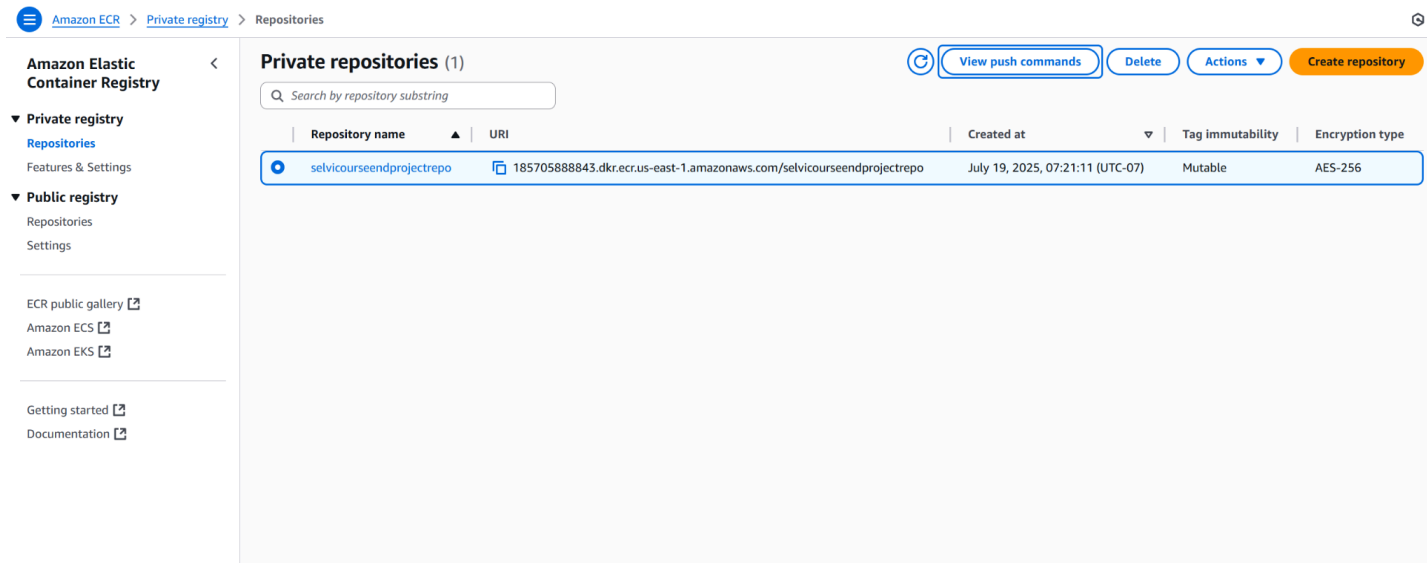
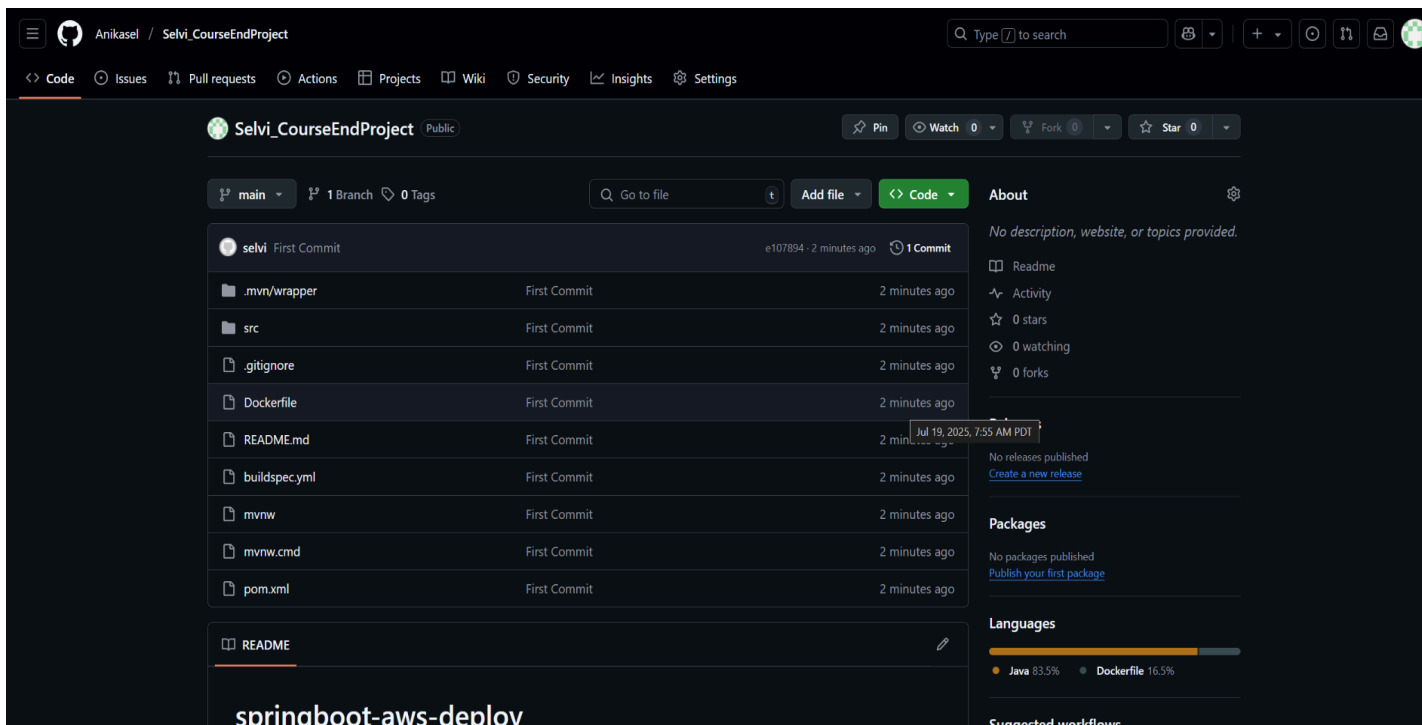


Step 1: Setup an AWS Elastic Container Registry with a repository



Step 2: Setup a GitHub repository and clone it to local machine



```
MINGW64/c/Selvi/CourseEndProject/Selvi_CourseEndProject

git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'selvi@Selvi-TUF.(none)')

selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$ git config user.name "selvi"

selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$ git config user.email "selvi@abc.com"

selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$ git commit -m "First Commit"
[main (root-commit) e107894] First Commit
13 files changed, 708 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/maven-wrapper.jar
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 Dockerfile
create mode 100644 README.md
create mode 100644 buildspec.yml
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/com/example/springbootawsdeploy/SpringbootAwsDeployApplication.java
create mode 100644 src/main/java/com/example/springbootawsdeploy/TestController.java
create mode 100644 src/main/resources/application.properties
create mode 100644 src/test/java/com/example/springbootawsdeploy/SpringbootAwsDeployApplicationTests.java

selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$ git push
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 16 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (29/29), 64.44 KiB | 10.74 MiB/s, done.
Total 29 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Anikase1/Selvi_CourseEndProject.git
 * [new branch]      main -> main

selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$
```

Step 3: Create a Code Build project

The screenshot shows the AWS CodeBuild console. On the left is a navigation menu with options like Source, Artifacts, Build, and Deploy. The main area displays a green banner indicating a successful build. Below this, the build status is shown as 'Succeeded' with a green checkmark. The build details table lists the status, initiator, build ARN, start time, end time, and build number. The build logs tab is selected, showing a table of build phases: SUBMITTED, QUEUED, and PROVISIONING, all with a status of 'Succeeded'.

Build started
You have successfully started the following build: Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2

[Developer Tools](#) > [CodeBuild](#) > [Build projects](#) > [Selvi-CodeBuildProject](#) > Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2

Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2

Build status

| | | |
|---------------------------------|---------------------------------|--|
| Status | Initiator | Build ARN |
| ✔ Succeeded | SelviRole/odl_user_1798140 | arn:aws:codebuild:us-east-1:185705888843:build/Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2 |
| Start time | End time | Build number |
| Jul 19, 2025 8:14 AM (UTC-7:00) | Jul 19, 2025 8:15 AM (UTC-7:00) | 1 |

[Build logs](#) [Phase details](#) [Reports](#) [Environment variables](#) [Build details](#) [Resource utilization](#)

| Name | Status | Context | Duration | Start time |
|--------------|-------------|---------|----------|---------------------------------|
| SUBMITTED | ✔ Succeeded | - | <1 sec | Jul 19, 2025 8:14 AM (UTC-7:00) |
| QUEUED | ✔ Succeeded | - | <1 sec | Jul 19, 2025 8:14 AM (UTC-7:00) |
| PROVISIONING | ✔ Succeeded | - | 4 secs | Jul 19, 2025 8:14 AM (UTC-7:00) |

Step 4: Establish an ECS cluster

The screenshot shows the Amazon Elastic Container Service (ECS) console. The left navigation menu includes Clusters, Namespaces, Task definitions, Account settings, Amazon ECR, Repositories, AWS Batch, Documentation, Discover products, and Subscriptions. The main area displays the 'SelviECScluster' overview. A notification at the top states that Amazon ECS changed the default log driver mode from blocking to non-blocking to improve application availability during CloudWatch outages. The cluster overview shows the ARN, status (Active), CloudWatch monitoring (Default), and registered container instances (0). The services section shows a table with columns for Service name, ARN, Status, Service..., Created at, Deployments and tasks, and Last deployment. The table is currently empty, displaying 'No services to display.'

[Amazon Elastic Container Service](#) > [Clusters](#) > [SelviECScluster](#) > Services

On June 25, 2025, Amazon ECS changed the default log driver mode from blocking to non-blocking to improve application availability during CloudWatch outages. [Learn more](#)

SelviECScluster

Last updated July 19, 2025 at 09:06 (UTC-7:00) [Update cluster](#) [Delete cluster](#)

Cluster overview

| | | | |
|--|----------|-----------------------|--------------------------------|
| ARN | Status | CloudWatch monitoring | Registered container instances |
| arn:aws:ecs:us-east-1:185705888843:cluster/SelviECScluster | ✔ Active | Default | - |
| Services | Tasks | | |
| Draining | Active | Pending | Running |
| - | - | - | - |

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

Services (0) [Info](#)

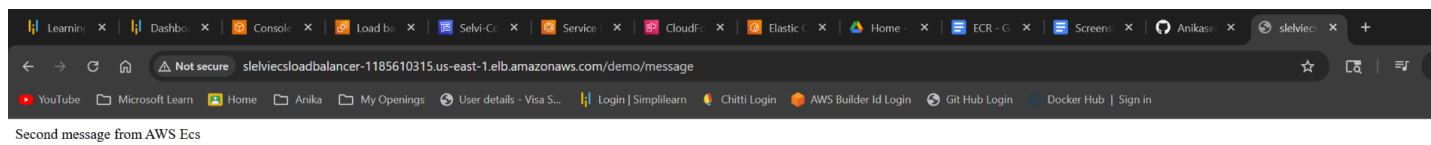
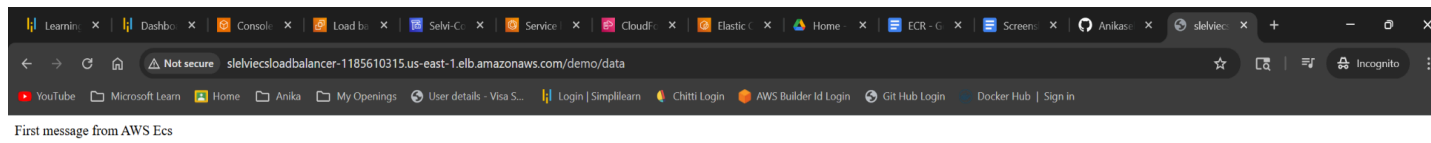
[Manage tags](#) [Update](#) [Delete service](#) [Create service](#)

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

Filter services by value

| Service name | ARN | Status | Service... | Created at | Deployments and tasks | Last deployment |
|-------------------------|-----|--------|------------|------------|-----------------------|-----------------|
| No services to display. | | | | | | |

Step 5: Validate the Application Deployment



Step 6. Construct and Execute a CodePipeline to automate the deployment process

