

CI/CD part 1

Create and push both service source code to github

Launch EC2 service and install all necessary things

-sudo apt install python3 python3-pip -y

-curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install nodejs -y

-sudo apt install git -y

Clone both repo and start both service
frontend on 3000 port and backend on 8000 port

```
MINGW64 ~/vpracice/CI/C1
$ express-frontend default 1.0.0 Port 0 0 0 stopped ON ON abarts abarts available disabled
$ express-frontend default N/A Port 18230 19m 0 online ON ON abarts abarts available disabled
$ Flask-backend default N/A Port 18230 19m 0 stopped ON ON abarts abarts available disabled

[0] Applying action deleteProcessId on app [express-frontend] (ids: [ 0, 2, 3 ])
[0] [express-frontend][0]
[0] [express-frontend][2]
[0] [express-frontend][3]

[1] id name tomcatversion version minid pid upTime u status cpu mem user watching
[1] 3 Flask-backend default N/A Port 18230 19m 0 online ON ON abarts abarts available disabled

ubuntu@ip-172-31-5-181:~/Frontend-node.js$ cd ..
ubuntu@ip-172-31-5-181:~$ ls
Backend-Flask Frontend-node.js
ubuntu@ip-172-31-5-181:~$ rm -rf Frontend-node.js/
ubuntu@ip-172-31-5-181:~$ git clone https://github.com/kumarpradeep78/Frontend-node.js.git
Cloning into 'Frontend-node.js'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (16/16), done.
remote: Writing objects: 100% (27/27), 12.37 KiB | 12.37 MiB/s, done.
Resolving deltas: 100% (4/4), done.
ubuntu@ip-172-31-5-181:~/Frontend-node.js$ npm start
[0] [error] Script not found: /home/ubuntu/app.js
ubuntu@ip-172-31-5-181:~/Frontend-node.js$ 
ubuntu@ip-172-31-5-181:~/Frontend-node.js$ npm install
added 71 packages, and audited 74 packages in 98ms

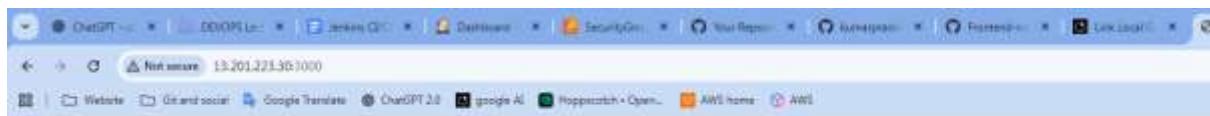
22 packages are funding for funding
  run `npm fund` for details

Found 0 vulnerabilities
ubuntu@ip-172-31-5-181:~/Frontend-node.js$ pm2 start app.js --name express-frontend -f
[0] [0] Starting /home/ubuntu/Frontend-node.js/app.js in fork_mode (0) instance
[0] [0] Done.

[1] id name tomcatversion version minid pid upTime u status cpu mem user watching
[1] 3 express-frontend default 1.0.0 Port 0 0 online ON ON abarts abarts available disabled
[1] 3 Flask-backend default N/A Port 18230 20m 0 online ON ON abarts abarts available disabled

ubuntu@ip-172-31-5-181:~/Frontend-node.js$ Read from socket: host ec2-3-201-223-30.ap-south-1.compute.amazonaws.com Connection reset by peer
Connection to ec2-3-201-223-30.ap-south-1.compute.amazonaws.com closed.
Client:Usage: send disconnect: Connection reset by peer
ubuntu@ip-172-31-5-181:~/Frontend-node.js$
```

both run and we also able to connect it with our local using public IP



PART 2

Install JDK(java 17) and Jenkins

1. sudo apt install openjdk-17-jdk -y
2. curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

3. echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
- https://pkg.jenkins.io/debian binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

4. sudo apt update

sudo apt install jenkins -y

5. sudo systemctl start jenkins

sudo systemctl enable jenkins

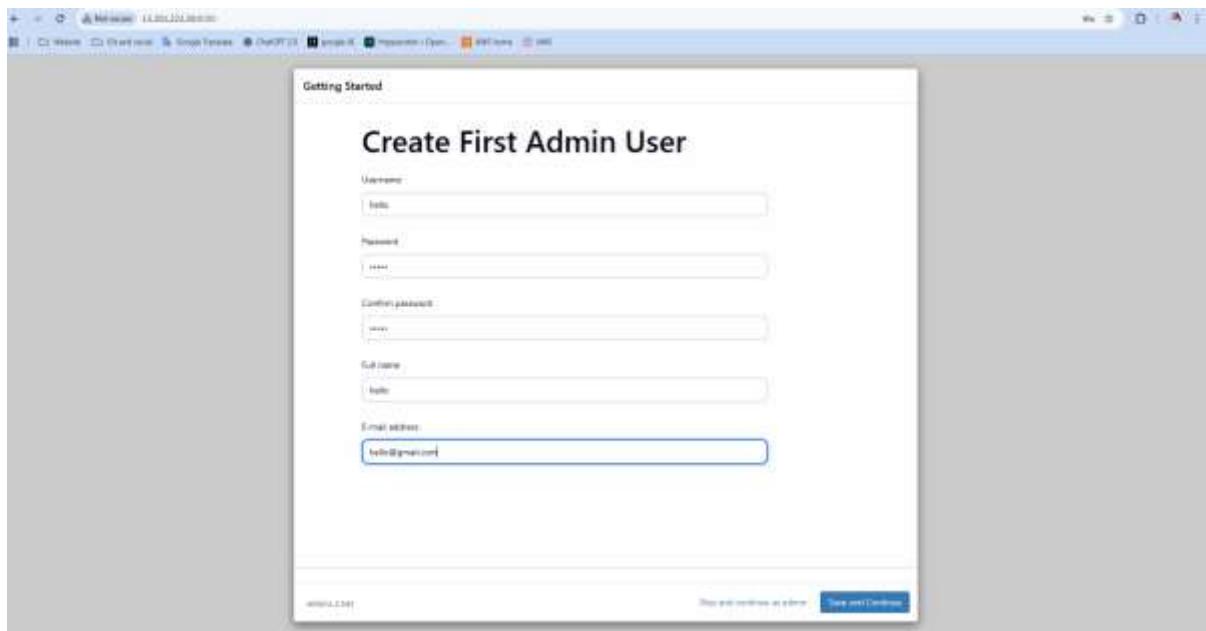
```
sudo systemctl status Jenkins
```

[these cmd used to start and enable auto start when you restart EC2 service]

6. Now access it with your local browser
[publicip:8080]

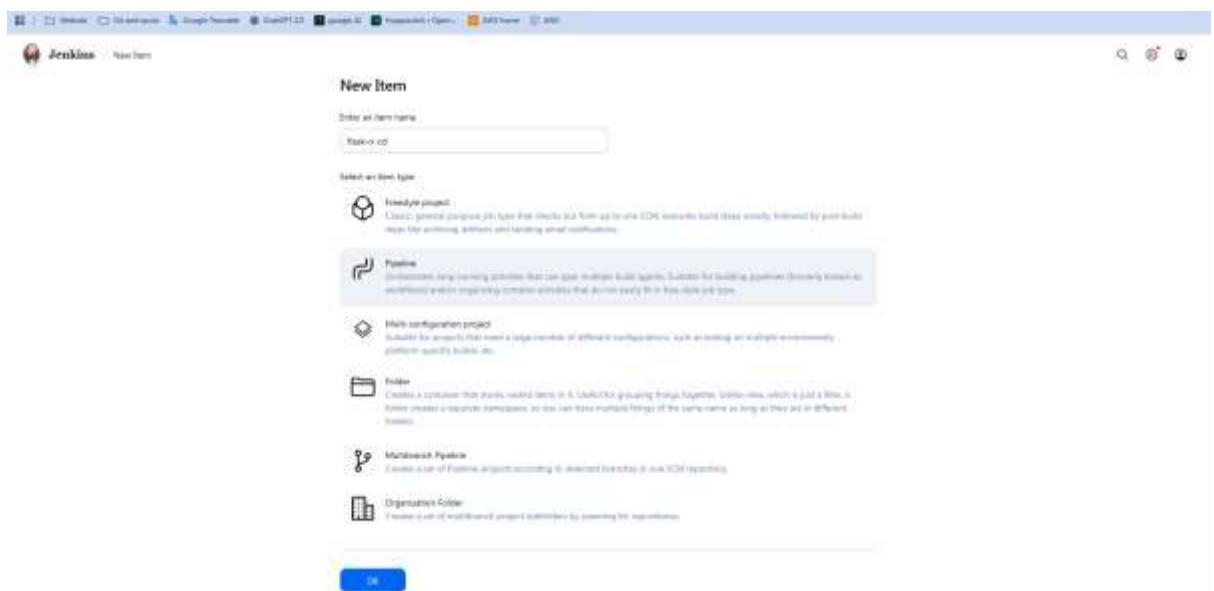


7. sudo cat /var/lib/jenkins/secrets/initialAdminPassword
copy that password and use it.



After that create your first user and password.

- ##Now we start to create pipeline (Backend Flask)



After this add script to this(what to do)

After this add webhook to git hub

Settings → Webhooks → Add Webhook

- Payload URL: `http://<EC2_PUBLIC_IP>:8080/github-webhook/`
- Content type: `application/json`
- Event: **Push**

General

Events

Webhooks **Add webhook**

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, XML, YAML, etc). More information can be found in our [developer documentation](#).

Payload URL *
http://13.201.221.30:8080/github-webhook/

Content type *
application/json

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.

Enable SSL verification Disable (not recommended)

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

Not push new commit .

After push github trigger Jenkins and Jenkins run script

User Form

Enter user name
Enter user email
Enter user password
Submit
View Test

Jenkins Build History

New Item Build History

ID	Build	Time Since	Status
1	jenkins-build	13 min	Back to normal
2	jenkins-build	1 hr 13 min	Failed for a long time
3	jenkins-build	1 hr 18 min	Failed since that build

After 2 failur 3 build successful

and our service also start

build console output status.