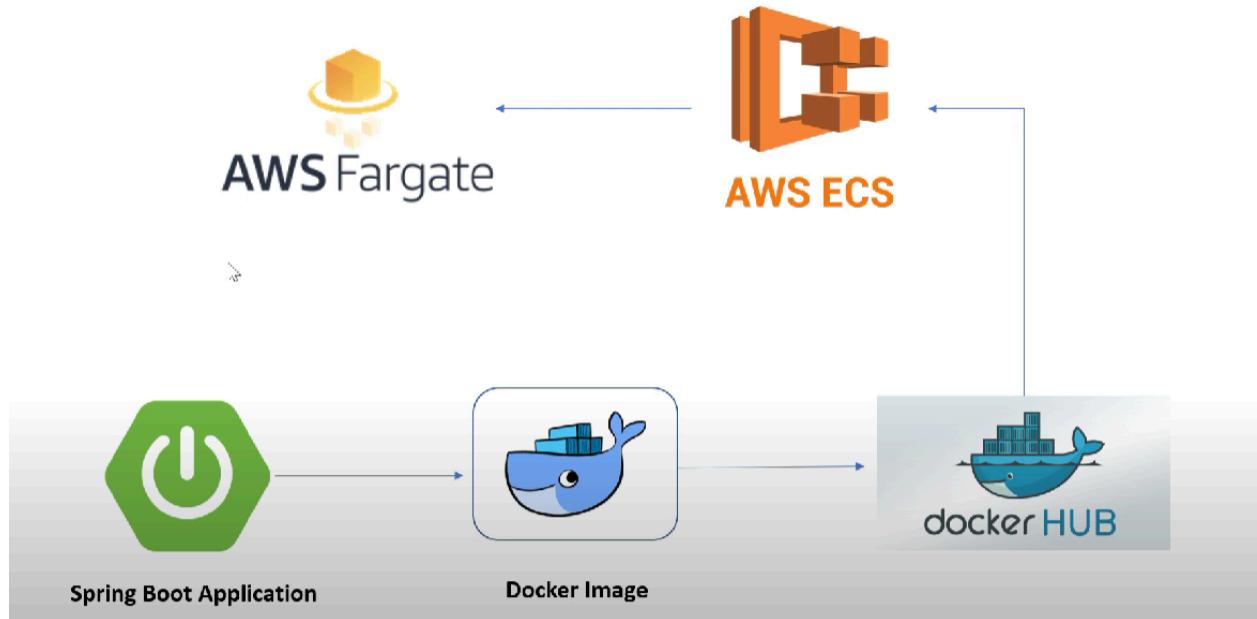


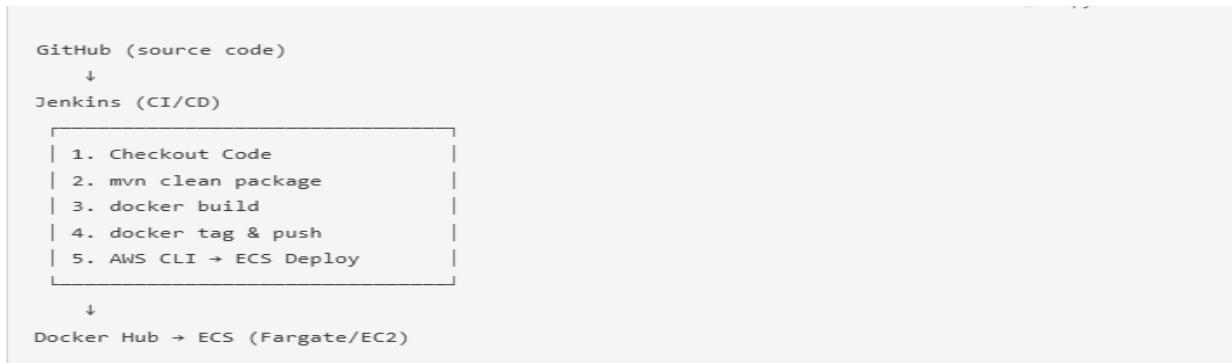
## AWS Deployment



We can also do all these by mentioned in jenkins to build and see the pipelines.

### Full CI/CD Flow:

1. **Clone your code** from GitHub
2. **Build your Spring Boot app** using Maven/Gradle
3. **Build Docker image**
4. **Tag & Push to Docker Hub**
5. **Deploy to ECS (via AWS CLI or ECS plugin)**



**But did it directly without using jenkins.**

1. [Spring Boot → Docker Image]
2. [Push to Docker Hub]
3. [Create ECS Task Definition]
4. [Create ECS Cluster]
5. [Run Task on ECS (Fargate or EC2)]

## Building docker image

Command to Build Docker Image via Maven

### Command to Build Docker Image via Maven

```
mvn spring-boot:build-image
-Dspring-boot.build-image.imageName=<your-dockerhub-username>/devops-integration
```

✓ Example:

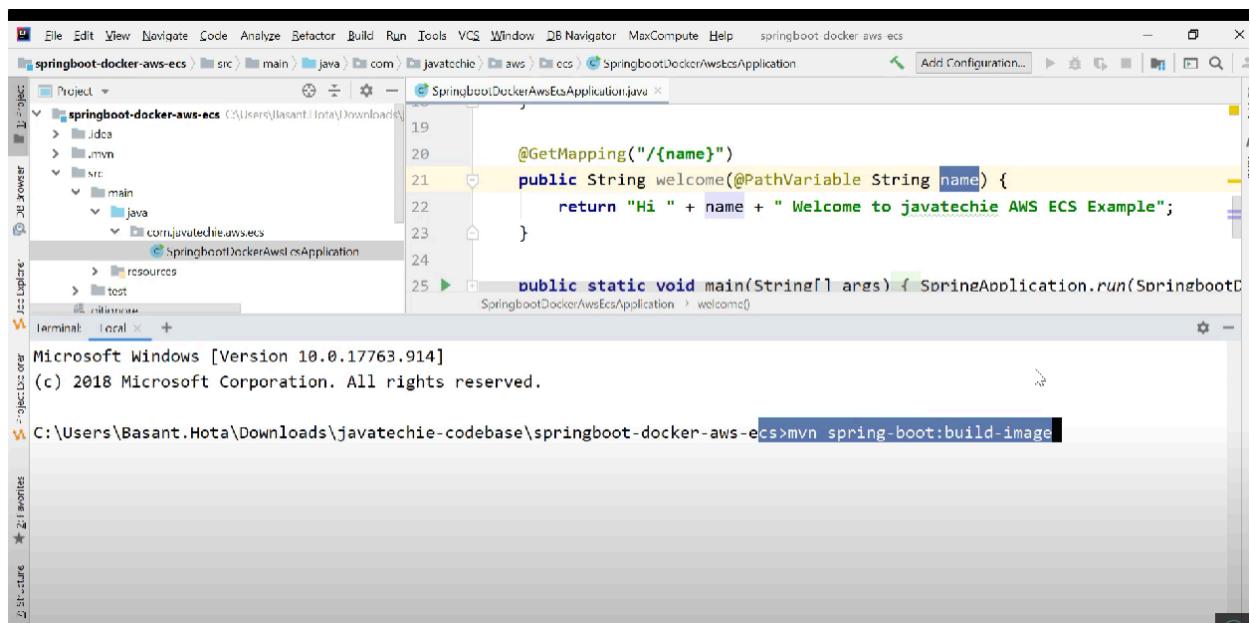
```
mvn spring-boot:build-image
-Dspring-boot.build-image.imageName=bharath19/devops-integration
```

# Build with Maven (build docker image)

```
mvn clean package dockerfile:build
```

```
# Run docker image locally  
docker run -d -p 8080:8080 yourdockerhubusername/your-image-name:tag
```

```
# Push to Docker Hub (after login)  
docker push yourdockerhubusername/your-image-name:tag
```



Run the docker image

The screenshot shows the IntelliJ IDEA interface with the following details:

- Code Editor:** The main window displays the file `SpringbootDockerAwsEcsApplication.java` containing Java code for a Spring Boot application.
- Terminal:** The bottom panel shows the command-line output of a Docker build process:

```
[INFO] [creator] Adding label 'org.springframework.boot.version'  
[INFO] [creator] *** Images (73457a32ba26):  
[INFO] [creator] docker.io/library/springboot-docker-aws-ecs:0.0.1-SNAPSHOT  
[INFO]  
[INFO] Successfully built image 'docker.io/library/springboot-docker-aws-ecs:0.0.1-SNAPSHOT'  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 27.225s  
[INFO] Finished at: Wed Aug 26 00:09:47 IST 2020  
[INFO] Final Memory: 26M/250M  
[INFO] -----  
C:\Users\Basant.Hota\Downloads\javatechie-codebase\springboot-docker-aws-ecs>docker run --tty --publish 8080:8080 springboot-docker-aws-ecs:0.0.1-SNAPSHOT
```

Push docker image to dockerHub

Suppose you have a local image called `myapp` with tag `v1`:

docker images

```
# REPOSITORY TAG IMAGE ID  
# myapp v1 abc123def456
```

You want to tag it for pushing to Docker Hub under your username `mydockeruser`:

```
docker tag myapp:v1 mydockeruser/myapp:v1
```

---

**Now you can push:**

```
docker push mydockeruser/myapp:v1
```

This creates a versioned image that Docker Hub understands.

```

MINGW64:/c/Program Files/Docker Toolbox
~~ { ~ ~ ~ ~ ~ ~ ~ ~ ~ / ==- ~ ~ ~
      \_o
        \_/_\

Docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Start interactive shell

rasant.Hota@INLT1931 MINGW64 /c/Program Files/Docker Toolbox (master)
$ docker image ls
REPOSITORY                                     TAG      IMAGE ID      CREATED     SIZE
paketobuildpacks/run                           latest   2e4cb9ee33b7  2 days ago  84.3MB
api-gateway-demo.jar                          latest   3a55b38f57c8  5 days ago  534MB
paketobuildpacks/run                           <none>  c99d8107f5cf  6 days ago  84.3MB
nginx                                           latest   4bb46517cac3  11 days ago  133MB
openjdk                                         8       5684f3366a1f  2 weeks ago  511MB
gcr.io/paketo-buildpacks/run                   base-cnb 610caf01e85e  3 months ago  71.1MB
gcr.io/paketo-buildpacks/builder              latest   48c94821db00  40 years ago  697MB
gcr.io/paketo-buildpacks/builder              <none>  4b268bb20aa3  40 years ago  648MB
gcr.io/paketo-buildpacks/builder              <none>  9328a12ebcd0  40 years ago  648MB
api-gateway-demo                            0.0.1-SNAPSHOT 5ef75a8f7e66  40 years ago  250MB
68977122149.dkr.ecr.ap-south-1.amazonaws.com/javatechie
javatechie/aws-img                         latest   c1acf01c1489  40 years ago  245MB
javatechie/demo-app                        0.0.1-SNAPSHOT c1acf01c1489  40 years ago  245MB
javatechie/demo-app                        0.0.1-SNAPSHOT c1acf01c1489  40 years ago  245MB
javatechie/demo-app                        0.0.1-SNAPSHOT 6cc341ce3e09  40 years ago  239MB
hello-world                                 0.0.1-SNAPSHOT 6ab875e34df0  40 years ago  697MB
gcr.io/paketo-buildpacks/builder              <none>  a20c33d4d781  40 years ago  243MB
javatechie/demo-app                        0.0.1-SNAPSHOT 89ca48ea7290  40 years ago  243MB
demo-app                                    0.0.1-SNAPSHOT 89ca48ea7290  40 years ago  243MB
springboot-docker-aws-ecs                  0.0.1-SNAPSHOT 73457a32ba26  40 years ago  243MB
spring-boot-docker-image                   0.0.1-SNAPSHOT a20c33d4d781  40 years ago  239MB
<none>                                     <none>  34b2986f5239  40 years ago  250MB

rasant.Hota@INLT1931 MINGW64 /c/Program Files/Docker Toolbox (master)
$ docker tag springboot-docker-aws-ecs:0.0.1-SNAPSHOT javatechie/springboot-docker-aws-ecs:0.0.1-SNAPSHOT

```

```

rasant.Hota@INLT1931 MINGW64 /c/Program Files/Docker Toolbox (master)
$ docker image ls
REPOSITORY                                     TAG      IMAGE ID      CREATED     SIZE
paketobuildpacks/run                           latest   2e4cb9ee33b7  2 days ago  84.3MB
api-gateway-demo.jar                          latest   3a55b38f57c8  5 days ago  534MB
paketobuildpacks/run                           <none>  c99d8107f5cf  6 days ago  84.3MB
nginx                                           latest   4bb46517cac3  11 days ago  133MB
openjdk                                         8       5684f3366a1f  2 weeks ago  511MB
gcr.io/paketo-buildpacks/run                   base-cnb 610caf01e85e  3 months ago  71.1MB
gcr.io/paketo-buildpacks/builder              latest   4b268bb20aa3  40 years ago  648MB
javatechie/demo-app                        0.0.1-SNAPSHOT 89ca48ea7290  40 years ago  243MB
javatechie/demo-app                        0.0.1-SNAPSHOT 6ab875e34df0  40 years ago  697MB
javatechie/springboot-docker-aws-ecs          <none>  73457a32ba26  40 years ago  243MB
springboot-docker-aws-ecs                  0.0.1-SNAPSHOT 73457a32ba26  40 years ago  243MB
68977122149.dkr.ecr.ap-south-1.amazonaws.com/javatechie
javatechie/aws-img                         latest   c1acf01c1489  40 years ago  245MB
javatechie/demo-app                        0.0.1-SNAPSHOT c1acf01c1489  40 years ago  245MB
javatechie/demo-app                        0.0.1-SNAPSHOT 5ef75a8f7e66  40 years ago  250MB
api-gateway-demo                            0.0.1-SNAPSHOT 9328a12ebcd0  40 years ago  648MB
gcr.io/paketo-buildpacks/builder              <none>  34b2986f5239  40 years ago  250MB
<none>                                     <none>  6cc341ce3e09  40 years ago  239MB
hello-world                                 0.0.1-SNAPSHOT 48c949821db0  40 years ago  697MB
gcr.io/paketo-buildpacks/builder              <none>  48c949821db0  40 years ago  697MB

```

Pushing

```

MINGW64:/c/Program Files/Docker Toolbox
Basant.Hota@INLT1931 MINGW64 /c/Program Files/Docker Toolbox (master)
$ 

Basant.Hota@INLT1931 MINGW64 /c/Program Files/Docker Toolbox (master)
$ docker push javatechie/springboot-docker-aws-ecs:0.0.1-SNAPSHOT
The push refers to repository [docker.io/javatechie/springboot-docker-aws-ecs]
8caef28d6182: Pushed
7cfffc3c153c7: Mounted from javatechie/demo-app
f98424725587: Pushing [=====] 16.71MB
6783b9e93792: Mounted from javatechie/demo-app
5cd38b221a5e: Mounted from javatechie/demo-app
4f53326d9ef5: waiting
810e950c6a17: Waiting
f5192aeff1e0: Waiting
e06e5b98e8e3: Waiting
170848Be8e8e: Waiting
d5202d622907: Waiting
e35889906a2ee: Waiting
e84232dd4559: Waiting
979718509333: Waiting
7292be7bbc48: Waiting
2542fe3c2e07: Waiting
6f5c8807c41a: Waiting
801e4a88973b: Waiting
2ba5b91c92b0: Waiting
2f37d1021287: Waiting
79bde4d54386: Waiting

```

The screenshot shows the Docker Hub interface with the user 'javatechie' logged in. The top navigation bar includes links for Explore, Repositories, Organizations, Get Help, and the user's name 'javatechie'. Below the navigation, there is a search bar and a dropdown menu set to 'javatechie'. A 'Create Repository' button is visible. The main content area displays five repository cards:

- javatechie / springboot-docker-aws-ecs** - Updated 5 minutes ago, 0 stars, 1 download, Public.
- javatechie / demo-app** - Updated 13 hours ago, 0 stars, 5 downloads, Public.
- javatechie / aws-img** - Updated 5 days ago, 0 stars, 5 downloads, Public.
- javatechie / docker-jenkins-integration-sample** - Updated 8 months ago, 0 stars, 9 downloads, Public.
- javatechie / spring-boot-docker-maven** - Updated 2 years ago, 0 stars, 6 downloads, Public.

**Amazon Elastic Container Service (ECS)** is a fully managed container orchestration service used to deploy, manage, and scale containerized applications on Amazon Web Services (AWS). It enables users to run containers with ease, whether it's a simple website or a complex, distributed application, and supports both EC2 instances and AWS Fargate.

**Amazon EC2** is a web service that provides **resizable compute capacity (virtual machines)** in the cloud. You can launch and manage **Linux or Windows servers** (called **EC2 instances**) and run any software on them, including containerized applications.

**AWS Fargate** is a **serverless compute engine for containers**. You don't need to provision or manage servers. You **only define your container requirements** (CPU, memory), and Fargate runs them automatically.

### **Example Analogy:**

- **EC2** is like **renting a car**: You drive and maintain it yourself.
- **Fargate** is like **using a taxi or Uber**: You just specify where to go (CPU & RAM) — AWS takes care of driving.

### **Key Features of ECS:**

#### **1. Fully Managed Service:**

- AWS handles the orchestration, availability, and scalability of your containerized workloads.
- No need to install Kubernetes or manage your own orchestrators unless using Amazon EKS.

#### **2. Launch Types:**

- **EC2 Launch Type**: You manage the EC2 instances (virtual machines) on which containers run.
- **Fargate Launch Type**: Serverless – you don't manage the infrastructure. Just specify CPU and memory; AWS does the rest.

#### **3. Tight AWS Integration:**

- Integrated with services like IAM, CloudWatch, ALB (Application Load Balancer), ECR (Elastic Container Registry), and more.

#### **4. Task Definitions:**

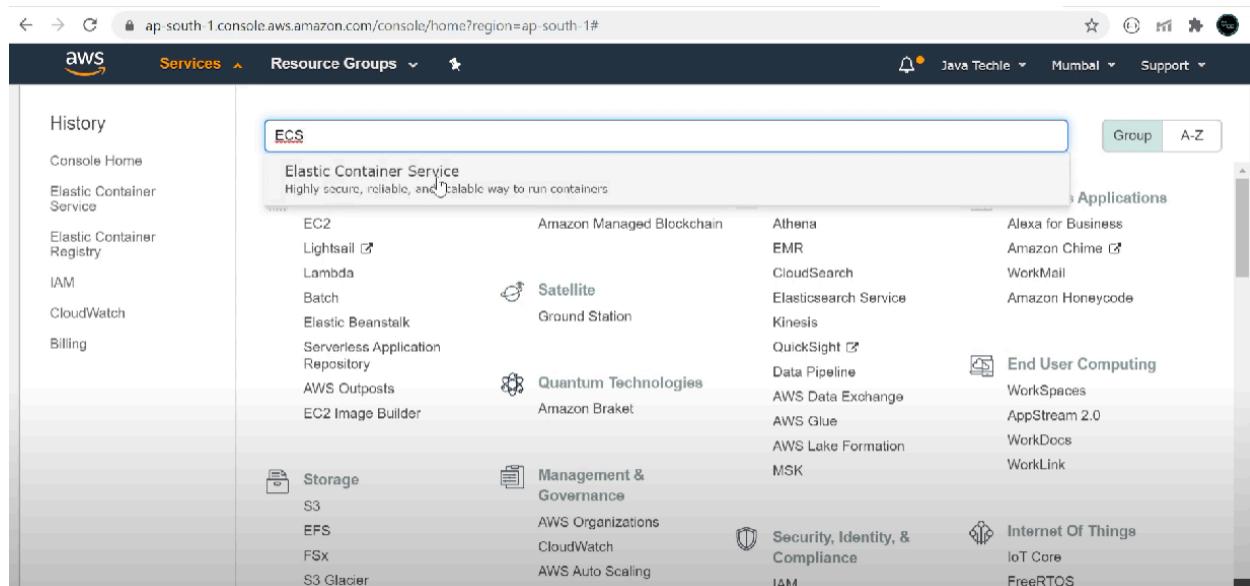
- JSON templates that define **how your containers should run** (image, CPU, memory, networking, etc.).

## 5. Service Management:

- Ensures that a specified number of tasks are always running and can do **rolling updates** to avoid downtime.

## 6. Networking Options:

- Support for **bridge**, **host**, and **AWS VPC networking** modes for flexibility.

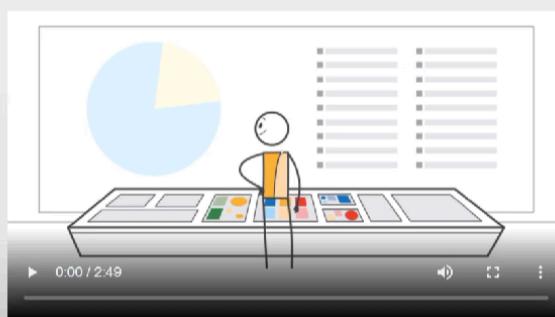


← → C ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/getStarted

AWS Services Resource Groups ★

Amazon ECS Clusters Task Definitions Account Settings Amazon ECR Repositories AWS Marketplace Discover software Subscriptions

## Amazon Elastic Container Service (ECS)



Amazon ECS makes it easy to deploy, manage, and scale Docker containers running applications, services, and batch processes. Amazon ECS places containers across your cluster based on your resource needs and is integrated with familiar features like Elastic Load Balancing, EC2 security groups, EBS volumes and IAM roles.

Explore - Docker Hub | How to Deploy Docker | (336) Java Techie - YouTube | Java-Techie-JT/Spring | 192.168.99.100:8080 | Amazon ECS | + - ×

← → C ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions

AWS Services Resource Groups ★

Amazon ECS Clusters **Task Definitions** Account Settings Amazon ECR Repositories AWS Marketplace Discover software Subscriptions

### Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. [Learn more](#)

Create new Task Definition Create new revision Actions Last updated on August 28, 2020 12:28:31 AM (0m ago) ⌂ ⓘ

Status: **ACTIVE** INACTIVE

Filter in this page

Task Definition	Latest revision status
No results	

Anyone of this select

AWS Services Resource Groups ★

Java Techie Mumbai Support

## Create new Task Definition

**Step 1: Select launch type compatibility**

Step 2: Configure task and container definitions

### Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

**FARGATE**

Price based on task size  
Requires network mode awsvpc  
AWS-managed infrastructure, no Amazon EC2 instances to manage

**EC2**

Price based on resource usage  
Multiple network modes available  
Self-managed infrastructure using Amazon EC2 instances

\*Required

Cancel Next step

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/create

AWS Services Resource Groups ★

Java Techie Mumbai Support

## Create new Task Definition

**Step 1: Select launch type compatibility**

**Step 2: Configure task and container definitions**

### Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

**Task Definition Name\***  ⓘ

**Requires Compatibilities\*** FARGATE

**Task Role**  ⓘ

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console ⓘ

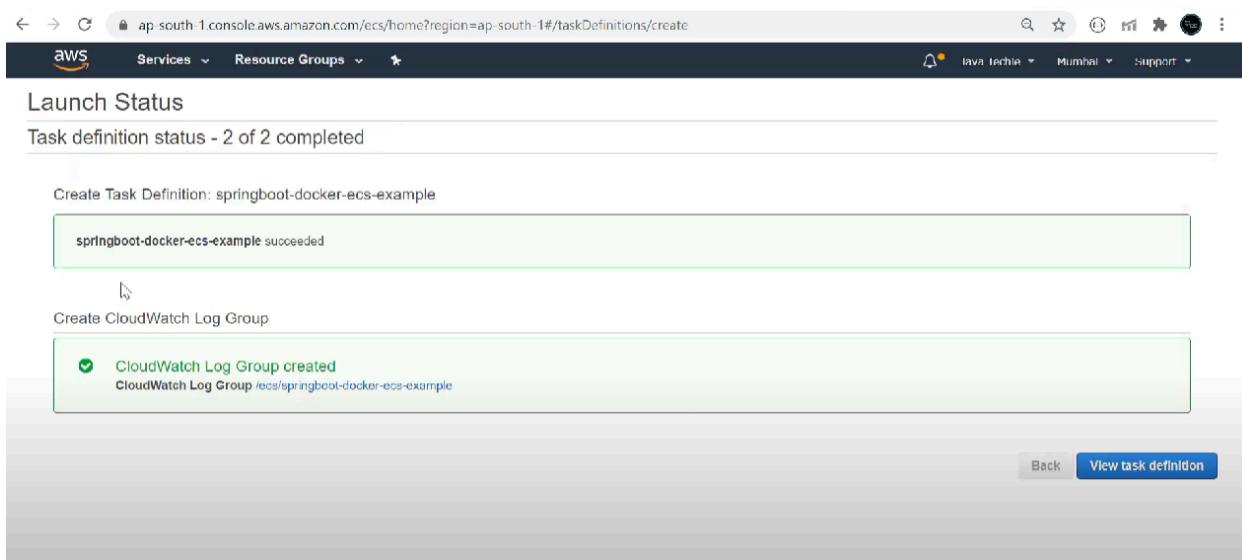
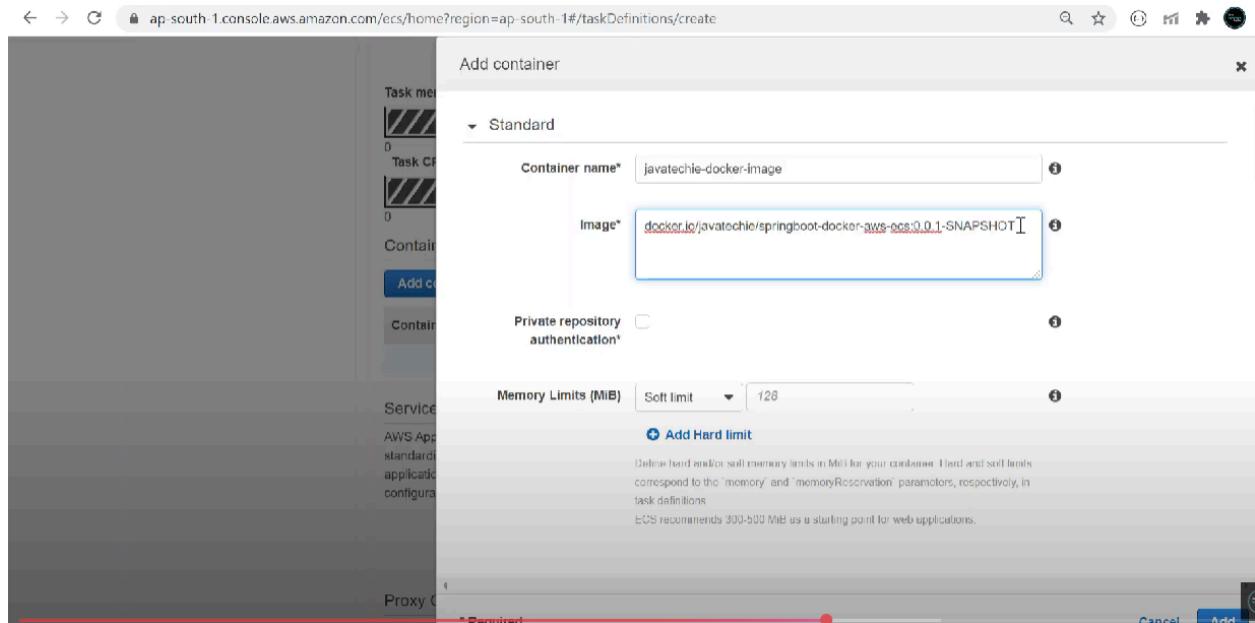
**Network Mode**  ⓘ

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

The screenshot shows the AWS ECS console interface. At the top, there are several tabs including 'Explore - Docker Hub', 'How to Deploy Docker', '(536) Java Iechic - Yo', 'Java-Iechic-jt/spring', '192.168.99.100:8080', and 'Amazon ECS'. The main content area has a header 'Task memory maximum allocation for container memory reservation' with a progress bar set at 1024 shared of 1024 MiB. Below it is another progress bar for 'Task CPU maximum allocation for containers' at 512 shared of 512 CPU units. A section titled 'Container Definitions' contains a button 'Add container' which is highlighted with a mouse cursor. A table below shows columns for Container Name, Image, Hard/Soft mem..., CPU Unit..., GPU, and Essential ...; the message 'No results' is displayed. Under 'Service Integration', there is a note about AWS App Mesh and a 'Enable App Mesh integration' checkbox. At the bottom, there is a 'Proxy Configuration' section.

Creating container and pasting docker url and tag from the docker hub

The screenshot shows the Docker Hub repository page for 'javatechie/springboot-docker-aws-ecs'. The top navigation bar includes 'Explore', 'Repositories', 'Organizations', 'Get Help', and a user dropdown. The repository path 'javatechie / springboot-docker-aws-ecs' is shown. A message indicates 'Using 0 of 1 private repositories. [Get more](#)'. The 'General' tab is selected, showing details like 'Last pushed: 6 minutes ago'. On the right, a 'Docker commands' section contains a button 'Public View' and a command line input field with the placeholder 'docker push javatechie/springboot-docker-aws-ecs:tagname'. Below this, the 'Tags' section lists '0.0.1-SNAPSHOT' with a download icon and a timestamp '8 minutes ago'. The 'Recent builds' section is also visible.



Task definition is created successfully

The screenshot shows the AWS ECS Task Definitions console. The URL is [ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/springboot-docker-ecs-example:1](https://ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/taskDefinitions/springboot-docker-ecs-example:1). The navigation bar includes 'Services' (selected), 'Resource Groups', and 'Support'. The left sidebar lists 'Amazon ECS', 'Clusters', 'Task Definitions' (selected), 'Account Settings', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. The main content area shows 'Task Definition: springboot-docker-ecs-example:1'. It has a 'Create new revision' button and an 'Actions' dropdown. Below that are tabs for 'Builder' (selected), 'JSON', and 'Tags'. The 'Task Definition Name' is set to 'springboot-docker-ecs-example'. The 'Task Role' is 'None'. A note says: 'Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the IAM Console.' The 'Network Mode' is set to 'awsvpc'. A note says: 'If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.' There is also a 'Create' button.

Now we need to create aws cluster where we can run our task definitions

The screenshot shows the AWS ECS console. The URL is [ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1](https://ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1). The navigation bar includes 'Services' (selected), 'Resource Groups', and 'Support'. The left sidebar lists 'Amazon ECS', 'Clusters', 'Task Definitions' (selected), 'Account Settings', 'Amazon ECR', 'Repositories', 'AWS Marketplace', 'Discover software', and 'Subscriptions'. The main content area is currently empty.

Sales Java Techie Mumbai Support

Amazon ECS

**Clusters**

Task Definitions

Account Settings

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

**Clusters**

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

**Create Cluster** **Get Started**

View **list** **card** 0 loaded of 0 clusters

**Filter in this page**

Cluster name	CloudWatch monitorin...	Services	Running tasks	Pending tasks	Container Instances
loading					

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/create/new

**Step 2: Configure cluster**

**Cluster name\***

**Networking**

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create a new VPC for this cluster

**Tags**

Key	Value
Add key	Add value

**CloudWatch Container Insights**

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices. It collects, aggregates, and summarizes compute utilization such as CPU, memory, disk, and network; and diagnostic information such as container restart failures to help you isolate issues with your clusters and resolve them quickly. [Learn more](#)

Enable Container Insights

\*Required **Create**

← → 🔍 ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/javatechie-aws-cluster/tasks

**Clusters**

- Task Definitions
- Account Settings
- Amazon ECR
- Repositories
- AWS Marketplace
- Discover software
- Subscriptions ↗

**Cluster : javatechie-aws-cluster**

Get a detailed view of the resources on your cluster.

**Cluster ARN** arn:aws:ecs:ap-south-1:688977122149:cluster/javatechie-aws-cluster

**Status** ACTIVE

**Registered container instances** 0

**Pending tasks count** 0 Fargate, 0 EC2

**Running tasks count** 0 Fargate, 0 EC2

**Active service count** 0 Fargate, 0 EC2

**Draining service count** 0 Fargate, 0 EC2

**Services** **Tasks** **ECS Instances** **Metrics** **Scheduled Tasks** **Tags** **Capacity Providers**

**Run new Task** **Stop** **Stop All** **Actions** ▾ Last updated on August 26, 2020 12:32:10 AM (0m ago) **⟳** **?**

Desired task status: **Running** Stopped

Filter in this page Launch type ALL

Task	Task definiti...	Container in...	Last status ...	Desired stat...	Started By	Group	Launch type...	Platform ver...
No results								

And select the task definition whatever we created

← → 🔍 ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/javatechie-aws-cluster/runTask

**Repositories**

- AWS Marketplace
- Discover software
- Subscriptions ↗

**Launch type**  FARGATE  EC2 **ⓘ**

[Switch to capacity provider strategy](#) **ⓘ**

**Task Definition** Family **springboot-docker-ecs-example** **ⓘ** Enter a value

Revision **1 (latest)** **ⓘ**

**Platform version** LATEST **ⓘ**

**Cluster** javatechie-aws-cluster **ⓘ**

**Number of tasks** 1 **ⓘ**

**Task Group** **ⓘ**

**VPC and security groups**

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

**Cluster VPC\*** **ⓘ**

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/javatechie-aws-cluster/runTask

VPC and security groups are configurable within your task definition unless the awsvpc network mode is selected.

**Cluster VPC\*** vpc-7c3a2414 (172.31.0.0/16)

**Subnets\*** subnet-bf50e7c4 (172.31.16.0/20) - ap-south-1c assign ipv6 on creation: Disabled

**Security groups\*** spring-8272 **Edit**

**Auto-assign public IP** ENABLED

**Advanced Options**

**Tagging requirements:** Tagging requires that you opt in to the new ARN and resource ID format. The IAM user/role has not opted in to the new ARN format. Opt-in to the new format to use it.

**Configure security groups**

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more](#)

**Assigned security groups**  Create new security group  Select existing security group

**Security group name\*** spring-8272

**Description** Wed Aug 26 2020 00:32:23 GMT+0530

**Inbound rules for security group**

Type	Protocol	Port range	Source
HTTP	TCP	80	Anywhere

**Add rule**

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/javatechie-aws-cluster/runTask

VPC and security groups are configurable within your task definition unless the awsvpc network mode is selected.

**Cluster VPC\*** vpc-7c3a2414 (172.31.0.0/16)

**Subnets\*** subnet-bf50e7c4 (172.31.16.0/20) - ap-south-1c assign ipv6 on creation: Disabled

**Security groups\*** spring-8272

**Auto-assign public IP** ENABLED

**Advanced Options**

**Tagging requirements:** Tagging requires that you opt in to the new ARN and resource ID format. The IAM user/role has not opted in to the new ARN format. Opt-in to the new format to use it.

**Configure security groups**

A security group is a set of firewall rules that control the traffic for your task. On this page, you can add rules to allow specific traffic to reach your task, or you can choose to use an existing security group. [Learn more](#)

**Assigned security groups**  Create new security group  Select existing security group

**Security group name\*** javatechie-all-access-security

**Description** Wed Aug 26 2020 00:32:23 GMT+0530

**Inbound rules for security group**

Type	Protocol	Port range	Source
All TCP	TCP	0 - 65535	Anywhere
All traffic	All	0 - 65535	Anywhere

**Add rule**

ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#/clusters/javatechie-aws-cluster/runTask

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

**Cluster VPC\***: vpc-7c3a2414 (172.31.0.0/16)

**Subnets\***: subnet-bf50e7c4 (172.31.16.0/20) - ap-south-1c  
assign ipv6 on creation: Disabled

**Security groups\***: javatechie-all-access-security [Edit](#)

**Auto-assign public IP**: ENABLED

**Advanced Options**

**Tagging requires that you opt in to the new ARN and resource ID format.**  
The IAM user/role has not opted in to the new ARN format. Opt-in to the new format to use this feature. [Manage your opt-in settings](#).

[Cancel](#) [Run Task](#)

Feedback English (US) © 2008 - 2020 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Services Resource Groups

**Clusters**

Created tasks successfully  
Task Ids : ["7720e3d6-d137-4c38-9611-b99bf3fdb75b"]

Clusters > javatechie-aws-cluster

### Cluster : javatechie-aws-cluster

Get a detailed view of the resources on your cluster.

**Cluster ARN**: arn:aws:ecs:ap-south-1:688977122149:cluster/javatechie-aws-cluster  
**Status**: ACTIVE

**Registered container instances**: 0

- Pending tasks count: 1 Fargate, 0 EC2
- Running tasks count: 0 Fargate, 0 EC2
- Active service count: 0 Fargate, 0 EC2
- Draining service count: 0 Fargate, 0 EC2

[Services](#) [Tasks](#) [ECS Instances](#) [Metrics](#) [Scheduled Tasks](#) [Tags](#) [Capacity Providers](#)

[Run new Task](#) [Stop](#) [Stop All](#) [Actions](#)

Last updated on August 26, 2020 12:33:46 AM (0m ago)

← → C ap-south-1.console.aws.amazon.com/ecs/home?region=ap-south-1#clusters/javatechie-aws-cluster/tasks

Clusters

Task Definitions

Account Settings

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions

Cluster : javatechie-aws-cluster

Get a detailed view of the resources on your cluster.

Update Cluster Delete Cluster

Cluster ARN: arn:aws:ecs:ap-south-1:688977122149:cluster/javatechie-aws-cluster

Status: ACTIVE

Registered container instances: 0

Pending tasks count: 0 Fargate, 0 EC2

Running tasks count: 1 Fargate, 0 EC2

Active service count: 0 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

Services Tasks ECS Instances Metrics Scheduled Tasks Tags Capacity Providers

Run new Task Stop Stop All Actions Last updated on August 26, 2020 12:34:36 AM (0m ago)

Desired task status: Running Stopped

Last updated on August 26, 2020 12:34:36 AM (0m ago)

Filter in this page Launch type ALL < 1-1 > Page size 50

Task	Task definition	Container In...	Last status	Desired stat...	Started By	Group	Launch type...	Platform ver...
7720e3d8-d1...	springboot-do...	--	RUNNING	RUNNING	family:springb...	FARGATE	1.3.0	

Screenshot of the AWS CloudWatch Logs console showing log entries for a Spring Boot application running in an ECS task. A context menu is open over a log entry, with the 'Copy link address' option selected.

Name	Container Runtime I...	Stat...	Image	Image Digest	CP...	Har...	Ess...	Res...
javatechie-...	e3036936aeaf7248b3...	RU...	docker.io/javatechie/springboot-doc...		0	102...	true	da6...

**Details**

Network bindings - not configured  
 Environment Variables - not configured  
 Environment Files - not configured

Docker labels - not config  
 Extra hosts - not config  
 Mount Points - not config  
 Volumes from - not config  
 Ulimits - not configured  
**Log Configuration**

Log driver: awslogs [View logs in CloudWatch Logs](#)

Key Value

- awslogs-group /ecs/springboot-docker-ecs-example
- awslogs-region ap-south-1
- awslogs-stream-prefix ecs

Screenshot of the AWS CloudWatch Logs console showing log entries for a Spring Boot application running in an ECS task. A context menu is open over a log entry, with the 'Copy link address' option selected.

Name	Container Runtime I...	Stat...	Image	Image Digest	CP...	Har...	Ess...	Res...
javatechie-...	e3036936aeaf7248b3...	RU...	docker.io/javatechie/springboot-doc...		0	102...	true	da6...

**AWS Marketplace**

- Discover software
- Subscriptions

**Platform version** 1.3.0

**Task definition** springboot-docker-ecs-example:1

- Group** family:springboot-docker-ecs-example
- Task role** None
- Last status** RUNNING
- Desired status** RUNNING
- Created at** 2020-08-26 00:33:45 +0530
- Started at** 2020-08-26 00:34:14 +0530

**Network**

Network mode awsvpc

ENI Id eni-099a617633d2ef389

Subnet Id subnet-bf58e7c4

Private IP 172.31.29.197

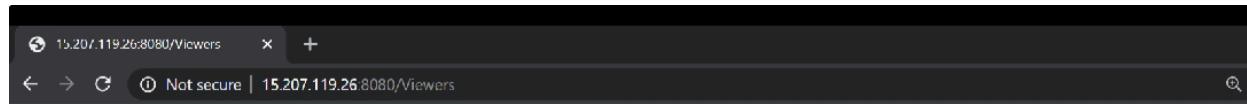
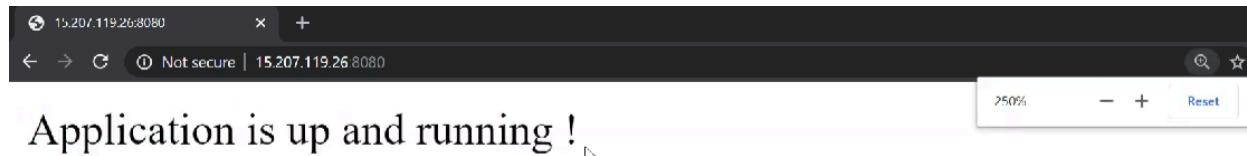
Public IP 35.207.119.26

Mac address 06:f1:84:5c:ac:f4

**Containers**

Last updated on August 26, 2020 12:34:16 AM (3m ago)

We copy the public ip along with port number then we can see the application result



# Hi Viewers Welcome to javatechie AWS ECS Example

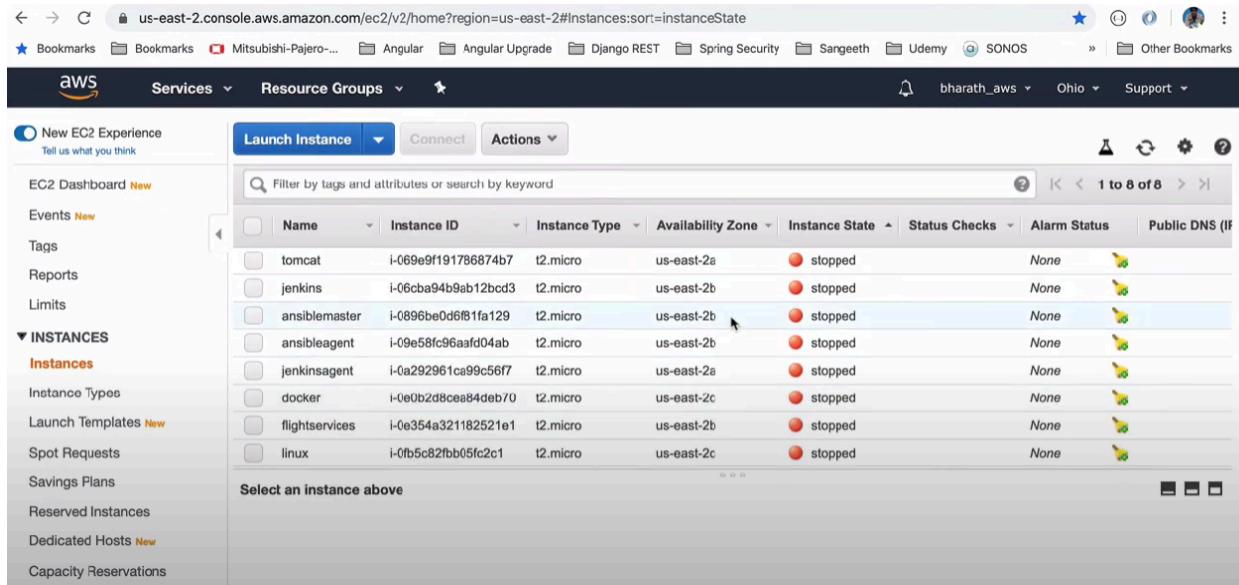
**Just other way , where we use EC2 ,S3 instead ECS**

We need to sign in to aws login page

The screenshot shows the AWS Services page with the following layout:

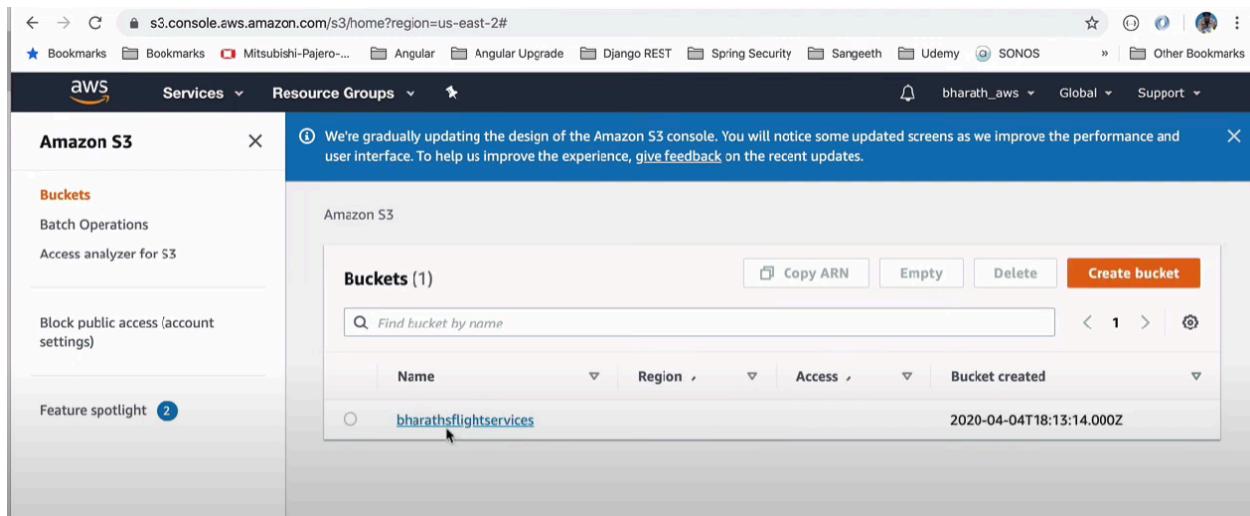
- Services** dropdown menu:
  - Compute
  - Storage
  - Analytics
  - End User Computing
  - Internet Of Things
- Resource Groups** dropdown menu:
  - History
  - Find a service by name or feature (for example, EC2, S3 or VM, storage).
  - Group A-Z
- EC2** link
- Console Home** link
- Compute** section:
  - EC2
  - Lightsail
  - Lambda
  - Batch
  - Elastic Beanstalk
  - Serverless Application Repository
  - AWS Outposts
  - EC2 Image Builder
- Blockchain** section:
  - Amazon Managed Blockchain
- Analytics** section:
  - Athena
  - EMR
  - CloudSearch
  - Elasticsearch Service
  - Kinesis
  - QuickSight
  - Data Pipeline
  - AWS Data Exchange
  - AWS Glue
  - AWS Lake Formation
  - MSK
- End User Computing** section:
  - WorkSpaces
  - AppStream 2.0
  - WorkDocs
  - WorkLink
- Internet Of Things** section:
  - IoT Core
  - FreeRTOS
  - IoT 1-Click
  - IoT Analytics
  - IoT Device Defender
  - IoT Device Management
  - IoT Events
- Storage** section:
  - S3
  - EFS
- Management & Governance** section:
  - AWS Organizations
  - CloudWatch
- Security, Identity, &** section

Here we need mainly EC2 instances under compute and here will create instances/machines where we can deploy our applications.



The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES' (selected), 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'. The main area has a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IF). There are 8 instances listed, all in the 'stopped' state. The instance 'bharathflightservices' is highlighted with a cursor over its 'us-east-2b' entry. A message at the bottom says 'Select an instance above'.

S3 under storage is used where we can apply jar files by creating buckets.



The screenshot shows the AWS S3 Buckets page. On the left, there's a sidebar with 'Amazon S3' selected, followed by 'Buckets', 'Batch Operations', 'Access analyzer for S3', 'Block public access (account settings)', and 'Feature spotlight'. The main area has a heading 'Amazon S3' and a table titled 'Buckets (1)'. The table has columns: Name, Region, Access, and Bucket created. It shows one bucket named 'bharathflightservices' created on '2020-04-04T18:13:14.000Z'. Buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket' are visible.

To create a new instance click on launch instance and complete the 7 step mentioned.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP)
test	i-012d5256477f7ecfa	t2.micro	us-east-2c	stopped	None	None	
tomcat	i-059e9f191786874b7	t2.micro	us-east-2a	stopped	None	None	
linux	i-05cb94b9ab12bcd3	t2.micro	us-east-2b	stopped	None	None	
ansiblemaster	i-0396be0d6f81fa129	t2.micro	us-east-2b	stopped	None	None	
ansibleagent	i-09e58fc96aaf04ab	t2.micro	us-east-2b	stopped	None	None	

Select the prebuilt AMI and other how many instance(machines) you need

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start

- My AMIs
- Amazon Linux (selected)
- AWS Marketplace
- Community AMIs
- Free tier only

1 to 40 of 40 AMIs

	<b>Amazon Linux 2 AMI (HVM), SSD Volume Type</b> - ami-0e01ce4ee18447327 (64-bit x86) / ami-03201f374ab66a29e (64-bit Arm) Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. Root device type: ebs Virtualization type: hvm ENA Enabled: Yes	<input type="button" value="Select"/>
	<b>Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type</b> - ami-01b01bbd08f24c7a8 The Amazon Linux AMI is FUD based. AWS recommends using the default instance AMI instead. Details	<input type="button" value="Select"/>

## Step 2: Choose an Instance Type

computing needs.

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## Step 3: Configure Instance Details

Number of instances  Launch into Auto Scaling Group

Purchasing option  Request Spot instances

Network

Subnet

Auto-assign Public IP

Placement group  Add instance to placement group

Capacity Reservation

IAM role

Cancel Previous Review and Launch Next: Add Storage

← → C us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:

★ Bookmarks Bookmarks Mitsubishi-Pajero... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS » | Other Bookmarks

aws Services Resource Groups ★ bharath\_aws Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## Step 4: Add Storage

edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0f54692056aaa4c20	<input type="text" value="8"/>	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	<input type="checkbox"/> Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

**Step 5: Add Tags**

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes
This resource currently has no tags					
Choose the Add tag button or <a href="#">click</a> to add a Name tag. Make sure your <a href="#">IAM policy</a> includes permissions to create tags.					
<a href="#">Add Tag</a> (Up to 50 tags maximum)					

**Step 6: Configure Security Group**

Assign a security group:  Create a new security group  
 Select an existing security group

Security group name: launch-wizard-19

Description: launch-wizard-19 created 2020-04-06T22:39:02.694+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

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

[Cancel](#) [Previous](#) [Review and Launch](#)

**Step 7: Review Instance Launch**

t2.micro	Variable	1	1	EBS only	-	Low to Moderate
----------	----------	---	---	----------	---	-----------------

**Security Groups** [Edit security groups](#)

Security group name: launch-wizard-19  
 Description: launch-wizard-19 created 2020-04-06T22:39:02.694+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

**Instance Details** [Edit instance details](#)

**Storage** [Edit storage](#)

[Cancel](#) [Previous](#) [Launch](#)

Screenshot of the AWS Launch Instance Wizard Step 7: Review Instance. The user is creating a new key pair named "awskeys". A note states: "You have to download the private key file (\*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created." The "Download Key Pair" button is visible.

Screenshot of the AWS Launch Instance Wizard Step 8: Launch Status. It shows a green success message: "Your instances are now launching" and "The following instance launches have been initiated: i-02ead5da393ae2f1e View launch log". Below this, there is a blue box with a tip: "Get notified of estimated charges" and "Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier)". Further down, there is a section titled "How to connect to your instances" with instructions and links to resources like "How to connect to your Linux instance" and "Amazon EC2: User Guide".

The screenshot shows the AWS Management Console with the EC2 service selected. On the left, the navigation pane is open, showing options like New EC2 Experience, EC2 Dashboard, Events, Tags, Reports, Limits, and INSTANCES. Under INSTANCES, the 'Instances' option is selected, showing sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main content area displays a table of instances. One instance is listed: i-0e354a321182521e1, which is an t2.micro instance running in the us-east-2b availability zone. The Public DNS is ec2-3-17-144-195.us-east-2.compute.amazonaws.com. The instance status is shown as 'running' with a green dot. Below the table, a detailed view for the selected instance is provided, including fields for Instance ID, Instance state, Instance type, and various network and finding details.

Click on connect button to connect with new instance(vm/machine)

The screenshot shows the 'Connect to your instance' wizard. The connection method is set to 'A standalone SSH client'. The instructions for accessing the instance are as follows:

- Open an SSH client. (find out how to connect using PuTTY)
- Locate your private key file (awskeys.pem). The wizard automatically detects the key you used to launch the instance.
- Your key must not be publicly viewable for SSH to work. Use this command if needed:  
chmod 400 awskeys.pem
- Connect to your instance using its Public DNS:  
ec2-3-17-144-195.us-east-2.compute.amazonaws.com

The example command shown is:

```
ssh -i "awskeys.pem" ec2-user@ec2-3-17-144-195.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#Instances:search=i-0e354a321182521e1;sort=instanceState

**Connect to your instance**

**Connection method**

- A standalone SSH client ⓘ
- Session Manager ⓘ
- EC2 Instance Connect (browser-based SSH connection) ⓘ

Connect using a custom user name, or default to the user name for the AMI used to launch the instance. [Learn more](#)

User name: ec2-user

**Close** **Connect**

Description	Status Checks	Monitoring	Tags
Instance ID: i-0e354a321182521e1	Public DNS (IPv4): ec2-3-17-144-195.us-east-2.compute.amazonaws.com	IPv4 Public IP: 3.17.144.195	IPv6 IPs: -
Instance state: running			
Instance type: t2.micro			

## Upload jar file into S3

s3.console.aws.amazon.com/s3/home?region=us-east-2

**History**

Find a service by name or feature (for example, EC2, S3 or VM, storage).

**Group A-Z**

S3	Compute	Blockchain	Analytics	End User Computing
Console Home	EC2	Amazon Managed Blockchain	Athena	WorkSpaces
EC2	Lightsail ⓘ	Satellite	EMR	AppStream 2.0
CodeDeploy	Lambda	Ground Station	CloudSearch	WorkDocs
	Batch		Elasticsearch Service	WorkLink
	Elastic Beanstalk		Kinesis	
	Serverless Application Repository		QuickSight ⓘ	
	AWS Outposts	Quantum Technologies	Data Pipeline	
	EC2 Image Builder	Amazon Braket ⓘ	AWS Data Exchange	
			AWS Glue	
			AWS Lake Formation	
			MSK	
	Storage	Management & Governance		Internet Of Things
	S3	AWS Organizations		IoT Core
	EFS	CloudWatch		FreeRTOS
				IoT 1-Click
				IoT Analytics
				IoT Device Defender
				IoT Device Management
				IoT Events

s3.console.aws.amazon.com/s3/home?region=us-east-2

Bookmarks Bookmarks Mitsubishi-Pajero-... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS Other Bookmarks

Services Resource Groups

Amazon S3

We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.

Buckets

Batch operations Access analyzer for S3

Block public access (account settings)

Feature spotlight

Amazon S3

Buckets (0)

Copy ARN Empty Delete Create bucket

Find bucket by name

Name	Region	Access	Bucket created
No buckets			

You don't have any buckets.

Create bucket

This screenshot shows the AWS S3 console homepage. On the left, there's a sidebar with links for 'Buckets', 'Batch operations', 'Access analyzer for S3', and 'Block public access (account settings)'. The main area is titled 'Amazon S3' and shows a message about the console's update. Below that is a table titled 'Buckets (0)' with columns for Name, Region, Access, and Bucket created. A search bar labeled 'Find bucket by name' is above the table. A large button labeled 'Create bucket' is at the bottom of the table area.

s3.console.aws.amazon.com/s3/bucket/create?region=us-east-2

Bookmarks Bookmarks Mitsubishi-Pajero-... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS Other Bookmarks

Services Resource Groups

Amazon S3 > Create bucket

We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.

Create bucket

General configuration

Bucket name

bharathsflight

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

Region

US East (Ohio) us-east-2

This screenshot shows the 'Create bucket' configuration page. It has a header 'Create bucket' and a section title 'General configuration'. Under 'Bucket name', the value 'bharathsflight' is entered into a text input field, which is highlighted with a blue border. Below it, a note says 'Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming'. Under 'Region', a dropdown menu shows 'US East (Ohio) us-east-2'.

s3.console.aws.amazon.com/s3/buckets/bharathsflightservices?region=us-east-2

Bookmarks Bookmarks Mitsubishi-Pajero-... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS Other Bookmarks

AWS Services Resource Groups

**Amazon S3**

Buckets

Batch operations Access analyzer for S3

Block public access (account settings)

Feature spotlight

We're gradually updating the design of the Amazon S3 console. You will notice some updated screens as we improve the performance and user interface. To help us improve the experience, give feedback on the recent updates.

Successfully created bucket bharathsflightservices

To upload files and folders, or to configure additional bucket settings such as Bucket Versioning, tags, and default encryption, choose Go to bucket details.

Go to bucket details

Amazon S3

Buckets (1)

Copy ARN Empty Delete Create bucket

Find bucket by name

Name	Region	Access	Bucket created
bharathsflightservices	US East (Ohio) us-east-2	Objects can be public	2020-04-04T18:13:14.000Z

Online Courses - Anytime, Anyw... X S3 Management Console +

Bookmarks Bookmarks Mitsubishi-Pajero-... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS Other Bookmarks

AWS Services Resource Groups

Amazon S3 > bharathsflightservices

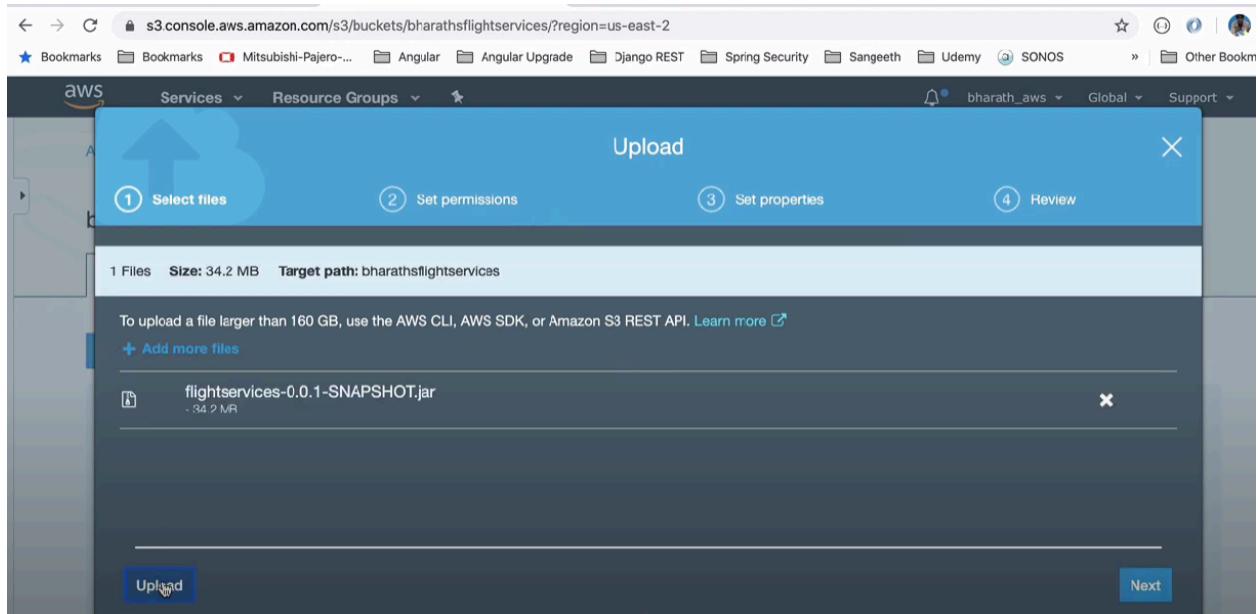
bharathsflightservices

Overview Properties Permissions Management Access points

Type a prefix and press Enter to search. Press ESC to clear.

Upload + Create folder Download Actions

Name	Last modified	Size	Storage class
------	---------------	------	---------------



The screenshot shows the AWS S3 console with the file 'flightservices-0.0.1-SNAPSHOT.jar' listed in the 'bharathsflightservices' bucket. The file details are as follows:

Key	flightservices-0.0.1-SNAPSHOT.jar
Size	34.2 MB
Expiration date	N/A
Expiration rule	N/A
ETag	4c365d6af4818e363a0a14e4a03b4744-3
Last modified	Apr 4, 2020 11:43:35 PM GMT+0530
Object URL	<a href="https://bharathsflightservices.s3-us-east-2.amazonaws.com/flightservices-0.0.1-SNAPSHOT.jar">https://bharathsflightservices.s3-us-east-2.amazonaws.com/flightservices-0.0.1-SNAPSHOT.jar</a>

Now run application using DNS url of EC2

us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#instances:sort=instanceState

Bookmarks Bookmarks Mitsubishi-Pajero... Angular Angular Upgrade Django REST Spring Security Sangeeth Udemy SONOS Other Bookmark

bharath\_aws Ohio Support

New EC2 Experience Tell us what you think

Services Resource Groups

Launch Instance Connect Actions

EC2 Dashboard Events Tags Reports Limits

INSTANCES Instances Instance Types Launch Templates New

Spot Requests Savings Plans Reserved Instances Dedicated Hosts New

Feedback English (US)

Filter by tags and attributes or search by keyword

1 to 11 of 11

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP)
flightservices	i-0e354a321182521e1	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-3-17-144-195.us-east-2.compute.amazonaws.com
test	i-012d5256477f7ecfa	t2.micro	us-east-2c	stopped	None	None	ec2-3-17-144-195.us-east-2.compute.amazonaws.com
tomcat	i-069e9f191786874b7	t2.micro	us-east-2a	stopped	None	None	ec2-3-17-144-195.us-east-2.compute.amazonaws.com
flightservices	i-02ab04b09119119ad9	t2.micro	us-east-2b	terminated	None	None	ec2-3-17-144-195.us-east-2.compute.amazonaws.com

Instance: i-0e354a321182521e1 (flightservices) Public DNS: ec2-3-17-144-195.us-east-2.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-0e354a321182521e1 Public DNS (IPv4): ec2-3-17-144-195.us-east-2.compute.amazonaws.com

Instance state: running IPv4 Public IP: 3.17.144.195

Instance type: t2.micro IPv6 IPs: -

Finding Opt-in to AWS Compute Optimizer for Elastic IPs

© 2008 - 2020, Amazon Internet Services Private Ltd, or its affiliates. All rights reserved. Privacy Policy Terms of Use