

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 ~/workspace/CI CD
0 express-frontend default 1.0.0 0 0 0 stopped 0% 0b ubuntu disabled
2 express-frontend default 1.0.0 0 0 0 stopped 0% 0b ubuntu disabled
3 flask-backend default N/A 18230 13m 0 0 0 0% 11.3mb ubuntu disabled

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

id name namespace version mode pid uptime s status cpu mem user watching
3 flask-backend default N/A 18230 13m 0 online 0% 11.3mb ubuntu disabled

ubuntu@ip-172-31-5-169:~/Frontend-node.js$ cd ..
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ ls
Backend-Flask frontend-node.js
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ cd ..
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ git clone https://github.com/kumarpradeep78/Frontend-node.js.git
Cloning into 'Frontend-node.js'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 22 (delta 4), reused 19 (delta 1), push-reused 0 (from 0)
Receiving objects: 100% (22/22), 12.37 KiB | 12.37 MiB/s, done.
Resolving deltas: 100% (4/4), done.
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ npm start app.js --name express-frontend -f
[aws] [color] Script not found: /home/ubuntu/app.js
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ cd Frontend-node.js/
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ npm install

added 73 packages, and audited 74 packages in 581ms

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

found 0 vulnerabilities
ubuntu@ip-172-31-5-169:~/Frontend-node.js$ npm start app.js --name express-frontend -f
[aws] Starting /home/ubuntu/Frontend-node.js/app.js in fork_mode (3 instance)
[aws] Done.

id name namespace version mode pid uptime s status cpu mem user watching
3 express-frontend default 1.0.0 18770 9s 0 online 0% 11.5mb ubuntu disabled
3 flask-backend default N/A 18230 20m 0 0 0 0% 11.3mb ubuntu disabled

ubuntu@ip-172-31-5-169:~/Frontend-node.js$ Read from remote host ec2-11-201-223-30.ap-south-1.compute.amazonaws.com: Connection reset by peer
Connection to ec2-11-201-223-30.ap-south-1.compute.amazonaws.com closed.
Client:loop: send disconnect: Connection reset by peer

MINGW64 ~/workspace/CI CD
```

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

The screenshot shows a web browser window with a 'User Form'. The browser's address bar displays '13.201.221.36:1000'. The form contains three text input fields labeled 'Enter your name', 'Enter your email', and 'Enter your password'. Below these fields are two buttons: 'Submit' and 'View Data'.

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`

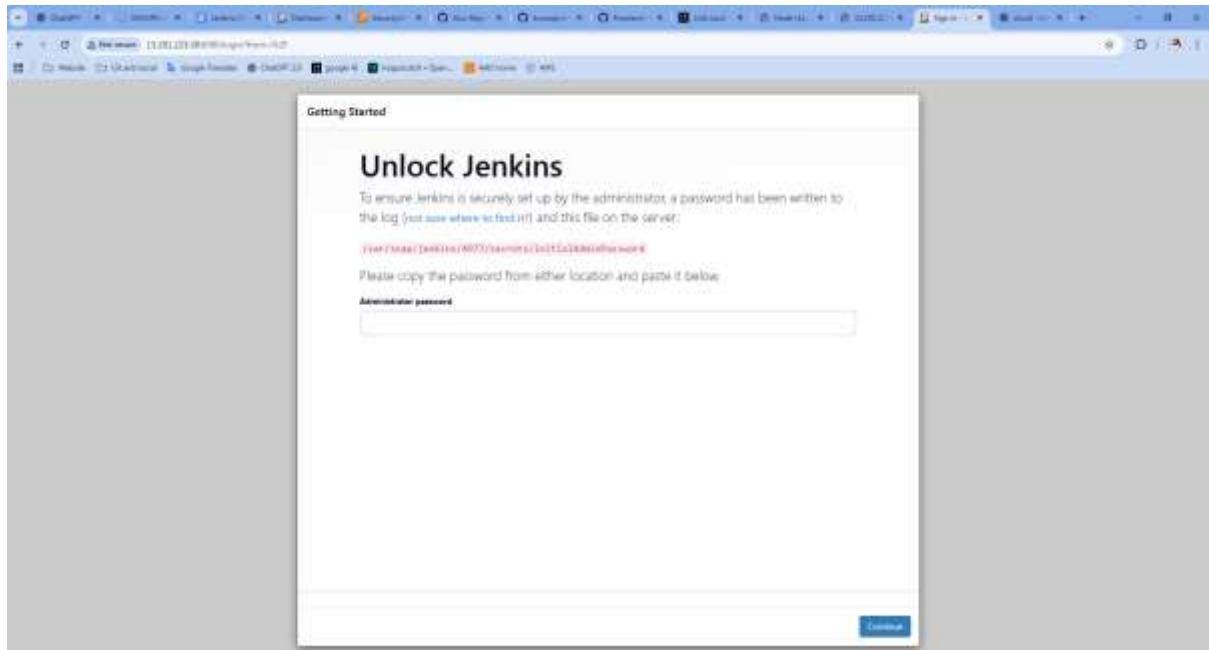
`sudo systemctl enable jenkins`

`sudo systemctl status Jenkins`

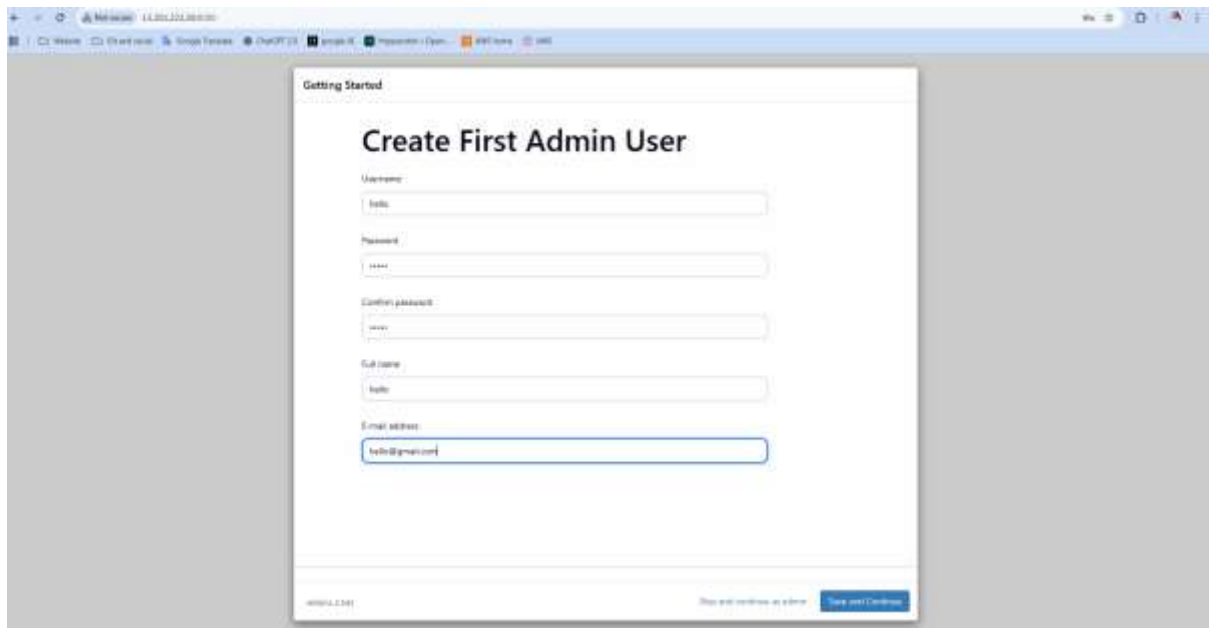
[these cmd used to start and enable auto start when you restart EC2 servic]

6. Now access it with your local browser

[publicip:8080]

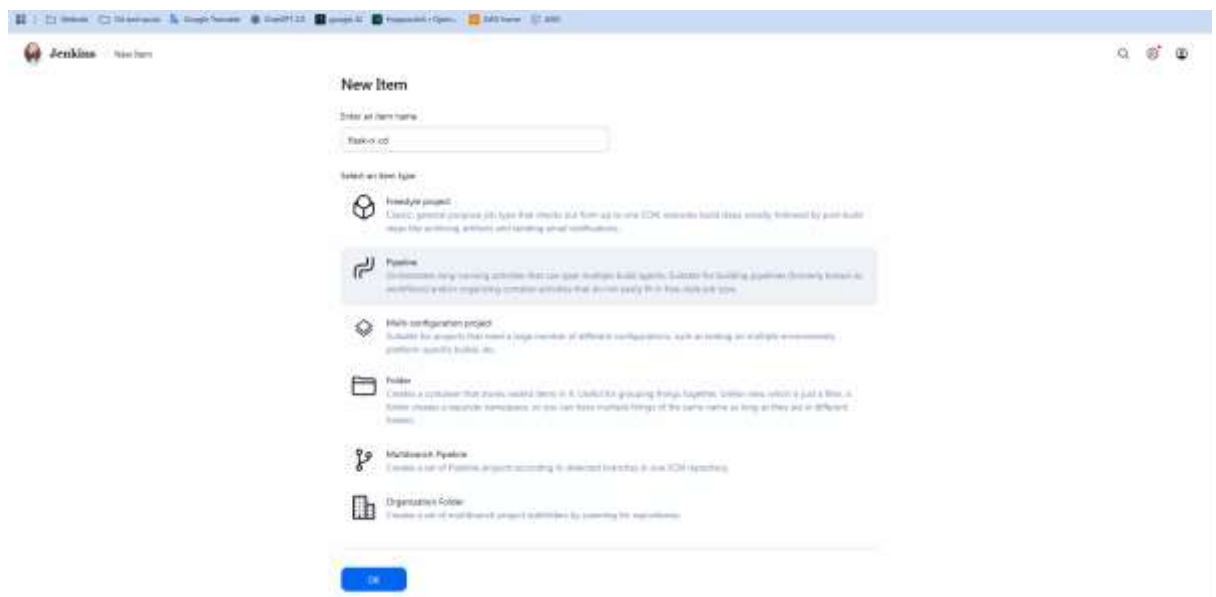


7. `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`
copy that password and use it.



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**

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, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

Payload URL *

Content type *

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 your name:

Enter your email:

Enter your password:

Jenkins Build History

#	Build	Time	Status
41	jenkins-build	12 min	Success
40	jenkins-build	1 hr 13 min	Failure
39	jenkins-build	1 hr 18 min	Failure

After 2 failur 3 build successful

and our service also start



The screenshot shows the Jenkins console output for a job named 'express-05-cd'. The output displays the PM2 daemon starting successfully, initializing the environment, and starting the 'express-frontend' process. A table lists the running process details, and the pipeline concludes with a 'SUCCESS' status.

```
+ [32m[PM2] + [30mSpawning PM2 daemon with pm2_home=/etc/.pm2
+ [32m[PM2] + [30mPM2 Successfully daemonized
+ [3m[34mUse --update-env to update environment variables+ [39m[22m
+ [31m[PM2][ERR00] + [36mProcess or Namespace express-frontend not found
+ pm2 start app.js --name express-frontend
[PM2][Initialization] Environment variable HOME (Linux) or HOMEPATH (Windows) are not set!
[PM2][Initialization] Defaulting to /etc/.pm2
+ [32m[PM2] + [30mStarting /var/snap/jenkins/4877/workspace/express-05-cd/app.js in fork_mode (1 instance)
+ [32m[PM2] + [30mDone.
```

id	name	namespace	version	mode	pid	uptime	♻	status	cpu	mem	user
0	express-frontend	default	1.0.0	fork	1827	0s	0	online	0s	0k	root

```
+ [32m[inline+ [22m[30m + [35m 23.9mb + [31mroot+ [22m + [30mdisabled+ [30m
+ [32m[PM2] + [30mDone.
```

```
[Pipeline] }
[Pipeline] // within
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // within
[Pipeline] }
[Pipeline] // within
[Pipeline] }
[Pipeline] // node
[Pipeline] // test of #pipeline
Finished: SUCCESS
```

build console output status.