EC2ContainerService

API Reference

API Version 2014-11-13

EC2ContainerService: API Reference

Table of Contents

Welcome

Amazon EC2 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 of Amazon EC2 instances. Amazon ECS lets you launch and stop container-enabled 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 like security groups, Amazon EBS volumes, and IAM roles.

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 EC2 Container Service 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 updated on September 3, 2015.

Actions

The following actions are supported:

- CreateCluster (p. 3)
- CreateService (p. 6)
- DeleteCluster (p. 11)
- DeleteService (p. 14)
- DeregisterContainerInstance (p. 18)
- DeregisterTaskDefinition (p. 23)
- DescribeClusters (p. 27)
- DescribeContainerInstances (p. 30)
- DescribeServices (p. 35)
- DescribeTaskDefinition (p. 39)
- DescribeTasks (p. 43)
- DiscoverPollEndpoint (p. 47)
- ListClusters (p. 49)
- ListContainerInstances (p. 52)
- ListServices (p. 55)
- ListTaskDefinitionFamilies (p. 58)
- ListTaskDefinitions (p. 62)
- ListTasks (p. 66)
- RegisterContainerInstance (p. 70)
- RegisterTaskDefinition (p. 73)
- RunTask (p. 79)
- StartTask (p. 84)
- StopTask (p. 89)
- SubmitContainerStateChange (p. 93)
- SubmitTaskStateChange (p. 95)
- UpdateContainerAgent (p. 97)
- UpdateService (p. 101)

CreateCluster

Creates a new Amazon ECS cluster. By default, your account will receive 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.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

clusterName

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

Type: String

Required: No

Response Syntax

```
{
    "cluster": {
        "activeServicesCount": number,
        "clusterArn": "string",
        "clusterName": "string",
        "pendingTasksCount": number,
        "registeredContainerInstancesCount": number,
        "runningTasksCount": number,
        "status": "string"
    }
}
```

Response Elements

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

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

cluster

The full description of your new cluster.

Type: Cluster (p. 106) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

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"
  }
}
```

CreateService

Runs and maintains a desired number of tasks from a specified task definition. If the number of tasks running in a service drops below `desiredCount`, Amazon ECS will spawn another instantiation of the task in the specified cluster.

Request Syntax

```
{
    "clientToken": "string",
    "cluster": "string",
    "desiredCount": number,
    "loadBalancers": [
        {
            "containerName": "string",
            "containerPort": number,
            "loadBalancerName": "string"
        }
    ],
    "role": "string",
    "serviceName": "string",
    "taskDefinition": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

clientToken

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

Type: String

Required: No

cluster

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

Type: String

Required: No

desiredCount

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

Type: Number

Required: Yes

loadBalancers

A list of 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. 114) objects

Required: No

role

The name or full Amazon Resource Name (ARN) of the IAM role that allows your Amazon ECS container agent to make calls to your load balancer on your behalf. This parameter is only required if you are using a load balancer with your service.

Type: String

Required: No

serviceName

The name of your service. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed. Service names must be unique within a cluster, but you can have similarly named services in multiple clusters within a region or across multiple regions.

Type: String

Required: Yes

taskDefinition

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

Type: String

Required: Yes

Response Syntax

```
{
    "service": {
        "clusterArn": "string",
        "deployments": [
            {
                "createdAt": number,
                "desiredCount": number,
                "id": "string",
                "pendingCount": number,
                "runningCount": number,
                "status": "string",
                "taskDefinition": "string",
                "updatedAt": number
            }
        ],
        "desiredCount": number,
        "events": [
            {
                "createdAt": number,
                "id": "string",
                "message": "string"
            }
        ],
        "loadBalancers": [
            {
                "containerName": "string",
                "containerPort": number,
                "loadBalancerName": "string"
            }
        ],
        "pendingCount": number,
        "roleArn": "string",
        "runningCount": number,
        "serviceArn": "string",
        "serviceName": "string",
        "status": "string",
        "taskDefinition": "string"
    }
}
```

Response Elements

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

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

service

The full description of your service following the create call.

Type: Service (p. 118) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

Example

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

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
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",
    "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-defini
tion/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"
  }
}
```

DeleteCluster

Deletes the specified cluster. You must deregister all container instances from this cluster before you may delete it. You can list the container instances in a cluster with ListContainerInstances (p. 52) and deregister them with DeregisterContainerInstance (p. 18).

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: Yes

Response Syntax

```
{
    "cluster": {
        "activeServicesCount": number,
        "clusterArn": "string",
        "clusterName": "string",
        "pendingTasksCount": number,
        "registeredContainerInstancesCount": number,
        "runningTasksCount": number,
        "status": "string"
    }
}
```

Response Elements

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

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

cluster

The full description of the deleted cluster.

Type: Cluster (p. 106) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

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

HTTP Status Code: 400

ClusterContainsContainerInstancesException

You cannot delete a cluster that has registered container instances. You must first deregister the container instances before you can delete the cluster. For more information, see DeregisterContainerInstance (p. 18).

HTTP Status Code: 400

ClusterContainsServicesException

You cannot delete a cluster that contains services. You must first update the service to reduce its desired task count to 0 and then delete the service. For more information, see UpdateService (p. 101) and DeleteService (p. 14).

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster could not be found. You can view your available clusters with ListClusters (p. 49). 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-side issue.

HTTP Status Code: 500

Examples

Example

This example request deletes the cluster called My-cluster.

Sample Request

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

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

Sample Response

```
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"
  }
}
```

DeleteService

Deletes a specified service within a cluster.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

The name of the cluster that hosts the service you want to delete. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

service

The name of the service you want to delete.

Type: String

Required: Yes

Response Syntax

```
{
    "service": {
        "clusterArn": "string",
        "deployments": [
            {
                "createdAt": number,
                "desiredCount": number,
                "id": "string",
                "pendingCount": number,
                "runningCount": number,
                "status": "string",
                "taskDefinition": "string",
                "updatedAt": number
            }
        ],
        "desiredCount": number,
        "events": [
            {
                "createdAt": number,
                "id": "string",
                "message": "string"
            }
        ],
        "loadBalancers": [
            {
                "containerName": "string",
                "containerPort": number,
                "loadBalancerName": "string"
            }
        ],
        "pendingCount": number,
        "roleArn": "string",
        "runningCount": number,
        "serviceArn": "string",
        "serviceName": "string",
        "status": "string",
        "taskDefinition": "string"
    }
}
```

Response Elements

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

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

service

Type: Service (p. 118) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 55). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Examples

Example

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

Sample Request

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

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

Sample Response

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

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deployments": [
      {
        "createdAt": 1430320735.285,
        "desiredCount": 0,
        "id": "ecs-svc/9223370606534040511",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/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-defini
tion/sleep360:27"
  }
}
```

DeregisterContainerInstance

Deregisters an Amazon ECS container instance from the specified cluster. This instance will no longer be 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 to avoid 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**
> When you terminate a container instance, it is automatically deregistered from your cluster.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstance

The container instance UUID or full Amazon Resource Name (ARN) of the container instance you want 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 UUID. For example, arn:aws:ecs:*region*:*aws_account_id*:container-instance/*container_instance_UUID*.

Type: String

Required: Yes

force

Force 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 and they will continue to pass Elastic Load Balancing load balancer health checks 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 will start another copy of that task on a different container instance if possible.

Type: Boolean

Required: No

Response Syntax

```
{
    "containerInstance": {
        "agentConnected": boolean,
        "agentUpdateStatus": "string",
        "containerInstanceArn": "string",
        "ec2InstanceId": "string",
        "pendingTasksCount": 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",
        "versionInfo": {
            "agentHash": "string",
            "agentVersion": "string",
            "dockerVersion": "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

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

Type: ContainerInstance (p. 110) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 61
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeregisterContainerInstance
X-Amz-Date: 20150429T182331Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "containerInstance": "f4292606-fbed-4b53-833b-92cad7c687c2"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 18:23:31 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 905
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "containerInstance": {
    "agentConnected": true,
    "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-in
stance/f4292606-fbed-4b53-833b-92cad7c687c2",
    "ec2InstanceId": "i-9281fb6f",
    "pendingTasksCount": 0,
    "registeredResources": [
      {
        "doubleValue": 0,
        "integerValue": 2048,
        "longValue": 0,
        "name": "CPU",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 3955,
        "longValue": 0,
        "name": "MEMORY",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS",
        "stringSetValue": [
          "2376",
          "22",
          "51678",
          "2375"
        ],
        "type": "STRINGSET"
      }
    ],
    "remainingResources": [
      {
        "doubleValue": 0,
        "integerValue": 2048,
        "longValue": 0,
        "name": "CPU",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 3955,
        "longValue": 0,
        "name": "MEMORY",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS",
        "stringSetValue": [
          "2376",
          "22",
          "51678",
          "2375"
        ],
        "type": "STRINGSET"
      }
    ],
    "runningTasksCount": 0,
    "status": "INACTIVE"
  }
}
```

DeregisterTaskDefinition

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

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

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

taskDefinition

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

Type: String

Required: Yes

Response Syntax

```
{
    "taskDefinition": {
        "containerDefinitions": [
            {
                "command": [
                    "string"
                ],
                "cpu": number,
                "entryPoint": [
                    "string"
                ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "essential": boolean,
                "image": "string",
                "links": [
                    "string"
                ],
                "memory": number,
                "mountPoints": [
                    {
                        "containerPath": "string",
                        "readOnly": boolean,
                        "sourceVolume": "string"
                    }
                ],
                "name": "string",
                "portMappings": [
                    {
                        "containerPort": number,
                        "hostPort": number,
                        "protocol": "string"
                    }
                ],
                "volumesFrom": [
                    {
                        "readOnly": boolean,
                        "sourceContainer": "string"
                    }
                ]
            }
        ],
        "family": "string",
        "revision": number,
        "status": "string",
        "taskDefinitionArn": "string",
        "volumes": [
            {
                "host": {
                    "sourcePath": "string"
                },
                "name": "string"
            }
        ]
    }
}
```

Response Elements

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

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

taskDefinition

The full description of the deregistered task.

Type: TaskDefinition (p. 121) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

Example

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

Sample Request

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

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

Sample Response

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

{
  "taskDefinition": {
    "containerDefinitions": [
      {
        "command": [
          "apt-get update; apt-get install stress; while true; do stress --cpu
 $(( RANDOM % 4 )) -t $(( RANDOM % 10 )); 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": []
  }
}
```

DescribeClusters

Describes one or more of your clusters.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

clusters

A space-separated list of 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

Response Syntax

```
{
    "clusters": [
        {
            "activeServicesCount": number,
            "clusterArn": "string",
            "clusterName": "string",
            "pendingTasksCount": number,
            "registeredContainerInstancesCount": number,
            "runningTasksCount": number,
            "status": "string"
        }
    ],
    "failures": [
        {
            "arn": "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.

clusters

The list of clusters.

Type: array of Cluster (p. 106) objects

failures

Type: array of Failure (p. 113) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

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": []
}
```

DescribeContainerInstances

Describes Amazon EC2 Container Service container instances. Returns metadata about registered and remaining resources on each container instance requested.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstances

A space-separated list of container instance UUIDs or full Amazon Resource Name (ARN) entries.

Type: array of Strings

Required: Yes

Response Syntax

```
{
    "containerInstances": [
        {
            "agentConnected": boolean,
            "agentUpdateStatus": "string",
            "containerInstanceArn": "string",
            "ec2InstanceId": "string",
            "pendingTasksCount": 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",
            "versionInfo": {
                "agentHash": "string",
                "agentVersion": "string",
                "dockerVersion": "string"
            }
        }
    ],
    "failures": [
        {
            "arn": "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
The list of container instances.

Type: array of ContainerInstance (p. 110) objects

failures
Type: array of Failure (p. 113) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException
These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

Example

This example request provides descriptive information about a container instance with an ID of 53ac7152-dcd1-4102-81f5-208962864132 in the update cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 85
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeContainerInstances
X-Amz-Date: 20150528T151428Z
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",
  "containerInstances": [
    "53ac7152-dcd1-4102-81f5-208962864132"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 28 May 2015 15:14:25 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1020
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "containerInstances": [
    {
      "agentConnected": true,
      "containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-
instance/53ac7152-dcd1-4102-81f5-208962864132",
      "ec2InstanceId": "i-f3c1de3a",
      "pendingTasksCount": 0,
      "registeredResources": [
        {
          "doubleValue": 0,
          "integerValue": 2048,
          "longValue": 0,
          "name": "CPU",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 3955,
          "longValue": 0,
          "name": "MEMORY",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
          "name": "PORTS",
          "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678"
          ],
          "type": "STRINGSET"
        }
      ],
      "remainingResources": [
        {
          "doubleValue": 0,
          "integerValue": 2048,
          "longValue": 0,
          "name": "CPU",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 3955,
          "longValue": 0,
          "name": "MEMORY",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
          "name": "PORTS",
          "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678"
          ],
          "type": "STRINGSET"
        }
      ],
      "runningTasksCount": 0,
      "status": "ACTIVE",
      "versionInfo": {
        "agentHash": "4023248",
        "agentVersion": "1.0.0",
        "dockerVersion": "DockerVersion: 1.5.0"
      }
    }
  ],
  "failures": []
}
```

DescribeServices

Describes the specified services running in your cluster.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster
The name of the cluster that hosts the service you want to describe. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

services
A list of services you want to describe.

Type: array of Strings

Required: Yes

Response Syntax

```
{
    "failures": [
        {
            "arn": "string",
            "reason": "string"
        }
    ],
    "services": [
        {
            "clusterArn": "string",
            "deployments": [
                {
                    "createdAt": number,
                    "desiredCount": number,
                    "id": "string",
                    "pendingCount": number,
                    "runningCount": number,
                    "status": "string",
                    "taskDefinition": "string",
                    "updatedAt": number
                }
            ],
            "desiredCount": number,
            "events": [
                {
                    "createdAt": number,
                    "id": "string",
                    "message": "string"
                }
            ],
            "loadBalancers": [
                {
                    "containerName": "string",
                    "containerPort": number,
                    "loadBalancerName": "string"
                }
            ],
            "pendingCount": number,
            "roleArn": "string",
            "runningCount": number,
            "serviceArn": "string",
            "serviceName": "string",
            "status": "string",
            "taskDefinition": "string"
        }
    ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: array of Failure (p. 113) objects

services

The list of services described.

Type: array of Service (p. 118) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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

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

{
  "failures": [],
  "services": [
    {
      "clusterArn": "arn:aws:ecs:us-west-2:012345678910:cluster/telemetry",
      "deployments": [
        {
          "createdAt": 1432829320.611,
          "desiredCount": 4,
          "id": "ecs-svc/9223370604025455196",
          "pendingCount": 0,
          "runningCount": 4,
          "status": "PRIMARY",
          "taskDefinition": "arn:aws:ecs:us-west-2:012345678910:task-defini
tion/hpcc-t2-medium:1",
          "updatedAt": 1432829320.611
        }
      ],
      "desiredCount": 4,
      "events": [],
      "loadBalancers": [],
      "pendingCount": 0,
      "runningCount": 4,
      "serviceArn": "arn:aws:ecs:us-west-2:012345678910:service/bunker-buster",

      "serviceName": "bunker-buster",
      "status": "ACTIVE",
      "taskDefinition": "arn:aws:ecs:us-west-2:012345678910:task-definition/hpcc-
t2-medium:1"
    }
  ]
}
```

DescribeTaskDefinition

Describes a task definition. You can specify a `family` and `revision` to find information on a specific task definition, or you can simply specify the family to find the latest `ACTIVE` revision in that family.

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

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

taskDefinition

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 that you want to describe.

Type: String

Required: Yes

Response Syntax

```
{
    "taskDefinition": {
        "containerDefinitions": [
            {
                "command": [
                    "string"
                ],
                "cpu": number,
                "entryPoint": [
                    "string"
                ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "essential": boolean,
                "image": "string",
                "links": [
                    "string"
                ],
                "memory": number,
                "mountPoints": [
                    {
                        "containerPath": "string",
                        "readOnly": boolean,
                        "sourceVolume": "string"
                    }
                ],
                "name": "string",
                "portMappings": [
                    {
                        "containerPort": number,
                        "hostPort": number,
                        "protocol": "string"
                    }
                ],
                "volumesFrom": [
                    {
                        "readOnly": boolean,
                        "sourceContainer": "string"
                    }
                ]
            }
        ],
        "family": "string",
        "revision": number,
        "status": "string",
        "taskDefinitionArn": "string",
        "volumes": [
            {
                "host": {
                    "sourcePath": "string"
                },
                "name": "string"
            }
        ]
    }
}
```

Response Elements

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

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

taskDefinition

The full task definition description.

Type: TaskDefinition (p. 121) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

Example

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

Sample Request

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

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

Sample Response

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

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

DescribeTasks

Describes a specified task or tasks.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

tasks

A space-separated list of task UUIDs or full Amazon Resource Name (ARN) entries.

Type: array of Strings

Required: Yes

Response Syntax

```
{
    "failures": [
        {
            "arn": "string",
            "reason": "string"
        }
    ],
    "tasks": [
        {
            "clusterArn": "string",
            "containerInstanceArn": "string",
            "containers": [
                {
                    "containerArn": "string",
                    "exitCode": number,
                    "lastStatus": "string",
                    "name": "string",
                    "networkBindings": [
                        {
                            "bindIP": "string",
                            "containerPort": number,
                            "hostPort": number,
                            "protocol": "string"
                        }
                    ],
                    "reason": "string",
                    "taskArn": "string"
                }
            ],
            "desiredStatus": "string",
            "lastStatus": "string",
            "overrides": {
                "containerOverrides": [
                    {
                        "command": [
                            "string"
                        ],
                        "environment": [
                            {
                                "name": "string",
                                "value": "string"
                            }
                        ],
                        "name": "string"
                    }
                ]
            },
            "startedBy": "string",
            "taskArn": "string",
            "taskDefinitionArn": "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
Type: array of Failure (p. 113) objects

tasks
The list of tasks.

Type: array of Task (p. 120) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

Example

This example request provides descriptive information about a task with an ID of c09f0188-7f87-4b0f-bfc3-16296622b6fe in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 51
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeTasks
X-Amz-Date: 20150429T190517Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "tasks": [
    "c09f0188-7f87-4b0f-bfc3-16296622b6fe"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:05:18 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1122
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/84818520-995f-4d94-9d70-7714bacc2953",
      "containers": [
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/76c980a8-
2454-4a9c-acc4-9eb103117273",
          "lastStatus": "RUNNING",
          "name": "mysql",
          "networkBindings": [],
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/c09f0188-7f87-
4b0f-bfc3-16296622b6fe"
        },
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e3c69b8f-
f15e-4d33-8093-282c2d2325e9",
          "lastStatus": "RUNNING",
          "name": "wordpress",
          "networkBindings": [
            {
              "bindIP": "0.0.0.0",
              "containerPort": 80,
              "hostPort": 80
            }
          ],
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/c09f0188-7f87-
4b0f-bfc3-16296622b6fe"
        }
      ],
      "desiredStatus": "RUNNING",
      "lastStatus": "RUNNING",
      "overrides": {
        "containerOverrides": [
          {
            "name": "mysql"
          },
          {
            "name": "wordpress"
          }
        ]
      },
      "startedBy": "ecs-svc/9223370606521064774",
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/c09f0188-7f87-4b0f-
bfc3-16296622b6fe",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/hello_world:10"
    }
  ]
}
```

DiscoverPollEndpoint

Note

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Returns an endpoint for the Amazon EC2 Container Service agent to poll for updates.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

The cluster that the container instance belongs to.

Type: String

Required: No

containerInstance

The container instance UUID or full 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 UUID. For example, arn:aws:ecs:*region*:*aws_account_id*:container-instance/*container_instance_UUID*.

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
The endpoint for the Amazon ECS agent to poll.

Type: String

telemetryEndpoint
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. 127).

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

HTTP Status Code: 400

ServerException
These errors are usually caused by a server-side issue.

HTTP Status Code: 500

ListClusters

Returns a list of existing clusters.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

maxResults

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: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListClusters` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

Required: No

Response Syntax

```
{
    "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

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

Type: array of Strings

nextToken

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. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

Example

This example request lists the clusters for your account.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListClusters
X-Amz-Date: 20150429T170621Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

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"
  ]
}
```

ListContainerInstances

Returns a list of container instances in a specified cluster.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

maxResults

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: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListContainerInstances` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

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
The list of container instance full Amazon Resource Name (ARN) entries for each container instance associated with the specified cluster.

Type: array of Strings

nextToken
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. 127).

ClientException
These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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
Date: Wed, 29 Apr 2015 17:53:06 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 492
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "containerInstanceArns": [
    "arn:aws:ecs:us-west-2:012345678910:container-instance/14e8cce9-0b16-4af4-
bfac-a85f7587aa98",
    "arn:aws:ecs:us-west-2:012345678910:container-instance/23bbf61b-45b4-4a4f-
b90c-c86290f066d6",
    "arn:aws:ecs:us-west-2:012345678910:container-instance/bd0abd43-94ce-4909-
9750-0dcc471ca4cb",
    "arn:aws:ecs:us-west-2:012345678910:container-instance/c967b2ee-68ea-415b-
b220-5936b26e6a04",
    "arn:aws:ecs:us-west-2:012345678910:container-instance/f5ec555b-8da4-48e1-
8427-0e03c3674a29"
  ]
}
```

ListServices

Lists the services that are running in a specified cluster.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

maxResults

The maximum number of container instance 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 100 results and a `nextToken` value if applicable.

Type: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListServices` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

Required: No

Response Syntax

```
{
    "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

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

The list of full Amazon Resource Name (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. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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"
  ]
}
```

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 definitions). You can filter the results with the `familyPrefix` parameter.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

familyPrefix

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

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: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListTaskDefinitionFamilies` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

Required: No

Response Syntax

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

Type: array of Strings

nextToken
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. 127).

ClientException
These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

Example

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

Sample Request

```
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

```
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",
    "node-dedupe",
    "port-mappings",
    "redis-volumes-from",
    "sleep360",
    "terrible-timer",
    "test-volumes-from",
    "tt-empty",
    "tt-empty-2vol",
    "tt-empty-volumes",
    "web-timer"
  ]
}
```

Example

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

Sample Request

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

{
  "familyPrefix": "hpcc"
}
```

Sample Response

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

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

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 common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

familyPrefix

The full family name that you want to filter the `ListTaskDefinitions` results with. Specifying a `familyPrefix` will limit the listed task definitions to task definition revisions that belong to that family.

Type: String

Required: No

maxResults

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: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListTaskDefinitions` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

Required: No

sort

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

The task definition status that you want to filter the `ListTaskDefinitions` results with. 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

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

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. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

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

{
  "taskDefinitionArns": [
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:1",
    "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:4",
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:5",
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:6",
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:7",
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:8",
    "arn:aws:ecs:us-east-1:012345678910:task-definition/hello_world:9",
    "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.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstance

The container instance UUID or full Amazon Resource Name (ARN) of the container instance that you want to filter the `ListTasks` results with. Specifying a `containerInstance` will limit the results to tasks that belong to that container instance.

Type: String

Required: No

desiredStatus

The task status that you want to filter the `ListTasks` results with. Specifying a `desiredStatus` of `STOPPED` will limit the results to tasks that are in the `STOPPED` status, which can be useful for debugging tasks that are not starting properly or have died or finished. The default status filter is `RUNNING`.

Type: String

Valid Values: `RUNNING | PENDING | STOPPED`

Required: No

family

The name of the family that you want to filter the `ListTasks` results with. Specifying a `family` will limit the results to tasks that belong to that family.

Type: String

Required: No

maxResults

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

Type: Number

Required: No

nextToken

The `nextToken` value returned from a previous paginated `ListTasks` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

Type: String

Required: No

serviceName

The name of the service that you want to filter the `ListTasks` results with. Specifying a `serviceName` will limit the results to tasks that belong to that service.

Type: String

Required: No

startedBy

The `startedBy` value that you want to filter the task results with. Specifying a `startedBy` value will limit the results to tasks that were started with that value.

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

nextToken

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

The list of task Amazon Resource Name (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. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 55). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Examples

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListTasks
```

```
X-Amz-Date: 20150429T192615Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:26:16 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 330
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "taskArns": [
    "arn:aws:ecs:us-east-1:012345678910:task/0b69d5c0-d655-4695-98cd-
5d2d526d9d5a",
    "arn:aws:ecs:us-east-1:012345678910:task/51a01bdf-d00e-487e-ab14-
7645330b6207",
    "arn:aws:ecs:us-east-1:012345678910:task/b0b28bb8-2be3-4810-b52b-
88df129d893c",
    "arn:aws:ecs:us-east-1:012345678910:task/c09f0188-7f87-4b0f-bfc3-
16296622b6fe"
  ]
}
```

RegisterContainerInstance

Note
This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Registers an Amazon EC2 instance into the specified cluster. This instance will become available to place containers on.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstanceArn

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

Type: String

Required: No

instanceIdentityDocument

The instance identity document for the Amazon 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

The instance identity document signature for the Amazon 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

totalResources

The resources available on the instance.

Type: array of Resource (p. 117) objects

Required: No

versionInfo

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

Type: VersionInfo (p. 122) object

Required: No

Response Syntax

```
{
    "containerInstance": {
        "agentConnected": boolean,
        "agentUpdateStatus": "string",
        "containerInstanceArn": "string",
        "ec2InstanceId": "string",
        "pendingTasksCount": 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",
        "versionInfo": {
            "agentHash": "string",
            "agentVersion": "string",
            "dockerVersion": "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

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

Type: ContainerInstance (p. 110) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

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

HTTP Status Code: 400

ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

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 on task definition parameters and defaults, see Amazon ECS Task Definitions in the *Amazon EC2 Container Service Developer Guide*.

Request Syntax

```
{
    "containerDefinitions": [
        {
            "command": [
                "string"
            ],
            "cpu": number,
            "entryPoint": [
                "string"
            ],
            "environment": [
                {
                    "name": "string",
                    "value": "string"
                }
            ],
            "essential": boolean,
            "image": "string",
            "links": [
                "string"
            ],
            "memory": number,
            "mountPoints": [
                {
                    "containerPath": "string",
                    "readOnly": boolean,
                    "sourceVolume": "string"
                }
            ],
            "name": "string",
            "portMappings": [
                {
                    "containerPort": number,
                    "hostPort": number,
                    "protocol": "string"
                }
            ],
            "volumesFrom": [
                {
                    "readOnly": boolean,
                    "sourceContainer": "string"
                }
            ]
        }
    ],
    "family": "string",
    "volumes": [
        {
            "host": {
                "sourcePath": "string"
            },
            "name": "string"
        }
    ]
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

containerDefinitions

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

Type: array of ContainerDefinition (p. 108) objects

Required: Yes

family

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

Type: String

Required: Yes

volumes

A list of volume definitions in JSON format that containers in your task may use.

Type: array of Volume (p. 123) objects

Required: No

Response Syntax

```
{
    "taskDefinition": {
        "containerDefinitions": [
            {
                "command": [
                    "string"
                ],
                "cpu": number,
                "entryPoint": [
                    "string"
                ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "essential": boolean,
                "image": "string",
                "links": [
                    "string"
                ],
                "memory": number,
                "mountPoints": [
                    {
                        "containerPath": "string",
                        "readOnly": boolean,
                        "sourceVolume": "string"
                    }
                ],
                "name": "string",
                "portMappings": [
                    {
                        "containerPort": number,
                        "hostPort": number,
                        "protocol": "string"
                    }
                ],
                "volumesFrom": [
                    {
                        "readOnly": boolean,
                        "sourceContainer": "string"
                    }
                ]
            }
        ],
        "family": "string",
        "revision": number,
        "status": "string",
        "taskDefinitionArn": "string",
        "volumes": [
            {
                "host": {
                    "sourcePath": "string"
                },
                "name": "string"
            }
        ]
    }
}
```

Response Elements

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

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

taskDefinition
Type: TaskDefinition (p. 121) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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-side issue.

HTTP Status Code: 500

Examples

Example

This example request registers a task definition in the hello_world family.

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:31:10 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": 11,
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/hello_world:11",
    "volumes": []
  }
}
```

RunTask

Start a task using random placement and the default Amazon ECS scheduler. If you want to use your own scheduler or place a task on a specific container instance, use `StartTask` instead.

Important
The count parameter is limited to 10 tasks per call.

Request Syntax

```
{
    "cluster": "string",
    "count": number,
    "overrides": {
        "containerOverrides": [
            {
                "command": [
                    "string"
                ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "name": "string"
            }
        ]
    },
    "startedBy": "string",
    "taskDefinition": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

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

Type: String

Required: No

count
The number of instantiations of the specified task that you would like to place on your cluster.

Important
The count parameter is limited to 10 tasks per call.

Type: Number

Required: No

overrides

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. 122) object

Required: No

startedBy

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. 66) call with the `startedBy` value.

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

Type: String

Required: No

taskDefinition

The `family` and `revision` (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want to run. If a `revision` is not specified, the latest `ACTIVE` revision is used.

Type: String

Required: Yes

Response Syntax

```
{
    "failures": [
        {
            "arn": "string",
            "reason": "string"
        }
    ],
    "tasks": [
        {
            "clusterArn": "string",
            "containerInstanceArn": "string",
            "containers": [
                {
                    "containerArn": "string",
                    "exitCode": number,
                    "lastStatus": "string",
                    "name": "string",
                    "networkBindings": [
                        {
                            "bindIP": "string",
                            "containerPort": number,
                            "hostPort": number,
                            "protocol": "string"
                        }
                    ],
                    "reason": "string",
                    "taskArn": "string"
                }
            ],
            "desiredStatus": "string",
            "lastStatus": "string",
            "overrides": {
                "containerOverrides": [
                    {
                        "command": [
                            "string"
                        ],
                        "environment": [
                            {
                                "name": "string",
                                "value": "string"
                            }
                        ],
                        "name": "string"
                    }
                ]
            },
            "startedBy": "string",
            "taskArn": "string",
            "taskDefinitionArn": "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

Any failed tasks from your `RunTask` action are listed here.

Type: array of Failure (p. 113) objects

tasks

A full description of the tasks that were run. Each task that was successfully placed on your cluster will be described here.

Type: array of Task (p. 120) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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: 20150429T193515Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:35:15 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 985
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/cf447635-790d-477d-be24-58a9cb819d45",
      "containers": [
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/e1ed7aac-
d9b2-4315-8726-d2432bf11868",
          "lastStatus": "PENDING",
          "name": "wordpress",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/d8c67b3c-ac87-
4ffe-a847-4785bc3a8b55"
        },
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/4b69fabd-
991d-4781-bbf9-7efa03c754aa",
          "lastStatus": "PENDING",
          "name": "mysql",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/d8c67b3c-ac87-
4ffe-a847-4785bc3a8b55"
        }
      ],
      "desiredStatus": "RUNNING",
      "lastStatus": "PENDING",
      "overrides": {
        "containerOverrides": [
          {
            "name": "wordpress"
          },
          {
            "name": "mysql"
          }
        ]
      },
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/d8c67b3c-ac87-4ffe-
a847-4785bc3a8b55",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/hello_world:11"
    }
  ]
}
```

StartTask

Starts a new task from the specified task definition on the specified container instance or instances. If you want to use the default Amazon ECS scheduler to place your task, use `RunTask` instead.

> **Important**
> The list of container instances to start tasks on is limited to 10.

Request Syntax

```
{
    "cluster": "string",
    "containerInstances": [
        "string"
    ],
    "overrides": {
        "containerOverrides": [
            {
                "command": [
                    "string"
                ],
                "environment": [
                    {
                        "name": "string",
                        "value": "string"
                    }
                ],
                "name": "string"
            }
        ]
    },
    "startedBy": "string",
    "taskDefinition": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstances

The container instance UUIDs or full Amazon Resource Name (ARN) entries for the container instances on which you would like to place your task.

> **Important**
> The list of container instances to start tasks on is limited to 10.

Type: array of Strings

Required: Yes

overrides

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. 122) object

Required: No

startedBy

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. 66) call with the `startedBy` value.

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

Type: String

Required: No

taskDefinition

The `family` and `revision` (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want to start. If a `revision` is not specified, the latest `ACTIVE` revision is used.

Type: String

Required: Yes

Response Syntax

```
{
    "failures": [
        {
            "arn": "string",
            "reason": "string"
        }
    ],
    "tasks": [
        {
            "clusterArn": "string",
            "containerInstanceArn": "string",
            "containers": [
                {
                    "containerArn": "string",
                    "exitCode": number,
                    "lastStatus": "string",
                    "name": "string",
                    "networkBindings": [
                        {
                            "bindIP": "string",
                            "containerPort": number,
                            "hostPort": number,
                            "protocol": "string"
                        }
                    ],
                    "reason": "string",
                    "taskArn": "string"
                }
            ],
            "desiredStatus": "string",
            "lastStatus": "string",
            "overrides": {
                "containerOverrides": [
                    {
                        "command": [
                            "string"
                        ],
                        "environment": [
                            {
                                "name": "string",
                                "value": "string"
                            }
                        ],
                        "name": "string"
                    }
                ]
            },
            "startedBy": "string",
            "taskArn": "string",
            "taskDefinitionArn": "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

Any failed tasks from your `StartTask` action are listed here.

Type: array of Failure (p. 113) objects

tasks

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

Type: array of Task (p. 120) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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 8db248d6-16a7-42b5-b9f9-43d3b1ad9430.

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: 20150429T194043Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "containerInstances": [
    "8db248d6-16a7-42b5-b9f9-43d3b1ad9430"
  ],
  "taskDefinition": "hello_world"
}
```

Sample Response

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

```
Date: Wed, 29 Apr 2015 19:40:44 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 985
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/8db248d6-16a7-42b5-b9f9-43d3b1ad9430",
      "containers": [
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/37234a82-
77f6-41d7-b54b-591f1e278093",
          "lastStatus": "PENDING",
          "name": "wordpress",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-
4b06-9d8f-1b56e75a99b5"
        },
        {
         "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/05a5528c-
77f6-4e5b-8f9a-2b0a1928a926",
          "lastStatus": "PENDING",
          "name": "mysql",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-
4b06-9d8f-1b56e75a99b5"
        }
      ],
      "desiredStatus": "RUNNING",
      "lastStatus": "PENDING",
      "overrides": {
        "containerOverrides": [
          {
            "name": "wordpress"
          },
          {
            "name": "mysql"
          }
        ]
      },
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-4b06-
9d8f-1b56e75a99b5",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/hello_world:11"
    }
  ]
}
```

StopTask

Stops a running task.

When StopTask (p. 89) is called on a task, the equivalent of `docker stop` is issued to the containers running in the task. This results in a `SIGTERM` and a 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.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

task

The task UUIDs or full Amazon Resource Name (ARN) entry of the task you would like to stop.

Type: String

Required: Yes

Response Syntax

```
{
    "task": {
        "clusterArn": "string",
        "containerInstanceArn": "string",
        "containers": [
            {
                "containerArn": "string",
                "exitCode": number,
                "lastStatus": "string",
                "name": "string",
                "networkBindings": [
                    {
                        "bindIP": "string",
                        "containerPort": number,
                        "hostPort": number,
                        "protocol": "string"
                    }
                ],
                "reason": "string",
                "taskArn": "string"
            }
        ],
        "desiredStatus": "string",
        "lastStatus": "string",
        "overrides": {
            "containerOverrides": [
                {
                    "command": [
                        "string"
                    ],
                    "environment": [
                        {
                            "name": "string",
                            "value": "string"
                        }
                    ],
                    "name": "string"
                }
            ]
        },
        "startedBy": "string",
        "taskArn": "string",
        "taskDefinitionArn": "string"
    }
}
```

Response Elements

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

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

task

Details on a task in a cluster.

Type: Task (p. 120) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

Examples

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: 48
X-Amz-Target: AmazonEC2ContainerServiceV20141113.StopTask
X-Amz-Date: 20150429T194253Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "task": "a126249b-b7e4-4b06-9d8f-1b56e75a99b5"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:42:53 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1063
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "task": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-in
stance/8db248d6-16a7-42b5-b9f9-43d3b1ad9430",
    "containers": [
      {
       "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/05a5528c-
77f6-4e5b-8f9a-2b0a1928a926",
        "lastStatus": "RUNNING",
        "name": "mysql",
        "networkBindings": [],
       "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-4b06-
9d8f-1b56e75a99b5"
      },
      {
       "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/37234a82-
77f6-41d7-b54b-591f1e278093",
        "lastStatus": "RUNNING",
        "name": "wordpress",
        "networkBindings": [
          {
            "bindIP": "0.0.0.0",
            "containerPort": 80,
            "hostPort": 80
          }
        ],
       "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-4b06-
9d8f-1b56e75a99b5"
      }
    ],
    "desiredStatus": "STOPPED",
    "lastStatus": "RUNNING",
    "overrides": {
      "containerOverrides": [
        {
          "name": "mysql"
        },
        {
          "name": "wordpress"
        }
      ]
    },
   "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/a126249b-b7e4-4b06-9d8f-
1b56e75a99b5",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-defini
tion/hello_world:11"
  }
}
```

SubmitContainerStateChange

Note

This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a container changed states.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

Type: String

Required: No

containerName

The name of the container.

Type: String

Required: No

exitCode

The exit code returned for the state change request.

Type: Number

Required: No

networkBindings

The network bindings of the container.

Type: array of NetworkBinding (p. 115) objects

Required: No

reason

The reason for the state change request.

Type: String

Required: No

status

The status of the state change request.

Type: String

Required: No

task

The task UUID or full Amazon Resource Name (ARN) of the task that hosts the container.

Type: String

Required: No

Response Syntax

```
{
    "acknowledgment": "string"
}
```

Response Elements

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

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

acknowledgment

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

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

HTTP Status Code: 400

ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

SubmitTaskStateChange

Note
This action is only used by the Amazon EC2 Container Service agent, and it is not intended for use outside of the agent.

Sent to acknowledge that a task changed states.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

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

Type: String

Required: No

reason
The reason for the state change request.

Type: String

Required: No

status
The status of the state change request.

Type: String

Required: No

task
The task UUID or full Amazon Resource Name (ARN) of the task in the state change request.

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

Acknowledgement of the state change.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

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

HTTP Status Code: 400

ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

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 EC2 Container Service Developer Guide*.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster
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
The container instance UUID or full Amazon Resource Name (ARN) entries for the container instance on which you would like to update the Amazon ECS container agent.

Type: String

Required: Yes

Response Syntax

```
{
    "containerInstance": {
        "agentConnected": boolean,
        "agentUpdateStatus": "string",
        "containerInstanceArn": "string",
        "ec2InstanceId": "string",
        "pendingTasksCount": 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",
        "versionInfo": {
            "agentHash": "string",
            "agentVersion": "string",
            "dockerVersion": "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

An Amazon EC2 instance that is running the Amazon ECS agent and has been registered with a cluster.

Type: ContainerInstance (p. 110) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side 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 will resume where it stopped previously.

HTTP Status Code: 400

Examples

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 82
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateContainerAgent
X-Amz-Date: 20150528T152756Z
User-Agent: aws-cli/1.7.30 Python/2.7.9 Darwin/14.3.0
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "cluster": "update",
  "containerInstance": "53ac7152-dcd1-4102-81f5-208962864132"
}
```

Sample Response

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

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

UpdateService

Modify the desired count or task definition used in a service.

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

You can use `UpdateService` to modify your task definition and deploy a new version of your service, one task at a time. 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. Because `UpdateService` starts a new version of the task before stopping an old version, your cluster must have capacity to support one more instantiation of the task when `UpdateService` is run. If your cluster cannot support another instantiation of the task used in your service, you can reduce the desired count of your service by one before modifying the task definition.

When UpdateService (p. 101) replaces a task during an update, 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.

Request Syntax

```
{
    "cluster": "string",
    "desiredCount": number,
    "service": "string",
    "taskDefinition": "string"
}
```

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 125).

The request requires the following data in JSON format.

cluster

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

desiredCount

The number of instantiations of the task that you would like to place and keep running in your service.

Type: Number

Required: No

service

The name of the service that you want to update.

Type: String

Required: Yes

taskDefinition

The `family` and `revision` (`family:revision`) or full Amazon Resource Name (ARN) of the task definition that you want 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": {
        "clusterArn": "string",
        "deployments": [
            {
                "createdAt": number,
                "desiredCount": number,
                "id": "string",
                "pendingCount": number,
                "runningCount": number,
                "status": "string",
                "taskDefinition": "string",
                "updatedAt": number
            }
        ],
        "desiredCount": number,
        "events": [
            {
                "createdAt": number,
                "id": "string",
                "message": "string"
            }
        ],
        "loadBalancers": [
            {
                "containerName": "string",
                "containerPort": number,
                "loadBalancerName": "string"
            }
        ],
        "pendingCount": number,
        "roleArn": "string",
        "runningCount": number,
        "serviceArn": "string",
        "serviceName": "string",
        "status": "string",
        "taskDefinition": "string"
    }
}
```

Response Elements

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

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

service

The full description of your service following the update call.

Type: Service (p. 118) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 127).

ClientException

These errors are usually caused by something the client did, such as use an action or resource on behalf of a user that doesn't have permission to use the action or resource, or specify 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. 49). 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-side issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service is not active. You cannot update a service that is not active. If you have previously deleted a service, you can recreate it with CreateService (p. 6).

HTTP Status Code: 400

ServiceNotFoundException

The specified service could not be found. You can view your available services with ListServices (p. 55). Amazon ECS services are cluster-specific and region-specific.

HTTP Status Code: 400

Examples

Example

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

Sample Request

```
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",
    "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-defini
tion/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-defini
tion/hello_world:10"
  }
}
```

Data Types

The EC2ContainerService API contains several data types that various actions use. This section describes each data type in detail.

Note
The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

Cluster

Description

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. 55).

Type: Number

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 can use to identify your cluster.

Type: String

Required: No

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

Type: Number

Required: No

registeredContainerInstancesCount
The number of container instances registered into the cluster.

Type: Number

Required: No

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

Type: Number

Required: No

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

Type: String

Required: No

Container

Description

A docker container that is part of a task.

Contents

containerArn

The Amazon Resource Name (ARN) of the container.

Type: String

Required: No

exitCode

The exit code returned from the container.

Type: Number

Required: No

lastStatus

The last known status of the container.

Type: String

Required: No

name

The name of the container.

Type: String

Required: No

networkBindings

The network bindings associated with the container.

Type: array of NetworkBinding (p. 115) objects

Required: No

reason

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

Type: String

Required: No

taskArn

The Amazon Resource Name (ARN) of the task.

Type: String

Required: No

ContainerDefinition

Description

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

Contents

command

The `CMD` that is passed to the container. For more information on the Docker `CMD` parameter, see https://docs.docker.com/reference/builder/#cmd.

Type: array of Strings

Required: No

cpu

The number of `cpu` units reserved for the container. A container instance has 1,024 `cpu` units for every CPU core. This parameter specifies the minimum amount of CPU to reserve for a container, and containers share unallocated CPU units with other containers on the instance with the same ratio as their allocated amount.

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

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

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 will allow is 2; however, the CPU parameter is not required, and you can use CPU values below 2 in your container definitions. For CPU values below 2 (including null), the behavior varies based on your Amazon ECS container agent version:

- **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.

Type: Number

Required: No

entryPoint

> **Important**
> Early versions of the Amazon ECS container agent do not properly handle entryPoint parameters. If you have problems using entryPoint, update your container agent or enter your commands and arguments as command array items instead.

The ENTRYPOINT that is passed to the container. For more information on the Docker ENTRYPOINT parameter, see https://docs.docker.com/reference/builder/#entrypoint.

Type: array of Strings

Required: No

environment

The environment variables to pass to a container.

Type: array of KeyValuePair (p. 114) objects

Required: No

essential

If the essential parameter of a container is marked as true, the failure of that container will stop the task. If the essential parameter of a container is marked as false, then its failure will not affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be essential.

Note

All tasks must have at least one essential container.

Type: Boolean

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 `repository-url/image:tag` .

Type: String

Required: No

links

The link parameter allows containers to communicate with each other without the need for port mappings, using the name parameter. The name:internalName construct is analogous to name:alias in Docker links. For more information on linking Docker containers, see https://docs.docker.com/userguide/dockerlinks/.

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

memory

The number of MiB of memory reserved for the container. If your container attempts to exceed the memory allocated here, the container is killed.

Type: Number

Required: No

mountPoints

The mount points for data volumes in your container.

Type: array of MountPoint (p. 115) 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.

Type: String

Required: No

portMappings

The list of port mappings for the container.

Type: array of PortMapping (p. 116) objects

Required: No

volumesFrom

Data volumes to mount from another container.

Type: array of VolumeFrom (p. 123) objects

Required: No

ContainerInstance

Description

An Amazon 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 actually connected to Amazon ECS. Registered instances with an agent that may be unhealthy or stopped will return `false`, and instances without a connected agent cannot accept placement request.

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

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 UUID. For example, arn:aws:ecs:*region*:*aws_account_id*:container-instance/*container_instance_UUID*.

Type: String

Required: No

ec2InstanceId
The Amazon 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: Number

Required: No

registeredResources
The registered resources on the container instance that are in use by current tasks.

Type: array of Resource (p. 117) objects

Required: No

remainingResources
The remaining resources of the container instance that are available for new tasks.

Type: array of Resource (p. 117) objects

Required: No

runningTasksCount
The number of tasks on the container instance that are in the `RUNNING` status.

Type: Number

Required: No

status
The status of the container instance. The valid values are `ACTIVE` or `INACTIVE`. `ACTIVE` indicates that the container instance can accept tasks.

Type: String

Required: No

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

Type: VersionInfo (p. 122) object

Required: No

ContainerOverride

Description

The overrides that should be sent to a container.

Contents

command

The command to send to the container that overrides the default command from the Docker image or the task definition.

Type: array of Strings

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.

Type: array of KeyValuePair (p. 114) objects

Required: No

name

The name of the container that receives the override.

Type: String

Required: No

Deployment

Description

The details of an Amazon ECS service deployment.

Contents

createdAt

The Unix time in seconds and milliseconds when the service was created.

Type: DateTime

Required: No

desiredCount

The most recent desired count of tasks that was specified for the service to deploy and/or maintain.

Type: Number

Required: No

id

The ID of the deployment.

Type: String

Required: No

pendingCount

The number of tasks in the deployment that are in the `PENDING` status.

Type: Number

Required: No

runningCount

The number of tasks in the deployment that are in the `RUNNING` status.

Type: Number

Required: No

status

The status of the deployment. Valid values are `PRIMARY` (for the most recent deployment), `ACTIVE` (for previous deployments that still have tasks running, but are being replaced with the `PRIMARY` deployment), and `INACTIVE` (for deployments that have been completely replaced).

Type: String

Required: No

taskDefinition

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

Type: String

Required: No

updatedAt

The Unix time in seconds and milliseconds when the service was last updated.

Type: DateTime

Required: No

Failure

Description

A failed resource.

Contents

arn

The Amazon Resource Name (ARN) of the failed resource.

Type: String

Required: No

reason

The reason for the failure.

Type: String

Required: No

HostVolumeProperties

Description

Details on a container instance host volume.

Contents

sourcePath

The path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you.

Type: String

Required: No

KeyValuePair

Description

A key and value pair object.

Contents

name

The name of the key value pair. For environment variables, this is the name of the environment variable.

Type: String

Required: No

value

The value of the key value pair. For environment variables, this is the value of the environment variable.

Type: String

Required: No

LoadBalancer

Description

Details on a load balancer that is used with a service.

Contents

containerName

The name of the container to associate with the load balancer.

Type: String

Required: No

containerPort

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the service's task definition. Your container instances must allow ingress traffic on the `hostPort` of the port mapping.

Type: Number

Required: No

loadBalancerName

The name of the load balancer.

Type: String

Required: No

MountPoint

Description

Details on a volume mount point that is used in a container definition.

Contents

containerPath

The path on the container to mount the host volume at.

Type: String

Required: No

readOnly

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

Type: Boolean

Required: No

sourceVolume

The name of the volume to mount.

Type: String

Required: No

NetworkBinding

Description

Details on the network bindings between a container and its host container instance.

Contents

bindIP

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

Type: String

Required: No

containerPort

The port number on the container that is be used with the network binding.

Type: Number

Required: No

hostPort

The port number on the host that is used with the network binding.

Type: Number

Required: No

protocol

The protocol used for the network binding.

Type: String

Valid Values: `tcp | udp`

Required: No

PortMapping

Description

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.

Contents

containerPort

The port number on the container that is bound to the user-specified or automatically assigned host port. If you specify a container port and not a host port, your container will automatically receive a host port in the ephemeral port range (for more information, see `hostPort`).

Type: Number

Required: No

hostPort

The port number on the container instance to reserve for your container. 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 will automatically receive a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range is 49153 to 65535, and this range is used for Docker versions prior to 1.6.0. For Docker version 1.6.0 and later, the Docker daemon tries to read the ephemeral port range from /proc/sys/net/ipv4/ip_local_port_range; if this kernel parameter is unavailable, the default ephemeral port range is used. You should not attempt to specify a host port in the ephemeral port range, since these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS Container Agent port 51678. Any host port that was previously specified in a running task is also reserved while the task is running (once a task stops, the host port is released).The current reserved ports are displayed in the remainingResources of DescribeContainerInstances (p. 30) output, and a container instance may have up to 50 reserved ports at a time, including the default reserved ports (automatically assigned ports do not count toward this limit).

Type: Number

Required: No

protocol

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

Type: String

Valid Values: tcp | udp

Required: No

Resource

Description

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: Number

Required: No

longValue

When the longValue type is set, the value of the resource must be an extended precision floating-point type.

Type: Long

Required: No

name

The name of the resource, such as CPU, MEMORY, PORTS, or a user-defined resource.

Type: String

Required: No

stringSetValue

When the `stringSetValue` 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

Service

Description

Details on a service within a cluster

Contents

clusterArn

The Amazon Resource Name (ARN) of the of the cluster that hosts the service.

Type: String

Required: No

deployments

The current state of deployments for the service.

Type: array of Deployment (p. 112) objects

Required: No

desiredCount

The desired number of instantiations of the task definition to keep running on the service. This value is specified when the service is created with CreateService (p. 6), and it can be modified with UpdateService (p. 101).

Type: Number

Required: No

events

The event stream for your service. A maximum of 100 of the latest events are displayed.

Type: array of ServiceEvent (p. 119) objects

Required: No

loadBalancers

A list of 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. 114) objects

Required: No

pendingCount

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

Type: Number

Required: No

roleArn

The Amazon Resource Name (ARN) of the IAM role associated with the service that allows the Amazon ECS container agent to register container instances with a load balancer.

Type: String

Required: No

runningCount

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

Type: Number

Required: No

serviceArn

The Amazon Resource Name (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

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

Type: String

Required: No

status

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

Type: String

Required: No

taskDefinition

The task definition to use for tasks in the service. This value is specified when the service is created with CreateService (p. 6), and it can be modified with UpdateService (p. 101).

Type: String

Required: No

ServiceEvent

Description

Details on an event associated with a service.

Contents

createdAt

The Unix time in seconds and milliseconds when the event was triggered.

Type: DateTime

Required: No

id

The ID string of the event.

Type: String

Required: No

message

The event message.

Type: String

Required: No

Task

Description

Details on a task in a cluster.

Contents

clusterArn

The Amazon Resource Name (ARN) of the of the cluster that hosts the task.

Type: String

Required: No

containerInstanceArn

The Amazon Resource Name (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. 107) objects

Required: No

desiredStatus

The desired status of the task.

Type: String

Required: No

lastStatus
The last known status of the task.

Type: String

Required: No

overrides
One or more container overrides.

Type: TaskOverride (p. 122) object

Required: No

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

Type: String

Required: No

taskArn
The Amazon Resource Name (ARN) of the task.

Type: String

Required: No

taskDefinitionArn
The Amazon Resource Name (ARN) of the of the task definition that creates the task.

Type: String

Required: No

TaskDefinition

Description

Details of a task definition.

Contents

containerDefinitions
A list of container definitions in JSON format that describe the different containers that make up your task. For more information on container definition parameters and defaults, see Amazon ECS Task Definitions in the *Amazon EC2 Container Service Developer Guide*.

Type: array of ContainerDefinition (p. 108) objects

Required: No

family
The family of your task definition. You can think of the `family` as the name of your task definition.

Type: String

Required: No

revision
The revision of the task in a particular family. You can think of the revision as a version number of a task definition in a family. When you register a task definition for the first time, the revision is `1`, and each time you register a new revision of a task definition in the same family, the revision value always increases by one (even if you have deregistered previous revisions in this family).

Type: Number

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 of the task definition.

Type: String

Required: No

volumes
The list of volumes in a task. For more information on volume definition parameters and defaults, see Amazon ECS Task Definitions in the *Amazon EC2 Container Service Developer Guide*.

Type: array of Volume (p. 123) objects

Required: No

TaskOverride

Description

The overrides associated with a task.

Contents

containerOverrides
One or more container overrides sent to a task.

Type: array of ContainerOverride (p. 111) objects

Required: No

VersionInfo

Description

The Docker and Amazon ECS container agent version information on a container instance.

Contents

agentHash

The Git commit hash for the Amazon ECS container agent build on the amazon-ecs-agent GitHub repository.

Type: String

Required: No

agentVersion

The version number of the Amazon ECS container agent.

Type: String

Required: No

dockerVersion

The Docker version running on the container instance.

Type: String

Required: No

Volume

Description

A data volume used in a task definition.

Contents

host

The path on the host container instance that is presented to the containers which access the volume. If this parameter is empty, then the Docker daemon assigns a host path for you.

Type: HostVolumeProperties (p. 114) object

Required: No

name

The name of the volume. This name is referenced in the `sourceVolume` parameter of container definition `mountPoints`.

Type: String

Required: No

VolumeFrom

Description

Details on a data volume from another container.

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 the container to mount volumes from.

Type: String

Required: No

Common Parameters

The following table lists the parameters that all actions use for signing Signature Version 4 requests. Any action-specific parameters are listed in the topic for that action. To view sample requests, see Examples of Signed Signature Version 4 Requests or Signature Version 4 Test Suite 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. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in *Using Temporary Security Credentials*.

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 common errors that all actions return. Any action-specific errors are listed in the topic for the action.

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

Throttling

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