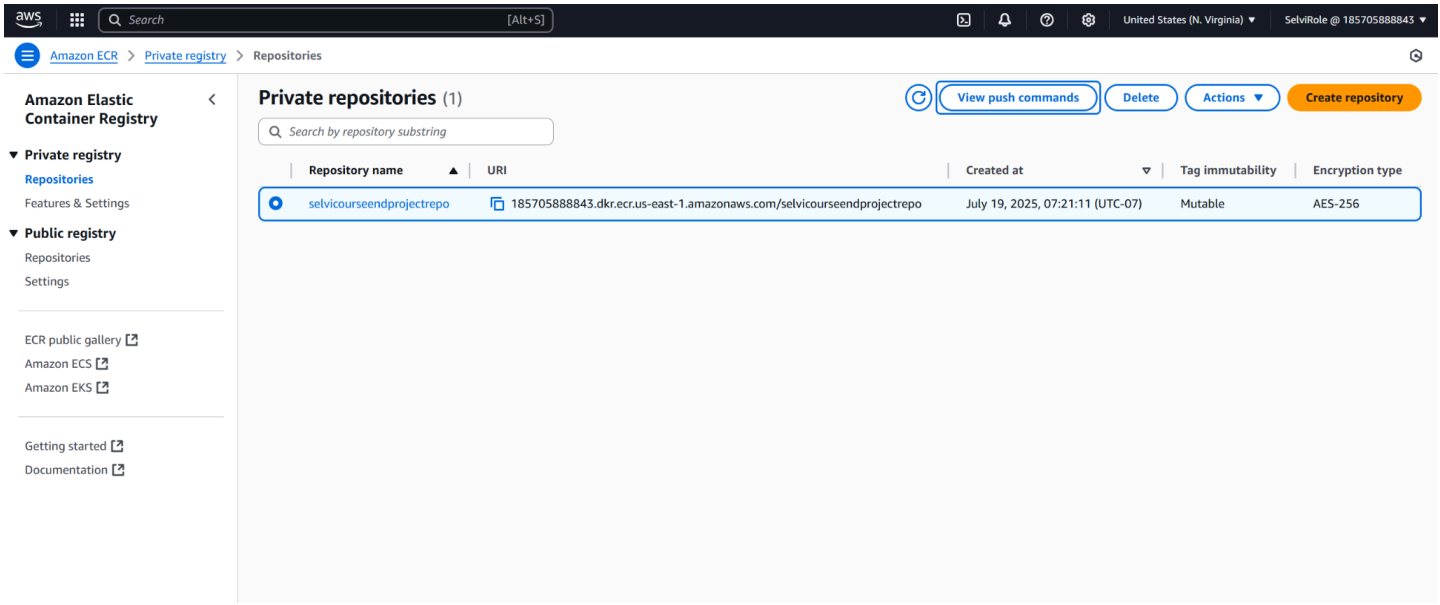
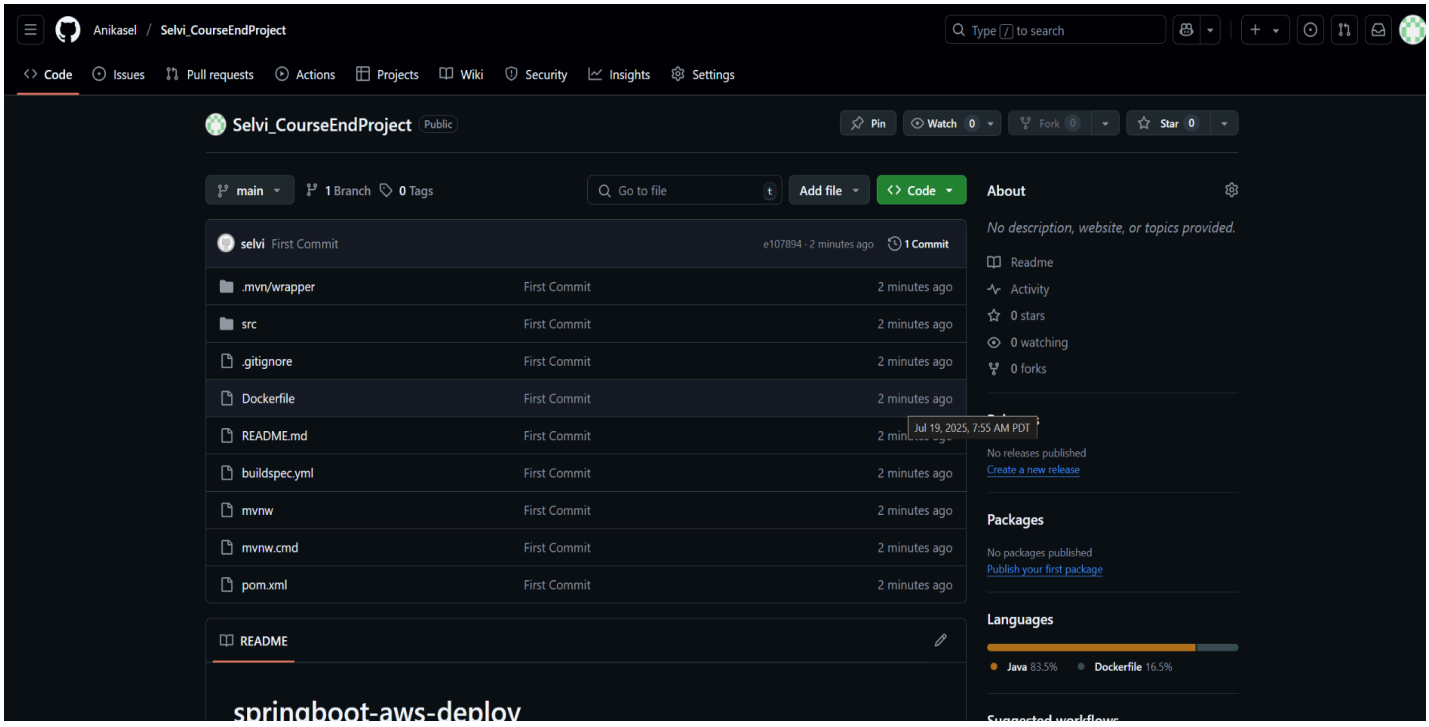
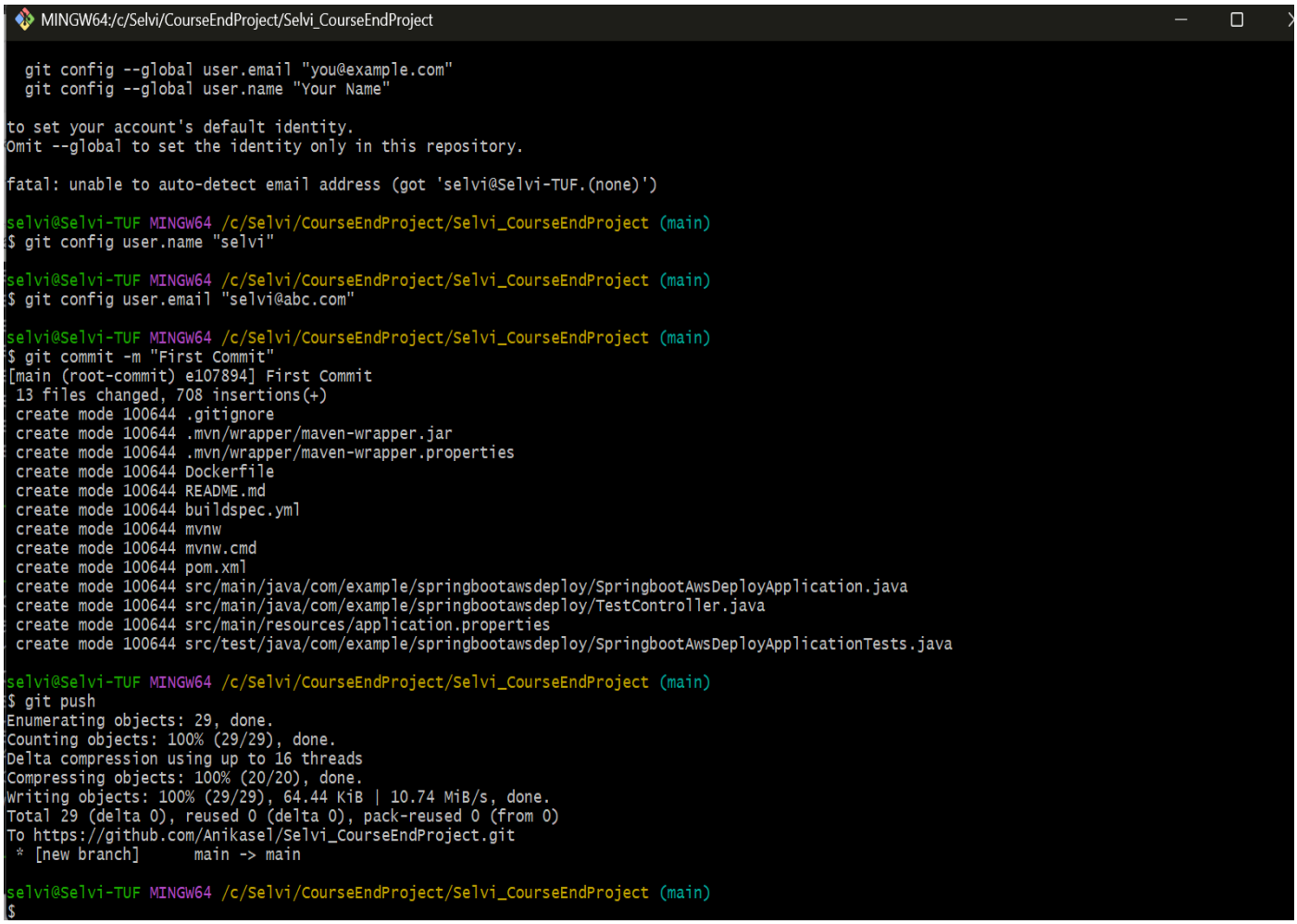
Step 1: Setup an AWS Elastic Container Registry with a repository

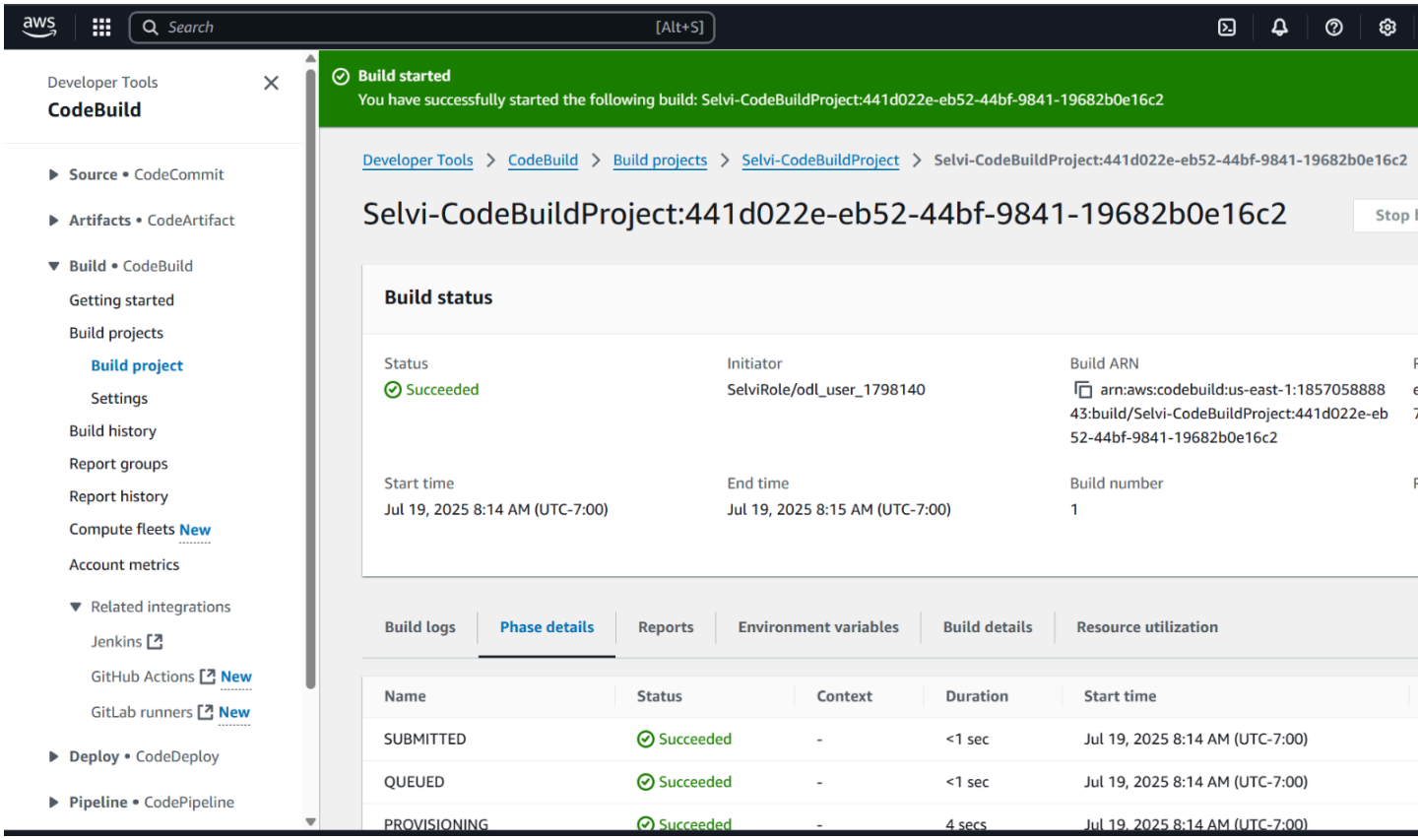


Step 2: Setup a GitHub repository and clone it to local machine

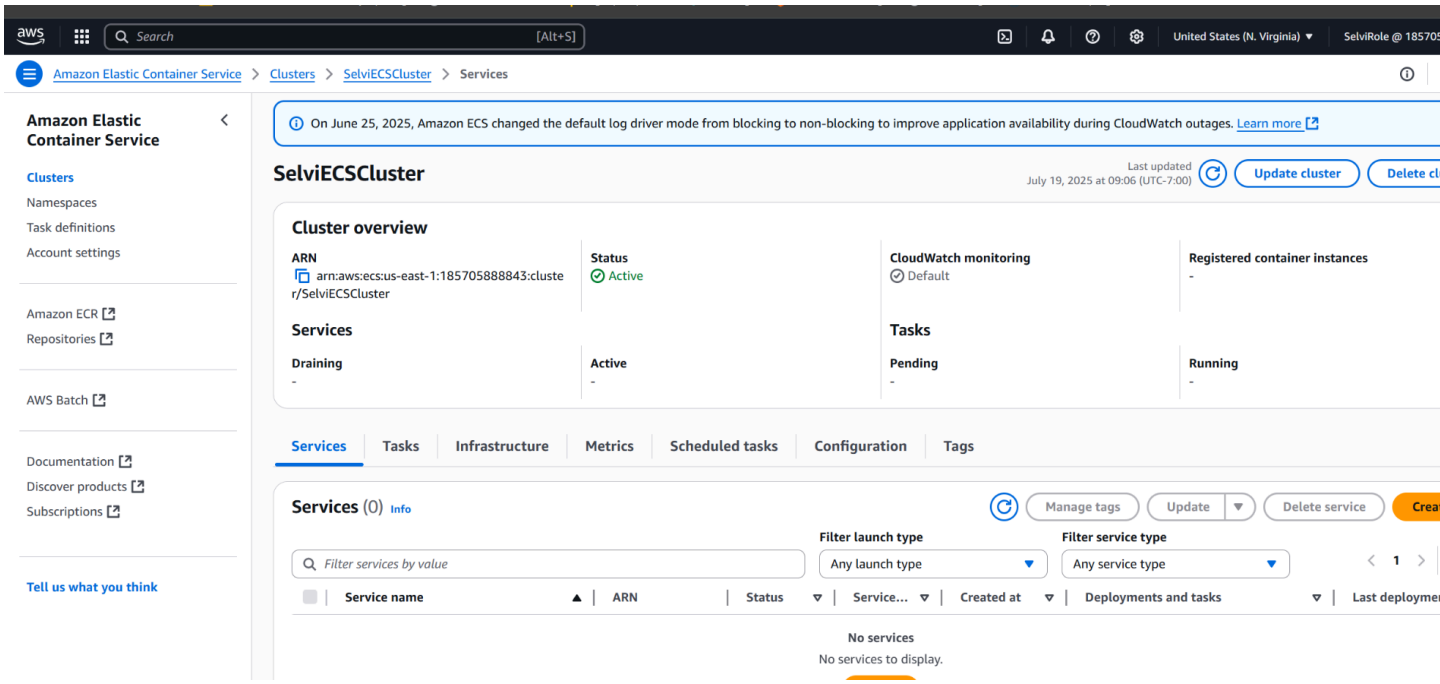



MINGW64:/c/Selvi/CourseEndProject/Selvi_CourseEndProject
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'selvi@Selvi-TUF.(none)')
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/Anikasel/Selvi_CourseEndProject.git
* [new branch] main -> main
selvi@Selvi-TUF MINGW64 /c/Selvi/CourseEndProject/Selvi_CourseEndProject (main)
$

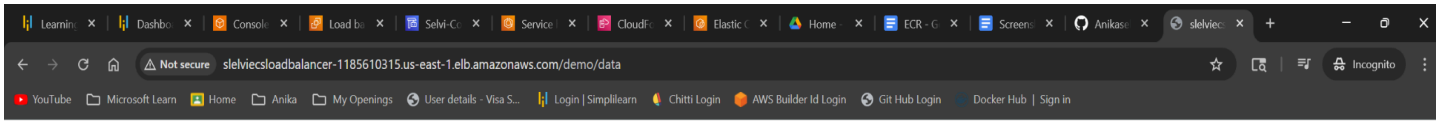
Step 3: Create a Code Build project



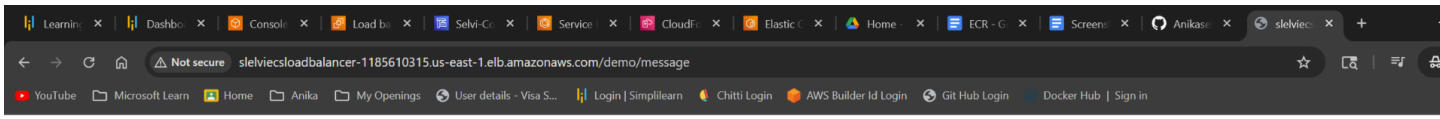
Step 4: Establish an ECS cluster



Step 5: Validate the Application Deployment



First message from AWS Ecs



Second message from AWS Ecs

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