Banking and Finance Domain

A PROJECT REPORT

SUBMITTED BY

ANMOL UPADHAYA

DEVOPS Course

SUBMITTED TO

Star Agile

APR & 2025

Certification Project – Finance Me

Banking and Finance Domain

FinanceMe is a Global leading Banking and Financial services provider based out of Germany. The company offers products and services like Banking, Funds Management, Loans, Debit Cards and Credits Cards, Investment Banking etc. Initially the company was using a Monolithic application architecture, As the company grown, It started facing difficulties in managing the application infrastructure and application deployments and Scaling of application when the traffic load increases.

FinanceMe has decided to opt for microservice architecture for its applications and decided to go DevOps by implementing necessary automations using CICD. **FinanceMe** has decided to use AWS as primary cloud services provider to create servers, databases and application deployments.

The company's goal is to deliver the product updates frequently to production automatically with High quality & Reliability. They also want to accelerate software delivery speed, quality and reducing feedback time between developers and testers.

Currently, they are facing following problems, because of various technologies involved in the project.

✓Building Complex Monolithic Application is difficult.

✓Manual efforts to test various components/modules of the project

✓Incremental builds are difficult to manage, test and deploy.

✓It was not possible to scale up individual modules independently.

✓Creation of infrastructure and configure it manually is very time consuming

✓Continuous manual monitoring the application is quite challenging.

Later, you need to implement Continuous Integration & Continuous Deployment using following tools:

✓Git - For version control for tracking changes in the code files

✓Maven – For Continuous Build

✓Jenkins - For continuous integration and continuous deployment

✓Docker - For deploying containerized applications

✓Ansible - Configuration management tools

✓Selenium - For automating tests on the deployed web application

✓Terraform - For creation of infrastructure.

✓Prometheus and Grafana – For Automated Monitoring and Report Visualization

Business challenge/requirement

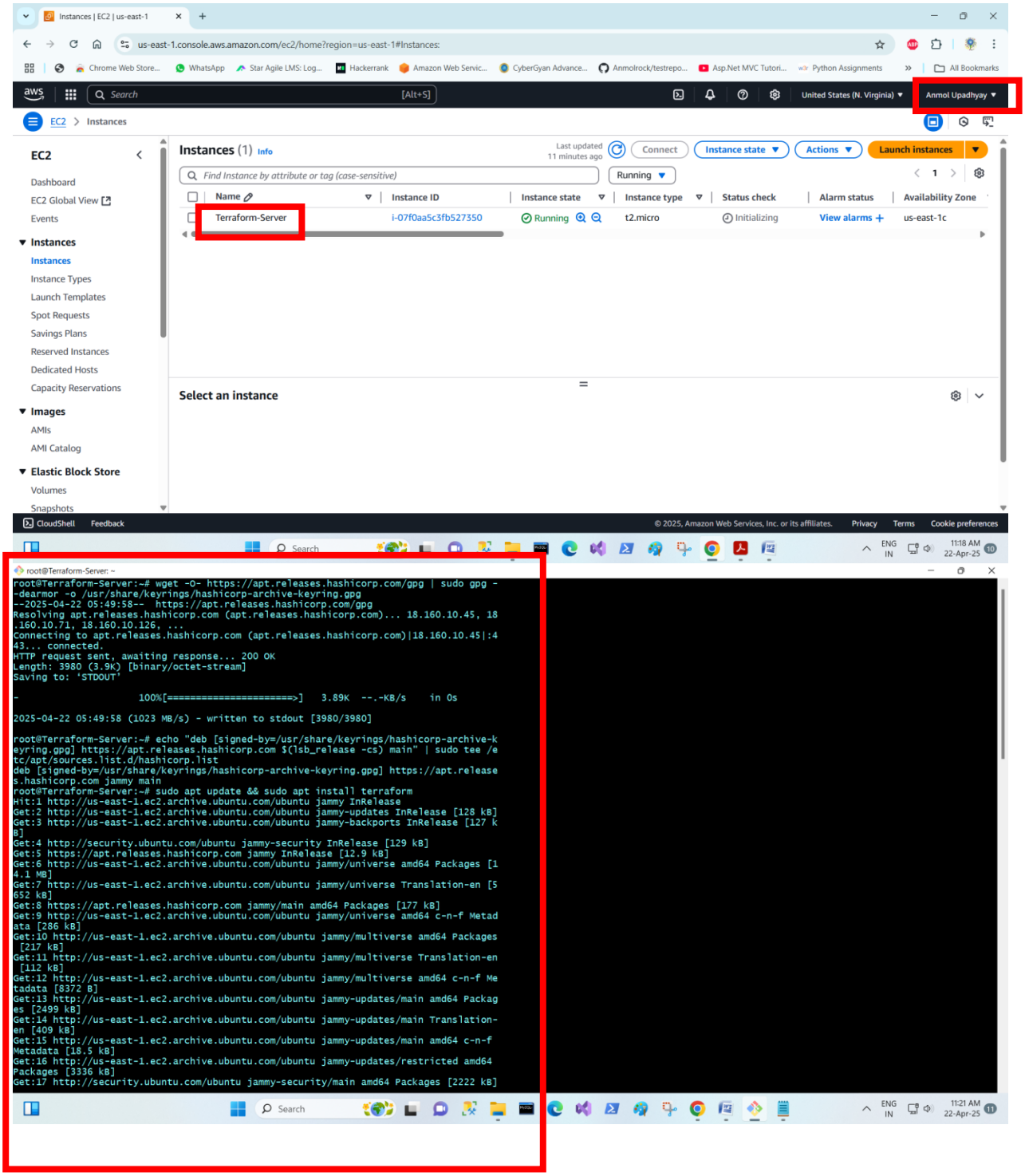
As soon as the developer pushes the updated code on the GIT master branch, the code should be checked out, compiled, tested, packaged and containerized. A new test-server should be provisioned using terraform and should be automatically configured using Ansible with all the required software's and as soon as the server is available, the application must be deployed to the test-server automatically.

The deployment should then be tested using a test automation tool, and if the build is

Successful, Prod server must be configured with all the software it should be pushed to the prod server. All this should happen automatically and should be triggered from a push to the GitHub master branch. Continuous monitoring server must be configured to monitor the test as well as prod server using Prometheus and Grafana should be configured to display a dashboard with following metrics.

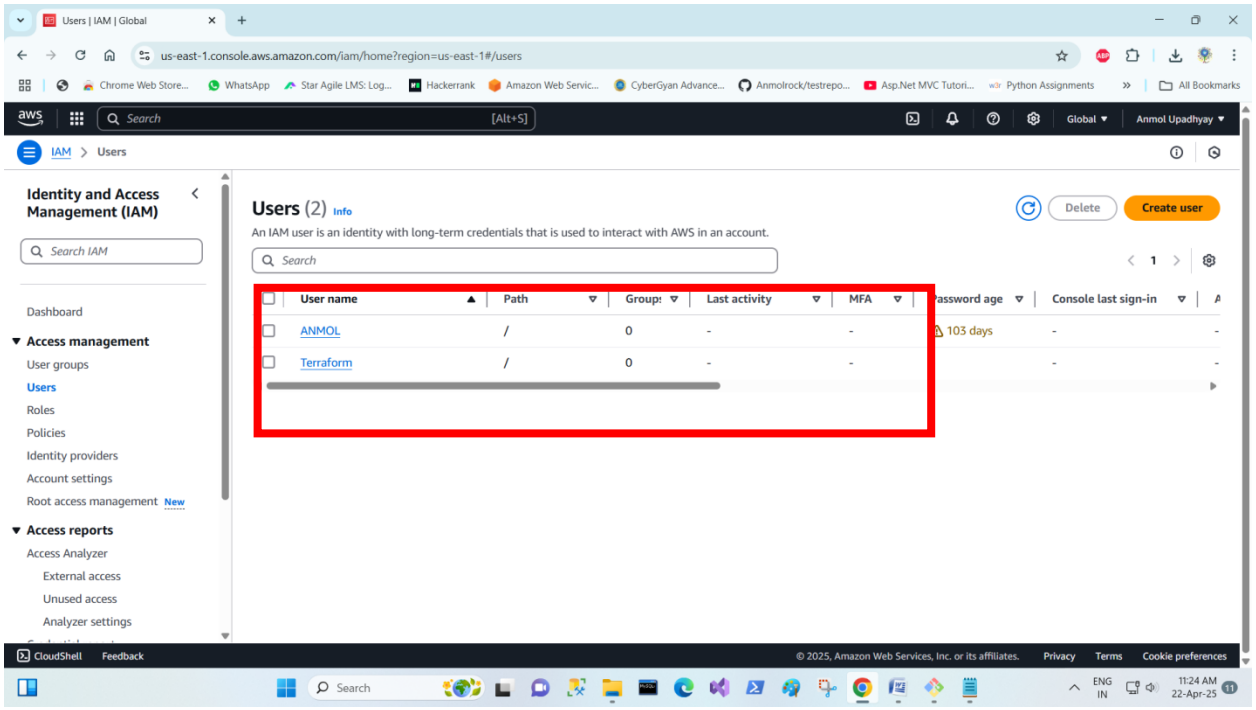
1. CPU utilization
2. Disk Space Utilization
3. Total Available Memory

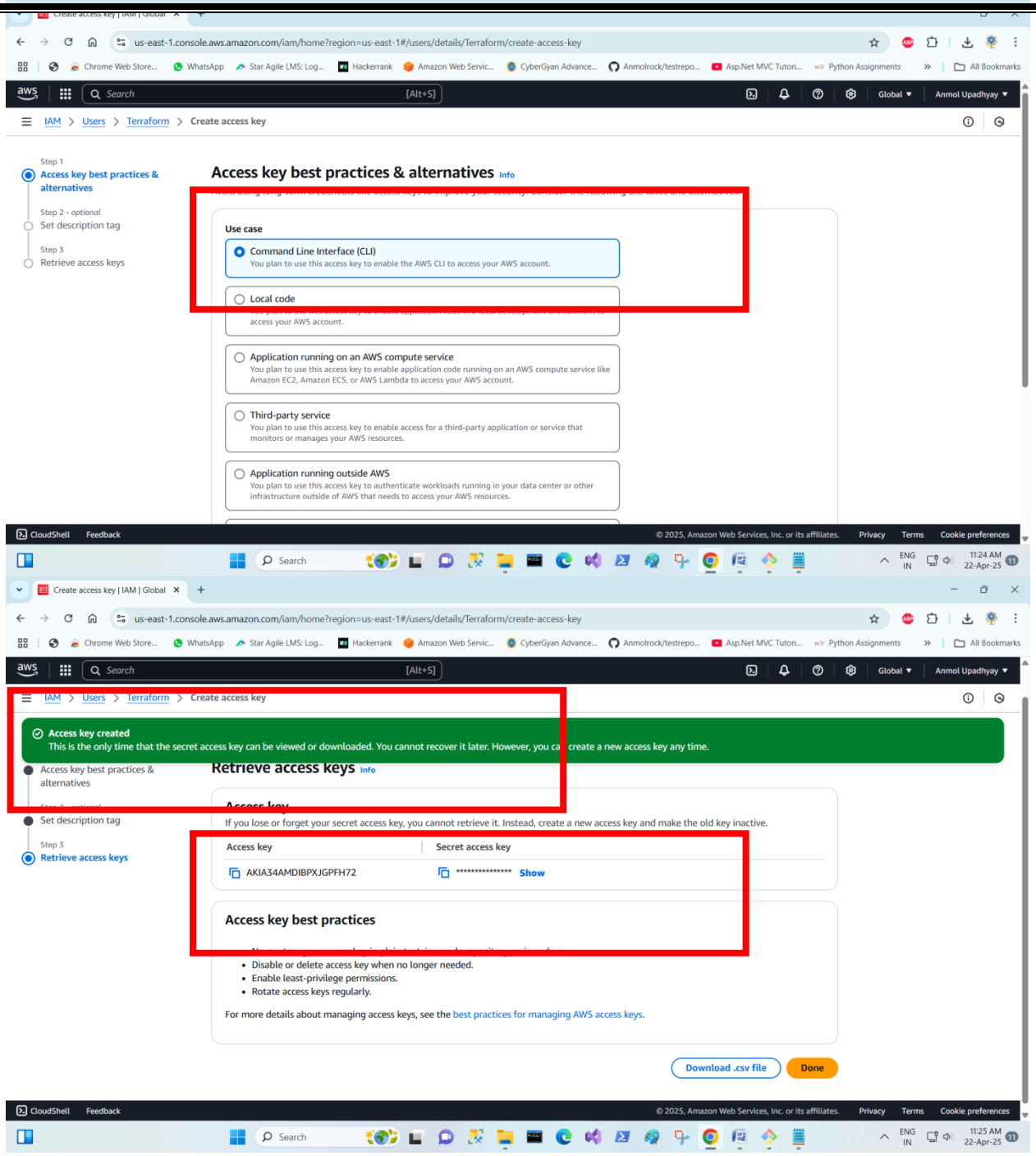
Step1: Create a Terraform – Server to Deploy a DEV ENV, TEST Server, and Prod Server for Further Process.



```
root@Terraform-Server:~# terraform -version
Terraform v1.11.4
on linux_amd64
root@Terraform-Server:~#
```

Step2: Create Accesskey and secret key in AWS for terraform to connect




us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/Terraform/create-access-key
IAM > Users > Terraform > Create access key
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Access key best practices & alternatives Info
Use case
Command Line Interface (CLI)
You plan to use this access key to enable the AWS CLI to access your AWS account.
Local code
Application running on an AWS compute service
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.
Retrieve access keys Info
Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.
Access key
Secret access key
AKIA34AMDIBPXJGPFH72
Show
Access key best practices
Disable or delete access key when no longer needed.
Enable least-privilege permissions.
Rotate access keys regularly.
For more details about managing access keys, see the best practices for managing AWS access keys.
Download .csv file
Done

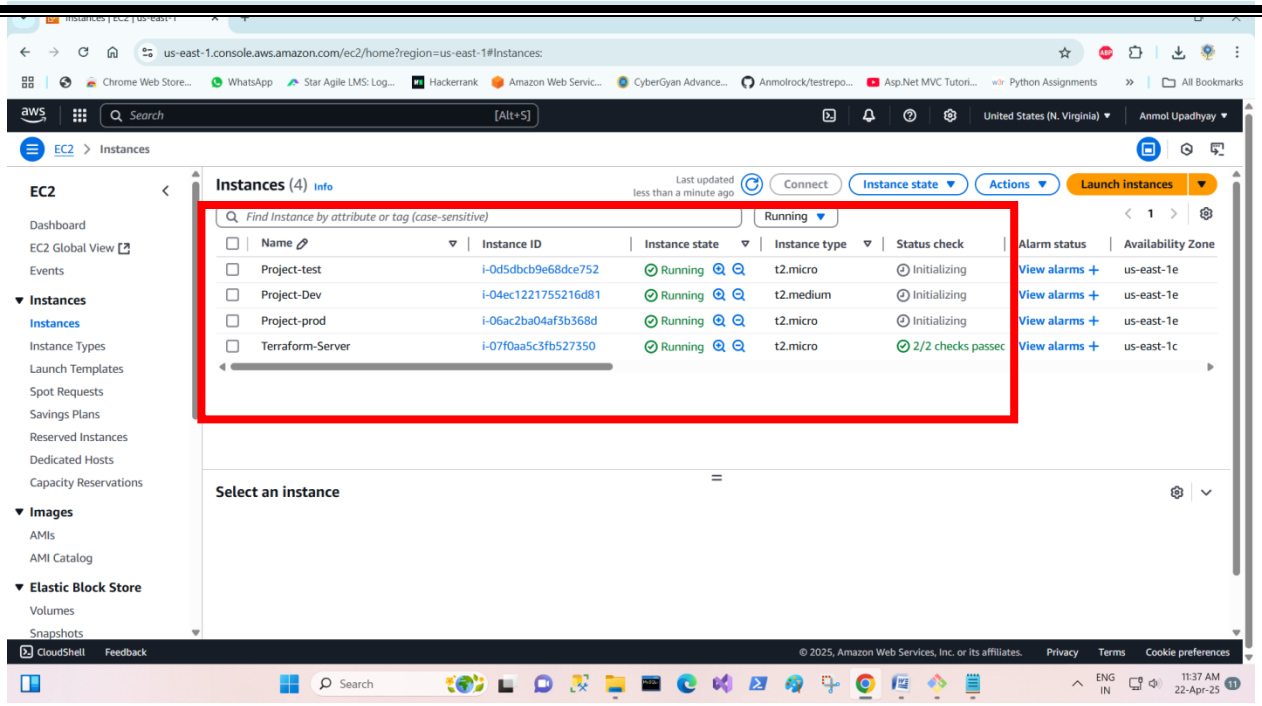
```
root@Terraform-Server:~# aws configure
AWS Access Key ID [None]: AKIA34AMDIBPXJGPFH72
AWS Secret Access Key [None]: OLcvNNigHMrRH+eFhrs8cUd59L1RNEVkMDpc
1eUo
Default region name [None]:
Default output format [None]:
root@Terraform-Server:~#
root@Terraform-Server:~#
```

Step3: Write terraform configuration file to create an EC2 server and install ansible on it.

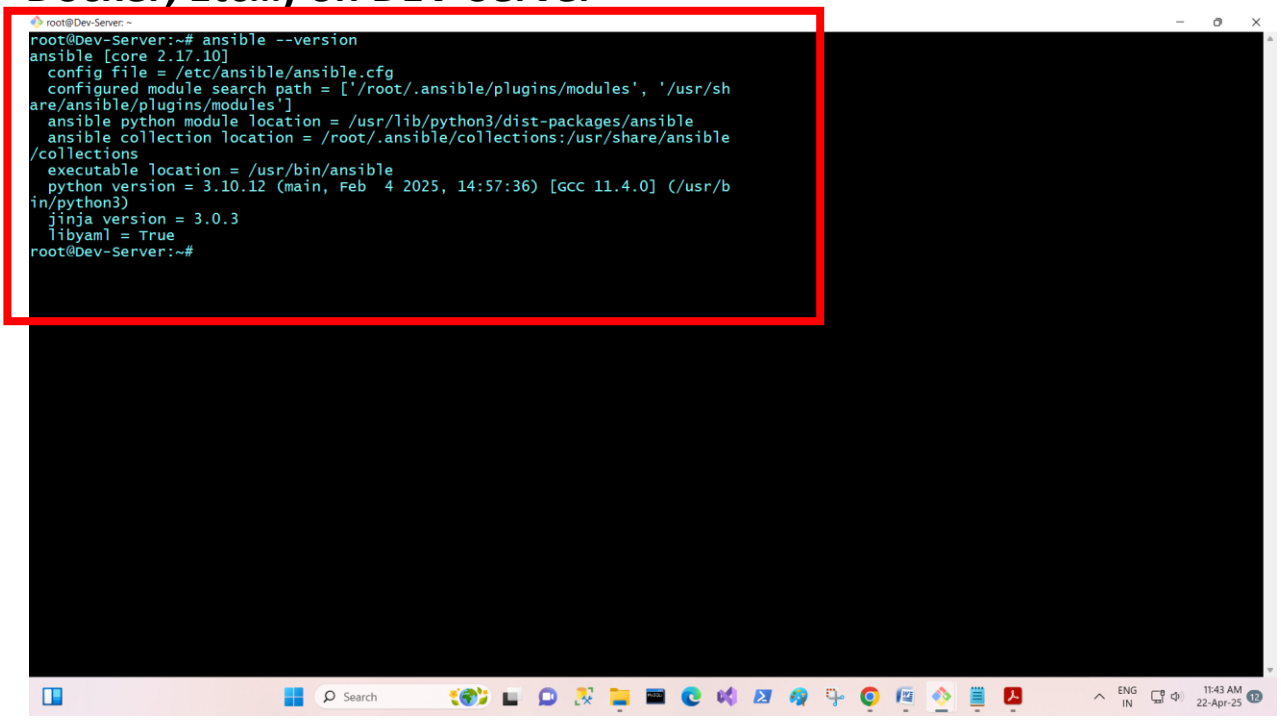
```
root@Terraform-Server:~# mkdir myproject
root@Terraform-Server:~#  cd myproject
root@Terraform-Server:~/myproject# vim aws_infra.tf
root@Terraform-Server:~/myproject# vim aws_infra.tf
root@Terraform-Server:~/myproject#
```

```
resource "aws_instance" "myec2-Dev" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.medium"
  key_name = "mykey"
  subnet_id = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.sl-sg.id]
  tags = {
    Name = "Project-Dev"
  }

}

resource "aws_instance" "myec2-test" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.micro"
  key_name = "mykey"
  subnet_id = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.sl-sg.id]
  tags = {
    Name = "Project-test"
  }

}

resource "aws_instance" "myec2-prod" {
  ami           = "ami-0f9de6e2d2f067fca"
  instance_type = "t2.micro"
  key_name = "mykey"
  subnet_id = aws_subnet.subnet-1.id
  security_groups = [aws_security_group.sl-sg.id]
  tags = {
    Name = "Project-prod"
-- INSERT --                                                    98,20         96%
```

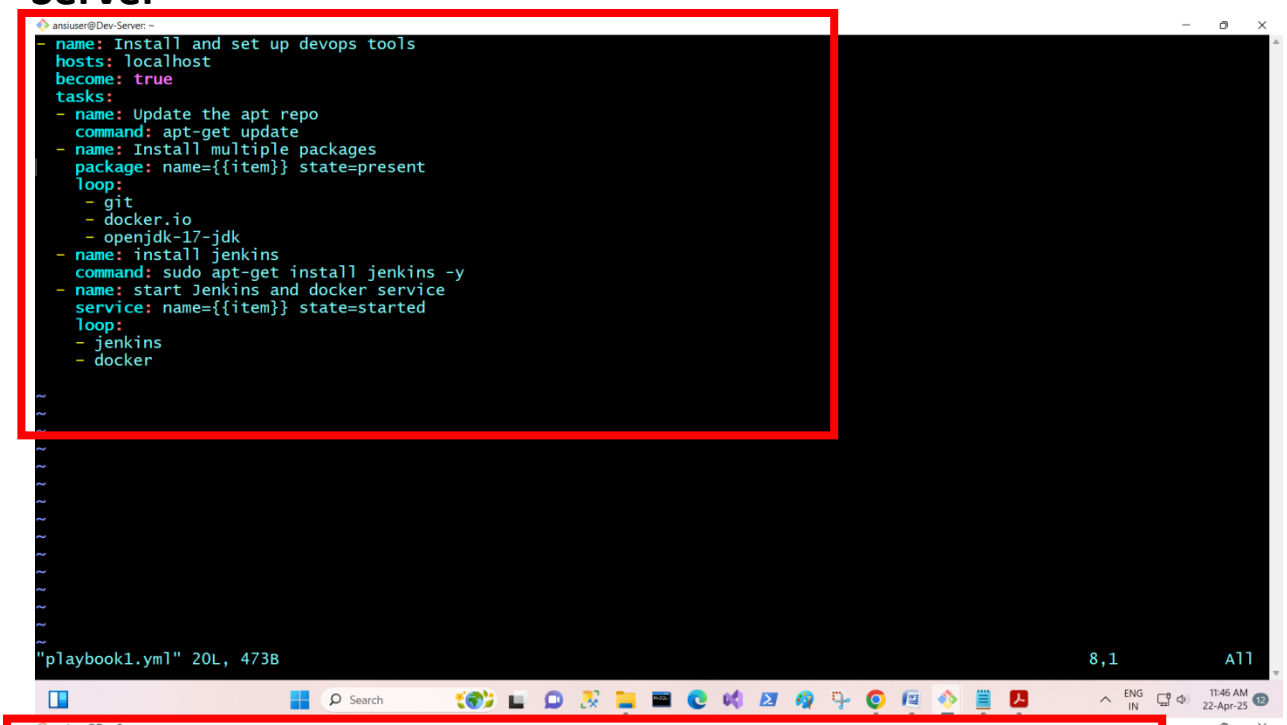
```
      }
    }

Plan: 10 to add, 0 to change, 0 to destroy.
aws_vpc.sl-vpc: Creating...
aws_vpc.sl-vpc: Creation complete after 1s [id=vpc-029b8b1c6e3a326a9]
aws_security_group.sl-sg: Creating...
aws_route_table.sl-route-table: Creating...
aws_internet_gateway.gw: Creating...
aws_subnet.subnet-1: Creating...
aws_internet_gateway.gw: Creation complete after 0s [id=igw-0e77aa1441cfc24c5
]
aws_route_table.sl-route-table: Creation complete after 0s [id=rtb-091575de87
a75a7ca]
aws_route.sl-route: Creating...
aws_route.sl-route: Creation complete after 1s [id=r-rtb-091575de87a75a7ca108
0289494]
aws_security_group.sl-sg: Creation complete after 2s [id=sg-05f17fae89461e8a6
]
aws_subnet.subnet-1: Still creating... [10s elapsed]
aws_subnet.subnet-1: Creation complete after 11s [id=subnet-072096b470819a337
]
aws_instance.myec2-test: Creating...
aws_instance.myec2-Dev: Creating...
aws_route_table_association.a: Creating...
aws_instance.myec2-prod: Creating...
aws_route_table_association.a: Creation complete after 0s [id=rtbassoc-05c4ce
efad583d63d]
aws_instance.myec2-test: Still creating... [10s elapsed]
aws_instance.myec2-Dev: Still creating... [10s elapsed]
aws_instance.myec2-prod: Still creating... [10s elapsed]
aws_instance.myec2-test: Still creating... [20s elapsed]
aws_instance.myec2-Dev: Still creating... [20s elapsed]
aws_instance.myec2-prod: Still creating... [20s elapsed]
aws_instance.myec2-test: Still creating... [30s elapsed]
aws_instance.myec2-Dev: Still creating... [30s elapsed]
aws_instance.myec2-prod: Still creating... [30s elapsed]
aws_instance.myec2-prod: Creation complete after 32s [id=i-06ac2ba04af3b368d]
aws_instance.myec2-Dev: Creation complete after 32s [id=i-04ec1221755216d81]
aws_instance.myec2-test: Still creating... [40s elapsed]
aws_instance.myec2-test: Creation complete after 42s [id=i-0d5dbcb9e68dce752]

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
root@Terraform-Server:~/myproject#
```



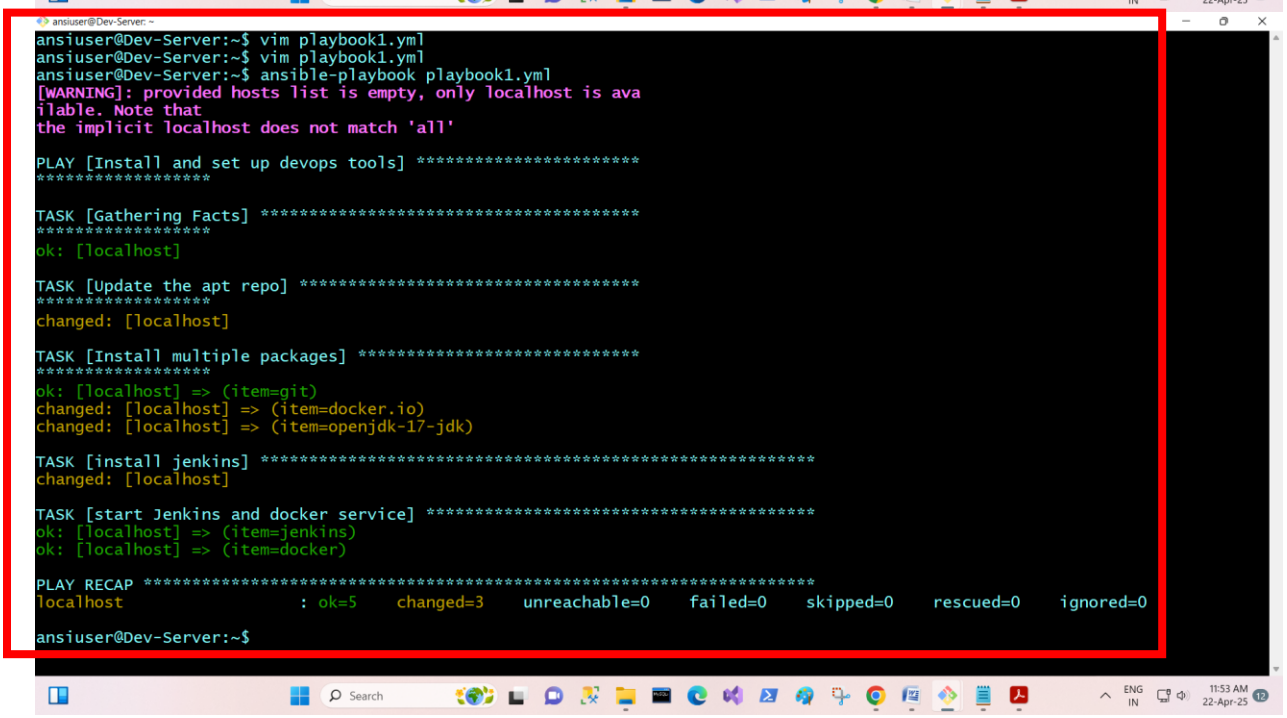
Step4: Install all the tools for CICD like (Ansible, Jenkins, Docker, Etc…) on DEV-Server



Step5: Create a Playbook for Install DevOps Tools on Dev-Server

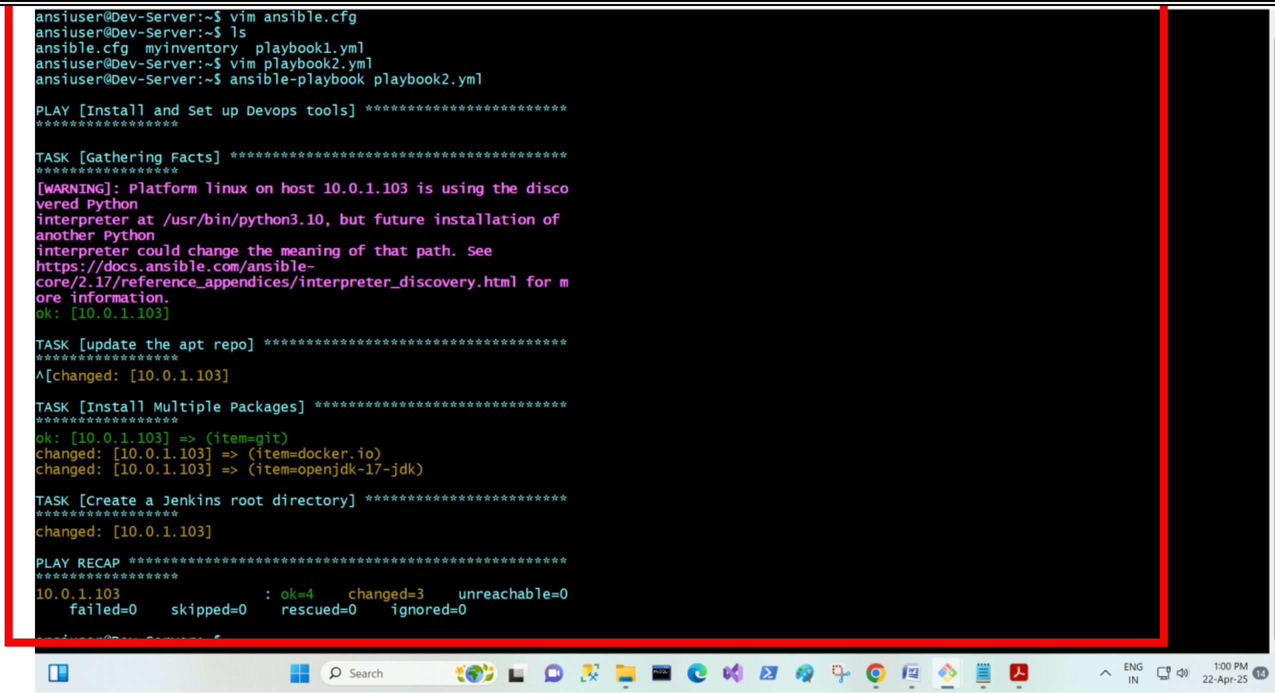


```
- name: Install and set up devops tools
  hosts: localhost
  become: true
  tasks:
  - name: Update the apt repo
    command: apt-get update
  - name: Install multiple packages
    package: name={{item}} state=present
    loop:
     - git
     - docker.io
     - openjdk-17-jdk
  - name: install jenkins
    command: sudo apt-get install jenkins -y
  - name: start Jenkins and docker service
    service: name={{item}} state=started
    loop:
    - jenkins
    - docker
```

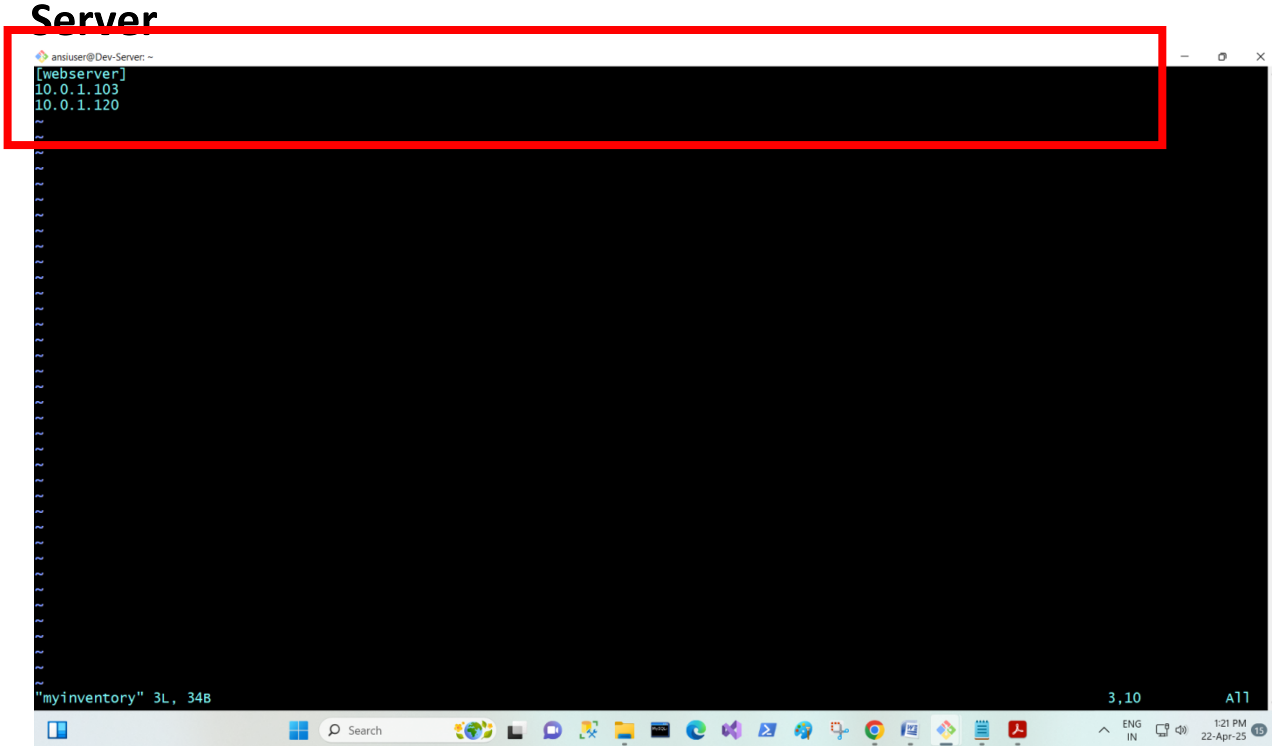


```
ansiuser@Dev-Server:~$ vim playbook1.yml
ansiuser@Dev-Server:~$ vim playbook1.yml
ansiuser@Dev-Server:~$ ansible-playbook playbook1.yml
[WARNING]: provided hosts list is empty, only localhost is ava
ilable. Note that
the implicit localhost does not match 'all'

PLAY [Install and set up devops tools] ************************
******************

TASK [Gathering Facts] ****************************************
******************
ok: [localhost]

TASK [Update the apt repo] ************************************
******************
changed: [localhost]

TASK [Install multiple packages] ******************************
******************
ok: [localhost] => (item=git)
changed: [localhost] => (item=docker.io)
changed: [localhost] => (item=openjdk-17-jdk)

TASK [install jenkins] ****************************************************
changed: [localhost]

TASK [start Jenkins and docker service] ***********************************
ok: [localhost] => (item=jenkins)
ok: [localhost] => (item=docker)

PLAY RECAP ********************************************************************
localhost                  : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansiuser@Dev-Server:~$
```

Step6: Create a Playbook2 for Install DevOps Tools on Test-Server

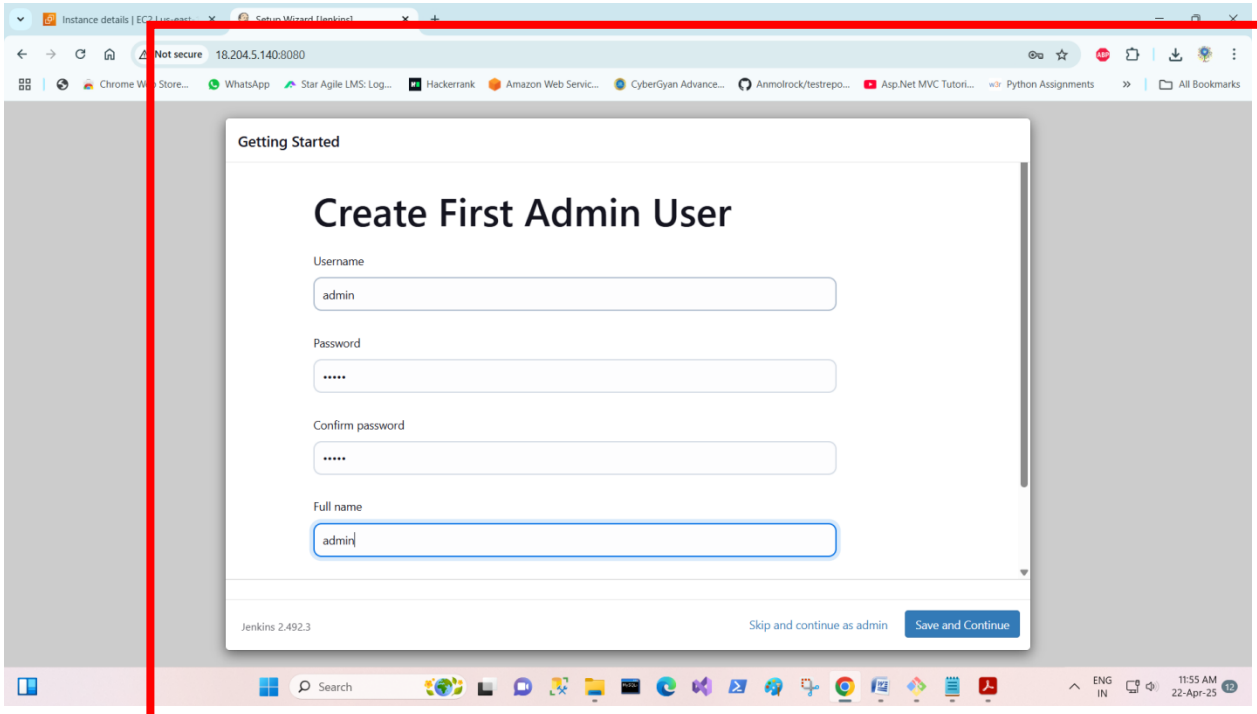


Step7: Create a Playbook2 for Install DevOps Tools on PROD-Server

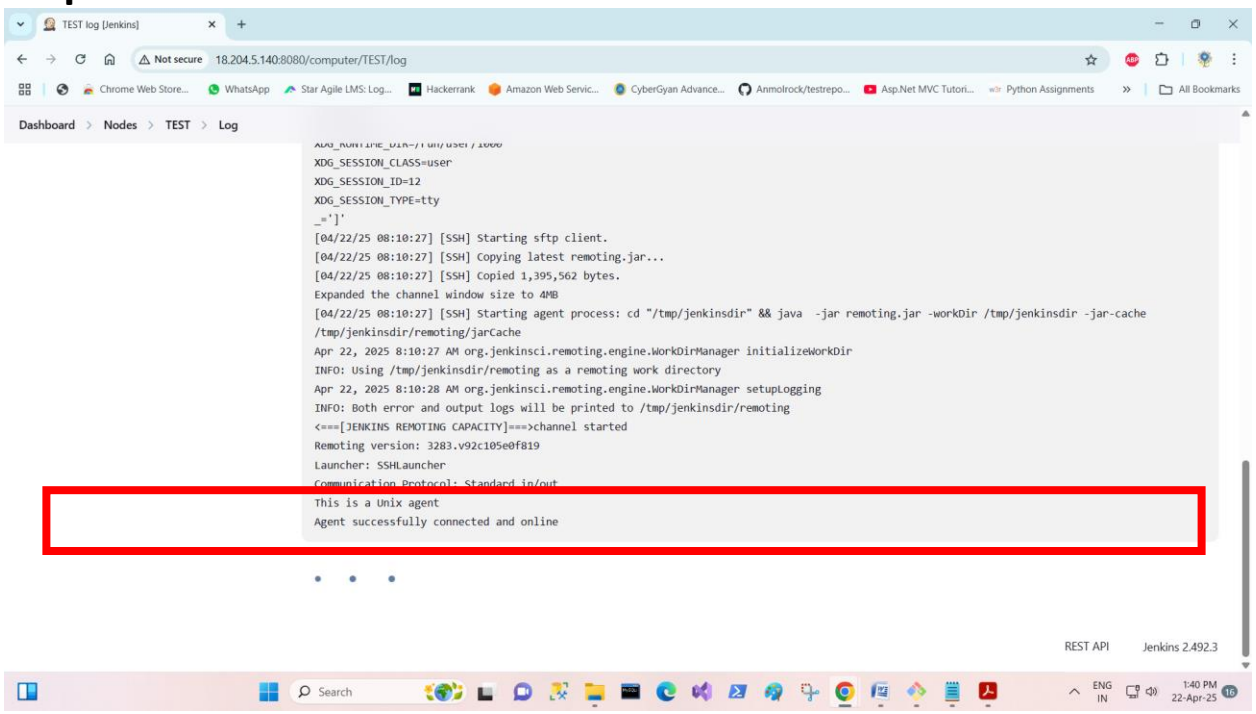


```
thon
interpreter at /usr/bin/python3.10, but future installation of another
Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more info
rmation.
ok: [10.0.1.120]
[WARNING]: Platform linux on host 10.0.1.103 is using the discovered Py
thon
interpreter at /usr/bin/python3.10, but future installation of another
Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.17/reference_appendices/interpreter_discovery.html for more info
rmation.
ok: [10.0.1.103]

TASK [update the apt repo] *****************************************
**********
changed: [10.0.1.103]
changed: [10.0.1.120]

TASK [Install Multiple Packages] ***********************************
**********
ok: [10.0.1.120] => (item=git)
ok: [10.0.1.103] => (item=git)
ok: [10.0.1.103] => (item=docker.io)
ok: [10.0.1.103] => (item=openjdk-17-jdk)
changed: [10.0.1.120] => (item=docker.io)
changed: [10.0.1.120] => (item=openjdk-17-jdk)

TASK [Create a Jenkins root directory] *****************************************
ok: [10.0.1.103]
changed: [10.0.1.120]

PLAY RECAP *********************************************************************
10.0.1.103                 : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
10.0.1.120                 : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ansiuser@Dev-Server:~$
```

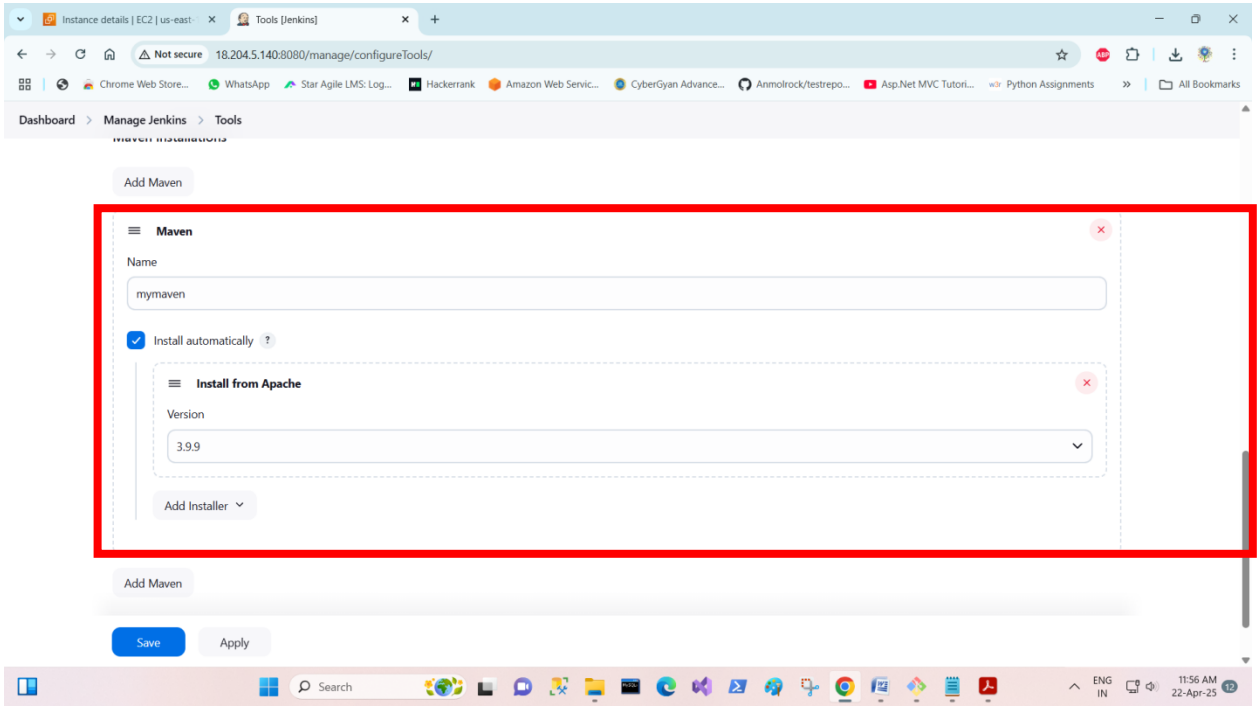
Step8: Setup Jenkins Dashboard



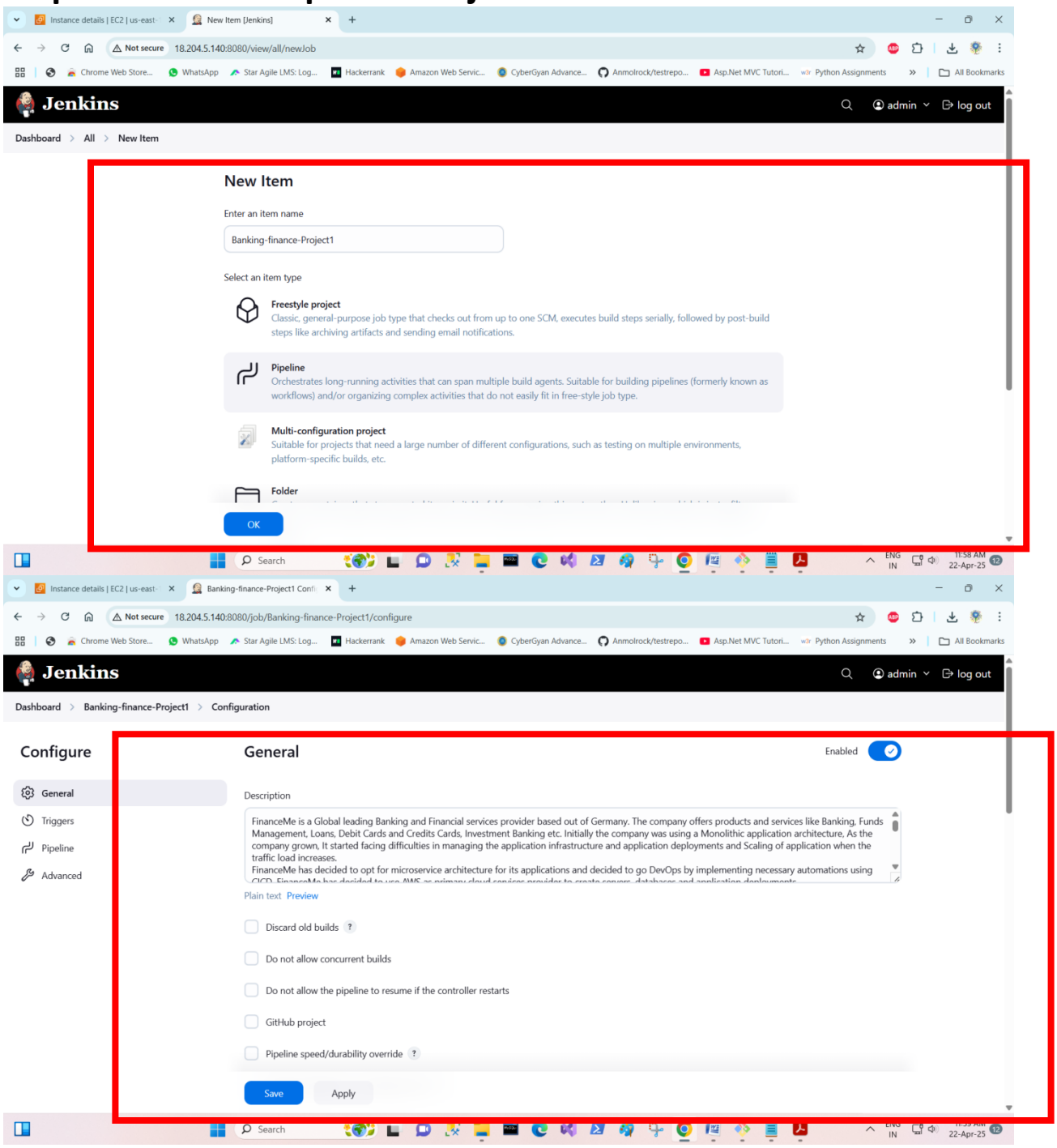
Step9: Then Create a Worker Nodes in Jenkins

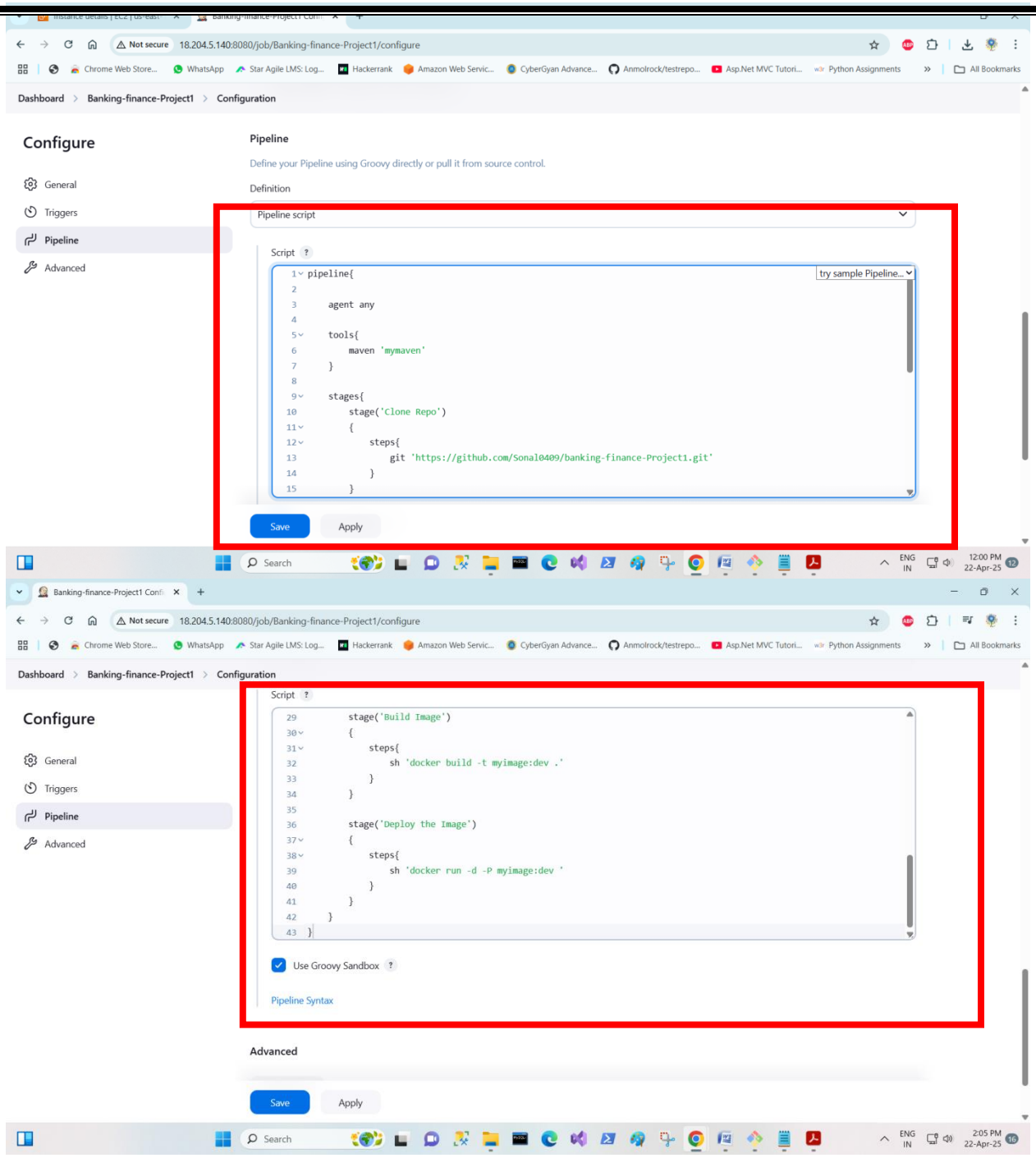


Step9: Setup Maven in Jenkins

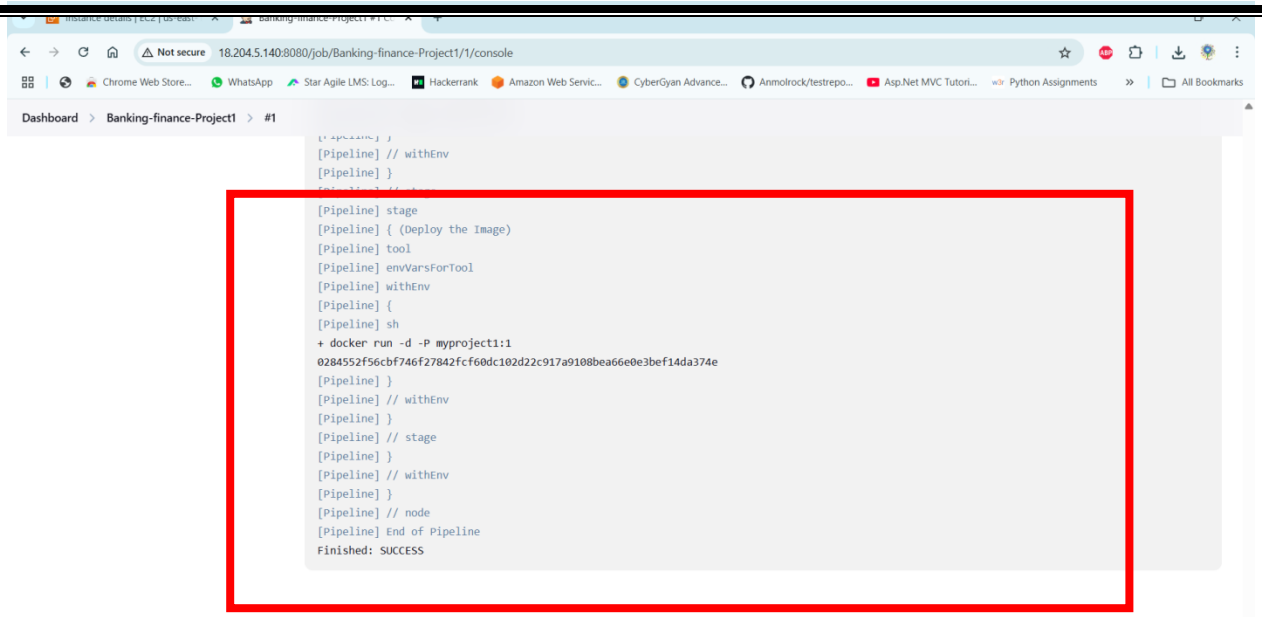


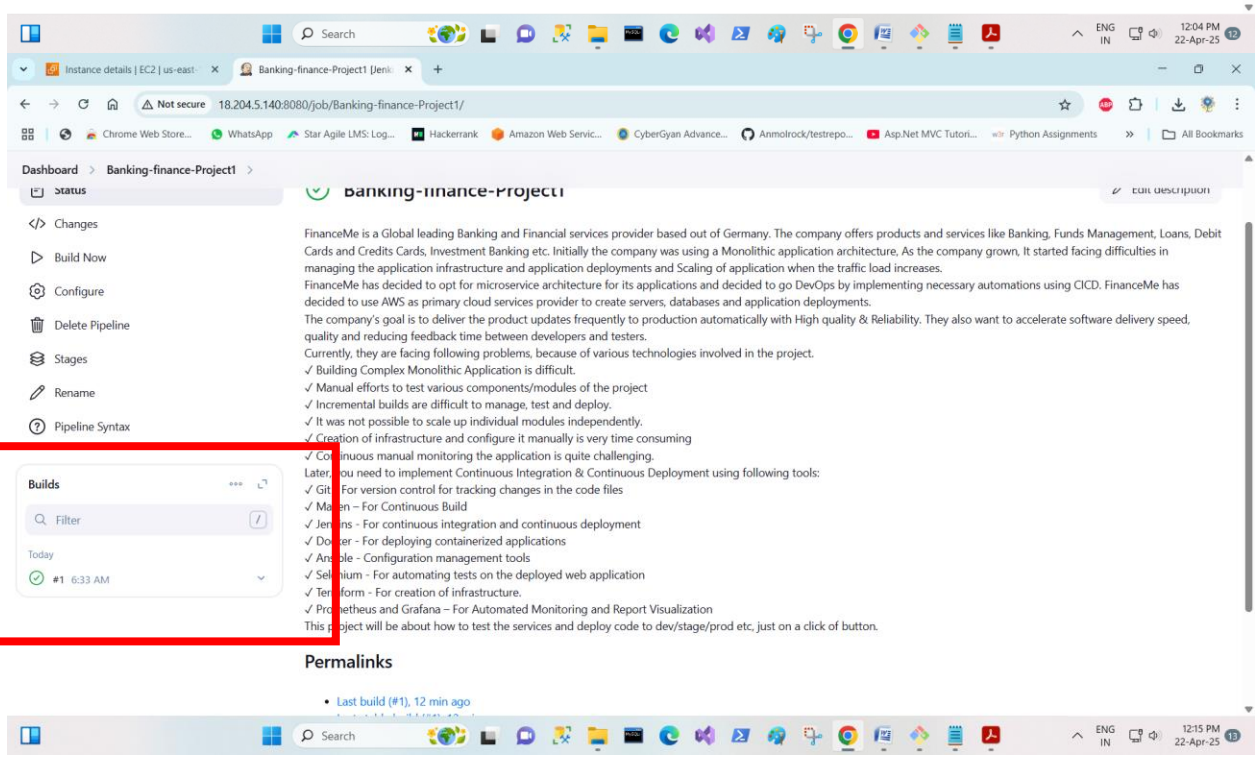
Step10: Create a Pipeline Project in Jenkins

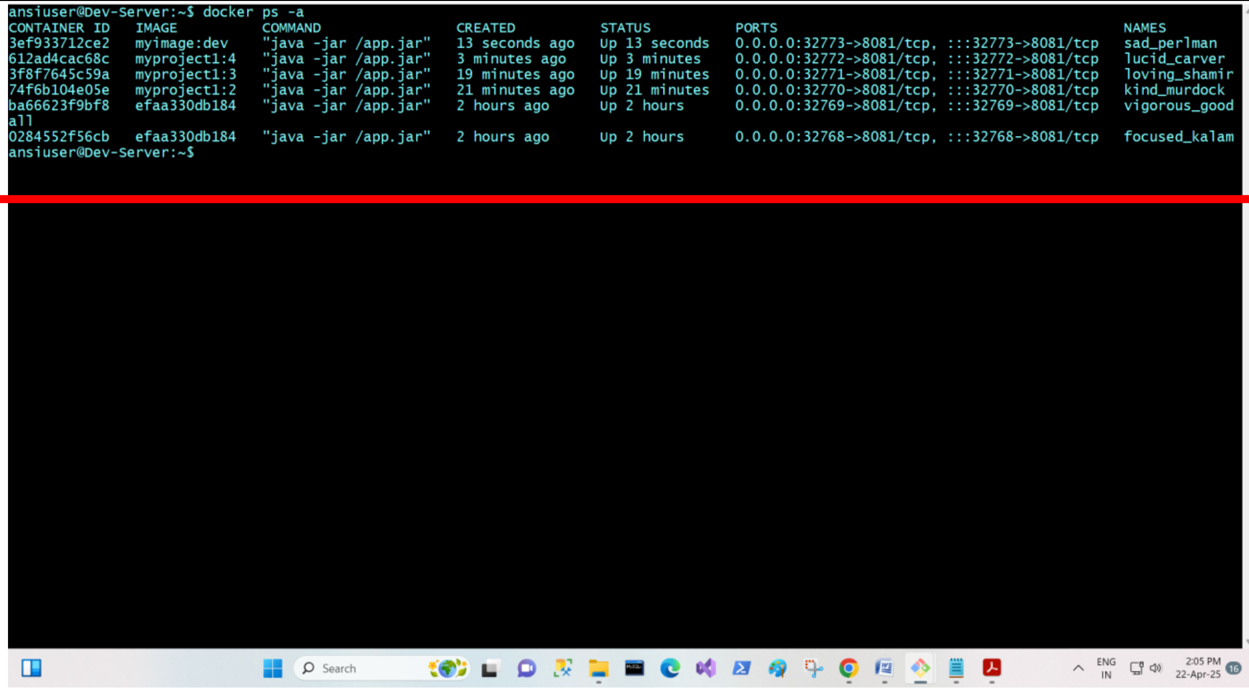



Not secure 18.204.5.140:8080/job/Banking-finance-Project1/configure
Dashboard > Banking-finance-Project1 > Configuration
Configure
General
Triggers
Pipeline
Advanced
Pipeline
Define your Pipeline using Groovy directly or pull it from source control.
Definition
Pipeline script
Script
try sample Pipeline...
pipeline{
agent any
tools{
maven 'mymaven'
}
stages{
stage('Clone Repo')
{
steps{
git 'https://github.com/Sonal0409/banking-finance-Project1.git'
}
}
Save
Apply

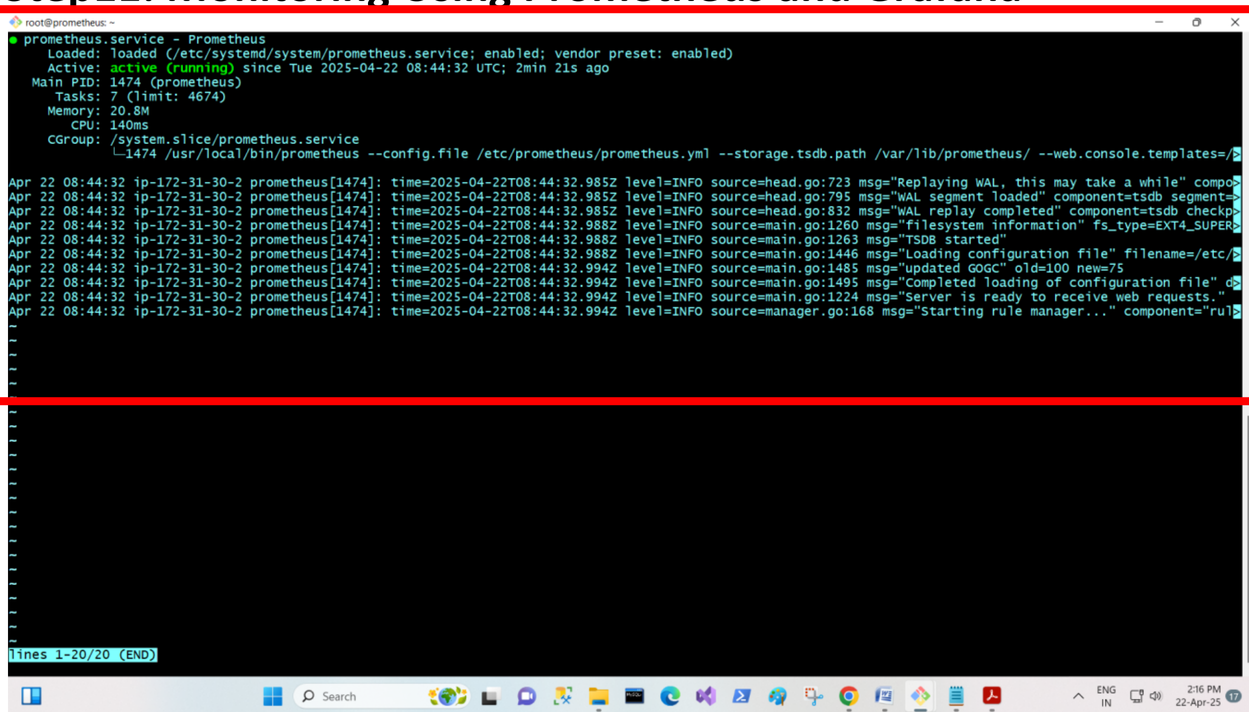
Not secure 18.204.5.140:8080/job/Banking-finance-Project1/configure
Dashboard > Banking-finance-Project1 > Configuration
Configure
General
Triggers
Pipeline
Advanced
Script
stage('Build Image')
{
steps{
sh 'docker build -t myimage:dev .'
}
}
stage('Deploy the Image')
{
steps{
sh 'docker run -d -P myimage:dev '
}
}
}
}
Use Groovy Sandbox
Pipeline Syntax
Advanced
Save
Apply

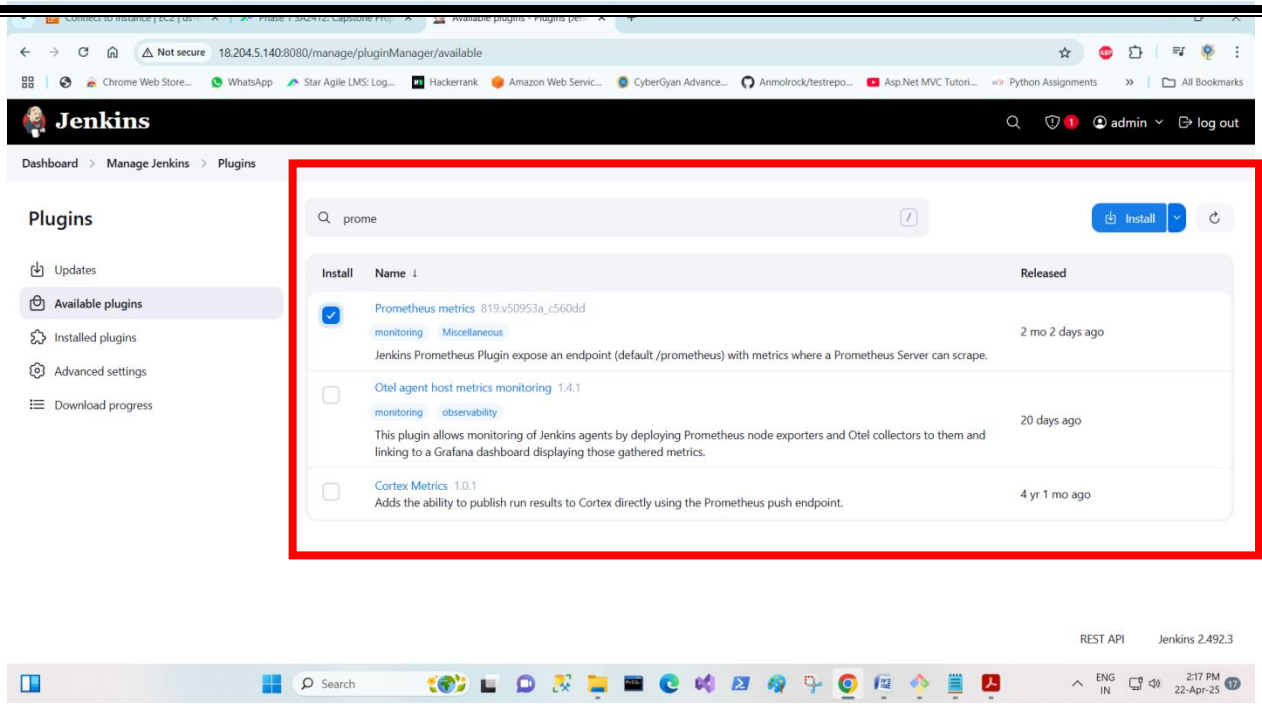

Not secure 18.204.5.140:8080/job/Banking-finance-Project1/1/console
Dashboard > Banking-finance-Project1 > #1
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] stage
[Pipeline] { (Deploy the Image)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker run -d -P myproject1:1
0284552f56cbf746f27842fcf60dc102d22c917a9108bea66e0e3bef14da374e
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
REST API
Jenkins 2.492.3

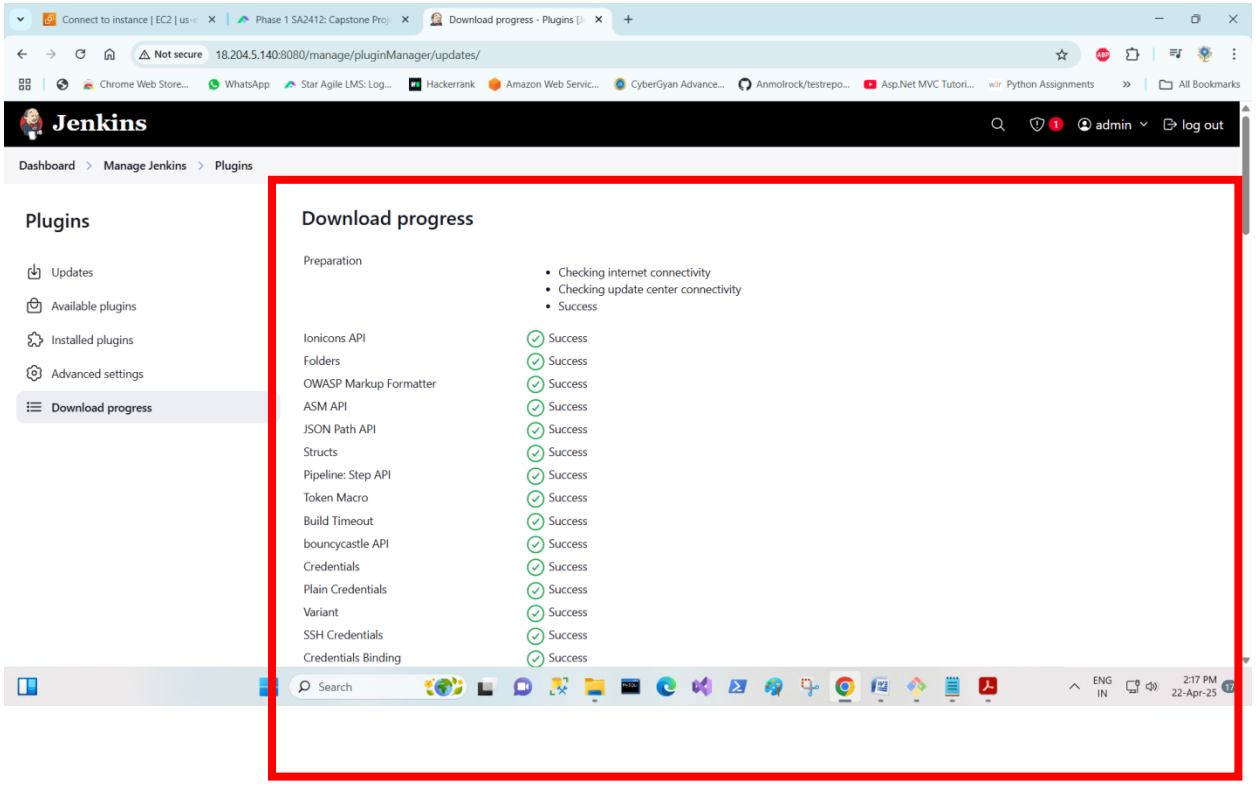

Not secure 18.204.5.140:8080/job/Banking-finance-Project1/
Dashboard > Banking-finance-Project1 >
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Builds
Filter
Today
#1 6:33 AM
Banking-finance-Project1
FinanceMe is a Global leading Banking and Financial services provider based out of Germany. The company offers products and services like Banking, Funds Management, Loans, Debit Cards and Credits Cards, Investment Banking etc. Initially the company was using a Monolithic application architecture, As the company grown, It started facing difficulties in managing the application infrastructure and application deployments and Scaling of application when the traffic load increases.
FinanceMe has decided to opt for microservice architecture for its applications and decided to go DevOps by implementing necessary automations using CICD. FinanceMe has decided to use AWS as primary cloud services provider to create servers, databases and application deployments.
The company's goal is to deliver the product updates frequently to production automatically with High quality & Reliability. They also want to accelerate software delivery speed, quality and reducing feedback time between developers and testers.
Currently, they are facing following problems, because of various technologies involved in the project.
✓ Building Complex Monolithic Application is difficult.
✓ Manual efforts to test various components/modules of the project
✓ Incremental builds are difficult to manage, test and deploy.
✓ It was not possible to scale up individual modules independently.
✓ Creation of infrastructure and configure it manually is very time consuming
✓ Continuous manual monitoring the application is quite challenging.
Later, you need to implement Continuous Integration & Continuous Deployment using following tools:
✓ Git – For version control for tracking changes in the code files
✓ Maven – For Continuous Build
✓ Jenkins - For continuous integration and continuous deployment
✓ Docker - For deploying containerized applications
✓ Ansible - Configuration management tools
✓ Selenium - For automating tests on the deployed web application
✓ Terraform - For creation of infrastructure.
✓ Prometheus and Grafana – For Automated Monitoring and Report Visualization
This project will be about how to test the services and deploy code to dev/stage/prod etc, just on a click of button.
Permalinks
Last build (#1), 12 min ago

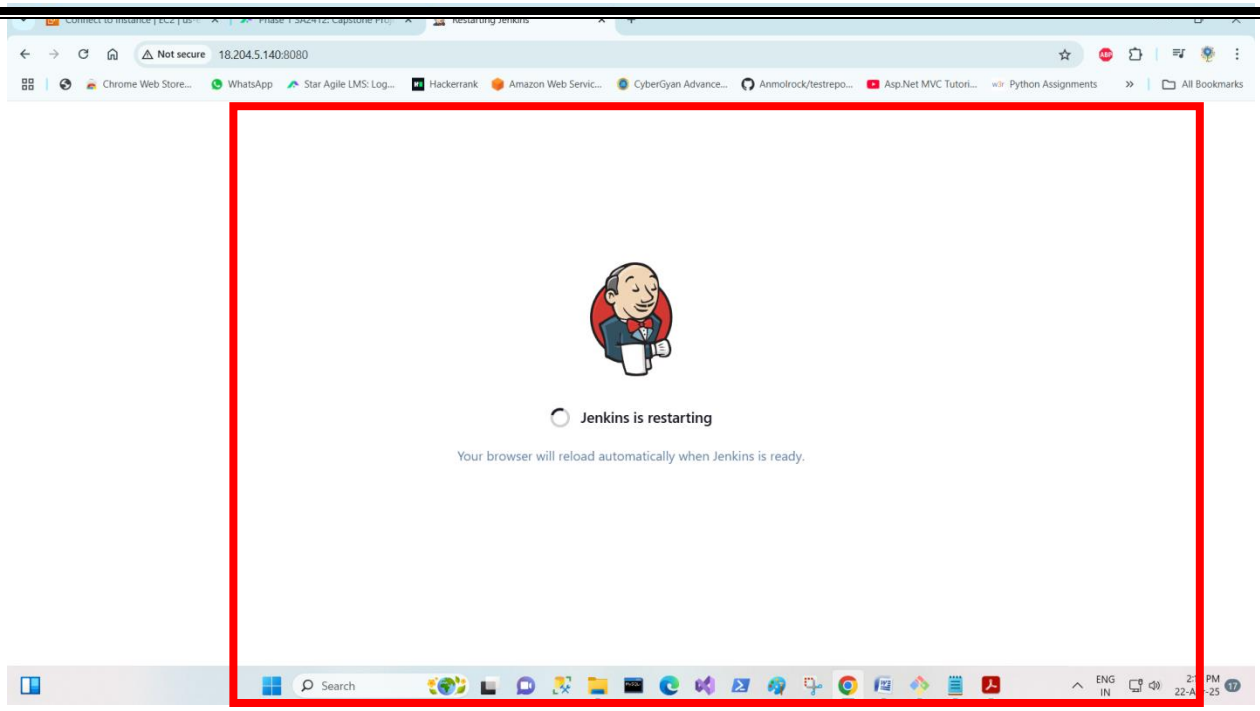


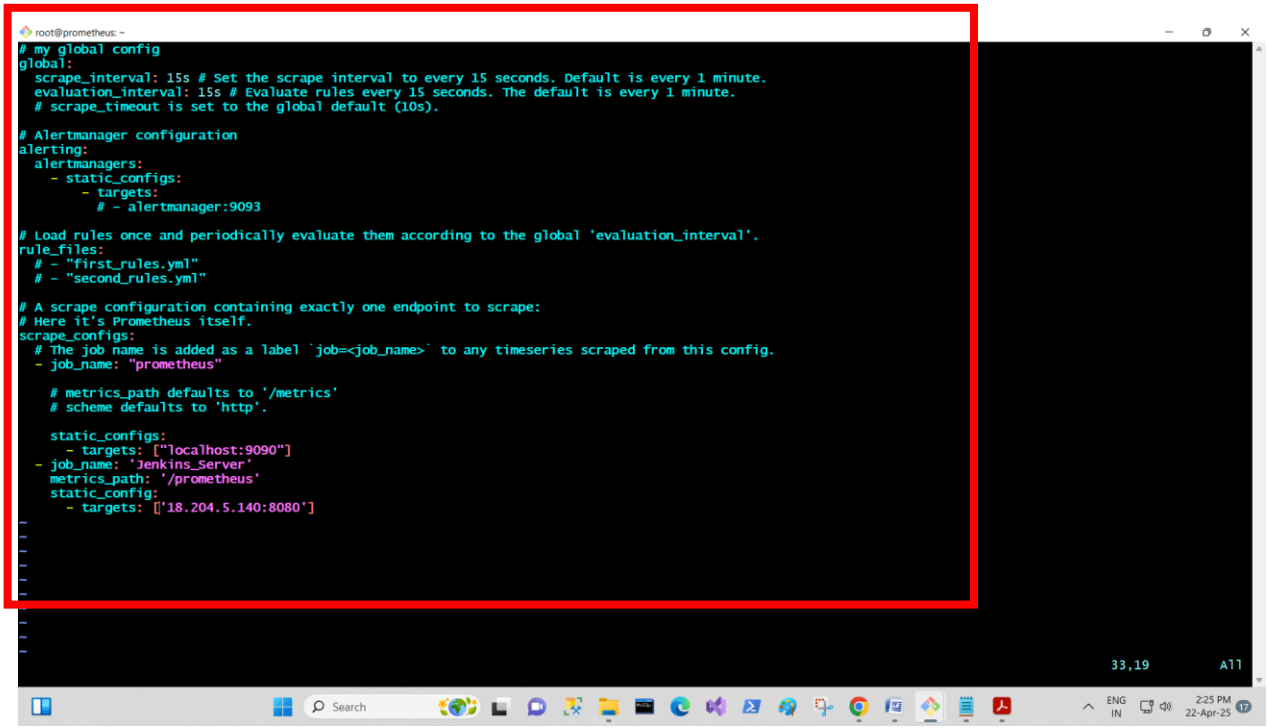
Step11: Monitoring Using Prometheus and Grafana

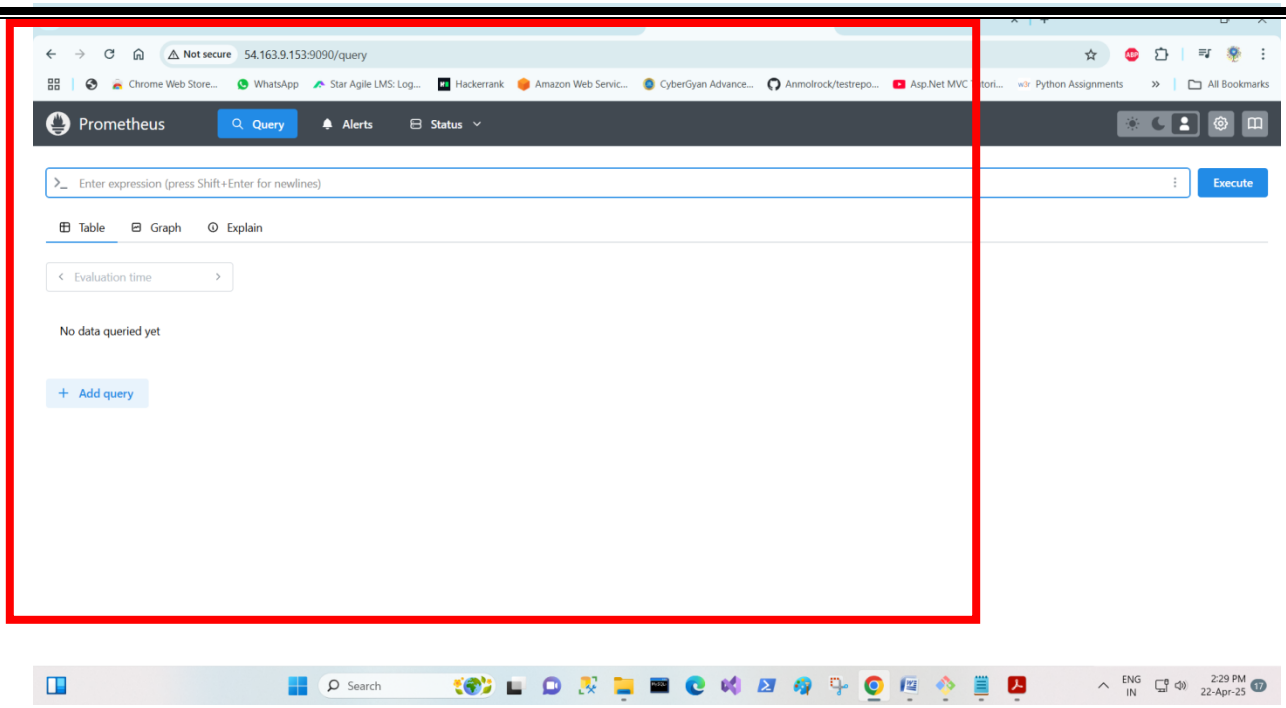



Not secure 18.204.5.140:8080/manage/pluginManager/available
Jenkins
admin
log out
Dashboard > Manage Jenkins > Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
prome
Install
Name
Released
Prometheus metrics 819.v50953a_c560dd
monitoring
Miscellaneous
Jenkins Prometheus Plugin expose an endpoint (default /prometheus) with metrics where a Prometheus Server can scrape.
2 mo 2 days ago
Otel agent host metrics monitoring 1.4.1
monitoring
observability
This plugin allows monitoring of Jenkins agents by deploying Prometheus node exporters and Otel collectors to them and linking to a Grafana dashboard displaying those gathered metrics.
20 days ago
Cortex Metrics 1.0.1
Adds the ability to publish run results to Cortex directly using the Prometheus push endpoint.
4 yr 1 mo ago
REST API
Jenkins 2.492.3

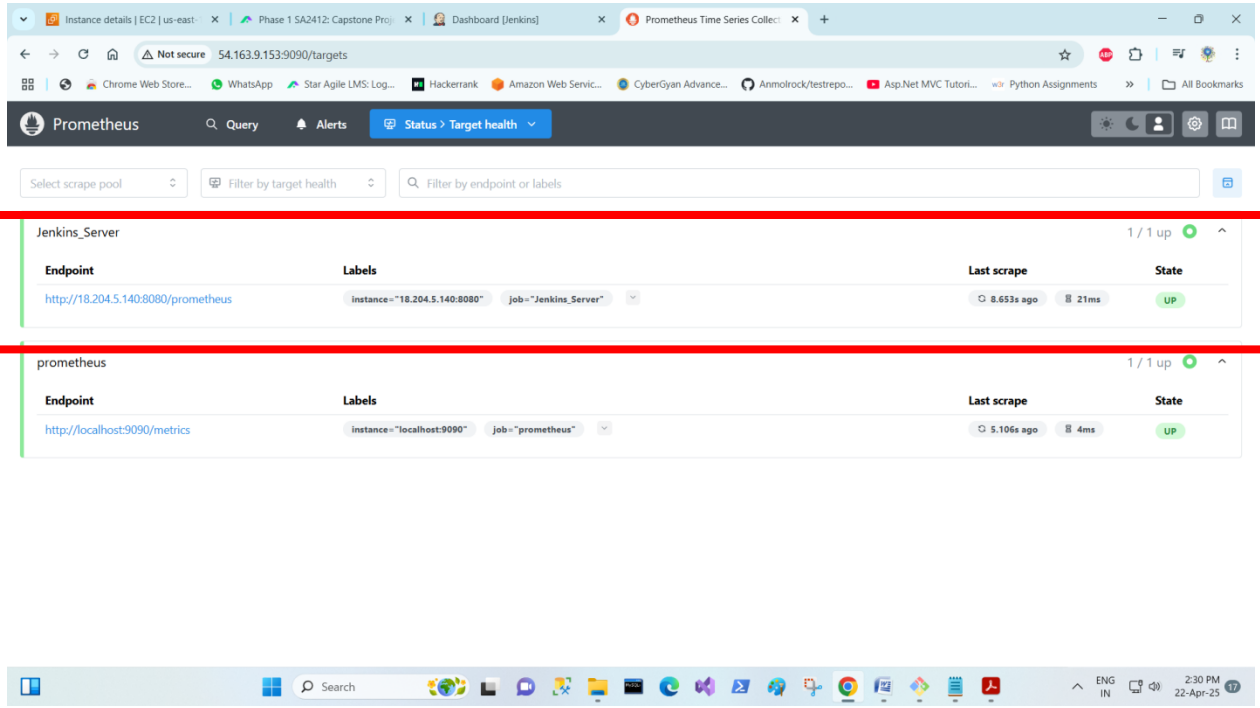

Not secure 18.204.5.140:8080/manage/pluginManager/updates/
Jenkins
admin
log out
Dashboard > Manage Jenkins > Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update center connectivity
Success
Ionicons API Success
Folders Success
OWASP Markup Formatter Success
ASM API Success
JSON Path API Success
Structs Success
Pipeline: Step API Success
Token Macro Success
Build Timeout Success
bouncycastle API Success
Credentials Success
Plain Credentials Success
Variant Success
SSH Credentials Success
Credentials Binding Success


Not secure 18.204.5.140:8080
Jenkins is restarting
Your browser will reload automatically when Jenkins is ready.

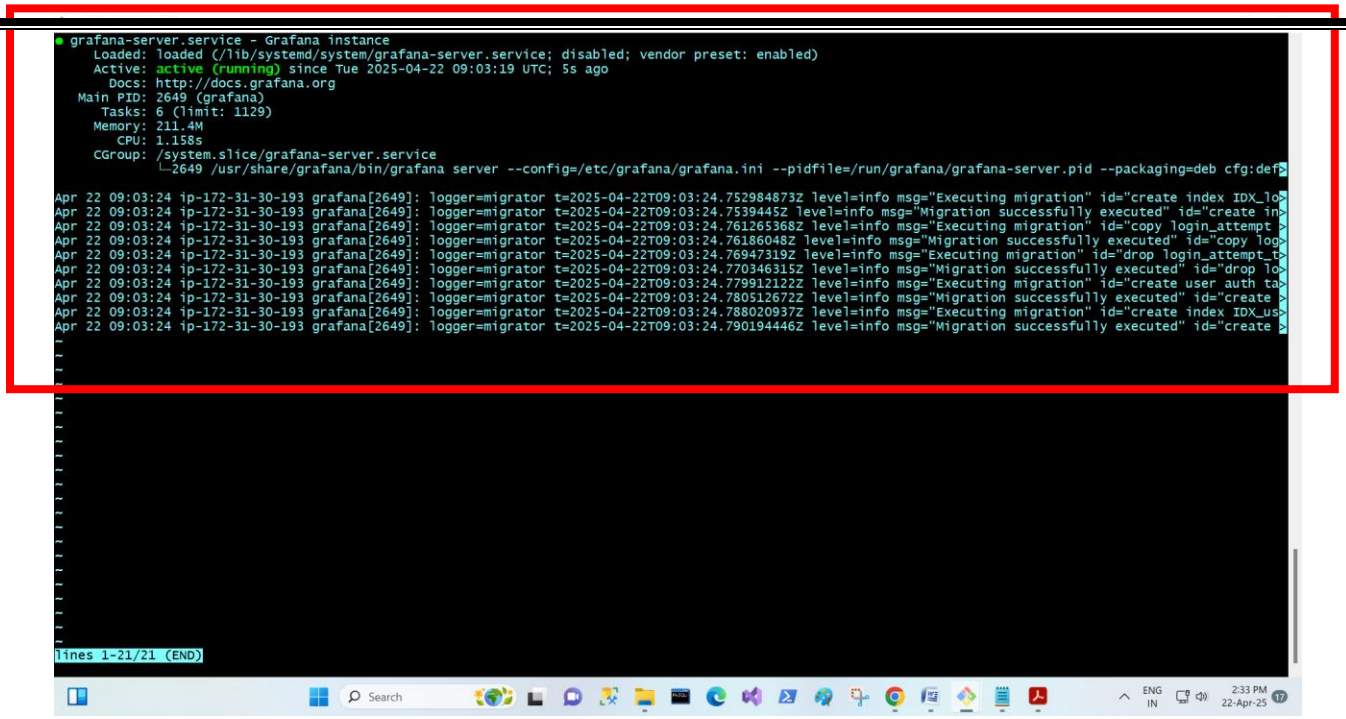

root@prometheus: ~
my global config
global:
scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
scrape_timeout is set to the global default (10s).
Alertmanager configuration
alerting:
alertmanagers:
- static_configs:
- targets:
- alertmanager:9093
Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
- "first_rules.yml"
- "second_rules.yml"
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: 'Jenkins_Server'
metrics_path: '/prometheus'
static_config:
- targets: ['18.204.5.140:8080']
33,19 All

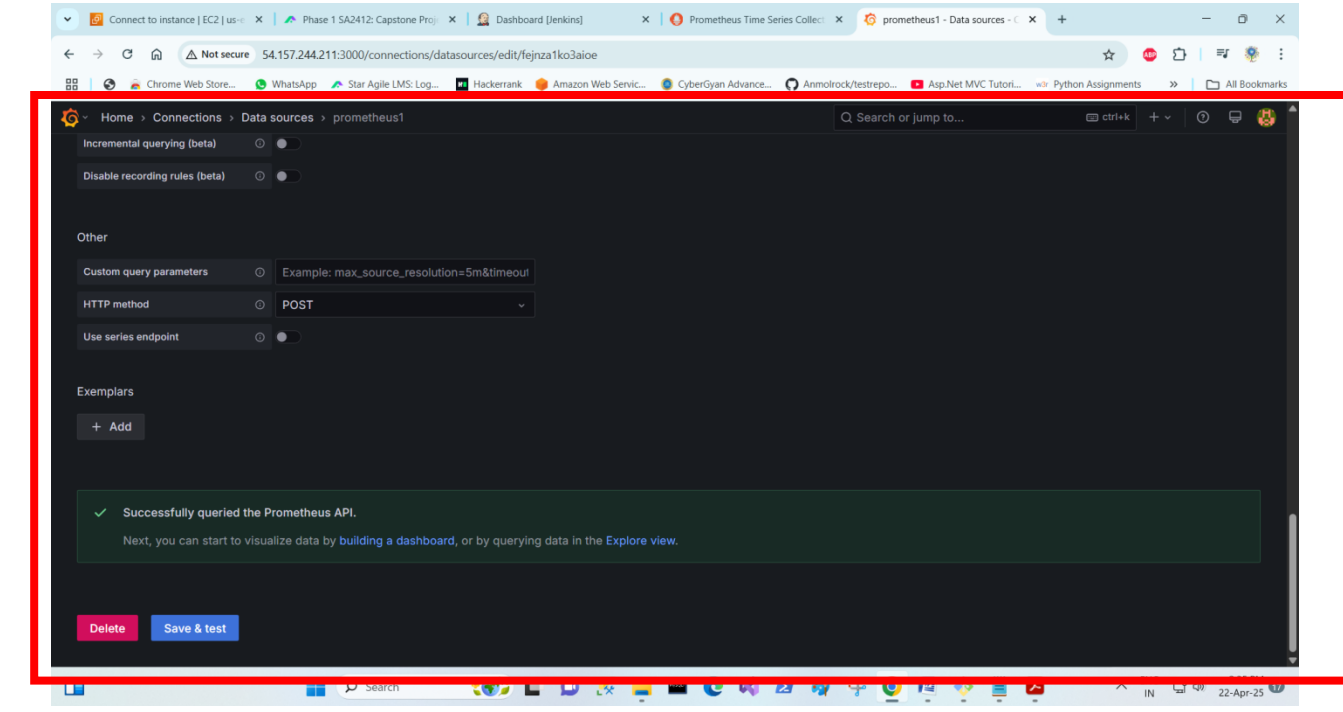

Not secure 54.163.9.153:9090/query
Prometheus
Query
Alerts
Status
Enter expression (press Shift+Enter for newlines)
Execute
Table
Graph
Explain
Evaluation time
No data queried yet
Add query

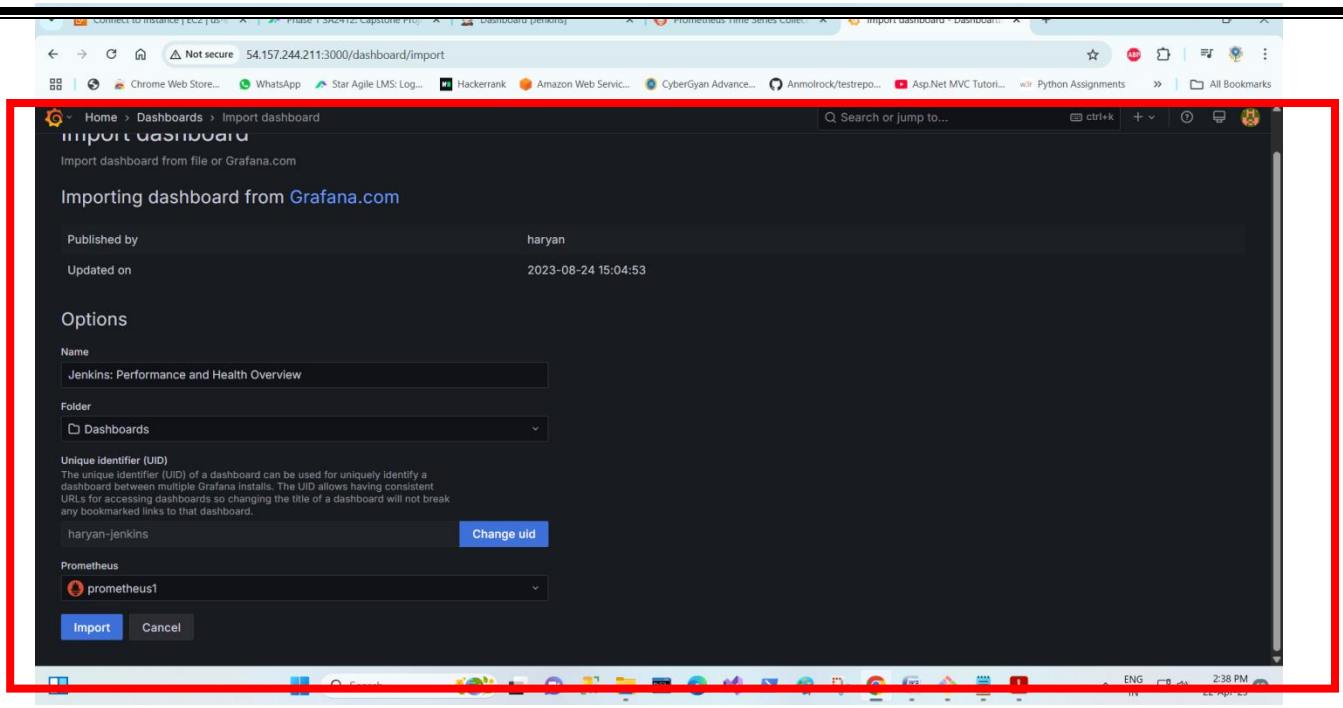
Search
ENG
IN
2:29 PM
22-Apr-25

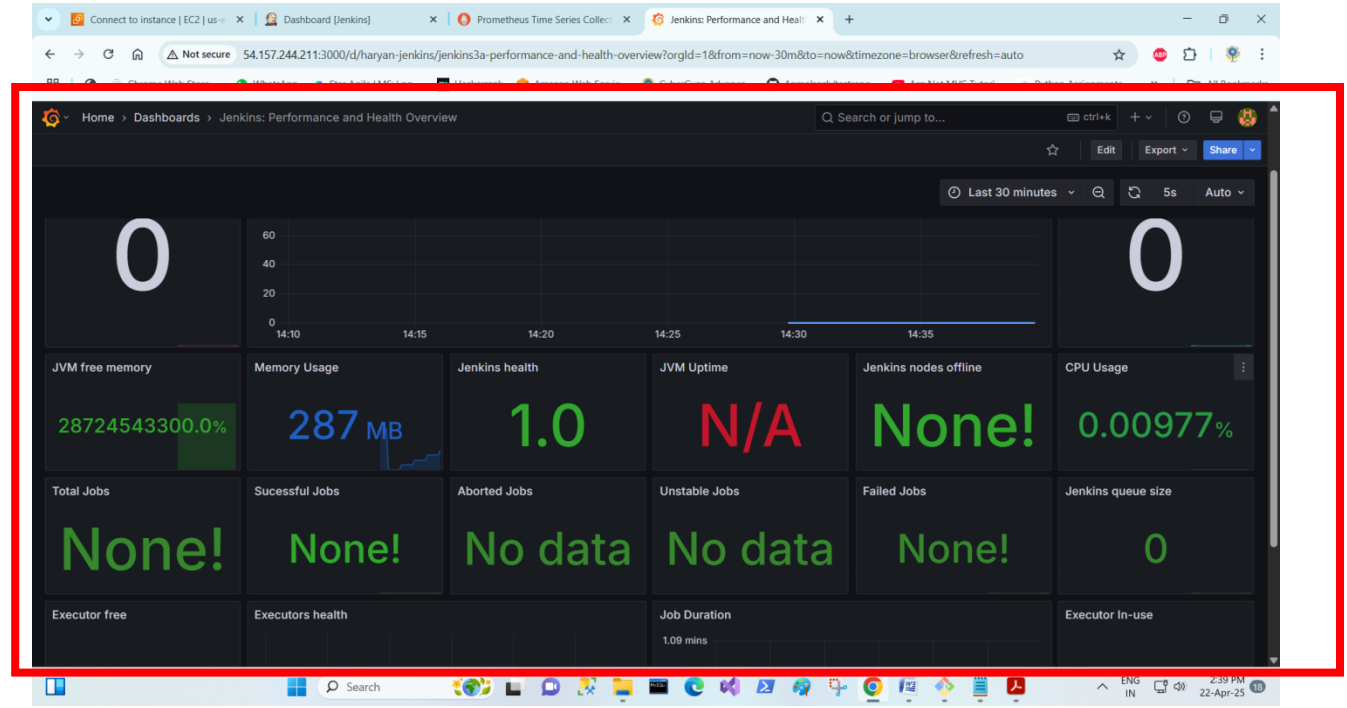

Not secure 54.163.9.153:9090/targets
Prometheus
Query
Alerts
Status > Target health
Select scrape pool
Filter by target health
Filter by endpoint or labels
Jenkins_Server
1 / 1 up
Endpoint
Labels
Last scrape
State
http://18.204.5.140:8080/prometheus
instance="18.204.5.140:8080"
job="Jenkins_Server"
8.653s ago
21ms
UP
prometheus
1 / 1 up
Endpoint
Labels
Last scrape
State
http://localhost:9090/metrics
instance="localhost:9090"
job="prometheus"
5.106s ago
4ms
UP

Search
ENG
IN
2:30 PM
22-Apr-25

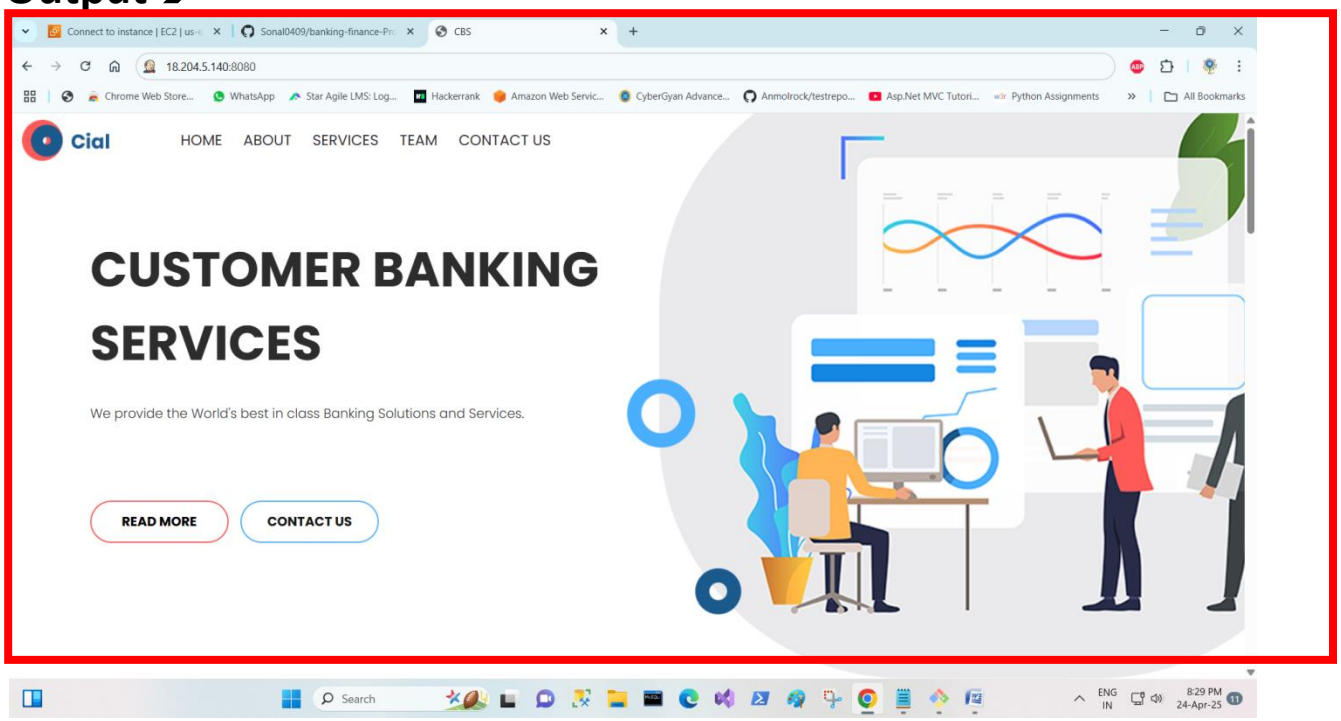

● grafana-server.service - Grafana instance
Loaded: loaded (/lib/systemd/system/grafana-server.service; disabled; vendor preset: enabled)
Active: active (running) since Tue 2025-04-22 09:03:19 UTC; 5s ago
Docs: http://docs.grafana.org
Main PID: 2649 (grafana)
Tasks: 6 (limit: 1129)
Memory: 211.4M
CPU: 1.158s
CGroup: /system.slice/grafana-server.service
└─2649 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana-server.pid --packaging=deb cfg:def
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.752984873Z level=info msg="Executing migration" id="create index IDX_lo
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.7539445Z level=info msg="Migration successfully executed" id="create in
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.761265368Z level=info msg="Executing migration" id="copy login_attempt
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.76186048Z level=info msg="Migration successfully executed" id="copy log
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.76947319Z level=info msg="Executing migration" id="drop login_attempt_t
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.770346315Z level=info msg="Migration successfully executed" id="drop lo
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.779912122Z level=info msg="Executing migration" id="create user auth ta
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.780512672Z level=info msg="Migration successfully executed" id="create
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.788020937Z level=info msg="Executing migration" id="create index IDX_us
Apr 22 09:03:24 ip-172-31-30-193 grafana[2649]: logger=migrator t=2025-04-22T09:03:24.790194446Z level=info msg="Migration successfully executed" id="create
lines 1-21/21 (END)
Search
ENG IN
2:33 PM 22-Apr-25

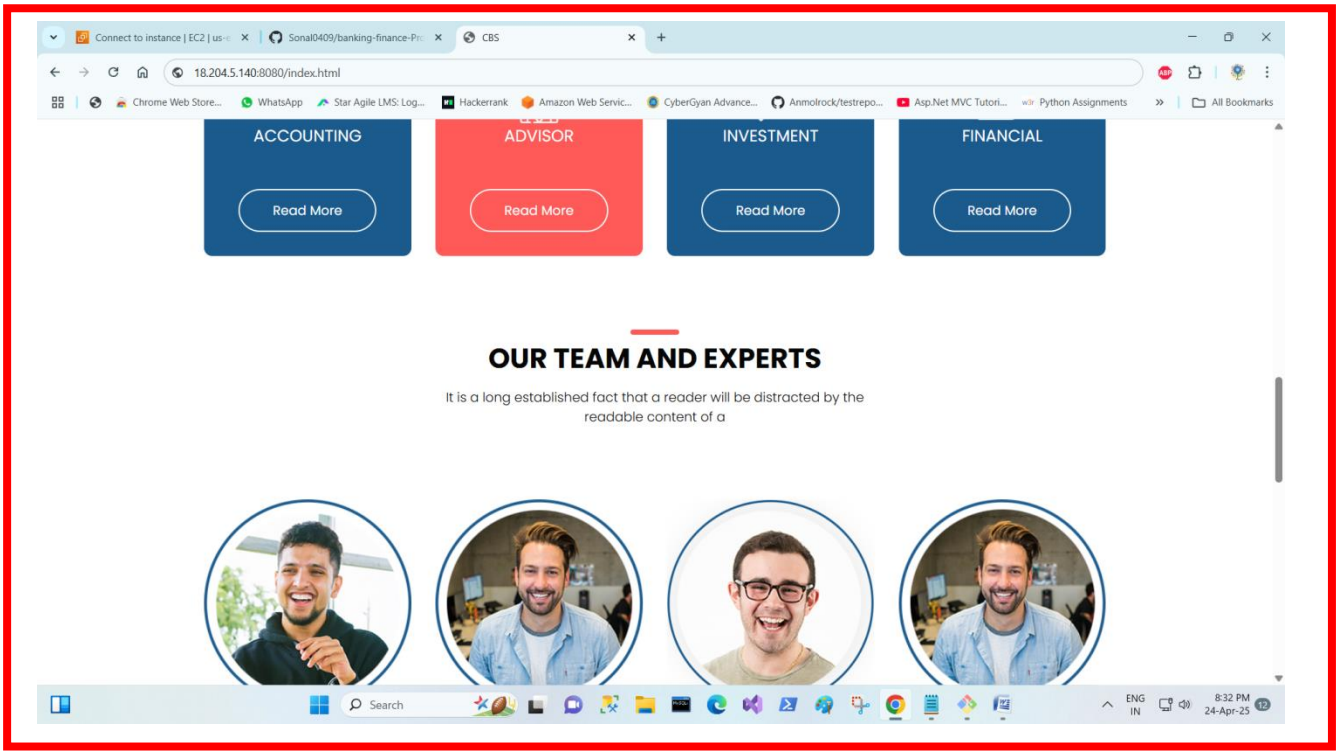

Connect to instance | EC2 | us-
Phase 1 SA2412: Capstone Proj
Dashboard [Jenkins]
Prometheus Time Series Collect
prometheus1 - Data sources -
Not secure
54.157.244.211:3000/connections/datasources/edit/fejnza1ko3aioe
Chrome Web Store...
WhatsApp
Star Agile LMS: Log...
Hackerrank
Amazon Web Servic...
CyberGyan Advance...
Animolrock/testrepo...
Asp.Net MVC Tutori...
Python Assignments
All Bookmarks
Home › Connections › Data sources › prometheus1
Search or jump to...
ctrl+k
Incremental querying (beta)
Disable recording rules (beta)
Other
Custom query parameters
Example: max_source_resolution=5m&timeout
HTTP method
POST
Use series endpoint
Exemplars
+ Add
Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Delete
Save & test
Search
IN
22-Apr-25


Not secure 54.157.244.211:3000/dashboard/import
Home › Dashboards › Import dashboard
Search or jump to...
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
haryan
Updated on
2023-08-24 15:04:53
Options
Name
Jenkins: Performance and Health Overview
Folder
Dashboards
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
haryan-jenkins
Change uid
Prometheus
prometheus1
Import
Cancel


Not secure 54.157.244.211:3000/d/haryan-jenkins/jenkins3a-performance-and-health-overview?orgId=1&from=now-30m&to=now&timezone=browser&refresh=auto
Home › Dashboards › Jenkins: Performance and Health Overview
Search or jump to...
Edit
Export
Share
Last 30 minutes
5s
Auto
0
60
40
20
0
14:10
14:15
14:20
14:25
14:30
14:35
0
JVM free memory
28724543300.0%
Memory Usage
287 MB
Jenkins health
1.0
JVM Uptime
N/A
Jenkins nodes offline
None!
CPU Usage
0.00977%
Total Jobs
None!
Sucessful Jobs
None!
Aborted Jobs
No data
Unstable Jobs
No data
Failed Jobs
None!
Jenkins queue size
0
Executor free
Executors health
Job Duration
1.09 mins
Executor In-use

Output➔


Connect to instance | EC2 | us-
Sonal0409/banking-finance-Pr
CBS
18.204.5.140:8080
Chrome Web Store...
WhatsApp
Star Agile LMS: Log...
Hackerrank
Amazon Web Servic...
CyberGyan Advance...
Anmolrock/testrepo...
Asp.Net MVC Tutori...
Python Assignments
All Bookmarks
Cial
HOME
ABOUT
SERVICES
TEAM
CONTACT US
CUSTOMER BANKING SERVICES
We provide the World's best in class Banking Solutions and Services.
READ MORE
CONTACT US
Search
ENG
IN
8:29 PM
24-Apr-25


Connect to instance | EC2 | us-
Sonal0409/banking-finance-Pr
CBS
18.204.5.140:8080/index.html
Chrome Web Store...
WhatsApp
Star Agile LMS: Log...
Hackerrank
Amazon Web Servic...
CyberGyan Advance...
Anmolrock/testrepo...
Asp.Net MVC Tutori...
Python Assignments
All Bookmarks
ACCOUNTING
Read More
ADVISOR
Read More
INVESTMENT
Read More
FINANCIAL
Read More
OUR TEAM AND EXPERTS
It is a long established fact that a reader will be distracted by the readable content of a
Search
ENG
IN
8:32 PM
24-Apr-25