

## Contents

1. Build Spring Boot Application docker image.....	1
1.1 Setup the SWS CLI .....	1
1.2 Dockerfile .....	1
1.3 Build .....	1
1.4 Run .....	1
2. ECR – Elastic Container Registry .....	2
2.1 Create Container Registry .....	2
2.2 Login in AWS by CLI.....	2
2.3 Tag the Docker image .....	2
2.4 Push the docker image to AWS Registry.....	2
2.5 Create Batch file to push the image.....	3
3 Deploy by AWS portal .....	3
4 Install ECS CLI .....	9
4 Creating a Cluster with a Fargate Task Using the Amazon ECS CLI.....	9
4.1 Create the Task Execution IAM Role .....	9
4.1.1 Create the task execution role resource file.....	9
4.1.2 Create the task execution role .....	9
4.1.3 Attach the task execution role policy .....	10
4.2 ECS Deploy .....	10
4.2.1 Create a Cluster.....	10
4.2.2 Register a Task Definition .....	10
4.2.3 List Task Definitions .....	10
4.2.4 Create a Service .....	10
4.2.5 List Services .....	11
4.2.6 Describe the Running Service .....	11

## 1. Build Spring Boot Application docker image

### 1.1 Setup the SWS CLI

Setup the AWS environments in development machine.

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnXEMI/XXXXXXXX/bPxRfiCYEXAMPLEKEY
```

```
Default region name [None]: ap-south-1
```

```
Default output format [None]: json
```

### 1.2 Dockerfile

Create the Dockerfile in the root folder and paste the following content in it.

```
FROM openjdk:8-jdk-alpine
```

```
RUN addgroup -S hexaware && adduser -S insurance -G hexaware
```

```
USER insurance:hexaware
```

```
COPY ./target/*.jar app.jar
```

```
EXPOSE 8080
```

```
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

### 1.3 Build

Build the above docker file with the following command.

```
docker build -t insurance-service .
```

### 1.4 Run

Run the following run command to run the docker image with required environment variables.

```
docker run -it -p 8080:8080 insurance-service
```

## 2. ECR – Elastic Container Registry

### 2.1 Create Container Registry

Create Elastic Container registry in AWS


ECR > Repositories > Create repository

### Create repository

#### Repository configuration

Repository name

319964789365.dkr.ecr.ap-south-1.amazonaws.com/

 Repository name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, and forward slashes. A namespace can be included with your repository name (e.g. namespace/repo-name).

Tag immutability

Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☐ Disabled

Scan on push

Enable scan on push to have each image automatically scanned after being pushed to a repository. If disabled, each image scan must be manually started to get scan results.

☐ Disabled

Cancel **Create repository**

### 2.2 Login in AWS by CLI

```
aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin <aws-account-id>.dkr.ecr.ap-south-1.amazonaws.com/email-service
```

### 2.3 Tag the Docker image

```
docker tag email-service:latest 319964789365.dkr.ecr.ap-south-1.amazonaws.com/email-service:latest
```

### 2.4 Push the docker image to AWS Registry

```
docker push 319964789365.dkr.ecr.ap-south-1.amazonaws.com/email-service:latest
```

## 2.5 Create Batch file to push the image

Create a batch file in root folder with name “docker-push.bat” with the following content.

```
call mvn clean install
call aws ecr get-login-password --region us-east-2 | docker login --
username AWS --password-stdin 628030530634.dkr.ecr.us-east-
2.amazonaws.com/insurance-service
call docker build -t insurance-service .
call docker tag insurance-service:latest 628030530634.dkr.ecr.us-east-
2.amazonaws.com/insurance-service:latest
call docker push 628030530634.dkr.ecr.us-east-2.amazonaws.com/insurance-
service:latest
```

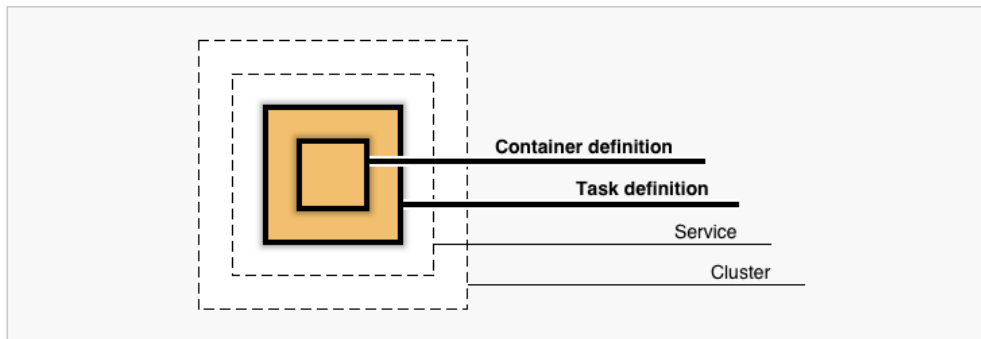
Run the above batch file to build the docker image and push the image to elastic container registry.

## 3 Deploy by AWS portal

After successful login in AWS portal navigate to the following URL for elastic container service.

<https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/clusters>

Diagram of ECS objects and how they relate



### Container definition

Edit

Choose an image for your container below to get started quickly or define the container image to use.

#### sample-app

image : httpd:2.4  
memory : 0.5GB (512)  
cpu : 0.25 vCPU (256)

#### nginx

image : nginx:latest  
memory : 0.5GB (512)  
cpu : 0.25 vCPU (256)

Click the configure button available in custom white box.

<b>tomcat-webserver</b> image : tomcat memory : 2GB (2048) cpu : 1 vCPU (1024)	<b>custom</b> image : -- memory : -- cpu : -- <a href="#">Configure</a>
---	---

## Task definition

[Edit](#)

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

Task definition name	first-run-task-definition	<a href="#">?</a>
Network mode	awsvpc	<a href="#">?</a>
Task execution role	Create new	<a href="#">?</a>
Compatibilities	FARGATE	<a href="#">?</a>
Task memory	0.5GB (512)	
Task CPU	0.25 vCPU (256)	

Fill the following form with appropriate vales.

▼ Standard

Container name\*

insurance-service



Image\*

628030530634.dkr.ecr.us-east-2.amazonaws.com/insurance-service:latest



Private repository authentication\*

☐

Memory Limits (MiB)

Soft limit

512



[+ Add Hard limit](#)

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions.

ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings

Container port

Protocol



8080

tcp



[+ Add port mapping](#)

### tomcat-webserver

image : tomcat  
memory : 2GB (2048)  
cpu : 1 vCPU (1024)

### insurance-service

[Configure](#)

image :  
628030530634.dkr.ecr.us-east-2.amazonaws.com/insurance-service:latest  
memory : 0.5GB (512)  
cpu :

## Task definition

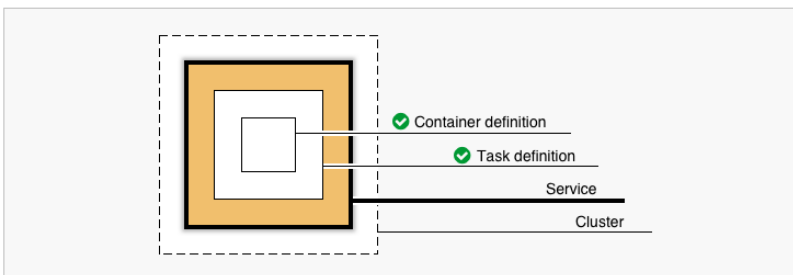
[Edit](#)

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

Task definition name	insurance-task-definition	<a href="#">i</a>
Network mode	awsvpc	<a href="#">i</a>
Task execution role	Create new	<a href="#">i</a>
Compatibilities	FARGATE	<a href="#">i</a>
Task memory	0.5GB (512)	
Task CPU	0.25 vCPU (256)	

Click Next button.

Diagram of ECS objects and how they relate



## Define your service

[Edit](#)

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name insurance-service-service

Number of desired tasks 1

Security group Automatically create new

A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

Load balancer type ☒ None  
☐ Application Load Balancer

\*Required

[Cancel](#)[Previous](#)[Next](#)

Enable the application load balancer.

## Define your service

[Edit](#)

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name insurance-service-service

Number of desired tasks 1

Security group Automatically create new

Two security groups are created to secure your service: An Application Load Balancer security group that allows all traffic on the Application Load Balancer port and an Amazon ECS security group that allows all traffic ONLY from the Application Load Balancer security group. You can further configure security groups and network access outside of this wizard.

Load balancer type ☐ None  
☒ Application Load Balancer

Load balancer listener port 8080

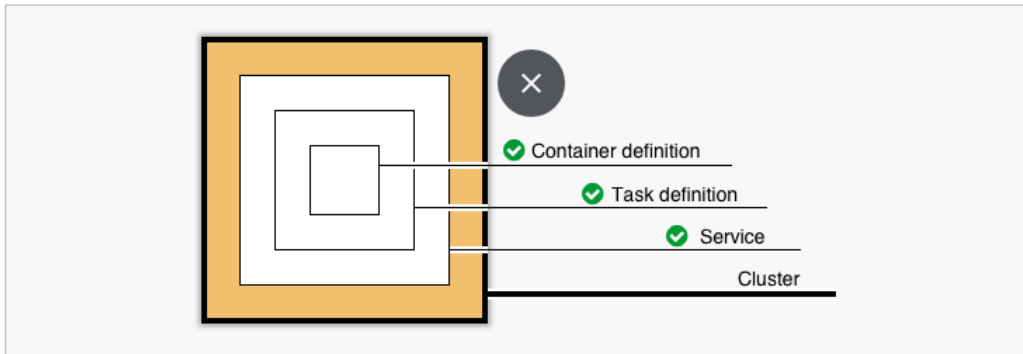
Load balancer listener protocol HTTP

\*Required

[Cancel](#)[Previous](#)[Next](#)

Click next button.





## Configure your cluster

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the [Amazon ECS documentation](#).

Cluster name

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.

VPC ID  ⓘ

Subnets  ⓘ

\*Required

Cancel

Previous

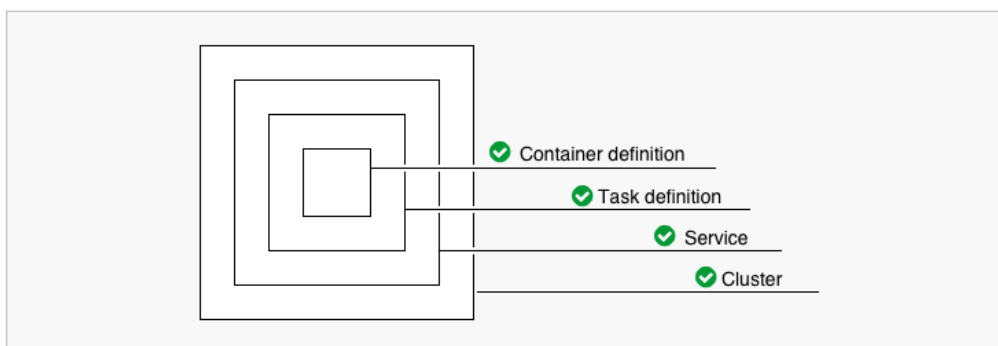
Next

Fill the cluster name and Click next button.

It will show the review page. Review all the configuration in once page and click create button. After 10 minutes the fargate cluster will be crated in the appropriate account.

The service URL will be available in Load balancer information page.

## Diagram of ECS objects and how they relate



## 4 Install ECS CLI

Follow the installation steps in the following URL.

[ECS CLI installation Steps](#)

## 4 Creating a Cluster with a Fargate Task Using the Amazon ECS CLI

### 4.1 Create the Task Execution IAM Role

To create the task execution IAM role using the AWS CLI

#### 4.1.1 Create the task execution role resource file

file `task-execution-assume-role.json`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

#### 4.1.2 Create the task execution role

```
aws iam --region ap-south-1 create-role --role-name ecsTaskExecutionRole --assume-role-policy-document file://task-execution-assume-role.json
```

#### 4.1.3 Attach the task execution role policy

```
aws iam --region ap-south-1 attach-role-policy --role-name ecsTaskExecutionRole --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

## 4.2 ECS Deploy

### 4.2.1 Create a Cluster

```
aws ecs create-cluster --cluster-name insurance-cluster
```

### 4.2.2 Register a Task Definition

```
aws ecs register-task-definition --cli-input-json file:///C:/Users/38904/Desktop/Amaze/email-service/ecs/fargate/task.json
```

[Reference](<https://docs.aws.amazon.com/cli/latest/reference/ecs/register-task-definition.html>)

### Role creation

[Reference]([https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task\\_execution\\_IAM\\_role.html](https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_execution_IAM_role.html))

### 4.2.3 List Task Definitions

```
aws ecs list-task-definitions
```

```
aws ecs describe-task-definitions --task-definition insurance-task-definition:1
```

### 4.2.4 Create a Service

**Note:** Subnet and security group need to be created before create service

```
aws ecs create-service `
--cluster amaze-cluster `
--service-name insurance-service `
--task-definition email:1 `
--desired-count 1 `
--launch-type "FARGATE" `
--network-configuration "awsvpcConfiguration={subnets=[subnet-28962353],securityGroups=[sg-61aa2c02]}"`
--load-balancers
"targetGroupArn=string,loadBalancerName=string,containerName=string,containerPort=integer"
```

#### 4.2.5 List Services

```
aws ecs list-services --cluster insurance-cluster
```

#### 4.2.6 Describe the Running Service

```
aws ecs describe-services --cluster insurance-cluster --services insurance-service
```