Amazon Elastic Container Service: API Reference
Copyright © 2020 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.
# Table of Contents

Welcome ........................................................................................................................................... 1  
Actions ............................................................................................................................................... 2  
CreateCapacityProvider ................................................................. 4  
  Request Syntax ................................................................................. 4  
  Request Parameters ........................................................................ 4  
  Response Syntax .............................................................................. 5  
  Response Elements ........................................................................ 5  
  Errors ............................................................................................... 6  
  See Also ........................................................................................... 6  
CreateCluster ............................................................................... 7  
  Request Syntax ................................................................................. 7  
  Request Parameters ........................................................................ 7  
  Response Syntax .............................................................................. 9  
  Response Elements ......................................................................... 10  
  Errors ............................................................................................... 10  
  Example ............................................................................................ 10  
  See Also ........................................................................................... 11  
CreateService .............................................................................. 12  
  Request Syntax ................................................................................. 13  
  Request Parameters ........................................................................ 14  
  Response Syntax .............................................................................. 19  
  Response Elements ......................................................................... 22  
  Errors ............................................................................................... 22  
  Examples .......................................................................................... 23  
  See Also ........................................................................................... 26  
CreateTaskSet .............................................................................. 27  
  Request Syntax ................................................................................. 27  
  Request Parameters ........................................................................ 28  
  Response Syntax .............................................................................. 30  
  Response Elements ......................................................................... 31  
  Errors ............................................................................................... 31  
  See Also ........................................................................................... 32  
DeleteAccountSetting ................................................................. 34  
  Request Syntax ................................................................................. 34  
  Request Parameters ........................................................................ 34  
  Response Syntax .............................................................................. 34  
  Response Elements ......................................................................... 34  
  Errors ............................................................................................... 35  
  See Also ........................................................................................... 35  
DeleteAttributes .......................................................................... 36  
  Request Syntax ................................................................................. 36  
  Request Parameters ........................................................................ 36  
  Response Syntax .............................................................................. 36  
  Response Elements ......................................................................... 36  
  Errors ............................................................................................... 37  
  Example ............................................................................................ 37  
  See Also ........................................................................................... 38  
DeleteCluster .............................................................................. 39  
  Request Syntax ................................................................................. 39  
  Request Parameters ........................................................................ 39  
  Response Syntax .............................................................................. 39  
  Response Elements ......................................................................... 40  
  Errors ............................................................................................... 40  
  Example ............................................................................................ 41
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>UpdateContainerInstancesState</td>
<td>223</td>
</tr>
<tr>
<td>UpdateClusterSettings</td>
<td>215</td>
</tr>
<tr>
<td>TagResource</td>
<td>209</td>
</tr>
<tr>
<td>SubmitAttachmentStateChanges</td>
<td>200</td>
</tr>
<tr>
<td>SubmitContainerStateChange</td>
<td>202</td>
</tr>
<tr>
<td>SubmitTaskStateChange</td>
<td>205</td>
</tr>
<tr>
<td>UntagResource</td>
<td>212</td>
</tr>
<tr>
<td>UpdateClusterSettings</td>
<td>215</td>
</tr>
<tr>
<td>UpdateContainerAgent</td>
<td>218</td>
</tr>
<tr>
<td>UpdateContainerInstancesState</td>
<td>223</td>
</tr>
</tbody>
</table>

**Response Elements** ................................................................. 197
**Errors** .................................................................................... 197
**Example** .................................................................................. 197
**See Also** ................................................................................ 199
**SubmitAttachmentStateChanges** ............................................. 200
**Request Syntax** ................................................................. 200
**Request Parameters** ........................................................... 200
**Response Syntax** ............................................................... 200
**Response Elements** ............................................................ 200
**Errors** .................................................................................. 201
**See Also** ................................................................................ 201
**SubmitContainerStateChange** .................................................. 202
**Request Syntax** ................................................................. 202
**Request Parameters** ........................................................... 202
**Response Syntax** ............................................................... 203
**Response Elements** ............................................................ 203
**Errors** .................................................................................. 203
**See Also** ................................................................................ 204
**SubmitTaskStateChange** ......................................................... 205
**Request Syntax** ................................................................. 205
**Request Parameters** ........................................................... 205
**Response Syntax** ............................................................... 206
**Response Elements** ............................................................ 207
**Errors** .................................................................................. 207
**See Also** ................................................................................ 207
**TagResource** ........................................................................... 209
**Request Syntax** ................................................................. 209
**Request Parameters** ........................................................... 209
**Response Elements** ............................................................ 210
**Errors** .................................................................................. 210
**Example** .................................................................................. 210
**See Also** ................................................................................ 211
**UntagResource** ....................................................................... 212
**Request Syntax** ................................................................. 212
**Request Parameters** ........................................................... 212
**Response Elements** ............................................................ 212
**Errors** .................................................................................. 213
**Example** .................................................................................. 213
**See Also** ................................................................................ 214
**UpdateClusterSettings** ......................................................... 215
**Request Syntax** ................................................................. 215
**Request Parameters** ........................................................... 215
**Response Syntax** ............................................................... 215
**Response Elements** ............................................................ 216
**Errors** .................................................................................. 216
**See Also** ................................................................................ 217
**UpdateContainerAgent** .......................................................... 218
**Request Syntax** ................................................................. 218
**Request Parameters** ........................................................... 218
**Response Syntax** ............................................................... 218
**Response Elements** ............................................................ 219
**Errors** .................................................................................. 220
**Example** .................................................................................. 221
**See Also** ................................................................................ 221
**UpdateContainerInstancesState** ............................................ 223
**Request Syntax** ................................................................. 223
**Request Parameters** ........................................................... 223
<table>
<thead>
<tr>
<th>Common Parameters</th>
<th>Common Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>VolumeFrom</td>
<td>371</td>
</tr>
<tr>
<td>Volume</td>
<td>373</td>
</tr>
<tr>
<td>VersionInfo</td>
<td>367</td>
</tr>
<tr>
<td>Ulimit</td>
<td>366</td>
</tr>
<tr>
<td>Tmpfs</td>
<td>365</td>
</tr>
<tr>
<td>TaskDefinitionPlacementConstraint</td>
<td>356</td>
</tr>
<tr>
<td>TaskOverride</td>
<td>358</td>
</tr>
<tr>
<td>TaskDefinition</td>
<td>352</td>
</tr>
<tr>
<td>TaskSet</td>
<td>360</td>
</tr>
<tr>
<td>Setting</td>
<td>342</td>
</tr>
<tr>
<td>SystemControl</td>
<td>343</td>
</tr>
<tr>
<td>Tag</td>
<td>344</td>
</tr>
<tr>
<td>Task</td>
<td>346</td>
</tr>
<tr>
<td>ServiceRegistry</td>
<td>340</td>
</tr>
<tr>
<td>ServiceEvent</td>
<td>339</td>
</tr>
<tr>
<td>Service</td>
<td>334</td>
</tr>
<tr>
<td>Common Errors</td>
<td>373</td>
</tr>
<tr>
<td>Common Parameters</td>
<td>371</td>
</tr>
<tr>
<td>See Also</td>
<td>333</td>
</tr>
<tr>
<td>See Also</td>
<td>338</td>
</tr>
<tr>
<td>See Also</td>
<td>339</td>
</tr>
<tr>
<td>See Also</td>
<td>340</td>
</tr>
<tr>
<td>See Also</td>
<td>342</td>
</tr>
<tr>
<td>See Also</td>
<td>343</td>
</tr>
<tr>
<td>See Also</td>
<td>344</td>
</tr>
<tr>
<td>See Also</td>
<td>346</td>
</tr>
<tr>
<td>See Also</td>
<td>351</td>
</tr>
<tr>
<td>See Also</td>
<td>352</td>
</tr>
<tr>
<td>See Also</td>
<td>356</td>
</tr>
<tr>
<td>See Also</td>
<td>357</td>
</tr>
<tr>
<td>See Also</td>
<td>358</td>
</tr>
<tr>
<td>See Also</td>
<td>360</td>
</tr>
<tr>
<td>See Also</td>
<td>363</td>
</tr>
<tr>
<td>See Also</td>
<td>365</td>
</tr>
<tr>
<td>See Also</td>
<td>366</td>
</tr>
<tr>
<td>See Also</td>
<td>366</td>
</tr>
<tr>
<td>See Also</td>
<td>367</td>
</tr>
<tr>
<td>See Also</td>
<td>367</td>
</tr>
<tr>
<td>See Also</td>
<td>368</td>
</tr>
<tr>
<td>See Also</td>
<td>369</td>
</tr>
<tr>
<td>See Also</td>
<td>370</td>
</tr>
<tr>
<td>See Also</td>
<td>370</td>
</tr>
<tr>
<td>See Also</td>
<td>371</td>
</tr>
<tr>
<td>See Also</td>
<td>373</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 April 5, 2020.
Actions

The following actions are supported:

- `CreateCapacityProvider` (p. 4)
- `CreateCluster` (p. 7)
- `CreateService` (p. 12)
- `CreateTaskSet` (p. 27)
- `DeleteAccountSetting` (p. 34)
- `DeleteAttributes` (p. 36)
- `DeleteCluster` (p. 39)
- `DeleteService` (p. 43)
- `DeleteTaskSet` (p. 50)
- `DeregisterContainerInstance` (p. 54)
- `DeregisterTaskDefinition` (p. 60)
- `DescribeCapacityProviders` (p. 67)
- `DescribeClusters` (p. 70)
- `DescribeContainerInstances` (p. 74)
- `DescribeServices` (p. 80)
- `DescribeTaskDefinition` (p. 87)
- `DescribeTasks` (p. 94)
- `DescribeTaskSets` (p. 100)
- `DiscoverPollEndpoint` (p. 104)
- `ListAccountSettings` (p. 106)
- `ListAttributes` (p. 109)
- `ListClusters` (p. 113)
- `ListContainerInstances` (p. 116)
- `ListServices` (p. 120)
- `ListTagsForResource` (p. 124)
- `ListTaskDefinitionFamilies` (p. 127)
- `ListTaskDefinitions` (p. 132)
- `ListTasks` (p. 136)
- `PutAccountSetting` (p. 141)
- `PutAccountSettingDefault` (p. 144)
- `PutAttributes` (p. 146)
- `PutClusterCapacityProviders` (p. 150)
- `RegisterContainerInstance` (p. 154)
- `RegisterTaskDefinition` (p. 159)
- `RunTask` (p. 174)
- `StartTask` (p. 185)
- `StopTask` (p. 194)
- `SubmitAttachmentStateChanges` (p. 200)
- `SubmitContainerStateChange` (p. 202)
- `SubmitTaskStateChange` (p. 205)
- TagResource (p. 209)
- UntagResource (p. 212)
- UpdateClusterSettings (p. 215)
- UpdateContainerAgent (p. 218)
- UpdateContainerInstancesState (p. 223)
- UpdateService (p. 231)
- UpdateServicePrimaryTaskSet (p. 241)
- UpdateTaskSet (p. 245)
CreateCapacityProvider

Creates a new capacity provider. Capacity providers are associated with an Amazon ECS cluster and are used in capacity provider strategies to facilitate cluster auto scaling.

Only capacity providers using an Auto Scaling group can be created. Amazon ECS tasks on AWS Fargate use the FARGATE and FARGATE_SPOT capacity providers which are already created and available to all accounts in Regions supported by AWS Fargate.

Request Syntax

```json
{
   "autoScalingGroupProvider": {
      "autoScalingGroupArn": "string",
      "managedScaling": {
         "maximumScalingStepSize": number,
         "minimumScalingStepSize": number,
         "status": "string",
         "targetCapacity": number
      },
      "managedTerminationProtection": "string"
   },
   "name": "string",
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**autoScalingGroupProvider (p. 4)**

The details of the Auto Scaling group for the capacity provider.

Type: AutoScalingGroupProvider (p. 254) object

Required: Yes

**name (p. 4)**

The name of the capacity provider. Up to 255 characters are allowed, including letters (upper and lowercase), numbers, underscores, and hyphens. The name cannot be prefixed with "aws", "ecs", or "fargate".

Type: String

Required: Yes

**tags (p. 4)**

The metadata that you apply to the capacity provider to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.
The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix.
- Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.
Required: No

Response Syntax

```json
{
    "capacityProvider": {
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "string",
            "managedScaling": {
                "maximumScalingStepSize": number,
                "minimumScalingStepSize": number,
                "status": "string",
                "targetCapacity": number
            },
            "managedTerminationProtection": "string"
        },
        "capacityProviderArn": "string",
        "name": "string",
        "status": "string",
        "tags": [
            {
                "key": "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.

`capacityProvider` (p. 5)

The full description of the new capacity provider.

Type: `CapacityProvider` (p. 256) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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

LimitExceededException

The limit for the resource has been exceeded.

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 V3
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. 7) API operation, Amazon ECS attempts to create the Amazon ECS 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
{
    "capacityProviders": [ "string" ],
    "clusterName": "string",
    "defaultCapacityProviderStrategy": [ {
        "base": number,
        "capacityProvider": "string",
        "weight": number
    } ],
    "settings": [ {
        "name": "string",
        "value": "string"
    } ],
    "tags": [ {
        "key": "string",
        "value": "string"
    } ]
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**capacityProviders** (p. 7)

The short name of one or more capacity providers to associate with the cluster.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created and not already associated with another cluster. New capacity providers can be created with the CreateCapacityProvider (p. 4) API operation.

To use a AWS Fargate capacity provider, specify either the FARGATE or FARGATE_SPOT capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.
The `PutClusterCapacityProviders` API operation is used to update the list of available capacity providers for a cluster after the cluster is created.

**Type:** Array of strings  
**Required:** No

### clusterName (p. 7)

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, and hyphens are allowed.

**Type:** String  
**Required:** No

### defaultCapacityProviderStrategy (p. 7)

The capacity provider strategy to use by default for the cluster.

When creating a service or running a task on a cluster, if no capacity provider or launch type is specified then the default capacity provider strategy for the cluster is used.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The `PutClusterCapacityProviders` API is used to associate a capacity provider with a cluster. Only capacity providers with an `ACTIVE` or `UPDATING` status can be used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the `CreateCapacityProvider` API operation.

To use a AWS Fargate capacity provider, specify either the `FARGATE` or `FARGATE_SPOT` capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

If a default capacity provider strategy is not defined for a cluster during creation, it can be defined later with the `PutClusterCapacityProviders` API operation.

**Type:** Array of `CapacityProviderStrategyItem` objects  
**Required:** No

### settings (p. 7)

The setting to use when creating a cluster. This parameter is used to enable CloudWatch Container Insights for a cluster. If this value is specified, it will override the `containerInsights` value set with `PutAccountSetting` or `PutAccountSettingDefault`.

**Type:** Array of `ClusterSetting` objects  
**Required:** No

### tags (p. 7)

The metadata that you apply to the cluster to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
• Maximum value length - 256 Unicode characters in UTF-8
• If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - . : / @.
• Tag keys and values are case-sensitive.
• Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```json
{
    "cluster": {
        "activeServicesCount": number,
        "attachments": [
            {
                "details": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "id": "string",
                "status": "string",
                "type": "string"
            }
        ],
        "attachmentsStatus": "string",
        "capacityProviders": [ "string" ],
        "clusterArn": "string",
        "clusterName": "string",
        "defaultCapacityProviderStrategy": [
            {
                "base": number,
                "capacityProvider": "string",
                "weight": number
            }
        ],
        "pendingTasksCount": number,
        "registeredContainerInstancesCount": number,
        "runningTasksCount": number,
        "settings": [
            {
                "name": "string",
                "value": "string"
            }
        ],
        "statistics": [
            {
                "name": "string",
                "value": "string"
            }
        ],
        "status": "string",
        "tags": [
        
```
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. 9)

The full description of your new cluster.

Type: `Cluster` (p. 259) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3

API Version 2014-11-13
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 the `desiredCount`, Amazon ECS runs another copy of the task in the specified cluster. To update an existing service, see the UpdateService action.

In addition to maintaining the desired count of tasks in your service, you can optionally run your service behind one or more load balancers. The load balancers distribute 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.

Tasks for services that do not use a load balancer are considered healthy if they're in the `RUNNING` state.

Tasks for services that do use a load balancer are considered healthy if they're in the `RUNNING` state and the container instance that they're hosted on is reported as healthy by the load balancer.

There are two service scheduler strategies available:

- **REPLICA** - The replica scheduling strategy places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions. For more information, see Service Scheduler Concepts in the Amazon Elastic Container Service Developer Guide.

- **DAEMON** - The daemon scheduling strategy deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks and will stop tasks that do not meet the placement constraints. When using this strategy, you don't need to specify a desired number of tasks, a task placement strategy, or use Service Auto Scaling policies. For more information, see Service Scheduler Concepts in the Amazon Elastic Container Service Developer Guide.

You can optionally specify a deployment configuration for your service. The deployment is triggered by changing properties, such as the task definition or the desired count of a service, with an UpdateService (p. 231) operation. The default value for a replica service for `minimumHealthyPercent` is 100%. The default value for a daemon service for `minimumHealthyPercent` is 0%.

If a service is using the ECS deployment controller, the minimum healthy percent represents a lower limit on the number of tasks in a service that must remain in the `RUNNING` state during a deployment, as a percentage of the desired number of tasks (rounded up to the nearest integer), and while any container instances are in the `DRAINING` state if the service contains tasks using the EC2 launch type. This parameter enables you to deploy without using additional cluster capacity. For example, if your service has a desired number of four tasks and a minimum healthy percent of 50%, the scheduler might 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're in the `RUNNING` state. Tasks for services that do use a load balancer are considered healthy if they're in the `RUNNING` state and they're reported as healthy by the load balancer. The default value for minimum healthy percent is 100%.

If a service is using the ECS deployment controller, the **maximum percent** parameter represents an upper limit on the number of tasks in a service that are allowed in the `RUNNING` or `PENDING` state during a deployment, as a percentage of the desired number of tasks (rounded down to the nearest integer), and while any container instances are in the `DRAINING` state if the service contains tasks using the EC2 launch type. This parameter enables you to define the deployment batch size. For example, if your service has a desired number of four tasks and a maximum percent value of 200%, the scheduler may 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 maximum percent is 200%.

If a service is using either the CODE_DEPLOY or EXTERNAL deployment controller types and tasks that use the EC2 launch type, the **minimum healthy percent** and **maximum percent** values are used only to
define the lower and upper limit on the number of the tasks in the service that remain in the **RUNNING** state while the container instances are in the **DRAINING** state. If the tasks in the service use the Fargate launch type, the minimum healthy percent and maximum percent values aren’t used, although they’re currently visible when describing your service.

When creating a service that uses the EXTERNAL deployment controller, you can specify only parameters that aren’t controlled at the task set level. The only required parameter is the service name. You control your services using the `CreateTaskSet` (p. 27) operation. For more information, see Amazon ECS Deployment Types in the *Amazon Elastic Container Service Developer Guide*.

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, giving priority to instances that have the fewest number of running tasks for this service in their respective Availability Zone. 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
{
    "capacityProviderStrategy": [
        {
            "base": number,
            "capacityProvider": "string",
            "weight": number
        }
    ],
    "clientToken": "string",
    "cluster": "string",
    "deploymentConfiguration": {
        "maximumPercent": number,
        "minimumHealthyPercent": number
    },
    "deploymentController": {
        "type": "string"
    },
    "desiredCount": number,
    "enableECSManagedTags": boolean,
    "healthCheckGracePeriodSeconds": number,
    "launchType": "string",
    "loadBalancers": [
        {
            "containerName": "string",
            "containerPort": number,
            "loadBalancerName": "string",
            "targetGroupArn": "string"
        }
    ],
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "assignPublicIp": "string",
        }
    }
}
```

*API Version 2014-11-13*
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

capacityProviderStrategy (p. 13)

The capacity provider strategy to use for the service.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The PutClusterCapacityProviders (p. 150) API is used to associate a capacity provider with a cluster. Only capacity providers with an ACTIVE or UPDATING status can be used.

If a capacityProviderStrategy is specified, the launchType parameter must be omitted. If no capacityProviderStrategy or launchType is specified, the defaultCapacityProviderStrategy for the cluster is used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the CreateCapacityProvider (p. 4) API operation.
To use a AWS Fargate capacity provider, specify either the FARGATE or FARGATE_SPOT capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

The PutClusterCapacityProviders (p. 150) API operation is used to update the list of available capacity providers for a cluster after the cluster is created.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

clientToken (p. 13)

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

Type: String

Required: No

cluster (p. 13)

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. 13)

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

Type: DeploymentConfiguration (p. 291) object

Required: No

deploymentController (p. 13)

The deployment controller to use for the service.

Type: DeploymentController (p. 293) object

Required: No

desiredCount (p. 13)

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

This is required if schedulingStrategy is REPLICA or is not specified. If schedulingStrategy is DAEMON then this is not required.

Type: Integer

Required: No

enableECSManagedTags (p. 13)

Specifies whether to enable Amazon ECS managed tags for the tasks within the service. For more information, see Tagging Your Amazon ECS Resources in the Amazon Elastic Container Service Developer Guide.

Type: Boolean
Required: No

**healthCheckGracePeriodSeconds (p. 13)**

The period of time, in seconds, that the Amazon ECS service scheduler should ignore unhealthy Elastic Load Balancing target health checks after a task has first started. This is only used when your service is configured to use a load balancer. If your service has a load balancer defined and you don't specify a health check grace period value, the default value of 0 is used.

If your service's tasks take a while to start and respond to Elastic Load Balancing health checks, you can specify a health check grace period of up to 2,147,483,647 seconds. During that time, the Amazon ECS service scheduler ignores health check status. This grace period can prevent the service scheduler from marking tasks as unhealthy and stopping them before they have time to come up.

Type: Integer

Required: No

**launchType (p. 13)**

The launch type on which to run your service. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

If a `launchType` is specified, the `capacityProviderStrategy` parameter must be omitted.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**loadBalancers (p. 13)**

A load balancer object representing the load balancers to use with your service. For more information, see Service Load Balancing in the Amazon Elastic Container Service Developer Guide.

If the service is using the rolling update (ECS) deployment controller and using either an Application Load Balancer or Network Load Balancer, you can specify multiple target groups to attach to the service. The service-linked role is required for services that make use of multiple target groups. For more information, see Using Service-Linked Roles for Amazon ECS in the Amazon Elastic Container Service Developer Guide.

If the service is using the CODE_DEPLOY deployment controller, the service is required to use either an Application Load Balancer or Network Load Balancer. When creating an AWS CodeDeploy deployment group, you specify two target groups (referred to as a `targetGroupPair`). During a deployment, AWS CodeDeploy determines which task set in your service has the status `PRIMARY` and associates one target group with it, and then associates the other target group with the replacement task set. The load balancer can also have up to two listeners: a required listener for production traffic and an optional listener that allows you perform validation tests with Lambda functions before routing production traffic to it.

After you create a service using the ECS deployment controller, the load balancer name or target group ARN, container name, and container port specified in the service definition are immutable. If you are using the CODE_DEPLOY deployment controller, these values can be changed when updating the service.

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.
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.

Services with tasks that use the `awsvpc` network mode (for example, those with the Fargate launch type) only support Application Load Balancers and Network Load Balancers. Classic Load Balancers are not supported. Also, when you create any target groups for these services, you must choose `ip` as the target type, not `instance`, because tasks that use the `awsvpc` network mode are associated with an elastic network interface, not an Amazon EC2 instance.

Type: Array of `LoadBalancer` (p. 311) objects

Required: No

**networkConfiguration (p. 13)**

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. 319) object

Required: No

**placementConstraints (p. 13)**

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 runtime).

Type: Array of `PlacementConstraint` (p. 321) objects

Required: No

**placementStrategy (p. 13)**

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. 322) objects

Required: No

**platformVersion (p. 13)**

The platform version that your tasks in the service are running on. A platform version is specified only for tasks using the Fargate launch type. If one isn't specified, the `LATEST` platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

**propagateTags (p. 13)**

Specifies whether to propagate the tags from the task definition or the service to the tasks in the service. If no value is specified, the tags are not propagated. Tags can only be propagated to the tasks within the service during service creation. To add tags to a task after service creation, use the `TagResource` (p. 209) API action.

Type: String
Valid Values: TASK_DEFINITION | SERVICE

Required: No

role (p. 13)

The name or full Amazon Resource Name (ARN) of the IAM role that allows Amazon ECS to make calls to your load balancer on your behalf. This parameter is only permitted if you are using a load balancer with your service and your task definition does not use the awsvpc network mode. If you specify the role parameter, you must also specify a load balancer object with the loadBalancers parameter.

Important
If your account has already created the Amazon ECS service-linked role, that role is used by default for your service unless you specify a role here. The service-linked role is required if your task definition uses the awsvpc network mode or if the service is configured to use service discovery, an external deployment controller, multiple target groups, or Elastic Inference accelerators in which case you should not specify a role here. For more information, see Using Service-Linked Roles for Amazon ECS in the Amazon Elastic Container Service Developer Guide.

If your specified role has a path other than /, then you must either specify the full role ARN (this is recommended) or prefix the role name with the path. For example, if a role with the name bar has a path of /foo/ then you would specify /foo/bar as the role name. For more information, see Friendly Names and Paths in the IAM User Guide.

Type: String

Required: No

schedulingStrategy (p. 13)

The scheduling strategy to use for the service. For more information, see Services.

There are two service scheduler strategies available:

• REPLICA-The replica scheduling strategy places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions. This scheduler strategy is required if the service is using the CODE_DEPLOY or EXTERNAL deployment controller types.

• DAEMON-The daemon scheduling strategy deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks and will stop tasks that do not meet the placement constraints. When you’re using this strategy, you don’t need to specify a desired number of tasks, a task placement strategy, or use Service Auto Scaling policies.

Note
Tasks using the Fargate launch type or the CODE_DEPLOY or EXTERNAL deployment controller types don’t support the DAEMON scheduling strategy.

Type: String

Valid Values: REPLICA | DAEMON

Required: No

serviceName (p. 13)

The name of your service. Up to 255 letters (uppercase and lowercase), numbers, and hyphens 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: Yes  

**serviceRegistries (p. 13)**  
The details of the service discovery registries to assign to this service. For more information, see **Service Discovery**.

**Note**
Service discovery is supported for Fargate tasks if you are using platform version v1.1.0 or later. For more information, see AWS Fargate Platform Versions.

Type: Array of **ServiceRegistry (p. 340)** objects  
Required: No

**tags (p. 13)**  
The metadata that you apply to the service to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define. When a service is deleted, the tags are deleted as well.

The following basic restrictions apply to tags:
- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of **Tag (p. 344)** objects  
Array Members: Minimum number of 0 items. Maximum number of 50 items.
Required: No

**taskDefinition (p. 13)**  
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.

A task definition must be specified if the service is using the **ECS** deployment controller.

Type: String  
Required: No

### Response Syntax

```json
{
  "service": {
    "capacityProviderStrategy": [
```
Response Syntax

```json
{
    "base": number,
    "capacityProvider": "string",
    "weight": number
}
",
"clusterArn": "string",
"createdAt": number,
"createdBy": "string",
"deploymentConfiguration": {
    "maximumPercent": number,
    "minimumHealthyPercent": number
},
"deploymentController": {
    "type": "string"
},
"deployments": [
    {
        "capacityProviderStrategy": [
            {
                "base": number,
                "capacityProvider": "string",
                "weight": number
            }
        ],
        "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,
"enableECSManagedTags": boolean,
"events": [
    {
        "createdAt": number,
        "id": "string",
        "message": "string"
    }
],
"healthCheckGracePeriodSeconds": number,
"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",
"propagateTags": "string",
"roleArn": "string",
"runningCount": number,
"schedulingStrategy": "string",
"serviceArn": "string",
"serviceName": "string",
"serviceRegistries": [ {
  "containerName": "string",
  "containerPort": number,
  "port": number,
  "registryArn": "string"
} ],
"status": "string",
"tags": [ {
  "key": "string",
  "value": "string"
} ],
"taskDefinition": "string",
"taskSets": [ {
  "capacityProviderStrategy": [ {
    "base": number,
    "capacityProvider": "string",
    "weight": number
  },
  "clusterArn": "string",
  "computedDesiredCount": number,
  "createdAt": number,
  "externalId": "string",
  "id": "string",
  "launchType": "string",
  "loadBalancers": [ {
    "containerName": "string",
    "containerPort": number,
    "loadBalancerName": "string",
    "targetGroupArn": "string"
  } ],
  "networkConfiguration": {
    "awsvpcConfiguration": { 
      "assignPublicIp": "string",
      "subnets": [ "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. 19)

The full description of your service following the create call.

If a service is using the ECS deployment controller, the deploymentController and taskSets parameters will not be returned.

If the service is using the CODE_DEPLOY deployment controller, the deploymentController, taskSets and deployments parameters will be returned, however the deployments parameter will be an empty list.

Type: Service (p. 334) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).
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. 113). 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

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 1

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
Server: Server
Date: Wed, 29 Apr 2015 17:01:27 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 636
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": 1430326887.362,
        "desiredCount": 10,
        "id": "ecs-svc/9223370606527888445",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-demo:1",
        "updatedAt": 1430326887.362
      }
    ],
    "desiredCount": 10,
    "events": [],
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/ecs-simple-service",
    "serviceName": "ecs-simple-service",
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-demo:1"
  }
}
```
Example 2

This example API request creates a service with multiple load balancer target groups.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerServiceV20141113.CreateService
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.16.190 Python/3.6.1 Darwin/16.7.0 botocore/1.12.180
X-Amz-Date: 20190723T001203Z
Authorization: AUTHPARAMS
Content-Length: 453

{
    "serviceName":"ecs-multiplealb-service",
    "taskDefinition":"ecs-multiplealb-demo",
    "loadBalancers":[
        {
            "targetGroupArn":"arn:aws:elasticloadbalancing:us-east-1:012345678910:targetgroup/tg1/18ce32cc074018ed",
            "containerName":"simple-app",
            "containerPort":80
        },
        {
            "targetGroupArn":"arn:aws:elasticloadbalancing:us-east-1:012345678910:targetgroup/tg2/737bead11d516e2a",
            "containerName":"simple-app",
            "containerPort":8080
        }
    ],
    "desiredCount":10
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 1440
Date: Tue, 23 Jul 2019 00:12:03 GMT
Connection: keep-alive

{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/default/ecs-multiplealb-service",
        "serviceName": "ecs-multiplealb-service",
        "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
        "loadBalancers": [
            {
                "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:012345678910:targetgroup/tg1/18ce32cc074018ed",
                "containerName": "simple-app",
                "containerPort": 80
            },
            {
                "targetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:012345678910:targetgroup/tg2/737bead11d516e2a",
                "containerName": "simple-app",
                "containerPort": 8080
            }
        ],
        "desiredCount": 10
    }
}
```
"containerPort": 8080
],
"serviceRegistries": [],
"status": "ACTIVE",
"desiredCount": 10,
"runningCount": 0,
"pendingCount": 0,
"launchType": "EC2",
"taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-
multiplealb-demo",
"deploymentConfiguration": {
  "maximumPercent": 200,
  "minimumHealthyPercent": 100
},
"deployments": [
  {
    "id": "ecs-svc/9223370473014051517",
    "status": "PRIMARY",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-
multiplealb-demo",
    "desiredCount": 10,
    "pendingCount": 0,
    "runningCount": 0,
    "createdAt": 1563840724.29,
    "updatedAt": 1563840724.29,
    "launchType": "EC2"
  }
],
"roleArn": "arn:aws:iam::012345678910:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS",
"events": [],
"createdAt": 1563840724.29,
"placementConstraints": [],
"placementStrategy": [],
"healthCheckGracePeriodSeconds": 0,
"schedulingStrategy": "REPLICA",
"enableECSManagedTags": false,
"propagateTags": "NONE"
}

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 V3
CreateTaskSet

Create a task set in the specified cluster and service. This is used when a service uses the EXTERNAL deployment controller type. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
  "capacityProviderStrategy": [
    {
      "base": number,
      "capacityProvider": "string",
      "weight": number
    }
  ],
  "clientToken": "string",
  "cluster": "string",
  "externalId": "string",
  "launchType": "string",
  "loadBalancers": [
    {
      "containerName": "string",
      "containerPort": number,
      "loadBalancerName": "string",
      "targetGroupArn": "string"
    }
  ],
  "networkConfiguration": {
    "awsVpcConfiguration": {
      "assignPublicIp": "string",
      "securityGroups": [ "string" ],
      "subnets": [ "string" ]
    }
  },
  "platformVersion": "string",
  "scale": {
    "unit": "string",
    "value": number
  },
  "service": "string",
  "serviceRegistries": [
    {
      "containerName": "string",
      "containerPort": number,
      "port": number,
      "registryArn": "string"
    }
  ],
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
  "taskDefinition": "string"
}
```
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**capacityProviderStrategy (p. 27)**

The capacity provider strategy to use for the task set.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The PutClusterCapacityProviders (p. 150) API is used to associate a capacity provider with a cluster. Only capacity providers with an ACTIVE or UPDATING status can be used.

If a capacityProviderStrategy is specified, the launchType parameter must be omitted. If no capacityProviderStrategy or launchType is specified, the defaultCapacityProviderStrategy for the cluster is used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the CreateCapacityProvider (p. 4) API operation.

To use a AWS Fargate capacity provider, specify either the FARGATE or FARGATE_SPOT capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

The PutClusterCapacityProviders (p. 150) API operation is used to update the list of available capacity providers for a cluster after the cluster is created.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

**clientToken (p. 27)**

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

Type: String

Required: No

**cluster (p. 27)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service to create the task set in.

Type: String

Required: Yes

**externalId (p. 27)**

An optional non-unique tag that identifies this task set in external systems. If the task set is associated with a service discovery registry, the tasks in this task set will have the ECS_TASK_SET_EXTERNAL_ID AWS Cloud Map attribute set to the provided value.

Type: String
Required: No

**launchType (p. 27)**

The launch type that new tasks in the task set will use. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

If a `launchType` is specified, the `capacityProviderStrategy` parameter must be omitted.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**loadBalancers (p. 27)**

A load balancer object representing the load balancer to use with the task set. The supported load balancer types are either an Application Load Balancer or a Network Load Balancer.

Type: Array of `LoadBalancer (p. 311)` objects

Required: No

**networkConfiguration (p. 27)**

An object representing the network configuration for a task or service.

Type: `NetworkConfiguration (p. 319)` object

Required: No

**platformVersion (p. 27)**

The platform version that the tasks in the task set should use. A platform version is specified only for tasks using the Fargate launch type. If one isn't specified, the LATEST platform version is used by default.

Type: String

Required: No

**scale (p. 27)**

A floating-point percentage of the desired number of tasks to place and keep running in the task set.

Type: `Scale (p. 332)` object

Required: No

**service (p. 27)**

The short name or full Amazon Resource Name (ARN) of the service to create the task set in.

Type: String

Required: Yes

**serviceRegistries (p. 27)**

The details of the service discovery registries to assign to this task set. For more information, see Service Discovery.

Type: Array of `ServiceRegistry (p. 340)` objects

Required: No
tags (p. 27)

The metadata that you apply to the task set to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define. When a service is deleted, the tags are deleted as well.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskDefinition (p. 27)

The task definition for the tasks in the task set to use.

Type: String

Required: Yes

Response Syntax

```
{
   "taskSet": {
       "capacityProviderStrategy": [
           {
               "base": number,
               "capacityProvider": "string",
               "weight": number
           }
       ],
       "clusterArn": "string",
       "computedDesiredCount": number,
       "createdAt": number,
       "externalId": "string",
       "id": "string",
       "launchType": "string",
       "loadBalancers": [
           {
               "containerName": "string",
               "containerPort": number,
               "loadBalancerName": "string",
               "targetGroupArn": "string"
           }
       ],
       "networkConfiguration": {
           "assignPublicIp": boolean,
           "awsvpcConfiguration": {
               "subnets": [
                   "string"
               ],
               "securityGroups": [
                   "string"
               ]
           }
       }
   }
}
```

API Version 2014-11-13
"awsvpcConfiguration": {
    "assignPublicIp": "string",
    "securityGroups": [ "string" ],
    "subnets": [ "string" ]
},
"pendingCount": number,
"platformVersion": "string",
"runningCount": number,
"scale": {
    "unit": "string",
    "value": number
},
"serviceArn": "string",
"serviceRegistries": [
    {
        "containerName": "string",
        "containerPort": number,
        "port": number,
        "registryArn": "string"
    }
],
"stabilityStatus": "string",
"stabilityStatusAt": number,
"startedBy": "string",
"status": "string",
"tags": [
    {
        "key": "string",
        "value": "string"
    }
],
"taskDefinition": "string",
"taskSetArn": "string",
"updatedAt": 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.

taskSet (p. 30)

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL
deployment. An Amazon ECS task set includes details such as the desired number of tasks, how
many tasks are running, and whether the task set serves production traffic.

Type: TaskSet (p. 360) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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. 12).

HTTP Status Code: 400

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 120). Amazon ECS services are cluster-specific and Region-specific.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task is not supported in this Region.

HTTP Status Code: 400

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
See Also

- 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 V3
DeleteAccountSetting

Disables an account setting for a specified IAM user, IAM role, or the root user for an account.

Request Syntax

```json
{
   "name": "string",
   "principalArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**name (p. 34)**

The resource name for which to disable the account setting. If `serviceLongArnFormat` is specified, the ARN for your Amazon ECS services is affected. If `taskLongArnFormat` is specified, the ARN and resource ID for your Amazon ECS tasks is affected. If `containerInstanceLongArnFormat` is specified, the ARN and resource ID for your Amazon ECS container instances is affected. If `awsvpcTrunking` is specified, the ENI limit for your Amazon ECS container instances is affected.

Type: String

Valid Values: `serviceLongArnFormat` | `taskLongArnFormat` | `containerInstanceLongArnFormat` | `awsvpcTrunking` | `containerInsights`

Required: Yes

**principalArn (p. 34)**

The ARN of the principal, which can be an IAM user, IAM role, or the root user. If you specify the root user, it disables the account setting for all IAM users, IAM roles, and the root user of the account unless an IAM user or role explicitly overrides these settings. If this field is omitted, the setting is changed only for the authenticated user.

Type: String

Required: No

Response Syntax

```json
{
   "setting": {
      "name": "string",
      "principalArn": "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.

setting (p. 34)

The account setting for the specified principal ARN.

Type: Setting (p. 342) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
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. 371).

The request accepts the following data in JSON format.

attributes (p. 36)

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. 253) objects

Required: Yes

cluster (p. 36)

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. 36)

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

Type: Array of Attribute (p. 253) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 113). 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. 116). 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
```
Content-Length: 169
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeleteAttributes
X-Amz-Date: 20161222T193851Z
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",
  "attributes": [
    {
      "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-
                     df30-47b4-8f1e-6e68ebd01f34",
      "name": "stack"
    }
  ]
}

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-1bd3d9928ca

{
  "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 V3
DeleteCluster

Deletes the specified cluster. The cluster will transition to the INACTIVE state. Clusters with an INACTIVE status may remain discoverable in your account for a period of time. However, this behavior is subject to change in the future, so you should not rely on INACTIVE clusters persisting.

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. 116) and deregister them with DeregisterContainerInstance (p. 54).

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 39)**

The short name or full Amazon Resource Name (ARN) of the cluster to delete.

Type: String

Required: Yes

Response Syntax

```json
{
    "cluster": {
        "activeServicesCount": number,
        "attachments": [
            {
                "details": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "id": "string",
                "status": "string",
                "type": "string"
            }
        ],
        "attachmentsStatus": "string",
        "capacityProviders": [ "string" ],
        "clusterArn": "string",
        "clusterName": "string",
        "defaultCapacityProviderStrategy": [ {
            "base": 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.

cluster (p. 39)

The full description of the deleted cluster.

Type: Cluster (p. 259) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. First, deregister the container instances before you can delete the cluster. For more information, see DeregisterContainerInstance (p. 54).

HTTP Status Code: 400
ClusterContainsServicesException
You cannot delete a cluster that contains services. First, update the service to reduce its desired task count to 0 and then delete the service. For more information, see UpdateService (p. 231) and DeleteService (p. 43).
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. 113). 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

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 request deletes the cluster called My-cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
```
Sample Response

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 V3
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. 231).

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 the ListServices (p. 120) API operation. After all tasks have transitioned to either STOPPING or STOPPED status, the service status moves from DRAINING to INACTIVE. Services in the DRAINING or INACTIVE status can still be viewed with the DescribeServices (p. 80) API operation. However, in the future, INACTIVE services may be cleaned up and purged from Amazon ECS record keeping, and DescribeServices (p. 80) calls on those services return a ServiceNotFoundException error.

Important
If you attempt to create a new service with the same name as an existing service in either ACTIVE or DRAINING status, you receive an error.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

cluster (p. 43)

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

force (p. 43)

If true, allows you to delete a service even if it has not been scaled down to zero tasks. It is only necessary to use this if the service is using the REPLICA scheduling strategy.

Type: Boolean

Required: No

service (p. 43)

The name of the service to delete.
Type: String

Required: Yes

Response Syntax

```json
{
  "service": {
    "capacityProviderStrategy": [
      {
        "base": number,
        "capacityProvider": "string",
        "weight": number
      }
    ],
    "clusterArn": "string",
    "createdAt": number,
    "createdBy": "string",
    "deploymentConfiguration": {
      "maximumPercent": number,
      "minimumHealthyPercent": number
    },
    "deploymentController": {
      "type": "string"
    },
    "deployments": [
      {
        "capacityProviderStrategy": [
          {
            "base": number,
            "capacityProvider": "string",
            "weight": number
          }
        ],
        "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,
    "enableECSManagedTags": boolean,
    "events": [
      {
        "createdAt": number,
        "id": "string",
        "message": "string"
      }
    ],
    "healthCheckGracePeriodSeconds": number,
    "healthCheckThresholdCount": number,
    "healthCheckType": "string",
    "inServiceCount": number,
    "launchType": "string",
    "registeredInstancesStatus": {
      "inServiceCount": number,
      "pendingDeletionCount": number,
      "pendingUpdateCount": number,
      "status": "string",
      "tasksInService": number
    },
    "runningCount": number,
    "updatedAt": number
  }
}
```

API Version 2014-11-13

44
"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",
"propagateTags": "string",
"roleArn": "string",
"runningCount": number,
"schedulingStrategy": "string",
"serviceArn": "string",
"serviceName": "string",
"serviceRegistries": [ 

"containerName": "string",
"containerPort": number,
"port": number,
"registryArn": "string"

],
"status": "string",
"tags": [ 

"key": "string",
"value": "string"

],
"taskDefinition": "string",
"taskSets": [ 

"capacityProviderStrategy": [ 

"base": number,
"capacityProvider": "string",
"weight": number

],
"clusterArn": "string",
"computedDesiredCount": number,
"createdAt": number,
"externalId": "string",
"id": "string"
"launchType": "string",
"loadBalancers": [
  {
    "containerName": "string",
    "containerPort": number,
    "loadBalancerName": "string",
    "targetGroupArn": "string"
  }
],
"networkConfiguration": {
  "awsvpcConfiguration": {
    "assignPublicIp": "string",
    "securityGroups": [ "string" ],
    "subnets": [ "string" ]
  }
},
"pendingCount": number,
"platformVersion": "string",
"runningCount": number,
"scale": {
  "unit": "string",
  "value": number
},
"serviceArn": "string",
"serviceRegistries": [
  {
    "containerName": "string",
    "containerPort": number,
    "port": number,
    "registryArn": "string"
  }
],
"stabilityStatus": "string",
"stabilityStatusAt": number,
"startedBy": "string",
"status": "string",
"tags": [
  {
    "key": "string",
    "value": "string"
  }
],
"taskDefinition": "string",
"taskSetArn": "string",
"updatedAt": 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.

**service (p. 44)**

The full description of the deleted service.

Type: Service (p. 334) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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. 120). 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

```plaintext
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/92337060653404511",
        "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"
  }
}

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 V3
DeleteTaskSet

Deletes a specified task set within a service. This is used when a service uses the EXTERNAL deployment controller type. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
    "cluster": "string",
    "force": boolean,
    "service": "string",
    "taskSet": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 50)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service that the task set exists in to delete.

Type: String

Required: Yes

**force (p. 50)**

If true, this allows you to delete a task set even if it hasn't been scaled down to zero.

Type: Boolean

Required: No

**service (p. 50)**

The short name or full Amazon Resource Name (ARN) of the service that hosts the task set to delete.

Type: String

Required: Yes

**taskSet (p. 50)**

The task set ID or full Amazon Resource Name (ARN) of the task set to delete.

Type: String

Required: Yes

Response Syntax

```json
{
}
```
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.

**taskSet (p. 50)**

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an `EXTERNAL` deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Type: TaskSet (p. 360) object

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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. 113). 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

**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. 12).

HTTP Status Code: 400

**ServiceNotFoundException**

The specified service could not be found. You can view your available services with ListServices (p. 120). Amazon ECS services are cluster-specific and Region-specific.

HTTP Status Code: 400

**TaskSetNotFoundException**

The specified task set could not be found. You can view your available task sets with DescribeTaskSets (p. 100). Task sets are specific to each cluster, service and Region.
HTTP Status Code: 400

**UnsupportedFeatureException**

The specified task is not supported in this Region.

HTTP Status Code: 400

### 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 V3
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 (p. 371).

The request accepts the following data in JSON format.

**cluster** (p. 54)

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. 54)

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. 54)

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"
      }
    ],
    "capacityProviderName": "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" ]
      }
    ]
  }
}
```
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. 55)

The container instance that was deregistered.

Type: ContainerInstance (p. 280) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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 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"
            }
        ]
    }
}
```
Amazon Elastic Container Service API Reference

Example

```
},
{
  "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-8805-9c57b9368fd0",
"ec2InstanceId": "i-0c3826c9",
"pendingTasksCount": 0,
"registeredResources": [
  {
    "doubleValue": 0,
    "integerValue": 1024,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 995,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "name": "PORTS",
    "stringSetValue": [
      "22",
      "2376",
      "2375",
      "51678"
    ],
    "type": "STRINGSET"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "name": "PORTS_UDP",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
],
"remainingResources": [
  {
    "doubleValue": 0,
    "integerValue": 1024,
    "name": "CPU",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 995,
    "name": "MEMORY",
    "type": "INTEGER"
  },
  {
    "doubleValue": 0,
    "integerValue": 0,
    "name": "PORTS",
    "stringSetValue": [],
    "type": "STRINGSET"
  }
],
```

API Version 2014-11-13

58
"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": "INACTIVE",
"versionInfo": {
  "agentHash": "b197edd",
  "agentVersion": "1.5.0",
  "dockerVersion": "DockerVersion: 1.7.1"
}
}
}

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 V3
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. However, 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**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**taskDefinition (p. 60)**

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**

```json
{
  "taskDefinition": {
    "compatibilities": [ "string" ],
    "containerDefinitions": [
      {
        "command": [ "string" ],
        "cpu": number,
        "dependsOn": [
          {
            "condition": "string",
            "containerName": "string"
          }
        ],
        "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"
  }
],
"firelensConfiguration": { 
  "options": { 
    "string": "string"
  },
  "type": "string"
},
"healthCheck": { 
  "command": [ "string" ],
  "interval": number,
  "retries": number,
  "startPeriod": number,
  "timeout": number
},
"hostname": "string",
"image": "string",
"interactive": boolean,
"links": [ "string" ],
"linuxParameters": { 
  "capabilities": { 
    "add": [ "string" ],
    "drop": [ "string" ]
  },
  "devices": [ 
    { 
      "containerPath": "string",
      "hostPath": "string",
      "permissions": [ "string" ]
    }
  ],
  "initProcessEnabled": boolean,
  "maxSwap": number,
  "sharedMemorySize": number,
  "swappiness": number,
  "tmpfs": [ 
    { 
      "containerPath": "string",
      "mountOptions": [ "string" ],
      "size": number
    }
  ]
},
"logConfiguration": { 
  "logDriver": "string",
  "options": { 
    "string": "string"
  },
  "secretOptions": [ 
    "string"
  ]
}
{  
  "name": "string",
  "valueFrom": "string"
}

"memory": number,
"memoryReservation": number,
"mountPoints": [
  {
    "containerPath": "string",
    "readOnly": boolean,
    "sourceVolume": "string"
  }
],

"name": "string",

"portMappings": [
  {
    "containerPort": number,
    "hostPort": number,
    "protocol": "string"
  }
],

"privileged": boolean,
"pseudoTerminal": boolean,
"readonlyRootFilesystem": boolean,
"repositoryCredentials": {
  "credentialsParameter": "string"
},

"resourceRequirements": [
  {
    "type": "string",
    "value": "string"
  }
],

"secrets": [
  {
    "name": "string",
    "valueFrom": "string"
  }
],

"startTimeout": number,
"stopTimeout": number,
"systemControls": [
  {
    "namespace": "string",
    "value": "string"
  }
],

"ulimits": [
  {
    "hardLimit": number,
    "name": "string",
    "softLimit": number
  }
],

"user": "string",
"volumesFrom": [
  {
    "readOnly": boolean,
    "sourceContainer": "string"
  }
],

"workingDirectory": "string"
}
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"inferenceAccelerators": [
  {
    "deviceName": "string",
    "deviceType": "string"
  }
],
"ipcMode": "string",
"memory": "string",
"networkMode": "string",
"pidMode": "string",
"placementConstraints": [
  {
    "expression": "string",
    "type": "string"
  }
],
"proxyConfiguration": {
  "containerName": "string",
  "properties": [
    {
      "name": "string",
      "value": "string"
    }
  ],
  "type": "string"
},
"requiresAttributes": [
  {
    "name": "string",
    "targetId": "string",
    "targetType": "string",
    "value": "string"
  }
],
"requiresCompatibilities": [ "string" ],
"revision": number,
"status": "string",
"taskDefinitionArn": "string",
"taskRoleArn": "string",
"volumes": [
  {
    "dockerVolumeConfiguration": {
      "autoprovision": boolean,
      "driver": "string",
      "driverOpts": {
        "string": "string"
      },
      "labels": {
        "string": "string"
      },
      "scope": "string"
    },
    "efsVolumeConfiguration": {
      "fileSystemId": "string",
      "rootDirectory": "string"
    },
    "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. 60)**

The full description of the deregistered task.

Type: TaskDefinition (p. 352) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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

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
```
**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

```json
{
  "taskDefinition": {
    "containerDefinitions": [ 
      {
        "command": [ "apt-get update; apt-get install stress; while true; do stress --cpu $(( RANDOM % 4 )); done"
        ],
        "cpu": 50,
        "entryPoint": [ "bash", 
        "-c"
        ],
        "environment": [],
        "essential": true,
        "image": "ubuntu",
        "memory": 100,
        "mountPoints": [],
        "name": "wave",
        "portMappings": [],
        "volumesFrom": []
      }
    ],
    "family": "cpu-wave",
    "revision": 1,
    "status": "INACTIVE",
    "taskDefinitionArn": "arn:aws:ecs:us-west-2:012345678910:task-definition/cpu-wave:1",
    "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
See Also

- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DescribeCapacityProviders

Describes one or more of your capacity providers.

Request Syntax

```
{
    "capacityProviders": [ "string" ],
    "include": [ "string" ],
    "maxResults": number,
    "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

capacityProviders (p. 67)

The short name or full Amazon Resource Name (ARN) of one or more capacity providers. Up to 100 capacity providers can be described in an action.

Type: Array of strings

Required: No

include (p. 67)

Specifies whether or not you want to see the resource tags for the capacity provider. If TAGS is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

Type: Array of strings

Valid Values: TAGS

Required: No

maxResults (p. 67)

The maximum number of account setting results returned by DescribeCapacityProviders in paginated output. When this parameter is used, DescribeCapacityProviders 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 DescribeCapacityProviders request with the returned nextToken value. This value can be between 1 and 10. If this parameter is not used, then DescribeCapacityProviders returns up to 10 results and a nextToken value if applicable.

Type: Integer

Required: No

nextToken (p. 67)

The nextToken value returned from a previous paginated DescribeCapacityProviders 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
{
  "capacityProviders": [
    {
      "autoScalingGroupProvider": {
        "autoScalingGroupArn": "string",
        "managedScaling": {
          "maximumScalingStepSize": number,
          "minimumScalingStepSize": number,
          "status": "string",
          "targetCapacity": number
        },
        "managedTerminationProtection": "string"
      },
      "capacityProviderArn": "string",
      "name": "string",
      "status": "string",
      "tags": [
        {
          "key": "string",
          "value": "string"
        }
      ]
    }
  ],
  "failures": [
    {
      "arn": "string",
      "detail": "string",
      "reason": "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.

**capacityProviders (p. 68)**

The list of capacity providers.

Type: Array of CapacityProvider (p. 256) objects

**failures (p. 68)**

Any failures associated with the call.

Type: Array of Failure (p. 298) objects
nextToken (p. 68)

The `nextToken` value to include in a future `DescribeCapacityProviders` request. When the results of a `DescribeCapacityProviders` 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. 373).

**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 V3
DescribeClusters

Describes one or more of your clusters.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**clusters (p. 70)**

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. 70)**

Whether to include additional information about your clusters in the response. If this field is omitted, the attachments, statistics, and tags are not included.

If ATTACHMENTS is specified, the attachments for the container instances or tasks within the cluster are included.

If SETTINGS is specified, the settings for the cluster are included.

If STATISTICS is specified, the following additional information, separated by launch type, is included:

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

If TAGS is specified, the metadata tags associated with the cluster are included.

Type: Array of strings

Valid Values: ATTACHMENTS | SETTINGS | STATISTICS | TAGS

Required: No
Response Syntax

```
{
  "clusters": [
    {
      "activeServicesCount": number,
      "attachments": [
        {
          "details": [
            {
              "name": "string",
              "value": "string"
            }
          ],
          "id": "string",
          "status": "string",
          "type": "string"
        }
      ],
      "attachmentsStatus": "string",
      "capacityProviders": [ "string" ],
      "clusterArn": "string",
      "clusterName": "string",
      "defaultCapacityProviderStrategy": [
        {
          "base": number,
          "capacityProvider": "string",
          "weight": number
        }
      ],
      "pendingTasksCount": number,
      "registeredContainerInstancesCount": number,
      "runningTasksCount": number,
      "settings": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "statistics": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "status": "string",
      "tags": [
        {
          "key": "string",
          "value": "string"
        }
      ]
    }
  ],
  "failures": [
    {
      "arn": "string",
      "detail": "string",
      "reason": "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. 71)

The list of clusters.

Type: Array of Cluster (p. 259) objects

failures (p. 71)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
DescribeContainerInstances

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

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

`cluster (p. 74)`

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. This parameter is required if the container instance or container instances you are describing were launched in any cluster other than the default cluster.

Type: String

Required: No

`containerInstances (p. 74)`

A list of up to 100 container instance IDs or full Amazon Resource Name (ARN) entries.

Type: Array of strings

Required: Yes

`include (p. 74)`

Specifies whether you want to see the resource tags for the container instance. If TAGS is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

Type: Array of strings

Valid Values: TAGS

Required: No

Response Syntax

```
{
  "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"
    }
],
"capacityProviderName": "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"
    }
],
"runningTasksCount": number,
"status": "string",
"statusReason": "string",
"tags": [
    {
        "key": "string",
        "value": "string"
    }
],
"version": number,
"versionInfo": {
    "agentHash": "string",
    "agentVersion": "string",
    "dockerVersion": "string"
}
],
"failures": [
    {
        "arn": "string",
        "description": "string",
        "id": "string",
        "reason": "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.

containerInstances (p. 74)

The list of container instances.

Type: Array of ContainerInstance (p. 280) objects

failures (p. 74)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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"
    },
    {
        "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": []
    }
]
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 V3
DescribeServices

Describes the specified services running in your cluster.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 80)**

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. This parameter is required if the service or services you are describing were launched in any cluster other than the default cluster.

Type: String

Required: No

**include (p. 80)**

Specifies whether you want to see the resource tags for the service. If TAGS is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

Type: Array of strings

Valid Values: TAGS

Required: No

**services (p. 80)**

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

```
{
    "failures": [
        {
            "arn": "string",
            "detail": "string",
            "reason": "string"
        }
    ],
    "services": [   ]
}
```
{
    "capacityProviderStrategy": [
        {
            "base": number,
            "capacityProvider": "string",
            "weight": number
        }
    ],
    "clusterArn": "string",
    "createdAt": number,
    "createdBy": "string",
    "deploymentConfiguration": {
        "maximumPercent": number,
        "minimumHealthyPercent": number
    },
    "deploymentController": {
        "type": "string"
    },
    "deployments": [
        {
            "capacityProviderStrategy": [
                {
                    "base": number,
                    "capacityProvider": "string",
                    "weight": number
                }
            ],
            "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,
    "enableECSManagedTags": boolean,
    "events": [
        {
            "createdAt": number,
            "id": "string",
            "message": "string"
        }
    ],
    "healthCheckGracePeriodSeconds": number,
    "launchType": "string",
    "loadBalancers": [
        {
            "containerName": "string",
            "containerPort": number,
            "loadBalancerName": "string",
            "targetGroupArn": "string"
        }
    ],
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
        }
    }
}
"awsvpcConfiguration": {
  "assignPublicIp": "string",
  "securityGroups": [ "string" ],
  "subnets": [ "string" ]
},
"pendingCount": number,
"placementConstraints": [ {
  "expression": "string",
  "type": "string"
} ],
"placementStrategy": [ {
  "field": "string",
  "type": "string"
} ],
"platformVersion": "string",
"propagateTags": "string",
"roleArn": "string",
"runningCount": number,
"schedulingStrategy": "string",
"serviceArn": "string",
"serviceName": "string",
"serviceRegistries": [ {
  "containerName": "string",
  "containerPort": number,
  "port": number,
  "registryArn": "string"
} ],
"status": "string",
"tags": [ {
  "key": "string",
  "value": "string"
} ],
"taskDefinition": "string",
"taskSets": [ {
  "capacityProviderStrategy": [ {
    "base": number,
    "capacityProvider": "string",
    "weight": number
  } ],
  "clusterArn": "string",
  "computedDesiredCount": number,
  "createdAt": number,
  "externalId": "string",
  "id": "string",
  "launchType": "string",
  "loadBalancers": [ {
    "containerName": "string",
    "containerPort": number,
    "loadBalancerName": "string",
    "targetGroupArn": "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.

failures (p. 80)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

services (p. 80)

The list of services described.

Type: Array of Service (p. 334) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).
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. 113). 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
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

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

```json
{
   "failures": [],
   "services": [
      {
         "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
         },
         "deployments": [
            {
               "createdAt": 1432829320.611,
               "desiredCount": 4,
               "id": "ecs-svc/9223370604025455196",
               "pendingCount": 0,
               "runningCount": 4,
               "status": "PRIMARY",
               "updatedAt": 1432829320.611
            }
         ],
         "desiredCount": 4,
         "events": [],
         "loadBalancers": [],
         "pendingCount": 0,
         "runningCount": 4,
         "serviceName": "bunker-buster",
         "status": "ACTIVE",
         "taskDefinition": "arn:aws:ecs:us-west-2:012345678910:task-definition/hpcc-t2-medium: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 V3

API Version 2014-11-13
DescribeTaskDefinition

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**

```json
{
   "include": [ "string" ],
   "taskDefinition": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**include (p. 87)**

Specifies whether to see the resource tags for the task definition. If TAGS is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

- **Type:** Array of strings
- **Valid Values:** TAGS
- **Required:** No

**taskDefinition (p. 87)**

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**

```json
{
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": {
      "compatibilities": [ "string" ],
      "containerDefinitions": [
         {
            "command": [ "string" ],
            "..."...
         }
      ]
   }
}
```
"cpu": number,
"dependsOn": [ 
  { 
    "condition": "string",
    "containerName": "string"
  }
],
"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"
  }
],
"firelensConfiguration": { 
  "options": { 
    "string": "string"
  },
  "type": "string"
},

"healthCheck": { 
  "command": [ "string" ],
  "interval": number,
  "retries": number,
  "startPeriod": number,
  "timeout": number
},

"hostname": "string",
"image": "string",
"interactive": boolean,
"links": [ "string" ],
"linuxParameters": { 
  "capabilities": { 
    "add": [ "string" ],
    "drop": [ "string" ]
  },
  "devices": [ 
    { 
      "containerPath": "string",
      "hostPath": "string",
      "permissions": [ "string" ]
    }
  ],
  "initProcessEnabled": boolean,
  "maxSwap": number,
  "sharedMemorySize": number,
  "swappiness": number,
  "tmpfs": [ 
    { 
      "containerPath": "string",
      "mountOptions": [ "string" ],
      "size": number
    }
  ]
}
Amazon Elastic Container Service API Reference

Response Syntax

```json
{
    "logConfiguration": {
      "logDriver": "string",
      "options": {
        "string": "string"
      },
      "secretOptions": [
        {
          "name": "string",
          "valueFrom": "string"
        }
      ],
      "memory": number,
      "memoryReservation": number,
      "mountPoints": [
        {
          "containerPath": "string",
          "readOnly": boolean,
          "sourceVolume": "string"
        }
      ],
      "name": "string",
      "portMappings": [
        {
          "containerPort": number,
          "hostPort": number,
          "protocol": "string"
        }
      ],
      "privileged": boolean,
      "pseudoTerminal": boolean,
      "readonlyRootFilesystem": boolean,
      "repositoryCredentials": {
        "credentialsParameter": "string"
      },
      "resourceRequirements": [
        {
          "type": "string",
          "value": "string"
        }
      ],
      "secrets": [
        {
          "name": "string",
          "valueFrom": "string"
        }
      ],
      "startTimeout": number,
      "stopTimeout": number,
      "systemControls": [
        {
          "namespace": "string",
          "value": "string"
        }
      ],
      "ulimits": [
        {
          "hardLimit": number,
          "name": "string",
          "softLimit": number
        }
      ],
      "user": "string",
    }
  }
```

API Version 2014-11-13
"volumesFrom": [
    {
        "readOnly": boolean,
        "sourceContainer": "string"
    }
],
"workingDirectory": "string"
],
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"inferenceAccelerators": [
    {
        "deviceName": "string",
        "deviceType": "string"
    }
],
"ipcMode": "string",
"memory": "string",
"networkMode": "string",
"pidMode": "string",
"placementConstraints": [
    {
        "expression": "string",
        "type": "string"
    }
],
"proxyConfiguration": {
    "containerName": "string",
    "properties": [
        {
            "name": "string",
            "value": "string"
        }
    ],
    "type": "string"
},
"requiresAttributes": [
    {
        "name": "string",
        "targetId": "string",
        "targetType": "string",
        "value": "string"
    }
],
"requiresCompatibilities": [ "string" ],
"revision": number,
"status": "string",
"taskDefinitionArn": "string",
"taskRoleArn": "string",
"volumes": [
    {
        "dockerVolumeConfiguration": {
            "autoprovision": boolean,
            "driver": "string",
            "driverOpts": {
                "string": "string"
            },
            "labels": {
                "string": "string"
            },
            "scope": "string"
        },
        "efsVolumeConfiguration": {
            "fileSystemId": "string",
            "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.

tags (p. 87)

The metadata that is applied to the task definition to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

taskDefinition (p. 87)

The full task definition description.

Type: TaskDefinition (p. 352) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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

```plaintext
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

```plaintext
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 V3
DescribeTasks

Describes a specified task or tasks.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 94)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task or tasks to describe. If you do not specify a cluster, the default cluster is assumed. This parameter is required if the task or tasks you are describing were launched in any cluster other than the default cluster.

Type: String

Required: No

**include (p. 94)**

Specifies whether you want to see the resource tags for the task. If **TAGS** is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

Type: Array of strings

Valid Values: **TAGS**

Required: No

**tasks (p. 94)**

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

Type: Array of strings

Required: Yes

Response Syntax

```json
{
   "failures": [
   
   {  
      "arn": "string",
      "detail": "string",
      "reason": "string"
   }
   ]
}
```
{  
  "tasks": [  
    {  
      "attachments": [  
        {  
          "details": [  
            {  
              "name": "string",  
              "value": "string"  
            }  
          ],  
          "id": "string",  
          "status": "string",  
          "type": "string"  
        }  
      ],  
      "attributes": [  
        {  
          "name": "string",  
          "targetId": "string",  
          "targetType": "string",  
          "value": "string"  
        }  
      ],  
      "availabilityZone": "string",  
      "capacityProviderName": "string",  
      "clusterArn": "string",  
      "connectivity": "string",  
      "connectivityAt": number,  
      "containerInstanceArn": "string",  
      "containers": [  
        {  
          "containerArn": "string",  
          "cpu": "string",  
          "exitCode": number,  
          "gpuIds": [ "string" ],  
          "healthStatus": "string",  
          "image": "string",  
          "imageDigest": "string",  
          "lastStatus": "string",  
          "memory": "string",  
          "memoryReservation": "string",  
          "name": "string",  
          "networkBindings": [  
            {  
              "bindIP": "string",  
              "containerPort": number,  
              "hostPort": number,  
              "protocol": "string"  
            }  
          ],  
          "networkInterfaces": [  
            {  
              "attachmentId": "string",  
              "ipv6Address": "string",  
              "privateIpv4Address": "string"  
            }  
          ],  
          "reason": "string",  
          "runtimeId": "string",  
          "taskArn": "string"  
        }  
      ],  
      "cpu": "string",  
      "createdAt": number,  
    }  
  ]  
}
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"healthStatus": "string",
"inferenceAccelerators": [
  {
    "deviceName": "string",
    "deviceType": "string"
  }
],
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
  "containerOverrides": [
    {
      "command": [ "string" ],
      "cpu": number,
      "environment": [
        {
          "name": "string",
          "value": "string"
        }
      ],
      "memory": number,
      "memoryReservation": number,
      "name": "string",
      "resourceRequirements": [
        {
          "type": "string",
          "value": "string"
        }
      ]
    }
  ],
  "cpu": "string",
  "executionRoleArn": "string",
  "inferenceAcceleratorOverrides": [
    {
      "deviceName": "string",
      "deviceType": "string"
    }
  ],
  "memory": "string",
  "taskRoleArn": "string"
},
"platformVersion": "string",
"pullStartedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "string",
"stopCode": "string",
"stoppedAt": number,
"stoppedReason": "string",
"stoppingAt": number,
"tags": [
  {
    "key": "string",
    "value": "string"
  }
],
"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. 94)
- Any failures associated with the call.
  - Type: Array of Failure (p. 298) objects

tasks (p. 94)
- The list of tasks.
  - Type: Array of Task (p. 346) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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

```plaintext
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

```plaintext
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": 1476822833.998,
      "desiredStatus": "RUNNING",
      "startedBy": "ecs-svc/9223370560032507596",
      "containers": [
```
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 V3
DescribeTaskSets

Describes the task sets in the specified cluster and service. This is used when a service uses the external deployment controller type. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```
{
   "cluster": "string",
   "include": [ "string" ],
   "service": "string",
   "taskSets": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

cluster (p. 100)

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service that the task sets exist in.

Type: String

Required: Yes

include (p. 100)

Specifies whether to see the resource tags for the task set. If TAGS is specified, the tags are included in the response. If this field is omitted, tags are not included in the response.

Type: Array of strings

Valid Values: TAGS

Required: No

service (p. 100)

The short name or full Amazon Resource Name (ARN) of the service that the task sets exist in.

Type: String

Required: Yes

taskSets (p. 100)

The ID or full Amazon Resource Name (ARN) of task sets to describe.

Type: Array of strings

Required: Yes
Response Syntax

```json
{
  "failures": [
    {
      "arn": "string",
      "detail": "string",
      "reason": "string"
    }
  ],
  "taskSets": [
    {
      "capacityProviderStrategy": [
        {
          "base": number,
          "capacityProvider": "string",
          "weight": number
        }
      ],
      "clusterArn": "string",
      "computedDesiredCount": number,
      "createdAt": number,
      "externalId": "string",
      "id": "string",
      "launchType": "string",
      "loadBalancers": [
        {
          "containerName": "string",
          "containerPort": number,
          "loadBalancerName": "string",
          "targetGroupArn": "string"
        }
      ],
      "networkConfiguration": {
        "awsvpcConfiguration": {
          "assignPublicIp": "string",
          "securityGroups": [ "string" ],
          "subnets": [ "string" ]
        }
      },
      "pendingCount": number,
      "platformVersion": "string",
      "runningCount": number,
      "scale": {
        "unit": "string",
        "value": number
      },
      "serviceArn": "string",
      "serviceRegistries": [
        {
          "containerName": "string",
          "containerPort": number,
          "port": number,
          "registryArn": "string"
        }
      ],
      "stabilityStatus": "string",
      "stabilityStatusAt": number,
      "startedBy": "string",
      "status": "string",
      "tags": [
        {
          "key": "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.

failures (p. 101)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

taskSets (p. 101)

The list of task sets described.

Type: Array of TaskSet (p. 360) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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
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. 12).

HTTP Status Code: 400

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 120). Amazon ECS services are cluster-specific and Region-specific.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task is not supported in this Region.

HTTP Status Code: 400

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 V3
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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 104)**

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

Type: String

Required: No

**containerInstance (p. 104)**

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

```
{
    "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. 104)

The endpoint for the Amazon ECS agent to poll.

Type: String
telemetryEndpoint (p. 104)

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. 373).

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 V3
ListAccountSettings

Lists the account settings for a specified principal.

**Request Syntax**

```json
{
  "effectiveSettings": boolean,
  "maxResults": number,
  "name": "string",
  "nextToken": "string",
  "principalArn": "string",
  "value": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**effectiveSettings (p. 106)**

Specifies whether to return the effective settings. If `true`, the account settings for the root user or the default setting for the `principalArn` are returned. If `false`, the account settings for the `principalArn` are returned if they are set. Otherwise, no account settings are returned.

Type: Boolean

Required: No

**maxResults (p. 106)**

The maximum number of account setting results returned by ListAccountSettings in paginated output. When this parameter is used, ListAccountSettings 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 ListAccountSettings request with the returned `nextToken` value. This value can be between 1 and 10. If this parameter is not used, then ListAccountSettings returns up to 10 results and a `nextToken` value if applicable.

Type: Integer

Required: No

**name (p. 106)**

The resource name you want to list the account settings for.

Type: String

Valid Values: serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights

Required: No

**nextToken (p. 106)**

The `nextToken` value returned from a ListAccountSettings request indicating that more results are available to fulfill the request and further calls will be needed. If `maxResults` was provided, it is possible the number of results to be fewer than `maxResults`. 
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

principalArn (p. 106)

The ARN of the principal, which can be an IAM user, IAM role, or the root user. If this field is omitted, the account settings are listed only for the authenticated user.

Type: String
Required: No

value (p. 106)

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

Type: String
Required: No

Response Syntax

```json
{
  "nextToken": "string",
  "settings": [
    {
      "name": "string",
      "principalArn": "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.

nextToken (p. 107)

The nextToken value to include in a future ListAccountSettings request. When the results of a ListAccountSettings 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

settings (p. 107)

The account settings for the resource.

Type: Array of Setting (p. 342) objects
Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
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. 371).

The request accepts the following data in JSON format.

**attributeName (p. 109)**

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

Type: String

Required: No

**attributeValue (p. 109)**

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. 109)**

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. 109)**

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.
Type: Integer
Required: No

nextToken (p. 109)

The `nextToken` value returned from a `ListAttributes` request indicating that more results are available to fulfill the request and further calls will be needed. If `maxResults` was provided, it is possible the number of results to be fewer than `maxResults`.

**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. 109)

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. 110)

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

Type: Array of Attribute (p. 253) objects

nextToken (p. 110)

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. 373).

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 113). 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 V3
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. 371).

The request accepts the following data in JSON format.

maxResults (p. 113)

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. 113)

The nextToken value returned from a ListClusters request indicating that more results are available to fulfill the request and further calls will be needed. If maxResults was provided, it is possible the number of results to be fewer than maxResults.

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. 113)**

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

Type: Array of strings

**nextToken (p. 113)**

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. 373)].

**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](../aws-general-ref) in the [AWS General Reference](../aws-general-ref).

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**

This example request lists the clusters for your account.

**Sample Request**

```
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-RequestId: 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 V3
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

```json
{
   "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. 371).

cluster (p. 116)

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. 116)

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. 116)

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. 116)***

The `nextToken` value returned from a `ListContainerInstances` request indicating that more results are available to fulfill the request and further calls will be needed. If `maxResults` was provided, it is possible the number of results to be fewer than `maxResults`.

**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. 116)**

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. 223)`. If you do not specify this parameter, the default is to include container instances set to all states other than `INACTIVE`.

Type: String

Valid Values:

- `ACTIVE` | `DRAINING` | `REGISTERING` | `DEREGISTERING` | `REGISTRATION_FAILED`

Required: No

---

**Response Syntax**

```
{
  "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. 117)**

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

Type: Array of strings

**nextToken (p. 117)**

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. 373).
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. 113). 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 V3
ListServices

Lists the services that are running in a specified cluster.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 120)**

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. 120)**

The launch type for the services to list.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**maxResults (p. 120)**

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 100. 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. 120)**

The nextToken value returned from a ListServices request indicating that more results are available to fulfill the request and further calls will be needed. If maxResults was provided, it is possible the number of results to be fewer than maxResults.
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

schedulingStrategy (p. 120)
The scheduling strategy for services to list.

Type: String
Valid Values: REPLICA | DAEMON
Required: No

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. 121)
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. 121)
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. 373).

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. 113). 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

```
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

```
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 V3
ListTagsForResource

List the tags for an Amazon ECS resource.

Request Syntax

```
{
  "resourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

`resourceArn (p. 124)`

The Amazon Resource Name (ARN) that identifies the resource for which to list the tags. Currently, the supported resources are Amazon ECS tasks, services, task definitions, clusters, and container instances.

Type: String

Required: Yes

Response Syntax

```
{
  "tags": [
    {
      "key": "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.

`tags (p. 124)`

The tags for the resource.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).
**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. 113). 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 lists the tags for the `dev` cluster.

**Sample Request**

```plaintext
POST / HTTP/1.1
Host: madison.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTagsForResource
Content-Type: application/x-amz-json-1.1
X-Amz-Date: 20181026T195430Z
Authorization: AUTHPARAMS
Content-Length: 72

{
   "resourceArn": "arn:aws:ecs:us-west-2:012345678910:cluster/dev"
}
```

**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 V3
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. 371).

The request accepts the following data in JSON format.

**familyPrefix (p. 127)**

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. 127)**

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. 127)**

The nextToken value returned from a ListTaskDefinitionFamilies request indicating that more results are available to fulfill the request and further calls will be needed. If maxResults was provided, it is possible the number of results to be fewer than maxResults.

- 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. 127)

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

Required: No

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. 128)

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

Type: Array of strings

nextToken (p. 128)

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. 373).

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 (\texttt{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

```plaintext
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

```plaintext
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
See Also

- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
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**

```
{
    "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. 371).

The request accepts the following data in JSON format.

**familyPrefix (p. 132)**

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. 132)**

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. 132)**

The `nextToken` value returned from a `ListTaskDefinitions` request indicating that more results are available to fulfill the request and further calls will be needed. If `maxResults` was provided, it is possible the number of results to be fewer than `maxResults`.

Type: String  
Required: No
sort (p. 132)

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. 132)

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. 133)

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. 133)

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. 373).
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 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 V3

```json
{
    "taskDefinitionArns": [
        "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:2",
        "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:3",
        "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:5",
        "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:8",
        "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:10"
    ]
}
```
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

```json
{
    "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. 371).

The request accepts the following data in JSON format.

cluster (p. 136)

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. 136)

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. 136)

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. This 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. Amazon ECS never sets the desired status of a task to that value (only a task's lastStatus may have a value of PENDING).
## Request Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Valid Values</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>family</td>
<td>String</td>
<td>RUNNING</td>
<td>PENDING</td>
<td>STOPPED</td>
</tr>
<tr>
<td>launchType</td>
<td>String</td>
<td>EC2</td>
<td>FARGATE</td>
<td></td>
</tr>
<tr>
<td>maxResults</td>
<td>Integer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nextToken</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>serviceName</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>startedBy</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Response Syntax

```json
{
    "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. 138)**

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. 138)**

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. 373).

**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. 113)`. 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. 120). 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

```plaintext
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

```plaintext
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-1:0123456789:task/0b69d5c0-d655-4695-98cd-5d2d526d95a",
    "arn:aws:ecs:us-east-1:0123456789:task/51a01bdf-d00e-487e-ab14-7645330b6207",
    "arn:aws:ecs:us-east-1:0123456789:task/b0b28bb8-2be3-4810-b52b-88df129d893c",
    "arn:aws:ecs:us-east-1:0123456789: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:
• 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 V3
PutAccountSetting

Modifies an account setting. Account settings are set on a per-Region basis.

If you change the account setting for the root user, the default settings for all of the IAM users and roles for which no individual account setting has been specified are reset. For more information, see Account Settings in the Amazon Elastic Container Service Developer Guide.

When serviceLongArnFormat, taskLongArnFormat, or containerInstanceLongArnFormat are specified, the Amazon Resource Name (ARN) and resource ID format of the resource type for a specified IAM user, IAM role, or the root user for an account is affected. The opt-in and opt-out account setting must be set for each Amazon ECS resource separately. The ARN and resource ID format of a resource will be defined by the opt-in status of the IAM user or role that created the resource. You must enable this setting to use Amazon ECS features such as resource tagging.

When awsvpcTrunking is specified, the elastic network interface (ENI) limit for any new container instances that support the feature is changed. If awsvpcTrunking is enabled, any new container instances that support the feature are launched have the increased ENI limits available to them. For more information, see Elastic Network Interface Trunking in the Amazon Elastic Container Service Developer Guide.

When containerInsights is specified, the default setting indicating whether CloudWatch Container Insights is enabled for your clusters is changed. If containerInsights is enabled, any new clusters that are created will have Container Insights enabled unless you disable it during cluster creation. For more information, see CloudWatch Container Insights in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```
{
  "name": "string",
  "principalArn": "string",
  "value": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

name (p. 141)

The Amazon ECS resource name for which to modify the account setting. If serviceLongArnFormat is specified, the ARN for your Amazon ECS services is affected. If taskLongArnFormat is specified, the ARN and resource ID for your Amazon ECS tasks is affected. If containerInstanceLongArnFormat is specified, the ARN and resource ID for your Amazon ECS container instances is affected. If awsvpcTrunking is specified, the elastic network interface (ENI) limit for your Amazon ECS container instances is affected. If containerInsights is specified, the default setting for CloudWatch Container Insights for your clusters is affected.

Type: String

Valid Values: serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights
Required: Yes

principalArn (p. 141)

The ARN of the principal, which can be an IAM user, IAM role, or the root user. If you specify the root user, it modifies the account setting for all IAM users, IAM roles, and the root user of the account unless an IAM user or role explicitly overrides these settings. If this field is omitted, the setting is changed only for the authenticated user.

Type: String

Required: No

value (p. 141)

The account setting value for the specified principal ARN. Accepted values are enabled and disabled.

Type: String

Required: Yes

Response Syntax

```
{
   "setting": {
      "name": "string",
      "principalArn": "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.

setting (p. 142)

The current account setting for a resource.

Type: Setting (p. 342) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
PutAccountSettingDefault

Modifies an account setting for all IAM users on an account for whom no individual account setting has been specified. Account settings are set on a per-Region basis.

Request Syntax

```
{
    "name": "string",
    "value": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

name (p. 144)

The resource name for which to modify the account setting. If serviceLongArnFormat is specified, the ARN for your Amazon ECS services is affected. If taskLongArnFormat is specified, the ARN and resource ID for your Amazon ECS tasks is affected. If containerInstanceLongArnFormat is specified, the ARN and resource ID for your Amazon ECS container instances is affected. If awsvpcTrunking is specified, the ENI limit for your Amazon ECS container instances is affected. If containerInsights is specified, the default setting for CloudWatch Container Insights for your clusters is affected.

Type: String

Valid Values: serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights

Required: Yes

value (p. 144)

The account setting value for the specified principal ARN. Accepted values are enabled and disabled.

Type: String

Required: Yes

Response Syntax

```
{
    "setting": {
        "name": "string",
        "principalArn": "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.

setting (p. 144)

The current account setting for a resource.

Type: Setting (p. 342) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
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. 36). For more information, see Attributes in the Amazon Elastic Container Service Developer Guide.

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. 371).

The request accepts the following data in JSON format.

attributes (p. 146)

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. 253) objects

Required: Yes

cluster (p. 146)

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

```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. 146)

The attributes applied to your resource.

Type: Array of Attribute (p. 253) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

AttributeLimitExceededException

You can apply up to 10 custom attributes per resource. You can view the attributes of a resource with ListAttributes (p. 109). You can remove existing attributes on a resource with DeleteAttributes (p. 36).

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 113). 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. 116). 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 V3
PutClusterCapacityProviders

Modifies the available capacity providers and the default capacity provider strategy for a cluster.

You must specify both the available capacity providers and a default capacity provider strategy for the cluster. If the specified cluster has existing capacity providers associated with it, you must specify all existing capacity providers in addition to any new ones you want to add. Any existing capacity providers associated with a cluster that are omitted from a PutClusterCapacityProviders API call will be disassociated with the cluster. You can only disassociate an existing capacity provider from a cluster if it's not being used by any existing tasks.

When creating a service or running a task on a cluster, if no capacity provider or launch type is specified, then the cluster's default capacity provider strategy is used. It is recommended to define a default capacity provider strategy for your cluster, however you may specify an empty array ([]) to bypass defining a default strategy.

Request Syntax

```
{  
    "capacityProviders": [ "string" ],  
    "cluster": "string",  
    "defaultCapacityProviderStrategy": [  
        {  
            "base": number,  
            "capacityProvider": "string",  
            "weight": number  
        }  
    ]  
}
```

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.

capacityProviders (p. 150)

The name of one or more capacity providers to associate with the cluster.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the CreateCapacityProvider API operation.

To use a AWS Fargate capacity provider, specify either the FARGATE or FARGATE_SPOT capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

Type: Array of strings

Required: Yes

cluster (p. 150)

The short name or full Amazon Resource Name (ARN) of the cluster to modify the capacity provider settings for. If you do not specify a cluster, the default cluster is assumed.
Type: String
Required: Yes

**defaultCapacityProviderStrategy (p. 150)**

The capacity provider strategy to use by default for the cluster.

When creating a service or running a task on a cluster, if no capacity provider or launch type is specified then the default capacity provider strategy for the cluster is used.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The `PutClusterCapacityProviders (p. 150)` API is used to associate a capacity provider with a cluster. Only capacity providers with an `ACTIVE` or `UPDATING` status can be used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the `CreateCapacityProvider (p. 4)` API operation.

To use an AWS Fargate capacity provider, specify either the `FARGATE` or `FARGATE_SPOT` capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

**Response Syntax**

```json
{
    "cluster": {
        "activeServicesCount": number,
        "attachments": [
            {
                "details": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "id": "string",
                "status": "string",
                "type": "string"
            }
        ],
        "attachmentsStatus": "string",
        "capacityProviders": [ "string" ],
        "clusterArn": "string",
        "clusterName": "string",
        "defaultCapacityProviderStrategy": [
            {
                "base": number,
                "capacityProvider": "string",
                "weight": number
            }
        ],
        "pendingTasksCount": number,
        "registeredContainerInstancesCount": number,
        "runningTasksCount": 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.

cluster (p. 151)

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.

Type: Cluster (p. 259) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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
ResourceInUseException

The specified resource is in-use and cannot be removed.

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

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 V3
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",
  "platformDevices": [
    {
      "id": "string",
      "type": "string"
    }
  ],
  "tags": [
    {
      "key": "string",
      "value": "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. 371).

The request accepts the following data in JSON format.
attributes (p. 154)

The container instance attributes that this container instance supports.

Type: Array of Attribute (p. 253) objects

Required: No

circle (p. 154)

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

circleInstanceArn (p. 154)

The ARN of the container instance (if it was previously registered).

Type: String

Required: No

instanceIdentityDocument (p. 154)

The instance identity document for the EC2 instance to register. This document can be found by running the following command from the instance: curl http://169.254.169.254/latest/dynamic/instance-identity/document/

Type: String

Required: No

instanceIdentityDocumentSignature (p. 154)

The instance identity document signature for the EC2 instance to register. This signature can be found by running the following command from the instance: curl http://169.254.169.254/latest/dynamic/instance-identity/signature/

Type: String

Required: No

platformDevices (p. 154)

The devices that are available on the container instance. The only supported device type is a GPU.

Type: Array of PlatformDevice (p. 323) objects

Required: No

tags (p. 154)

The metadata that you apply to the container instance to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:
- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.

Tag keys and values are case-sensitive.

Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

**Type:** Array of [Tag](p. 344) objects

**Array Members:** Minimum number of 0 items. Maximum number of 50 items.

**Required:** No

**totalResources (p. 154)**

The resources available on the instance.

**Type:** Array of [Resource](p. 329) objects

**Required:** No

**versionInfo (p. 154)**

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

**Type:** [VersionInfo](p. 367) object

**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"
      }
    ],
    "capacityProviderName": "string",
    "containerInstanceArn": "string",
    "ec2InstanceId": "string"
  }
}
```

---

API Version 2014-11-13

156
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. 156)

The container instance that was registered.

Type: ContainerInstance (p. 280) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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 V3
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. 319) 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,
            "dependsOn": [ {
                        "condition": "string",
                        "containerName": "string"
                    } ],
            "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" } ],
            "firelensConfiguration": { "options": { "string": "string" }, "type": "string" },
            "healthCheck": [ { "command": [ "string" ] } ]
        }
    ]
}
```

API Version 2014-11-13

159
"interval": number,
"retries": number,
"startPeriod": number,
"timeout": number
},
"hostname": "string",
"image": "string",
"interactive": boolean,
"links": [ "string" ],
"linuxParameters": {
  "capabilities": {
    "add": [ "string" ],
    "drop": [ "string" ]
  },
  "devices": [
    { "containerPath": "string",
      "hostPath": "string",
      "permissions": [ "string" ]
  ],
  "initProcessEnabled": boolean,
  "maxSwap": number,
  "sharedMemorySize": number,
  "swappiness": number,
  "tmpfs": [
    { "containerPath": "string",
      "mountOptions": [ "string" ],
      "size": number
    }
  ]
},
"logConfiguration": {
  "logDriver": "string",
  "options": {
    "string": "string"
  },
  "secretOptions": [
    { "name": "string",
      "valueFrom": "string"
    }
  ]
},
"memory": number,
"memoryReservation": number,
"mountPoints": [
  { "containerPath": "string",
    "readOnly": boolean,
    "sourceVolume": "string"
  }
],
"name": "string",
"portMappings": [
  { "containerPort": number,
    "hostPort": number,
    "protocol": "string"
  }
],
"privileged": boolean,
"pseudoTerminal": boolean,
"readonlyRootFilesystem": boolean,
"repositoryCredentials": {
"credentialsParameter": "string",
],
"resourceRequirements": [ 
{ 
  "type": "string",
  "value": "string"
 }
],
"secrets": [ 
{ 
  "name": "string",
  "valueFrom": "string"
 }
],
"startTimeout": number,
"stopTimeout": number,
"systemControls": [ 
{ 
  "namespace": "string",
  "value": "string"
 }
],
"ulimits": [ 
{ 
  "hardLimit": number,
  "name": "string",
  "softLimit": number
 }
],
"user": "string",
"volumesFrom": [ 
{ 
  "readOnly": boolean,
  "sourceContainer": "string"
 }
],
"workingDirectory": "string"
],
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"inferenceAccelerators": [ 
{ 
  "deviceName": "string",
  "deviceType": "string"
 }
],
"ipcMode": "string",
"memory": "string",
"networkMode": "string",
"pidMode": "string",
"placementConstraints": [ 
{ 
  "expression": "string",
  "type": "string"
 }
],
"proxyConfiguration": { 
  "containerName": "string",
  "properties": [ 
  { 
    "name": "string",
    "value": "string"
  }
  ],
}
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

containerDefinitions (p. 159)

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

Type: Array of ContainerDefinition (p. 267) objects

Required: Yes

cpu (p. 159)

The number of CPU units used by the task. It can be expressed as an integer using CPU units, for example 1024, or as a string using vCPUs, for example 1 vCPU or 1 vcpu, in a task definition. String values are converted to an integer indicating the CPU units when the task definition is registered.

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 EC2 launch type, this field is optional. Supported values are between 128 CPU units (0.125 vCPUs) and 10240 CPU units (10 vCPUs).
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 supported values for the memory parameter:

- **256 (0.25 vCPU)** - Available memory values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
- **512 (0.5 vCPU)** - Available memory values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
- **1024 (1 vCPU)** - Available memory values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
- **2048 (2 vCPU)** - Available memory values: Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
- **4096 (4 vCPU)** - Available memory values: Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)

Type: String

Required: No

e**xecutionRoleArn (p. 159)**

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. 159)**

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, and hyphens are allowed.

Type: String

Required: Yes

**inferenceAccelerators (p. 159)**

The Elastic Inference accelerators to use for the containers in the task.

Type: Array of **InferenceAccelerator (p. 304)** objects

Required: No

**ipcMode (p. 159)**

The IPC resource namespace to use for the containers in the task. The valid values are host, task, or none. If host is specified, then all containers within the tasks that specified the host IPC mode on the same container instance share the same IPC resources with the host Amazon EC2 instance. If task is specified, all containers within the specified task share the same IPC resources. If none is specified, then IPC resources within the containers of a task are private and not shared with other containers in a task or on the container instance. If no value is specified, then the IPC resource namespace sharing depends on the Docker daemon setting on the container instance. For more information, see IPC settings in the Docker run reference.

If the host IPC mode is used, be aware that there is a heightened risk of undesired IPC namespace expose. For more information, see Docker security.

If you are setting namespaced kernel parameters using systemControls for the containers in the task, the following will apply to your IPC resource namespace. For more information, see System Controls in the Amazon Elastic Container Service Developer Guide.

- For tasks that use the host IPC mode, IPC namespace related systemControls are not supported.
- For tasks that use the task IPC mode, IPC namespace related systemControls will apply to all containers within a task.
Note
This parameter is not supported for Windows containers or tasks using the Fargate launch type.

Type: String

Valid Values: host | task | none

Required: No

memory (p. 159)
The amount of memory (in MiB) used by the task. It can be expressed as an integer using MiB, for example 1024, or as a string using GB, for example 1GB or 1 GB, in a task definition. String values are converted to an integer indicating the MiB when the task definition is registered.

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

If using the EC2 launch type, this field is optional.

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

- 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available cpu values: 256 (.25 vCPU)
- 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available cpu values: 512 (.5 vCPU)
- 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available cpu values: 1024 (1 vCPU)
- Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB) - Available cpu values: 2048 (2 vCPU)
- Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB) - Available cpu values: 4096 (4 vCPU)

Type: String

Required: No

networkMode (p. 159)
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 you are using the Fargate launch type, the awsvpc network mode is required. If you are using the EC2 launch type, any network mode can be used. If the network mode is set to none, you cannot specify port mappings in your container definitions, and the tasks 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. 319) value 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 Amazon ECS-optimized AMIs, 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 cannot 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

**pidMode (p. 159)**

The process namespace to use for the containers in the task. The valid values are host or task. If host is specified, then all containers within the tasks that specified the host PID mode on the same container instance share the same process namespace with the host Amazon EC2 instance. If task is specified, all containers within the specified task share the same process namespace. If no value is specified, the default is a private namespace. For more information, see PID settings in the Docker run reference.

If the host PID mode is used, be aware that there is a heightened risk of undesired process namespace expose. For more information, see Docker security.

Note
This parameter is not supported for Windows containers or tasks using the Fargate launch type.

Type: String

Valid Values: host | task

Required: No

**placementConstraints (p. 159)**

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 runtime).

Type: Array of TaskDefinitionPlacementConstraint (p. 357) objects

Required: No

**proxyConfiguration (p. 159)**

The configuration details for the App Mesh proxy.

For tasks using the EC2 launch type, the container instances require at least version 1.26.0 of the container agent and at least version 1.26.0-1 of the ecs-init package to enable a proxy configuration. If your container instances are launched from the Amazon ECS-optimized AMI version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

For tasks using the Fargate launch type, the task or service requires platform version 1.3.0 or later.

Type: ProxyConfiguration (p. 326) object

Required: No

**requiresCompatibilities (p. 159)**

The launch type required by the task. If no value is specified, it defaults to EC2.
Type: Array of strings

Valid Values: EC2 | FARGATE

Required: No
tags (p. 159)
The metadata that you apply to the task definition to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:
- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No
taskRoleArn (p. 159)
The short name or full 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. For more information, see IAM Roles for Tasks in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No
volumes (p. 159)
A list of volume definitions in JSON format that containers in your task may use.

Type: Array of Volume (p. 368) objects

Required: No

Response Syntax

```json
{
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "taskDefinition": {
        "compatibilities": [ "string" ],
        "containerDefinitions": [
            {
```
"command": [ "string" ],
"cpu": number,
"dependsOn": [ {
  "condition": "string",
  "containerName": "string"
},
"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"
} ],
"firelensConfiguration": {
  "options": { "string": "string" },
  "type": "string"
},
"healthCheck": { "command": [ "string" ],
  "interval": number,
  "retries": number,
  "startPeriod": number,
  "timeout": number
},
"hostname": "string",
"image": "string",
"interactive": boolean,
"links": [ "string" ],
"linuxParameters": { "capabilities": { "add": [ "string" ],
  "drop": [ "string" ]
},
"devices": [ {
  "containerPath": "string",
  "hostPath": "string",
  "permissions": [ "string" ]
} ],
"initProcessEnabled": boolean,
"maxSwap": number,
"sharedMemorySize": number,
"swappiness": number,
"tmpfs": [ {
  "containerPath": "string",
  "mountOptions": [ "string" ]
} ]}
"size": number
}
},
"logConfiguration": {
  "logDriver": "string",
  "options": {
    "string": "string"
  },
  "secretOptions": [
    {
      "name": "string",
      "valueFrom": "string"
    }
  ]
},
"memory": number,
"memoryReservation": number,
"mountPoints": [
  {
    "containerPath": "string",
    "readOnly": boolean,
    "sourceVolume": "string"
  }
],
"name": "string",
"portMappings": [
  {
    "containerPort": number,
    "hostPort": number,
    "protocol": "string"
  }
],
"privileged": boolean,
"pseudoTerminal": boolean,
"readonlyRootFilesystem": boolean,
"repositoryCredentials": {
  "credentialsParameter": "string"
},
"resourceRequirements": [
  {
    "type": "string",
    "value": "string"
  }
],
"secrets": [
  {
    "name": "string",
    "valueFrom": "string"
  }
],
"startTimeout": number,
"stopTimeout": number,
"systemControls": [
  {
    "namespace": "string",
    "value": "string"
  }
],
"ulimits": [
  {
    "hardLimit": number,
    "name": "string",
    "softLimit": number
  }
]
"user": "string",
"volumesFrom": [
  {
    "readOnly": boolean,
    "sourceContainer": "string"
  }
],
"workingDirectory": "string"
],
"cpu": "string",
"executionRoleArn": "string",
"family": "string",
"inferenceAccelerators": [
  {
    "deviceName": "string",
    "deviceType": "string"
  }
],
"ipcMode": "string",
"memory": "string",
"networkMode": "string",
"pidMode": "string",
"placementConstraints": [
  {
    "expression": "string",
    "type": "string"
  }
],
"proxyConfiguration": {
  "containerName": "string",
  "properties": [
    {
      "name": "string",
      "value": "string"
    }
  ],
  "type": "string"
},
"requiresAttributes": [
  {
    "name": "string",
    "targetId": "string",
    "targetType": "string",
    "value": "string"
  }
],
"requiresCompatibilities": [ "string" ],
"revision": number,
"status": "string",
"taskDefinitionArn": "string",
"taskRoleArn": "string",
"volumes": [
  {
    "dockerVolumeConfiguration": {
      "autoprovision": boolean,
      "driver": "string",
      "driverOpts": {
        "string": "string"
      },
      "labels": {
        "string": "string"
      },
      "scope": "string"
    },
    "efsVolumeConfiguration": {
      "autoprovision": boolean,
      "driver": "string",
      "driverOpts": {
        "string": "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.

tags (p. 166)

The list of tags associated with the task definition.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

taskDefinition (p. 166)

The full description of the registered task definition.

Type: TaskDefinition (p. 352) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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
```


```
{
  "taskDefinition": {
    "containerDefinitions": [
      {
        "cpu": 10,
        "environment": [],
        "essential": true,
        "image": "wordpress",
        "links": [
          "mysql"
        ],
        "memory": 500,
        "mountPoints": [],
        "name": "wordpress",
        "portMappings": [
          {
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
          }
        ],
        "volumesFrom": []
      },
      {
        "cpu": 10,
        "environment": [
          {
            "name": "MYSQL_ROOT_PASSWORD",
            "value": "password"
          }
        ],
        "essential": true,
        "image": "mysql",
        "memory": 500,
        "mountPoints": [],
        "name": "mysql",
        "portMappings": [],
        "volumesFrom": []
      }
    ],
    "family": "hello_world",
    "networkMode": "host",
    "requiresAttributes": [
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
      }
    ],
    "revision": 4,
    "status": "ACTIVE",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:4",
    "volumes": []
  }
}
```

### 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 V3
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. 185) to use your own scheduler or place tasks manually on specific container instances.

The Amazon ECS API follows an eventual consistency model, due to the distributed nature of the system supporting the API. This means that the result of an API command you run that affects your Amazon ECS resources might not be immediately visible to all subsequent commands you run. Keep this in mind when you carry out an API command that immediately follows a previous API command.

To manage eventual consistency, you can do the following:

- Confirm the state of the resource before you run a command to modify it. Run the DescribeTasks command using an exponential backoff algorithm to ensure that you allow enough time for the previous command to propagate through the system. To do this, run the DescribeTasks command repeatedly, starting with a couple of seconds of wait time and increasing gradually up to five minutes of wait time.
- Add wait time between subsequent commands, even if the DescribeTasks command returns an accurate response. Apply an exponential backoff algorithm starting with a couple of seconds of wait time, and increase gradually up to about five minutes of wait time.

Request Syntax

```json
{
   "capacityProviderStrategy": [  
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
   ],
   "cluster": "string",
   "count": number,
   "enableECSManagedTags": boolean,
   "group": "string",
   "launchType": "string",
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
   "overrides": {
      "containerOverrides": [  
         {
            "command": [ "string" ],
            "cpu": number,
            "environment": [  
               {
                  "name": "string",
                  "value": "string"
               }
            ]
         }
      ]
   }
}
```

API Version 2014-11-13

174
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**capacityProviderStrategy (p. 174)**

The capacity provider strategy to use for the task.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The PutClusterCapacityProviders (p. 150) API is used to associate a
capacity provider with a cluster. Only capacity providers with an **ACTIVE** or **UPDATING** status can be used.

If a `capacityProviderStrategy` is specified, the `launchType` parameter must be omitted. If no `capacityProviderStrategy` or `launchType` is specified, the `defaultCapacityProviderStrategy` for the cluster is used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the `CreateCapacityProvider` API operation.

To use a AWS Fargate capacity provider, specify either the `FARGATE` or `FARGATE_SPOT` capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

The `PutClusterCapacityProviders` API operation is used to update the list of available capacity providers for a cluster after the cluster is created.

Type: Array of `CapacityProviderStrategyItem` objects

Required: No

**cluster** (p. 174)

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. 174)

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

**enableECSManagedTags** (p. 174)

Specifies whether to enable Amazon ECS managed tags for the task. For more information, see Tagging Your Amazon ECS Resources in the Amazon Elastic Container Service Developer Guide.

Type: Boolean

Required: No

**group** (p. 174)

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. 174)

The launch type on which to run your task. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

If a `launchType` is specified, the `capacityProviderStrategy` parameter must be omitted.

Type: String
Valid Values: EC2 | FARGATE

Required: No

networkConfiguration (p. 174)

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. 319) object

Required: No

overrides (p. 174)

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. 358) object

Required: No

placementConstraints (p. 174)

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 runtime).

Type: Array of PlacementConstraint (p. 321) objects

Required: No

placementStrategy (p. 174)

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. 322) objects

Required: No

platformVersion (p. 174)

The platform version the task should run. A platform version is only specified for tasks using the Fargate launch type. If one is not specified, the LATEST platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

propagateTags (p. 174)

Specifies whether to propagate the tags from the task definition to the task. If no value is specified, the tags are not propagated. Tags can only be propagated to the task during task creation. To add tags to a task after task creation, use the TagResource (p. 209) API action.
Note
An error will be received if you specify the SERVICE option when running a task.

Type: String

Valid Values: TASK_DEFINITION | SERVICE

Required: No

referenceId (p. 174)
The reference ID to use for the task.

Type: String

Required: No

startedBy (p. 174)
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. 136) 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

tags (p. 174)
The metadata that you apply to the task to help you categorize and organize them. Each tag consists
of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:
• Maximum number of tags per resource - 50
• For each resource, each tag key must be unique, and each tag key can have only one value.
• Maximum key length - 128 Unicode characters in UTF-8
• Maximum value length - 256 Unicode characters in UTF-8
• If your tagging schema is used across multiple services and resources, remember that other
services may have restrictions on allowed characters. Generally allowed characters are: letters,
numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
• Tag keys and values are case-sensitive.
• Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys
or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix.
Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskDefinition (p. 174)
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.
Response Syntax

```json
{
  "failures": [  
    {  
      "arn": "string",
      "detail": "string",
      "reason": "string"
    }
  ],  
  "tasks": [  
    {  
      "attachments": [  
        {  
          "details": [  
            {  
              "name": "string",
              "value": "string"
            }
          ],  
          "id": "string",
          "status": "string",
          "type": "string"
        }
      ],  
      "attributes": [  
        {  
          "name": "string",
          "targetId": "string",
          "targetType": "string",
          "value": "string"
        }
      ],  
      "availabilityZone": "string",
      "capacityProviderName": "string",
      "clusterArn": "string",
      "connectivity": "string",
      "connectivityAt": number,
      "containerInstanceArn": "string",
      "containers": [  
        {  
          "containerArn": "string",
          "cpu": "string",
          "exitCode": number,
          "gpuIds": [ "string" ],
          "healthStatus": "string",
          "image": "string",
          "imageDigest": "string",
          "lastStatus": "string",
          "memory": "string",
          "memoryReservation": "string",
          "name": "string",
          "networkBindings": [  
            {  
              "bindIP": "string",
              "containerPort": number,
              "hostPort": number,
              "protocol": "string"
            }
          ]
        }
      ]
    }
  ]
}
```
],
"networkInterfaces": [
{
"attachmentId": "string",
"ipv6Address": "string",
"privateIpv4Address": "string"
}
],
"reason": "string",
"runtimeId": "string",
"taskArn": "string"
}
],
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"healthStatus": "string",
"inferenceAccelerators": [
{
"deviceName": "string",
"deviceType": "string"
}
],
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
"containerOverrides": [
{
"command": [ "string" ],
"cpu": number,
"environment": [ 
{
"name": "string",
"value": "string"
}
],
"memory": number,
"memoryReservation": number,
"name": "string",
"resourceRequirements": [
{
"type": "string",
"value": "string"
}
]
}
],
"cpu": "string",
"executionRoleArn": "string",
"inferenceAcceleratorOverrides": [
{
"deviceName": "string",
"deviceType": "string"
}
],
"memory": "string",
"taskRoleArn": "string"
},
"platformVersion": "string",
"pull_startedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "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. 179)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

tasks (p. 179)

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. 346) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

AccessDeniedException

You do not have authorization to perform the requested action.

HTTP Status Code: 400

BlockedException

Your AWS account has been blocked. For more information, contact AWS Support.

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. 113). 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
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

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

API Version 2014-11-13
182
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": {
        "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 V3
StartTask

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

Alternatively, you can use RunTask (p. 174) to place tasks for you. For more information, see Scheduling Tasks in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
   "cluster": "string",
   "containerInstances": [ "string" ],
   "enableECSManagedTags": boolean,
   "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",
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
            ]
         },
         {
            "command": [ "string" ],
            "cpu": number,
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "memory": "string",
            "taskRoleArn": "string"
         },
         {
            "cpu": "string",
            "executionRoleArn": "string",
            "inferenceAcceleratorOverrides": [
               {
                  "deviceName": "string",
                  "deviceType": "string"
               }
            ],
            "memory": "string",
            "tags": [
               {
                  "key": "string",
                  "value": "string"
               }
            ]
         }
      ],
      "propagateTags": "string",
      "referenceId": "string",
      "startedBy": "string",
      "tags": ["string"],
      "containerInstances": ["string"],
      "enableECSManagedTags": "boolean",
      "group": "string",
      "networkConfiguration": {
         "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": ["string"],
            "subnets": ["string"]
         }
      },
      "overrides": {
         "containerOverrides": [
            {
               "command": ["string"],
               "cpu": number,
               "environment": ["string"],
               "memory": number,
               "memoryReservation": number,
               "name": "string",
               "resourceRequirements": ["string"]
            }
         ]
      },
      "propagateTags": "string",
      "referenceId": "string",
      "startedBy": "string",
      "tags": ["string"]
   }
}
```
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

cluster (p. 185)

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

countryGlans (p. 185)

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

enableECSManagedTags (p. 185)

Specifies whether to enable Amazon ECS managed tags for the task. For more information, see Tagging Your Amazon ECS Resources in the Amazon Elastic Container Service Developer Guide.

Type: Boolean

Required: No

group (p. 185)

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. 185)

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. 319) object

Required: No

overrides (p. 185)

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. 358) object

Required: No

propagateTags (p. 185)

Specifies whether to propagate the tags from the task definition or the service to the task. If no value is specified, the tags are not propagated.

Type: String

Valid Values: TASK_DEFINITION | SERVICE

Required: No

referenceId (p. 185)

The reference ID to use for the task.

Type: String

Required: No

startedBy (p. 185)

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. 136) 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

tags (p. 185)

The metadata that you apply to the task to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects
Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

**taskDefinition (p. 185)**

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",
      "detail": "string",
      "reason": "string"
    }
  ],
  "tasks": [
    {
      "attachments": [
        {
          "details": [
            {
              "name": "string",
              "value": "string"
            }
          ],
          "id": "string",
          "status": "string",
          "type": "string"
        }
      ],
      "attributes": [
        {
          "name": "string",
          "targetId": "string",
          "targetType": "string",
          "value": "string"
        }
      ],
      "availabilityZone": "string",
      "capacityProviderName": "string",
      "clusterArn": "string",
      "connectivity": "string",
      "connectivityAt": number,
      "containerInstanceArn": "string",
      "containers": [
        {
          "containerArn": "string",
          "cpu": "string",
          "exitCode": number,
          "gpuIds": [ "string" ],
          "healthStatus": "string",
          "image": "string",
          "imageDigest": "string",
          "lastStatus": "string",
          "memory": "string"
        }
      ]
    }
  ]
}
```
"memoryReservation": "string",
"name": "string",
"networkBindings": [
{
  "bindIP": "string",
  "containerPort": number,
  "hostPort": number,
  "protocol": "string"
}
],
"networkInterfaces": [
{
  "attachmentId": "string",
  "ipv6Address": "string",
  "privateIpv4Address": "string"
}
],
"reason": "string",
"runtimeId": "string",
"taskArn": "string"
},
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"healthStatus": "string",
"inferenceAccelerators": [
{
  "deviceName": "string",
  "deviceType": "string"
}
],
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
  "containerOverrides": [
  {
    "command": [ "string" ],
    "cpu": number,
    "environment": [
    {
      "name": "string",
      "value": "string"
    }
    ],
    "memory": number,
    "memoryReservation": number,
    "name": "string",
    "resourceRequirements": [
    {
      "type": "string",
      "value": "string"
    }
    ]
  }
  ],
  "cpu": "string",
  "executionRoleArn": "string",
  "inferenceAcceleratorOverrides": [
  {
    "deviceName": "string",
    "deviceType": "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. 188)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

tasks (p. 188)

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. 346) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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
```
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 V3
StopTask

Stops a running task. Any tags associated with the task will be deleted.

When StopTask (p. 194) is called on a task, the equivalent of `docker stop` is issued to the containers running in the task. This results in a `SIGTERM` value and a default 30-second timeout, after which the `SIGKILL` value is sent and the containers are forcibly stopped. If the container handles the `SIGTERM` value gracefully and exits within 30 seconds from receiving it, no `SIGKILL` value 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**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

`cluster (p. 194)`

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. 194)`

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. 94) API operations on this task. Up to 255 characters are allowed in this message.

Type: String

Required: No

`task (p. 194)`

The task ID or full Amazon Resource Name (ARN) 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"
      }
    ],
    "attributes": [
      {
        "name": "string",
        "targetId": "string",
        "targetType": "string",
        "value": "string"
      }
    ],
    "availabilityZone": "string",
    "capacityProviderName": "string",
    "clusterArn": "string",
    "connectivity": "string",
    "connectivityAt": number,
    "containerInstanceArn": "string",
    "containers": [
      {
        "containerArn": "string",
        "cpu": "string",
        "exitCode": number,
        "gpuIds": [ "string" ],
        "healthStatus": "string",
        "image": "string",
        "imageDigest": "string",
        "lastStatus": "string",
        "memory": "string",
        "memoryReservation": "string",
        "name": "string",
        "networkBindings": [
          {
            "bindIP": "string",
            "containerPort": number,
            "hostPort": number,
            "protocol": "string"
          }
        ],
        "networkInterfaces": [
          {
            "attachmentId": "string",
            "ipv6Address": "string",
            "privateIpv4Address": "string"
          }
        ],
        "reason": "string",
        "runtimeId": "string",
        "taskArn": "string"
      }
    ]
  }
}
```

API Version 2014-11-13
"cpu": "string",
"createdAt": number,
"desiredStatus": "string",
"executionStoppedAt": number,
"group": "string",
"healthStatus": "string",
"inferenceAccelerators": [
    {
        "deviceName": "string",
        "deviceType": "string"
    }
],
"lastStatus": "string",
"launchType": "string",
"memory": "string",
"overrides": {
    "containerOverrides": [
        {
            "command": [ "string" ],
            "cpu": number,
            "environment": [
                {"name": "string",
                 "value": "string"
            }
        ],
        "memory": number,
        "memoryReservation": number,
        "name": "string",
        "resourceRequirements": [
            {"type": "string",
             "value": "string"
        }
    ]
},
"cpu": "string",
"executionRoleArn": "string",
"inferenceAcceleratorOverride": {
    "deviceName": "string",
    "deviceType": "string"
},
"memory": "string",
"taskRoleArn": "string"
},
"platformVersion": "string",
"pullStartedAt": number,
"pullStoppedAt": number,
"startedAt": number,
"startedBy": "string",
"stopCode": "string",
"stoppedAt": number,
"stoppedReason": "string",
"stoppingAt": number,
"tags": [
    {
        "key": "string",
        "value": "string"
    }
],
"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.

**task (p. 195)**

The task that was stopped.

Type: Task (p. 346) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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. 113). 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-a079-961675296646",
        "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/e09064f7-7361-4c87-8ab9-8d073bbdbcb9",
        "lastStatus": "RUNNING",
        "name": "busybox",
        "networkBindings": [],
        "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-b493-371970c6c4d6"
      }
    ]
  }
}
```
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 V3
SubmitAttachmentStateChanges

**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 an attachment changed states.

**Request Syntax**

```
{
   "attachments": [
   {
      "attachmentArn": "string",
      "status": "string"
   }
   ],
   "cluster": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**attachments (p. 200)**

Any attachments associated with the state change request.

Type: Array of AttachmentStateChange (p. 252) objects

Required: Yes

**cluster (p. 200)**

The short name or full ARN of the cluster that hosts the container instance the attachment belongs to.

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. 200)

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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

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 V3
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",
  "runtimeId": "string",
  "status": "string",
  "task": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 202)**

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

Type: String

Required: No

**containerName (p. 202)**

The name of the container.

Type: String

Required: No

**exitCode (p. 202)**

The exit code returned for the state change request.

Type: Integer

Required: No

**networkBindings (p. 202)**

The network bindings of the container.
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. 203)**

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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 V3
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

```
{
  "attachments": [
    {
      "attachmentArn": "string",
      "status": "string"
    }
  ],
  "cluster": "string",
  "containers": [
    {
      "containerName": "string",
      "exitCode": number,
      "imageDigest": "string",
      "networkBindings": [
        {
          "bindIP": "string",
          "containerPort": number,
          "hostPort": number,
          "protocol": "string"
        }
      ],
      "reason": "string",
      "runtimeId": "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. 371).

The request accepts the following data in JSON format.

**attachments (p. 205)**

Any attachments associated with the state change request.

Type: Array of AttachmentStateChange (p. 252) objects

Required: No

**cluster (p. 205)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the task.
Type: String
Required: No

**containers (p. 205)**
Any containers associated with the state change request.
Type: Array of ContainerStateChange (p. 286) objects
Required: No

**executionStoppedAt (p. 205)**
The Unix timestamp for when the task execution stopped.
Type: Timestamp
Required: No

**pullStartedAt (p. 205)**
The Unix timestamp for when the container image pull began.
Type: Timestamp
Required: No

**pullStoppedAt (p. 205)**
The Unix timestamp for when the container image pull completed.
Type: Timestamp
Required: No

**reason (p. 205)**
The reason for the state change request.
Type: String
Required: No

**status (p. 205)**
The status of the state change request.
Type: String
Required: No

**task (p. 205)**
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. 206)

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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

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 V3
TagResource

Associates the specified tags to a resource with the specified resourceArn. If existing tags on a resource are not specified in the request parameters, they are not changed. When a resource is deleted, the tags associated with that resource are deleted as well.

Request Syntax

```json
{
  "resourceArn": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

resourceArn (p. 209)

The Amazon Resource Name (ARN) of the resource to which to add tags. Currently, the supported resources are Amazon ECS capacity providers, tasks, services, task definitions, clusters, and container instances.

Type: String

Required: Yes

tags (p. 209)

The tags to add to the resource. A tag is an array of key-value pairs.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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

ResourceNotFoundException

The specified resource could not be found.

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 tags the dev cluster with key team and value dev.
Sample Request

POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerServiceV20141113.TagResource
Content-Type: application/x-amz-json-1.1
X-Amz-Date: 20181026T194744Z
Authorization: AUTHPARAMS
Content-Length: 115

{
  "tags":[
    {
      "key":"team",
      "value":"dev"
    }
  ]
}

Sample Response

HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 2
Date: Fri, 26 Oct 2018 20:01:34 GMT

{}

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 V3
UntagResource

Deletes specified tags from a resource.

**Request Syntax**

```json
{
   "resourceArn": "string",
   "tagKeys": [ "string" ]
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**resourceArn (p. 212)**

The Amazon Resource Name (ARN) of the resource from which to delete tags. Currently, the supported resources are Amazon ECS capacity providers, tasks, services, task definitions, clusters, and container instances.

- Type: String
- Required: Yes

**tagKeys (p. 212)**

The keys of the tags to be removed.

- Type: Array of strings
- Pattern: `^(\[\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`
- Required: Yes

**Response Elements**

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

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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. 113). 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

ResourceNotFoundException

The specified resource could not be found.

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 tags the dev cluster with key team and value dev.

Sample Request

```
POST / HTTP/1.1
Host: madison.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UntagResource
Content-Type: application/x-amz-json-1.1
X-Amz-Date: 20181026T200134Z
Authorization: AUTHPARAMS
Content-Length: 93

{
  "tagKeys": ["team"
  ]
}
```

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 V3
UpdateClusterSettings

Modifies the settings to use for a cluster.

Request Syntax

```json
{
  "cluster": "string",
  "settings": [
    {
      "name": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

- **cluster** (p. 215)
  - The name of the cluster to modify the settings for.
  - Type: String
  - Required: Yes

- **settings** (p. 215)
  - The setting to use by default for a cluster. This parameter is used to enable CloudWatch Container Insights for a cluster. If this value is specified, it will override the `containerInsights` value set with PutAccountSetting (p. 141) or PutAccountSettingDefault (p. 144).
  - Type: Array of ClusterSetting (p. 263) objects
  - Required: Yes

Response Syntax

```json
{
  "cluster": {
    "activeServicesCount": number,
    "attachments": [
      {
        "details": [
          {
            "name": "string",
            "value": "string"
          }
        ],
        "id": "string",
        "status": "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.

**cluster (p. 215)**

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.

Type: Cluster (p. 259) object

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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. 113). 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

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 V3
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. 371).

The request accepts the following data in JSON format.

cluster (p. 218)

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. 218)

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": [
          {
            "name": "string",
..."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. 218)

The container instance for which the container agent was updated.

Type: ContainerInstance (p. 280) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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 (\texttt{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

```plaintext
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

```plaintext
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 V3
UpdateContainerInstancesState

Modifies the status of an Amazon ECS container instance.

Once a container instance has reached an **ACTIVE** state, 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.

**Important**
A container instance cannot be changed to **DRAINING** until it has reached an **ACTIVE** status. If the instance is in any other status, an error will be received.

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. 231).

- 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** is 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. 136).

When a container instance has been drained, you can set a container instance to **ACTIVE** status and once it has reached that status 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. 371).
The request accepts the following data in JSON format.

**cluster (p. 223)**

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. 223)**

A list of container instance IDs or full ARN entries.

Type: Array of strings

Required: Yes

**status (p. 223)**

The container instance state with which to update the container instance. The only valid values for this action are ACTIVE and DRAINING. A container instance can only be updated to DRAINING status once it has reached an ACTIVE state. If a container instance is in REGISTERING, DEREGISTERING, or REGISTRATION_FAILED state you can describe the container instance but will be unable to update the container instance state.

Type: String

Valid Values: ACTIVE | DRAINING | REGISTERING | DEREGISTERING | REGISTRATION_FAILED

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"
        }
      ]
    }
  ]
}```
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. 224)

The list of container instances.

Type: Array of ContainerInstance (p. 280) objects
failures (p. 224)

Any failures associated with the call.

Type: Array of Failure (p. 298) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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 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
Server: Server
Date: Tue, 20 Dec 2016 22:11:42 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 2344
Connection: keep-alive
x-amzn-RequestId: 49d68928-c701-11e6-8f99-6103d648cdad

{
  "containerInstances": [
    {
      "agentConnected": true,
      "attributes": ["ecs.availability-zone", "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-containter"
    },
    {
      "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
    }
  ]
}
Example

```json
{
"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-6e68ebd01f34",
"ec2InstanceId": "i-05d99c76f957727ef6",
"pendingTasksCount": 0,
"registeredResources": [
{
"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": [
"22",
"2376",
"2375",
"51678",
"51679"
],
"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
See Also

- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
Important

Updating the task placement strategies and constraints on an Amazon ECS service remains in preview and is a Beta Service as defined by and subject to the Beta Service Participation Service Terms located at https://aws.amazon.com/service-terms ("Beta Terms"). These Beta Terms apply to your participation in this preview.

Modifies the parameters of a service.

For services using the rolling update (ECS) deployment controller, the desired count, deployment configuration, network configuration, task placement constraints and strategies, or task definition used can be updated.

For services using the blue/green (CODE_DEPLOY) deployment controller, only the desired count, deployment configuration, task placement constraints and strategies, and health check grace period can be updated using this API. If the network configuration, platform version, or task definition need to be updated, a new AWS CodeDeploy deployment should be created. For more information, see CreateDeployment in the AWS CodeDeploy API Reference.

For services using an external deployment controller, you can update only the desired count, task placement constraints and strategies, and health check grace period using this API. If the launch type, load balancer, network configuration, platform version, or task definition need to be updated, you should create a new task set. For more information, see CreateTaskSet (p. 27).

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.

If you have updated the Docker image of your application, you can create a new task definition with that image and deploy it to your service. The service scheduler uses the minimum healthy percent and maximum percent parameters (in the service's deployment configuration) to determine the deployment strategy.

Note

If your updated Docker image uses the same tag as what is in the existing task definition for your service (for example, my_image:latest), you do not need to create a new revision of your task definition. You can update the service using the forceNewDeployment option. The new tasks launched by the deployment pull the current image/tag combination from your repository when they start.

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. 231) 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

```json
{
    "capacityProviderStrategy": [
        {
            "base": number,
            "capacityProvider": "string",
            "weight": number
        }
    ],
    "cluster": "string",
    "deploymentConfiguration": {
        "maximumPercent": number,
        "minimumHealthyPercent": number
    },
    "desiredCount": number,
    "forceNewDeployment": boolean,
    "healthCheckGracePeriodSeconds": number,
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
        }
    },
    "placementConstraints": [
        {
            "expression": "string",
            "type": "string"
        }
    ]
}
```

**API Version 2014-11-13**
Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

capacityProviderStrategy (p. 232)

The capacity provider strategy to update the service to use.

If the service is using the default capacity provider strategy for the cluster, the service can be updated to use one or more capacity providers as opposed to the default capacity provider strategy. However, when a service is using a capacity provider strategy that is not the default capacity provider strategy, the service cannot be updated to use the cluster's default capacity provider strategy.

A capacity provider strategy consists of one or more capacity providers along with the base and weight to assign to them. A capacity provider must be associated with the cluster to be used in a capacity provider strategy. The PutClusterCapacityProviders (p. 150) API is used to associate a capacity provider with a cluster. Only capacity providers with an ACTIVE or UPDATING status can be used.

If specifying a capacity provider that uses an Auto Scaling group, the capacity provider must already be created. New capacity providers can be created with the CreateCapacityProvider (p. 4) API operation.

To use a AWS Fargate capacity provider, specify either the FARGATE or FARGATE_SPOT capacity providers. The AWS Fargate capacity providers are available to all accounts and only need to be associated with a cluster to be used.

The PutClusterCapacityProviders (p. 150) API operation is used to update the list of available capacity providers for a cluster after the cluster is created.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

cluster (p. 232)

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. 232)

Optional deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.
Type: DeploymentConfiguration (p. 291) object

Requested: No

desiredCount (p. 232)

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

Type: Integer

Requested: No

forceNewDeployment (p. 232)

Whether to force a new deployment of the service. Deployments are not forced by default. You can use this option to trigger a new deployment with no service definition changes. For example, you can update a service's tasks to use a newer Docker image with the same image/tag combination (my_image:latest) or to roll Fargate tasks onto a newer platform version.

Type: Boolean

Requested: No

healthCheckGracePeriodSeconds (p. 232)

The period of time, in seconds, that the Amazon ECS service scheduler should ignore unhealthy Elastic Load Balancing target health checks after a task has first started. This is only valid if your service is configured to use a load balancer. If your service's tasks take a while to start and respond to Elastic Load Balancing health checks, you can specify a health check grace period of up to 2,147,483,647 seconds. During that time, the Amazon ECS service scheduler ignores the Elastic Load Balancing health check status. This grace period can prevent the ECS service scheduler from marking tasks as unhealthy and stopping them before they have time to come up.

Type: Integer

Requested: No

networkConfiguration (p. 232)

An object representing the network configuration for a task or service.

Type: NetworkConfiguration (p. 319) object

Requested: No

placementConstraints (p. 232)

An array of task placement constraint objects to update the service to use. If no value is specified, the existing placement constraints for the service will remain unchanged. If this value is specified, it will override any existing placement constraints defined for the service. To remove all existing placement constraints, specify an empty array.

You can specify a maximum of 10 constraints per task (this limit includes constraints in the task definition and those specified at runtime).

Type: Array of PlacementConstraint (p. 321) objects

Requested: No

placementStrategy (p. 232)

The task placement strategy objects to update the service to use. If no value is specified, the existing placement strategy for the service will remain unchanged. If this value is specified, it will override
the existing placement strategy defined for the service. To remove an existing placement strategy, specify an empty object.

You can specify a maximum of five strategy rules per service.

Type: Array of PlacementStrategy (p. 322) objects

Required: No

platformVersion (p. 232)

The platform version on which your tasks in the service are running. A platform version is only specified for tasks using the Fargate launch type. If a platform version is not specified, the LATEST platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

service (p. 232)

The name of the service to update.

Type: String

Required: Yes

taskDefinition (p. 232)

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

```
{ 
  "service": { 
    "capacityProviderStrategy": [ 
      { 
        "base": number, 
        "capacityProvider": "string", 
        "weight": number 
      } 
    ], 
  "clusterArn": "string", 
  "createdAt": number, 
  "createdBy": "string", 
  "deploymentConfiguration": { 
    "maximumPercent": number, 
    "minimumHealthyPercent": number 
  }, 
  "deploymentController": { 
    "type": "string" 
  }, 
  "deployments": [ 
  
```

API Version 2014-11-13

235
"capacityProviderStrategy": [
  {
    "base": number,
    "capacityProvider": "string",
    "weight": number
  }
],
"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,
"enableECSManagedTags": boolean,
"events": [
  {
    "createdAt": number,
    "id": "string",
    "message": "string"
  }
],
"healthCheckGracePeriodSeconds": number,
"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"
"propagateTags": "string",
"roleArn": "string",
"runningCount": number,
"schedulingStrategy": "string",
"serviceArn": "string",
"serviceName": "string",
"serviceRegistries": [
    {
        "containerName": "string",
        "containerPort": number,
        "port": number,
        "registryArn": "string"
    }
],
"status": "string",
"tags": [
    {
        "key": "string",
        "value": "string"
    }
],
"taskDefinition": "string",
"taskSets": [
    {
        "capacityProviderStrategy": [
            {
                "base": number,
                "capacityProvider": "string",
                "weight": number
            }
        ],
        "clusterArn": "string",
        "computedDesiredCount": number,
        "createdAt": number,
        "externalId": "string",
        "id": "string",
        "launchType": "string",
        "loadBalancers": [
            {
                "containerName": "string",
                "containerPort": number,
                "loadBalancerName": "string",
                "targetGroupArn": "string"
            }
        ],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "assignPublicIp": "string",
                "securityGroups": [ "string" ],
                "subnets": [ "string" ]
            }
        },
        "pendingCount": number,
        "platformVersion": "string",
        "runningCount": number,
        "scale": {
            "unit": "string",
            "value": number
        },
        "serviceArn": "string",
        "serviceRegistries": [
            {
                "containerName": "string",
                "containerPort": number,
                "port": number,
                "registryArn": "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. 235)**

The full description of your service following the update call.

Type: Service (p. 334) object

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 373).

**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. 113). 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. 12).

HTTP Status Code: 400

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 120). 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,
        "id": "ecs-svc/9223370606521064774",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:10",
        "updatedAt": 1430336267.173
      }
    ],
    "desiredCount": 3,
    "events": [],
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/hello_world",
    "serviceName": "hello_world",
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:10"
  }
}

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 V3

API Version 2014-11-13 240
UpdateServicePrimaryTaskSet

Modifies which task set in a service is the primary task set. Any parameters that are updated on the primary task set in a service will transition to the service. This is used when a service uses the EXTERNAL deployment controller type. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 241)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service that the task set exists in.

- Type: String
- Required: Yes

**primaryTaskSet (p. 241)**

The short name or full Amazon Resource Name (ARN) of the task set to set as the primary task set in the deployment.

- Type: String
- Required: Yes

**service (p. 241)**

The short name or full Amazon Resource Name (ARN) of the service that the task set exists in.

- Type: String
- Required: Yes

Response Syntax

```json
{
    "taskSet": {
        "capacityProviderStrategy": [
            {
                "base": number,
                "capacityProvider": "string",
                "weight": number
            }
        ]
    }
}
```
If the action is successful, the service sends back an HTTP 200 response. The following data is returned in JSON format by the service.

taskSet (p. 241)

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

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. 113). 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

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. 12).

HTTP Status Code: 400

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 120). Amazon ECS services are cluster-specific and Region-specific.

HTTP Status Code: 400

TaskSetNotFoundException

The specified task set could not be found. You can view your available task sets with DescribeTaskSets (p. 100). Task sets are specific to each cluster, service and Region.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task is not supported in this Region.

HTTP Status Code: 400
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 V3
UpdateTaskSet

Modifies a task set. This is used when a service uses the EXTERNAL deployment controller type. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Request Syntax

```json
{
   "cluster": "string",
   "scale": {
      "unit": "string",
      "value": number
   },
   "service": "string",
   "taskSet": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 371).

The request accepts the following data in JSON format.

**cluster (p. 245)**

The short name or full Amazon Resource Name (ARN) of the cluster that hosts the service that the task set exists in.

Type: String

Required: Yes

**scale (p. 245)**

A floating-point percentage of the desired number of tasks to place and keep running in the task set.

Type: Scale (p. 332) object

Required: Yes

**service (p. 245)**

The short name or full Amazon Resource Name (ARN) of the service that the task set exists in.

Type: String

Required: Yes

**taskSet (p. 245)**

The short name or full Amazon Resource Name (ARN) of the task set to update.

Type: String

Required: Yes

Response Syntax

```json
{
}
```
"taskSet": {  
  "capacityProviderStrategy": [  
    {  
      "base": number,  
      "capacityProvider": "string",  
      "weight": number  
    },  
    "clusterArn": "string",  
    "computedDesiredCount": number,  
    "createdAt": number,  
    "externalId": "string",  
    "id": "string",  
    "launchType": "string",  
    "loadBalancers": [  
      {  
        "containerName": "string",  
        "containerPort": number,  
        "loadBalancerName": "string",  
        "targetGroupArn": "string"  
      }  
    ],  
    "networkConfiguration": {  
      "awsvpcConfiguration": {  
        "assignPublicIp": "string",  
        "securityGroups": [ "string" ],  
        "subnets": [ "string" ]  
      }  
    },  
    "pendingCount": number,  
    "platformVersion": "string",  
    "runningCount": number,  
    "scale": {  
      "unit": "string",  
      "value": number  
    },  
    "serviceArn": "string",  
    "serviceRegistries": [  
      {  
        "containerName": "string",  
        "containerPort": number,  
        "port": number,  
        "registryArn": "string"  
      }  
    ],  
    "stabilityStatus": "string",  
    "stabilityStatusAt": number,  
    "startedBy": "string",  
    "status": "string",  
    "tags": [  
      {  
        "key": "string",  
        "value": "string"  
      }  
    ],  
    "taskDefinition": "string",  
    "taskSetArn": "string",  
    "updatedAt": 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.

**taskSet (p. 245)**

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Type: TaskSet (p. 360) object

---

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 373).

### 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. 113). 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

### 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. 12).

HTTP Status Code: 400

### ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 120). Amazon ECS services are cluster-specific and Region-specific.

HTTP Status Code: 400

### TaskSetNotFoundException

The specified task set could not be found. You can view your available task sets with DescribeTaskSets (p. 100). Task sets are specific to each cluster, service, and Region.
HTTP Status Code: 400

**UnsupportedFeatureException**

The specified task is not supported in this Region.

HTTP Status Code: 400

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 V3
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. 251)
- AttachmentStateChange (p. 252)
- Attribute (p. 253)
- AutoScalingGroupProvider (p. 254)
- AwsVpcConfiguration (p. 255)
- CapacityProvider (p. 256)
- CapacityProviderStrategyItem (p. 258)
- Cluster (p. 259)
- ClusterSetting (p. 263)
- Container (p. 264)
- ContainerDefinition (p. 267)
- ContainerDependency (p. 278)
- ContainerInstance (p. 280)
- ContainerOverride (p. 284)
- ContainerStateChange (p. 286)
- Deployment (p. 288)
- DeploymentConfiguration (p. 291)
- DeploymentController (p. 293)
- Device (p. 294)
- DockerVolumeConfiguration (p. 295)
- EFSVolumeConfiguration (p. 297)
- Failure (p. 298)
- FirelensConfiguration (p. 299)
- HealthCheck (p. 300)
- HostEntry (p. 302)
- HostVolumeProperties (p. 303)
- InferenceAccelerator (p. 304)
- InferenceAcceleratorOverride (p. 305)
- KernelCapabilities (p. 306)
- KeyValuePair (p. 308)
- LinuxParameters (p. 309)
- LoadBalancer (p. 311)
- LogConfiguration (p. 313)
- ManagedScaling (p. 315)
- MountPoint (p. 317)
• NetworkBinding (p. 318)
• NetworkConfiguration (p. 319)
• NetworkInterface (p. 320)
• PlacementConstraint (p. 321)
• PlacementStrategy (p. 322)
• PlatformDevice (p. 323)
• PortMapping (p. 324)
• ProxyConfiguration (p. 326)
• RepositoryCredentials (p. 328)
• Resource (p. 329)
• ResourceRequirement (p. 331)
• Scale (p. 332)
• Secret (p. 333)
• Service (p. 334)
• ServiceEvent (p. 339)
• ServiceRegistry (p. 340)
• Setting (p. 342)
• SystemControl (p. 343)
• Tag (p. 344)
• Task (p. 346)
• TaskDefinition (p. 352)
• TaskDefinitionPlacementConstraint (p. 357)
• TaskOverride (p. 358)
• TaskSet (p. 360)
• Tmpfs (p. 365)
• Ulimit (p. 366)
• VersionInfo (p. 367)
• Volume (p. 368)
• VolumeFrom (p. 370)
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. 308) 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 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 V3
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 V3
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 V3
AutoScalingGroupProvider

The details of the Auto Scaling group for the capacity provider.

Contents

autoScalingGroupArn

The Amazon Resource Name (ARN) that identifies the Auto Scaling group.

Type: String

Required: Yes

managedScaling

The managed scaling settings for the Auto Scaling group capacity provider.

Type: ManagedScaling (p. 315) object

Required: No

managedTerminationProtection

The managed termination protection setting to use for the Auto Scaling group capacity provider. This determines whether the Auto Scaling group has managed termination protection.

Important
When using managed termination protection, managed scaling must also be used otherwise managed termination protection will not work.

When managed termination protection is enabled, Amazon ECS prevents the Amazon EC2 instances in an Auto Scaling group that contain tasks from being terminated during a scale-in action. The Auto Scaling group and each instance in the Auto Scaling group must have instance protection from scale-in actions enabled as well. For more information, see Instance Protection in the AWS Auto Scaling User Guide.

When managed termination protection is disabled, your Amazon EC2 instances are not protected from termination when the Auto Scaling group scales in.

Type: String

Valid Values: ENABLED | DISABLED

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 V3
AwsVpcConfiguration

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

Contents

**assignPublicIp**

Whether the task's elastic network interface receives a public IP address. The default value is `DISABLED`.

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. There is a limit of 5 security groups that can be specified per `AwsVpcConfiguration`.

*Note*

All specified security groups must be from the same VPC.

Type: Array of strings

Required: No

**subnets**

The subnets associated with the task or service. There is a limit of 16 subnets that can be specified per `AwsVpcConfiguration`.

*Note*

All specified subnets must be from the same VPC.

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 V3
CapacityProvider

The details of a capacity provider.

Contents

autoScalingGroupProvider

The Auto Scaling group settings for the capacity provider.

Type: AutoScalingGroupProvider (p. 254) object

Required: No

capacityProviderArn

The Amazon Resource Name (ARN) that identifies the capacity provider.

Type: String

Required: No

name

The name of the capacity provider.

Type: String

Required: No

status

The current status of the capacity provider. Only capacity providers in an ACTIVE state can be used in a cluster.

Type: String

Valid Values: ACTIVE

Required: No

tags

The metadata that you apply to the capacity provider to help you categorize and organize it. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

API Version 2014-11-13
Array Members: Minimum number of 0 items. Maximum number of 50 items.

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 V3
CapacityProviderStrategyItem

The details of a capacity provider strategy.

Contents

base

The base value designates how many tasks, at a minimum, to run on the specified capacity provider. Only one capacity provider in a capacity provider strategy can have a base defined.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 100000.

Required: No

capacityProvider

The short name of the capacity provider.

Type: String

Required: Yes

weight

The weight value designates the relative percentage of the total number of tasks launched that should use the specified capacity provider.

For example, if you have a strategy that contains two capacity providers and both have a weight of 1, then when the base is satisfied, the tasks will be split evenly across the two capacity providers. Using that same logic, if you specify a weight of 1 for capacityProviderA and a weight of 4 for capacityProviderB, then for every one task that is run using capacityProviderA, four tasks would use capacityProviderB.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 1000.

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 V3
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. 120).

Type: Integer

Required: No

attachments

The resources attached to a cluster. When using a capacity provider with a cluster, the Auto Scaling plan that is created will be returned as a cluster attachment.

Type: Array of Attachment (p. 251) objects

Required: No

attachmentsStatus

The status of the capacity providers associated with the cluster. The following are the states that will be returned:

UPDATE_IN_PROGRESS

The available capacity providers for the cluster are updating. This occurs when the Auto Scaling plan is provisioning or deprovisioning.

UPDATE_COMPLETE

The capacity providers have successfully updated.

UPDATE_FAILED

The capacity provider updates failed.

Type: String

Required: No

capacityProviders

The capacity providers associated with the cluster.

Type: Array of strings

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

`defaultCapacityProviderStrategy`

The default capacity provider strategy for the cluster. When services or tasks are run in the cluster with no launch type or capacity provider strategy specified, the default capacity provider strategy is used.

Type: Array of `CapacityProviderStrategyItem (p. 258)` objects

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. This includes container instances in both `ACTIVE` and `DRAINING` status.

Type: Integer

Required: No

`runningTasksCount`

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

Type: Integer

Required: No

`settings`

The settings for the cluster. This parameter indicates whether CloudWatch Container Insights is enabled or disabled for a cluster.

Type: Array of `ClusterSetting (p. 263)` objects

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. 308) objects

Required: No

status

The status of the cluster. The following are the possible states that will be returned.

ACTIVE

The cluster is ready to accept tasks and if applicable you can register container instances with the cluster.

PROVISIONING

The cluster has capacity providers associated with it and the resources needed for the capacity provider are being created.

DEPROVISIONING

The cluster has capacity providers associated with it and the resources needed for the capacity provider are being deleted.

FAILED

The cluster has capacity providers associated with it and the resources needed for the capacity provider have failed to create.

INACTIVE

The cluster has been deleted. Clusters with an INACTIVE status may remain discoverable in your account for a period of time. However, this behavior is subject to change in the future, so you should not rely on INACTIVE clusters persisting.

Type: String

Required: No

tags

The metadata that you apply to the cluster to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

• Maximum number of tags per resource - 50
• For each resource, each tag key must be unique, and each tag key can have only one value.
• Maximum key length - 128 Unicode characters in UTF-8
• Maximum value length - 256 Unicode characters in UTF-8
• If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
• Tag keys and values are case-sensitive.
• Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

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 V3
ClusterSetting

The settings to use when creating a cluster. This parameter is used to enable CloudWatch Container Insights for a cluster.

Contents

name

The name of the cluster setting. The only supported value is containerInsights.

Type: String

Valid Values: containerInsights

Required: No

value

The value to set for the cluster setting. The supported values are enabled and disabled. If enabled is specified, CloudWatch Container Insights will be enabled for the cluster, otherwise it will be disabled unless the containerInsights account setting is enabled. If a cluster value is specified, it will override the containerInsights value set with PutAccountSetting (p. 141) or PutAccountSettingDefault (p. 144).

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 V3
Container

A Docker container that is part of a task.

Contents

containerArn

The Amazon Resource Name (ARN) of the container.

Type: String

Required: No

cpu

The number of CPU units set for the container. The value will be 0 if no value was specified in the container definition when the task definition was registered.

Type: String

Required: No

exitCode

The exit code returned from the container.

Type: Integer

Required: No

gpulids

The IDs of each GPU assigned to the container.

Type: Array of strings

Required: No

healthStatus

The health status of the container. If health checks are not configured for this container in its task definition, then it reports the health status as UNKNOWN.

Type: String

Valid Values: HEALTHY | UNHEALTHY | UNKNOWN

Required: No

image

The image used for the container.

Type: String

Required: No

imageDigest

The container image manifest digest.

Note

The imageDigest is only returned if the container is using an image hosted in Amazon ECR, otherwise it is omitted.
Type: String
Required: No

lastStatus
The last known status of the container.
Type: String
Required: No

memory
The hard limit (in MiB) of memory set for the container.
Type: String
Required: No

memoryReservation
The soft limit (in MiB) of memory set for 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. 318) objects
Required: No

networkInterfaces
The network interfaces associated with the container.
Type: Array of NetworkInterface (p. 320) 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

runtimeId
The ID of the Docker 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 V3
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 `Cmd` in the Create a container section of the Docker Remote API and the `COMMAND` parameter to `docker run`. For more information, see https://docs.docker.com/engine/reference/builder/#cmd. If there are multiple arguments, each argument should be a separated string in the array.

Type: Array of strings

Required: No

cpu

The number of CPU units reserved for the container. This parameter maps to `CpuShares` in the Create a container section of the Docker Remote API and the `--cpu-shares` option to `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 `cpu` value.

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.

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 CPU share constraint in the Docker documentation. The minimum valid CPU share value that the Linux kernel allows 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:

- **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.
- **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
dependsOn

The dependencies defined for container startup and shutdown. A container can contain multiple dependencies. When a dependency is defined for container startup, for container shutdown it is reversed.

For tasks using the EC2 launch type, the container instances require at least version 1.26.0 of the container agent to enable container dependencies. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the Amazon Elastic Container Service Developer Guide. If you are using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the ecs-init package. If your container instances are launched from version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

For tasks using the Fargate launch type, the task or service requires platform version 1.3.0 or later.

Type: Array of ContainerDependency (p. 278) objects

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 --format '{{.Server.APIVersion}}'

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.

With Windows containers, this parameter can be used to reference a credential spec file when
configuring a container for Active Directory authentication. For more information, see Using gMSAs
for Windows Containers in the Amazon Elastic Container Service Developer Guide.

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.

Type: Array of strings

Required: No

effective

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. 308) 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. 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 or tasks that use the `awsvpc` network mode.

Type: Array of HostEntry objects
Required: No
firelensconfiguration

The FireLens configuration for the container. This is used to specify and configure a log router for container logs. For more information, see Custom Log Routing in the Amazon Elastic Container Service Developer Guide.

Type: FirelensConfiguration object
Required: No
healthcheck

The container health check command and associated configuration parameters for the container. This parameter maps to `HealthCheck` in the Create a container section of the Docker Remote API and the `HEALTHCHECK` parameter of docker run.

Type: HealthCheck object
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.

Note
The hostname parameter is not supported if you are using the `awsvpc` network mode.

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`.

- When a new task starts, the Amazon ECS container agent pulls the latest version of the specified image and tag for the container to use. However, subsequent updates to a repository image are not propagated to already running tasks.
- 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>@sha256:94afd1f264d908bc90dbca0035a5b567EXAMPLE`.
- 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

`interactive`

When this parameter is `true`, this allows you to deploy containerized applications that require `stdin` or `tty` to be allocated. This parameter maps to `OpenStdin` in the `Create a container` section of the Docker Remote API and the `--interactive` option to `docker run`.

Type: Boolean

Required: No

`links`

The `links` parameter allows containers to communicate with each other without the need for port mappings. This parameter is only supported if the network mode of a task definition is `bridge`. The `name:internalName` construct is analogous to `name:alias` in Docker links. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed. For more information about linking Docker containers, go to Legacy container links in the Docker documentation. 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 kernel capabilities. For more information see `KernelCapabilities (p. 306)`.

**Note**
This parameter is not supported for Windows containers.

Type: `LinuxParameters (p. 309)` object
**logConfiguration**

The log configuration specification for the container.

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` 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 --format '{{.Server.APIVersion}}'
```

**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` object

**memory**

The amount (in MiB) of memory to present to the container. If your container attempts to exceed the memory specified here, the container is killed. The total amount of memory reserved for all containers within a task must be lower than the task `memory` value, if one is specified. This parameter maps to `Memory` in the `Create a container` section of the Docker Remote API and the `--memory` option to `docker run`.

If using the Fargate launch type, this parameter is optional.

If using the EC2 launch type, you must specify either a task-level memory value or a container-level memory value. If you specify both a container-level `memory` and `memoryReservation` value, `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.

Type: Integer

**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`.

If a task-level memory value is not specified, you must specify a non-zero integer for one or both of `memory` or `memoryReservation` in a container definition. 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.

Type: Integer

Required: No

`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. 317)` objects

Required: No

`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, and hyphens 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

Required: No

`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 `PortBinding`s 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. The assignments are also visible in the networkBindings section DescribeTasks (p. 94) responses.

Type: Array of PortMapping (p. 324) 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

**pseudoTerminal**

When this parameter is true, a TTY is allocated. This parameter maps to `Tty` in the Create a container section of the Docker Remote API and the `--tty` option to `docker run`.

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

**repositoryCredentials**

The private repository authentication credentials to use.

Type: RepositoryCredentials (p. 328) object

Required: No

**resourceRequirements**

The type and amount of a resource to assign to a container. The only supported resource is a GPU.

Type: Array of ResourceRequirement (p. 331) objects

Required: No
secrets

The secrets to pass to the container. For more information, see Specifying Sensitive Data in the Amazon Elastic Container Service Developer Guide.

Type: Array of Secret (p. 333) objects

Required: No

startTimeout

Time duration (in seconds) to wait before giving up on resolving dependencies for a container. For example, you specify two containers in a task definition with containerA having a dependency on containerB reaching a COMPLETE, SUCCESS, or HEALTHY status. If a startTimeout value is specified for containerB and it does not reach the desired status within that time then containerA will give up and not start. This results in the task transitioning to a STOPPED state.

For tasks using the Fargate launch type, this parameter requires that the task or service uses platform version 1.3.0 or later. If this parameter is not specified, the default value of 3 minutes is used.

For tasks using the EC2 launch type, if the startTimeout parameter is not specified, the value set for the Amazon ECS container agent configuration variable ECS_CONTAINER_START_TIMEOUT is used by default. If neither the startTimeout parameter or the ECS_CONTAINER_START_TIMEOUT agent configuration variable are set, then the default values of 3 minutes for Linux containers and 8 minutes on Windows containers are used. Your container instances require at least version 1.26.0 of the container agent to enable a container start timeout value. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the Amazon Elastic Container Service Developer Guide. If you are using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the ecs-init package. If your container instances are launched from version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

Type: Integer

Required: No

stopTimeout

Time duration (in seconds) to wait before the container is forcefully killed if it doesn't exit normally on its own.

For tasks using the Fargate launch type, the task or service requires platform version 1.3.0 or later. The max stop timeout value is 120 seconds and if the parameter is not specified, the default value of 30 seconds is used.

For tasks using the EC2 launch type, if the stopTimeout parameter is not specified, the value set for the Amazon ECS container agent configuration variable ECS_CONTAINER_STOP_TIMEOUT is used by default. If neither the stopTimeout parameter or the ECS_CONTAINER_STOP_TIMEOUT agent configuration variable are set, then the default values of 30 seconds for Linux containers and 30 seconds on Windows containers are used. Your container instances require at least version 1.26.0 of the container agent to enable a container stop timeout value. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the Amazon Elastic Container Service Developer Guide. If you are using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the ecs-init package. If your container instances are launched from version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.
Type: Integer
Required: No

**systemControls**

A list of namespaced kernel parameters to set in the container. This parameter maps to `Sysctl`s in the Create a container section of the Docker Remote API and the `--sysctl` option to `docker run`.

**Note**

It is not recommended that you specify network-related `systemControls` parameters for multiple containers in a single task that also uses either the `awsVpc` or `host` network modes. For tasks that use the `awsVpc` network mode, the container that is started last determines which `systemControls` parameters take effect. For tasks that use the `host` network mode, it changes the container instance's namespaced kernel parameters as well as the containers.

Type: Array of `SystemControl` (p. 343) objects

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` (p. 366) 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 --format '{{.Server.APIVersion}}'`

**Note**

This parameter is not supported for Windows containers.

Type: Array of `Ulimit` (p. 366) 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`.

You can use the following formats. If specifying a UID or GID, you must specify it as a positive integer.

- `user`
- `user:group`
- `uid`
- `uid:gid`
- `user:gid`
- `uid:group`

**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. 370) 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 V3
ContainerDependency

The dependencies defined for container startup and shutdown. A container can contain multiple dependencies. When a dependency is defined for container startup, for container shutdown it is reversed.

Your Amazon ECS container instances require at least version 1.26.0 of the container agent to enable container dependencies. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the Amazon Elastic Container Service Developer Guide. If you are using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the ecs-init package. If your container instances are launched from version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

Note
For tasks using the Fargate launch type, this parameter requires that the task or service uses platform version 1.3.0 or later.

Contents

condition

The dependency condition of the container. The following are the available conditions and their behavior:

- **START** - This condition emulates the behavior of links and volumes today. It validates that a dependent container is started before permitting other containers to start.
- **COMPLETE** - This condition validates that a dependent container runs to completion (exits) before permitting other containers to start. This can be useful for nonessential containers that run a script and then exit.
- **SUCCESS** - This condition is the same as **COMPLETE**, but it also requires that the container exits with a zero status.
- **HEALTHY** - This condition validates that the dependent container passes its Docker health check before permitting other containers to start. This requires that the dependent container has health checks configured. This condition is confirmed only at task startup.

Type: String

Valid Values: START | COMPLETE | SUCCESS | HEALTHY

Required: Yes

containerName

The name of a container.

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 V3
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. Only instances connected to an agent can 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 resources attached to a container instance, such as elastic network interfaces.

Type: Array of Attachment (p. 251) 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. 146) operation.

Type: Array of Attribute (p. 253) objects
Required: No

capacityProviderName

The capacity provider associated with the container instance.

Type: String
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
**ec2InstanceId**

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 timestamp for when the container instance was registered.

Type: Timestamp

Required: No

**registeredResources**

For CPU and memory resource types, this parameter describes the amount of each resource that was available on the container instance when the container agent registered it with Amazon ECS. This value represents the total amount of CPU and memory that can be allocated on this container instance to 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. 329) objects

Required: No

**remainingResources**

For CPU and memory resource types, this parameter describes the remaining CPU and memory that has not already been allocated to tasks and is therefore available for new tasks. For port resource types, this parameter describes the ports that were reserved by the Amazon ECS container agent (at instance registration time) and any task containers that have reserved port mappings on the host (with the host or bridge network mode). Any port that is not specified here is available for new tasks.

Type: Array of Resource (p. 329) 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 REGISTERING, REGISTRATION_FAILED, ACTIVE, INACTIVE, DEREGISTERING, or DRAINING.

If your account has opted in to the awsvpcTrunking account setting, then any newly registered container instance will transition to a REGISTERING status while the trunk elastic network...
interface is provisioned for the instance. If the registration fails, the instance will transition to a REGISTRATION_FAILED status. You can describe the container instance and see the reason for failure in the statusReason parameter. Once the container instance is terminated, the instance transitions to a Deregistering status while the trunk elastic network interface is deprovisioned. The instance then transitions to an INACTIVE status.

The ACTIVE status indicates that the container instance can accept tasks. The 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

statusReason

The reason that the container instance reached its current status.

Type: String
Required: No

tags

The metadata that you apply to the container instance to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.
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](#) 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 V3
ContainerOverride

The overrides that should be sent to a container. An empty container override can be passed in. An example of an empty container override would be `{"containerOverrides": [ ]}`. If a non-empty container override is specified, the name parameter must be included.

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. 308) 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
resourceRequirements

The type and amount of a resource to assign to a container, instead of the default value from the task definition. The only supported resource is a GPU.

Type: Array of ResourceRequirement (p. 331) 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 V3
ContainerStateChange

An object representing a change in state for a container.

Contents

c-containerName
  The name of the container.
  Type: String
  Required: No

c-exitCode
  The exit code for the container, if the state change is a result of the container exiting.
  Type: Integer
  Required: No

c-imageDigest
  The container image SHA 256 digest.
  Type: String
  Required: No

c-networkBindings
  Any network bindings associated with the container.
  Type: Array of NetworkBinding (p. 318) objects
  Required: No

c-reason
  The reason for the state change.
  Type: String
  Required: No

c-runtimeId
  The ID of the Docker container.
  Type: String
  Required: No

c-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 V3
Deployment

The details of an Amazon ECS service deployment. This is used only when a service uses the ECS deployment controller type.

Contents

capacityProviderStrategy

The capacity provider strategy that the deployment is using.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

createdAt

The Unix timestamp for when the service deployment 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 the tasks in the service are using. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

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. 319) 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 tasks in the service are running. A platform version is only specified for tasks using the Fargate launch type. If one is not specified, the LATEST platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

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. The following describes each state:

- PRIMARY
  - The most recent deployment of a service.

- ACTIVE
  - A service deployment that still has running tasks, but are in the process of being replaced with a new PRIMARY deployment.

- INACTIVE
  - A deployment that has been completely replaced.

Type: String

Required: No

**taskDefinition**

The most recent task definition that was specified for the tasks in the service to use.

Type: String

Required: No

**updatedAt**

The Unix timestamp for when the service deployment 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 V3
DeploymentConfiguration

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

Contents

maximumPercent

If a service is using the rolling update (ECS) deployment type, the maximum percent parameter
represents an upper limit on the number of tasks in a service that are allowed in the RUNNING or
PENDING state during a deployment, as a percentage of the desired number of tasks (rounded
down to the nearest integer), and while any container instances are in the DRAINING state if
the service contains tasks using the EC2 launch type. This parameter enables you to define the
deployment batch size. For example, if your service has a desired number of four tasks and a
maximum percent value of 200%, the scheduler may 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 maximum percent is 200%.

If a service is using the blue/green (CODE_DEPLOY) or EXTERNAL deployment types and tasks that
use the EC2 launch type, the maximum percent value is set to the default value and is used to
define the upper limit on the number of the tasks in the service that remain in the RUNNING state
while the container instances are in the DRAINING state. If the tasks in the service use the Fargate
launch type, the maximum percent value is not used, although it is returned when describing your
service.

Type: Integer

Required: No

minimumHealthyPercent

If a service is using the rolling update (ECS) deployment type, the minimum healthy percent parameter
represents a lower limit on the number of tasks in a service that must remain in the RUNNING state
during a deployment, as a percentage of the desired number of tasks (rounded up to the nearest
integer), and while any container instances are in the DRAINING state if the service contains tasks
using the EC2 launch type. This parameter enables you to deploy without using additional cluster
capacity. For example, if your service has a desired number of four tasks and a minimum healthy
percent of 50%, the scheduler may 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 they are reported as healthy by the load balancer. The default value for
minimum healthy percent is 100%.

If a service is using the blue/green (CODE_DEPLOY) or EXTERNAL deployment types and tasks that
use the EC2 launch type, the minimum healthy percent value is set to the default value and is used to
define the lower limit on the number of the tasks in the service that remain in the RUNNING state
while the container instances are in the DRAINING state. If the tasks in the service use the Fargate
launch type, the minimum healthy percent value is not used, although it is returned when describing
your service.

Type: Integer

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for Ruby V3
DeploymentController

The deployment controller to use for the service. For more information, see Amazon ECS Deployment Types in the Amazon Elastic Container Service Developer Guide.

Contents

type

The deployment controller type to use.

There are three deployment controller types available:

ECS

The rolling update (ECS) deployment type involves replacing the current running version of the container with the latest version. The number of containers Amazon ECS adds or removes from the service during a rolling update is controlled by adjusting the minimum and maximum number of healthy tasks allowed during a service deployment, as specified in the DeploymentConfiguration (p. 291).

CODE_DEPLOY

The blue/green (CODE_DEPLOY) deployment type uses the blue/green deployment model powered by AWS CodeDeploy, which allows you to verify a new deployment of a service before sending production traffic to it.

EXTERNAL

The external (EXTERNAL) deployment type enables you to use any third-party deployment controller for full control over the deployment process for an Amazon ECS service.

Type: String

Valid Values: ECS | CODE_DEPLOY | EXTERNAL

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 V3
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 V3
DockerVolumeConfiguration

This parameter is specified when you are using Docker volumes. Docker volumes are only supported when you are using the EC2 launch type. Windows containers only support the use of the local driver. To use bind mounts, specify a host instead.

Contents

autoprovision

If this value is true, the Docker volume is created if it does not already exist.

Note
This field is only used if the scope is shared.

Type: Boolean
Required: No

driver

The Docker volume driver to use. The driver value must match the driver name provided by Docker because it is used for task placement. If the driver was installed using the Docker plugin CLI, use docker plugin ls to retrieve the driver name from your container instance. If the driver was installed using another method, use Docker plugin discovery to retrieve the driver name. For more information, see Docker plugin discovery. This parameter maps to Driver in the Create a volume section of the Docker Remote API and the --driver option to docker volume create.

Type: String
Required: No

driverOpts

A map of Docker driver-specific options passed through. This parameter maps to DriverOpts in the Create a volume section of the Docker Remote API and the --opt option to docker volume create.

Type: String to string map
Required: No

labels

Custom metadata to add to your Docker volume. This parameter maps to Labels in the Create a volume section of the Docker Remote API and the --label option to docker volume create.

Type: String to string map
Required: No

scope

The scope for the Docker volume that determines its lifecycle. Docker volumes that are scoped to a task are automatically provisioned when the task starts and destroyed when the task stops. Docker volumes that are scoped as shared persist after the task stops.

Type: String
Valid Values: task | shared
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 V3
EFSVolumeConfiguration

This parameter is specified when you are using an Amazon Elastic File System (Amazon EFS) file storage. Amazon EFS file systems are only supported when you are using the EC2 launch type.

**Important**
EFSVolumeConfiguration remains in preview and is a Beta Service as defined by and subject to the Beta Service Participation Service Terms located at https://aws.amazon.com/service-terms ("Beta Terms"). These Beta Terms apply to your participation in this preview of EFSVolumeConfiguration.

**Contents**

**fileSystemId**

The Amazon EFS file system ID to use.

Type: String

Required: Yes

**rootDirectory**

The directory within the Amazon EFS file system to mount as the root directory inside the host.

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 V3
Failure

A failed resource.

Contents

arn
The Amazon Resource Name (ARN) of the failed resource.
Type: String
Required: No

detail
The details of the failure.
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 V3
FirelensConfiguration

The FireLens configuration for the container. This is used to specify and configure a log router for container logs. For more information, see Custom Log Routing in the Amazon Elastic Container Service Developer Guide.

Contents

options

The options to use when configuring the log router. This field is optional and can be used to specify a custom configuration file or to add additional metadata, such as the task, task definition, cluster, and container instance details to the log event. If specified, the syntax to use is "options": "{"enable-ecs-log-metadata": "true|false","config-file-type": "s3|file","config-file-value": "arn:aws:s3:::mybucket/fluent.conf|filepath"}". For more information, see Creating a Task Definition that Uses a FireLens Configuration in the Amazon Elastic Container Service Developer Guide.

Type: String to string map

Required: No

type

The log router to use. The valid values are fluentd or fluentbit.

Type: String

Valid Values: fluentd | fluentbit

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 V3
HealthCheck

An object representing a container health check. Health check parameters that are specified in a container definition override any Docker health checks that exist in the container image (such as those specified in a parent image or from the image's Dockerfile).

You can view the health status of both individual containers and a task with the DescribeTasks API operation or when viewing the task details in the console.

The following describes the possible healthStatus values for a container:

- **HEALTHY**—The container health check has passed successfully.
- **UNHEALTHY**—The container health check has failed.
- **UNKNOWN**—The container health check is being evaluated or there is no container health check defined.

The following describes the possible healthStatus values for a task. The container health check status of nonessential containers do not have an effect on the health status of a task.

- **HEALTHY**—All essential containers within the task have passed their health checks.
- **UNHEALTHY**—One or more essential containers have failed their health check.
- **UNKNOWN**—The essential containers within the task are still having their health checks evaluated or there are no container health checks defined.

If a task is run manually, and not as part of a service, the task will continue its lifecycle regardless of its health status. For tasks that are part of a service, if the task reports as unhealthy then the task will be stopped and the service scheduler will replace it.

The following are notes about container health check support:

- Container health checks require version 1.17.0 or greater of the Amazon ECS container agent. For more information, see Updating the Amazon ECS Container Agent.
- Container health checks are supported for Fargate tasks if you are using platform version 1.1.0 or greater. For more information, see AWS Fargate Platform Versions.
- Container health checks are not supported for tasks that are part of a service that is configured to use a Classic Load Balancer.

**Contents**

command

A string array representing the command that the container runs to determine if it is healthy. The string array must start with CMD to execute the command arguments directly, or CMD-SHELL to run the command with the container's default shell. For example:

```
[ "CMD-SHELL", "curl -f http://localhost/ || exit 1" ]
```

An exit code of 0 indicates success, and non-zero exit code indicates failure. For more information, see `HealthCheck` in the Create a container section of the Docker Remote API.

Type: Array of strings

Required: Yes
interval

The time period in seconds between each health check execution. You may specify between 5 and 300 seconds. The default value is 30 seconds.

Type: Integer

Required: No

retries

The number of times to retry a failed health check before the container is considered unhealthy. You may specify between 1 and 10 retries. The default value is 3.

Type: Integer

Required: No

startPeriod

The optional grace period within which to provide containers time to bootstrap before failed health checks count towards the maximum number of retries. You may specify between 0 and 300 seconds. The startPeriod is disabled by default.

Note

If a health check succeeds within the startPeriod, then the container is considered healthy and any subsequent failures count toward the maximum number of retries.

Type: Integer

Required: No

timeout

The time period in seconds to wait for a health check to succeed before it is considered a failure. You may specify between 2 and 60 seconds. The default value is 5.

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 V3
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. 267).

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 V3
HostVolumeProperties

Details on a container instance bind mount host volume.

Contents

sourcePath

When the host parameter is used, specify a sourcePath to declare 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 V3
InferenceAccelerator

Details on a Elastic Inference accelerator. For more information, see Working with Amazon Elastic Inference on Amazon ECS in the Amazon Elastic Container Service Developer Guide.

Contents

deviceName

The Elastic Inference accelerator device name. The deviceName must also be referenced in a container definition as a ResourceRequirement (p. 331).

Type: String
Required: Yes
deviceType

The Elastic Inference accelerator type to use.

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 V3
InferenceAcceleratorOverride

Details on an Elastic Inference accelerator task override. This parameter is used to override the Elastic Inference accelerator specified in the task definition. For more information, see Working with Amazon Elastic Inference on Amazon ECS in the Amazon Elastic Container Service Developer Guide.

Contents

deviceName

The Elastic Inference accelerator device name to override for the task. This parameter must match a deviceName specified in the task definition.

Type: String
Required: No

deviceType

The Elastic Inference accelerator type to use.

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 V3
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.

Note

If you are using tasks that use the Fargate launch type, the add parameter is not supported.

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 V3
KeyValuePair

A key-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 V3
LinuxParameters

Linux-specific options that are applied to the container, such as Linux KernelCapabilities (p. 306).

Contents

capabilities

The Linux capabilities for the container that are added to or dropped from the default configuration provided by Docker.

Note

If you are using tasks that use the Fargate launch type, capabilities is supported but the add parameter is not supported.

Type: KernelCapabilities (p. 306) 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.

Note

If you are using tasks that use the Fargate launch type, the devices parameter is not supported.

Type: Array of Device (p. 294) 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 --format '{{.Server.APIVersion}}'

Type: Boolean

Required: No

maxSwap

The total amount of swap memory (in MiB) a container can use. This parameter will be translated to the --memory=swap option to docker run where the value would be the sum of the container memory plus the maxSwap value.

If a maxSwap value of 0 is specified, the container will not use swap. Accepted values are 0 or any positive integer. If the maxSwap parameter is omitted, the container will use the swap configuration for the container instance it is running on. A maxSwap value must be set for the swappiness parameter to be used.

Note

If you are using tasks that use the Fargate launch type, the maxSwap parameter is not supported.

Type: Integer

Required: No
sharedMemorySize

The value for the size (in MiB) of the /dev/shm volume. This parameter maps to the --shm-size option to docker run.

Note
If you are using tasks that use the Fargate launch type, the sharedMemorySize parameter is not supported.

Type: Integer
Required: No

swappiness

This allows you to tune a container's memory swappiness behavior. A swappiness value of 0 will cause swapping to not happen unless absolutely necessary. A swappiness value of 100 will cause pages to be swapped very aggressively. Accepted values are whole numbers between 0 and 100. If the swappiness parameter is not specified, a default value of 60 is used. If a value is not specified for maxSwap then this parameter is ignored. This parameter maps to the --memory-swappiness option to docker run.

Note
If you are using tasks that use the Fargate launch type, the swappiness parameter is not supported.

Type: Integer
Required: No

tmpfs

The container path, mount options, and size (in MiB) of the tmpfs mount. This parameter maps to the --tmpfs option to docker run.

Note
If you are using tasks that use the Fargate launch type, the tmpfs parameter is not supported.

Type: Array of Tmpfs (p. 365) 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 V3
LoadBalancer

The load balancer configuration to use with a service or task set.

For specific notes and restrictions regarding the use of load balancers with services and task sets, see the CreateService and CreateTaskSet actions.

Contents

ccontAINERNAME

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 task definition the tasks in the service are using. For tasks that use the EC2 launch type, the container instance they are launched on must allow ingress traffic on the hostPort of the port mapping.

Type: Integer

Required: No

loadBalancerName

The name of the load balancer to associate with the Amazon ECS service or task set.

A load balancer name is only specified when using a Classic Load Balancer. If you are using an Application Load Balancer or a Network Load Balancer the load balancer name parameter should be omitted.

Type: String

Required: No

targetGroupArn

The full Amazon Resource Name (ARN) of the Elastic Load Balancing target group or groups associated with a service or task set.

A target group ARN is only specified when using an Application Load Balancer or Network Load Balancer. If you are using a Classic Load Balancer the target group ARN should be omitted.

For services using the ECS deployment controller, you can specify one or multiple target groups. For more information, see Registering Multiple Target Groups with a Service in the Amazon Elastic Container Service Developer Guide.

For services using the CODE_DEPLOY deployment controller, you are required to define two target groups for the load balancer. For more information, see Blue/Green Deployment with CodeDeploy in the Amazon Elastic Container Service Developer Guide.

Important

If your service's task definition uses the awsvpc network mode (which is required for the Fargate launch type), you must choose ip as the target type, not instance, when creating
your target groups because tasks that use the *aws-vpc* network mode are associated with an elastic network interface, not an Amazon EC2 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 V3
LogConfiguration

The log configuration specification for the container.

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.

The following should be noted when specifying a log configuration for your containers:

- Amazon ECS currently supports a subset of the logging drivers available to the Docker daemon (shown in the valid values below). 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.
- For tasks using the EC2 launch type, 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.
- For tasks using the Fargate launch type, because you do not have access to the underlying infrastructure your tasks are hosted on, any additional software needed will have to be installed outside of the task. For example, the Fluentd output aggregators or a remote host running Logstash to send Gelf logs to.

Contents

logDriver

The log driver to use for the container. The valid values listed earlier are log drivers that the Amazon ECS container agent can communicate with by default.

For tasks using the Fargate launch type, the supported log drivers are awslogs, splunk, and awsfirelens.

For tasks using the EC2 launch type, the supported log drivers are awslogs, fluentd, gelf, json-file, journald, logentries, syslog, splunk, and awsfirelens.

For more information about using the awslogs log driver, see Using the awslogs Log Driver in the Amazon Elastic Container Service Developer Guide.

For more information about using the awsfirelens log driver, see Custom Log Routing in the Amazon Elastic Container Service Developer Guide.

Note

If you have a custom driver that is not listed, 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, we do not currently provide support for running modified copies of this software.

Type: String

Valid Values: json-file | syslog | journald | gelf | fluentd | awslogs | splunk | awsfirelens
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 --format '{{.Server.APIVersion}}'`

Type: String to string map

Required: No

**secretOptions**

The secrets to pass to the log configuration. For more information, see Specifying Sensitive Data in the Amazon Elastic Container Service Developer Guide.

Type: Array of Secret (p. 333) 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 V3
ManagedScaling

The managed scaling settings for the Auto Scaling group capacity provider.

When managed scaling is enabled, Amazon ECS manages the scale-in and scale-out actions of the Auto Scaling group. Amazon ECS manages a target tracking scaling policy using an Amazon ECS-managed CloudWatch metric with the specified targetCapacity value as the target value for the metric. For more information, see Using Managed Scaling in the Amazon Elastic Container Service Developer Guide.

If managed scaling is disabled, the user must manage the scaling of the Auto Scaling group.

Contents

maximumScalingStepSize

The maximum number of container instances that Amazon ECS will scale in or scale out at one time. If this parameter is omitted, the default value of 10000 is used.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

minimumScalingStepSize

The minimum number of container instances that Amazon ECS will scale in or scale out at one time. If this parameter is omitted, the default value of 1 is used.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

status

Whether or not to enable managed scaling for the capacity provider.

Type: String

Valid Values: ENABLED | DISABLED

Required: No

targetCapacity

The target capacity value for the capacity provider. The specified value must be greater than 0 and less than or equal to 100. A value of 100 will result in the Amazon EC2 instances in your Auto Scaling group being completely utilized.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

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 V3
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. Must be a volume name referenced in the name parameter of task definition volume.

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 V3
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. 94) API responses.

Contents

bindIP

The IP address that the container is bound to on the container instance.

Type: String

Required: No

ccontainerPort

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 V3
NetworkConfiguration

An object representing the network configuration for a task or service.

Contents

awsVpcConfiguration

The VPC subnets and security groups associated with a task.

Note
All specified subnets and security groups must be from the same VPC.

Type: AwsVpcConfiguration (p. 255) 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 V3
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 V3
PlacementConstraint

An object representing a constraint on task placement. For more information, see Task Placement Constraints in the Amazon Elastic Container Service Developer Guide.

Note
If you are using the Fargate launch type, task placement constraints are not supported.

Contents

expression

A cluster query language expression to apply to the constraint. 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.

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 V3
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 V3
PlatformDevice

The devices that are available on the container instance. The only supported device type is a GPU.

Contents

id

The ID for the GPU(s) on the container instance. The available GPU IDs can also be obtained on the container instance in the /var/lib/ecs/gpu/nvidia_gpu_info.json file.

Type: String
Required: Yes

type

The type of device that is available on the container instance. The only supported value is GPU.

Type: String
Valid Values: GPU
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 V3
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 you are 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 (p. 94) API responses.

Contents

containerPort

The port number on the container that is bound to the user-specified or automatically assigned host port.

If you are using containers in a task with the awsvpc or host network mode, exposed ports should be specified using containerPort.

If you are 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.

Important

You cannot expose the same container port for multiple protocols. An error will be returned if this is attempted.

Type: Integer

Required: No

hostPort

The port number on the container instance to reserve for your container.

If you are using containers in a task with the awsvpc or host network mode, the hostPort can either be left blank or set to the same value as the containerPort.

If you are 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. Do 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 51678-51680. 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 \texttt{remainingResources} of \texttt{DescribeContainerInstances} output. A container instance can have up to 100 reserved ports at a time, including the default reserved ports. Automatically assigned ports don't count toward the 100 reserved ports limit.

Type: Integer

Required: No

\textbf{protocol}

The protocol used for the port mapping. Valid values are \texttt{tcp} and \texttt{udp}. The default is \texttt{tcp}.

Type: String

Valid Values: \texttt{tcp} | \texttt{udp}

Required: No

\section*{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 V3
ProxyConfiguration

The configuration details for the App Mesh proxy.

For tasks using the EC2 launch type, the container instances require at least version 1.26.0 of the container agent and at least version 1.26.0-1 of the `ecs-init` package to enable a proxy configuration. If your container instances are launched from the Amazon ECS-optimized AMI version 20190301 or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

For tasks using the Fargate launch type, the task or service requires platform version 1.3.0 or later.

Contents

**containerName**

The name of the container that will serve as the App Mesh proxy.

Type: String

Required: Yes

**properties**

The set of network configuration parameters to provide the Container Network Interface (CNI) plugin, specified as key-value pairs.

- **IgnoredUID** - (Required) The user ID (UID) of the proxy container as defined by the `user` parameter in a container definition. This is used to ensure the proxy ignores its own traffic. If `IgnoredGID` is specified, this field can be empty.

- **IgnoredGID** - (Required) The group ID (GID) of the proxy container as defined by the `user` parameter in a container definition. This is used to ensure the proxy ignores its own traffic. If `IgnoredUID` is specified, this field can be empty.

- **AppPorts** - (Required) The list of ports that the application uses. Network traffic to these ports is forwarded to the `ProxyIngressPort` and `ProxyEgressPort`.

- **ProxyIngressPort** - (Required) Specifies the port that incoming traffic to the `AppPorts` is directed to.

- **ProxyEgressPort** - (Required) Specifies the port that outgoing traffic from the `AppPorts` is directed to.

- **EgressIgnoredPorts** - (Required) The egress traffic going to the specified ports is ignored and not redirected to the `ProxyEgressPort`. It can be an empty list.

- **EgressIgnoredIPs** - (Required) The egress traffic going to the specified IP addresses is ignored and not redirected to the `ProxyEgressPort`. It can be an empty list.

Type: Array of `KeyValuePair` objects

Required: No

**type**

The proxy type. The only supported value is `APPMESH`.

Type: String

Valid Values: `APPMESH`

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 V3
RepositoryCredentials

The repository credentials for private registry authentication.

Contents

credentialsParameter

The Amazon Resource Name (ARN) of the secret containing the private repository credentials.

**Note**
When you are using the Amazon ECS API, AWS CLI, or AWS SDK, if the secret exists in the same Region as the task that you are launching then you can use either the full ARN or the name of the secret. When you are using the AWS Management Console, you must specify the full ARN of the secret.

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 V3
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, PORTS_UDP, 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++
• AWS SDK for Go
• AWS SDK for Java
• AWS SDK for Ruby V3
ResourceRequirement

The type and amount of a resource to assign to a container. The supported resource types are GPUs and Elastic Inference accelerators. For more information, see Working with GPUs on Amazon ECS or Working with Amazon Elastic Inference on Amazon ECS in the Amazon Elastic Container Service Developer Guide.

Contents

type

The type of resource to assign to a container. The supported values are GPU or InferenceAccelerator.

Type: String

Valid Values: GPU | InferenceAccelerator

Required: Yes

value

The value for the specified resource type.

If the GPU type is used, the value is the number of physical GPUs the Amazon ECS container agent will reserve for the container. The number of GPUs reserved for all containers in a task should not exceed the number of available GPUs on the container instance the task is launched on.

If the InferenceAccelerator type is used, the value should match the deviceName for an InferenceAccelerator (p. 304) specified in a task definition.

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 V3
Scale

A floating-point percentage of the desired number of tasks to place and keep running in the task set.

Contents

unit

The unit of measure for the scale value.

Type: String

Valid Values: PERCENT

Required: No

value

The value, specified as a percent total of a service's desiredCount, to scale the task set. Accepted values are numbers between 0 and 100.

Type: Double

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 V3
Secret

An object representing the secret to expose to your container. Secrets can be exposed to a container in the following ways:

- To inject sensitive data into your containers as environment variables, use the `secrets` container definition parameter.
- To reference sensitive information in the log configuration of a container, use the `secretOptions` container definition parameter.

For more information, see [Specifying Sensitive Data](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data.html) in the *Amazon Elastic Container Service Developer Guide*.

Contents

name

The name of the secret.

Type: String

Required: Yes

valueFrom

The secret to expose to the container. The supported values are either the full ARN of the AWS Secrets Manager secret or the full ARN of the parameter in the AWS Systems Manager Parameter Store.

Note

If the AWS Systems Manager Parameter Store parameter exists in the same Region as the task you are launching, then you can use either the full ARN or name of the parameter. If the parameter exists in a different Region, then the full ARN must be specified.

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++](https://aws.amazon.com/sdk-for-cpp/)
- [AWS SDK for Go](https://aws.amazon.com/sdk-for-go/)
- [AWS SDK for Java](https://aws.amazon.com/sdk-for-java/)
- [AWS SDK for Ruby V3](https://aws.amazon.com/sdk-for-ruby/)

API Version 2014-11-13

333
Service

Details on a service within a cluster

Contents

capacityProviderStrategy

The capacity provider strategy associated with the service.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

clusterArn

The Amazon Resource Name (ARN) of the cluster that hosts the service.

Type: String

Required: No

createdAt

The Unix timestamp for when the service was created.

Type: Timestamp

Required: No

createdBy

The principal that created the service.

Type: String

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. 291) object

Required: No

deploymentController

The deployment controller type the service is using. When using the DescribeServices API, this field is omitted if the service is using the ECS deployment controller type.

Type: DeploymentController (p. 293) object

Required: No

deployments

The current state of deployments for the service.

Type: Array of Deployment (p. 288) 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. 12), and it can be modified with UpdateService (p. 231).

Type: Integer

Required: No

disableECSManagedTags

Specifies whether to enable Amazon ECS managed tags for the tasks in the service. For more information, see Tagging Your Amazon ECS Resources in the Amazon Elastic Container Service Developer Guide.

Type: Boolean

Required: No

events

The event stream for your service. A maximum of 100 of the latest events are displayed.

Type: Array of ServiceEvent (p. 339) objects

Required: No

healthCheckGracePeriodSeconds

The period of time, in seconds, that the Amazon ECS service scheduler ignores unhealthy Elastic Load Balancing target health checks after a task has first started.

Type: Integer

Required: No

launchType

The launch type on which your service is running. If no value is specified, it will default to EC2. Valid values include EC2 and FARGATE. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

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. 311) 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. 319) 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. 321) objects

Required: No

**placementStrategy**

The placement strategy that determines how tasks for the service are placed.

Type: Array of PlacementStrategy (p. 322) objects

Required: No

**platformVersion**

The platform version on which to run your service. A platform version is only specified for tasks using the Fargate launch type. If one is not specified, the LATEST platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

**propagateTags**

Specifies whether to propagate the tags from the task definition or the service to the task. If no value is specified, the tags are not propagated.

Type: String

Valid Values: TASK_DEFINITION | SERVICE

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

Required: No

**schedulingStrategy**

The scheduling strategy to use for the service. For more information, see Services.
There are two service scheduler strategies available:

- **REPLICA** - The replica scheduling strategy places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions.

- **DAEMON** - The daemon scheduling strategy deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks and will stop tasks that do not meet the placement constraints.

  **Note**
  
  Fargate tasks do not support the **DAEMON** scheduling strategy.

**serviceScheduler**

Type: String

Valid Values: REPLICA | DAEMON

Required: No

**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, and hyphens 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

**serviceRegistries**

The details of the service discovery registries to assign to this service. For more information, see Service Discovery.

Type: Array of ServiceRegistry (p. 340) objects

Required: No

**status**

The status of the service. The valid values are ACTIVE, DRAINING, or INACTIVE.

Type: String

Required: No

**tags**

The metadata that you apply to the service to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
• For each resource, each tag key must be unique, and each tag key can have only one value.
• Maximum key length - 128 Unicode characters in UTF-8
• Maximum value length - 256 Unicode characters in UTF-8
• If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
• Tag keys and values are case-sensitive.
• Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

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. 12), and it can be modified with UpdateService (p. 231).

Type: String

Required: No

taskSets

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Type: Array of TaskSet (p. 360) 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 V3
ServiceEvent

Details on an event associated with a service.

Contents

createdAt

The Unix timestamp 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 V3
ServiceRegistry

Details of the service registry.

Contents

countainerName

The container name value, already specified in the task definition, to be used for your service discovery service. If the task definition that your service task specifies uses the bridge or host network mode, you must specify a containerName and containerPort combination from the task definition. If the task definition that your service task specifies uses the awsvpc network mode and a type SRV DNS record is used, you must specify either a containerName and containerPort combination or a port value, but not both.

Type: String
Required: No

countainerPort

The port value, already specified in the task definition, to be used for your service discovery service. If the task definition your service task specifies uses the bridge or host network mode, you must specify a containerName and containerPort combination from the task definition. If the task definition your service task specifies uses the awsvpc network mode and a type SRV DNS record is used, you must specify either a containerName and containerPort combination or a port value, but not both.

Type: Integer
Required: No

port

The port value used if your service discovery service specified an SRV record. This field may be used if both the awsvpc network mode and SRV records are used.

Type: Integer
Required: No

registryArn

The Amazon Resource Name (ARN) of the service registry. The currently supported service registry is AWS Cloud Map. For more information, see CreateService.

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 V3
Setting

The current account setting for a resource.

Contents

name

The Amazon ECS resource name.

Type: String

Valid Values: serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights

Required: No

principalArn

The ARN of the principal, which can be an IAM user, IAM role, or the root user. If this field is omitted, the authenticated user is assumed.

Type: String

Required: No

value

Whether the account setting is enabled or disabled for the specified resource.

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 V3
SystemControl

A list of namespaced kernel parameters to set in the container. This parameter maps to Sysctls in the Create a container section of the Docker Remote API and the --sysctl option to docker run.

It is not recommended that you specify network-related systemControls parameters for multiple containers in a single task that also uses either the awsvpc or host network mode for the following reasons:

- For tasks that use the awsvpc network mode, if you set systemControls for any container, it applies to all containers in the task. If you set different systemControls for multiple containers in a single task, the container that is started last determines which systemControls take effect.
- For tasks that use the host network mode, the systemControls parameter applies to the container instance's kernel parameter as well as that of all containers of any tasks running on that container instance.

Contents

namespace

The namespaced kernel parameter for which to set a value.

Type: String

Required: No

driver

value

The value for the namespaced kernel parameter specified in namespace.

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 V3
Tag

The metadata that you apply to a resource to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Contents

key

One part of a key-value pair that make up a tag. A key is a general label that acts like a category for more specific tag values.

Type: String


Pattern: ^([\p{L}\p{Z}\p{N}_.:/=+-@]*)$

Required: No

value

The optional part of a key-value pair that make up a tag. A value acts as a descriptor within a tag category (key).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: ^([\p{L}\p{Z}\p{N}_.:/=+-@]*)$

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 V3
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. 251) objects
  - Required: No
- attributes
  - The attributes of the task
  - Type: Array of Attribute (p. 253) objects
  - Required: No
- availabilityZone
  - The availability zone of the task.
  - Type: String
  - Required: No
- capacityProviderName
  - The capacity provider associated with the task.
  - Type: String
  - 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 timestamp 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. 264) objects
Required: No

cpu

The number of CPU units used by the task as expressed in a task definition. It can be expressed as an integer using CPU units, for example 1024. It can also be expressed as a string using vCPUs, for example 1 vCPU or 1 vcpu. String values are converted to an integer indicating the CPU units when the task definition is registered.

If you are using the EC2 launch type, this field is optional. Supported values are between 128 CPU units (0.125 vCPUs) and 10240 CPU units (10 vCPUs).

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 supported values for the memory parameter:

- 256 (.25 vCPU) - Available memory values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
- 512 (.5 vCPU) - Available memory values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
- 1024 (1 vCPU) - Available memory values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
- 2048 (2 vCPU) - Available memory values: Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
- 4096 (4 vCPU) - Available memory values: Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)

Type: String
Required: No

createdAt

The Unix timestamp for when the task was created (the task entered the PENDING state).

Type: Timestamp
Required: No

desiredStatus

The desired status of the task. For more information, see Task Lifecycle.

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

**healthStatus**

The health status for the task, which is determined by the health of the essential containers in the task. If all essential containers in the task are reporting as **HEALTHY**, then the task status also reports as **HEALTHY**. If any essential containers in the task are reporting as **UNHEALTHY** or **UNKNOWN**, then the task status also reports as **UNHEALTHY** or **UNKNOWN**, accordingly.

**Note**

The Amazon ECS container agent does not monitor or report on Docker health checks that are embedded in a container image (such as those specified in a parent image or from the image's Dockerfile) and not specified in the container definition. Health check parameters that are specified in a container definition override any Docker health checks that exist in the container image.

Type: String

Valid Values: HEALTHY | UNHEALTHY | UNKNOWN

Required: No

**inferenceAccelerators**

The Elastic Inference accelerator associated with the task.

Type: Array of InferenceAccelerator (p. 304) objects

Required: No

**lastStatus**

The last known status of the task. For more information, see Task Lifecycle.

Type: String

Required: No

**launchType**

The launch type on which your task is running. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

Type: String

Valid Values: EC2 | FARGATE

Required: No

**memory**

The amount of memory (in MiB) used by the task as expressed in a task definition. It can be expressed as an integer using MiB, for example 1024. It can also be expressed as a string using GB, for example 1GB or 1 GB. String values are converted to an integer indicating the MiB when the task definition is registered.

If you are using the EC2 launch type, this field is optional.
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 supported values for the `cpu` parameter:

- 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available `cpu` values: 256 (.25 vCPU)
- 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available `cpu` values: 512 (1 vCPU)
- 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available `cpu` values: 1024 (1 vCPU)
- Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB) - Available `cpu` values: 2048 (2 vCPU)
- Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB) - Available `cpu` values: 4096 (4 vCPU)

Type: String

Required: No

overrides

One or more container overrides.

Type: `TaskOverride (p. 358)` object

Required: No

platformVersion

The platform version on which your task is running. A platform version is only specified for tasks using the Fargate launch type. If one is not specified, the `LATEST` platform version is used by default. 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.
stopCode

The stop code indicating why a task was stopped. The stoppedReason may contain additional details.

Type: String

Valid Values: TaskFailedToStart | EssentialContainerExited | UserInitiated

stoppedAt

The Unix timestamp for when the task was stopped (the task transitioned from the RUNNING state to the STOPPED state).

Type: Timestamp

stoppedReason

The reason that the task was stopped.

Type: String

stoppingAt

The Unix timestamp for when the task stops (transitions from the RUNNING state to STOPPED).

Type: Timestamp

tags

The metadata that you apply to the task to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use aws:, AWS:, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

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 API actions 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 V3
TaskDefinition

The details of a task definition which describes the container and volume definitions of an Amazon Elastic Container Service task. You can specify which Docker images to use, the required resources, and other configurations related to launching the task definition through an Amazon ECS service or task.

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 (p. 267) objects

Required: No

cpu

The number of cpu units used by the task. If you are using the EC2 launch type, this field is optional and any value can be used. 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: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
• 512 (.5 vCPU) - Available memory values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
• 1024 (1 vCPU) - Available memory values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
• 2048 (2 vCPU) - Available memory values: Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
• 4096 (4 vCPU) - Available memory values: Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)

Type: String

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution 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

family

The name of a family that this task definition is registered to. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
A family groups multiple versions of a task definition. Amazon ECS gives the first task definition that you registered to a family a revision number of 1. Amazon ECS gives sequential revision numbers to each task definition that you add.

Type: String
Required: No

inferenceAccelerators

The Elastic Inference accelerator associated with the task.

Type: Array of InferenceAccelerator (p. 304) objects
Required: No

ipcMode

The IPC resource namespace to use for the containers in the task. The valid values are host, task, or none. If host is specified, then all containers within the tasks that specified the host IPC mode on the same container instance share the same IPC resources with the host Amazon EC2 instance. If task is specified, all containers within the specified task share the same IPC resources. If none is specified, then IPC resources within the containers of a task are private and not shared with other containers in a task or on the container instance. If no value is specified, then the IPC resource namespace sharing depends on the Docker daemon setting on the container instance. For more information, see IPC settings in the Docker run reference.

If the host IPC mode is used, be aware that there is a heightened risk of undesired IPC namespace expose. For more information, see Docker security.

If you are setting namespaced kernel parameters using systemControls for the containers in the task, the following will apply to your IPC resource namespace. For more information, see System Controls in the Amazon Elastic Container Service Developer Guide.

• For tasks that use the host IPC mode, IPC namespace related systemControls are not supported.
• For tasks that use the task IPC mode, IPC namespace related systemControls will apply to all containers within a task.

Note
This parameter is not supported for Windows containers or tasks using the Fargate launch type.

Type: String
Valid Values: host | task | none
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 a task-level memory value is specified then the container-level memory value is optional.

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:

• 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available cpu values: 256 (.25 vCPU)
• 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available cpu values: 512 (.5 vCPU)
• 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available cpu values: 1024 (1 vCPU)
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 you are using the Fargate launch type, the awsvpc network mode is required. If you are using the EC2 launch type, any network mode can be used. If the network mode is set to none, you cannot specify port mappings in your container definitions, and the tasks 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. 319) value 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 Amazon ECS-optimized AMIs, 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 cannot 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.

pidMode

The process namespace to use for the containers in the task. The valid values are host or task. If host is specified, then all containers within the tasks that specified the host PID mode on the same container instance share the same process namespace with the host Amazon EC2 instance. If task is specified, all containers within the specified task share the same process namespace. If no value is specified, the default is a private namespace. For more information, see PID settings in the Docker run reference.

Note

If the host PID mode is used, be aware that there is a heightened risk of undesired process namespace expose. For more information, see Docker security.
Type: String

Valid Values: host | task

Required: No

**placementConstraints**

An array of placement constraint objects to use for tasks. This field is not valid if you are using the Fargate launch type for your task.

Type: Array of TaskDefinitionPlacementConstraint (p. 357) objects

Required: No

**proxyConfiguration**

The configuration details for the App Mesh proxy.

Your Amazon ECS container instances require at least version 1.26.0 of the container agent and at least version 1.26.0-1 of the ecs-init package to enable a proxy configuration. If your container instances are launched from the Amazon ECS-optimized AMI version 20190301 or later, then they contain the required versions of the container agent and ecs-init. For more information, see Amazon ECS-optimized Linux AMI in the Amazon Elastic Container Service Developer Guide.

Type: ProxyConfiguration (p. 326) object

Required: No

**requiresAttributes**

The container instance attributes required by your task. This field is not valid if you are using the Fargate launch type for your task.

Type: Array of Attribute (p. 253) objects

Required: No

**requiresCompatibilities**

The launch type the task requires. If no value is specified, it will default to EC2. Valid values include EC2 and FARGATE.

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 that 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 short name or full Amazon Resource Name (ARN) of the AWS Identity and Access Management (IAM) role that grants containers in the task permission to call AWS APIs on your behalf. For more information, see [Amazon ECS Task Role](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task Stops.html) in the Amazon Elastic Container Service Developer Guide.

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](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/windows-task-steps.html) in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

**volumes**

The list of volume definitions for the task.

If your tasks 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 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 V3
TaskDefinitionPlacementConstraint

An object representing a constraint on task placement in the task definition. For more information, see Task Placement Constraints in the Amazon Elastic Container Service Developer Guide.

Note
If you are using the Fargate launch type, task placement constraints are not supported.

Contents

eexpression
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 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 V3
TaskOverride

The overrides associated with a task.

Contents

containerOverrides

One or more container overrides sent to a task.
Type: Array of ContainerOverride (p. 284) objects
Required: No

cpu

The cpu override for the task.
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

inferenceAcceleratorOverrides

The Elastic Inference accelerator override for the task.
Type: Array of InferenceAcceleratorOverride (p. 305) objects
Required: No

memory

The memory override for the task.
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 V3
TaskSet

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Contents

capacityProviderStrategy

The capacity provider strategy associated with the task set.

Type: Array of CapacityProviderStrategyItem (p. 258) objects

Required: No

clusterArn

The Amazon Resource Name (ARN) of the cluster that the service that hosts the task set exists in.

Type: String

Required: No

computedDesiredCount

The computed desired count for the task set. This is calculated by multiplying the service's desiredCount by the task set's scale percentage. The result is always rounded up. For example, if the computed desired count is 1.2, it rounds up to 2 tasks.

Type: Integer

Required: No

createdAt

The Unix timestamp for when the task set was created.

Type: Timestamp

Required: No

externalId

The external ID associated with the task set.

If a task set is created by an AWS CodeDeploy deployment, the externalId parameter contains the AWS CodeDeploy deployment ID.

If a task set is created for an external deployment and is associated with a service discovery registry, the externalId parameter contains the ECS_TASK_SET_EXTERNAL_ID AWS Cloud Map attribute.

Type: String

Required: No

id

The ID of the task set.

Type: String

Required: No
launchType

The launch type the tasks in the task set are using. For more information, see Amazon ECS Launch Types in the Amazon Elastic Container Service Developer Guide.

Type: String

Valid Values: EC2 | FARGATE

Required: No

loadBalancers

Details on a load balancer that is used with a task set.

Type: Array of LoadBalancer (p. 311) objects

Required: No

networkConfiguration

The network configuration for the task set.

Type: NetworkConfiguration (p. 319) object

Required: No

pendingCount

The number of tasks in the task set that are in the PENDING status during a deployment. A task in the PENDING state is preparing to enter the RUNNING state. A task set enters the PENDING status when it launches for the first time or when it is restarted after being in the STOPPED state.

Type: Integer

Required: No

platformVersion

The platform version on which the tasks in the task set are running. A platform version is only specified for tasks using the Fargate launch type. If one is not specified, the LATEST platform version is used by default. For more information, see AWS Fargate Platform Versions in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: No

runningCount

The number of tasks in the task set that are in the RUNNING status during a deployment. A task in the RUNNING state is running and ready for use.

Type: Integer

Required: No

scale

A floating-point percentage of the desired number of tasks to place and keep running in the task set.

Type: Scale (p. 332) object

Required: No
serviceArn

The Amazon Resource Name (ARN) of the service the task set exists in.

Type: String
Required: No

serviceRegistries

The details of the service discovery registries to assign to this task set. For more information, see Service Discovery.

Type: Array of ServiceRegistry (p. 340) objects
Required: No

stabilityStatus

The stability status, which indicates whether the task set has reached a steady state. If the following conditions are met, the task set will be in STEADY_STATE:
• The task runningCount is equal to the computedDesiredCount.
• The pendingCount is 0.
• There are no tasks running on container instances in the DRAINING status.
• All tasks are reporting a healthy status from the load balancers, service discovery, and container health checks.

If any of those conditions are not met, the stability status returns STABILIZING.

Type: String
Valid Values: STEADY_STATE | STABILIZING
Required: No

stabilityStatusAt

The Unix timestamp for when the task set stability status was retrieved.

Type: Timestamp
Required: No

startedBy

The tag specified when a task set is started. If the task set is created by an AWS CodeDeploy deployment, the startedBy parameter is CODE_DEPLOY. For a task set created for an external deployment, the startedBy field isn't used.

Type: String
Required: No

status

The status of the task set. The following describes each state:
PRIMAR

The task set is serving production traffic.

ACTIVE

The task set is not serving production traffic.
DRAINING

The tasks in the task set are being stopped and their corresponding targets are being
deregistered from their target group.

Type: String

Required: No

tags

The metadata that you apply to the task set to help you categorize and organize them. Each tag
consists of a key and an optional value, both of which you define.

The following basic restrictions apply to tags:
- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other
  services may have restrictions on allowed characters. Generally allowed characters are: letters,
  numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys
  or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix.
  Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag (p. 344) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskDefinition

The task definition the task set is using.

Type: String

Required: No

taskSetArn

The Amazon Resource Name (ARN) of the task set.

Type: String

Required: No

updatedAt

The Unix timestamp for when the task set 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 V3
Tmpfs

The container path, mount options, and size of the tmpfs mount.

Contents

containerPath

The absolute file path where the tmpfs volume is to be mounted.

Type: String

Required: Yes

mountOptions

The list of tmpfs volume mount options.

Valid values: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirmount" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relative" | "nrelative" | "strictatime" | "nstrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

Type: Array of strings

Required: No

size

The size (in MiB) of the tmpfs volume.

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 V3
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 | rtime | 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 V3
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-ecz-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 V3
Volume

A data volume used in a task definition. For tasks that use a Docker volume, specify a `DockerVolumeConfiguration`. For tasks that use a bind mount host volume, specify a `host` and optional `sourcePath`. For more information, see Using Data Volumes in Tasks.

Contents

dockerVolumeConfiguration

This parameter is specified when you are using Docker volumes. Docker volumes are only supported when you are using the EC2 launch type. Windows containers only support the use of the `local` driver. To use bind mounts, specify the `host` parameter instead.

Type: `DockerVolumeConfiguration (p. 295)` object

Required: No

efsVolumeConfiguration

This parameter is specified when you are using an Amazon Elastic File System (Amazon EFS) file storage. Amazon EFS file systems are only supported when you are using the EC2 launch type.

Important

`EFSVolumeConfiguration` remains in preview and is a Beta Service as defined by and subject to the Beta Service Participation Service Terms located at https://aws.amazon.com/service-terms/ (“Beta Terms”). These Beta Terms apply to your participation in this preview of `EFSVolumeConfiguration`.

Type: `EFSVolumeConfiguration (p. 297)` object

Required: No

host

This parameter is specified when you are using bind mount host volumes. Bind mount host volumes are supported when you are using either the EC2 or Fargate launch types. The contents of the `host` parameter determine whether your bind mount host 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. However, 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`.

Type: `HostVolumeProperties (p. 303)` object

Required: No

name

The name of the volume. Up to 255 letters (uppercase and lowercase), numbers, and hyphens 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 V3
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 from which to mount volumes.

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 V3
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

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400