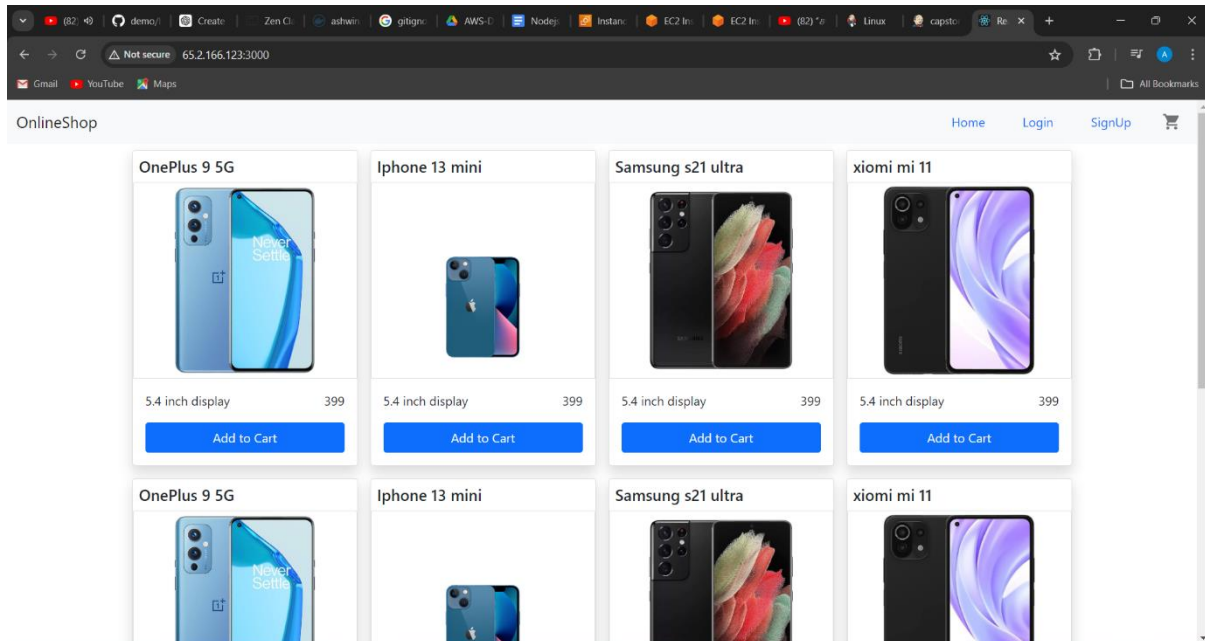


1.GITHUB link

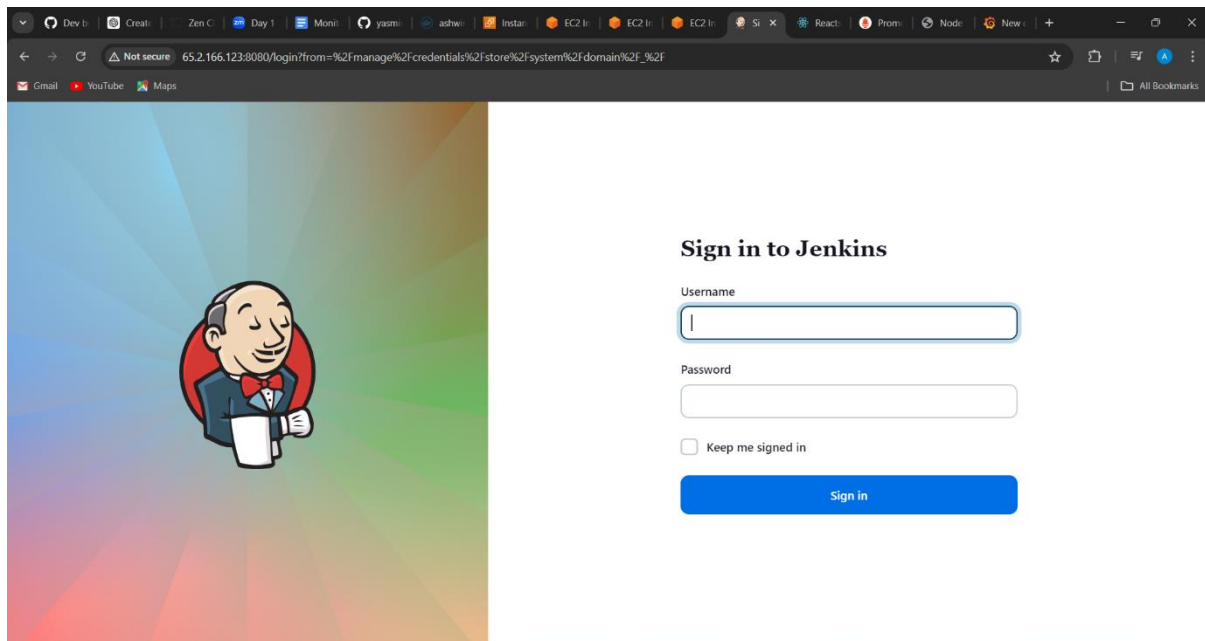
<https://github.com/ashwin31081999/demo>

2.Application can be accessed by the ip address

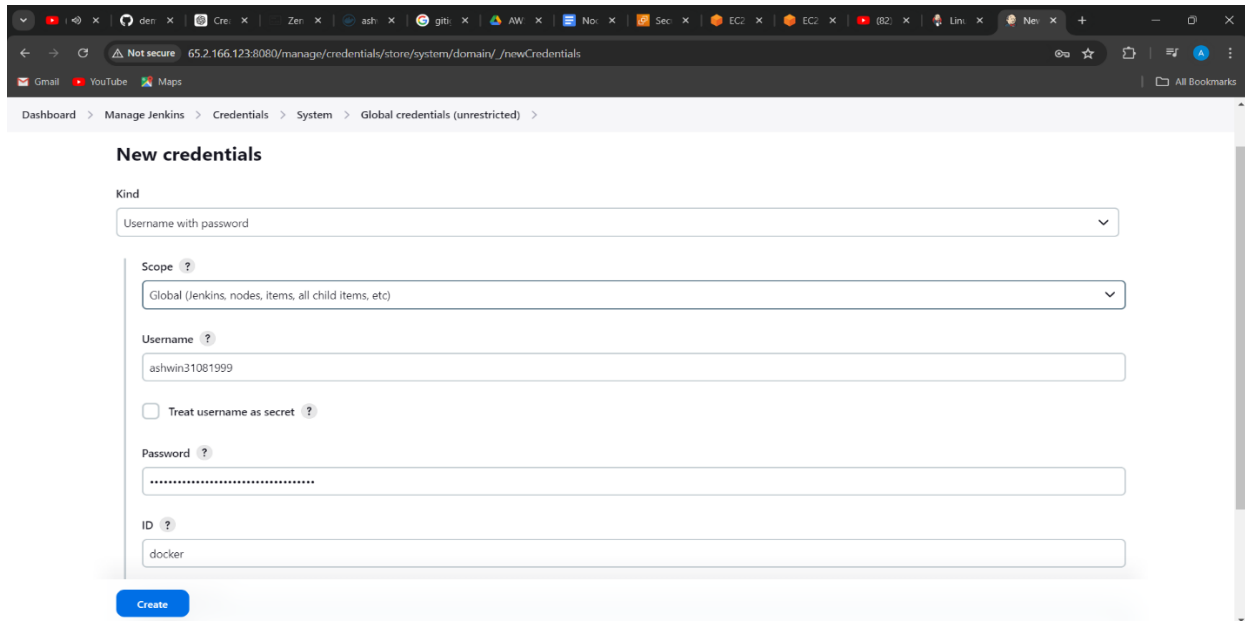
<http://65.2.166.123:3000/>



3.Jenkins login page



4. Docker hub credentials added in Jenkins.



The screenshot shows the 'New credentials' form in Jenkins. The browser address bar indicates the URL: 65.2.166.123:8080/manage/credentials/store/system/domain/_/newCredentials. The breadcrumb trail is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >. The form fields are: Kind (Username with password), Scope (Global (Jenkins, nodes, items, all child items, etc)), Username (ashwin31081999), Password (masked with asterisks), ID (docker), and a checkbox for 'Treat username as secret' which is unchecked. A 'Create' button is at the bottom left.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Username with password

Scope: Global (Jenkins, nodes, items, all child items, etc)

Username: ashwin31081999

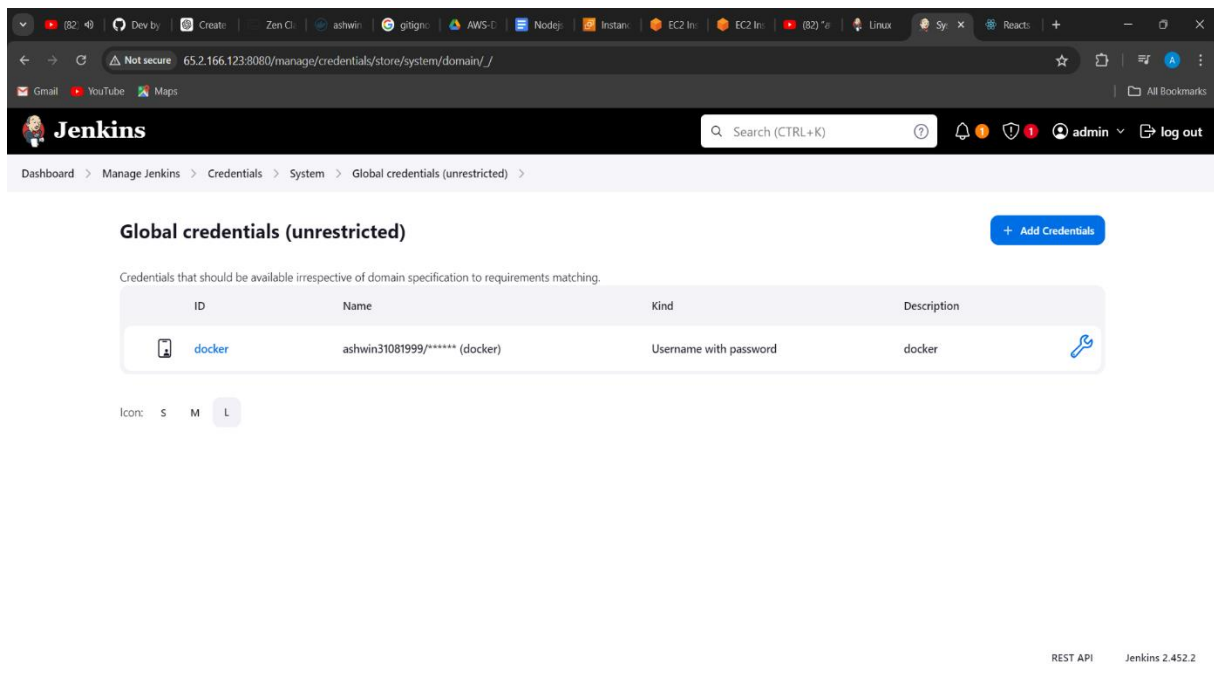
☐ Treat username as secret

Password: *****

ID: docker

Create

5. Global credential.





The screenshot shows the 'Global credentials (unrestricted)' page in Jenkins. The browser address bar indicates the URL: 65.2.166.123:8080/manage/credentials/store/system/domain/_/. The breadcrumb trail is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >. The page title is 'Global credentials (unrestricted)' with an '+ Add Credentials' button. Below the title, it says 'Credentials that should be available irrespective of domain specification to requirements matching.' There is a table with 4 columns: ID, Name, Kind, and Description. The table contains one entry: ID 'docker', Name 'ashwin31081999/***** (docker)', Kind 'Username with password', and Description 'docker'. At the bottom right of the table is a wrench icon. Below the table, there are filters for 'Icon', 'S', 'M', and 'L', with 'L' selected. The footer shows 'REST API' and 'Jenkins 2.452.2'.

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 docker	ashwin31081999/***** (docker)	Username with password	docker 

Icon: S M L

REST API Jenkins 2.452.2

6.Environmental variables added -1

The screenshot shows the Jenkins 'Manage Jenkins' > 'System' configuration page. The 'Environment variables' section is checked. It contains a 'List of variables' with two entries:

Name	Value
DOCKER_BFLASK_IMAGE1	ashwin31081999/prod
DOCKER_BFLASK_IMAGE2	ashwin31081999/dev

At the bottom of the section are 'Save' and 'Apply' buttons.

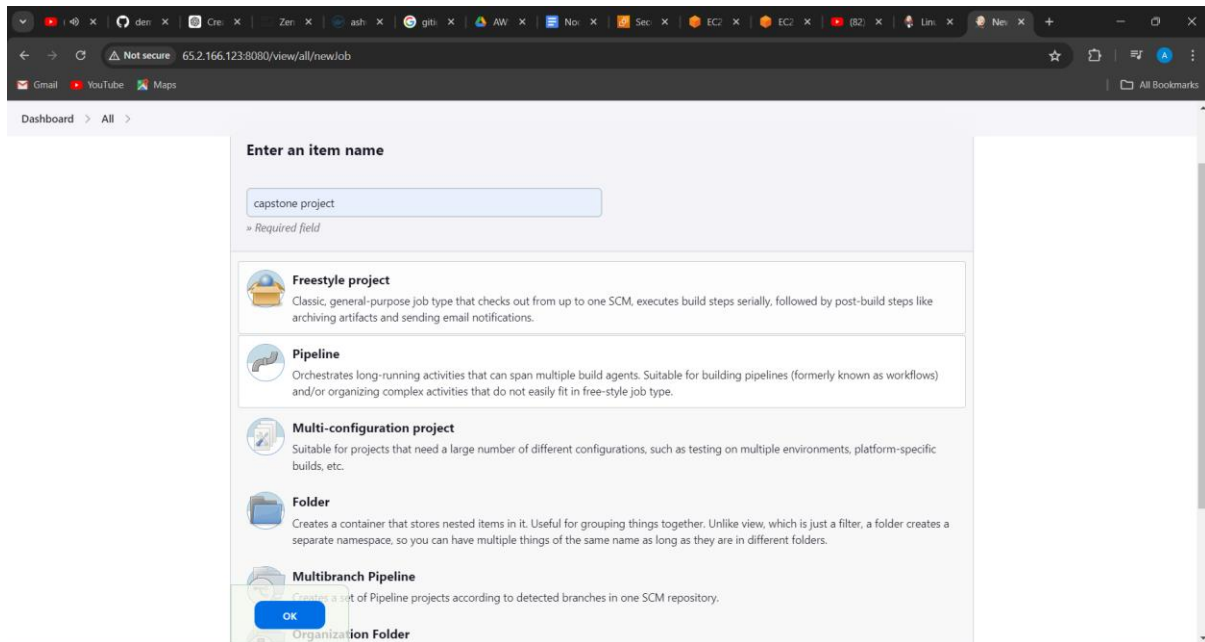
7. Environmental variables added -2

This screenshot shows the same Jenkins 'Manage Jenkins' > 'System' configuration page, but with two additional environment variables added:

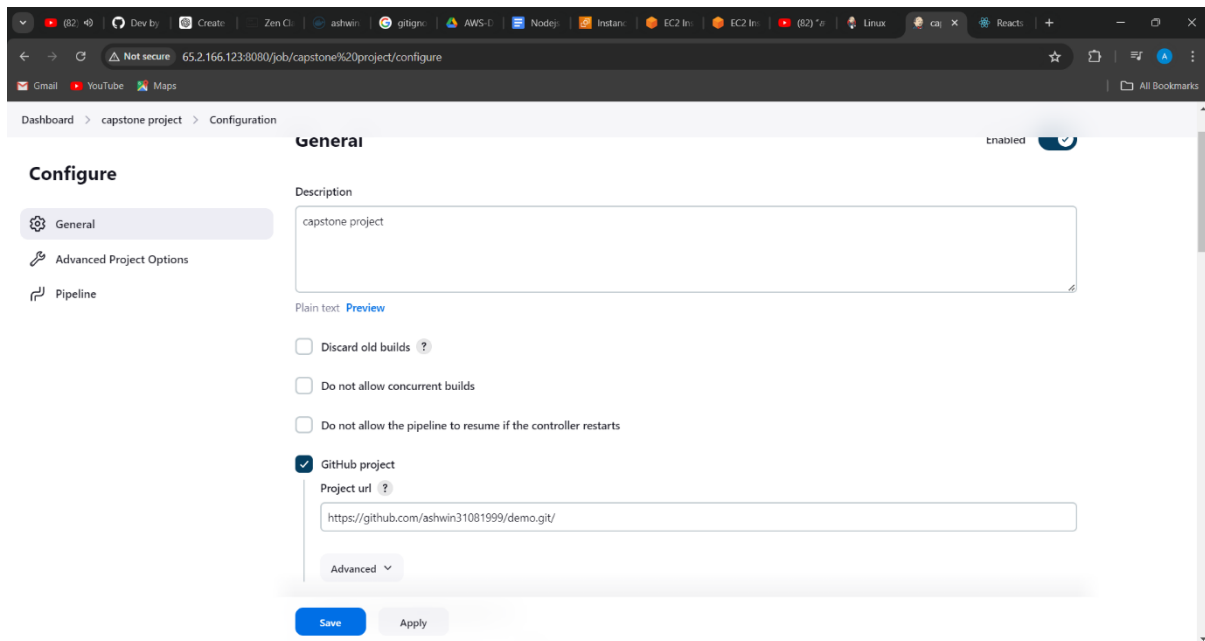
Name	Value
DOCKER_USER	ashwin31081999
DOCKER_PASS	dckr_pat_m-4FR8ZvjhSX6by0uMLVVCigoX8

The 'Save' and 'Apply' buttons remain at the bottom.

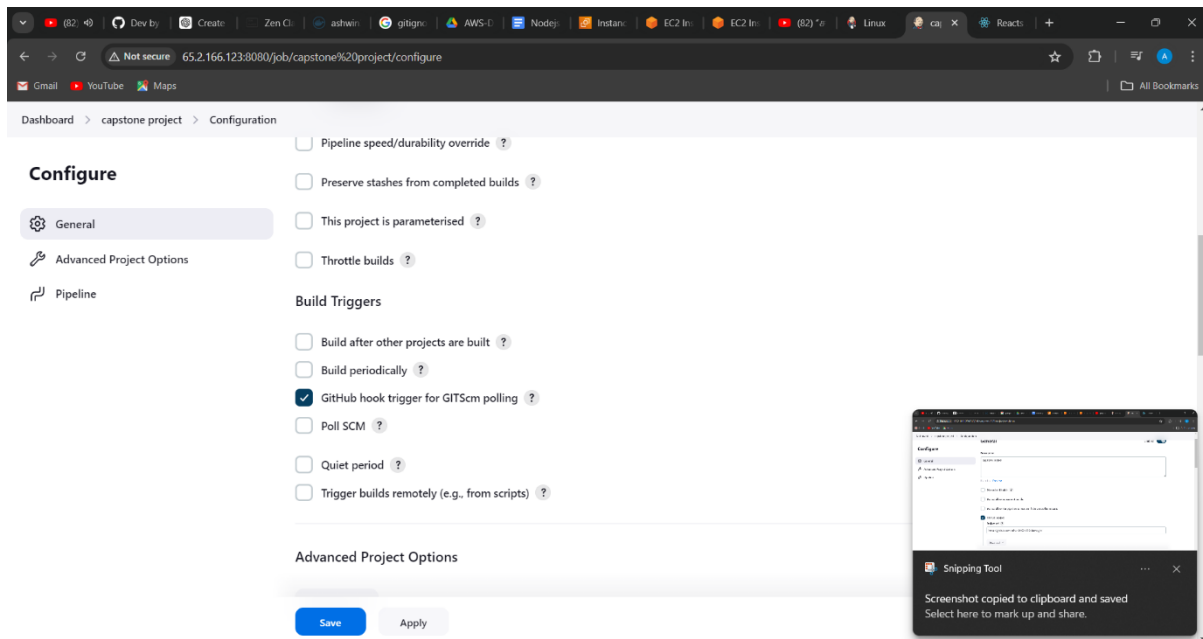
8.Created a pipeline project in Jenkins.



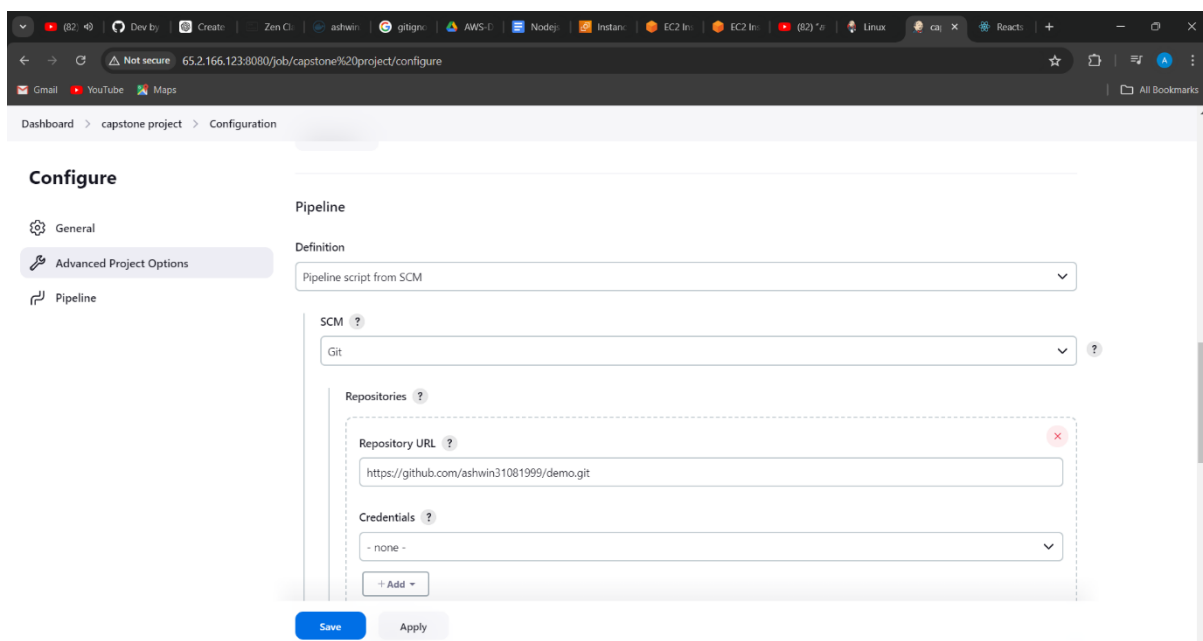
9.Configuring the pipeline-1



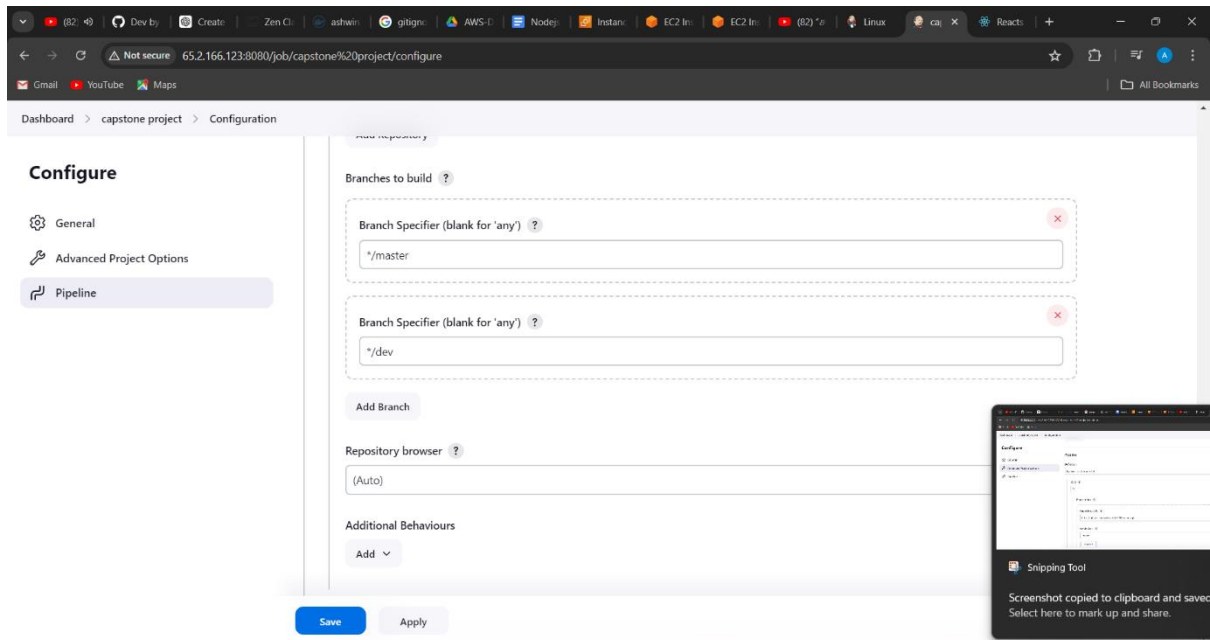
10. Configuring the pipeline-2



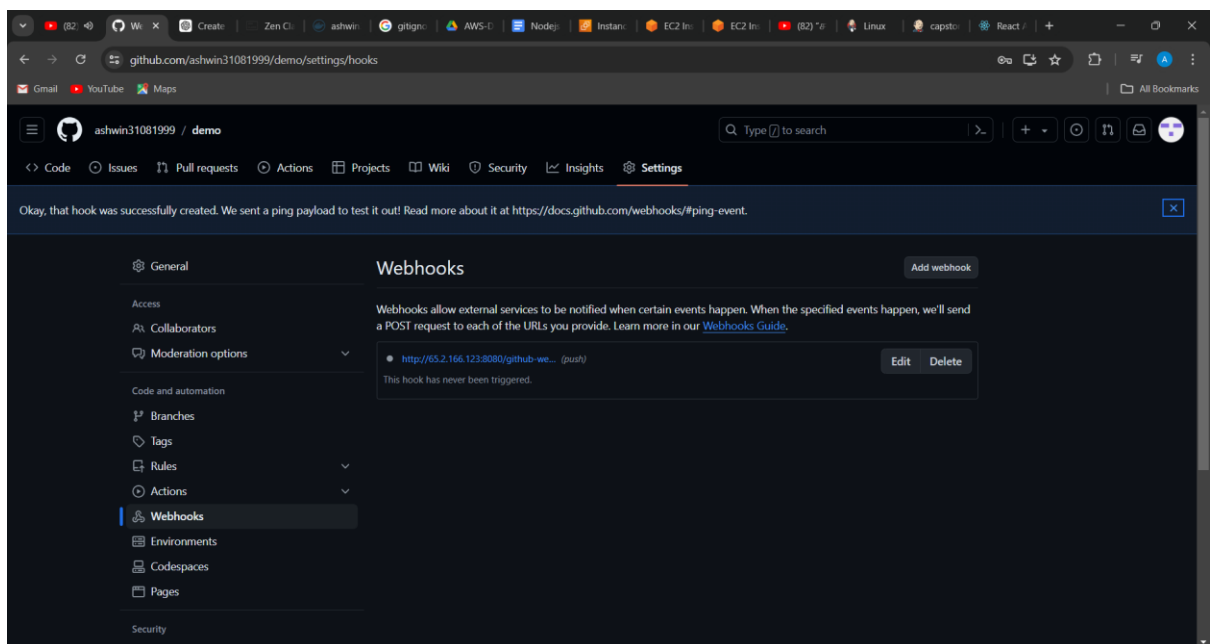
11. Configuring the pipeline-3



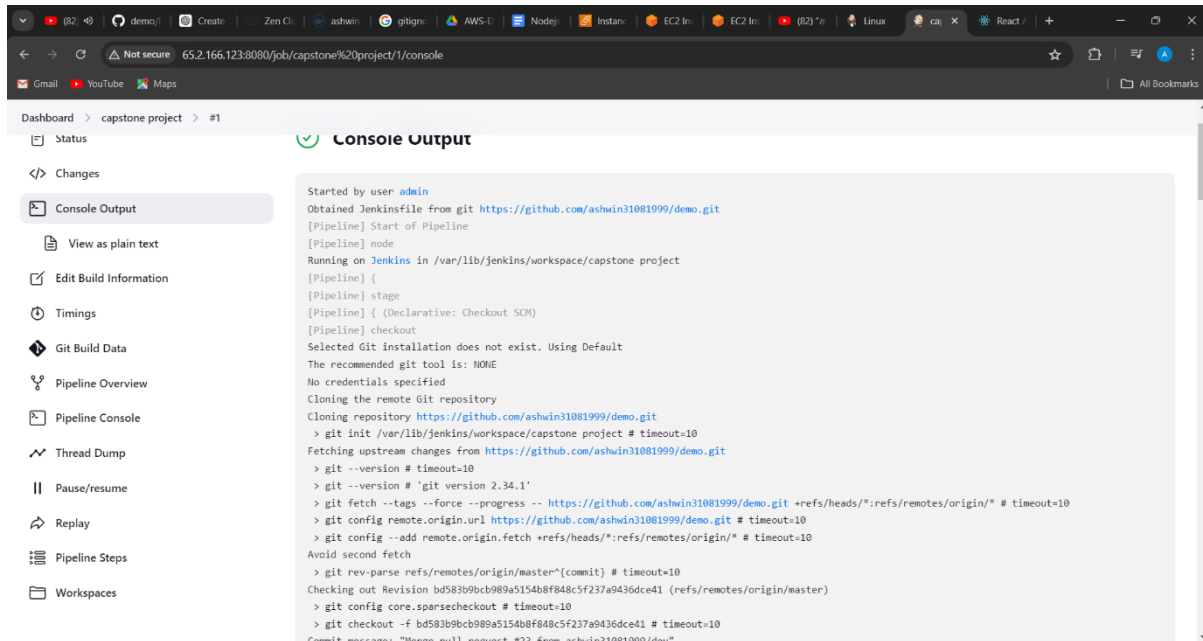
12. Configuring the pipeline-4



13. Webhook configuration



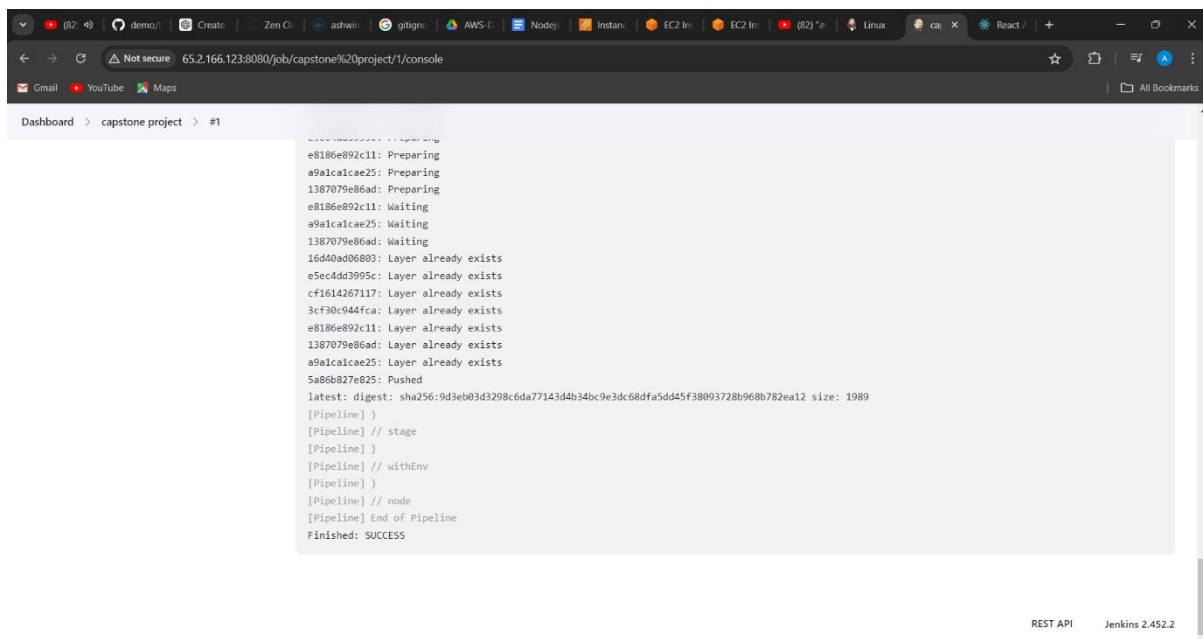
14. Console output -1



The screenshot shows the Jenkins web interface for a pipeline run. The left sidebar contains navigation links: Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Timings, Git Build Data, Pipeline Overview, Pipeline Console, Thread Dump, Pause/resume, Replay, Pipeline Steps, and Workspaces. The main area displays the console output for the pipeline, which includes the following text:

```
Started by user admin
Obtained Jenkinsfile from git https://github.com/ashwin31081999/demo.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/capstone project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/ashwin31081999/demo.git
> git init /var/lib/jenkins/workspace/capstone project # timeout=10
Fetching upstream changes from https://github.com/ashwin31081999/demo.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/ashwin31081999/demo.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/ashwin31081999/demo.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/master^{commit} # timeout=10
Checking out Revision bd583b9bcb989a5154b8f848c5f237a9436dce41 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f bd583b9bcb989a5154b8f848c5f237a9436dce41 # timeout=10
Commit message: "Merge pull request #23 from ashwin31081999/dev"
```

15. Console output -2

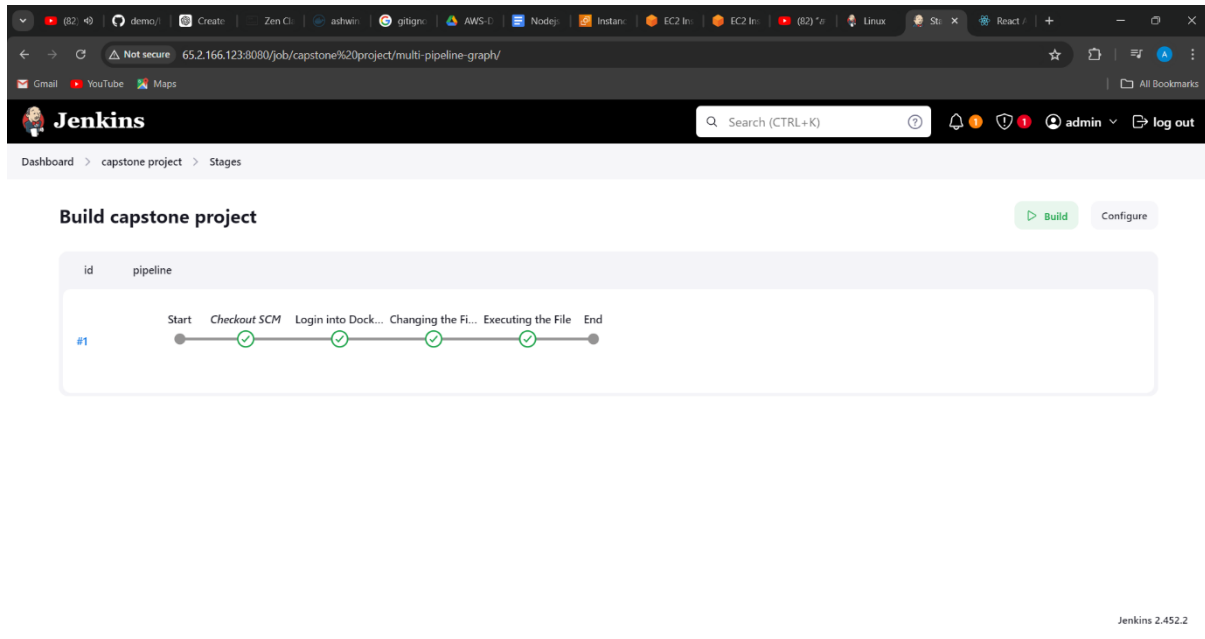


The screenshot shows the Jenkins web interface for a pipeline run. The left sidebar contains navigation links: Dashboard, capstone project, #1. The main area displays the console output for the pipeline, which includes the following text:

```
2023-08-22 10:11:11.111 [INFO] [Pipeline] node
e8186e892c11: Preparing
a9a1ca1cae25: Preparing
1387079e86ad: Preparing
e8186e892c11: Waiting
a9a1ca1cae25: Waiting
1387079e86ad: Waiting
16d40ad06803: Layer already exists
e5ec4dd3995c: Layer already exists
cf1614267117: Layer already exists
3cf30c944fca: Layer already exists
e8186e892c11: Layer already exists
1387079e86ad: Layer already exists
a9a1ca1cae25: Layer already exists
5a86b827e825: Pushed
latest: digest: sha256:9d3eb03d3298c6da77143d4b34bc9e3dc68dfa5d45f38093728b968b782ea12 size: 1989
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.452.2

16. Stages output



The screenshot shows the Jenkins web interface for a pipeline named 'Build capstone project'. The pipeline is in a 'Completed' state, indicated by green checkmarks on all stages. The stages are: Start, Checkout SCM, Login into Dock..., Changing the Fi..., Executing the File, and End. The pipeline is displayed in a 'graph' view. The Jenkins version is 2.452.2.

Build capstone project

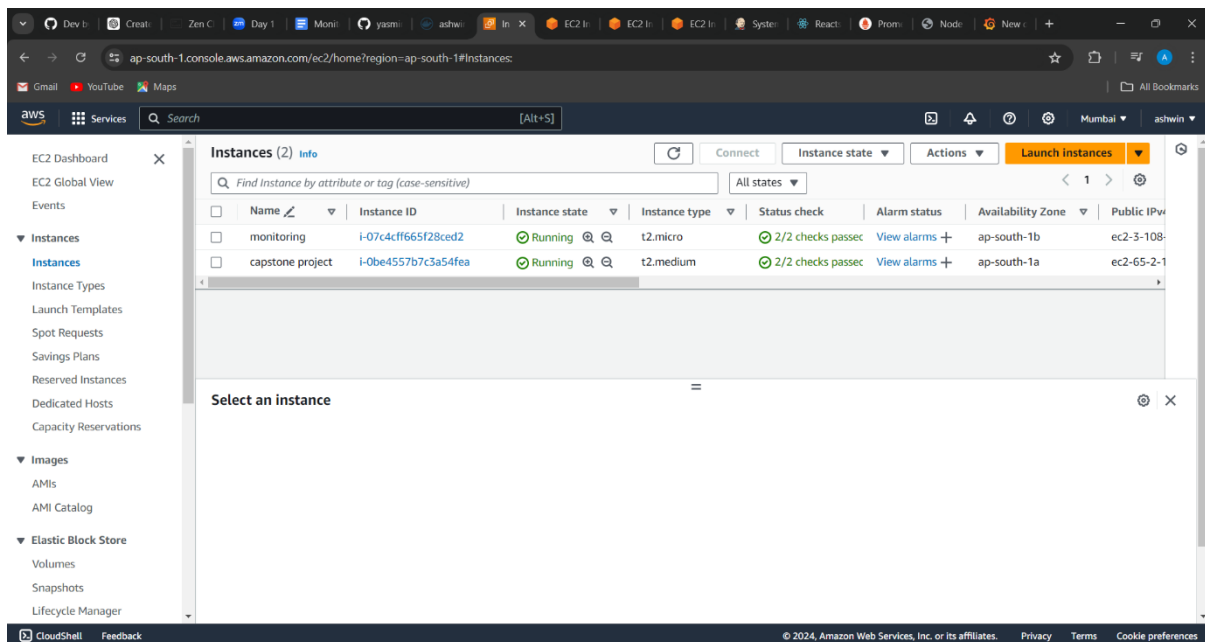
id pipeline

#1

Start Checkout SCM Login into Dock... Changing the Fi... Executing the File End

Jenkins 2.452.2

17. EC2 machines created



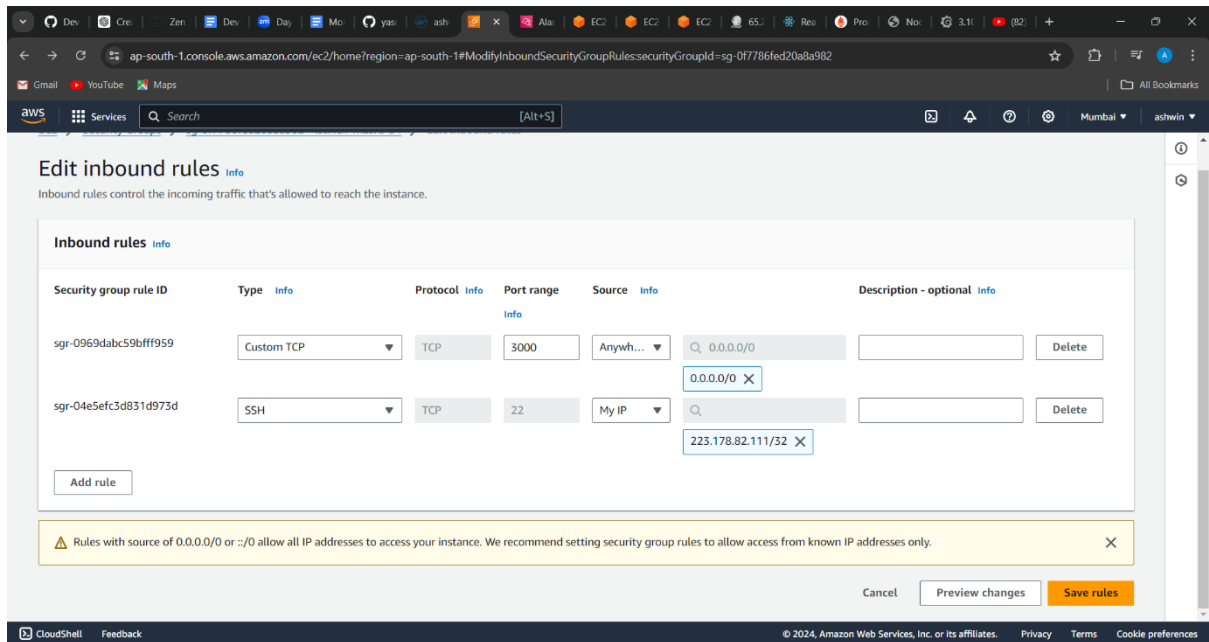
The screenshot shows the AWS Management Console 'Instances' page. Two EC2 instances are listed: 'monitoring' (t2.micro) and 'capstone project' (t2.medium). Both instances are in the 'Running' state. The 'capstone project' instance is highlighted. The console also shows a 'Select an instance' dialog box.

Instances (2)

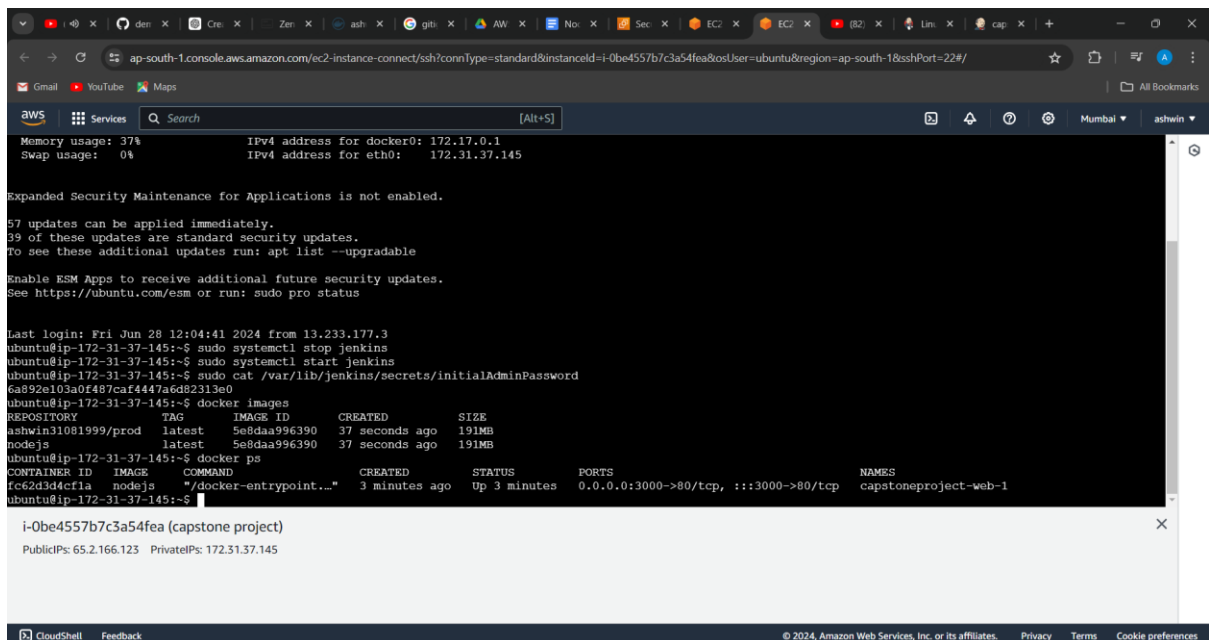
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
monitoring	i-07c4cf665f28ced2	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	ec2-3-108-
capstone project	i-0be4557b7c3a54fea	Running	t2.medium	2/2 checks passed	View alarms	ap-south-1a	ec2-65-2-1

Select an instance

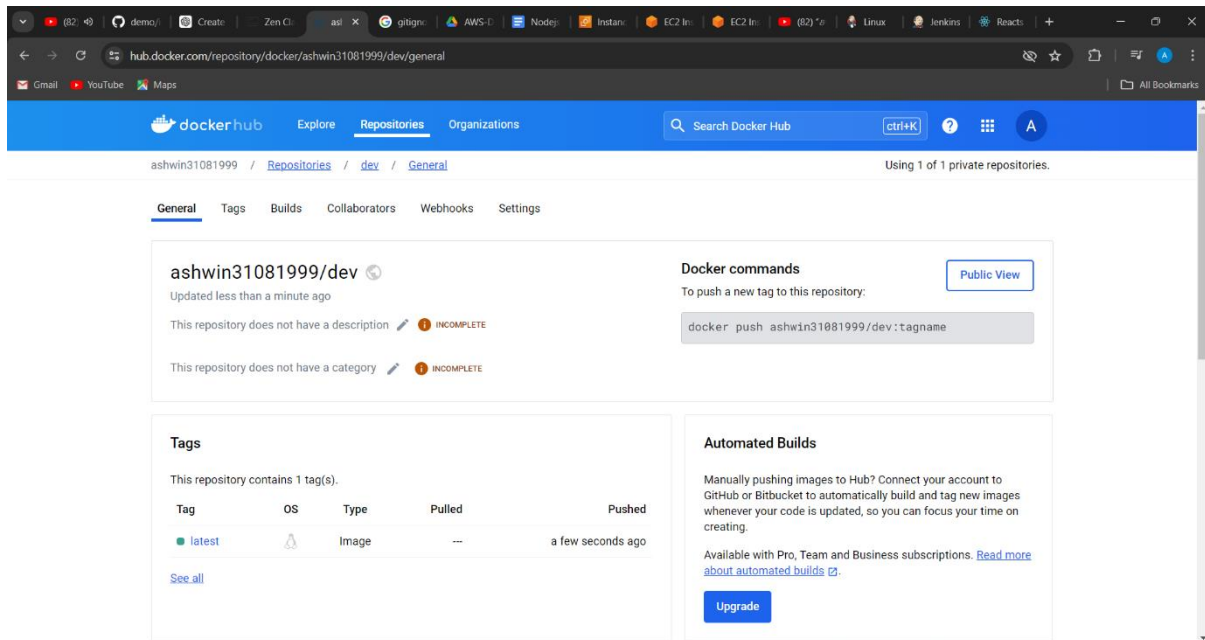
18. Security group configuration



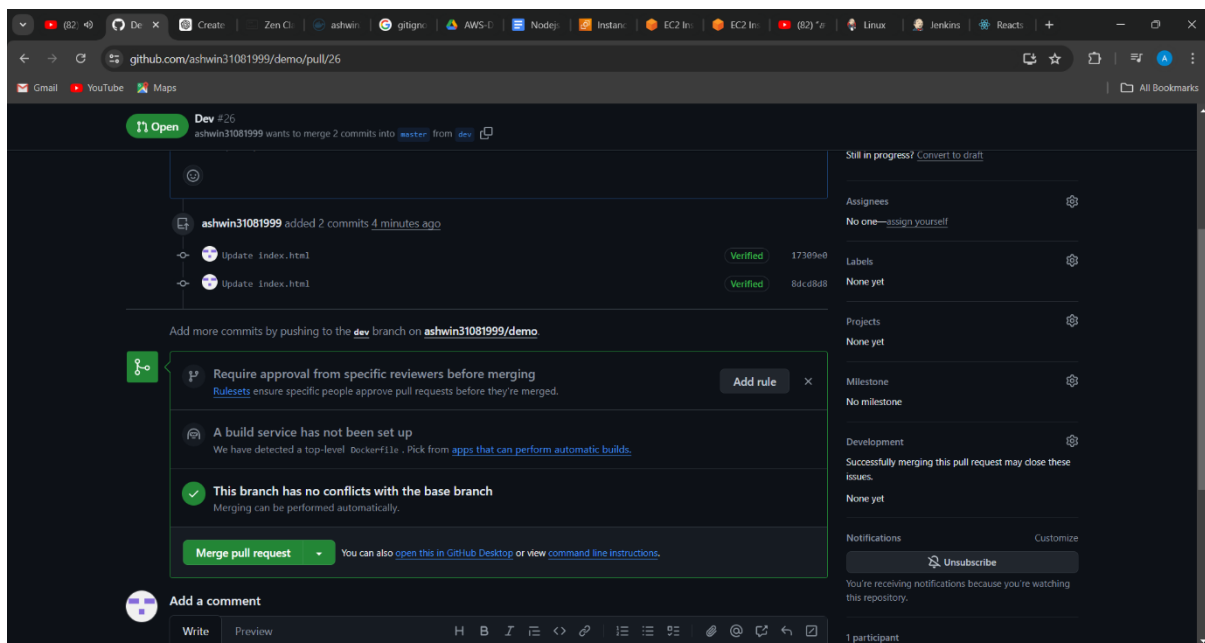
19. Images and containers created



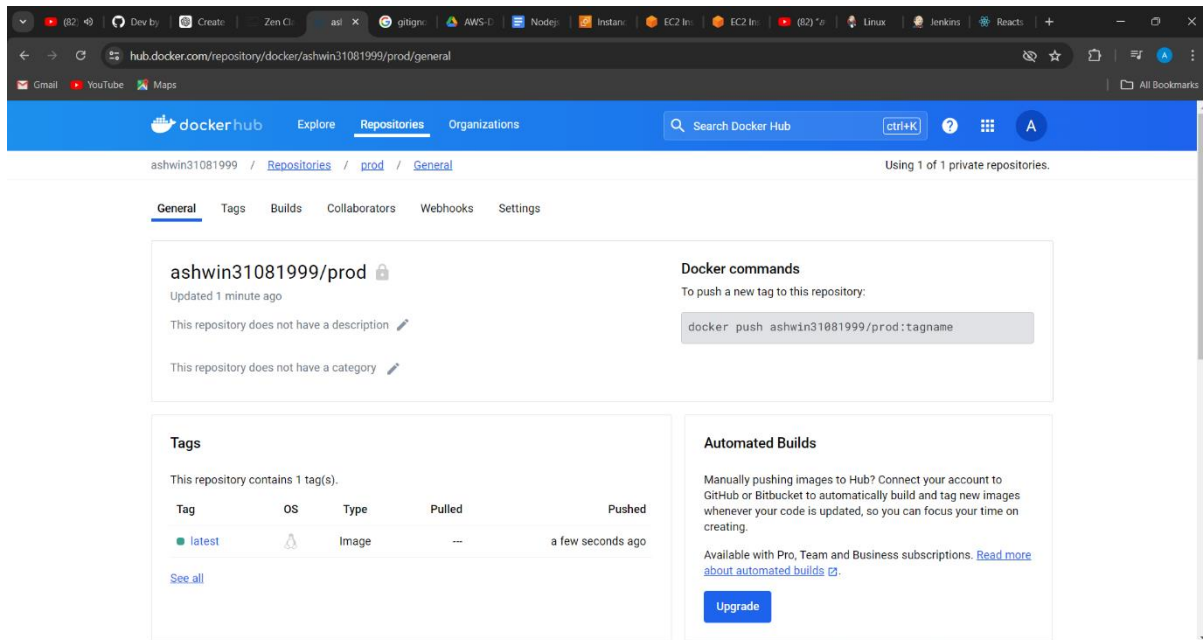
20. Images pushed to docker dev repo



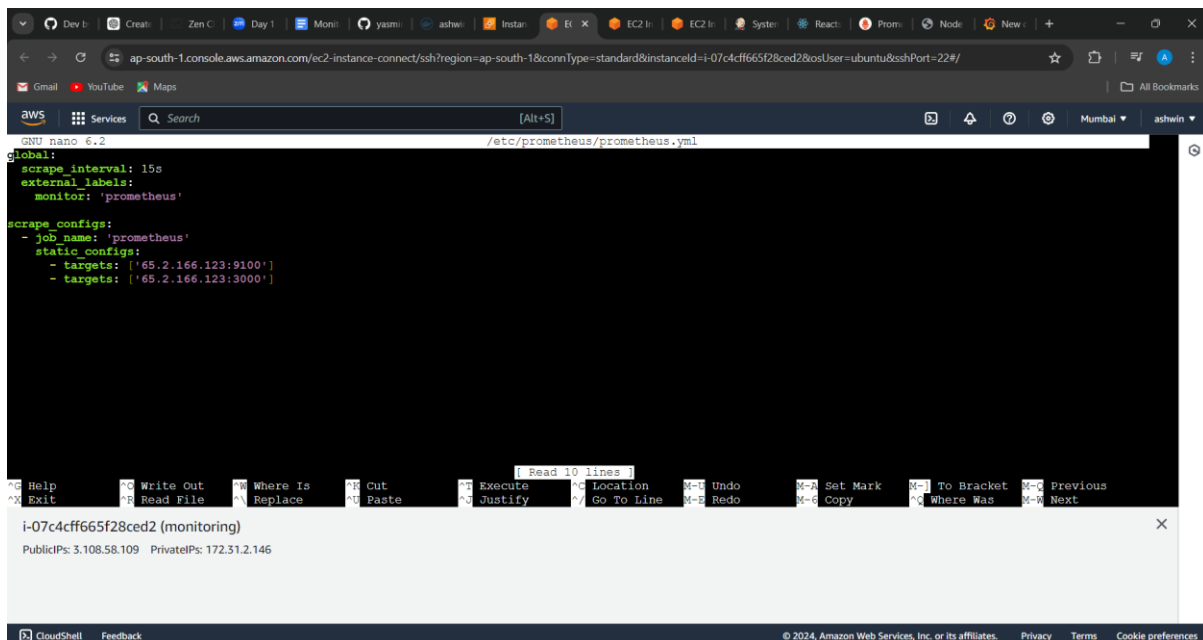
21. The changes merged to master repo in git



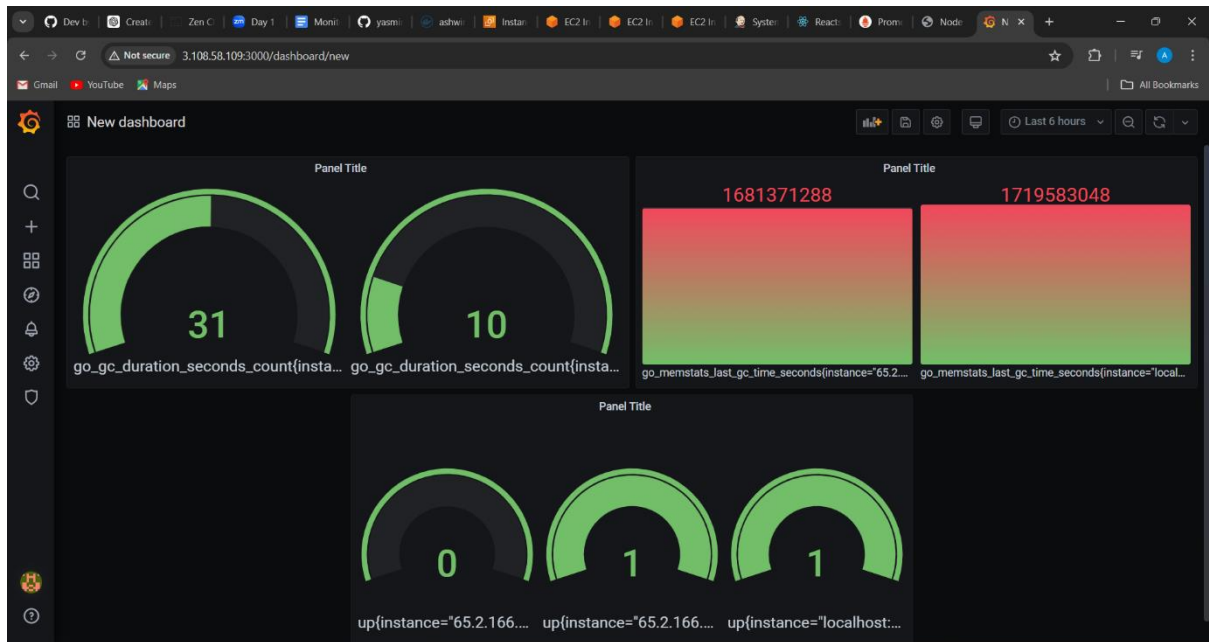
22. After second run the image is pushed to prod (private) repo in docker



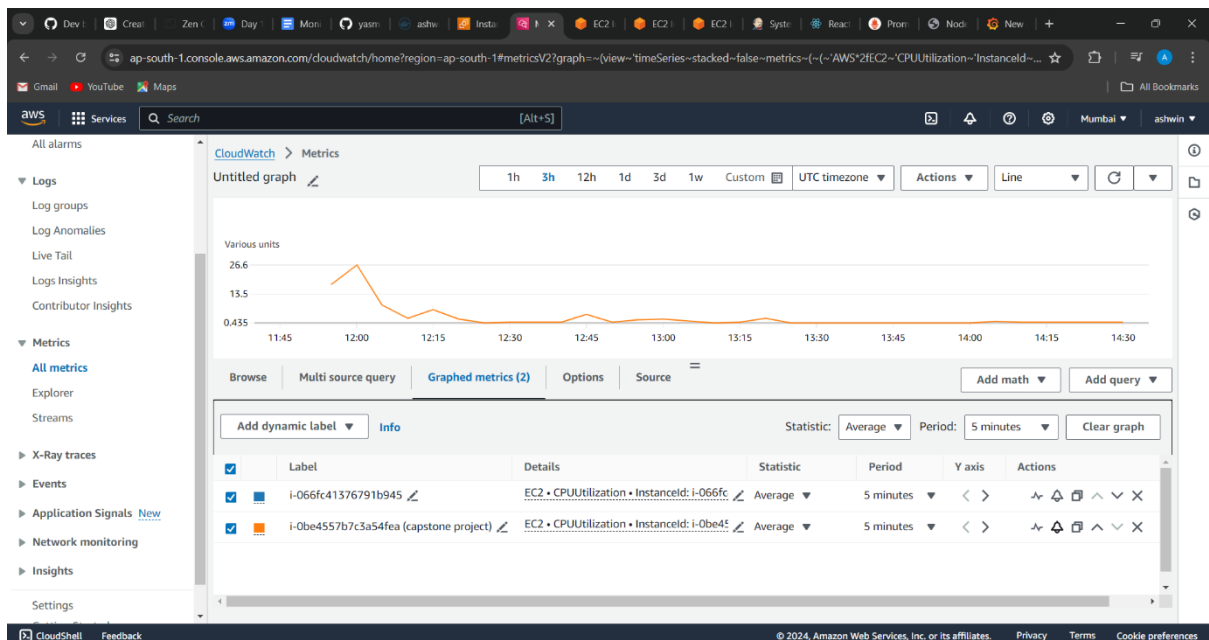
23. Configuring the Prometheus file



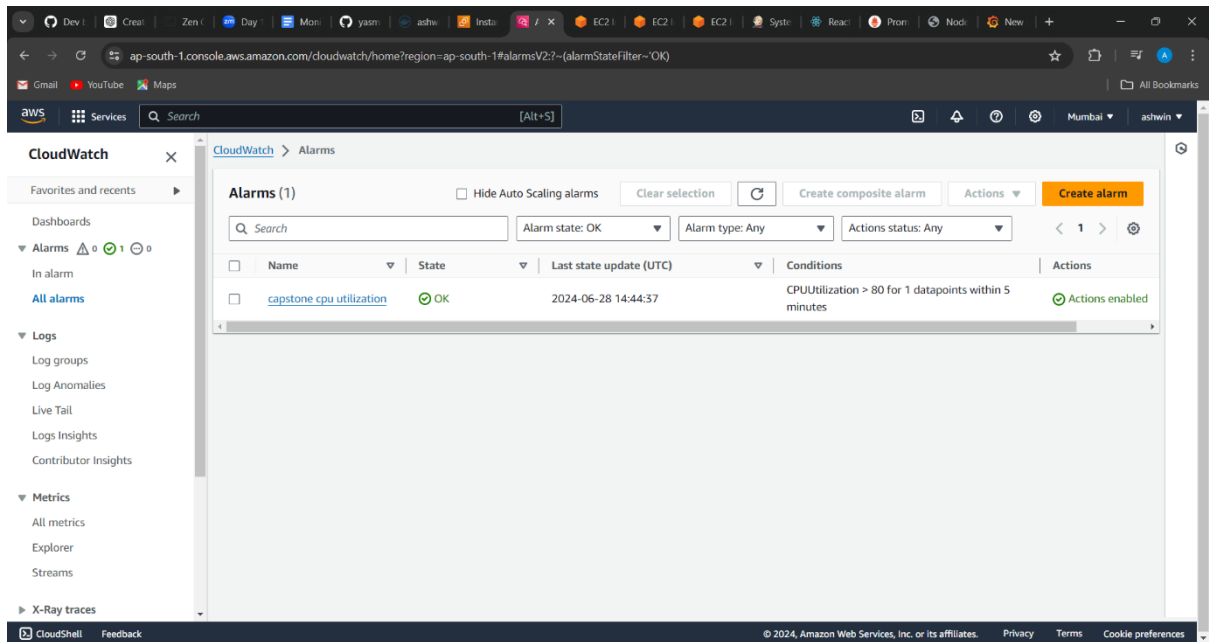
24. Monitoring output using Grafana



25. Monitoring using cloud watch



26 Setting an alarm to get notified via email



The screenshot displays the AWS CloudWatch Alarms console. The left-hand navigation pane includes sections for 'Dashboards', 'Alarms' (with a sub-link for 'All alarms'), 'Logs', 'Metrics', and 'X-Ray traces'. The main content area is titled 'Alarms (1)' and features a table with the following columns: Name, State, Last state update (UTC), Conditions, and Actions. A single alarm, 'capstone cpu utilization', is listed with a state of 'OK' and a last update of '2024-06-28 14:44:37'. The conditions are specified as 'CPUUtilization > 80 for 1 datapoints within 5 minutes', and the actions are marked as 'Actions enabled'. Above the table, there are filters for 'Alarm state: OK', 'Alarm type: Any', and 'Actions status: Any'. A 'Create alarm' button is visible in the top right corner of the main area. The bottom of the page shows the 'CloudShell' icon, a 'Feedback' link, and the copyright notice '© 2024, Amazon Web Services, Inc. or its affiliates.' along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Name	State	Last state update (UTC)	Conditions	Actions
capstone cpu utilization	OK	2024-06-28 14:44:37	CPUUtilization > 80 for 1 datapoints within 5 minutes	Actions enabled