Building Docker image for Nodejs App via Guthub to Docker: CICD Jenkins.

Ubuntu, Linux
Github.

My github page: https://github.com/UrysKingdom/node-dockerized-projects/blob/main/Dockerfile

Intro Brief:

This project demonstrates the continuous integration and continuous deployment (CICD) pipeline for building a Docker image for a Node.js application via GitHub using Jenkins. The process involves setting up Node.js, creating a Node.js application with the Express framework, testing with Mocha, creating a GitHub repository, configuring Jenkins pipelines, and finally, deploying the Docker image. This end-to-end workflow ensures efficient and automated deployment of the Node.js application.

Process in detail:

1. To install Nodejs

```
: sudo apt install nodejs
```

2. To install npm (& check the version)

```
: sudo apt install npm
node --version
npm --version
```

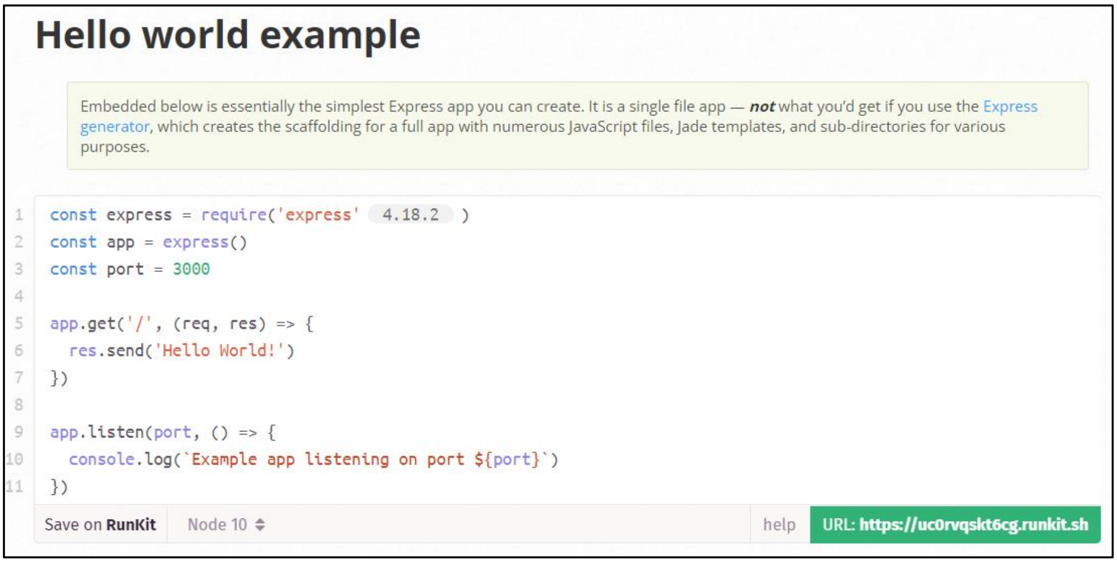
3.Go to <www.nodejs.org/download>, and start our nodejs application
Build using 'Express framework' (expressjs.com)
Initiate after making 'package.json file' in a directory 'apps'.

```
:mkdir apps
 cd apps

 ~/apps# npm init
      npm install express --save
```

4.To create 'index.js' file.

```
~apps#  vi index.js
```

(This will open a blank space to write a file. Copy "Hello World' example from expressjs.com at 'Getting started section and paste, and save it: press ‘I’ to insert text, press ‘esc’ to exit, the type ‘:wq’ and press ‘Enter’ to save the file and exit.)



Hello world example

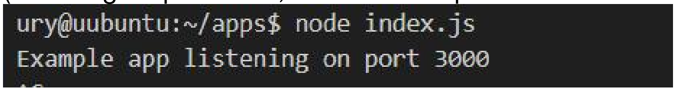
Embedded below is essentially the simplest Express app you can create. It is a single file app — ***not*** what you'd get if you use the Express generator, which creates the scaffolding for a full app with numerous JavaScript files, Jade templates, and sub-directories for various purposes.

```
const express = require('express'  4.18.2 )
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

https://expressjs.com/en/starter/hello-world.html

5. ~apps# node index.js

(Listening on port 3000, click www.https://localhost:3000 to see the '*Hello world*' example)



```
ury@uubuntu:~/apps$ node index.js
Example app listening on port 3000
```

6. ~apps# npm install --save-dev mocha

7. Verify the 'package.jason' file using '**vi package.json**' command.

Then:

```
scripts: {
    "test":"mocha"
    "build": "echo 'running Build script'"
```

8. ~apps# npm test

9. ~apps# mkdir test
              cd test

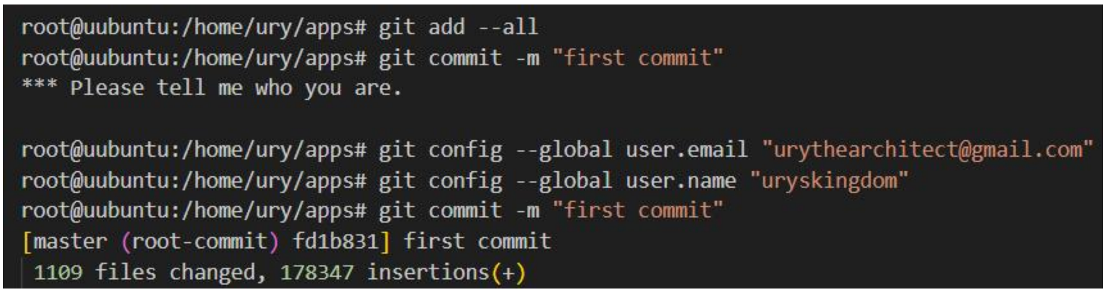
To create a file,
~apps# touch mytest.test.js/test

10. ~apps# npm test

11. ~apps# npm run build
      ~apps# vi pacakge.json

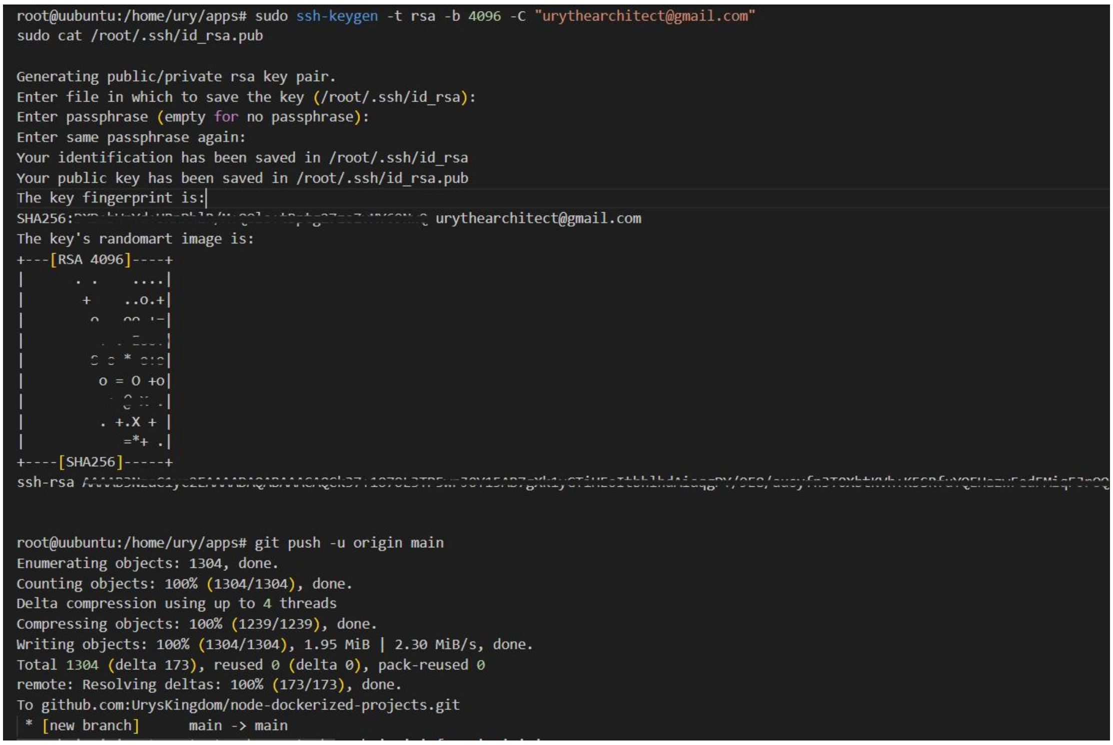
12. Now to create a Github repository at the Github website. (Repo name example: node-project).

```
    ~apps# git init
~apps# . git/
~apps# git status
~apps# git add -all
~apps# git commit -m "first commit"
~apps# git branch -m main
```



```
root@uubuntu:/home/ury/apps# git add --all
root@uubuntu:/home/ury/apps# git commit -m "first commit"
*** Please tell me who you are.

root@uubuntu:/home/ury/apps# git config --global user.email "urythearchitect@gmail.com"
root@uubuntu:/home/ury/apps# git config --global user.name "uryskingdom"
root@uubuntu:/home/ury/apps# git commit -m "first commit"
[master (root-commit) fd1b831] first commit
 1109 files changed, 178347 insertions(+)
```

13. Using 'ssh' on the git web address click & copy "git remote add origin ~".
to see all the file uploaded to the github page.



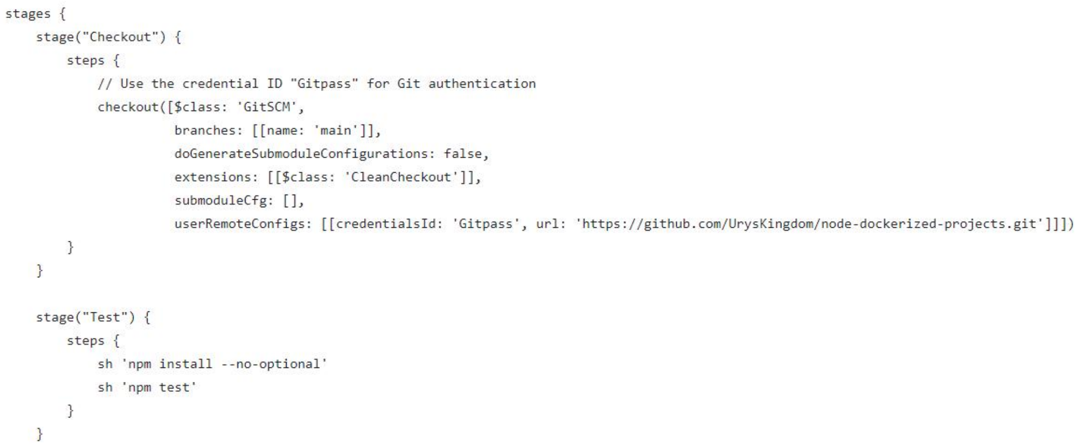
```
root@uubuntu:/home/ury/apps# sudo ssh-keygen -t rsa -b 4096 -C "urythearchitect@gmail.com"
sudo cat /root/.ssh/id_rsa.pub

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:[redacted] urythearchitect@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|      . .    ....|
|       +    ..o.+|
[redacted]
|        o = O +o|
[redacted]
|        . +.X + |
|           =*+ .|
+----[SHA256]-----+
ssh-rsa [redacted]


root@uubuntu:/home/ury/apps# git push -u origin main
Enumerating objects: 1304, done.
Counting objects: 100% (1304/1304), done.
Delta compression using up to 4 threads
Compressing objects: 100% (1239/1239), done.
Writing objects: 100% (1304/1304), 1.95 MiB | 2.30 MiB/s, done.
Total 1304 (delta 173), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (173/173), done.
To github.com:UrysKingdom/node-dockerized-projects.git
 * [new branch]      main -> main
```

Jenkins Pipeline

14. Create a new pipeline, name for Github project.

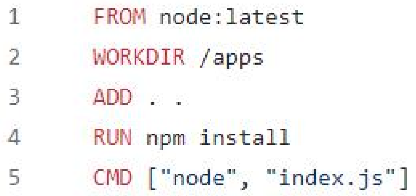
15. Create a Jenkins file with steps.

https://github.com/UrysKingdom/node-dockerized-projects/blob/main/Jenskinsfile



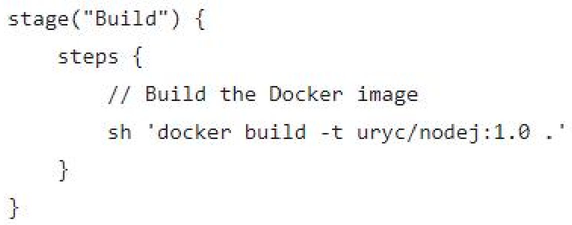
```
stages {
    stage("Checkout") {
        steps {
            // Use the credential ID "Gitpass" for Git authentication
            checkout([$class: 'GitSCM',
                      branches: [[name: 'main']],
                      doGenerateSubmoduleConfigurations: false,
                      extensions: [[$class: 'CleanCheckout']],
                      submoduleCfg: [],
                      userRemoteConfigs: [[credentialsId: 'Gitpass', url: 'https://github.com/UrysKingdom/node-dockerized-projects.git']]])
        }
    }

    stage("Test") {
        steps {
            sh 'npm install --no-optional'
            sh 'npm test'
        }
    }
```

16. Create a docker file for **nodejs** at git.
https://github.com/UrysKingdom/node-dockerized-projects/blob/main/Dockerfile



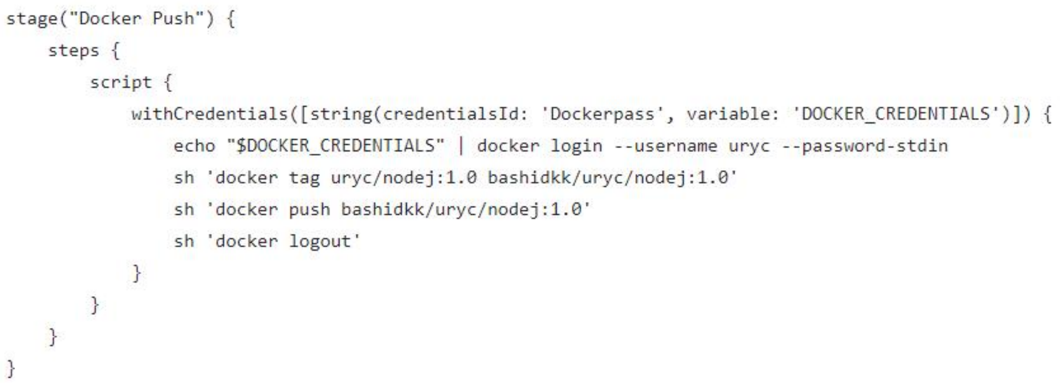
```
FROM node:latest
WORKDIR /apps
ADD . .
RUN npm install
CMD ["node", "index.js"]
```

17. Add new stage, stage for 'Build image' in the Jenkins script.



```
stage("Build") {
    steps {
        // Build the Docker image
        sh 'docker build -t uryc/nodej:1.0 .'
    }
}
```

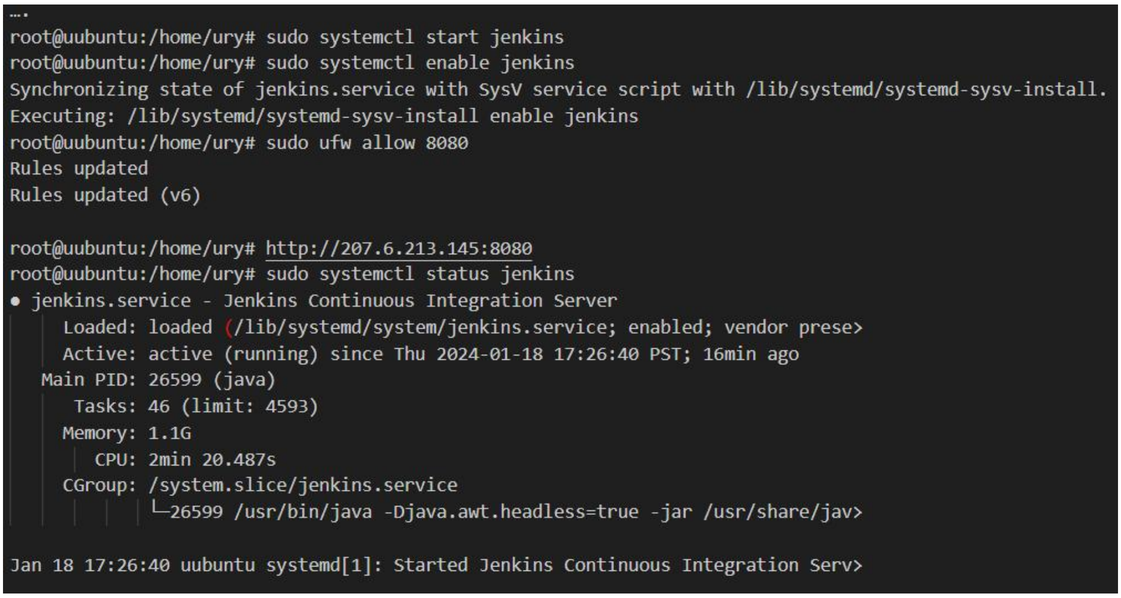
17. Add new stage, stage for 'Docker Push' in the Jenkins script with credentials.



```
stage("Docker Push") {
    steps {
        script {
            withCredentials([string(credentialsId: 'Dockerpass', variable: 'DOCKER_CREDENTIALS')]) {
                echo "$DOCKER_CREDENTIALS" | docker login --username uryc --password-stdin
                sh 'docker tag uryc/nodej:1.0 bashidkk/uryc/nodej:1.0'
                sh 'docker push bashidkk/uryc/nodej:1.0'
                sh 'docker logout'
            }
        }
    }
}
```

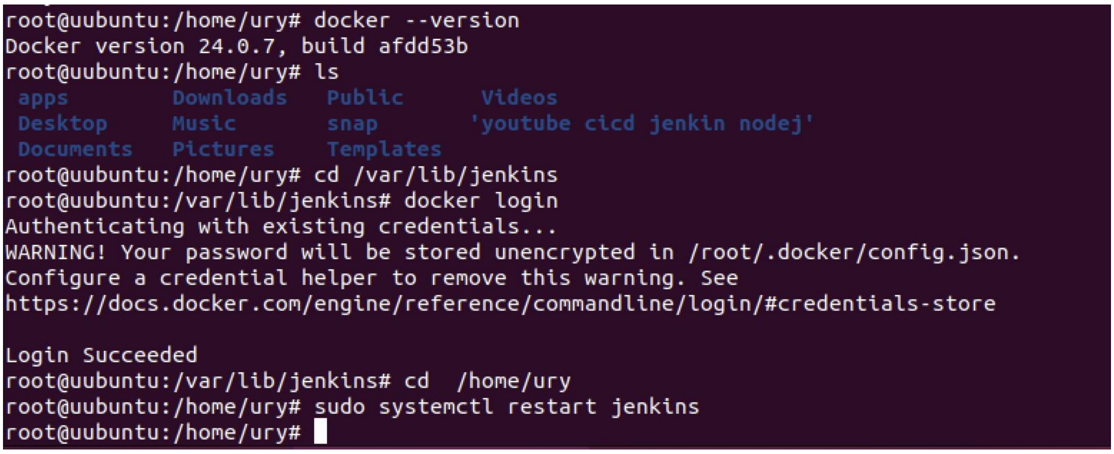
19. ssh Jenkins and connect.

```
~apps# ssh userid@http://xxx.x.xxx.xxx:8080  (e.g. ury@http://Local_ip_address:8080)
```



```
root@uubuntu:/home/ury# sudo systemctl start jenkins
root@uubuntu:/home/ury# sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable jenkins
root@uubuntu:/home/ury# sudo ufw allow 8080
Rules updated
Rules updated (v6)

root@uubuntu:/home/ury# http://207.6.213.145:8080
root@uubuntu:/home/ury# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor prese>
     Active: active (running) since Thu 2024-01-18 17:26:40 PST; 16min ago
   Main PID: 26599 (java)
      Tasks: 46 (limit: 4593)
     Memory: 1.1G
        CPU: 2min 20.487s
     CGroup: /system.slice/jenkins.service
             └─26599 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/jav>

Jan 18 17:26:40 uubuntu systemd[1]: Started Jenkins Continuous Integration Serv>
```

Docker

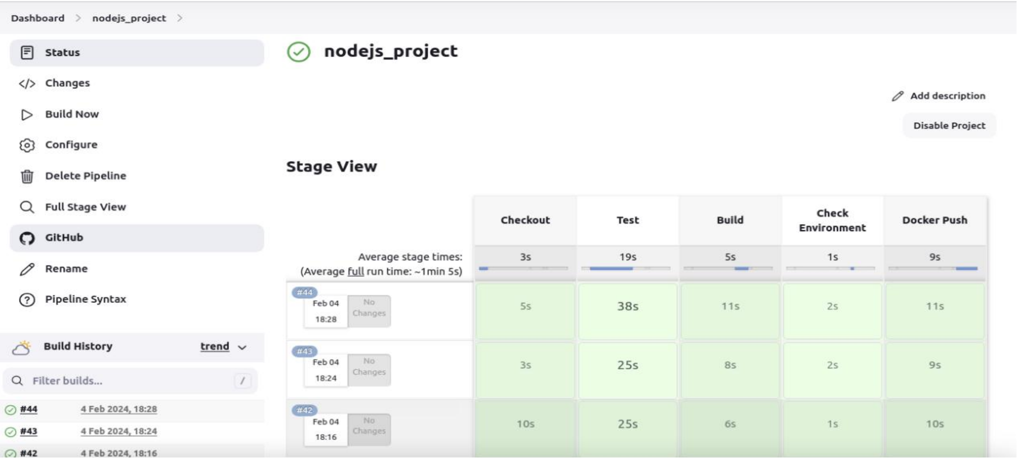


```
root@uubuntu:/home/ury# docker --version
Docker version 24.0.7, build afdd53b
root@uubuntu:/home/ury# ls
 apps        Downloads   Public      Videos
 Desktop     Music       snap       'youtube cicd jenkin nodej'
 Documents   Pictures    Templates
root@uubuntu:/home/ury# cd /var/lib/jenkins
root@uubuntu:/var/lib/jenkins# docker login
Authenticating with existing credentials...
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/#credentials-store

Login Succeeded
root@uubuntu:/var/lib/jenkins# cd  /home/ury
root@uubuntu:/home/ury# sudo systemctl restart jenkins
root@uubuntu:/home/ury# 
```

20. Docker images
    docker run -p 88/3000 -d uryc/nodej:1.0

21. Check the "Hello world" in the web ip address.

22. Jenkins pipeline success!



nodejs_project

Stage View

	Checkout	Test	Build	Check Environment	Docker Push
Average stage times: (Average full run time: ~1min 5s)	3s	19s	5s	1s	9s
#44 Feb 04 18:28 No Changes	5s	38s	11s	2s	11s
#43 Feb 04 18:24 No Changes	3s	25s	8s	2s	9s
#42 Feb 04 18:16 No Changes	10s	25s	6s	1s	10s

#44 4 Feb 2024, 18:28
#43 4 Feb 2024, 18:24
#42 4 Feb 2024, 18:16