Build and Deploy Dockerized React JS Application with AWS CI-CD and ElasticBeanstalk

Description:

Build and deploy applications with AWS native CI-CD methodologies.

Steps:

Application and Dockerfile Creation

Create a simple reactjs application and create Dockerfile for that application.

Code Storage (CodeCommit)

Create a Codecommit repository in AWS and push created Local codes into Codecommit using

CLI commands.

Docker Build (CodeBuild)

Use CodeBuild service to build the code using dockerfile and push created docker image into

docker hub (Use buildspec.yml file).

Continuous Deployment (ElasticBeanstalk)

Create a ElasticBeanstalk docker based application to deploy our reactjs application.

Continuous Integration (CodePipeline)

Create a Codepipeline and choose source as codecommit and build stage as codebuild and

deploy stage as elasticbeanstalk to deploy our created Docker application. Deploy using

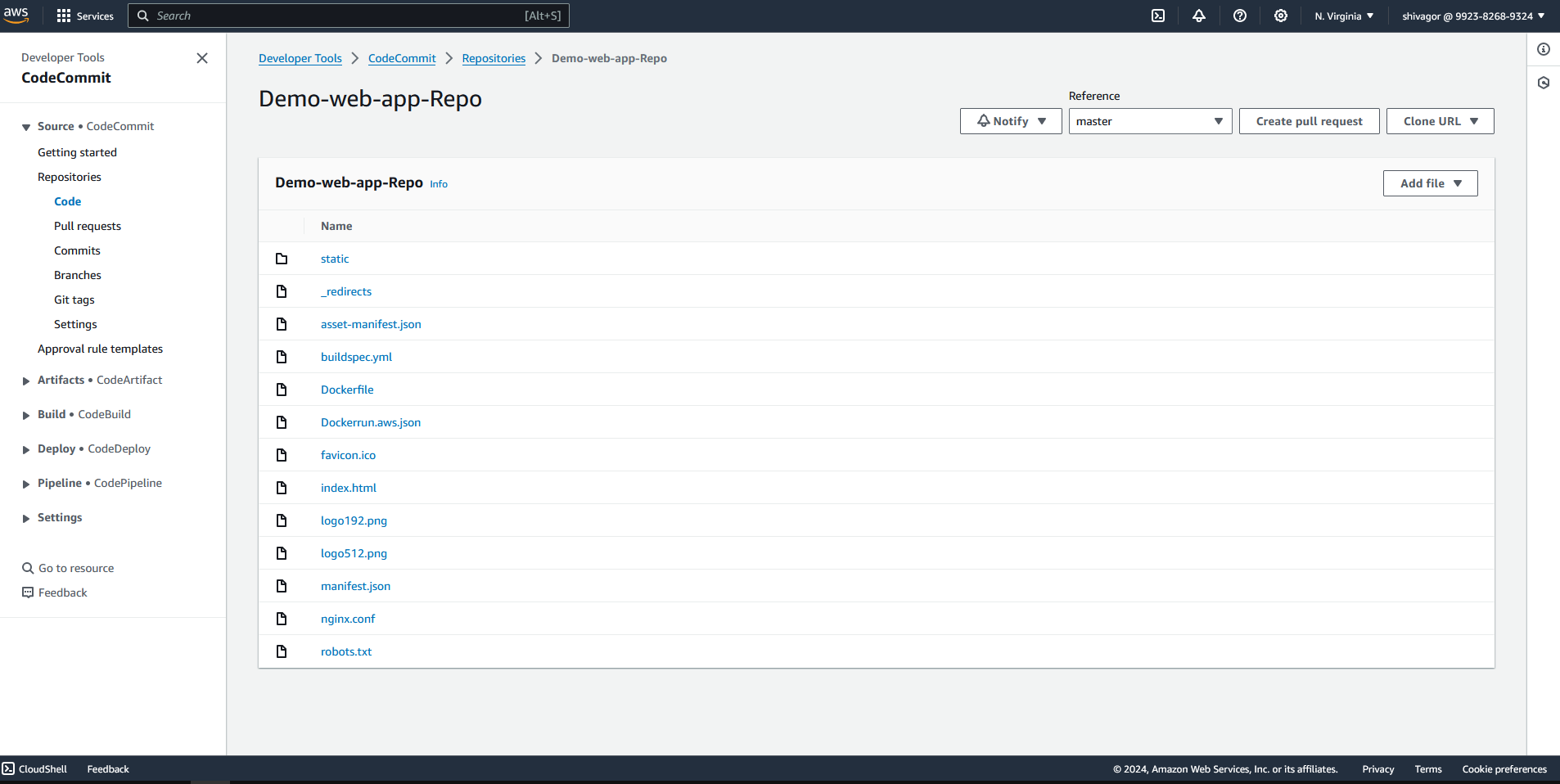
Dockerrun.aws.json file.

Used existing ecom react based app for CI/CD with codebuild, code commit , elastic beanstalk and codepipeline deploye using Dockerrun.aws.json file

Created a repo in code commit named [Demo-web-app-Repo](https://us-east-1.console.aws.amazon.com/codesuite/codecommit/repositories/Demo-web-app-Repo/browse?region=us-east-1)

Logged in with gitbash with aws credentials of git

Used cli for uploading source code to codecommit repo



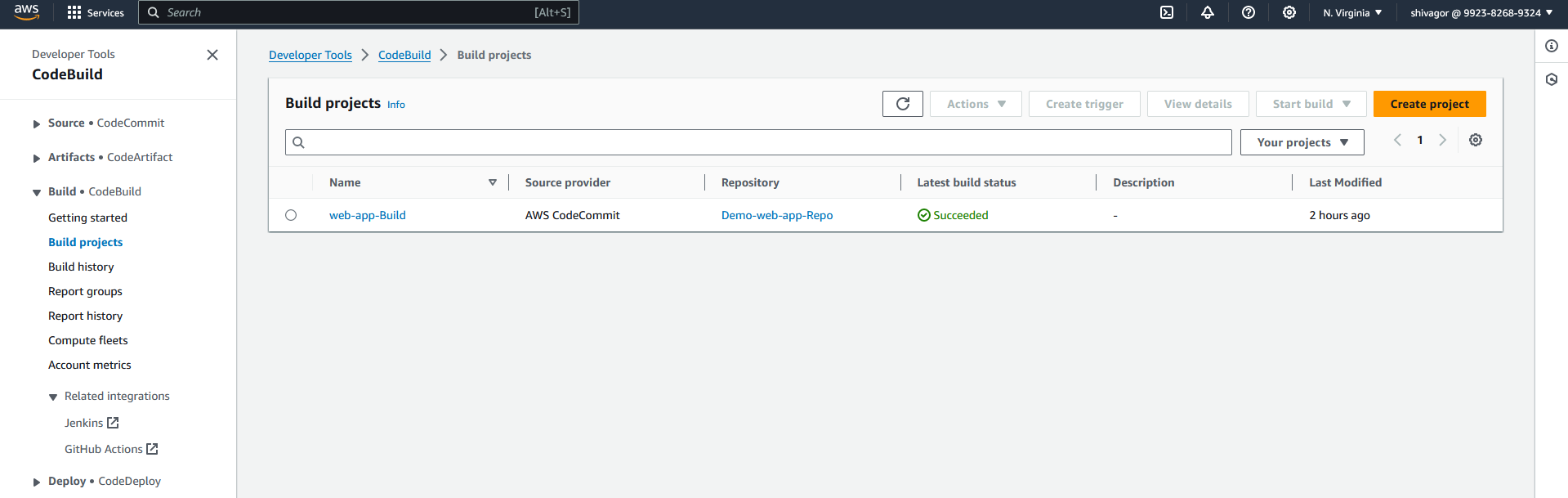
Create a build called web-app-build source provider as aws code commit

Leave every thing in env as default and change service role or create a new service role

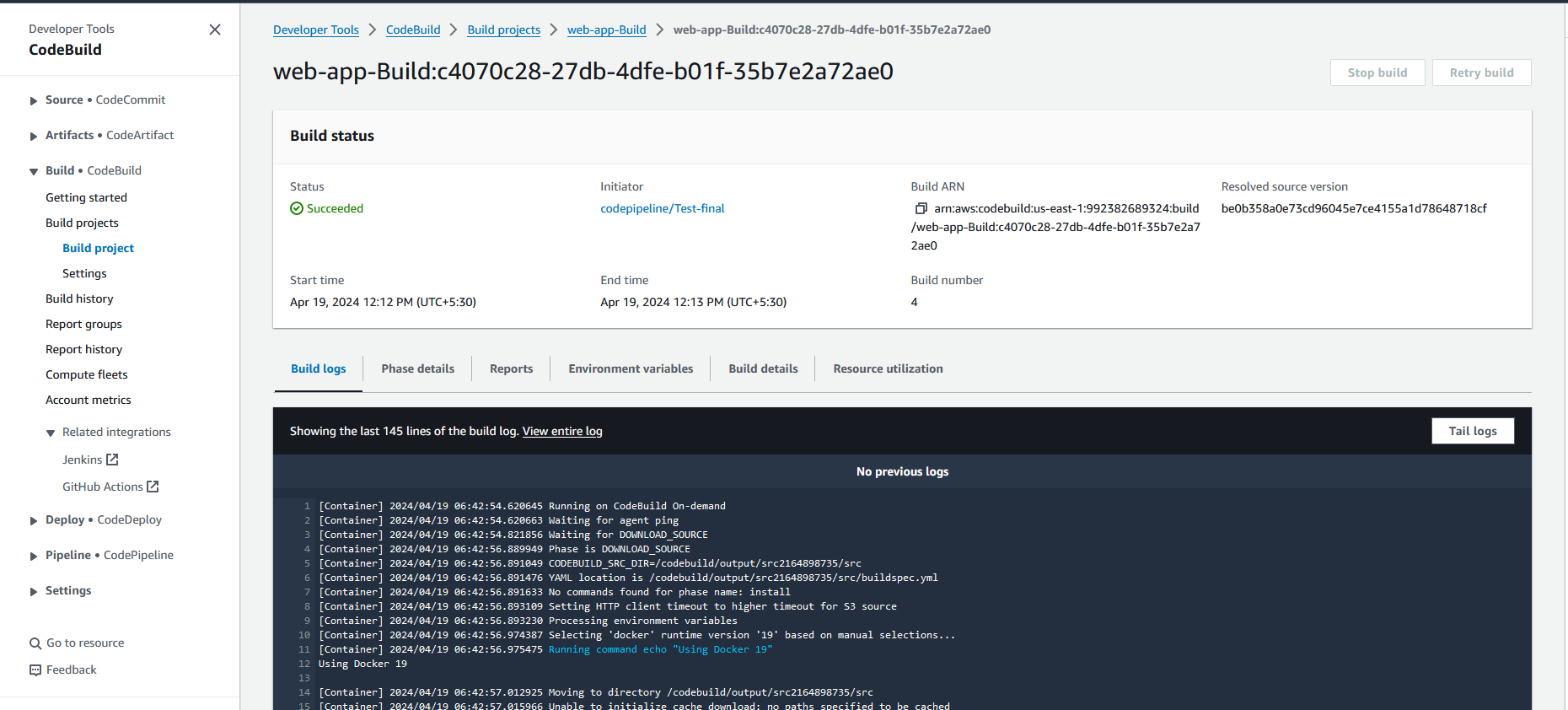
Under environment add env variables dockerhub username and docker hub password

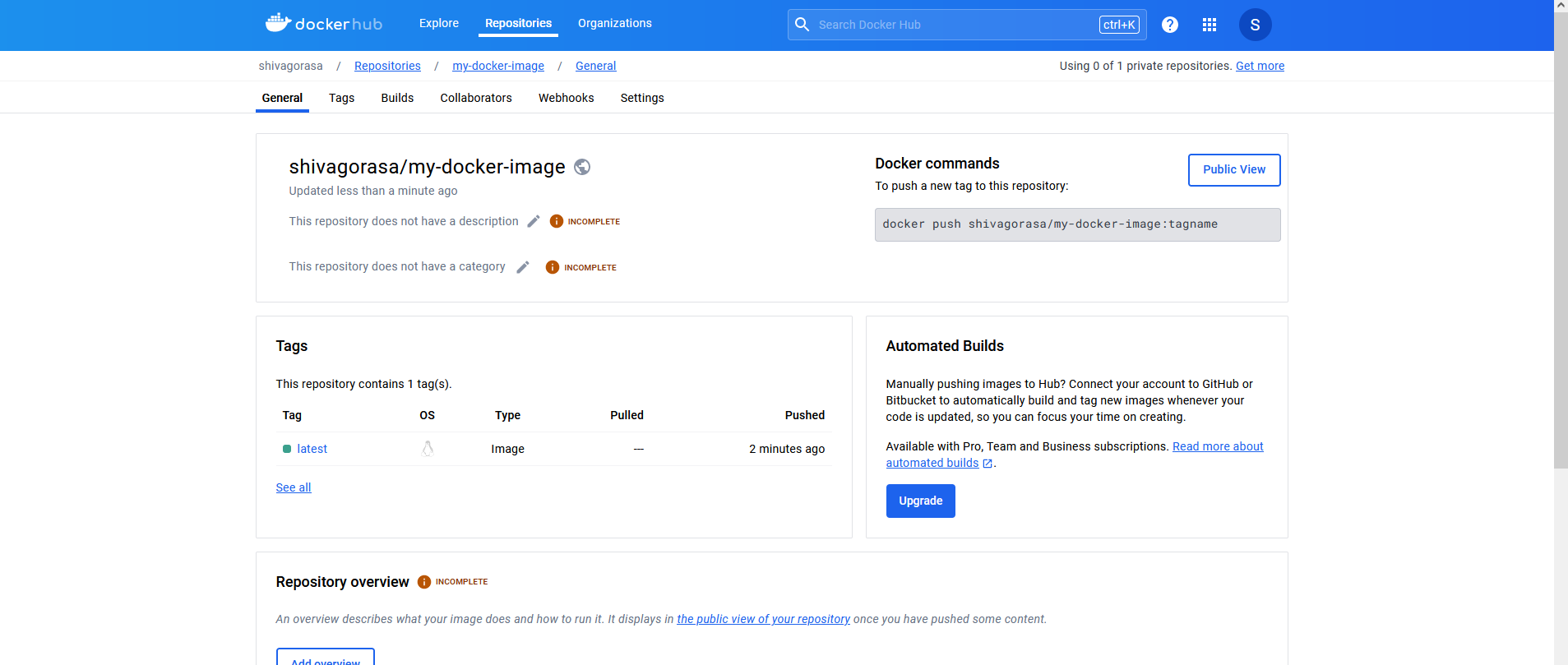
Under build spec use a buildspec.yml file which is present in our source code repo on code commit

Then create build project



If we click on start build , a docker image is pushed to our docker hub registry with username/my-docker-image:latest





Continuous Deployment (ElasticBeanstalk)

Create a ElasticBeanstalk docker based application to deploy our reactjs application.

Choose create application -> applicaton name

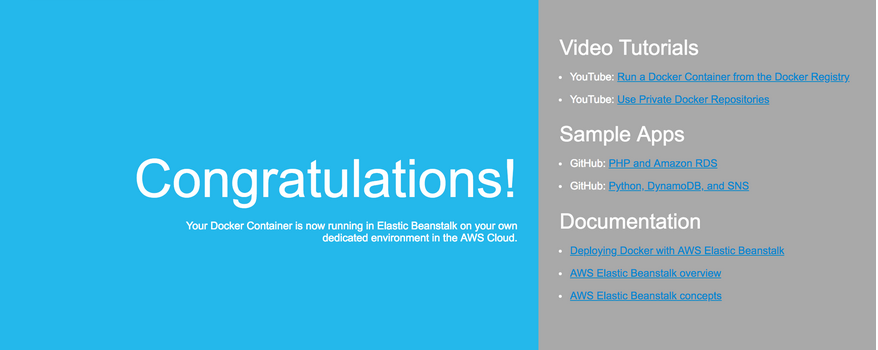
Then select application name and select application environment

Choose a sample docker application

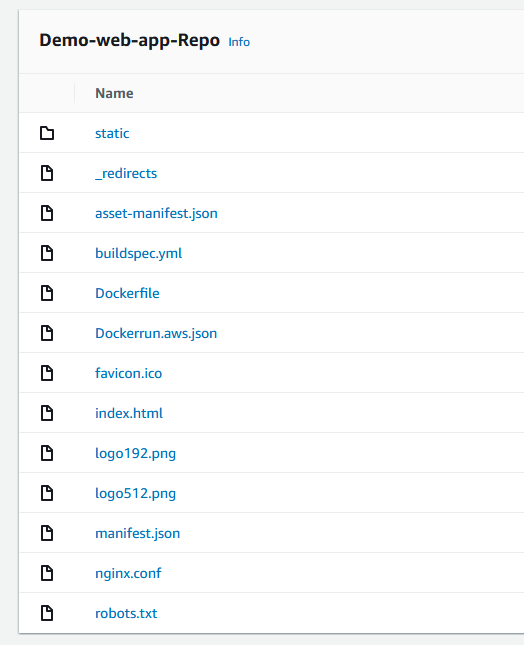
Choose above service role and choose a key to login to a instance , choose vpc, for instance and traffic scaling

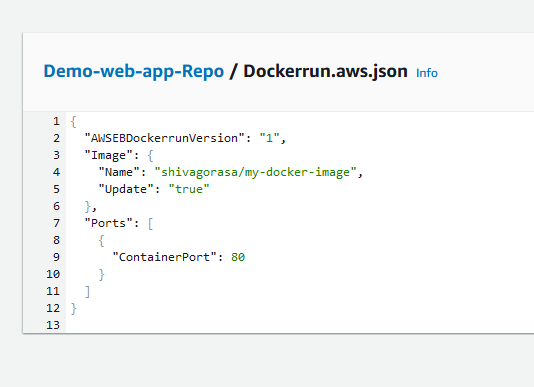
Select single instance , use default under monitoring and create application

We get default docker application running on elastic bean stalk

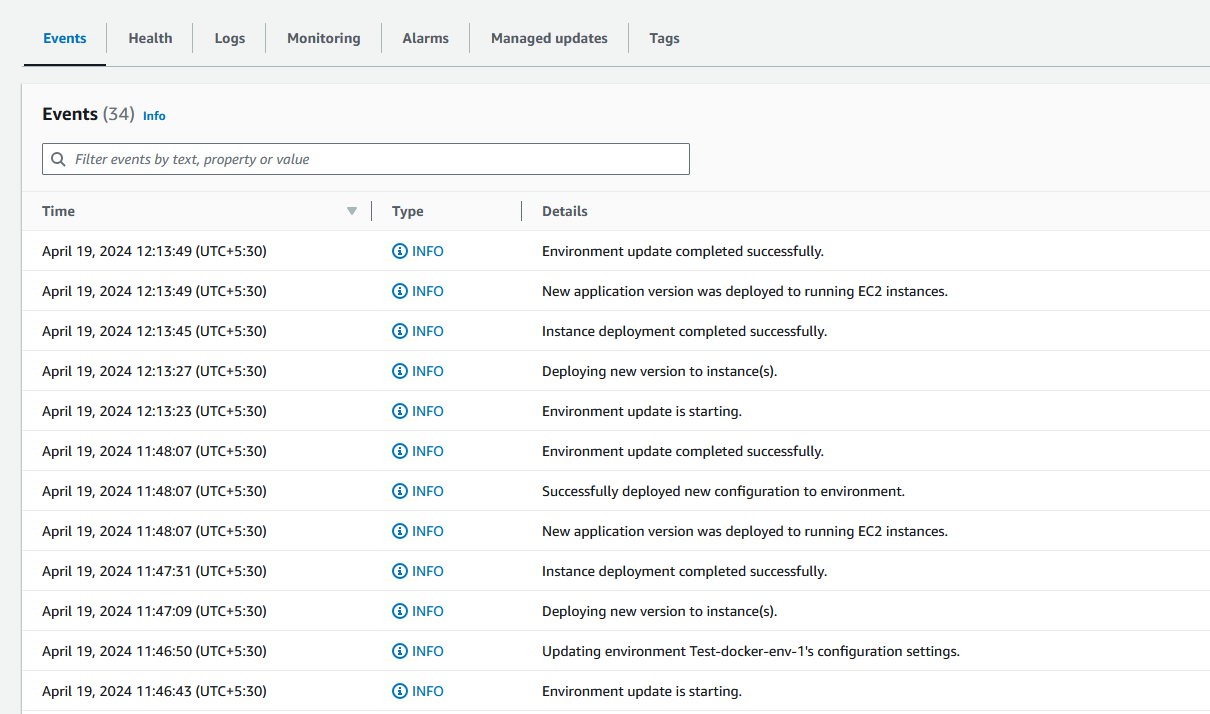


Later on elastic beanstalk upload source code as a zip file with buildspec.yml and Dockerrun.aws.json in the sourcecode directory

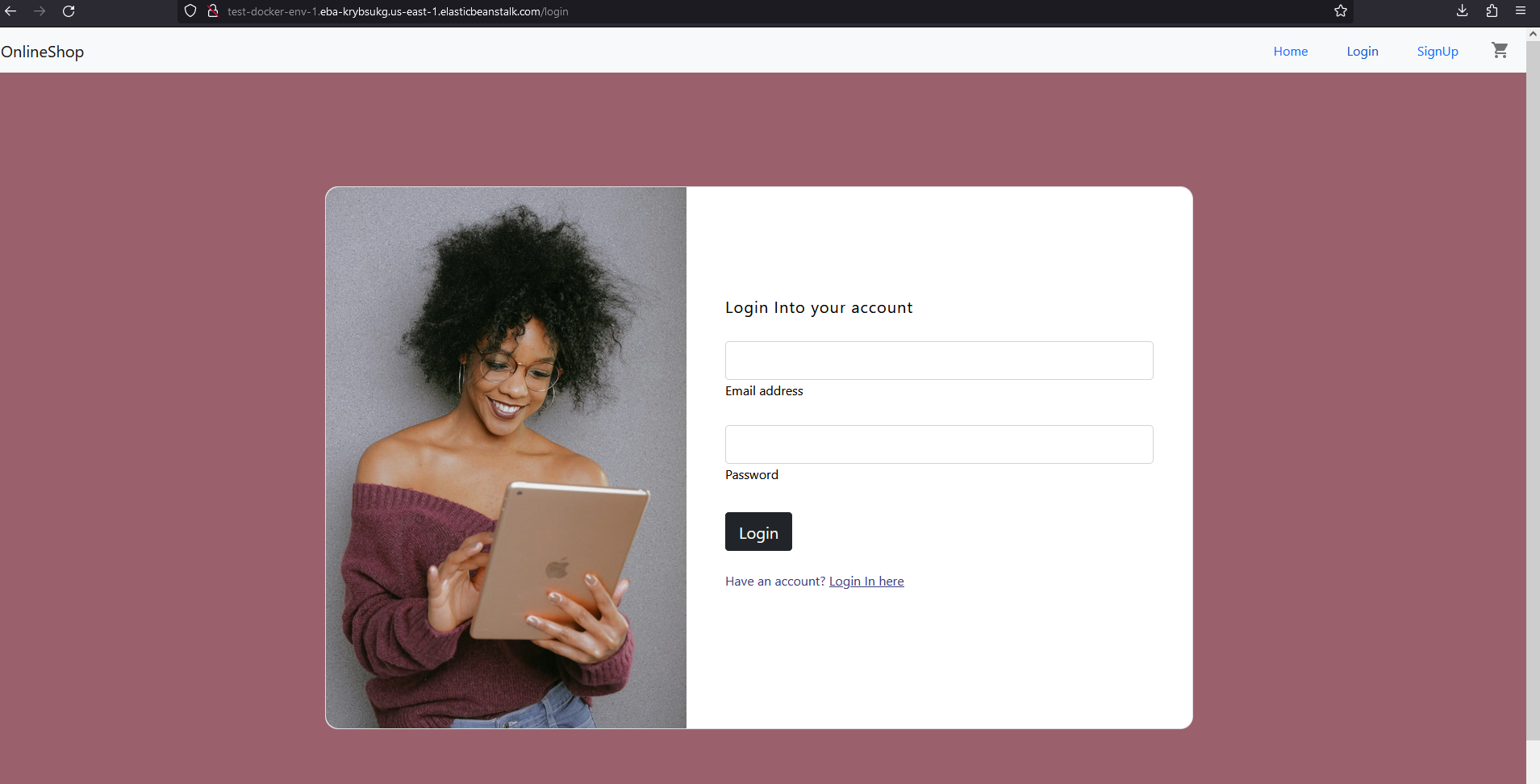




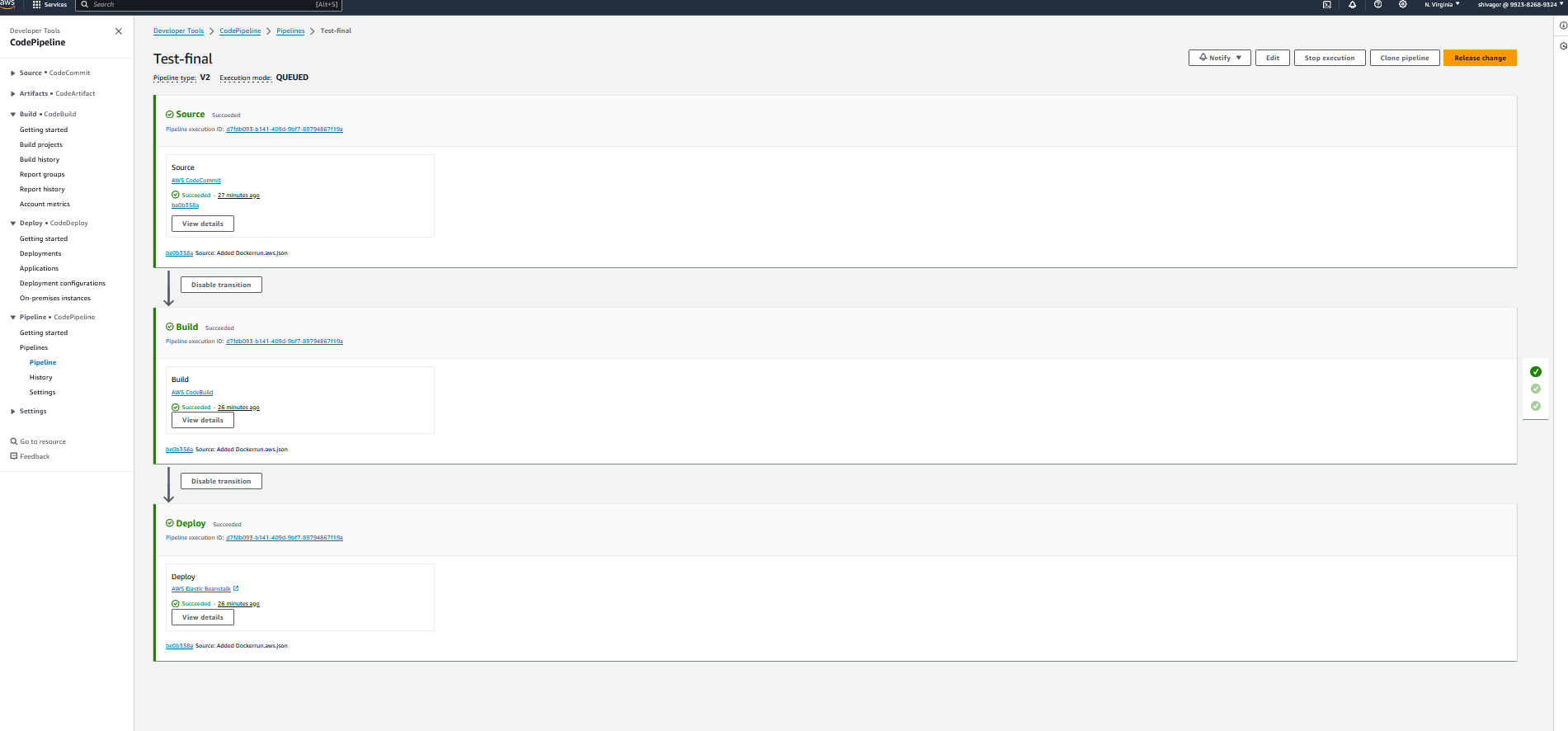
Usiing above create a EBS TO DEPLOY OUR APPLICATION , IF SUCCESSFUL WE GET FOLLOWING UNDER LOGS:



WE CAN ACCESS OUR APP RUNNING USING DOMAIN UNDER EBS



NOW TO CREATE A Code pipeline Select source as code commit, artifacts like source code is sent to aws s3 bucket, for build process select build as code build, and finally for code deploy use aws elastic beanstalk with required IAM role and policies , if everything works well we get following



And similarly we use domain on ebs to access our application running on aws

