

# How to deploy SpringBoot WebAPI to AWS ECS

---

## 1. Create Spring Boot Web API application with VSCode

---

See this repo: [https://github.com/luiscoco/SpringBoot\\_Sample2-created-WebAPI-with-VSCode](https://github.com/luiscoco/SpringBoot_Sample2-created-WebAPI-with-VSCode)

We set, in the `application.properties` file, the application port to 80

```
# Server Configuration
server.port=80

# Logging
logging.level.org.springframework.web=INFO
logging.level.org.hibernate=ERROR

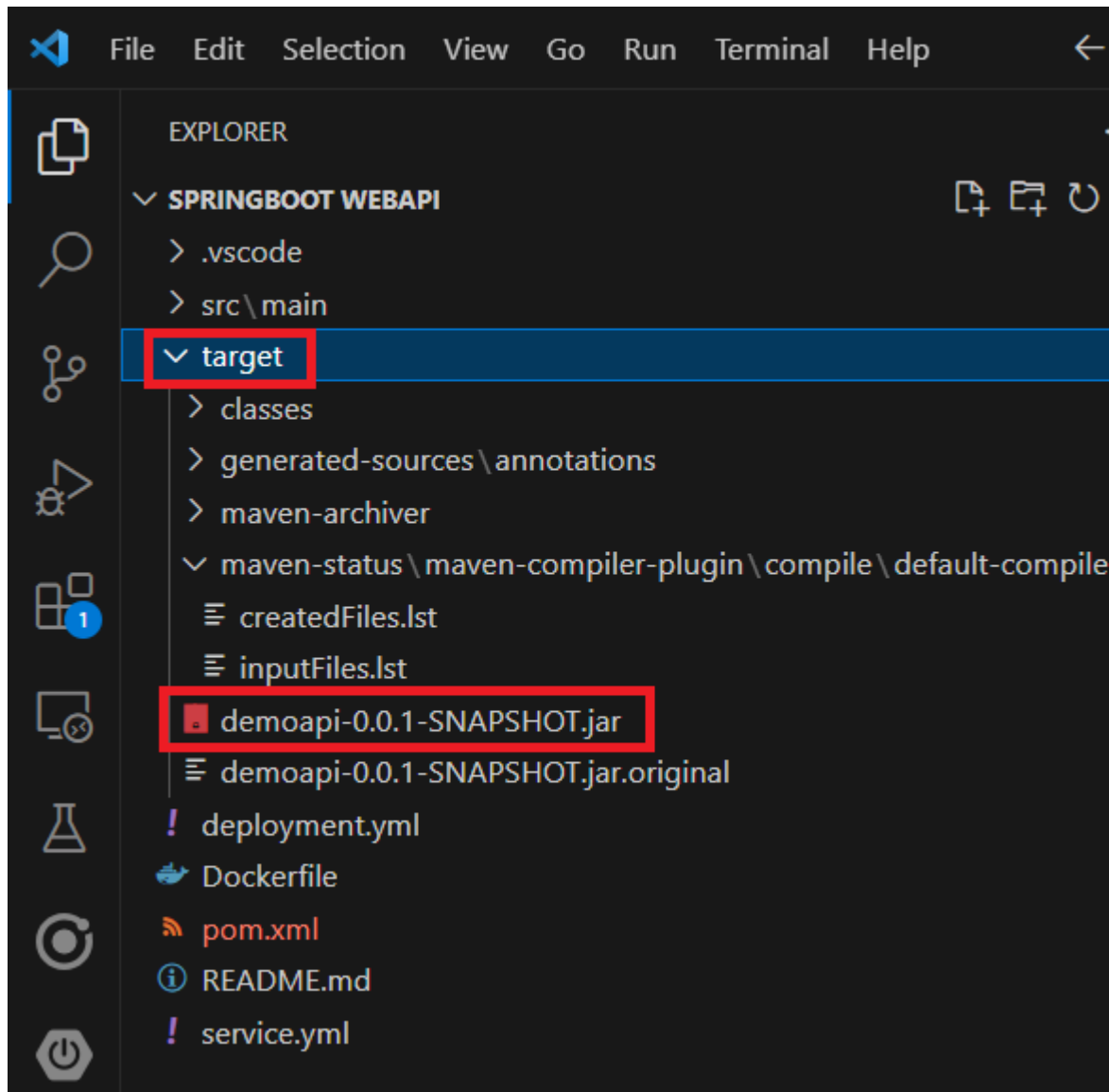
# Actuator Endpoints
management.endpoints.web.exposure.include=*
management.endpoint.health.show-details=always

# Actuator Info
info.app.name=My Spring Boot Application
info.app.description=A simple demo application
info.app.version=1.0.0
```

Before creating the docker image we should create the **JAR** file with the following command:

```
mvn clean install
```

We confirm we created the **target** folder



After creating the JAR file we run the application with this command:

```
java -jar .\target\demoapi-0.0.1-SNAPSHOT.jar
```

We verify the application endpoints:

<http://localhost:80/hello>

<http://localhost:80/actuator/health>

## 2. Create a Docker file

Create a Dockerfile and copy this content inside:

```
# Start with a base image containing Java runtime
FROM openjdk:11-jdk-slim as build

# Add Maintainer Info
LABEL maintainer="your_email@example.com"
```

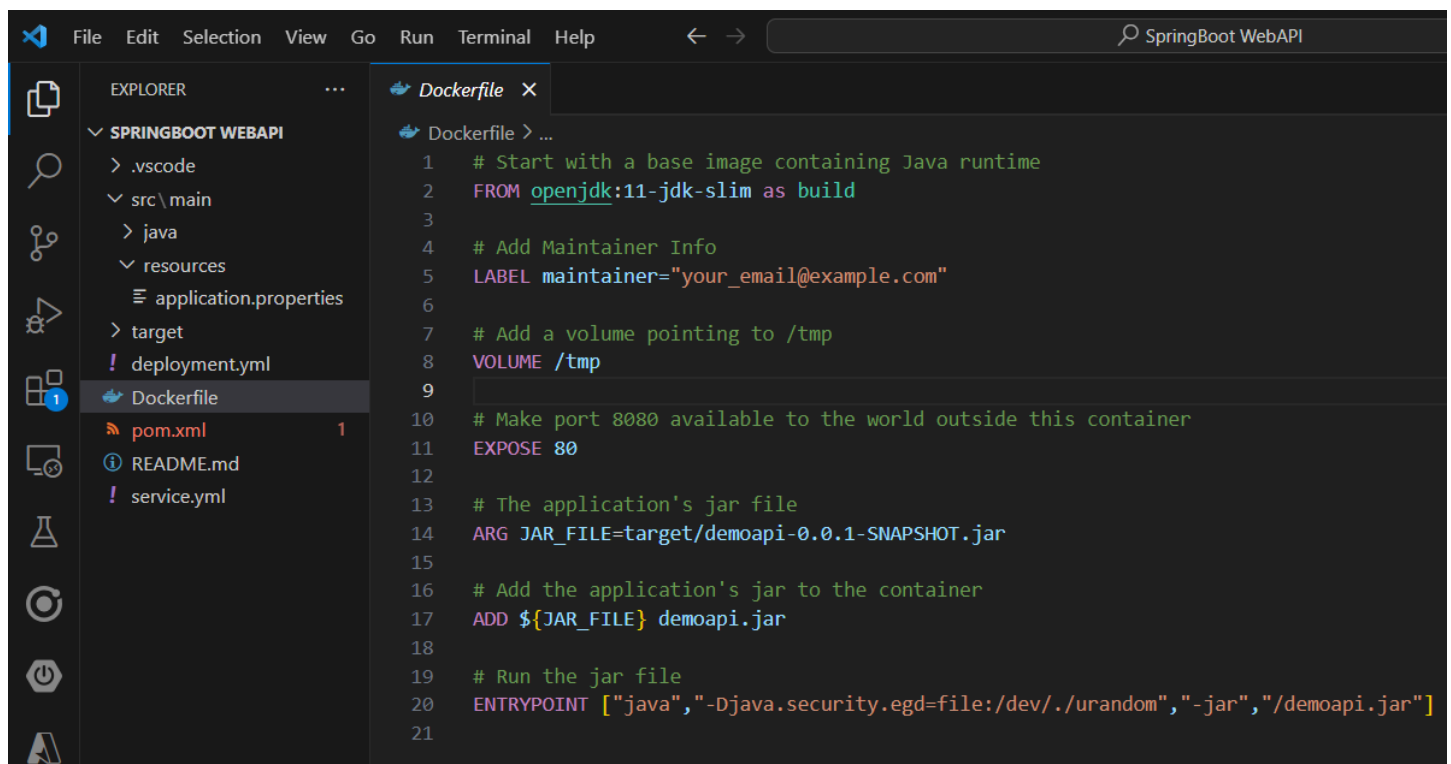
```
# Add a volume pointing to /tmp
VOLUME /tmp

# Make port 80 available to the world outside this container
EXPOSE 80

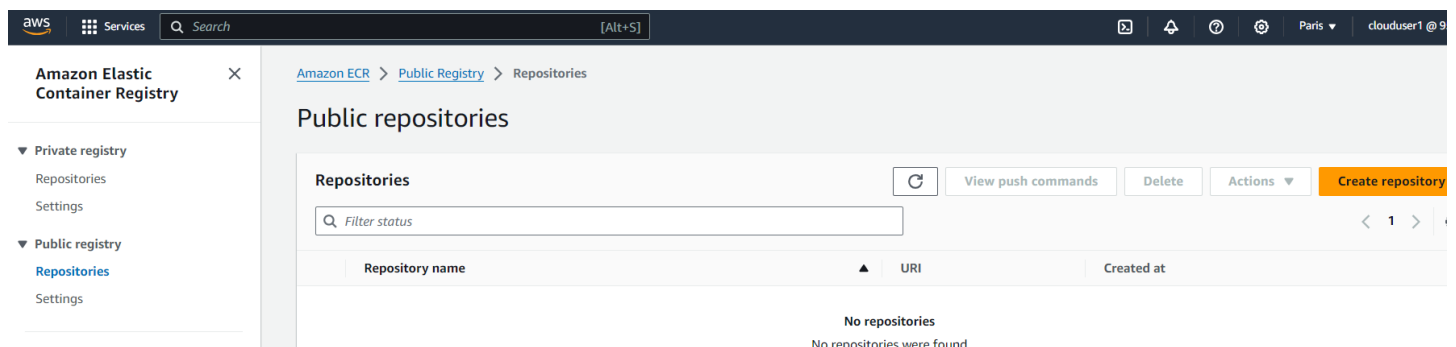
# The application's jar file
ARG JAR_FILE=target/demoapi-0.0.1-SNAPSHOT.jar


# Add the application's jar to the container
ADD ${JAR_FILE} demoapi.jar

# Run the jar file
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/demoapi.jar"]
```



### 3. Create a AWS Elastic Container Registry ECR Public repo



 Services  [Alt+S]

Amazon ECR > Private registry > Repositories > Create repository

## Create repository

### General settings


Visibility settings

[Info](#)

Choose the visibility setting for the repository.

☐ Private  
Access is managed by IAM and repository policy permissions.

☒ Public  
Publicly visible and accessible for image pulls.

 Once a repository is created, the visibility setting of the repository can't be changed.

### Detail


Repository name

[Info](#)

A namespace can be included with your repository name (e.g. namespace/repo-name).

public.ecr.aws/registry-alias/

10 out of 205 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

 Services  [Alt+S]

Repository logo - optional [Info](#)

Choose a local image file to use as the repository logo.

The supported file format is PNG. The supported image dimensions for both height and width should be a minimum of 60 pixels and a maximum of 2048 pixels. The maximum file size is 500 KB.

Short description - optional

The short description is displayed in search results and on the repository detail page.


0 out of 255 characters maximum.

Content types - optional

[Info](#)

Select the operating systems and system architectures that are compatible with the images in your repository.

Operating systems	Architectures
<input checked="" type="checkbox"/> Linux	<input checked="" type="checkbox"/> ARM
<input type="checkbox"/> Windows	<input checked="" type="checkbox"/> ARM 64
	<input checked="" type="checkbox"/> x86
	<input checked="" type="checkbox"/> x86-64

 Services  [Alt+S]

☰

Describe this repository

0 out of 10,240 characters maximum. Use GitHub Flavored Markdown format for the text. [Learn more](#)

Preview

**Usage - optional** [Info](#) [View example](#)

Provide detailed information about how to use the images in the repository. This provides context, support information, and additional usage details for users of the repository.


Usage information

0 out of 10,240 characters maximum. Use GitHub Flavored Markdown format for the text. [Learn more](#)

Preview

Cancel

Create repository

 Services  [Alt+S]

Amazon Elastic Container Registry

▼ Private registry

Repositories

Settings

▼ Public registry

**Repositories**

Settings

ECR public gallery

Amazon ECS

Created public repository

webapirepo has been successfully created in public registry

[Amazon ECR](#) > [Public Registry](#) > Repositories

Public repositories

Repositories (1)

View push commands

Delete

Actions

Create repository

	Repository name	URI	Created at
	webapirepo	public.ecr.aws/x7p6e5r6/webapirepo	December 29, 2023, 21:17:46 (UTC+01)

https://md2pdf.netlify.app

5/21

## Push commands for webapirepo



Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see [Getting Started with Amazon ECR](#).

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see [Registry Authentication](#).

1. Retrieve an authentication token and authenticate your Docker client to your registry.

Use the AWS CLI:

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/x7p6e5r6
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions [here](#). You can skip this step if your image is already built:

```
docker build -t webapirepo .
```

3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag webapirepo:latest public.ecr.aws/x7p6e5r6/webapirepo:latest
```

4. Run the following command to push this image to your newly created AWS repository:

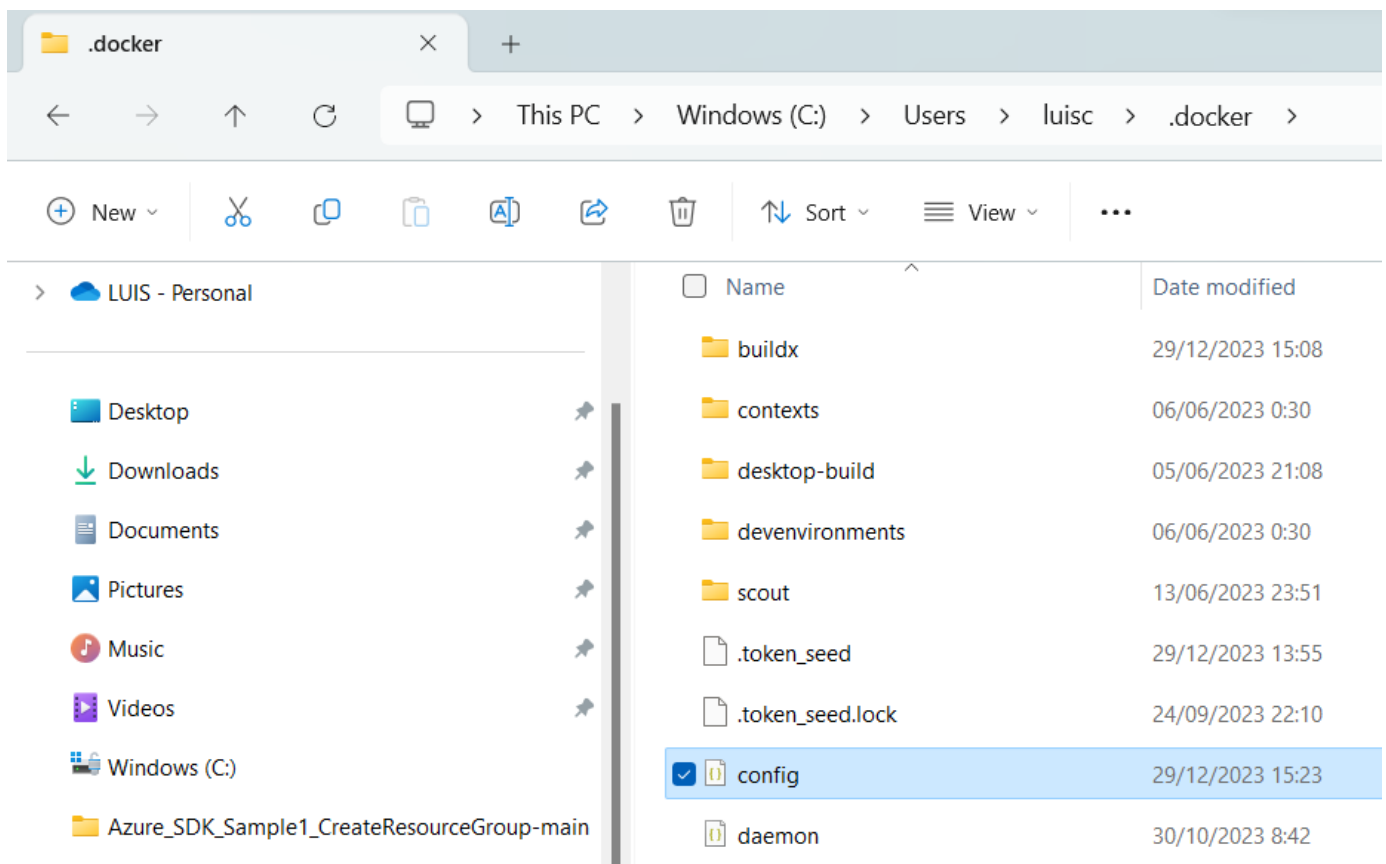
```
docker push public.ecr.aws/x7p6e5r6/webapirepo:latest
```

Close

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-
```

**IMPORTANT:** if you cannot enter in the AWS ECR follow these steps

- Delete **config.json** file in the following location:



- Type the command:

`docker login`

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE
PS C:\SpringBoot WebAPI> docker login
Authenticating with existing credentials...
Login Succeeded
```

- Delete the letter "s" the "credsStore": "desktop" the result is this word: "credStore": "desktop"

```
{ } config.json X
C: > Users > luisc > .docker > { } config.json > credsStore
1 {
2   "auths": {
3     "https://index.docker.io/v1/": {}
4   },
5   "credStore": "desktop",
6   "currentContext": "default"
7 }
```

```

config.json X
C: > Users > luisc > .docker > config.json > ...
1 {
2   "auths": {
3     "https://index.docker.io/v1/": {}
4   },
5   "credStore": "desktop",
6   "currentContext": "default"
7 }

```

- Configure AWS account with access key and secret key

The screenshot shows the AWS Management Console for the Amazon Elastic Container Registry (ECR). The main view is for the 'webapirepo' repository, which currently has 0 images. A sidebar on the right displays account information for 'clouduser1', including the account ID (9547-1817-7936) and IAM user (clouduser1). The sidebar also lists various account management options like 'Switch role' and 'Sign out'.

The screenshot shows the AWS IAM console 'Access keys' page. It displays 0 access keys and a 'Create access key' button. A message states: 'No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials.' The page also includes a 'Learn more' link for additional information.

The screenshot shows the AWS IAM console 'Create access key' page. It displays the 'Access key best practices & alternatives' section, which lists various use cases and recommended alternatives for using access keys. The page includes a 'Confirmation' section where the user must agree to the recommendations before creating the access key.



aws Services Search [Alt+S]

IAM > Security credentials > Create access key

Step 1  
Access key best practices & alternatives

Step 2 - optional  
**Set description tag**

Step 3  
Retrieve access keys

### Set description tag - optional [Info](#)

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value  
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

accessECRrepo

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: \_ . : / = + - @

Cancel Previous **Create access key**

aws Services Search [Alt+S]

IAM > Security credentials > Create access key

Step 1  
Access key best practices & alternatives

Step 2 - optional  
Set description tag

Step 3  
**Retrieve access keys**

### Retrieve access keys [Info](#)

**Access key**  
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIA54SNDJKIKDYXZ4MJ	***** <a href="#">Show</a>

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file Done

```
PS C:\Users\luisc\.docker> aws configure
AWS Access Key ID [*****24MJ]: 
AWS Secret Access Key [*****V7BM]: 
Default region name [us-east-1]: eu-west-3
Default output format [json]:
```

We try to enter now again in the AWS ECR with the following command:

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS AZURE powershell +

```
PS C:\Users\luisc\.docker> aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin public.ecr.aws/x7p6e5r6
WARNING! Your password will be stored unencrypted in C:\Users\luisc\.docker\config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- Now we can create the SpringBoot WebAPI Docker image with this command

```
docker build -t webapirepo .
```

- Rename the Docker image for pushing it to the AWE ECR repo

```
docker tag webapirepo:latest public.ecr.aws/x7p6e5r6/webapirepo:latest
```

- Also we can verify the application docker image running it in our local Docker Desktop with this command:

```
docker run -p 80:80 --name myapp-container webapirepo:latest
```

- Verify the application endpoints:

<http://localhost:80/hello>

<http://localhost:80/actuator/health>

## 4. Push the Docker image to AWS ECR

Push the Docker image to AWS ECR

```
docker push public.ecr.aws/x7p6e5r6/webapirepo:latest
```

```
PS C:\SpringBoot WebAPI> docker push public.ecr.aws/x7p6e5r6/webapirepo:latest
The push refers to repository [public.ecr.aws/x7p6e5r6/webapirepo]
632870e20190: Pushed
eb6ee5b9581f: Pushed
e3abdc2e9252: Pushed
eafe6e032dbd: Pushed
92a4e8a3140f: Pushed
latest: digest: sha256:bbc1283153d8ab75f0521d4bffa7b23a5e4d03feb5296ce703e097fcaa45952c6 size: 1372
PS C:\SpringBoot WebAPI>
```

The screenshot shows the Amazon ECR console interface. On the left, the 'Amazon Elastic Container Registry' sidebar is visible with options for Private and Public registries. The main panel displays the 'webapirepo' repository under the Public Registry. It shows a list of images with one entry: 'latest'. The image details include its digest (sha256:bbc1283153d8ab75f0521d4bffa7b23a5e4d03feb5296ce703e097fcaa45952c6) and size (252.56 MB). Buttons for 'View public listing' and 'View push commands' are present at the top right of the repository view.

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
latest	Image	December 30, 2023, 12:32:54 (UTC+01)	252.56	Copy URI	sha256:bbc1283153d8ab...

The screenshot shows the Amazon Elastic Container Registry (ECR) console. The breadcrumb navigation is: Amazon ECR > Public Registry > Repositories > webapirepo > sha256:bbc1283153d8ab75f0521d4bff7b23a5e4d03feb5296ce703e097fcaa45952c6. The main section is titled "Image". Under "Details", the "Image tags" section shows "latest". The "URI" is highlighted with a red box: `public.ecr.aws/x7p6e5r6/webapirepo:latest`. The "Digest" is `sha256:bbc1283153d8ab75f0521d4bff7b23a5e4d03feb5296ce703e097fcaa45952c6`. The "General information" section shows the "Artifact type" as `Image`, the "Repository" as `webapirepo`, and the "Pushed at" date as "December 30, 2023, 12:32:54 (UTC+01)". The "Size (MB)" is 252.56.

## 5. Deploy Docker image from AWS ECR to AWS ECS

The screenshot shows the AWS Services search results for "ECS". The search bar contains "ECS". The results are categorized into "Services (26)", "Features (62)", "Resources (New)", "Documentation (32,780)", "Knowledge Articles (790)", "Marketplace (352)", "Blogs (2,952)", "Events (46)", and "Tutorials (26)". The "Elastic Container Service" is highlighted with a red box. It is described as "Highly secure, reliable, and scalable way to run containers". The "Top features" section lists "Clusters", "Task definitions", "Account settings", "Get started", and "Namespaces". Other services shown include "Batch" and "AWS FIS".

### 5.1. Create a AWS ECS cluster

We press in the Create cluster button:

The screenshot shows the Amazon Elastic Container Service (ECS) console. The breadcrumb navigation is: Amazon Elastic Container Service > Clusters. The main section is titled "Clusters (0) Info". There is a search bar for "Search clusters". The "Create cluster" button is highlighted with a red box. Below the search bar, there are tabs for "Cluster", "Services", "Tasks", "Container instances", "CloudWatch monitoring", and "Capacity provider strategy". The "Cluster" tab is selected, showing "No clusters" and "No clusters to display".

We set the AWS Cluster name

The screenshot shows the AWS Management Console interface for creating a new ECS cluster. The left sidebar contains navigation links for 'Amazon Elastic Container Service', 'Clusters', 'Namespaces', 'Task definitions', 'Account settings', 'Install AWS Copilot', 'Amazon ECR', 'Repositories', 'AWS Batch', 'Documentation', 'Discover products', and 'Subscriptions'. The main content area is titled 'Create cluster' and includes a description of ECS clusters. Below this, the 'Cluster configuration' section contains a 'Cluster name' field with the value 'myspringbootcluster2' and a 'Default namespace - optional' dropdown menu also set to 'myspringbootcluster2'. The 'Infrastructure' section is expanded, showing the 'Serverless' option selected. Under 'Infrastructure', the 'AWS Fargate (serverless)' option is checked, while 'Amazon EC2 instances' and 'External instances using ECS Anywhere' are unchecked. The 'Monitoring - optional' and 'Tags - optional' sections are also visible at the bottom.

30/12/23, 15:20

How to deploy SpringBoot WebAPI to AWS ECS

aws Services Search [Alt+S]

Tell us what you think X

**Amazon Elastic Container Service**

Clusters

Namespaces

Task definitions

Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

**Create cluster** Info

An Amazon ECS cluster groups together tasks, and services, and allows for shared capacity and common configurations. All of your tasks, services, and capacity must belong to a cluster.

**Cluster configuration**

Cluster name

myspringbootcluster2

There can be a maximum of 255 characters. The valid characters are letters (uppercase and lowercase), numbers, hyphens, and underscores.

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

myspringbootcluster2 X

**Infrastructure** Info **Serverless**

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

☒ **AWS Fargate (serverless)**

Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

☐ **Amazon EC2 instances**

Manual configurations. Use for large workloads with consistent resource demands.

We leave the Fargate option checked and we press the **Create** button

This screenshot shows the bottom portion of the 'Create cluster' page. The 'Default namespace' dropdown remains at 'myspringbootcluster2'. The 'Infrastructure' section is still expanded, showing 'AWS Fargate (serverless)' as the selected option. Below this, the 'Monitoring - optional' section is expanded, indicating that 'Container Insights' is off by default. The 'Tags - optional' section is also expanded, stating that tags help identify and organize clusters. At the bottom right, there are 'Cancel' and 'Create' buttons.

aws Services Search [Alt+S]

Tell us what you think X

**Amazon Elastic Container Service**

Clusters

Namespaces

Task definitions

Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

Documentation

Discover products

Subscriptions

Default namespace - *optional*

Select the namespace to specify a group of services that make up your application. You can overwrite this value at the service level.

myspringbootcluster2 X

**Infrastructure** Info **Serverless**

Your cluster is automatically configured for AWS Fargate (serverless) with two capacity providers. Add Amazon EC2 instances, or external instances using ECS Anywhere.

☒ **AWS Fargate (serverless)**

Pay as you go. Use if you have tiny, batch, or burst workloads or for zero maintenance overhead. The cluster has Fargate and Fargate Spot capacity providers by default.

☐ **Amazon EC2 instances**

Manual configurations. Use for large workloads with consistent resource demands.

☐ **External instances using ECS Anywhere**

Manual configurations. Use to add data center compute.

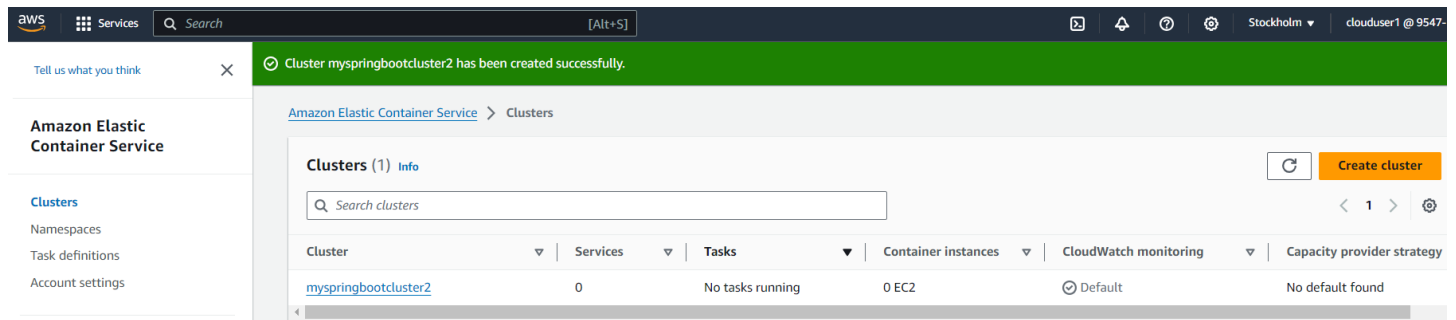
**Monitoring - optional** Info

Container Insights is off by default. When you use Container Insights, there is a cost associated with it.

**Tags - optional** Info

Tags help you to identify and organize your clusters.

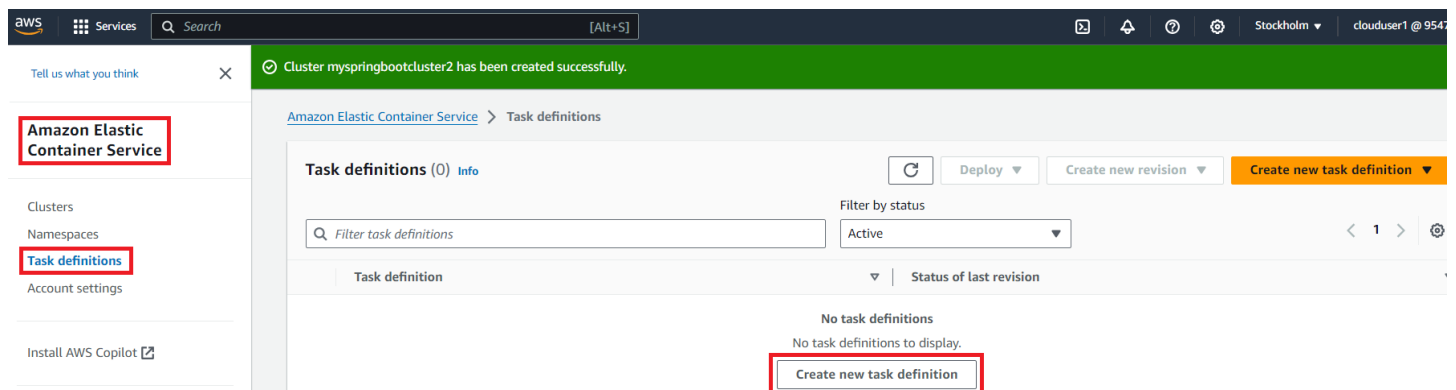
Cancel Create



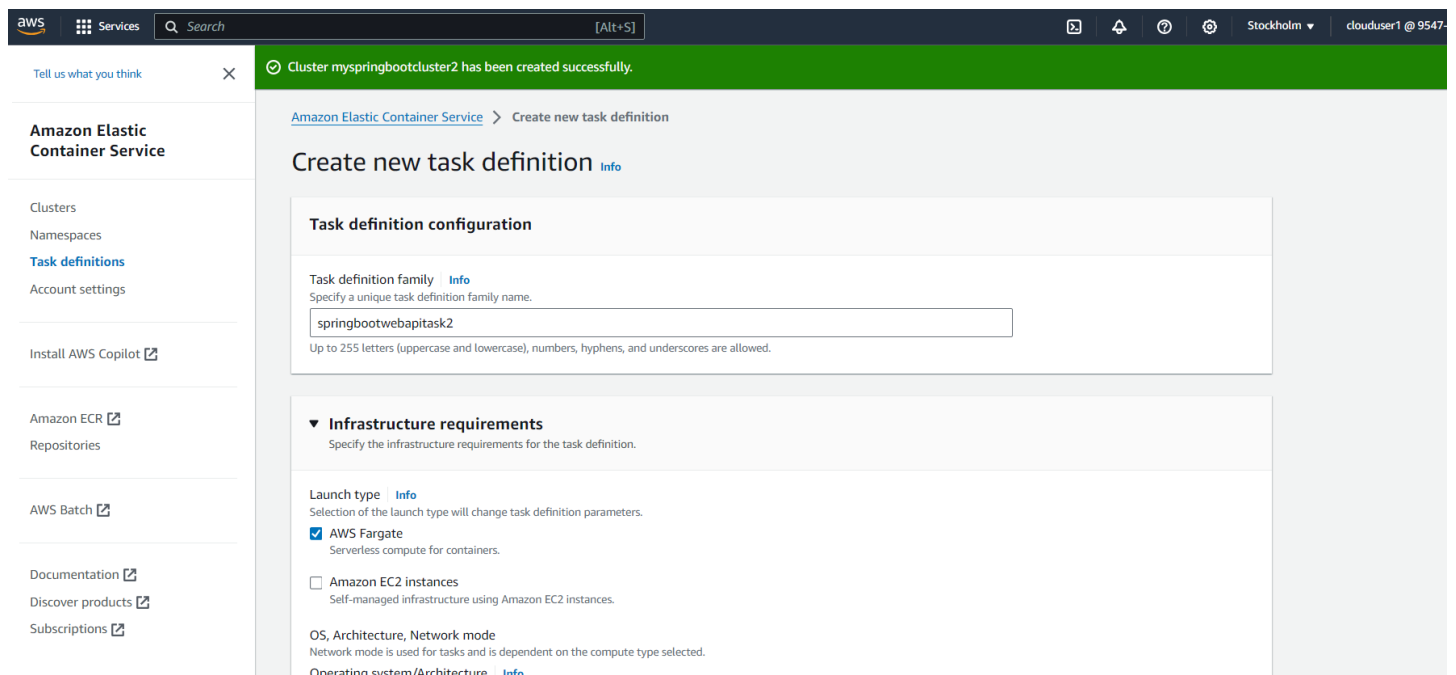
**Note:** we selected the eu-north-1 region but the more advisable solution is select eu-west-3 because it is the nearest to our location in Spain. But today I had some technical problems creating cluster in Paris region and I decided to select the Stockholm one.

## 5.2. Create a Task Definition

We press the Create new task definition button



We select the infrastructure option **Fargate**



We leave the rest of options as they are by default

The screenshot shows the 'Task definition' configuration page in the AWS Management Console. The left sidebar contains navigation links for Amazon Elastic Container Service, Clusters, Namespaces, Task definitions (selected), Account settings, Install AWS Copilot, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions.

The main configuration area includes the following settings:

- OS, Architecture, Network mode:** Network mode is used for tasks and is dependent on the compute type selected.
- Operating system/Architecture:** Linux/X86\_64
- Network mode:** awsvpc
- Task size:** Specify the amount of CPU and memory to reserve for your task.
  - CPU:** 1 vCPU
  - Memory:** 3 GB
- Task roles - conditional:**
  - Task role:** -
  - Task execution role:** ecsTaskExecutionRole
- Task placement - optional:** Task placement constraints are not supported for AWS Fargate launch type.

We have to set container image name

The screenshot shows the Amazon Elastic Container Registry (ECR) console. The left sidebar contains navigation links for Amazon Elastic Container Registry, Private registry, Public registry (selected), Repositories, Settings, Images, Gallery detail, and ECR public gallery.

The main area shows the 'webapirepo' repository. The 'Images (1)' section displays a table with the following data:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
latest	Image	December 30, 2023, 14:06:07 (UTC+01)	252.56	Copy URI	sha256:5a661dd0860426...

The screenshot shows the 'Image' details page in the Amazon Elastic Container Registry (ECR) console. The left sidebar contains navigation links for Amazon Elastic Container Registry, Private registry, Public registry (selected), Repositories, Settings, Images, Gallery detail, Permissions, Repository tags, Settings, ECR public gallery, Amazon ECS, Amazon EKS, Getting started, and Documentation.

The main area shows the 'Image' details for the 'latest' tag. The 'URI' field is highlighted with a red box, showing the image URI: `public.ecr.aws/x7p6e5r6/webapirepo:latest`. The 'Digest' field shows the digest: `sha256:5a661dd086042613b2c573ec501a0d7e79e2ad3b5d468cea81d3c076fd2b4fb4`.

The 'General information' section displays the following details:

- Artifact type:** Image
- Repository:** webapirepo
- Pushed at:** December 30, 2023, 14:06:07 (UTC+01)
- Size (MB):** 252.56

We set the container URL and the port mapping

**Container - 1** Info

Container details  
Specify a name, container image, and whether the container should be marked as essential. Each task definition must have at least one essential container.

Name	Image URI	Essential container
myspringbootcontainer	public.ecr.aws/x7p6e5r6/webapirepo:latest	Yes

Private registry Info  
Store credentials in Secrets Manager, and then use the credentials to reference images in private registries.

☐ Private registry authentication

Port mappings Info  
Add port mappings to allow the container to access ports on the host to send or receive traffic. For port name, a default will be assigned if left blank.

Container port	Protocol	Port name	App protocol
80	TCP	http	None

[Add port mapping](#)

Read only root file system Info  
When this parameter is turned on, the container is given read-only access to its root file system.

☐ Read only

Resource allocation limits - conditional Info  
Container-level CPU, GPU, and memory limits are different from task-level values. They define how much resources are allocated for the container. If container attempts to exceed the memory specified in hard limit, the container is terminated.

CPU	GPU	Memory hard limit	Memory soft limit
1	1	3	1

in vCPU in GB in GB in GB

We leave the rest of options as they are by default and we press the \*\*\*\*

Ephemeral storage Info  
The amount of ephemeral storage, in GiB, to allocate for the task. By default, your tasks hosted on AWS Fargate receive a minimum of 20 GiB of ephemeral storage.

Amount  
21

To specify a custom amount of ephemeral storage, specify a value between 21 GiB up to a maximum of 200 GiB.

Volumes Info  
Add one or more data volumes for your task to provide additional storage for the containers in the task. For each data volume, you should add a mount point to specify where to mount the data volume in the container.

[Add volume](#)

Volumes from Info  
Mount data volumes from another container.

[Add volume from](#)

► **Monitoring - optional**  
Configure your application trace and metric collection settings using the AWS Distro for OpenTelemetry integration.

► **Tags - optional** Info  
Tags help you to identify and organize your task definitions.

Cancel [Create](#)

Cluster myspringbootcluster2 has been created successfully.

Task definition successfully created  
springbootwebapitask2:1 has been successfully created. You can use this task definition to deploy a service or run a task. [Deploy](#)

Amazon Elastic Container Service > Task definitions

Task definitions (1) Info

Filter by status: Active

Task definition	Status of last revision
<a href="#">springbootwebapitask2</a>	ACTIVE

## 5.3. Create a Service

The screenshot shows the AWS Management Console for the 'myspringbootcluster2' cluster. The breadcrumb navigation at the top is 'Amazon Elastic Container Service > Clusters > myspringbootcluster2 > Services'. The left sidebar shows the 'Amazon Elastic Container Service' menu with 'Clusters' highlighted. The main content area displays the 'Cluster overview' for 'myspringbootcluster2'. The 'ARN' is 'arn:aws:ecs:eu-north-1:954718177936:cluster/myspringbootcluster2', the 'Status' is 'Active', and 'CloudWatch monitoring' is 'Default'. The 'Services' section shows a table with columns for 'Draining', 'Active', 'Pending', and 'Running'. The 'Services' tab is selected, showing a table with columns for 'Service name', 'ARN', 'Status', 'Service type', 'Deployments and tasks', 'Last deployment', and 'Task definition'. The 'Create' button is highlighted in the top right corner of the 'Services' section.

We select the launch type Fargate

The screenshot shows the 'Create service' page for the 'myspringbootcluster2' cluster. The breadcrumb navigation is 'Amazon Elastic Container Service > Clusters > myspringbootcluster2 > Create service'. The left sidebar shows the 'Amazon Elastic Container Service' menu with 'Clusters' highlighted. The main content area displays the 'Create' page for 'myspringbootcluster2'. The 'Environment' section shows 'AWS Fargate' selected. The 'Existing cluster' section shows 'myspringbootcluster2' selected. The 'Compute configuration (advanced)' section shows 'Launch type' selected. The 'Launch type' section shows 'FARGATE' selected. The 'Platform version' section shows 'LATEST' selected.



**Deployment configuration**

Application type [Info](#)  
Specify what type of application you want to run.

☒ **Service**  
Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.

☐ **Task**  
Launch a standalone task that runs and terminates. For example, a batch job.

Task definition  
Select an existing task definition. To create a new task definition, go to [Task definitions](#).

☐ **Specify the revision manually**  
Manually input the revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family  
springbootwebapitask2

Revision  
1 (LATEST)

Service name  
Assign a unique name for this service.  
myspringbootservice

Service type [Info](#)  
Specify the service type that the service scheduler will follow.

☒ **Replica**  
Place and maintain a desired number of tasks across your cluster.

☐ **Daemon**  
Place and maintain one copy of your task on each container instance.

We press the **Create** button

**Optional configuration**

☐ **Service Connect - optional**  
Configure this service in a namespace to create and resolve endpoints. Services can resolve endpoints within the same namespace without task or application configuration.

☐ **Turn on Service Connect** [Info](#)  
Turn off Service Connect to remove the configuration.

**Service discovery - optional**  
Service discovery uses Amazon Route 53 to create a namespace for your service, which allows it to be discoverable via DNS.

**Networking**

**Load balancing - optional**

**Service auto scaling - optional**  
Automatically adjust your service's desired count up and down within a specified range in response to CloudWatch alarms. You can modify your service auto scaling configuration at any time to meet the needs of your application.

**Tags - optional** [Info](#)  
Tags help you to identify and organize your resources.

Cancel **Create**

It takes few minutes to create the new Service

myspringbootcluster2 deployment is in progress. It takes a few minutes. [View in CloudFormation](#)

Amazon Elastic Container Service > Clusters > myspringbootcluster2 > Services

### myspringbootcluster2

[Refresh](#) [Update cluster](#) [Delete cluster](#)

#### Cluster overview

ARN arn:aws:ecs:eu-north-1:954718177936:cluster/myspringbootcluster2	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining -	Active -	Tasks Pending -	Running -

[Services](#) [Tasks](#) [Infrastructure](#) [Metrics](#) [Scheduled tasks](#) [Tags](#)

#### Services (0) Info

[Refresh](#) [Manage tags](#) [Update](#) [Delete service](#) [Create](#)

Filter launch type: Any launch type Filter service type: Any service type

Filter services by value:

Service name | ARN | Status | Service type | Deployments and tasks | Last deployment | Task definition

myspringbootcluster2 [Refresh](#) [Update cluster](#) [Delete cluster](#)

#### Cluster overview

ARN arn:aws:ecs:eu-north-1:954718177936:cluster/myspringbootcluster2	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining -	Active 1	Tasks Pending -	Running 1

[Services](#) [Tasks](#) [Infrastructure](#) [Metrics](#) [Scheduled tasks](#) [Tags](#)

#### Services (1) Info

[Refresh](#) [Manage tags](#) [Update](#) [Delete service](#) [Create](#)

Filter launch type: Any launch type Filter service type: Any service type

Filter services by value:

<input type="checkbox"/>	Service name	ARN	Status	Service type	Deployments and tasks	Last deployment	Task definition
<input type="checkbox"/>	myspringbootservice	arn:aws:ecs:eu-north-1:954718177936:service/myspringbootcluster2/myspringbootservice	Active	REPLICA	1/1 Tasks running	Completed	springboot

Now we select the **Tasks** tab and we press on the task

**Cluster overview**

ARN arn:aws:ecs:eu-north-1:954718177936:cluster/myspringbootcluster2	Status Active	CloudWatch monitoring Default	Registered container instances -
Services Draining -	Active 1	Tasks Pending -	Running 1

**Tasks (1)**

Filter desired status: Running | Filter launch type: Any launch type

Task	Last status	Desired status	Task name	Health status	Started at	Container instance	Launch type	Platform version
6b37e...	Running	Running	springbo...	Unknown	5 minutes ago	-	FARGATE	1.4.0

## 5.4. How to create a new inbound rule to allow traffic on port 80

We press in the Networking tab

**Configuration** | Logs | **Networking** | Tags

**Task overview**

ARN arn:aws:ecs:eu-north-1:954718177936:task/myspringbootcluster2/6b37ec5f23c347a0ad88b380c932303a	Last status Running	Desired status Running	Started/Created at 2023-12-30T13:59:10.671Z 2023-12-30T13:58:38.758Z
---	------------------------	---------------------------	--

**Configuration**

Operating system/Architecture Linux/X86_64	Capacity provider -	ENI ID eni-00340e248c6f7f43e	Public IP 16.170.245.89   open address
CPU   Memory 1 vCPU   3 GB	Launch type FARGATE	Network mode awsvpc	Private IP 172.31.37.64

Container details for myspringbootcontainer

We select the Security Group

aws Services Search [Alt+S] Stockholm clouser1 @ 9547-18

Tell us what you think X

**Amazon Elastic Container Service**

Clusters

Namespaces

Task definitions

Account settings

Install AWS Copilot

Amazon ECR

Repositories

AWS Batch

myspringbootservice has been deployed successfully.

Amazon Elastic Container Service > Clusters > myspringbootcluster2 > Tasks > 6b37ec5f23c347a0ad88b380c932303a > Networking

6b37ec5f23c347a0ad88b380c932303a

Configuration Logs **Networking** Tags

Network

Run Reachability Analyzer

ENI ID  
eni-00340e248c6f7f43e

Subnet  
subnet-0baed75ce183fdd6f

Security groups  
sg-050cb9f74ed81537b (default)

Task role  
-

Task execution role  
ecsTaskExecutionRole

Public IP  
16.170.245.89 | open address

Private IP  
172.31.37.64

IPv6 address  
-

MAC address  
0a:2f:44:d1:66:be

aws Services Search [Alt+S] Stockholm clouser1 @ 9547-18

VPC dashboard EC2 Global View Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

**Security groups**

VPC > Security Groups > sg-050cb9f74ed81537b - default

sg-050cb9f74ed81537b - default

Actions

Details

Security group name  
default

Security group ID  
sg-050cb9f74ed81537b

Description  
default VPC security group

VPC ID  
vpc-08e469e2c38722f9b

Owner  
954718177936

Inbound rules count  
2 Permission entries

Outbound rules count  
1 Permission entry

Inbound rules Outbound rules Tags

Inbound rules (1/2)

Search

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
<input checked="" type="checkbox"/>	-	sgr-0c7d750ff3c15d69	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sgr-051aa7de5781d44...	-	All traffic	All	All	sg-050cb9f74ed81537b

aws Services Search [Alt+S] Stockholm clouser1 @ 9547-18

VPC > Security Groups > sg-050cb9f74ed81537b - default > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0c7d750ff3c15d69	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-051aa7de5781d44d4	All traffic	All	All	Custom	sg-050cb9f74ed81537b

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

## 5.5. Verify the application endpoints

We copy the Public IP address and we check the application endpoints

myspringbootservice has been deployed successfully.

Amazon Elastic Container Service > Clusters > myspringbootcluster2 > Tasks > 6b37ec5f23c347a0ad88b380c932303a > Networking

6b37ec5f23c347a0ad88b380c932303a

Configuration | Logs | **Networking** | Tags

**Network** Run Reachability Analyzer

ENI ID <a href="#">eni-00340e248c6f7f43e</a>	Task role -	<b>Public IP</b> <a href="#">16.170.245.89</a>   <a href="#">open address</a>
Subnet <a href="#">subnet-0baed75ce183fdd6f</a>	Task execution role <a href="#">ecsTaskExecutionRole</a>	Private IP <a href="#">172.31.37.64</a>
Security groups <a href="#">sg-050cb9f74ed81537b (default)</a>		IPv6 address -
		MAC address <a href="#">0a:2f:44:d1:66:be</a>

These are the application endpoints:

<http://16.170.245.89/hello>

ChatGPT | 16.170.245.89/hello

Not secure | 16.170.245.89/hello

Import favorites | Gmail | YouTube | Maps | Traducir | Noticias

Hello, World!

<http://16.170.245.89/actuator/health>

ChatGPT | 16.170.245.89/actuator/health

Not secure | 16.170.245.89/actuator/health

Import favorites | Gmail | YouTube | Maps | Traducir | Noticias

```
1 {
2   "status": "UP",
3   "components": {
4     "diskSpace": {
5       "status": "UP",
6       "details": {
7         "total": 31526391808,
8         "free": 19154272256,
9         "threshold": 10485760,
10        "exists": true
11      }
12    },
13    "ping": {
14      "status": "UP"
15    }
16  }
17 }
```