

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

The screenshot shows the AWS ECR console. On the left, there's a sidebar with navigation links for Amazon Elastic Container Registry, Private registry (Repositories, Features & Settings), Public registry (Repositories, Settings), ECR public gallery, Amazon ECS, and Amazon EKS. The main area is titled "Private repositories (1)" and shows a table with one item: "selvicourseendprojectrepo". The table includes columns for Repository name, URI, Created at, Tag immutability, and Encryption type. The repository details show it was created on July 19, 2025, at 07:21:11 (UTC-07), is mutable, and uses AES-256 encryption.

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

The screenshot shows a GitHub repository page for "Selvi.CourseEndProject". The repository is public and has 1 branch and 0 tags. The code tab is selected, showing a list of files: .mvn/wrapper, src, .gitignore, Dockerfile, README.md, buildspec.yml, mvnw, mvnw.cmd, and pom.xml. All files were committed by "selvi" as their first commit, 2 minutes ago. The README file contains the text "springboot-aws-deploy". The repository has 0 stars, 0 forks, and 0 releases published. It also has 0 watching users. The Languages section shows Java at 83.5% and Dockerfile at 16.5%. A "Suggested workflows" section is visible at the bottom.

```
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/Anikasel/Selvi_CourseEndProject.git
 * [new branch]      main -> main

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

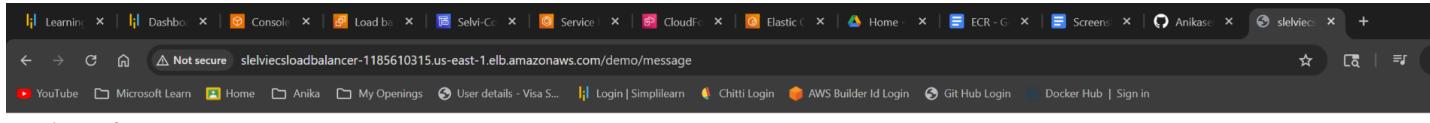
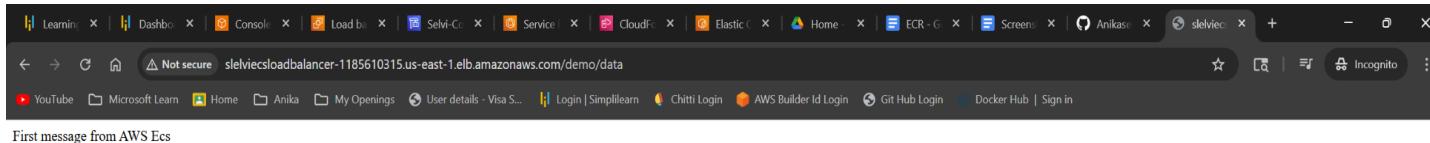
Step 3: Create a Code Build project

The screenshot shows the AWS CodeBuild console. The left sidebar has a tree view with 'CodeBuild' selected. Under 'Build projects', 'Build project' is also selected. The main area shows a green banner at the top stating 'Build started' and 'You have successfully started the following build: Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2'. Below this, the build details are listed: Status (Succeeded), Initiator (SelviRole/odl_user_1798140), Build ARN (arn:aws:codebuild:us-east-1:185705888843:build/Selvi-CodeBuildProject:441d022e-eb52-44bf-9841-19682b0e16c2), Start time (Jul 19, 2025 8:14 AM (UTC-7:00)), End time (Jul 19, 2025 8:15 AM (UTC-7:00)), and Build number (1). A table below shows the Phase details: SUBMITTED (Status Succeeded, Duration <1 sec, Start time Jul 19, 2025 8:14 AM (UTC-7:00)); QUEUED (Status Succeeded, Duration <1 sec, Start time Jul 19, 2025 8:14 AM (UTC-7:00)); and PROVISIONING (Status Succeeded, Duration 4 secs, Start time Jul 19, 2025 8:14 AM (UTC-7:00)).

Step 4: Establish an ECS cluster

The screenshot shows the AWS Elastic Container Service (ECS) console. The left sidebar has 'Clusters' selected under 'Amazon Elastic Container Service'. The main area shows the 'SelviECSCluster' overview. The 'Cluster overview' section includes the ARN (arn:aws:ecs:us-east-1:185705888843:cluster/SelviECSCluster), Status (Active), CloudWatch monitoring (Default), and Registered container instances (-). The 'Services' section shows Draining (-) and Active. The 'Tasks' section shows Pending and Running. Below this, a table lists Services (0) with columns for Service name, ARN, Status, Service..., Created at, Deployments and tasks, and Last deployment. A note at the top states: '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](#)'.

Step 5: Validate the Application Deployment



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

