Deployment Architecture Document

This document outlines the deployment architecture for the project, detailing the processes and technologies used for development, staging, and production environments, setting up and managing various tasks and applications, including Prometheus, Grafana, Node Exporter, MySQL backup and migration, as well as CI/CD pipeline using Jenkins and Kubernetes cluster installation and configuration.

1.Git Repository Setup

Branches:

- **Development Branch**: For active development and feature integration.
- **Staging Branch**: For pre-production testing and QA.
- **Production Branch**: For live deployments.

Branch Protection Rules:

- Code reviews and approvals required for merges.
- Continuous Integration (CI) checks must pass before merging.

Git Workflow:

- **Feature Branches**: Created from the development branch.
- **Pull Requests**: For merging feature branches into development.
- **Release Process**: Merging development into staging for testing, then staging into production after approval.

Git command used:
Initialize the Git Repository

- **Create a new directory for your project:**

  mkdir git-project
  cd git-project

- **Initialize a new Git repository:**

git init

Create the Initial Branches
Firstly, we need to change branch from master to main:

git init –b main or,

git symbolic-ref HEAD refs/heads/main

- **Create and switch to the development branch:**

  git checkout -b development
  git commit –m “ ”
- **Create and switch to the staging branch:**

  git checkout -b staging
- **Create and switch to the production branch:**

  git checkout -b production
- **Switch back to the development branch to continue working:**

  git checkout development (branch change)

If you are in development branch and want to merge development brach with main branch, you can use git merge command.

git merge main

Push to Remote Repository

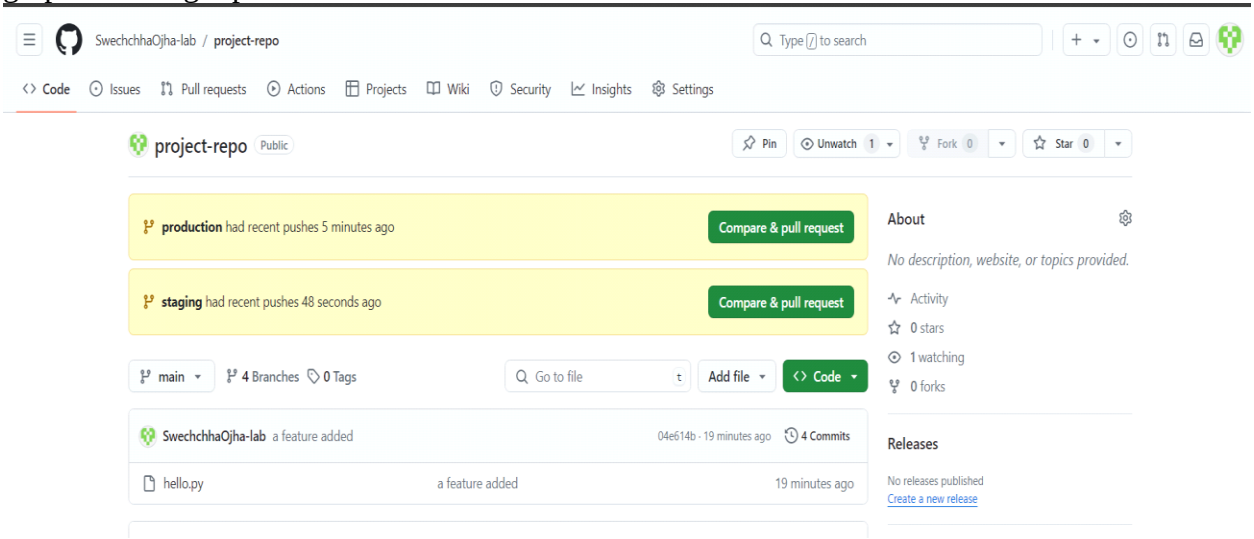
- **Add the remote repository:**

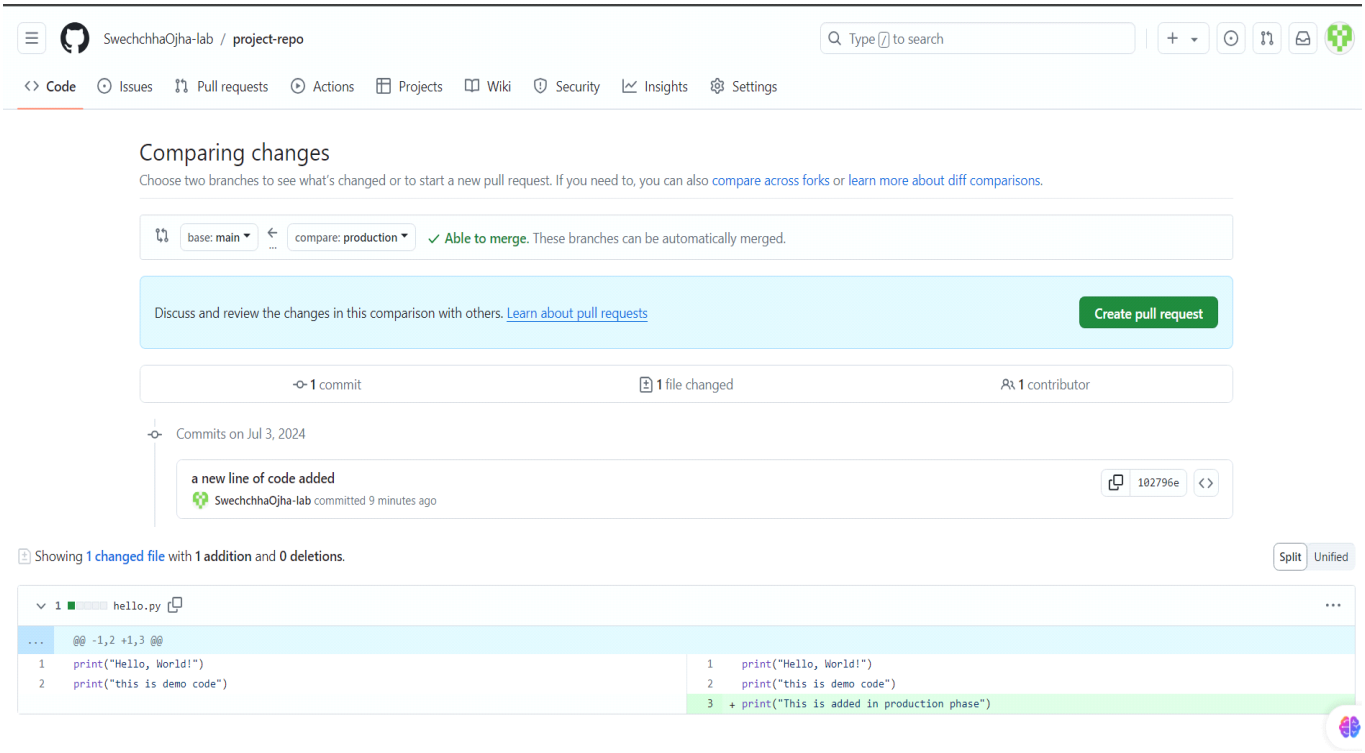
  git remote add origin git@github.com:SwechchhaOjha-lab/project-repo.git
- **Push the development branch:**

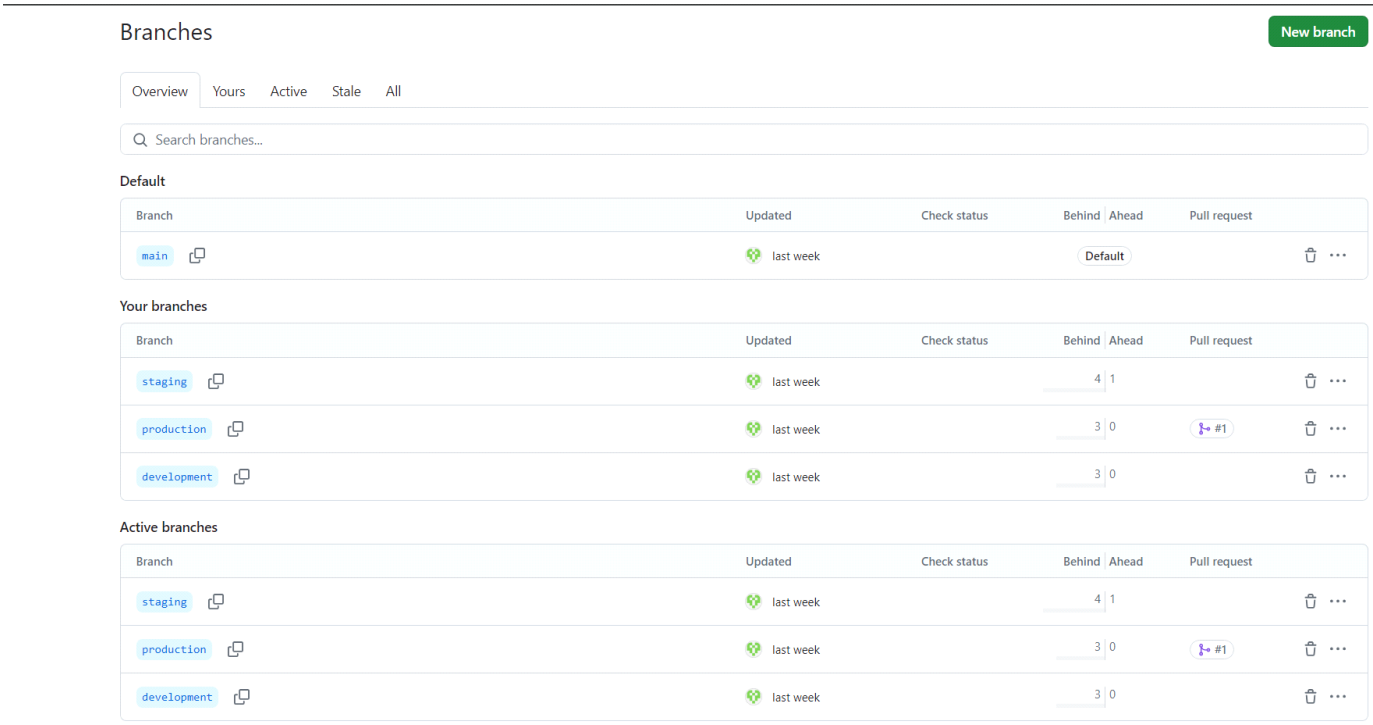
  git push -u origin development
- **Push the staging branch:**

  git push -u origin staging
- **Push the production branch:**

git push -u origin production




SwechchhaOjha-lab / project-repo
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Comparing changes
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.
base: main
compare: production
Able to merge. These branches can be automatically merged.
Discuss and review the changes in this comparison with others. Learn about pull requests
Create pull request
1 commit
1 file changed
1 contributor
Commits on Jul 3, 2024
a new line of code added
SwechchhaOjha-lab committed 9 minutes ago
102796e
Showing 1 changed file with 1 addition and 0 deletions.
Split
Unified
1 hello.py
@@ -1,2 +1,3 @@
print("Hello, World!")
print("this is demo code")
print("Hello, World!")
print("this is demo code")
+ print("This is added in production phase")


Branches
New branch
Overview
Yours
Active
Stale
All
Search branches...
Default
Branch
Updated
Check status
Behind
Ahead
Pull request
main
last week
Default
Your branches
Branch
Updated
Check status
Behind
Ahead
Pull request
staging
last week
4
1
production
last week
3
0
#1
development
last week
3
0
Active branches
Branch
Updated
Check status
Behind
Ahead
Pull request
staging
last week
4
1
production
last week
3
0
#1
development
last week
3
0

2. Infrastructure Definition using Vagrant

Vagrant Configuration:

- Vagrantfile defines the environment.
- Providers: VirtualBox
- Base Boxes: Windows

Create a directory for your Vagrant project.

mkdir vagrant_test

cd vagrant_test

vagrant init

(Initialize a new Vagrant environment. This will create a Vagrantfile in your project directory.)

```
Vagrant.configure("2") do |config|

  config.vm.define "Centos-test-vm" do |centos1|
    centos1.vm.box = "centos/7"
    centos1.vm.network "forwarded_port", guest: 80, host: 8086

  #config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
  centos1.vm.network "private_network", ip: "192.168.56.10", name: "VirtualBox Host-Only Ethernet Adapter"
  centos1.vm.network "public_network", type: "dhcp", bridge: "Intel(R) Dual Band Wireless-AC 8265"
  centos1.vm.hostname="devops.com"
  centos1.vm.provider "virtualbox" do |vb|

    vb.memory = "1024"
    vb.cpus= "1"
    vb.name= "centos vm 1"
  end


   config.vm.provision "shell", inline: <<-SHELL
      yum install -y epel-release
      yum install -y nginx
      systemctl start nginx
      systemctl enable nginx
    SHELL
end
```

```
  config.vm.define "ubuntu-test-vm" do |ubuntu1|
   ubuntu1.vm.box = "ubuntu/jammy64"
   ubuntu1.vm.network "forwarded_port", guest: 80, host: 8083


 #config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"
 ubuntu1.vm.network "private_network", ip: "192.168.56.10", name: "VirtualBox Host-Only Ethernet Adapter"
 ubuntu1.vm.network "public_network", type: "dhcp", bridge: "Intel(R) Dual Band Wireless-AC 8265"
 ubuntu1.vm.hostname="devops.com"
 ubuntu1.vm.provider "virtualbox" do |vb|

   vb.memory = "1024"
   vb.cpus= "1"
   vb.name= "ubuntu vm 1"
  end

config.vm.provision "shell", inline: <<-SHELL
   apt-get update
    #apt-get install -y nginx
    #systemctl start nginx
    #systemctl enable nginx
   SHELL

   ubuntu1.ssh.insert_key = false
   ubuntu1.ssh.timeout = 300


end
```

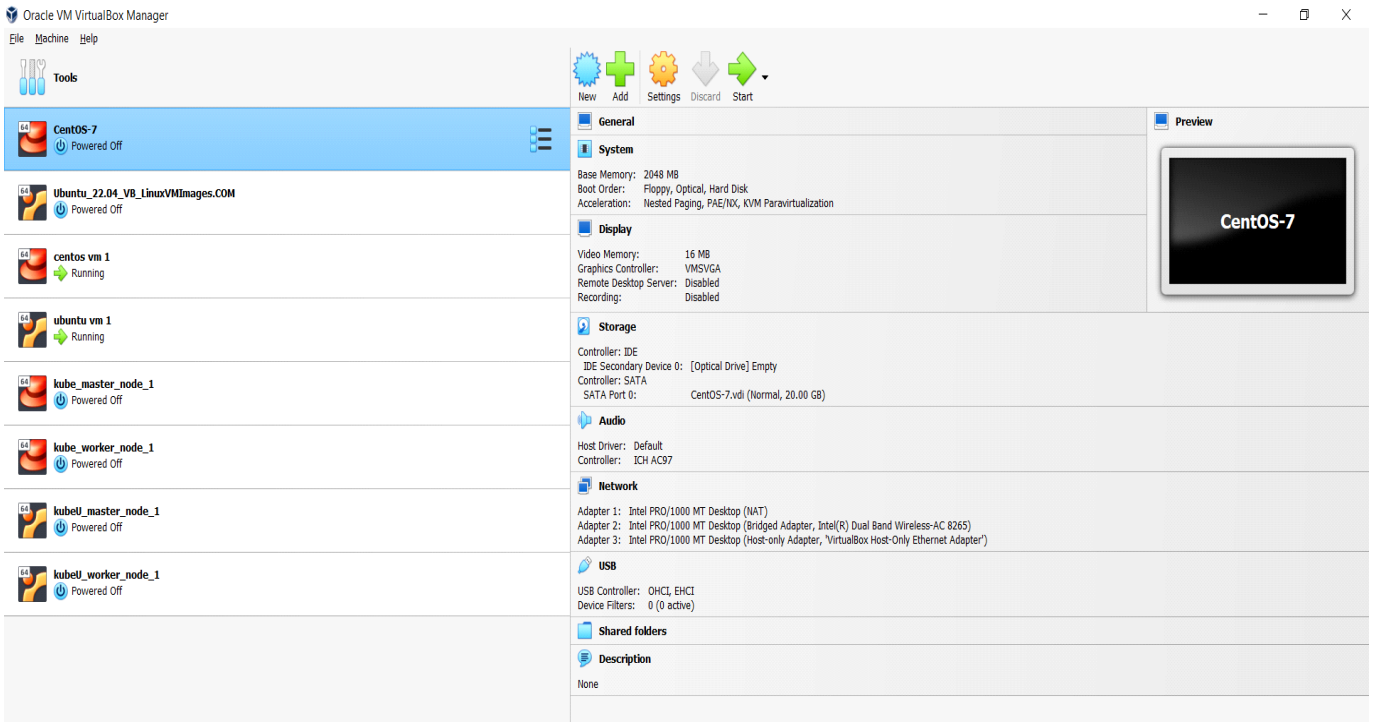
Start the Virtual Machines:

To start centos vm 1 VM:

Vagrant up < vm config name> Centos-test-vm (It will give new centos7 server in Virtual Machince)

To start ubuntu vm 1 VM server:

Vagrant up ubuntu-test-vm(It will give new ubuntu server in VM



Access the Virtual Machines

You can SSH into each virtual machine using the following commands in terminal:

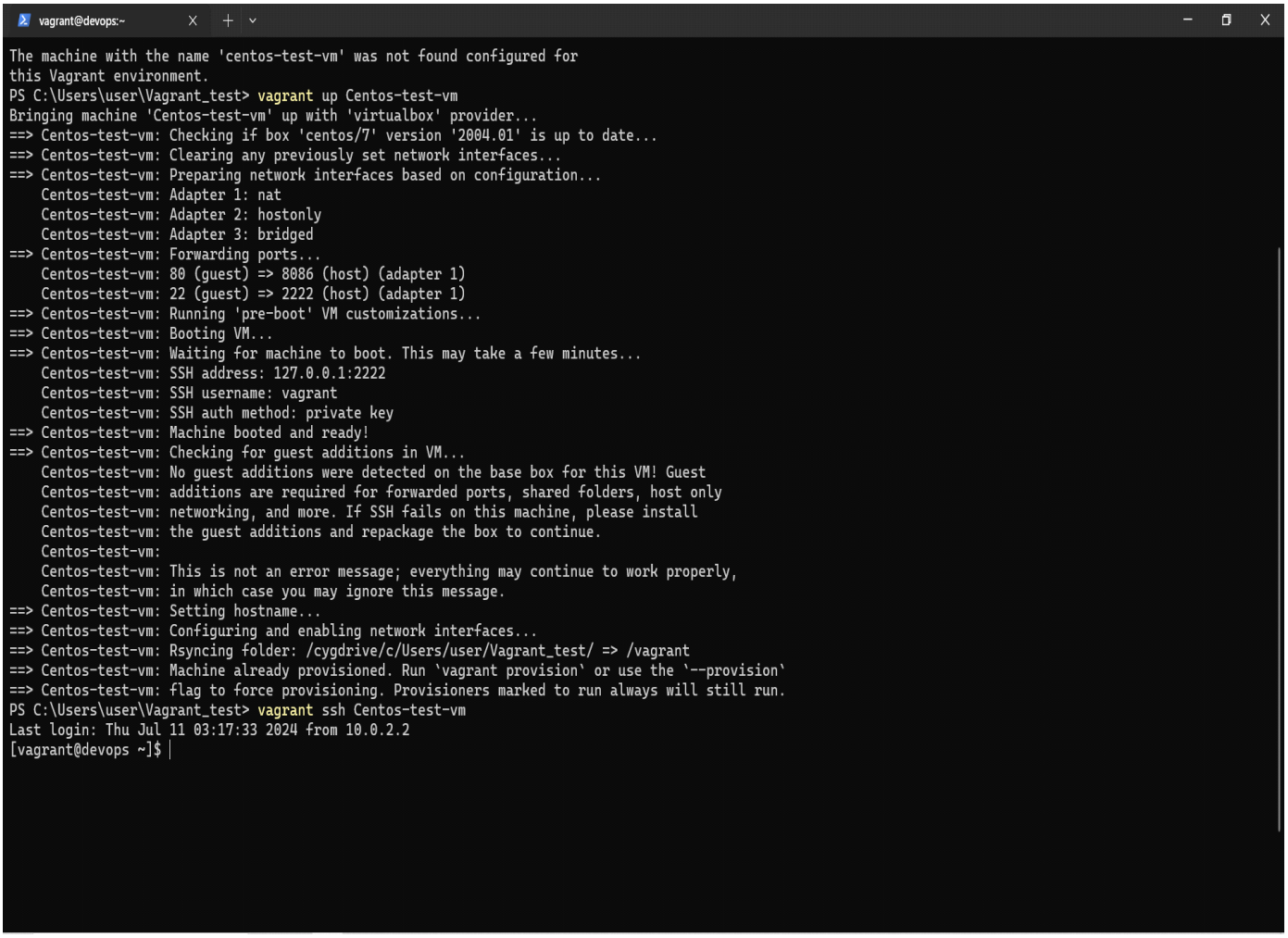
- To access the centos VM:

vagrant ssh centos-test-vm

- To access the ubuntu VM:

vagrant ssh ubuntu-test-vm

To destroy VMs: vagrant destroy –f
In case of provision: vagrant reload --provision


vagrant@devops:~
The machine with the name 'centos-test-vm' was not found configured for
this Vagrant environment.
PS C:\Users\user\Vagrant_test> vagrant up Centos-test-vm
Bringing machine 'Centos-test-vm' up with 'virtualbox' provider...
==> Centos-test-vm: Checking if box 'centos/7' version '2004.01' is up to date...
==> Centos-test-vm: Clearing any previously set network interfaces...
==> Centos-test-vm: Preparing network interfaces based on configuration...
Centos-test-vm: Adapter 1: nat
Centos-test-vm: Adapter 2: hostonly
Centos-test-vm: Adapter 3: bridged
==> Centos-test-vm: Forwarding ports...
Centos-test-vm: 80 (guest) => 8086 (host) (adapter 1)
Centos-test-vm: 22 (guest) => 2222 (host) (adapter 1)
==> Centos-test-vm: Running 'pre-boot' VM customizations...
==> Centos-test-vm: Booting VM...
==> Centos-test-vm: Waiting for machine to boot. This may take a few minutes...
Centos-test-vm: SSH address: 127.0.0.1:2222
Centos-test-vm: SSH username: vagrant
Centos-test-vm: SSH auth method: private key
==> Centos-test-vm: Machine booted and ready!
==> Centos-test-vm: Checking for guest additions in VM...
Centos-test-vm: No guest additions were detected on the base box for this VM! Guest
Centos-test-vm: additions are required for forwarded ports, shared folders, host only
Centos-test-vm: networking, and more. If SSH fails on this machine, please install
Centos-test-vm: the guest additions and repackage the box to continue.
Centos-test-vm:
Centos-test-vm: This is not an error message; everything may continue to work properly,
Centos-test-vm: in which case you may ignore this message.
==> Centos-test-vm: Setting hostname...
==> Centos-test-vm: Configuring and enabling network interfaces...
==> Centos-test-vm: Rsyncing folder: /cygdrive/c/Users/user/Vagrant_test/ => /vagrant
==> Centos-test-vm: Machine already provisioned. Run `vagrant provision` or use the `--provision`
==> Centos-test-vm: flag to force provisioning. Provisioners marked to run always will still run.
PS C:\Users\user\Vagrant_test> vagrant ssh Centos-test-vm
Last login: Thu Jul 11 03:17:33 2024 from 10.0.2.2
[vagrant@devops ~]$

3. CI Pipeline with Jenkins

Jenkins Setup:

- Create bridge network in docker
- Access Jenkins on a server with docker run command to execute docker command inside Jenkins node
- Configure Jenkins with necessary plugins: Git, Docker, Pipeline.

Command :

Docker run

Create docker file for Jenkins docker images

Docker login

Docker build –t <image name>

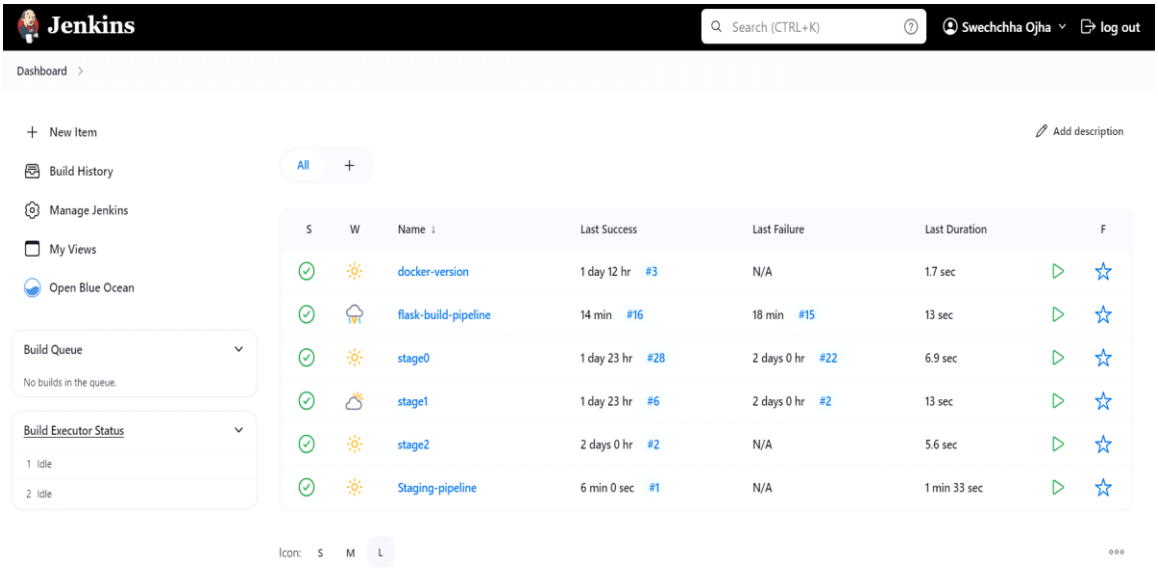
Docker exec –it <container name> bash(to get into Jenkins container)

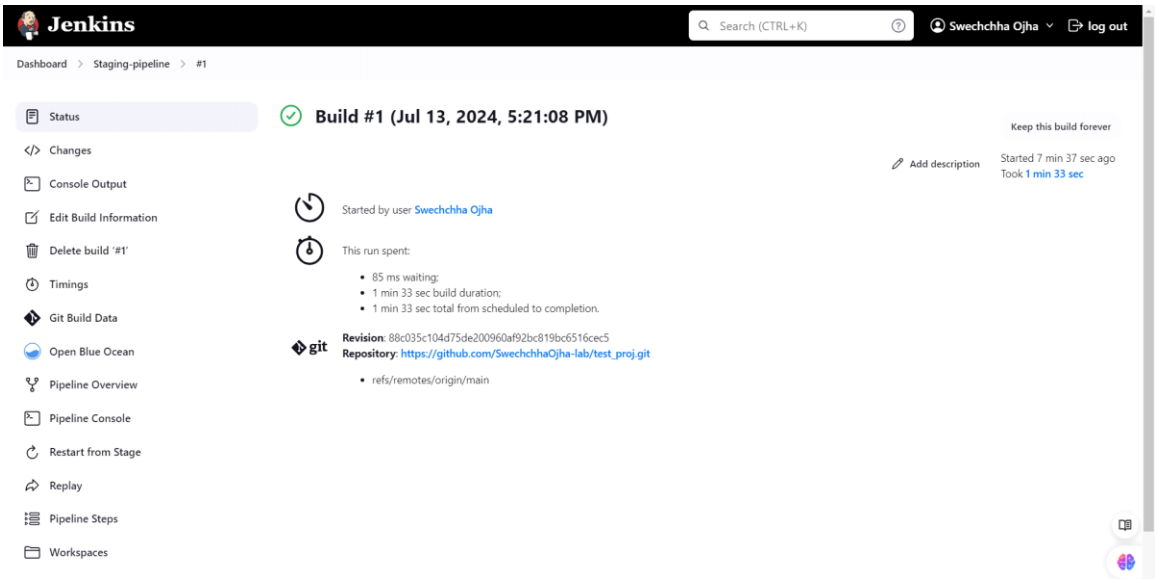
Docker restart <Jenkins id>

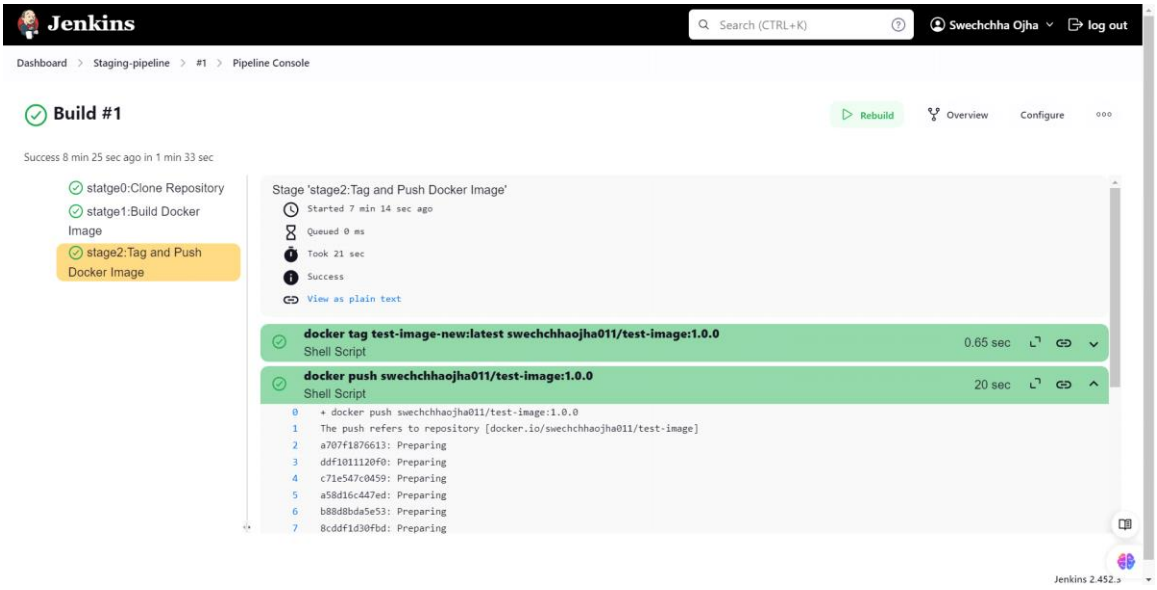
<ip address eth1>:8080(listen 8080 port)

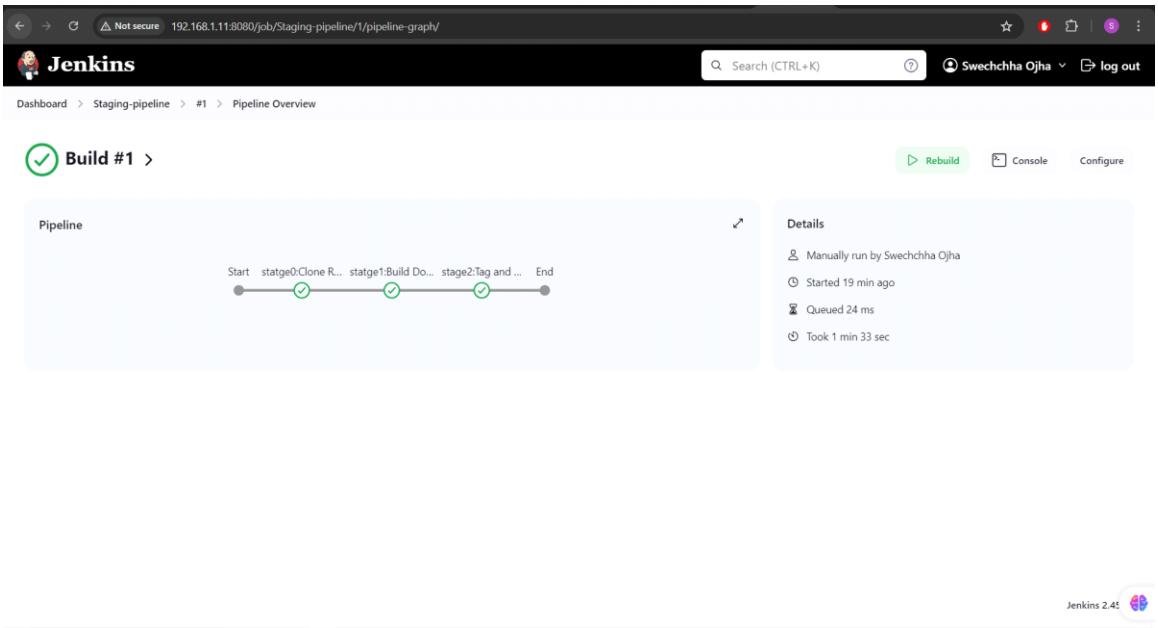
Pipeline Script:

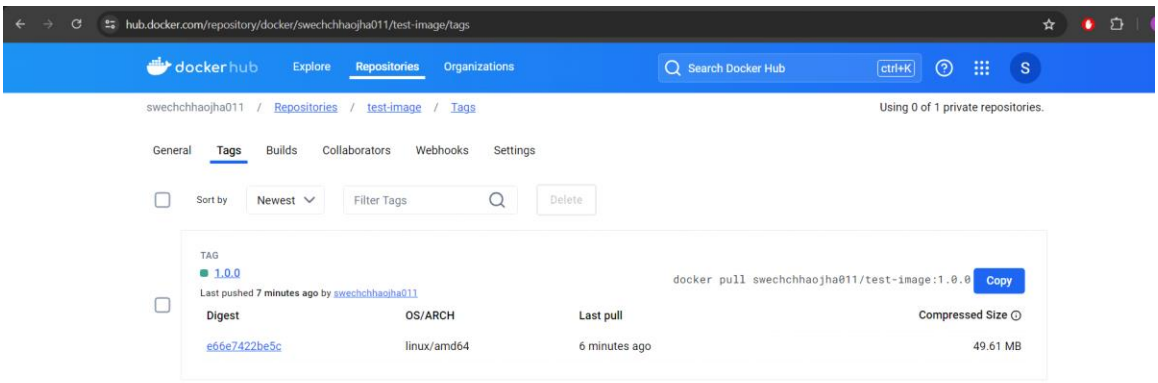
```
pipeline {
agent any

stages {
stage('statge0:Clone Repository') {
steps {
git branch: 'main', url: 'https://github.com/SwechchhaOjha-lab/test_proj.git'
}
}
stage('statge1:Build Docker Image') {
steps {
sh 'docker build -t test-image-new:latest .'
}
}
stage('stage2:Tag and Push Docker Image') {
steps {
sh 'docker tag test-image-new:latest swechchhaojha011/test-image:1.0.0'
sh 'docker push swechchhaojha011/test-image:1.0.0'
}
}
}
}
```


Jenkins
Search (CTRL+K)
Swechchha Ojha
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Open Blue Ocean
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
F
docker-version
1 day 12 hr #3
N/A
1.7 sec
flask-build-pipeline
14 min #16
18 min #15
13 sec
stage0
1 day 23 hr #28
2 days 0 hr #22
6.9 sec
stage1
1 day 23 hr #6
2 days 0 hr #2
13 sec
stage2
2 days 0 hr #2
N/A
5.6 sec
Staging-pipeline
6 min 0 sec #1
N/A
1 min 33 sec
Icon: S M L


Jenkins
Search (CTRL+K)
Swechchha Ojha
log out
Dashboard
Staging-pipeline
#1
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
Git Build Data
Open Blue Ocean
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Build #1 (Jul 13, 2024, 5:21:08 PM)
Keep this build forever
Add description
Started 7 min 37 sec ago
Took 1 min 33 sec
Started by user Swechchha Ojha
This run spent:
85 ms waiting;
1 min 33 sec build duration;
1 min 33 sec total from scheduled to completion.
Revision: 88c035c104d75de200960af92bc819bc6516cec5
Repository: https://github.com/SwechchhaOjha-lab/test_proj.git
refs/remotes/origin/main


Jenkins
Search (CTRL+K)
Swechchha Ojha
log out
Dashboard > Staging-pipeline > #1 > Pipeline Console
Build #1
Rebuild
Overview
Configure
Success 8 min 25 sec ago in 1 min 33 sec
statge0:Clone Repository
statge1:Build Docker Image
stage2:Tag and Push Docker Image
Stage 'stage2:Tag and Push Docker Image'
Started 7 min 14 sec ago
Queued 0 ms
Took 21 sec
Success
View as plain text
docker tag test-image-new:latest swechchhaojha011/test-image:1.0.0
Shell Script
0.65 sec
docker push swechchhaojha011/test-image:1.0.0
Shell Script
20 sec
+ docker push swechchhaojha011/test-image:1.0.0
The push refers to repository [docker.io/swechchhaojha011/test-image]
a707f1876613: Preparing
ddf1011120f0: Preparing
c71e547c0459: Preparing
a58d16c447ed: Preparing
b88d8bda5e53: Preparing
8cddf1d30fbd: Preparing


Not secure
192.168.1.11:8080/job/Staging-pipeline/1/pipeline-graph/
Jenkins
Search (CTRL+K)
Swechchha Ojha
log out
Dashboard > Staging-pipeline > #1 > Pipeline Overview
Build #1
Rebuild
Console
Configure
Pipeline
Start
statge0:Clone R...
statge1:Build Do...
stage2:Tag and ...
End
Details
Manually run by Swechchha Ojha
Started 19 min ago
Queued 24 ms
Took 1 min 33 sec


hub.docker.com/repository/docker/swechchhaojha011/test-image/tags
docker hub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
swechchhaojha011 / Repositories / test-image / Tags
Using 0 of 1 private repositories.
General
Tags
Builds
Collaborators
Webhooks
Settings
Sort by
Newest
Filter Tags
Delete
TAG
1.0.0
Last pushed 7 minutes ago by swechchhaojha011
docker pull swechchhaojha011/test-image:1.0.0
Copy
Digest
OS/ARCH
Last pull
Compressed Size
e66e7422be5c
linux/amd64
6 minutes ago
49.61 MB

4. Prometheus and Grafana Monitoring Setup

Prometheus is used for monitoring and alerting, while Grafana is used for visualization. The Node Exporter is installed on each server to expose hardware and OS metrics.

Architecture:

Prometheus Server

- **Function:** Collects and stores metrics from various endpoints.
- **Configuration File:** /etc/prometheus/prometheus.yml
- **Port:** 9090

Grafana Server

- **Function:** Visualizes metrics collected by Prometheus.
- **Configuration:** Add Prometheus as a data source.
- **Port:** 3000

Node Exporter

- **Function:** Exposes hardware and OS metrics from servers.
- **Installation Path:** /usr/local/bin/node_exporter
- **Port:** 9100

Installation Steps

Prometheus Installation

wget https://github.com/prometheus/prometheus/releases/download/v2.32.1/prometheus-2.32.1.linux-amd64.tar.gz
tar xvf prometheus-2.32.1.linux-amd64.tar.gz

Grafana Installation

wget https://dl.grafana.com/oss/release/grafana-8.3.3.linux-amd64.tar.gz
tar -zxvf grafana-8.3.3.linux-amd64.tar.gz

Node Exporter Installation

wget https://github.com/prometheus/node_exporter/releases/download/v1.3.1/node_exporter-1.3.1.linux-amd64.tar.gz
tar xvf node_exporter-1.3.1.linux-amd64.tar.gz

Configure Prometheus to Scrape Node Exporter Metrics

Yaml

```
global:
scrape_interval: 10s

scrape_configs:
- job_name: 'prometheus_master'
scrape_interval: 5s
static_configs:
- targets: ['127.0.0.1:9090']

- job_name: 'node_exporter_centos'
scrape_interval: 5s
static_configs:
- targets: ['192.168.56.101:9100']
```

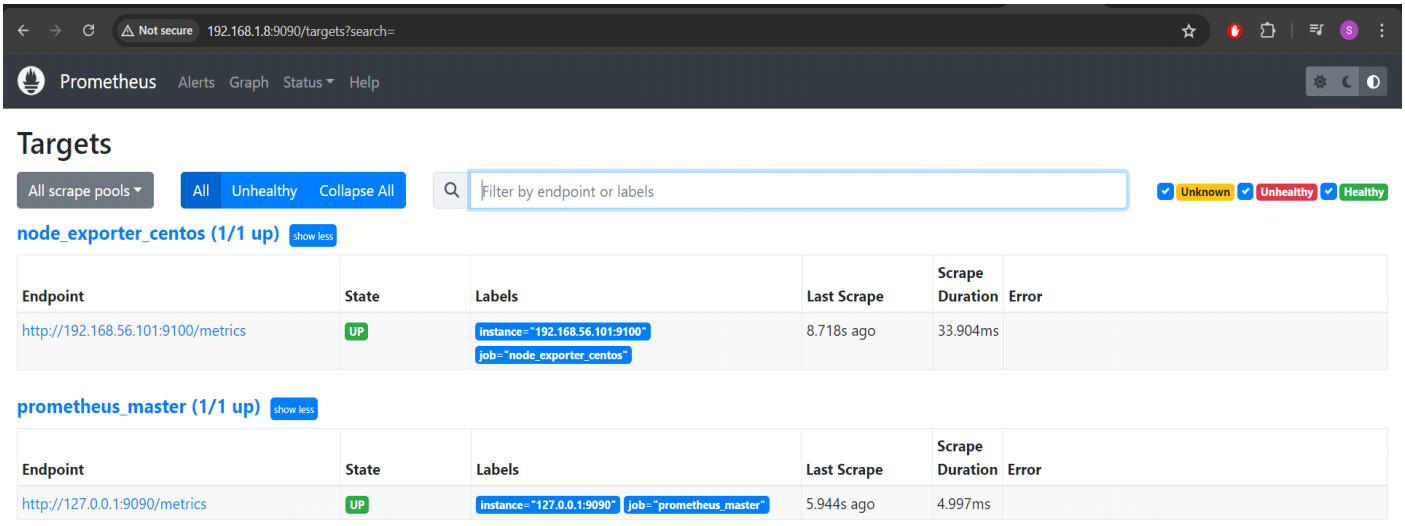
Security Measures:

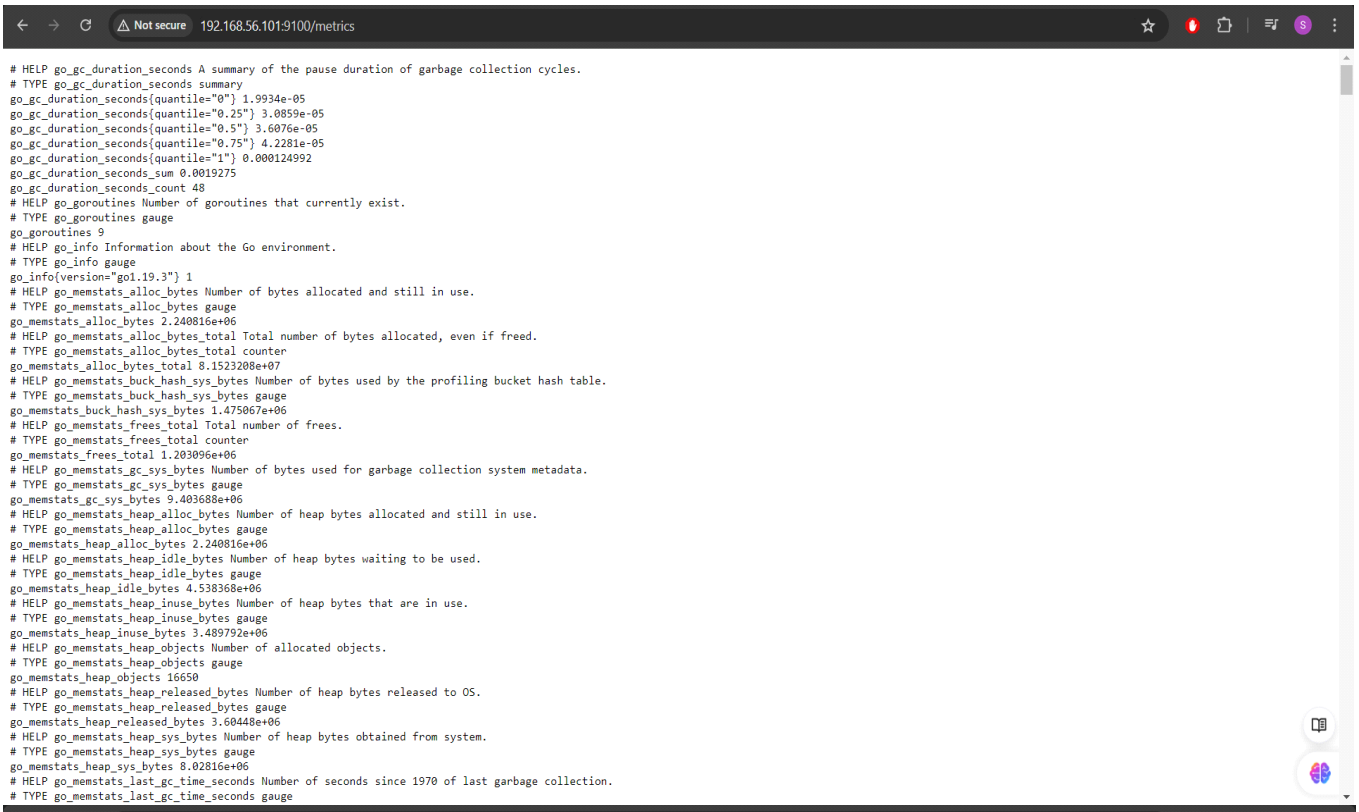
RBAC in Grafana

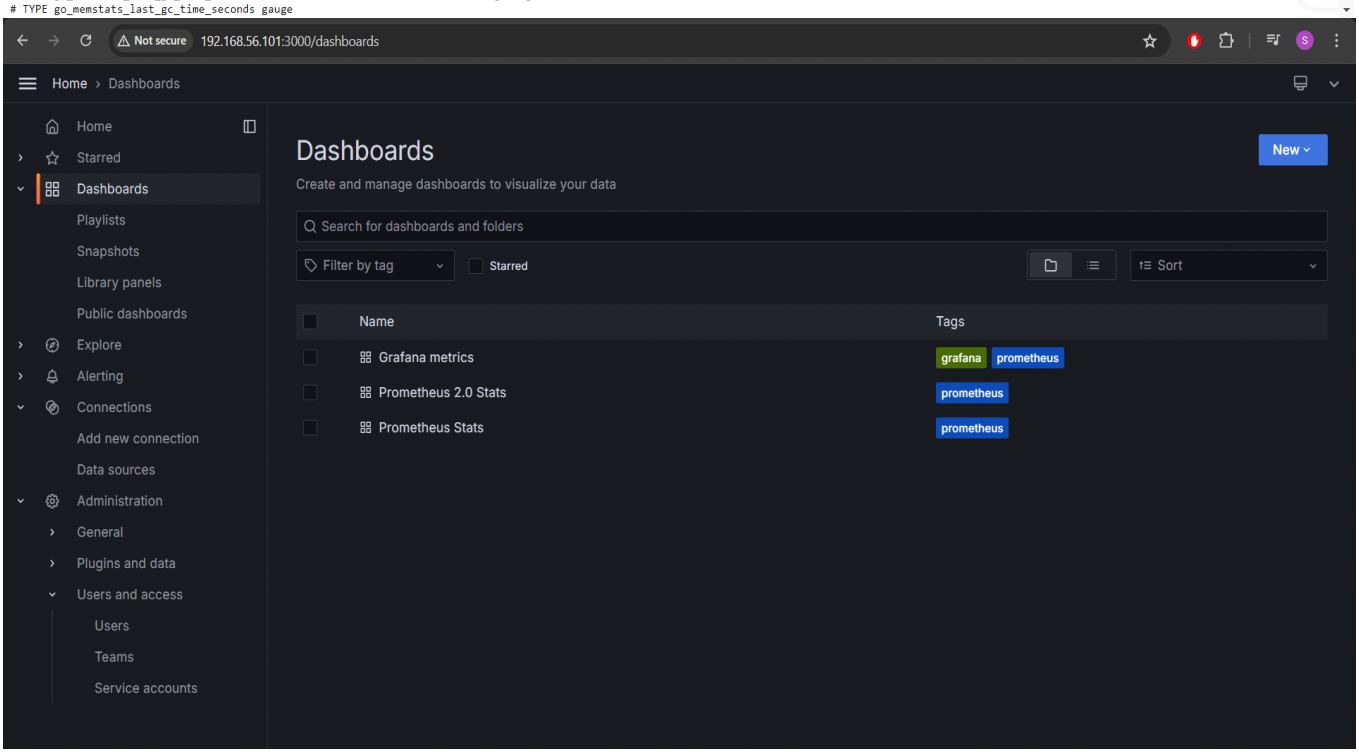
Go to Configuration -> Users to create roles and users.

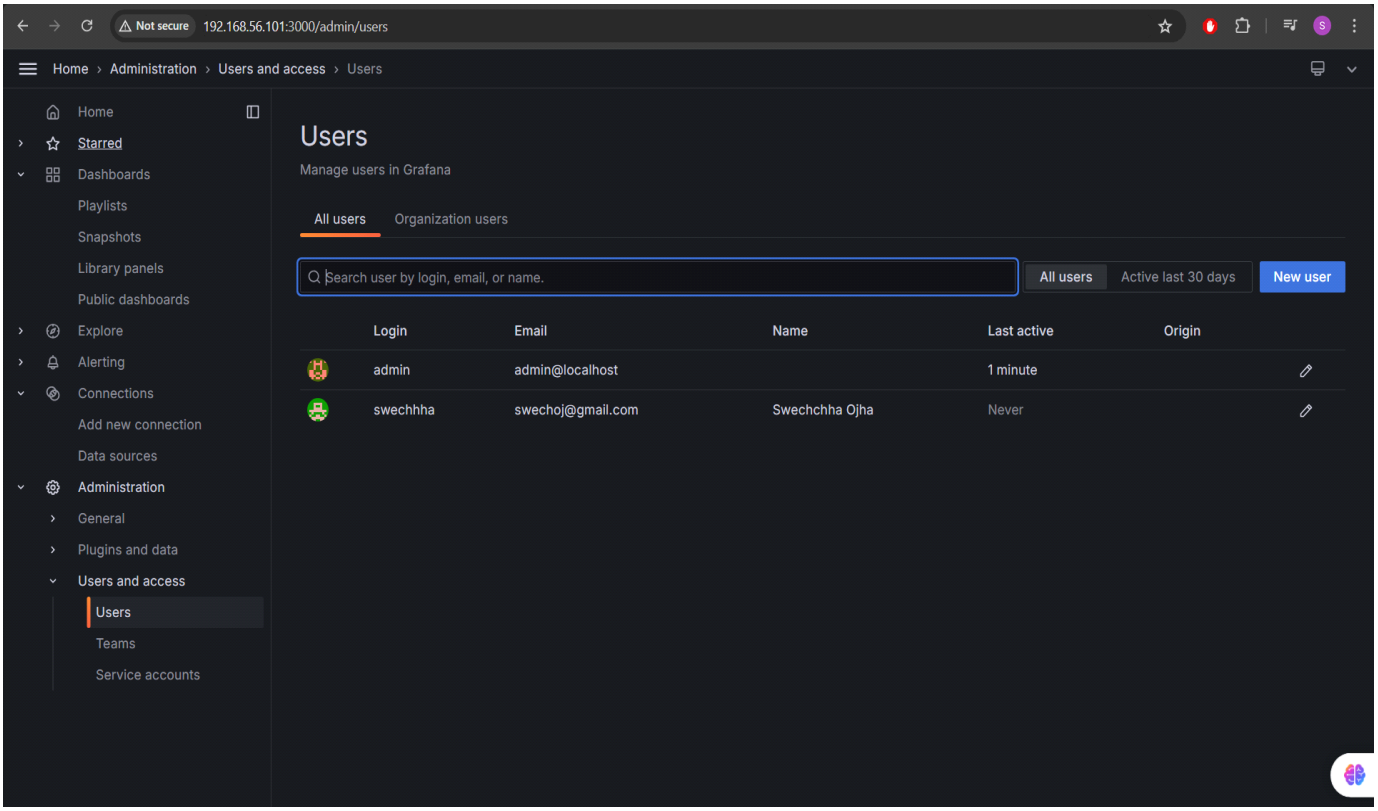
Encryption

Enable HTTPS for Prometheus and Grafana, use secure credentials.




Not secure
192.168.56.101:9100/metrics
HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 1.9934e-05
go_gc_duration_seconds{quantile="0.25"} 3.0859e-05
go_gc_duration_seconds{quantile="0.5"} 3.6076e-05
go_gc_duration_seconds{quantile="0.75"} 4.2281e-05
go_gc_duration_seconds{quantile="1"} 0.000124992
go_gc_duration_seconds_sum 0.0019275
go_gc_duration_seconds_count 48
HELP go_goroutines Number of goroutines that currently exist.
TYPE go_goroutines gauge
go_goroutines 9
HELP go_info Information about the Go environment.
TYPE go_info gauge
go_info{version="go1.19.3"} 1
HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.240816e+06
HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 8.1523208e+07
HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.475067e+06
HELP go_memstats_frees_total Total number of frees.
TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.203096e+06
HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 9.403688e+06
HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.240816e+06
HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 4.538368e+06
HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 3.489792e+06
HELP go_memstats_heap_objects Number of allocated objects.
TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 16650
HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 3.60448e+06
HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 8.02816e+06
HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
TYPE go_memstats_last_gc_time_seconds gauge


Not secure
192.168.56.101:3000/dashboards
Home › Dashboards
Home
Starred
Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
General
Plugins and data
Users and access
Users
Teams
Service accounts
Dashboards
Create and manage dashboards to visualize your data
New
Search for dashboards and folders
Filter by tag
Starred
Sort
Name
Tags
Grafana metrics
grafana
prometheus
Prometheus 2.0 Stats
prometheus
Prometheus Stats
prometheus


Not secure
192.168.56.101:3000/admin/users
Home › Administration › Users and access › Users
Home
Starred
Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
General
Plugins and data
Users and access
Users
Teams
Service accounts
Users
Manage users in Grafana
All users
Organization users
Search user by login, email, or name.
All users
Active last 30 days
New user
Login
Email
Name
Last active
Origin
admin
admin@localhost
1 minute
swechhha
swechoj@gmail.com
Swechchha Ojha
Never

5. MySQL Backups and Migration

Backup Strategy:

- Script Automatic daily backupsin mysql with corntab
- Store backups securely on a remote server.

Backup Command:

mysql –u root –p (check database to backup)

cd /etc

mkdir backup_mysql

cd backup_mysql

vi backup_mysql.sh (create new sricpt name backup_mysql.sh)

ls –l

gunzip employees_db-20240711_100818.sql.gz

chmod +x backup_mysql.sh (grant execute permission to the script)

Sudo ./backup_mysql.sh (testing)

Corntab –e (create a schedule for automatic backup execution)

Systemctl restart crond

head employees_db-20240711_100818.sql

tail employees_db-20240711_100818.sql

Migration Steps:

- Perform a full backup on Server A.
- Transfer backup file to Server B.
- Restore the database on Server B.

Restore Command:

telnet <ip address> 3306

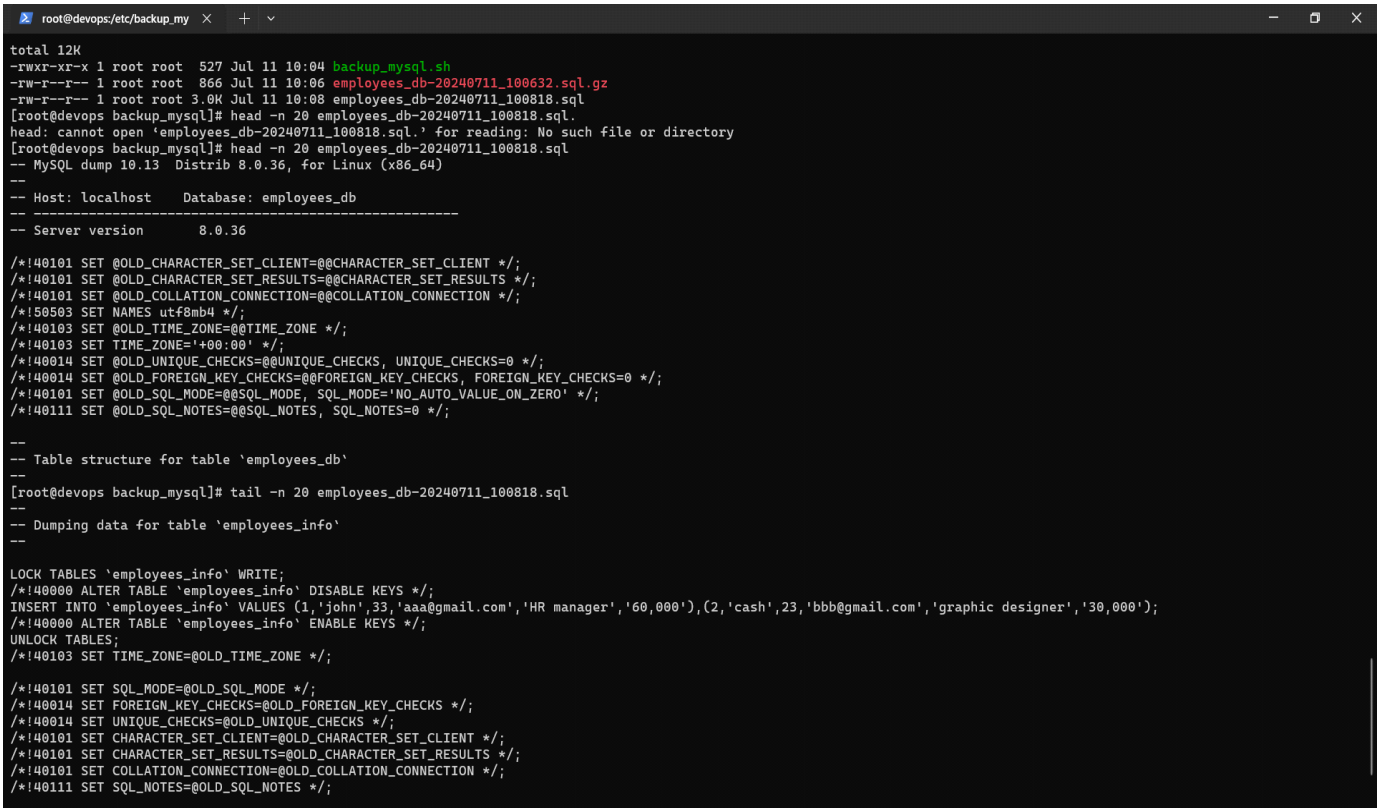
Mysql –u root –p -h<ip address>

```
root@devops:~  ubuntu@ubuntu-2204: ~
    ->
    ->
    ->
    -> ^C
mysql> exit
Bye
[root@devops ~]# mysql -u newuser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| employees_db       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

mysql> exit
Bye
[root@devops ~]# netstat -utlpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      1103/java
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      1120/nginx: master
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1045/sshd
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      1120/nginx: master
tcp        0      0 127.0.0.1:8005          0.0.0.0:*               LISTEN      1103/java
tcp6       0      0 :::3306                 :::*                    LISTEN      1550/mysqld
tcp6       0      0 :::9100                 :::*                    LISTEN      1046/node_exporter
tcp6       0      0 :::80                   :::*                    LISTEN      1120/nginx: master
tcp6       0      0 :::22                   :::*                    LISTEN      1045/sshd
```

```
root@devops:/etc/backup_my  ubuntu@ubuntu-2204: ~
#!/bin/bash

# MySQL database credentials
DB_USER="newuser"
DB_PASS="Centos@123"
DB_NAME="employees_db"

# Backup directory
BACKUP_DIR="/etc/backup_mysql"

# Date format for backup file
DATE=$(date +"%Y%m%d_%H%M%S")

# Backup filename
BACKUP_FILE="$BACKUP_DIR/$DB_NAME-$DATE.sql.gz"

# Dump the MySQL database
mysqldump -u $DB_USER -p$DB_PASS $DB_NAME | gzip > $BACKUP_FILE

# Check if backup was successful
if [ $? -eq 0 ]; then
    echo "Backup completed successfully: $BACKUP_FILE"
else
    echo "Error: Backup failed!"
fi

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"backup_mysql.sh" 26L, 527C
```

```
root@devops:~ | ubuntu@ubuntu-2204: ~

ubuntu@ubuntu-2204:~$ telnet 192.168.1.7 3306
Trying 192.168.1.7...
Connected to 192.168.1.7.
Escape character is '^]'.
J
n] ]f^u<h#mysql_native_passwordubuntu
!#08S01Got packets out of orderConnection closed by foreign host.
ubuntu@ubuntu-2204:~$ mysql -u newuser -p -h 192.168.1.7
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.36 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| employees_db       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0.02 sec)

mysql> show tables;
ERROR 1046 (3D000): No database selected
mysql> use employees_db
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> shoe tables;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 's
hoe tables' at line 1
mysql> show tables;
+------------------------+
| Tables_in_employees_db |
```

```
root@devops:/etc/backup_my | ubuntu@ubuntu-2204: ~

[root@devops /]# cd etc/
[root@devops etc]# mkdir backup_mysql
[root@devops etc]# cd backup_mysql
[root@devops backup_mysql]# sudo vi backup_mysql.sh
[root@devops backup_mysql]# sudo vi backup_mysql.sh
[root@devops backup_mysql]# sudo vi backup_mysql.sh
[root@devops backup_mysql]# ls -l
total 4
-rw-r--r-- 1 root root 527 Jul 11 10:04 backup_mysql.sh
[root@devops backup_mysql]# chmod +x backup_mysql.sh
[root@devops backup_mysql]#
[root@devops backup_mysql]# ls -l
total 4
-rwxr-xr-x 1 root root 527 Jul 11 10:04 backup_mysql.sh
[root@devops backup_mysql]# ./backup_mysql.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this operation' when trying to dump tablespaces
Backup completed successfully: /etc/backup_mysql/employees_db-20240711_100632.sql.gz
[root@devops backup_mysql]# sudo ./backup_mysql.sh
mysqldump: [Warning] Using a password on the command line interface can be insecure.
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this operation' when trying to dump tablespaces
Backup completed successfully: /etc/backup_mysql/employees_db-20240711_100818.sql.gz
[root@devops backup_mysql]# ls -l
total 12
-rwxr-xr-x 1 root root 527 Jul 11 10:04 backup_mysql.sh
-rw-r--r-- 1 root root 866 Jul 11 10:06 employees_db-20240711_100632.sql.gz
-rw-r--r-- 1 root root 864 Jul 11 10:08 employees_db-20240711_100818.sql.gz
[root@devops backup_mysql]#
[root@devops backup_mysql]#
[root@devops backup_mysql]# crontab -e
no crontab for root - using an empty one
crontab: no changes made to crontab
crontab: no changes made to crontab
[root@devops backup_mysql]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@devops backup_mysql]#  1L, 45C written
crontab: installing new crontab
[root@devops backup_mysql]# systemctl restart cron
Failed to restart cron.service: Unit not found.
[root@devops backup_mysql]# systemctl restart crond
[root@devops backup_mysql]# crontab -l
0 23 * * * /etc/backup_mysql/backup_mysql.sh
[root@devops backup_mysql]# crontab -e
```


root@devops:/etc/backup_my
total 12K
-rwxr-xr-x 1 root root 527 Jul 11 10:04 backup_mysql.sh
-rw-r--r-- 1 root root 866 Jul 11 10:06 employees_db-20240711_100632.sql.gz
-rw-r--r-- 1 root root 3.0K Jul 11 10:08 employees_db-20240711_100818.sql
[root@devops backup_mysql]# head -n 20 employees_db-20240711_100818.sql.
head: cannot open ‘employees_db-20240711_100818.sql.’ for reading: No such file or directory
[root@devops backup_mysql]# head -n 20 employees_db-20240711_100818.sql
-- MySQL dump 10.13 Distrib 8.0.36, for Linux (x86_64)
--
-- Host: localhost   Database: employees_db
-- --
-- Server version   8.0.36
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
--
-- Table structure for table `employees_db`
--
[root@devops backup_mysql]# tail -n 20 employees_db-20240711_100818.sql
--
-- Dumping data for table `employees_info`
--
LOCK TABLES `employees_info` WRITE;
/*!40000 ALTER TABLE `employees_info` DISABLE KEYS */;
INSERT INTO `employees_info` VALUES (1,'john',33,'aaa@gmail.com','HR manager','60,000'),(2,'cash',23,'bbb@gmail.com','graphic designer','30,000');
/*!40000 ALTER TABLE `employees_info` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

Ubuntu installation

- **Downloaded the image** :

  By visiting linuxvmimages website and downloaded the Ubuntu 22.04(jammy jellyfish) virtual machine image .

- **Imported the Image**:

  Opening virtualization software and import the downloaded Ubuntu image. This process typically involves selecting "Import Appliance" or "Import VM" from the software's menu and choosing the downloaded file.

- **Configuring Virtual Machine Settings**:

  Before starting the virtual machine, adjusting settings like allocated RAM, number of CPU cores, and virtual disk size based on system's capabilities and needs.

- **Starting the Virtual Machine**:

  Once configured, starting the virtual machine. The Ubuntu operating system will boot up within the virtual environment.

- **Loging Into Ubuntu server**:

  Using the default credentials provided by linuxvmimages(often username: ubuntu, password: ubuntu) to log into the Ubuntu desktop environment.

- ssh ubuntu@<ip address> (inorder to ssh in windows terminal)