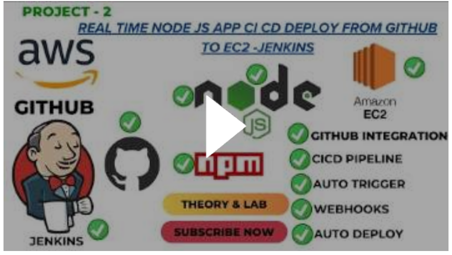
# DevOps Project-1

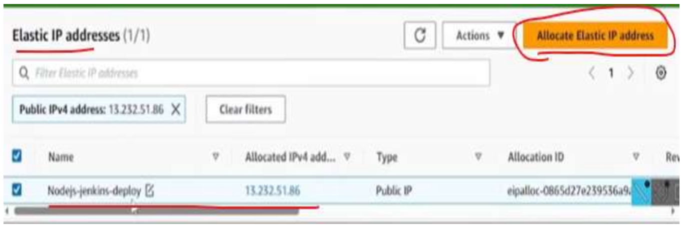
Friday, February 9, 2024 4:57 PM

- Project Name = **Real-time NODE.js App deployment with Jenkins CI CD pipeline|Github Repo auto trigger webhook.**
- Project Link = Real-time NODE.js App deployment with Jenkins CI CD pipeline|Github Repo auto trigger webhook.



- Project Prerequisite:- NODE.js Application Deployement on EC2 server with CICD pipeline using nxing webserver for reverse Proxy.

1. **Server AWS = done**
   i. Create ec2 instance on aws. (name = Nodes-Jenkins-deploy, rest same)
   ii. For connect web server we have to create Elastic IP Address.
   iii. For elastic we need to click elastic IP on ec2 and it's created.

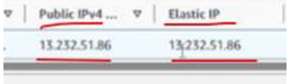


Elastic IP addresses (1/1)

| Name | Allocated IPv4 add... | Type | Allocation ID | Rev |
|---|---|---|---|---|
| Nodejs-jenkins-deploy | 13.232.51.86 | Public IP | eipalloc-0865d27e239536a9 | |

Public IPv4 address: 13.232.51.86

   iv. Now click on associate elastic ip and select instance, private IP Address and click on associate button.
   v. Once we associate with private ip, my ec2 public ip will change and both public and elastic ip will same.



| Public IPv4 ... | Elastic IP |
|---|---|
| 13.232.51.86 | 13.232.51.86 |

   vi. now connect this ip to mobastream application

2. **Ubuntu OS = done**
   i. Choose ubuntu os on ec2 instance and open in mobastream
   ii. Now we connect to server.
   iii. Now we have to perform below command to update the machine.
      1) sudo su - (make root user)
      2) Sudo apt update = (make machine update and package)

   iv. cc

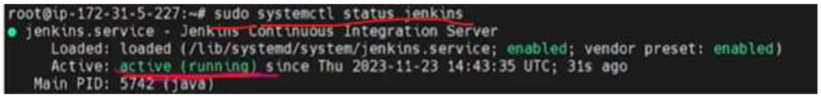
3. **Git = done**
   i. Install Git on ubuntu machine
   ii. Git --version

4. **Jenkins Installation = done**

   Jenkins and java installation process = Linux (jenkins.io)

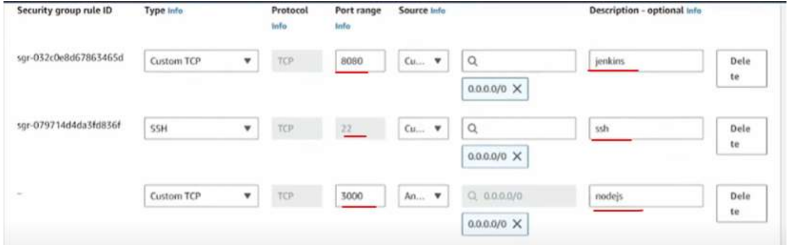
   i. Before installing Jenkins we have to install first JAVA JDK.
      1) sudo apt update
      2) sudo apt install fontconfig openjdk-17-jre
      3) java -version
         openjdk version "17.0.8" 2023-07-18
         OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)
         OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)

   ii. Now we have to install Jenkins using below step on ubuntu.
      1) In ubuntu we have to install Jenkins LTS version below (LTS = Long Term Support release).
         a) sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
            https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
         b) echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
            https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
            /etc/apt/sources.list.d/jenkins.list > /dev/null
         c) sudo apt-get update
         d) sudo apt-get install jenkins
         e) sudo systemctl enable jenkins
         f) sudo systemctl start jenkins
         g) sudo systemctl status jenkins
      2) Now our jenkins is running successfully.



```
root@ip-172-31-5-227:~# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-11-23 14:43:35 UTC; 31s ago
   Main PID: 5742 (java)
```

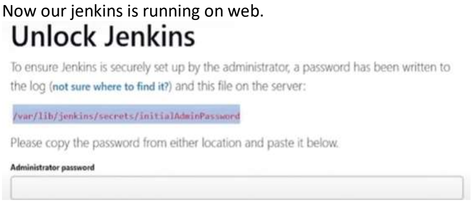
3) Now access jenkins on web.
   a) Before open jenkins on web we need to enable jenkins port on ec2 security  inbound rules.



| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | | Description - optional Info | |
|---|---|---|---|---|---|---|---|
| sgr-032c0e8d67863465d | Custom TCP ▼ | TCP | 8080 | Cu... ▼ | Q<br>0.0.0.0/0 X | jenkins | Delete |
| sgr-079714d4da3fd836f | SSH ▼ | TCP | 22 | Cu... ▼ | Q<br>0.0.0.0/0 X | ssh | Delete |
| - | Custom TCP ▼ | TCP | 3000 | An... ▼ | Q 0.0.0.0/0<br>0.0.0.0/0 X | nodejs | Delete |

   b) cc

4) Now our jenkins is running on web.



## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

/var/lib/jenkins/secrets/initialAdminPassword

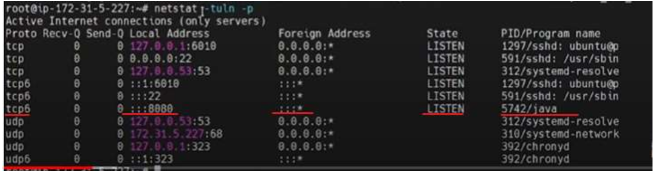
Please copy the password from either location and paste it below.

Administrator password

5) Now we have to install some suggest plugin and required plugin.

### 5. netstat -tuln -p   =  done

i. Now we have to check our port is working or not using below command.



```
root@ip-172-31-5-227:~# netstat -tuln -p
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:6010          0.0.0.0:*               LISTEN      1297/sshd: ubuntu@p
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      591/sshd: /usr/sbin
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      312/systemd-resolve
tcp6       0      0 ::1:6010                :::*                    LISTEN      1297/sshd: ubuntu@p
tcp6       0      0 :::22                   :::*                    LISTEN      591/sshd: /usr/sbin
tcp6       0      0 :::8080                 :::*                    LISTEN      5742/java
udp        0      0 127.0.0.53:53           0.0.0.0:*                           312/systemd-resolve
udp        0      0 172.31.5.227:68         0.0.0.0:*                           310/systemd-network
udp        0      0 127.0.0.1:323           0.0.0.0:*                           392/chronyd
udp6       0      0 ::1:323                 :::*                                392/chronyd
```

### 6. Node.js & NPM Installation on server = node 18 and npm 9  = done

i. Now we have to install node.js & NPM on Ubuntu machine using below command.

ii. Here is the github link command to install nodejs .
Link = https://github.com/nodesource/distributions

**iii. Download and import the nodesource GPG key**

1) sudo apt-get update
2) sudo apt-get install -y ca-certificates curl gnupg
3) sudo mkdir -p /etc/apt/keyrings
4) curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

**iv. Create deb repository.**

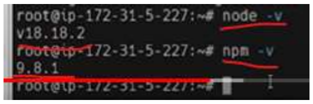
1) NODE_MAJOR=18
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | sudo tee /etc/apt/sources.list.d/nodesource.list

v. Run Update and install

1) sudo apt-get update
2) sudo apt-get install nodejs -y

vi. Now we have to validate the node js and NPM version.

1) node -v



```
root@ip-172-31-5-227:~# node -v
v18.18.2
root@ip-172-31-5-227:~# npm -v
9.8.1
root@ip-172-31-5-227:~#
```

2) Cc

vii. Choose and try below command if above is not work

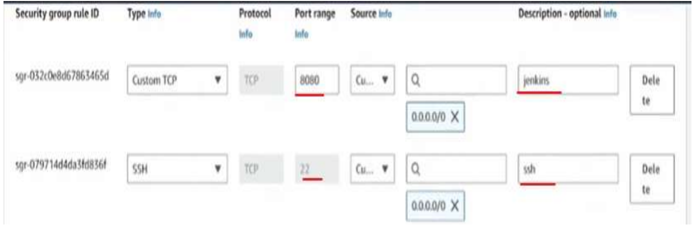
1) curl -fsSL https://deb.nodesource.com/setup_21.x | sudo -E bash - &&\
2) sudo apt-get install -y nodejs

viii. cc

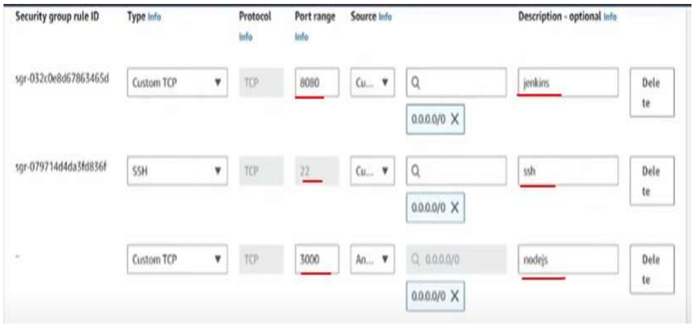
### 7. Firewall Configuration  =  done

i. Now we have to set security port for above application like ssh, jenkins, nodejs.

| | Name | Port Number |
|---|---|---|
| 1. | ssh | 22 |
| 2. | Jenkins | 8080 |
| 3. | nodejs | 3000 |



| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | | Description - optional Info | |
|---|---|---|---|---|---|---|---|
| sgr-032c0e8d67863465d | Custom TCP ▼ | TCP | 8080 | Cu... ▼ | Q<br>0.0.0.0/0 X | jenkins | Delete |
| sgr-079714d4da3fd836f | SSH ▼ | TCP | 22 | Cu... ▼ | Q<br>0.0.0.0/0 X | ssh | Delete |



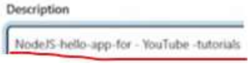
| Security group rule ID | Type Info | Protocol Info | Port range Info | Source Info | | Description - optional Info | |
|---|---|---|---|---|---|---|---|
| sgr-032c0e8d67863465d | Custom TCP | TCP | 8080 | Cu... | 0.0.0.0/0 X | jenkins | Delete |
| sgr-079714d4da3fd836f | SSH | TCP | 22 | Cu... | 0.0.0.0/0 X | ssh | Delete |
| - | Custom TCP | TCP | 3000 | An... | 0.0.0.0/0<br>0.0.0.0/0 X | nodejs | Delete |

### 8. GITHUB integration with Jenkins. = done

i. Now we have create a jenkins job using github source code.

ii. Now we have to setup jenkins and click on new item(nodeJS-hello-app) project name.

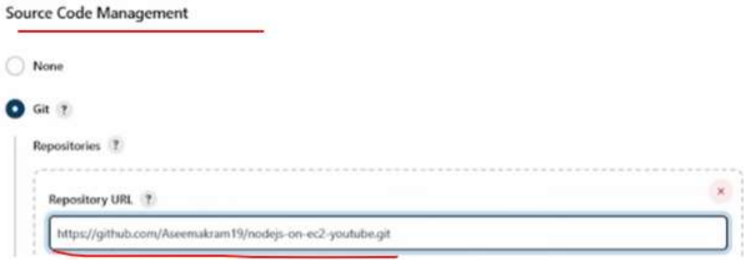
iii. Now give des = nodejs-hello-app-for-youtube.



Description

NodeJS-hello-app-for - YouTube -tutorials

iv. Now we have to give source code github repository URL and rest same.



Source Code Management

○ None

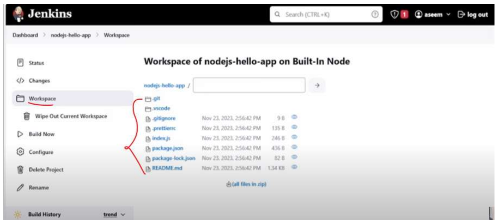
● Git

Repositories

Repository URL

https://github.com/Aseemakram19/nodejs-on-ec2-youtube.git

v. Now click on save and click on Build now. Once build is successfully then click on workspace and in workspace we can see all the project source code on jenkins



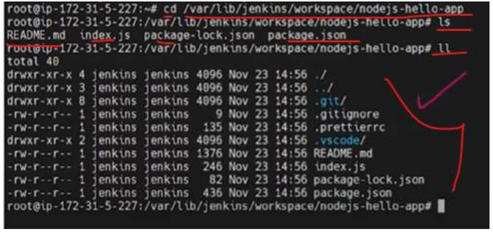
Workspace of nodejs-hello-app on Built-In Node

nodejs-hello-app /

| | | |
|---|---|---|
| .git | | |
| .vscode | | |
| .gitignore | Nov 23, 2023, 2:56:42 PM | 9 B |
| .prettierrc | Nov 23, 2023, 2:56:42 PM | 135 B |
| index.js | Nov 23, 2023, 2:56:42 PM | 246 B |
| package.json | Nov 23, 2023, 2:56:42 PM | 436 B |
| package-lock.json | Nov 23, 2023, 2:56:42 PM | 82 B |
| README.md | Nov 23, 2023, 2:56:42 PM | 1.34 KB |

(all files in zip)

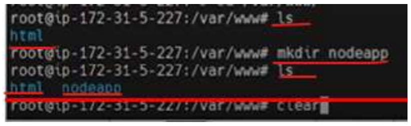
vi. After build check the path.



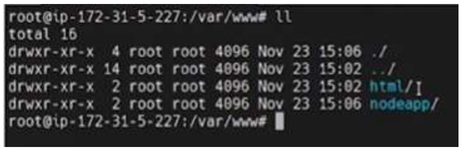
```
root@ip-172-31-5-227:~# cd /var/lib/jenkins/workspace/nodejs-hello-app
root@ip-172-31-5-227:/var/lib/jenkins/workspace/nodejs-hello-app# ls
README.md  index.js  package-lock.json  package.json
root@ip-172-31-5-227:/var/lib/jenkins/workspace/nodejs-hello-app# ll
total 40
drwxr-xr-x 4 jenkins jenkins 4096 Nov 23 14:56 ./
drwxr-xr-x 3 jenkins jenkins 4096 Nov 23 14:56 ../
drwxr-xr-x 8 jenkins jenkins 4096 Nov 23 14:56 .git/
-rw-r--r-- 1 jenkins jenkins    9 Nov 23 14:56 .gitignore
-rw-r--r-- 1 jenkins jenkins  135 Nov 23 14:56 .prettierrc
drwxr-xr-x 2 jenkins jenkins 4096 Nov 23 14:56 .vscode/
-rw-r--r-- 1 jenkins jenkins 1376 Nov 23 14:56 README.md
-rw-r--r-- 1 jenkins jenkins  246 Nov 23 14:56 index.js
-rw-r--r-- 1 jenkins jenkins   82 Nov 23 14:56 package-lock.json
-rw-r--r-- 1 jenkins jenkins  436 Nov 23 14:56 package.json
root@ip-172-31-5-227:/var/lib/jenkins/workspace/nodejs-hello-app# 
```

vii. Copy to directory -build Workspace - /var/www/nodeapp (before doing this step we need to do step-9 for NGINX webserver installation)
Permission for jenkins to copy dir in server machines.

1) Now we have to create one directory.
   a) Ls
   b) Mkdir nodeapp
   c) Ls

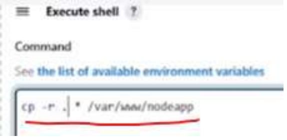


```
root@ip-172-31-5-227:/var/www# ls
html
root@ip-172-31-5-227:/var/www# mkdir nodeapp
root@ip-172-31-5-227:/var/www# ls
html  nodeapp
root@ip-172-31-5-227:/var/www# clear
```



```
root@ip-172-31-5-227:/var/www# ll
total 16
drwxr-xr-x  4 root root 4096 Nov 23 15:06 ./
drwxr-xr-x 14 root root 4096 Nov 23 15:02 ../
drwxr-xr-x  2 root root 4096 Nov 23 15:02 html/
drwxr-xr-x  2 root root 4096 Nov 23 15:06 nodeapp/
root@ip-172-31-5-227:/var/www# 
```

2) Now we have to give nodeapp path in jenkins and build but build faild because nodeapp have permission issue.



Execute shell

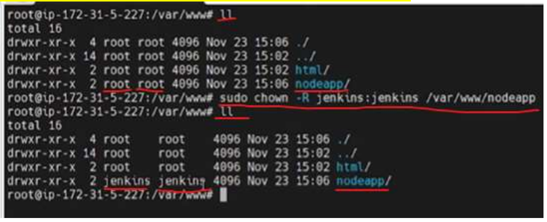
Command

See the list of available environment variables

```
cp -r ./* /var/www/nodeapp
```

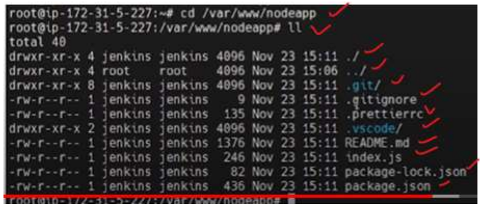
3) Now we have to give jenkin:jenkins permission to nodeapp using below command.

sudo chown -R jenkins:jenkins /var/www/nodeapp



```
root@ip-172-31-5-227:/var/www# ll
total 16
drwxr-xr-x  4 root root 4096 Nov 23 15:06 ./
drwxr-xr-x 14 root root 4096 Nov 23 15:02 ../
drwxr-xr-x  2 root root 4096 Nov 23 15:02 html/
drwxr-xr-x  2 root root 4096 Nov 23 15:06 nodeapp/
root@ip-172-31-5-227:/var/www# sudo chown -R jenkins:jenkins /var/www/nodeapp
root@ip-172-31-5-227:/var/www# ll
total 16
drwxr-xr-x  4 root    root    4096 Nov 23 15:06 ./
drwxr-xr-x 14 root    root    4096 Nov 23 15:02 ../
drwxr-xr-x  2 root    root    4096 Nov 23 15:02 html/
drwxr-xr-x  2 jenkins jenkins 4096 Nov 23 15:06 nodeapp/
root@ip-172-31-5-227:/var/www#
```

4) Now we have to build again and see our source code is present in nodeapp.



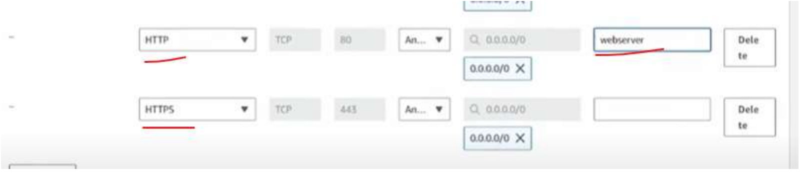
```
root@ip-172-31-5-227:~# cd /var/www/nodeapp
root@ip-172-31-5-227:/var/www/nodeapp# ll
total 40
drwxr-xr-x 4 jenkins jenkins 4096 Nov 23 15:11 ./
drwxr-xr-x 4 root    root    4096 Nov 23 15:06 ../
drwxr-xr-x 8 jenkins jenkins 4096 Nov 23 15:11 .git/
-rw-r--r-- 1 jenkins jenkins    9 Nov 23 15:11 .gitignore
-rw-r--r-- 1 jenkins jenkins  135 Nov 23 15:11 .prettierrc
drwxr-xr-x 2 jenkins jenkins 4096 Nov 23 15:11 .vscode/
-rw-r--r-- 1 jenkins jenkins 1376 Nov 23 15:11 README.md
-rw-r--r-- 1 jenkins jenkins  246 Nov 23 15:11 index.js
-rw-r--r-- 1 jenkins jenkins   82 Nov 23 15:11 package-lock.json
-rw-r--r-- 1 jenkins jenkins  436 Nov 23 15:11 package.json
root@ip-172-31-5-227:/var/www/nodeapp#
```

5)

## 9. NGINX webserver installation. = done

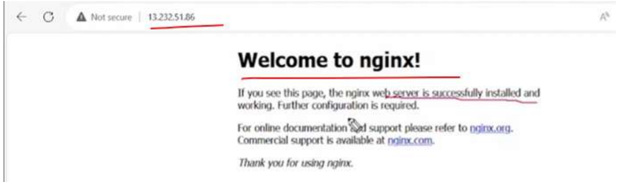
i. Ask this to chatgpt = installation of Nginx webserver on ubuntu os.

1) sudo apt update = update the package list.
2) sudo apt install nginx = install nginx
3) sudo service nginx start = start nginx
4) sudo systemctl enable nginx = Enable Nginx to start to Boot
5) Now we have to validate the nginx is working or not.
   a) Copy the ec2 public ip and paste on web page but before doing we have to set the port number in security inbound rules.



HTTP ▼ TCP 80 An... ▼ 0.0.0.0/0 webserver Dele te
0.0.0.0/0 ✕
HTTPS ▼ TCP 443 An... ▼ 0.0.0.0/0 Dele te
0.0.0.0/0 ✕

   b) Now refresh the page and found that our nginx webserver is running successfully.



Not secure | 13.232.51.86

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

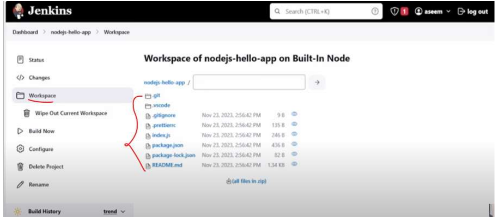
*Thank you for using nginx.*

   c) cc

6) cc

## 10. Creating builds = done

No need to do again it already done above



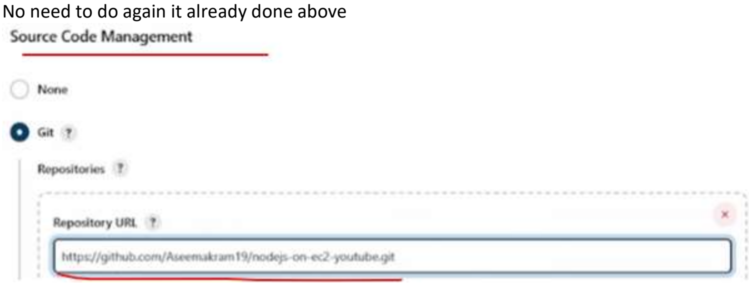
Workspace of nodejs-hello-app on Built-In Node

nodejs-hello-app /

.git
.vscode
.gitignore Nov 23, 2023, 2:56:42 PM 9 B
.prettierrc Nov 23, 2023, 2:56:42 PM 135 B
index.js Nov 23, 2023, 2:56:42 PM 246 B
package.json Nov 23, 2023, 2:56:42 PM 436 B
package-lock.json Nov 23, 2023, 2:56:42 PM 82 B
README.md Nov 23, 2023, 2:56:42 PM 1.34 KB
(all files in zip)

## 11. Source code @ GitHub = done

No need to do again it already done above



Source Code Management

○ None

● Git

Repositories

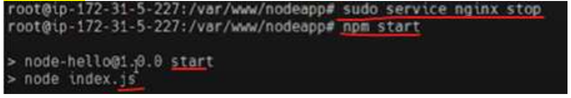
Repository URL

https://github.com/Aseemakram19/nodejs-on-ec2-youtube.git

## 12. Readme = If you stuck any step the go to readme file to see all step. = done

## 13. NPM Install = done

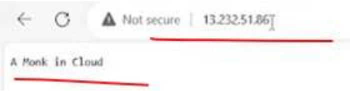
i. **npm start** = before start the npm we need to stop the nginx first (**sudo service nginx stop**)



```
root@ip-172-31-5-227:/var/www/nodeapp# sudo service nginx stop
root@ip-172-31-5-227:/var/www/nodeapp# npm start

> node-hello@1.0.0 start
> node index.js
```

ii. Now npm is started then copy the ec2 public ip and paste on web.



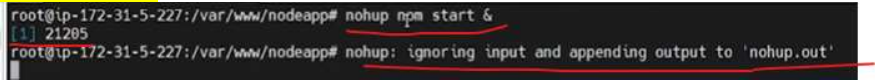
Not secure | 13.232.51.86

A Monk in Cloud

iii. How to run npm in background.

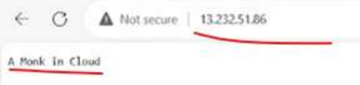
1) Npm run in background using below command.

   **nohup npm start &**



```
root@ip-172-31-5-227:/var/www/nodeapp# nohup npm start &
[1] 21205
root@ip-172-31-5-227:/var/www/nodeapp# nohup: ignoring input and appending output to 'nohup.out'
```

2) After npm start we have to validate using refresh the ec2 public link in web page.



Not secure | 13.232.51.86

A Monk in Cloud

3) Kill all npm using below command.

   **pkill -9 -f "npm"**

4) cc

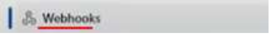
iv. cc

## 14. GitHub-webhooks auto deployment =

**i. How to create webhook on github.**

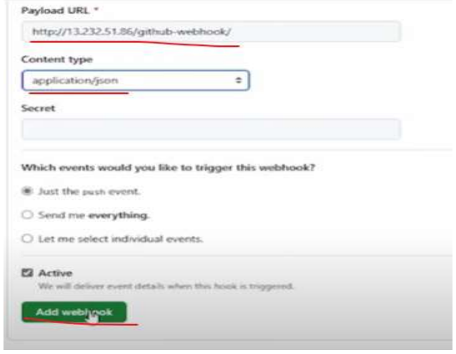
**1) Goto github.**

**2) Click on webhook.**



**3) Click on add webhook.**

**4) Copy the ec2 public ip and paste in payload URL in github.**



Payload URL *

http://13.232.51.86/github-webhook/

Content type

application/json

Secret

Which events would you like to trigger this webhook?

● Just the push event.

○ Send me everything.

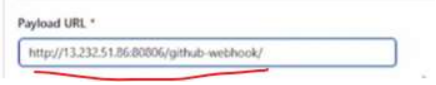
○ Let me select individual events.

☑ Active

We will deliver event details when this hook is triggered.

Add webhook

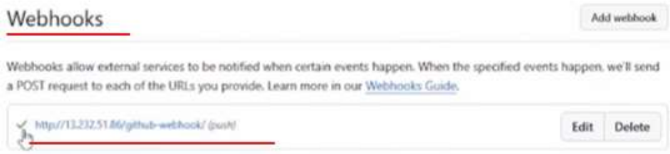
If we get error then we do this.



Payload URL *

http://13.232.51.86:80806/github-webhook/

**5) Now webhook is created.**



Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.232.51.86/github-webhook/ (push)

Edit Delete

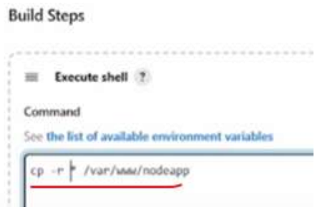
**6) Now we have to install few plugin in jenkins like below.**

**a) Generic Webhook Trigger**

**b) Git server**

**c) GitHub Integration**

**7) Now we have to build now the jenkins but it's faild so we need to fix it go to below step to remove dot(.) and build again.**



Build Steps

Execute shell ?

Command

See the list of available environment variables

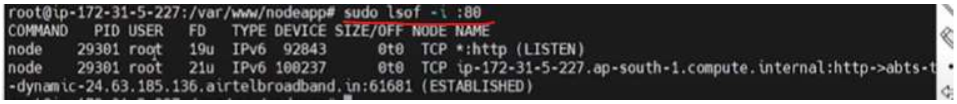
```
cp -r  /var/www/nodeapp
```

**8) Now refresh the page and still it's runing.**

## 15. Now we have to check the 80 port how many pid is runing.

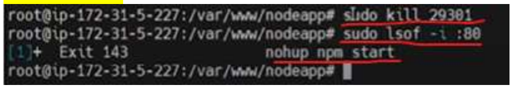
**i. Check PID**

sudo lsof -i :80



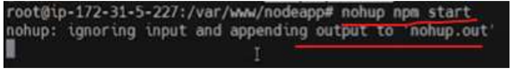
```
root@ip-172-31-5-227:/var/www/nodeapp# sudo lsof -i :80
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
node    29301 root   19u  IPv6  92843      0t0  TCP *:http (LISTEN)
node    29301 root   21u  IPv6 100237      0t0  TCP ip-172-31-5-227.ap-south-1.compute.internal:http->abts-t
-dynamic-24.63.185.136.airtelbroadband.in:61681 (ESTABLISHED)
```

ii. **Now we have to kill this pid using below command**
Sudo kill 29301
sudo lsof -I :80



```
root@ip-172-31-5-227:/var/www/nodeapp# sudo kill 29301
root@ip-172-31-5-227:/var/www/nodeapp# sudo lsof -i :80
[1]+  Exit 143                nohup npm start
root@ip-172-31-5-227:/var/www/nodeapp# 
```

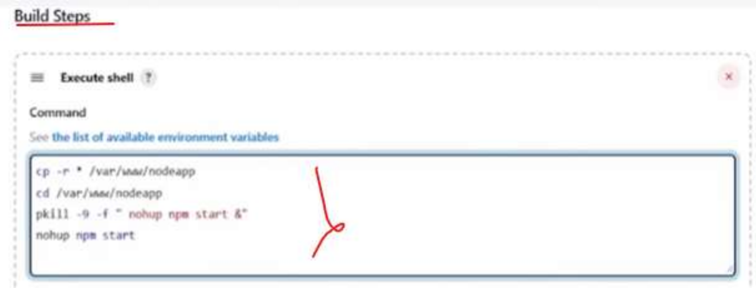
iii. **Now we refresh the page and it's not running . Now we are enable this application using below command.**
nohup npm start



```
root@ip-172-31-5-227:/var/www/nodeapp# nohup npm start
nohup: ignoring input and appending output to 'nohup.out'
```

iv. **Now refesh the webpage and it's working application.**

16. **Now we have to automate this. So goto jenkins and write below command in build step tab and save and build now and refresh web page.**



Build Steps

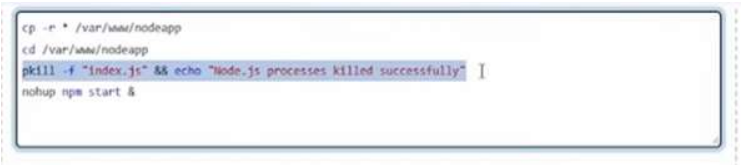
Execute shell

Command

See the list of available environment variables

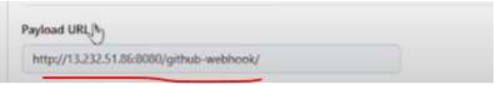
```
cp -r * /var/www/nodeapp
cd /var/www/nodeapp
pkill -9 -f " nohup npm start &"
nohup npm start
```

cp -r * /var/www/nodeapp
cd /var/www/nodeapp
pkill -f "index.js" && echo "node.js process killed successfully"
nohup npm start &



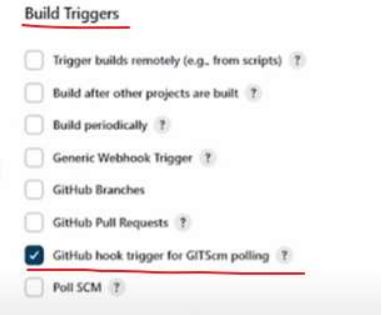
```
cp -r * /var/www/nodeapp
cd /var/www/nodeapp
pkill -f "index.js" && echo "Node.js processes killed successfully"
nohup npm start &
```

17. **Now every thing working fine but we need to validate whether my pipeline is automate or not.**
    i. Before change we need to update the webhook.



Payload URL
http://13.232.51.86:8080/github-webhook/

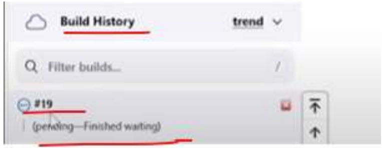
ii. Now we have to enable the Github Hook and save.



Build Triggers

☐ Trigger builds remotely (e.g., from scripts)
☐ Build after other projects are built
☐ Build periodically
☐ Generic Webhook Trigger
☐ GitHub Branches
☐ GitHub Pull Requests
☑ GitHub hook trigger for GITScm polling
☐ Poll SCM

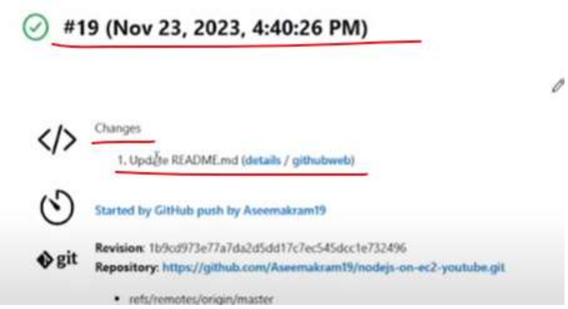
iii. For that go to github source code and goto readme file and try to change some thing and commit there.
iv. Once we change I can see that jenkins automatic build the project and change I can see in jenkins workspace readme file.



Build History trend

Filter builds...

#19
(pending—Finished waiting)



#19 (Nov 23, 2023, 4:40:26 PM)

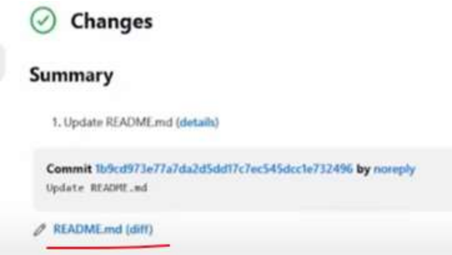
Changes

1. Update README.md (details / githubweb)

Started by GitHub push by Aseemakram19

Revision: 1b9cd973e77a7da2d5dd17c7ec545dcc1e732496
Repository: https://github.com/Aseemakram19/nodejs-on-ec2-youtube.git

- refs/remotes/origin/master



Changes

Summary

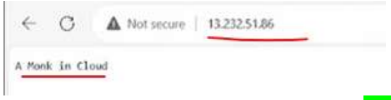
1. Update README.md (details)

Commit 1b9cd973e77a7da2d5dd17c7ec545dcc1e732496 by noreply
Update README.md

README.md (diff)

v. Now we can see that if we change any thing in our giyhub source code it's automatic build and change made on jenkins workspace.

vi. Now this is the process to  NODE.js App deployment with Jenkins CI CD pipeline|Github Repo auto trigger webhook succesfully.



A Monk in Cloud

Now build run successfully.

1. After project run successfully then we have to terminate the ec2 instance and elastic ip.