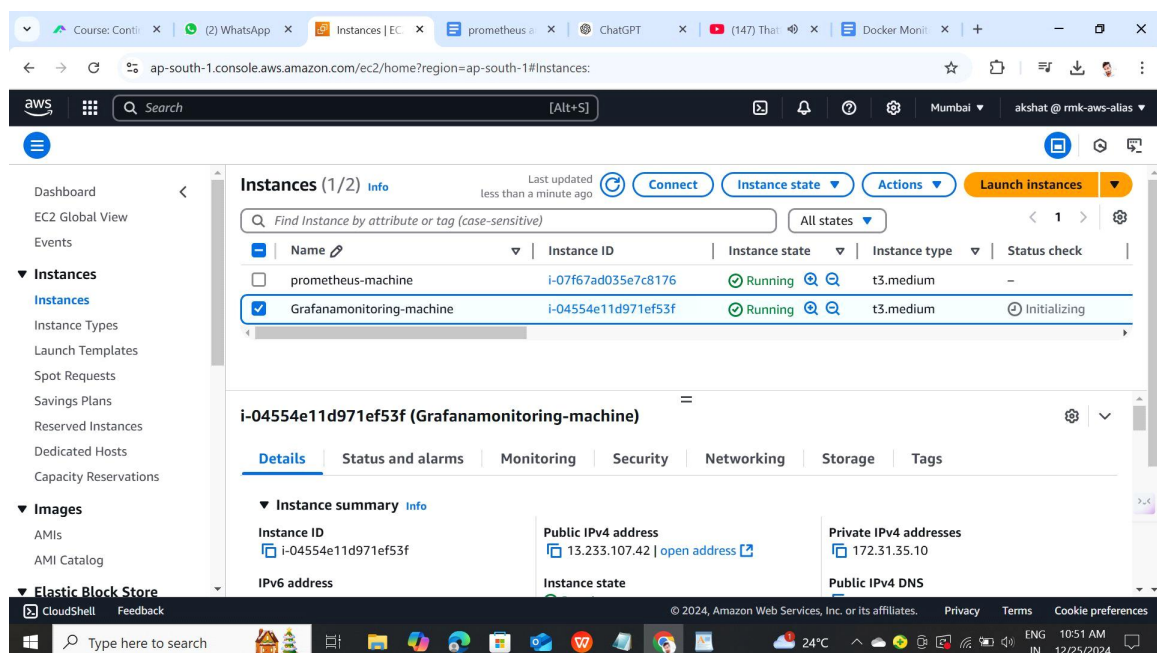


## Continuous Monitoring using Prometheus and Grafana Assignment

### 1) INSTANCE -1 FROM GRAFANA

- successfully launch 2 instance one is grafana and another one is prometheus



- Successfully install grafana

Course x (2) WhatsApp x Instance x EC2 Inst x EC2 Inst x prometheus x ChatGP x (14) x Docker x +

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-04554e11d... [Alt+S]

aws Search [Mumbai] akshat @ rmk-aws-alias

```
grafana-8.4.4/scripts/circle-test-frontend.sh
grafana-8.4.4/scripts/circle-test-mysql.sh
grafana-8.4.4/scripts/circle-test-postgres.sh
grafana-8.4.4/scripts/clean-git-or-error.sh
grafana-8.4.4/scripts/generate-ally-report.sh
grafana-8.4.4/scripts/import_many_dashboards.sh
grafana-8.4.4/scripts/mixin-check.sh
grafana-8.4.4/scripts/protobuf-check.sh
grafana-8.4.4/scripts/stripnulls.sh
grafana-8.4.4/scripts/tag_release.sh
grafana-8.4.4/scripts/trigger_docker_build.sh
grafana-8.4.4/scripts/trigger_grafana_packer.sh
grafana-8.4.4/scripts/trigger_windows_build.sh
grafana-8.4.4/scripts/validate-devenv-dashboards.sh
root@ip-172-31-35-10:/home/ubuntu# ls
grafana-8.4.4 grafana-enterprise-8.4.4.linux-amd64.tar.gz
root@ip-172-31-35-10:/home/ubuntu# cd grafana-8.4.4
root@ip-172-31-35-10:/home/ubuntu/grafana-8.4.4# ./bin/grafana-server
Grafana server is running with elevated privileges. This is not recommended
INFO[12-25]05:24:18] Starting Grafana logger=settings version=8.4.4 commit=fcb01fae branch=HEAD compiled=2022-03-16T13:36:26+0000
```

i-04554e11d971ef53f (Grafanamonitoring-machine)

PublicIPs: 13.233.107.42 PrivateIPs: 172.31.35.10

CloudShell Feedback

Type here to search

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

24°C 10:54 AM 12/25/2024

- we know that grafana is access with port 3000 so we add custom tcp for grafana access to internet

Course x (2) WhatsApp x Modify x EC2 Inst x EC2 Inst x prometheus x ChatGP x (14) x Docker x +

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ModifyInboundSecurityGroupRules.securityGroupId=sg-073245b79c9e701d6 - launch-wizard-7 Edit inbound rules

aws Search [Mumbai] akshat @ rmk-aws-alias

EC2 > Security Groups > sg-073245b79c9e701d6 - launch-wizard-7 > Edit inbound rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sgr-06140406abc72ab97	HTTPS	TCP	443	C...	0.0.0.0/0	Delete
sgr-01e3b4a4ed8fcf240	SSH	TCP	22	C...	0.0.0.0/0	Delete
sgr-0a2ca9f54c7abee4d	HTTP	TCP	80	C...	0.0.0.0/0	Delete
-	Custom TCP	TCP	3000	A...	0.0.0.0/0	Delete

Add rule

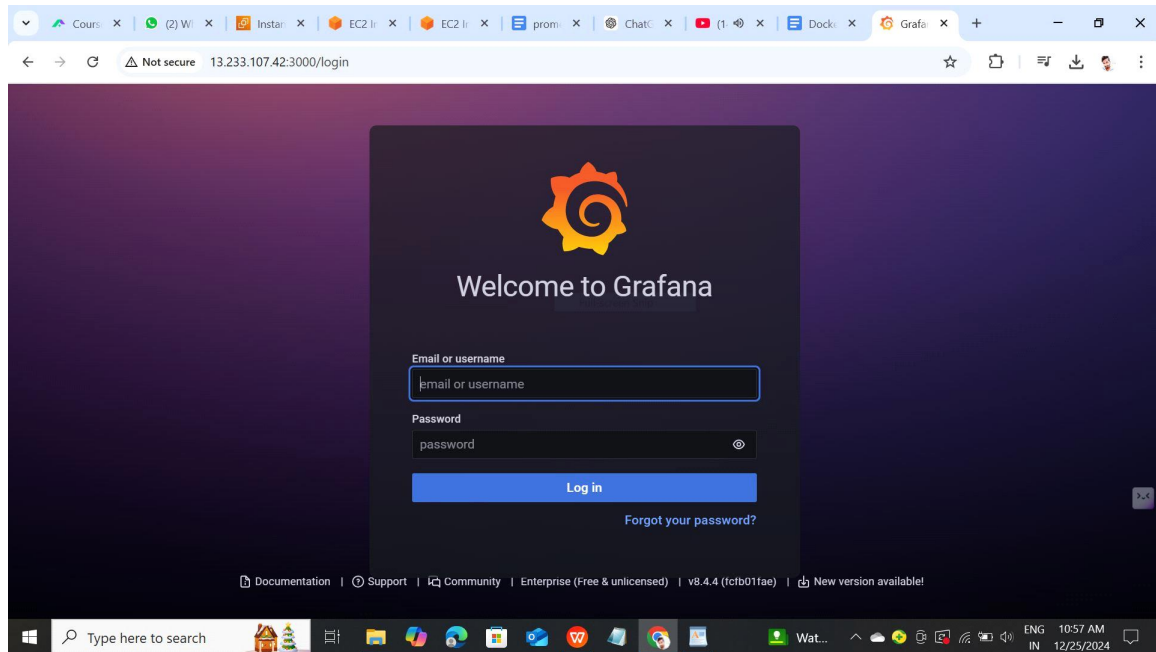
CloudShell Feedback

Type here to search

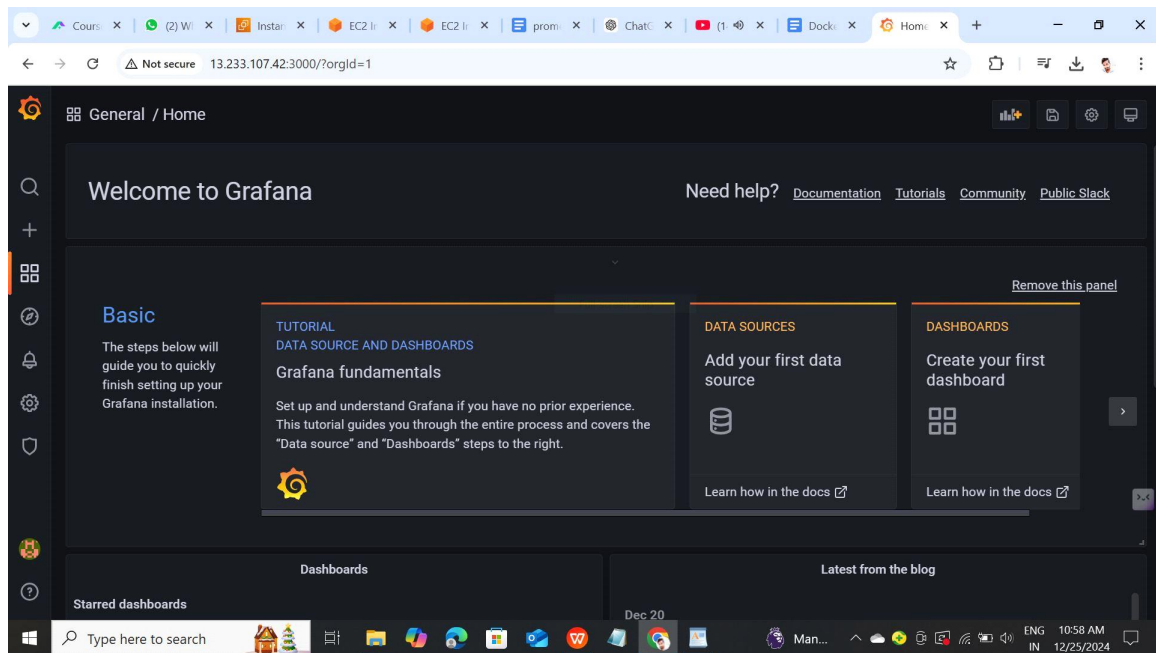
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

10:55 AM 12/25/2024

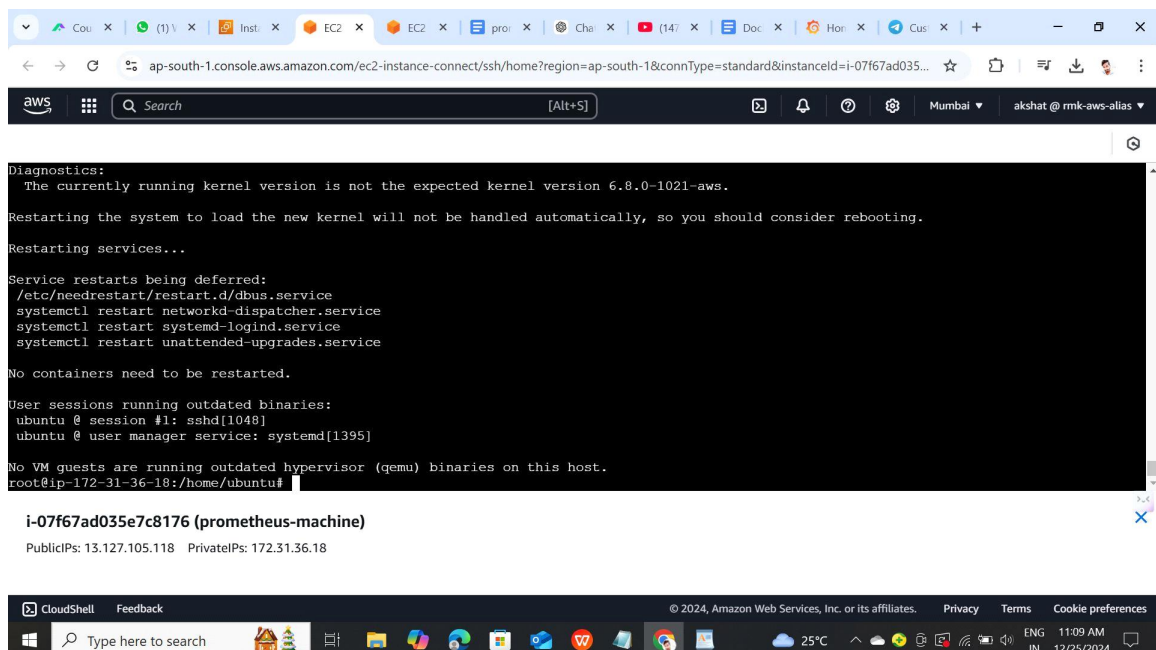
- now we saw successfully access grafana through port 3000



- enter username and password and successfully enter into grafana dashboard



## 2) INSTANCE 2 FOR PROMETHEUS



- Successfully install prometheus

The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output indicates that the Prometheus 2.34.0 Linux AMD64 tarball has been successfully extracted. The files listed include console templates, HTML examples, and the Prometheus binary itself. The terminal prompt is at the root of the instance.

```
2024-12-25 05:44:24 (46.2 MB/s) - 'prometheus-2.34.0.linux-amd64.tar.gz' saved [76299772/76299772]
root@ip-172-31-36-18:/home/ubuntu# tar zxvf prometheus-2.34.0.linux-amd64.tar.gz
prometheus-2.34.0.linux-amd64/
prometheus-2.34.0.linux-amd64/consoles/
prometheus-2.34.0.linux-amd64/consoles/index.html.example
prometheus-2.34.0.linux-amd64/consoles/node-cpu.html
prometheus-2.34.0.linux-amd64/consoles/node-disk.html
prometheus-2.34.0.linux-amd64/consoles/node-overview.html
prometheus-2.34.0.linux-amd64/consoles/node.html
prometheus-2.34.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.34.0.linux-amd64/consoles/prometheus.html
prometheus-2.34.0.linux-amd64/console_libraries/
prometheus-2.34.0.linux-amd64/console_libraries/menu.lib
prometheus-2.34.0.linux-amd64/console_libraries/prom.lib
prometheus-2.34.0.linux-amd64/prometheus.yml
prometheus-2.34.0.linux-amd64/LICENSE
prometheus-2.34.0.linux-amd64/NOTICE
prometheus-2.34.0.linux-amd64/prometheus
prometheus-2.34.0.linux-amd64/promtool
root@ip-172-31-36-18:/home/ubuntu#
```

**i-07f67ad035e7c8176 (prometheus-machine)**  
PublicIPs: 13.127.105.118 PrivateIPs: 172.31.36.18

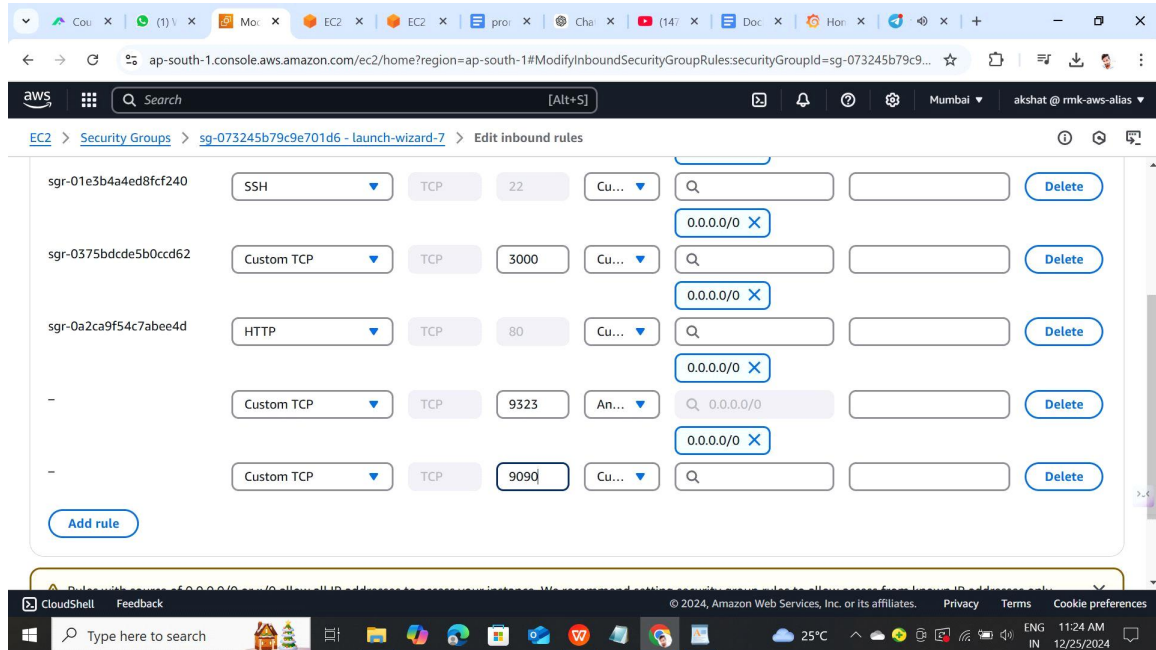
- prometheus working on port 9293

The screenshot shows the AWS CloudShell interface with a terminal window displaying the Prometheus configuration file. The configuration sets the metrics address to 0.0.0.0:9293 and enables experimental features.

```
{
  "metrics-addr": "0.0.0.0:9293",
  "experimental": true
}
```

**i-07f67ad035e7c8176 (prometheus-machine)**  
PublicIPs: 13.127.105.118 PrivateIPs: 172.31.36.18

- Edit inbound rule to add custom tcp port 9090 for prometheus and add 9100 for jenkins-server



- we access metrics through internet



```
13.127.105.118:9323/metrics
# HELP engine_daemon_container_actions_seconds The number of seconds it takes to process each container action
# TYPE engine_daemon_container_actions_seconds histogram
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="changes"} 0
engine_daemon_container_actions_seconds_count{action="changes"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="commit"} 0
engine_daemon_container_actions_seconds_count{action="commit"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="create",le="2.5"} 1
```

- we add port of jenkins server and docker also add for visualization

```
ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=ap-south-1&connType=standard&instanceId=i-07f67ad035...

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: "docker"

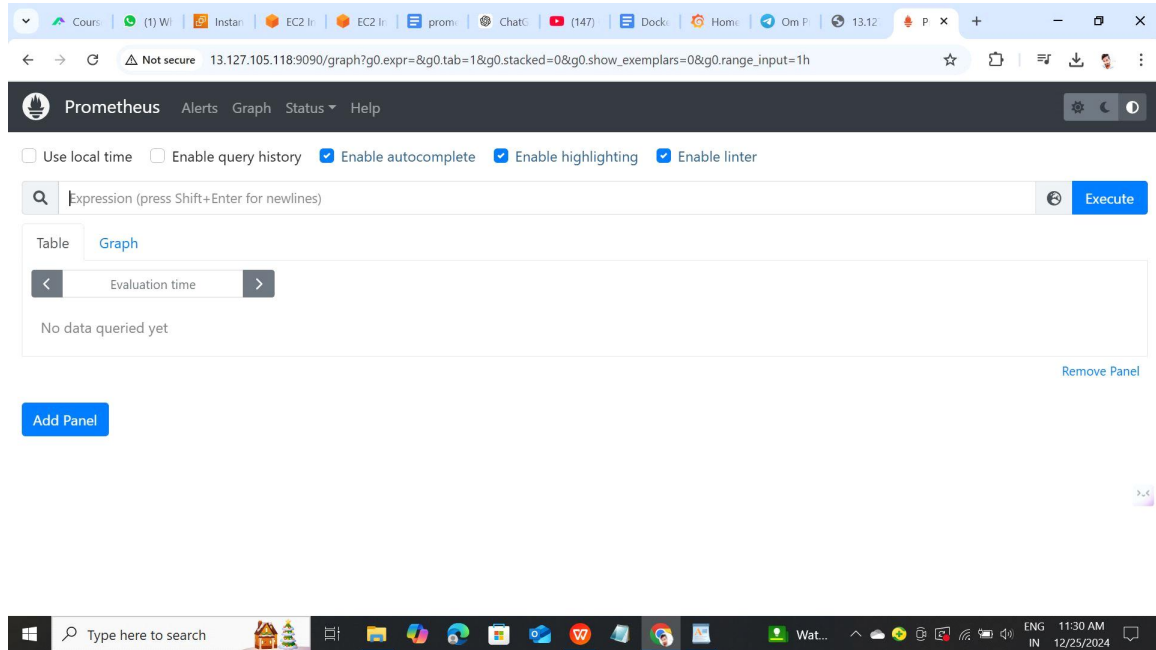
    # metrics path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9323"]

-- INSERT --

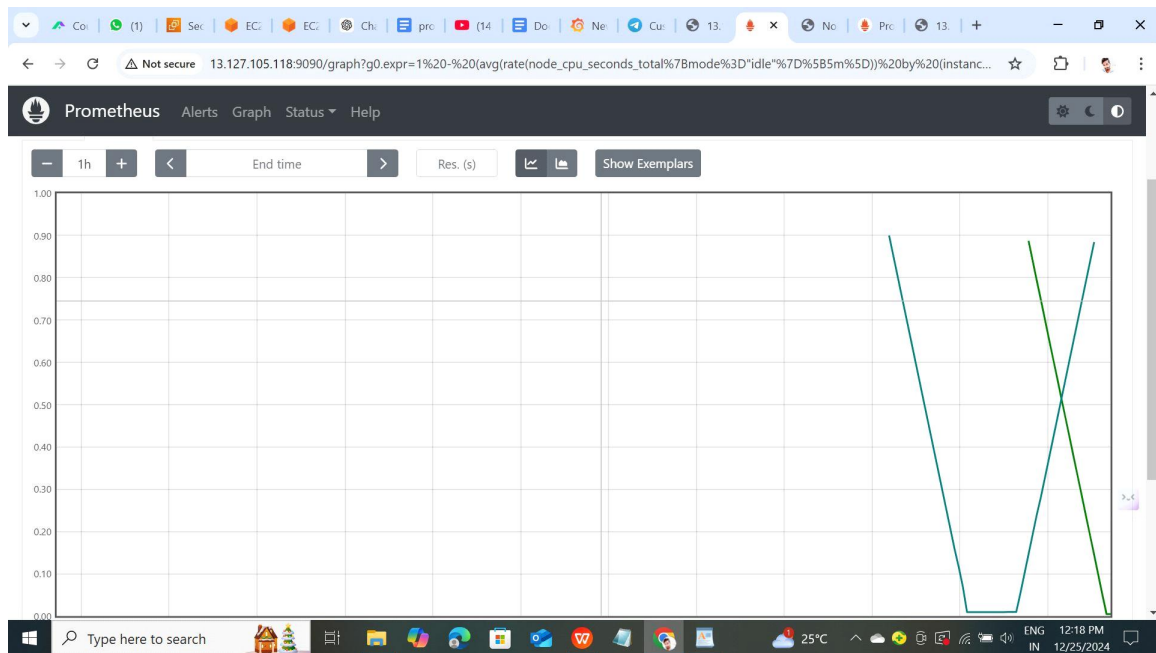
i-07f67ad035e7c8176 (prometheus-machine)
PublicIPs: 13.127.105.118 PrivateIPs: 172.31.36.18
```

- Successfully access prometheus through port 9090

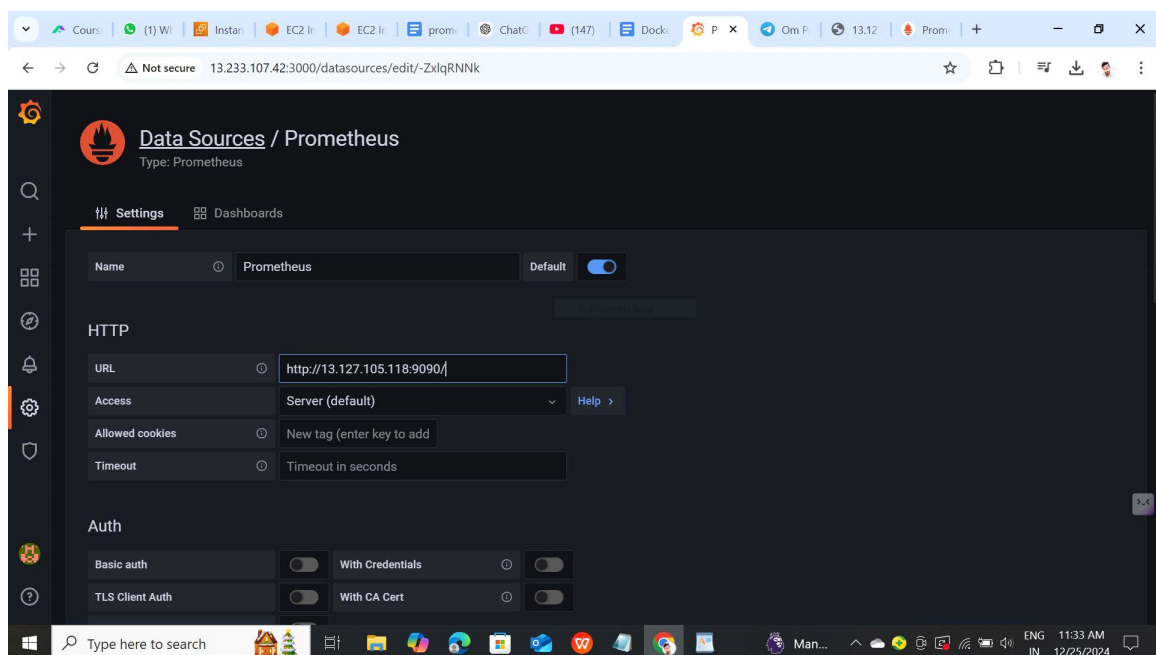


- we saw prometheus successfully allow jenkins server metrics

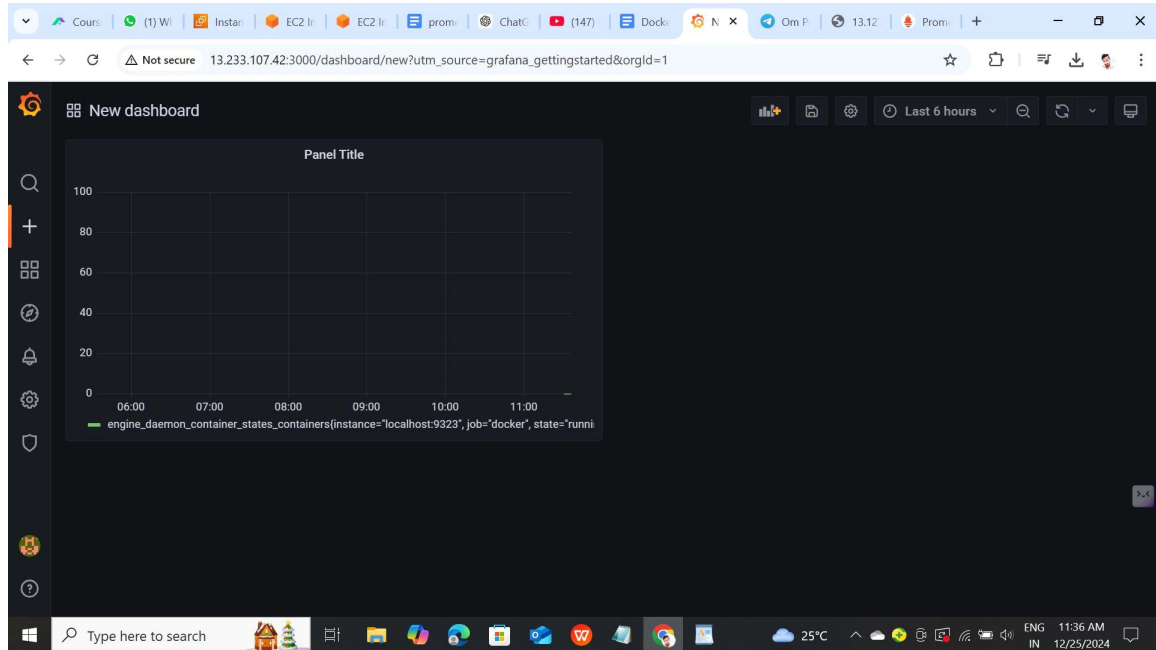




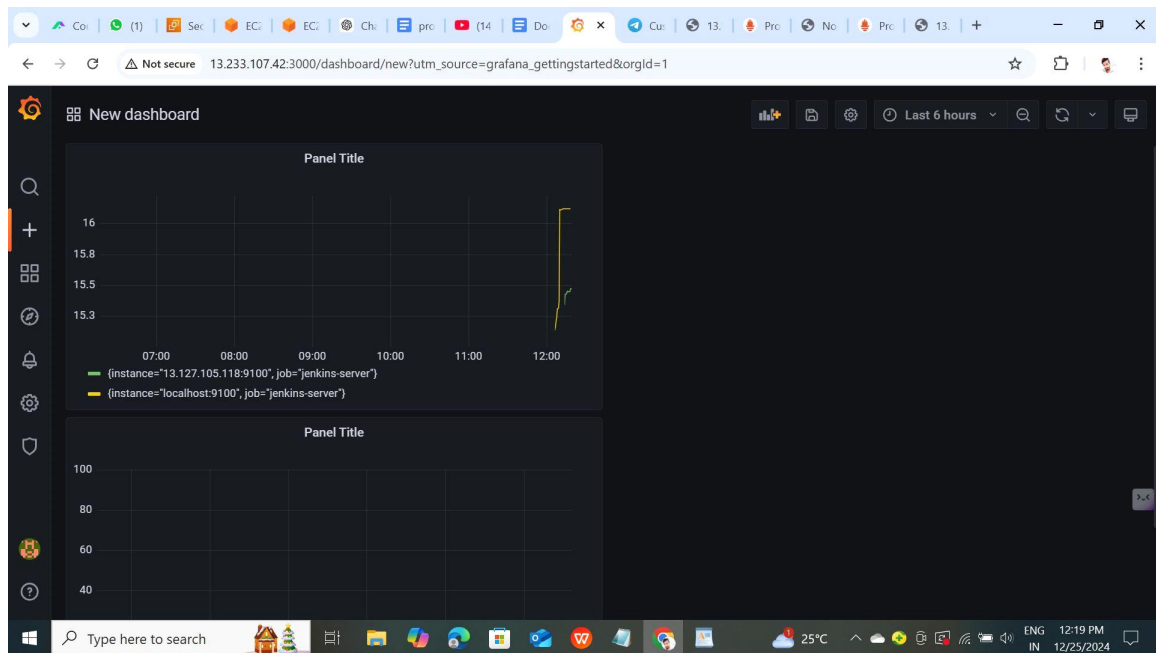
- Successfully add prometheus in grafana for access the metrics in visualization mood



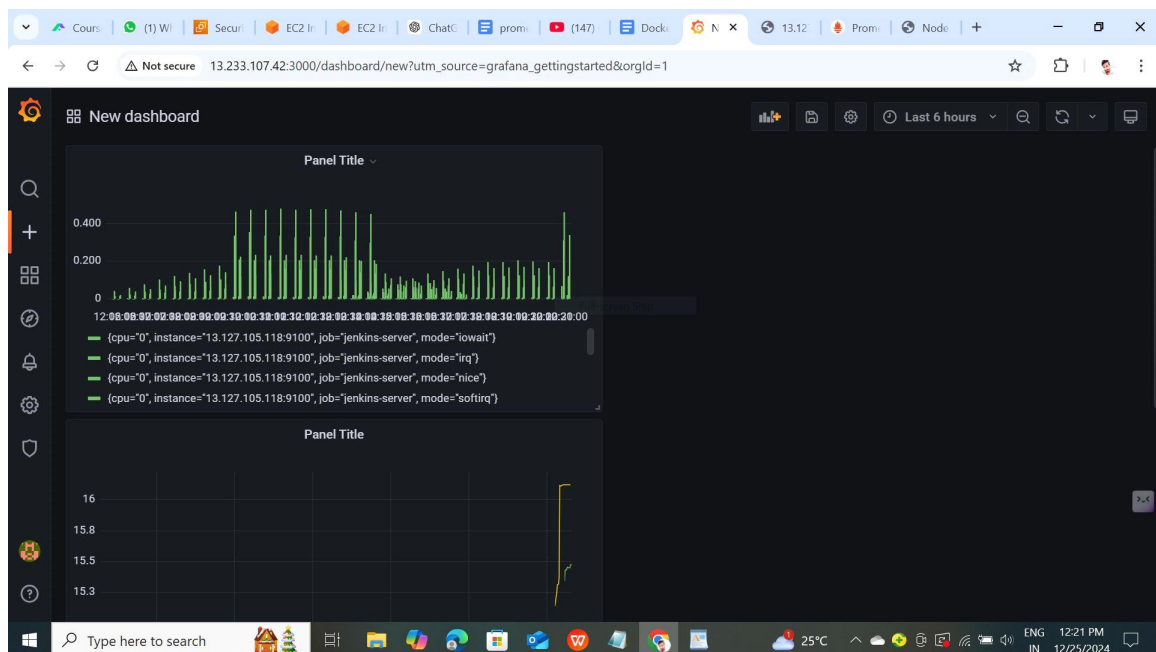
- we saw the docker matrices their no conatiner running at



- we saw the cup utilization graph at below



we saw the memoery utilization graph below at



**YOU**

**THANK**

