CICD-DEPLOY

K V, Vignesh

+91-9962623973
Vigneshkv96@gmail.com

Table of Contents

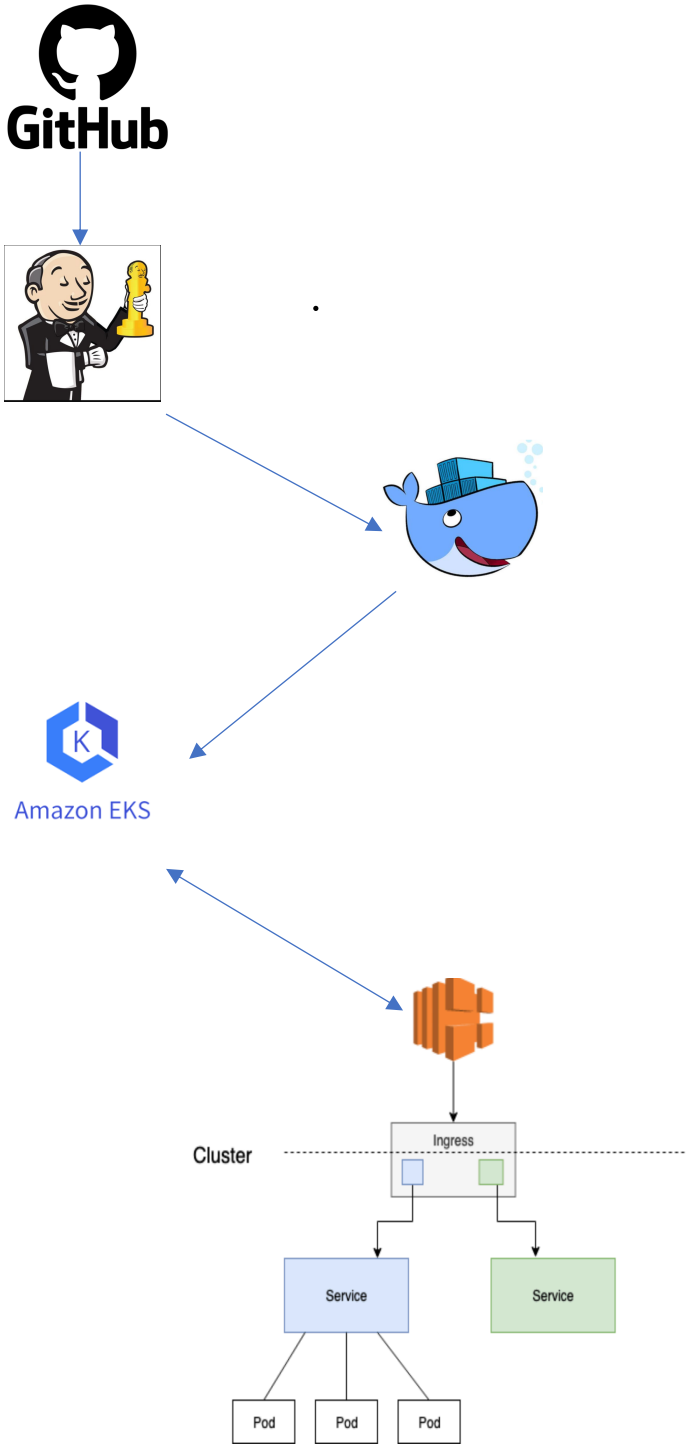
Problem Statement

- You will find here a private Github repository containing a Node.js web application here - https://github.com/devopsruby/cicd-deploy.
- You should have received an invite to the repository that is sent to your email address. Fork the repository to your own Github repository or any other public Git server (Note: You should not make any changes directly in the repository above)
- Create a CI/CD pipeline that tracks the ‘release’ and ‘staging’ branches of the repository using bash scripts or the CI/CD tool that you are most experienced in. The pipeline should be triggered on new commits and perform at a minimum the following:
- The pipeline should build and run tests on the application b. Containerise and deploy the application on a public cloud instance c. The application makes use of a built-in JSON data file, whose filename can be specified in the “DATA_FILE” environment variable. If the environment variable is not found, then it will default to using “Responses.json”
- .There are two branches, “staging” and “release”: i. The “staging” branch should use “Responses-other.json” as its input Datafile ii. The “release” branch should use “Responses.json” as its input datafile e.
- Consider how versioning can be done to differentiate the build f. If any of the tasks fails at any point, the pipeline should be stopped and an email alert should be sent with relevant information.

Tools

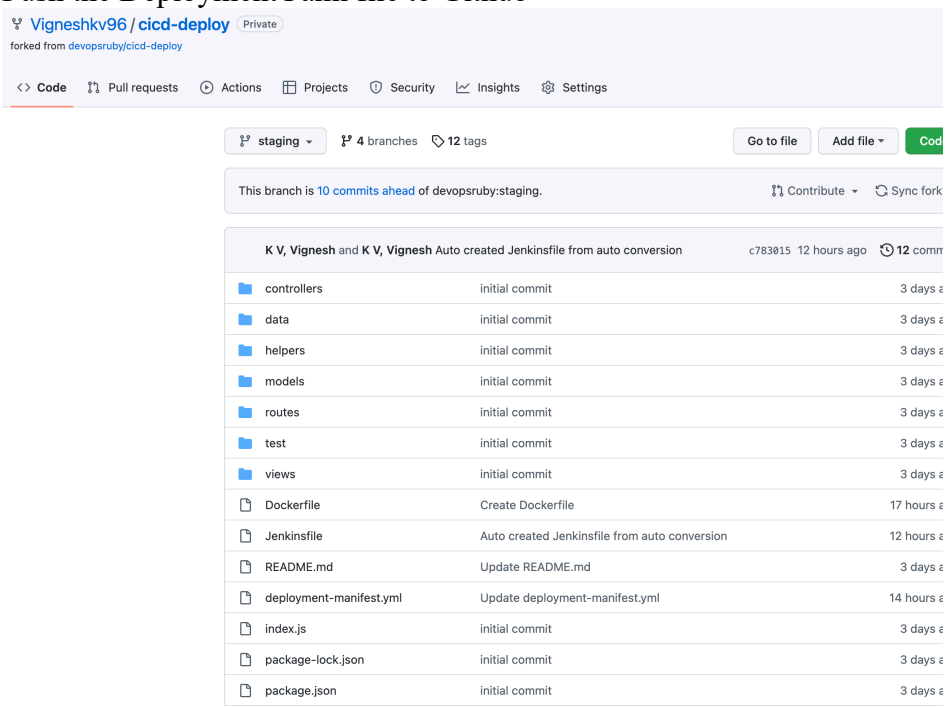
SCM	GitHub
CICD	Jenkins
ContainerRegistry	DockerHub
Deployment	AWS EKS
Build	NPM

Flow Diagram

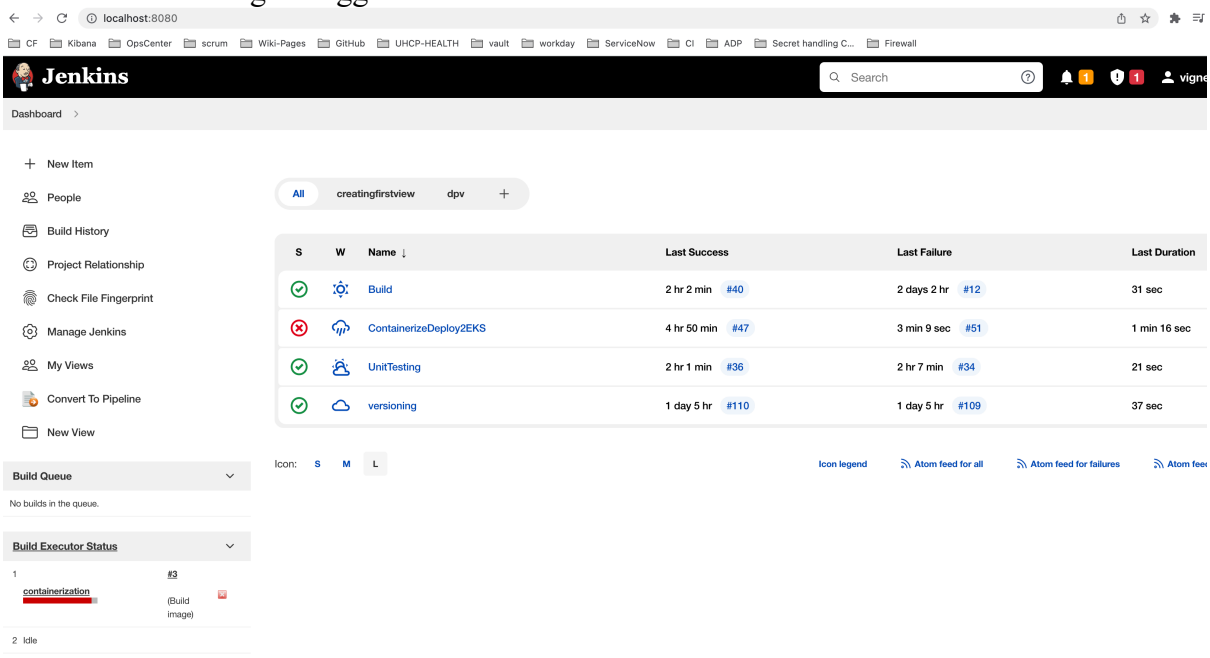


Implementation Screenshots

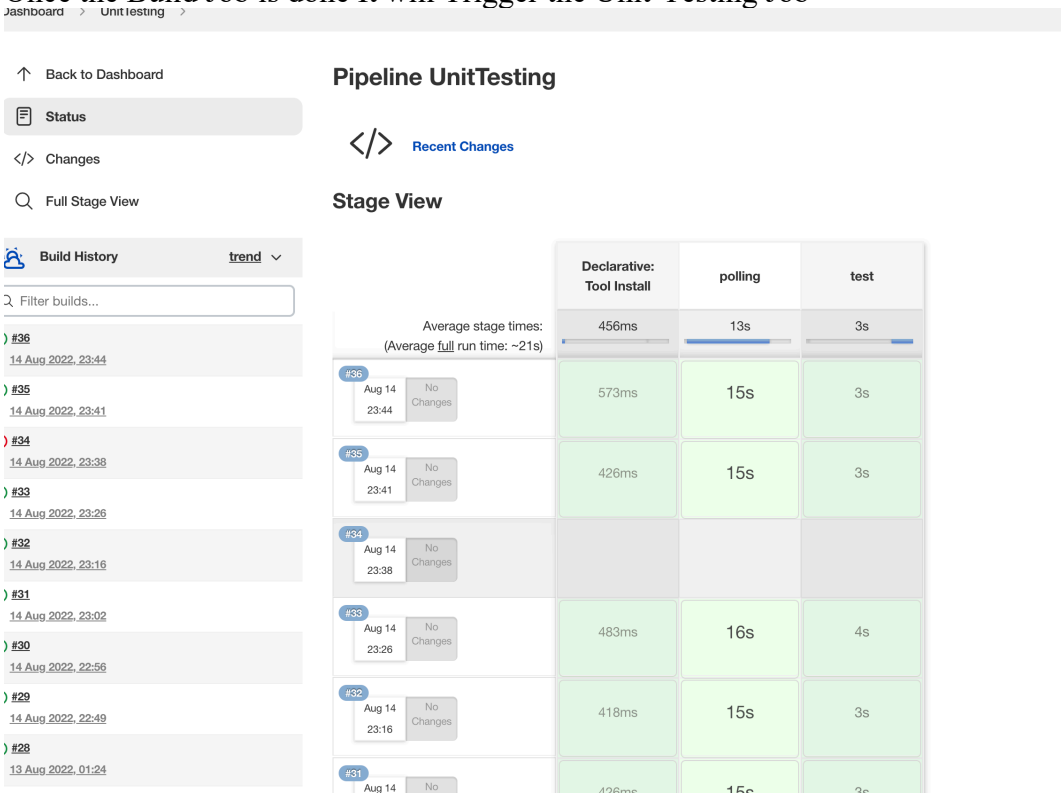
Push the DeploymentYaml file to Github



The Build Job will get Triggered



Once the Build Job is done It will Trigger the Unit-Testing Job



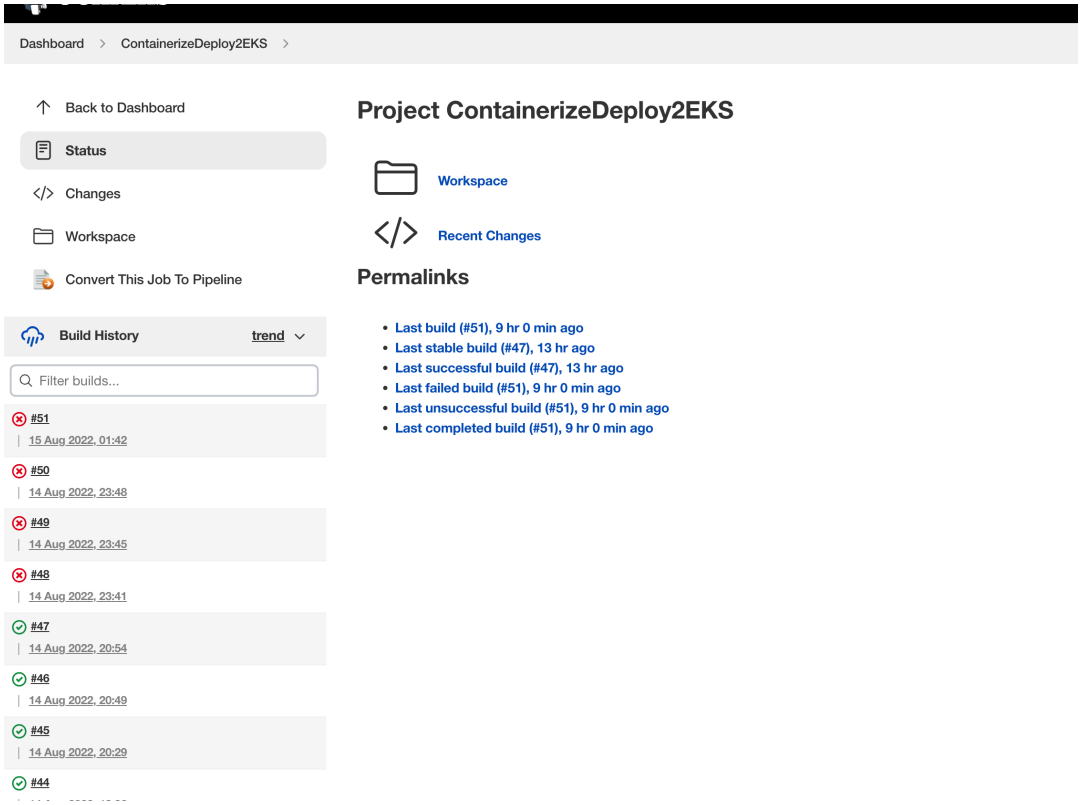
After Build passes the UnitTesting, The containerizeDeploy2EKS will get triggered
The Build wil be converted to docker image considering the below scenario

```
If (Github branch == Stagging )
        Then
          DATA_FILE=Responses-others.json
Else:
        DATA_FILE=Responses.json
```

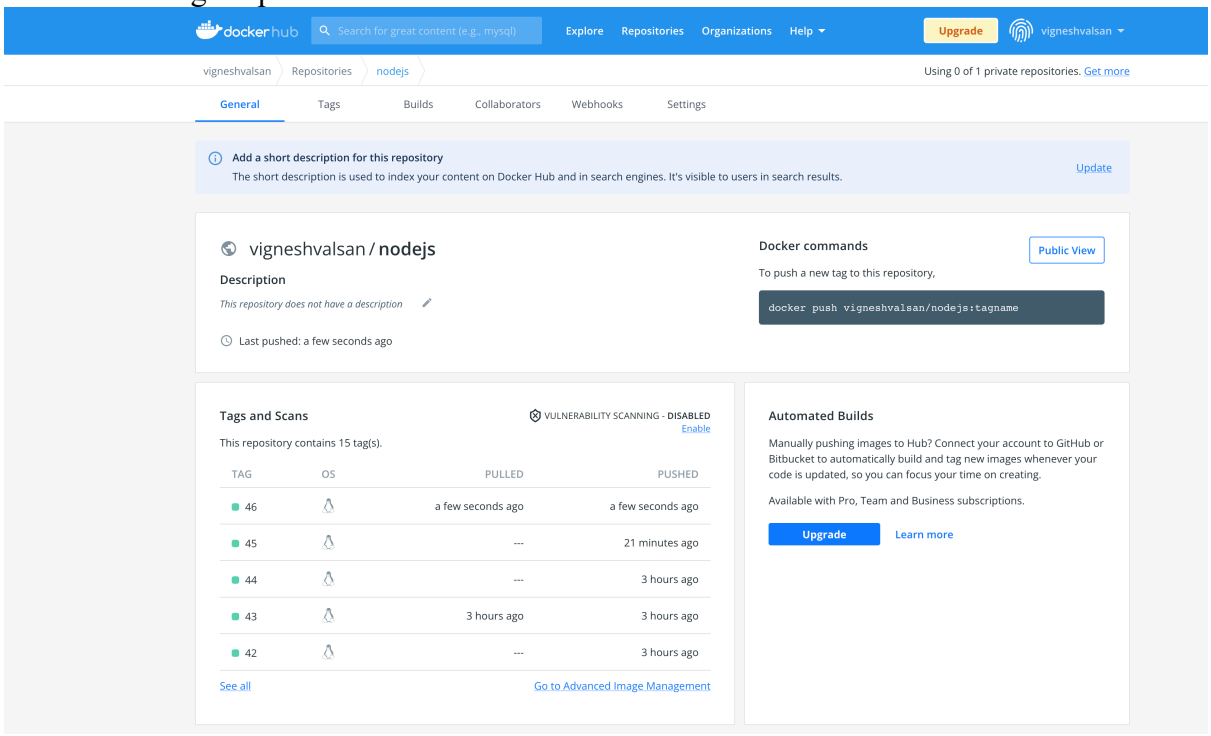
The above condition is set in the environment variable of the image using the below command

```
docker build . -t vigneshvalsan/nodejs:$BUILD_NUMBER_$GITHUBBRANCH --build-arg data=$data
```

where $data could be Responses-others.json / Responses.json Based on github branch



The Build Image is pushed to DockerHub



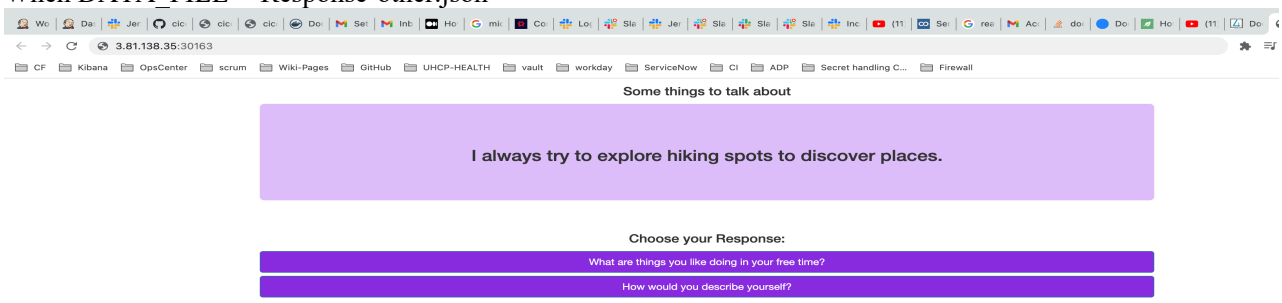
Then the Image is deployed to AWS EKS using kubectl apply command

Dashboard > ContainerizeDeploy2EKS > #47

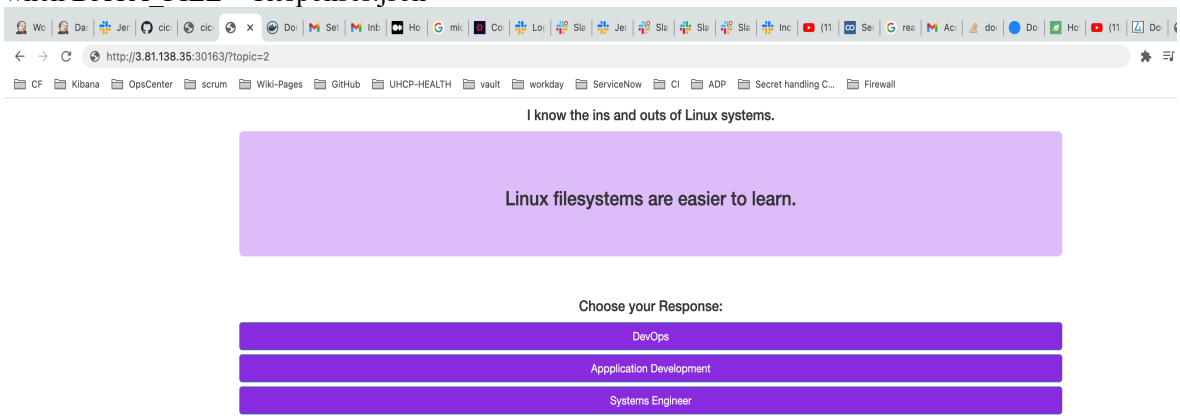
```
a442da27876b: Preparing
08a6d4d7d5f9: Preparing
cbf26b63836e: Preparing
83874e80f7b0: Preparing
12228ba7a3b1: Preparing
9b55156abf26: Preparing
293d5db30c9f: Preparing
03127cdb479b: Preparing
9c742cd6c7a5: Preparing
08a6d4d7d5f9: Waiting
cbf26b63836e: Waiting
83874e80f7b0: Waiting
12228ba7a3b1: Waiting
9b55156abf26: Waiting
293d5db30c9f: Waiting
03127cdb479b: Waiting
9c742cd6c7a5: Waiting
aeaf4a79a0be: Layer already exists
0ff45ac92715: Layer already exists
888f5fc41637: Layer already exists
a442da27876b: Layer already exists
08a6d4d7d5f9: Layer already exists
cbf26b63836e: Layer already exists
83874e80f7b0: Layer already exists
12228ba7a3b1: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9b55156abf26: Layer already exists
9c742cd6c7a5: Layer already exists
044b87333170: Pushed
47: digest: sha256:26158de698165b62813362feb092488f99f8949fd5a0b0a06678bb4b99b7f7e2 size: 3056
+ envsubst
+ kubectl apply -f -
deployment.apps/nodeapp-deployment configured
service/app-service configured
+ docker logout
Removing login credentials for https://index.docker.io/v1/
Finished: SUCCESS
```

The Snippet of Deployed Application Based on DATA_FILLE are shown below

When DATA_FILE==Response-other.json

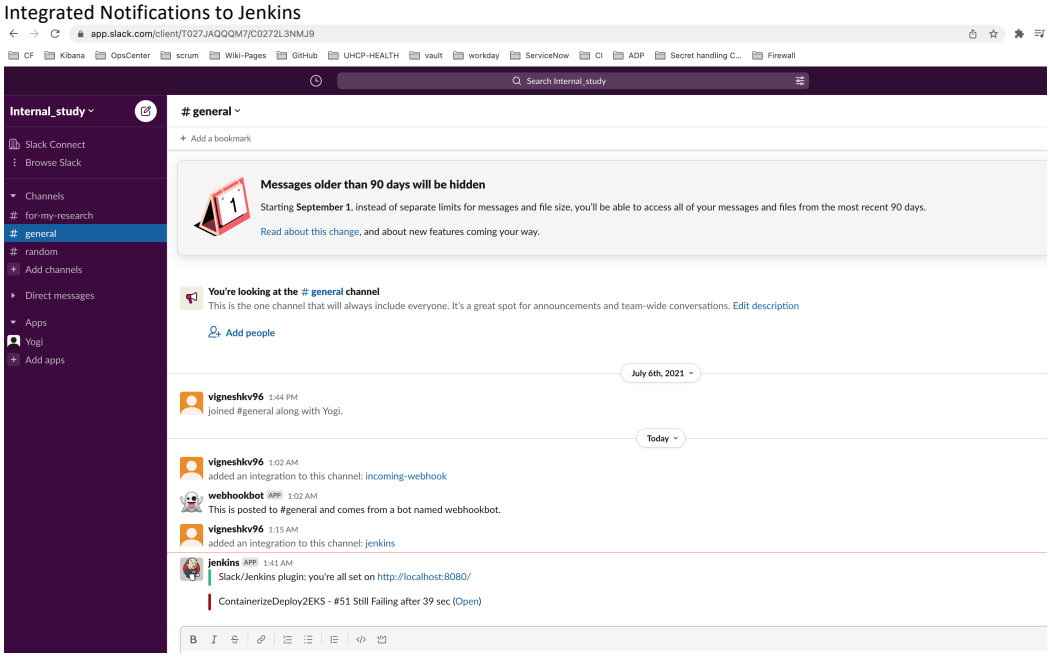


when DATA_FILE==Responses.json



In Any step if the build fails an Notification will be sent to Slack as below

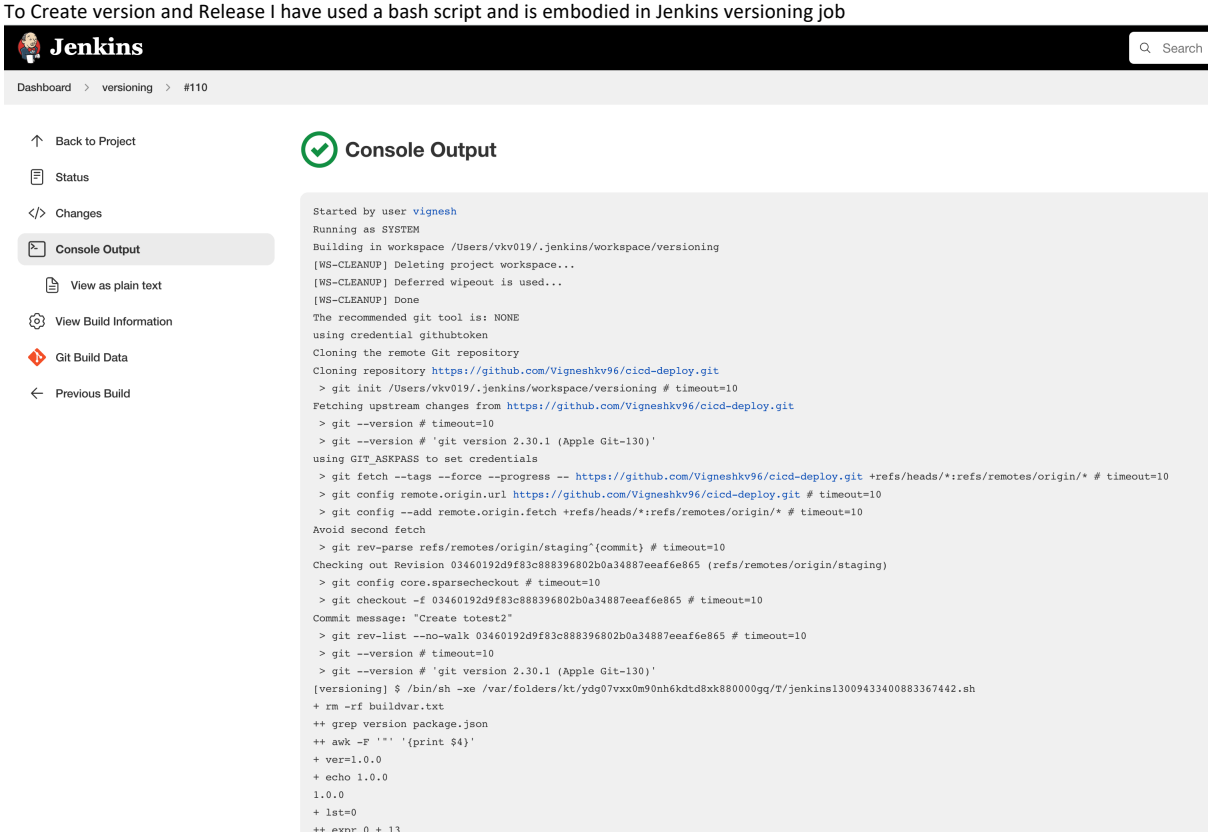
Integrated Notifications to Jenkins



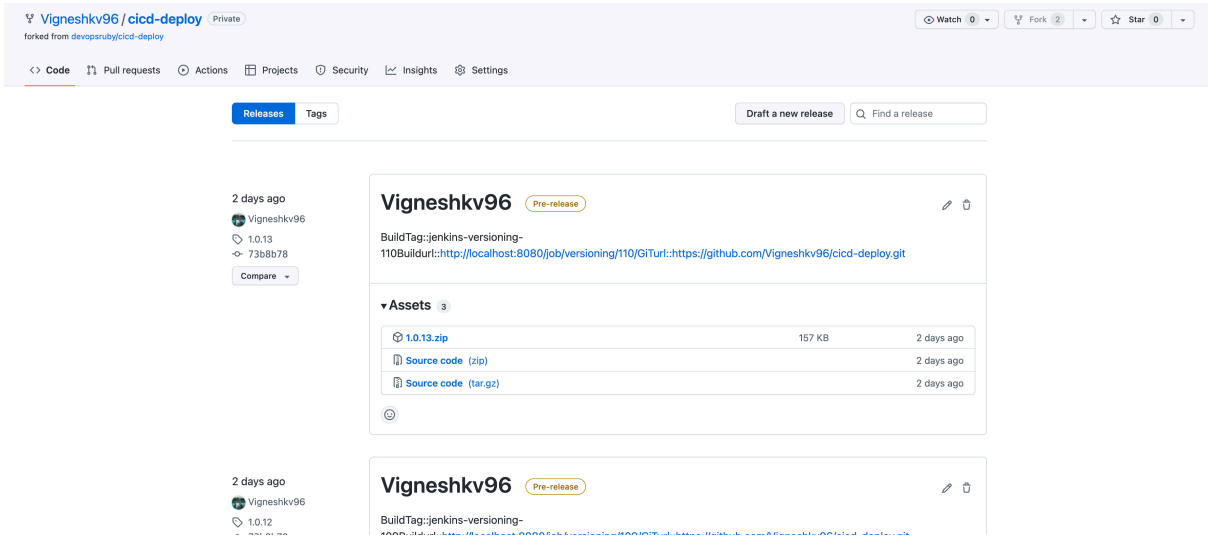
Jenkins Jobs

Versioning

To Create version and Release I have used a bash script and is embodied in Jenkins versioning job



GitHub Release Screenshot



References Websites

- https://docs.aws.amazon.com/eks/latest/userguide/create-cluster.html
- https://levelup.gitconnected.com/send-slack-notifications-with-jenkins-f8e8b2d2e748
- https://stackoverflow.com/questions/49425534/how-to-add-conditional-postbuild-actions-using-jenkins-declarative-pipeline
- https://www.jenkins.io/doc/book/pipeline/pipeline-best-practices/
- https://www.densify.com/kubernetes-autoscaling/kubernetes-service-load-balancer
- https://eksctl.io/
- https://techblog.geekyants.com
- https://docs.aws.amazon.com/eks/latest/userguide/sample-deployment.html