aws

API Reference

Amazon Elastic Container Service

API Version 2014-11-13

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Elastic Container Service: API Reference

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Welcome **1**
Actions by Amazon ECS resource **2**
Account setting actions 2
Cluster actions 2
Cluster capacity provider actions 3
Container actions 3
Container agent actions 3
Container instance actions 3
Namespace actions 4
Service actions 4
Task definition actions 4
Task actions 5
Task set actions 5
Tag actions 5
Actions **7**
CreateCapacityProvider 9
Request Syntax 9
Request Parameters 9
Response Syntax 11
Response Elements 11
Errors 12
See Also 12
CreateCluster 14
Request Syntax 14
Request Parameters 15
Response Syntax 18
Response Elements 19
Errors 20
Examples 20
See Also 21
CreateService 23
Request Syntax 25
Request Parameters 28
Response Syntax 38

Response Elements ... 44
Errors ... 44
Examples ... 45
See Also ... 53
CreateTaskSet ... 54
Request Syntax ... 54
Request Parameters ... 55
Response Syntax ... 59
Response Elements ... 60
Errors ... 61
See Also ... 62
DeleteAccountSetting ... 64
Request Syntax ... 64
Request Parameters ... 64
Response Syntax ... 65
Response Elements ... 65
Errors ... 65
See Also ... 66
DeleteAttributes ... 67
Request Syntax ... 67
Request Parameters ... 67
Response Syntax ... 68
Response Elements ... 68
Errors ... 68
Examples ... 69
See Also ... 70
DeleteCapacityProvider ... 72
Request Syntax ... 72
Request Parameters ... 72
Response Syntax ... 73
Response Elements ... 73
Errors ... 74
See Also ... 74
DeleteCluster ... 75
Request Syntax ... 75
Request Parameters ... 75

Response Syntax 75
Response Elements 77
Errors 77
Examples 79
See Also 80
DeleteService 81
Request Syntax 81
Request Parameters 81
Response Syntax 82
Response Elements 88
Errors 88
Examples 89
See Also 91
DeleteTaskDefinitions 92
Request Syntax 92
Request Parameters 92
Response Syntax 93
Response Elements 99
Errors 99
Examples 100
See Also 102
DeleteTaskSet 103
Request Syntax 103
Request Parameters 103
Response Syntax 104
Response Elements 105
Errors 106
See Also 107
DeregisterContainerInstance 108
Request Syntax 108
Request Parameters 108
Response Syntax 109
Response Elements 111
Errors 111
Examples 112
See Also 116

DeregisterTaskDefinition ... 117
Request Syntax ... 117
Request Parameters ... 117
Response Syntax ... 118
Response Elements ... 123
Errors ... 124
Examples ... 124
See Also ... 126
DescribeCapacityProviders ... 127
Request Syntax ... 127
Request Parameters ... 127
Response Syntax ... 128
Response Elements ... 129
Errors ... 130
See Also ... 130
DescribeClusters ... 132
Request Syntax ... 132
Request Parameters ... 132
Response Syntax ... 133
Response Elements ... 134
Errors ... 135
Examples ... 135
See Also ... 139
DescribeContainerInstances ... 140
Request Syntax ... 140
Request Parameters ... 140
Response Syntax ... 141
Response Elements ... 143
Errors ... 143
Examples ... 144
See Also ... 148
DescribeServices ... 149
Request Syntax ... 149
Request Parameters ... 149
Response Syntax ... 150
Response Elements ... 156

Errors ... 156
Examples ... 157
See Also ... 159
DescribeTaskDefinition ... 160
Request Syntax ... 160
Request Parameters ... 160
Response Syntax ... 161
Response Elements ... 167
Errors ... 167
Examples ... 168
See Also ... 170
DescribeTasks ... 171
Request Syntax ... 171
Request Parameters ... 171
Response Syntax ... 172
Response Elements ... 176
Errors ... 176
Examples ... 177
See Also ... 179
DescribeTaskSets ... 181
Request Syntax ... 181
Request Parameters ... 181
Response Syntax ... 182
Response Elements ... 184
Errors ... 184
See Also ... 185
DiscoverPollEndpoint ... 187
Request Syntax ... 187
Request Parameters ... 187
Response Syntax ... 188
Response Elements ... 188
Errors ... 188
See Also ... 189
ExecuteCommand ... 190
Request Syntax ... 190
Request Parameters ... 190

Response Syntax ... 191
Response Elements ... 191
Errors ... 192
See Also ... 194
GetTaskProtection ... 195
Request Syntax ... 195
Request Parameters ... 195
Response Syntax ... 195
Response Elements ... 196
Errors ... 196
Examples ... 197
See Also ... 199
ListAccountSettings ... 200
Request Syntax ... 200
Request Parameters ... 200
Response Syntax ... 202
Response Elements ... 202
Errors ... 203
See Also ... 203
ListAttributes ... 205
Request Syntax ... 205
Request Parameters ... 205
Response Syntax ... 207
Response Elements ... 207
Errors ... 207
Examples ... 208
See Also ... 209
ListClusters ... 211
Request Syntax ... 211
Request Parameters ... 211
Response Syntax ... 212
Response Elements ... 212
Errors ... 212
Examples ... 213
See Also ... 214
ListContainerInstances ... 215

Request Syntax ... 215
Request Parameters ... 215
Response Syntax ... 217
Response Elements ... 217
Errors ... 217
Examples ... 218
See Also ... 219
ListServices ... 221
Request Syntax ... 221
Request Parameters ... 221
Response Syntax ... 222
Response Elements ... 223
Errors ... 223
Examples ... 224
See Also ... 225
ListServicesByNamespace ... 226
Request Syntax ... 226
Request Parameters ... 226
Response Syntax ... 227
Response Elements ... 227
Errors ... 228
See Also ... 228
ListTagsForResource ... 230
Request Syntax ... 230
Request Parameters ... 230
Response Syntax ... 230
Response Elements ... 230
Errors ... 231
Examples ... 232
See Also ... 233
ListTaskDefinitionFamilies ... 234
Request Syntax ... 234
Request Parameters ... 234
Response Syntax ... 235
Response Elements ... 236
Errors ... 236

Examples ... 237
See Also ... 239
ListTaskDefinitions ... 240
Request Syntax ... 240
Request Parameters ... 240
Response Syntax ... 242
Response Elements ... 242
Errors ... 242
Examples ... 243
See Also ... 244
ListTasks ... 246
Request Syntax ... 246
Request Parameters ... 246
Response Syntax ... 249
Response Elements ... 249
Errors ... 249
Examples ... 250
See Also ... 251
PutAccountSetting ... 253
Request Syntax ... 253
Request Parameters ... 253
Response Syntax ... 256
Response Elements ... 256
Errors ... 256
See Also ... 257
PutAccountSettingDefault ... 258
Request Syntax ... 258
Request Parameters ... 258
Response Syntax ... 260
Response Elements ... 261
Errors ... 261
See Also ... 261
PutAttributes ... 263
Request Syntax ... 263
Request Parameters ... 263
Response Syntax ... 264

Response Elements ... 264
Errors ... 264
Examples ... 265
See Also ... 266
PutClusterCapacityProviders ... 268
Request Syntax ... 268
Request Parameters ... 268
Response Syntax ... 270
Response Elements ... 271
Errors ... 272
See Also ... 273
RegisterContainerInstance ... 274
Request Syntax ... 274
Request Parameters ... 275
Response Syntax ... 277
Response Elements ... 279
Errors ... 279
See Also ... 280
RegisterTaskDefinition ... 281
Request Syntax ... 281
Request Parameters ... 287
Response Syntax ... 295
Response Elements ... 300
Errors ... 301
Examples ... 302
See Also ... 305
RunTask ... 306
Request Syntax ... 307
Request Parameters ... 310
Response Syntax ... 316
Response Elements ... 319
Errors ... 320
Examples ... 322
See Also ... 324
StartTask ... 325
Request Syntax ... 325

Request Parameters ... 327
Response Syntax ... 331
Response Elements ... 335
Errors ... 335
Examples ... 336
See Also ... 338
StopTask ... 340
Request Syntax ... 340
Request Parameters ... 340
Response Syntax ... 341
Response Elements ... 345
Errors ... 345
Examples ... 346
See Also ... 348
SubmitAttachmentStateChanges ... 349
Request Syntax ... 349
Request Parameters ... 349
Response Syntax ... 350
Response Elements ... 350
Errors ... 350
See Also ... 351
SubmitContainerStateChange ... 352
Request Syntax ... 352
Request Parameters ... 352
Response Syntax ... 354
Response Elements ... 354
Errors ... 354
See Also ... 355
SubmitTaskStateChange ... 356
Request Syntax ... 356
Request Parameters ... 357
Response Syntax ... 359
Response Elements ... 359
Errors ... 359
See Also ... 360
TagResource ... 361

Request Syntax ... 361
Request Parameters ... 361
Response Elements ... 362
Errors ... 362
Examples ... 363
See Also ... 365
UntagResource ... 366
Request Syntax ... 366
Request Parameters ... 366
Response Elements ... 366
Errors ... 367
Examples ... 367
See Also ... 368
UpdateCapacityProvider ... 370
Request Syntax ... 370
Request Parameters ... 370
Response Syntax ... 371
Response Elements ... 371
Errors ... 372
See Also ... 372
UpdateCluster ... 373
Request Syntax ... 373
Request Parameters ... 373
Response Syntax ... 374
Response Elements ... 376
Errors ... 376
See Also ... 377
UpdateClusterSettings ... 378
Request Syntax ... 378
Request Parameters ... 378
Response Syntax ... 379
Response Elements ... 380
Errors ... 381
See Also ... 381
UpdateContainerAgent ... 383
Request Syntax ... 383

Request Parameters ... 383
Response Syntax ... 384
Response Elements ... 386
Errors ... 386
Examples ... 387
See Also ... 389
UpdateContainerInstancesState ... 390
Request Syntax ... 391
Request Parameters ... 391
Response Syntax ... 392
Response Elements ... 394
Errors ... 394
Examples ... 395
See Also ... 400
UpdateService ... 401
Request Syntax ... 404
Request Parameters ... 407
Response Syntax ... 413
Response Elements ... 419
Errors ... 419
Examples ... 421
See Also ... 423
UpdateServicePrimaryTaskSet ... 424
Request Syntax ... 424
Request Parameters ... 424
Response Syntax ... 425
Response Elements ... 426
Errors ... 426
See Also ... 428
UpdateTaskProtection ... 429
Request Syntax ... 429
Request Parameters ... 430
Response Syntax ... 431
Response Elements ... 431
Errors ... 432
Examples ... 433

See Also ... 436
UpdateTaskSet ... 438
Request Syntax ... 438
Request Parameters ... 438
Response Syntax ... 439
Response Elements ... 440
Errors ... 441
See Also ... 442
Data Types ... 443
Attachment ... 447
Contents ... 447
See Also ... 448
AttachmentStateChange ... 449
Contents ... 449
See Also ... 449
Attribute ... 450
Contents ... 450
See Also ... 451
AutoScalingGroupProvider ... 452
Contents ... 452
See Also ... 453
AutoScalingGroupProviderUpdate ... 454
Contents ... 454
See Also ... 455
AwsVpcConfiguration ... 456
Contents ... 456
See Also ... 457
CapacityProvider ... 458
Contents ... 458
See Also ... 460
CapacityProviderStrategyItem ... 461
Contents ... 461
See Also ... 462
Cluster ... 463
Contents ... 463
See Also ... 467

ClusterConfiguration ... 469
Contents ... 469
See Also ... 469
ClusterServiceConnectDefaults ... 470
Contents ... 470
See Also ... 470
ClusterServiceConnectDefaultsRequest ... 471
Contents ... 471
See Also ... 472
ClusterSetting ... 473
Contents ... 473
See Also ... 473
Container ... 474
Contents ... 474
See Also ... 477
ContainerDefinition ... 478
Contents ... 478
See Also ... 497
ContainerDependency ... 498
Contents ... 498
See Also ... 499
ContainerInstance ... 500
Contents ... 500
See Also ... 504
ContainerInstanceHealthStatus ... 505
Contents ... 505
See Also ... 505
ContainerOverride ... 506
Contents ... 506
See Also ... 507
ContainerStateChange ... 509
Contents ... 509
See Also ... 510
Deployment ... 511
Contents ... 511
See Also ... 515

DeploymentAlarms ... 517
Contents ... 517
See Also ... 518
DeploymentCircuitBreaker ... 519
Contents ... 519
See Also ... 519
DeploymentConfiguration ... 521
Contents ... 521
See Also ... 523
DeploymentController ... 524
Contents ... 524
See Also ... 524
Device ... 526
Contents ... 526
See Also ... 526
DockerVolumeConfiguration ... 528
Contents ... 528
See Also ... 529
EBSTagSpecification ... 530
Contents ... 530
See Also ... 530
EFSAuthorizationConfig ... 532
Contents ... 532
See Also ... 532
EFSVolumeConfiguration ... 534
Contents ... 534
See Also ... 535
EnvironmentFile ... 536
Contents ... 536
See Also ... 537
EphemeralStorage ... 538
Contents ... 538
See Also ... 538
ExecuteCommandConfiguration ... 539
Contents ... 539
See Also ... 540

ExecuteCommandLogConfiguration 541
Contents 541
See Also 542
Failure 543
Contents 543
See Also 543
FirelensConfiguration 544
Contents 544
See Also 544
FSxWindowsFileServerAuthorizationConfig 546
Contents 546
See Also 546
FSxWindowsFileServerVolumeConfiguration 547
Contents 547
See Also 547
HealthCheck 549
Contents 551
See Also 552
HostEntry 554
Contents 554
See Also 554
HostVolumeProperties 555
Contents 555
See Also 555
InferenceAccelerator 556
Contents 556
See Also 556
InferenceAcceleratorOverride 557
Contents 557
See Also 557
InstanceHealthCheckResult 558
Contents 558
See Also 558
KernelCapabilities 560
Contents 560
See Also 561

KeyValuePair ... 562
Contents ... 562
See Also ... 562
LinuxParameters ... 563
Contents ... 563
See Also ... 565
LoadBalancer ... 567
Contents ... 567
See Also ... 568
LogConfiguration ... 570
Contents ... 570
See Also ... 572
ManagedAgent ... 573
Contents ... 573
See Also ... 573
ManagedAgentStateChange ... 575
Contents ... 575
See Also ... 575
ManagedScaling ... 577
Contents ... 577
See Also ... 578
MountPoint ... 580
Contents ... 580
See Also ... 580
NetworkBinding ... 581
Contents ... 581
See Also ... 583
NetworkConfiguration ... 584
Contents ... 584
See Also ... 584
NetworkInterface ... 585
Contents ... 585
See Also ... 585
PlacementConstraint ... 586
Contents ... 586
See Also ... 586

PlacementStrategy 588
Contents 588
See Also 588
PlatformDevice 590
Contents 590
See Also 590
PortMapping 591
Contents 591
See Also 595
ProtectedTask 596
Contents 596
See Also 596
ProxyConfiguration 597
Contents 597
See Also 598
RepositoryCredentials 599
Contents 599
See Also 599
Resource 600
Contents 600
See Also 601
ResourceRequirement 602
Contents 602
See Also 602
RuntimePlatform 604
Contents 604
See Also 604
Scale 606
Contents 606
See Also 606
Secret 607
Contents 607
See Also 608
Service 609
Contents 609
See Also 616

ServiceConnectClientAlias 617
Contents 617
See Also 618
ServiceConnectConfiguration 619
Contents 619
See Also 621
ServiceConnectService 622
Contents 622
See Also 623
ServiceConnectServiceResource 625
Contents 625
See Also 626
ServiceConnectTlsCertificateAuthority 627
Contents 627
See Also 627
ServiceConnectTlsConfiguration 628
Contents 628
See Also 628
ServiceEvent 629
Contents 629
See Also 629
ServiceManagedEBSVolumeConfiguration 630
Contents 630
See Also 633
ServiceRegistry 635
Contents 635
See Also 636
ServiceVolumeConfiguration 637
Contents 637
See Also 637
Session 638
Contents 638
See Also 638
Setting 639
Contents 639
See Also 640

SystemControl ... 641
Contents ... 642
See Also ... 642
Tag ... 643
Contents ... 643
See Also ... 644
Task ... 645
Contents ... 645
See Also ... 654
TaskDefinition ... 655
Contents ... 655
See Also ... 664
TaskDefinitionPlacementConstraint ... 665
Contents ... 665
See Also ... 665
TaskManagedEBSVolumeConfiguration ... 667
Contents ... 667
See Also ... 671
TaskManagedEBSVolumeTerminationPolicy ... 672
Contents ... 672
See Also ... 672
TaskOverride ... 673
Contents ... 673
See Also ... 674
TaskSet ... 675
Contents ... 675
See Also ... 681
TaskVolumeConfiguration ... 682
Contents ... 682
See Also ... 682
TimeoutConfiguration ... 683
Contents ... 683
See Also ... 684
Tmpfs ... 685
Contents ... 685
See Also ... 685

Ulimit ... 687
Contents ... 687
See Also ... 687
VersionInfo ... 689
Contents ... 689
See Also ... 689
Volume ... 690
Contents ... 690
See Also ... 692
VolumeFrom ... 693
Contents ... 693
See Also ... 693
API request throttling ... 694
How throttling Is applied ... 694
Throttling quotas ... 695
Adjusting API throttling quotas ... 698
Handling API throttling ... 699
Ensuring idempotency ... 700
Idempotency in Amazon ECS ... 700
Idempotency for RunTask ... 701
RunTask retry rules and responses ... 702
Examples ... 702
AWS CLI command examples ... 702
API request examples ... 703
Retry recommendations for idempotent requests ... 706
Common Parameters ... 707
Common Errors ... 710

Welcome

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service. It makes it easy to run, stop, and manage Docker containers. You can host your cluster on a serverless infrastructure that's managed by Amazon ECS by launching your services or tasks on AWS Fargate. For more control, you can host your tasks on a cluster of Amazon Elastic Compute Cloud (Amazon EC2) or External (on-premises) instances that you manage.

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

You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs, isolation policies, and availability requirements. With Amazon ECS, you don't need to operate your own cluster management and configuration management systems. You also don't need to worry about scaling your management infrastructure.

This document was last published on April 24, 2024.

Actions by Amazon ECS resource

The following section lists the API actions by resource.

Resource actions

- Account setting actions
- Cluster actions
- Cluster capacity provider actions
- Container actions
- Container agent actions
- Container instance actions
- Namespace actions
- Service actions
- Task definition actions
- Task actions
- Task set actions
- Tag actions

Account setting actions

The following API actions are available for account settings:

- DeleteAccountSetting
- ListAccountSettings
- PutAccountSetting
- PutAccountSettingDefault

Cluster actions

The following API actions are available for clusters:

- CreateCluster
- DeleteCluster

- DescribeClusters
- ListClusters
- UpdateCluster
- UpdateClusterSettings

Cluster capacity provider actions

The following APIs are available for cluster capacity providers:

- CreateCapacityProvider
- DeleteCapacityProvider
- DescribeCapacityProviders
- PutClusterCapacityProviders
- UpdateCapacityProvider

Container actions

The following APIs are available for containers:

- ExecuteCommand

Container agent actions

The following APIs are available for container agents:

- UpdateContainerAgent

Container instance actions

The following APIs are available for container instances:

- DeregisterContainerInstance
- DescribeContainerInstances
- DeleteAttributes

- ListAttributes
- ListContainerInstances
- PutAttributes
- UpdateContainerInstancesState

Namespace actions

> **Note**
>
> Namespaces are an AWS Cloud Map resource. Namespaces are necessary in Amazon ECS to use the ECS Service Connect or ECS service discovery features. Amazon ECS requests namespaces from AWS Cloud Map and then displays your namespaces in the AWS Management Console. Use the AWS Cloud Map console or API for additional configuration options.

The following APIs are available for namespaces:

- CreateCluster

Service actions

The following APIs are available for services:

- CreateService
- DeleteService
- DescribeServices
- ListServices
- UpdateService

Task definition actions

The following APIs are available for task-definitions:

- DeleteTaskDefinitions

- DeregisterTaskDefinition
- DescribeTaskDefinition
- ListTaskDefinitionFamilies
- ListTaskDefinitions
- RegisterTaskDefinition

Task actions

The following APIs are available for tasks:

- DescribeTasks
- GetTaskProtection
- ListTasks
- RunTask
- StartTask
- StopTask
- UpdateTaskProtection

Task set actions

The following APIs are available for task sets:

- CreateTaskSet
- DeleteTaskSet
- DescribeTaskSets
- UpdateServicePrimaryTaskSet
- UpdateTaskSet

Tag actions

The following APIs are available for tags:

- ListTagsForResource

- TagResource
- UntagResource

Actions

The following actions are supported:

- CreateCapacityProvider
- CreateCluster
- CreateService
- CreateTaskSet
- DeleteAccountSetting
- DeleteAttributes
- DeleteCapacityProvider
- DeleteCluster
- DeleteService
- DeleteTaskDefinitions
- DeleteTaskSet
- DeregisterContainerInstance
- DeregisterTaskDefinition
- DescribeCapacityProviders
- DescribeClusters
- DescribeContainerInstances
- DescribeServices
- DescribeTaskDefinition
- DescribeTasks
- DescribeTaskSets
- DiscoverPollEndpoint
- ExecuteCommand
- GetTaskProtection
- ListAccountSettings
- ListAttributes
- ListClusters
- ListContainerInstances

- ListServices
- ListServicesByNamespace
- ListTagsForResource
- ListTaskDefinitionFamilies
- ListTaskDefinitions
- ListTasks
- PutAccountSetting
- PutAccountSettingDefault
- PutAttributes
- PutClusterCapacityProviders
- RegisterContainerInstance
- RegisterTaskDefinition
- RunTask
- StartTask
- StopTask
- SubmitAttachmentStateChanges
- SubmitContainerStateChange
- SubmitTaskStateChange
- TagResource
- UntagResource
- UpdateCapacityProvider
- UpdateCluster
- UpdateClusterSettings
- UpdateContainerAgent
- UpdateContainerInstancesState
- UpdateService
- UpdateServicePrimaryTaskSet
- UpdateTaskProtection
- UpdateTaskSet

CreateCapacityProvider

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

Only capacity providers that use an Auto Scaling group can be created. Amazon ECS tasks on AWS Fargate use the FARGATE and FARGATE_SPOT capacity providers. These providers are available to all accounts in the AWS Regions that AWS Fargate supports.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

autoScalingGroupProvider

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

Type: AutoScalingGroupProvider object

Required: Yes

name

The name of the capacity provider. Up to 255 characters are allowed. They include letters (both upper and lowercase letters), numbers, underscores (_), and hyphens (-). The name can't be prefixed with "`aws`", "`ecs`", or "`fargate`".

Type: String

Required: Yes

tags

The metadata that you apply to the capacity provider to categorize and organize them more conveniently. Each tag consists of a key and an optional value. You define both of them.

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

Response Syntax

```
{
   "capacityProvider": {
      "autoScalingGroupProvider": {
         "autoScalingGroupArn": "string",
         "managedDraining": "string",
         "managedScaling": {
            "instanceWarmupPeriod": number,
            "maximumScalingStepSize": number,
            "minimumScalingStepSize": number,
            "status": "string",
            "targetCapacity": number
         },
         "managedTerminationProtection": "string"
      },
      "capacityProviderArn": "string",
      "name": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "updateStatus": "string",
      "updateStatusReason": "string"
   }
}
```

Response Elements

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

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

capacityProvider

The full description of the new capacity provider.

Type: CapacityProvider object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

LimitExceededException

The limit for the resource was exceeded.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UpdateInProgressException

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

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateCluster

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

> **Note**
>
> When you call the CreateCluster API operation, Amazon ECS attempts to create the Amazon ECS service-linked role for your account. This is so that it can manage required resources in other AWS services on your behalf. However, if the user that makes the call doesn't have permissions to create the service-linked role, it isn't created. For more information, see Using service-linked roles for Amazon ECS in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

```
{
   "capacityProviders": [ "string" ],
   "clusterName": "string",
   "configuration": {
      "executeCommandConfiguration": {
         "kmsKeyId": "string",
         "logConfiguration": {
            "cloudWatchEncryptionEnabled": boolean,
            "cloudWatchLogGroupName": "string",
            "s3BucketName": "string",
            "s3EncryptionEnabled": boolean,
            "s3KeyPrefix": "string"
         },
         "logging": "string"
      }
   },
   "defaultCapacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
```

```
    ],
    "serviceConnectDefaults": {
       "namespace": "string"
    },
    "settings": [
       {
          "name": "string",
          "value": "string"
       }
    ],
    "tags": [
       {
          "key": "string",
          "value": "string"
       }
    ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviders

The short name of one or more capacity providers to associate with the cluster. A capacity provider must be associated with a cluster before it can be included as part of the default capacity provider strategy of the cluster or used in a capacity provider strategy when calling the CreateService or RunTask actions.

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

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

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

Type: Array of strings

Required: No

clusterName

The name of your cluster. If you don't specify a name for your cluster, you create a cluster that's named `default`. Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed.

Type: String

Required: No

configuration

The `execute` command configuration for the cluster.

Type: ClusterConfiguration object

Required: No

defaultCapacityProviderStrategy

The capacity provider strategy to set as the default for the cluster. After a default capacity provider strategy is set for a cluster, when you call the CreateService or RunTask APIs with no capacity provider strategy or launch type specified, the default capacity provider strategy for the cluster is used.

If a default capacity provider strategy isn't defined for a cluster when it was created, it can be defined later with the PutClusterCapacityProviders API operation.

Type: Array of CapacityProviderStrategyItem objects

Required: No

serviceConnectDefaults

Use this parameter to set a default Service Connect namespace. After you set a default Service Connect namespace, any new services with Service Connect turned on that are created in the cluster are added as client services in the namespace. This setting only applies to new services that set the `enabled` parameter to `true` in the `ServiceConnectConfiguration`. You can

set the namespace of each service individually in the `ServiceConnectConfiguration` to override this default parameter.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ClusterServiceConnectDefaultsRequest object

Required: No

settings

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

Type: Array of ClusterSetting objects

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

Response Syntax

```
{
   "cluster": {
      "activeServicesCount": number,
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attachmentsStatus": "string",
      "capacityProviders": [ "string" ],
      "clusterArn": "string",
      "clusterName": "string",
      "configuration": {
         "executeCommandConfiguration": {
            "kmsKeyId": "string",
            "logConfiguration": {
               "cloudWatchEncryptionEnabled": boolean,
               "cloudWatchLogGroupName": "string",
               "s3BucketName": "string",
               "s3EncryptionEnabled": boolean,
               "s3KeyPrefix": "string"
            },
            "logging": "string"
         }
      },
      "defaultCapacityProviderStrategy": [
         {
```

```
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "pendingTasksCount": number,
      "registeredContainerInstancesCount": number,
      "runningTasksCount": number,
      "serviceConnectDefaults": {
         "namespace": "string"
      },
      "settings": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "statistics": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ]
   }
}
```

Response Elements

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

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

cluster

The full description of your new cluster.

Type: Cluster object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request creates a cluster called `My-cluster`.

Sample Request

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

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

Sample Response

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

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateService

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

> **Note**
>
> On March 21, 2024, a change was made to resolve the task definition revision before authorization. When a task definition revision is not specified, authorization will occur using the latest revision of a task definition.

In addition to maintaining the desired count of tasks in your service, you can optionally run your service behind one or more load balancers. The load balancers distribute traffic across the tasks that are associated with the service. For more information, see Service load balancing in the *Amazon Elastic Container Service Developer Guide*.

You can attach Amazon EBS volumes to Amazon ECS tasks by configuring the volume when creating or updating a service. `volumeConfigurations` is only supported for REPLICA service and not DAEMON service. For more infomation, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*.

Tasks for services that don't use a load balancer are considered healthy if they're in the `RUNNING` state. Tasks for services that use a load balancer are considered healthy if they're in the `RUNNING` state and are reported as healthy by the load balancer.

There are two service scheduler strategies available:

- `REPLICA` - The replica scheduling strategy places and maintains your desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions. For more information, see Service scheduler concepts in the *Amazon Elastic Container Service Developer Guide*.
- `DAEMON` - The daemon scheduling strategy deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks. It also stops tasks that don't meet the placement constraints. When using this strategy, you don't need to

specify a desired number of tasks, a task placement strategy, or use Service Auto Scaling policies. For more information, see Service scheduler concepts in the *Amazon Elastic Container Service Developer Guide*.

You can optionally specify a deployment configuration for your service. The deployment is initiated by changing properties. For example, the deployment might be initiated by the task definition or by your desired count of a service. This is done with an UpdateService operation. The default value for a replica service for `minimumHealthyPercent` is 100%. The default value for a daemon service for `minimumHealthyPercent` is 0%.

If a service uses the ECS deployment controller, the minimum healthy percent represents a lower limit on the number of tasks in a service that must remain in the `RUNNING` state during a deployment. Specifically, it represents it as a percentage of your desired number of tasks (rounded up to the nearest integer). This happens when any of your container instances are in the `DRAINING` state if the service contains tasks using the EC2 launch type. Using this parameter, you can deploy without using additional cluster capacity. For example, if you set your service to have desired number of four tasks and a minimum healthy percent of 50%, the scheduler might stop two existing tasks to free up cluster capacity before starting two new tasks. If they're in the `RUNNING` state, tasks for services that don't use a load balancer are considered healthy . If they're in the `RUNNING` state and reported as healthy by the load balancer, tasks for services that *do* use a load balancer are considered healthy . The default value for minimum healthy percent is 100%.

If a service uses the ECS deployment controller, the **maximum percent** parameter represents an upper limit on the number of tasks in a service that are allowed in the `RUNNING` or `PENDING` state during a deployment. Specifically, it represents it as a percentage of the desired number of tasks (rounded down to the nearest integer). This happens when any of your container instances are in the `DRAINING` state if the service contains tasks using the EC2 launch type. Using this parameter, you can define the deployment batch size. For example, if your service has a desired number of four tasks and a maximum percent value of 200%, the scheduler may start four new tasks before stopping the four older tasks (provided that the cluster resources required to do this are available). The default value for maximum percent is 200%.

If a service uses either the `CODE_DEPLOY` or `EXTERNAL` deployment controller types and tasks that use the EC2 launch type, the **minimum healthy percent** and **maximum percent** values are used only to define the lower and upper limit on the number of the tasks in the service that remain in the `RUNNING` state. This is while the container instances are in the `DRAINING` state. If the tasks

in the service use the Fargate launch type, the minimum healthy percent and maximum percent values aren't used. This is the case even if they're currently visible when describing your service.

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

When the service scheduler launches new tasks, it determines task placement. For information about task placement and task placement strategies, see Amazon ECS task placement in the *Amazon Elastic Container Service Developer Guide*

Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

Request Syntax

```
{
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
   ],
   "clientToken": "string",
   "cluster": "string",
   "deploymentConfiguration": {
      "alarms": {
         "alarmNames": [ "string" ],
         "enable": boolean,
         "rollback": boolean
      },
      "deploymentCircuitBreaker": {
         "enable": boolean,
         "rollback": boolean
      },
      "maximumPercent": number,
```

```
      "minimumHealthyPercent": number
   },
   "deploymentController": {
      "type": "string"
   },
   "desiredCount": number,
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "healthCheckGracePeriodSeconds": number,
   "launchType": "string",
   "loadBalancers": [
      {
         "containerName": "string",
         "containerPort": number,
         "loadBalancerName": "string",
         "targetGroupArn": "string"
      }
   ],
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
   "placementConstraints": [
      {
         "expression": "string",
         "type": "string"
      }
   ],
   "placementStrategy": [
      {
         "field": "string",
         "type": "string"
      }
   ],
   "platformVersion": "string",
   "propagateTags": "string",
   "role": "string",
   "schedulingStrategy": "string",
   "serviceConnectConfiguration": {
      "enabled": boolean,
      "logConfiguration": {
```

```
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "namespace": "string",
      "services": [
         {
            "clientAliases": [
               {
                  "dnsName": "string",
                  "port": number
               }
            ],
            "discoveryName": "string",
            "ingressPortOverride": number,
            "portName": "string",
            "timeout": {
               "idleTimeoutSeconds": number,
               "perRequestTimeoutSeconds": number
            },
            "tls": {
               "issuerCertificateAuthority": {
                  "awsPcaAuthorityArn": "string"
               },
               "kmsKey": "string",
               "roleArn": "string"
            }
         }
      ]
   },
   "serviceName": "string",
   "serviceRegistries": [
      {
         "containerName": "string",
         "containerPort": number,
         "port": number,
         "registryArn": "string"
```

```
      }
   ],
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": "string",
   "volumeConfigurations": [
      {
         "managedEBSVolume": {
            "encrypted": boolean,
            "filesystemType": "string",
            "iops": number,
            "kmsKeyId": "string",
            "roleArn": "string",
            "sizeInGiB": number,
            "snapshotId": "string",
            "tagSpecifications": [
               {
                  "propagateTags": "string",
                  "resourceType": "string",
                  "tags": [
                     {
                        "key": "string",
                        "value": "string"
                     }
                  ]
               }
            ],
            "throughput": number,
            "volumeType": "string"
         },
         "name": "string"
      }
   ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviderStrategy

The capacity provider strategy to use for the service.

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

A capacity provider strategy may contain a maximum of 6 capacity providers.

Type: Array of CapacityProviderStrategyItem objects

Required: No

clientToken

An identifier that you provide to ensure the idempotency of the request. It must be unique and is case sensitive. Up to 36 ASCII characters in the range of 33-126 (inclusive) are allowed.

Type: String

Required: No

cluster

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

Type: String

Required: No

deploymentConfiguration

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

Type: DeploymentConfiguration object

Required: No

deploymentController

The deployment controller to use for the service. If no deployment controller is specified, the default value of ECS is used.

Type: DeploymentController object

Required: No

desiredCount

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

This is required if `schedulingStrategy` is `REPLICA` or isn't specified. If `schedulingStrategy` is `DAEMON` then this isn't required.

Type: Integer

Required: No

enableECSManagedTags

Specifies whether to turn on Amazon ECS managed tags for the tasks within the service. For more information, see Tagging your Amazon ECS resources in the *Amazon Elastic Container Service Developer Guide*.

When you use Amazon ECS managed tags, you need to set the `propagateTags` request parameter.

Type: Boolean

Required: No

enableExecuteCommand

Determines whether the execute command functionality is turned on for the service. If `true`, this enables execute command functionality on all containers in the service tasks.

Type: Boolean

Required: No

healthCheckGracePeriodSeconds

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

If you do not use an Elastic Load Balancing, we recommend that you use the `startPeriod` in the task definition health check parameters. For more information, see Health check.

If your service's tasks take a while to start and respond to Elastic Load Balancing health checks, you can specify a health check grace period of up to 2,147,483,647 seconds (about 69 years). During that time, the Amazon ECS service scheduler ignores health check status. This grace period can prevent the service scheduler from marking tasks as unhealthy and stopping them before they have time to come up.

Type: Integer

Required: No

launchType

The infrastructure that you run your service on. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

The `FARGATE` launch type runs your tasks on AWS Fargate On-Demand infrastructure.

> **Note**
>
> Fargate Spot infrastructure is available for use but a capacity provider strategy must be used. For more information, see AWS Fargate capacity providers in the *Amazon ECS Developer Guide*.

The `EC2` launch type runs your tasks on Amazon EC2 instances registered to your cluster.

The `EXTERNAL` launch type runs your tasks on your on-premises server or virtual machine (VM) capacity registered to your cluster.

A service can use either a launch type or a capacity provider strategy. If a `launchType` is specified, the `capacityProviderStrategy` parameter must be omitted.

Type: String

Valid Values: `EC2` | `FARGATE` | `EXTERNAL`

Required: No

loadBalancers

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

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

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

If you use the `CODE_DEPLOY` deployment controller, these values can be changed when updating the service.

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

For Classic Load Balancers, this object must contain the load balancer name, the container name , and the container port to access from the load balancer. The container name must be as it appears in a container definition. The target group ARN parameter must be omitted. When a task from this service is placed on a container instance, the container instance is registered with the load balancer that's specified here.

Services with tasks that use the `awsvpc` network mode (for example, those with the Fargate launch type) only support Application Load Balancers and Network Load Balancers. Classic Load Balancers aren't supported. Also, when you create any target groups for these services, you

must choose `ip` as the target type, not `instance`. This is because tasks that use the `awsvpc` network mode are associated with an elastic network interface, not an Amazon EC2 instance.

Type: Array of LoadBalancer objects

Required: No

networkConfiguration

The network configuration for the service. This parameter is required for task definitions that use the `awsvpc` network mode to receive their own elastic network interface, and it isn't supported for other network modes. For more information, see Task networking in the *Amazon Elastic Container Service Developer Guide*.

Type: NetworkConfiguration object

Required: No

placementConstraints

An array of placement constraint objects to use for tasks in your service. You can specify a maximum of 10 constraints for each task. This limit includes constraints in the task definition and those specified at runtime.

Type: Array of PlacementConstraint objects

Required: No

placementStrategy

The placement strategy objects to use for tasks in your service. You can specify a maximum of 5 strategy rules for each service.

Type: Array of PlacementStrategy objects

Required: No

platformVersion

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

Type: String

Required: No

propagateTags

Specifies whether to propagate the tags from the task definition to the task. If no value is specified, the tags aren't propagated. Tags can only be propagated to the task during task creation. To add tags to a task after task creation, use the TagResource API action.

You must set this to a value other than `NONE` when you use Cost Explorer. For more information, see Amazon ECS usage reports in the *Amazon Elastic Container Service Developer Guide*.

The default is `NONE`.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

role

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

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

If your specified role has a path other than `/`, then you must either specify the full role ARN (this is recommended) or prefix the role name with the path. For example, if a role with the

name `bar` has a path of `/foo/` then you would specify `/foo/bar` as the role name. For more information, see Friendly names and paths in the *IAM User Guide*.

Type: String

Required: No

schedulingStrategy

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

There are two service scheduler strategies available:

- `REPLICA`-The replica scheduling strategy places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions. This scheduler strategy is required if the service uses the `CODE_DEPLOY` or `EXTERNAL` deployment controller types.
- `DAEMON`-The daemon scheduling strategy deploys exactly one task on each active container instance that meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks and will stop tasks that don't meet the placement constraints. When you're using this strategy, you don't need to specify a desired number of tasks, a task placement strategy, or use Service Auto Scaling policies.

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

Type: String

Valid Values: `REPLICA | DAEMON`

Required: No

serviceConnectConfiguration

The configuration for this service to discover and connect to services, and be discovered by, and connected from, other services within a namespace.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ServiceConnectConfiguration object

Required: No

serviceName

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

Type: String

Required: Yes

serviceRegistries

The details of the service discovery registry to associate with this service. For more information, see Service discovery.

> **Note**
>
> Each service may be associated with one service registry. Multiple service registries for each service isn't supported.

Type: Array of ServiceRegistry objects

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

taskDefinition

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

A task definition must be specified if the service uses either the `ECS` or `CODE_DEPLOY` deployment controllers.

For more information about deployment types, see Amazon ECS deployment types.

Type: String

Required: No

volumeConfigurations

The configuration for a volume specified in the task definition as a volume that is configured at launch time. Currently, the only supported volume type is an Amazon EBS volume.

Type: Array of ServiceVolumeConfiguration objects

Required: No

Response Syntax

```
{
   "service": {
      "capacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "clusterArn": "string",
      "createdAt": number,
      "createdBy": "string",
      "deploymentConfiguration": {
         "alarms": {
            "alarmNames": [ "string" ],
            "enable": boolean,
            "rollback": boolean
         },
         "deploymentCircuitBreaker": {
            "enable": boolean,
            "rollback": boolean
         },
         "maximumPercent": number,
         "minimumHealthyPercent": number
      },
      "deploymentController": {
         "type": "string"
      },
      "deployments": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
               }
            ],
            "createdAt": number,
            "desiredCount": number,
            "failedTasks": number,
            "id": "string",
```

```
            "launchType": "string",
            "networkConfiguration": {
               "awsvpcConfiguration": {
                  "assignPublicIp": "string",
                  "securityGroups": [ "string" ],
                  "subnets": [ "string" ]
               }
            },
            "pendingCount": number,
            "platformFamily": "string",
            "platformVersion": "string",
            "rolloutState": "string",
            "rolloutStateReason": "string",
            "runningCount": number,
            "serviceConnectConfiguration": {
               "enabled": boolean,
               "logConfiguration": {
                  "logDriver": "string",
                  "options": {
                     "string" : "string"
                  },
                  "secretOptions": [
                     {
                        "name": "string",
                        "valueFrom": "string"
                     }
                  ]
               },
               "namespace": "string",
               "services": [
                  {
                     "clientAliases": [
                        {
                           "dnsName": "string",
                           "port": number
                        }
                     ],
                     "discoveryName": "string",
                     "ingressPortOverride": number,
                     "portName": "string",
                     "timeout": {
                        "idleTimeoutSeconds": number,
                        "perRequestTimeoutSeconds": number
                     },
```

```
                "tls": {
                   "issuerCertificateAuthority": {
                      "awsPcaAuthorityArn": "string"
                   },
                   "kmsKey": "string",
                   "roleArn": "string"
                }
             }
          ]
       },
       "serviceConnectResources": [
          {
             "discoveryArn": "string",
             "discoveryName": "string"
          }
       ],
       "status": "string",
       "taskDefinition": "string",
       "updatedAt": number,
       "volumeConfigurations": [
          {
             "managedEBSVolume": {
                "encrypted": boolean,
                "filesystemType": "string",
                "iops": number,
                "kmsKeyId": "string",
                "roleArn": "string",
                "sizeInGiB": number,
                "snapshotId": "string",
                "tagSpecifications": [
                   {
                      "propagateTags": "string",
                      "resourceType": "string",
                      "tags": [
                         {
                            "key": "string",
                            "value": "string"
                         }
                      ]
                   }
                ],
                "throughput": number,
                "volumeType": "string"
             },
```

```
                "name": "string"
            }
        ]
    }
],
"desiredCount": number,
"enableECSManagedTags": boolean,
"enableExecuteCommand": boolean,
"events": [
    {
        "createdAt": number,
        "id": "string",
        "message": "string"
    }
],
"healthCheckGracePeriodSeconds": number,
"launchType": "string",
"loadBalancers": [
    {
        "containerName": "string",
        "containerPort": number,
        "loadBalancerName": "string",
        "targetGroupArn": "string"
    }
],
"networkConfiguration": {
    "awsvpcConfiguration": {
        "assignPublicIp": "string",
        "securityGroups": [ "string" ],
        "subnets": [ "string" ]
    }
},
"pendingCount": number,
"placementConstraints": [
    {
        "expression": "string",
        "type": "string"
    }
],
"placementStrategy": [
    {
        "field": "string",
        "type": "string"
    }
```

```
      ],
      "platformFamily": "string",
      "platformVersion": "string",
      "propagateTags": "string",
      "roleArn": "string",
      "runningCount": number,
      "schedulingStrategy": "string",
      "serviceArn": "string",
      "serviceName": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSets": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
               }
            ],
            "clusterArn": "string",
            "computedDesiredCount": number,
            "createdAt": number,
            "externalId": "string",
            "id": "string",
            "launchType": "string",
            "loadBalancers": [
               {
                  "containerName": "string",
                  "containerPort": number,
```

```
                "loadBalancerName": "string",
                "targetGroupArn": "string"
             }
          ],
          "networkConfiguration": {
             "awsvpcConfiguration": {
                "assignPublicIp": "string",
                "securityGroups": [ "string" ],
                "subnets": [ "string" ]
             }
          },
          "pendingCount": number,
          "platformFamily": "string",
          "platformVersion": "string",
          "runningCount": number,
          "scale": {
             "unit": "string",
             "value": number
          },
          "serviceArn": "string",
          "serviceRegistries": [
             {
                "containerName": "string",
                "containerPort": number,
                "port": number,
                "registryArn": "string"
             }
          ],
          "stabilityStatus": "string",
          "stabilityStatusAt": number,
          "startedBy": "string",
          "status": "string",
          "tags": [
             {
                "key": "string",
                "value": "string"
             }
          ],
          "taskDefinition": "string",
          "taskSetArn": "string",
          "updatedAt": number
       }
    ]
}
```

```
}
```

Response Elements

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

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

service

The full description of your service following the create call.

A service will return either a `capacityProviderStrategy` or `launchType` parameter, but not both, depending where one was specified when it was created.

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

if the service uses the `CODE_DEPLOY` deployment controller, the `deploymentController`, `taskSets` and `deployments` parameters will be returned, however the `deployments` parameter will be an empty list.

Type: Service object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version doesn't satisfy the required capabilities of the task definition.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version doesn't exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example 1

This example API request creates a service in your default Region called `ecs-simple-service`. The service uses the `ecs-demo` task definition and it maintains 10 instantiations of that task. It uses both the FARGATE and FARGATE_SPOT capacity providers.

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,
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "FARGATE",
         "weight": 1
      }
   ],
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "FARGATE_SPOT",
         "weight": 1
      }
   ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 17:01:27 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 636
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deploymentConfiguration": {
        "maximumPercent": 200,
        "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "createdAt": 1430326887.362,
        "desiredCount": 10,
        "id": "ecs-svc/9223370606527888445",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-
demo:1",
        "updatedAt": 1430326887.362
      }
    ],
    "capacityProviderStrategy": [
      {
        "capacityProvider": "FARGATE",
        "weight": 1,
        "base": 0
      },
      {
        "capacityProvider": "FARGATE_SPOT",
        "weight": 1,
        "base": 0
      }
    ],
    "desiredCount": 10,
    "events": [],
```

```
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/default/ecs-simple-
service",
    "serviceName": "ecs-simple-service",
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-demo:1"
  }
}
```

Example 2

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

Sample Request

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

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

```
   ],
   "desiredCount":10
}
```

Sample Response

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

{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/default/ecs-
multiplealb-service",
        "serviceName": "ecs-multiplealb-service",
        "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
        "loadBalancers": [
            {
                "targetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:012345678910:targetgroup/tg1/18ce32cc074018ed",
                "containerName": "simple-app",
                "containerPort": 80
            },
            {
                "targetGroupArn": "arn:aws:elasticloadbalancing:us-
east-1:012345678910:targetgroup/tg2/737bead11d516e2a",
                "containerName": "simple-app",
                "containerPort": 8080
            }
        ],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 10,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/ecs-
multiplealb-demo",
        "deploymentConfiguration": {
            "maximumPercent": 200,
```

```
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/9223370473014051517",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/
ecs-multiplealb-demo",
                "desiredCount": 10,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1563840724.29,
                "updatedAt": 1563840724.29,
                "launchType": "EC2"
            }
        ],
        "roleArn": "arn:aws:iam::012345678910:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS",
        "events": [],
        "createdAt": 1563840724.29,
        "placementConstraints": [],
        "placementStrategy": [],
        "healthCheckGracePeriodSeconds": 0,
        "schedulingStrategy": "REPLICA",
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

Example 3

This example API request creates a service with a strategy that distributes tasks evenly across Availability Zones and then bin packs tasks based on memory within each Availability Zone.

Sample Request

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

```
Authorization: AUTHPARAMS
Content-Length: 453

{
   "serviceName":"example-placement1",
   "taskDefinition":"windows-simple-iis",
   "placementStrategy": [
        {
            "field": "attribute:ecs.availability-zone",
            "type": "spread"
        },
        {
            "field": "memory",
            "type": "binpack"
        }
   ]
   "desiredCount":3
}
```

Sample Response

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

{
    "service": {
        "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/windows-ec2/default/
example-placement1",
        "serviceName": "example-placement1",
        "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/windows-ec2",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 3,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/windows-
simple-iis:2",
```

```
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": false,
                "rollback": false
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/409552086333EXAMPLE",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-east-1:123456789012:task-definition/
windows-simple-iis:2",
                "desiredCount": 3,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": "2022-07-21T15:06:08.787000-04:00",
                "updatedAt": "2022-07-21T15:06:08.787000-04:00",
                "launchType": "EC2",
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-svc/409552086333EXAMPLE in
 progress."
            }
        ],
        "events": [],
        "createdAt": "2022-07-21T15:06:08.787000-04:00",
        "placementConstraints": [],
        "placementStrategy": [
            {
                "type": "spread",
                "field": "attribute:ecs.availability-zone"
            },
            {
                "type": "binpack",
                "field": "MEMORY"
            }
        ],
        "schedulingStrategy": "REPLICA",
        "createdBy": "arn:aws:iam::123456789012:user/johndoe",
        "enableECSManagedTags": false,
        "propagateTags": "NONE",
        "enableExecuteCommand": false
```

```
    }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateTaskSet

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

> **Note**
>
> On March 21, 2024, a change was made to resolve the task definition revision before authorization. When a task definition revision is not specified, authorization will occur using the latest revision of a task definition.

For information about the maximum number of task sets and otther quotas, see Amazon ECS service quotas in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

```
{
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
   ],
   "clientToken": "string",
   "cluster": "string",
   "externalId": "string",
   "launchType": "string",
   "loadBalancers": [
      {
         "containerName": "string",
         "containerPort": number,
         "loadBalancerName": "string",
         "targetGroupArn": "string"
      }
   ],
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
```

```
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      },
      "platformVersion": "string",
      "scale": {
         "unit": "string",
         "value": number
      },
      "service": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviderStrategy

The capacity provider strategy to use for the task set.

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

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

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

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

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

Type: Array of CapacityProviderStrategyItem objects

Required: No

clientToken

An identifier that you provide to ensure the idempotency of the request. It must be unique and is case sensitive. Up to 36 ASCII characters in the range of 33-126 (inclusive) are allowed.

Type: String

Required: No

cluster

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

Type: String

Required: Yes

externalId

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

Type: String

Required: No

launchType

The launch type that new tasks in the task set uses. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

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

Type: String

Valid Values: `EC2` | `FARGATE` | `EXTERNAL`

Required: No

loadBalancers

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

Type: Array of LoadBalancer objects

Required: No

networkConfiguration

An object representing the network configuration for a task set.

Type: NetworkConfiguration object

Required: No

platformVersion

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

Type: String

Required: No

scale

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

Type: Scale object

Required: No

service

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

Type: String

Required: Yes

serviceRegistries

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

Type: Array of ServiceRegistry objects

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

taskDefinition

The task definition for the tasks in the task set to use. If a revision isn't specified, the latest `ACTIVE` revision is used.

Type: String

Required: Yes

Response Syntax

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

```
      "platformFamily": "string",
      "platformVersion": "string",
      "runningCount": number,
      "scale": {
         "unit": "string",
         "value": number
      },
      "serviceArn": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "stabilityStatus": "string",
      "stabilityStatusAt": number,
      "startedBy": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSetArn": "string",
      "updatedAt": number
   }
}
```

Response Elements

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

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

taskSet

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

Type: TaskSet object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version doesn't satisfy the required capabilities of the task definition.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version doesn't exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

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

DeleteAccountSetting

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

name

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

Type: String

Valid Values: `serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights | fargateFIPSMode | tagResourceAuthorization | fargateTaskRetirementWaitPeriod | guardDutyActivate`

Required: Yes

principalArn

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

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

setting

The account setting for the specified principal ARN.

Type: Setting object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteAttributes

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

attributes

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

Type: Array of Attribute objects

Required: Yes

cluster

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

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

attributes

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

Type: Array of Attribute objects

Errors

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

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

TargetNotFoundException

The specified target wasn't found. You can view your available container instances with ListContainerInstances. Amazon ECS container instances are cluster-specific and Region-specific.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 169
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DeleteAttributes
X-Amz-Date: 20161222T193851Z
User-Agent: aws-cli/1.11.30 Python/2.7.12 Darwin/16.3.0 botocore/1.4.87
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "cluster": "default",
  "attributes": [
    {
      "targetId": "arn:aws:ecs:us-west-2:123456789012:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
      "name": "stack"
```

```
    }
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 22 Dec 2016 19:38:51 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 158
Connection: keep-alive
x-amzn-RequestId: 445193ca-c87e-11e6-86db-1bd3d9928caf

{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:123456789012:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteCapacityProvider

Deletes the specified capacity provider.

> **Note**
>
> The FARGATE and FARGATE_SPOT capacity providers are reserved and can't be deleted. You can disassociate them from a cluster using either the PutClusterCapacityProviders API or by deleting the cluster.

Prior to a capacity provider being deleted, the capacity provider must be removed from the capacity provider strategy from all services. The UpdateService API can be used to remove a capacity provider from a service's capacity provider strategy. When updating a service, the `forceNewDeployment` option can be used to ensure that any tasks using the Amazon EC2 instance capacity provided by the capacity provider are transitioned to use the capacity from the remaining capacity providers. Only capacity providers that aren't associated with a cluster can be deleted. To remove a capacity provider from a cluster, you can either use PutClusterCapacityProviders or delete the cluster.

Request Syntax

```
{
   "capacityProvider": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProvider

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

Type: String

Required: Yes

Response Syntax

```
{
   "capacityProvider": {
      "autoScalingGroupProvider": {
         "autoScalingGroupArn": "string",
         "managedDraining": "string",
         "managedScaling": {
            "instanceWarmupPeriod": number,
            "maximumScalingStepSize": number,
            "minimumScalingStepSize": number,
            "status": "string",
            "targetCapacity": number
         },
         "managedTerminationProtection": "string"
      },
      "capacityProviderArn": "string",
      "name": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "updateStatus": "string",
      "updateStatusReason": "string"
   }
}
```

Response Elements

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

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

capacityProvider

The details of the capacity provider.

Type: CapacityProvider object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteCluster

Deletes the specified cluster. The cluster transitions to the `INACTIVE` state. Clusters with an `INACTIVE` status might remain discoverable in your account for a period of time. However, this behavior is subject to change in the future. We don't recommend that you rely on `INACTIVE` clusters persisting.

You must deregister all container instances from this cluster before you may delete it. You can list the container instances in a cluster with ListContainerInstances and deregister them with DeregisterContainerInstance.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

Response Syntax

```
{
   "cluster": {
      "activeServicesCount": number,
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
```

```
                "value": "string"
            }
        ],
        "id": "string",
        "status": "string",
        "type": "string"
    }
],
"attachmentsStatus": "string",
"capacityProviders": [ "string" ],
"clusterArn": "string",
"clusterName": "string",
"configuration": {
    "executeCommandConfiguration": {
        "kmsKeyId": "string",
        "logConfiguration": {
            "cloudWatchEncryptionEnabled": boolean,
            "cloudWatchLogGroupName": "string",
            "s3BucketName": "string",
            "s3EncryptionEnabled": boolean,
            "s3KeyPrefix": "string"
        },
        "logging": "string"
    }
},
"defaultCapacityProviderStrategy": [
    {
        "base": number,
        "capacityProvider": "string",
        "weight": number
    }
],
"pendingTasksCount": number,
"registeredContainerInstancesCount": number,
"runningTasksCount": number,
"serviceConnectDefaults": {
    "namespace": "string"
},
"settings": [
    {
        "name": "string",
        "value": "string"
    }
],
```

```
      "statistics": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ]
   }
}
```

Response Elements

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

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

cluster

The full description of the deleted cluster.

Type: Cluster object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterContainsContainerInstancesException

You can't delete a cluster that has registered container instances. First, deregister the container instances before you can delete the cluster. For more information, see DeregisterContainerInstance.

HTTP Status Code: 400

ClusterContainsServicesException

You can't delete a cluster that contains services. First, update the service to reduce its desired task count to 0, and then delete the service. For more information, see UpdateService and DeleteService.

HTTP Status Code: 400

ClusterContainsTasksException

You can't delete a cluster that has active tasks.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UpdateInProgressException

There's already a current Amazon ECS container agent update in progress on the container instance that's specified. If the container agent becomes disconnected while it's in a transitional

stage, such as PENDING or STAGING, the update process can get stuck in that state. However, when the agent reconnects, it resumes where it stopped previously.

HTTP Status Code: 400

Examples

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

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

Example

This example request deletes the cluster called `My-cluster`.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
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"
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteService

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

Note

When you delete a service, if there are still running tasks that require cleanup, the service status moves from `ACTIVE` to `DRAINING`, and the service is no longer visible in the console or in the ListServices API operation. After all tasks have transitioned to either `STOPPING` or `STOPPED` status, the service status moves from `DRAINING` to `INACTIVE`. Services in the `DRAINING` or `INACTIVE` status can still be viewed with the DescribeServices API operation. However, in the future, `INACTIVE` services may be cleaned up and purged from Amazon ECS record keeping, and DescribeServices calls on those services return a `ServiceNotFoundException` error.

Important

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

force

If `true`, allows you to delete a service even if it wasn't scaled down to zero tasks. It's only necessary to use this if the service uses the `REPLICA` scheduling strategy.

Type: Boolean

Required: No

service

The name of the service to delete.

Type: String

Required: Yes

Response Syntax

```
{
   "service": {
      "capacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "clusterArn": "string",
      "createdAt": number,
      "createdBy": "string",
      "deploymentConfiguration": {
         "alarms": {
            "alarmNames": [ "string" ],
            "enable": boolean,
            "rollback": boolean
```

```
         },
         "deploymentCircuitBreaker": {
            "enable": boolean,
            "rollback": boolean
         },
         "maximumPercent": number,
         "minimumHealthyPercent": number
      },
      "deploymentController": {
         "type": "string"
      },
      "deployments": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
               }
            ],
            "createdAt": number,
            "desiredCount": number,
            "failedTasks": number,
            "id": "string",
            "launchType": "string",
            "networkConfiguration": {
               "awsvpcConfiguration": {
                  "assignPublicIp": "string",
                  "securityGroups": [ "string" ],
                  "subnets": [ "string" ]
               }
            },
            "pendingCount": number,
            "platformFamily": "string",
            "platformVersion": "string",
            "rolloutState": "string",
            "rolloutStateReason": "string",
            "runningCount": number,
            "serviceConnectConfiguration": {
               "enabled": boolean,
               "logConfiguration": {
                  "logDriver": "string",
                  "options": {
                     "string" : "string"
```

```
            },
            "secretOptions": [
               {
                  "name": "string",
                  "valueFrom": "string"
               }
            ]
         },
         "namespace": "string",
         "services": [
            {
               "clientAliases": [
                  {
                     "dnsName": "string",
                     "port": number
                  }
               ],
               "discoveryName": "string",
               "ingressPortOverride": number,
               "portName": "string",
               "timeout": {
                  "idleTimeoutSeconds": number,
                  "perRequestTimeoutSeconds": number
               },
               "tls": {
                  "issuerCertificateAuthority": {
                     "awsPcaAuthorityArn": "string"
                  },
                  "kmsKey": "string",
                  "roleArn": "string"
               }
            }
         ]
      },
      "serviceConnectResources": [
         {
            "discoveryArn": "string",
            "discoveryName": "string"
         }
      ],
      "status": "string",
      "taskDefinition": "string",
      "updatedAt": number,
      "volumeConfigurations": [
```

```
            {
               "managedEBSVolume": {
                  "encrypted": boolean,
                  "filesystemType": "string",
                  "iops": number,
                  "kmsKeyId": "string",
                  "roleArn": "string",
                  "sizeInGiB": number,
                  "snapshotId": "string",
                  "tagSpecifications": [
                     {
                        "propagateTags": "string",
                        "resourceType": "string",
                        "tags": [
                           {
                              "key": "string",
                              "value": "string"
                           }
                        ]
                     }
                  ],
                  "throughput": number,
                  "volumeType": "string"
               },
               "name": "string"
            }
         ]
      }
   ],
   "desiredCount": number,
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "events": [
      {
         "createdAt": number,
         "id": "string",
         "message": "string"
      }
   ],
   "healthCheckGracePeriodSeconds": number,
   "launchType": "string",
   "loadBalancers": [
      {
         "containerName": "string",
```

```
            "containerPort": number,
            "loadBalancerName": "string",
            "targetGroupArn": "string"
         }
      ],
      "networkConfiguration": {
         "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      },
      "pendingCount": number,
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "placementStrategy": [
         {
            "field": "string",
            "type": "string"
         }
      ],
      "platformFamily": "string",
      "platformVersion": "string",
      "propagateTags": "string",
      "roleArn": "string",
      "runningCount": number,
      "schedulingStrategy": "string",
      "serviceArn": "string",
      "serviceName": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
```

```
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSets": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
               }
            ],
            "clusterArn": "string",
            "computedDesiredCount": number,
            "createdAt": number,
            "externalId": "string",
            "id": "string",
            "launchType": "string",
            "loadBalancers": [
               {
                  "containerName": "string",
                  "containerPort": number,
                  "loadBalancerName": "string",
                  "targetGroupArn": "string"
               }
            ],
            "networkConfiguration": {
               "awsvpcConfiguration": {
                  "assignPublicIp": "string",
                  "securityGroups": [ "string" ],
                  "subnets": [ "string" ]
               }
            },
            "pendingCount": number,
            "platformFamily": "string",
            "platformVersion": "string",
            "runningCount": number,
            "scale": {
               "unit": "string",
               "value": number
            },
            "serviceArn": "string",
```

```
            "serviceRegistries": [
               {
                  "containerName": "string",
                  "containerPort": number,
                  "port": number,
                  "registryArn": "string"
               }
            ],
            "stabilityStatus": "string",
            "stabilityStatusAt": number,
            "startedBy": "string",
            "status": "string",
            "tags": [
               {
                  "key": "string",
                  "value": "string"
               }
            ],
            "taskDefinition": "string",
            "taskSetArn": "string",
            "updatedAt": number
         }
      ]
   }
}
```

Response Elements

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

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

service

The full description of the deleted service.

Type: Service object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

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

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deploymentConfiguration": {
        "maximumPercent": 200,
        "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "createdAt": 1430320735.285,
        "desiredCount": 0,
        "id": "ecs-svc/9223370606534040511",
        "pendingCount": 0,
        "runningCount": 0,
```

```
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/
sleep360:27",
        "updatedAt": 1430320735.285
      }
    ],
    "desiredCount": 0,
    "events": [],
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/default/test",
    "serviceName": "test",
    "status": "DRAINING",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/sleep360:27"
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteTaskDefinitions

Deletes one or more task definitions.

You must deregister a task definition revision before you delete it. For more information, see DeregisterTaskDefinition.

When you delete a task definition revision, it is immediately transitions from the `INACTIVE` to `DELETE_IN_PROGRESS`. Existing tasks and services that reference a `DELETE_IN_PROGRESS` task definition revision continue to run without disruption. Existing services that reference a `DELETE_IN_PROGRESS` task definition revision can still scale up or down by modifying the service's desired count.

You can't use a `DELETE_IN_PROGRESS` task definition revision to run new tasks or create new services. You also can't update an existing service to reference a `DELETE_IN_PROGRESS` task definition revision.

A task definition revision will stay in `DELETE_IN_PROGRESS` status until all the associated tasks and services have been terminated.

When you delete all `INACTIVE` task definition revisions, the task definition name is not displayed in the console and not returned in the API. If a task definition revisions are in the `DELETE_IN_PROGRESS` state, the task definition name is displayed in the console and returned in the API. The task definition name is retained by Amazon ECS and the revision is incremented the next time you create a task definition with that name.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

taskDefinitions

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

You can specify up to 10 task definitions as a comma separated list.

Type: Array of strings

Required: Yes

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "taskDefinitions": [
      {
         "compatibilities": [ "string" ],
         "containerDefinitions": [
            {
               "command": [ "string" ],
               "cpu": number,
               "credentialSpecs": [ "string" ],
               "dependsOn": [
                  {
                     "condition": "string",
                     "containerName": "string"
                  }
               ],
               "disableNetworking": boolean,
               "dnsSearchDomains": [ "string" ],
               "dnsServers": [ "string" ],
               "dockerLabels": {
                  "string" : "string"
               },
               "dockerSecurityOptions": [ "string" ],
               "entryPoint": [ "string" ],
```

```
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "environmentFiles": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "essential": boolean,
            "extraHosts": [
               {
                  "hostname": "string",
                  "ipAddress": "string"
               }
            ],
            "firelensConfiguration": {
               "options": {
                  "string" : "string"
               },
               "type": "string"
            },
            "healthCheck": {
               "command": [ "string" ],
               "interval": number,
               "retries": number,
               "startPeriod": number,
               "timeout": number
            },
            "hostname": "string",
            "image": "string",
            "interactive": boolean,
            "links": [ "string" ],
            "linuxParameters": {
               "capabilities": {
                  "add": [ "string" ],
                  "drop": [ "string" ]
               },
               "devices": [
                  {
                     "containerPath": "string",
```

```
                  "hostPath": "string",
                  "permissions": [ "string" ]
               }
            ],
            "initProcessEnabled": boolean,
            "maxSwap": number,
            "sharedMemorySize": number,
            "swappiness": number,
            "tmpfs": [
               {
                  "containerPath": "string",
                  "mountOptions": [ "string" ],
                  "size": number
               }
            ]
         },
         "logConfiguration": {
            "logDriver": "string",
            "options": {
               "string" : "string"
            },
            "secretOptions": [
               {
                  "name": "string",
                  "valueFrom": "string"
               }
            ]
         },
         "memory": number,
         "memoryReservation": number,
         "mountPoints": [
            {
               "containerPath": "string",
               "readOnly": boolean,
               "sourceVolume": "string"
            }
         ],
         "name": "string",
         "portMappings": [
            {
               "appProtocol": "string",
               "containerPort": number,
               "containerPortRange": "string",
               "hostPort": number,
```

```
            "name": "string",
            "protocol": "string"
         }
      ],
      "privileged": boolean,
      "pseudoTerminal": boolean,
      "readonlyRootFilesystem": boolean,
      "repositoryCredentials": {
         "credentialsParameter": "string"
      },
      "resourceRequirements": [
         {
            "type": "string",
            "value": "string"
         }
      ],
      "secrets": [
         {
            "name": "string",
            "valueFrom": "string"
         }
      ],
      "startTimeout": number,
      "stopTimeout": number,
      "systemControls": [
         {
            "namespace": "string",
            "value": "string"
         }
      ],
      "ulimits": [
         {
            "hardLimit": number,
            "name": "string",
            "softLimit": number
         }
      ],
      "user": "string",
      "volumesFrom": [
         {
            "readOnly": boolean,
            "sourceContainer": "string"
         }
      ],
```

```
            "workingDirectory": "string"
         }
      ],
      "cpu": "string",
      "deregisteredAt": number,
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "family": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "ipcMode": "string",
      "memory": "string",
      "networkMode": "string",
      "pidMode": "string",
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "proxyConfiguration": {
         "containerName": "string",
         "properties": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "type": "string"
      },
      "registeredAt": number,
      "registeredBy": "string",
      "requiresAttributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
```

```
            }
         ],
         "requiresCompatibilities": [ "string" ],
         "revision": number,
         "runtimePlatform": {
            "cpuArchitecture": "string",
            "operatingSystemFamily": "string"
         },
         "status": "string",
         "taskDefinitionArn": "string",
         "taskRoleArn": "string",
         "volumes": [
            {
               "configuredAtLaunch": boolean,
               "dockerVolumeConfiguration": {
                  "autoprovision": boolean,
                  "driver": "string",
                  "driverOpts": {
                     "string" : "string"
                  },
                  "labels": {
                     "string" : "string"
                  },
                  "scope": "string"
               },
               "efsVolumeConfiguration": {
                  "authorizationConfig": {
                     "accessPointId": "string",
                     "iam": "string"
                  },
                  "fileSystemId": "string",
                  "rootDirectory": "string",
                  "transitEncryption": "string",
                  "transitEncryptionPort": number
               },
               "fsxWindowsFileServerVolumeConfiguration": {
                  "authorizationConfig": {
                     "credentialsParameter": "string",
                     "domain": "string"
                  },
                  "fileSystemId": "string",
                  "rootDirectory": "string"
               },
               "host": {
```

```
                "sourcePath": "string"
            },
            "name": "string"
        }
    ]
  }
]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

taskDefinitions

The list of deleted task definitions.

Type: Array of TaskDefinition objects

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request deletes the task definition named `Example-task-definition:1`.

Sample Request

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

{
  "taskDefinitions": [
    "Example-task-definition:1"
```

```
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 7 Dec 2022 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

{
  "failures": [],
  "taskDefinitions": [
    {
      "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": "DELETE_IN_PROGRESS",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/Example-
task-definition:1",
```

```
      "volumes": []
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteTaskSet

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

force

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

Type: Boolean

Required: No

service

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

Type: String

Required: Yes

taskSet

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

Type: String

Required: Yes

Response Syntax

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

```
      },
      "pendingCount": number,
      "platformFamily": "string",
      "platformVersion": "string",
      "runningCount": number,
      "scale": {
         "unit": "string",
         "value": number
      },
      "serviceArn": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "stabilityStatus": "string",
      "stabilityStatusAt": number,
      "startedBy": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSetArn": "string",
      "updatedAt": number
   }
}
```

Response Elements

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

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

taskSet

Details about the task set.

Type: TaskSet object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

TaskSetNotFoundException

The specified task set wasn't found. You can view your available task sets with DescribeTaskSets. Task sets are specific to each cluster, service and Region.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeregisterContainerInstance

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

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

Deregistering a container instance removes the instance from a cluster, but it doesn't terminate the EC2 instance. If you are finished using the instance, be sure to terminate it in the Amazon EC2 console to stop billing.

> **Note**
>
> If you terminate a running container instance, Amazon ECS automatically deregisters the instance from your cluster (stopped container instances or instances with disconnected agents aren't automatically deregistered when terminated).

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstance

The container instance ID or full ARN of the container instance to deregister. For more information about the ARN format, see Amazon Resource Name (ARN) in the *Amazon ECS Developer Guide*.

Type: String

Required: Yes

force

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

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

Type: Boolean

Required: No

Response Syntax

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

```
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
         }
      ],
      "capacityProviderName": "string",
      "containerInstanceArn": "string",
      "ec2InstanceId": "string",
      "healthStatus": {
         "details": [
            {
               "lastStatusChange": number,
               "lastUpdated": number,
               "status": "string",
               "type": "string"
            }
         ],
         "overallStatus": "string"
      },
      "pendingTasksCount": number,
      "registeredAt": number,
      "registeredResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "remainingResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
```

```
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "runningTasksCount": number,
      "status": "string",
      "statusReason": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "version": number,
      "versionInfo": {
         "agentHash": "string",
         "agentVersion": "string",
         "dockerVersion": "string"
      }
   }
}
```

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

The container instance that was deregistered.

Type: ContainerInstance object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

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

{
  "containerInstance": {
    "agentConnected": true,
    "attributes": [
      {
        "name": "com.amazonaws.ecs.capability.privileged-container"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
      },
      {
```

```
        "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
      },
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
      }
    ],
    "containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-instance/
default/c9c9a6f2-8766-464b-8805-9c57b9368fb0",
    "ec2InstanceId": "i-0c3826c9",
    "pendingTasksCount": 0,
    "registeredResources": [
      {
        "doubleValue": 0,
        "integerValue": 1024,
        "longValue": 0,
        "name": "CPU",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 995,
        "longValue": 0,
        "name": "MEMORY",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS",
        "stringSetValue": [
          "22",
          "2376",
          "2375",
          "51678"
        ],
        "type": "STRINGSET"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS_UDP",
        "stringSetValue": [],
```

```
        "type": "STRINGSET"
      }
    ],
    "remainingResources": [
      {
        "doubleValue": 0,
        "integerValue": 1024,
        "longValue": 0,
        "name": "CPU",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 995,
        "longValue": 0,
        "name": "MEMORY",
        "type": "INTEGER"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS",
        "stringSetValue": [
          "22",
          "2376",
          "2375",
          "51678"
        ],
        "type": "STRINGSET"
      },
      {
        "doubleValue": 0,
        "integerValue": 0,
        "longValue": 0,
        "name": "PORTS_UDP",
        "stringSetValue": [],
        "type": "STRINGSET"
      }
    ],
    "runningTasksCount": 0,
    "status": "INACTIVE",
    "versionInfo": {
      "agentHash": "b197edd",
```

```
      "agentVersion": "1.5.0",
      "dockerVersion": "DockerVersion: 1.7.1"
    }
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeregisterTaskDefinition

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

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

> **Note**
>
> At this time, `INACTIVE` task definitions remain discoverable in your account indefinitely. However, this behavior is subject to change in the future. We don't recommend that you rely on `INACTIVE` task definitions persisting beyond the lifecycle of any associated tasks and services.

You must deregister a task definition revision before you delete it. For more information, see DeleteTaskDefinitions.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

taskDefinition

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

Type: String

Required: Yes

Response Syntax

```
{
   "taskDefinition": {
      "compatibilities": [ "string" ],
      "containerDefinitions": [
         {
            "command": [ "string" ],
            "cpu": number,
            "credentialSpecs": [ "string" ],
            "dependsOn": [
               {
                  "condition": "string",
                  "containerName": "string"
               }
            ],
            "disableNetworking": boolean,
            "dnsSearchDomains": [ "string" ],
            "dnsServers": [ "string" ],
            "dockerLabels": {
               "string" : "string"
            },
            "dockerSecurityOptions": [ "string" ],
            "entryPoint": [ "string" ],
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "environmentFiles": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "essential": boolean,
            "extraHosts": [
               {
```

```
            "hostname": "string",
            "ipAddress": "string"
         }
      ],
      "firelensConfiguration": {
         "options": {
            "string" : "string"
         },
         "type": "string"
      },
      "healthCheck": {
         "command": [ "string" ],
         "interval": number,
         "retries": number,
         "startPeriod": number,
         "timeout": number
      },
      "hostname": "string",
      "image": "string",
      "interactive": boolean,
      "links": [ "string" ],
      "linuxParameters": {
         "capabilities": {
            "add": [ "string" ],
            "drop": [ "string" ]
         },
         "devices": [
            {
               "containerPath": "string",
               "hostPath": "string",
               "permissions": [ "string" ]
            }
         ],
         "initProcessEnabled": boolean,
         "maxSwap": number,
         "sharedMemorySize": number,
         "swappiness": number,
         "tmpfs": [
            {
               "containerPath": "string",
               "mountOptions": [ "string" ],
               "size": number
            }
         ]
```

```
            },
            "logConfiguration": {
               "logDriver": "string",
               "options": {
                  "string" : "string"
               },
               "secretOptions": [
                  {
                     "name": "string",
                     "valueFrom": "string"
                  }
               ]
            },
            "memory": number,
            "memoryReservation": number,
            "mountPoints": [
               {
                  "containerPath": "string",
                  "readOnly": boolean,
                  "sourceVolume": "string"
               }
            ],
            "name": "string",
            "portMappings": [
               {
                  "appProtocol": "string",
                  "containerPort": number,
                  "containerPortRange": "string",
                  "hostPort": number,
                  "name": "string",
                  "protocol": "string"
               }
            ],
            "privileged": boolean,
            "pseudoTerminal": boolean,
            "readonlyRootFilesystem": boolean,
            "repositoryCredentials": {
               "credentialsParameter": "string"
            },
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
```

```
            ],
            "secrets": [
               {
                  "name": "string",
                  "valueFrom": "string"
               }
            ],
            "startTimeout": number,
            "stopTimeout": number,
            "systemControls": [
               {
                  "namespace": "string",
                  "value": "string"
               }
            ],
            "ulimits": [
               {
                  "hardLimit": number,
                  "name": "string",
                  "softLimit": number
               }
            ],
            "user": "string",
            "volumesFrom": [
               {
                  "readOnly": boolean,
                  "sourceContainer": "string"
               }
            ],
            "workingDirectory": "string"
         }
      ],
      "cpu": "string",
      "deregisteredAt": number,
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "family": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
```

```
      ],
      "ipcMode": "string",
      "memory": "string",
      "networkMode": "string",
      "pidMode": "string",
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "proxyConfiguration": {
         "containerName": "string",
         "properties": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "type": "string"
      },
      "registeredAt": number,
      "registeredBy": "string",
      "requiresAttributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
         }
      ],
      "requiresCompatibilities": [ "string" ],
      "revision": number,
      "runtimePlatform": {
         "cpuArchitecture": "string",
         "operatingSystemFamily": "string"
      },
      "status": "string",
      "taskDefinitionArn": "string",
      "taskRoleArn": "string",
      "volumes": [
         {
            "configuredAtLaunch": boolean,
            "dockerVolumeConfiguration": {
```

```
            "autoprovision": boolean,
            "driver": "string",
            "driverOpts": {
               "string" : "string"
            },
            "labels": {
               "string" : "string"
            },
            "scope": "string"
         },
         "efsVolumeConfiguration": {
            "authorizationConfig": {
               "accessPointId": "string",
               "iam": "string"
            },
            "fileSystemId": "string",
            "rootDirectory": "string",
            "transitEncryption": "string",
            "transitEncryptionPort": number
         },
         "fsxWindowsFileServerVolumeConfiguration": {
            "authorizationConfig": {
               "credentialsParameter": "string",
               "domain": "string"
            },
            "fileSystemId": "string",
            "rootDirectory": "string"
         },
         "host": {
            "sourcePath": "string"
         },
         "name": "string"
      }
   ]
}
```

Response Elements

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

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

taskDefinition

The full description of the deregistered task.

Type: TaskDefinition object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
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": []
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeCapacityProviders

Describes one or more of your capacity providers.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviders

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

Type: Array of strings

Required: No

include

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

Type: Array of strings

Valid Values: `TAGS`

Required: No

maxResults

The maximum number of account setting results returned by `DescribeCapacityProviders` in paginated output. When this parameter is used, `DescribeCapacityProviders`

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

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

Response Syntax

```
{
   "capacityProviders": [
      {
         "autoScalingGroupProvider": {
            "autoScalingGroupArn": "string",
            "managedDraining": "string",
            "managedScaling": {
               "instanceWarmupPeriod": number,
               "maximumScalingStepSize": number,
               "minimumScalingStepSize": number,
               "status": "string",
               "targetCapacity": number
```

```
            },
            "managedTerminationProtection": "string"
         },
         "capacityProviderArn": "string",
         "name": "string",
         "status": "string",
         "tags": [
            {
               "key": "string",
               "value": "string"
            }
         ],
         "updateStatus": "string",
         "updateStatusReason": "string"
      }
   ],
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

capacityProviders

The list of capacity providers.

Type: Array of CapacityProvider objects

failures

Any failures associated with the call.

Type: Array of Failure objects

nextToken

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

Type: String

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeClusters

Describes one or more of your clusters.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

clusters

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

Type: Array of strings

Required: No

include

Determines whether to include additional information about the clusters in the response. If this field is omitted, this information isn't included.

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

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

If `CONFIGURATIONS` is specified, the configuration for the cluster is included.

If `STATISTICS` is specified, the task and service count is included, separated by launch type.

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

Type: Array of strings

Valid Values: ATTACHMENTS | CONFIGURATIONS | SETTINGS | STATISTICS | TAGS

Required: No

Response Syntax

```
{
   "clusters": [
      {
         "activeServicesCount": number,
         "attachments": [
            {
               "details": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "id": "string",
               "status": "string",
               "type": "string"
            }
         ],
         "attachmentsStatus": "string",
         "capacityProviders": [ "string" ],
         "clusterArn": "string",
         "clusterName": "string",
         "configuration": {
            "executeCommandConfiguration": {
               "kmsKeyId": "string",
               "logConfiguration": {
                  "cloudWatchEncryptionEnabled": boolean,
                  "cloudWatchLogGroupName": "string",
                  "s3BucketName": "string",
                  "s3EncryptionEnabled": boolean,
                  "s3KeyPrefix": "string"
               },
               "logging": "string"
            }
         },
         "defaultCapacityProviderStrategy": [
            {
               "base": number,
```

```
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "pendingTasksCount": number,
      "registeredContainerInstancesCount": number,
      "runningTasksCount": number,
      "serviceConnectDefaults": {
         "namespace": "string"
      },
      "settings": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "statistics": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ]
   }
],
"failures": [
   {
      "arn": "string",
      "detail": "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 objects

failures

Any failures associated with the call.

Type: Array of Failure objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request provides descriptive information about the default cluster.

Sample Request

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

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

Sample Response

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

{
  "clusters": [
    {
      "activeServicesCount": 1,
      "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
      "clusterName": "default",
      "pendingTasksCount": 0,
```

```
      "registeredContainerInstancesCount": 0,
      "runningTasksCount": 0,
      "status": "ACTIVE"
    }
  ],
  "failures": []
}
```

Example

This example request provides descriptive information about a cluster that has an associated capacity provider with managed scaling turned on. The capacity provider and the scaling policy that Amazon ECS creates on your behalf are included as attachments in the response.

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: 20220429T185014Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "clusters": [
    "cluster-with-capacity-provider",
    "include": ["ATTACHMENTS"]
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2022 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": [
        {
            "clusterArn": "arn:aws:ecs:us-west-2:012345678910:cluster/cluster-with-
capacity-provider",
            "clusterName": "cluster-with-capacity-provider",
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 0,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [],
            "settings": [],
            "capacityProviders": [
                "FARGATE",
                "FARGATE_SPOT",
                "Infra-ECS-Cluster-45ff394c-615f-4496-a336-8019a3b38d7a-
EC2CapacityProvider-EXAMPLE"
            ],
            "defaultCapacityProviderStrategy": [],
            "attachments": [
                {
                    "id": "bf5ecfe0-f813-4356-b862-1dbe6025c57f",
                    "type": "as_policy",
                    "status": "CREATED",
                    "details": [
                        {
                            "name": "capacityProviderName",
                            "value": "Infra-ECS-Cluster-45ff394c-615f-4496-
a336-8019a3b38d7a-EC2CapacityProvider-EXAMPLE"
                        },
                        {
                            "name": "scalingPolicyName",
                            "value": "ECSManagedAutoScalingPolicy-55c67464-7691-4310-
a725-EXAMPLE"
                        }
                    ]
                }
            ],
            "attachmentsStatus": "UPDATE_COMPLETE"
        }
    ],
    "failures": []
```

```
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeContainerInstances

Describes one or more container instances. Returns metadata about each container instance requested.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstances

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

Type: Array of strings

Required: Yes

include

Specifies whether you want to see the resource tags for the container instance. If `TAGS` is specified, the tags are included in the response. If `CONTAINER_INSTANCE_HEALTH` is specified,

the container instance health is included in the response. If this field is omitted, tags and container instance health status aren't included in the response.

Type: Array of strings

Valid Values: `TAGS | CONTAINER_INSTANCE_HEALTH`

Required: No

Response Syntax

```
{
   "containerInstances": [
      {
         "agentConnected": boolean,
         "agentUpdateStatus": "string",
         "attachments": [
            {
               "details": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "id": "string",
               "status": "string",
               "type": "string"
            }
         ],
         "attributes": [
            {
               "name": "string",
               "targetId": "string",
               "targetType": "string",
               "value": "string"
            }
         ],
         "capacityProviderName": "string",
         "containerInstanceArn": "string",
         "ec2InstanceId": "string",
         "healthStatus": {
            "details": [
               {
```

```
            "lastStatusChange": number,
            "lastUpdated": number,
            "status": "string",
            "type": "string"
         }
      ],
      "overallStatus": "string"
   },
   "pendingTasksCount": number,
   "registeredAt": number,
   "registeredResources": [
      {
         "doubleValue": number,
         "integerValue": number,
         "longValue": number,
         "name": "string",
         "stringSetValue": [ "string" ],
         "type": "string"
      }
   ],
   "remainingResources": [
      {
         "doubleValue": number,
         "integerValue": number,
         "longValue": number,
         "name": "string",
         "stringSetValue": [ "string" ],
         "type": "string"
      }
   ],
   "runningTasksCount": number,
   "status": "string",
   "statusReason": "string",
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "version": number,
   "versionInfo": {
      "agentHash": "string",
      "agentVersion": "string",
      "dockerVersion": "string"
```

```
            }
         }
      ],
      "failures": [
         {
            "arn": "string",
            "detail": "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 objects

failures

Any failures associated with the call.

Type: Array of Failure objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 64
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeContainerInstances
```

```
X-Amz-Date: 20160520T171518Z
User-Agent: aws-cli/1.10.30 Python/2.7.11 Darwin/15.4.0 botocore/1.4.17
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "containerInstances": [
        "f9cc75bb-0c94-46b9-bf6d-49d320bc1551"
    ]
}
```

Sample Response

```
{
    "containerInstances": [
        {
            "agentConnected": true,
            "attributes": [
                {
                    "name": "com.amazonaws.ecs.capability.privileged-container"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                },
                {
```

```
                    "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
                },
                {
                    "name": "com.amazonaws.ecs.capability.ecr-auth"
                }
            ],
            "containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-
instance/default/f9cc75bb-0c94-46b9-bf6d-49d320bc1551",
            "ec2InstanceId": "i-042f39dc",
            "pendingTasksCount": 0,
            "registeredResources": [
                {
                    "doubleValue": 0,
                    "integerValue": 1024,
                    "longValue": 0,
                    "name": "CPU",
                    "type": "INTEGER"
                },
                {
                    "doubleValue": 0,
                    "integerValue": 995,
                    "longValue": 0,
                    "name": "MEMORY",
                    "type": "INTEGER"
                },
                {
                    "doubleValue": 0,
                    "integerValue": 0,
                    "longValue": 0,
                    "name": "PORTS",
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET"
                },
                {
                    "doubleValue": 0,
                    "integerValue": 0,
                    "longValue": 0,
                    "name": "PORTS_UDP",
                    "stringSetValue": [],
```

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

```
                "agentHash": "0931217",
                "agentVersion": "1.9.0",
                "dockerVersion": "DockerVersion: 1.9.1"
            }
        }
    ],
    "failures": []
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeServices

Describes the specified services running in your cluster.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

include

Determines whether you want to see the resource tags for the service. If TAGS is specified, the tags are included in the response. If this field is omitted, tags aren't included in the response.

Type: Array of strings

Valid Values: TAGS

Required: No

services

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

Type: Array of strings

Required: Yes

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "services": [
      {
         "capacityProviderStrategy": [
            {
               "base": number,
               "capacityProvider": "string",
               "weight": number
            }
         ],
         "clusterArn": "string",
         "createdAt": number,
         "createdBy": "string",
         "deploymentConfiguration": {
            "alarms": {
               "alarmNames": [ "string" ],
               "enable": boolean,
               "rollback": boolean
            },
            "deploymentCircuitBreaker": {
               "enable": boolean,
               "rollback": boolean
            },
            "maximumPercent": number,
            "minimumHealthyPercent": number
         },
         "deploymentController": {
            "type": "string"
         },
         "deployments": [
```

```
{
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
   ],
   "createdAt": number,
   "desiredCount": number,
   "failedTasks": number,
   "id": "string",
   "launchType": "string",
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
   "pendingCount": number,
   "platformFamily": "string",
   "platformVersion": "string",
   "rolloutState": "string",
   "rolloutStateReason": "string",
   "runningCount": number,
   "serviceConnectConfiguration": {
      "enabled": boolean,
      "logConfiguration": {
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "namespace": "string",
      "services": [
         {
            "clientAliases": [
```

```
                        {
                           "dnsName": "string",
                           "port": number
                        }
                     ],
                     "discoveryName": "string",
                     "ingressPortOverride": number,
                     "portName": "string",
                     "timeout": {
                        "idleTimeoutSeconds": number,
                        "perRequestTimeoutSeconds": number
                     },
                     "tls": {
                        "issuerCertificateAuthority": {
                           "awsPcaAuthorityArn": "string"
                        },
                        "kmsKey": "string",
                        "roleArn": "string"
                     }
                  }
               ]
            },
            "serviceConnectResources": [
               {
                  "discoveryArn": "string",
                  "discoveryName": "string"
               }
            ],
            "status": "string",
            "taskDefinition": "string",
            "updatedAt": number,
            "volumeConfigurations": [
               {
                  "managedEBSVolume": {
                     "encrypted": boolean,
                     "filesystemType": "string",
                     "iops": number,
                     "kmsKeyId": "string",
                     "roleArn": "string",
                     "sizeInGiB": number,
                     "snapshotId": "string",
                     "tagSpecifications": [
                        {
                           "propagateTags": "string",
```

```
                        "resourceType": "string",
                        "tags": [
                           {
                              "key": "string",
                              "value": "string"
                           }
                        ]
                     }
                  ],
                  "throughput": number,
                  "volumeType": "string"
               },
               "name": "string"
            }
         ]
      }
   ],
   "desiredCount": number,
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "events": [
      {
         "createdAt": number,
         "id": "string",
         "message": "string"
      }
   ],
   "healthCheckGracePeriodSeconds": number,
   "launchType": "string",
   "loadBalancers": [
      {
         "containerName": "string",
         "containerPort": number,
         "loadBalancerName": "string",
         "targetGroupArn": "string"
      }
   ],
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
```

```
      "pendingCount": number,
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "placementStrategy": [
         {
            "field": "string",
            "type": "string"
         }
      ],
      "platformFamily": "string",
      "platformVersion": "string",
      "propagateTags": "string",
      "roleArn": "string",
      "runningCount": number,
      "schedulingStrategy": "string",
      "serviceArn": "string",
      "serviceName": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSets": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
```

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

```
                "tags": [
                   {
                      "key": "string",
                      "value": "string"
                   }
                ],
                "taskDefinition": "string",
                "taskSetArn": "string",
                "updatedAt": number
             }
          ]
       }
    ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

services

The list of services described.

Type: Array of Service objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request provides a full description of the `test-service` 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": [
    "test-service"
  ],
  "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",
      "deploymentConfiguration": {
          "maximumPercent": 200,
          "minimumHealthyPercent": 100
      },
      "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-definition/hpcc-
t2-medium:1",
          "updatedAt": 1432829320.611
```

```
        }
      ],
      "desiredCount": 4,
      "events": [],
      "loadBalancers": [],
      "pendingCount": 0,
      "runningCount": 4,
      "serviceArn": "arn:aws:ecs:us-west-2:012345678910:service/default/test-service",
      "serviceName": "test-service",
      "status": "ACTIVE",
      "taskDefinition": "arn:aws:ecs:us-west-2:012345678910:task-definition/hpcc-t2-
medium:1"
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTaskDefinition

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

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

include

Determines whether to see the resource tags for the task definition. If `TAGS` is specified, the tags are included in the response. If this field is omitted, tags aren't included in the response.

Type: Array of strings

Valid Values: `TAGS`

Required: No

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 to describe.

Type: String

Required: Yes

Response Syntax

```
{
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": {
      "compatibilities": [ "string" ],
      "containerDefinitions": [
         {
            "command": [ "string" ],
            "cpu": number,
            "credentialSpecs": [ "string" ],
            "dependsOn": [
               {
                  "condition": "string",
                  "containerName": "string"
               }
            ],
            "disableNetworking": boolean,
            "dnsSearchDomains": [ "string" ],
            "dnsServers": [ "string" ],
            "dockerLabels": {
               "string" : "string"
            },
            "dockerSecurityOptions": [ "string" ],
            "entryPoint": [ "string" ],
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "environmentFiles": [
               {
                  "type": "string",
```

```
            "value": "string"
         }
      ],
      "essential": boolean,
      "extraHosts": [
         {
            "hostname": "string",
            "ipAddress": "string"
         }
      ],
      "firelensConfiguration": {
         "options": {
            "string" : "string"
         },
         "type": "string"
      },
      "healthCheck": {
         "command": [ "string" ],
         "interval": number,
         "retries": number,
         "startPeriod": number,
         "timeout": number
      },
      "hostname": "string",
      "image": "string",
      "interactive": boolean,
      "links": [ "string" ],
      "linuxParameters": {
         "capabilities": {
            "add": [ "string" ],
            "drop": [ "string" ]
         },
         "devices": [
            {
               "containerPath": "string",
               "hostPath": "string",
               "permissions": [ "string" ]
            }
         ],
         "initProcessEnabled": boolean,
         "maxSwap": number,
         "sharedMemorySize": number,
         "swappiness": number,
         "tmpfs": [
```

```
            {
               "containerPath": "string",
               "mountOptions": [ "string" ],
               "size": number
            }
         ]
      },
      "logConfiguration": {
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "memory": number,
      "memoryReservation": number,
      "mountPoints": [
         {
            "containerPath": "string",
            "readOnly": boolean,
            "sourceVolume": "string"
         }
      ],
      "name": "string",
      "portMappings": [
         {
            "appProtocol": "string",
            "containerPort": number,
            "containerPortRange": "string",
            "hostPort": number,
            "name": "string",
            "protocol": "string"
         }
      ],
      "privileged": boolean,
      "pseudoTerminal": boolean,
      "readonlyRootFilesystem": boolean,
      "repositoryCredentials": {
         "credentialsParameter": "string"
```

```
            },
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "secrets": [
               {
                  "name": "string",
                  "valueFrom": "string"
               }
            ],
            "startTimeout": number,
            "stopTimeout": number,
            "systemControls": [
               {
                  "namespace": "string",
                  "value": "string"
               }
            ],
            "ulimits": [
               {
                  "hardLimit": number,
                  "name": "string",
                  "softLimit": number
               }
            ],
            "user": "string",
            "volumesFrom": [
               {
                  "readOnly": boolean,
                  "sourceContainer": "string"
               }
            ],
            "workingDirectory": "string"
         }
      ],
      "cpu": "string",
      "deregisteredAt": number,
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
```

```
      "family": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "ipcMode": "string",
      "memory": "string",
      "networkMode": "string",
      "pidMode": "string",
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "proxyConfiguration": {
         "containerName": "string",
         "properties": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "type": "string"
      },
      "registeredAt": number,
      "registeredBy": "string",
      "requiresAttributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
         }
      ],
      "requiresCompatibilities": [ "string" ],
      "revision": number,
      "runtimePlatform": {
         "cpuArchitecture": "string",
         "operatingSystemFamily": "string"
      },
      "status": "string",
```

```
      "taskDefinitionArn": "string",
      "taskRoleArn": "string",
      "volumes": [
         {
            "configuredAtLaunch": boolean,
            "dockerVolumeConfiguration": {
               "autoprovision": boolean,
               "driver": "string",
               "driverOpts": {
                  "string" : "string"
               },
               "labels": {
                  "string" : "string"
               },
               "scope": "string"
            },
            "efsVolumeConfiguration": {
               "authorizationConfig": {
                  "accessPointId": "string",
                  "iam": "string"
               },
               "fileSystemId": "string",
               "rootDirectory": "string",
               "transitEncryption": "string",
               "transitEncryptionPort": number
            },
            "fsxWindowsFileServerVolumeConfiguration": {
               "authorizationConfig": {
                  "credentialsParameter": "string",
                  "domain": "string"
               },
               "fileSystemId": "string",
               "rootDirectory": "string"
            },
            "host": {
               "sourcePath": "string"
            },
            "name": "string"
         }
      ]
   }
}
```

Response Elements

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

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

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

taskDefinition

The full task definition description.

Type: TaskDefinition object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 36
```

```
X-Amz-Target: AmazonEC2ContainerServiceV20141113.DescribeTaskDefinition
X-Amz-Date: 20150429T190902Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

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

Sample Response

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

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

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTasks

Describes a specified task or tasks.

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

If you have tasks with tags, and then delete the cluster, the tagged tasks are returned in the response. If you create a new cluster with the same name as the deleted cluster, the tagged tasks are not included in the response.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

include

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

Type: Array of strings

Valid Values: TAGS

Required: No

tasks

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

Type: Array of strings

Required: Yes

Response Syntax

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

```
      "availabilityZone": "string",
      "capacityProviderName": "string",
      "clusterArn": "string",
      "connectivity": "string",
      "connectivityAt": number,
      "containerInstanceArn": "string",
      "containers": [
         {
            "containerArn": "string",
            "cpu": "string",
            "exitCode": number,
            "gpuIds": [ "string" ],
            "healthStatus": "string",
            "image": "string",
            "imageDigest": "string",
            "lastStatus": "string",
            "managedAgents": [
               {
                  "lastStartedAt": number,
                  "lastStatus": "string",
                  "name": "string",
                  "reason": "string"
               }
            ],
            "memory": "string",
            "memoryReservation": "string",
            "name": "string",
            "networkBindings": [
               {
                  "bindIP": "string",
                  "containerPort": number,
                  "containerPortRange": "string",
                  "hostPort": number,
                  "hostPortRange": "string",
                  "protocol": "string"
               }
            ],
            "networkInterfaces": [
               {
                  "attachmentId": "string",
                  "ipv6Address": "string",
                  "privateIpv4Address": "string"
               }
            ],
```

```
            "reason": "string",
            "runtimeId": "string",
            "taskArn": "string"
         }
      ],
      "cpu": "string",
      "createdAt": number,
      "desiredStatus": "string",
      "enableExecuteCommand": boolean,
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionStoppedAt": number,
      "group": "string",
      "healthStatus": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "lastStatus": "string",
      "launchType": "string",
      "memory": "string",
      "overrides": {
         "containerOverrides": [
            {
               "command": [ "string" ],
               "cpu": number,
               "environment": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "environmentFiles": [
                  {
                     "type": "string",
                     "value": "string"
                  }
               ],
               "memory": number,
               "memoryReservation": number,
               "name": "string",
```

```
                "resourceRequirements": [
                   {
                      "type": "string",
                      "value": "string"
                   }
                ]
             }
          ],
          "cpu": "string",
          "ephemeralStorage": {
             "sizeInGiB": number
          },
          "executionRoleArn": "string",
          "inferenceAcceleratorOverrides": [
             {
                "deviceName": "string",
                "deviceType": "string"
             }
          ],
          "memory": "string",
          "taskRoleArn": "string"
       },
       "platformFamily": "string",
       "platformVersion": "string",
       "pullStartedAt": number,
       "pullStoppedAt": number,
       "startedAt": number,
       "startedBy": "string",
       "stopCode": "string",
       "stoppedAt": number,
       "stoppedReason": "string",
       "stoppingAt": number,
       "tags": [
          {
             "key": "string",
             "value": "string"
          }
       ],
       "taskArn": "string",
       "taskDefinitionArn": "string",
       "version": number
    }
 ]
```

```
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

tasks

The list of tasks.

Type: Array of Task objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 21:49:16 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1238
Connection: keep-alive

x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "failures": [],
  "tasks": [
    {
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-
b493-371970c6c4d6",
      "overrides": {
        "containerOverrides": [
          {
            "name": "simple-app"
          },
          {
            "name": "busybox"
          }
        ]
      },
      "lastStatus": "RUNNING",
      "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-instance/
default/5991d8da-1d59-49d2-a31f-4230f9e73140",
      "createdAt": 1476822811.295,
      "version": 0,
      "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
      "startedAt": 1476822833.998,
      "desiredStatus": "RUNNING",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/console-
sample-app-dynamic-ports:1",
      "startedBy": "ecs-svc/9223370560032507596",
      "containers": [
        {
          "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/4df26bb4-
f057-467b-a079-961675296e64",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/
default/1dc5c17a-422b-4dc4-b493-371970c6c4d6",
```

```
          "lastStatus": "RUNNING",
          "name": "simple-app",
          "networkBindings": [
            {
              "protocol": "tcp",
              "bindIP": "0.0.0.0",
              "containerPort": 80,
              "hostPort": 32774
            }
          ]
        },
        {
          "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/
e09064f7-7361-4c87-8ab9-8d073bbdbcb9",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/
default/1dc5c17a-422b-4dc4-b493-371970c6c4d6",
          "lastStatus": "RUNNING",
          "name": "busybox",
          "networkBindings": []
        }
      ]
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTaskSets

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

include

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

Type: Array of strings

Valid Values: TAGS

Required: No

service

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

Type: String

Required: Yes

taskSets

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

Type: Array of strings

Required: No

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "taskSets": [
      {
         "capacityProviderStrategy": [
            {
               "base": number,
               "capacityProvider": "string",
               "weight": number
            }
         ],
         "clusterArn": "string",
         "computedDesiredCount": number,
         "createdAt": number,
         "externalId": "string",
         "id": "string",
         "launchType": "string",
         "loadBalancers": [
            {
               "containerName": "string",
               "containerPort": number,
               "loadBalancerName": "string",
               "targetGroupArn": "string"
            }
```

```
         ],
         "networkConfiguration": {
            "awsvpcConfiguration": {
               "assignPublicIp": "string",
               "securityGroups": [ "string" ],
               "subnets": [ "string" ]
            }
         },
         "pendingCount": number,
         "platformFamily": "string",
         "platformVersion": "string",
         "runningCount": number,
         "scale": {
            "unit": "string",
            "value": number
         },
         "serviceArn": "string",
         "serviceRegistries": [
            {
               "containerName": "string",
               "containerPort": number,
               "port": number,
               "registryArn": "string"
            }
         ],
         "stabilityStatus": "string",
         "stabilityStatusAt": number,
         "startedBy": "string",
         "status": "string",
         "tags": [
            {
               "key": "string",
               "value": "string"
            }
         ],
         "taskDefinition": "string",
         "taskSetArn": "string",
         "updatedAt": number
      }
   ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

taskSets

The list of task sets described.

Type: Array of TaskSet objects

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

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

DiscoverPollEndpoint

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

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstance

The container instance ID or full ARN of the container instance. For more information about the ARN format, see Amazon Resource Name (ARN) in the *Amazon ECS Developer Guide*.

Type: String

Required: No

Response Syntax

```
{
   "endpoint": "string",
   "serviceConnectEndpoint": "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

serviceConnectEndpoint

The endpoint for the Amazon ECS agent to poll for Service Connect configuration. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteCommand

Runs a command remotely on a container within a task.

If you use a condition key in your IAM policy to refine the conditions for the policy statement, for example limit the actions to a specific cluster, you receive an `AccessDeniedException` when there is a mismatch between the condition key value and the corresponding parameter value.

For information about required permissions and considerations, see Using Amazon ECS Exec for debugging in the *Amazon ECS Developer Guide*.

Request Syntax

```
{
   "cluster": "string",
   "command": "string",
   "container": "string",
   "interactive": boolean,
   "task": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The Amazon Resource Name (ARN) or short name of the cluster the task is running in. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

command

The command to run on the container.

Type: String

Required: Yes

container

The name of the container to execute the command on. A container name only needs to be specified for tasks containing multiple containers.

Type: String

Required: No

interactive

Use this flag to run your command in interactive mode.

Type: Boolean

Required: Yes

task

The Amazon Resource Name (ARN) or ID of the task the container is part of.

Type: String

Required: Yes

Response Syntax

```
{
   "clusterArn": "string",
   "containerArn": "string",
   "containerName": "string",
   "interactive": boolean,
   "session": {
      "sessionId": "string",
      "streamUrl": "string",
      "tokenValue": "string"
   },
   "taskArn": "string"
}
```

Response Elements

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

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

clusterArn

The Amazon Resource Name (ARN) of the cluster.

Type: String

containerArn

The Amazon Resource Name (ARN) of the container.

Type: String

containerName

The name of the container.

Type: String

interactive

Determines whether the execute command session is running in interactive mode. Amazon ECS only supports initiating interactive sessions, so you must specify `true` for this value.

Type: Boolean

session

The details of the SSM session that was created for this instance of execute-command.

Type: Session object

taskArn

The Amazon Resource Name (ARN) of the task.

Type: String

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

TargetNotConnectedException

The execute command cannot run. This error can be caused by any of the following configuration issues:

- Incorrect IAM permissions
- The SSM agent is not installed or is not running
- There is an interface Amazon VPC endpoint for Amazon ECS, but there is not one for Systems Manager Session Manager

For information about how to troubleshoot the issues, see Troubleshooting issues with ECS Exec in the *Amazon Elastic Container Service Developer Guide*.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetTaskProtection

Retrieves the protection status of tasks in an Amazon ECS service.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

tasks

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

Type: Array of strings

Required: No

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
```

```
         "reason": "string"
      }
   ],
   "protectedTasks": [
      {
         "expirationDate": number,
         "protectionEnabled": boolean,
         "taskArn": "string"
      }
   ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

protectedTasks

A list of tasks with the following information.

- `taskArn`: The task ARN.
- `protectionEnabled`: The protection status of the task. If scale-in protection is turned on for a task, the value is `true`. Otherwise, it is `false`.
- `expirationDate`: The epoch time when protection for the task will expire.

Type: Array of ProtectedTask objects

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example

This example request gets the protection status for a task.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length:81
X-Amz-Target: AmazonEC2ContainerServiceV20141113.GetTaskProtection
X-Amz-Date: 20221102T190406Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "test-task-protection",
    "tasks": [
        "b8b1cf532d0e46ba8d44a40d1de16772"
    ]
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 02 Nov 2022 19:04:06 GMT
Content-Type: application/x-amz-json-1.1
Content-Length:177
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:012345678910:task/
b8b1cf532d0e46ba8d44a40d1de16772",
```

```
            "protectionEnabled": true,
            "expirationDate": 1667416437.0
        }
    ],
    "failures": []
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListAccountSettings

Lists the account settings for a specified principal.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

effectiveSettings

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

Type: Boolean

Required: No

maxResults

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

Type: Integer

Required: No

name

The name of the account setting you want to list the settings for.

Type: String

Valid Values: `serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights | fargateFIPSMode | tagResourceAuthorization | fargateTaskRetirementWaitPeriod | guardDutyActivate`

Required: No

nextToken

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

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

Type: String

Required: No

principalArn

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

> **Note**
>
> Federated users assume the account setting of the root user and can't have explicit account settings set for them.

Type: String

Required: No

value

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

Type: String

Required: No

Response Syntax

```
{
   "nextToken": "string",
   "settings": [
      {
         "name": "string",
         "principalArn": "string",
         "type": "string",
         "value": "string"
      }
   ]
}
```

Response Elements

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

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

nextToken

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

Type: String

settings

The account settings for the resource.

Type: Array of Setting objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

ListAttributes

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

attributeName

The name of the attribute to filter the results with.

Type: String

Required: No

attributeValue

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

Type: String

Required: No

cluster

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

Type: String

Required: No

maxResults

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

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

targetType

The type of the target to list attributes with.

Type: String

Valid Values: `container-instance`

Required: Yes

Response Syntax

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

Response Elements

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

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

attributes

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

Type: Array of Attribute objects

nextToken

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

Type: String

Errors

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

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 22 Dec 2016 18:16:00 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 158
Connection: keep-alive
x-amzn-RequestId: b0eb3407-c872-11e6-a3b0-295902c79de2

{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:123456789012:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

ListClusters

Returns a list of existing clusters.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

maxResults

The maximum number of cluster results that `ListClusters` returned 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 isn't used, then `ListClusters` returns up to 100 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

Response Syntax

```
{
   "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 that's 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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request lists the clusters for your account.

Sample Request

```
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"
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListContainerInstances

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

filter

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

Type: String

Required: No

maxResults

The maximum number of container instance results that `ListContainerInstances` returned 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 isn't used, then `ListContainerInstances` returns up to 100 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

status

Filters the container instances by status. For example, if you specify the `DRAINING` status, the results include only container instances that have been set to `DRAINING` using UpdateContainerInstancesState. If you don't specify this parameter, the default is to include container instances set to all states other than `INACTIVE`.

Type: String

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

Required: No

Response Syntax

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

Response Elements

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

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

containerInstanceArns

The list of container instances with full 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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.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

{"containerInstanceArns":["arn:aws:ecs:us-east-1:012345678910:container-
instance/example/1eb22c8ab33149b397dc769f68cc1319","arn:aws:ecs:us-
east-1:012345678910:container-instance/example/5cf7e311a2b74d3882650353cf3b2214"]}
```

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-east-1:012345678910:container-instance/
example/1eb22c8ab33149b397dc769f68cc1319",
    "arn:aws:ecs:us-east-1:012345678910:container-instance/
example/5cf7e311a2b74d3882650353cf3b2214"
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

ListServices

Returns a list of services. You can filter the results by cluster, launch type, and scheduling strategy.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The short name or full Amazon Resource Name (ARN) of the cluster to use when filtering the `ListServices` results. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

launchType

The launch type to use when filtering the `ListServices` results.

Type: String

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

maxResults

The maximum number of service results that `ListServices` returned 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 isn't used, then `ListServices` returns up to 10 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

schedulingStrategy

The scheduling strategy to use when filtering the `ListServices` results.

Type: String

Valid Values: `REPLICA | DAEMON`

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 ARN entries for each service that's associated with the specified cluster.

Type: Array of strings

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request lists the services in the default cluster.

Sample Request

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

{}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 29 Apr 2015 19:13:42 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 138
```

```
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListServicesByNamespace

This operation lists all of the services that are associated with a AWS Cloud Map namespace. This list might include services in different clusters. In contrast, `ListServices` can only list services in one cluster at a time. If you need to filter the list of services in a single cluster by various parameters, use `ListServices`. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

maxResults

The maximum number of service results that `ListServicesByNamespace` returns in paginated output. When this parameter is used, `ListServicesByNamespace` 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 `ListServicesByNamespace` request with the returned `nextToken` value. This value can be between 1 and 100. If this parameter isn't used, then `ListServicesByNamespace` returns up to 10 results and a `nextToken` value if applicable.

Type: Integer

Required: No

namespace

The namespace name or full Amazon Resource Name (ARN) of the AWS Cloud Map namespace to list the services in.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: Yes

nextToken

The `nextToken` value that's returned from a `ListServicesByNamespace` request. It indicates that more results are available to fulfill the request and further calls are needed. If `maxResults` is returned, it is possible the number of results is less than `maxResults`.

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 `ListServicesByNamespace` request. When the results of a `ListServicesByNamespace` request exceed `maxResults`, this value can be used to retrieve the next page of results. When there are no more results to return, this value is `null`.

Type: String

serviceArns

The list of full ARN entries for each service that's associated with the specified namespace.

Type: Array of strings

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

List the tags for an Amazon ECS resource.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

resourceArn

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

Type: String

Required: Yes

Response Syntax

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

Response Elements

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

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

tags

The tags for the resource.

Type: Array of Tag objects

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

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example lists the tags for the dev cluster.

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 39
Date: Fri, 26 Oct 2018 19:54:31 GMT

{
   "tags":[
      {
         "key":"team",
```

```
            "value":"dev"
         }
      ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTaskDefinitionFamilies

Returns a list of task definition families that are registered to your account. This list includes task definition families that no longer have any `ACTIVE` task definition revisions.

You can filter out task definition families that don't contain any `ACTIVE` task definition revisions by setting the `status` parameter to `ACTIVE`. You can also filter the results with the `familyPrefix` parameter.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

familyPrefix

The `familyPrefix` is a string that's 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 that `ListTaskDefinitionFamilies` returned 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 isn't used, then `ListTaskDefinitionFamilies` returns up to 100 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

status

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

Type: String

Valid Values: `ACTIVE | INACTIVE | ALL`

Required: No

Response Syntax

```
{
```

```
    "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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

```
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"
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTaskDefinitions

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

familyPrefix

The full family name to filter the `ListTaskDefinitions` results with. Specifying a `familyPrefix` limits 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 that `ListTaskDefinitions` returned 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 isn't used, then `ListTaskDefinitions` returns up to 100 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

sort

The order to sort the results in. 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. This is 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 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 | DELETE_IN_PROGRESS`

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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

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

{
  "familyPrefix": "hello_world"
```

```
}
```

Sample Response

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

{
  "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"
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

ListTasks

Returns a list of tasks. You can filter the results by cluster, task definition family, container instance, launch type, what IAM principal started the task, or by the desired status of the task.

Recently stopped tasks might appear in the returned results.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The short name or full Amazon Resource Name (ARN) of the cluster to use when filtering the `ListTasks` results. If you do not specify a cluster, the default cluster is assumed.

Type: String

Required: No

containerInstance

The container instance ID or full ARN of the container instance to use when filtering the `ListTasks` results. Specifying a `containerInstance` limits the results to tasks that belong to that container instance.

Type: String

Required: No

desiredStatus

The task desired status to use when filtering the `ListTasks` results. Specifying a `desiredStatus` of `STOPPED` limits the results to tasks that Amazon ECS has set the desired status to `STOPPED`. This can be useful for debugging tasks that aren't starting properly or have died or finished. The default status filter is `RUNNING`, which shows tasks that Amazon ECS has set the desired status to `RUNNING`.

> **Note**
>
> Although you can filter results based on a desired status of `PENDING`, this doesn't return any results. Amazon ECS never sets the desired status of a task to that value (only a task's `lastStatus` may have a value of `PENDING`).

Type: String

Valid Values: `RUNNING | PENDING | STOPPED`

Required: No

family

The name of the task definition family to use when filtering the `ListTasks` results. Specifying a `family` limits the results to tasks that belong to that family.

Type: String

Required: No

launchType

The launch type to use when filtering the `ListTasks` results.

Type: String

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

maxResults

The maximum number of task results that `ListTasks` returned 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 isn't used, then `ListTasks` returns up to 100 results and a `nextToken` value if applicable.

Type: Integer

Required: No

nextToken

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

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

Type: String

Required: No

serviceName

The name of the service to use when filtering the `ListTasks` results. Specifying a `serviceName` limits the results to tasks that belong to that service.

Type: String

Required: No

startedBy

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

When you specify `startedBy` as the filter, it must be the only filter that you use.

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 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.

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

{}
```

Sample Response

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

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

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

PutAccountSetting

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

If you change the root user account setting, the default settings are reset for users and roles that do not have specified individual account settings. For more information, see Account Settings in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

name

The Amazon ECS account setting name to modify.

The following are the valid values for the account setting name.

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

- `containerInstanceLongArnFormat` - When modified, the Amazon Resource Name (ARN) and resource ID format of the resource type for a specified user, role, or the root user for an account is affected. The opt-in and opt-out account setting must be set for each Amazon ECS resource separately. The ARN and resource ID format of a resource is defined by the opt-in status of the user or role that created the resource. You must turn on this setting to use Amazon ECS features such as resource tagging.
- `awsvpcTrunking` - When modified, the elastic network interface (ENI) limit for any new container instances that support the feature is changed. If `awsvpcTrunking` is turned on, any new container instances that support the feature are launched have the increased ENI limits available to them. For more information, see Elastic Network Interface Trunking in the *Amazon Elastic Container Service Developer Guide*.
- `containerInsights` - When modified, the default setting indicating whether AWS CloudWatch Container Insights is turned on for your clusters is changed. If `containerInsights` is turned on, any new clusters that are created will have Container Insights turned on unless you disable it during cluster creation. For more information, see CloudWatch Container Insights in the *Amazon Elastic Container Service Developer Guide*.
- `dualStackIPv6` - When turned on, when using a VPC in dual stack mode, your tasks using the `awsvpc` network mode can have an IPv6 address assigned. For more information on using IPv6 with tasks launched on Amazon EC2 instances, see Using a VPC in dual-stack mode. For more information on using IPv6 with tasks launched on Fargate, see Using a VPC in dual-stack mode.
- `fargateFIPSMode` - If you specify `fargateFIPSMode`, AWS Fargate FIPS 140 compliance is affected.
- `fargateTaskRetirementWaitPeriod` - When AWS determines that a security or infrastructure update is needed for an Amazon ECS task hosted on Fargate, the tasks need to be stopped and new tasks launched to replace them. Use `fargateTaskRetirementWaitPeriod` to configure the wait time to retire a Fargate task. For information about the Fargate tasks maintenance, see AWS Fargate task maintenance in the *Amazon ECS Developer Guide*.
- `tagResourceAuthorization` - Amazon ECS is introducing tagging authorization for resource creation. Users must have permissions for actions that create the resource, such as `ecsCreateCluster`. If tags are specified when you create a resource, AWS performs additional authorization to verify if users or roles have permissions to create tags. Therefore, you must grant explicit permissions to use the `ecs:TagResource` action. For more

information, see Grant permission to tag resources on creation in the *Amazon ECS Developer Guide*.

- `guardDutyActivate` - The `guardDutyActivate` parameter is read-only in Amazon ECS and indicates whether Amazon ECS Runtime Monitoring is enabled or disabled by your security administrator in your Amazon ECS account. Amazon GuardDuty controls this account setting on your behalf. For more information, see Protecting Amazon ECS workloads with Amazon ECS Runtime Monitoring.

Type: String

Valid Values: `serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights | fargateFIPSMode | tagResourceAuthorization | fargateTaskRetirementWaitPeriod | guardDutyActivate`

Required: Yes

principalArn

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

> **Note**
>
> You must use the root user when you set the Fargate wait time (`fargateTaskRetirementWaitPeriod`).
> Federated users assume the account setting of the root user and can't have explicit account settings set for them.

Type: String

Required: No

value

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

When you specify `fargateTaskRetirementWaitPeriod` for the `name`, the following are the valid values:

- `0` - AWS sends the notification, and immediately retires the affected tasks.
- `7` - AWS sends the notification, and waits 7 calendar days to retire the tasks.
- `14` - AWS sends the notification, and waits 14 calendar days to retire the tasks.

Type: String

Required: Yes

Response Syntax

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

Response Elements

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

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

setting

The current account setting for a resource.

Type: Setting object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutAccountSettingDefault

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

name

The resource name for which to modify the account setting.

The following are the valid values for the account setting name.

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

ECS resource separately. The ARN and resource ID format of a resource is defined by the opt-in status of the user or role that created the resource. You must turn on this setting to use Amazon ECS features such as resource tagging.

- `awsvpcTrunking` - When modified, the elastic network interface (ENI) limit for any new container instances that support the feature is changed. If `awsvpcTrunking` is turned on, any new container instances that support the feature are launched have the increased ENI limits available to them. For more information, see Elastic Network Interface Trunking in the *Amazon Elastic Container Service Developer Guide*.
- `containerInsights` - When modified, the default setting indicating whether AWS CloudWatch Container Insights is turned on for your clusters is changed. If `containerInsights` is turned on, any new clusters that are created will have Container Insights turned on unless you disable it during cluster creation. For more information, see CloudWatch Container Insights in the *Amazon Elastic Container Service Developer Guide*.
- `dualStackIPv6` - When turned on, when using a VPC in dual stack mode, your tasks using the `awsvpc` network mode can have an IPv6 address assigned. For more information on using IPv6 with tasks launched on Amazon EC2 instances, see Using a VPC in dual-stack mode. For more information on using IPv6 with tasks launched on Fargate, see Using a VPC in dual-stack mode.
- `fargateFIPSMode` - If you specify `fargateFIPSMode`, AWS Fargate FIPS 140 compliance is affected.
- `fargateTaskRetirementWaitPeriod` - When AWS determines that a security or infrastructure update is needed for an Amazon ECS task hosted on Fargate, the tasks need to be stopped and new tasks launched to replace them. Use `fargateTaskRetirementWaitPeriod` to configure the wait time to retire a Fargate task. For information about the Fargate tasks maintenance, see AWS Fargate task maintenance in the *Amazon ECS Developer Guide*.
- `tagResourceAuthorization` - Amazon ECS is introducing tagging authorization for resource creation. Users must have permissions for actions that create the resource, such as `ecsCreateCluster`. If tags are specified when you create a resource, AWS performs additional authorization to verify if users or roles have permissions to create tags. Therefore, you must grant explicit permissions to use the `ecs:TagResource` action. For more information, see Grant permission to tag resources on creation in the *Amazon ECS Developer Guide*.
- `guardDutyActivate` - The `guardDutyActivate` parameter is read-only in Amazon ECS and indicates whether Amazon ECS Runtime Monitoring is enabled or disabled by your

security administrator in your Amazon ECS account. Amazon GuardDuty controls this account setting on your behalf. For more information, see Protecting Amazon ECS workloads with Amazon ECS Runtime Monitoring.

Type: String

Valid Values: `serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights | fargateFIPSMode | tagResourceAuthorization | fargateTaskRetirementWaitPeriod | guardDutyActivate`

Required: Yes

value

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

When you specify `fargateTaskRetirementWaitPeriod` for the name, the following are the valid values:

- `0` - AWS sends the notification, and immediately retires the affected tasks.
- 7 - AWS sends the notification, and waits 7 calendar days to retire the tasks.
- 14 - AWS sends the notification, and waits 14 calendar days to retire the tasks.

Type: String

Required: Yes

Response Syntax

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

Response Elements

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

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

setting

The current setting for a resource.

Type: Setting object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutAttributes

Create or update an attribute on an Amazon ECS resource. If the attribute doesn't exist, it's created. If the attribute exists, its value is replaced with the specified value. To delete an attribute, use DeleteAttributes. For more information, see Attributes in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

attributes

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

Type: Array of Attribute objects

Required: Yes

cluster

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

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

attributes

The attributes applied to your resource.

Type: Array of Attribute objects

Errors

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

AttributeLimitExceededException

You can apply up to 10 custom attributes for each resource. You can view the attributes of a resource with ListAttributes. You can remove existing attributes on a resource with DeleteAttributes.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

TargetNotFoundException

The specified target wasn't found. You can view your available container instances with ListContainerInstances. Amazon ECS container instances are cluster-specific and Region-specific.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 192
X-Amz-Target: AmazonEC2ContainerServiceV20141113.PutAttributes
```

```
X-Amz-Date: 20161222T180005Z
User-Agent: aws-cli/1.11.30 Python/2.7.12 Darwin/16.3.0 botocore/1.4.87
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "cluster": "default",
  "attributes": [
    {
      "targetId": "arn:aws:ecs:us-west-2:123456789012:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
      "name": "stack",
      "value": "production"
    }
  ]
}
```

Sample Response

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

{
  "attributes": [
    {
      "name": "stack",
      "targetId": "arn:aws:ecs:us-west-2:123456789012:container-instance/1c3be8ed-
df30-47b4-8f1e-6e68ebd01f34",
      "value": "production"
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutClusterCapacityProviders

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

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

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviders

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

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

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

Type: Array of strings

Required: Yes

cluster

The short name or full Amazon Resource Name (ARN) of the cluster to modify the capacity provider settings for. If you don't specify a cluster, the default cluster is assumed.

Type: String

Required: Yes

defaultCapacityProviderStrategy

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

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

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

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

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

Type: Array of CapacityProviderStrategyItem objects

Required: Yes

Response Syntax

```
{
   "cluster": {
      "activeServicesCount": number,
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attachmentsStatus": "string",
      "capacityProviders": [ "string" ],
      "clusterArn": "string",
      "clusterName": "string",
      "configuration": {
         "executeCommandConfiguration": {
            "kmsKeyId": "string",
            "logConfiguration": {
               "cloudWatchEncryptionEnabled": boolean,
               "cloudWatchLogGroupName": "string",
               "s3BucketName": "string",
               "s3EncryptionEnabled": boolean,
               "s3KeyPrefix": "string"
            },
            "logging": "string"
         }
      },
      "defaultCapacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
```

```
            "weight": number
         }
      ],
      "pendingTasksCount": number,
      "registeredContainerInstancesCount": number,
      "runningTasksCount": number,
      "serviceConnectDefaults": {
         "namespace": "string"
      },
      "settings": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "statistics": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ]
   }
}
```

Response Elements

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

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

cluster

Details about the cluster.

Type: Cluster object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ResourceInUseException

The specified resource is in-use and can't be removed.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UpdateInProgressException

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

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RegisterContainerInstance

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

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

Request Syntax

```
{
   "attributes": [
      {
         "name": "string",
         "targetId": "string",
         "targetType": "string",
         "value": "string"
      }
   ],
   "cluster": "string",
   "containerInstanceArn": "string",
   "instanceIdentityDocument": "string",
   "instanceIdentityDocumentSignature": "string",
   "platformDevices": [
      {
         "id": "string",
         "type": "string"
      }
   ],
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "totalResources": [
      {
         "doubleValue": number,
         "integerValue": number,
```

```
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "versionInfo": {
         "agentHash": "string",
         "agentVersion": "string",
         "dockerVersion": "string"
      }
}
```

Request Parameters

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

The request accepts the following data in JSON format.

attributes

The container instance attributes that this container instance supports.

Type: Array of Attribute objects

Required: No

cluster

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

Type: String

Required: No

containerInstanceArn

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

Type: String

Required: No

instanceIdentityDocument

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

Type: String

Required: No

instanceIdentityDocumentSignature

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

Type: String

Required: No

platformDevices

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

Type: Array of PlatformDevice objects

Required: No

tags

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

The following basic restrictions apply to tags:

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

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

Type: Array of Tag objects

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

Required: No

totalResources

The resources available on the instance.

Type: Array of Resource objects

Required: No

versionInfo

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

Type: VersionInfo object

Required: No

Response Syntax

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

```
      }
   ],
   "attributes": [
      {
         "name": "string",
         "targetId": "string",
         "targetType": "string",
         "value": "string"
      }
   ],
   "capacityProviderName": "string",
   "containerInstanceArn": "string",
   "ec2InstanceId": "string",
   "healthStatus": {
      "details": [
         {
            "lastStatusChange": number,
            "lastUpdated": number,
            "status": "string",
            "type": "string"
         }
      ],
      "overallStatus": "string"
   },
   "pendingTasksCount": number,
   "registeredAt": number,
   "registeredResources": [
      {
         "doubleValue": number,
         "integerValue": number,
         "longValue": number,
         "name": "string",
         "stringSetValue": [ "string" ],
         "type": "string"
      }
   ],
   "remainingResources": [
      {
         "doubleValue": number,
         "integerValue": number,
         "longValue": number,
         "name": "string",
         "stringSetValue": [ "string" ],
         "type": "string"
```

```
         }
      ],
      "runningTasksCount": number,
      "status": "string",
      "statusReason": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "version": number,
      "versionInfo": {
         "agentHash": "string",
         "agentVersion": "string",
         "dockerVersion": "string"
      }
   }
}
```

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

The container instance that was registered.

Type: ContainerInstance object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RegisterTaskDefinition

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

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

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

Request Syntax

```
{
   "containerDefinitions": [
      {
         "command": [ "string" ],
         "cpu": number,
         "credentialSpecs": [ "string" ],
         "dependsOn": [
            {
               "condition": "string",
               "containerName": "string"
            }
         ],
         "disableNetworking": boolean,
         "dnsSearchDomains": [ "string" ],
         "dnsServers": [ "string" ],
         "dockerLabels": {
            "string" : "string"
         },
         "dockerSecurityOptions": [ "string" ],
         "entryPoint": [ "string" ],
```

```
      "environment": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "environmentFiles": [
         {
            "type": "string",
            "value": "string"
         }
      ],
      "essential": boolean,
      "extraHosts": [
         {
            "hostname": "string",
            "ipAddress": "string"
         }
      ],
      "firelensConfiguration": {
         "options": {
            "string" : "string"
         },
         "type": "string"
      },
      "healthCheck": {
         "command": [ "string" ],
         "interval": number,
         "retries": number,
         "startPeriod": number,
         "timeout": number
      },
      "hostname": "string",
      "image": "string",
      "interactive": boolean,
      "links": [ "string" ],
      "linuxParameters": {
         "capabilities": {
            "add": [ "string" ],
            "drop": [ "string" ]
         },
         "devices": [
            {
               "containerPath": "string",
```

```
                "hostPath": "string",
                "permissions": [ "string" ]
            }
         ],
         "initProcessEnabled": boolean,
         "maxSwap": number,
         "sharedMemorySize": number,
         "swappiness": number,
         "tmpfs": [
            {
               "containerPath": "string",
               "mountOptions": [ "string" ],
               "size": number
            }
         ]
      },
      "logConfiguration": {
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "memory": number,
      "memoryReservation": number,
      "mountPoints": [
         {
            "containerPath": "string",
            "readOnly": boolean,
            "sourceVolume": "string"
         }
      ],
      "name": "string",
      "portMappings": [
         {
            "appProtocol": "string",
            "containerPort": number,
            "containerPortRange": "string",
            "hostPort": number,
```

```
            "name": "string",
            "protocol": "string"
         }
      ],
      "privileged": boolean,
      "pseudoTerminal": boolean,
      "readonlyRootFilesystem": boolean,
      "repositoryCredentials": {
         "credentialsParameter": "string"
      },
      "resourceRequirements": [
         {
            "type": "string",
            "value": "string"
         }
      ],
      "secrets": [
         {
            "name": "string",
            "valueFrom": "string"
         }
      ],
      "startTimeout": number,
      "stopTimeout": number,
      "systemControls": [
         {
            "namespace": "string",
            "value": "string"
         }
      ],
      "ulimits": [
         {
            "hardLimit": number,
            "name": "string",
            "softLimit": number
         }
      ],
      "user": "string",
      "volumesFrom": [
         {
            "readOnly": boolean,
            "sourceContainer": "string"
         }
      ],
```

```
            "workingDirectory": "string"
         }
      ],
      "cpu": "string",
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "family": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "ipcMode": "string",
      "memory": "string",
      "networkMode": "string",
      "pidMode": "string",
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "proxyConfiguration": {
         "containerName": "string",
         "properties": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "type": "string"
      },
      "requiresCompatibilities": [ "string" ],
      "runtimePlatform": {
         "cpuArchitecture": "string",
         "operatingSystemFamily": "string"
      },
      "tags": [
         {
            "key": "string",
            "value": "string"
```

```
      }
   ],
   "taskRoleArn": "string",
   "volumes": [
      {
         "configuredAtLaunch": boolean,
         "dockerVolumeConfiguration": {
            "autoprovision": boolean,
            "driver": "string",
            "driverOpts": {
               "string" : "string"
            },
            "labels": {
               "string" : "string"
            },
            "scope": "string"
         },
         "efsVolumeConfiguration": {
            "authorizationConfig": {
               "accessPointId": "string",
               "iam": "string"
            },
            "fileSystemId": "string",
            "rootDirectory": "string",
            "transitEncryption": "string",
            "transitEncryptionPort": number
         },
         "fsxWindowsFileServerVolumeConfiguration": {
            "authorizationConfig": {
               "credentialsParameter": "string",
               "domain": "string"
            },
            "fileSystemId": "string",
            "rootDirectory": "string"
         },
         "host": {
            "sourcePath": "string"
         },
         "name": "string"
      }
   ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

containerDefinitions

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

Type: Array of ContainerDefinition objects

Required: Yes

cpu

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

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

If you're using the EC2 launch type, this field is optional. Supported values are between 128 CPU units (`0.125` vCPUs) and `10240` CPU units (`10` vCPUs). If you do not specify a value, the parameter is ignored.

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

The CPU units cannot be less than 1 vCPU when you use Windows containers on Fargate.

- 256 (.25 vCPU) - Available `memory` values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
- 512 (.5 vCPU) - Available `memory` values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
- 1024 (1 vCPU) - Available `memory` values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)

- 2048 (2 vCPU) - Available `memory` values: 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
- 4096 (4 vCPU) - Available `memory` values: 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)
- 8192 (8 vCPU) - Available `memory` values: 16 GB and 60 GB in 4 GB increments

  This option requires Linux platform `1.4.0` or later.
- 16384 (16vCPU) - Available `memory` values: 32GB and 120 GB in 8 GB increments

  This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

ephemeralStorage

The amount of ephemeral storage to allocate for the task. This parameter is used to expand the total amount of ephemeral storage available, beyond the default amount, for tasks hosted on AWS Fargate. For more information, see Using data volumes in tasks in the *Amazon ECS Developer Guide*.

> **Note**
>
> For tasks using the Fargate launch type, the task requires the following platforms:
>
> - Linux platform version `1.4.0` or later.
> - Windows platform version `1.0.0` or later.

Type: EphemeralStorage object

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution role that grants the Amazon ECS container agent permission to make AWS API calls on your behalf. The task execution IAM role is required depending on the requirements of your task. For more information, see Amazon ECS task execution IAM role in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

family

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

Type: String

Required: Yes

inferenceAccelerators

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

Type: Array of InferenceAccelerator objects

Required: No

ipcMode

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

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

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

- For tasks that use the `host` IPC mode, IPC namespace related `systemControls` are not supported.
- For tasks that use the `task` IPC mode, IPC namespace related `systemControls` will apply to all containers within a task.

> **Note**
>
> This parameter is not supported for Windows containers or tasks run on AWS Fargate.

Type: String

Valid Values: `host | task | none`

Required: No

memory

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

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

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

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

The CPU units cannot be less than 1 vCPU when you use Windows containers on Fargate.

- 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available `cpu` values: 256 (.25 vCPU)
- 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available `cpu` values: 512 (.5 vCPU)
- 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available `cpu` values: 1024 (1 vCPU)
- Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB) - Available `cpu` values: 2048 (2 vCPU)
- Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB) - Available `cpu` values: 4096 (4 vCPU)
- Between 16 GB and 60 GB in 4 GB increments - Available `cpu` values: 8192 (8 vCPU)

This option requires Linux platform `1.4.0` or later.

- Between 32GB and 120 GB in 8 GB increments - Available `cpu` values: 16384 (16 vCPU)

  This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

networkMode

The Docker networking mode to use for the containers in the task. The valid values are `none`, `bridge`, `awsvpc`, and `host`. If no network mode is specified, the default is `bridge`.

For Amazon ECS tasks on Fargate, the `awsvpc` network mode is required. For Amazon ECS tasks on Amazon EC2 Linux instances, any network mode can be used. For Amazon ECS tasks on Amazon EC2 Windows instances, `<default>` or `awsvpc` can be used. If the network mode is set to `none`, you cannot specify port mappings in your container definitions, and the tasks containers do not have external connectivity. The `host` and `awsvpc` network modes offer the highest networking performance for containers because they use the EC2 network stack instead of the virtualized network stack provided by the `bridge` mode.

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

> **Important**
>
> When using the `host` network mode, you should not run containers using the root user (UID 0). It is considered best practice to use a non-root user.

If the network mode is `awsvpc`, the task is allocated an elastic network interface, and you must specify a NetworkConfiguration value when you create a service or run a task with the task definition. For more information, see Task Networking in the *Amazon Elastic Container Service Developer Guide*.

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

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

Type: String

Valid Values: `bridge | host | awsvpc | none`

Required: No

pidMode

The process namespace to use for the containers in the task. The valid values are `host` or `task`. On Fargate for Linux containers, the only valid value is `task`. For example, monitoring sidecars might need `pidMode` to access information about other containers running in the same task.

If `host` is specified, all containers within the tasks that specified the `host` PID mode on the same container instance share the same process namespace with the host Amazon EC2 instance.

If `task` is specified, all containers within the specified task share the same process namespace.

If no value is specified, the default is a private namespace for each container. For more information, see PID settings in the *Docker run reference*.

If the `host` PID mode is used, there's a heightened risk of undesired process namespace exposure. For more information, see Docker security.

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

> **Note**
>
> This parameter is only supported for tasks that are hosted on AWS Fargate if the tasks are using platform version `1.4.0` or later (Linux). This isn't supported for Windows containers on Fargate.

Type: String

Valid Values: `host | task`

Required: No

placementConstraints

An array of placement constraint objects to use for the task. You can specify a maximum of 10 constraints for each task. This limit includes constraints in the task definition and those specified at runtime.

Type: Array of TaskDefinitionPlacementConstraint objects

Required: No

proxyConfiguration

The configuration details for the App Mesh proxy.

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

Type: ProxyConfiguration object

Required: No

requiresCompatibilities

The task launch type that Amazon ECS validates the task definition against. A client exception is returned if the task definition doesn't validate against the compatibilities specified. If no value is specified, the parameter is omitted from the response.

Type: Array of strings

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

runtimePlatform

The operating system that your tasks definitions run on. A platform family is specified only for tasks using the Fargate launch type.

Type: RuntimePlatform object

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

taskRoleArn

The short name or full Amazon Resource Name (ARN) of the IAM role that containers in this task can assume. All containers in this task are granted the permissions that are specified in this role. For more information, see IAM Roles for Tasks in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

volumes

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

Type: Array of Volume objects

Required: No

Response Syntax

```
{
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": {
      "compatibilities": [ "string" ],
      "containerDefinitions": [
         {
            "command": [ "string" ],
            "cpu": number,
            "credentialSpecs": [ "string" ],
            "dependsOn": [
               {
                  "condition": "string",
                  "containerName": "string"
               }
            ],
            "disableNetworking": boolean,
            "dnsSearchDomains": [ "string" ],
            "dnsServers": [ "string" ],
            "dockerLabels": {
               "string" : "string"
            },
            "dockerSecurityOptions": [ "string" ],
            "entryPoint": [ "string" ],
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "environmentFiles": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "essential": boolean,
```

```
            "extraHosts": [
               {
                  "hostname": "string",
                  "ipAddress": "string"
               }
            ],
            "firelensConfiguration": {
               "options": {
                  "string" : "string"
               },
               "type": "string"
            },
            "healthCheck": {
               "command": [ "string" ],
               "interval": number,
               "retries": number,
               "startPeriod": number,
               "timeout": number
            },
            "hostname": "string",
            "image": "string",
            "interactive": boolean,
            "links": [ "string" ],
            "linuxParameters": {
               "capabilities": {
                  "add": [ "string" ],
                  "drop": [ "string" ]
               },
               "devices": [
                  {
                     "containerPath": "string",
                     "hostPath": "string",
                     "permissions": [ "string" ]
                  }
               ],
               "initProcessEnabled": boolean,
               "maxSwap": number,
               "sharedMemorySize": number,
               "swappiness": number,
               "tmpfs": [
                  {
                     "containerPath": "string",
                     "mountOptions": [ "string" ],
                     "size": number
```

```
            }
         ]
      },
      "logConfiguration": {
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "memory": number,
      "memoryReservation": number,
      "mountPoints": [
         {
            "containerPath": "string",
            "readOnly": boolean,
            "sourceVolume": "string"
         }
      ],
      "name": "string",
      "portMappings": [
         {
            "appProtocol": "string",
            "containerPort": number,
            "containerPortRange": "string",
            "hostPort": number,
            "name": "string",
            "protocol": "string"
         }
      ],
      "privileged": boolean,
      "pseudoTerminal": boolean,
      "readonlyRootFilesystem": boolean,
      "repositoryCredentials": {
         "credentialsParameter": "string"
      },
      "resourceRequirements": [
         {
            "type": "string",
```

```
                "value": "string"
            }
        ],
        "secrets": [
            {
                "name": "string",
                "valueFrom": "string"
            }
        ],
        "startTimeout": number,
        "stopTimeout": number,
        "systemControls": [
            {
                "namespace": "string",
                "value": "string"
            }
        ],
        "ulimits": [
            {
                "hardLimit": number,
                "name": "string",
                "softLimit": number
            }
        ],
        "user": "string",
        "volumesFrom": [
            {
                "readOnly": boolean,
                "sourceContainer": "string"
            }
        ],
        "workingDirectory": "string"
    }
],
"cpu": "string",
"deregisteredAt": number,
"ephemeralStorage": {
    "sizeInGiB": number
},
"executionRoleArn": "string",
"family": "string",
"inferenceAccelerators": [
    {
        "deviceName": "string",
```

```
            "deviceType": "string"
         }
      ],
      "ipcMode": "string",
      "memory": "string",
      "networkMode": "string",
      "pidMode": "string",
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "proxyConfiguration": {
         "containerName": "string",
         "properties": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "type": "string"
      },
      "registeredAt": number,
      "registeredBy": "string",
      "requiresAttributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
         }
      ],
      "requiresCompatibilities": [ "string" ],
      "revision": number,
      "runtimePlatform": {
         "cpuArchitecture": "string",
         "operatingSystemFamily": "string"
      },
      "status": "string",
      "taskDefinitionArn": "string",
      "taskRoleArn": "string",
      "volumes": [
         {
```

```
            "configuredAtLaunch": boolean,
            "dockerVolumeConfiguration": {
               "autoprovision": boolean,
               "driver": "string",
               "driverOpts": {
                  "string" : "string"
               },
               "labels": {
                  "string" : "string"
               },
               "scope": "string"
            },
            "efsVolumeConfiguration": {
               "authorizationConfig": {
                  "accessPointId": "string",
                  "iam": "string"
               },
               "fileSystemId": "string",
               "rootDirectory": "string",
               "transitEncryption": "string",
               "transitEncryptionPort": number
            },
            "fsxWindowsFileServerVolumeConfiguration": {
               "authorizationConfig": {
                  "credentialsParameter": "string",
                  "domain": "string"
               },
               "fileSystemId": "string",
               "rootDirectory": "string"
            },
            "host": {
               "sourcePath": "string"
            },
            "name": "string"
         }
      ]
   }
}
```

Response Elements

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

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

tags

The list of tags associated with the task definition.

Type: Array of Tag objects

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

taskDefinition

The full description of the registered task definition.

Type: TaskDefinition object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

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

{
  "networkMode": "host",
  "containerDefinitions": [
    {
      "name": "wordpress",
      "links": [
        "mysql"
      ],
      "image": "wordpress",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80
        }
```

```
      ],
      "memory": 500,
      "cpu": 10
    },
    {
      "name": "mysql",
      "image": "mysql",
      "cpu": 10,
      "environment": [
        {
          "name": "MYSQL_ROOT_PASSWORD",
          "value": "password"
        }
      ],
      "memory": 500,
      "healthCheck": {
          "retries": 3,
          "command": [
              "CMD-SHELL",
              "curl -f http://localhost:8080/ || exit 1"
          ],
          "timeout": 5,
          "interval": 30,
      },
      "essential": true
    }
  ],
  "family": "hello_world"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 12 Aug 2016 22:17:20 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 714
Connection: keep-alive
x-amzn-RequestId: 896d7e0f-60da-11e6-8e21-55c97a4b6423

{
  "taskDefinition": {
    "containerDefinitions": [
```

```
      {
        "cpu": 10,
        "environment": [],
        "healthCheck": {
              "retries": 3,
              "command": [
                  "CMD-SHELL",
                  "curl -f http://localhost:8080/ || exit 1"
              ],
              "timeout": 5,
              "interval": 30,
              "startPeriod": null
        },
        "essential": true,
        "image": "wordpress",
        "links": [
          "mysql"
        ],
        "memory": 500,
        "mountPoints": [],
        "name": "wordpress",
        "portMappings": [
          {
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
          }
        ],
        "volumesFrom": []
      },
      {
        "cpu": 10,
        "environment": [
          {
            "name": "MYSQL_ROOT_PASSWORD",
            "value": "password"
          }
        ],
        "essential": true,
        "image": "mysql",
        "memory": 500,
        "mountPoints": [],
        "name": "mysql",
        "portMappings": [],
```

```
        "volumesFrom": []
      }
    ],
    "family": "hello_world",
    "networkMode": "host",
    "runtimePlatform":LINUX
    "requiresAttributes": [
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
      }
    ],
    "revision": 4,
    "status": "ACTIVE",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:4",
    "volumes": []
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RunTask

Starts a new task using the specified task definition.

> **Note**
>
> On March 21, 2024, a change was made to resolve the task definition revision before authorization. When a task definition revision is not specified, authorization will occur using the latest revision of a task definition.

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

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

Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

You can attach Amazon EBS volumes to Amazon ECS tasks by configuring the volume when creating or updating a service. For more infomation, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*.

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

To manage eventual consistency, you can do the following:

- Confirm the state of the resource before you run a command to modify it. Run the DescribeTasks command using an exponential backoff algorithm to ensure that you allow enough time for

the previous command to propagate through the system. To do this, run the DescribeTasks command repeatedly, starting with a couple of seconds of wait time and increasing gradually up to five minutes of wait time.

- Add wait time between subsequent commands, even if the DescribeTasks command returns an accurate response. Apply an exponential backoff algorithm starting with a couple of seconds of wait time, and increase gradually up to about five minutes of wait time.

Request Syntax

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

```
            ],
            "environmentFiles": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "memory": number,
            "memoryReservation": number,
            "name": "string",
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
            ]
         }
      ],
      "cpu": "string",
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "inferenceAcceleratorOverrides": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "memory": "string",
      "taskRoleArn": "string"
   },
   "placementConstraints": [
      {
         "expression": "string",
         "type": "string"
      }
   ],
   "placementStrategy": [
      {
         "field": "string",
         "type": "string"
      }
   ],
```

```
   "platformVersion": "string",
   "propagateTags": "string",
   "referenceId": "string",
   "startedBy": "string",
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": "string",
   "volumeConfigurations": [
      {
         "managedEBSVolume": {
            "encrypted": boolean,
            "filesystemType": "string",
            "iops": number,
            "kmsKeyId": "string",
            "roleArn": "string",
            "sizeInGiB": number,
            "snapshotId": "string",
            "tagSpecifications": [
               {
                  "propagateTags": "string",
                  "resourceType": "string",
                  "tags": [
                     {
                        "key": "string",
                        "value": "string"
                     }
                  ]
               }
            ],
            "terminationPolicy": {
               "deleteOnTermination": boolean
            },
            "throughput": number,
            "volumeType": "string"
         },
         "name": "string"
      }
   ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviderStrategy

The capacity provider strategy to use for the task.

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

When you use cluster auto scaling, you must specify `capacityProviderStrategy` and not `launchType`.

A capacity provider strategy may contain a maximum of 6 capacity providers.

Type: Array of CapacityProviderStrategyItem objects

Required: No

clientToken

An identifier that you provide to ensure the idempotency of the request. It must be unique and is case sensitive. Up to 64 characters are allowed. The valid characters are characters in the range of 33-126, inclusive. For more information, see Ensuring idempotency.

Type: String

Required: No

cluster

The short name or full Amazon Resource Name (ARN) of the cluster 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 to place on your cluster. You can specify up to 10 tasks for each call.

Type: Integer

Required: No

enableECSManagedTags

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

Type: Boolean

Required: No

enableExecuteCommand

Determines whether to use the execute command functionality for the containers in this task. If `true`, this enables execute command functionality on all containers in the task.

If `true`, then the task definition must have a task role, or you must provide one as an override.

Type: Boolean

Required: No

group

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

Type: String

Required: No

launchType

The infrastructure to run your standalone task on. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

The `FARGATE` launch type runs your tasks on AWS Fargate On-Demand infrastructure.

> **Note**
>
> Fargate Spot infrastructure is available for use but a capacity provider strategy must be used. For more information, see AWS Fargate capacity providers in the *Amazon ECS Developer Guide*.

The EC2 launch type runs your tasks on Amazon EC2 instances registered to your cluster.

The EXTERNAL launch type runs your tasks on your on-premises server or virtual machine (VM) capacity registered to your cluster.

A task can use either a launch type or a capacity provider strategy. If a `launchType` is specified, the `capacityProviderStrategy` parameter must be omitted.

When you use cluster auto scaling, you must specify `capacityProviderStrategy` and not `launchType`.

Type: String

Valid Values: `EC2` | `FARGATE` | `EXTERNAL`

Required: No

networkConfiguration

The network configuration for the task. This parameter is required for task definitions that use the `awsvpc` network mode to receive their own elastic network interface, and it isn't supported for other network modes. For more information, see Task networking in the *Amazon Elastic Container Service Developer Guide*.

Type: NetworkConfiguration object

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's 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.

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

Type: TaskOverride object

Required: No

placementConstraints

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

Type: Array of PlacementConstraint objects

Required: No

placementStrategy

The placement strategy objects to use for the task. You can specify a maximum of 5 strategy rules for each task.

Type: Array of PlacementStrategy objects

Required: No

platformVersion

The platform version the task uses. A platform version is only specified for tasks hosted on Fargate. If one isn't specified, the `LATEST` platform version is used. For more information, see AWS Fargate platform versions in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

propagateTags

Specifies whether to propagate the tags from the task definition to the task. If no value is specified, the tags aren't propagated. Tags can only be propagated to the task during task creation. To add tags to a task after task creation, use the TagResource API action.

> **Note**
>
> An error will be received if you specify the `SERVICE` option when running a task.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

referenceId

The reference ID to use for the task. The reference ID can have a maximum length of 1024 characters.

Type: String

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 call with the `startedBy` value. Up to 128 letters (uppercase and lowercase), numbers, hyphens (-), and underscores (_) are allowed.

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

Type: String

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

taskDefinition

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

The full ARN value must match the value that you specified as the `Resource` of the principal's permissions policy.

When you specify a task definition, you must either specify a specific revision, or all revisions in the ARN.

To specify a specific revision, include the revision number in the ARN. For example, to specify revision 2, use `arn:aws:ecs:us-east-1:111122223333:task-definition/TaskFamilyName:2`.

To specify all revisions, use the wildcard (*) in the ARN. For example, to specify all revisions, use `arn:aws:ecs:us-east-1:111122223333:task-definition/TaskFamilyName:*`.

For more information, see Policy Resources for Amazon ECS in the Amazon Elastic Container Service Developer Guide.

Type: String

Required: Yes

volumeConfigurations

The details of the volume that was `configuredAtLaunch`. You can configure the size, volumeType, IOPS, throughput, snapshot and encryption in in TaskManagedEBSVolumeConfiguration. The `name` of the volume must match the `name` from the task definition.

Type: Array of TaskVolumeConfiguration objects

Required: No

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "tasks": [
      {
         "attachments": [
            {
               "details": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "id": "string",
               "status": "string",
               "type": "string"
            }
         ],
         "attributes": [
            {
               "name": "string",
               "targetId": "string",
               "targetType": "string",
               "value": "string"
            }
         ],
         "availabilityZone": "string",
         "capacityProviderName": "string",
         "clusterArn": "string",
         "connectivity": "string",
         "connectivityAt": number,
         "containerInstanceArn": "string",
         "containers": [
            {
               "containerArn": "string",
               "cpu": "string",
```

```
            "exitCode": number,
            "gpuIds": [ "string" ],
            "healthStatus": "string",
            "image": "string",
            "imageDigest": "string",
            "lastStatus": "string",
            "managedAgents": [
               {
                  "lastStartedAt": number,
                  "lastStatus": "string",
                  "name": "string",
                  "reason": "string"
               }
            ],
            "memory": "string",
            "memoryReservation": "string",
            "name": "string",
            "networkBindings": [
               {
                  "bindIP": "string",
                  "containerPort": number,
                  "containerPortRange": "string",
                  "hostPort": number,
                  "hostPortRange": "string",
                  "protocol": "string"
               }
            ],
            "networkInterfaces": [
               {
                  "attachmentId": "string",
                  "ipv6Address": "string",
                  "privateIpv4Address": "string"
               }
            ],
            "reason": "string",
            "runtimeId": "string",
            "taskArn": "string"
         }
      ],
      "cpu": "string",
      "createdAt": number,
      "desiredStatus": "string",
      "enableExecuteCommand": boolean,
      "ephemeralStorage": {
```

```
            "sizeInGiB": number
         },
         "executionStoppedAt": number,
         "group": "string",
         "healthStatus": "string",
         "inferenceAccelerators": [
            {
               "deviceName": "string",
               "deviceType": "string"
            }
         ],
         "lastStatus": "string",
         "launchType": "string",
         "memory": "string",
         "overrides": {
            "containerOverrides": [
               {
                  "command": [ "string" ],
                  "cpu": number,
                  "environment": [
                     {
                        "name": "string",
                        "value": "string"
                     }
                  ],
                  "environmentFiles": [
                     {
                        "type": "string",
                        "value": "string"
                     }
                  ],
                  "memory": number,
                  "memoryReservation": number,
                  "name": "string",
                  "resourceRequirements": [
                     {
                        "type": "string",
                        "value": "string"
                     }
                  ]
               }
            ],
            "cpu": "string",
            "ephemeralStorage": {
```

```
               "sizeInGiB": number
            },
            "executionRoleArn": "string",
            "inferenceAcceleratorOverrides": [
               {
                  "deviceName": "string",
                  "deviceType": "string"
               }
            ],
            "memory": "string",
            "taskRoleArn": "string"
         },
         "platformFamily": "string",
         "platformVersion": "string",
         "pullStartedAt": number,
         "pullStoppedAt": number,
         "startedAt": number,
         "startedBy": "string",
         "stopCode": "string",
         "stoppedAt": number,
         "stoppedReason": "string",
         "stoppingAt": number,
         "tags": [
            {
               "key": "string",
               "value": "string"
            }
         ],
         "taskArn": "string",
         "taskDefinitionArn": "string",
         "version": number
      }
   ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

For information about how to address failures, see Service event messages and API failure reasons in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of Failure objects

tasks

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

Type: Array of Task objects

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

BlockedException

Your AWS account was blocked. For more information, contact AWS Support.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

ConflictException

The RunTask request could not be processed due to conflicts. The provided clientToken is already in use with a different RunTask request. The resourceIds are the existing task ARNs which are already associated with the clientToken.

To fix this issue:

- Run RunTask with a unique clientToken.
- Run RunTask with the clientToken and the original set of parameters

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version doesn't satisfy the required capabilities of the task definition.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version doesn't exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

{
  "count": 1,
  "taskDefinition": "hello_world",
  ""clientToken": "550e8400-e29b-41d4-a716-446655440000"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 21:57:40 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1025
Connection: keep-alive
```

```
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

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

```
      "version": 1
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartTask

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

> **Note**
>
> On March 21, 2024, a change was made to resolve the task definition revision before authorization. When a task definition revision is not specified, authorization will occur using the latest revision of a task definition.

Starting April 15, 2023, AWS will not onboard new customers to Amazon Elastic Inference (EI), and will help current customers migrate their workloads to options that offer better price and performance. After April 15, 2023, new customers will not be able to launch instances with Amazon EI accelerators in Amazon SageMaker, Amazon ECS, or Amazon EC2. However, customers who have used Amazon EI at least once during the past 30-day period are considered current customers and will be able to continue using the service.

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

You can attach Amazon EBS volumes to Amazon ECS tasks by configuring the volume when creating or updating a service. For more infomation, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*.

Request Syntax

```
{
   "cluster": "string",
   "containerInstances": [ "string" ],
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "group": "string",
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
   "overrides": {
```

```
      "containerOverrides": [
         {
            "command": [ "string" ],
            "cpu": number,
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "environmentFiles": [
               {
                  "type": "string",
                  "value": "string"
               }
            ],
            "memory": number,
            "memoryReservation": number,
            "name": "string",
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
            ]
         }
      ],
      "cpu": "string",
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "inferenceAcceleratorOverrides": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "memory": "string",
      "taskRoleArn": "string"
   },
   "propagateTags": "string",
   "referenceId": "string",
   "startedBy": "string",
```

```
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskDefinition": "string",
   "volumeConfigurations": [
      {
         "managedEBSVolume": {
            "encrypted": boolean,
            "filesystemType": "string",
            "iops": number,
            "kmsKeyId": "string",
            "roleArn": "string",
            "sizeInGiB": number,
            "snapshotId": "string",
            "tagSpecifications": [
               {
                  "propagateTags": "string",
                  "resourceType": "string",
                  "tags": [
                     {
                        "key": "string",
                        "value": "string"
                     }
                  ]
               }
            ],
            "terminationPolicy": {
               "deleteOnTermination": boolean
            },
            "throughput": number,
            "volumeType": "string"
         },
         "name": "string"
      }
   ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstances

The container instance IDs or full ARN entries for the container instances where you would like to place your task. You can specify up to 10 container instances.

Type: Array of strings

Required: Yes

enableECSManagedTags

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

Type: Boolean

Required: No

enableExecuteCommand

Whether or not the execute command functionality is turned on for the task. If `true`, this turns on the execute command functionality on all containers in the task.

Type: Boolean

Required: No

group

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

Type: String

Required: No

networkConfiguration

The VPC subnet and security group configuration for tasks that receive their own elastic network interface by using the `awsvpc` networking mode.

Type: NetworkConfiguration object

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 receives. You can override the default command for a container (that's 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 object

Required: No

propagateTags

Specifies whether to propagate the tags from the task definition or the service to the task. If no value is specified, the tags aren't propagated.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

referenceId

The reference ID to use for the task.

Type: String

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 call with the `startedBy` value. Up to 36 letters (uppercase and lowercase), numbers, hyphens (-), and underscores (_) are allowed.

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

Type: String

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

taskDefinition

The `family` and `revision` (`family:revision`) or full ARN of the task definition to start. If a `revision` isn't specified, the latest `ACTIVE` revision is used.

Type: String

Required: Yes

volumeConfigurations

The details of the volume that was `configuredAtLaunch`. You can configure the size, volumeType, IOPS, throughput, snapshot and encryption in TaskManagedEBSVolumeConfiguration. The `name` of the volume must match the `name` from the task definition.

Type: Array of TaskVolumeConfiguration objects

Required: No

Response Syntax

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

```
            }
         ],
         "attributes": [
            {
               "name": "string",
               "targetId": "string",
               "targetType": "string",
               "value": "string"
            }
         ],
         "availabilityZone": "string",
         "capacityProviderName": "string",
         "clusterArn": "string",
         "connectivity": "string",
         "connectivityAt": number,
         "containerInstanceArn": "string",
         "containers": [
            {
               "containerArn": "string",
               "cpu": "string",
               "exitCode": number,
               "gpuIds": [ "string" ],
               "healthStatus": "string",
               "image": "string",
               "imageDigest": "string",
               "lastStatus": "string",
               "managedAgents": [
                  {
                     "lastStartedAt": number,
                     "lastStatus": "string",
                     "name": "string",
                     "reason": "string"
                  }
               ],
               "memory": "string",
               "memoryReservation": "string",
               "name": "string",
               "networkBindings": [
                  {
                     "bindIP": "string",
                     "containerPort": number,
                     "containerPortRange": "string",
                     "hostPort": number,
                     "hostPortRange": "string",
```

```
                "protocol": "string"
            }
         ],
         "networkInterfaces": [
            {
               "attachmentId": "string",
               "ipv6Address": "string",
               "privateIpv4Address": "string"
            }
         ],
         "reason": "string",
         "runtimeId": "string",
         "taskArn": "string"
      }
   ],
   "cpu": "string",
   "createdAt": number,
   "desiredStatus": "string",
   "enableExecuteCommand": boolean,
   "ephemeralStorage": {
      "sizeInGiB": number
   },
   "executionStoppedAt": number,
   "group": "string",
   "healthStatus": "string",
   "inferenceAccelerators": [
      {
         "deviceName": "string",
         "deviceType": "string"
      }
   ],
   "lastStatus": "string",
   "launchType": "string",
   "memory": "string",
   "overrides": {
      "containerOverrides": [
         {
            "command": [ "string" ],
            "cpu": number,
            "environment": [
               {
                  "name": "string",
                  "value": "string"
               }
```

```
                ],
                "environmentFiles": [
                   {
                      "type": "string",
                      "value": "string"
                   }
                ],
                "memory": number,
                "memoryReservation": number,
                "name": "string",
                "resourceRequirements": [
                   {
                      "type": "string",
                      "value": "string"
                   }
                ]
             }
          ],
          "cpu": "string",
          "ephemeralStorage": {
             "sizeInGiB": number
          },
          "executionRoleArn": "string",
          "inferenceAcceleratorOverrides": [
             {
                "deviceName": "string",
                "deviceType": "string"
             }
          ],
          "memory": "string",
          "taskRoleArn": "string"
       },
       "platformFamily": "string",
       "platformVersion": "string",
       "pullStartedAt": number,
       "pullStoppedAt": number,
       "startedAt": number,
       "startedBy": "string",
       "stopCode": "string",
       "stoppedAt": number,
       "stoppedReason": "string",
       "stoppingAt": number,
       "tags": [
          {
```

```
                "key": "string",
                "value": "string"
            }
         ],
         "taskArn": "string",
         "taskDefinitionArn": "string",
         "version": number
      }
   ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

tasks

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

Type: Array of Task objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

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

{
  "failures": [],
  "tasks": [
    {
      "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
      "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-instance/
default/4c543eed-f83f-47da-b1d8-3d23f1da4c64",
      "containers": [
        {
          "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/
e76594d4-27e1-4c74-98b5-46a6435eb769",
          "lastStatus": "PENDING",
          "name": "wordpress",
```

```
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/default/
fdf2c302-468c-4e55-b884-5331d816e7fb"
        },
        {
          "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/default/
b19106ea-4fa8-4f1d-9767-96922c82b070",
          "lastStatus": "PENDING",
          "name": "mysql",
          "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/default/
fdf2c302-468c-4e55-b884-5331d816e7fb"
        }
      ],
      "createdAt": 1479765460.842,
      "desiredStatus": "RUNNING",
      "lastStatus": "PENDING",
      "overrides": {
        "containerOverrides": [
          {
            "name": "wordpress"
          },
          {
            "name": "mysql"
          }
        ]
      },
      "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/default/fdf2c302-468c-4e55-
b884-5331d816e7fb",
      "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:6",
      "version": 1
    }
  ]
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StopTask

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

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

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

reason

An optional message specified when a task is stopped. For example, if you're using a custom scheduler, you can use this parameter to specify the reason for stopping the task here, and the message appears in subsequent DescribeTasks API operations on this task.

Type: String

Required: No

task

The task ID of the task to stop.

Type: String

Required: Yes

Response Syntax

```
{
   "task": {
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attributes": [
         {
            "name": "string",
            "targetId": "string",
            "targetType": "string",
            "value": "string"
         }
      ],
```

```
      "availabilityZone": "string",
      "capacityProviderName": "string",
      "clusterArn": "string",
      "connectivity": "string",
      "connectivityAt": number,
      "containerInstanceArn": "string",
      "containers": [
         {
            "containerArn": "string",
            "cpu": "string",
            "exitCode": number,
            "gpuIds": [ "string" ],
            "healthStatus": "string",
            "image": "string",
            "imageDigest": "string",
            "lastStatus": "string",
            "managedAgents": [
               {
                  "lastStartedAt": number,
                  "lastStatus": "string",
                  "name": "string",
                  "reason": "string"
               }
            ],
            "memory": "string",
            "memoryReservation": "string",
            "name": "string",
            "networkBindings": [
               {
                  "bindIP": "string",
                  "containerPort": number,
                  "containerPortRange": "string",
                  "hostPort": number,
                  "hostPortRange": "string",
                  "protocol": "string"
               }
            ],
            "networkInterfaces": [
               {
                  "attachmentId": "string",
                  "ipv6Address": "string",
                  "privateIpv4Address": "string"
               }
            ],
```

```
            "reason": "string",
            "runtimeId": "string",
            "taskArn": "string"
         }
      ],
      "cpu": "string",
      "createdAt": number,
      "desiredStatus": "string",
      "enableExecuteCommand": boolean,
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionStoppedAt": number,
      "group": "string",
      "healthStatus": "string",
      "inferenceAccelerators": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "lastStatus": "string",
      "launchType": "string",
      "memory": "string",
      "overrides": {
         "containerOverrides": [
            {
               "command": [ "string" ],
               "cpu": number,
               "environment": [
                  {
                     "name": "string",
                     "value": "string"
                  }
               ],
               "environmentFiles": [
                  {
                     "type": "string",
                     "value": "string"
                  }
               ],
               "memory": number,
               "memoryReservation": number,
               "name": "string",
```

```
            "resourceRequirements": [
               {
                  "type": "string",
                  "value": "string"
               }
            ]
         }
      ],
      "cpu": "string",
      "ephemeralStorage": {
         "sizeInGiB": number
      },
      "executionRoleArn": "string",
      "inferenceAcceleratorOverrides": [
         {
            "deviceName": "string",
            "deviceType": "string"
         }
      ],
      "memory": "string",
      "taskRoleArn": "string"
   },
   "platformFamily": "string",
   "platformVersion": "string",
   "pullStartedAt": number,
   "pullStoppedAt": number,
   "startedAt": number,
   "startedBy": "string",
   "stopCode": "string",
   "stoppedAt": number,
   "stoppedReason": "string",
   "stoppingAt": number,
   "tags": [
      {
         "key": "string",
         "value": "string"
      }
   ],
   "taskArn": "string",
   "taskDefinitionArn": "string",
   "version": number
  }
}
```

Response Elements

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

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

task

The task that was stopped.

Type: Task object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example request stops a task with the ID 1dc5c17a-422b-4dc4-b493-371970c6c4d6 in the default cluster.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 88
X-Amz-Target: AmazonEC2ContainerServiceV20141113.StopTask
X-Amz-Date: 20161121T220318Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "task": "1dc5c17a-422b-4dc4-b493-371970c6c4d6"
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 21 Nov 2016 22:03:18 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1260
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
```

```
{
  "task": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "containerInstanceArn": "arn:aws:ecs:us-east-1:012345678910:container-instance/
default/5991d8da-1d59-49d2-a31f-4230f9e73140",
    "containers": [
      {
        "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/4df26bb4-
f057-467b-a079-961675296e64",
        "lastStatus": "RUNNING",
        "name": "simple-app",
        "networkBindings": [
          {
            "bindIP": "0.0.0.0",
            "containerPort": 80,
            "hostPort": 32774,
            "protocol": "tcp"
          }
        ],
        "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/default/1dc5c17a-422b-4dc4-
b493-371970c6c4d6"
      },
      {
        "containerArn": "arn:aws:ecs:us-east-1:012345678910:container/
e09064f7-7361-4c87-8ab9-8d073bbdbcb9",
        "lastStatus": "RUNNING",
        "name": "busybox",
        "networkBindings": [],
        "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/default/1dc5c17a-422b-4dc4-
b493-371970c6c4d6"
      }
    ],
    "createdAt": 1476822811.295,
    "desiredStatus": "STOPPED",
    "lastStatus": "RUNNING",
    "overrides": {
      "containerOverrides": [
        {
          "name": "simple-app"
        },
        {
          "name": "busybox"
        }
      ]
```

```
    },
    "startedAt": 1476822833.998,
    "startedBy": "ecs-svc/9223370560032507596",
    "stoppedReason": "Task stopped by user",
    "taskArn": "arn:aws:ecs:us-east-1:012345678910:task/1dc5c17a-422b-4dc4-
b493-371970c6c4d6",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/default/
console-sample-app-dynamic-ports:1",
    "version": 0
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SubmitAttachmentStateChanges

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

Sent to acknowledge that an attachment changed states.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

attachments

Any attachments associated with the state change request.

Type: Array of AttachmentStateChange objects

Required: Yes

cluster

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

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

acknowledgment

Acknowledgement of the state change.

Type: String

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SubmitContainerStateChange

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

Sent to acknowledge that a container changed states.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The short name or full 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 that's returned for the state change request.

Type: Integer

Required: No

networkBindings

The network bindings of the container.

Type: Array of NetworkBinding objects

Required: No

reason

The reason for the state change request.

Type: String

Required: No

runtimeId

The ID of the Docker container.

Type: String

Required: No

status

The status of the state change request.

Type: String

Required: No

task

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

Type: String

Required: No

Response Syntax

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

Response Elements

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

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

acknowledgment

Acknowledgement of the state change.

Type: String

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SubmitTaskStateChange

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

Sent to acknowledge that a task changed states.

Request Syntax

```
{
   "attachments": [
      {
         "attachmentArn": "string",
         "status": "string"
      }
   ],
   "cluster": "string",
   "containers": [
      {
         "containerName": "string",
         "exitCode": number,
         "imageDigest": "string",
         "networkBindings": [
            {
               "bindIP": "string",
               "containerPort": number,
               "containerPortRange": "string",
               "hostPort": number,
               "hostPortRange": "string",
               "protocol": "string"
            }
         ],
         "reason": "string",
         "runtimeId": "string",
         "status": "string"
      }
   ],
   "executionStoppedAt": number,
   "managedAgents": [
```

```
      {
         "containerName": "string",
         "managedAgentName": "string",
         "reason": "string",
         "status": "string"
      }
   ],
   "pullStartedAt": number,
   "pullStoppedAt": number,
   "reason": "string",
   "status": "string",
   "task": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

attachments

Any attachments associated with the state change request.

Type: Array of AttachmentStateChange objects

Required: No

cluster

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

Type: String

Required: No

containers

Any containers that's associated with the state change request.

Type: Array of ContainerStateChange objects

Required: No

executionStoppedAt

The Unix timestamp for the time when the task execution stopped.

Type: Timestamp

Required: No

managedAgents

The details for the managed agent that's associated with the task.

Type: Array of ManagedAgentStateChange objects

Required: No

pullStartedAt

The Unix timestamp for the time when the container image pull started.

Type: Timestamp

Required: No

pullStoppedAt

The Unix timestamp for the time when the container image pull completed.

Type: Timestamp

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 ID or full 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.

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

resourceArn

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

Type: String

Required: Yes

tags

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

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50

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

Type: Array of Tag objects

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

Required: Yes

Response Elements

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

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example tags the `dev` cluster with key `team` and value `dev`.

Sample Request

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

```
{
   "resourceArn":"arn:aws:ecs:us-west-2:012345678910:cluster/dev",
   "tags":[
      {
         "key":"team",
         "value":"dev"
      }
   ]
}
```

Sample Response

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

{}
```

Example

This example tags the dev cluster with key `team` and value dev and the key `second-key` and value dev-key2.

Sample Request

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

{
   "resourceArn":"arn:aws:ecs:us-west-2:012345678910:cluster/dev",
   "tags":[
      {
         "key":"team",
```

```
         "value":"dev",
         "second-key",
         "dev-key2"
      }
   ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 2
Date: Wed, 19 Oct 2022 20:01:34 GMT

{}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Deletes specified tags from a resource.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

resourceArn

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

Type: String

Required: Yes

tagKeys

The keys of the tags to be removed.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 128.

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

Required: Yes

Response Elements

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

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

This example tags the `dev` cluster with key `team` and value `dev`.

Sample Request

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

{
   "resourceArn":"arn:aws:ecs:us-west-2:012345678910:cluster/devcluster",
   "tagKeys":[
      "team"
   ]
}
```

Sample Response

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

{}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateCapacityProvider

Modifies the parameters for a capacity provider.

Request Syntax

```
{
   "autoScalingGroupProvider": {
      "managedDraining": "string",
      "managedScaling": {
         "instanceWarmupPeriod": number,
         "maximumScalingStepSize": number,
         "minimumScalingStepSize": number,
         "status": "string",
         "targetCapacity": number
      },
      "managedTerminationProtection": "string"
   },
   "name": "string"
}
```

Request Parameters

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

The request accepts the following data in JSON format.

autoScalingGroupProvider

An object that represent the parameters to update for the Auto Scaling group capacity provider.

Type: AutoScalingGroupProviderUpdate object

Required: Yes

name

The name of the capacity provider to update.

Type: String

Required: Yes

Response Syntax

```
{
   "capacityProvider": {
      "autoScalingGroupProvider": {
         "autoScalingGroupArn": "string",
         "managedDraining": "string",
         "managedScaling": {
            "instanceWarmupPeriod": number,
            "maximumScalingStepSize": number,
            "minimumScalingStepSize": number,
            "status": "string",
            "targetCapacity": number
         },
         "managedTerminationProtection": "string"
      },
      "capacityProviderArn": "string",
      "name": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "updateStatus": "string",
      "updateStatusReason": "string"
   }
}
```

Response Elements

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

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

capacityProvider

Details about the capacity provider.

Type: CapacityProvider object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateCluster

Updates the cluster.

Request Syntax

```
{
   "cluster": "string",
   "configuration": {
      "executeCommandConfiguration": {
         "kmsKeyId": "string",
         "logConfiguration": {
            "cloudWatchEncryptionEnabled": boolean,
            "cloudWatchLogGroupName": "string",
            "s3BucketName": "string",
            "s3EncryptionEnabled": boolean,
            "s3KeyPrefix": "string"
         },
         "logging": "string"
      }
   },
   "serviceConnectDefaults": {
      "namespace": "string"
   },
   "settings": [
      {
         "name": "string",
         "value": "string"
      }
   ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The name of the cluster to modify the settings for.

Type: String

Required: Yes

configuration

The execute command configuration for the cluster.

Type: ClusterConfiguration object

Required: No

serviceConnectDefaults

Use this parameter to set a default Service Connect namespace. After you set a default Service Connect namespace, any new services with Service Connect turned on that are created in the cluster are added as client services in the namespace. This setting only applies to new services that set the `enabled` parameter to `true` in the `ServiceConnectConfiguration`. You can set the namespace of each service individually in the `ServiceConnectConfiguration` to override this default parameter.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ClusterServiceConnectDefaultsRequest object

Required: No

settings

The cluster settings for your cluster.

Type: Array of ClusterSetting objects

Required: No

Response Syntax

```
{
   "cluster": {
      "activeServicesCount": number,
      "attachments": [
         {
```

```
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attachmentsStatus": "string",
      "capacityProviders": [ "string" ],
      "clusterArn": "string",
      "clusterName": "string",
      "configuration": {
         "executeCommandConfiguration": {
            "kmsKeyId": "string",
            "logConfiguration": {
               "cloudWatchEncryptionEnabled": boolean,
               "cloudWatchLogGroupName": "string",
               "s3BucketName": "string",
               "s3EncryptionEnabled": boolean,
               "s3KeyPrefix": "string"
            },
            "logging": "string"
         }
      },
      "defaultCapacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "pendingTasksCount": number,
      "registeredContainerInstancesCount": number,
      "runningTasksCount": number,
      "serviceConnectDefaults": {
         "namespace": "string"
      },
      "settings": [
         {
            "name": "string",
```

```
            "value": "string"
         }
      ],
      "statistics": [
         {
            "name": "string",
            "value": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ]
   }
}
```

Response Elements

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

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

cluster

Details about the cluster.

Type: Cluster object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateClusterSettings

Modifies the settings to use for a cluster.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

The name of the cluster to modify the settings for.

Type: String

Required: Yes

settings

The setting to use by default for a cluster. This parameter is used to turn on CloudWatch Container Insights for a cluster. If this value is specified, it overrides the `containerInsights` value set with PutAccountSetting or PutAccountSettingDefault.

> **Important**
>
> Currently, if you delete an existing cluster that does not have Container Insights turned on, and then create a new cluster with the same name with Container Insights tuned

on, Container Insights will not actually be turned on. If you want to preserve the same name for your existing cluster and turn on Container Insights, you must wait 7 days before you can re-create it.

Type: Array of ClusterSetting objects

Required: Yes

Response Syntax

```
{
   "cluster": {
      "activeServicesCount": number,
      "attachments": [
         {
            "details": [
               {
                  "name": "string",
                  "value": "string"
               }
            ],
            "id": "string",
            "status": "string",
            "type": "string"
         }
      ],
      "attachmentsStatus": "string",
      "capacityProviders": [ "string" ],
      "clusterArn": "string",
      "clusterName": "string",
      "configuration": {
         "executeCommandConfiguration": {
            "kmsKeyId": "string",
            "logConfiguration": {
               "cloudWatchEncryptionEnabled": boolean,
               "cloudWatchLogGroupName": "string",
               "s3BucketName": "string",
               "s3EncryptionEnabled": boolean,
               "s3KeyPrefix": "string"
            },
            "logging": "string"
```

```
            }
         },
         "defaultCapacityProviderStrategy": [
            {
               "base": number,
               "capacityProvider": "string",
               "weight": number
            }
         ],
         "pendingTasksCount": number,
         "registeredContainerInstancesCount": number,
         "runningTasksCount": number,
         "serviceConnectDefaults": {
            "namespace": "string"
         },
         "settings": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "statistics": [
            {
               "name": "string",
               "value": "string"
            }
         ],
         "status": "string",
         "tags": [
            {
               "key": "string",
               "value": "string"
            }
         ]
      }
}
```

Response Elements

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

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

cluster

Details about the cluster

Type: Cluster object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

See Also

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

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateContainerAgent

Updates the Amazon ECS container agent on a specified container instance. Updating the Amazon ECS container agent doesn't 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.

> **Note**
>
> The `UpdateContainerAgent` API isn't supported for container instances using the Amazon ECS-optimized Amazon Linux 2 (arm64) AMI. To update the container agent, you can update the `ecs-init` package. This updates the agent. For more information, see Updating the Amazon ECS container agent in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> Agent updates with the `UpdateContainerAgent` API operation do not apply to Windows container instances. We recommend that you launch new container instances to update the agent version in your Windows clusters.

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

Response Syntax

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

```
            "targetType": "string",
            "value": "string"
         }
      ],
      "capacityProviderName": "string",
      "containerInstanceArn": "string",
      "ec2InstanceId": "string",
      "healthStatus": {
         "details": [
            {
               "lastStatusChange": number,
               "lastUpdated": number,
               "status": "string",
               "type": "string"
            }
         ],
         "overallStatus": "string"
      },
      "pendingTasksCount": number,
      "registeredAt": number,
      "registeredResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "remainingResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "runningTasksCount": number,
      "status": "string",
      "statusReason": "string",
      "tags": [
```

```
         {
            "key": "string",
            "value": "string"
         }
      ],
      "version": number,
      "versionInfo": {
         "agentHash": "string",
         "agentVersion": "string",
         "dockerVersion": "string"
      }
   }
}
```

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

The container instance that the container agent was updated for.

Type: ContainerInstance object

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

MissingVersionException

Amazon ECS can't determine the current version of the Amazon ECS container agent on the container instance and doesn't have enough information to proceed with an update. This could be because the agent running on the container instance is a previous or custom version that doesn't use our version information.

HTTP Status Code: 400

NoUpdateAvailableException

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

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UpdateInProgressException

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

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

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

{
  "containerInstance": {
    "agentConnected": true,
    "agentUpdateStatus": "PENDING",
...
```

```
    "versionInfo": {
      "agentHash": "4023248",
      "agentVersion": "1.0.0",
      "dockerVersion": "DockerVersion: 1.5.0"
    }
  }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateContainerInstancesState

Modifies the status of an Amazon ECS container instance.

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

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

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

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

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

Any PENDING or RUNNING tasks that do not belong to a service aren't affected. You must wait for them to finish or stop them manually.

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

When a container instance has been drained, you can set a container instance to ACTIVE status and once it has reached that status the Amazon ECS scheduler can begin scheduling tasks on the instance again.

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: No

containerInstances

A list of up to 10 container instance IDs or full ARN entries.

Type: Array of strings

Required: Yes

status

The container instance state to update the container instance with. The only valid values for this action are ACTIVE and DRAINING. A container instance can only be updated to DRAINING status once it has reached an ACTIVE state. If a container instance is in REGISTERING, DEREGISTERING, or REGISTRATION_FAILED state you can describe the container instance but can't update the container instance state.

Type: String

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

Required: Yes

Response Syntax

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

```
      ],
      "capacityProviderName": "string",
      "containerInstanceArn": "string",
      "ec2InstanceId": "string",
      "healthStatus": {
         "details": [
            {
               "lastStatusChange": number,
               "lastUpdated": number,
               "status": "string",
               "type": "string"
            }
         ],
         "overallStatus": "string"
      },
      "pendingTasksCount": number,
      "registeredAt": number,
      "registeredResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "remainingResources": [
         {
            "doubleValue": number,
            "integerValue": number,
            "longValue": number,
            "name": "string",
            "stringSetValue": [ "string" ],
            "type": "string"
         }
      ],
      "runningTasksCount": number,
      "status": "string",
      "statusReason": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
```

```
            }
         ],
         "version": number,
         "versionInfo": {
            "agentHash": "string",
            "agentVersion": "string",
            "dockerVersion": "string"
         }
      }
   ],
   "failures": [
      {
         "arn": "string",
         "detail": "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 objects

failures

Any failures associated with the call.

Type: Array of Failure objects

Errors

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

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

Examples

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

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

Example

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

Sample Request

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

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 20 Dec 2016 22:11:42 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 2344
Connection: keep-alive
x-amzn-RequestId: 49d68928-c701-11e6-8f99-6103d648cdad

{
  "containerInstances": [
    {
      "agentConnected": true,
      "attributes": [
        {
          "name": "ecs.availability-zone",
          "value": "us-west-2b"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
        },
        {
```

```
          "name": "ecs.instance-type",
          "value": "c4.xlarge"
        },
        {
          "name": "ecs.ami-id",
          "value": "ami-a2ca61c2"
        },
        {
          "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
        },
        {
          "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
        },
        {
          "name": "com.amazonaws.ecs.capability.privileged-container"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
        },
        {
          "name": "com.amazonaws.ecs.capability.ecr-auth"
        },
        {
          "name": "ecs.os-type",
          "value": "linux"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
        },
```

```
        {
          "name": "com.amazonaws.ecs.capability.task-iam-role"
        },
        {
          "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
        }
      ],
      "containerInstanceArn": "arn:aws:ecs:us-west-2:012345678910:container-instance/
default/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
      "ec2InstanceId": "i-05d99c76955727ec6",
      "pendingTasksCount": 0,
      "registeredResources": [
        {
          "doubleValue": 0,
          "integerValue": 4096,
          "longValue": 0,
          "name": "CPU",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 7482,
          "longValue": 0,
          "name": "MEMORY",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
          "name": "PORTS",
          "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
          ],
          "type": "STRINGSET"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
```

```
          "name": "PORTS_UDP",
          "stringSetValue": [],
          "type": "STRINGSET"
        }
      ],
      "remainingResources": [
        {
          "doubleValue": 0,
          "integerValue": 4096,
          "longValue": 0,
          "name": "CPU",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 7482,
          "longValue": 0,
          "name": "MEMORY",
          "type": "INTEGER"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
          "name": "PORTS",
          "stringSetValue": [
            "22",
            "2376",
            "2375",
            "51678",
            "51679"
          ],
          "type": "STRINGSET"
        },
        {
          "doubleValue": 0,
          "integerValue": 0,
          "longValue": 0,
          "name": "PORTS_UDP",
          "stringSetValue": [],
          "type": "STRINGSET"
        }
      ],
      "runningTasksCount": 0,
```

```
      "status": "DRAINING",
      "version": 30,
      "versionInfo": {
        "agentHash": "efe53c6",
        "agentVersion": "1.13.1",
        "dockerVersion": "DockerVersion: 1.11.2"
      }
    }
  ],
  "failures": []
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateService

Modifies the parameters of a service.

> **Note**
>
> On March 21, 2024, a change was made to resolve the task definition revision before authorization. When a task definition revision is not specified, authorization will occur using the latest revision of a task definition.

For services using the rolling update (ECS) you can update the desired count, deployment configuration, network configuration, load balancers, service registries, enable ECS managed tags option, propagate tags option, task placement constraints and strategies, and task definition. When you update any of these parameters, Amazon ECS starts new tasks with the new configuration.

You can attach Amazon EBS volumes to Amazon ECS tasks by configuring the volume when starting or running a task, or when creating or updating a service. For more infomation, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*. You can update your volume configurations and trigger a new deployment. `volumeConfigurations` is only supported for REPLICA service and not DAEMON service. If you leave `volumeConfigurations` `null`, it doesn't trigger a new deployment. For more infomation on volumes, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*.

For services using the blue/green (`CODE_DEPLOY`) deployment controller, only the desired count, deployment configuration, health check grace period, task placement constraints and strategies, enable ECS managed tags option, and propagate tags can be updated using this API. If the network configuration, platform version, task definition, or load balancer need to be updated, create a new AWS CodeDeploy deployment. For more information, see CreateDeployment in the *AWS CodeDeploy API Reference*.

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

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 attach Amazon EBS volumes to Amazon ECS tasks by configuring the volume when starting or running a task, or when creating or updating a service. For more infomation, see Amazon EBS volumes in the *Amazon Elastic Container Service Developer Guide*.

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

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

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

- If `minimumHealthyPercent` is below 100%, the scheduler can ignore `desiredCount` temporarily during a deployment. For example, if `desiredCount` is four tasks, a minimum of 50% allows the scheduler to stop two existing tasks before starting two new tasks. Tasks for services that don't use a load balancer are considered healthy if they're in the `RUNNING` state. Tasks for services that use a load balancer are considered healthy if they're in the `RUNNING` state and are reported as healthy by the load balancer.
- The `maximumPercent` parameter represents an upper limit on the number of running tasks during a deployment. You can use it to define the deployment batch size. For example, if `desiredCount` is four tasks, a maximum of 200% starts four new tasks before stopping the four older tasks (provided that the cluster resources required to do this are available).

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

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

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

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

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

Note

You must have a service-linked role when you update any of the following service properties:

- loadBalancers,
- serviceRegistries

For more information about the role see the CreateService request parameter role.

Request Syntax

```
{
   "capacityProviderStrategy": [
      {
         "base": number,
         "capacityProvider": "string",
         "weight": number
      }
   ],
   "cluster": "string",
   "deploymentConfiguration": {
      "alarms": {
         "alarmNames": [ "string" ],
         "enable": boolean,
         "rollback": boolean
      },
      "deploymentCircuitBreaker": {
         "enable": boolean,
         "rollback": boolean
      },
      "maximumPercent": number,
      "minimumHealthyPercent": number
   },
   "desiredCount": number,
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "forceNewDeployment": boolean,
   "healthCheckGracePeriodSeconds": number,
   "loadBalancers": [
      {
         "containerName": "string",
         "containerPort": number,
         "loadBalancerName": "string",
         "targetGroupArn": "string"
```

```
      }
   ],
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
         "securityGroups": [ "string" ],
         "subnets": [ "string" ]
      }
   },
   "placementConstraints": [
      {
         "expression": "string",
         "type": "string"
      }
   ],
   "placementStrategy": [
      {
         "field": "string",
         "type": "string"
      }
   ],
   "platformVersion": "string",
   "propagateTags": "string",
   "service": "string",
   "serviceConnectConfiguration": {
      "enabled": boolean,
      "logConfiguration": {
         "logDriver": "string",
         "options": {
            "string" : "string"
         },
         "secretOptions": [
            {
               "name": "string",
               "valueFrom": "string"
            }
         ]
      },
      "namespace": "string",
      "services": [
         {
            "clientAliases": [
               {
                  "dnsName": "string",
```

```
                "port": number
             }
          ],
          "discoveryName": "string",
          "ingressPortOverride": number,
          "portName": "string",
          "timeout": {
             "idleTimeoutSeconds": number,
             "perRequestTimeoutSeconds": number
          },
          "tls": {
             "issuerCertificateAuthority": {
                "awsPcaAuthorityArn": "string"
             },
             "kmsKey": "string",
             "roleArn": "string"
          }
       }
    ]
 },
 "serviceRegistries": [
    {
       "containerName": "string",
       "containerPort": number,
       "port": number,
       "registryArn": "string"
    }
 ],
 "taskDefinition": "string",
 "volumeConfigurations": [
    {
       "managedEBSVolume": {
          "encrypted": boolean,
          "filesystemType": "string",
          "iops": number,
          "kmsKeyId": "string",
          "roleArn": "string",
          "sizeInGiB": number,
          "snapshotId": "string",
          "tagSpecifications": [
             {
                "propagateTags": "string",
                "resourceType": "string",
                "tags": [
```

```
                    {
                       "key": "string",
                       "value": "string"
                    }
                 ]
              }
           ],
           "throughput": number,
           "volumeType": "string"
        },
        "name": "string"
     }
  ]
}
```

Request Parameters

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

The request accepts the following data in JSON format.

capacityProviderStrategy

The capacity provider strategy to update the service to use.

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

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

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

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

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

Type: Array of CapacityProviderStrategyItem objects

Required: No

cluster

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

Type: String

Required: No

deploymentConfiguration

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

Type: DeploymentConfiguration object

Required: No

desiredCount

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

Type: Integer

Required: No

enableECSManagedTags

Determines whether to turn on Amazon ECS managed tags for the tasks in the service. For more information, see Tagging Your Amazon ECS Resources in the *Amazon Elastic Container Service Developer Guide*.

Only tasks launched after the update will reflect the update. To update the tags on all tasks, set `forceNewDeployment` to `true`, so that Amazon ECS starts new tasks with the updated tags.

Type: Boolean

Required: No

enableExecuteCommand

If `true`, this enables execute command functionality on all task containers.

If you do not want to override the value that was set when the service was created, you can set this to `null` when performing this action.

Type: Boolean

Required: No

forceNewDeployment

Determines whether to force a new deployment of the service. By default, deployments aren't forced. You can use this option to start a new deployment with no service definition changes. For example, you can update a service's tasks to use a newer Docker image with the same image/tag combination (`my_image:latest`) or to roll Fargate tasks onto a newer platform version.

Type: Boolean

Required: No

healthCheckGracePeriodSeconds

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

Type: Integer

Required: No

loadBalancers

A list of Elastic Load Balancing load balancer objects. It contains the load balancer name, the container name, and the container port to access from the load balancer. The container name is as it appears in a container definition.

When you add, update, or remove a load balancer configuration, Amazon ECS starts new tasks with the updated Elastic Load Balancing configuration, and then stops the old tasks when the new tasks are running.

For services that use rolling updates, you can add, update, or remove Elastic Load Balancing target groups. You can update from a single target group to multiple target groups and from multiple target groups to a single target group.

For services that use blue/green deployments, you can update Elastic Load Balancing target groups by using `CreateDeployment` through AWS CodeDeploy. Note that multiple target groups are not supported for blue/green deployments. For more information see Register multiple target groups with a service in the *Amazon Elastic Container Service Developer Guide*.

For services that use the external deployment controller, you can add, update, or remove load balancers by using CreateTaskSet. Note that multiple target groups are not supported for external deployments. For more information see Register multiple target groups with a service in the *Amazon Elastic Container Service Developer Guide*.

You can remove existing `loadBalancers` by passing an empty list.

Type: Array of LoadBalancer objects

Required: No

networkConfiguration

An object representing the network configuration for the service.

Type: NetworkConfiguration object

Required: No

placementConstraints

An array of task placement constraint objects to update the service to use. If no value is specified, the existing placement constraints for the service will remain unchanged. If this

value is specified, it will override any existing placement constraints defined for the service. To remove all existing placement constraints, specify an empty array.

You can specify a maximum of 10 constraints for each task. This limit includes constraints in the task definition and those specified at runtime.

Type: Array of PlacementConstraint objects

Required: No

placementStrategy

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

You can specify a maximum of five strategy rules for each service.

Type: Array of PlacementStrategy objects

Required: No

platformVersion

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

Type: String

Required: No

propagateTags

Determines whether to propagate the tags from the task definition or the service to the task. If no value is specified, the tags aren't propagated.

Only tasks launched after the update will reflect the update. To update the tags on all tasks, set `forceNewDeployment` to `true`, so that Amazon ECS starts new tasks with the updated tags.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

service

The name of the service to update.

Type: String

Required: Yes

serviceConnectConfiguration

The configuration for this service to discover and connect to services, and be discovered by, and connected from, other services within a namespace.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ServiceConnectConfiguration object

Required: No

serviceRegistries

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

When you add, update, or remove the service registries configuration, Amazon ECS starts new tasks with the updated service registries configuration, and then stops the old tasks when the new tasks are running.

You can remove existing `serviceRegistries` by passing an empty list.

Type: Array of ServiceRegistry objects

Required: No

taskDefinition

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

Type: String

Required: No

volumeConfigurations

The details of the volume that was `configuredAtLaunch`. You can configure the size, volumeType, IOPS, throughput, snapshot and encryption in ServiceManagedEBSVolumeConfiguration. The `name` of the volume must match the `name` from the task definition. If set to null, no new deployment is triggered. Otherwise, if this configuration differs from the existing one, it triggers a new deployment.

Type: Array of ServiceVolumeConfiguration objects

Required: No

Response Syntax

```
{
   "service": {
      "capacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "clusterArn": "string",
      "createdAt": number,
      "createdBy": "string",
      "deploymentConfiguration": {
         "alarms": {
            "alarmNames": [ "string" ],
            "enable": boolean,
            "rollback": boolean
         },
         "deploymentCircuitBreaker": {
            "enable": boolean,
            "rollback": boolean
         },
         "maximumPercent": number,
         "minimumHealthyPercent": number
      },
```

```
      "deploymentController": {
         "type": "string"
      },
      "deployments": [
         {
            "capacityProviderStrategy": [
               {
                  "base": number,
                  "capacityProvider": "string",
                  "weight": number
               }
            ],
            "createdAt": number,
            "desiredCount": number,
            "failedTasks": number,
            "id": "string",
            "launchType": "string",
            "networkConfiguration": {
               "awsvpcConfiguration": {
                  "assignPublicIp": "string",
                  "securityGroups": [ "string" ],
                  "subnets": [ "string" ]
               }
            },
            "pendingCount": number,
            "platformFamily": "string",
            "platformVersion": "string",
            "rolloutState": "string",
            "rolloutStateReason": "string",
            "runningCount": number,
            "serviceConnectConfiguration": {
               "enabled": boolean,
               "logConfiguration": {
                  "logDriver": "string",
                  "options": {
                     "string" : "string"
                  },
                  "secretOptions": [
                     {
                        "name": "string",
                        "valueFrom": "string"
                     }
                  ]
               },
```

```
            "namespace": "string",
            "services": [
               {
                  "clientAliases": [
                     {
                        "dnsName": "string",
                        "port": number
                     }
                  ],
                  "discoveryName": "string",
                  "ingressPortOverride": number,
                  "portName": "string",
                  "timeout": {
                     "idleTimeoutSeconds": number,
                     "perRequestTimeoutSeconds": number
                  },
                  "tls": {
                     "issuerCertificateAuthority": {
                        "awsPcaAuthorityArn": "string"
                     },
                     "kmsKey": "string",
                     "roleArn": "string"
                  }
               }
            ]
         },
         "serviceConnectResources": [
            {
               "discoveryArn": "string",
               "discoveryName": "string"
            }
         ],
         "status": "string",
         "taskDefinition": "string",
         "updatedAt": number,
         "volumeConfigurations": [
            {
               "managedEBSVolume": {
                  "encrypted": boolean,
                  "filesystemType": "string",
                  "iops": number,
                  "kmsKeyId": "string",
                  "roleArn": "string",
                  "sizeInGiB": number,
```

```
                  "snapshotId": "string",
                  "tagSpecifications": [
                     {
                        "propagateTags": "string",
                        "resourceType": "string",
                        "tags": [
                           {
                              "key": "string",
                              "value": "string"
                           }
                        ]
                     }
                  ],
                  "throughput": number,
                  "volumeType": "string"
               },
               "name": "string"
            }
         ]
      }
   ],
   "desiredCount": number,
   "enableECSManagedTags": boolean,
   "enableExecuteCommand": boolean,
   "events": [
      {
         "createdAt": number,
         "id": "string",
         "message": "string"
      }
   ],
   "healthCheckGracePeriodSeconds": number,
   "launchType": "string",
   "loadBalancers": [
      {
         "containerName": "string",
         "containerPort": number,
         "loadBalancerName": "string",
         "targetGroupArn": "string"
      }
   ],
   "networkConfiguration": {
      "awsvpcConfiguration": {
         "assignPublicIp": "string",
```

```
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      },
      "pendingCount": number,
      "placementConstraints": [
         {
            "expression": "string",
            "type": "string"
         }
      ],
      "placementStrategy": [
         {
            "field": "string",
            "type": "string"
         }
      ],
      "platformFamily": "string",
      "platformVersion": "string",
      "propagateTags": "string",
      "roleArn": "string",
      "runningCount": number,
      "schedulingStrategy": "string",
      "serviceArn": "string",
      "serviceName": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSets": [
         {
            "capacityProviderStrategy": [
```

```
            {
               "base": number,
               "capacityProvider": "string",
               "weight": number
            }
         ],
         "clusterArn": "string",
         "computedDesiredCount": number,
         "createdAt": number,
         "externalId": "string",
         "id": "string",
         "launchType": "string",
         "loadBalancers": [
            {
               "containerName": "string",
               "containerPort": number,
               "loadBalancerName": "string",
               "targetGroupArn": "string"
            }
         ],
         "networkConfiguration": {
            "awsvpcConfiguration": {
               "assignPublicIp": "string",
               "securityGroups": [ "string" ],
               "subnets": [ "string" ]
            }
         },
         "pendingCount": number,
         "platformFamily": "string",
         "platformVersion": "string",
         "runningCount": number,
         "scale": {
            "unit": "string",
            "value": number
         },
         "serviceArn": "string",
         "serviceRegistries": [
            {
               "containerName": "string",
               "containerPort": number,
               "port": number,
               "registryArn": "string"
            }
         ],
```

```
            "stabilityStatus": "string",
            "stabilityStatusAt": number,
            "startedBy": "string",
            "status": "string",
            "tags": [
               {
                  "key": "string",
                  "value": "string"
               }
            ],
            "taskDefinition": "string",
            "taskSetArn": "string",
            "updatedAt": number
         }
      ]
   }
}
```

Response Elements

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

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

service

The full description of your service following the update call.

Type: Service object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

NamespaceNotFoundException

The specified namespace wasn't found.

HTTP Status Code: 400

PlatformTaskDefinitionIncompatibilityException

The specified platform version doesn't satisfy the required capabilities of the task definition.

HTTP Status Code: 400

PlatformUnknownException

The specified platform version doesn't exist.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example

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

Sample Request

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

{
  "service": "hello_world",
```

```
  "desiredCount": 3
}
```

Sample Response

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

{
  "service": {
    "clusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/default",
    "deploymentConfiguration": {
        "maximumPercent": 200,
        "minimumHealthyPercent": 100
    },
    "deployments": [
      {
        "createdAt": 1430333711.033,
        "desiredCount": 3,
        "id": "ecs-svc/9223370606521064774",
        "pendingCount": 0,
        "runningCount": 0,
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:10",
        "updatedAt": 1430336267.173
      }
    ],
    "desiredCount": 3,
    "events": [],
    "loadBalancers": [],
    "pendingCount": 0,
    "runningCount": 0,
    "serviceArn": "arn:aws:ecs:us-east-1:012345678910:service/default/hello_world",
    "serviceName": "hello_world",
    "status": "ACTIVE",
    "taskDefinition": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:10"
```

```
    }
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateServicePrimaryTaskSet

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

primaryTaskSet

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

Type: String

Required: Yes

service

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

Type: String

Required: Yes

Response Syntax

```
{
   "taskSet": {
      "capacityProviderStrategy": [
         {
            "base": number,
            "capacityProvider": "string",
            "weight": number
         }
      ],
      "clusterArn": "string",
      "computedDesiredCount": number,
      "createdAt": number,
      "externalId": "string",
      "id": "string",
      "launchType": "string",
      "loadBalancers": [
         {
            "containerName": "string",
            "containerPort": number,
            "loadBalancerName": "string",
            "targetGroupArn": "string"
         }
      ],
      "networkConfiguration": {
         "awsvpcConfiguration": {
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      },
      "pendingCount": number,
      "platformFamily": "string",
      "platformVersion": "string",
      "runningCount": number,
      "scale": {
         "unit": "string",
         "value": number
```

```
      },
      "serviceArn": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "stabilityStatus": "string",
      "stabilityStatusAt": number,
      "startedBy": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSetArn": "string",
      "updatedAt": number
   }
}
```

Response Elements

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

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

taskSet

The details about the task set.

Type: TaskSet object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

TaskSetNotFoundException

The specified task set wasn't found. You can view your available task sets with DescribeTaskSets. Task sets are specific to each cluster, service and Region.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTaskProtection

Updates the protection status of a task. You can set `protectionEnabled` to `true` to protect your task from termination during scale-in events from Service Autoscaling or deployments.

Task-protection, by default, expires after 2 hours at which point Amazon ECS clears the `protectionEnabled` property making the task eligible for termination by a subsequent scale-in event.

You can specify a custom expiration period for task protection from 1 minute to up to 2,880 minutes (48 hours). To specify the custom expiration period, set the `expiresInMinutes` property. The `expiresInMinutes` property is always reset when you invoke this operation for a task that already has `protectionEnabled` set to `true`. You can keep extending the protection expiration period of a task by invoking this operation repeatedly.

To learn more about Amazon ECS task protection, see Task scale-in protection in the Amazon Elastic Container Service Developer Guide .

Note

This operation is only supported for tasks belonging to an Amazon ECS service. Invoking this operation for a standalone task will result in an `TASK_NOT_VALID` failure. For more information, see API failure reasons.

Important

If you prefer to set task protection from within the container, we recommend using the Task scale-in protection endpoint.

Request Syntax

```
{
   "cluster": "string",
   "expiresInMinutes": number,
   "protectionEnabled": boolean,
   "tasks": [ "string" ]
```

```
}
```

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

expiresInMinutes

If you set `protectionEnabled` to `true`, you can specify the duration for task protection in minutes. You can specify a value from 1 minute to up to 2,880 minutes (48 hours). During this time, your task will not be terminated by scale-in events from Service Auto Scaling or deployments. After this time period lapses, `protectionEnabled` will be reset to `false`.

If you don't specify the time, then the task is automatically protected for 120 minutes (2 hours).

Type: Integer

Required: No

protectionEnabled

Specify `true` to mark a task for protection and `false` to unset protection, making it eligible for termination.

Type: Boolean

Required: Yes

tasks

A list of up to 10 task IDs or full ARN entries.

Type: Array of strings

Required: Yes

Response Syntax

```
{
   "failures": [
      {
         "arn": "string",
         "detail": "string",
         "reason": "string"
      }
   ],
   "protectedTasks": [
      {
         "expirationDate": number,
         "protectionEnabled": boolean,
         "taskArn": "string"
      }
   ]
}
```

Response Elements

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

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

failures

Any failures associated with the call.

Type: Array of Failure objects

protectedTasks

A list of tasks with the following information.

- `taskArn`: The task ARN.
- `protectionEnabled`: The protection status of the task. If scale-in protection is turned on for a task, the value is `true`. Otherwise, it is `false`.

- `expirationDate`: The epoch time when protection for the task will expire.

Type: Array of ProtectedTask objects

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

Examples

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

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

Example 1

This example request enables scale-in protection for a task for 60 minutes.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 132
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateTaskProtection
X-Amz-Date: 20221102T184632Z Content-Type:
application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "test-task-protection",
    "tasks": [
        "b8b1cf532d0e46ba8d44a40d1de16772"
    ],
    "protectionEnabled": true,
    "expiresInMinutes": 60
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Wed, 02 Nov 2022 17:56:32 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 180
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:012345678910:task/default/
b8b1cf532d0e46ba8d44a40d1de16772",
            "protectionEnabled": true,
            "expirationDate": 1667415392.553
        }
    ],
    "failures": []
}
```

Example 2

This example request enables scale-in protection for a task without specifying the `expiresInMinutes` parameter. By default, the task will be protected from scale-in events for 120 minutes.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 132
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateTaskProtection
X-Amz-Date: 20221103T212558Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "test-task-protection",
    "tasks": [
        "b8b1cf532d0e46ba8d44a40d1de16772"
```

```
    ],
    "protectionEnabled": true
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 03 Nov 2022 21:25:57 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 180
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:012345678910:task/default/
b8b1cf532d0e46ba8d44a40d1de16772",
            "protectionEnabled": true,
            "expirationDate": 1667517958.355
        }
    ],
    "failures": []
}
```

Example 3

This example request turns off scale-in protection for a task.

Sample Request

```
POST / HTTP/1.1
Host: ecs.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length:109
X-Amz-Target: AmazonEC2ContainerServiceV20141113.UpdateTaskProtection
X-Amz-Date: 20221103T213134Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
    "cluster": "test-task-protection",
```

```
    "tasks": [
        "b8b1cf532d0e46ba8d44a40d1de16772"
    ],
    "protectionEnabled":false
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Thu, 03 Nov 2022 21:31:34 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 180
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
    "protectedTasks": [
        {
            "taskArn": "arn:aws:ecs:us-west-2:012345678910:task/default/
b8b1cf532d0e46ba8d44a40d1de16772",
            "protectionEnabled": false
        }
    ],
    "failures": []
}
```

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTaskSet

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

Request Syntax

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

Request Parameters

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

The request accepts the following data in JSON format.

cluster

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

Type: String

Required: Yes

scale

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

Type: Scale object

Required: Yes

service

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

Type: String

Required: Yes

taskSet

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

Type: String

Required: Yes

Response Syntax

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

```
            "assignPublicIp": "string",
            "securityGroups": [ "string" ],
            "subnets": [ "string" ]
         }
      },
      "pendingCount": number,
      "platformFamily": "string",
      "platformVersion": "string",
      "runningCount": number,
      "scale": {
         "unit": "string",
         "value": number
      },
      "serviceArn": "string",
      "serviceRegistries": [
         {
            "containerName": "string",
            "containerPort": number,
            "port": number,
            "registryArn": "string"
         }
      ],
      "stabilityStatus": "string",
      "stabilityStatusAt": number,
      "startedBy": "string",
      "status": "string",
      "tags": [
         {
            "key": "string",
            "value": "string"
         }
      ],
      "taskDefinition": "string",
      "taskSetArn": "string",
      "updatedAt": number
   }
}
```

Response Elements

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

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

taskSet

Details about the task set.

Type: TaskSet object

Errors

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

AccessDeniedException

You don't have authorization to perform the requested action.

HTTP Status Code: 400

ClientException

These errors are usually caused by a client action. This client action might be using an action or resource on behalf of a user that doesn't have permissions to use the action or resource. Or, it might be specifying an identifier that isn't valid.

HTTP Status Code: 400

ClusterNotFoundException

The specified cluster wasn't found. You can view your available clusters with ListClusters. Amazon ECS clusters are Region specific.

HTTP Status Code: 400

InvalidParameterException

The specified parameter isn't valid. Review the available parameters for the API request.

HTTP Status Code: 400

ServerException

These errors are usually caused by a server issue.

HTTP Status Code: 500

ServiceNotActiveException

The specified service isn't active. You can't update a service that's inactive. If you have previously deleted a service, you can re-create it with CreateService.

HTTP Status Code: 400

ServiceNotFoundException

The specified service wasn't found. You can view your available services with ListServices. Amazon ECS services are cluster specific and Region specific.

HTTP Status Code: 400

TaskSetNotFoundException

The specified task set wasn't found. You can view your available task sets with DescribeTaskSets. Task sets are specific to each cluster, service and Region.

HTTP Status Code: 400

UnsupportedFeatureException

The specified task isn't supported in this Region.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

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

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

The following data types are supported:

- Attachment
- AttachmentStateChange
- Attribute
- AutoScalingGroupProvider
- AutoScalingGroupProviderUpdate
- AwsVpcConfiguration
- CapacityProvider
- CapacityProviderStrategyItem
- Cluster
- ClusterConfiguration
- ClusterServiceConnectDefaults
- ClusterServiceConnectDefaultsRequest
- ClusterSetting
- Container
- ContainerDefinition
- ContainerDependency
- ContainerInstance
- ContainerInstanceHealthStatus
- ContainerOverride
- ContainerStateChange

- Deployment
- DeploymentAlarms
- DeploymentCircuitBreaker
- DeploymentConfiguration
- DeploymentController
- Device
- DockerVolumeConfiguration
- EBSTagSpecification
- EFSAuthorizationConfig
- EFSVolumeConfiguration
- EnvironmentFile
- EphemeralStorage
- ExecuteCommandConfiguration
- ExecuteCommandLogConfiguration
- Failure
- FirelensConfiguration
- FSxWindowsFileServerAuthorizationConfig
- FSxWindowsFileServerVolumeConfiguration
- HealthCheck
- HostEntry
- HostVolumeProperties
- InferenceAccelerator
- InferenceAcceleratorOverride
- InstanceHealthCheckResult
- KernelCapabilities
- KeyValuePair
- LinuxParameters
- LoadBalancer
- LogConfiguration
- ManagedAgent

- ManagedAgentStateChange
- ManagedScaling
- MountPoint
- NetworkBinding
- NetworkConfiguration
- NetworkInterface
- PlacementConstraint
- PlacementStrategy
- PlatformDevice
- PortMapping
- ProtectedTask
- ProxyConfiguration
- RepositoryCredentials
- Resource
- ResourceRequirement
- RuntimePlatform
- Scale
- Secret
- Service
- ServiceConnectClientAlias
- ServiceConnectConfiguration
- ServiceConnectService
- ServiceConnectServiceResource
- ServiceConnectTlsCertificateAuthority
- ServiceConnectTlsConfiguration
- ServiceEvent
- ServiceManagedEBSVolumeConfiguration
- ServiceRegistry
- ServiceVolumeConfiguration
- Session

- Setting
- SystemControl
- Tag
- Task
- TaskDefinition
- TaskDefinitionPlacementConstraint
- TaskManagedEBSVolumeConfiguration
- TaskManagedEBSVolumeTerminationPolicy
- TaskOverride
- TaskSet
- TaskVolumeConfiguration
- TimeoutConfiguration
- Tmpfs
- Ulimit
- VersionInfo
- Volume
- VolumeFrom

Attachment

An object representing a container instance or task attachment.

Contents

details

Details of the attachment.

For elastic network interfaces, this includes the network interface ID, the MAC address, the subnet ID, and the private IPv4 address.

For Service Connect services, this includes `portName`, `clientAliases`, `discoveryName`, and `ingressPortOverride`.

For Elastic Block Storage, this includes `roleArn`, `deleteOnTermination`, `volumeName`, `volumeId`, and `statusReason` (only when the attachment fails to create or attach).

Type: Array of KeyValuePair objects

Required: No

id

The unique identifier for the attachment.

Type: String

Required: No

status

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

Type: String

Required: No

type

The type of the attachment, such as `ElasticNetworkInterface`, `Service Connect`, and `AmazonElasticBlockStorage`.

Type: String

Required: No

See Also

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

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

AttachmentStateChange

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

Contents

attachmentArn

The Amazon Resource Name (ARN) of the attachment.

Type: String

Required: Yes

status

The status of the attachment.

Type: String

Required: Yes

See Also

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

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

Attribute

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

Contents

name

The name of the attribute. The `name` must contain between 1 and 128 characters. The name may contain letters (uppercase and lowercase), numbers, hyphens (-), underscores (_), forward slashes (/), back slashes (\), or periods (.).

Type: String

Required: Yes

targetId

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

Type: String

Required: No

targetType

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

Type: String

Valid Values: `container-instance`

Required: No

value

The value of the attribute. The `value` must contain between 1 and 128 characters. It can contain letters (uppercase and lowercase), numbers, hyphens (-), underscores (_), periods (.), at signs (@), forward slashes (/), back slashes (\), colons (:), or spaces. The value can't start or end with a space.

Type: String

Required: No

See Also

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

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

AutoScalingGroupProvider

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

Contents

autoScalingGroupArn

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

Type: String

Required: Yes

managedDraining

The managed draining option for the Auto Scaling group capacity provider. When you enable this, Amazon ECS manages and gracefully drains the EC2 container instances that are in the Auto Scaling group capacity provider.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

managedScaling

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

Type: ManagedScaling object

Required: No

managedTerminationProtection

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

> **Important**
>
> When using managed termination protection, managed scaling must also be used otherwise managed termination protection doesn't work.

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

When managed termination protection is off, your Amazon EC2 instances aren't protected from termination when the Auto Scaling group scales in.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

See Also

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

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

AutoScalingGroupProviderUpdate

The details of the Auto Scaling group capacity provider to update.

Contents

managedDraining

The managed draining option for the Auto Scaling group capacity provider. When you enable this, Amazon ECS manages and gracefully drains the EC2 container instances that are in the Auto Scaling group capacity provider.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

managedScaling

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

Type: ManagedScaling object

Required: No

managedTerminationProtection

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

> **Important**
>
> When using managed termination protection, managed scaling must also be used otherwise managed termination protection doesn't work.

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

When managed termination protection is off, your Amazon EC2 instances aren't protected from termination when the Auto Scaling group scales in.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

See Also

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

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

AwsVpcConfiguration

An object representing the networking details for a task or service. For example `awsvpcConfiguration={subnets=["subnet-12344321"],securityGroups=["sg-12344321"]}`

Contents

subnets

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

> **Note**
>
> All specified subnets must be from the same VPC.

Type: Array of strings

Required: Yes

assignPublicIp

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

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

securityGroups

The IDs of the security groups associated with the task or service. If you don't specify a security group, the default security group for the VPC is used. There's a limit of 5 security groups that can be specified per `AwsVpcConfiguration`.

> **Note**
>
> All specified security groups must be from the same VPC.

Type: Array of strings

Required: No

See Also

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

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

CapacityProvider

The details for a capacity provider.

Contents

autoScalingGroupProvider

The Auto Scaling group settings for the capacity provider.

Type: AutoScalingGroupProvider object

Required: No

capacityProviderArn

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

Type: String

Required: No

name

The name of the capacity provider.

Type: String

Required: No

status

The current status of the capacity provider. Only capacity providers in an `ACTIVE` state can be used in a cluster. When a capacity provider is successfully deleted, it has an `INACTIVE` status.

Type: String

Valid Values: `ACTIVE | INACTIVE`

Required: No

tags

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

The following basic restrictions apply to tags:

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

Type: Array of Tag objects

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

Required: No

updateStatus

The update status of the capacity provider. The following are the possible states that is returned.

DELETE_IN_PROGRESS

The capacity provider is in the process of being deleted.

DELETE_COMPLETE

The capacity provider was successfully deleted and has an `INACTIVE` status.

DELETE_FAILED

The capacity provider can't be deleted. The update status reason provides further details about why the delete failed.

Type: String

Valid Values: `DELETE_IN_PROGRESS | DELETE_COMPLETE | DELETE_FAILED | UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED`

Required: No

updateStatusReason

The update status reason. This provides further details about the update status for the capacity provider.

Type: String

Required: No

See Also

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

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

CapacityProviderStrategyItem

The details of a capacity provider strategy. A capacity provider strategy can be set when using the RunTask or CreateCluster APIs or as the default capacity provider strategy for a cluster with the CreateCluster API.

Only capacity providers that are already associated with a cluster and have an `ACTIVE` or `UPDATING` status can be used in a capacity provider strategy. The PutClusterCapacityProviders API is used to associate a capacity provider with a cluster.

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

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

A capacity provider strategy may contain a maximum of 6 capacity providers.

Contents

capacityProvider

The short name of the capacity provider.

Type: String

Required: Yes

base

The *base* value designates how many tasks, at a minimum, to run on the specified capacity provider. Only one capacity provider in a capacity provider strategy can have a *base* defined. If no value is specified, the default value of `0` is used.

Type: Integer

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

Required: No

weight

The *weight* value designates the relative percentage of the total number of tasks launched that should use the specified capacity provider. The `weight` value is taken into consideration after the `base` value, if defined, is satisfied.

If no `weight` value is specified, the default value of `0` is used. When multiple capacity providers are specified within a capacity provider strategy, at least one of the capacity providers must have a weight value greater than zero and any capacity providers with a weight of `0` can't be used to place tasks. If you specify multiple capacity providers in a strategy that all have a weight of `0`, any `RunTask` or `CreateService` actions using the capacity provider strategy will fail.

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

Type: Integer

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

Required: No

See Also

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

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

Cluster

A regional grouping of one or more container instances where 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.

Type: Integer

Required: No

attachments

The resources attached to a cluster. When using a capacity provider with a cluster, the capacity provider and associated resources are returned as cluster attachments.

Type: Array of Attachment objects

Required: No

attachmentsStatus

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

UPDATE_IN_PROGRESS

The available capacity providers for the cluster are updating.

UPDATE_COMPLETE

The capacity providers have successfully updated.

UPDATE_FAILED

The capacity provider updates failed.

Type: String

Required: No

capacityProviders

The capacity providers associated with the cluster.

Type: Array of strings

Required: No

clusterArn

The Amazon Resource Name (ARN) that identifies the cluster. For more information about the ARN format, see Amazon Resource Name (ARN) in the *Amazon ECS Developer Guide*.

Type: String

Required: No

clusterName

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

Type: String

Required: No

configuration

The execute command configuration for the cluster.

Type: ClusterConfiguration object

Required: No

defaultCapacityProviderStrategy

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

Type: Array of CapacityProviderStrategyItem objects

Required: No

pendingTasksCount

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

Type: Integer

Required: No

registeredContainerInstancesCount

The number of container instances registered into the cluster. This includes container instances in both `ACTIVE` and `DRAINING` status.

Type: Integer

Required: No

runningTasksCount

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

Type: Integer

Required: No

serviceConnectDefaults

Use this parameter to set a default Service Connect namespace. After you set a default Service Connect namespace, any new services with Service Connect turned on that are created in the cluster are added as client services in the namespace. This setting only applies to new services that set the `enabled` parameter to `true` in the `ServiceConnectConfiguration`. You can set the namespace of each service individually in the `ServiceConnectConfiguration` to override this default parameter.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ClusterServiceConnectDefaults object

Required: No

settings

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

Type: Array of ClusterSetting objects

Required: No

statistics

Additional information about your clusters that are separated by launch type. They include the following:

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

Type: Array of KeyValuePair objects

Required: No

status

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

ACTIVE

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

PROVISIONING

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

DEPROVISIONING

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

FAILED

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

INACTIVE

The cluster has been deleted. Clusters with an `INACTIVE` status may remain discoverable in your account for a period of time. However, this behavior is subject to change in the future. We don't recommend that you rely on `INACTIVE` clusters persisting.

Type: String

Required: No

tags

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

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterConfiguration

The execute command configuration for the cluster.

Contents

executeCommandConfiguration

The details of the execute command configuration.

Type: ExecuteCommandConfiguration object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterServiceConnectDefaults

Use this parameter to set a default Service Connect namespace. After you set a default Service Connect namespace, any new services with Service Connect turned on that are created in the cluster are added as client services in the namespace. This setting only applies to new services that set the `enabled` parameter to `true` in the `ServiceConnectConfiguration`. You can set the namespace of each service individually in the `ServiceConnectConfiguration` to override this default parameter.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Contents

namespace

The namespace name or full Amazon Resource Name (ARN) of the AWS Cloud Map namespace. When you create a service and don't specify a Service Connect configuration, this namespace is used.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterServiceConnectDefaultsRequest

Use this parameter to set a default Service Connect namespace. After you set a default Service Connect namespace, any new services with Service Connect turned on that are created in the cluster are added as client services in the namespace. This setting only applies to new services that set the `enabled` parameter to `true` in the `ServiceConnectConfiguration`. You can set the namespace of each service individually in the `ServiceConnectConfiguration` to override this default parameter.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Contents

namespace

The namespace name or full Amazon Resource Name (ARN) of the AWS Cloud Map namespace that's used when you create a service and don't specify a Service Connect configuration. The namespace name can include up to 1024 characters. The name is case-sensitive. The name can't include hyphens (-), tilde (~), greater than (>), less than (<), or slash (/).

If you enter an existing namespace name or ARN, then that namespace will be used. Any namespace type is supported. The namespace must be in this account and this AWS Region.

If you enter a new name, a AWS Cloud Map namespace will be created. Amazon ECS creates a AWS Cloud Map namespace with the "API calls" method of instance discovery only. This instance discovery method is the "HTTP" namespace type in the AWS Command Line Interface. Other types of instance discovery aren't used by Service Connect.

If you update the cluster with an empty string `""` for the namespace name, the cluster configuration for Service Connect is removed. Note that the namespace will remain in AWS Cloud Map and must be deleted separately.

For more information about AWS Cloud Map, see Working with Services in the *AWS Cloud Map Developer Guide*.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClusterSetting

The settings to use when creating a cluster. This parameter is used to turn on CloudWatch Container Insights for a cluster.

Contents

name

The name of the cluster setting. The value is `containerInsights` .

Type: String

Valid Values: `containerInsights`

Required: No

value

The value to set for the cluster setting. The supported values are `enabled` and `disabled`.

If you set `name` to `containerInsights` and `value` to `enabled`, CloudWatch Container Insights will be on for the cluster, otherwise it will be off unless the `containerInsights` account setting is turned on. If a cluster value is specified, it will override the `containerInsights` value set with PutAccountSetting or PutAccountSettingDefault.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Container

A Docker container that's part of a task.

Contents

containerArn

The Amazon Resource Name (ARN) of the container.

Type: String

Required: No

cpu

The number of CPU units set for the container. The value is `0` if no value was specified in the container definition when the task definition was registered.

Type: String

Required: No

exitCode

The exit code returned from the container.

Type: Integer

Required: No

gpuIds

The IDs of each GPU assigned to the container.

Type: Array of strings

Required: No

healthStatus

The health status of the container. If health checks aren't configured for this container in its task definition, then it reports the health status as `UNKNOWN`.

Type: String

Valid Values: `HEALTHY | UNHEALTHY | UNKNOWN`

Required: No

image

The image used for the container.

Type: String

Required: No

imageDigest

The container image manifest digest.

Type: String

Required: No

lastStatus

The last known status of the container.

Type: String

Required: No

managedAgents

The details of any Amazon ECS managed agents associated with the container.

Type: Array of ManagedAgent objects

Required: No

memory

The hard limit (in MiB) of memory set for the container.

Type: String

Required: No

memoryReservation

The soft limit (in MiB) of memory set for the container.

Type: String

Required: No

name

The name of the container.

Type: String

Required: No

networkBindings

The network bindings associated with the container.

Type: Array of NetworkBinding objects

Required: No

networkInterfaces

The network interfaces associated with the container.

Type: Array of NetworkInterface objects

Required: No

reason

A short (255 max characters) human-readable string to provide additional details about a running or stopped container.

Type: String

Required: No

runtimeId

The ID of the Docker container.

Type: String

Required: No

taskArn

The ARN of the task.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerDefinition

Container definitions are used in task definitions to describe the different containers that are launched as part of a task.

Contents

command

The command that's passed to the container. This parameter maps to `Cmd` in the Create a container section of the Docker Remote API and the `COMMAND` parameter to docker run. For more information, see https://docs.docker.com/engine/reference/builder/#cmd. If there are multiple arguments, each argument is a separated string in the array.

Type: Array of strings

Required: No

cpu

The number of `cpu` units reserved for the container. This parameter maps to `CpuShares` in the Create a container section of the Docker Remote API and the `--cpu-shares` option to docker run.

This field is optional for tasks using the Fargate launch type, and the only requirement is that the total amount of CPU reserved for all containers within a task be lower than the task-level `cpu` value.

> **Note**
>
> You can determine the number of CPU units that are available per EC2 instance type by multiplying the vCPUs listed for that instance type on the Amazon EC2 Instances detail page by 1,024.

Linux containers share unallocated CPU units with other containers on the container instance with the same ratio as their allocated amount. For example, if you run a single-container task on a single-core instance type with 512 CPU units specified for that container, and that's 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 is guaranteed a minimum of 512 CPU units when needed. Moreover, each container could float to higher CPU usage if the other container was not using it. If both tasks were 100% active all of the time, they would be limited to 512 CPU units.

On Linux container instances, the Docker daemon on the container instance uses the CPU value to calculate the relative CPU share ratios for running containers. For more information, see CPU share constraint in the Docker documentation. The minimum valid CPU share value that the Linux kernel allows is 2. However, the CPU parameter isn't 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 two CPU shares.
- **Agent versions greater than or equal to 1.2.0:** Null, zero, and CPU values of 1 are passed to Docker as 2.

On Windows container instances, the CPU limit is enforced as an absolute limit, or a quota. Windows containers only have access to the specified amount of CPU that's described in the task definition. A null or zero CPU value is passed to Docker as `0`, which Windows interprets as 1% of one CPU.

Type: Integer

Required: No

credentialSpecs

A list of ARNs in SSM or Amazon S3 to a credential spec (`CredSpec`) file that configures the container for Active Directory authentication. We recommend that you use this parameter instead of the `dockerSecurityOptions`. The maximum number of ARNs is 1.

There are two formats for each ARN.

credentialspecdomainless:MyARN

You use `credentialspecdomainless:MyARN` to provide a `CredSpec` with an additional section for a secret in AWS Secrets Manager. You provide the login credentials to the domain in the secret.

Each task that runs on any container instance can join different domains.

You can use this format without joining the container instance to a domain.

credentialspec:MyARN

You use `credentialspec:MyARN` to provide a `CredSpec` for a single domain.

You must join the container instance to the domain before you start any tasks that use this task definition.

In both formats, replace `MyARN` with the ARN in SSM or Amazon S3.

If you provide a `credentialspecdomainless:MyARN`, the `credspec` must provide a ARN in AWS Secrets Manager for a secret containing the username, password, and the domain to connect to. For better security, the instance isn't joined to the domain for domainless authentication. Other applications on the instance can't use the domainless credentials. You can use this parameter to run tasks on the same instance, even it the tasks need to join different domains. For more information, see Using gMSAs for Windows Containers and Using gMSAs for Linux Containers.

Type: Array of strings

Required: No

dependsOn

The dependencies defined for container startup and shutdown. A container can contain multiple dependencies on other containers in a task definition. When a dependency is defined for container startup, for container shutdown it is reversed.

For tasks using the EC2 launch type, the container instances require at least version 1.26.0 of the container agent to turn on container dependencies. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the *Amazon Elastic Container Service Developer Guide*. If you're using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the `ecs-init` package. If your container instances are launched from version `20190301` or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the *Amazon Elastic Container Service Developer Guide*.

For tasks using the Fargate launch type, the task or service requires the following platforms:

- Linux platform version `1.3.0` or later.

- Windows platform version `1.0.0` or later.

Type: Array of ContainerDependency objects

Required: No

disableNetworking

When this parameter is true, networking is off within the container. This parameter maps to `NetworkDisabled` in the Create a container section of the Docker Remote API.

> **Note**
>
> This parameter is not supported for Windows containers.

Type: Boolean

Required: No

dnsSearchDomains

A list of DNS search domains that are presented to the container. This parameter maps to `DnsSearch` in the Create a container section of the Docker Remote API and the `--dns-search` option to docker run.

> **Note**
>
> This parameter is not supported for Windows containers.

Type: Array of strings

Required: No

dnsServers

A list of DNS servers that are presented to the container. This parameter maps to `Dns` in the Create a container section of the Docker Remote API and the `--dns` option to docker run.

> **Note**
>
> This parameter is not supported for Windows containers.

Type: Array of strings

Required: No

dockerLabels

A key/value map of labels to add to the container. This parameter maps to `Labels` in the Create a container section of the Docker Remote API and the `--label` option to docker run. This parameter requires version 1.18 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version --format '{{.Server.APIVersion}}'`

Type: String to string map

Required: No

dockerSecurityOptions

A list of strings to provide custom configuration for multiple security systems. For more information about valid values, see Docker Run Security Configuration. This field isn't valid for containers in tasks using the Fargate launch type.

For Linux tasks on EC2, this parameter can be used to reference custom labels for SELinux and AppArmor multi-level security systems.

For any tasks on EC2, this parameter can be used to reference a credential spec file that configures a container for Active Directory authentication. For more information, see Using gMSAs for Windows Containers and Using gMSAs for Linux Containers in the *Amazon Elastic Container Service Developer Guide*.

This parameter maps to `SecurityOpt` in the Create a container section of the Docker Remote API and the `--security-opt` option to docker run.

> **Note**
>
> The Amazon ECS container agent running on a container instance must register with the `ECS_SELINUX_CAPABLE=true` or `ECS_APPARMOR_CAPABLE=true` environment variables before containers placed on that instance can use these security options. For more information, see Amazon ECS Container Agent Configuration in the *Amazon Elastic Container Service Developer Guide*.

For more information about valid values, see Docker Run Security Configuration.

Valid values: "no-new-privileges" | "apparmor:PROFILE" | "label:value" | "credentialspec:CredentialSpecFilePath"

Type: Array of strings

Required: No

entryPoint

> **Important**
>
> Early versions of the Amazon ECS container agent don't properly handle `entryPoint` parameters. If you have problems using `entryPoint`, update your container agent or enter your commands and arguments as `command` array items instead.

The entry point that's passed to the container. This parameter maps to `Entrypoint` in the Create a container section of the Docker Remote API and the `--entrypoint` option to docker run. For more information, see https://docs.docker.com/engine/reference/builder/#entrypoint.

Type: Array of strings

Required: No

environment

The environment variables to pass to a container. This parameter maps to `Env` in the Create a container section of the Docker Remote API and the `--env` option to docker run.

> **Important**
>
> We don't recommend that you use plaintext environment variables for sensitive information, such as credential data.

Type: Array of KeyValuePair objects

Required: No

environmentFiles

A list of files containing the environment variables to pass to a container. This parameter maps to the `--env-file` option to docker run.

You can specify up to ten environment files. The file must have a `.env` file extension. Each line in an environment file contains an environment variable in `VARIABLE=VALUE` format. Lines beginning with # are treated as comments and are ignored. For more information about the environment variable file syntax, see Declare default environment variables in file.

If there are environment variables specified using the `environment` parameter in a container definition, they take precedence over the variables contained within an environment file. If multiple environment files are specified that contain the same variable, they're processed from the top down. We recommend that you use unique variable names. For more information, see Specifying Environment Variables in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of EnvironmentFile objects

Required: No

essential

If the `essential` parameter of a container is marked as `true`, and that container fails or stops for any reason, all other containers that are part of the task are stopped. If the `essential` parameter of a container is marked as `false`, its failure doesn't affect the rest of the containers in a task. If this parameter is omitted, a container is assumed to be essential.

All tasks must have at least one essential container. If you have an application that's composed of multiple containers, group containers that are used for a common purpose into components, and separate the different components into multiple task definitions. For more information, see Application Architecture in the *Amazon Elastic Container Service Developer Guide*.

Type: Boolean

Required: No

extraHosts

A list of hostnames and IP address mappings to append to the `/etc/hosts` file on the container. This parameter maps to `ExtraHosts` in the Create a container section of the Docker Remote API and the `--add-host` option to docker run.

> **Note**
>
> This parameter isn't supported for Windows containers or tasks that use the `awsvpc` network mode.

Type: Array of HostEntry objects

Required: No

firelensConfiguration

The FireLens configuration for the container. This is used to specify and configure a log router for container logs. For more information, see Custom Log Routing in the *Amazon Elastic Container Service Developer Guide*.

Type: FirelensConfiguration object

Required: No

healthCheck

The container health check command and associated configuration parameters for the container. This parameter maps to `HealthCheck` in the Create a container section of the Docker Remote API and the `HEALTHCHECK` parameter of docker run.

Type: HealthCheck object

Required: No

hostname

The hostname to use for your container. This parameter maps to `Hostname` in the Create a container section of the Docker Remote API and the `--hostname` option to docker run.

> **Note**
>
> The `hostname` parameter is not supported if you're using the `awsvpc` network mode.

Type: String

Required: No

image

The image used to start a container. This string is passed directly to the Docker daemon. By default, images in the Docker Hub registry are available. Other repositories are specified with either *repository-url/image*:*tag* or *repository-url/image*@*digest* . Up to 255 letters (uppercase and lowercase), numbers, hyphens, underscores, colons, periods, forward slashes, and number signs are allowed. This parameter maps to `Image` in the Create a container section of the Docker Remote API and the `IMAGE` parameter of docker run.

- When a new task starts, the Amazon ECS container agent pulls the latest version of the specified image and tag for the container to use. However, subsequent updates to a repository image aren't propagated to already running tasks.
- Images in Amazon ECR repositories can be specified by either using the full `registry/repository:tag` or `registry/repository@digest`. For example, `012345678910.dkr.ecr.<region-name>.amazonaws.com/<repository-name>:latest` or `012345678910.dkr.ecr.<region-name>.amazonaws.com/<repository-name>@sha256:94afd1f2e64d908bc90dbca0035a5b567EXAMPLE`.
- Images in official repositories on Docker Hub use a single name (for example, `ubuntu` or `mongo`).
- Images in other repositories on Docker Hub are qualified with an organization name (for example, `amazon/amazon-ecs-agent`).
- Images in other online repositories are qualified further by a domain name (for example, `quay.io/assemblyline/ubuntu`).

Type: String

Required: No

interactive

When this parameter is `true`, you can deploy containerized applications that require `stdin` or a `tty` to be allocated. This parameter maps to `OpenStdin` in the Create a container section of the Docker Remote API and the `--interactive` option to docker run.

Type: Boolean

Required: No

links

The `links` parameter allows containers to communicate with each other without the need for port mappings. This parameter is only supported if the network mode of a task definition is `bridge`. The `name:internalName` construct is analogous to `name:alias` in Docker links. Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed. For more information about linking Docker containers, go to Legacy container links in the Docker documentation. This parameter maps to `Links` in the Create a container section of the Docker Remote API and the `--link` option to docker run.

> **Note**
>
> This parameter is not supported for Windows containers.

> **Important**
>
> Containers that are collocated on a single container instance may be able to communicate with each other without requiring links or host port mappings. Network isolation is achieved on the container instance using security groups and VPC settings.

Type: Array of strings

Required: No

linuxParameters

Linux-specific modifications that are applied to the container, such as Linux kernel capabilities. For more information see KernelCapabilities.

> **Note**
>
> This parameter is not supported for Windows containers.

Type: LinuxParameters object

Required: No

logConfiguration

The log configuration specification for the container.

This parameter maps to `LogConfig` in the Create a container section of the Docker Remote API and the `--log-driver` option to docker run. By default, containers use the same logging driver that the Docker daemon uses. However the container can use a different logging driver than the Docker daemon by specifying a log driver with this parameter in the container definition. To use a different logging driver for a container, the log system must be configured properly on the container instance (or on a different log server for remote logging options). For more information about the options for different supported log drivers, see Configure logging drivers in the Docker documentation.

> **Note**
>
> Amazon ECS currently supports a subset of the logging drivers available to the Docker daemon (shown in the LogConfiguration data type). Additional log drivers may be available in future releases of the Amazon ECS container agent.

This parameter requires version 1.18 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version --format '{{.Server.APIVersion}}'`

> **Note**
>
> The Amazon ECS container agent running on a container instance must register the logging drivers available on that instance with the `ECS_AVAILABLE_LOGGING_DRIVERS` environment variable before containers placed on that instance can use these log configuration options. For more information, see Amazon ECS Container Agent Configuration in the *Amazon Elastic Container Service Developer Guide*.

Type: LogConfiguration object

Required: No

memory

The amount (in MiB) of memory to present to the container. If your container attempts to exceed the memory specified here, the container is killed. The total amount of memory reserved for all containers within a task must be lower than the task `memory` value, if one is specified. This parameter maps to `Memory` in the Create a container section of the Docker Remote API and the `--memory` option to docker run.

If using the Fargate launch type, this parameter is optional.

If using the EC2 launch type, you must specify either a task-level memory value or a container-level memory value. If you specify both a container-level `memory` and `memoryReservation` value, `memory` must be greater than `memoryReservation`. If you specify `memoryReservation`, then that value is subtracted from the available memory resources for the container instance where the container is placed. Otherwise, the value of `memory` is used.

The Docker 20.10.0 or later daemon reserves a minimum of 6 MiB of memory for a container. So, don't specify less than 6 MiB of memory for your containers.

The Docker 19.03.13-ce or earlier daemon reserves a minimum of 4 MiB of memory for a container. So, don't specify less than 4 MiB of memory for your containers.

Type: Integer

Required: No

memoryReservation

The soft limit (in MiB) of memory to reserve for the container. When system memory is under heavy contention, Docker attempts to keep the container memory to this soft limit. However, your container can consume more memory when it needs to, up to either the hard limit specified with the `memory` parameter (if applicable), or all of the available memory on the container instance, whichever comes first. This parameter maps to `MemoryReservation` in the Create a container section of the Docker Remote API and the `--memory-reservation` option to docker run.

If a task-level memory value is not specified, you must specify a non-zero integer for one or both of `memory` or `memoryReservation` in a container definition. If you specify both, `memory` must be greater than `memoryReservation`. If you specify `memoryReservation`, then that

value is subtracted from the available memory resources for the container instance where the container is placed. Otherwise, the value of `memory` is used.

For example, if your container normally uses 128 MiB of memory, but occasionally bursts to 256 MiB of memory for short periods of time, you can set a `memoryReservation` of 128 MiB, and a `memory` hard limit of 300 MiB. This configuration would allow the container to only reserve 128 MiB of memory from the remaining resources on the container instance, but also allow the container to consume more memory resources when needed.

The Docker 20.10.0 or later daemon reserves a minimum of 6 MiB of memory for a container. So, don't specify less than 6 MiB of memory for your containers.

The Docker 19.03.13-ce or earlier daemon reserves a minimum of 4 MiB of memory for a container. So, don't specify less than 4 MiB of memory for your containers.

Type: Integer

Required: No

mountPoints

The mount points for data volumes in your container.

This parameter maps to `Volumes` in the Create a container section of the Docker Remote API and the `--volume` option to docker run.

Windows containers can mount whole directories on the same drive as `$env:ProgramData`. Windows containers can't mount directories on a different drive, and mount point can't be across drives.

Type: Array of MountPoint objects

Required: No

name

The name of a container. If you're linking multiple containers together in a task definition, the `name` of one container can be entered in the `links` of another container to connect the containers. Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed. This parameter maps to `name` in the Create a container section of the Docker Remote API and the `--name` option to docker run.

Type: String

Required: No

portMappings

The list of port mappings for the container. Port mappings allow containers to access ports on the host container instance to send or receive traffic.

For task definitions that use the `awsvpc` network mode, only specify the `containerPort`. The `hostPort` can be left blank or it must be the same value as the `containerPort`.

Port mappings on Windows use the `NetNAT` gateway address rather than `localhost`. There's no loopback for port mappings on Windows, so you can't access a container's mapped port from the host itself.

This parameter maps to `PortBindings` in the Create a container section of the Docker Remote API and the `--publish` option to docker run. If the network mode of a task definition is set to none, then you can't specify port mappings. If the network mode of a task definition is set to `host`, then host ports must either be undefined or they must match the container port in the port mapping.

> **Note**
>
> After a task reaches the `RUNNING` status, manual and automatic host and container port assignments are visible in the **Network Bindings** section of a container description for a selected task in the Amazon ECS console. The assignments are also visible in the `networkBindings` section DescribeTasks responses.

Type: Array of PortMapping objects

Required: No

privileged

When this parameter is true, the container is given elevated privileges on the host container instance (similar to the `root` user). This parameter maps to `Privileged` in the Create a container section of the Docker Remote API and the `--privileged` option to docker run.

> **Note**
>
> This parameter is not supported for Windows containers or tasks run on AWS Fargate.

Type: Boolean

Required: No

pseudoTerminal

When this parameter is `true`, a TTY is allocated. This parameter maps to `Tty` in the Create a container section of the Docker Remote API and the `--tty` option to docker run.

Type: Boolean

Required: No

readonlyRootFilesystem

When this parameter is true, the container is given read-only access to its root file system. This parameter maps to `ReadonlyRootfs` in the Create a container section of the Docker Remote API and the `--read-only` option to docker run.

> **Note**
>
> This parameter is not supported for Windows containers.

Type: Boolean

Required: No

repositoryCredentials

The private repository authentication credentials to use.

Type: RepositoryCredentials object

Required: No

resourceRequirements

The type and amount of a resource to assign to a container. The only supported resource is a GPU.

Type: Array of ResourceRequirement objects

Required: No

secrets

The secrets to pass to the container. For more information, see Specifying Sensitive Data in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of Secret objects

Required: No

startTimeout

Time duration (in seconds) to wait before giving up on resolving dependencies for a container. For example, you specify two containers in a task definition with containerA having a dependency on containerB reaching a `COMPLETE`, `SUCCESS`, or `HEALTHY` status. If a `startTimeout` value is specified for containerB and it doesn't reach the desired status within that time then containerA gives up and not start. This results in the task transitioning to a `STOPPED` state.

> **Note**
>
> When the `ECS_CONTAINER_START_TIMEOUT` container agent configuration variable is used, it's enforced independently from this start timeout value.

For tasks using the Fargate launch type, the task or service requires the following platforms:

- Linux platform version `1.3.0` or later.
- Windows platform version `1.0.0` or later.

For tasks using the EC2 launch type, your container instances require at least version `1.26.0` of the container agent to use a container start timeout value. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the *Amazon Elastic Container Service Developer Guide*. If you're using an Amazon ECS-optimized Linux AMI, your instance needs at least version `1.26.0-1` of the `ecs-init` package. If your container instances are launched from version `20190301` or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the *Amazon Elastic Container Service Developer Guide*.

The valid values are 2-120 seconds.

Type: Integer

Required: No

stopTimeout

Time duration (in seconds) to wait before the container is forcefully killed if it doesn't exit normally on its own.

For tasks using the Fargate launch type, the task or service requires the following platforms:

- Linux platform version `1.3.0` or later.
- Windows platform version `1.0.0` or later.

The max stop timeout value is 120 seconds and if the parameter is not specified, the default value of 30 seconds is used.

For tasks that use the EC2 launch type, if the `stopTimeout` parameter isn't specified, the value set for the Amazon ECS container agent configuration variable `ECS_CONTAINER_STOP_TIMEOUT` is used. If neither the `stopTimeout` parameter or the `ECS_CONTAINER_STOP_TIMEOUT` agent configuration variable are set, then the default values of 30 seconds for Linux containers and 30 seconds on Windows containers are used. Your container instances require at least version 1.26.0 of the container agent to use a container stop timeout value. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the *Amazon Elastic Container Service Developer Guide*. If you're using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the `ecs-init` package. If your container instances are launched from version `20190301` or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the *Amazon Elastic Container Service Developer Guide*.

The valid values are 2-120 seconds.

Type: Integer

Required: No

systemControls

A list of namespaced kernel parameters to set in the container. This parameter maps to `Sysctls` in the Create a container section of the Docker Remote API and the `--sysctl` option

to docker run. For example, you can configure `net.ipv4.tcp_keepalive_time` setting to maintain longer lived connections.

Type: Array of SystemControl objects

Required: No

ulimits

A list of `ulimits` to set in the container. If a `ulimit` value is specified in a task definition, it overrides the default values set by Docker. This parameter maps to `Ulimits` in the Create a container section of the Docker Remote API and the `--ulimit` option to docker run. Valid naming values are displayed in the Ulimit data type.

Amazon ECS tasks hosted on AWS Fargate use the default resource limit values set by the operating system with the exception of the `nofile` resource limit parameter which AWS Fargate overrides. The `nofile` resource limit sets a restriction on the number of open files that a container can use. The default `nofile` soft limit is `1024` and the default hard limit is `65535`.

This parameter requires version 1.18 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version --format '{{.Server.APIVersion}}'`

> **Note**
>
> This parameter is not supported for Windows containers.

Type: Array of Ulimit objects

Required: No

user

The user to use inside the container. This parameter maps to `User` in the Create a container section of the Docker Remote API and the `--user` option to docker run.

> **Important**
>
> When running tasks using the `host` network mode, don't run containers using the root user (UID 0). We recommend using a non-root user for better security.

You can specify the `user` using the following formats. If specifying a UID or GID, you must specify it as a positive integer.

- `user`
- `user:group`
- `uid`
- `uid:gid`
- `user:gid`
- `uid:group`

> **Note**
>
> This parameter is not supported for Windows containers.

Type: String

Required: No

volumesFrom

Data volumes to mount from another container. This parameter maps to `VolumesFrom` in the Create a container section of the Docker Remote API and the `--volumes-from` option to docker run.

Type: Array of VolumeFrom objects

Required: No

workingDirectory

The working directory to run commands inside the container in. This parameter maps to `WorkingDir` in the Create a container section of the Docker Remote API and the `--workdir` option to docker run.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerDependency

The dependencies defined for container startup and shutdown. A container can contain multiple dependencies. When a dependency is defined for container startup, for container shutdown it is reversed.

Your Amazon ECS container instances require at least version 1.26.0 of the container agent to use container dependencies. However, we recommend using the latest container agent version. For information about checking your agent version and updating to the latest version, see Updating the Amazon ECS Container Agent in the *Amazon Elastic Container Service Developer Guide*. If you're using an Amazon ECS-optimized Linux AMI, your instance needs at least version 1.26.0-1 of the `ecs-init` package. If your container instances are launched from version `20190301` or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> For tasks that use the Fargate launch type, the task or service requires the following platforms:
>
> - Linux platform version `1.3.0` or later.
> - Windows platform version `1.0.0` or later.

For more information about how to create a container dependency, see Container dependency in the *Amazon Elastic Container Service Developer Guide*.

Contents

condition

The dependency condition of the container. The following are the available conditions and their behavior:

- `START` - This condition emulates the behavior of links and volumes today. It validates that a dependent container is started before permitting other containers to start.
- `COMPLETE` - This condition validates that a dependent container runs to completion (exits) before permitting other containers to start. This can be useful for nonessential containers that run a script and then exit. This condition can't be set on an essential container.

- SUCCESS - This condition is the same as COMPLETE, but it also requires that the container exits with a zero status. This condition can't be set on an essential container.
- HEALTHY - This condition validates that the dependent container passes its Docker health check before permitting other containers to start. This requires that the dependent container has health checks configured. This condition is confirmed only at task startup.

Type: String

Valid Values: START | COMPLETE | SUCCESS | HEALTHY

Required: Yes

containerName

The name of a container.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerInstance

An Amazon EC2 or External instance that's running the Amazon ECS agent and has been registered with a cluster.

Contents

agentConnected

This parameter returns `true` if the agent is connected to Amazon ECS. An instance with an agent that may be unhealthy or stopped return `false`. Only instances connected to an agent can accept task placement requests.

Type: Boolean

Required: No

agentUpdateStatus

The status of the most recent agent update. If an update wasn't ever requested, this value is `NULL`.

Type: String

Valid Values: `PENDING | STAGING | STAGED | UPDATING | UPDATED | FAILED`

Required: No

attachments

The resources attached to a container instance, such as an elastic network interface.

Type: Array of Attachment objects

Required: No

attributes

The attributes set for the container instance, either by the Amazon ECS container agent at instance registration or manually with the PutAttributes operation.

Type: Array of Attribute objects

Required: No

capacityProviderName

The capacity provider that's associated with the container instance.

Type: String

Required: No

containerInstanceArn

The Amazon Resource Name (ARN) of the container instance. For more information about the ARN format, see Amazon Resource Name (ARN) in the *Amazon ECS Developer Guide*.

Type: String

Required: No

ec2InstanceId

The ID of the container instance. For Amazon EC2 instances, this value is the Amazon EC2 instance ID. For external instances, this value is the AWS Systems Manager managed instance ID.

Type: String

Required: No

healthStatus

An object representing the health status of the container instance.

Type: ContainerInstanceHealthStatus object

Required: No

pendingTasksCount

The number of tasks on the container instance that are in the `PENDING` status.

Type: Integer

Required: No

registeredAt

The Unix timestamp for the time when the container instance was registered.

Type: Timestamp

Required: No

registeredResources

For CPU and memory resource types, this parameter describes the amount of each resource that was available on the container instance when the container agent registered it with Amazon ECS. This value represents the total amount of CPU and memory that can be allocated on this container instance to tasks. For port resource types, this parameter describes the ports that were reserved by the Amazon ECS container agent when it registered the container instance with Amazon ECS.

Type: Array of Resource objects

Required: No

remainingResources

For CPU and memory resource types, this parameter describes the remaining CPU and memory that wasn't already allocated to tasks and is therefore available for new tasks. For port resource types, this parameter describes the ports that were reserved by the Amazon ECS container agent (at instance registration time) and any task containers that have reserved port mappings on the host (with the `host` or `bridge` network mode). Any port that's not specified here is available for new tasks.

Type: Array of Resource objects

Required: No

runningTasksCount

The number of tasks on the container instance that have a desired status (`desiredStatus`) of `RUNNING`.

Type: Integer

Required: No

status

The status of the container instance. The valid values are `REGISTERING`, `REGISTRATION_FAILED`, `ACTIVE`, `INACTIVE`, `DEREGISTERING`, or `DRAINING`.

If your account has opted in to the `awsvpcTrunking` account setting, then any newly registered container instance will transition to a `REGISTERING` status while the trunk elastic

network interface is provisioned for the instance. If the registration fails, the instance will transition to a `REGISTRATION_FAILED` status. You can describe the container instance and see the reason for failure in the `statusReason` parameter. Once the container instance is terminated, the instance transitions to a `DEREGISTERING` status while the trunk elastic network interface is deprovisioned. The instance then transitions to an `INACTIVE` status.

The `ACTIVE` status indicates that the container instance can accept tasks. The `DRAINING` indicates that new tasks aren't placed on the container instance and any service tasks running on the container instance are removed if possible. For more information, see Container instance draining in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

statusReason

The reason that the container instance reached its current status.

Type: String

Required: No

tags

The metadata that you apply to the container instance to help you categorize and organize them. Each tag consists of a key and an optional value. You define both.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

version

The version counter for the container instance. Every time a container instance experiences a change that triggers a CloudWatch event, the version counter is incremented. If you're replicating your Amazon ECS container instance state with CloudWatch Events, you can compare the version of a container instance reported by the Amazon ECS APIs with the version reported in CloudWatch Events for the container instance (inside the `detail` object) to verify that the version in your event stream is current.

Type: Long

Required: No

versionInfo

The version information for the Amazon ECS container agent and Docker daemon running on the container instance.

Type: VersionInfo object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerInstanceHealthStatus

An object representing the health status of the container instance.

Contents

details

An array of objects representing the details of the container instance health status.

Type: Array of InstanceHealthCheckResult objects

Required: No

overallStatus

The overall health status of the container instance. This is an aggregate status of all container instance health checks.

Type: String

Valid Values: `OK | IMPAIRED | INSUFFICIENT_DATA | INITIALIZING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerOverride

The overrides that are sent to a container. An empty container override can be passed in. An example of an empty container override is `{"containerOverrides": [ ] }`. If a non-empty container override is specified, the `name` parameter must be included.

You can use Secrets Manager or AWS Systems Manager Parameter Store to store the sensitive data. For more information, see Retrieve secrets through environment variables in the Amazon ECS Developer Guide.

Contents

command

The command to send to the container that overrides the default command from the Docker image or the task definition. You must also specify a container name.

Type: Array of strings

Required: No

cpu

The number of `cpu` units reserved for the container, instead of the default value from the task definition. You must also specify a container name.

Type: Integer

Required: No

environment

The environment variables to send to the container. You can add new environment variables, which are added to the container at launch, or you can override the existing environment variables from the Docker image or the task definition. You must also specify a container name.

Type: Array of KeyValuePair objects

Required: No

environmentFiles

A list of files containing the environment variables to pass to a container, instead of the value from the container definition.

Type: Array of EnvironmentFile objects

Required: No

memory

The hard limit (in MiB) of memory to present to the container, instead of the default value from the task definition. If your container attempts to exceed the memory specified here, the container is killed. You must also specify a container name.

Type: Integer

Required: No

memoryReservation

The soft limit (in MiB) of memory to reserve for the container, instead of the default value from the task definition. You must also specify a container name.

Type: Integer

Required: No

name

The name of the container that receives the override. This parameter is required if any override is specified.

Type: String

Required: No

resourceRequirements

The type and amount of a resource to assign to a container, instead of the default value from the task definition. The only supported resource is a GPU.

Type: Array of ResourceRequirement objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContainerStateChange

An object that represents a change in state for a container.

Contents

containerName

The name of the container.

Type: String

Required: No

exitCode

The exit code for the container, if the state change is a result of the container exiting.

Type: Integer

Required: No

imageDigest

The container image SHA 256 digest.

Type: String

Required: No

networkBindings

Any network bindings that are associated with the container.

Type: Array of NetworkBinding objects

Required: No

reason

The reason for the state change.

Type: String

Required: No

runtimeId

The ID of the Docker container.

Type: String

Required: No

status

The status of the container.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Deployment

The details of an Amazon ECS service deployment. This is used only when a service uses the ECS deployment controller type.

Contents

capacityProviderStrategy

The capacity provider strategy that the deployment is using.

Type: Array of CapacityProviderStrategyItem objects

Required: No

createdAt

The Unix timestamp for the time when the service deployment was created.

Type: Timestamp

Required: No

desiredCount

The most recent desired count of tasks that was specified for the service to deploy or maintain.

Type: Integer

Required: No

failedTasks

The number of consecutively failed tasks in the deployment. A task is considered a failure if the service scheduler can't launch the task, the task doesn't transition to a `RUNNING` state, or if it fails any of its defined health checks and is stopped.

> **Note**
>
> Once a service deployment has one or more successfully running tasks, the failed task count resets to zero and stops being evaluated.

Type: Integer

Required: No

id

The ID of the deployment.

Type: String

Required: No

launchType

The launch type the tasks in the service are using. For more information, see Amazon ECS Launch Types in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Valid Values: `EC2` | `FARGATE` | `EXTERNAL`

Required: No

networkConfiguration

The VPC subnet and security group configuration for tasks that receive their own elastic network interface by using the `awsvpc` networking mode.

Type: NetworkConfiguration object

Required: No

pendingCount

The number of tasks in the deployment that are in the `PENDING` status.

Type: Integer

Required: No

platformFamily

The operating system that your tasks in the service, or tasks are running on. A platform family is specified only for tasks using the Fargate launch type.

All tasks that run as part of this service must use the same `platformFamily` value as the service, for example, `LINUX.`.

Type: String

Required: No

platformVersion

The platform version that your tasks in the service run on. A platform version is only specified for tasks using the Fargate launch type. If one isn't specified, the `LATEST` platform version is used. For more information, see AWS Fargate Platform Versions in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

rolloutState

> **Note**
>
> The `rolloutState` of a service is only returned for services that use the rolling update (`ECS`) deployment type that aren't behind a Classic Load Balancer.

The rollout state of the deployment. When a service deployment is started, it begins in an `IN_PROGRESS` state. When the service reaches a steady state, the deployment transitions to a `COMPLETED` state. If the service fails to reach a steady state and circuit breaker is turned on, the deployment transitions to a `FAILED` state. A deployment in `FAILED` state doesn't launch any new tasks. For more information, see DeploymentCircuitBreaker.

Type: String

Valid Values: `COMPLETED | FAILED | IN_PROGRESS`

Required: No

rolloutStateReason

A description of the rollout state of a deployment.

Type: String

Required: No

runningCount

The number of tasks in the deployment that are in the `RUNNING` status.

Type: Integer

Required: No

serviceConnectConfiguration

The details of the Service Connect configuration that's used by this deployment. Compare the configuration between multiple deployments when troubleshooting issues with new deployments.

The configuration for this service to discover and connect to services, and be discovered by, and connected from, other services within a namespace.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: ServiceConnectConfiguration object

Required: No

serviceConnectResources

The list of Service Connect resources that are associated with this deployment. Each list entry maps a discovery name to a AWS Cloud Map service name.

Type: Array of ServiceConnectServiceResource objects

Required: No

status

The status of the deployment. The following describes each state.

PRIMARY

The most recent deployment of a service.

ACTIVE

A service deployment that still has running tasks, but are in the process of being replaced with a new PRIMARY deployment.

INACTIVE

A deployment that has been completely replaced.

Type: String

Required: No

taskDefinition

The most recent task definition that was specified for the tasks in the service to use.

Type: String

Required: No

updatedAt

The Unix timestamp for the time when the service deployment was last updated.

Type: Timestamp

Required: No

volumeConfigurations

The details of the volume that was `configuredAtLaunch`. You can configure different settings like the size, throughput, volumeType, and ecryption in ServiceManagedEBSVolumeConfiguration. The `name` of the volume must match the `name` from the task definition.

Type: Array of ServiceVolumeConfiguration objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeploymentAlarms

One of the methods which provide a way for you to quickly identify when a deployment has failed, and then to optionally roll back the failure to the last working deployment.

When the alarms are generated, Amazon ECS sets the service deployment to failed. Set the rollback parameter to have Amazon ECS to roll back your service to the last completed deployment after a failure.

You can only use the `DeploymentAlarms` method to detect failures when the `DeploymentController` is set to `ECS` (rolling update).

For more information, see Rolling update in the Amazon Elastic Container Service Developer Guide .

Contents

alarmNames

One or more CloudWatch alarm names. Use a "," to separate the alarms.

Type: Array of strings

Required: Yes

enable

Determines whether to use the CloudWatch alarm option in the service deployment process.

Type: Boolean

Required: Yes

rollback

Determines whether to configure Amazon ECS to roll back the service if a service deployment fails. If rollback is used, when a service deployment fails, the service is rolled back to the last deployment that completed successfully.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeploymentCircuitBreaker

> **Note**
>
> The deployment circuit breaker can only be used for services using the rolling update (ECS) deployment type.

The **deployment circuit breaker** determines whether a service deployment will fail if the service can't reach a steady state. If it is turned on, a service deployment will transition to a failed state and stop launching new tasks. You can also configure Amazon ECS to roll back your service to the last completed deployment after a failure. For more information, see Rolling update in the *Amazon Elastic Container Service Developer Guide*.

For more information about API failure reasons, see API failure reasons in the *Amazon Elastic Container Service Developer Guide*.

Contents

enable

Determines whether to use the deployment circuit breaker logic for the service.

Type: Boolean

Required: Yes

rollback

Determines whether to configure Amazon ECS to roll back the service if a service deployment fails. If rollback is on, when a service deployment fails, the service is rolled back to the last deployment that completed successfully.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeploymentConfiguration

Optional deployment parameters that control how many tasks run during a deployment and the ordering of stopping and starting tasks.

Contents

alarms

Information about the CloudWatch alarms.

Type: DeploymentAlarms object

Required: No

deploymentCircuitBreaker

> **Note**
>
> The deployment circuit breaker can only be used for services using the rolling update (ECS) deployment type.

The **deployment circuit breaker** determines whether a service deployment will fail if the service can't reach a steady state. If you use the deployment circuit breaker, a service deployment will transition to a failed state and stop launching new tasks. If you use the rollback option, when a service deployment fails, the service is rolled back to the last deployment that completed successfully. For more information, see Rolling update in the *Amazon Elastic Container Service Developer Guide*

Type: DeploymentCircuitBreaker object

Required: No

maximumPercent

If a service is using the rolling update (ECS) deployment type, the `maximumPercent` parameter represents an upper limit on the number of your service's tasks that are allowed in the `RUNNING` or `PENDING` state during a deployment, as a percentage of the `desiredCount` (rounded down to the nearest integer). This parameter enables you to define the deployment batch size. For example, if your service is using the `REPLICA` service scheduler and has a `desiredCount` of

four tasks and a `maximumPercent` value of 200%, the scheduler may start four new tasks before stopping the four older tasks (provided that the cluster resources required to do this are available). The default `maximumPercent` value for a service using the `REPLICA` service scheduler is 200%.

If a service is using either the blue/green (`CODE_DEPLOY`) or `EXTERNAL` deployment types and tasks that use the EC2 launch type, the **maximum percent** value is set to the default value and is used to define the upper limit on the number of the tasks in the service that remain in the `RUNNING` state while the container instances are in the `DRAINING` state. If the tasks in the service use the Fargate launch type, the maximum percent value is not used, although it is returned when describing your service.

Type: Integer

Required: No

minimumHealthyPercent

If a service is using the rolling update (`ECS`) deployment type, the `minimumHealthyPercent` represents a lower limit on the number of your service's tasks that must remain in the `RUNNING` state during a deployment, as a percentage of the `desiredCount` (rounded up to the nearest integer). This parameter enables you to deploy without using additional cluster capacity. For example, if your service has a `desiredCount` of four tasks and a `minimumHealthyPercent` of 50%, the service scheduler may stop two existing tasks to free up cluster capacity before starting two new tasks.

For services that *do not* use a load balancer, the following should be noted:

- A service is considered healthy if all essential containers within the tasks in the service pass their health checks.
- If a task has no essential containers with a health check defined, the service scheduler will wait for 40 seconds after a task reaches a `RUNNING` state before the task is counted towards the minimum healthy percent total.
- If a task has one or more essential containers with a health check defined, the service scheduler will wait for the task to reach a healthy status before counting it towards the minimum healthy percent total. A task is considered healthy when all essential containers within the task have passed their health checks. The amount of time the service scheduler can wait for is determined by the container health check settings.

For services that *do* use a load balancer, the following should be noted:

- If a task has no essential containers with a health check defined, the service scheduler will wait for the load balancer target group health check to return a healthy status before counting the task towards the minimum healthy percent total.
- If a task has an essential container with a health check defined, the service scheduler will wait for both the task to reach a healthy status and the load balancer target group health check to return a healthy status before counting the task towards the minimum healthy percent total.

The default value for a replica service for `minimumHealthyPercent` is 100%. The default `minimumHealthyPercent` value for a service using the `DAEMON` service schedule is 0% for the AWS CLI, the AWS SDKs, and the APIs and 50% for the AWS Management Console.

The minimum number of healthy tasks during a deployment is the `desiredCount` multiplied by the `minimumHealthyPercent`/100, rounded up to the nearest integer value.

If a service is using either the blue/green (`CODE_DEPLOY`) or `EXTERNAL` deployment types and is running tasks that use the EC2 launch type, the **minimum healthy percent** value is set to the default value and is used to define the lower limit on the number of the tasks in the service that remain in the `RUNNING` state while the container instances are in the `DRAINING` state. If a service is using either the blue/green (`CODE_DEPLOY`) or `EXTERNAL` deployment types and is running tasks that use the Fargate launch type, the minimum healthy percent value is not used, although it is returned when describing your service.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeploymentController

The deployment controller to use for the service. For more information, see Amazon ECS deployment types in the *Amazon Elastic Container Service Developer Guide*.

Contents

type

The deployment controller type to use.

There are three deployment controller types available:

ECS

The rolling update (ECS) deployment type involves replacing the current running version of the container with the latest version. The number of containers Amazon ECS adds or removes from the service during a rolling update is controlled by adjusting the minimum and maximum number of healthy tasks allowed during a service deployment, as specified in the DeploymentConfiguration.

CODE_DEPLOY

The blue/green (CODE_DEPLOY) deployment type uses the blue/green deployment model powered by AWS CodeDeploy, which allows you to verify a new deployment of a service before sending production traffic to it.

EXTERNAL

The external (EXTERNAL) deployment type enables you to use any third-party deployment controller for full control over the deployment process for an Amazon ECS service.

Type: String

Valid Values: ECS | CODE_DEPLOY | EXTERNAL

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Device

An object representing a container instance host device.

Contents

hostPath

The path for the device on the host container instance.

Type: String

Required: Yes

containerPath

The path inside the container at which to expose the host device.

Type: String

Required: No

permissions

The explicit permissions to provide to the container for the device. By default, the container has permissions for `read`, `write`, and `mknod` for the device.

Type: Array of strings

Valid Values: `read | write | mknod`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DockerVolumeConfiguration

This parameter is specified when you're using Docker volumes. Docker volumes are only supported when you're using the EC2 launch type. Windows containers only support the use of the `local` driver. To use bind mounts, specify a `host` instead.

Contents

autoprovision

If this value is `true`, the Docker volume is created if it doesn't already exist.

> **Note**
>
> This field is only used if the `scope` is `shared`.

Type: Boolean

Required: No

driver

The Docker volume driver to use. The driver value must match the driver name provided by Docker because it is used for task placement. If the driver was installed using the Docker plugin CLI, use `docker plugin ls` to retrieve the driver name from your container instance. If the driver was installed using another method, use Docker plugin discovery to retrieve the driver name. For more information, see Docker plugin discovery. This parameter maps to `Driver` in the Create a volume section of the Docker Remote API and the `xxdriver` option to docker volume create.

Type: String

Required: No

driverOpts

A map of Docker driver-specific options passed through. This parameter maps to `DriverOpts` in the Create a volume section of the Docker Remote API and the `xxopt` option to docker volume create.

Type: String to string map

Required: No

labels

Custom metadata to add to your Docker volume. This parameter maps to `Labels` in the Create a volume section of the Docker Remote API and the `xxlabel` option to docker volume create.

Type: String to string map

Required: No

scope

The scope for the Docker volume that determines its lifecycle. Docker volumes that are scoped to a `task` are automatically provisioned when the task starts and destroyed when the task stops. Docker volumes that are scoped as `shared` persist after the task stops.

Type: String

Valid Values: `task | shared`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EBSTagSpecification

The tag specifications of an Amazon EBS volume.

Contents

resourceType

The type of volume resource.

Type: String

Valid Values: `volume`

Required: Yes

propagateTags

Determines whether to propagate the tags from the task definition to the Amazon EBS volume. Tags can only propagate to a `SERVICE` specified in `ServiceVolumeConfiguration`. If no value is specified, the tags aren't propagated.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

tags

The tags applied to this Amazon EBS volume. `AmazonECSCreated` and `AmazonECSManaged` are reserved tags that can't be used.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EFSAuthorizationConfig

The authorization configuration details for the Amazon EFS file system.

Contents

accessPointId

The Amazon EFS access point ID to use. If an access point is specified, the root directory value specified in the `EFSVolumeConfiguration` must either be omitted or set to `/` which will enforce the path set on the EFS access point. If an access point is used, transit encryption must be on in the `EFSVolumeConfiguration`. For more information, see Working with Amazon EFS access points in the *Amazon Elastic File System User Guide*.

Type: String

Required: No

iam

Determines whether to use the Amazon ECS task role defined in a task definition when mounting the Amazon EFS file system. If it is turned on, transit encryption must be turned on in the `EFSVolumeConfiguration`. If this parameter is omitted, the default value of `DISABLED` is used. For more information, see Using Amazon EFS access points in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EFSVolumeConfiguration

This parameter is specified when you're using an Amazon Elastic File System file system for task storage. For more information, see Amazon EFS volumes in the *Amazon Elastic Container Service Developer Guide*.

Contents

fileSystemId

The Amazon EFS file system ID to use.

Type: String

Required: Yes

authorizationConfig

The authorization configuration details for the Amazon EFS file system.

Type: EFSAuthorizationConfig object

Required: No

rootDirectory

The directory within the Amazon EFS file system to mount as the root directory inside the host. If this parameter is omitted, the root of the Amazon EFS volume will be used. Specifying / will have the same effect as omitting this parameter.

> **Important**
>
> If an EFS access point is specified in the `authorizationConfig`, the root directory parameter must either be omitted or set to / which will enforce the path set on the EFS access point.

Type: String

Required: No

transitEncryption

Determines whether to use encryption for Amazon EFS data in transit between the Amazon ECS host and the Amazon EFS server. Transit encryption must be turned on if Amazon EFS IAM authorization is used. If this parameter is omitted, the default value of `DISABLED` is used. For more information, see Encrypting data in transit in the *Amazon Elastic File System User Guide*.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

transitEncryptionPort

The port to use when sending encrypted data between the Amazon ECS host and the Amazon EFS server. If you do not specify a transit encryption port, it will use the port selection strategy that the Amazon EFS mount helper uses. For more information, see EFS mount helper in the *Amazon Elastic File System User Guide*.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EnvironmentFile

A list of files containing the environment variables to pass to a container. You can specify up to ten environment files. The file must have a `.env` file extension. Each line in an environment file should contain an environment variable in `VARIABLE=VALUE` format. Lines beginning with # are treated as comments and are ignored.

If there are environment variables specified using the `environment` parameter in a container definition, they take precedence over the variables contained within an environment file. If multiple environment files are specified that contain the same variable, they're processed from the top down. We recommend that you use unique variable names. For more information, see Use a file to pass environment variables to a container in the *Amazon Elastic Container Service Developer Guide*.

Environment variable files are objects in Amazon S3 and all Amazon S3 security considerations apply.

You must use the following platforms for the Fargate launch type:

- Linux platform version `1.4.0` or later.
- Windows platform version `1.0.0` or later.

Consider the following when using the Fargate launch type:

- The file is handled like a native Docker env-file.
- There is no support for shell escape handling.
- The container entry point interperts the `VARIABLE` values.

Contents

type

The file type to use. Environment files are objects in Amazon S3. The only supported value is `s3`.

Type: String

Valid Values: `s3`

Required: Yes

value

The Amazon Resource Name (ARN) of the Amazon S3 object containing the environment variable file.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EphemeralStorage

The amount of ephemeral storage to allocate for the task. This parameter is used to expand the total amount of ephemeral storage available, beyond the default amount, for tasks hosted on AWS Fargate. For more information, see Using data volumes in tasks in the *Amazon ECS Developer Guide;*.

> **Note**
>
> For tasks using the Fargate launch type, the task requires the following platforms:
>
> - Linux platform version `1.4.0` or later.
> - Windows platform version `1.0.0` or later.

Contents

sizeInGiB

The total amount, in GiB, of ephemeral storage to set for the task. The minimum supported value is `20` GiB and the maximum supported value is `200` GiB.

Type: Integer

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ExecuteCommandConfiguration

The details of the execute command configuration.

Contents

kmsKeyId

Specify an AWS Key Management Service key ID to encrypt the data between the local client and the container.

Type: String

Required: No

logConfiguration

The log configuration for the results of the execute command actions. The logs can be sent to CloudWatch Logs or an Amazon S3 bucket. When `logging=OVERRIDE` is specified, a `logConfiguration` must be provided.

Type: ExecuteCommandLogConfiguration object

Required: No

logging

The log setting to use for redirecting logs for your execute command results. The following log settings are available.

- `NONE`: The execute command session is not logged.
- `DEFAULT`: The `awslogs` configuration in the task definition is used. If no logging parameter is specified, it defaults to this value. If no `awslogs` log driver is configured in the task definition, the output won't be logged.
- `OVERRIDE`: Specify the logging details as a part of `logConfiguration`. If the `OVERRIDE` logging option is specified, the `logConfiguration` is required.

Type: String

Valid Values: `NONE | DEFAULT | OVERRIDE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ExecuteCommandLogConfiguration

The log configuration for the results of the execute command actions. The logs can be sent to CloudWatch Logs or an Amazon S3 bucket.

Contents

cloudWatchEncryptionEnabled

Determines whether to use encryption on the CloudWatch logs. If not specified, encryption will be off.

Type: Boolean

Required: No

cloudWatchLogGroupName

The name of the CloudWatch log group to send logs to.

> **Note**
> The CloudWatch log group must already be created.

Type: String

Required: No

s3BucketName

The name of the S3 bucket to send logs to.

> **Note**
> The S3 bucket must already be created.

Type: String

Required: No

s3EncryptionEnabled

Determines whether to use encryption on the S3 logs. If not specified, encryption is not used.

Type: Boolean

Required: No

s3KeyPrefix

An optional folder in the S3 bucket to place logs in.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Failure

A failed resource. For a list of common causes, see API failure reasons in the *Amazon Elastic Container Service Developer Guide*.

Contents

arn

The Amazon Resource Name (ARN) of the failed resource.

Type: String

Required: No

detail

The details of the failure.

Type: String

Required: No

reason

The reason for the failure.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FirelensConfiguration

The FireLens configuration for the container. This is used to specify and configure a log router for container logs. For more information, see Custom log routing in the *Amazon Elastic Container Service Developer Guide*.

Contents

type

The log router to use. The valid values are `fluentd` or `fluentbit`.

Type: String

Valid Values: `fluentd | fluentbit`

Required: Yes

options

The options to use when configuring the log router. This field is optional and can be used to specify a custom configuration file or to add additional metadata, such as the task, task definition, cluster, and container instance details to the log event. If specified, the syntax to use is `"options":{"enable-ecs-log-metadata":"true|false","config-file-type:"s3|file","config-file-value":"arn:aws:s3:::mybucket/fluent.conf|filepath"}`. For more information, see Creating a task definition that uses a FireLens configuration in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> Tasks hosted on AWS Fargate only support the `file` configuration file type.

Type: String to string map

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FSxWindowsFileServerAuthorizationConfig

The authorization configuration details for Amazon FSx for Windows File Server file system. See FSxWindowsFileServerVolumeConfiguration in the *Amazon ECS API Reference*.

For more information and the input format, see Amazon FSx for Windows File Server Volumes in the *Amazon Elastic Container Service Developer Guide*.

Contents

credentialsParameter

The authorization credential option to use. The authorization credential options can be provided using either the Amazon Resource Name (ARN) of an AWS Secrets Manager secret or SSM Parameter Store parameter. The ARN refers to the stored credentials.

Type: String

Required: Yes

domain

A fully qualified domain name hosted by an AWS Directory Service Managed Microsoft AD (Active Directory) or self-hosted AD on Amazon EC2.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FSxWindowsFileServerVolumeConfiguration

This parameter is specified when you're using Amazon FSx for Windows File Server file system for task storage.

For more information and the input format, see Amazon FSx for Windows File Server volumes in the *Amazon Elastic Container Service Developer Guide*.

Contents

authorizationConfig

The authorization configuration details for the Amazon FSx for Windows File Server file system.

Type: FSxWindowsFileServerAuthorizationConfig object

Required: Yes

fileSystemId

The Amazon FSx for Windows File Server file system ID to use.

Type: String

Required: Yes

rootDirectory

The directory within the Amazon FSx for Windows File Server file system to mount as the root directory inside the host.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

HealthCheck

An object representing a container health check. Health check parameters that are specified in a container definition override any Docker health checks that exist in the container image (such as those specified in a parent image or from the image's Dockerfile). This configuration maps to the `HEALTHCHECK` parameter of docker run.

> **Note**
>
> The Amazon ECS container agent only monitors and reports on the health checks specified in the task definition. Amazon ECS does not monitor Docker health checks that are embedded in a container image and not specified in the container definition. Health check parameters that are specified in a container definition override any Docker health checks that exist in the container image.

You can view the health status of both individual containers and a task with the DescribeTasks API operation or when viewing the task details in the console.

The health check is designed to make sure that your containers survive agent restarts, upgrades, or temporary unavailability.

Amazon ECS performs health checks on containers with the default that launched the container instance or the task.

The following describes the possible `healthStatus` values for a container:

- `HEALTHY`-The container health check has passed successfully.
- `UNHEALTHY`-The container health check has failed.
- `UNKNOWN`-The container health check is being evaluated, there's no container health check defined, or Amazon ECS doesn't have the health status of the container.

The following describes the possible `healthStatus` values based on the container health checker status of essential containers in the task with the following priority order (high to low):

- `UNHEALTHY`-One or more essential containers have failed their health check.
- `UNKNOWN`-Any essential container running within the task is in an `UNKNOWN` state and no other essential containers have an `UNHEALTHY` state.

- HEALTHY-All essential containers within the task have passed their health checks.

Consider the following task health example with 2 containers.

- If Container1 is UNHEALTHY and Container2 is UNKNOWN, the task health is UNHEALTHY.
- If Container1 is UNHEALTHY and Container2 is HEALTHY, the task health is UNHEALTHY.
- If Container1 is HEALTHY and Container2 is UNKNOWN, the task health is UNKNOWN.
- If Container1 is HEALTHY and Container2 is HEALTHY, the task health is HEALTHY.

Consider the following task health example with 3 containers.

- If Container1 is UNHEALTHY and Container2 is UNKNOWN, and Container3 is UNKNOWN, the task health is UNHEALTHY.
- If Container1 is UNHEALTHY and Container2 is UNKNOWN, and Container3 is HEALTHY, the task health is UNHEALTHY.
- If Container1 is UNHEALTHY and Container2 is HEALTHY, and Container3 is HEALTHY, the task health is UNHEALTHY.
- If Container1 is HEALTHY and Container2 is UNKNOWN, and Container3 is HEALTHY, the task health is UNKNOWN.
- If Container1 is HEALTHY and Container2 is UNKNOWN, and Container3 is UNKNOWN, the task health is UNKNOWN.
- If Container1 is HEALTHY and Container2 is HEALTHY, and Container3 is HEALTHY, the task health is HEALTHY.

If a task is run manually, and not as part of a service, the task will continue its lifecycle regardless of its health status. For tasks that are part of a service, if the task reports as unhealthy then the task will be stopped and the service scheduler will replace it.

The following are notes about container health check support:

- If the Amazon ECS container agent becomes disconnected from the Amazon ECS service, this won't cause a container to transition to an UNHEALTHY status. This is by design, to ensure that containers remain running during agent restarts or temporary unavailability. The health check status is the "last heard from" response from the Amazon ECS agent, so if the container was considered HEALTHY prior to the disconnect, that status will remain until the agent reconnects

and another health check occurs. There are no assumptions made about the status of the container health checks.

- Container health checks require version `1.17.0` or greater of the Amazon ECS container agent. For more information, see Updating the Amazon ECS container agent.
- Container health checks are supported for Fargate tasks if you're using platform version `1.1.0` or greater. For more information, see AWS Fargate platform versions.
- Container health checks aren't supported for tasks that are part of a service that's configured to use a Classic Load Balancer.

Contents

command

A string array representing the command that the container runs to determine if it is healthy. The string array must start with `CMD` to run the command arguments directly, or `CMD-SHELL` to run the command with the container's default shell.

When you use the AWS Management Console JSON panel, the AWS Command Line Interface, or the APIs, enclose the list of commands in double quotes and brackets.

```
[ "CMD-SHELL", "curl -f http://localhost/ || exit 1" ]
```

You don't include the double quotes and brackets when you use the AWS Management Console.

```
CMD-SHELL, curl -f http://localhost/ || exit 1
```

An exit code of 0 indicates success, and non-zero exit code indicates failure. For more information, see `HealthCheck` in the Create a container section of the Docker Remote API.

Type: Array of strings

Required: Yes

interval

The time period in seconds between each health check execution. You may specify between 5 and 300 seconds. The default value is 30 seconds.

Type: Integer

Required: No

retries

The number of times to retry a failed health check before the container is considered unhealthy. You may specify between 1 and 10 retries. The default value is 3.

Type: Integer

Required: No

startPeriod

The optional grace period to provide containers time to bootstrap before failed health checks count towards the maximum number of retries. You can specify between 0 and 300 seconds. By default, the `startPeriod` is off.

> **Note**
>
> If a health check succeeds within the `startPeriod`, then the container is considered healthy and any subsequent failures count toward the maximum number of retries.

Type: Integer

Required: No

timeout

The time period in seconds to wait for a health check to succeed before it is considered a failure. You may specify between 2 and 60 seconds. The default value is 5.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

HostEntry

Hostnames and IP address entries that are added to the `/etc/hosts` file of a container via the `extraHosts` parameter of its ContainerDefinition.

Contents

hostname

The hostname to use in the `/etc/hosts` entry.

Type: String

Required: Yes

ipAddress

The IP address to use in the `/etc/hosts` entry.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HostVolumeProperties

Details on a container instance bind mount host volume.

Contents

sourcePath

When the `host` parameter is used, specify a `sourcePath` to declare the path on the host container instance that's presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you. If the `host` parameter contains a `sourcePath` file location, then the data volume persists at the specified location on the host container instance until you delete it manually. If the `sourcePath` value doesn't exist on the host container instance, the Docker daemon creates it. If the location does exist, the contents of the source path folder are exported.

If you're using the Fargate launch type, the `sourcePath` parameter is not supported.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InferenceAccelerator

Details on an Elastic Inference accelerator. For more information, see Working with Amazon Elastic Inference on Amazon ECS in the *Amazon Elastic Container Service Developer Guide*.

Contents

deviceName

The Elastic Inference accelerator device name. The `deviceName` must also be referenced in a container definition as a ResourceRequirement.

Type: String

Required: Yes

deviceType

The Elastic Inference accelerator type to use.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InferenceAcceleratorOverride

Details on an Elastic Inference accelerator task override. This parameter is used to override the Elastic Inference accelerator specified in the task definition. For more information, see Working with Amazon Elastic Inference on Amazon ECS in the *Amazon Elastic Container Service Developer Guide*.

Contents

deviceName

The Elastic Inference accelerator device name to override for the task. This parameter must match a `deviceName` specified in the task definition.

Type: String

Required: No

deviceType

The Elastic Inference accelerator type to use.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InstanceHealthCheckResult

An object representing the result of a container instance health status check.

Contents

lastStatusChange

The Unix timestamp for when the container instance health status last changed.

Type: Timestamp

Required: No

lastUpdated

The Unix timestamp for when the container instance health status was last updated.

Type: Timestamp

Required: No

status

The container instance health status.

Type: String

Valid Values: `OK | IMPAIRED | INSUFFICIENT_DATA | INITIALIZING`

Required: No

type

The type of container instance health status that was verified.

Type: String

Valid Values: `CONTAINER_RUNTIME`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KernelCapabilities

The Linux capabilities to add or remove from the default Docker configuration for a container defined in the task definition. For more information about the default capabilities and the non-default available capabilities, see Runtime privilege and Linux capabilities in the *Docker run reference*. For more detailed information about these Linux capabilities, see the capabilities(7) Linux manual page.

Contents

add

The Linux capabilities for the container that have been added to the default configuration provided by Docker. This parameter maps to CapAdd in the Create a container section of the Docker Remote API and the `--cap-add` option to docker run.

> **Note**
>
> Tasks launched on AWS Fargate only support adding the SYS_PTRACE kernel capability.

Valid values: `"ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"`

Type: Array of strings

Required: No

drop

The Linux capabilities for the container that have been removed from the default configuration provided by Docker. This parameter maps to CapDrop in the Create a container section of the Docker Remote API and the `--cap-drop` option to docker run.

Valid values: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KeyValuePair

A key-value pair object.

Contents

name

The name of the key-value pair. For environment variables, this is the name of the environment variable.

Type: String

Required: No

value

The value of the key-value pair. For environment variables, this is the value of the environment variable.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LinuxParameters

The Linux-specific options that are applied to the container, such as Linux KernelCapabilities.

Contents

capabilities

The Linux capabilities for the container that are added to or dropped from the default configuration provided by Docker.

> **Note**
>
> For tasks that use the Fargate launch type, `capabilities` is supported for all platform versions but the `add` parameter is only supported if using platform version 1.4.0 or later.

Type: KernelCapabilities object

Required: No

devices

Any host devices to expose to the container. This parameter maps to `Devices` in the Create a container section of the Docker Remote API and the `--device` option to docker run.

> **Note**
>
> If you're using tasks that use the Fargate launch type, the `devices` parameter isn't supported.

Type: Array of Device objects

Required: No

initProcessEnabled

Run an `init` process inside the container that forwards signals and reaps processes. This parameter maps to the `--init` option to docker run. This parameter requires version 1.25

of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version --format '{{.Server.APIVersion}}'`

Type: Boolean

Required: No

maxSwap

The total amount of swap memory (in MiB) a container can use. This parameter will be translated to the `--memory-swap` option to docker run where the value would be the sum of the container memory plus the `maxSwap` value.

If a `maxSwap` value of `0` is specified, the container will not use swap. Accepted values are `0` or any positive integer. If the `maxSwap` parameter is omitted, the container will use the swap configuration for the container instance it is running on. A `maxSwap` value must be set for the `swappiness` parameter to be used.

> **Note**
>
> If you're using tasks that use the Fargate launch type, the `maxSwap` parameter isn't supported.
> If you're using tasks on Amazon Linux 2023 the `swappiness` parameter isn't supported.

Type: Integer

Required: No

sharedMemorySize

The value for the size (in MiB) of the `/dev/shm` volume. This parameter maps to the `--shm-size` option to docker run.

> **Note**
>
> If you are using tasks that use the Fargate launch type, the `sharedMemorySize` parameter is not supported.

Type: Integer

Required: No

swappiness

This allows you to tune a container's memory swappiness behavior. A `swappiness` value of `0` will cause swapping to not happen unless absolutely necessary. A `swappiness` value of `100` will cause pages to be swapped very aggressively. Accepted values are whole numbers between `0` and `100`. If the `swappiness` parameter is not specified, a default value of `60` is used. If a value is not specified for `maxSwap` then this parameter is ignored. This parameter maps to the `--memory-swappiness` option to docker run.

> **Note**
>
> If you're using tasks that use the Fargate launch type, the `swappiness` parameter isn't supported.
> If you're using tasks on Amazon Linux 2023 the `swappiness` parameter isn't supported.

Type: Integer

Required: No

tmpfs

The container path, mount options, and size (in MiB) of the tmpfs mount. This parameter maps to the `--tmpfs` option to docker run.

> **Note**
>
> If you're using tasks that use the Fargate launch type, the `tmpfs` parameter isn't supported.

Type: Array of Tmpfs objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LoadBalancer

The load balancer configuration to use with a service or task set.

When you add, update, or remove a load balancer configuration, Amazon ECS starts a new deployment with the updated Elastic Load Balancing configuration. This causes tasks to register to and deregister from load balancers.

We recommend that you verify this on a test environment before you update the Elastic Load Balancing configuration.

A service-linked role is required for services that use multiple target groups. For more information, see Using service-linked roles in the *Amazon Elastic Container Service Developer Guide*.

Contents

containerName

The name of the container (as it appears in a container definition) to associate with the load balancer.

You need to specify the container name when configuring the target group for an Amazon ECS load balancer.

Type: String

Required: No

containerPort

The port on the container to associate with the load balancer. This port must correspond to a `containerPort` in the task definition the tasks in the service are using. For tasks that use the EC2 launch type, the container instance they're launched on must allow ingress traffic on the `hostPort` of the port mapping.

Type: Integer

Required: No

loadBalancerName

The name of the load balancer to associate with the Amazon ECS service or task set.

If you are using an Application Load Balancer or a Network Load Balancer the load balancer name parameter should be omitted.

Type: String

Required: No

targetGroupArn

The full Amazon Resource Name (ARN) of the Elastic Load Balancing target group or groups associated with a service or task set.

A target group ARN is only specified when using an Application Load Balancer or Network Load Balancer.

For services using the ECS deployment controller, you can specify one or multiple target groups. For more information, see Registering multiple target groups with a service in the *Amazon Elastic Container Service Developer Guide*.

For services using the `CODE_DEPLOY` deployment controller, you're required to define two target groups for the load balancer. For more information, see Blue/green deployment with CodeDeploy in the *Amazon Elastic Container Service Developer Guide*.

> **Important**
>
> If your service's task definition uses the `awsvpc` network mode, you must choose `ip` as the target type, not `instance`. Do this when creating your target groups because tasks that use the `awsvpc` network mode are associated with an elastic network interface, not an Amazon EC2 instance. This network mode is required for the Fargate launch type.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

LogConfiguration

The log configuration for the container. This parameter maps to `LogConfig` in the Create a container section of the Docker Remote API and the `--log-driver` option to `docker run`.

By default, containers use the same logging driver that the Docker daemon uses. However, the container might use a different logging driver than the Docker daemon by specifying a log driver configuration in the container definition. For more information about the options for different supported log drivers, see Configure logging drivers in the Docker documentation.

Understand the following when specifying a log configuration for your containers.

- Amazon ECS currently supports a subset of the logging drivers available to the Docker daemon. Additional log drivers may be available in future releases of the Amazon ECS container agent.

  For tasks on AWS Fargate, the supported log drivers are `awslogs`, `splunk`, and `awsfirelens`.

  For tasks hosted on Amazon EC2 instances, the supported log drivers are `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`,`syslog`, `splunk`, and `awsfirelens`.
- This parameter requires version 1.18 of the Docker Remote API or greater on your container instance.
- For tasks that are hosted on Amazon EC2 instances, the Amazon ECS container agent must register the available logging drivers with the `ECS_AVAILABLE_LOGGING_DRIVERS` environment variable before containers placed on that instance can use these log configuration options. For more information, see Amazon ECS container agent configuration in the *Amazon Elastic Container Service Developer Guide*.
- For tasks that are on AWS Fargate, because you don't have access to the underlying infrastructure your tasks are hosted on, any additional software needed must be installed outside of the task. For example, the Fluentd output aggregators or a remote host running Logstash to send Gelf logs to.

Contents

logDriver

The log driver to use for the container.

For tasks on AWS Fargate, the supported log drivers are `awslogs`, `splunk`, and `awsfirelens`.

For tasks hosted on Amazon EC2 instances, the supported log drivers are `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`,`syslog`, `splunk`, and `awsfirelens`.

For more information about using the `awslogs` log driver, see Using the awslogs log driver in the *Amazon Elastic Container Service Developer Guide*.

For more information about using the `awsfirelens` log driver, see Custom log routing in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> If you have a custom driver that isn't listed, you can fork the Amazon ECS container agent project that's available on GitHub and customize it to work with that driver. We encourage you to submit pull requests for changes that you would like to have included. However, we don't currently provide support for running modified copies of this software.

Type: String

Valid Values: `json-file | syslog | journald | gelf | fluentd | awslogs | splunk | awsfirelens`

Required: Yes

options

The configuration options to send to the log driver. This parameter requires version 1.19 of the Docker Remote API or greater on your container instance. To check the Docker Remote API version on your container instance, log in to your container instance and run the following command: `sudo docker version --format '{{.Server.APIVersion}}'`

Type: String to string map

Required: No

secretOptions

The secrets to pass to the log configuration. For more information, see Specifying sensitive data in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of Secret objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ManagedAgent

Details about the managed agent status for the container.

Contents

lastStartedAt

The Unix timestamp for the time when the managed agent was last started.

Type: Timestamp

Required: No

lastStatus

The last known status of the managed agent.

Type: String

Required: No

name

The name of the managed agent. When the execute command feature is turned on, the managed agent name is `ExecuteCommandAgent`.

Type: String

Valid Values: `ExecuteCommandAgent`

Required: No

reason

The reason for why the managed agent is in the state it is in.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ManagedAgentStateChange

An object representing a change in state for a managed agent.

Contents

containerName

The name of the container that's associated with the managed agent.

Type: String

Required: Yes

managedAgentName

The name of the managed agent.

Type: String

Valid Values: `ExecuteCommandAgent`

Required: Yes

status

The status of the managed agent.

Type: String

Required: Yes

reason

The reason for the status of the managed agent.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ManagedScaling

The managed scaling settings for the Auto Scaling group capacity provider.

When managed scaling is turned on, Amazon ECS manages the scale-in and scale-out actions of the Auto Scaling group. Amazon ECS manages a target tracking scaling policy using an Amazon ECS managed CloudWatch metric with the specified `targetCapacity` value as the target value for the metric. For more information, see Using managed scaling in the *Amazon Elastic Container Service Developer Guide*.

If managed scaling is off, the user must manage the scaling of the Auto Scaling group.

Contents

instanceWarmupPeriod

The period of time, in seconds, after a newly launched Amazon EC2 instance can contribute to CloudWatch metrics for Auto Scaling group. If this parameter is omitted, the default value of `300` seconds is used.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 10000.

Required: No

maximumScalingStepSize

The maximum number of Amazon EC2 instances that Amazon ECS will scale out at one time. The scale in process is not affected by this parameter. If this parameter is omitted, the default value of `10000` is used.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

minimumScalingStepSize

The minimum number of Amazon EC2 instances that Amazon ECS will scale out at one time. The scale in process is not affected by this parameter If this parameter is omitted, the default value of 1 is used.

When additional capacity is required, Amazon ECS will scale up the minimum scaling step size even if the actual demand is less than the minimum scaling step size.

If you use a capacity provider with an Auto Scaling group configured with more than one Amazon EC2 instance type or Availability Zone, Amazon ECS will scale up by the exact minimum scaling step size value and will ignore both the maximum scaling step size as well as the capacity demand.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10000.

Required: No

status

Determines whether to use managed scaling for the capacity provider.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

targetCapacity

The target capacity utilization as a percentage for the capacity provider. The specified value must be greater than `0` and less than or equal to `100`. For example, if you want the capacity provider to maintain 10% spare capacity, then that means the utilization is 90%, so use a `targetCapacity` of `90`. The default value of `100` percent results in the Amazon EC2 instances in your Auto Scaling group being completely used.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MountPoint

The details for a volume mount point that's used in a container definition.

Contents

containerPath

The path on the container to mount the host volume at.

Type: String

Required: No

readOnly

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

Type: Boolean

Required: No

sourceVolume

The name of the volume to mount. Must be a volume name referenced in the `name` parameter of task definition `volume`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NetworkBinding

Details on the network bindings between a container and its host container instance. After a task reaches the `RUNNING` status, manual and automatic host and container port assignments are visible in the `networkBindings` section of DescribeTasks API responses.

Contents

bindIP

The IP address that the container is bound to on the container instance.

Type: String

Required: No

containerPort

The port number on the container that's used with the network binding.

Type: Integer

Required: No

containerPortRange

The port number range on the container that's bound to the dynamically mapped host port range.

The following rules apply when you specify a `containerPortRange`:

- You must use either the `bridge` network mode or the `awsvpc` network mode.
- This parameter is available for both the EC2 and AWS Fargate launch types.
- This parameter is available for both the Linux and Windows operating systems.
- The container instance must have at least version 1.67.0 of the container agent and at least version 1.67.0-1 of the `ecs-init` package
- You can specify a maximum of 100 port ranges per container.
- You do not specify a `hostPortRange`. The value of the `hostPortRange` is set as follows:
  - For containers in a task with the `awsvpc` network mode, the `hostPortRange` is set to the same value as the `containerPortRange`. This is a static mapping strategy.

- For containers in a task with the `bridge` network mode, the Amazon ECS agent finds open host ports from the default ephemeral range and passes it to docker to bind them to the container ports.

- The `containerPortRange` valid values are between 1 and 65535.
- A port can only be included in one port mapping per container.
- You cannot specify overlapping port ranges.
- The first port in the range must be less than last port in the range.
- Docker recommends that you turn off the docker-proxy in the Docker daemon config file when you have a large number of ports.

  For more information, see Issue #11185 on the Github website.

  For information about how to turn off the docker-proxy in the Docker daemon config file, see Docker daemon in the *Amazon ECS Developer Guide*.

You can call `DescribeTasks` to view the `hostPortRange` which are the host ports that are bound to the container ports.

Type: String

Required: No

hostPort

The port number on the host that's used with the network binding.

Type: Integer

Required: No

hostPortRange

The port number range on the host that's used with the network binding. This is assigned is assigned by Docker and delivered by the Amazon ECS agent.

Type: String

Required: No

protocol

The protocol used for the network binding.

Type: String

Valid Values: `tcp | udp`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NetworkConfiguration

The network configuration for a task or service.

Contents

awsvpcConfiguration

The VPC subnets and security groups that are associated with a task.

> **Note**
>
> All specified subnets and security groups must be from the same VPC.

Type: AwsVpcConfiguration object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NetworkInterface

An object representing the elastic network interface for tasks that use the `awsvpc` network mode.

Contents

attachmentId

The attachment ID for the network interface.

Type: String

Required: No

ipv6Address

The private IPv6 address for the network interface.

Type: String

Required: No

privateIpv4Address

The private IPv4 address for the network interface.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PlacementConstraint

An object representing a constraint on task placement. For more information, see Task placement constraints in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> If you're using the Fargate launch type, task placement constraints aren't supported.

Contents

expression

A cluster query language expression to apply to the constraint. The expression can have a maximum length of 2000 characters. You can't specify an expression if the constraint type is `distinctInstance`. For more information, see Cluster query language in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

type

The type of constraint. Use `distinctInstance` to ensure that each task in a particular group is running on a different container instance. Use `memberOf` to restrict the selection to a group of valid candidates.

Type: String

Valid Values: `distinctInstance | memberOf`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

PlacementStrategy

The task placement strategy for a task or service. For more information, see Task placement strategies in the *Amazon Elastic Container Service Developer Guide*.

Contents

field

The field to apply the placement strategy against. For the `spread` placement strategy, valid values are `instanceId` (or `host`, which has the same effect), or any platform or custom attribute that's applied to a container instance, such as `attribute:ecs.availability-zone`. For the `binpack` placement strategy, valid values are `cpu` and `memory`. For the `random` placement strategy, this field is not used.

Type: String

Required: No

type

The type of placement strategy. The `random` placement strategy randomly places tasks on available candidates. The `spread` placement strategy spreads placement across available candidates evenly based on the `field` parameter. The `binpack` strategy places tasks on available candidates that have the least available amount of the resource that's specified with the `field` parameter. For example, if you binpack on memory, a task is placed on the instance with the least amount of remaining memory but still enough to run the task.

Type: String

Valid Values: `random | spread | binpack`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

PlatformDevice

The devices that are available on the container instance. The only supported device type is a GPU.

Contents

id

The ID for the GPUs on the container instance. The available GPU IDs can also be obtained on the container instance in the `/var/lib/ecs/gpu/nvidia_gpu_info.json` file.

Type: String

Required: Yes

type

The type of device that's available on the container instance. The only supported value is `GPU`.

Type: String

Valid Values: `GPU`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PortMapping

Port mappings allow containers to access ports on the host container instance to send or receive traffic. Port mappings are specified as part of the container definition.

If you use containers in a task with the `awsvpc` or `host` network mode, specify the exposed ports using `containerPort`. The `hostPort` can be left blank or it must be the same value as the `containerPort`.

Most fields of this parameter (`containerPort`, `hostPort`, `protocol`) maps to `PortBindings` in the Create a container section of the Docker Remote API and the `--publish` option to `docker run`. If the network mode of a task definition is set to `host`, host ports must either be undefined or match the container port in the port mapping.

> **Note**
>
> You can't expose the same container port for multiple protocols. If you attempt this, an error is returned.

After a task reaches the `RUNNING` status, manual and automatic host and container port assignments are visible in the `networkBindings` section of DescribeTasks API responses.

Contents

appProtocol

The application protocol that's used for the port mapping. This parameter only applies to Service Connect. We recommend that you set this parameter to be consistent with the protocol that your application uses. If you set this parameter, Amazon ECS adds protocol-specific connection handling to the Service Connect proxy. If you set this parameter, Amazon ECS adds protocol-specific telemetry in the Amazon ECS console and CloudWatch.

If you don't set a value for this parameter, then TCP is used. However, Amazon ECS doesn't add protocol-specific telemetry for TCP.

`appProtocol` is immutable in a Service Connect service. Updating this field requires a service deletion and redeployment.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Valid Values: `http | http2 | grpc`

Required: No

containerPort

The port number on the container that's bound to the user-specified or automatically assigned host port.

If you use containers in a task with the `awsvpc` or `host` network mode, specify the exposed ports using `containerPort`.

If you use containers in a task with the `bridge` network mode and you specify a container port and not a host port, your container automatically receives a host port in the ephemeral port range. For more information, see `hostPort`. Port mappings that are automatically assigned in this way do not count toward the 100 reserved ports limit of a container instance.

Type: Integer

Required: No

containerPortRange

The port number range on the container that's bound to the dynamically mapped host port range.

The following rules apply when you specify a `containerPortRange`:

- You must use either the `bridge` network mode or the `awsvpc` network mode.
- This parameter is available for both the EC2 and AWS Fargate launch types.
- This parameter is available for both the Linux and Windows operating systems.
- The container instance must have at least version 1.67.0 of the container agent and at least version 1.67.0-1 of the `ecs-init` package

- You can specify a maximum of 100 port ranges per container.
- You do not specify a `hostPortRange`. The value of the `hostPortRange` is set as follows:
  - For containers in a task with the `awsvpc` network mode, the `hostPortRange` is set to the same value as the `containerPortRange`. This is a static mapping strategy.
  - For containers in a task with the `bridge` network mode, the Amazon ECS agent finds open host ports from the default ephemeral range and passes it to docker to bind them to the container ports.
- The `containerPortRange` valid values are between 1 and 65535.
- A port can only be included in one port mapping per container.
- You cannot specify overlapping port ranges.
- The first port in the range must be less than last port in the range.
- Docker recommends that you turn off the docker-proxy in the Docker daemon config file when you have a large number of ports.

  For more information, see Issue #11185 on the Github website.

  For information about how to turn off the docker-proxy in the Docker daemon config file, see Docker daemon in the *Amazon ECS Developer Guide*.

You can call `DescribeTasks` to view the `hostPortRange` which are the host ports that are bound to the container ports.

Type: String

Required: No

hostPort

The port number on the container instance to reserve for your container.

If you specify a `containerPortRange`, leave this field empty and the value of the `hostPort` is set as follows:

- For containers in a task with the `awsvpc` network mode, the `hostPort` is set to the same value as the `containerPort`. This is a static mapping strategy.
- For containers in a task with the `bridge` network mode, the Amazon ECS agent finds open ports on the host and automatically binds them to the container ports. This is a dynamic mapping strategy.

If you use containers in a task with the `awsvpc` or `host` network mode, the `hostPort` can either be left blank or set to the same value as the `containerPort`.

If you use containers in a task with the `bridge` network mode, you can specify a non-reserved host port for your container port mapping, or you can omit the `hostPort` (or set it to `0`) while specifying a `containerPort` and your container automatically receives a port in the ephemeral port range for your container instance operating system and Docker version.

The default ephemeral port range for Docker version 1.6.0 and later is listed on the instance under `/proc/sys/net/ipv4/ip_local_port_range`. If this kernel parameter is unavailable, the default ephemeral port range from 49153 through 65535 (Linux) or 49152 through 65535 (Windows) is used. Do not attempt to specify a host port in the ephemeral port range as these are reserved for automatic assignment. In general, ports below 32768 are outside of the ephemeral port range.

The default reserved ports are 22 for SSH, the Docker ports 2375 and 2376, and the Amazon ECS container agent ports 51678-51680. Any host port that was previously specified in a running task is also reserved while the task is running. That is, after a task stops, the host port is released. The current reserved ports are displayed in the `remainingResources` of DescribeContainerInstances output. A container instance can have up to 100 reserved ports at a time. This number includes the default reserved ports. Automatically assigned ports aren't included in the 100 reserved ports quota.

Type: Integer

Required: No

name

The name that's used for the port mapping. This parameter only applies to Service Connect. This parameter is the name that you use in the `serviceConnectConfiguration` of a service. The name can include up to 64 characters. The characters can include lowercase letters, numbers, underscores (_), and hyphens (-). The name can't start with a hyphen.

For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

protocol

The protocol used for the port mapping. Valid values are `tcp` and `udp`. The default is `tcp`. `protocol` is immutable in a Service Connect service. Updating this field requires a service deletion and redeployment.

Type: String

Valid Values: `tcp | udp`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProtectedTask

An object representing the protection status details for a task. You can set the protection status with the UpdateTaskProtection API and get the status of tasks with the GetTaskProtection API.

Contents

expirationDate

The epoch time when protection for the task will expire.

Type: Timestamp

Required: No

protectionEnabled

The protection status of the task. If scale-in protection is on for a task, the value is `true`. Otherwise, it is `false`.

Type: Boolean

Required: No

taskArn

The task ARN.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProxyConfiguration

The configuration details for the App Mesh proxy.

For tasks that use the EC2 launch type, the container instances require at least version 1.26.0 of the container agent and at least version 1.26.0-1 of the `ecs-init` package to use a proxy configuration. If your container instances are launched from the Amazon ECS optimized AMI version `20190301` or later, then they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI

Contents

containerName

The name of the container that will serve as the App Mesh proxy.

Type: String

Required: Yes

properties

The set of network configuration parameters to provide the Container Network Interface (CNI) plugin, specified as key-value pairs.

- `IgnoredUID` - (Required) The user ID (UID) of the proxy container as defined by the `user` parameter in a container definition. This is used to ensure the proxy ignores its own traffic. If `IgnoredGID` is specified, this field can be empty.
- `IgnoredGID` - (Required) The group ID (GID) of the proxy container as defined by the `user` parameter in a container definition. This is used to ensure the proxy ignores its own traffic. If `IgnoredUID` is specified, this field can be empty.
- `AppPorts` - (Required) The list of ports that the application uses. Network traffic to these ports is forwarded to the `ProxyIngressPort` and `ProxyEgressPort`.
- `ProxyIngressPort` - (Required) Specifies the port that incoming traffic to the `AppPorts` is directed to.
- `ProxyEgressPort` - (Required) Specifies the port that outgoing traffic from the `AppPorts` is directed to.
- `EgressIgnoredPorts` - (Required) The egress traffic going to the specified ports is ignored and not redirected to the `ProxyEgressPort`. It can be an empty list.

- `EgressIgnoredIPs` - (Required) The egress traffic going to the specified IP addresses is ignored and not redirected to the `ProxyEgressPort`. It can be an empty list.

Type: Array of KeyValuePair objects

Required: No

type

The proxy type. The only supported value is `APPMESH`.

Type: String

Valid Values: `APPMESH`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RepositoryCredentials

The repository credentials for private registry authentication.

Contents

credentialsParameter

The Amazon Resource Name (ARN) of the secret containing the private repository credentials.

> **Note**
>
> When you use the Amazon ECS API, AWS CLI, or AWS SDK, if the secret exists in the same Region as the task that you're launching then you can use either the full ARN or the name of the secret. When you use the AWS Management Console, you must specify the full ARN of the secret.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Resource

Describes the resources available for a container instance.

Contents

doubleValue

When the `doubleValue` type is set, the value of the resource must be a double precision floating-point type.

Type: Double

Required: No

integerValue

When the `integerValue` type is set, the value of the resource must be an integer.

Type: Integer

Required: No

longValue

When the `longValue` type is set, the value of the resource must be an extended precision floating-point type.

Type: Long

Required: No

name

The name of the resource, such as `CPU`, `MEMORY`, `PORTS`, `PORTS_UDP`, or a user-defined resource.

Type: String

Required: No

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. Valid values: `INTEGER`, `DOUBLE`, `LONG`, or `STRINGSET`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourceRequirement

The type and amount of a resource to assign to a container. The supported resource types are GPUs and Elastic Inference accelerators. For more information, see Working with GPUs on Amazon ECS or Working with Amazon Elastic Inference on Amazon ECS in the *Amazon Elastic Container Service Developer Guide*

Contents

type

The type of resource to assign to a container. The supported values are `GPU` or `InferenceAccelerator`.

Type: String

Valid Values: `GPU | InferenceAccelerator`

Required: Yes

value

The value for the specified resource type.

If the `GPU` type is used, the value is the number of physical `GPUs` the Amazon ECS container agent reserves for the container. The number of GPUs that's reserved for all containers in a task can't exceed the number of available GPUs on the container instance that the task is launched on.

If the `InferenceAccelerator` type is used, the `value` matches the `deviceName` for an InferenceAccelerator specified in a task definition.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RuntimePlatform

Information about the platform for the Amazon ECS service or task.

For more information about `RuntimePlatform`, see RuntimePlatform in the *Amazon Elastic Container Service Developer Guide*.

Contents

cpuArchitecture

The CPU architecture.

You can run your Linux tasks on an ARM-based platform by setting the value to `ARM64`. This option is available for tasks that run on Linux Amazon EC2 instance or Linux containers on Fargate.

Type: String

Valid Values: `X86_64 | ARM64`

Required: No

operatingSystemFamily

The operating system.

Type: String

Valid Values: `WINDOWS_SERVER_2019_FULL | WINDOWS_SERVER_2019_CORE | WINDOWS_SERVER_2016_FULL | WINDOWS_SERVER_2004_CORE | WINDOWS_SERVER_2022_CORE | WINDOWS_SERVER_2022_FULL | WINDOWS_SERVER_20H2_CORE | LINUX`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Scale

A floating-point percentage of the desired number of tasks to place and keep running in the task set.

Contents

unit

The unit of measure for the scale value.

Type: String

Valid Values: PERCENT

Required: No

value

The value, specified as a percent total of a service's `desiredCount`, to scale the task set. Accepted values are numbers between 0 and 100.

Type: Double

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Secret

An object representing the secret to expose to your container. Secrets can be exposed to a container in the following ways:

- To inject sensitive data into your containers as environment variables, use the `secrets` container definition parameter.
- To reference sensitive information in the log configuration of a container, use the `secretOptions` container definition parameter.

For more information, see Specifying sensitive data in the *Amazon Elastic Container Service Developer Guide*.

Contents

name

The name of the secret.

Type: String

Required: Yes

valueFrom

The secret to expose to the container. The supported values are either the full ARN of the AWS Secrets Manager secret or the full ARN of the parameter in the SSM Parameter Store.

For information about the require AWS Identity and Access Management permissions, see Required IAM permissions for Amazon ECS secrets (for Secrets Manager) or Required IAM permissions for Amazon ECS secrets (for Systems Manager Parameter store) in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> If the SSM Parameter Store parameter exists in the same Region as the task you're launching, then you can use either the full ARN or name of the parameter. If the parameter exists in a different Region, then the full ARN must be specified.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Service

Details on a service within a cluster.

Contents

capacityProviderStrategy

The capacity provider strategy the service uses. When using the DescribeServices API, this field is omitted if the service was created using a launch type.

Type: Array of CapacityProviderStrategyItem objects

Required: No

clusterArn

The Amazon Resource Name (ARN) of the cluster that hosts the service.

Type: String

Required: No

createdAt

The Unix timestamp for the time when the service was created.

Type: Timestamp

Required: No

createdBy

The principal that created the service.

Type: String

Required: No

deploymentConfiguration

Optional deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.

Type: DeploymentConfiguration object

Required: No

deploymentController

The deployment controller type the service is using.

Type: DeploymentController object

Required: No

deployments

The current state of deployments for the service.

Type: Array of Deployment 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, and it can be modified with UpdateService.

Type: Integer

Required: No

enableECSManagedTags

Determines whether to use Amazon ECS managed tags for the tasks in the service. For more information, see Tagging Your Amazon ECS Resources in the *Amazon Elastic Container Service Developer Guide*.

Type: Boolean

Required: No

enableExecuteCommand

Determines whether the execute command functionality is turned on for the service. If `true`, the execute command functionality is turned on for all containers in tasks as part of the service.

Type: Boolean

Required: No

events

The event stream for your service. A maximum of 100 of the latest events are displayed.

Type: Array of ServiceEvent objects

Required: No

healthCheckGracePeriodSeconds

The period of time, in seconds, that the Amazon ECS service scheduler ignores unhealthy Elastic Load Balancing target health checks after a task has first started.

Type: Integer

Required: No

launchType

The launch type the service is using. When using the DescribeServices API, this field is omitted if the service was created using a capacity provider strategy.

Type: String

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

loadBalancers

A list of Elastic Load Balancing load balancer objects. It contains the load balancer name, the container name, and the container port to access from the load balancer. The container name is as it appears in a container definition.

Type: Array of LoadBalancer objects

Required: No

networkConfiguration

The VPC subnet and security group configuration for tasks that receive their own elastic network interface by using the `awsvpc` networking mode.

Type: NetworkConfiguration object

Required: No

pendingCount

The number of tasks in the cluster that are in the `PENDING` state.

Type: Integer

Required: No

placementConstraints

The placement constraints for the tasks in the service.

Type: Array of PlacementConstraint objects

Required: No

placementStrategy

The placement strategy that determines how tasks for the service are placed.

Type: Array of PlacementStrategy objects

Required: No

platformFamily

The operating system that your tasks in the service run on. A platform family is specified only for tasks using the Fargate launch type.

All tasks that run as part of this service must use the same `platformFamily` value as the service (for example, `LINUX`).

Type: String

Required: No

platformVersion

The platform version to run your service on. A platform version is only specified for tasks that are hosted on AWS Fargate. If one isn't specified, the `LATEST` platform version is used. For more information, see AWS Fargate Platform Versions in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

propagateTags

Determines whether to propagate the tags from the task definition or the service to the task. If no value is specified, the tags aren't propagated.

Type: String

Valid Values: `TASK_DEFINITION | SERVICE | NONE`

Required: No

roleArn

The ARN of the IAM role that's associated with the service. It allows the Amazon ECS container agent to register container instances with an Elastic Load Balancing load balancer.

Type: String

Required: No

runningCount

The number of tasks in the cluster that are in the `RUNNING` state.

Type: Integer

Required: No

schedulingStrategy

The scheduling strategy to use for the service. For more information, see Services.

There are two service scheduler strategies available.

- `REPLICA`-The replica scheduling strategy places and maintains the desired number of tasks across your cluster. By default, the service scheduler spreads tasks across Availability Zones. You can use task placement strategies and constraints to customize task placement decisions.
- `DAEMON`-The daemon scheduling strategy deploys exactly one task on each active container instance. This task meets all of the task placement constraints that you specify in your cluster. The service scheduler also evaluates the task placement constraints for running tasks. It stop tasks that don't meet the placement constraints.

> **Note**
>
> Fargate tasks don't support the `DAEMON` scheduling strategy.

Type: String

Valid Values: `REPLICA | DAEMON`

Required: No

serviceArn

The ARN that identifies the service. For more information about the ARN format, see Amazon Resource Name (ARN) in the *Amazon ECS Developer Guide*.

Type: String

Required: No

serviceName

The name of your service. Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed. Service names must be unique within a cluster. However, you can have similarly named services in multiple clusters within a Region or across multiple Regions.

Type: String

Required: No

serviceRegistries

The details for the service discovery registries to assign to this service. For more information, see Service Discovery.

Type: Array of ServiceRegistry objects

Required: No

status

The status of the service. The valid values are `ACTIVE`, `DRAINING`, or `INACTIVE`.

Type: String

Required: No

tags

The metadata that you apply to the service to help you categorize and organize them. Each tag consists of a key and an optional value. You define bot the key and value.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskDefinition

The task definition to use for tasks in the service. This value is specified when the service is created with CreateService, and it can be modified with UpdateService.

Type: String

Required: No

taskSets

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Type: Array of TaskSet objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectClientAlias

Each alias ("endpoint") is a fully-qualified name and port number that other tasks ("clients") can use to connect to this service.

Each name and port mapping must be unique within the namespace.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Contents

port

The listening port number for the Service Connect proxy. This port is available inside of all of the tasks within the same namespace.

To avoid changing your applications in client Amazon ECS services, set this to the same port that the client application uses by default. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 65535.

Required: Yes

dnsName

The `dnsName` is the name that you use in the applications of client tasks to connect to this service. The name must be a valid DNS name but doesn't need to be fully-qualified. The name can include up to 127 characters. The name can include lowercase letters, numbers, underscores (_), hyphens (-), and periods (.). The name can't start with a hyphen.

If this parameter isn't specified, the default value of `discoveryName.namespace` is used. If the `discoveryName` isn't specified, the port mapping name from the task definition is used in `portName.namespace`.

To avoid changing your applications in client Amazon ECS services, set this to the same name that the client application uses by default. For example, a few common names are `database`, db, or the lowercase name of a database, such as `mysql` or `redis`. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectConfiguration

The Service Connect configuration of your Amazon ECS service. The configuration for this service to discover and connect to services, and be discovered by, and connected from, other services within a namespace.

Tasks that run in a namespace can use short names to connect to services in the namespace. Tasks can connect to services across all of the clusters in the namespace. Tasks connect through a managed proxy container that collects logs and metrics for increased visibility. Only the tasks that Amazon ECS services create are supported with Service Connect. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Contents

enabled

Specifies whether to use Service Connect with this service.

Type: Boolean

Required: Yes

logConfiguration

The log configuration for the container. This parameter maps to `LogConfig` in the Create a container section of the Docker Remote API and the `--log-driver` option to `docker run`.

By default, containers use the same logging driver that the Docker daemon uses. However, the container might use a different logging driver than the Docker daemon by specifying a log driver configuration in the container definition. For more information about the options for different supported log drivers, see Configure logging drivers in the Docker documentation.

Understand the following when specifying a log configuration for your containers.

- Amazon ECS currently supports a subset of the logging drivers available to the Docker daemon. Additional log drivers may be available in future releases of the Amazon ECS container agent.

  For tasks on AWS Fargate, the supported log drivers are `awslogs`, `splunk`, and `awsfirelens`.

For tasks hosted on Amazon EC2 instances, the supported log drivers are `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`,`syslog`, `splunk`, and `awsfirelens`.

- This parameter requires version 1.18 of the Docker Remote API or greater on your container instance.
- For tasks that are hosted on Amazon EC2 instances, the Amazon ECS container agent must register the available logging drivers with the `ECS_AVAILABLE_LOGGING_DRIVERS` environment variable before containers placed on that instance can use these log configuration options. For more information, see Amazon ECS container agent configuration in the *Amazon Elastic Container Service Developer Guide*.
- For tasks that are on AWS Fargate, because you don't have access to the underlying infrastructure your tasks are hosted on, any additional software needed must be installed outside of the task. For example, the Fluentd output aggregators or a remote host running Logstash to send Gelf logs to.

Type: LogConfiguration object

Required: No

namespace

The namespace name or full Amazon Resource Name (ARN) of the AWS Cloud Map namespace for use with Service Connect. The namespace must be in the same AWS Region as the Amazon ECS service and cluster. The type of namespace doesn't affect Service Connect. For more information about AWS Cloud Map, see Working with Services in the *AWS Cloud Map Developer Guide*.

Type: String

Required: No

services

The list of Service Connect service objects. These are names and aliases (also known as endpoints) that are used by other Amazon ECS services to connect to this service.

This field is not required for a "client" Amazon ECS service that's a member of a namespace only to connect to other services within the namespace. An example of this would be a frontend application that accepts incoming requests from either a load balancer that's attached to the service or by other means.

An object selects a port from the task definition, assigns a name for the AWS Cloud Map service, and a list of aliases (endpoints) and ports for client applications to refer to this service.

Type: Array of ServiceConnectService objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectService

The Service Connect service object configuration. For more information, see Service Connect in the *Amazon Elastic Container Service Developer Guide*.

Contents

portName

The `portName` must match the name of one of the `portMappings` from all the containers in the task definition of this Amazon ECS service.

Type: String

Required: Yes

clientAliases

The list of client aliases for this Service Connect service. You use these to assign names that can be used by client applications. The maximum number of client aliases that you can have in this list is 1.

Each alias ("endpoint") is a fully-qualified name and port number that other Amazon ECS tasks ("clients") can use to connect to this service.

Each name and port mapping must be unique within the namespace.

For each `ServiceConnectService`, you must provide at least one `clientAlias` with one `port`.

Type: Array of ServiceConnectClientAlias objects

Required: No

discoveryName

The `discoveryName` is the name of the new AWS Cloud Map service that Amazon ECS creates for this Amazon ECS service. This must be unique within the AWS Cloud Map namespace. The name can contain up to 64 characters. The name can include lowercase letters, numbers, underscores (_), and hyphens (-). The name can't start with a hyphen.

If the `discoveryName` isn't specified, the port mapping name from the task definition is used in `portName.namespace`.

Type: String

Required: No

ingressPortOverride

The port number for the Service Connect proxy to listen on.

Use the value of this field to bypass the proxy for traffic on the port number specified in the named `portMapping` in the task definition of this application, and then use it in your VPC security groups to allow traffic into the proxy for this Amazon ECS service.

In `awsvpc` mode and Fargate, the default value is the container port number. The container port number is in the `portMapping` in the task definition. In bridge mode, the default value is the ephemeral port of the Service Connect proxy.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 65535.

Required: No

timeout

A reference to an object that represents the configured timeouts for Service Connect.

Type: TimeoutConfiguration object

Required: No

tls

A reference to an object that represents a Transport Layer Security (TLS) configuration.

Type: ServiceConnectTlsConfiguration object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectServiceResource

The Service Connect resource. Each configuration maps a discovery name to a AWS Cloud Map service name. The data is stored in AWS Cloud Map as part of the Service Connect configuration for each discovery name of this Amazon ECS service.

A task can resolve the `dnsName` for each of the `clientAliases` of a service. However a task can't resolve the discovery names. If you want to connect to a service, refer to the `ServiceConnectConfiguration` of that service for the list of `clientAliases` that you can use.

Contents

discoveryArn

The Amazon Resource Name (ARN) for the namespace in AWS Cloud Map that matches the discovery name for this Service Connect resource. You can use this ARN in other integrations with AWS Cloud Map. However, Service Connect can't ensure connectivity outside of Amazon ECS.

Type: String

Required: No

discoveryName

The discovery name of this Service Connect resource.

The `discoveryName` is the name of the new AWS Cloud Map service that Amazon ECS creates for this Amazon ECS service. This must be unique within the AWS Cloud Map namespace. The name can contain up to 64 characters. The name can include lowercase letters, numbers, underscores (_), and hyphens (-). The name can't start with a hyphen.

If the `discoveryName` isn't specified, the port mapping name from the task definition is used in `portName.namespace`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectTlsCertificateAuthority

An object that represents the AWS Private Certificate Authority certificate.

Contents

awsPcaAuthorityArn

The ARN of the AWS Private Certificate Authority certificate.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceConnectTlsConfiguration

An object that represents the configuration for Service Connect TLS.

Contents

issuerCertificateAuthority

The signer certificate authority.

Type: ServiceConnectTlsCertificateAuthority object

Required: Yes

kmsKey

The AWS Key Management Service key.

Type: String

Required: No

roleArn

The Amazon Resource Name (ARN) of the IAM role that's associated with the Service Connect TLS.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceEvent

The details for an event that's associated with a service.

Contents

createdAt

The Unix timestamp for the time when the event was triggered.

Type: Timestamp

Required: No

id

The ID string for the event.

Type: String

Required: No

message

The event message.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceManagedEBSVolumeConfiguration

The configuration for the Amazon EBS volume that Amazon ECS creates and manages on your behalf. These settings are used to create each Amazon EBS volume, with one volume created for each task in the service.

Many of these parameters map 1:1 with the Amazon EBS `CreateVolume` API request parameters.

Contents

roleArn

The ARN of the IAM role to associate with this volume. This is the Amazon ECS infrastructure IAM role that is used to manage your AWS infrastructure. We recommend using the Amazon ECS-managed `AmazonECSInfrastructureRolePolicyForVolumes` IAM policy with this role. For more information, see Amazon ECS infrastructure IAM role in the *Amazon ECS Developer Guide*.

Type: String

Required: Yes

encrypted

Indicates whether the volume should be encrypted. If no value is specified, encryption is turned on by default. This parameter maps 1:1 with the `Encrypted` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Boolean

Required: No

filesystemType

The Linux filesystem type for the volume. For volumes created from a snapshot, you must specify the same filesystem type that the volume was using when the snapshot was created. If there is a filesystem type mismatch, the task will fail to start.

The available filesystem types are `ext3`, `ext4`, and `xfs`. If no value is specified, the `xfs` filesystem type is used by default.

Type: String

Valid Values: `ext3 | ext4 | xfs`

Required: No

iops

The number of I/O operations per second (IOPS). For `gp3`, `io1`, and `io2` volumes, this represents the number of IOPS that are provisioned for the volume. For `gp2` volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting.

The following are the supported values for each volume type.

- `gp3`: 3,000 - 16,000 IOPS
- `io1`: 100 - 64,000 IOPS
- `io2`: 100 - 256,000 IOPS

This parameter is required for `io1` and `io2` volume types. The default for `gp3` volumes is `3,000 IOPS`. This parameter is not supported for `st1`, `sc1`, or `standard` volume types.

This parameter maps 1:1 with the `Iops` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Integer

Required: No

kmsKeyId

The Amazon Resource Name (ARN) identifier of the AWS Key Management Service key to use for Amazon EBS encryption. When encryption is turned on and no AWS Key Management Service key is specified, the default AWS managed key for Amazon EBS volumes is used. This parameter maps 1:1 with the `KmsKeyId` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

> **Important**
>
> AWS authenticates the AWS Key Management Service key asynchronously. Therefore, if you specify an ID, alias, or ARN that is invalid, the action can appear to complete, but eventually fails.

Type: String

Required: No

sizeInGiB

The size of the volume in GiB. You must specify either a volume size or a snapshot ID. If you specify a snapshot ID, the snapshot size is used for the volume size by default. You can optionally specify a volume size greater than or equal to the snapshot size. This parameter maps 1:1 with the `Size` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

The following are the supported volume size values for each volume type.

- `gp2` and `gp3`: 1-16,384
- `io1` and `io2`: 4-16,384
- `st1` and `sc1`: 125-16,384
- `standard`: 1-1,024

Type: Integer

Required: No

snapshotId

The snapshot that Amazon ECS uses to create the volume. You must specify either a snapshot ID or a volume size. This parameter maps 1:1 with the `SnapshotId` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: String

Required: No

tagSpecifications

The tags to apply to the volume. Amazon ECS applies service-managed tags by default. This parameter maps 1:1 with the `TagSpecifications.N` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Array of EBSTagSpecification objects

Required: No

throughput

The throughput to provision for a volume, in MiB/s, with a maximum of 1,000 MiB/s. This parameter maps 1:1 with the `Throughput` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

> **Important**
>
> This parameter is only supported for the gp3 volume type.

Type: Integer

Required: No

volumeType

The volume type. This parameter maps 1:1 with the `VolumeType` parameter of the CreateVolume API in the *Amazon EC2 API Reference*. For more information, see Amazon EBS volume types in the *Amazon EC2 User Guide*.

The following are the supported volume types.

- General Purpose SSD: `gp2|gp3`
- Provisioned IOPS SSD: `io1|io2`
- Throughput Optimized HDD: `st1`
- Cold HDD: `sc1`
- Magnetic: `standard`

> **Note**
>
> The magnetic volume type is not supported on Fargate.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceRegistry

The details for the service registry.

Each service may be associated with one service registry. Multiple service registries for each service are not supported.

When you add, update, or remove the service registries configuration, Amazon ECS starts a new deployment. New tasks are registered and deregistered to the updated service registry configuration.

Contents

containerName

The container name value to be used for your service discovery service. It's already specified in the task definition. If the task definition that your service task specifies uses the `bridge` or `host` network mode, you must specify a `containerName` and `containerPort` combination from the task definition. If the task definition that your service task specifies uses the `awsvpc` network mode and a type SRV DNS record is used, you must specify either a `containerName` and `containerPort` combination or a `port` value. However, you can't specify both.

Type: String

Required: No

containerPort

The port value to be used for your service discovery service. It's already specified in the task definition. If the task definition your service task specifies uses the `bridge` or `host` network mode, you must specify a `containerName` and `containerPort` combination from the task definition. If the task definition your service task specifies uses the `awsvpc` network mode and a type SRV DNS record is used, you must specify either a `containerName` and `containerPort` combination or a `port` value. However, you can't specify both.

Type: Integer

Required: No

port

The port value used if your service discovery service specified an SRV record. This field might be used if both the `awsvpc` network mode and SRV records are used.

Type: Integer

Required: No

registryArn

The Amazon Resource Name (ARN) of the service registry. The currently supported service registry is AWS Cloud Map. For more information, see CreateService.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServiceVolumeConfiguration

The configuration for a volume specified in the task definition as a volume that is configured at launch time. Currently, the only supported volume type is an Amazon EBS volume.

Contents

name

The name of the volume. This value must match the volume name from the `Volume` object in the task definition.

Type: String

Required: Yes

managedEBSVolume

The configuration for the Amazon EBS volume that Amazon ECS creates and manages on your behalf. These settings are used to create each Amazon EBS volume, with one volume created for each task in the service. The Amazon EBS volumes are visible in your account in the Amazon EC2 console once they are created.

Type: ServiceManagedEBSVolumeConfiguration object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Session

The details for the execute command session.

Contents

sessionId

The ID of the execute command session.

Type: String

Required: No

streamUrl

A URL to the managed agent on the container that the SSM Session Manager client uses to send commands and receive output from the container.

Type: String

Required: No

tokenValue

An encrypted token value containing session and caller information. It's used to authenticate the connection to the container.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Setting

The current account setting for a resource.

Contents

name

The Amazon ECS resource name.

Type: String

Valid Values: `serviceLongArnFormat | taskLongArnFormat | containerInstanceLongArnFormat | awsvpcTrunking | containerInsights | fargateFIPSMode | tagResourceAuthorization | fargateTaskRetirementWaitPeriod | guardDutyActivate`

Required: No

principalArn

The ARN of the principal. It can be a user, role, or the root user. If this field is omitted, the authenticated user is assumed.

Type: String

Required: No

type

Indicates whether AWS manages the account setting, or if the user manages it.

`aws_managed` account settings are read-only, as AWS manages such on the customer's behalf. Currently, the `guardDutyActivate` account setting is the only one AWS manages.

Type: String

Valid Values: `user | aws_managed`

Required: No

value

Determines whether the account setting is on or off for the specified resource.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SystemControl

A list of namespaced kernel parameters to set in the container. This parameter maps to `Sysctls` in the Create a container section of the Docker Remote API and the `--sysctl` option to docker run. For example, you can configure `net.ipv4.tcp_keepalive_time` setting to maintain longer lived connections.

We don't recommend that you specify network-related `systemControls` parameters for multiple containers in a single task that also uses either the `awsvpc` or `host` network mode. Doing this has the following disadvantages:

- For tasks that use the `awsvpc` network mode including Fargate, if you set `systemControls` for any container, it applies to all containers in the task. If you set different `systemControls` for multiple containers in a single task, the container that's started last determines which `systemControls` take effect.
- For tasks that use the `host` network mode, the network namespace `systemControls` aren't supported.

If you're setting an IPC resource namespace to use for the containers in the task, the following conditions apply to your system controls. For more information, see IPC mode.

- For tasks that use the `host` IPC mode, IPC namespace `systemControls` aren't supported.
- For tasks that use the `task` IPC mode, IPC namespace `systemControls` values apply to all containers within a task.

Note

This parameter is not supported for Windows containers.

Note

This parameter is only supported for tasks that are hosted on AWS Fargate if the tasks are using platform version `1.4.0` or later (Linux). This isn't supported for Windows containers on Fargate.

Contents

namespace

The namespaced kernel parameter to set a `value` for.

Type: String

Required: No

value

The namespaced kernel parameter to set a `value` for.

Valid IPC namespace values: `"kernel.msgmax" | "kernel.msgmnb" | "kernel.msgmni" | "kernel.sem" | "kernel.shmall" | "kernel.shmmax" | "kernel.shmmni" | "kernel.shm_rmid_forced"`, and `Sysctls` that start with `"fs.mqueue.*"`

Valid network namespace values: `Sysctls` that start with `"net.*"`

All of these values are supported by Fargate.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

The metadata that you apply to a resource to help you categorize and organize them. Each tag consists of a key and an optional value. You define them.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Contents

key

One part of a key-value pair that make up a tag. A `key` is a general label that acts like a category for more specific tag values.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

value

The optional part of a key-value pair that make up a tag. A `value` acts as a descriptor within a tag category (key).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Task

Details on a task in a cluster.

Contents

attachments

The Elastic Network Adapter that's associated with the task if the task uses the `awsvpc` network mode.

Type: Array of Attachment objects

Required: No

attributes

The attributes of the task

Type: Array of Attribute objects

Required: No

availabilityZone

The Availability Zone for the task.

Type: String

Required: No

capacityProviderName

The capacity provider that's associated with the task.

Type: String

Required: No

clusterArn

The ARN of the cluster that hosts the task.

Type: String

Required: No

connectivity

The connectivity status of a task.

Type: String

Valid Values: `CONNECTED | DISCONNECTED`

Required: No

connectivityAt

The Unix timestamp for the time when the task last went into `CONNECTED` status.

Type: Timestamp

Required: No

containerInstanceArn

The ARN of the container instances that host the task.

Type: String

Required: No

containers

The containers that's associated with the task.

Type: Array of Container objects

Required: No

cpu

The number of CPU units used by the task as expressed in a task definition. It can be expressed as an integer using CPU units (for example, `1024`). It can also be expressed as a string using vCPUs (for example, `1 vCPU` or `1 vcpu`). String values are converted to an integer that indicates the CPU units when the task definition is registered.

If you use the EC2 launch type, this field is optional. Supported values are between `128` CPU units (`0.125` vCPUs) and `10240` CPU units (`10` vCPUs).

If you use the Fargate launch type, this field is required. You must use one of the following values. These values determine the range of supported values for the `memory` parameter:

The CPU units cannot be less than 1 vCPU when you use Windows containers on Fargate.

- 256 (.25 vCPU) - Available `memory` values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
- 512 (.5 vCPU) - Available `memory` values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
- 1024 (1 vCPU) - Available `memory` values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
- 2048 (2 vCPU) - Available `memory` values: 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
- 4096 (4 vCPU) - Available `memory` values: 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)
- 8192 (8 vCPU) - Available `memory` values: 16 GB and 60 GB in 4 GB increments

  This option requires Linux platform `1.4.0` or later.
- 16384 (16vCPU) - Available `memory` values: 32GB and 120 GB in 8 GB increments

  This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

createdAt

The Unix timestamp for the time when the task was created. More specifically, it's for the time when the task entered the `PENDING` state.

Type: Timestamp

Required: No

desiredStatus

The desired status of the task. For more information, see Task Lifecycle.

Type: String

Required: No

enableExecuteCommand

Determines whether execute command functionality is turned on for this task. If `true`, execute command functionality is turned on all the containers in the task.

Type: Boolean

Required: No

ephemeralStorage

The ephemeral storage settings for the task.

Type: EphemeralStorage object

Required: No

executionStoppedAt

The Unix timestamp for the time when the task execution stopped.

Type: Timestamp

Required: No

group

The name of the task group that's associated with the task.

Type: String

Required: No

healthStatus

The health status for the task. It's determined by the health of the essential containers in the task. If all essential containers in the task are reporting as `HEALTHY`, the task status also reports as `HEALTHY`. If any essential containers in the task are reporting as `UNHEALTHY` or `UNKNOWN`, the task status also reports as `UNHEALTHY` or `UNKNOWN`.

> **Note**
>
> The Amazon ECS container agent doesn't monitor or report on Docker health checks that are embedded in a container image and not specified in the container definition. For example, this includes those specified in a parent image or from the image's Dockerfile. Health check parameters that are specified in a container definition override any Docker health checks that are found in the container image.

Type: String

Valid Values: `HEALTHY | UNHEALTHY | UNKNOWN`

Required: No

inferenceAccelerators

The Elastic Inference accelerator that's associated with the task.

Type: Array of InferenceAccelerator objects

Required: No

lastStatus

The last known status for the task. For more information, see Task Lifecycle.

Type: String

Required: No

launchType

The infrastructure where your task runs on. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

memory

The amount of memory (in MiB) that the task uses as expressed in a task definition. It can be expressed as an integer using MiB (for example, `1024`). If it's expressed as a string using GB (for example, `1GB` or `1 GB`), it's converted to an integer indicating the MiB when the task definition is registered.

If you use the EC2 launch type, this field is optional.

If you use the Fargate launch type, this field is required. You must use one of the following values. The value that you choose determines the range of supported values for the `cpu` parameter.

- 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available `cpu` values: 256 (.25 vCPU)

- 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available `cpu` values: 512 (.5 vCPU)
- 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available `cpu` values: 1024 (1 vCPU)
- Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB) - Available `cpu` values: 2048 (2 vCPU)
- Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB) - Available `cpu` values: 4096 (4 vCPU)
- Between 16 GB and 60 GB in 4 GB increments - Available `cpu` values: 8192 (8 vCPU)

  This option requires Linux platform `1.4.0` or later.
- Between 32GB and 120 GB in 8 GB increments - Available `cpu` values: 16384 (16 vCPU)

  This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

overrides

One or more container overrides.

Type: TaskOverride object

Required: No

platformFamily

The operating system that your tasks are running on. A platform family is specified only for tasks that use the Fargate launch type.

All tasks that run as part of this service must use the same `platformFamily` value as the service (for example, `LINUX.`).

Type: String

Required: No

platformVersion

The platform version where your task runs on. A platform version is only specified for tasks that use the Fargate launch type. If you didn't specify one, the `LATEST` platform version is used. For

more information, see AWS Fargate Platform Versions in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

pullStartedAt

The Unix timestamp for the time when the container image pull began.

Type: Timestamp

Required: No

pullStoppedAt

The Unix timestamp for the time when the container image pull completed.

Type: Timestamp

Required: No

startedAt

The Unix timestamp for the time when the task started. More specifically, it's for the time when the task transitioned from the `PENDING` state to the `RUNNING` state.

Type: Timestamp

Required: No

startedBy

The tag specified when a task is started. If an Amazon ECS service started the task, the `startedBy` parameter contains the deployment ID of that service.

Type: String

Required: No

stopCode

The stop code indicating why a task was stopped. The `stoppedReason` might contain additional details.

For more information about stop code, see Stopped tasks error codes in the *Amazon ECS Developer Guide*.

Type: String

Valid Values: `TaskFailedToStart | EssentialContainerExited | UserInitiated | ServiceSchedulerInitiated | SpotInterruption | TerminationNotice`

Required: No

stoppedAt

The Unix timestamp for the time when the task was stopped. More specifically, it's for the time when the task transitioned from the `RUNNING` state to the `STOPPED` state.

Type: Timestamp

Required: No

stoppedReason

The reason that the task was stopped.

Type: String

Required: No

stoppingAt

The Unix timestamp for the time when the task stops. More specifically, it's for the time when the task transitions from the `RUNNING` state to `STOPPING`.

Type: Timestamp

Required: No

tags

The metadata that you apply to the task to help you categorize and organize the task. Each tag consists of a key and an optional value. You define both the key and value.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.

- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskArn

The Amazon Resource Name (ARN) of the task.

Type: String

Required: No

taskDefinitionArn

The ARN of the task definition that creates the task.

Type: String

Required: No

version

The version counter for the task. Every time a task experiences a change that starts a CloudWatch event, the version counter is incremented. If you replicate your Amazon ECS task state with CloudWatch Events, you can compare the version of a task reported by the Amazon ECS API actions with the version reported in CloudWatch Events for the task (inside the `detail` object) to verify that the version in your event stream is current.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskDefinition

The details of a task definition which describes the container and volume definitions of an Amazon Elastic Container Service task. You can specify which Docker images to use, the required resources, and other configurations related to launching the task definition through an Amazon ECS service or task.

Contents

compatibilities

The task launch types the task definition validated against during task definition registration. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of strings

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

containerDefinitions

A list of container definitions in JSON format that describe the different containers that make up your task. For more information about container definition parameters and defaults, see Amazon ECS Task Definitions in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of ContainerDefinition objects

Required: No

cpu

The number of `cpu` units used by the task. If you use the EC2 launch type, this field is optional. Any value can be used. If you use the Fargate launch type, this field is required. You must use one of the following values. The value that you choose determines your range of valid values for the `memory` parameter.

The CPU units cannot be less than 1 vCPU when you use Windows containers on Fargate.

- 256 (.25 vCPU) - Available `memory` values: 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB)
- 512 (.5 vCPU) - Available `memory` values: 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)

- 1024 (1 vCPU) - Available `memory` values: 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
- 2048 (2 vCPU) - Available `memory` values: 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
- 4096 (4 vCPU) - Available `memory` values: 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)
- 8192 (8 vCPU) - Available `memory` values: 16 GB and 60 GB in 4 GB increments

  This option requires Linux platform `1.4.0` or later.
- 16384 (16vCPU) - Available `memory` values: 32GB and 120 GB in 8 GB increments

  This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

deregisteredAt

The Unix timestamp for the time when the task definition was deregistered.

Type: Timestamp

Required: No

ephemeralStorage

The ephemeral storage settings to use for tasks run with the task definition.

Type: EphemeralStorage object

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution role that grants the Amazon ECS container agent permission to make AWS API calls on your behalf. The task execution IAM role is required depending on the requirements of your task. For more information, see Amazon ECS task execution IAM role in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

family

The name of a family that this task definition is registered to. Up to 255 characters are allowed. Letters (both uppercase and lowercase letters), numbers, hyphens (-), and underscores (_) are allowed.

A family groups multiple versions of a task definition. Amazon ECS gives the first task definition that you registered to a family a revision number of 1. Amazon ECS gives sequential revision numbers to each task definition that you add.

Type: String

Required: No

inferenceAccelerators

The Elastic Inference accelerator that's associated with the task.

Type: Array of InferenceAccelerator objects

Required: No

ipcMode

The IPC resource namespace to use for the containers in the task. The valid values are `host`, `task`, or `none`. If `host` is specified, then all containers within the tasks that specified the `host` IPC mode on the same container instance share the same IPC resources with the host Amazon EC2 instance. If `task` is specified, all containers within the specified task share the same IPC resources. If `none` is specified, then IPC resources within the containers of a task are private and not shared with other containers in a task or on the container instance. If no value is specified, then the IPC resource namespace sharing depends on the Docker daemon setting on the container instance. For more information, see IPC settings in the *Docker run reference*.

If the `host` IPC mode is used, be aware that there is a heightened risk of undesired IPC namespace expose. For more information, see Docker security.

If you are setting namespaced kernel parameters using `systemControls` for the containers in the task, the following will apply to your IPC resource namespace. For more information, see System Controls in the *Amazon Elastic Container Service Developer Guide*.

- For tasks that use the `host` IPC mode, IPC namespace related `systemControls` are not supported.

- For tasks that use the `task` IPC mode, IPC namespace related `systemControls` will apply to all containers within a task.

> **Note**
>
> This parameter is not supported for Windows containers or tasks run on AWS Fargate.

Type: String

Valid Values: `host | task | none`

Required: No

memory

The amount (in MiB) of memory used by the task.

If your tasks runs on Amazon EC2 instances, you must specify either a task-level memory value or a container-level memory value. This field is optional and any value can be used. If a task-level memory value is specified, the container-level memory value is optional. For more information regarding container-level memory and memory reservation, see ContainerDefinition.

If your tasks runs on AWS Fargate, this field is required. You must use one of the following values. The value you choose determines your range of valid values for the cpu parameter.

- 512 (0.5 GB), 1024 (1 GB), 2048 (2 GB) - Available cpu values: 256 (.25 vCPU)
- 1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB) - Available cpu values: 512 (.5 vCPU)
- 2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB) - Available cpu values: 1024 (1 vCPU)
- Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB) - Available cpu values: 2048 (2 vCPU)
- Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB) - Available cpu values: 4096 (4 vCPU)
- Between 16 GB and 60 GB in 4 GB increments - Available cpu values: 8192 (8 vCPU)

  This option requires Linux platform `1.4.0` or later.
- Between 32GB and 120 GB in 8 GB increments - Available cpu values: 16384 (16 vCPU)

This option requires Linux platform `1.4.0` or later.

Type: String

Required: No

networkMode

The Docker networking mode to use for the containers in the task. The valid values are none, `bridge`, `awsvpc`, and `host`. If no network mode is specified, the default is `bridge`.

For Amazon ECS tasks on Fargate, the `awsvpc` network mode is required. For Amazon ECS tasks on Amazon EC2 Linux instances, any network mode can be used. For Amazon ECS tasks on Amazon EC2 Windows instances, `<default>` or `awsvpc` can be used. If the network mode is set to none, you cannot specify port mappings in your container definitions, and the tasks containers do not have external connectivity. The `host` and `awsvpc` network modes offer the highest networking performance for containers because they use the EC2 network stack instead of the virtualized network stack provided by the `bridge` mode.

With the `host` and `awsvpc` network modes, exposed container ports are mapped directly to the corresponding host port (for the `host` network mode) or the attached elastic network interface port (for the `awsvpc` network mode), so you cannot take advantage of dynamic host port mappings.

> **Important**
>
> When using the `host` network mode, you should not run containers using the root user (UID 0). It is considered best practice to use a non-root user.

If the network mode is `awsvpc`, the task is allocated an elastic network interface, and you must specify a NetworkConfiguration value when you create a service or run a task with the task definition. For more information, see Task Networking in the *Amazon Elastic Container Service Developer Guide*.

If the network mode is `host`, you cannot run multiple instantiations of the same task on a single container instance when port mappings are used.

For more information, see Network settings in the *Docker run reference*.

Type: String

Valid Values: `bridge | host | awsvpc | none`

Required: No

pidMode

The process namespace to use for the containers in the task. The valid values are `host` or `task`. On Fargate for Linux containers, the only valid value is `task`. For example, monitoring sidecars might need `pidMode` to access information about other containers running in the same task.

If `host` is specified, all containers within the tasks that specified the `host` PID mode on the same container instance share the same process namespace with the host Amazon EC2 instance.

If `task` is specified, all containers within the specified task share the same process namespace.

If no value is specified, the default is a private namespace for each container. For more information, see PID settings in the *Docker run reference*.

If the `host` PID mode is used, there's a heightened risk of undesired process namespace exposure. For more information, see Docker security.

> **Note**
>
> This parameter is not supported for Windows containers.

> **Note**
>
> This parameter is only supported for tasks that are hosted on AWS Fargate if the tasks are using platform version `1.4.0` or later (Linux). This isn't supported for Windows containers on Fargate.

Type: String

Valid Values: `host | task`

Required: No

placementConstraints

An array of placement constraint objects to use for tasks.

> **Note**
>
> This parameter isn't supported for tasks run on AWS Fargate.

Type: Array of TaskDefinitionPlacementConstraint objects

Required: No

proxyConfiguration

The configuration details for the App Mesh proxy.

Your Amazon ECS container instances require at least version 1.26.0 of the container agent and at least version 1.26.0-1 of the `ecs-init` package to use a proxy configuration. If your container instances are launched from the Amazon ECS optimized AMI version `20190301` or later, they contain the required versions of the container agent and `ecs-init`. For more information, see Amazon ECS-optimized Linux AMI in the *Amazon Elastic Container Service Developer Guide*.

Type: ProxyConfiguration object

Required: No

registeredAt

The Unix timestamp for the time when the task definition was registered.

Type: Timestamp

Required: No

registeredBy

The principal that registered the task definition.

Type: String

Required: No

requiresAttributes

The container instance attributes required by your task. When an Amazon EC2 instance is registered to your cluster, the Amazon ECS container agent assigns some standard attributes to

the instance. You can apply custom attributes. These are specified as key-value pairs using the Amazon ECS console or the PutAttributes API. These attributes are used when determining task placement for tasks hosted on Amazon EC2 instances. For more information, see Attributes in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> This parameter isn't supported for tasks run on AWS Fargate.

Type: Array of Attribute objects

Required: No

requiresCompatibilities

The task launch types the task definition was validated against. The valid values are EC2, FARGATE, and EXTERNAL. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

Type: Array of strings

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

revision

The revision of the task in a particular family. The revision is a version number of a task definition in a family. When you register a task definition for the first time, the revision is 1. Each time that you register a new revision of a task definition in the same family, the revision value always increases by one. This is even if you deregistered previous revisions in this family.

Type: Integer

Required: No

runtimePlatform

The operating system that your task definitions are running on. A platform family is specified only for tasks using the Fargate launch type.

When you specify a task in a service, this value must match the `runtimePlatform` value of the service.

Type: RuntimePlatform object

Required: No

status

The status of the task definition.

Type: String

Valid Values: `ACTIVE | INACTIVE | DELETE_IN_PROGRESS`

Required: No

taskDefinitionArn

The full Amazon Resource Name (ARN) of the task definition.

Type: String

Required: No

taskRoleArn

The short name or full Amazon Resource Name (ARN) of the AWS Identity and Access Management role that grants containers in the task permission to call AWS APIs on your behalf. For more information, see Amazon ECS Task Role in the *Amazon Elastic Container Service Developer Guide*.

IAM roles for tasks on Windows require that the `-EnableTaskIAMRole` option is set when you launch the Amazon ECS-optimized Windows AMI. Your containers must also run some configuration code to use the feature. For more information, see Windows IAM roles for tasks in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

volumes

The list of data volume definitions for the task. For more information, see Using data volumes in tasks in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> The `host` and `sourcePath` parameters aren't supported for tasks run on AWS Fargate.

Type: Array of Volume objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskDefinitionPlacementConstraint

The constraint on task placement in the task definition. For more information, see Task placement constraints in the *Amazon Elastic Container Service Developer Guide*.

> **Note**
>
> Task placement constraints aren't supported for tasks run on AWS Fargate.

Contents

expression

A cluster query language expression to apply to the constraint. For more information, see Cluster query language in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

type

The type of constraint. The `MemberOf` constraint restricts selection to be from a group of valid candidates.

Type: String

Valid Values: `memberOf`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskManagedEBSVolumeConfiguration

The configuration for the Amazon EBS volume that Amazon ECS creates and manages on your behalf. These settings are used to create each Amazon EBS volume, with one volume created for each task.

Contents

roleArn

The ARN of the IAM role to associate with this volume. This is the Amazon ECS infrastructure IAM role that is used to manage your AWS infrastructure. We recommend using the Amazon ECS-managed `AmazonECSInfrastructureRolePolicyForVolumes` IAM policy with this role. For more information, see Amazon ECS infrastructure IAM role in the *Amazon ECS Developer Guide*.

Type: String

Required: Yes

encrypted

Indicates whether the volume should be encrypted. If no value is specified, encryption is turned on by default. This parameter maps 1:1 with the `Encrypted` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Boolean

Required: No

filesystemType

The Linux filesystem type for the volume. For volumes created from a snapshot, you must specify the same filesystem type that the volume was using when the snapshot was created. If there is a filesystem type mismatch, the task will fail to start.

The available filesystem types are `ext3`, `ext4`, and `xfs`. If no value is specified, the `xfs` filesystem type is used by default.

Type: String

Valid Values: `ext3 | ext4 | xfs`

Required: No

iops

The number of I/O operations per second (IOPS). For `gp3`, `io1`, and `io2` volumes, this represents the number of IOPS that are provisioned for the volume. For `gp2` volumes, this represents the baseline performance of the volume and the rate at which the volume accumulates I/O credits for bursting.

The following are the supported values for each volume type.

- `gp3`: 3,000 - 16,000 IOPS
- `io1`: 100 - 64,000 IOPS
- `io2`: 100 - 256,000 IOPS

This parameter is required for `io1` and `io2` volume types. The default for `gp3` volumes is `3,000 IOPS`. This parameter is not supported for `st1`, `sc1`, or `standard` volume types.

This parameter maps 1:1 with the `Iops` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Integer

Required: No

kmsKeyId

The Amazon Resource Name (ARN) identifier of the AWS Key Management Service key to use for Amazon EBS encryption. When encryption is turned on and no AWS Key Management Service key is specified, the default AWS managed key for Amazon EBS volumes is used. This parameter maps 1:1 with the `KmsKeyId` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

> **Important**
>
> AWS authenticates the AWS Key Management Service key asynchronously. Therefore, if you specify an ID, alias, or ARN that is invalid, the action can appear to complete, but eventually fails.

Type: String

Required: No

sizeInGiB

The size of the volume in GiB. You must specify either a volume size or a snapshot ID. If you specify a snapshot ID, the snapshot size is used for the volume size by default. You can optionally specify a volume size greater than or equal to the snapshot size. This parameter maps 1:1 with the `Size` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

The following are the supported volume size values for each volume type.

- `gp2` and `gp3`: 1-16,384
- `io1` and `io2`: 4-16,384
- `st1` and `sc1`: 125-16,384
- `standard`: 1-1,024

Type: Integer

Required: No

snapshotId

The snapshot that Amazon ECS uses to create the volume. You must specify either a snapshot ID or a volume size. This parameter maps 1:1 with the `SnapshotId` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: String

Required: No

tagSpecifications

The tags to apply to the volume. Amazon ECS applies service-managed tags by default. This parameter maps 1:1 with the `TagSpecifications.N` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

Type: Array of EBSTagSpecification objects

Required: No

terminationPolicy

The termination policy for the volume when the task exits. This provides a way to control whether Amazon ECS terminates the Amazon EBS volume when the task stops.

Type: TaskManagedEBSVolumeTerminationPolicy object

Required: No

throughput

The throughput to provision for a volume, in MiB/s, with a maximum of 1,000 MiB/s. This parameter maps 1:1 with the `Throughput` parameter of the CreateVolume API in the *Amazon EC2 API Reference*.

> **Important**
>
> This parameter is only supported for the `gp3` volume type.

Type: Integer

Required: No

volumeType

The volume type. This parameter maps 1:1 with the `VolumeType` parameter of the CreateVolume API in the *Amazon EC2 API Reference*. For more information, see Amazon EBS volume types in the *Amazon EC2 User Guide*.

The following are the supported volume types.

- General Purpose SSD: `gp2|gp3`
- Provisioned IOPS SSD: `io1|io2`
- Throughput Optimized HDD: `st1`
- Cold HDD: `sc1`
- Magnetic: `standard`

> **Note**
>
> The magnetic volume type is not supported on Fargate.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskManagedEBSVolumeTerminationPolicy

The termination policy for the Amazon EBS volume when the task exits. For more information, see Amazon ECS volume termination policy.

Contents

deleteOnTermination

Indicates whether the volume should be deleted on when the task stops. If a value of `true` is specified,
Amazon ECS deletes the Amazon EBS volume on your behalf when the task goes into the STOPPED state. If no value is specified, the
default value is `true` is used. When set to `false`, Amazon ECS leaves the volume in your account.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskOverride

The overrides that are associated with a task.

Contents

containerOverrides

One or more container overrides that are sent to a task.

Type: Array of ContainerOverride objects

Required: No

cpu

The CPU override for the task.

Type: String

Required: No

ephemeralStorage

The ephemeral storage setting override for the task.

> **Note**
>
> This parameter is only supported for tasks hosted on Fargate that use the following platform versions:
>
> - Linux platform version `1.4.0` or later.
> - Windows platform version `1.0.0` or later.

Type: EphemeralStorage object

Required: No

executionRoleArn

The Amazon Resource Name (ARN) of the task execution role override for the task. For more information, see Amazon ECS task execution IAM role in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

inferenceAcceleratorOverrides

The Elastic Inference accelerator override for the task.

Type: Array of InferenceAcceleratorOverride objects

Required: No

memory

The memory override for the task.

Type: String

Required: No

taskRoleArn

The Amazon Resource Name (ARN) of the role that containers in this task can assume. All containers in this task are granted the permissions that are specified in this role. For more information, see IAM Role for Tasks in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskSet

Information about a set of Amazon ECS tasks in either an AWS CodeDeploy or an EXTERNAL deployment. An Amazon ECS task set includes details such as the desired number of tasks, how many tasks are running, and whether the task set serves production traffic.

Contents

capacityProviderStrategy

The capacity provider strategy that are associated with the task set.

Type: Array of CapacityProviderStrategyItem objects

Required: No

clusterArn

The Amazon Resource Name (ARN) of the cluster that the service that hosts the task set exists in.

Type: String

Required: No

computedDesiredCount

The computed desired count for the task set. This is calculated by multiplying the service's `desiredCount` by the task set's `scale` percentage. The result is always rounded up. For example, if the computed desired count is 1.2, it rounds up to 2 tasks.

Type: Integer

Required: No

createdAt

The Unix timestamp for the time when the task set was created.

Type: Timestamp

Required: No

externalId

The external ID associated with the task set.

If an AWS CodeDeploy deployment created a task set, the `externalId` parameter contains the AWS CodeDeploy deployment ID.

If a task set is created for an external deployment and is associated with a service discovery registry, the `externalId` parameter contains the `ECS_TASK_SET_EXTERNAL_ID` AWS Cloud Map attribute.

Type: String

Required: No

id

The ID of the task set.

Type: String

Required: No

launchType

The launch type the tasks in the task set are using. For more information, see Amazon ECS launch types in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Valid Values: `EC2 | FARGATE | EXTERNAL`

Required: No

loadBalancers

Details on a load balancer that are used with a task set.

Type: Array of LoadBalancer objects

Required: No

networkConfiguration

The network configuration for the task set.

Type: NetworkConfiguration object

Required: No

pendingCount

The number of tasks in the task set that are in the PENDING status during a deployment. A task in the PENDING state is preparing to enter the RUNNING state. A task set enters the PENDING status when it launches for the first time or when it's restarted after being in the STOPPED state.

Type: Integer

Required: No

platformFamily

The operating system that your tasks in the set are running on. A platform family is specified only for tasks that use the Fargate launch type.

All tasks in the set must have the same value.

Type: String

Required: No

platformVersion

The AWS Fargate platform version where the tasks in the task set are running. A platform version is only specified for tasks run on AWS Fargate. For more information, see AWS Fargate platform versions in the *Amazon Elastic Container Service Developer Guide*.

Type: String

Required: No

runningCount

The number of tasks in the task set that are in the RUNNING status during a deployment. A task in the RUNNING state is running and ready for use.

Type: Integer

Required: No

scale

A floating-point percentage of your desired number of tasks to place and keep running in the task set.

Type: Scale object

Required: No

serviceArn

The Amazon Resource Name (ARN) of the service the task set exists in.

Type: String

Required: No

serviceRegistries

The details for the service discovery registries to assign to this task set. For more information, see Service discovery.

Type: Array of ServiceRegistry objects

Required: No

stabilityStatus

The stability status. This indicates whether the task set has reached a steady state. If the following conditions are met, the task set are in `STEADY_STATE`:

- The task `runningCount` is equal to the `computedDesiredCount`.
- The `pendingCount` is `0`.
- There are no tasks that are running on container instances in the `DRAINING` status.
- All tasks are reporting a healthy status from the load balancers, service discovery, and container health checks.

If any of those conditions aren't met, the stability status returns `STABILIZING`.

Type: String

Valid Values: `STEADY_STATE | STABILIZING`

Required: No

stabilityStatusAt

The Unix timestamp for the time when the task set stability status was retrieved.

Type: Timestamp

Required: No

startedBy

The tag specified when a task set is started. If an AWS CodeDeploy deployment created the task set, the `startedBy` parameter is `CODE_DEPLOY`. If an external deployment created the task set, the `startedBy` field isn't used.

Type: String

Required: No

status

The status of the task set. The following describes each state.

PRIMARY

The task set is serving production traffic.

ACTIVE

The task set isn't serving production traffic.

DRAINING

The tasks in the task set are being stopped, and their corresponding targets are being deregistered from their target group.

Type: String

Required: No

tags

The metadata that you apply to the task set to help you categorize and organize them. Each tag consists of a key and an optional value. You define both.

The following basic restrictions apply to tags:

- Maximum number of tags per resource - 50
- For each resource, each tag key must be unique, and each tag key can have only one value.
- Maximum key length - 128 Unicode characters in UTF-8
- Maximum value length - 256 Unicode characters in UTF-8
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally allowed characters are: letters, numbers, and spaces representable in UTF-8, and the following characters: + - = . _ : / @.
- Tag keys and values are case-sensitive.
- Do not use `aws:`, `AWS:`, or any upper or lowercase combination of such as a prefix for either keys or values as it is reserved for AWS use. You cannot edit or delete tag keys or values with this prefix. Tags with this prefix do not count against your tags per resource limit.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

taskDefinition

The task definition that the task set is using.

Type: String

Required: No

taskSetArn

The Amazon Resource Name (ARN) of the task set.

Type: String

Required: No

updatedAt

The Unix timestamp for the time when the task set was last updated.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TaskVolumeConfiguration

Configuration settings for the task volume that was `configuredAtLaunch` that weren't set during `RegisterTaskDef`.

Contents

name

The name of the volume. This value must match the volume name from the `Volume` object in the task definition.

Type: String

Required: Yes

managedEBSVolume

The configuration for the Amazon EBS volume that Amazon ECS creates and manages on your behalf. These settings are used to create each Amazon EBS volume, with one volume created for each task. The Amazon EBS volumes are visible in your account in the Amazon EC2 console once they are created.

Type: TaskManagedEBSVolumeConfiguration object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TimeoutConfiguration

An object that represents the timeout configurations for Service Connect.

> **Note**
>
> If `idleTimeout` is set to a time that is less than `perRequestTimeout`, the connection will close when the `idleTimeout` is reached and not the `perRequestTimeout`.

Contents

idleTimeoutSeconds

The amount of time in seconds a connection will stay active while idle. A value of `0` can be set to disable `idleTimeout`.

The `idleTimeout` default for `HTTP/HTTP2/GRPC` is 5 minutes.

The `idleTimeout` default for `TCP` is 1 hour.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: No

perRequestTimeoutSeconds

The amount of time waiting for the upstream to respond with a complete response per request. A value of `0` can be set to disable `perRequestTimeout`. `perRequestTimeout` can only be set if Service Connect `appProtocol` isn't `TCP`. Only `idleTimeout` is allowed for `TCP` `appProtocol`.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tmpfs

The container path, mount options, and size of the tmpfs mount.

Contents

containerPath

The absolute file path where the tmpfs volume is to be mounted.

Type: String

Required: Yes

size

The maximum size (in MiB) of the tmpfs volume.

Type: Integer

Required: Yes

mountOptions

The list of tmpfs volume mount options.

Valid values: `"defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"`

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Ulimit

The `ulimit` settings to pass to the container.

Amazon ECS tasks hosted on AWS Fargate use the default resource limit values set by the operating system with the exception of the `nofile` resource limit parameter which AWS Fargate overrides. The `nofile` resource limit sets a restriction on the number of open files that a container can use. The default `nofile` soft limit is `1024` and the default hard limit is `65535`.

You can specify the `ulimit` settings for a container in a task definition.

Contents

hardLimit

The hard limit for the `ulimit` type.

Type: Integer

Required: Yes

name

The `type` of the `ulimit`.

Type: String

Valid Values: `core | cpu | data | fsize | locks | memlock | msgqueue | nice | nofile | nproc | rss | rtprio | rttime | sigpending | stack`

Required: Yes

softLimit

The soft limit for the `ulimit` type.

Type: Integer

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VersionInfo

The Docker and Amazon ECS container agent version information about a container instance.

Contents

agentHash

The Git commit hash for the Amazon ECS container agent build on the amazon-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 that's running on the container instance.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Volume

The data volume configuration for tasks launched using this task definition. Specifying a volume configuration in a task definition is optional. The volume configuration may contain multiple volumes but only one volume configured at launch is supported. Each volume defined in the volume configuration may only specify a `name` and one of either `configuredAtLaunch`, `dockerVolumeConfiguration`, `efsVolumeConfiguration`, `fsxWindowsFileServerVolumeConfiguration`, or `host`. If an empty volume configuration is specified, by default Amazon ECS uses a host volume. For more information, see Using data volumes in tasks.

Contents

configuredAtLaunch

Indicates whether the volume should be configured at launch time. This is used to create Amazon EBS volumes for standalone tasks or tasks created as part of a service. Each task definition revision may only have one volume configured at launch in the volume configuration.

To configure a volume at launch time, use this task definition revision and specify a `volumeConfigurations` object when calling the `CreateService`, `UpdateService`, `RunTask` or `StartTask` APIs.

Type: Boolean

Required: No

dockerVolumeConfiguration

This parameter is specified when you use Docker volumes.

Windows containers only support the use of the `local` driver. To use bind mounts, specify the `host` parameter instead.

> **Note**
>
> Docker volumes aren't supported by tasks run on AWS Fargate.

Type: DockerVolumeConfiguration object

Required: No

efsVolumeConfiguration

This parameter is specified when you use an Amazon Elastic File System file system for task storage.

Type: EFSVolumeConfiguration object

Required: No

fsxWindowsFileServerVolumeConfiguration

This parameter is specified when you use Amazon FSx for Windows File Server file system for task storage.

Type: FSxWindowsFileServerVolumeConfiguration object

Required: No

host

This parameter is specified when you use bind mount host volumes. The contents of the `host` parameter determine whether your bind mount host volume persists on the host container instance and where it's stored. If the `host` parameter is empty, then the Docker daemon assigns a host path for your data volume. However, the data isn't guaranteed to persist after the containers that are associated with it stop running.

Windows containers can mount whole directories on the same drive as `$env:ProgramData`. Windows containers can't mount directories on a different drive, and mount point can't be across drives. For example, you can mount `C:\my\path:C:\my\path` and `D:\:D:\`, but not `D:\my\path:C:\my\path` or `D:\:C:\my\path`.

Type: HostVolumeProperties object

Required: No

name

The name of the volume. Up to 255 letters (uppercase and lowercase), numbers, underscores, and hyphens are allowed.

When using a volume configured at launch, the `name` is required and must also be specified as the volume name in the `ServiceVolumeConfiguration` or `TaskVolumeConfiguration` parameter when creating your service or standalone task.

For all other types of volumes, this name is referenced in the `sourceVolume` parameter of the `mountPoints` object in the container definition.

When a volume is using the `efsVolumeConfiguration`, the name is required.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VolumeFrom

Details on a data volume from another container in the same task definition.

Contents

readOnly

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

Type: Boolean

Required: No

sourceContainer

The name of another container within the same task definition to mount volumes from.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Request throttling for the Amazon ECS API

Amazon Elastic Container Service throttles all API requests for each AWS account on a per-Region basis. We do this to ensure consistent performance and fair usage of the service for all Amazon ECS customers. Throttling ensures that calls to the Amazon ECS API do not exceed the maximum allowed API request quotas for both Amazon ECS and the other AWS services that it integrates with. API calls are subject to the request quotas whether they originate from:

- A third-party application
- A command line tool
- The Amazon ECS console

If you exceed an API throttling quota, you get the `ThrottlingException` error code.

```
An error occurred (ThrottlingException) when calling the DescribeClusters operation
 (reached max retries: 4): Rate exceeded.
com.amazonaws.services.ecs.model.AmazonECSException: Rate exceeded (Service: AmazonECS;
 Status Code: 400; Error
Code: ThrottlingException; Request ID: 5ed90669-e454-464d-9b2f-6523bc86f537; Proxy:
 null)
```

How throttling Is applied

Amazon ECS uses the token bucket algorithm to implement API throttling. With this algorithm, your account has a *bucket* that holds a specific number of *tokens*. The number of tokens in the bucket represents your throttling quota at any given second.

Amazon ECS examines the rate of API request submissions for all Amazon ECS APIs in your account, per Region, and applies two types of API throttling quotas: *sustained* and *burst*. The sustained rate is the average number of API requests allowed per second over time for an operation. The burst rate is the maximum number of API requests allowed in any one second. With burst, you can periodically make a higher number of API requests than the sustained rate. Following which, Amazon ECS throttles subsequent API requests until the rate of API requests allowed over time stabilizes to the sustained rate. In the token bucket algorithm, the *bucket maximum capacity* signifies the burst rate and the *bucket refill rate* is the sustained rate. We will use these terms to provide you an illustration of Amazon ECS API request throttling in the following example.

You are throttled on the number of API requests you make and each request removes one token from the token bucket. For example, the bucket size for *Cluster read actions*, such as the `DescribeClusters` API, is 50 tokens, so you can make up to 50 `DescribeClusters` requests in one second. If you exceed 50 requests in a second, you are throttled and the remaining requests within that second fail.

Buckets automatically refill at a set rate. If the bucket is below its maximum capacity, a set number of tokens is added back to it every second until it reaches its maximum capacity. If the bucket is full when refill tokens arrive, they are discarded. The bucket cannot hold more than its maximum number of tokens. For example, the bucket size for *Cluster read actions*, such as the `DescribeClusters` API, is 50 tokens, and the refill rate is 20 tokens per second. If you make 50 `DescribeClusters` API requests in a second, the bucket is immediately reduced to zero tokens. The bucket is then refilled by 20 tokens every second, until it reaches its maximum capacity of 50 tokens. This means that the previously empty bucket reaches its maximum capacity after 2.5 seconds.

You do not need to wait for the bucket to be completely full before you can make API requests. You can use tokens as they are added to the bucket. If you immediately use the refill tokens, the bucket does not reach its maximum capacity. For example, the bucket size for *Cluster read actions*, such as the `DescribeClusters` API, is 50 tokens, and the refill rate is 20 tokens per second. If you deplete the bucket by making 50 API requests in a second, you can continue to make 20 API requests per second. The bucket can refill to the maximum capacity only if you make fewer than 20 API requests per second.

Request Token Bucket Sizes and Refill Rates

For request rate limiting purposes, API actions are grouped into categories. All API actions in a category share the same token bucket. For instance, `DescribeClusters` and `ListClusters` APIs share the *Cluster read actions* bucket, for which capacity is 50 and refill rate is 20. This means that the cumulative number of API requests for all *Cluster read actions* is throttled by the same burst rate quota of 50 API requests. Thus, you can make 25 `DescribeClusters` and 25 `ListClusters` API requests in one second, or 30 `DescribeClusters` and 20 `ListClusters`, or 50 `DescribeClusters` and 0 `ListClusters`, or 0 `DescribeClusters` and 50 `ListClusters`, but you cannot make 50 `DescribeClusters` and 50 `ListClusters` requests at the same time. Sustained rate is similarly applied cumulatively to all API requests within a bucket.

The following table shows the bucket capacity (or burst) and refill rate (or sustained) for all AWS Regions. All API action categories enforce rate quotas for each AWS account on a per-Region basis.

API action category	Actions	Bucket maximum capacity (or Burst rate)	Bucket refill rate (or Sustained rate)
Cluster modify actions	• CreateCluster • DeleteCluster • PutClusterCapacityProviders • UpdateCluster • UpdateClusterSettings	20	1
Cluster read actions	• DescribeClusters • ListClusters	50	20
Task definition modify actions	• DeregisterTaskDefinition • RegisterTaskDefinition	20	1
Task definition read actions	• DescribeTaskDefinition • ListTaskDefinitions • ListTaskDefinitionFamilies	50	20
Task definition deletion actions	• DeleteTaskDefinitions	5	1
Capacity provider modify actions	• CreateCapacityProvider • DeleteCapacityProvider • UpdateCapacityProvider	10	1
Capacity provider read actions	• DescribeCapacityProviders	50	20
Tag modify actions	• TagResource • UntagResource	20	10

API action category	Actions	Bucket maximum capacity (or Burst rate)	Bucket refill rate (or Sustained rate)
Tag read actions	• ListTagsForResource	50	20
Setting modify actions	• DeleteAccountSetting • PutAccountSetting • PutAccountSettingDefault	10	1
Setting read actions	• ListAccountSettings	50	20
Cluster resource modify actions	• DeleteAttributes • DeregisterContainerInstance • ExecuteCommand • PutAttributes • RunTask[1] • StartTask • StopTask • UpdateContainerAgent • UpdateContainerInstancesStates	100	40
Cluster resource read actions	• DescribeContainerInstances • DescribeTasks • ListAttributes • ListContainerInstances • ListTasks	100	20

API action category	Actions	Bucket maximum capacity (or Burst rate)	Bucket refill rate (or Sustained rate)
Agent modify actions	• `RegisterContainerInstance` • `SubmitAttachmentStateChanges` • `SubmitContainerStateChange` • `SubmitTaskStateChange`	200	120
Service modify actions	• `CreateService` • `DeleteService` • `UpdateService`	50	5
Service read actions	• `DescribeServices` • `ListServices`	100	20
Task protection actions	• `UpdateTaskProtection` • `GetTaskProtection`	200	80
Cluster service resource read actions	• `ListServicesByNamespace`	10	1

[1] AWS Fargate (Fargate) additionally throttles Amazon ECS `RunTask` API to the rates listed here in the *Amazon ECS Developer Guide*.

Adjusting API throttling quotas

You can request an increase for API throttling quotas for your AWS account. To request a quota adjustment, contact the AWS Support Center.

Handling API throttling

You can implement an error retry and exponential back-off strategy to avoid the impact of throttling errors on your workloads. If you use AWS SDK, the automatic retry logic is already built-in and configurable. You can refer to the following resources for more details:

- Error retries and exponential backoff in AWS in the AWS General Reference Guide
- Exponential backoff and jitter blog post
- Timeouts, retries, and backoff with jitter article in the Amazon Builder's Library

Ensuring idempotency

When you perform a mutating operation, you might see an exception because of timeouts or server issues occurring after the resources are mutated. This can make it difficult to determine whether the mutation occurred, and could lead to multiple retries. However, if the original operation and the subsequent retries actually performed the mutations you might have applied stacking changes or created more resources than you intended to. Idempotency ensures that an operation mutates resources no more than one time. With an idempotent request, if the original request mutated successfully, any subsequent retries complete successfully without performing any further mutation.

Topics

- Idempotency in Amazon ECS
- Idempotency for RunTask
- Examples
- Retry recommendations for idempotent requests

Idempotency in Amazon ECS

The following API actions optionally support idempotency using a *client token*. The corresponding AWS CLI commands also support idempotency using a client token. A client token is a unique, case-sensitive string. To make an idempotent API request using one of these actions, specify a client token in the request. You should not reuse the same client token for other API requests. If you retry a request that completed successfully using the same client token and the same parameters, the retry succeeds without performing any further actions.

Idempotent using a client token

- CreateService

  The client token can be up to 36 ASCII characters in the range of 33-126 (inclusive).
- CreateTaskSet

  The client token can be up to 36 ASCII characters in the range of 33-126 (inclusive).
- RunTask

The client token can be up to 64 ASCII characters in the range of 33-126 (inclusive).

Types of idempotency

- cluster – Requests with the same token in the same cluster are idempotent. For example, ClientToken A can only be used as a request parameter one time for `RunTask` requests in Cluster X. `RunTask` requests to other clusters are considered a separate request. Therefore, you can use ClientToken A for a `RunTask` request for cluster Y.

Idempotency for RunTask

The `RunTask` API supports idempotency using a client token. A client token is a unique string that you specify when you make an API request. If you retry an API request with the same client token and the same request parameters after it has completed successfully, the result of the original request is returned. If you retry a successful request using the same client token, but one or more of the parameters are different, other than the Region or Availability Zone, the retry fails with a `ConflictException`. If you do not specify your own client token, the AWS SDK and AWS Command Line Interface automatically generate a client token for the request to ensure that it is idempotent. A client token can be any string that includes up to 64 ASCII characters in the range of 33-126 (inclusive).

The time to live (TTL) for the `RunTask` client token is 24 hours. You should not reuse the same client token for different requests. The client token maximum TTL is valid for whichever of the following two values is lower:

- 24 hours
- The lifetime of the resource plus one hour

  The lifetime of a resource is the timestamp at which the task was created to the timestamp at which the last status (`lastStatus`) transitioned to `STOPPED`. When you use `RunTask` to launch more than one task, the lifetime of the resource equals the lifetime of the last task that transitioned to `STOPPED`.

RunTask retry rules and responses

When you retry a request because you received a 5xx exception, the retried successful response normally includes all of the information that the original request would have returned. Tasks that have been stopped for under an hour only include the task ARN, last status, and desired status.

The following is an example snippet of the response from a retry when there is one running task, one stopped task, and one task that failed to launch.

```
{
  "failures": [
    {
      "arn": "arn:aws:ecs:us-east-1:123456789012:container/4df26bb4-f057-467b-
a079-961675296e64",
      "reason": "RESOURCE:MEMORY"
    }
  ],
  "tasks": [
    {
      "desiredStatus": "RUNNING",
      "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/fdf2c302-468c-4e55-
b884-5331d816e7fb",
      ...
    },
    {
      "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/default/fdf2c302-468c-4e55-
b884-5331d819999",
      "lastStatus": "STOPPED",
      ...
     }
  ]
}
```

Failures that are over an hour old only include the number of failed tasks.

Examples

AWS CLI command examples

To make an AWS CLI command idempotent, add the `--client-token` option.

Example: create-service

The following create-service command uses idempotency as it includes a client token.

```
aws ecs create-service \
    --cluster MyCluster \
    --service MyService \
    --task-definition MyTaskDefinition:2 \
    --desired-count 2 \
    --launch-type FARGATE \
    --platform-version LATEST \
    --network-configuration
 "awsvpcConfiguration={subnets=["subnet-12344321"],securityGroups=["sg-12344321"],assignPublicI
 \
    --client-token 550e8400-e29b-41d4-a716-44665544
```

Example: create-task-set

The following create-task-set command uses idempotency as it includes a client token.

```
aws ecs create-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-definition MyTaskDefinition:2 \
    --network-configuration
 "awsvpcConfiguration={subnets=["subnet-12344321"],securityGroups=["sg-12344321"]}" \
    --client-token 550e8400-e29b-41d4-a716-44665544
```

Example: run-task

The following run-task command uses idempotency as it includes a client token.

```
aws ecs run-task \
    --cluster MyCluster \
    --task-definition MyTaskDefinition:2 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

API request examples

To make an API request idempotent, add the `clientToken` parameter.

Example: CreateService

The following CreateService API request uses idempotency as it includes a client token.

```
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": "MyService",
  "taskDefinition": "MyTaskDefinition:2",
  "desiredCount": 10,
   "capacityProviderStrategy": [
      {
         "base": "number",
         "capacityProvider": "FARGATE",
         "weight": 1
      }
   ],
   "capacityProviderStrategy": [
      {
         "base": "number",
         "capacityProvider": "FARGATE_SPOT",
         "weight": 1
      }
   ],
   "clientToken": "550e8400-e29b-41d4-a716-44665544"
}
```

Example: CreateTaskSet

The following CreateTaskSet API request uses idempotency as it includes a client token.

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 87
X-Amz-Target: AmazonEC2ContainerServiceV20141113.CreateTaskSet
X-Amz-Date: 20150429T170125Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
```

```
{
  "serviceName": "MyService",
  "taskDefinition": "mytask:1",
  "desiredCount": 1,
   "capacityProviderStrategy": [
      {
         "base": "number",
         "capacityProvider": "FARGATE",
         "weight": 1
      }
   ],
   "capacityProviderStrategy": [
      {
         "base": "number",
         "capacityProvider": "FARGATE_SPOT",
         "weight": 1
      }
   ],
    "clientToken": "550e8400-e29b-41d4-a716-44665544"
}
```

Example: RunTask

The following RunTask API request uses idempotency as it includes a client token.

```
POST / HTTP/1.1
Host: ecs.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 45
X-Amz-Target: AmazonEC2ContainerServiceV20141113.RunTask
X-Amz-Date: 20161121T215740Z
User-Agent: aws-cli/1.11.13 Python/2.7.12 Darwin/16.1.0 botocore/1.4.66
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "count": 1,
  "taskDefinition": "mytask:1",
  "clientToken": "550e8400-e29b-41d4-a716-446655440000"
}
```

Retry recommendations for idempotent requests

The following table shows some common responses that you might get for idempotent API requests, and provides retry recommendations.

Response	Recommendation	Comments
200 (OK)	Do not retry	The original request completed successfully. Any subsequent retries return successfully.
400-series response codes (client errors)	Do not retry	There is a problem with the request, from among the following: • It includes a parameter or parameter combination that is not valid. • It uses an action or resource for which you do not have permissions. • It uses a resource that is in the process of changing states. If the request involves a resource that is in the process of changing states, retrying the request could possibly succeed.
500-series response codes (server errors)	Retry	The error is caused by an AWS server-side issue and is generally transient. Repeat the request with an appropriate back-off strategy.

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

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 Create a signed AWS API request in the *IAM User Guide*.

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 Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, 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 Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400