

Creating a Cluster

You can create a Amazon ECS cluster using the AWS Management Console, as described in this topic. Before you begin, be sure that you've completed the steps in [Setting Up with Amazon ECS](#). After you've created your cluster, you can register container instances into it and run tasks and services.

To create a cluster

1. Open the Amazon ECS console at <https://console.aws.amazon.com/ecs/>.
2. From the navigation bar, select the region to use.

Note

Amazon ECS is available in the following regions:

Region Name	Region
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1
EU (Frankfurt)	eu-central-1
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

3. In the navigation pane, choose **Clusters**.
4. On the **Clusters** page, select **Create Cluster**.
5. In the **Cluster name** field, enter a name for your cluster. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
6. Choose **Create** to create your cluster.

Launching an Amazon ECS Container Instance

You can launch an Amazon ECS container instance using the AWS Management Console, as described in this topic. Before you begin, be sure that you've completed the steps in [Setting Up with Amazon ECS](#). After you've launched your instance, you can use it to run tasks.

To launch a container instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation bar, select the region to use.

Note

Amazon ECS is available in the following regions:

Region Name	Region
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1
EU (Frankfurt)	eu-central-1
Asia Pacific (Tokyo)	ap-northeast-1
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

3. From the console dashboard, choose **Launch Instance**.
4. On the **Choose an Amazon Machine Image (AMI)** page, choose **Community AMIs**.
5. Choose an AMI for your container instance. You can choose the Amazon ECS-optimized AMI, or another operating system, such as CoreOS or Ubuntu. If you do not choose the Amazon ECS-optimized AMI, you need to follow the procedures in [Installing the Amazon ECS Container Agent](#).

Note

For Amazon ECS-specific CoreOS installation instructions, see <https://coreos.com/docs/running-coreos/cloud-providers/ecs/>.

To use the Amazon ECS-optimized AMI, type **amazon-ecs-optimized** in the **Search community AMIs** field and press the **Enter** key. Choose **Select** next to the **amzn-ami-2016.03.g-amazon-ecs-optimized** AMI. The current Amazon ECS-optimized AMI IDs by region are listed below for reference.

Region	AMI ID
us-east-1	ami-52cd5445
us-west-1	ami-efale28f
us-west-2	ami-a426edc4
eu-west-1	ami-7b244e08
eu-central-1	ami-721aec1d
ap-northeast-1	ami-058a4964
ap-southeast-1	ami-0d9f466e
ap-southeast-2	ami-7df2c61e

6. On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. The `t2.micro` instance type is selected by default. The instance type that you select determines the resources available for your tasks to run on.
7. Choose **Next: Configure Instance Details**.
8. On the **Configure Instance Details** page, set the **Auto-assign Public IP** field depending on whether or not you want your instance to be accessible from the public Internet. If your instance should be accessible from the Internet, verify that the **Auto-assign Public IP** field is set to **Enable**. If your instance should not be accessible from the Internet, set this field to **Disable**.
9. On the **Configure Instance Details** page, select the `ecsInstanceRole` **IAM role** value that you created for your container instances in [Setting Up with Amazon ECS](#).

Important

If you do not launch your container instance with the proper IAM permissions, your Amazon ECS agent will not connect to your cluster. For more information, see [Amazon ECS Container Instance IAM Role](#).

10. (Optional) Configure your Amazon ECS container instance with user data, such as the agent environment variables from [Amazon ECS Container Agent Configuration](#); Amazon EC2 user data scripts are executed only once, when the instance is first launched.

By default, your container instance launches into your default cluster. If you want to launch into your own cluster instead of the default, choose the **Advanced Details** list and paste the following script into the **User data** field, replacing *your_cluster_name* with the name of your cluster.

```
#!/bin/bash
echo ECS_CLUSTER=your_cluster_name >> /etc/ecs/ecs.config
```

Or, if you have an `ecs.config` file in Amazon S3 and have enabled Amazon S3 read-only access to your container instance role, choose the **Advanced Details** list and paste the following script into the **User data** field, replacing *your_bucket_name* with the name of your bucket to install the AWS CLI and write your configuration file at launch time.

Note

For more information about this configuration, see [Storing Container Instance Configuration in Amazon S3](#).

```
#!/bin/bash
yum install -y aws-cli
aws s3 cp s3://your_bucket_name/ecs.config /etc/ecs/ecs.config
```

11. Choose **Next: Add Storage**.
12. On the **Add Storage** page, configure the storage for your container instance.

If you are using an Amazon ECS-optimized AMI prior to the **2015.09.d** version, your instance has a single volume that is shared by the operating system and Docker.

If you are using the **2015.09.d** or later Amazon ECS-optimized AMI, your instance has two volumes configured. The **Root** volume is for the operating system's use, and the second Amazon EBS volume (attached to `/dev/xvdcz`) is for Docker's use.

You can optionally increase or decrease the volume sizes for your instance to meet your application needs.

13. Choose **Review and Launch**.

14. On the **Review Instance Launch** page, under **Security Groups**, you'll see that the wizard created and selected a security group for you. Instead, select the security group that you created in [Setting Up with Amazon ECS](#) using the following steps:

- a. Choose **Edit security groups**.
- b. On the **Configure Security Group** page, ensure that the **Select an existing security group** option is selected.
- c. Select the security group you created for your container instance from the list of existing security groups, and choose **Review and Launch**.

15. On the **Review Instance Launch** page, choose **Launch**.

16. In the **Select an existing key pair or create a new key pair** dialog box, choose **Choose an existing key pair**, then select the key pair that you created when getting set up.

When you are ready, select the acknowledgment field, and then choose **Launch Instances**.

17. A confirmation page lets you know that your instance is launching. Choose **View Instances** to close the confirmation page and return to the console.

18. On the **Instances** screen, you can view the status of your instance. It takes a short time for an instance to launch. When you launch an instance, its initial state is `pending`. After the instance starts, its state changes to `running`, and it receives a public DNS name. (If the **Public DNS** column is hidden, choose the **Show/Hide** icon and select **Public DNS**.)

Creating a Task Definition

Before you can run Docker containers on Amazon ECS, you need to create a task definition.

To create a new task definition

1. Open the Amazon ECS console at <https://console.aws.amazon.com/ecs/>.
2. From the navigation bar, choose the region to register your task definition in.
3. In the navigation pane, choose **Task Definitions**.
4. On the **Task Definitions** page, select **Create new Task Definition**.
5. (Optional) If you have a JSON representation of your task definition that you would like to use, complete the following steps:
 - a. On the **Create a Task Definition** page, scroll to the bottom of the page and choose **Configure via JSON**.
 - b. Paste your task definition JSON into the text area and choose **Save**.
 - c. Verify your information and select **Create**.
6. In the **Task Definition Name** field, enter a name for your task definition. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
7. (Optional) In the **Task Role** field, choose an IAM role that provides permissions for containers in your task to make calls to AWS APIs on your behalf. For more information, see [IAM Roles for Tasks](#).

Note

Only roles that have the **Amazon EC2 Container Service Task Role** trust relationship are shown here. For help creating an IAM role for your tasks, see [Creating an IAM Role and Policy for your Tasks](#).

8. (Optional) In the **Network Mode** field, choose the Docker network mode that you would like to use for the containers in your task. The available network modes correspond to those described in [Network settings](#) in the Docker run reference.

The default Docker network mode is `bridge`. If the network mode is set to `none`, you cannot specify port mappings in your container definitions, and the task's containers do not have external connectivity. The `hostnetwork` mode offers the highest networking performance for containers because they use the host network stack instead of the virtualized network stack provided by the `bridge` mode; however, exposed container

ports are mapped directly to the corresponding host port, so you cannot take advantage of dynamic host port mappings or run multiple instantiations of the same task on a single container instance if port mappings are used.

9. For each container in your task definition, complete the following steps.
 - a. Choose **Add Container Definition**.
 - b. Fill out each required field and any optional fields to use in your container definitions (more container definition parameters are available in the **Advanced container configuration** menu). For more information, see [Task Definition Parameters](#).
 - c. Select **Add** to add your container to the task definition.
10. (Optional) To define data volumes for your task, choose **Add volume**. For more information, see [Using Data Volumes in Tasks](#).
 - a. In the **Name** field, enter a name for your volume. Up to 255 letters (uppercase and lowercase), numbers, hyphens, and underscores are allowed.
 - b. (Optional) In the **Source Path** field, enter the path on the host container instance to present to the container. If this you leave this field empty, then the Docker daemon assigns a host path for you. If you specify a source path, then the data volume persists at the specified location on the host container instance until you delete it manually. If the source path does not exist on the host container instance, the Docker daemon creates it. If the location does exist, the contents of the source path folder are exported to the container.
11. Choose **Create** to finish.

Configuring Basic Service Parameters

All services require some basic configuration parameters that define the service, such as the task definition to use, which cluster the service should run on, how many tasks should be placed for the service, and so on; this is called the service definition. For more information about the parameters defined in a service definition, see [Service Definition Parameters](#).

This section covers creating a service with the basic service definition parameters that are required; when you have configured these parameters, you can create your service or move on to the next sections for optional service definition configuration, such as configuring your service to use a load balancer.

To configure the basic service definition parameters

1. Open the Amazon ECS console at <https://console.aws.amazon.com/ecs/>.
2. On the navigation bar, select the region that your cluster is in.
3. In the navigation pane, choose **Task Definitions**.
4. On the **Task Definitions** page, choose the name of the task definition from which to create your service.
5. On the **Task Definition name** page, choose the revision of the task definition from which to create your service.
6. Review the task definition, and choose **Create Service**.
7. On the **Create Service** page, for **Cluster**, choose the cluster in which to create your service.
8. For **Service name**, enter a unique name for your service.
9. For **Number of tasks**, enter the number of tasks to launch and maintain on your cluster.

Note

If your task definition uses static host port mappings on your container instances, then you need at least one container instance with the specified port available in your cluster for each task in your service. This restriction does not apply if your task definition uses dynamic host port mappings. For more information, see `portMappings` in the [Task Definition Parameters](#) topic.

10. (Optional) You can specify deployment parameters that control how many tasks run during the deployment and the ordering of stopping and starting tasks.
- **Minimum healthy percent:** Specify a lower limit on the number of running tasks during a deployment, evaluated as a percentage of the service's desired number of tasks. For example, if your service has a desired number of four tasks, a minimum healthy percent of 50% allows the scheduler to stop two existing tasks before starting two new tasks. Tasks for services that do not use a load balancer are considered healthy if they are in the `RUNNING` state; tasks for services that do use a load balancer are considered healthy if they are in the `RUNNING` state and the container instance it is hosted on is reported as healthy by the load balancer. The default value for minimum healthy percent is 50% in the console, and 100% with the AWS CLI or SDKs.
 - **Maximum percent:** Specify an upper limit on the number of running tasks during a deployment, evaluated as a percentage of the service's desired number of tasks. For example, if your service has a desired number of four tasks, a maximum percent value of 200% starts 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%.
11. If you do not want to run your service behind a load balancer or configure your service to use Service Auto Scaling, then you can proceed to [Review and Create Your Service](#). Otherwise, proceed to the next sections.

(Optional) Configuring Your Service to Use a Load Balancer

If you have an available Elastic Load Balancing load balancer configured, you can attach it to your service with the following procedures, or you can configure a new load balancer in the Amazon EC2 console; for more information see [Creating a Load Balancer](#).

Note

You must create your Elastic Load Balancing load balancer resources prior to following these procedures.

First, you must choose the load balancer type to use with your service. Then you can configure your service to work with the load balancer.

To choose a load balancer type

1. If you have not done so already, follow the basic service creation procedures in [Configuring Basic Service Parameters](#).
2. On the **Create Service** page, choose **Configure ELB**.
3. Choose the load balancer type to use with your service:

Application Load Balancer

Allows containers to use dynamic host port mapping (multiple tasks allowed per container instance). Multiple services can use the same listener port on a single load balancer with rule-based routing and paths.

Classic Load Balancer

Requires static host port mappings (only one task allowed per container instance); rule-based routing and paths are not supported.

We recommend that you use Application Load Balancers for your Amazon ECS services so that you can take advantage of the advanced features available to them.

4. For **Select IAM role for service**, choose **Create new role** to create a new role for your service, or select an existing IAM role to use for your service (by default, this is `ecsServiceRole`).

Important

If you choose to use an existing `ecsServiceRole` IAM role, you must verify that the role has the proper permissions to use Application Load Balancers and Classic Load Balancers, as shown in [Amazon ECS Service Scheduler IAM Role](#).

5. For **ELB Name**, choose the name of the load balancer to use with your service. Only load balancers that correspond to the load balancer type you selected earlier are visible here.
6. The following steps differ based on the load balancer type for your service. If you've chosen an Application Load Balancer, follow the steps in [To configure an Application Load Balancer](#). If you've chosen a Classic Load Balancer, follow the steps in [To configure a Classic Load Balancer](#).

To configure an Application Load Balancer

1. For **Select a Container**, choose the container and port combination from your task definition that your load balancer should distribute traffic to, and choose **Add to ELB**.
2. For **Listener port**, choose the listener port and protocol of the listener that you created in [Creating an Application Load Balancer](#) (if applicable), or choose **create new** to create a new listener and then enter a port number and choose a port protocol in **Listener protocol**.
3. For **Target group name**, choose the target group that you created in [Creating an Application Load Balancer](#) (if applicable), or choose **create new** to create a new target group.
4. (Optional) If you chose to create a new target group, complete the following fields as follows:
 - For **Target group name**, enter a name for your target group.
 - For **Target group protocol**, enter the protocol to use for routing traffic to your tasks.
 - For **Path pattern**, if your listener does not have any existing rules, the default path pattern (/) is used. If your listener already has a default rule, then you must enter a path pattern that matches traffic that you want to have sent to your service's target group. For example, if your service is a web application called `web-app`, and you want traffic that matches `http://my-elb-url/web-app` to route to your service, then you would enter `/web-app*` as your path pattern. For more information, see [Listeners for Your Application Load Balancers](#) in the *Application Load Balancer Guide*.
 - For **Health check path**, enter the path to which the load balancer should send health check pings.
5. When you are finished configuring your Application Load Balancer, choose **Save** to save your configuration and proceed to [Review and Create Your Service](#).

To configure a Classic Load Balancer

1. The **Health check port**, **Health check protocol**, and **Health check path** fields are all pre-populated with the values you configured in [Creating a Classic Load Balancer](#) (if applicable). You can update these settings in the Amazon EC2 console.
2. For **Container for ELB health check**, choose the container to send health checks.
3. When you are finished configuring your Classic Load Balancer, choose **Save** to save your configuration and proceed to [Review and Create Your Service](#).

(Optional) Configuring Your Service to Use Service Auto Scaling

Your Amazon ECS service can optionally be configured to use Auto Scaling to adjust its desired count up or down in response to CloudWatch alarms. For more information see [Service Auto Scaling](#).

Note

Service Auto Scaling is available in the following regions:

Region Name	Region
US East (N. Virginia)	us-east-1
US West (Oregon)	us-west-2
EU (Ireland)	eu-west-1

To configure basic Service Auto Scaling parameters

1. If you have not done so already, follow the basic service creation procedures in [Configuring Basic Service Parameters](#).
2. On the **Create Service** page, choose **Configure Service Auto Scaling**.
3. On the **Service Auto Scaling** page, select **Configure Service Auto Scaling to adjust your service's desired count**.
4. For **Minimum number of tasks**, enter the lower limit of the number of tasks for Service Auto Scaling to use. Your service's desired count will not be automatically adjusted below this amount.
5. For **Desired number of tasks**, this field is pre-populated with the value you entered earlier. You can change your service's desired count at this time, but this value must be between the minimum and maximum number of tasks specified on this page.
6. For **Maximum number of tasks**, enter the upper limit of the number of tasks for Service Auto Scaling to use. Your service's desired count will not be automatically adjusted above this amount.
7. For **IAM role for Service Auto Scaling**, choose an IAM role to authorize the Application Auto Scaling service to adjust your service's desired count on your behalf. If you have not previously created such a role, choose **Create new role** and the role will be created for you. For future reference, the role that is created for you is

called `ecsAutoscaleRole`. For more information, see [Amazon ECS Service Auto Scaling IAM Role](#).

To configure scaling policies for your service

These steps will help you create scaling policies and CloudWatch alarms that can be used to trigger scaling activities for your service. You can create a **Scale out** alarm to increase the desired count of your service, and a **Scale in** alarm to decrease the desired count of your service.

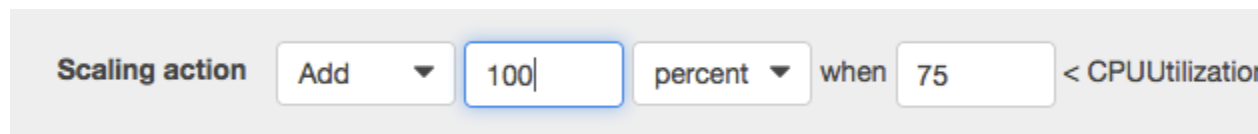
1. For **Policy name**, enter a descriptive name for your policy, or use the default policy name that is already entered.
2. For **Execute policy when**, select the CloudWatch alarm that you want to use to scale your service up or down.

You can use an existing CloudWatch alarm that you have previously created, or you can choose to create a new alarm. The **Create new alarm** workflow allows you to create CloudWatch alarms that are based on the `CPUUtilization` and `MemoryUtilization` of the service that you are creating. To use other metrics, you can create your alarm in the CloudWatch console and then return to this wizard to choose that alarm.

3. (Optional) If you've chosen to create a new alarm, complete the following steps.
 - a. For **Alarm name**, enter a descriptive name for your alarm. For example, if your alarm should trigger when your service CPU utilization exceeds 75%, you could call the alarm `service_name-cpu-gt-75`.
 - b. For **ECS service metric**, choose the service metric to use for your alarm. For more information about these service utilization metrics, see [Service Utilization](#).
 - c. For **Alarm threshold**, enter the following information to configure your alarm:
 - Choose the CloudWatch statistic for your alarm (the default value of **Average** works in many cases). For more information, see [Statistics](#) in the *Amazon CloudWatch Developer Guide*.
 - Choose the comparison operator for your alarm and enter the value that the comparison operator checks against (for example, `>` and `75`).
 - Enter the number of consecutive periods before the alarm is triggered and the period length. For example, a 2 consecutive periods of 5 minutes would take 10 minutes before the alarm triggered. Because your Amazon ECS tasks can scale up and down quickly, you should consider using a

low number of consecutive periods and a short period duration to react to alarms as soon as possible.

- d. Choose **Save** to save your alarm.
4. For **Scaling action**, enter the following information to configure how your service responds to the alarm:
 - Choose whether to add to, subtract from, or set a specific desired count for your service.
 - If you chose to add or subtract tasks, enter the number of tasks (or percent of existing tasks) to add or subtract when the scaling action is triggered. If you chose to set the desired count, enter the desired count that your service should be set to when the scaling action is triggered.
 - (Optional) If you chose to add or subtract tasks, choose whether the previous value is used as an integer or a percent value of the existing desired count.
 - Enter the lower boundary of your step scaling adjustment. By default, for your first scaling action, this value is the metric amount where your alarm is triggered. For example, the following scaling action adds 100% of the existing desired count when the CPU utilization is greater than 75%.



Scaling action Add ▼ 100 percent ▼ when 75 < CPUUtilization

5. (Optional) You can repeat [Step 4](#) to configure multiple scaling actions for a single alarm (for example, to add 1 task if CPU utilization is between 75-85%, and to add 2 tasks if CPU utilization is greater than 85%).
6. (Optional) If you chose to add or subtract a percentage of the existing desired count, enter a minimum increment value for **Add tasks in increments of *n* task(s)**.
7. For **Cooldown period**, enter the number of seconds between scaling actions.
8. Repeat [Step 1](#) through [Step 7](#) for the **Scale in** policy and choose **Save** to save your Service Auto Scaling configuration.

Review and Create Your Service

After you have configured your basic service definition parameters and optionally configured your service to use a load balancer, you can review your configuration and then choose **Create Service** to finish creating your service.

Note

After you create a service, the target group ARN or load balancer name, container name, and container port specified in the service definition are immutable. You cannot add, remove, or change the load balancer configuration of an existing service. If you update the task definition for the service, the container name and container port that were specified when the service was created must remain in the task definition