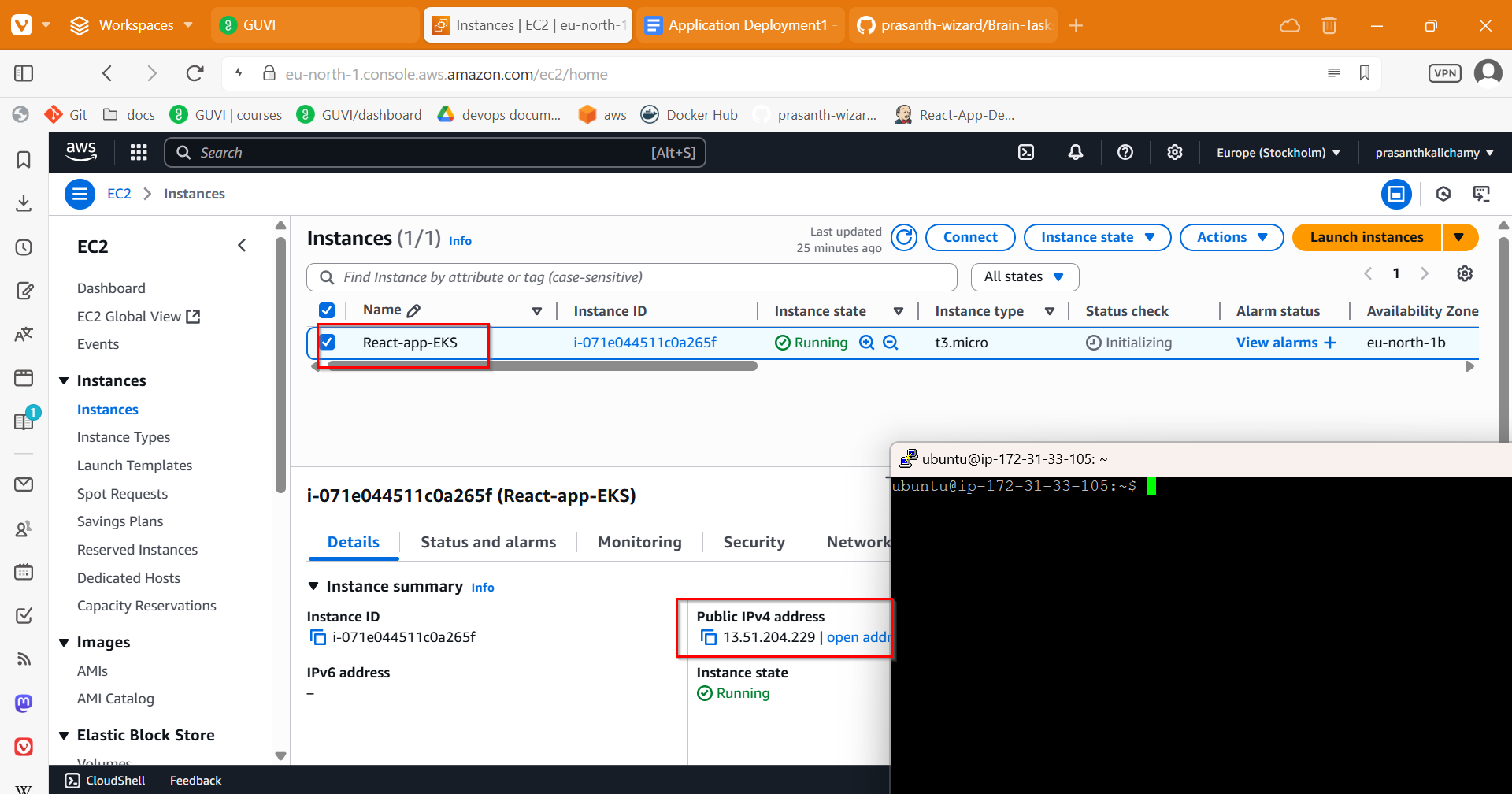
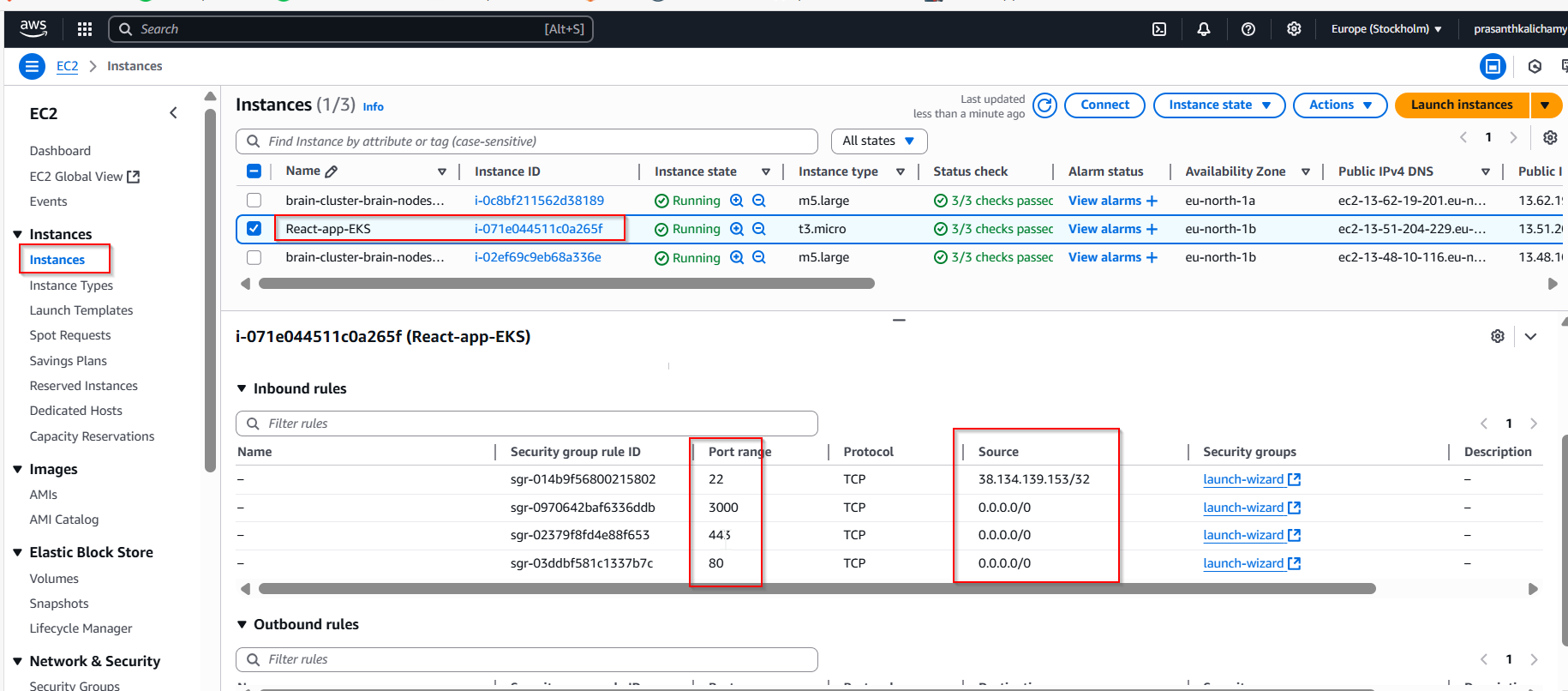
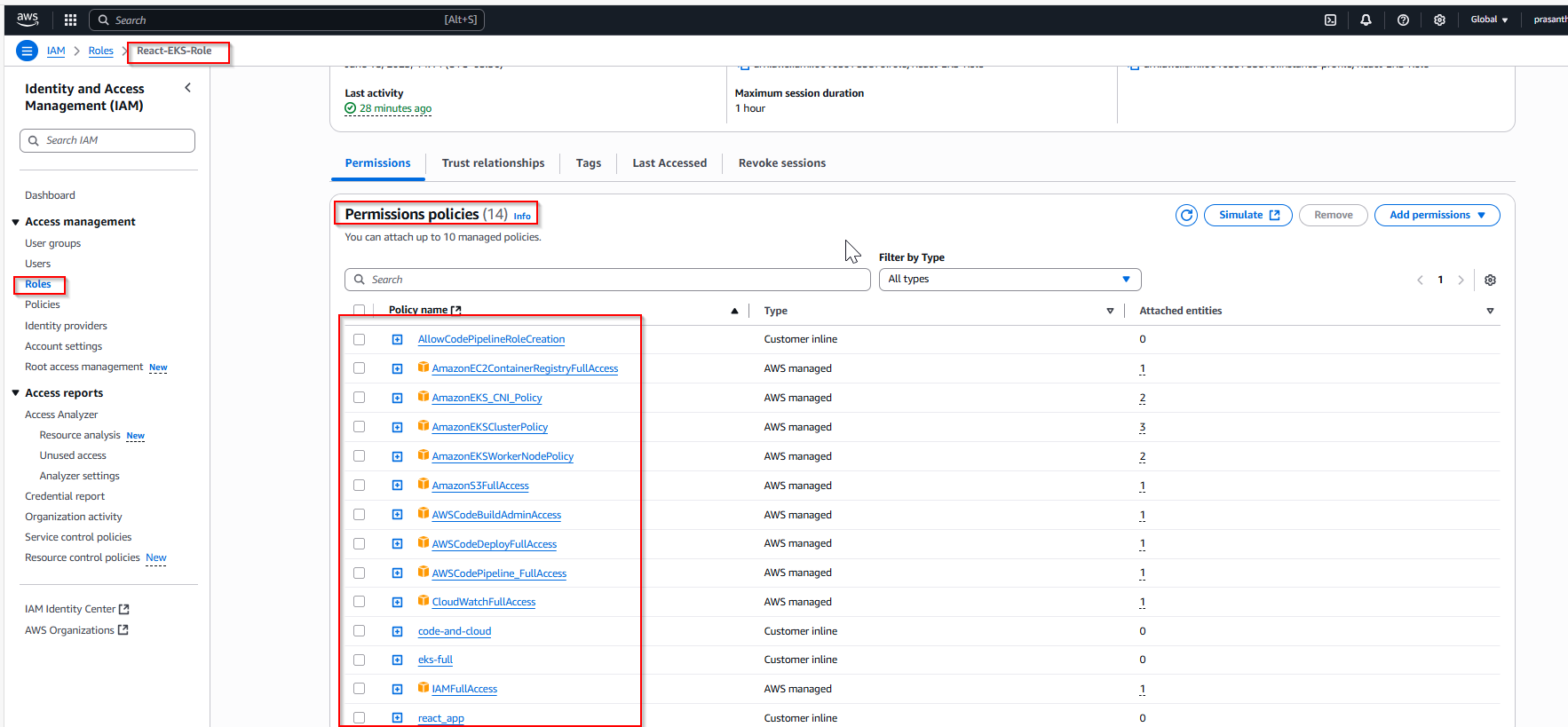
**Instance has been created and connected with putty:**



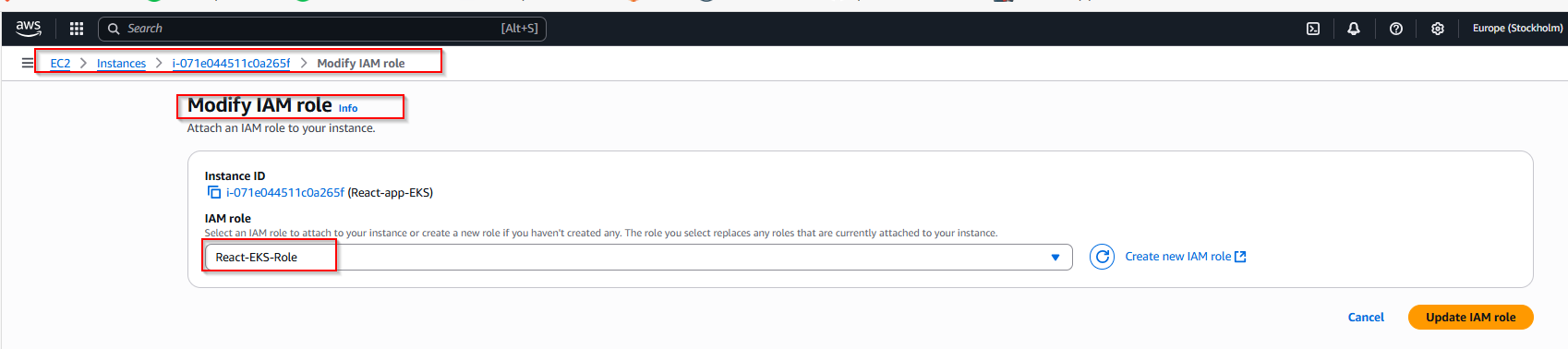
**Security Group:**

****

**IAM Role Setup on EC2:**

****

**IAM role Attached in instance:**



**Installing Required Tools:**

sudo apt update && sudo apt upgrade -y

**Docker Installing:**

sudo apt install docker.io -y

sudo systemctl enable docker

sudo systemctl start docker

sudo usermod -aG docker $USER

**Git installing:**

sudo apt install git -y

**Installing unzip and curl:**

sudo apt install unzip curl -y

**Installing AWS CLI:**

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

unzip awscliv2.zip

sudo ./aws/install

**Installing kubectl:**

curl -LO "https://dl.k8s.io/release/$(curl -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

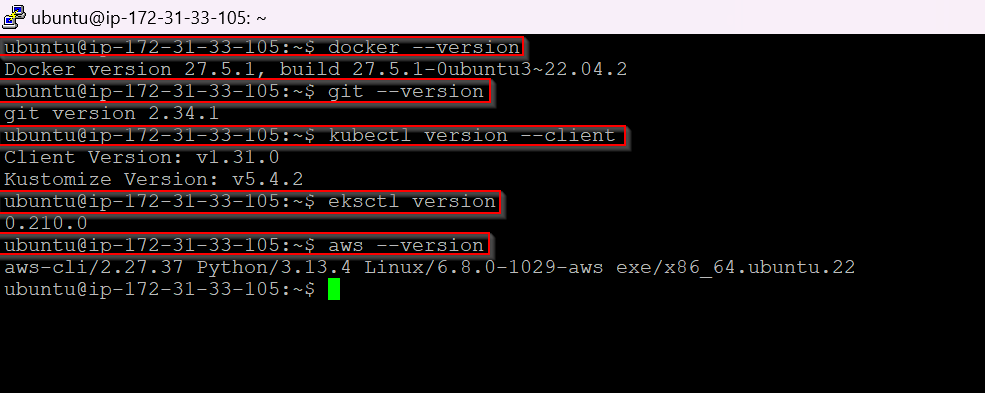
**Installing eksctl:**

KUBECTL\_VERSION=$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)

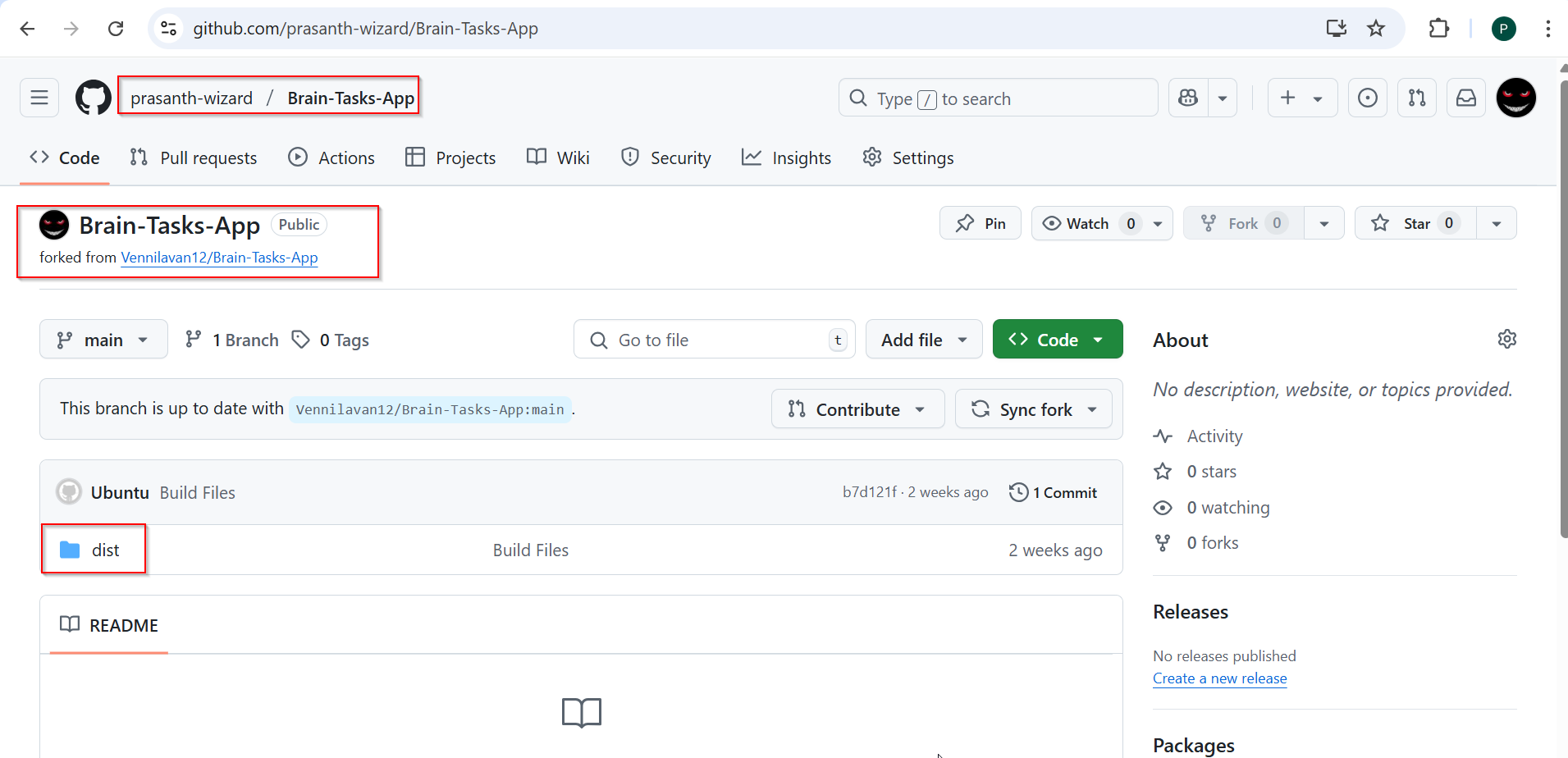
curl -LO [https://dl.k8s.io/release/${KUBECTL\_VERSION}/bin/linux/amd64/kubectl](https://dl.k8s.io/release/$%7bKUBECTL_VERSION%7d/bin/linux/amd64/kubectl)

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

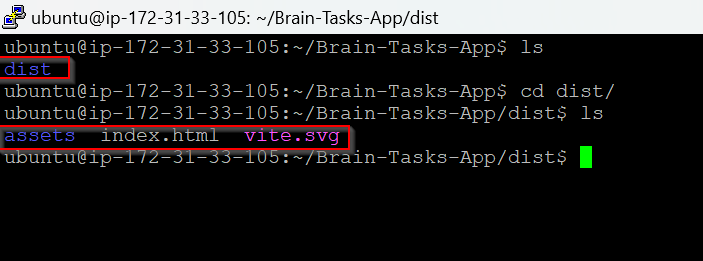


**Forked GitHub Repo:**

****

git clone https://github.com/prasanth-wizard/Brain-Tasks-App.git

cd Brain-Tasks-App



**Create Dockerfile:**

**Dockerfile**

FROM nginx:alpine

RUN rm -rf /usr/share/nginx/html/\*

COPY nginx.conf /etc/nginx/conf.d/default.conf

COPY dist/ /usr/share/nginx/html/

EXPOSE 3000

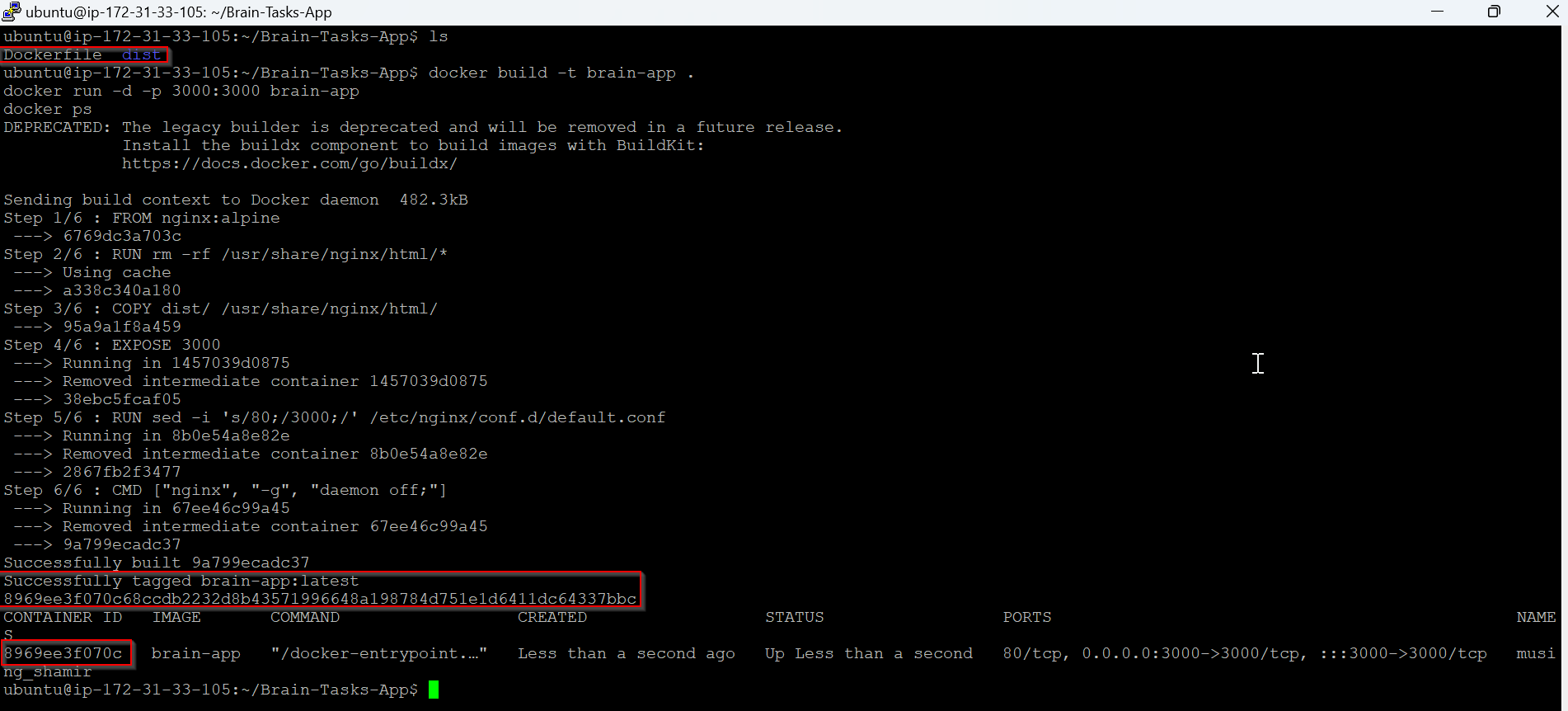
CMD ["sh", "-c", "nginx -g 'daemon off;' 2>&1 | tee -a /app-logs/app.log"]

**Build & Test the Docker Image Locally:**

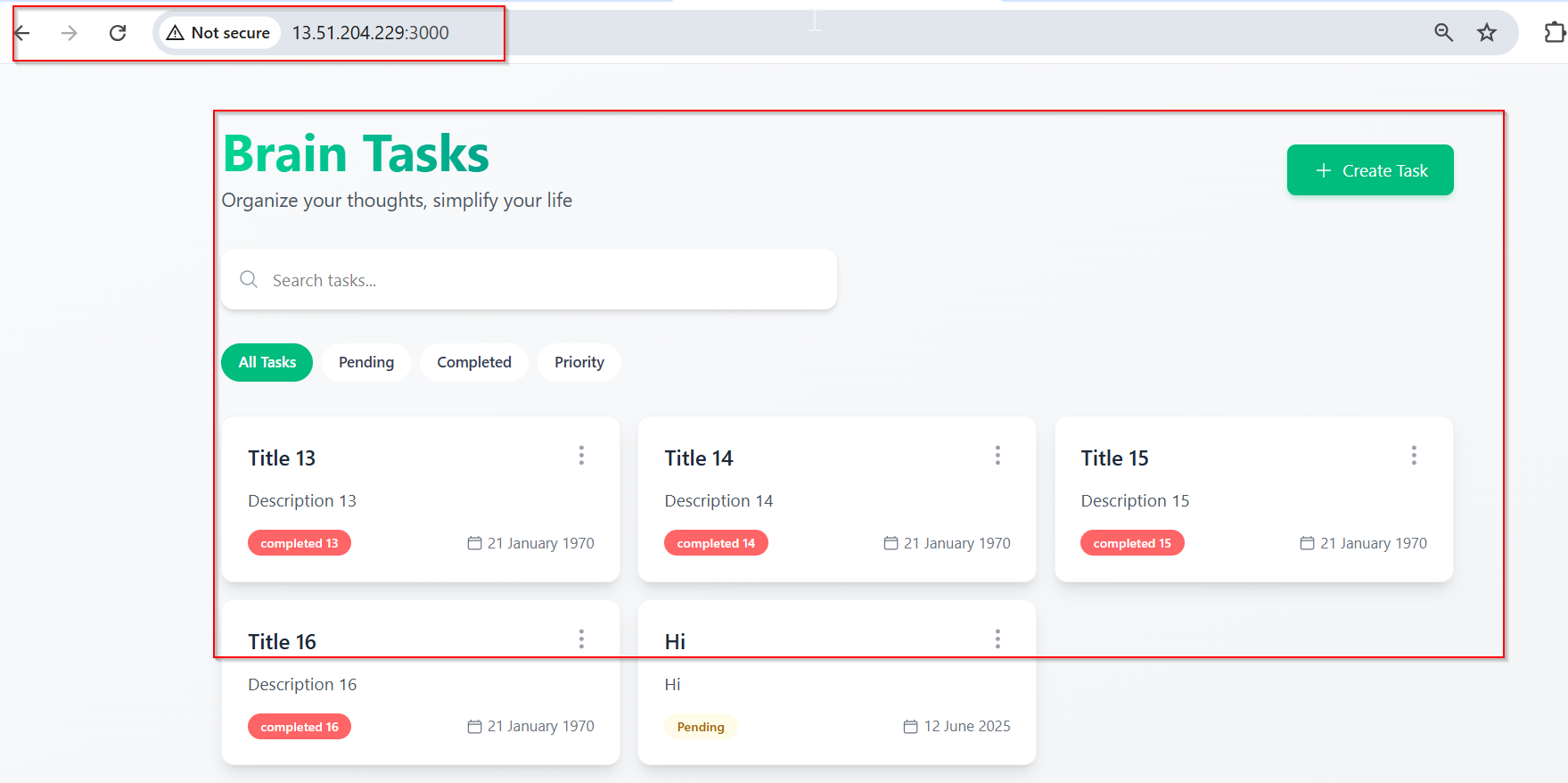
docker build -t brain-app .

docker run -d -p 3000:3000 brain-app

docker ps



**Container is running and mapped to port 3000:**

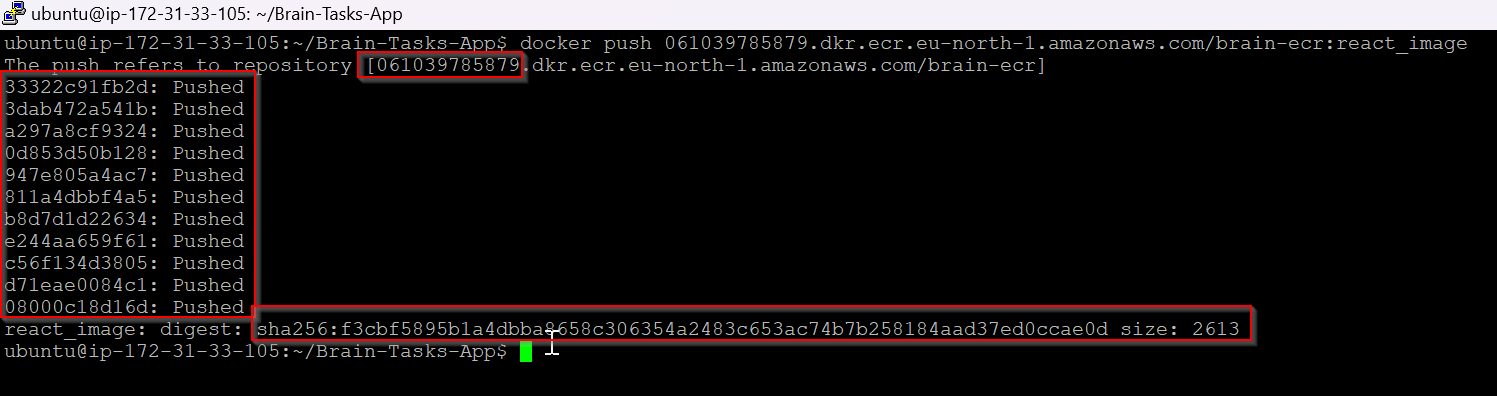


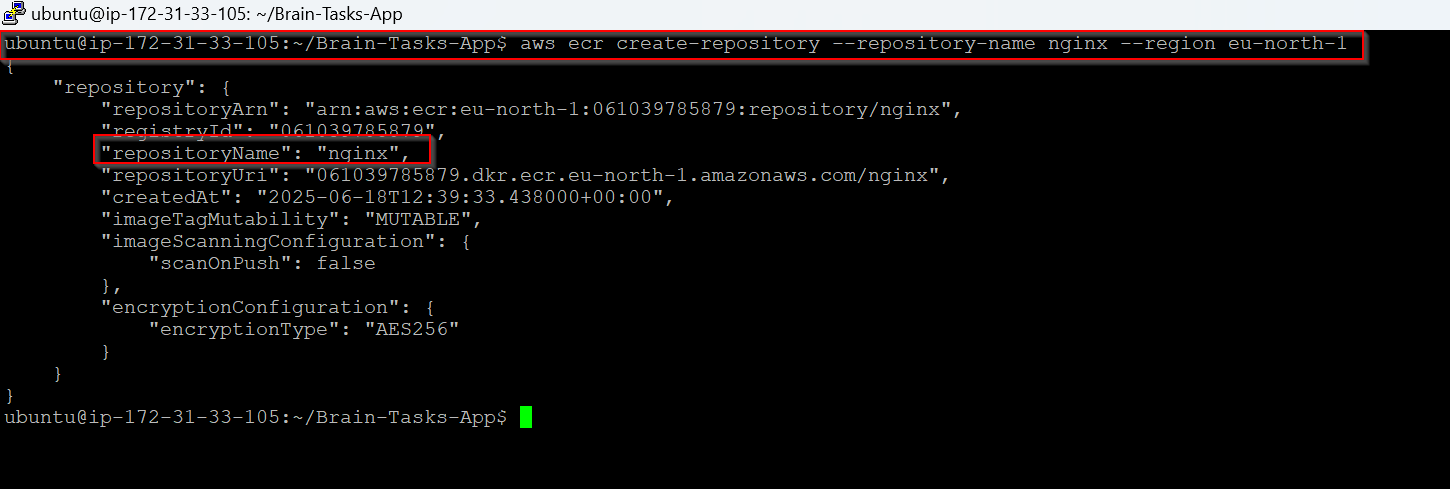
**Push Docker Image to ECR:**

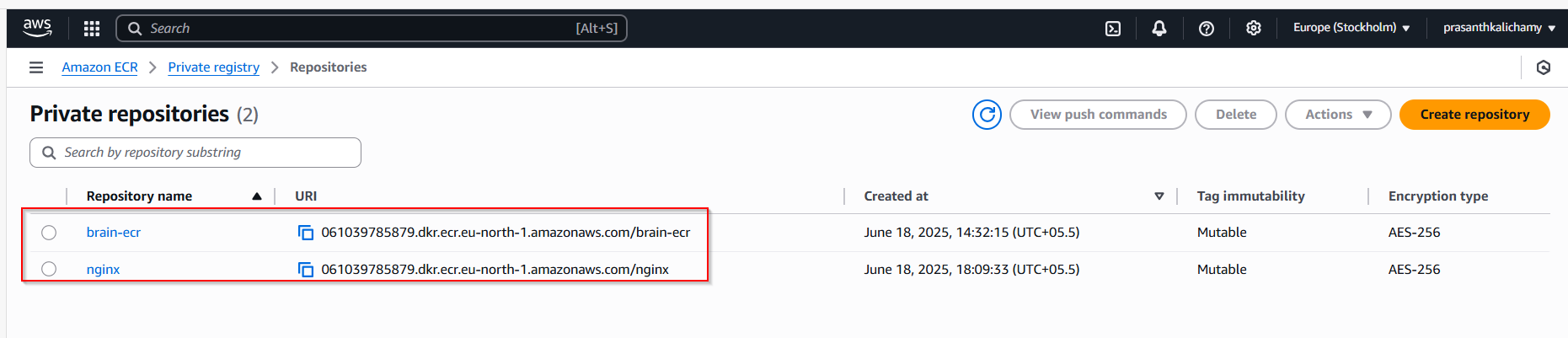
aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 061039785879.dkr.ecr.eu-north-1.amazonaws.com

docker tag brain-app:latest 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

docker push 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image







**Creating the EKS cluster:**

eksctl create cluster \

--name brain-cluster \

--region eu-north-1 \

--nodegroup-name brain-nodes \

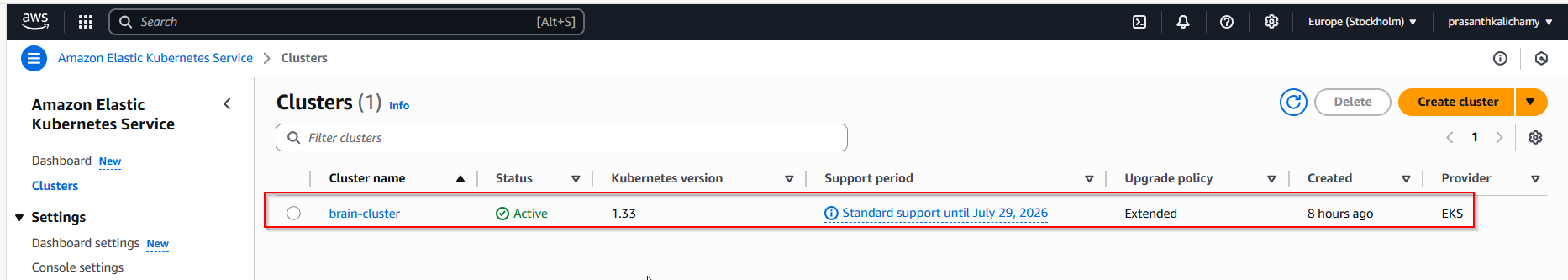
--nodes 2 \

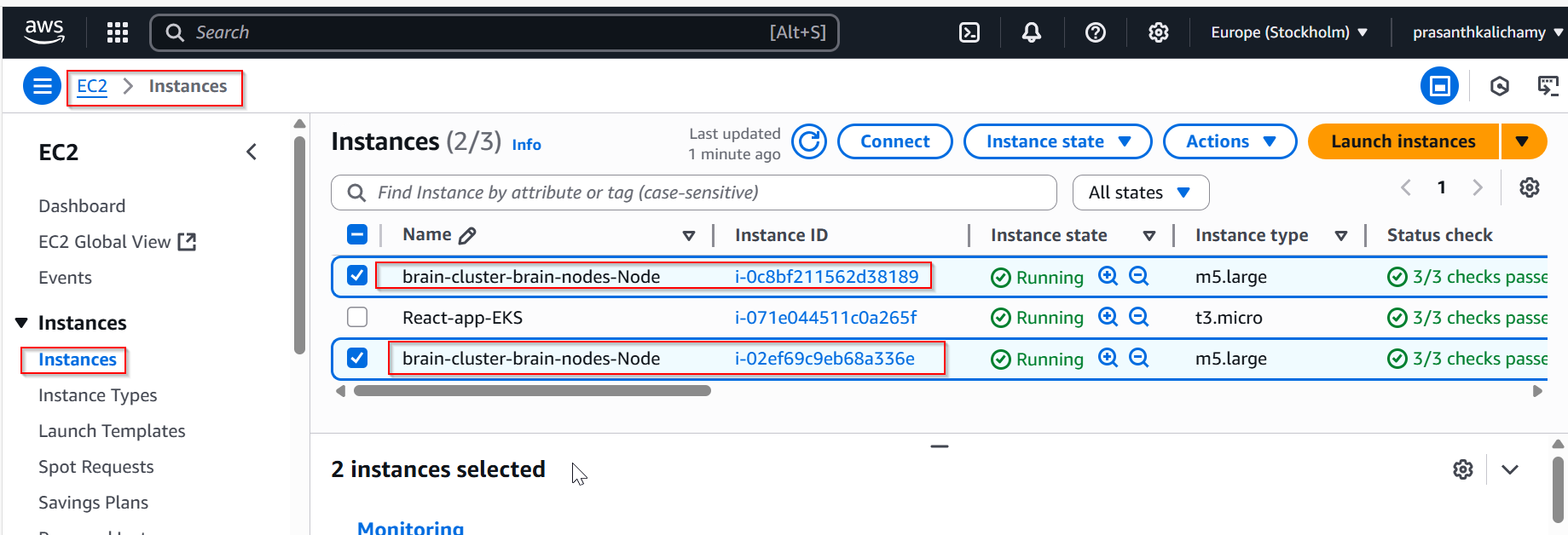
--nodes-min 1 \

--nodes-max 3 \

--managed

**EKS Cluster:**





**Deploy to EKS:**

**Creating deployment.yaml file:**

apiVersion: apps/v1

kind: Deployment

metadata:

name: brain-app-deployment

spec:

replicas: 1

selector:

matchLabels:

app: brain-app

template:

metadata:

labels:

app: brain-app

spec:

containers:

- name: brain-app

image: 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

ports:

- containerPort: 3000

volumeMounts:

- name: applogs

mountPath: /app-logs

- name: cloudwatch-agent

image: amazon/cloudwatch-agent:latest

volumeMounts:

- name: applogs

mountPath: /app-logs

- name: cwagentconfig

mountPath: /etc/cwagentconfig

readOnly: true

args:

- -config

- /etc/cwagentconfig/cwagentconfig.json

- -envconfig

volumes:

- name: applogs

emptyDir: {}

- name: cwagentconfig

configMap:

name: cwagentconfig

**Creating service.yaml file:**

apiVersion: v1

kind: Service

metadata:

name: brain-app-service

spec:

type: LoadBalancer

selector:

app: brain-app

ports:

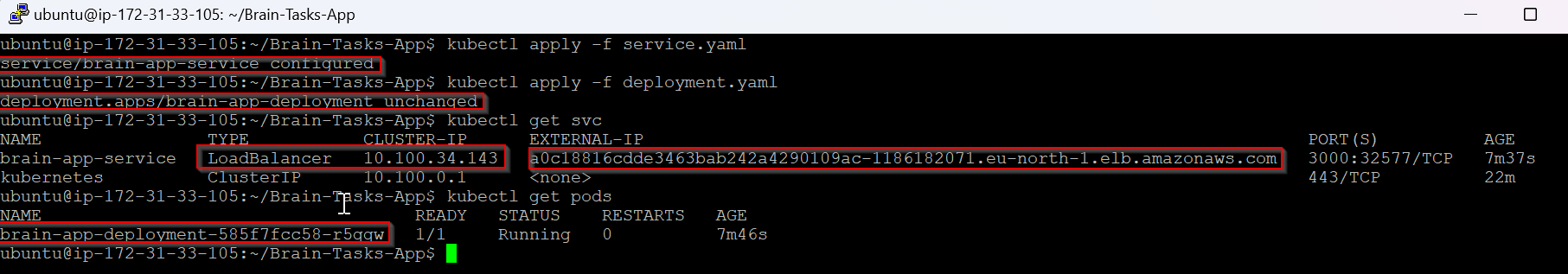
- protocol: TCP

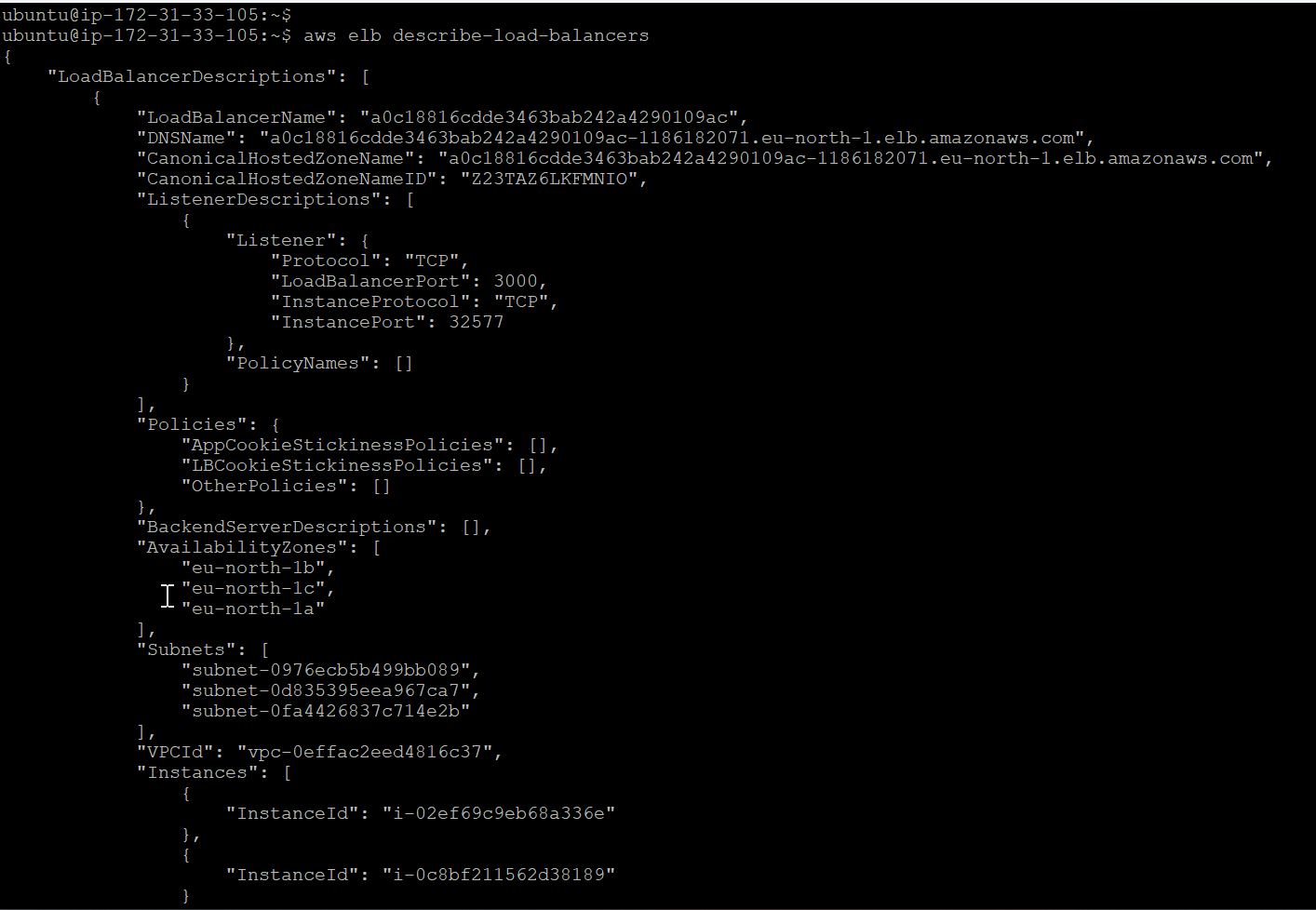
port: 3000

targetPort: 3000

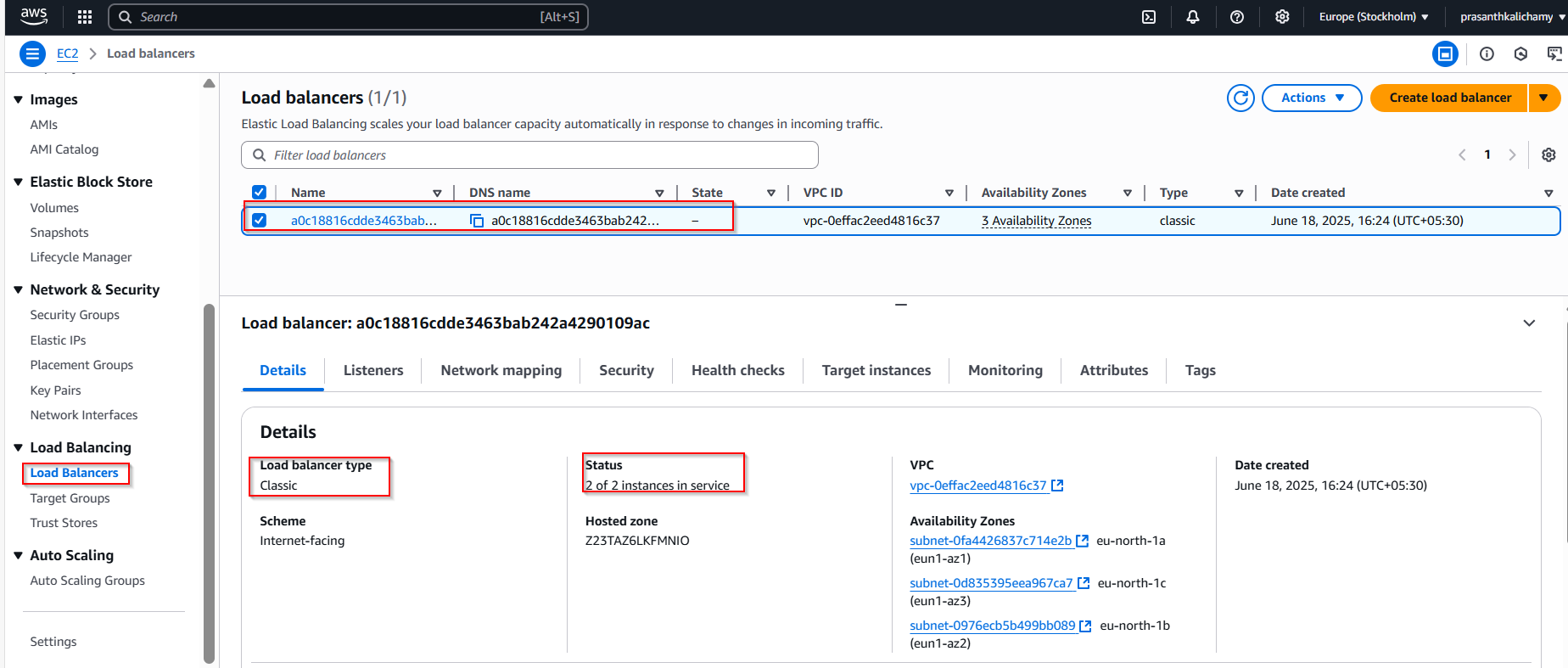
kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

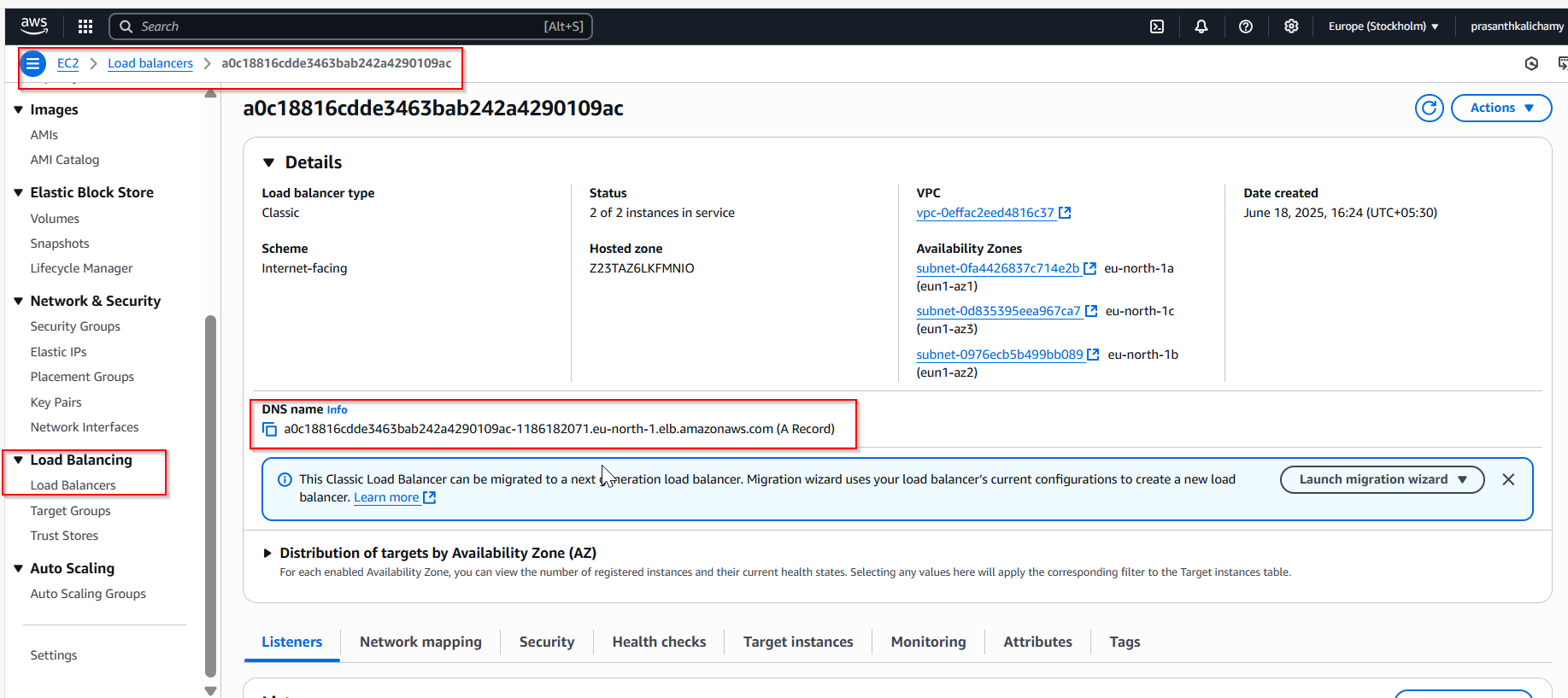




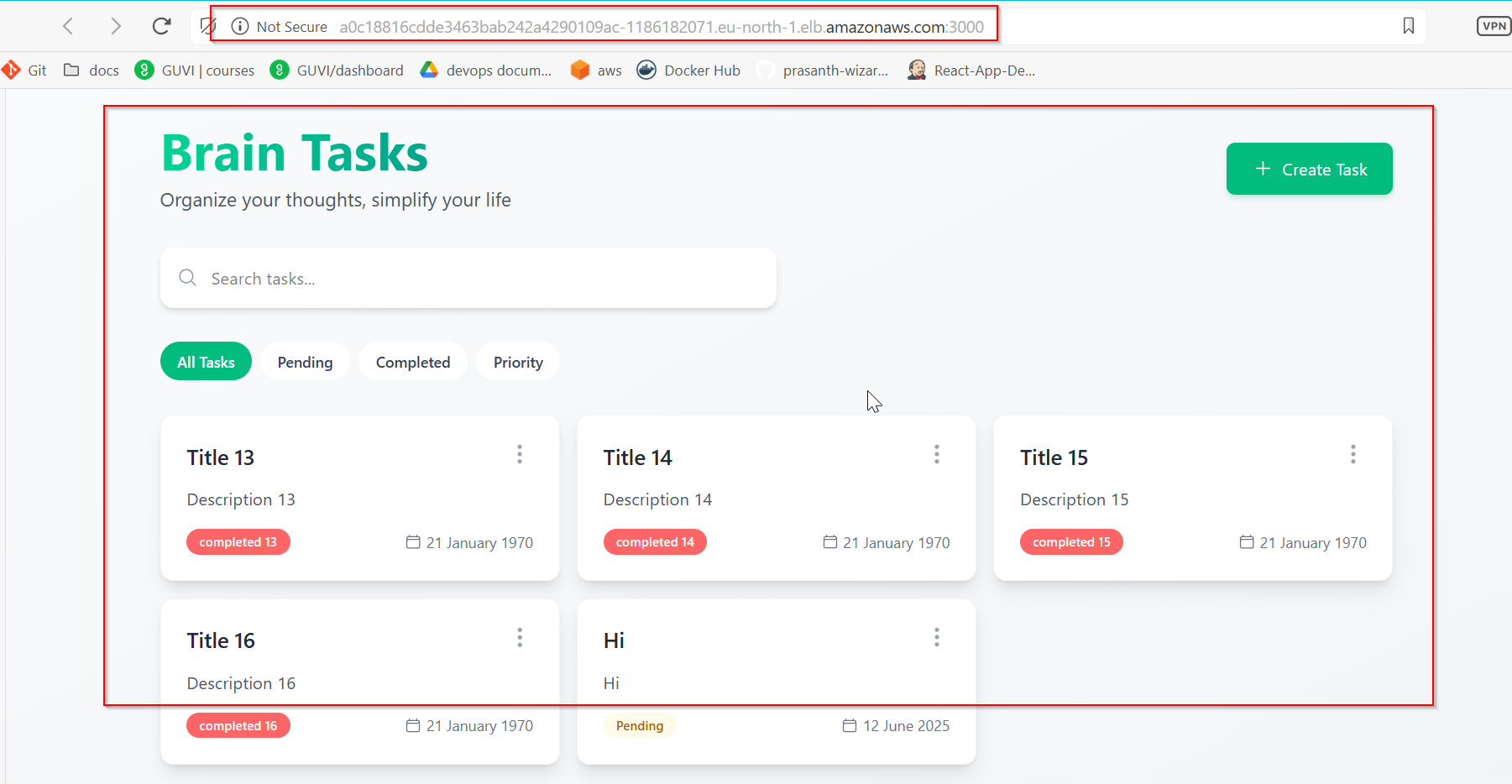
**Load Balancer:**

****

**Load balancer ARN:**

****

**Application running in load balancer ARN:**



**AWS CodeBuild:**

**Configuration for CodeBuild Project:**

**Project Configuration:**

Project name: brain-react-build

Leave description blank or write: Builds React app and pushes Docker image to ECR

**Source:**

Source provider: GitHub

**Repository:**

https://github.com/prasanth-wizard/Brain-Tasks-App

Webhook: Enable webhook to trigger build on code push

**Environment:**

Environment image: Managed image

Operating system: Amazon Linux

Runtime: Standard

Image: aws/codebuild/amazonlinux-x86\_64-standard:5.0

Image version: Always use the latest

Privileged mode: Enable this (to allow Docker commands)

**Service role:**

Choose New service role

Name it: codebuild-brain-react-build-service-role

**Artifacts:**

Type: No artifacts

(because you are pushing the Docker image to ECR)

**Logs:**

Enable CloudWatch logs

(You can leave the group/stream name blank unless you want custom names)

Click "Create build project"

**Creating buildspec.yml file:**

version: 0.2

phases:

pre\_build:

commands:

- echo Logging in to Amazon ECR...

- aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 061039785879.dkr.ecr.eu-north-1.amazonaws.com

- echo Logging in to Docker Hub...

- echo "$DOCKERHUB\_PASS" | docker login -u "prasanth0003" --password-stdin

build:

commands:

- echo Build started on `date`

- docker build -t brain-app .

- docker tag brain-app:latest 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

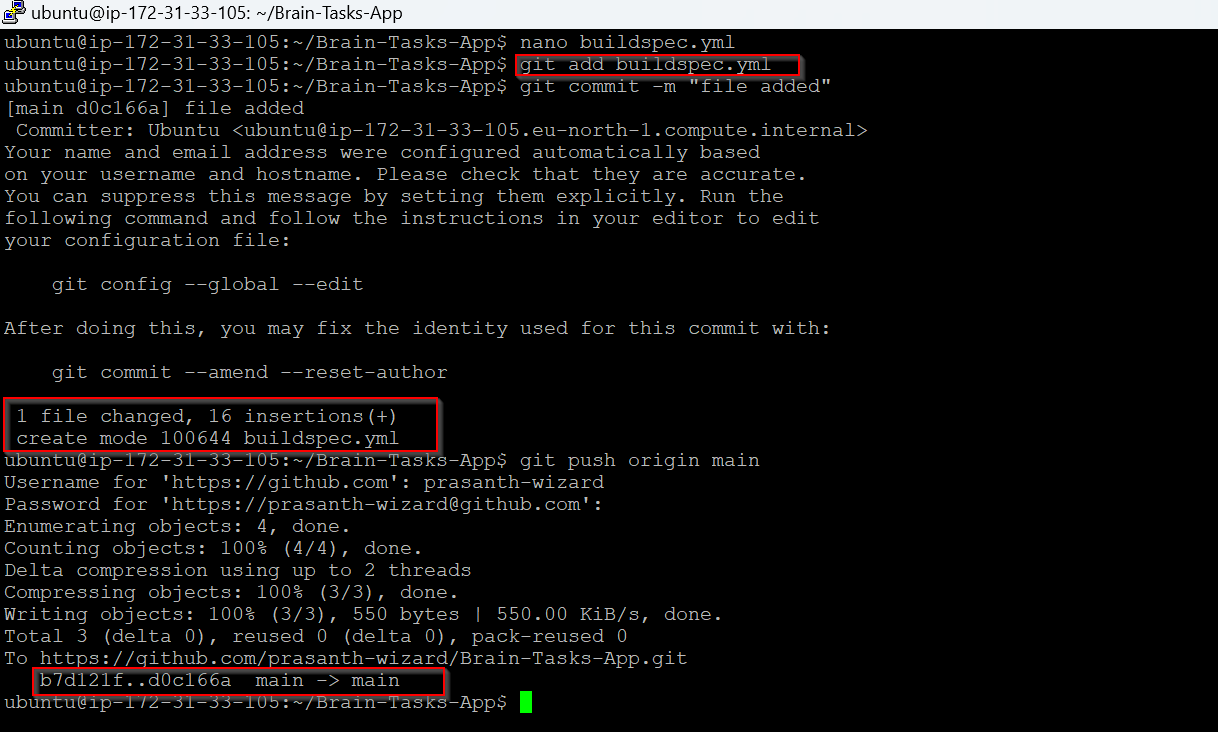
post\_build:

commands:

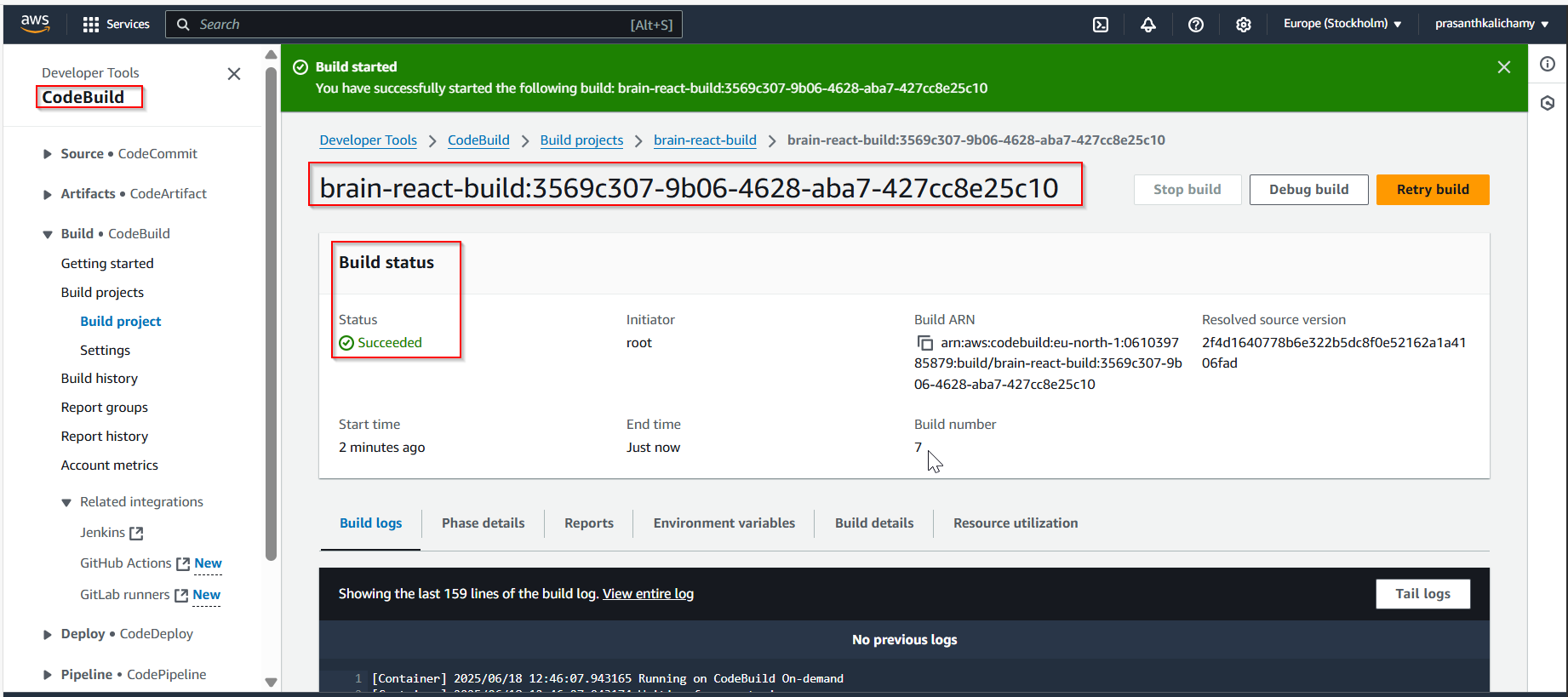
- echo Build completed on `date`

- docker push 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

**File created and pushed to github:**



**Build created successfully:**



**AWS CodeDeploy for EKS Deployment:**

**Create appspec.yaml**

version: 0.0

Resources:

- TargetService:

Type: AWS::EKS::Service

Properties:

TaskDefinition: ""

LoadBalancerInfo:

ContainerName: brain-app

ContainerPort: 3000

hooks:

AfterAllowTestTraffic:

- location: scripts/deploy.sh

timeout: 300

runas: root

**Create the deployment script:**

mkdir -p scripts

nano scripts/deploy.sh

#!/bin/bash

echo "Deploying to EKS..."

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

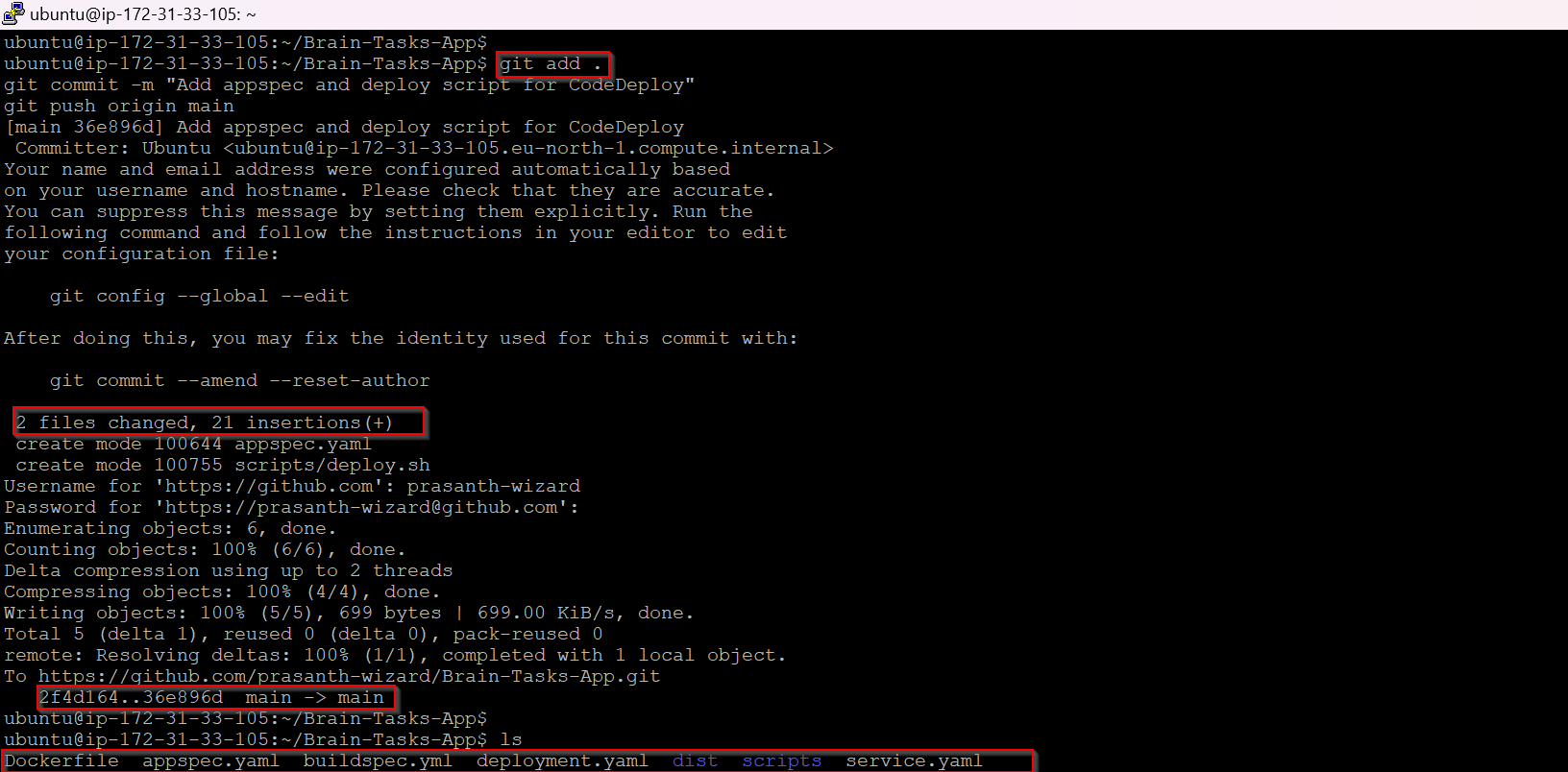
echo "Deployment complete!"

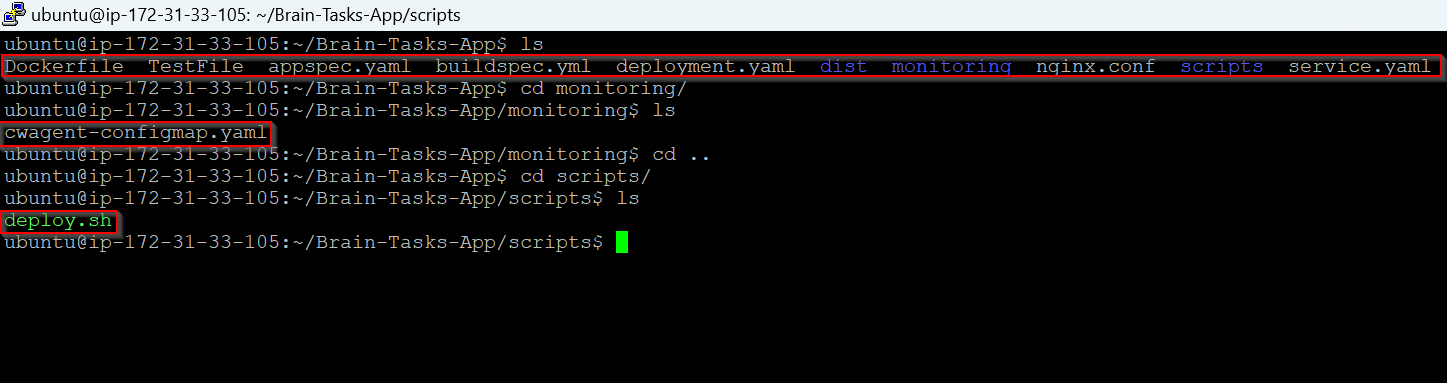
chmod +x scripts/deploy.sh

git add .

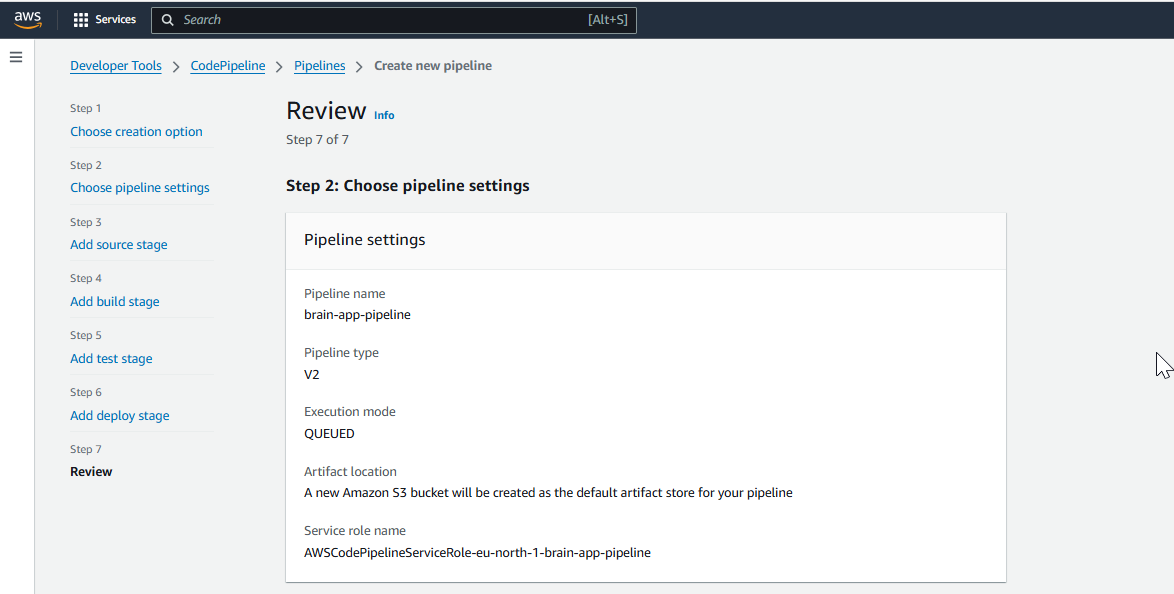
git commit -m "Add appspec and deploy script for CodeDeploy"

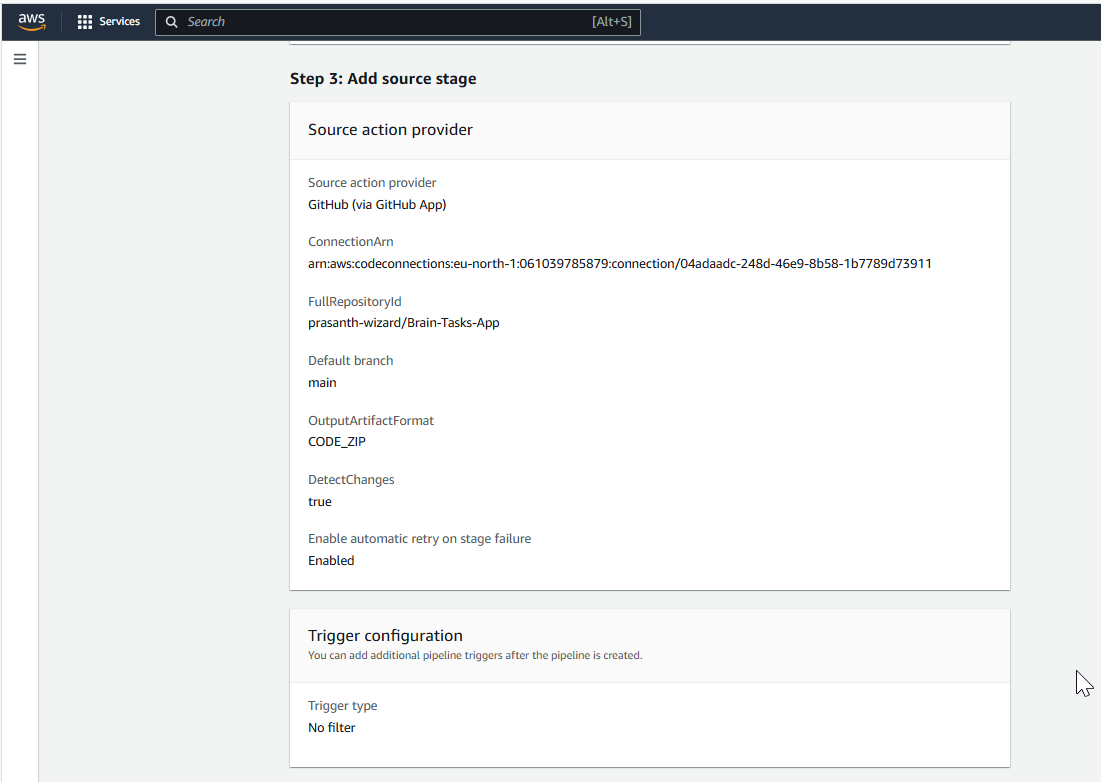
git push origin main

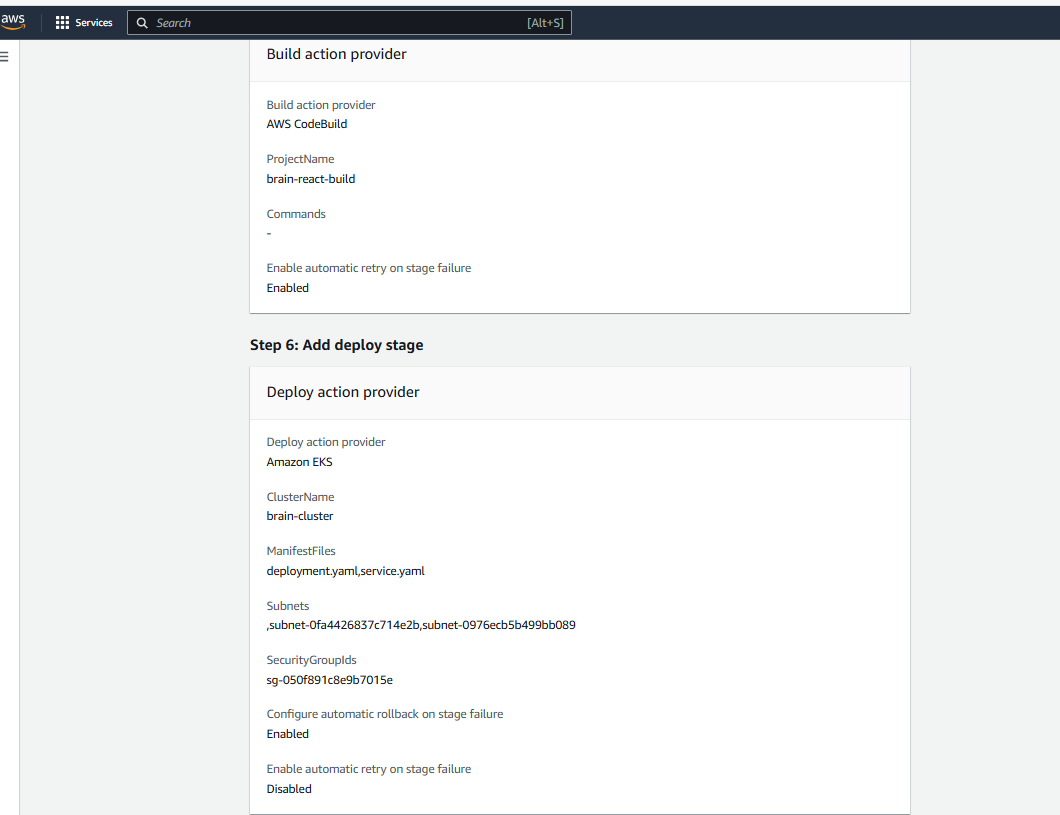


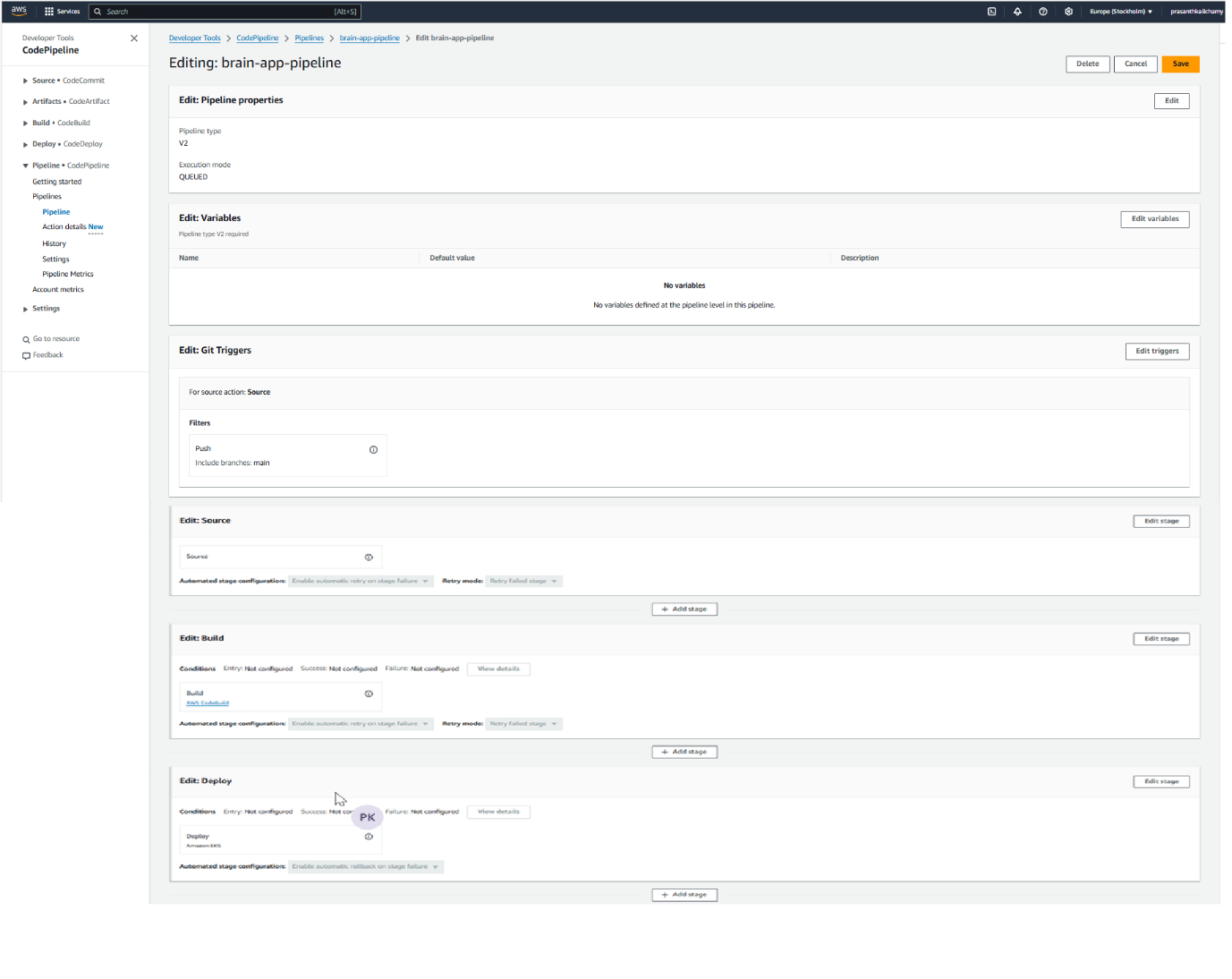


**Code Deploy and Pipeline settings:**

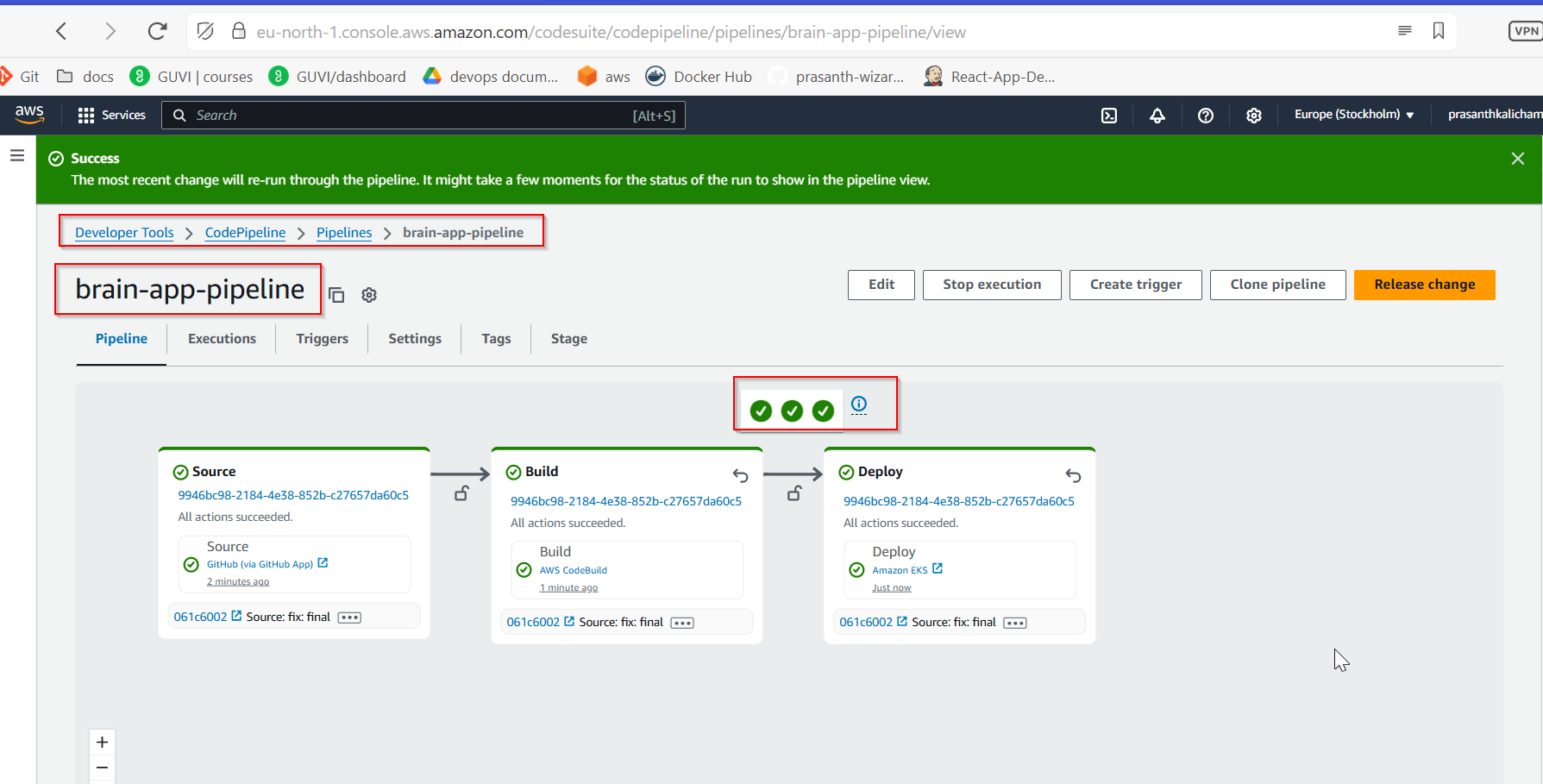


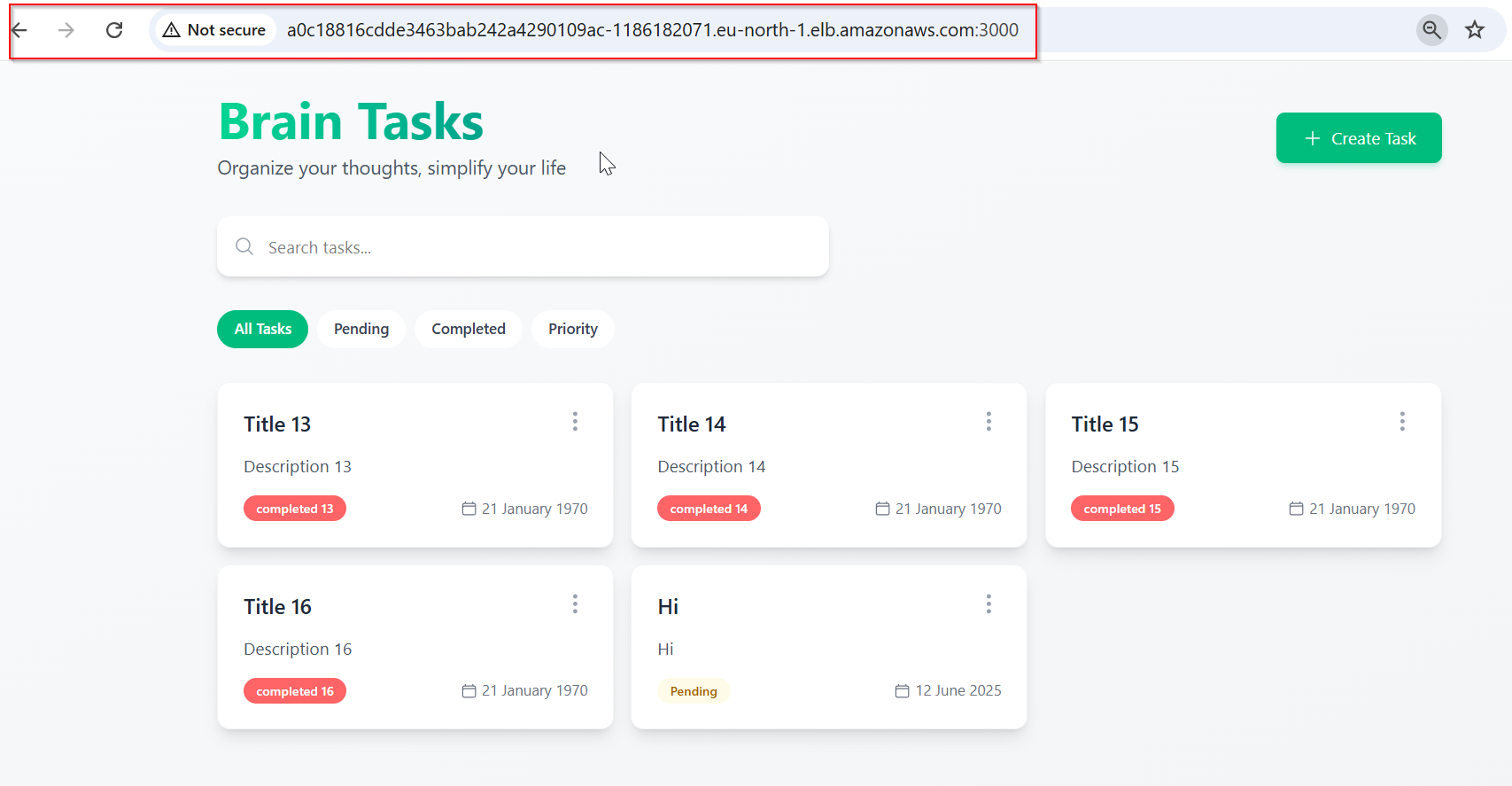




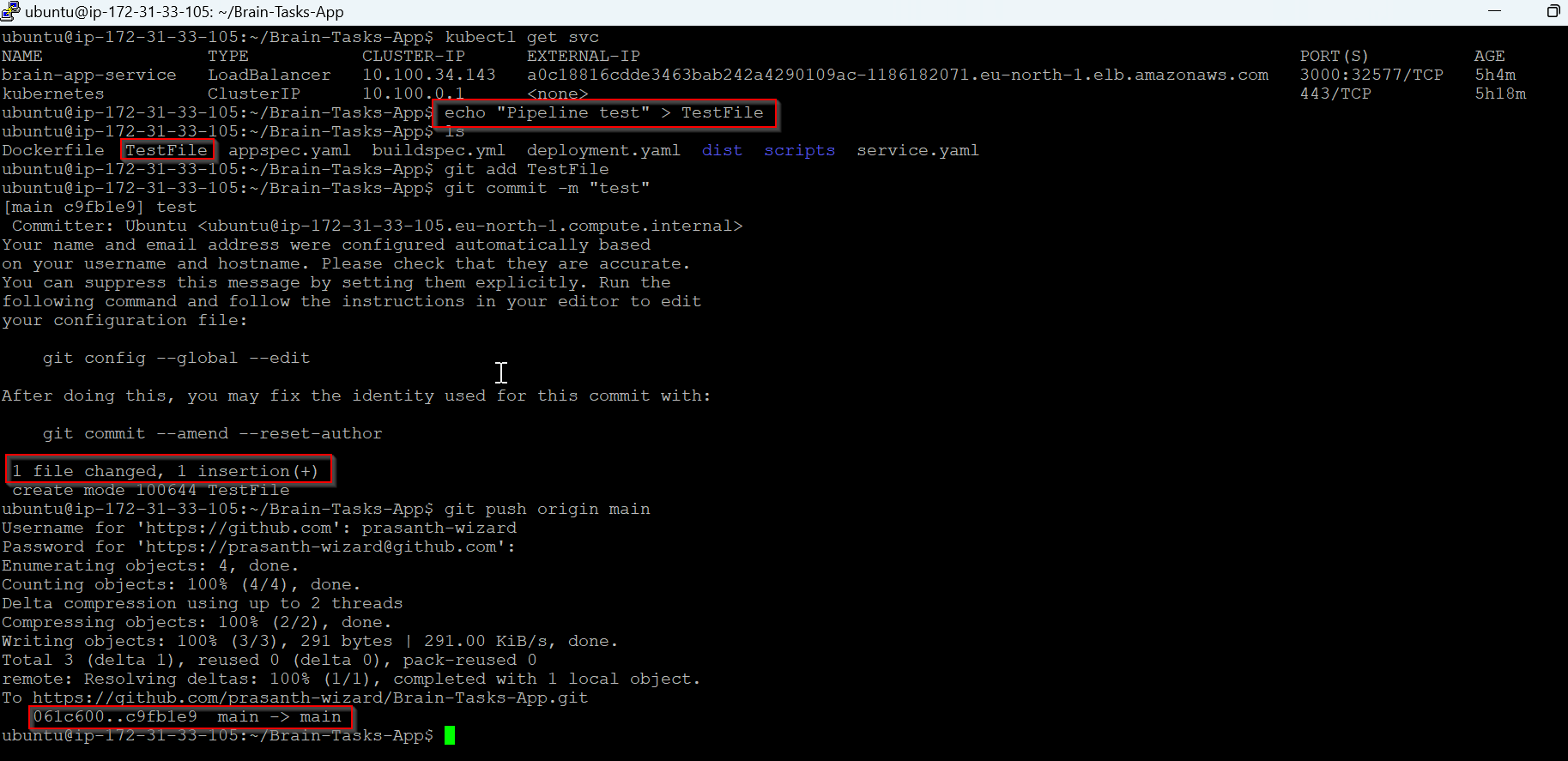


**Automation pipeline Success:**

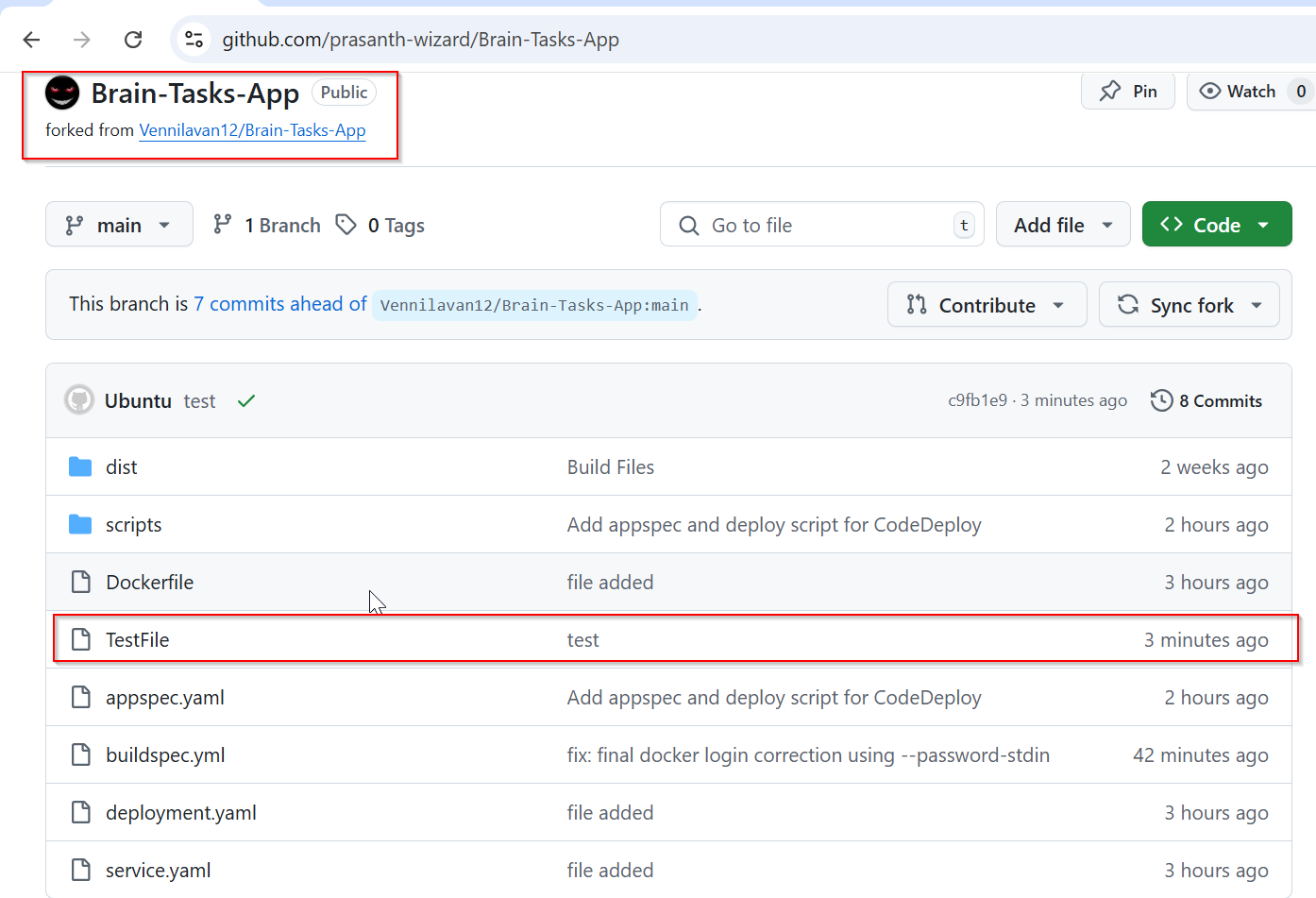




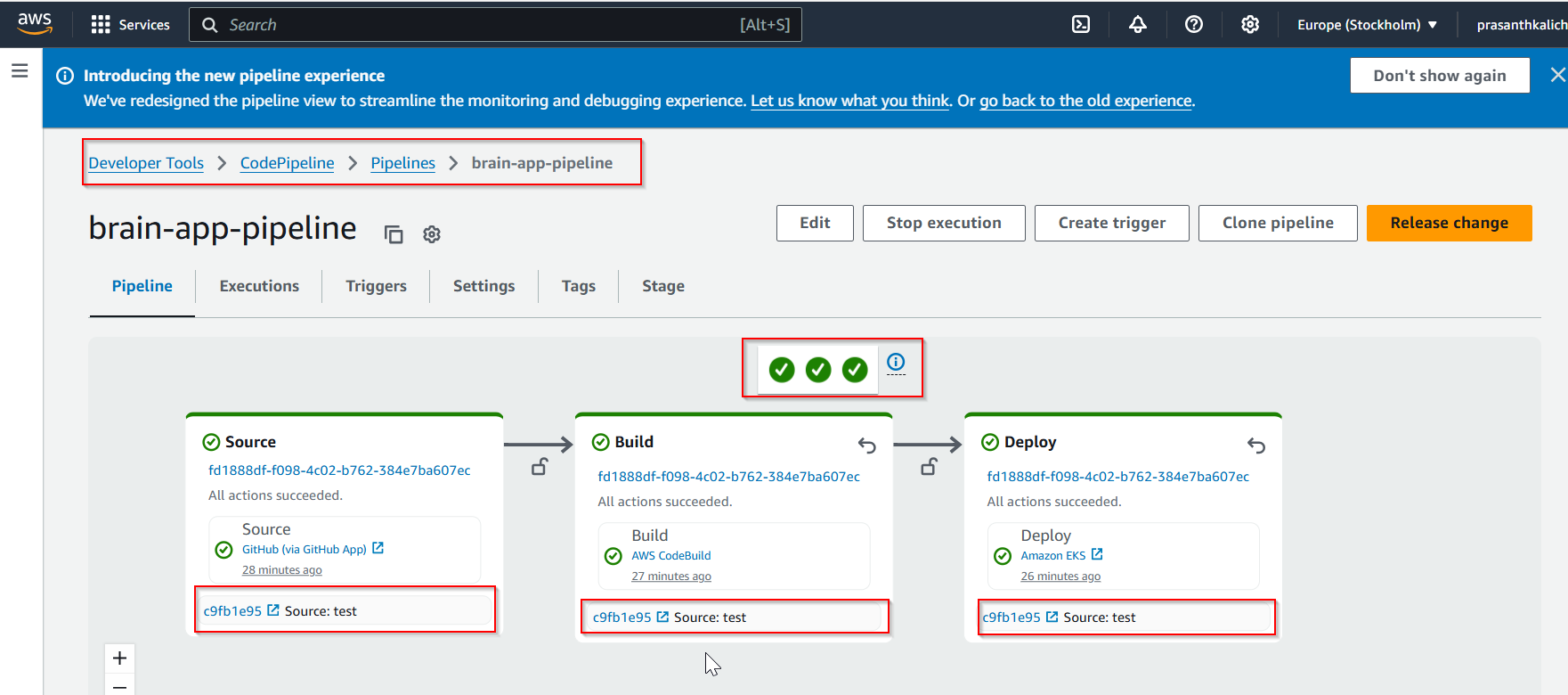
**Automation Pipeline Test:**



**Git Repo:**

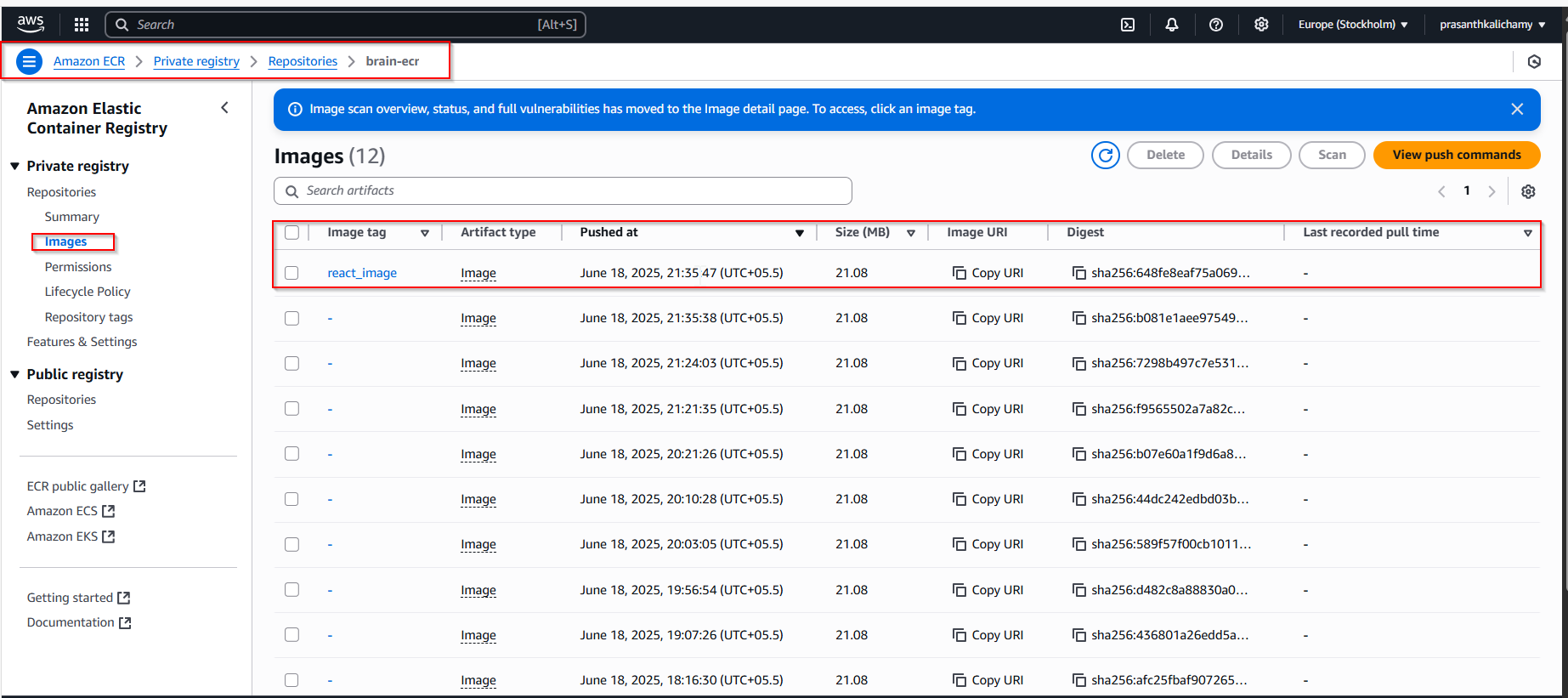


**Pipeline test triggered successfully:**

****

**ECR:**

**Image pushed to ECR repo correctly:**



**Build Log:**

[Container] 2025/06/20 00:10:17.156554 Running on CodeBuild On-demand

[Container] 2025/06/20 00:10:17.156563 Waiting for agent ping

[Container] 2025/06/20 00:10:17.759659 Waiting for DOWNLOAD\_SOURCE

[Container] 2025/06/20 00:10:19.280028 Phase is DOWNLOAD\_SOURCE

[Container] 2025/06/20 00:10:19.282234 CODEBUILD\_SRC\_DIR=/codebuild/output/src2422080798/src

[Container] 2025/06/20 00:10:19.282686 YAML location is /codebuild/output/src2422080798/src/buildspec.yml

[Container] 2025/06/20 00:10:19.284340 Setting HTTP client timeout to higher timeout for S3 source

[Container] 2025/06/20 00:10:19.284420 Processing environment variables

[Container] 2025/06/20 00:10:19.879846 No runtime version selected in buildspec.

[Container] 2025/06/20 00:10:19.907381 Moving to directory /codebuild/output/src2422080798/src

[Container] 2025/06/20 00:10:19.907404 Cache is not defined in the buildspec

[Container] 2025/06/20 00:10:20.125625 Skip cache due to: no paths specified to be cached

[Container] 2025/06/20 00:10:20.125899 Registering with agent

[Container] 2025/06/20 00:10:20.369534 Phases found in YAML: 3

[Container] 2025/06/20 00:10:20.369558 PRE\_BUILD: 4 commands

[Container] 2025/06/20 00:10:20.369563 BUILD: 3 commands

[Container] 2025/06/20 00:10:20.369568 POST\_BUILD: 2 commands

[Container] 2025/06/20 00:10:20.369882 Phase complete: DOWNLOAD\_SOURCE State: SUCCEEDED

[Container] 2025/06/20 00:10:20.369901 Phase context status code: Message:

[Container] 2025/06/20 00:10:20.626329 Entering phase INSTALL

[Container] 2025/06/20 00:10:20.850555 Phase complete: INSTALL State: SUCCEEDED

[Container] 2025/06/20 00:10:20.850580 Phase context status code: Message:

[Container] 2025/06/20 00:10:20.880871 Entering phase PRE\_BUILD

[Container] 2025/06/20 00:10:21.119354 Running command echo Logging in to Amazon ECR...

Logging in to Amazon ECR...

[Container] 2025/06/20 00:10:21.126920 Running command aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 061039785879.dkr.ecr.eu-north-1.amazonaws.com

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded

[Container] 2025/06/20 00:10:30.828360 Running command echo Logging in to Docker Hub...

Logging in to Docker Hub...

[Container] 2025/06/20 00:10:30.835682 Running command echo "$DOCKERHUB\_PASS" | docker login -u "prasanth0003" --password-stdin

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded

[Container] 2025/06/20 00:10:31.895662 Phase complete: PRE\_BUILD State: SUCCEEDED

[Container] 2025/06/20 00:10:31.895681 Phase context status code: Message:

[Container] 2025/06/20 00:10:31.935195 Entering phase BUILD

[Container] 2025/06/20 00:10:31.936210 Running command echo Build started on `date`

Build started on Fri Jun 20 12:10:31 AM UTC 2025

[Container] 2025/06/20 00:10:31.944824 Running command docker build -t brain-app .

#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile

#1 transferring dockerfile: 511B done

#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/nginx:alpine

#2 ...

#3 [auth] library/nginx:pull token for registry-1.docker.io

#3 DONE 0.0s

#2 [internal] load metadata for docker.io/library/nginx:alpine

#2 DONE 1.5s

#4 [internal] load .dockerignore

#4 transferring context: 2B done

#4 DONE 0.0s

#5 [internal] load build context

#5 transferring context: 318.29kB done

#5 DONE 0.0s

#6 [1/4] FROM docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10

#6 resolve docker.io/library/nginx:alpine@sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10 0.0s done

#6 sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc15fb093229f648674 0B / 1.79MB 0.1s

#6 sha256:b464cfdf2a6319875aeb27359ec549790ce14d8214fcb16ef915e4530e5ed235 0B / 629B 0.1s

#6 sha256:65645c7bb6a0661892a8b03b89d0743208a18dd2f3f17a54ef4b76fb8e2f2a10 10.33kB / 10.33kB done

#6 sha256:62223d644fa234c3a1cc785ee14242ec47a77364226f1c811d2f669f96dc2ac8 2.50kB / 2.50kB done

#6 sha256:6769dc3a703c719c1d2756bda113659be28ae16cf0da58dd5fd823d6b9a050ea 10.79kB / 10.79kB done

#6 sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0B / 3.64MB 0.1s

#6 sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB 0.2s done

#6 sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947d00628cb5f2d5773 0B / 955B 0.2s

#6 sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc15fb093229f648674 1.79MB / 1.79MB 0.3s done

#6 extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.1s

#6 sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73cd6bca1350f582948 0B / 402B 0.3s

#6 sha256:b464cfdf2a6319875aeb27359ec549790ce14d8214fcb16ef915e4530e5ed235 629B / 629B 0.4s done

#6 sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947d00628cb5f2d5773 955B / 955B 0.4s done

#6 extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.1s done

#6 sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73cd6bca1350f582948 402B / 402B 0.4s done

#6 extracting sha256:61ca4f733c802afd9e05a32f0de0361b6d713b8b53292dc15fb093229f648674 0.1s done

#6 sha256:197eb75867ef4fcecd4724f17b0972ab0489436860a594a9445f8eaff8155053 0B / 1.21kB 0.5s

#6 sha256:34a64644b756511a2e217f0508e11d1a572085d66cd6dc9a555a082ad49a3102 0B / 1.40kB 0.5s

#6 sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f7ccc2954004f2a7d 0B / 15.52MB 0.5s

#6 sha256:197eb75867ef4fcecd4724f17b0972ab0489436860a594a9445f8eaff8155053 1.21kB / 1.21kB 0.5s done

#6 sha256:34a64644b756511a2e217f0508e11d1a572085d66cd6dc9a555a082ad49a3102 1.40kB / 1.40kB 0.5s done

#6 sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f7ccc2954004f2a7d 5.24MB / 15.52MB 0.6s

#6 extracting sha256:b464cfdf2a6319875aeb27359ec549790ce14d8214fcb16ef915e4530e5ed235 done

#6 extracting sha256:d7e5070240863957ebb0b5a44a5729963c3462666baa2947d00628cb5f2d5773 done

#6 sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f7ccc2954004f2a7d 15.52MB / 15.52MB 0.6s done

#6 extracting sha256:81bd8ed7ec6789b0cb7f1b47ee731c522f6dba83201ec73cd6bca1350f582948 done

#6 extracting sha256:197eb75867ef4fcecd4724f17b0972ab0489436860a594a9445f8eaff8155053 done

#6 extracting sha256:34a64644b756511a2e217f0508e11d1a572085d66cd6dc9a555a082ad49a3102 done

#6 extracting sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f7ccc2954004f2a7d 0.1s

#6 extracting sha256:39c2ddfd6010082a4a646e7ca44e95aca9bf3eaebc00f17f7ccc2954004f2a7d 0.3s done

#6 DONE 1.2s

#7 [2/4] RUN rm -rf /usr/share/nginx/html/\*

#7 DONE 0.6s

#8 [3/4] COPY nginx.conf /etc/nginx/conf.d/default.conf

#8 DONE 0.1s

#9 [4/4] COPY dist/ /usr/share/nginx/html/

#9 DONE 0.1s

#10 exporting to image

#10 exporting layers 0.0s done

#10 writing image sha256:8daa3cc6b23f0231de9d44d7dc6c055b8cc5f5ec83b365a06429b9a3ba005bcf done

#10 naming to docker.io/library/brain-app done

#10 DONE 0.1s

[Container] 2025/06/20 00:10:35.842125 Running command docker tag brain-app:latest 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

[Container] 2025/06/20 00:10:35.864601 Phase complete: BUILD State: SUCCEEDED

[Container] 2025/06/20 00:10:35.864616 Phase context status code: Message:

[Container] 2025/06/20 00:10:35.897964 Entering phase POST\_BUILD

[Container] 2025/06/20 00:10:35.898968 Running command echo Build completed on `date`

Build completed on Fri Jun 20 12:10:35 AM UTC 2025

[Container] 2025/06/20 00:10:35.907369 Running command docker push 061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr:react\_image

The push refers to repository [061039785879.dkr.ecr.eu-north-1.amazonaws.com/brain-ecr]

2ff41745db90: Preparing

d0750d479125: Preparing

0927251c1064: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

811a4dbbf4a5: Waiting

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

0d853d50b128: Layer already exists

947e805a4ac7: Layer already exists

b8d7d1d22634: Layer already exists

811a4dbbf4a5: Layer already exists

c56f134d3805: Layer already exists

e244aa659f61: Layer already exists

d71eae0084c1: Layer already exists

08000c18d16d: Layer already exists

d0750d479125: Pushed

0927251c1064: Pushed

2ff41745db90: Pushed

react\_image: digest: sha256:1f8f4a86cd996a0ff0d3b2a458fbe22ef95210ebeabb9bcfce34bd73dd4f0470 size: 2613

[Container] 2025/06/20 00:10:37.651297 Phase complete: POST\_BUILD State: SUCCEEDED

[Container] 2025/06/20 00:10:37.651313 Phase context status code: Message:

[Container] 2025/06/20 00:10:37.697753 Set report auto-discover timeout to 5 seconds

[Container] 2025/06/20 00:10:37.697798 Expanding base directory path: .

[Container] 2025/06/20 00:10:37.700767 Assembling file list

[Container] 2025/06/20 00:10:37.700779 Expanding .

[Container] 2025/06/20 00:10:37.703838 Expanding file paths for base directory .

[Container] 2025/06/20 00:10:37.703851 Assembling file list

[Container] 2025/06/20 00:10:37.703854 Expanding \*\*/\*

[Container] 2025/06/20 00:10:37.707073 No matching auto-discover report paths found

[Container] 2025/06/20 00:10:37.707092 Report auto-discover file discovery took 0.009340 seconds

[Container] 2025/06/20 00:10:37.707108 Phase complete: UPLOAD\_ARTIFACTS State: SUCCEEDED

[Container] 2025/06/20 00:10:37.707116 Phase context status code: Message:

**Deploy Log:**

[Container] 2025/06/20 00:11:16.538234 Running on CodeBuild On-demand

[Container] 2025/06/20 00:11:16.538249 Waiting for agent ping

[Container] 2025/06/20 00:11:16.739659 Waiting for DOWNLOAD\_SOURCE

[Container] 2025/06/20 00:11:17.949591 Phase is DOWNLOAD\_SOURCE

[Container] 2025/06/20 00:11:17.950942 CODEBUILD\_SRC\_DIR=/codebuild/output/src823152702/src

[Container] 2025/06/20 00:11:17.951579 YAML location is /codebuild/readonly/buildspec.yml

[Container] 2025/06/20 00:11:17.955434 Setting HTTP client timeout to higher timeout for S3 source

[Container] 2025/06/20 00:11:17.955609 Processing environment variables

[Container] 2025/06/20 00:11:18.162225 No runtime version selected in buildspec.

[Container] 2025/06/20 00:11:18.211706 Moving to directory /codebuild/output/src823152702/src

[Container] 2025/06/20 00:11:18.211730 Cache is not defined in the buildspec

[Container] 2025/06/20 00:11:18.259879 Skip cache due to: no paths specified to be cached

[Container] 2025/06/20 00:11:18.260254 Registering with agent

[Container] 2025/06/20 00:11:18.306719 Phases found in YAML: 1

[Container] 2025/06/20 00:11:18.306738 BUILD: 7 commands

[Container] 2025/06/20 00:11:18.306974 Phase complete: DOWNLOAD\_SOURCE State: SUCCEEDED

[Container] 2025/06/20 00:11:18.306984 Phase context status code: Message:

[Container] 2025/06/20 00:11:18.390813 Entering phase INSTALL

[Container] 2025/06/20 00:11:18.455676 Phase complete: INSTALL State: SUCCEEDED

[Container] 2025/06/20 00:11:18.455695 Phase context status code: Message:

[Container] 2025/06/20 00:11:18.503173 Entering phase PRE\_BUILD

[Container] 2025/06/20 00:11:18.521195 Phase complete: PRE\_BUILD State: SUCCEEDED

[Container] 2025/06/20 00:11:18.521214 Phase context status code: Message:

[Container] 2025/06/20 00:11:18.567956 Entering phase BUILD

[Container] 2025/06/20 00:11:18.615332 Running command mkdir -p /tmp/cp-action-source

[Container] 2025/06/20 00:11:18.622099 Running command export CODEPIPELINE\_INPUT\_ACTION\_SOURCE\_PATH=/tmp/cp-action-source

[Container] 2025/06/20 00:11:18.627539 Running command curl "https://d2lz4e8o9u2n9f.cloudfront.net/deploy-eks-aws-1/0.1.0.tgz" -o /tmp/cp-action-source/action-archive.tgz

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0

100 33.3M 100 33.3M 0 0 167M 0 --:--:-- --:--:-- --:--:-- 167M

[Container] 2025/06/20 00:11:19.681602 Running command tar -xvzf /tmp/cp-action-source/action-archive.tgz --strip-components=1 -C /tmp/cp-action-source

package/dist/helm/helm

package/dist/kubectl/kubectl

package/dist/index.js

package/dist/validationUtils.js

package/package.json

package/dist/index.d.ts.map

package/dist/index.js.map

package/dist/validationUtils.d.ts.map

package/dist/validationUtils.js.map

package/dist/index.d.ts

package/dist/validationUtils.d.ts

[Container] 2025/06/20 00:11:20.530778 Running command node $CODEPIPELINE\_INPUT\_ACTION\_SOURCE\_PATH/dist/index.js

Logging in to Amazon EKS cluster.

Running command: aws eks update-kubeconfig --region eu-north-1 --name brain-cluster

Installing kubectl...

Running command: chmod +x /tmp/cp-action-source/dist/kubectl/kubectl && sudo mv /tmp/cp-action-source/dist/kubectl/kubectl /usr/local/bin/kubectl

Applying Kubernetes manifests.

Running command: kubectl apply -f deployment.yaml -f service.yaml

Resources to watch: [ ]

Action completed.

[Container] 2025/06/20 00:11:38.812773 Running command [ -f /tmp/cp-action-source/action-output-variables.sh ] && chmod 755 /tmp/cp-action-source/action-output-variables.sh && source /tmp/cp-action-source/action-output-variables.sh || true

[Container] 2025/06/20 00:11:38.819278 Running command if [[ ! -z $CodePipeline\_ErrorCode || ! -z $CodePipeline\_ErrorSummary ]]; then exit 1; fi

[Container] 2025/06/20 00:11:38.840159 Phase complete: BUILD State: SUCCEEDED

[Container] 2025/06/20 00:11:38.840174 Phase context status code: Message:

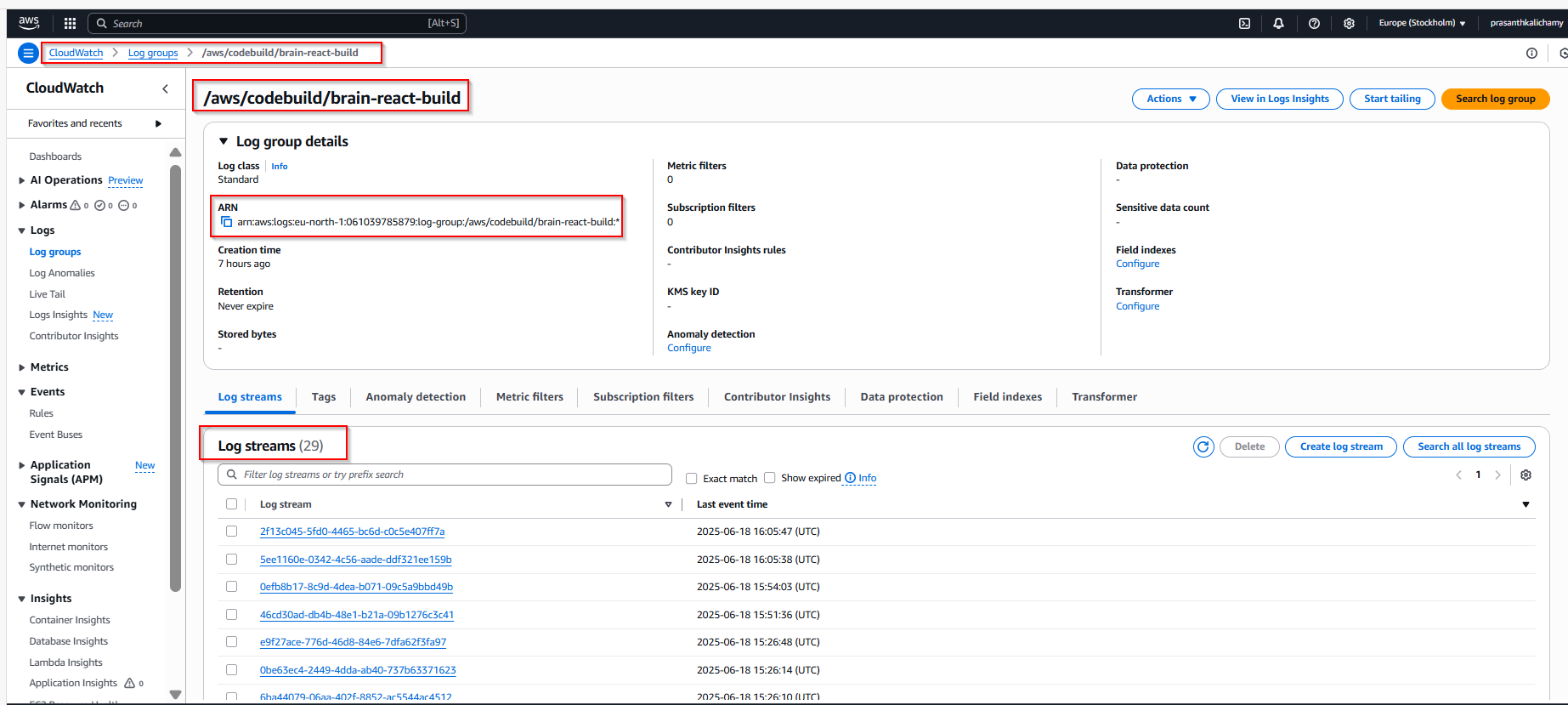
[Container] 2025/06/20 00:11:38.885161 Entering phase POST\_BUILD

[Container] 2025/06/20 00:11:38.902625 Phase complete: POST\_BUILD State: SUCCEEDED

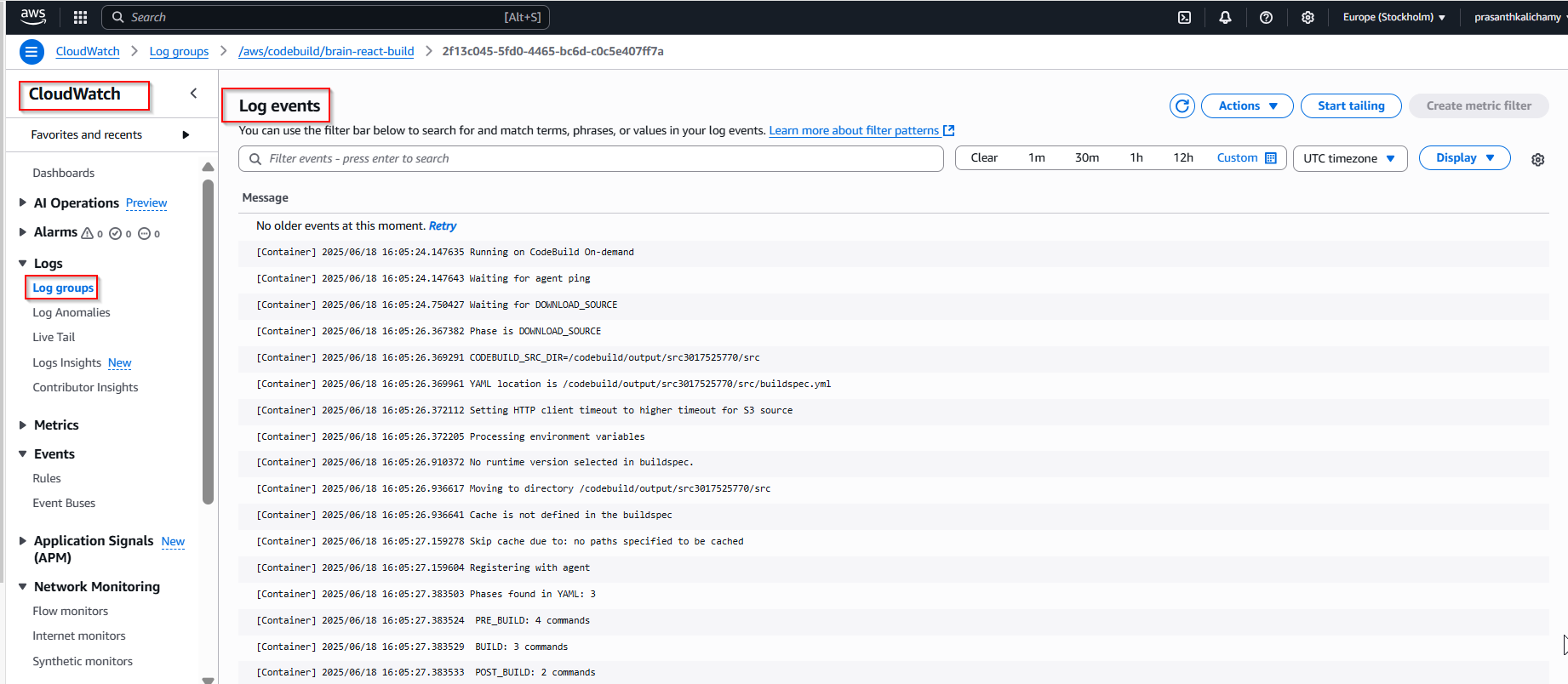
[Container] 2025/06/20 00:11:38.902642 Phase context status code: Message:

**Monitoring:**

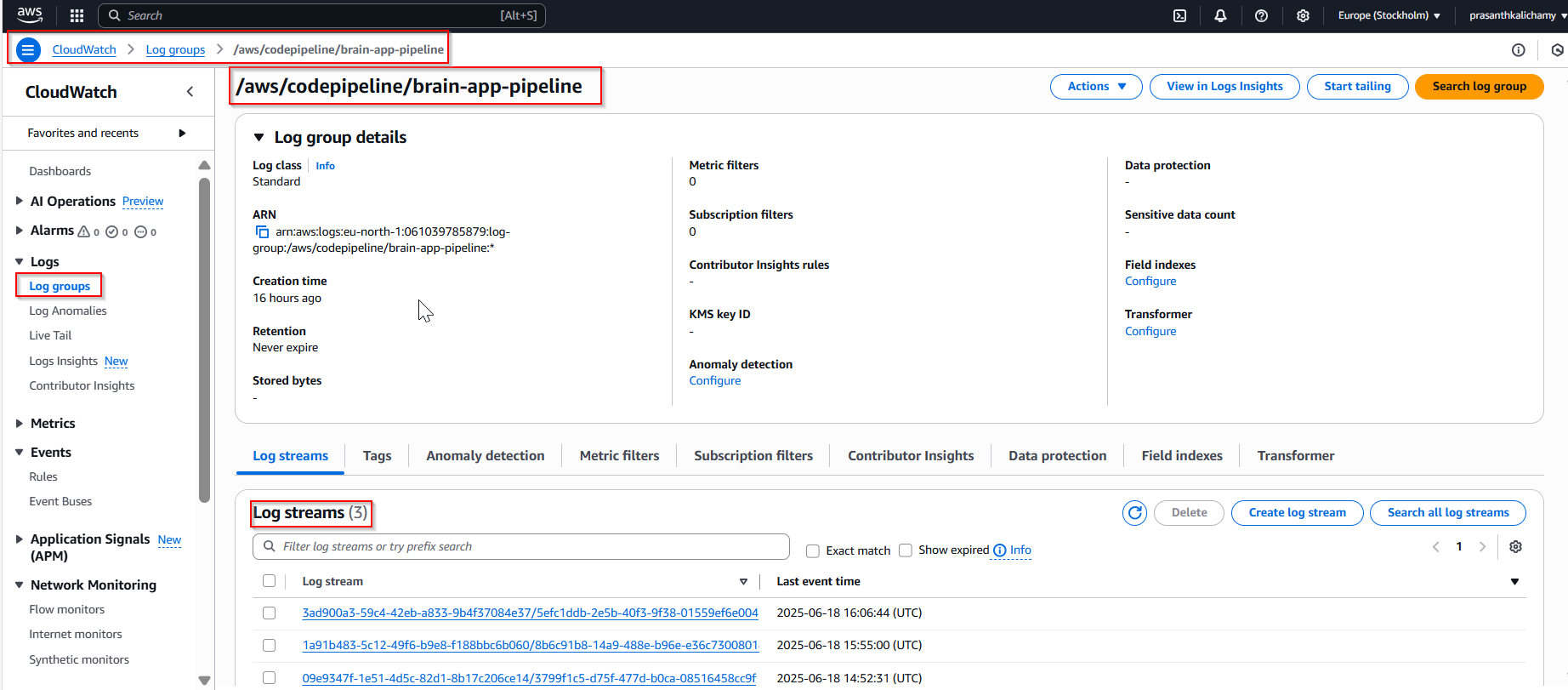
**Build Logs:**

****

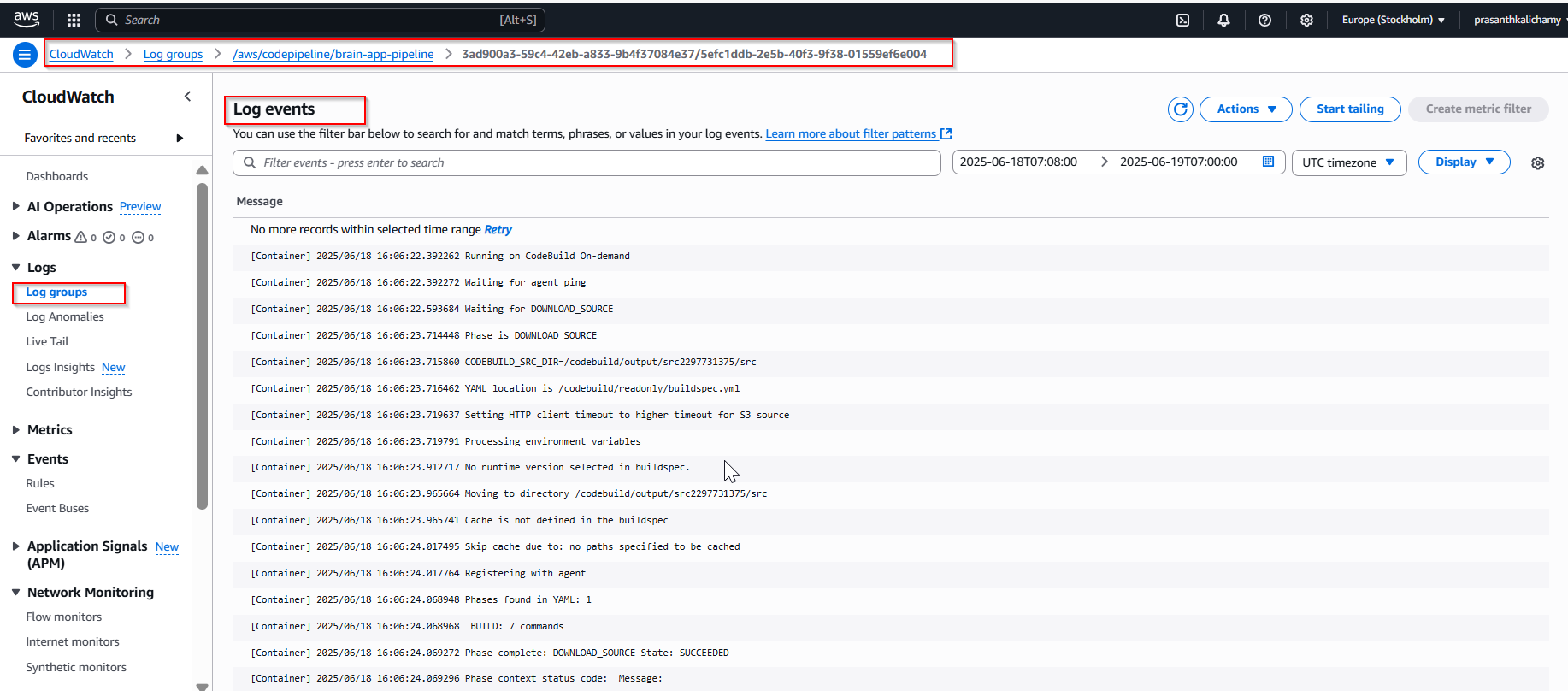
**Log Events:**

****

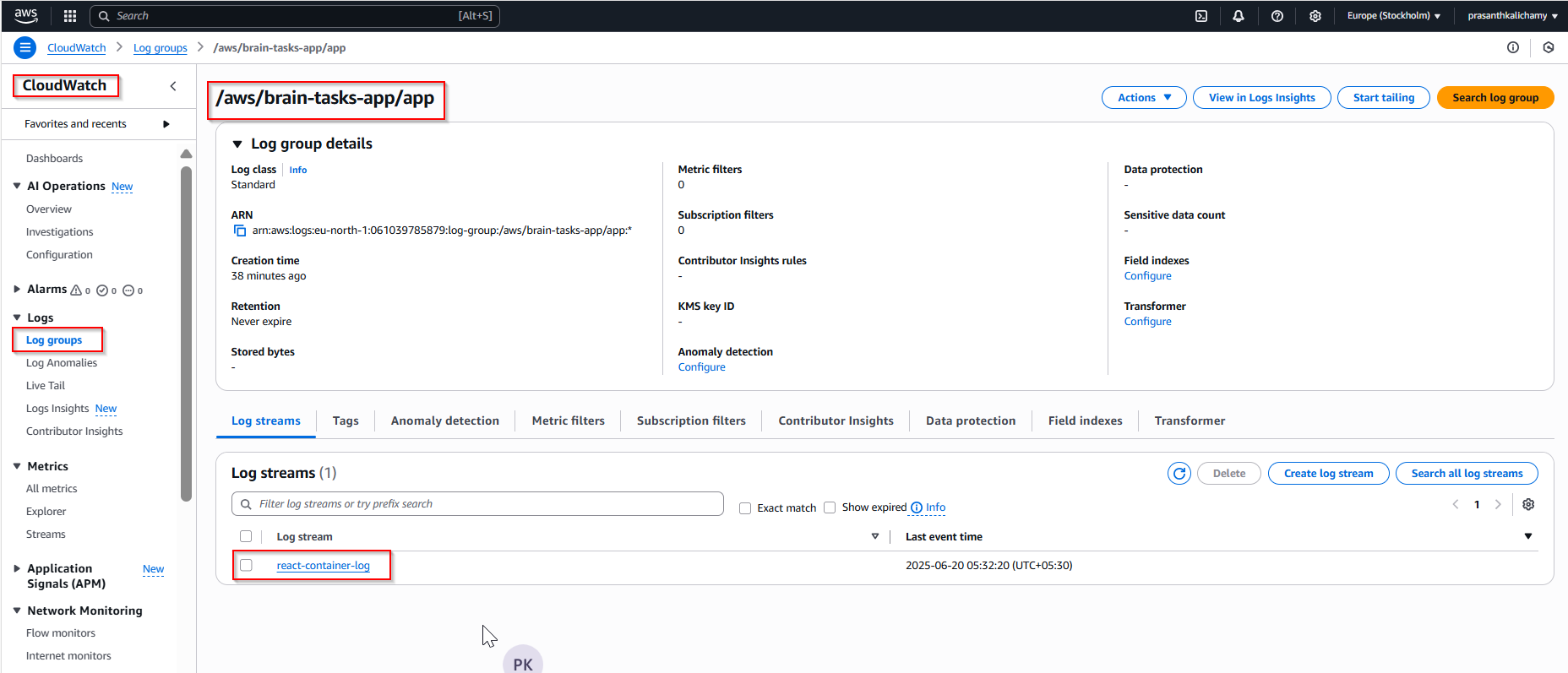
**Deploy and Pipeline Logs:**

****

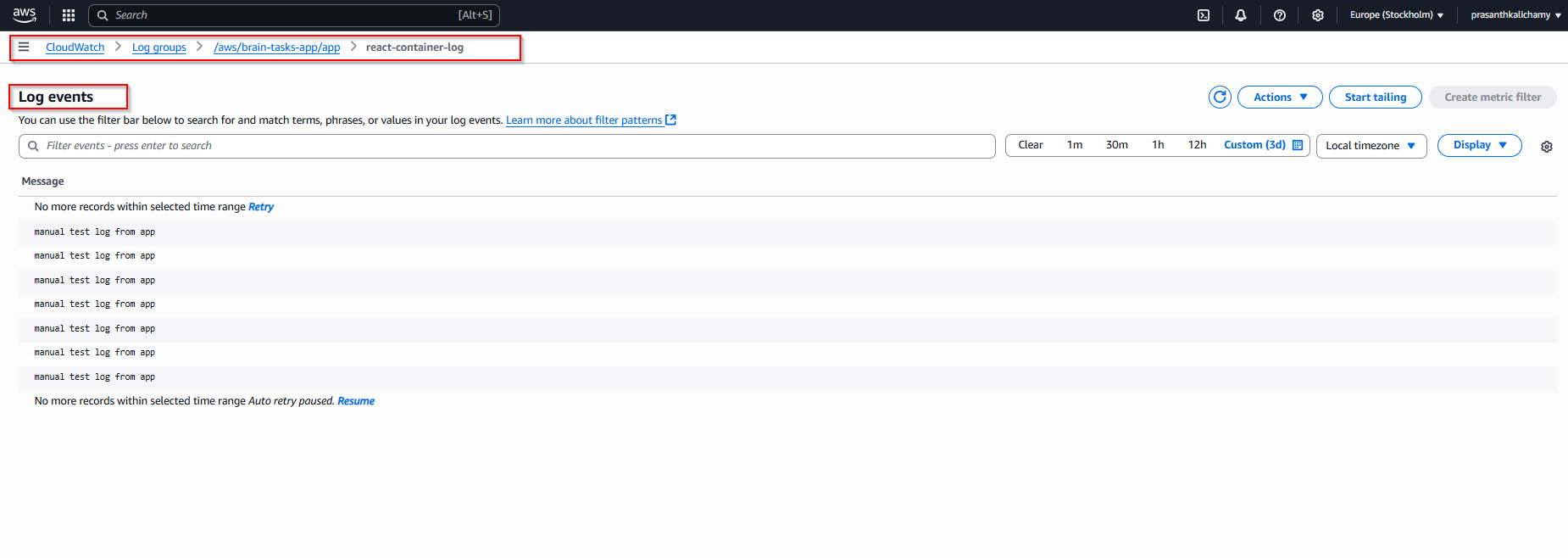
**Log Events:**

****

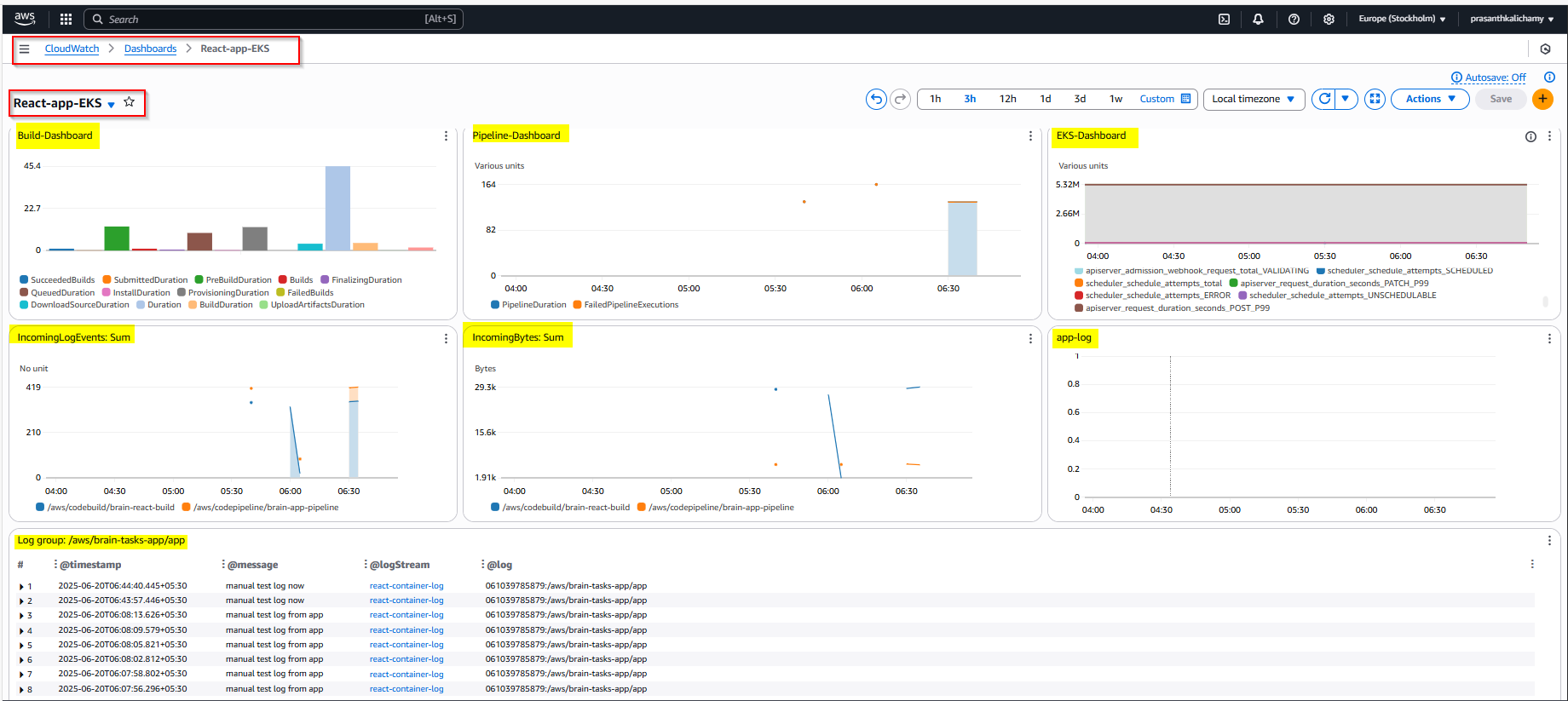
**Application Log:**

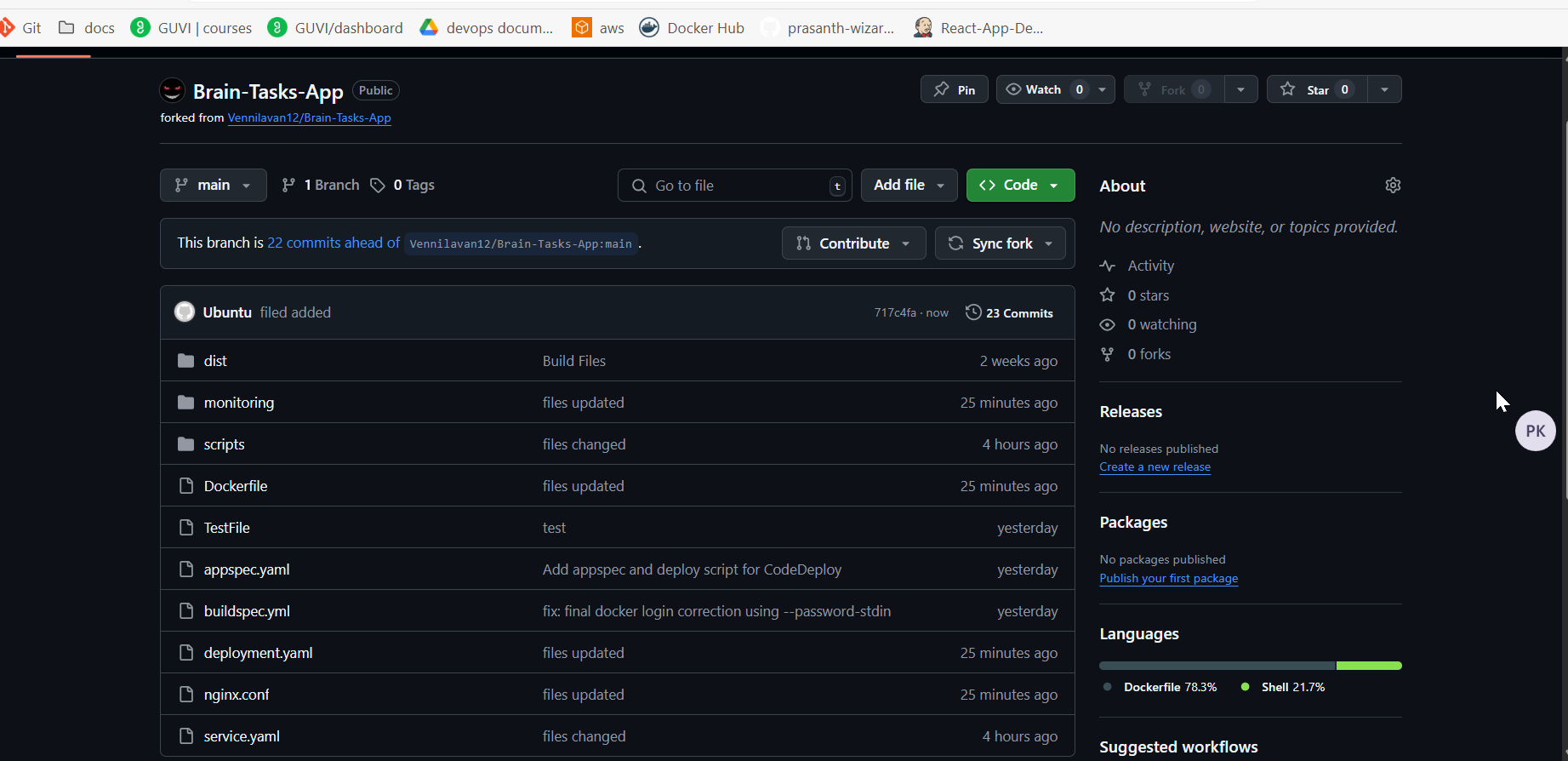
****

**Log Events:**

****

**Dashboard:**

****

****