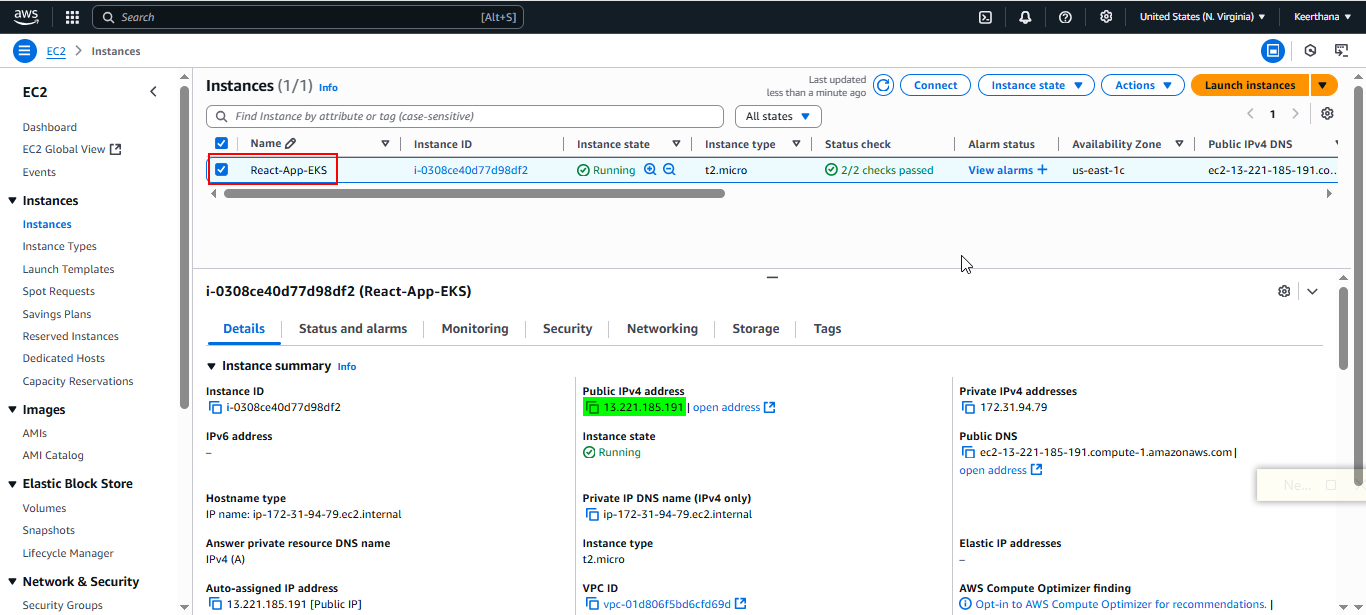
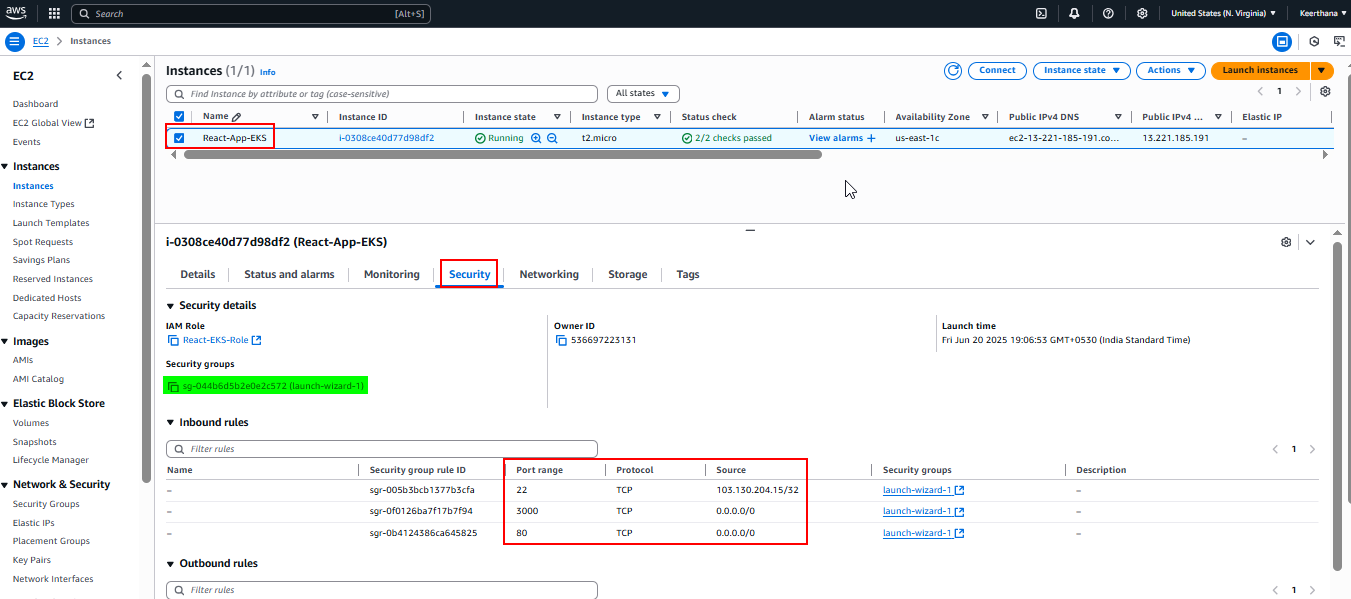
**React Application Deployment**

**Launch EC2 instance and connect with putty:**



**Instance – Security group:**



**Install Docker**

sudo apt update

sudo apt install docker.io -y

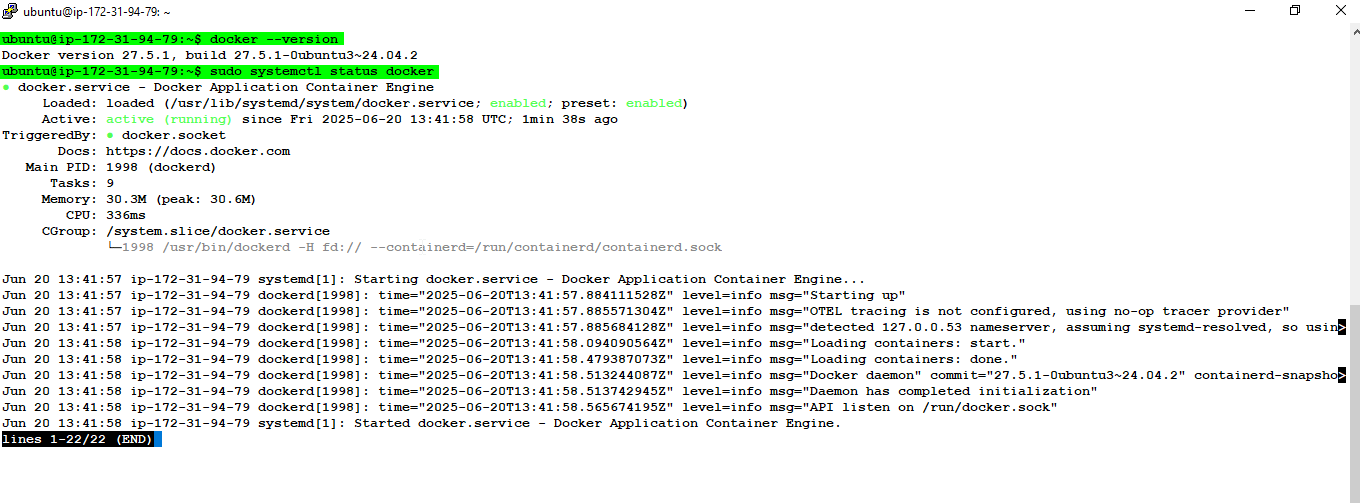
sudo systemctl enable docker

sudo systemctl start docker

sudo usermod -aG docker ubuntu

sudo usermod -aG docker $USER

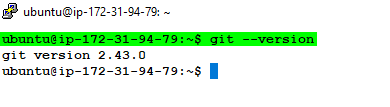
docker --version



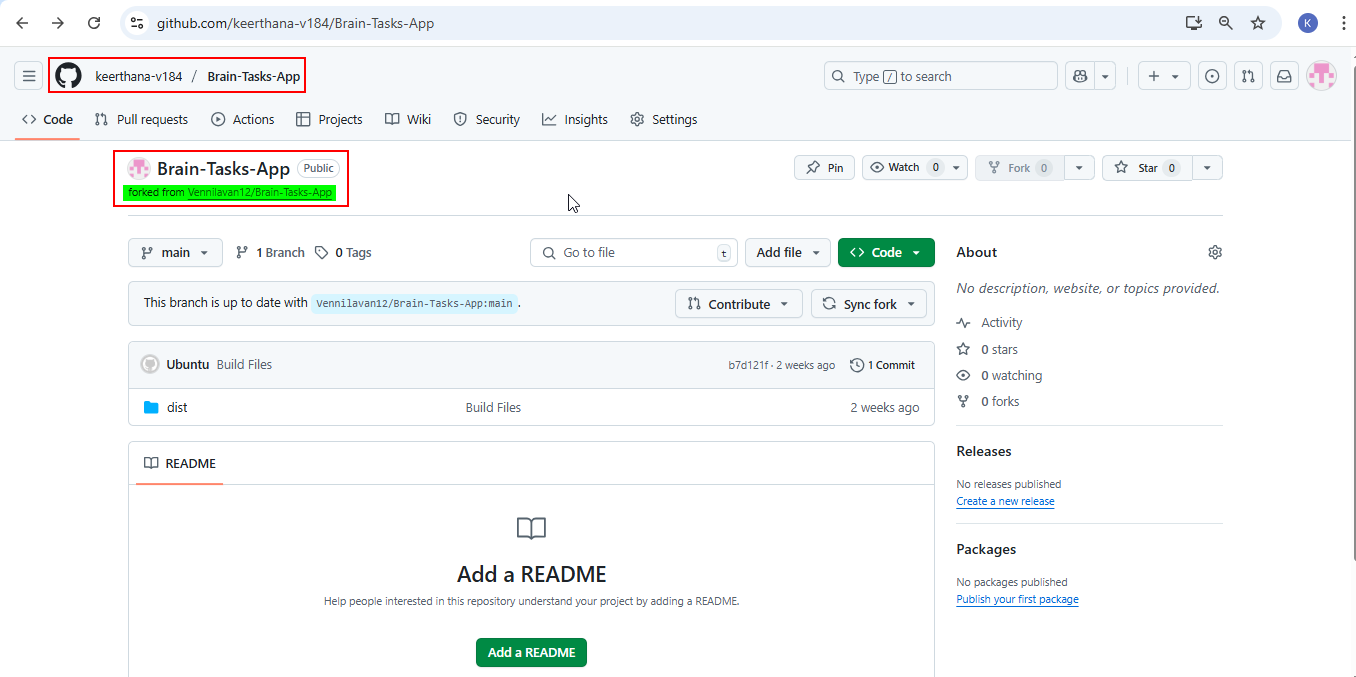
**Install git**

sudo apt install git -y

git --version



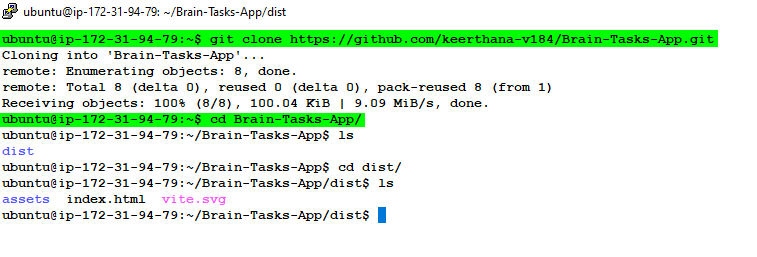
**Forked the given repo to my GitHub**



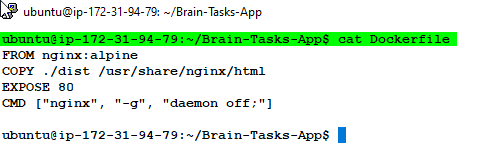
**Clone the Forked Repo and get into the directory**

git clone <https://github.com/keerthana-v184/Brain-Tasks-App.git>

cd Brain-Tasks-App



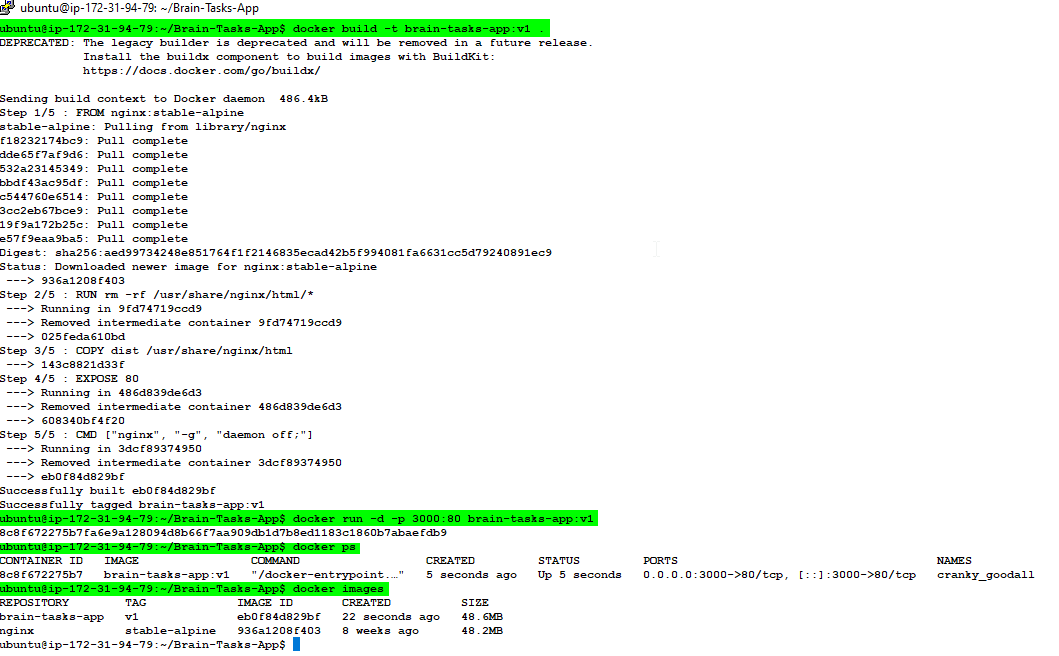
**Create Dockerfile:**



**Build & Run Docker Container to check locally:**

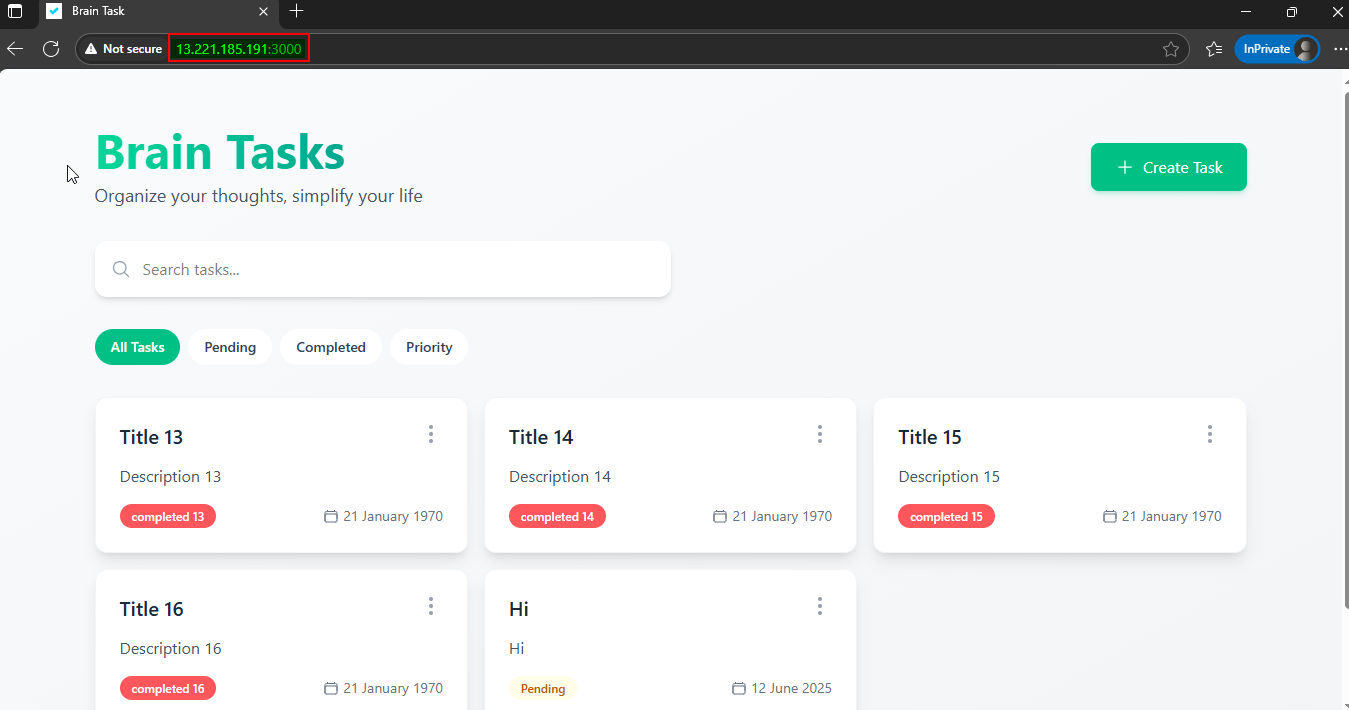
docker build -t brain-tasks-app:v1 .

docker run -d -p 3000:80 brain-tasks-app:v1



**Access the public IP with port 3000 in browser:**

* <http://13.221.185.191:3000/>



**Install AWS CLI**

sudo apt update

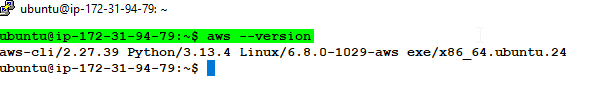
sudo apt install unzip -y

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

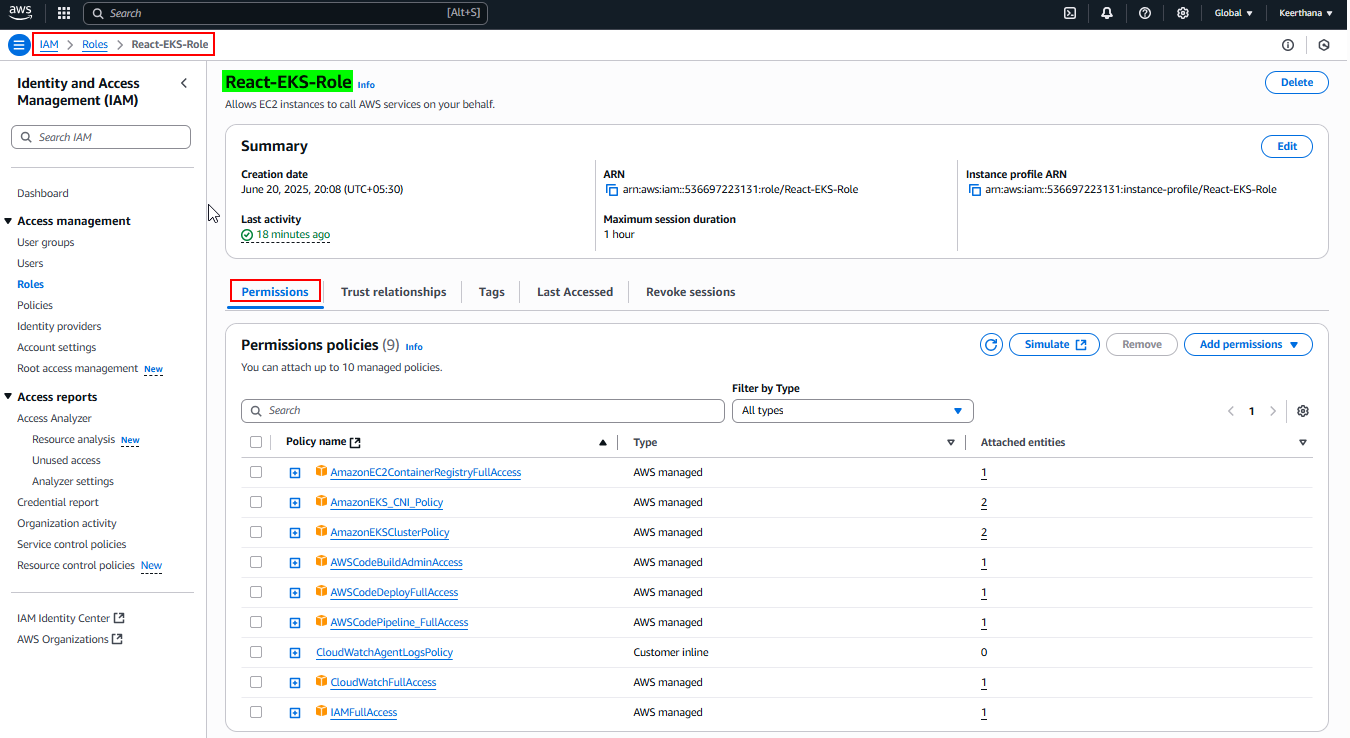
unzip awscliv2.zip

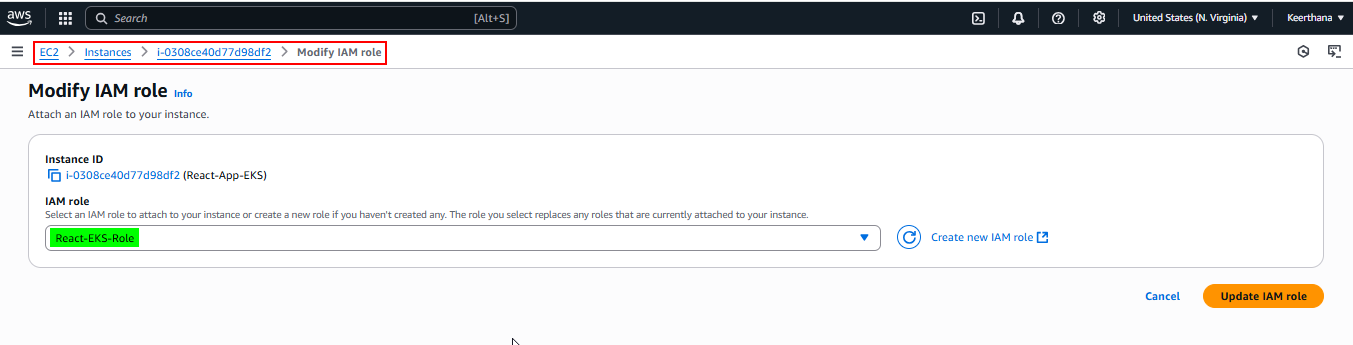
sudo ./aws/install

aws --version



**Create IAM Role and attach with instance:**





**Create ECR Repository**

aws ecr create-repository --repository-name brain-tasks-app --region us-east-1

**Authenticate Docker to ECR**

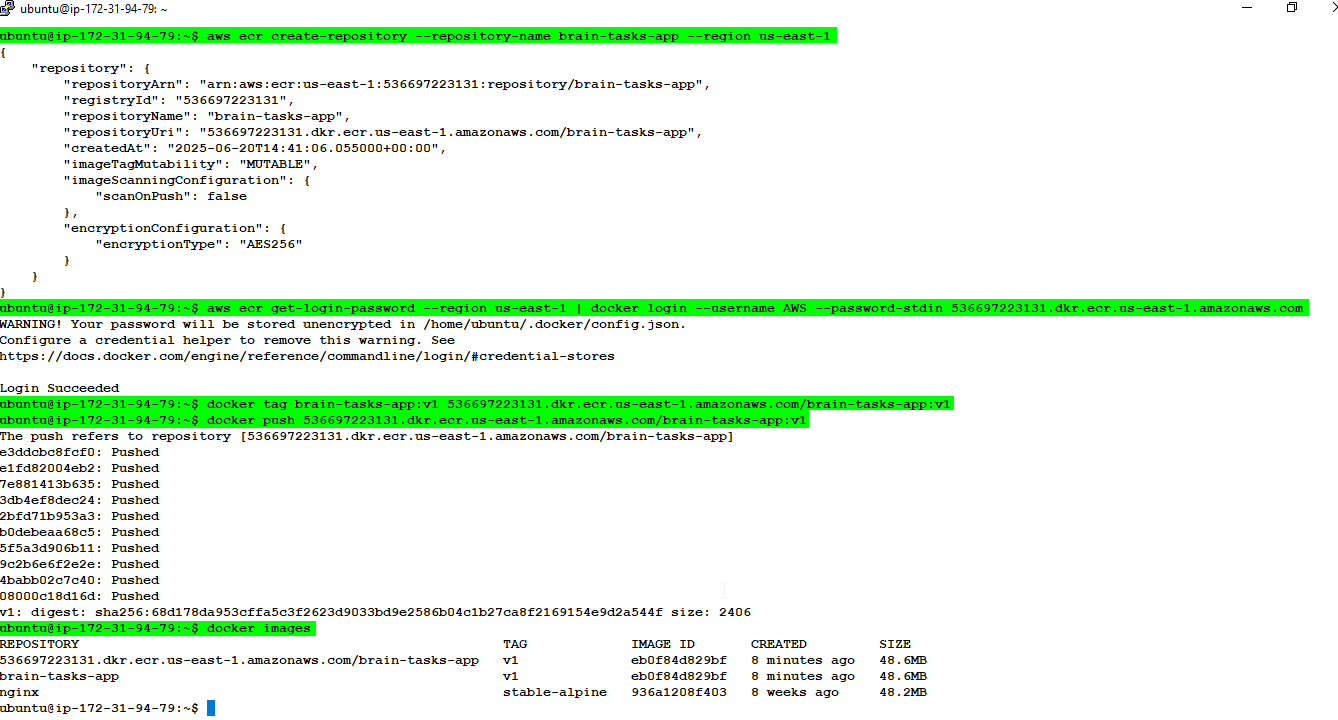
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 536697223131.dkr.ecr.us-east-1.amazonaws.com

**Tag added to Docker image for ECR**

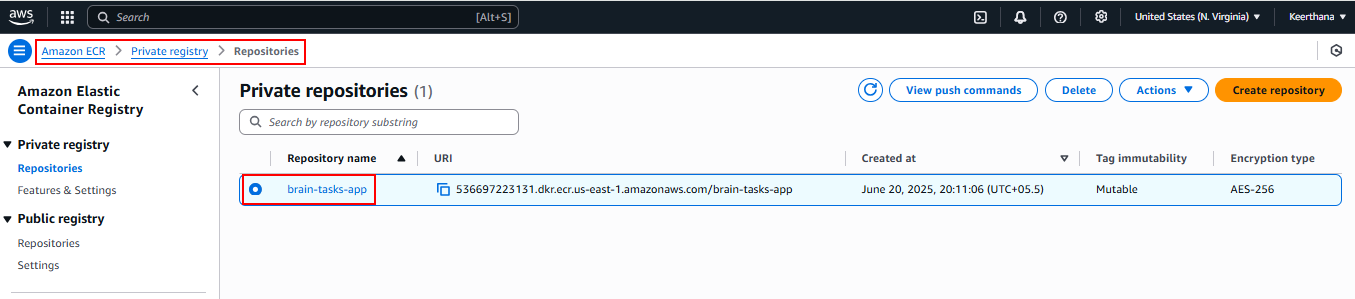
docker tag brain-tasks-app:v1 536697223131.dkr.ecr.us-east-1.amazonaws.com/brain-tasks-app:v1

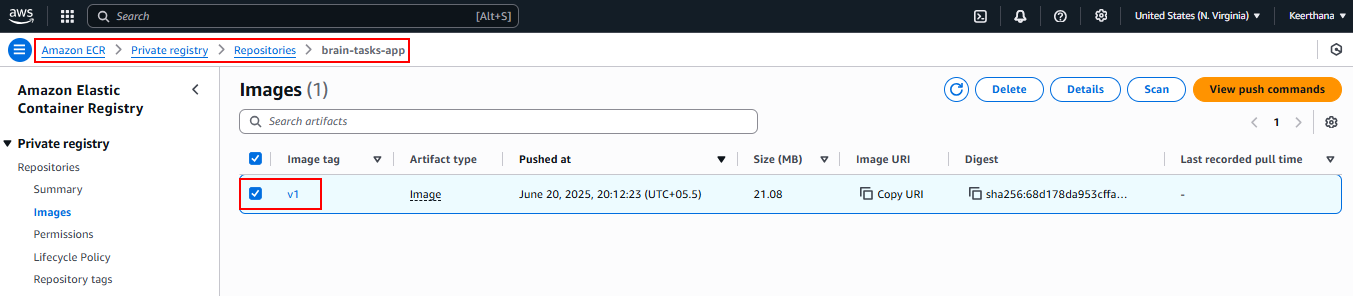
**Push the image to ECR**

docker push 536697223131.dkr.ecr.us-east-1.amazonaws.com/brain-tasks-app:v1



**AWS ECR:**





**Kubernetes:**

**Install kubectl**

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

chmod +x kubectl

sudo mv kubectl /usr/local/bin

**Install eksctl**

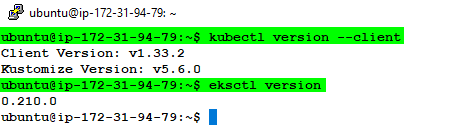
curl --location "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" -o eksctl.tar.gz

tar -xzf eksctl.tar.gz

sudo mv eksctl /usr/local/bin

kubectl version --client

eksctl version



**Create EKS Cluster**

eksctl create cluster \

--name brain-tasks-cluster \

--region us-east-1 \

--nodegroup-name brain-tasks-nodegroup \

--node-type t2.micro \

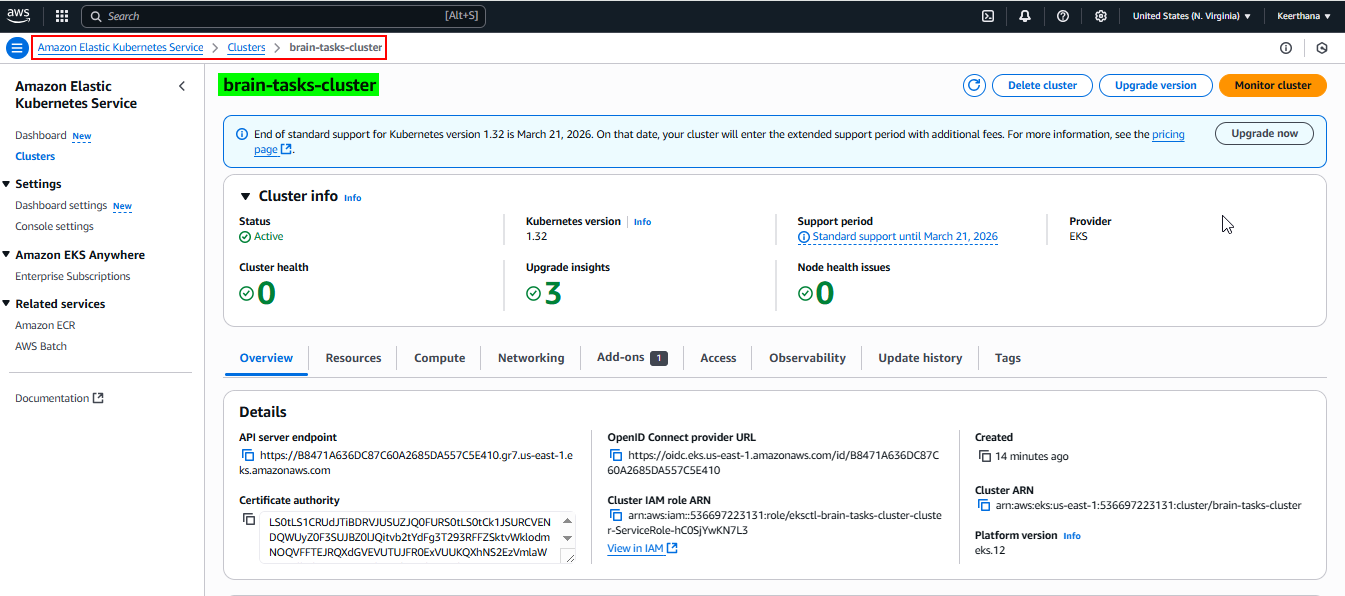
--nodes 3 \

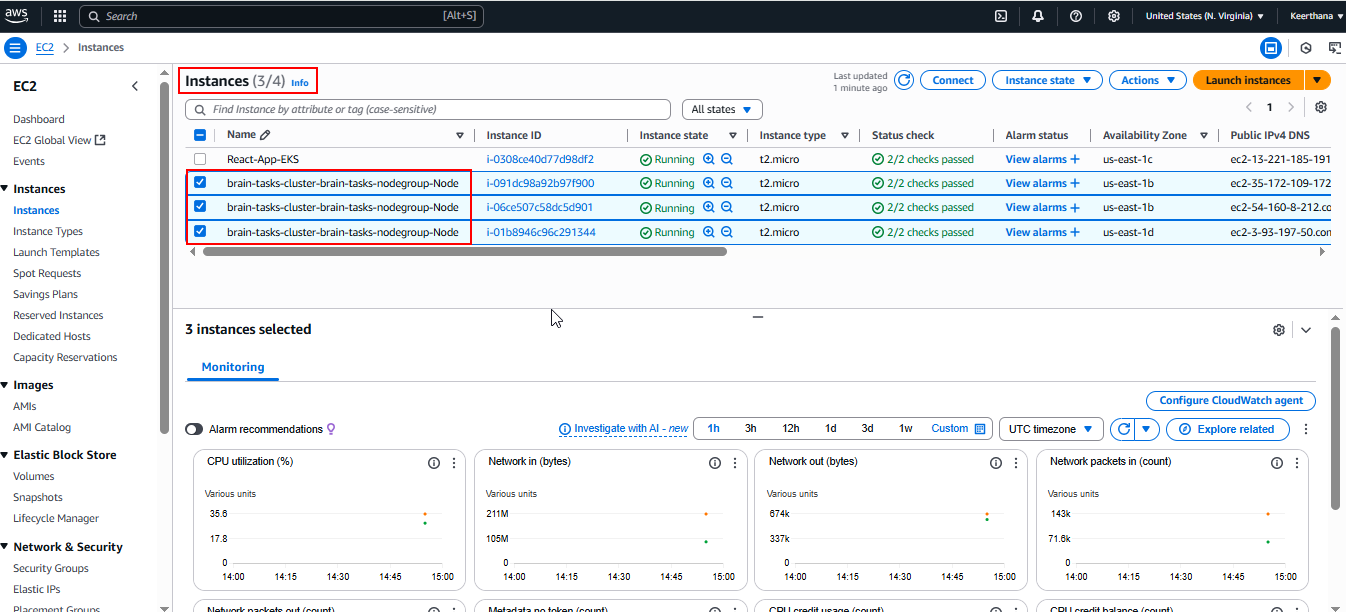
--nodes-min 2 \

--nodes-max 4 \

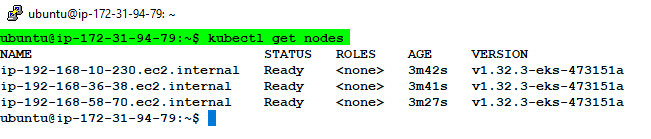
--managed

**AWS EKS:**





**Nodes created**

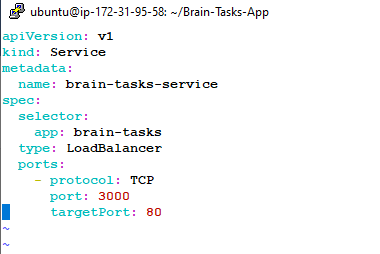


**Create Deployment and Service YAML inside GitHub repo:**

**deployment.yml**

****

**Service.yml**

****

**Confirmed They Work in EKS Cluster**

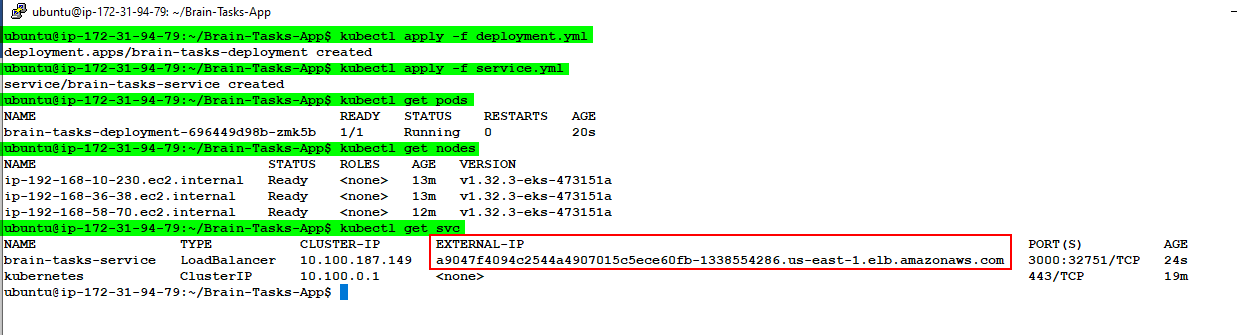
kubectl apply -f deployment.yml

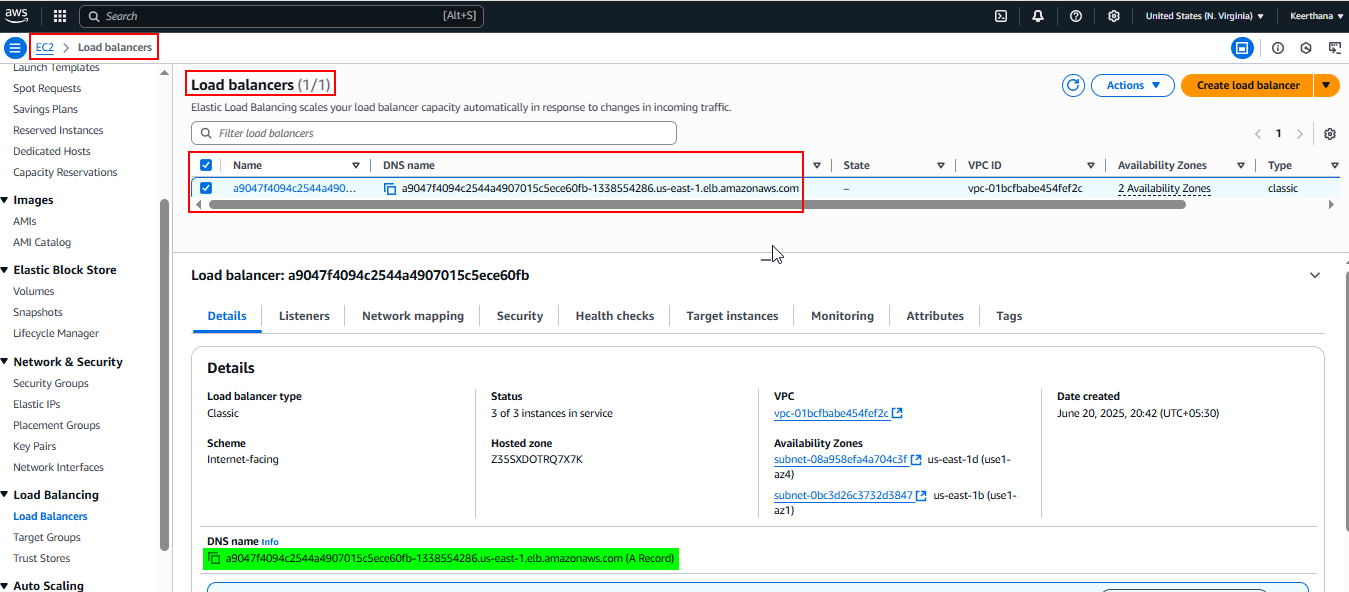
kubectl apply -f service.yml

kubectl get pods

kubectl get nodes

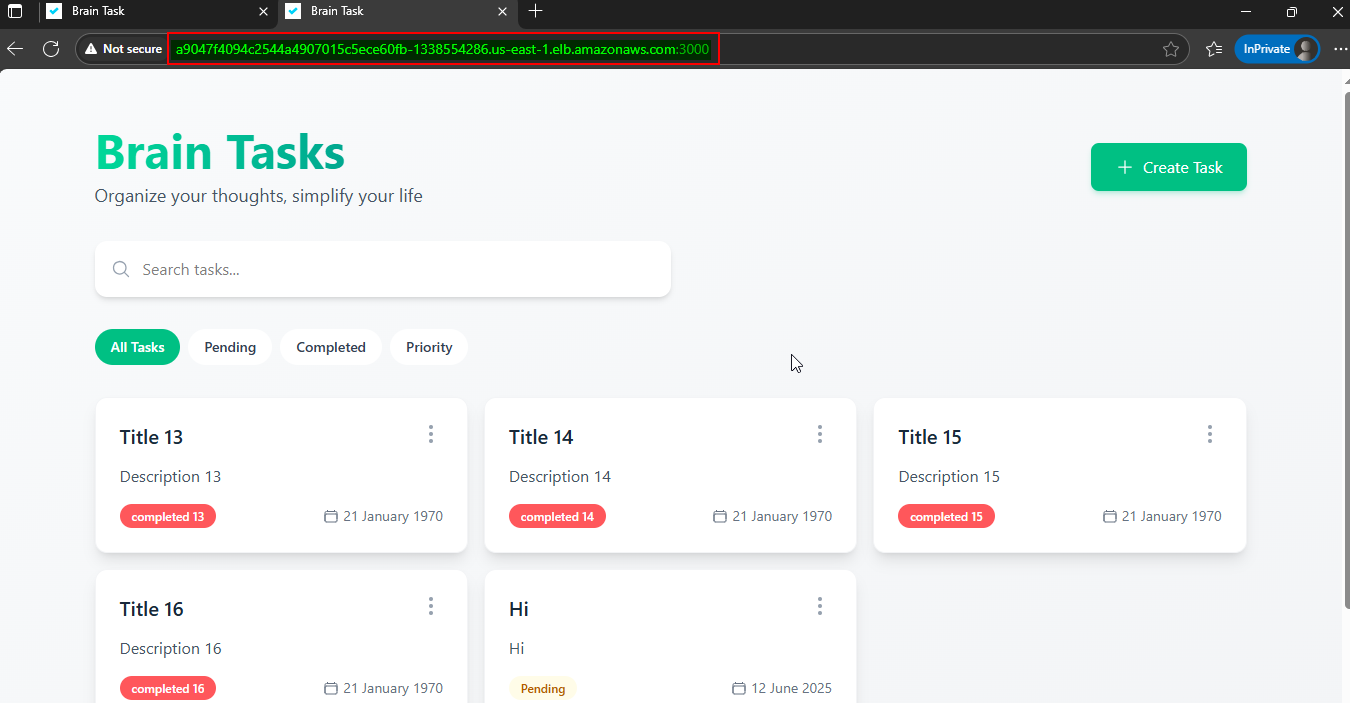
kubectl get svc



**Load Balancer created**  
  


**Access External-IP in browser with port 3000**

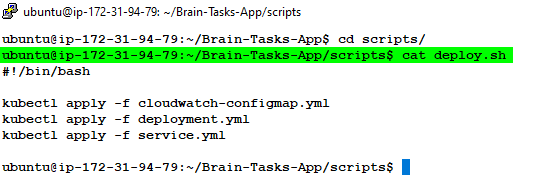
* <http://a9047f4094c2544a4907015c5ece60fb-1338554286.us-east-1.elb.amazonaws.com:3000/>



**Create script file to apply YAML files:**

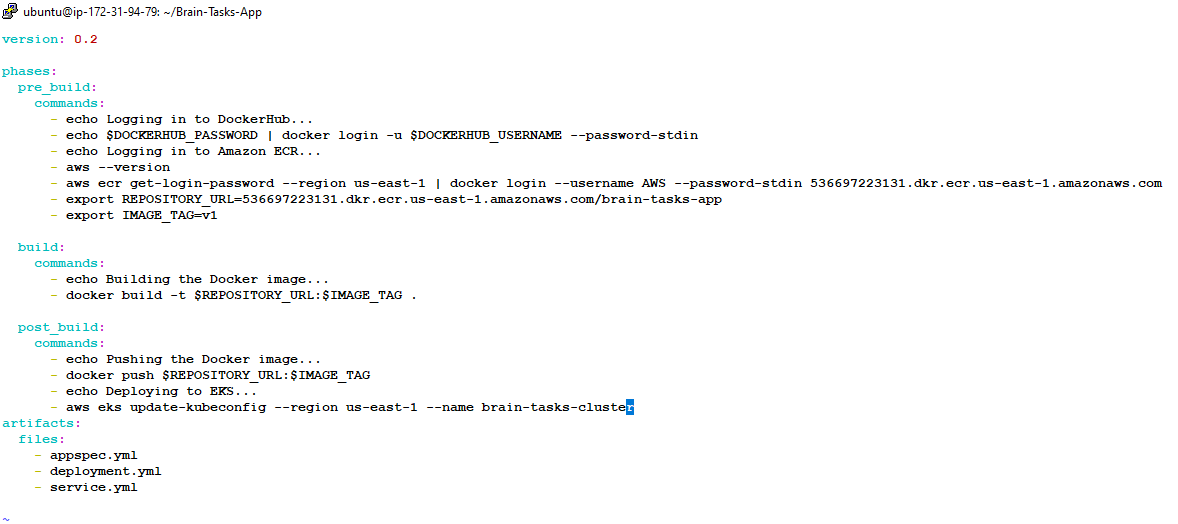
mkdir scripts && cd scripts

vi deploy.sh

****

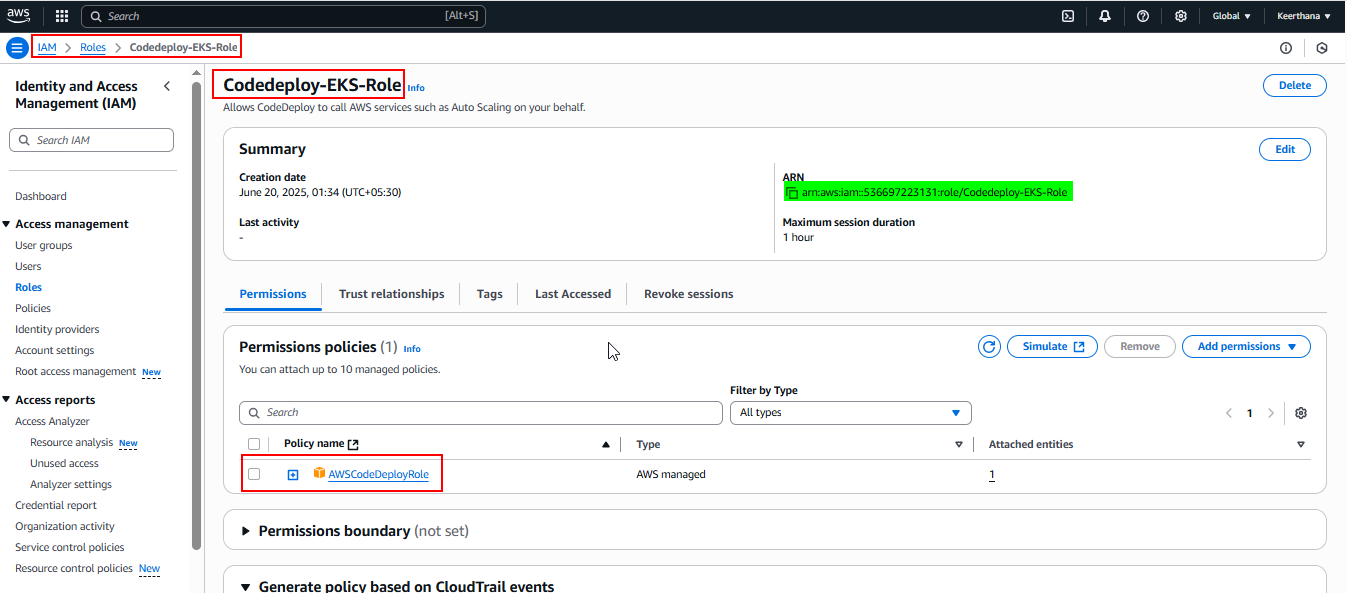
**Code Build:**

**Create buildspec.yml file**

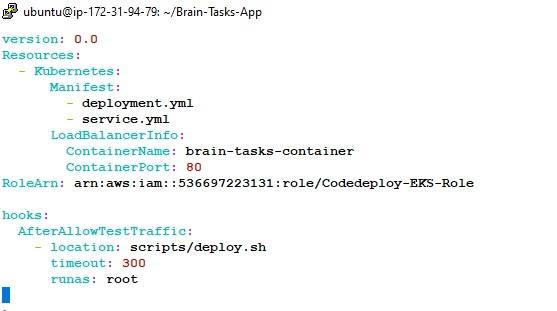


**Code Deploy:**

**Create IAM Role for code deploy and add in appspec.yml file:**



**Create appspec.yml file**

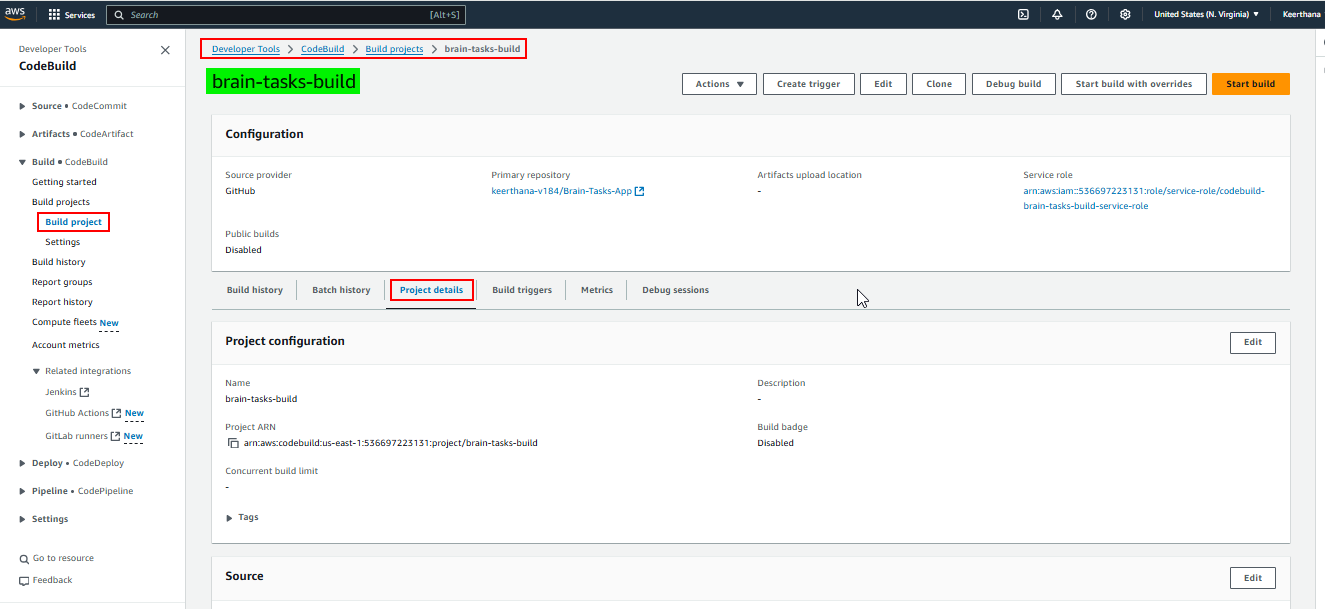
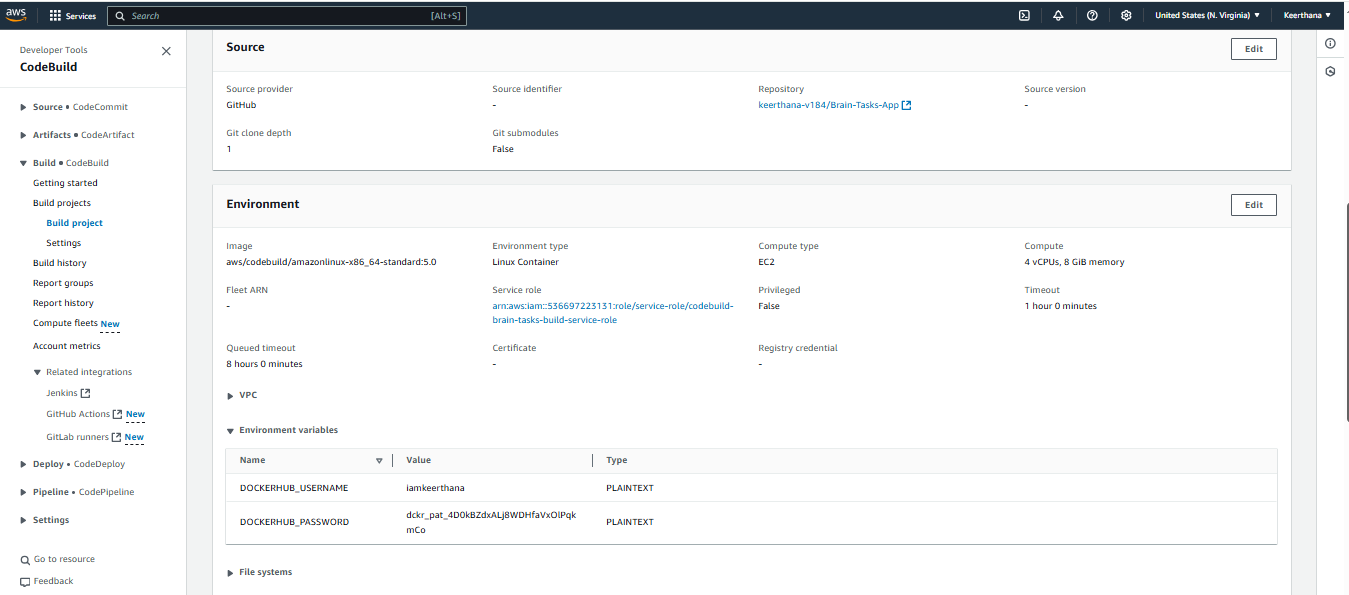


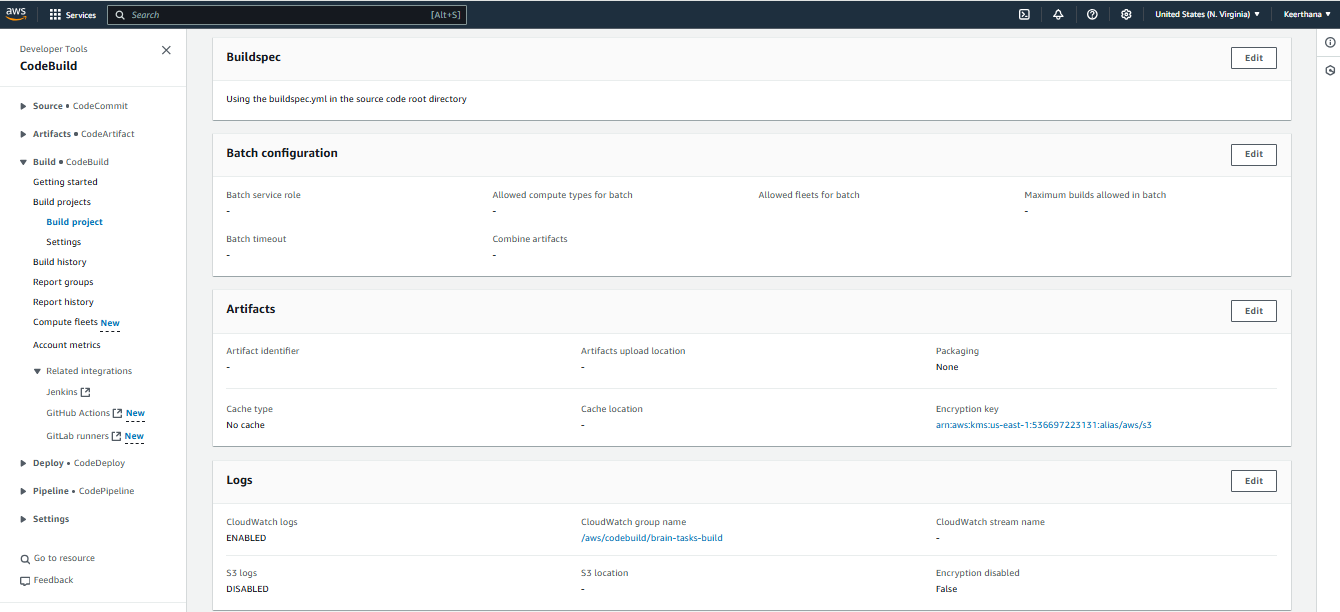
**Create cloudwatch-configmap.yml for collecting logs:**



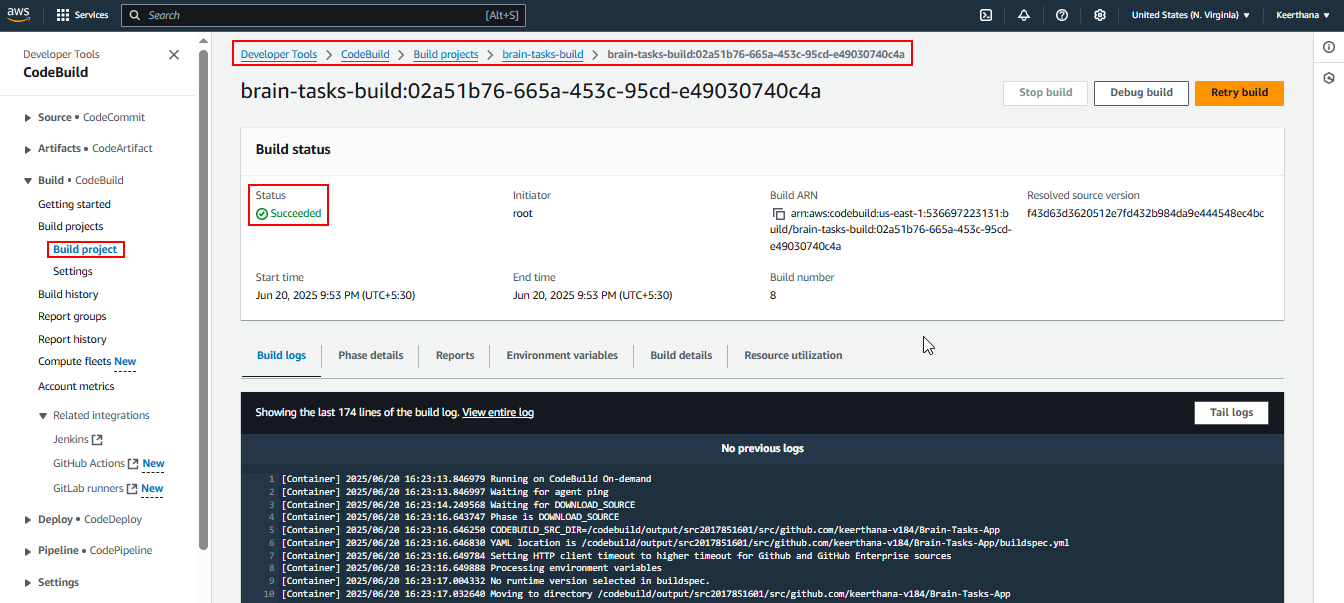
**Code Build in AWS Console**

**Build project creation:**

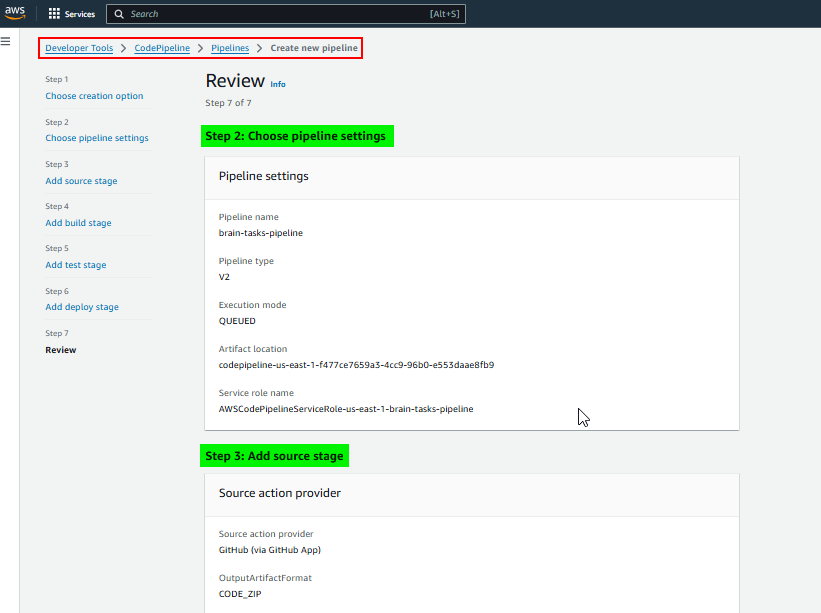
**  
  
**

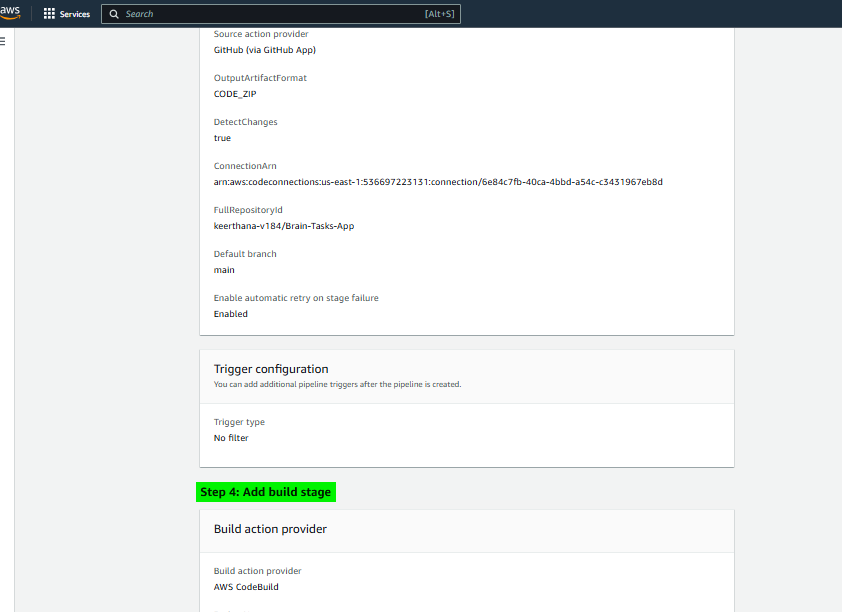
****

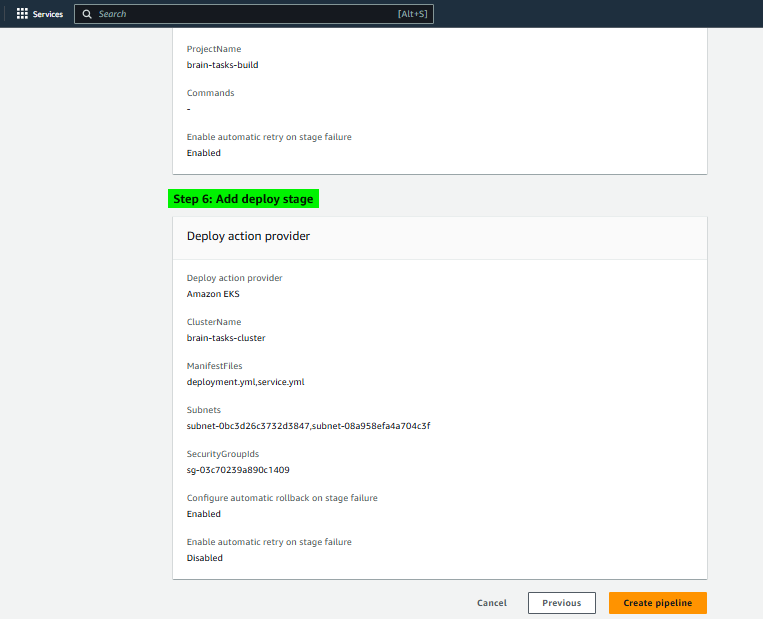
**Build succeed**

****

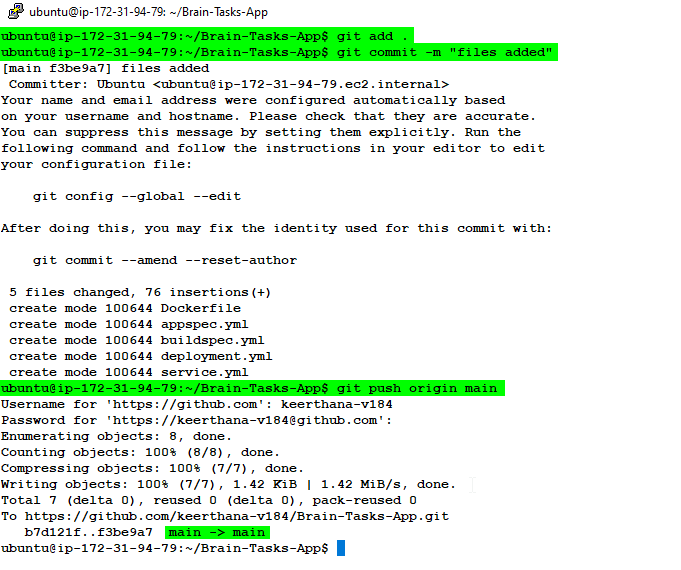
**Code Pipeline:**



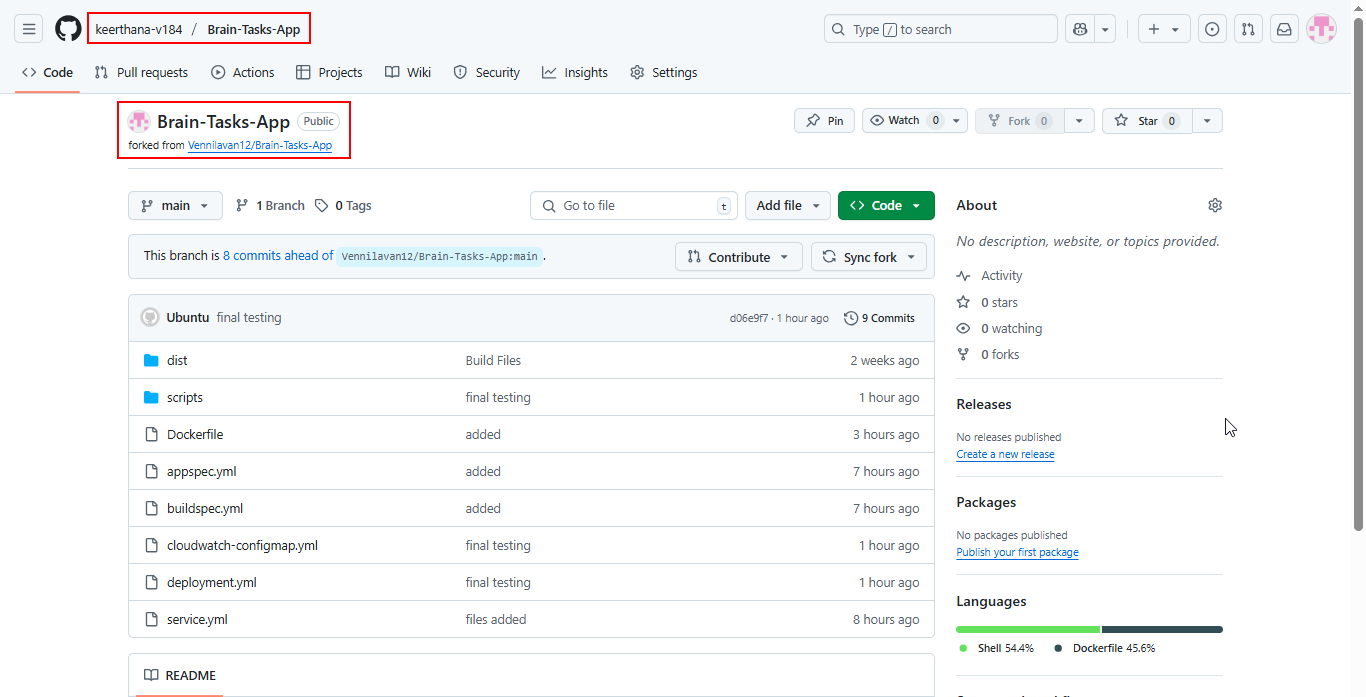




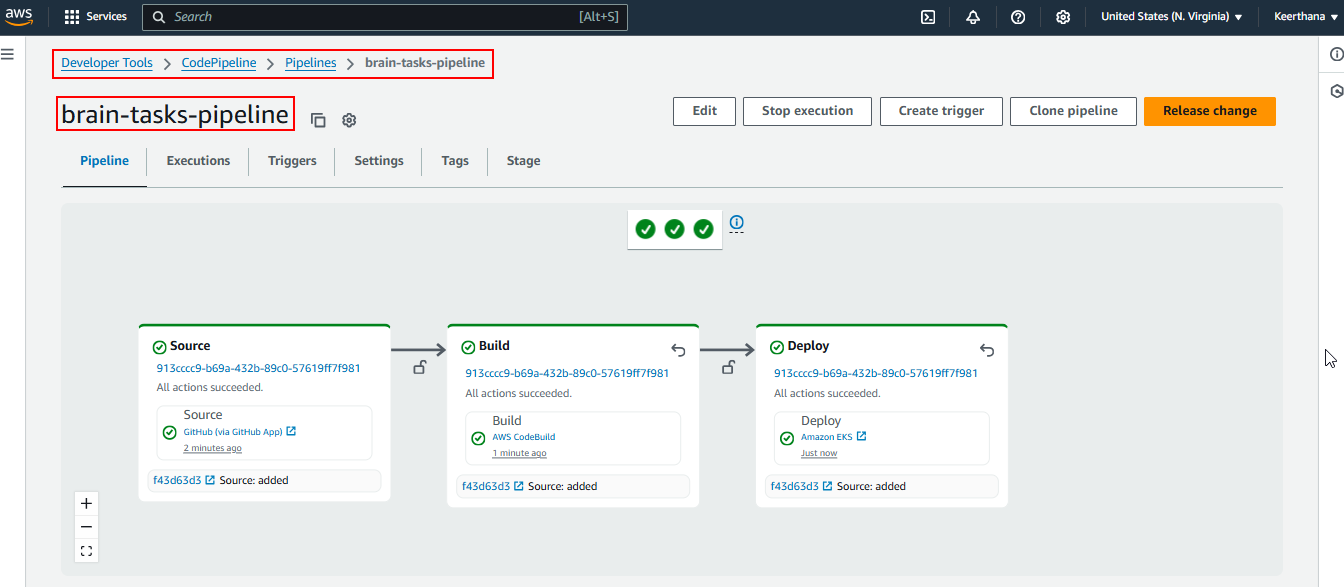
**Push the code files to a GitHub:**



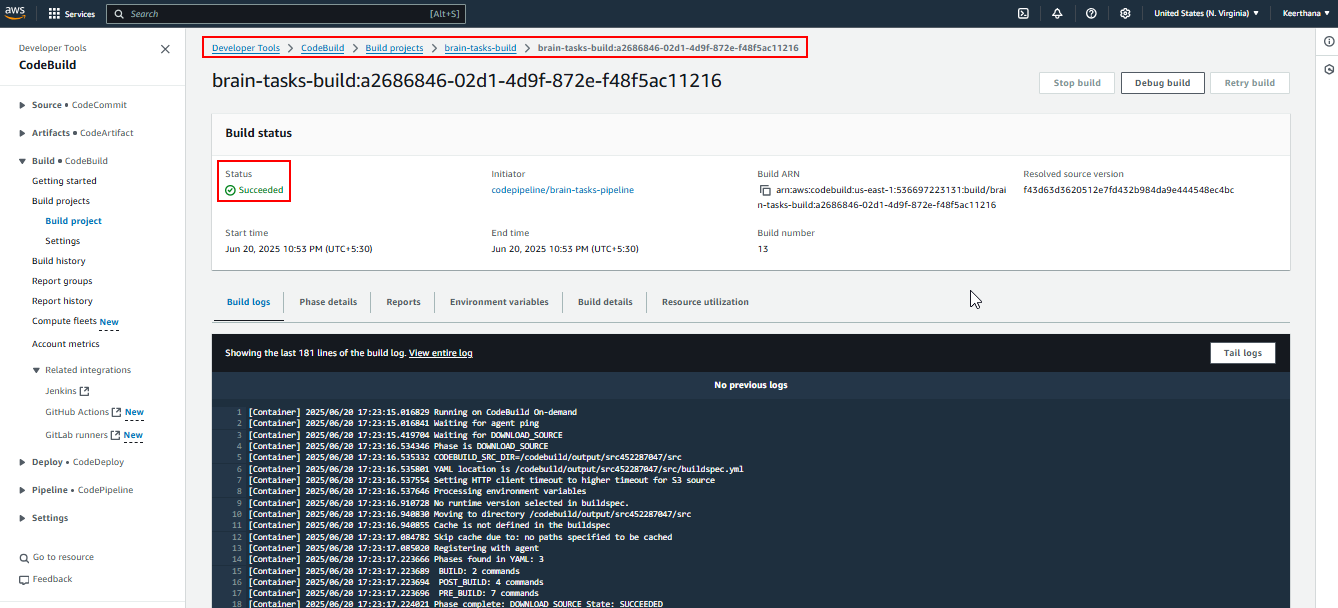
**GitHub Repo**

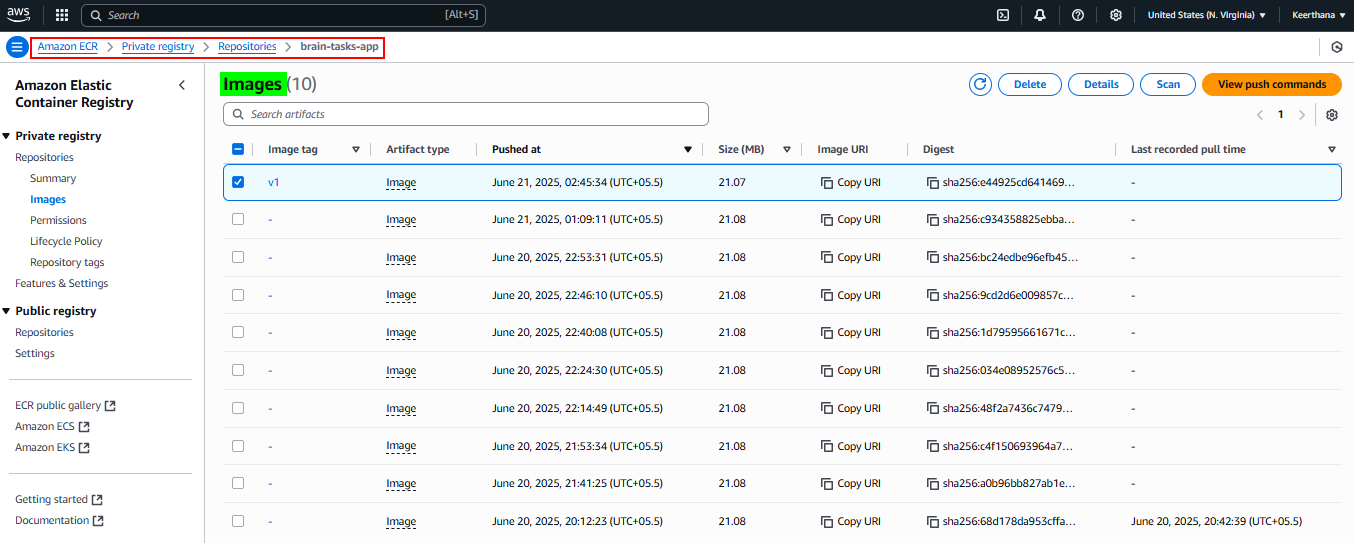


**Pipeline triggered and succeed:**



**After automation Build succeed and Image pushed to ECR:**





**Build console logs:**

1 [Container] 2025/06/20 17:23:15.016829 Running on CodeBuild On-demand

2 [Container] 2025/06/20 17:23:15.016841 Waiting for agent ping

3 [Container] 2025/06/20 17:23:15.419704 Waiting for DOWNLOAD\_SOURCE

4 [Container] 2025/06/20 17:23:16.534346 Phase is DOWNLOAD\_SOURCE

5 [Container] 2025/06/20 17:23:16.535332 CODEBUILD\_SRC\_DIR=/codebuild/output/src452287047/src

6 [Container] 2025/06/20 17:23:16.535801 YAML location is /codebuild/output/src452287047/src/buildspec.yml

7 [Container] 2025/06/20 17:23:16.537554 Setting HTTP client timeout to higher timeout for S3 source

8 [Container] 2025/06/20 17:23:16.537646 Processing environment variables

9 [Container] 2025/06/20 17:23:16.910728 No runtime version selected in buildspec.

10 [Container] 2025/06/20 17:23:16.940830 Moving to directory /codebuild/output/src452287047/src

11 [Container] 2025/06/20 17:23:16.940855 Cache is not defined in the buildspec

12 [Container] 2025/06/20 17:23:17.084782 Skip cache due to: no paths specified to be cached

13 [Container] 2025/06/20 17:23:17.085020 Registering with agent

14 [Container] 2025/06/20 17:23:17.223666 Phases found in YAML: 3

15 [Container] 2025/06/20 17:23:17.223689 BUILD: 2 commands

16 [Container] 2025/06/20 17:23:17.223694 POST\_BUILD: 4 commands

17 [Container] 2025/06/20 17:23:17.223696 PRE\_BUILD: 7 commands

18 [Container] 2025/06/20 17:23:17.224021 Phase complete: DOWNLOAD\_SOURCE State: SUCCEEDED

19 [Container] 2025/06/20 17:23:17.224034 Phase context status code: Message:

20 [Container] 2025/06/20 17:23:17.396068 Entering phase INSTALL

21 [Container] 2025/06/20 17:23:17.527662 Phase complete: INSTALL State: SUCCEEDED

22 [Container] 2025/06/20 17:23:17.527683 Phase context status code: Message:

23 [Container] 2025/06/20 17:23:17.564582 Entering phase PRE\_BUILD

24 [Container] 2025/06/20 17:23:17.699017 Running command echo Logging in to DockerHub...

25 Logging in to DockerHub...

26

27 [Container] 2025/06/20 17:23:17.706656 Running command echo $DOCKERHUB\_PASSWORD | docker login -u $DOCKERHUB\_USERNAME --password-stdin

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

29 Configure a credential helper to remove this warning. See

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

31

32 Login Succeeded

33

34 [Container] 2025/06/20 17:23:17.802656 Running command echo Logging in to Amazon ECR...

35 Logging in to Amazon ECR...

36

37 [Container] 2025/06/20 17:23:17.809964 Running command aws --version

38 aws-cli/2.27.35 Python/3.13.3 Linux/4.14.355-275.572.amzn2.x86\_64 exec-env/AWS\_ECS\_EC2 exe/x86\_64.amzn.2023

39

40 [Container] 2025/06/20 17:23:26.340949 Running command aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 536697223131.dkr.ecr.us-east-1.amazonaws.com

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

42 Configure a credential helper to remove this warning. See

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

44

45 Login Succeeded

46

47 [Container] 2025/06/20 17:23:27.087891 Running command export REPOSITORY\_URL=536697223131.dkr.ecr.us-east-1.amazonaws.com/brain-tasks-app

48

49 [Container] 2025/06/20 17:23:27.095609 Running command export IMAGE\_TAG=v1

50

51 [Container] 2025/06/20 17:23:27.102639 Phase complete: PRE\_BUILD State: SUCCEEDED

52 [Container] 2025/06/20 17:23:27.102652 Phase context status code: Message:

53 [Container] 2025/06/20 17:23:27.139193 Entering phase BUILD

54 [Container] 2025/06/20 17:23:27.140127 Running command echo Building the Docker image...

55 Building the Docker image...

56

57 [Container] 2025/06/20 17:23:27.147674 Running command docker build -t $REPOSITORY\_URL:$IMAGE\_TAG .

58 #0 building with "default" instance using docker driver

59

60 #1 [internal] load build definition from Dockerfile

61 #1 transferring dockerfile: 176B done

62 #1 DONE 0.0s

63

64 #2 [auth] library/nginx:pull token for registry-1.docker.io

65 #2 DONE 0.0s

66

67 #3 [internal] load metadata for docker.io/library/nginx:stable-alpine

68 #3 DONE 0.3s

69

70 #4 [internal] load .dockerignore

71 #4 transferring context: 2B done

72 #4 DONE 0.0s

73

74 #5 [internal] load build context

75 #5 transferring context: 317.98kB done

76 #5 DONE 0.0s

77

78 #6 [1/3] FROM docker.io/library/nginx:stable-alpine@sha256:aed99734248e851764f1f2146835ecad42b5f994081fa6631cc5d79240891ec9

79 #6 resolve docker.io/library/nginx:stable-alpine@sha256:aed99734248e851764f1f2146835ecad42b5f994081fa6631cc5d79240891ec9 0.0s done

80 #6 sha256:aed99734248e851764f1f2146835ecad42b5f994081fa6631cc5d79240891ec9 10.32kB / 10.32kB done

81 #6 sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB 0.1s

82 #6 sha256:532a231453496d98bd24f9d17442565e55a30c355843a7106ca30d455828ab0c 628B / 628B 0.0s done

83 #6 sha256:bbdf43ac95df903e0e0aba895e4fde36b9994a02e21e6f9ee01a90b874330b2b 955B / 955B 0.1s done

84 #6 sha256:c544760e65143e93084253d9559bde21e5b5ed47cec5db7ffce3cf52c6c1ea4d 0B / 404B 0.1s

85 #6 sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 3.64MB / 3.64MB 0.1s done

86 #6 sha256:c544760e65143e93084253d9559bde21e5b5ed47cec5db7ffce3cf52c6c1ea4d 404B / 404B 0.1s done

87 #6 sha256:37075895d8461222f53afa7804aec2c57d69f9842995705cc54a0c4a70d68fc9 2.49kB / 2.49kB done

88 #6 sha256:936a1208f403bdd270e7f1ef76a0d92cd48fd343ccc474a9cf13bed3b5199a46 10.77kB / 10.77kB done

89 #6 sha256:dde65f7af9d6bd24c25436572f04a4590e1d24cc3261e490b4dc04f8ddc95f55 1.79MB / 1.79MB 0.1s done

90 #6 extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.1s

91 #6 sha256:19f9a172b25c0bb4945785ec1043c39f16f4af6efc5f40b0000ef423c84c1a1d 1.40kB / 1.40kB 0.1s done

92 #6 sha256:e57f9eaa9ba5fcc77182b2bac87e8ff12915363b7f083eb096744ac3c762a138 4.19MB / 15.52MB 0.2s

93 #6 sha256:3cc2eb67bce91b1eacb09e48d14caba02f056ba09772b34f083b54b7b369cc8c 1.21kB / 1.21kB 0.1s done

94 #6 extracting sha256:f18232174bc91741fdf3da96d85011092101a032a93a388b79e99e69c2d5c870 0.1s done

95 #6 sha256:e57f9eaa9ba5fcc77182b2bac87e8ff12915363b7f083eb096744ac3c762a138 15.52MB / 15.52MB 0.3s done

96 #6 extracting sha256:dde65f7af9d6bd24c25436572f04a4590e1d24cc3261e490b4dc04f8ddc95f55 0.1s done

97 #6 extracting sha256:532a231453496d98bd24f9d17442565e55a30c355843a7106ca30d455828ab0c done

98 #6 extracting sha256:bbdf43ac95df903e0e0aba895e4fde36b9994a02e21e6f9ee01a90b874330b2b done

99 #6 extracting sha256:c544760e65143e93084253d9559bde21e5b5ed47cec5db7ffce3cf52c6c1ea4d

100 #6 extracting sha256:c544760e65143e93084253d9559bde21e5b5ed47cec5db7ffce3cf52c6c1ea4d done

101 #6 extracting sha256:3cc2eb67bce91b1eacb09e48d14caba02f056ba09772b34f083b54b7b369cc8c done

102 #6 extracting sha256:19f9a172b25c0bb4945785ec1043c39f16f4af6efc5f40b0000ef423c84c1a1d done

103 #6 extracting sha256:e57f9eaa9ba5fcc77182b2bac87e8ff12915363b7f083eb096744ac3c762a138

104 #6 extracting sha256:e57f9eaa9ba5fcc77182b2bac87e8ff12915363b7f083eb096744ac3c762a138 0.3s done

105 #6 DONE 1.2s

106

107 #7 [2/3] RUN rm -rf /usr/share/nginx/html/\*

108 #7 DONE 0.5s

109

110 #8 [3/3] COPY dist /usr/share/nginx/html

111 #8 DONE 0.1s

112

113 #9 exporting to image

114 #9 exporting layers 0.0s done

115 #9 writing image sha256:3c9914470215f497962a744e7bb91d7eb50b71ffa1793a7e9e9c6146177ad75c done

116 #9 naming to 536697223131.dkr.ecr.us-east-1.amazonaws.com/brain-tasks-app:v1 done

117 #9 DONE 0.1s

118

119 [Container] 2025/06/20 17:23:29.532474 Phase complete: BUILD State: SUCCEEDED

120 [Container] 2025/06/20 17:23:29.532492 Phase context status code: Message:

121 [Container] 2025/06/20 17:23:29.569556 Entering phase POST\_BUILD

122 [Container] 2025/06/20 17:23:29.570630 Running command echo Pushing the Docker image...

123 Pushing the Docker image...

124

125 [Container] 2025/06/20 17:23:29.579017 Running command docker push $REPOSITORY\_URL:$IMAGE\_TAG

126 The push refers to repository [536697223131.dkr.ecr.us-east-1.amazonaws.com/brain-tasks-app]

127 2cbde43240e9: Preparing

128 1b4148c919e6: Preparing

129 7e881413b635: Preparing

130 3db4ef8dec24: Preparing

131 2bfd71b953a3: Preparing

132 b0debeaa68c5: Preparing

133 5f5a3d906b11: Preparing

134 9c2b6e6f2e2e: Preparing

135 4babb02c7c40: Preparing

136 08000c18d16d: Preparing

137 5f5a3d906b11: Waiting

138 9c2b6e6f2e2e: Waiting

139 4babb02c7c40: Waiting

140 08000c18d16d: Waiting

141 b0debeaa68c5: Waiting

142 7e881413b635: Layer already exists

143 3db4ef8dec24: Layer already exists

144 2bfd71b953a3: Layer already exists

145 b0debeaa68c5: Layer already exists

146 5f5a3d906b11: Layer already exists

147 9c2b6e6f2e2e: Layer already exists

148 4babb02c7c40: Layer already exists

149 08000c18d16d: Layer already exists

150 2cbde43240e9: Pushed

151 1b4148c919e6: Pushed

152 v1: digest: sha256:bc24edbe96efb4571afc723c74d022fd59c20232acbd4fc64a9327cbac87d4fb size: 2406

153

154 [Container] 2025/06/20 17:23:31.155594 Running command echo Deploying to EKS...

155 Deploying to EKS...

156

157 [Container] 2025/06/20 17:23:31.163136 Running command aws eks update-kubeconfig --region us-east-1 --name brain-tasks-cluster

158 Added new context arn:aws:eks:us-east-1:536697223131:cluster/brain-tasks-cluster to /root/.kube/config

159

160 [Container] 2025/06/20 17:23:31.924103 Phase complete: POST\_BUILD State: SUCCEEDED

161 [Container] 2025/06/20 17:23:31.924119 Phase context status code: Message:

162 [Container] 2025/06/20 17:23:32.121984 Expanding base directory path: .

163 [Container] 2025/06/20 17:23:32.126292 Assembling file list

164 [Container] 2025/06/20 17:23:32.126309 Expanding .

165 [Container] 2025/06/20 17:23:32.130113 Expanding file paths for base directory .

166 [Container] 2025/06/20 17:23:32.130129 Assembling file list

167 [Container] 2025/06/20 17:23:32.130132 Expanding appspec.yml

168 [Container] 2025/06/20 17:23:32.133213 Expanding deployment.yml

169 [Container] 2025/06/20 17:23:32.136346 Expanding service.yml

170 [Container] 2025/06/20 17:23:32.139298 Found 3 file(s)

171 [Container] 2025/06/20 17:23:32.140655 Set report auto-discover timeout to 5 seconds

172 [Container] 2025/06/20 17:23:32.140698 Expanding base directory path: .

173 [Container] 2025/06/20 17:23:32.143684 Assembling file list

174 [Container] 2025/06/20 17:23:32.143694 Expanding .

175 [Container] 2025/06/20 17:23:32.146770 Expanding file paths for base directory .

176 [Container] 2025/06/20 17:23:32.146782 Assembling file list

177 [Container] 2025/06/20 17:23:32.146785 Expanding \*\*/\*

178 [Container] 2025/06/20 17:23:32.149915 No matching auto-discover report paths found

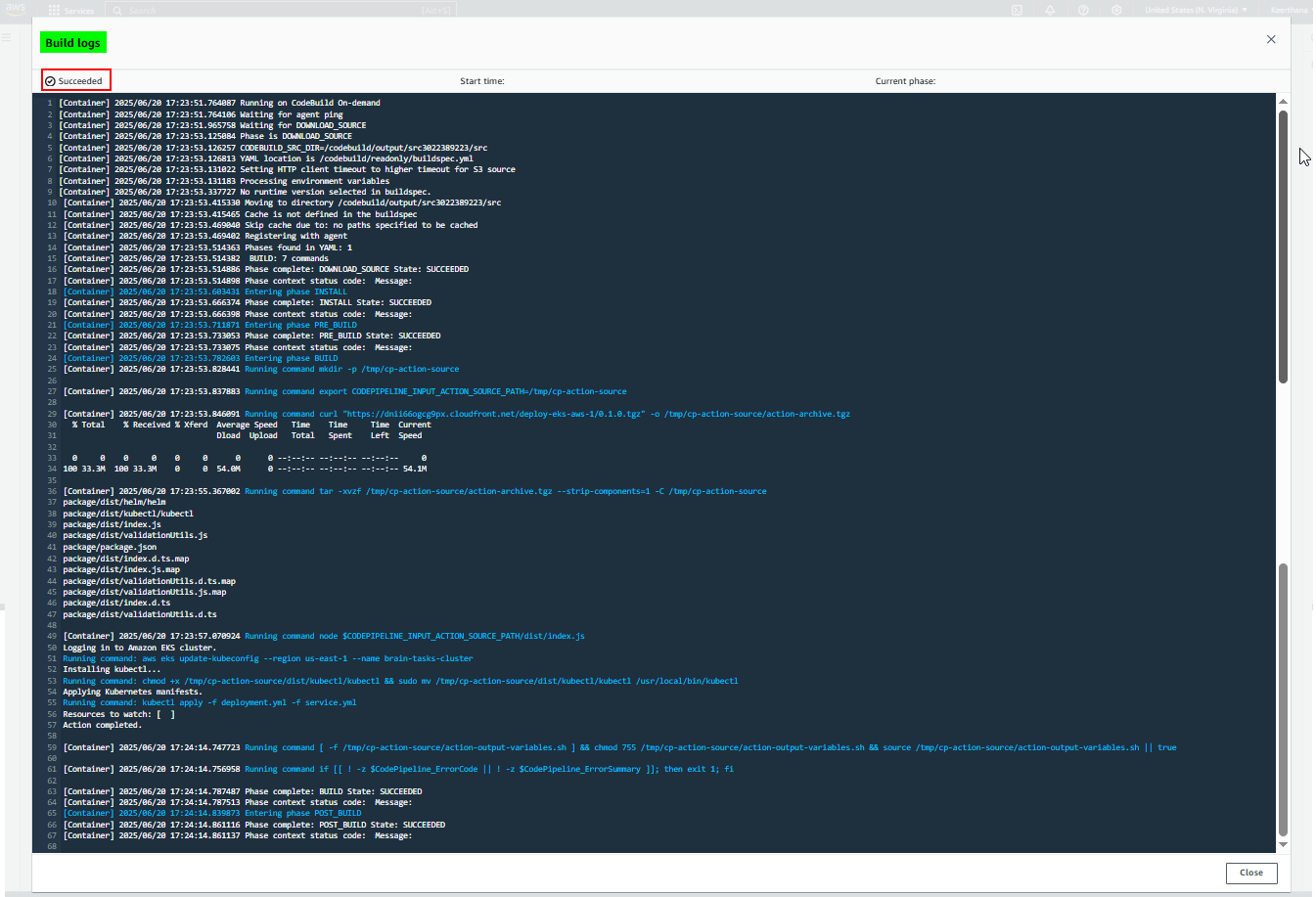
179 [Container] 2025/06/20 17:23:32.149934 Report auto-discover file discovery took 0.009279 seconds

180 [Container] 2025/06/20 17:23:32.149943 Phase complete: UPLOAD\_ARTIFACTS State: SUCCEEDED

181 [Container] 2025/06/20 17:23:32.149951 Phase context status code: Message:

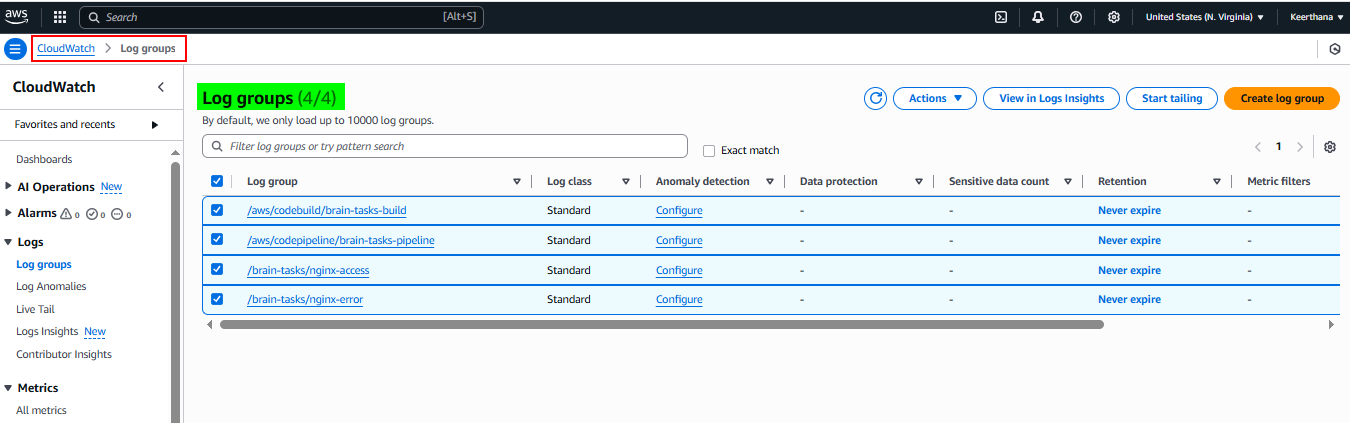
182

**Deploy console logs**



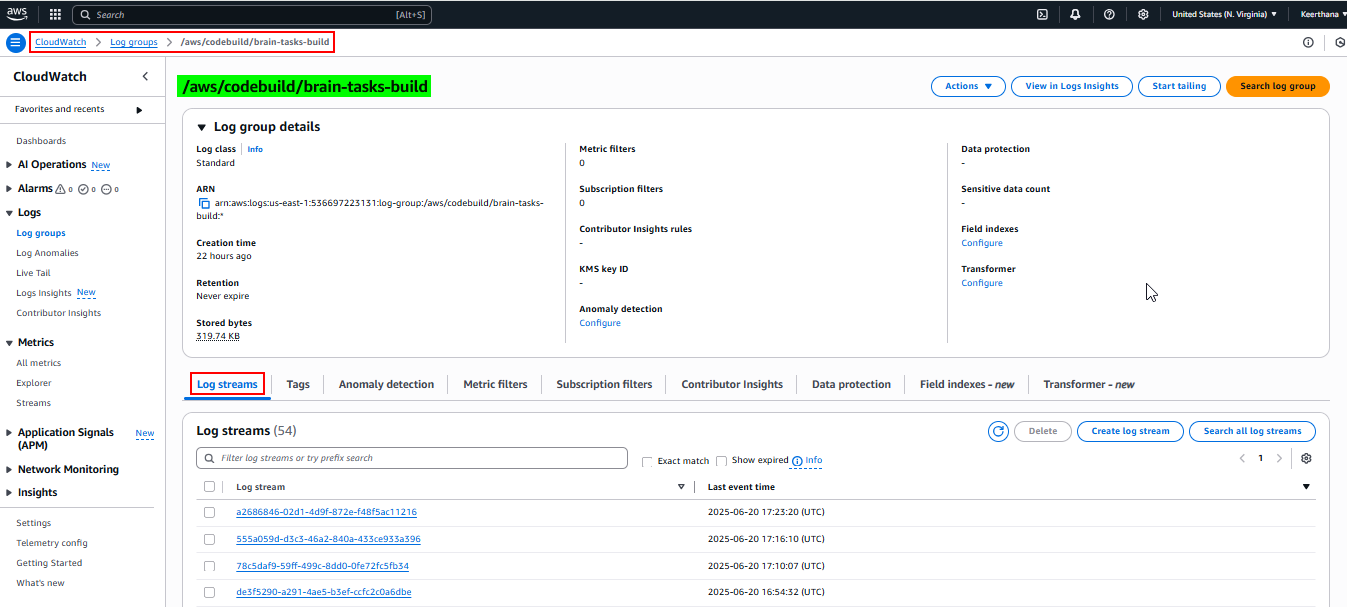
**Cloud Watch Monitoring**

* Log groups are created for the build, deploy and EKS

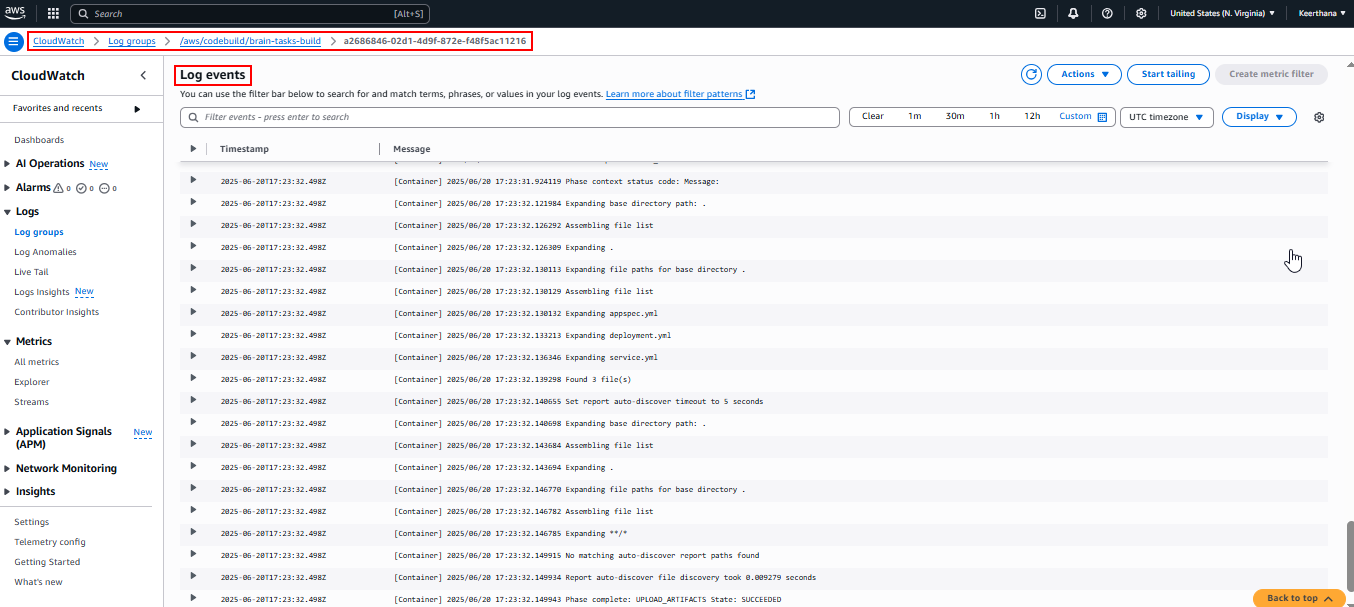


**Build logs:**

**Log streams – build**

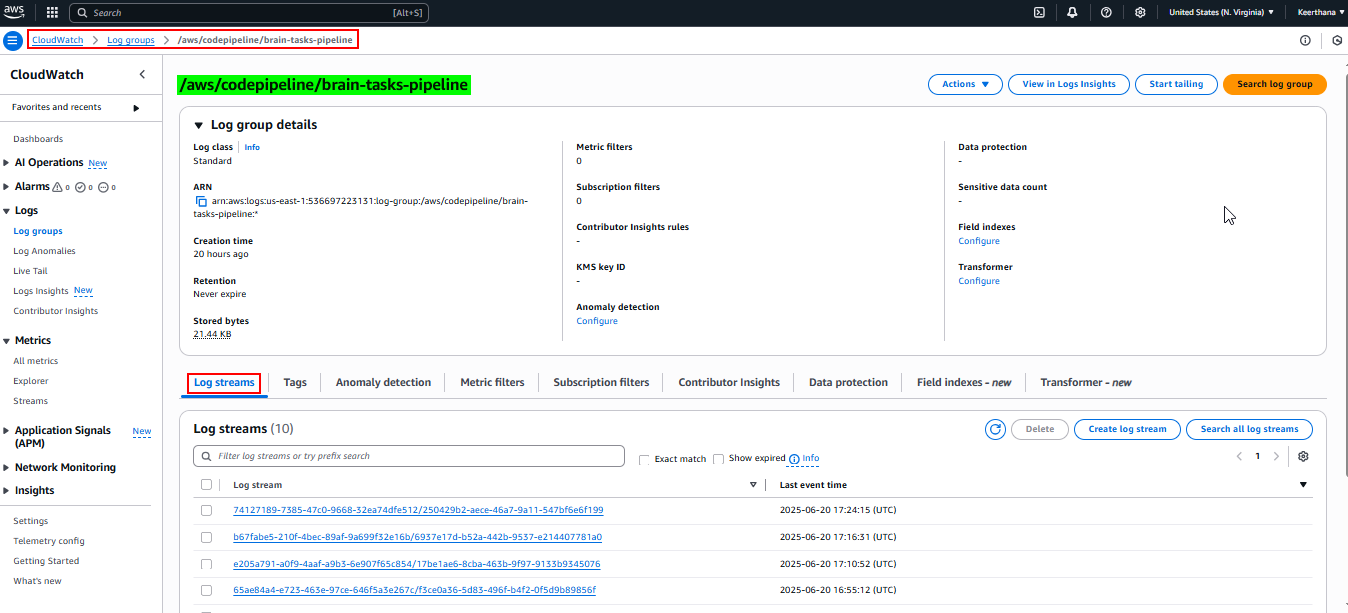


**Log events – build**

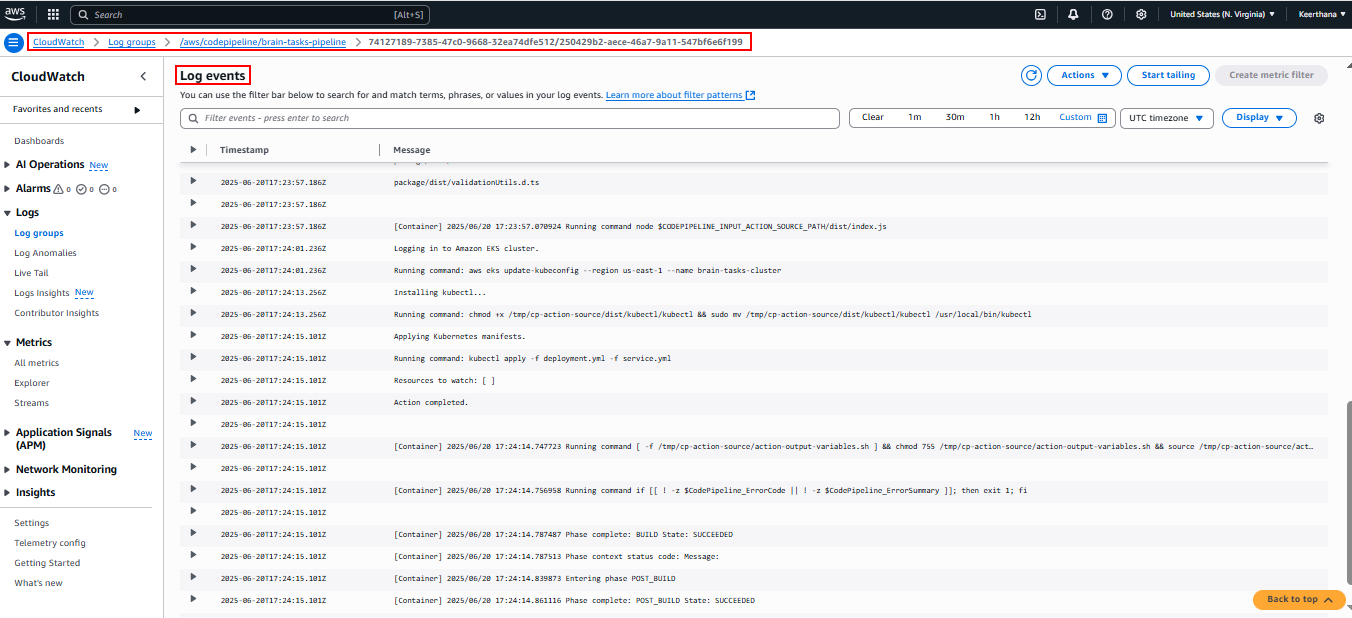


**Deploy logs:**

**Log streams – deploy**

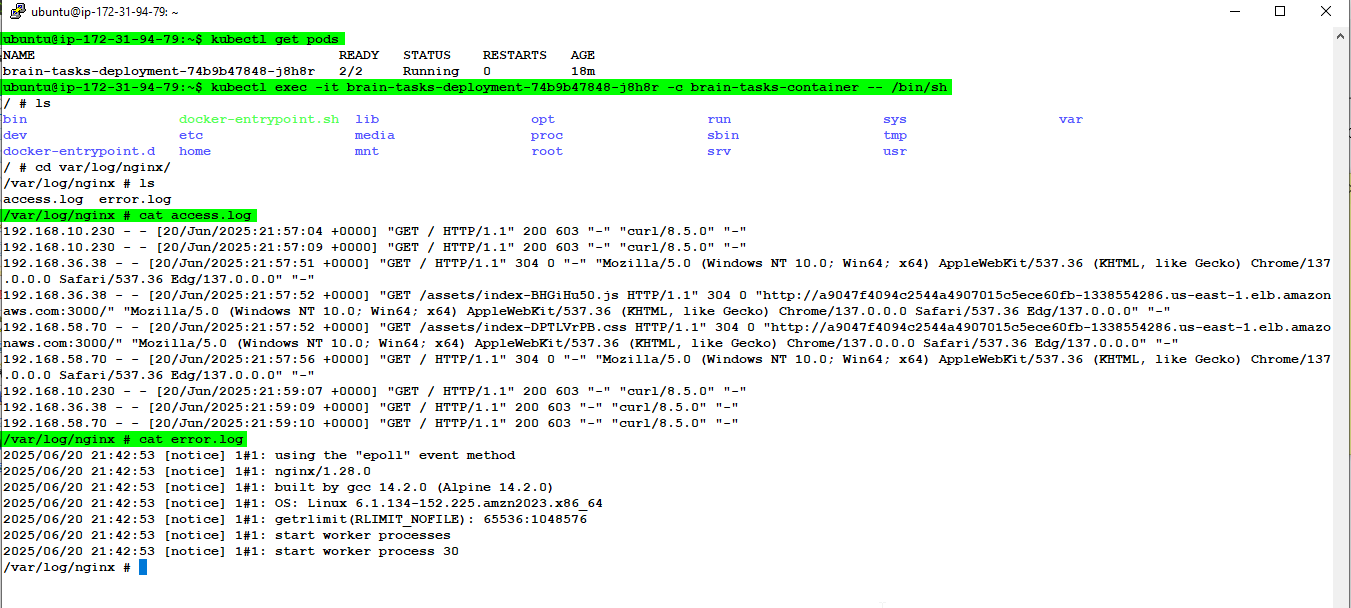


**Log events – deploy**

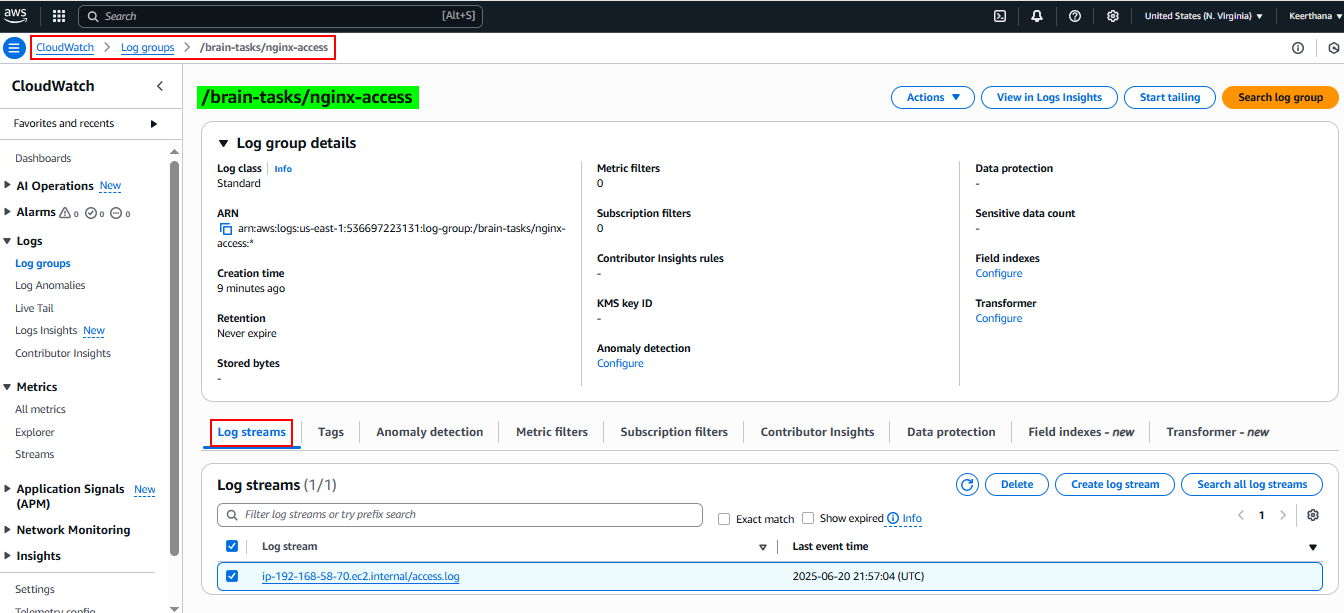


**Application logs inside container:**

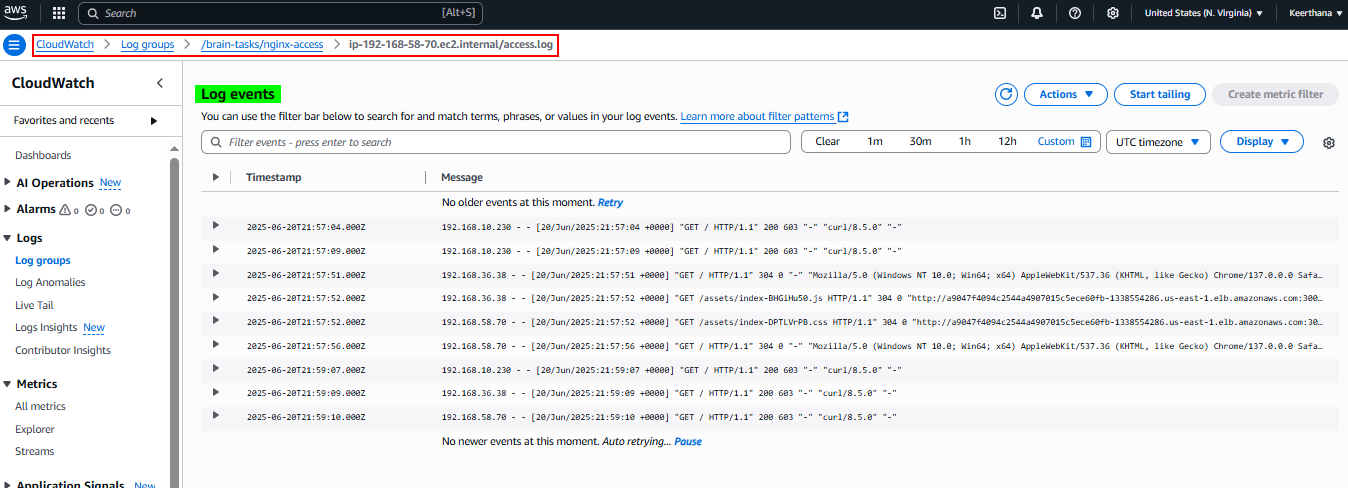
**Access.log and error.log**



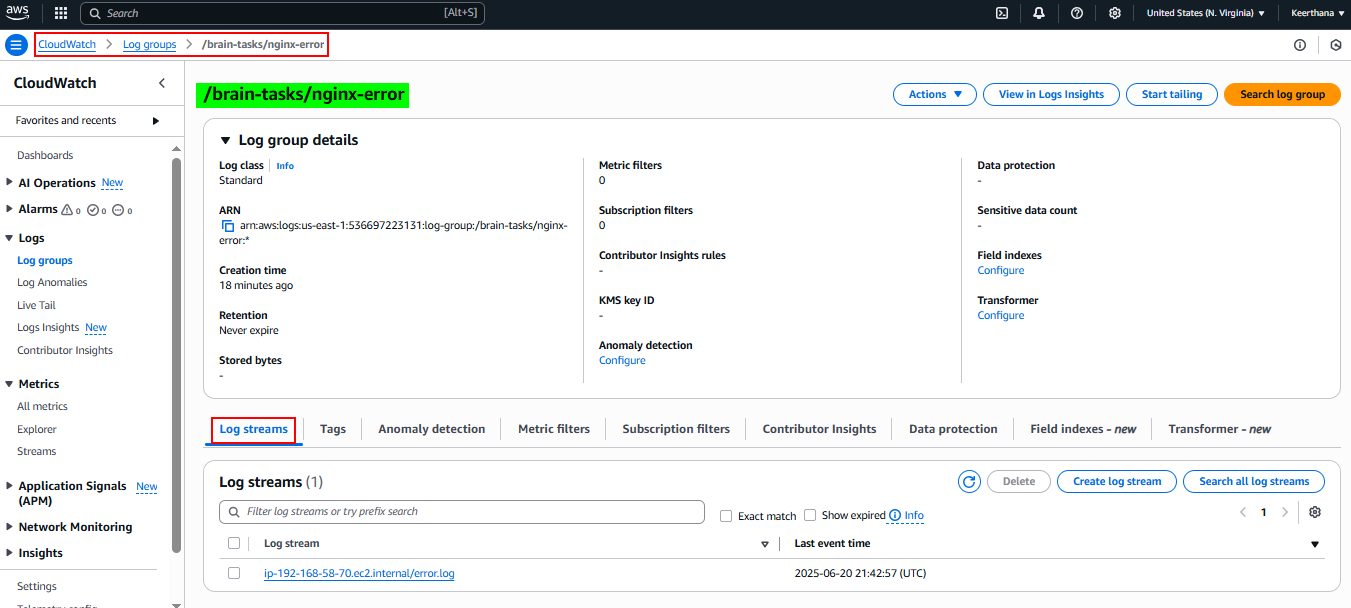
**Log streams – Application (access)**



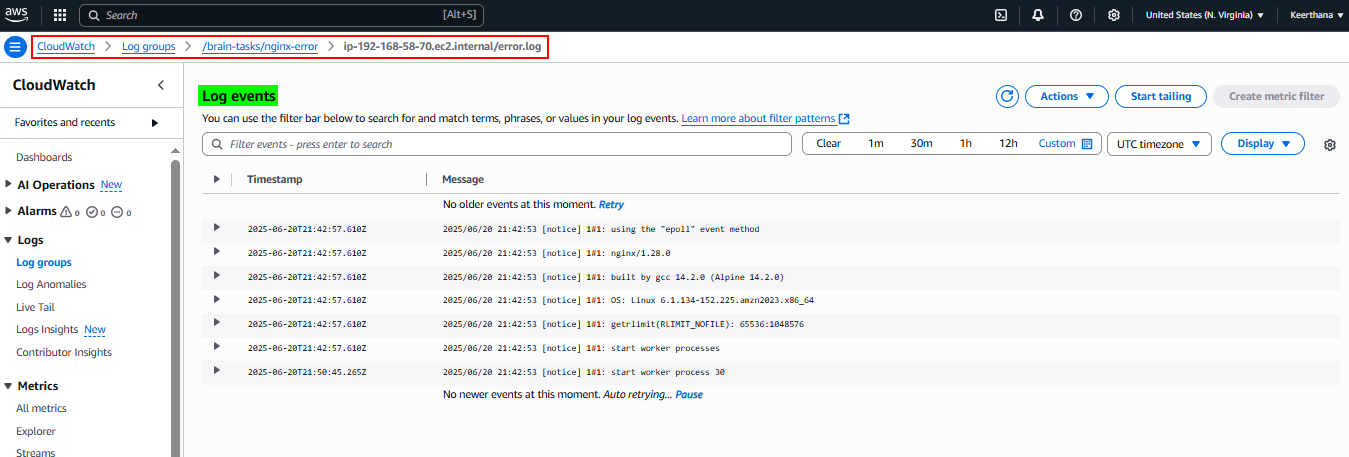
**Log events – Application (access)**



**Log streams – Application (error)**



**Log events – Application (error)**



**Cloud Watch Dashboard**

