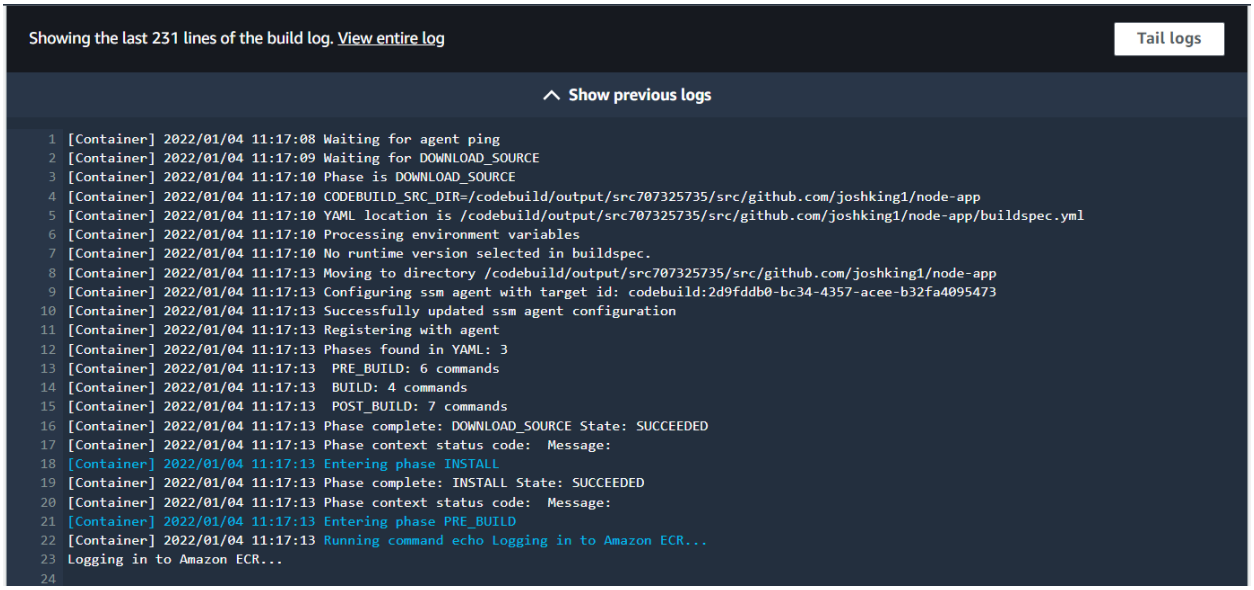
AWS CodeBuild | AWS ECR | AWS CoudeBuild Push Docker Image to ECR | AWS CodeBuild Example

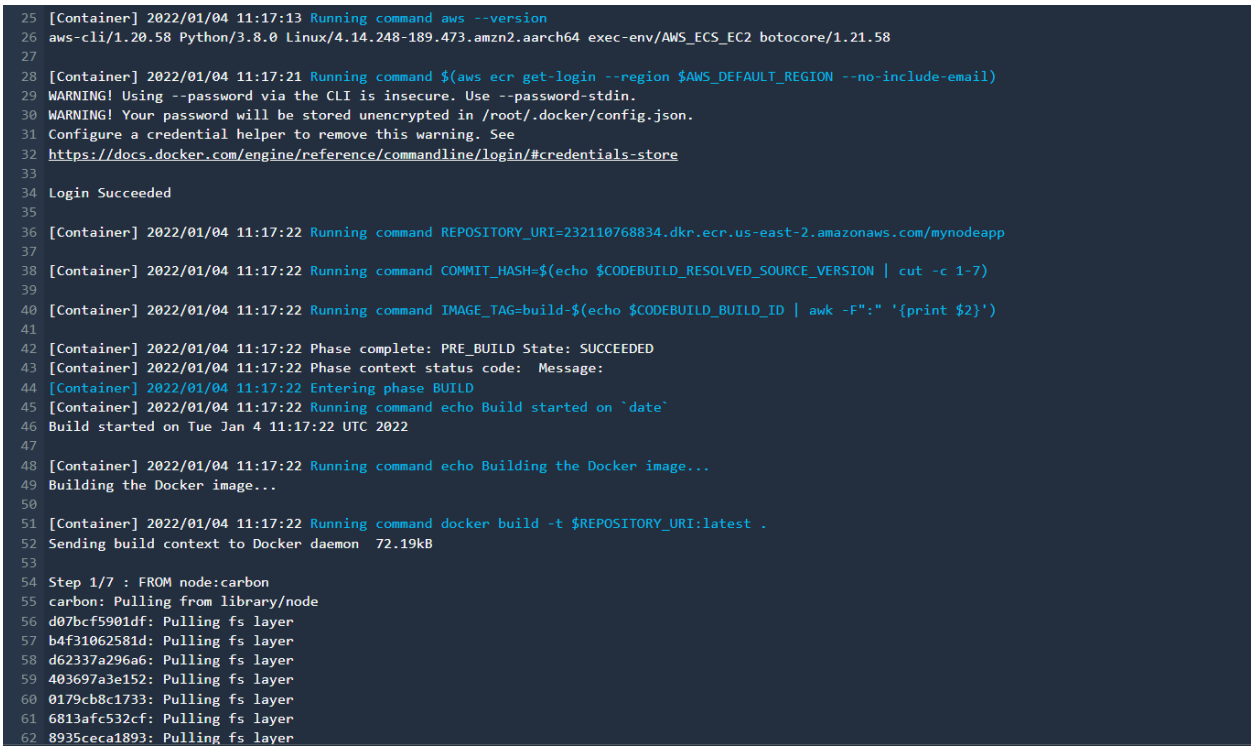
AWS CodeBuild is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers.

Repository to Clone

https://github.com/joshking1/node-app.git

- Create an Elastic Container Registry on AWS
- Allow the code build to automatically assign name of the IAM role
- Go to the IAM, select ROLE and assign the created IAM role by code build, full access to the amazonecrfullaccess
- Create Personal Access Token (PAT) in the github and assign it to the codebuild
- Select the operating system where the docker is installed
- Enable this flag if you want to build Docker images or want your builds to get elevated privileges
- Add the details for logs
- Start the build
- Images from my build


Showing the last 231 lines of the build log. View entire log
Tail logs
Show previous logs
1 [Container] 2022/01/04 11:17:08 Waiting for agent ping
2 [Container] 2022/01/04 11:17:09 Waiting for DOWNLOAD_SOURCE
3 [Container] 2022/01/04 11:17:10 Phase is DOWNLOAD_SOURCE
4 [Container] 2022/01/04 11:17:10 CODEBUILD_SRC_DIR=/codebuild/output/src707325735/src/github.com/joshking1/node-app
5 [Container] 2022/01/04 11:17:10 YAML location is /codebuild/output/src707325735/src/github.com/joshking1/node-app/buildspec.yml
6 [Container] 2022/01/04 11:17:10 Processing environment variables
7 [Container] 2022/01/04 11:17:10 No runtime version selected in buildspec.
8 [Container] 2022/01/04 11:17:13 Moving to directory /codebuild/output/src707325735/src/github.com/joshking1/node-app
9 [Container] 2022/01/04 11:17:13 Configuring ssm agent with target id: codebuild:2d9fddb0-bc34-4357-acee-b32fa4095473
10 [Container] 2022/01/04 11:17:13 Successfully updated ssm agent configuration
11 [Container] 2022/01/04 11:17:13 Registering with agent
12 [Container] 2022/01/04 11:17:13 Phases found in YAML: 3
13 [Container] 2022/01/04 11:17:13 PRE_BUILD: 6 commands
14 [Container] 2022/01/04 11:17:13 BUILD: 4 commands
15 [Container] 2022/01/04 11:17:13 POST_BUILD: 7 commands
16 [Container] 2022/01/04 11:17:13 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
17 [Container] 2022/01/04 11:17:13 Phase context status code: Message:
18 [Container] 2022/01/04 11:17:13 Entering phase INSTALL
19 [Container] 2022/01/04 11:17:13 Phase complete: INSTALL State: SUCCEEDED
20 [Container] 2022/01/04 11:17:13 Phase context status code: Message:
21 [Container] 2022/01/04 11:17:13 Entering phase PRE_BUILD
22 [Container] 2022/01/04 11:17:13 Running command echo Logging in to Amazon ECR...
23 Logging in to Amazon ECR...
24


25 [Container] 2022/01/04 11:17:13 Running command aws --version
26 aws-cli/1.20.58 Python/3.8.0 Linux/4.14.248-189.473.amzn2.aarch64 exec-env/AWS_ECS_EC2 botocore/1.21.58
27
28 [Container] 2022/01/04 11:17:21 Running command $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
29 WARNING! Using --password via the CLI is insecure. Use --password-stdin.
30 WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
31 Configure a credential helper to remove this warning. See
32 https://docs.docker.com/engine/reference/commandline/login/#credentials-store
33
34 Login Succeeded
35
36 [Container] 2022/01/04 11:17:22 Running command REPOSITORY_URI=232110768834.dkr.ecr.us-east-2.amazonaws.com/mynodeapp
37
38 [Container] 2022/01/04 11:17:22 Running command COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
39
40 [Container] 2022/01/04 11:17:22 Running command IMAGE_TAG=build-$(echo $CODEBUILD_BUILD_ID | awk -F":" '{print $2}')
41
42 [Container] 2022/01/04 11:17:22 Phase complete: PRE_BUILD State: SUCCEEDED
43 [Container] 2022/01/04 11:17:22 Phase context status code: Message:
44 [Container] 2022/01/04 11:17:22 Entering phase BUILD
45 [Container] 2022/01/04 11:17:22 Running command echo Build started on `date`
46 Build started on Tue Jan 4 11:17:22 UTC 2022
47
48 [Container] 2022/01/04 11:17:22 Running command echo Building the Docker image...
49 Building the Docker image...
50
51 [Container] 2022/01/04 11:17:22 Running command docker build -t $REPOSITORY_URI:latest .
52 Sending build context to Docker daemon 72.19kB
53
54 Step 1/7 : FROM node:carbon
55 carbon: Pulling from library/node
56 d07bcf5901df: Pulling fs layer
57 b4f31062581d: Pulling fs layer
58 d62337a296a6: Pulling fs layer
59 403697a3e152: Pulling fs layer
60 0179cb8c1733: Pulling fs layer
61 6813afc532cf: Pulling fs layer
62 8935ceca1893: Pulling fs layer

```
0179cb8c1733: Pulling fs layer
6813afc532cf: Pulling fs layer
8935ceca1893: Pulling fs layer
dfbca36bef87: Pulling fs layer
5195d0adf884: Pulling fs layer
403697a3e152: Waiting
0179cb8c1733: Waiting
6813afc532cf: Waiting
8935ceca1893: Waiting
dfbca36bef87: Waiting
5195d0adf884: Waiting
d62337a296a6: Verifying Checksum
d62337a296a6: Download complete
b4f31062581d: Verifying Checksum
b4f31062581d: Download complete
d07bcf5901df: Verifying Checksum
d07bcf5901df: Download complete
6813afc532cf: Verifying Checksum
6813afc532cf: Download complete
403697a3e152: Verifying Checksum
403697a3e152: Download complete
dfbca36bef87: Verifying Checksum
dfbca36bef87: Download complete
8935ceca1893: Verifying Checksum
8935ceca1893: Download complete
5195d0adf884: Verifying Checksum
5195d0adf884: Download complete
0179cb8c1733: Verifying Checksum
0179cb8c1733: Download complete
d07bcf5901df: Pull complete
b4f31062581d: Pull complete
d62337a296a6: Pull complete
403697a3e152: Pull complete
0179cb8c1733: Pull complete
6813afc532cf: Pull complete
8935ceca1893: Pull complete
dfbca36bef87: Pull complete
```

```
Digest: sha256:a681bf74805b80d03eb21a6c0ef168a976108a287a74167ab593fc953aac34df
Status: Downloaded newer image for node:carbon
 ---> 50c57a9369c7
Step 2/7 : WORKDIR /usr/src/app
 ---> Running in 741e3fdcb8dd
Removing intermediate container 741e3fdcb8dd
 ---> 6116c96971e0
Step 3/7 : COPY package*.json ./
 ---> 8f8d6aa6e1c2
Step 4/7 : RUN npm install
 ---> Running in dce6fa7bc068
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN docker_web_app@1.0.0 No repository field.
npm WARN docker_web_app@1.0.0 No license field.

added 50 packages from 37 contributors and audited 50 packages in 1.62s

1 package is looking for funding
  run `npm fund` for details

found 0 vulnerabilities

Removing intermediate container dce6fa7bc068
 ---> 935f21809941
Step 5/7 : COPY . .
 ---> 2cb039949400
Step 6/7 : EXPOSE 8080
 ---> Running in 16d7d1efa461
Removing intermediate container 16d7d1efa461
 ---> 41336e5c20a2
Step 7/7 : CMD [ "npm", "start" ]
 ---> Running in 5f8d4911d04a
Removing intermediate container 5f8d4911d04a
 ---> a5e607377b80
Successfully built a5e607377b80
Successfully tagged 232110768834.dkr.ecr.us-east-2.amazonaws.com/mynodeapp:latest
```

```
[Container] 2022/01/04 11:17:47 Running command docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG

[Container] 2022/01/04 11:17:48 Phase complete: BUILD State: SUCCEEDED
[Container] 2022/01/04 11:17:48 Phase context status code:  Message:
[Container] 2022/01/04 11:17:48 Entering phase POST_BUILD
[Container] 2022/01/04 11:17:48 Running command echo Build completed on `date`
Build completed on Tue Jan 4 11:17:48 UTC 2022

[Container] 2022/01/04 11:17:48 Running command echo Pushing the Docker images...
Pushing the Docker images...

[Container] 2022/01/04 11:17:48 Running command docker push $REPOSITORY_URI:latest
The push refers to repository [232110768834.dkr.ecr.us-east-2.amazonaws.com/mynodeapp]
26be1c2f610a: Preparing
9c240075a6c2: Preparing
c234806490d4: Preparing
c7238b9bed9b: Preparing
8d57f55b8217: Preparing
4abc4549efbb: Preparing
d4cfcbee98ee: Preparing
2f48a5f89dc0: Preparing
d76a5d76b78d: Preparing
bae4fdae8c7d: Preparing
33b482d3030f: Preparing
870768294883: Preparing
f319319793e0: Preparing
4abc4549efbb: Waiting
d4cfcbee98ee: Waiting
2f48a5f89dc0: Waiting
d76a5d76b78d: Waiting
bae4fdae8c7d: Waiting
33b482d3030f: Waiting
870768294883: Waiting
f319319793e0: Waiting
c234806490d4: Pushed
c7238b9bed9b: Pushed
8d57f55b8217: Pushed
26be1c2f610a: Pushed
```

```
870768294883: Pushed
d4cfcbee98ee: Pushed
f319319793e0: Pushed
bae4fdae8c7d: Pushed
d76a5d76b78d: Pushed
latest: digest: sha256:136e1cdff0bc6f944e9a9251735d23f09ad475caacb9cfb5e1dd7adcde43340c size: 3048

[Container] 2022/01/04 11:18:19 Running command docker push $REPOSITORY_URI:$IMAGE_TAG
The push refers to repository [232110768834.dkr.ecr.us-east-2.amazonaws.com/mynodeapp]
26be1c2f610a: Preparing
9c240075a6c2: Preparing
c234806490d4: Preparing
c7238b9bed9b: Preparing
8d57f55b8217: Preparing
4abc4549efbb: Preparing
d4cfcbee98ee: Preparing
2f48a5f89dc0: Preparing
d76a5d76b78d: Preparing
bae4fdae8c7d: Preparing
33b482d3030f: Preparing
870768294883: Preparing
f319319793e0: Preparing
4abc4549efbb: Waiting
d4cfcbee98ee: Waiting
33b482d3030f: Waiting
2f48a5f89dc0: Waiting
d76a5d76b78d: Waiting
870768294883: Waiting
bae4fdae8c7d: Waiting
f319319793e0: Waiting
c234806490d4: Layer already exists
26be1c2f610a: Layer already exists
c7238b9bed9b: Layer already exists
8d57f55b8217: Layer already exists
9c240075a6c2: Layer already exists
4abc4549efbb: Layer already exists
d76a5d76b78d: Layer already exists
```

```
c234806490d4: Layer already exists
26be1c2f610a: Layer already exists
c7238b9bed9b: Layer already exists
8d57f55b8217: Layer already exists
9c240075a6c2: Layer already exists
4abc4549efbb: Layer already exists
d76a5d76b78d: Layer already exists
2f48a5f89dc0: Layer already exists
d4cfcbee98ee: Layer already exists
bae4fdae8c7d: Layer already exists
870768294883: Layer already exists
f319319793e0: Layer already exists
33b482d3030f: Layer already exists
build-2d9fddb0-bc34-4357-acee-b32fa4095473: digest: sha256:136e1cdff0bc6f944e9a9251735d23f09ad475caacb9cfb5e1dd7adcde43340c size: 3048

[Container] 2022/01/04 11:18:20 Running command echo Writing image definitions file...
Writing image definitions file...

[Container] 2022/01/04 11:18:20 Running command printf '[{"name":"nodeapp","imageUri":"%s"}]' $REPOSITORY_URI:$IMAGE_TAG > imagedefinitions.json

[Container] 2022/01/04 11:18:20 Running command cat imagedefinitions.json
[{"name":"nodeapp","imageUri":"232110768834.dkr.ecr.us-east-2.amazonaws.com/mynodeapp:build-2d9fddb0-bc34-4357-acee-b32fa4095473"}]
[Container] 2022/01/04 11:18:20 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2022/01/04 11:18:20 Phase context status code:  Message:
[Container] 2022/01/04 11:18:20 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
[Container] 2022/01/04 11:18:20 Phase context status code:  Message:

```