Learn how to create an Amazon ECS Linux task for the Fargate launch type

On this page

Related resources

Amazon ECS API Reference (https://docs.aws.amazon.com/AmazonECS/latest/APIReference/index.html)

AWS CLI commands for Amazon ECS (https://docs.aws.amazon.com/cli/latest/reference/ecs/)

SDKs & Tools (https://aws.amazon.com/tools/)

Recently added to this guide

Amazon Elastic Container Service (Amazon ECS) is a highly scalable, fast, container management service that makes it easy to run, stop, and manage your containers. You your containers on a serverless infrastructure that is managed by Amazon ECS by laun your services or tasks on AWS Fargate. For more information on Fargate, see AWS Farg Amazon ECS (./AWS_Fargate.html) .

Get started with Amazon ECS on AWS Fargate by using the Fargate launch type for yo in the Regions where Amazon ECS supports AWS Fargate.

- Troubleshooting (troubleshooting.html)
- Security (security.html)
- Tutorials (ecs-tutorials.html)

Complete the following steps to get started with Amazon ECS on AWS Fargate.

Prerequisites

Before you begin, complete the steps in Set up to use Amazon ECS (./get-set-up-for-ama ecs.html) and that your AWS user has the permissions specified in the `Administrator` IAM policy example.

The console attempts to automatically create the task execution IAM role, which is req Fargate tasks. To ensure that the console is able to create this IAM role, one of the foll must be true:

- Your user has administrator access. For more information, see Set up to use Amaz (./get-set-up-for-amazon-ecs.html) .
- Your user has the IAM permissions to create a service role. For more information, s Creating a Role to Delegate Permissions to an AWS Service (https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-service.html) .
- A user with administrator access has manually created the task execution role so t available on the account to be used. For more information, see Amazon ECS task e IAM role (./task_execution_IAM_role.html) .

> **Important**
>
> The security group you select when creating a service with your task definition have port 80 open for inbound traffic. Add the following inbound rule to your security group. For information about how to create a security group, see Creat security group for your Amazon EC2 instance (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-security-group.html the *Amazon EC2 User Guide*.
>
> - Type: HTTP
> - Protocol: TCP
> - Port range: 80
> - Source: Anywhere (`0.0.0.0/0`)

Step 1: Create the cluster

Create a cluster that uses the default VPC.

Before you begin, assign the appropriate IAM permission. For more information, see A ECS cluster examples (./security_iam_id-based-policy-examples.html#IAM_cluster_policies) .

1. Open the console at https://console.aws.amazon.com/ecs/v2 (https://console.aws.amazon.com/ecs/v2) .
2. From the navigation bar, select the Region to use.
3. In the navigation pane, choose **Clusters**.
4. On the **Clusters** page, choose **Create cluster**.
5. Under **Cluster configuration**, for **Cluster name**, enter a unique name.

   The name can contain up to 255 letters (uppercase and lowercase), numbers, and

6. (Optional) To turn on Container Insights, expand **Monitoring**, and then turn on **Us Container Insights**.
7. (Optional) To help identify your cluster, expand **Tags**, and then configure your tag

   [Add a tag] Choose **Add tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose **Remove** to the right of the tag's Key and Value.
8. Choose **Create**.

Step 2: Create a task definition

A task definition is like a blueprint for your application. Each time you launch a task in ECS, you specify a task definition. The service then knows which Docker image to use f containers, how many containers to use in the task, and the resource allocation for eac container.

1. In the navigation pane, choose **Task Definitions**.
2. Choose **Create new Task Definition**, **Create new revision with JSON**.
3. Copy and paste the following example task definition into the box and then choos

```
{
    "family": "sample-fargate",
    "networkMode": "awsvpc",
    "containerDefinitions": [
        {
            "name": "fargate-app",
            "image":
"public.ecr.aws/docker/library/httpd:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": [
                "sh",
                "-c"
            ],
            "command": [
                "/bin/sh -c \"echo '<html> <head>
<title>Amazon ECS Sample App</title> <style>body {margin-top
40px; background-color: #333;} </style> </head><body> <div
style=color:white;text-align:center> <h1>Amazon ECS Sample
App</h1> <h2>Congratulations!</h2> <p>Your application is no
running on a container in Amazon ECS.</p> </div></body>
</html>' >  /usr/local/apache2/htdocs/index.html && httpd-
foreground\""
            ]
        }
```

```
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "256",
    "memory": "512"
}
```

4. Choose **Create**.

Step 3: Create the service

Create a service using the task definition.

1. In the navigation pane, choose **Clusters**, and then select the cluster you created in Create the cluster (#get-started-fargate-cluster) .
2. From the **Services** tab, choose **Create**.
3. Under **Deployment configuration**, specify how your application is deployed.
   a. For **Task definition**, choose the task definition you created in Step 2: Create a definition (#get-started-fargate-task-def) .
   b. For **Service name**, enter a name for your service.
   c. For **Desired tasks**, enter **1**.
4. Under **Networking**, you can create a new security group or choose an existing secu group for your task. Ensure that the security group you use has the inbound rule li under Prerequisites (#first-run-linux-prereqs) .
5. Choose **Create**.

Step 4: View your service

1. Open the console at https://console.aws.amazon.com/ecs/v2 (https://console.aws.amazon.com/ecs/v2) .
2. In the navigation pane, choose **Clusters**.
3. Choose the cluster where you ran the service.
4. In the **Services** tab, under **Service name**, choose the service you created in Step 3: the service (#create-linux-service) .
5. Choose the **Tasks** tab, and then choose the task in your service.
6. On the task page, in the **Configuration** section, under **Public IP**, choose **Open add**

Step 5: Clean up

When you are finished using an Amazon ECS cluster, you should clean up the resources associated with it to avoid incurring charges for resources that you are not using.

Some Amazon ECS resources, such as tasks, services, clusters, and container instances, cleaned up using the Amazon ECS console. Other resources, such as Amazon EC2 insta Elastic Load Balancing load balancers, and Auto Scaling groups, must be cleaned up m in the Amazon EC2 console or by deleting the AWS CloudFormation stack that created

1. In the navigation pane, choose **Clusters**.
2. On the **Clusters** page, select the cluster you created for this tutorial.
3. Choose the **Services** tab.
4. Select the service, and then choose **Delete**.
5. At the confirmation prompt, enter **delete** and then choose **Delete**. Alternatively, y use the `Force delete` option to have Amazon ECS scale the service down on yo before deleting it.

   Wait until the service is deleted.
6. Choose **Delete Cluster**. At the confirmation prompt, enter **delete** *cluster-name* then choose **Delete**. Deleting the cluster cleans up the associated resources that w created with the cluster, including Auto Scaling groups, VPCs, or load balancers.

Related resources

Amazon ECS API Reference (https://docs.aws.amazon.com/AmazonECS/latest/APIReference/index.html)

AWS CLI commands for Amazon ECS (https://docs.aws.amazon.com/cli/latest/reference/ecs/)

SDKs & Tools (https://aws.amazon.com/tools/)

▶ **Recently added to this guide**

View related pages *Abstracts generated by AI*

1 2 3 4

AmazonCloudWatch › monitoring

Setting up Container Insights on Amazon ECS using AWS Distro for...

Task roles, execution roles created for AWS OpenTelemetry Collector on Amazon ECS to set up Container Insights, metrics visible in CloudWatch console

January 25, 2024

AmazonECS › developerguide

Learn how to create an Amazon EC Windows task for the Fargate laun

Create an Amazon ECS Windows task o Fargate. Register a Windows task defin create a cluster, and launch a service w your task definition.

October 26, 2024

Discover highly rated pages *Abstracts generated by AI*

1 2 3 4

AmazonECS › developerguide

What is Amazon Elastic Container Service?...

Amazon ECS manages containerized applications, deploys applications on containers, creates task definitions, runs tasks, monitors resource utilization, manages clusters.

December 21, 2024

AmazonECS › developerguide

Amazon ECS task definition param (https://docs.aws.amazon.com/Amazo

January 30, 2025