

# CICD-DEPLOY

K V, Vignesh

+91-9962623973  
Vigneshkv96@gmail.com

# Table of Contents

- Problem statement
- Tools
- Flow diagram
- Implementation Screenshot
- Versioning
- Reference Websites

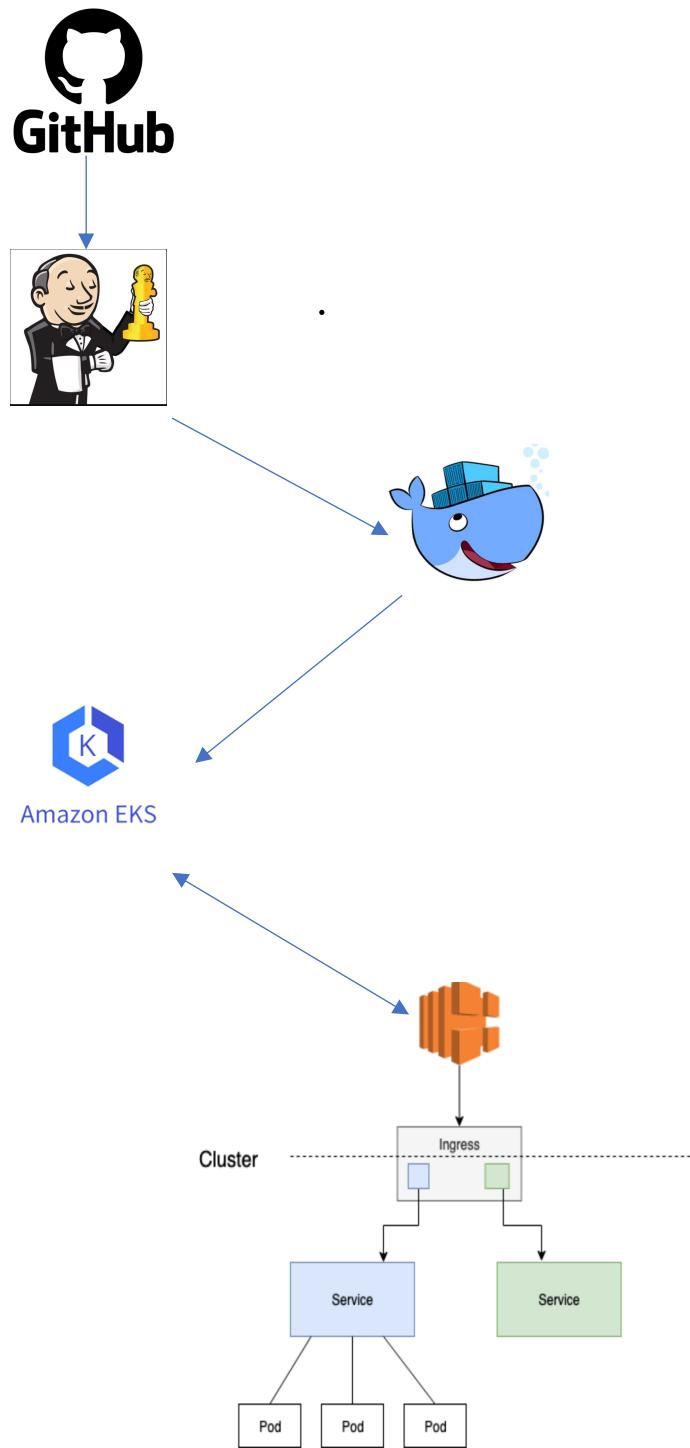
# Problem Statement

- 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

This screenshot shows a GitHub repository page for a private fork of `cicd-deploy`. The repository has 4 branches and 12 tags. A commit from `K V, Vignesh and K V, Vignesh` was made 12 hours ago, titled "Auto created Jenkinsfile from auto conversion". The commit details show the creation of various files like controllers, data, helpers, models, routes, test, views, Dockerfile, Jenkinsfile, README.md, deployment-manifest.yml, index.js, package-lock.json, and package.json. All commits are marked as "initial commit".

The Build Job will get Triggered

This screenshot shows the Jenkins dashboard at `localhost:8080`. It displays a list of builds: "Build" (last success 2 hr 2 min, last failure 2 days 2 hr), "ContainerizeDeploy2EKS" (last success 4 hr 50 min, last failure 3 min 9 sec), "UnitTesting" (last success 2 hr 1 min, last failure 2 hr 7 min), and "versioning" (last success 1 day 5 hr, last failure 1 day 5 hr). The "Build Queue" section shows one build named "containerization" (Build #3) currently running. The "Build Executor Status" section shows one executor named "Idle".

# CICD-DEPLOY

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

Dashboard > Unit testing >

↑ Back to Dashboard

Status

</> Recent Changes

Full Stage View

### Pipeline UnitTesting

#### Stage View

Build History trend ▾

Filter builds...

Average stage times:  
(Average full run time: ~21s)

#	Date	No Changes	Declarative: Tool Install	polling	test
#36	Aug 14	23:44	456ms	13s	3s
#35	Aug 14	23:41	573ms	15s	3s
#34	Aug 14	23:38	426ms	15s	3s
#33	Aug 14	23:26	483ms	16s	4s
#32	Aug 14	23:16	418ms	15s	3s
#31	Aug 14	23:02	420ms	15s	3s
#30	Aug 14	22:56	420ms	15s	3s
#29	Aug 14	22:49	420ms	15s	3s
#28	Aug 14	01:24	420ms	15s	3s

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

# CICD-DEPLOY

The screenshot shows a project dashboard titled "Project ContainerizeDeploy2EKS". On the left, there's a sidebar with links: "Back to Dashboard", "Status" (which is currently selected), "Changes", "Workspace", and "Convert This Job To Pipeline". Below the sidebar, there's a "Build History" section with a search bar and a table of builds:

Build ID	Date
#51	15 Aug 2022, 01:42
#50	14 Aug 2022, 23:48
#49	14 Aug 2022, 23:45
#48	14 Aug 2022, 23:41
#47	14 Aug 2022, 20:54
#46	14 Aug 2022, 20:49
#45	14 Aug 2022, 20:29
#44	...

On the right side, there are sections for "Workspace" and "Recent Changes". Under "Recent Changes", there's a list of the last completed builds:

- Last build (#51), 9 hr 0 min ago
- Last stable build (#47), 13 hr ago
- Last successful build (#47), 13 hr ago
- Last failed build (#51), 9 hr 0 min ago
- Last unsuccessful build (#51), 9 hr 0 min ago
- Last completed build (#51), 9 hr 0 min ago

## Permalinks

A list of build permalinks:

- [Last build \(#51\)](#), 9 hr 0 min ago
- [Last stable build \(#47\)](#), 13 hr ago
- [Last successful build \(#47\)](#), 13 hr ago
- [Last failed build \(#51\)](#), 9 hr 0 min ago
- [Last unsuccessful build \(#51\)](#), 9 hr 0 min ago
- [Last completed build \(#51\)](#), 9 hr 0 min ago

## The Build Image is pushed to DockerHub

The screenshot shows a DockerHub repository page for "vigneshvalsan/nodejs". The top navigation bar includes "dockerhub", a search bar, "Explore", "Repositories", "Organizations", "Help", and a user profile "vigneshvalsan".

The repository details page shows the following information:

- Description:** "This repository does not have a description" with an edit icon.
- Last pushed:** "a few seconds ago"
- Docker commands:** "To push a new tag to this repository, `docker push vigneshvalsan/nodejs:tagname`". A "Public View" button is also present.
- Tags and Scans:** "This repository contains 15 tag(s)." A table of tags:

TAG	OS	PULLED	PUSHED
46	Ubuntu	a few seconds ago	a few seconds ago
45	Ubuntu	---	21 minutes ago
44	Ubuntu	---	3 hours ago
43	Ubuntu	3 hours ago	3 hours ago
42	Ubuntu	---	3 hours ago

[See all](#) [Go to Advanced Image Management](#)

**Automated Builds:** "Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating." Available with Pro, Team and Business subscriptions. Buttons for "Upgrade" and "Learn more".

## CICD-DEPLOY

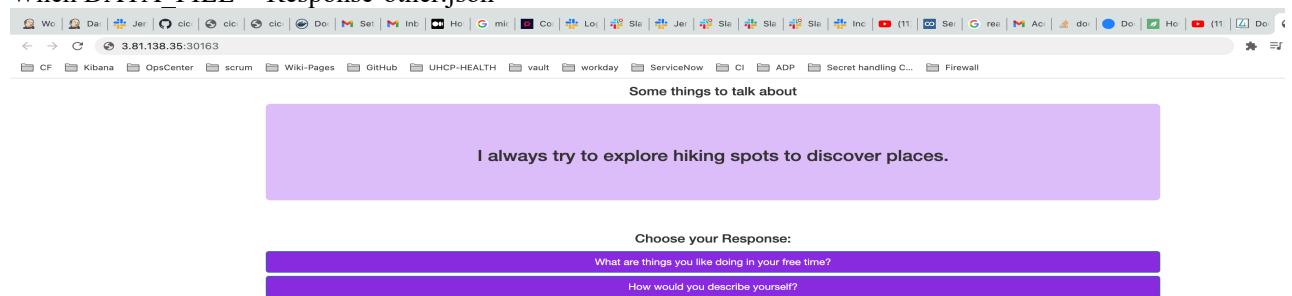
Then the Image is deployed to AWS EKS using kubectl apply command

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\_FILE are shown below

When DATA\_FILE==Response-other.json



# CICD-DEPLOY

when DATA\_FILE==Responses.json

The screenshot shows a web-based survey or poll. At the top, there's a navigation bar with various icons and links. Below it, a message says "I know the ins and outs of Linux systems." A purple box contains the text "Linux filesystems are easier to learn." At the bottom, there are three purple buttons labeled "Choose your Response:" followed by "DevOps", "Application Development", and "Systems Engineer".

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

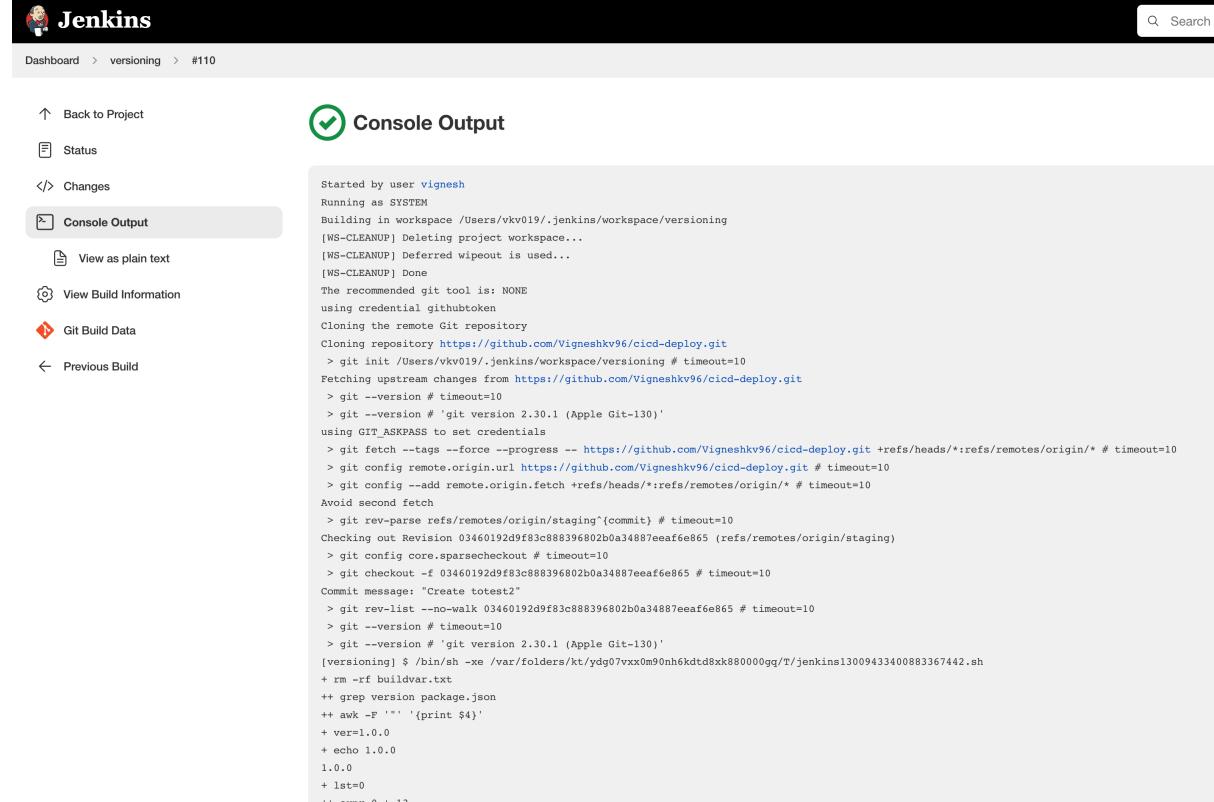
Integrated Notifications to Jenkins

The screenshot shows a Slack channel named "# general". It displays a message from "vigneshkv96" at 1:44 PM stating they joined the channel along with "Yogi". Below this, another message from "vigneshkv96" at 1:02 AM indicates an integration with Jenkins. A bot named "webhookdot" posted a message at 1:02 AM. A Jenkins job named "ContainerizeDeploy2EKS" failed at 1:15 AM. A Jenkins plugin message at 1:41 AM says the plugin is set up on http://localhost:8080.

Jenkins Jobs

# Versioning

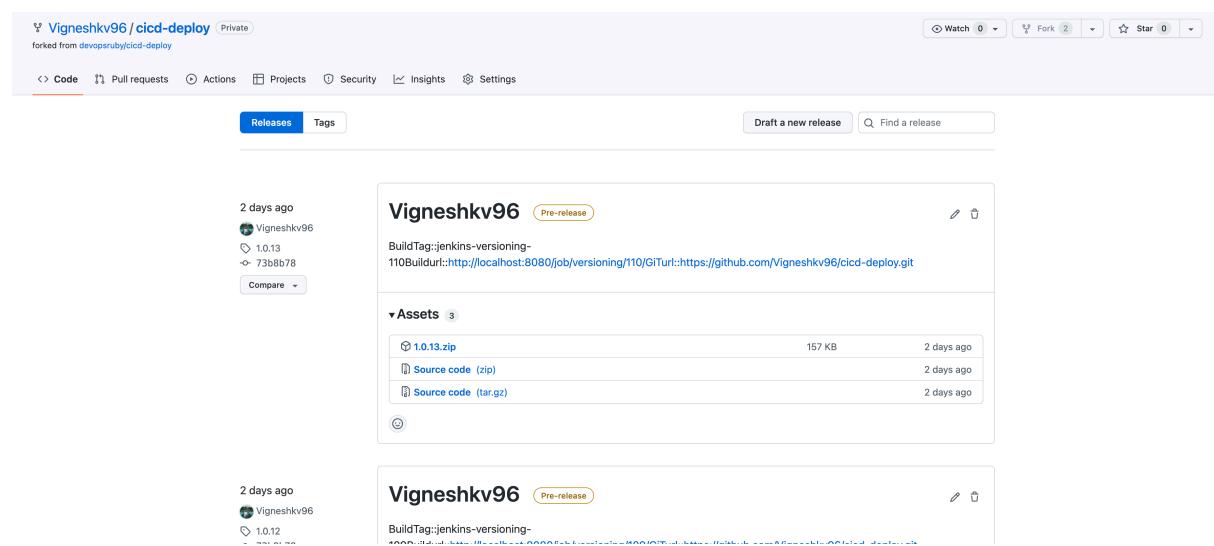
To Create version and Release I have used a bash script and is embodied in Jenkins versioning job



The screenshot shows the Jenkins interface for a build job named "versioning". The "Console Output" tab is selected, displaying the command-line logs of the build process. The logs show the script performing several actions:

- Started by user vignesh
- Running as SYSTEM
- Building in workspace /Users/vkv019/.jenkins/workspace/versioning
- [WS-CLEANUP] Deleting project workspace...
- [WS-CLEANUP] Deferred wipeout is used...
- [WS-CLEANUP] Done
- The recommended git tool is: NONE
- using credential githubtoken
- Cloning the remote Git repository <https://github.com/Vigneshkv96/cicd-deploy.git>
- > git init /Users/vkv019/.jenkins/workspace/versioning # timeout=10
- Fetching upstream changes from <https://github.com/Vigneshkv96/cicd-deploy.git>
- > git --version # timeout=10
- > git --version # 'git version 2.30.1 (Apple Git-130)'
- using GIT\_ASKPASS to set credentials
- > git fetch --tags --force --progress -- <https://github.com/Vigneshkv96/cicd-deploy.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10
- > git config remote.origin.url <https://github.com/Vigneshkv96/cicd-deploy.git> # timeout=10
- > git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10
- Avoid second fetch
- > git rev-parse refs/remotes/origin/staging^{commit} # timeout=10
- Checking out Revision 03460192d9f83c888396802b0a34887eef6e865 (refs/remotes/origin/staging)
- > git config core.sparsecheckout # timeout=10
- > git checkout -f 03460192d9f83c888396802b0a34887eef6e865 # timeout=10
- Commit message: "Create totest2"
- > git rev-list --no-walk 03460192d9f83c888396802b0a34887eef6e865 # timeout=10
- > git --version # timeout=10
- > git --version # 'git version 2.30.1 (Apple Git-130)'
- [versioning] \$ /bin/sh -xe /var/folders/kt/ydg07vxx0m90nh6ktd8xk880000gq/T/jenkinsl3009433400883367442.sh
- + rm -rf buildvar.txt
- + grep version package.json
- + awk -F '"' '{print \$4}'
- + ver=1.0.0
- + echo 1.0.0
- 1.0.0
- + lstat=
- + eval \$0 + 13

## GitHub Release Screenshot



The screenshot shows the GitHub Releases page for the repository "Vigneshkv96 / cicd-deploy". It displays two releases:

- 1.0.13 (Pre-release)**: Published 2 days ago. Assets include "1.0.13.zip" (157 KB, 2 days ago), "Source code (zip)" (2 days ago), and "Source code (tar.gz)" (2 days ago).
- 1.0.12 (Pre-release)**: Published 2 days ago. Assets include "1.0.12.zip" (157 KB, 2 days ago), "Source code (zip)" (2 days ago), and "Source code (tar.gz)" (2 days ago).

# 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>