your container instance, log in to your
container instance and run the following command: sudo docker version | grep "Server
API version"

Type: String

Valid Values: json-file | syslog | journald | gelf | fluentd | awslogs | splunk

Required: Yes

options

The configuration options to send to the log driver. This parameter requires version 1.19 of the
Docker Remote API or greater on your container instance. To check the Docker Remote API version
on your container instance, log in to your container instance and run the following command: sudo
docker version | grep "Server API version"

Type: String to string map

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
MountPoint

Details on a volume mount point that is used in a container definition.

Contents

containerPath

The path on the container to mount the host volume at.

Type: String
Required: No

readOnly

If this value is true, the container has read-only access to the volume. If this value is false, then the container can write to the volume. The default value is false.

Type: Boolean
Required: No

sourceVolume

The name of the volume to mount.

Type: String
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NetworkBinding

Details on the network bindings between a container and its host container instance. After a task reaches the RUNNING status, manual and automatic host and container port assignments are visible in the networkBindings section of DescribeTasks (p. 57) API responses.

Contents

bindIP

The IP address that the container is bound to on the container instance.

Type: String

Required: No

containerPort

The port number on the container that is used with the network binding.

Type: Integer

Required: No

hostPort

The port number on the host that is used with the network binding.

Type: Integer

Required: No

protocol

The protocol used for the network binding.

Type: String

Valid Values: tcp | udp

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NetworkConfiguration

An object representing the network configuration for a task or service.

Contents

awsVpcConfiguration

The VPC subnets and security groups associated with a task.

Type: AwsVpcConfiguration (p. 161) 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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
NetworkInterface

An object representing the Elastic Network Interface for tasks that use the awsvpc network mode.

Contents

attachmentId

The attachment ID for the network interface.

Type: String

Required: No

ipv6Address

The private IPv6 address for the network interface.

Type: String

Required: No

privateIpv4Address

The private IPv4 address for the network interface.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PlacementConstraint

An object representing a constraint on task placement. For more information, see Task Placement Constraints in the Amazon Elastic Container Service Developer Guide.

Contents

eexpression

A cluster query language expression to apply to the constraint. Note you cannot specify an expression if the constraint type is distinctInstance. For more information, see Cluster Query Language in the Amazon Elastic Container Service Developer Guide.

Type: String
Required: No

type

The type of constraint. Use distinctInstance to ensure that each task in a particular group is running on a different container instance. Use memberOf to restrict the selection to a group of valid candidates. The value distinctInstance is not supported in task definitions.

Type: String
Valid Values: distinctInstance | memberOf
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PlacementStrategy

The task placement strategy for a task or service. For more information, see Task Placement Strategies in the Amazon Elastic Container Service Developer Guide.

Contents

field

The field to apply the placement strategy against. For the spread placement strategy, valid values are `instanceId` (or `host`, which has the same effect), or any platform or custom attribute that is applied to a container instance, such as `attribute:ecs.availability-zone`. For the binpack placement strategy, valid values are `cpu` and `memory`. For the random placement strategy, this field is not used.

Type: String

Required: No

type

The type of placement strategy. The random placement strategy randomly places tasks on available candidates. The spread placement strategy spreads placement across available candidates evenly based on the field parameter. The binpack strategy places tasks on available candidates that have the least available amount of the resource that is specified with the field parameter. For example, if you binpack on memory, a task is placed on the instance with the least amount of remaining memory (but still enough to run the task).

Type: String

Valid Values: random | spread | binpack

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
PortMapping

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

If using containers in a task with the `awsvpc` or `host` network mode, exposed ports should be specified using `containerPort`. The `hostPort` can be left blank or it must be the same value as the `containerPort`.

After a task reaches the `RUNNING` status, manual and automatic host and container port assignments are visible in the `networkBindings` section of `DescribeTasks` API responses.

Contents

**containerPort**

The port number on the container that is bound to the user-specified or automatically assigned host port.

If using containers in a task with the `awsvpc` or `host` network mode, exposed ports should be specified using `containerPort`.

If using containers in a task with the `bridge` network mode and you specify a container port and not a host port, your container automatically receives a host port in the ephemeral port range (for more information, see `hostPort`). Port mappings that are automatically assigned in this way do not count toward the 100 reserved ports limit of a container instance.

Type: Integer

Required: No

**hostPort**

The port number on the container instance to reserve for your container.

If using containers in a task with the `awsvpc` or `host` network mode, the `hostPort` can either be left blank or needs to be the same value as the `containerPort`.

If using containers in a task with the `bridge` network mode, you can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to 0) while specifying a `containerPort` and your container automatically receives a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range for Docker version 1.6.0 and later is listed on the instance under `/proc/sys/net/ipv4/ip_local_port_range`; if this kernel parameter is unavailable, the default ephemeral port range from 49153 through 65535 is used. You should not attempt to specify a host port in the ephemeral port range as these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

**Note**  
The default ephemeral port range from 49153 through 65535 is always used for Docker versions before 1.6.0.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS container agent ports 51678 and 51679. Any host port that was previously specified in a running task is also reserved while the task is running (after a task stops, the host port is released). The current reserved ports are displayed in the `remainingResources` of `DescribeContainerInstances` output, and a container instance may have up to 100 reserved...
ports at a time, including the default reserved ports (automatically assigned ports do not count toward the 100 reserved ports limit).

`Type:` Integer

`Required:` No

**protocol**

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`.

`Type:` String

`Valid Values:` `tcp` | `udp`

`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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Resource

Describes the resources available for a container instance.

Contents

doubleValue

When the doubleValue type is set, the value of the resource must be a double precision floating-point type.

Type: Double
Required: No

integerValue

When the integerValue type is set, the value of the resource must be an integer.

Type: Integer
Required: No

longValue

When the longValue type is set, the value of the resource must be an extended precision floating-point type.

Type: Long
Required: No

name

The name of the resource, such as cpu, memory, ports, or a user-defined resource.

Type: String
Required: No

stringValue

When the stringValue type is set, the value of the resource must be a string type.

Type: Array of strings
Required: No

type

The type of the resource, such as INTEGER, DOUBLE, LONG, or STRINGSET.

Type: String
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++
See Also

- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Service
Details on a service within a cluster

Contents

clusterArn
The Amazon Resource Name (ARN) of the cluster that hosts the service.
Type: String
Required: No

createdAt
The Unix time stamp for when the service was created.
Type: Timestamp
Required: No

deploymentConfiguration
Optional deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.
Type: DeploymentConfiguration (p. 182) object
Required: No

deployments
The current state of deployments for the service.
Type: Array of Deployment (p. 180) objects
Required: No

desiredCount
The desired number of instantiations of the task definition to keep running on the service. This value is specified when the service is created with CreateService (p. 6), and it can be modified with UpdateService (p. 149).
Type: Integer
Required: No

events
The event stream for your service. A maximum of 100 of the latest events are displayed.
Type: Array of ServiceEvent (p. 206) objects
Required: No

launchType
The launch type on which your service is running.
Type: String
Valid Values: EC2 | FARGATE
Required: No

**loadBalancers**

A list of Elastic Load Balancing load balancer objects, containing the load balancer name, the container name (as it appears in a container definition), and the container port to access from the load balancer.

Type: Array of LoadBalancer (p. 191) objects

Required: No

**networkConfiguration**

The VPC subnet and security group configuration for tasks that receive their own Elastic Network Interface by using the awsvpc networking mode.

Type: NetworkConfiguration (p. 195) object

Required: No

**pendingCount**

The number of tasks in the cluster that are in the PENDING state.

Type: Integer

Required: No

**placementConstraints**

The placement constraints for the tasks in the service.

Type: Array of PlacementConstraint (p. 197) objects

Required: No

**placementStrategy**

The placement strategy that determines how tasks for the service are placed.

Type: Array of PlacementStrategy (p. 198) objects

Required: No

**platformVersion**

The platform version on which your task is running. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

**roleArn**

The ARN of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with an Elastic Load Balancing load balancer.

Type: String

Required: No

**runningCount**

The number of tasks in the cluster that are in the RUNNING state.

Type: Integer
serviceArn

The ARN that identifies the service. The ARN contains the arn:aws:ecs namespace, followed by the region of the service, the AWS account ID of the service owner, the service namespace, and then the service name. For example, arn:aws:ecs:region:012345678910:service/my-service.

Type: String
Required: No

serviceName

The name of your service. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. Service names must be unique within a cluster, but you can have similarly named services in multiple clusters within a region or across multiple regions.

Type: String
Required: No

status

The status of the service. The valid values are ACTIVE, DRAINING, or INACTIVE.

Type: String
Required: No

taskDefinition

The task definition to use for tasks in the service. This value is specified when the service is created with CreateService (p. 6), and it can be modified with UpdateService (p. 149).

Type: String
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
ServiceEvent

Details on an event associated with a service.

Contents

createdAt

The Unix time stamp for when the event was triggered.

Type: Timestamp

Required: No

id

The ID string of the event.

Type: String

Required: No

message

The event message.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Task

Details on a task in a cluster.

Contents

attachments

The Elastic Network Adapter associated with the task if the task uses the awsvpc network mode.

Type: Array of Attachment (p. 158) objects

Required: No

clusterArn

The ARN of the cluster that hosts the task.

Type: String

Required: No

connectivity

The connectivity status of a task.

Type: String

Valid Values: CONNECTED | DISCONNECTED

Required: No

connectivityAt

The Unix time stamp for when the task last went into CONNECTED status.

Type: Timestamp

Required: No

containerInstanceArn

The ARN of the container instances that host the task.

Type: String

Required: No

containers

The containers associated with the task.

Type: Array of Container (p. 164) objects

Required: No

cpu

The number of cpu units used by the task. If using the EC2 launch type, this field is optional and any value can be used. If using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the memory parameter:

- 256 (.25 vCPU) - Available memory values: 0.5GB, 1GB, 2GB
- 512 (.5 vCPU) - Available memory values: 1GB, 2GB, 3GB, 4GB
• 1024 (1 vCPU) - Available memory values: 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB
• 2048 (2 vCPU) - Available memory values: Between 4GB and 16GB in 1GB increments
• 4096 (4 vCPU) - Available memory values: Between 8GB and 30GB in 1GB increments

Type: String
Required: No

**createdAt**

The Unix time stamp for when the task was created (the task entered the **PENDING** state).

Type: Timestamp
Required: No

**desiredStatus**

The desired status of the task.

Type: String
Required: No

**executionStoppedAt**

The Unix timestamp for when the task execution stopped.

Type: Timestamp
Required: No

**group**

The name of the task group associated with the task.

Type: String
Required: No

**lastStatus**

The last known status of the task.

Type: String
Required: No

**launchType**

The launch type on which your task is running.

Type: String

Valid Values: **EC2** | **FARGATE**

Required: No

**memory**

The amount (in MiB) of memory used by the task. If using the EC2 launch type, this field is optional and any value can be used. If using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the *cpu* parameter:

- 0.5GB, 1GB, 2GB - Available *cpu* values: 256 (.25 vCPU)
- 1GB, 2GB, 3GB, 4GB - Available *cpu* values: 512 (.5 vCPU)
• 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB - Available CPU values: 1024 (1 vCPU)
• Between 4GB and 16GB in 1GB increments - Available CPU values: 2048 (2 vCPU)
• Between 8GB and 30GB in 1GB increments - Available CPU values: 4096 (4 vCPU)

Type: String
Required: No

overrides
One or more container overrides.

Type: TaskOverride (p. 216) object
Required: No

platformVersion
The platform version on which your task is running. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String
Required: No

pullStartedAt
The Unix timestamp for when the container image pull began.

Type: Timestamp
Required: No

pullStoppedAt
The Unix timestamp for when the container image pull completed.

Type: Timestamp
Required: No

startedAt
The Unix timestamp for when the task started (the task transitioned from the PENDING state to the RUNNING state).

Type: Timestamp
Required: No

startedBy
The tag specified when a task is started. If the task is started by an Amazon ECS service, then the startedBy parameter contains the deployment ID of the service that starts it.

Type: String
Required: No

stoppedAt
The Unix timestamp for when the task was stopped (the task transitioned from the RUNNING state to the STOPPED state).

Type: Timestamp
Required: No

**stoppedReason**

The reason the task was stopped.

Type: String

Required: No

**stoppingAt**

The Unix time stamp for when the task will stop (the task transitioned from the **RUNNING** state to the **STOPPED** state).

Type: Timestamp

Required: No

**taskArn**

The Amazon Resource Name (ARN) of the task.

Type: String

Required: No

**taskDefinitionArn**

The ARN of the task definition that creates the task.

Type: String

Required: No

**version**

The version counter for the task. Every time a task experiences a change that triggers a CloudWatch event, the version counter is incremented. If you are replicating your Amazon ECS task state with CloudWatch Events, you can compare the version of a task reported by the Amazon ECS APIs with the version reported in CloudWatch Events for the task (inside the `detail` object) to verify that the version in your event stream is current.

Type: Long

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
TaskDefinition

Details of a task definition.

Contents

compatibilities

The launch type to use with your task. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

Type: Array of strings

Valid Values: EC2 | FARGATE

Required: No

containerDefinitions

A list of container definitions in JSON format that describe the different containers that make up your task. For more information about container definition parameters and defaults, see Amazon ECS Task Definitions in the Amazon Elastic Container Service Developer Guide.

Type: Array of ContainerDefinition objects

Required: No

cpu

The number of CPU units used by the task. If using the EC2 launch type, this field is optional and any value can be used. If using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the memory parameter:

- 256 (.25 vCPU) - Available memory values: 0.5GB, 1GB, 2GB
- 512 (.5 vCPU) - Available memory values: 1GB, 2GB, 3GB, 4GB
- 1024 (1 vCPU) - Available memory values: 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB
- 2048 (2 vCPU) - Available memory values: Between 4GB and 16GB in 1GB increments
- 4096 (4 vCPU) - Available memory values: Between 8GB and 30GB in 1GB increments

Type: String

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution role that the Amazon ECS container agent and the Docker daemon can assume.

Type: String

Required: No

family

The family of your task definition, used as the definition name.

Type: String

Required: No
memory

The amount (in MiB) of memory used by the task. If using the EC2 launch type, this field is optional and any value can be used. If using the Fargate launch type, this field is required and you must use one of the following values, which determines your range of valid values for the cpu parameter:

- 0.5GB, 1GB, 2GB - Available cpu values: 256 (.25 vCPU)
- 1GB, 2GB, 3GB, 4GB - Available cpu values: 512 (.5 vCPU)
- 2GB, 3GB, 4GB, 5GB, 6GB, 7GB, 8GB - Available cpu values: 1024 (1 vCPU)
- Between 4GB and 16GB in 1GB increments - Available cpu values: 2048 (2 vCPU)
- Between 8GB and 30GB in 1GB increments - Available cpu values: 4096 (4 vCPU)

Type: String
Required: No

networkMode

The Docker networking mode to use for the containers in the task. The valid values are none, bridge, awsvpc, and host. The default Docker network mode is bridge. If using the Fargate launch type, the awsvpc network mode is required. If using the EC2 launch type, any network mode can be used. If the network mode is set to none, you can't specify port mappings in your container definitions, and the task's containers do not have external connectivity. The host and awsvpc network modes offer the highest networking performance for containers because they use the EC2 network stack instead of the virtualized network stack provided by the bridge mode.

With the host and awsvpc network modes, exposed container ports are mapped directly to the corresponding host port (for the host network mode) or the attached elastic network interface port (for the awsvpc network mode), so you cannot take advantage of dynamic host port mappings.

If the network mode is awsvpc, the task is allocated an Elastic Network Interface, and you must specify a NetworkConfiguration (p. 195) when you create a service or run a task with the task definition. For more information, see Task Networking in the Amazon Elastic Container Service Developer Guide.

Note
Currently, only the Amazon ECS-optimized AMI, other Amazon Linux variants with the ecs-init package, or AWS Fargate infrastructure support the awsvpc network mode.

If the network mode is host, you can't run multiple instantiations of the same task on a single container instance when port mappings are used.

Docker for Windows uses different network modes than Docker for Linux. When you register a task definition with Windows containers, you must not specify a network mode. If you use the console to register a task definition with Windows containers, you must choose the <default> network mode object.

For more information, see Network settings in the Docker run reference.

Type: String
Valid Values: bridge | host | awsvpc | none
Required: No

placementConstraints

An array of placement constraint objects to use for tasks. This field is not valid if using the Fargate launch type for your task.

Type: Array of TaskDefinitionPlacementConstraint (p. 215) objects
Required: No

**requiresAttributes**

The container instance attributes required by your task. This field is not valid if using the Fargate launch type for your task.

Type: Array of [Attribute](p. 160) objects

Required: No

**requiresCompatibilities**

The launch type the task is using.

Type: Array of strings

Valid Values: EC2 | FARGATE

Required: No

**revision**

The revision of the task in a particular family. The revision is a version number of a task definition in a family. When you register a task definition for the first time, the revision is 1; each time you register a new revision of a task definition in the same family, the revision value always increases by one (even if you have deregistered previous revisions in this family).

Type: Integer

Required: No

**status**

The status of the task definition.

Type: String

Valid Values: ACTIVE | INACTIVE

Required: No

**taskDefinitionArn**

The full Amazon Resource Name (ARN) of the task definition.

Type: String

Required: No

**taskRoleArn**

The ARN of the IAM role that containers in this task can assume. All containers in this task are granted the permissions that are specified in this role.

IAM roles for tasks on Windows require that the -EnableTaskIamRole option is set when you launch the Amazon ECS-optimized Windows AMI. Your containers must also run some configuration code in order to take advantage of the feature. For more information, see [Windows IAM Roles for Tasks](#) in the [Amazon Elastic Container Service Developer Guide](#).

Type: String

Required: No

**volumes**

The list of volumes in a task.
If you are using the Fargate launch type, the `host` and `sourcePath` parameters are not supported.

For more information about volume definition parameters and defaults, see Amazon ECS Task Definitions in the Amazon Elastic Container Service Developer Guide.

Type: Array of Volume (p. 219) objects

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
TaskDefinitionPlacementConstraint

An object representing a constraint on task placement in the task definition.

If you are using the Fargate launch type, task placement constraints are not supported.

For more information, see Task Placement Constraints in the Amazon Elastic Container Service Developer Guide.

Contents

expression

A cluster query language expression to apply to the constraint. For more information, see Cluster Query Language in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

type

The type of constraint. The DistinctInstance constraint ensures that each task in a particular group is running on a different container instance. The MemberOf constraint restricts selection to be from a group of valid candidates.

Type: String

Valid Values: memberOf

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
TaskOverride

The overrides associated with a task.

Contents

containerOverrides

One or more container overrides sent to a task.

Type: Array of ContainerOverride (p. 177) objects

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution role that the Amazon ECS container agent and the Docker daemon can assume.

Type: String

Required: No

taskRoleArn

The Amazon Resource Name (ARN) of the IAM role that containers in this task can assume. All containers in this task are granted the permissions that are specified in this role.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Ulimit

The ulimit settings to pass to the container.

Contents

hardLimit

The hard limit for the ulimit type.

Type: Integer

Required: Yes

name

The type of the ulimit.

Type: String

Valid Values: core | cpu | data | fsize | locks | memlock | msgqueue | nice | nofile | nproc | rss | rtprio | rttme | sigpending | stack

Required: Yes

softLimit

The soft limit for the ulimit type.

Type: Integer

Required: Yes

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
VersionInfo

The Docker and Amazon ECS container agent version information about a container instance.

Contents

agentHash

The Git commit hash for the Amazon ECS container agent build on the [amazon-ecs-agent](https://github.com/aws/amazon-ecs-agent) GitHub repository.

Type: String

Required: No

agentVersion

The version number of the Amazon ECS container agent.

Type: String

Required: No

dockerVersion

The Docker version running on the container instance.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Volume

A data volume used in a task definition.

Contents

host

The contents of the host parameter determine whether your data volume persists on the host container instance and where it is stored. If the host parameter is empty, then the Docker daemon assigns a host path for your data volume, but the data is not guaranteed to persist after the containers associated with it stop running.

Windows containers can mount whole directories on the same drive as $env:ProgramData. Windows containers cannot mount directories on a different drive, and mount point cannot be across drives. For example, you can mount C:\my\path:C:\my\path and D:\D:\, but not D:\my \path:C:\my\path or D:\C:\my\path.

Type: HostVolumeProperties (p. 186) object

name

The name of the volume. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. This name is referenced in the sourceVolume parameter of container definition mountPoints.

Type: String

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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
VolumeFrom

Details on a data volume from another container in the same task definition.

Contents

readOnly

If this value is true, the container has read-only access to the volume. If this value is false, then the container can write to the volume. The default value is false.

Type: Boolean
Required: No

sourceContainer

The name of another container within the same task definition to mount volumes from.

Type: String
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 Go
- AWS SDK for Java
- AWS SDK for Ruby V2
Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the Amazon Web Services General Reference.

**Action**

The action to be performed.

Type: string

Required: Yes

**Version**

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

**X-Amz-Algorithm**

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

**X-Amz-Credential**

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string (“aws4_request”). The value is expressed in the following format: access_key/YYYYMMDD/region/service/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

**X-Amz-Date**

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is
not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see **Handling Dates in Signature Version 4** in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

**X-Amz-Security-Token**

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to *AWS Services That Work with IAM* in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

**X-Amz-Signature**

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

**X-Amz-SignedHeaders**

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see **Task 1: Create a Canonical Request For Signature Version 4** in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional
Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

**AccessDeniedException**

You do not have sufficient access to perform this action.

HTTP Status Code: 400

**IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

**InvalidAction**

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

**InvalidClientTokenId**

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

**InvalidParameterCombination**

Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

**InvalidQueryParameter**

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

**MalformedQueryString**

The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**

The request is missing an action or a required parameter.

HTTP Status Code: 400
**MissingAuthenticationToken**

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**OptInRequired**

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

**RequestExpired**

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

**ServiceUnavailable**

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

**ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationException**

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400