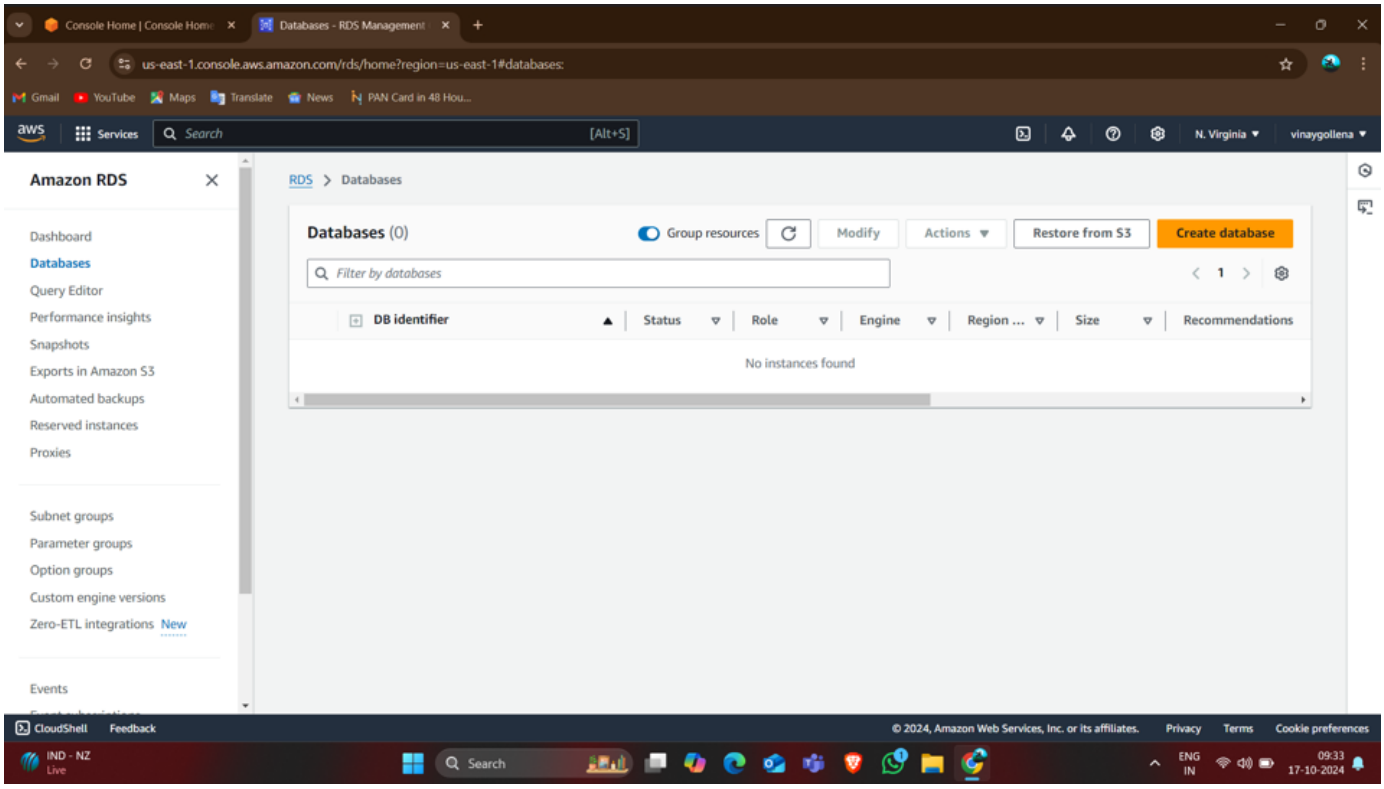
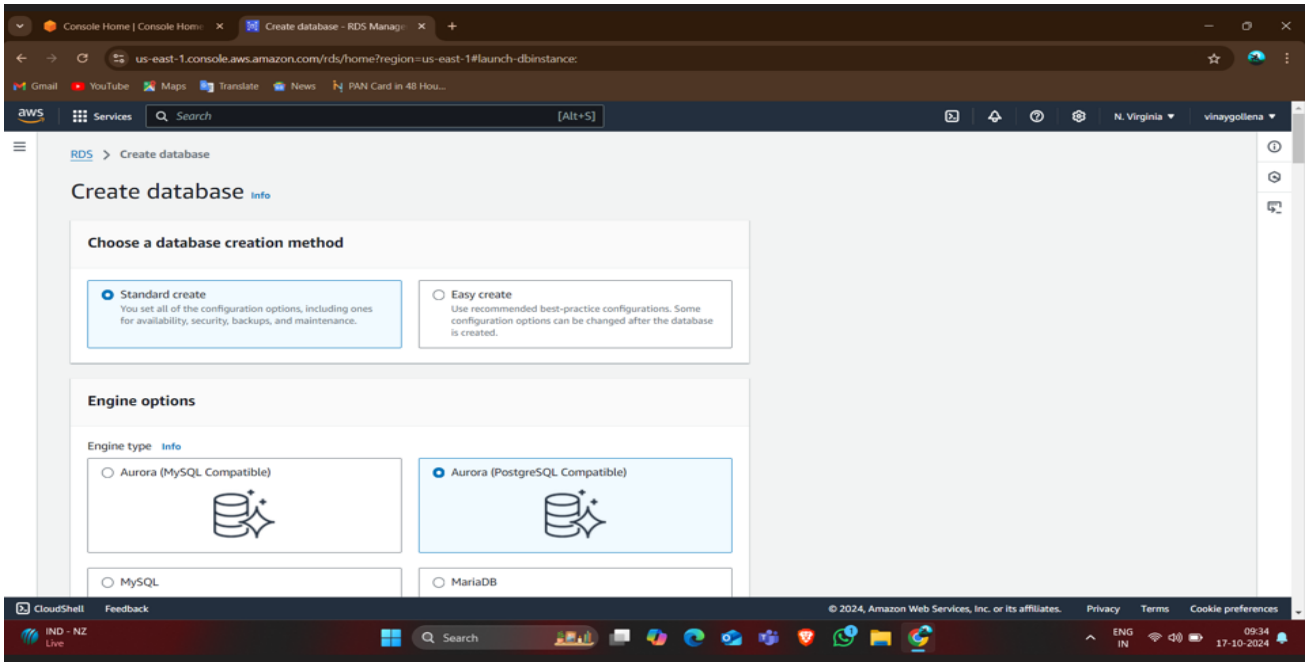
WordPress Working Methods

Deploy WordPress web application by using AWS RDS(MYSQL) service (manually).

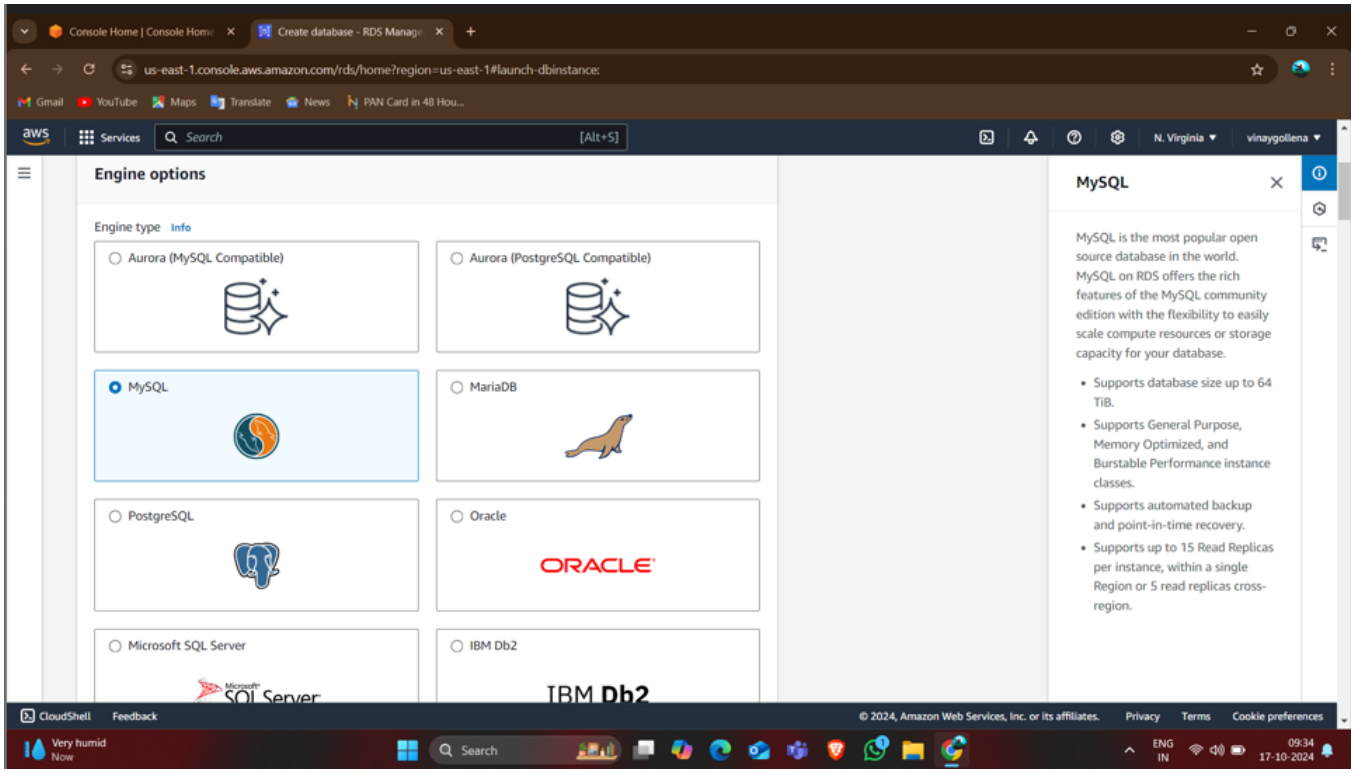
- Now first go to aws account and log into aws and search RDS and click on create database.



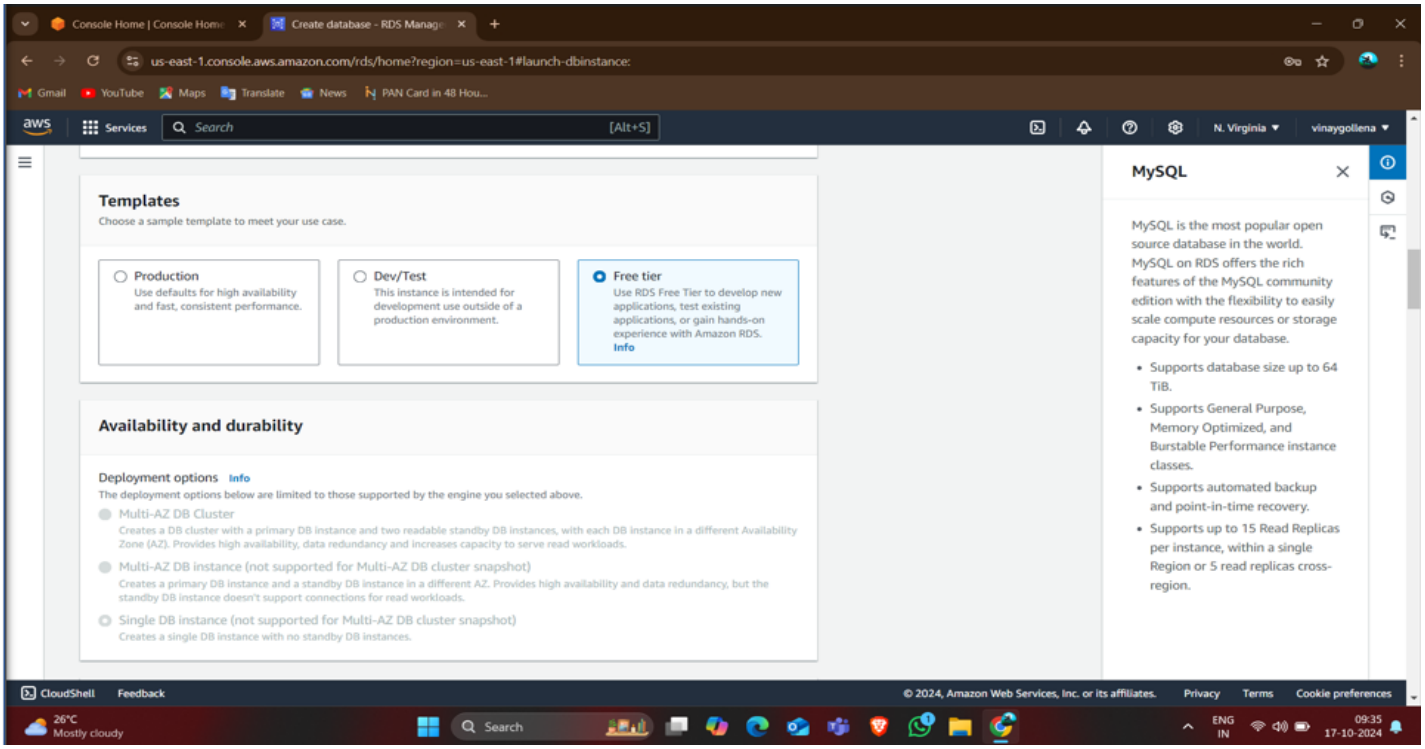
- Now select the database creation method (select standard method) because by selecting easy creation methods then its disables free tier template.



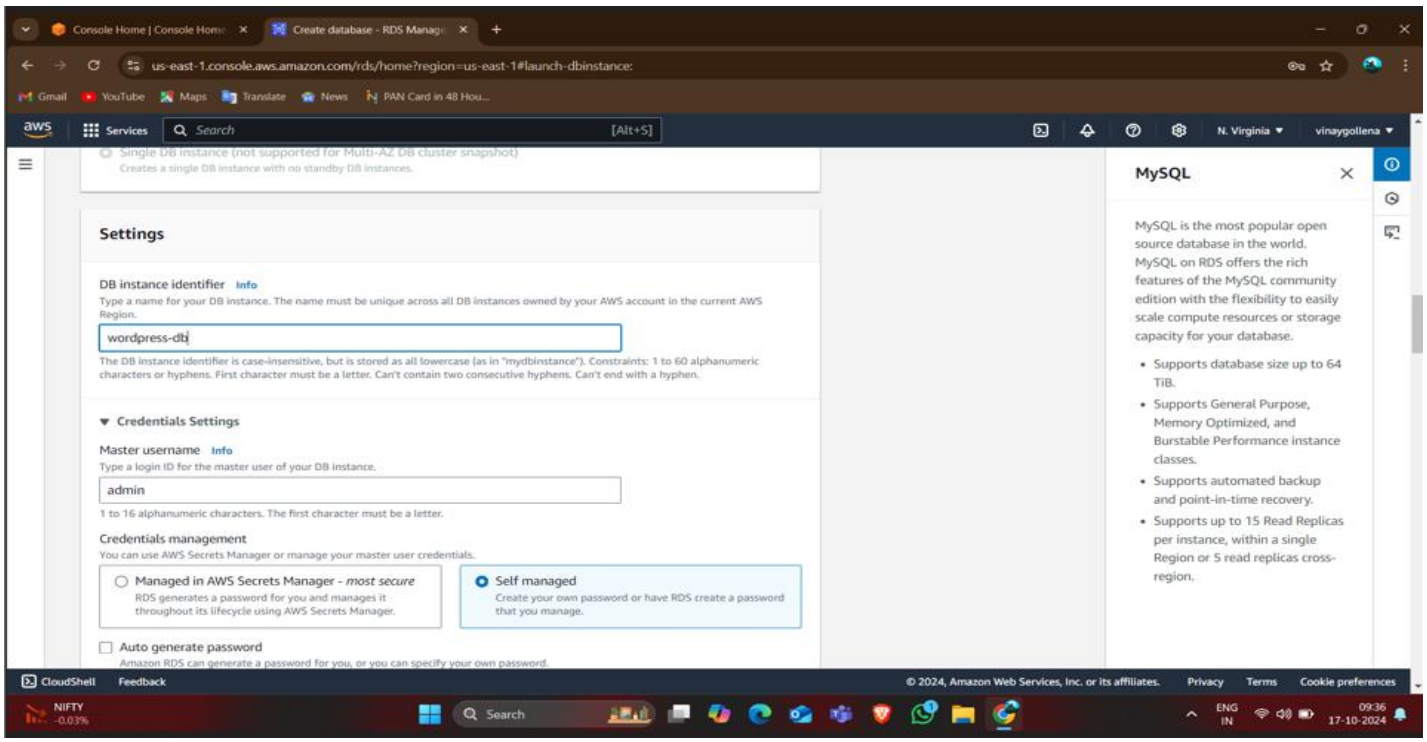
- Now select the database engine with version but I select the MySQL database engine because it’s a huge usage in real time organizations.



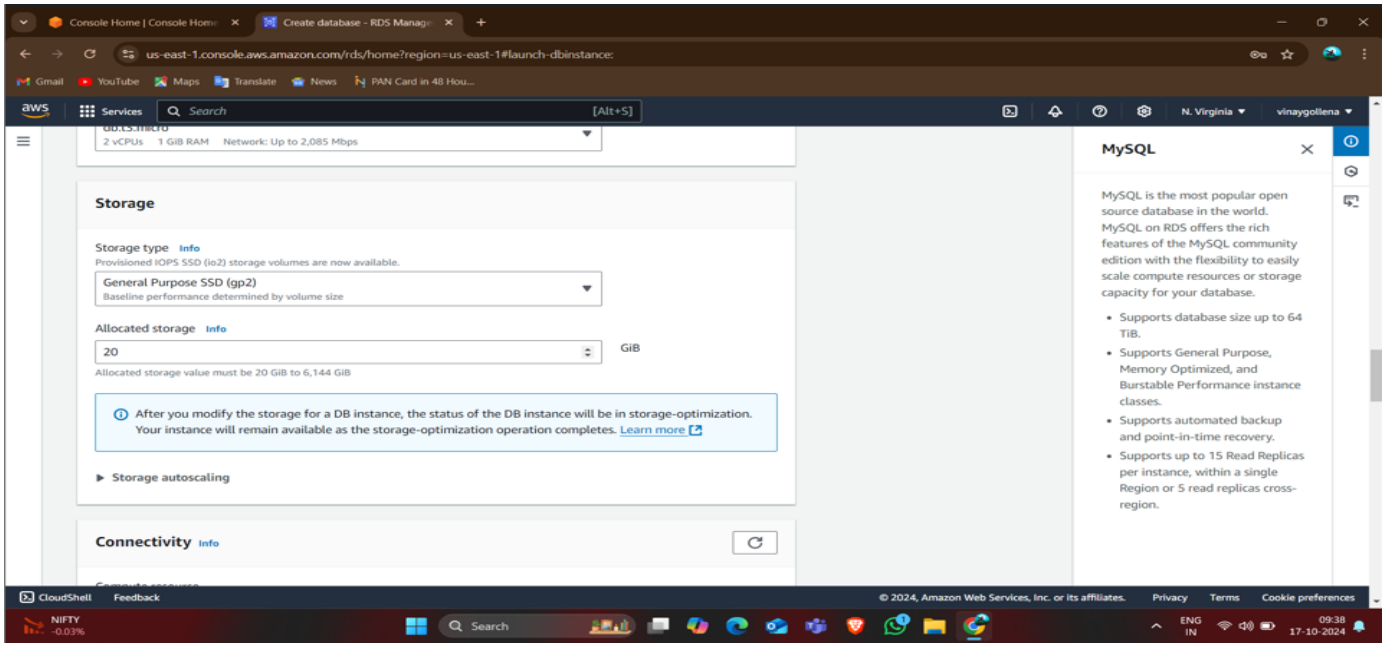
- Now select the template as free tier and by selecting the free tier it disables the availability zone section options.



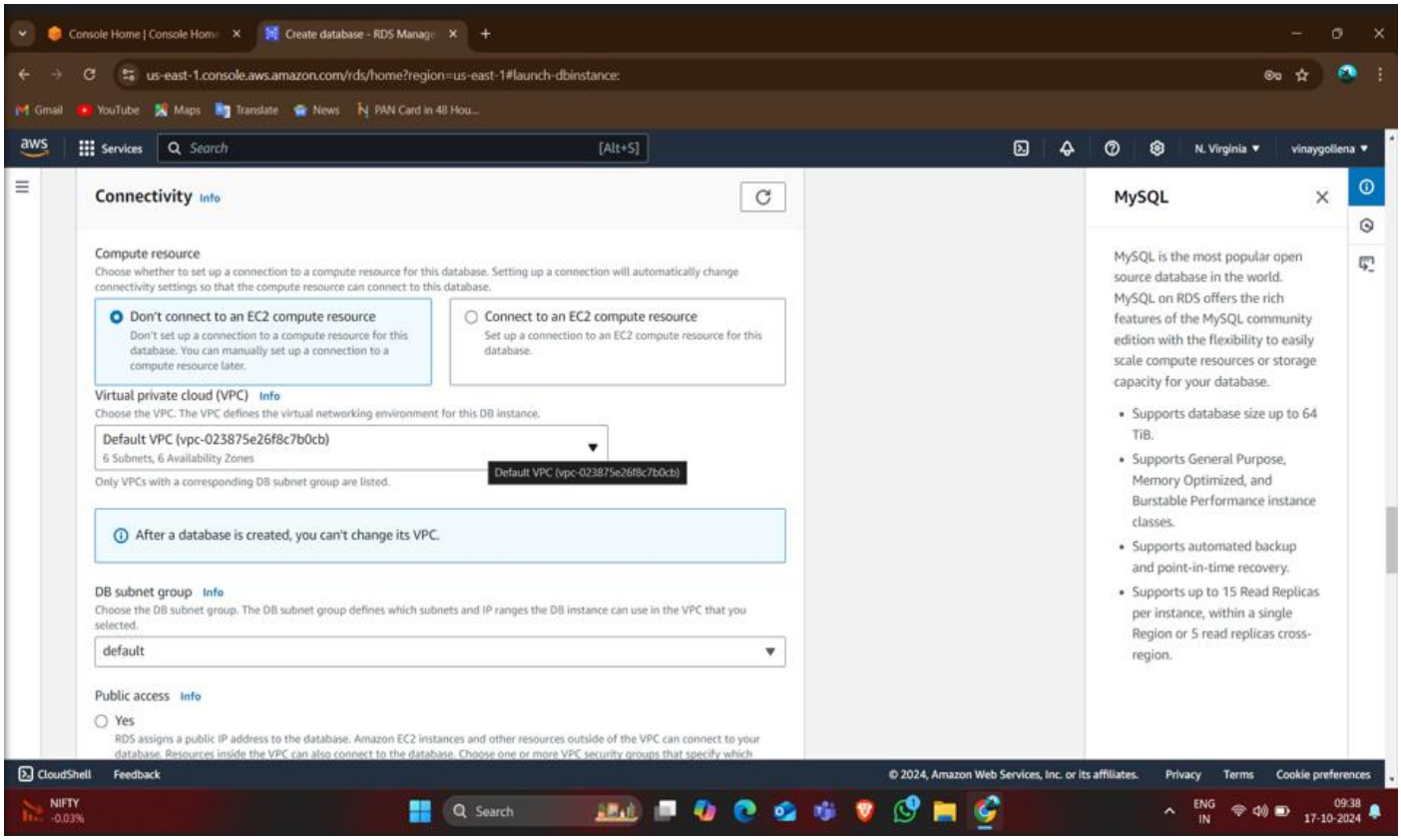
- Now give some name to your database and give username and password as credentials for your databases.



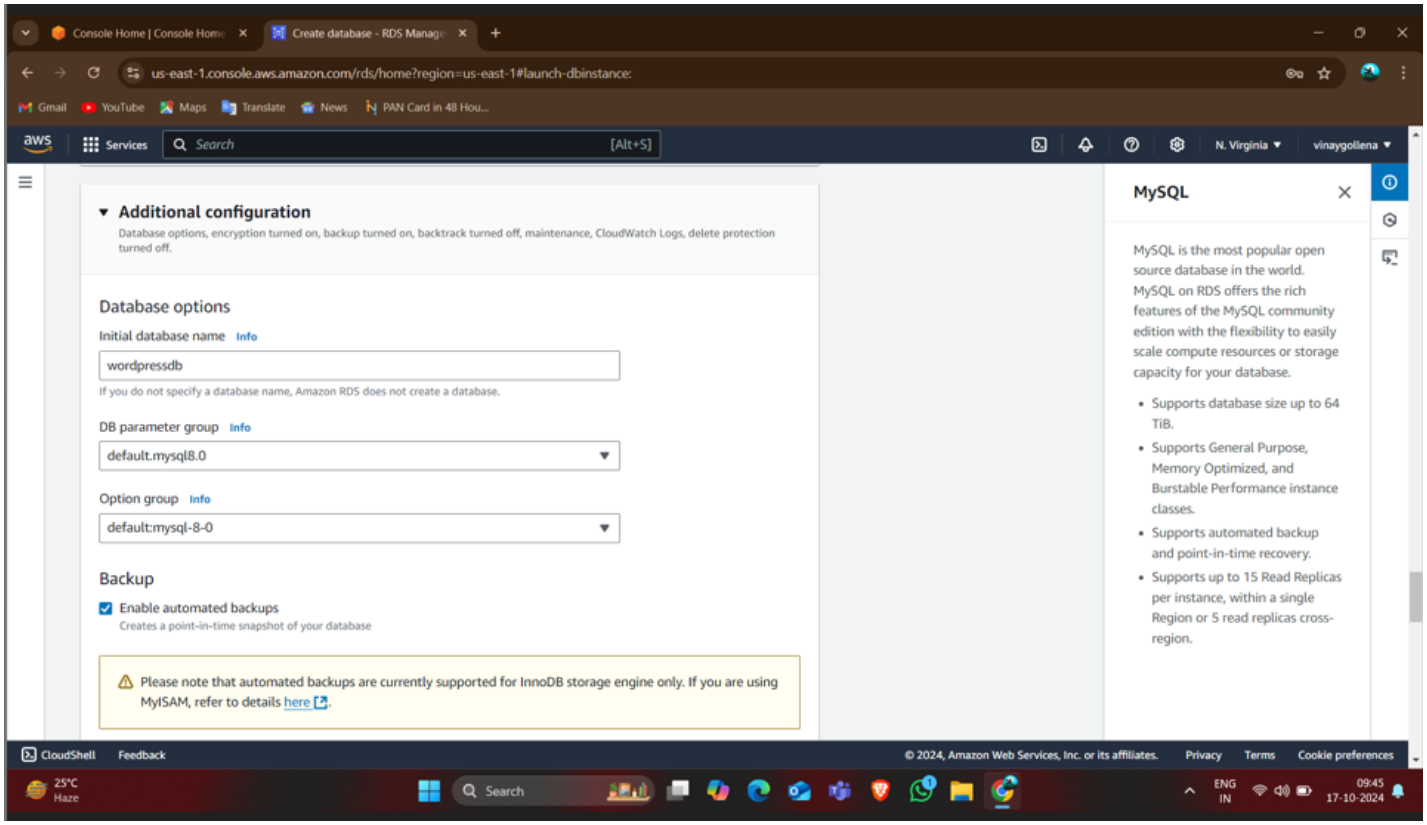
- Now select the storage type as general purpose (SSD (gp2)) and enter the storage values as maximum (1000gb) and minimum (20gb).



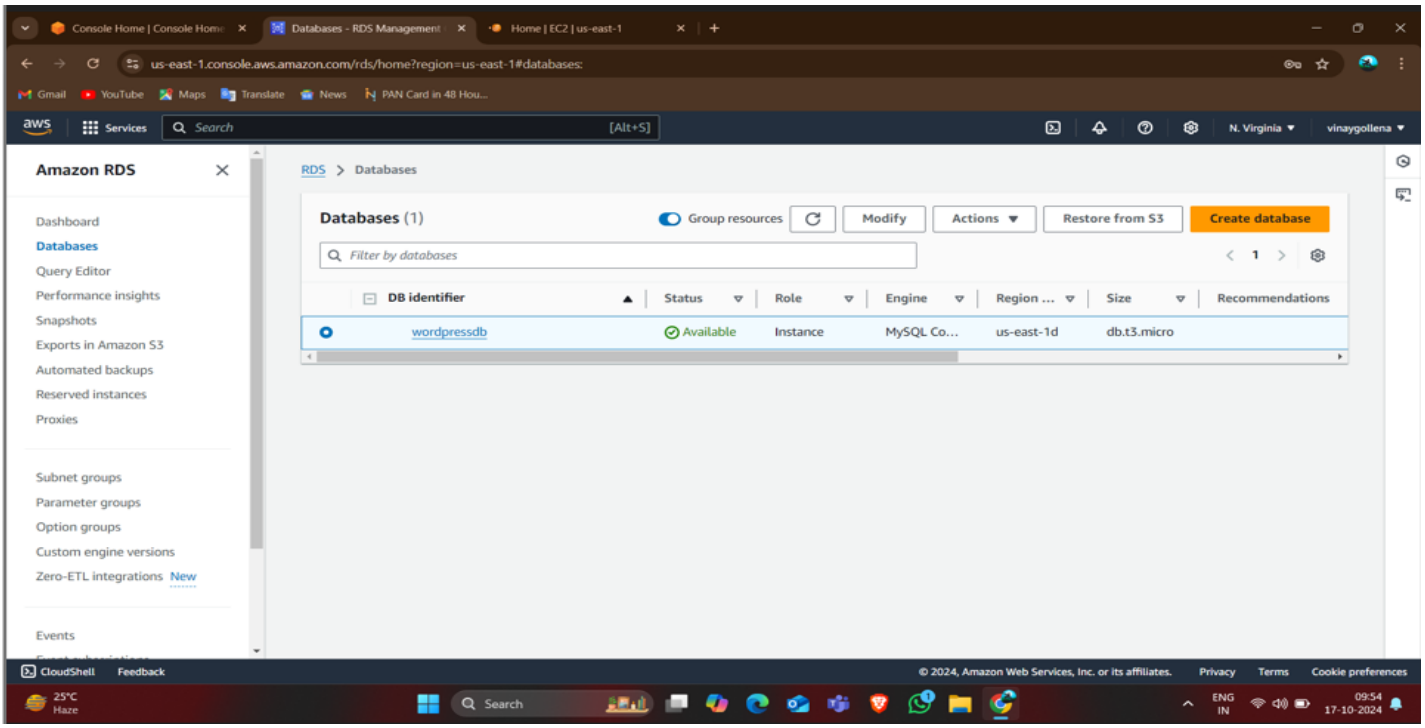
- Now select your created vpc or select default vpc and it automatically selects the database subnet group.



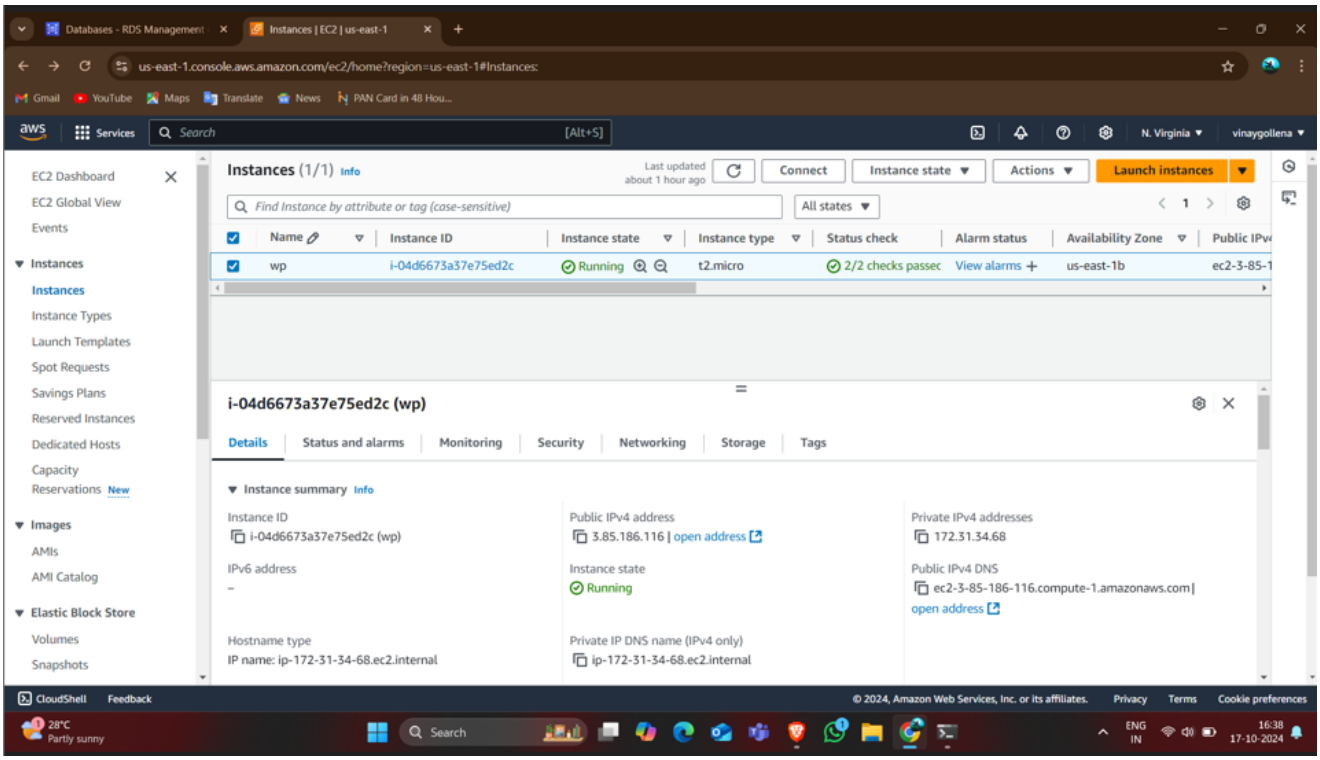
- Now give the name of the database which you give at the stage of DB instance identifier enter same name here.

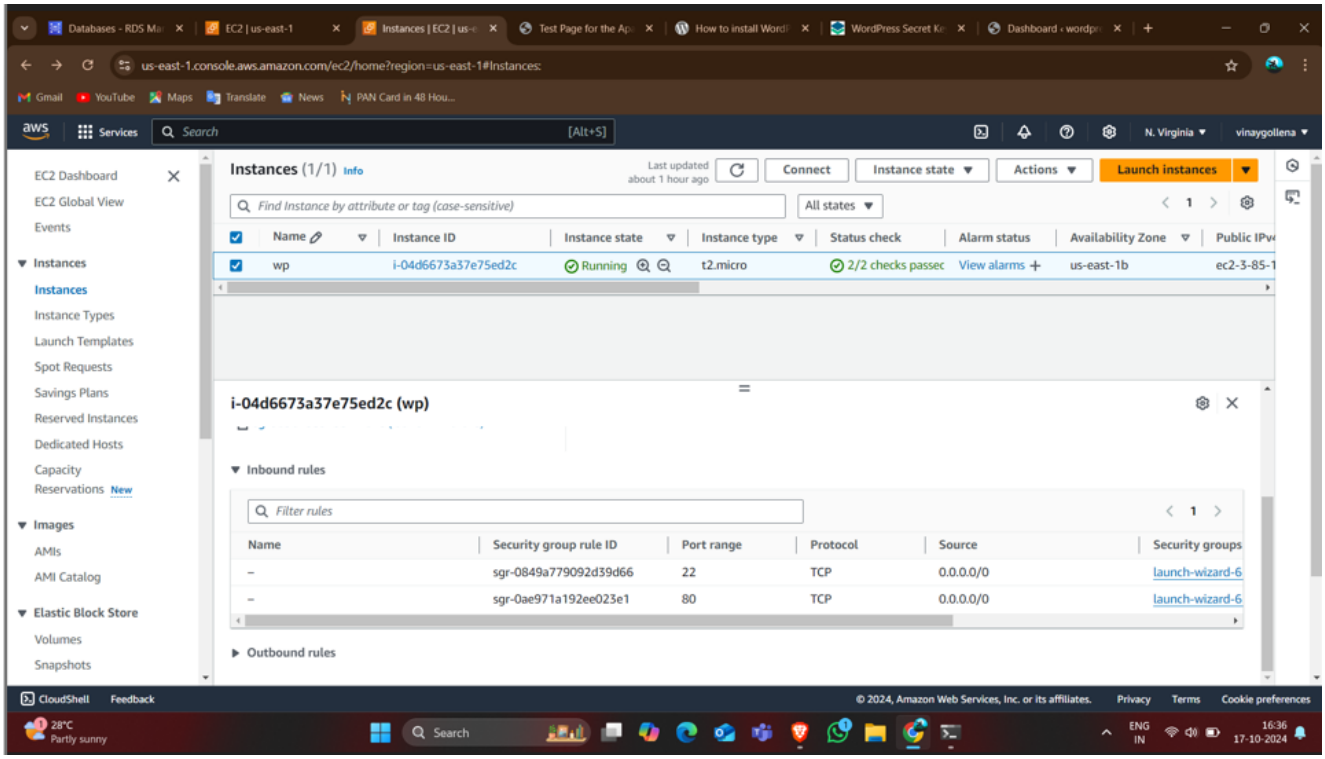


- Now click on the create database button and will create the MYSQL database.



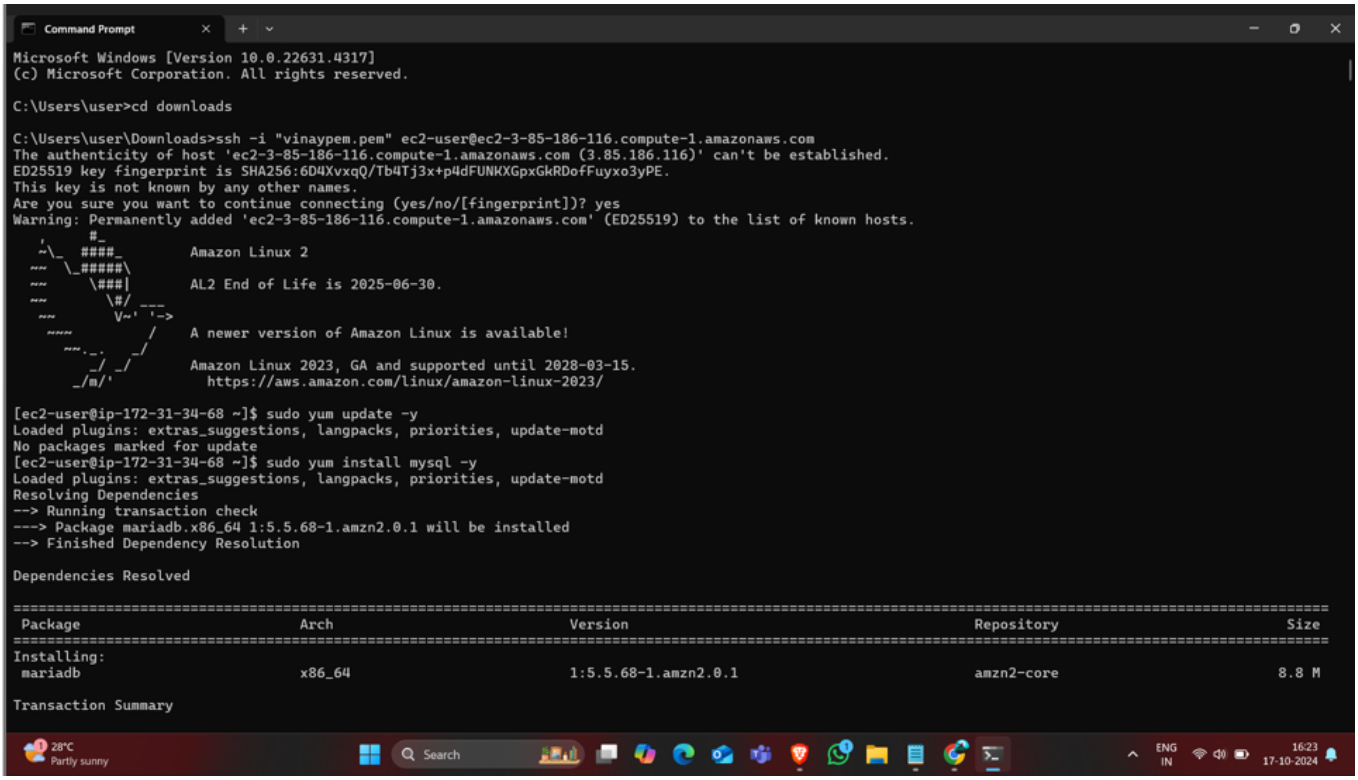
- Now create the ec2 instance by selecting ec2 instance and launch the instance by selecting Amazon Linux -2 version and giving security group with ssh(22) and http(80).



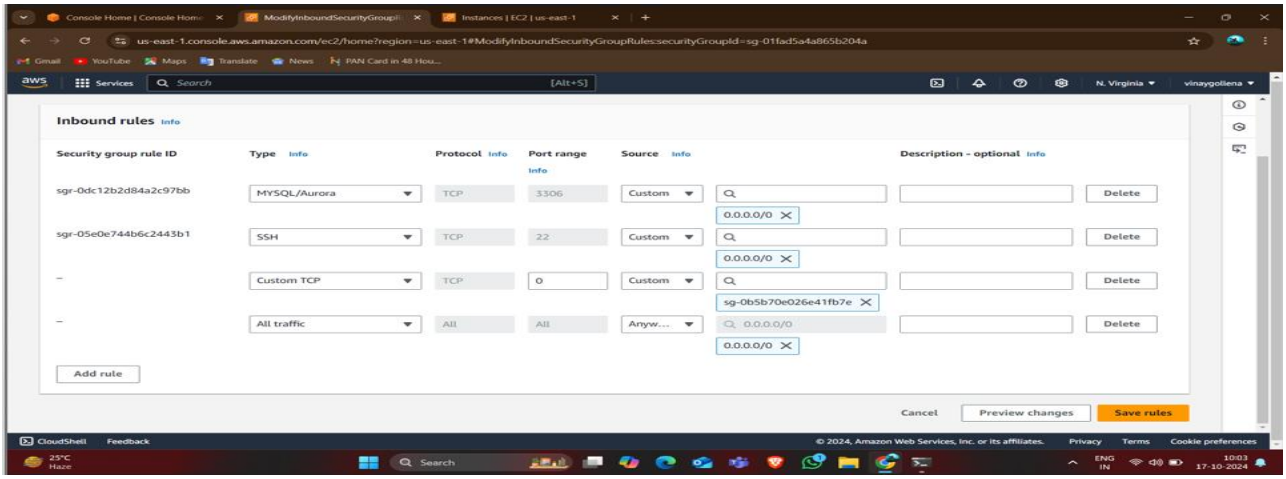


- Now connect the virtual server through the terminal shown in fig.

- Now update the Linux version by using command <sudo yum update -y> and install MySQL by using the command sudo yum install -y mysql2



- Now to allow create traffic from ec2 instance into rds database for that go to rds service and select your database.
- Now go inside the created database and go to the security under this option these is security group id click on that.



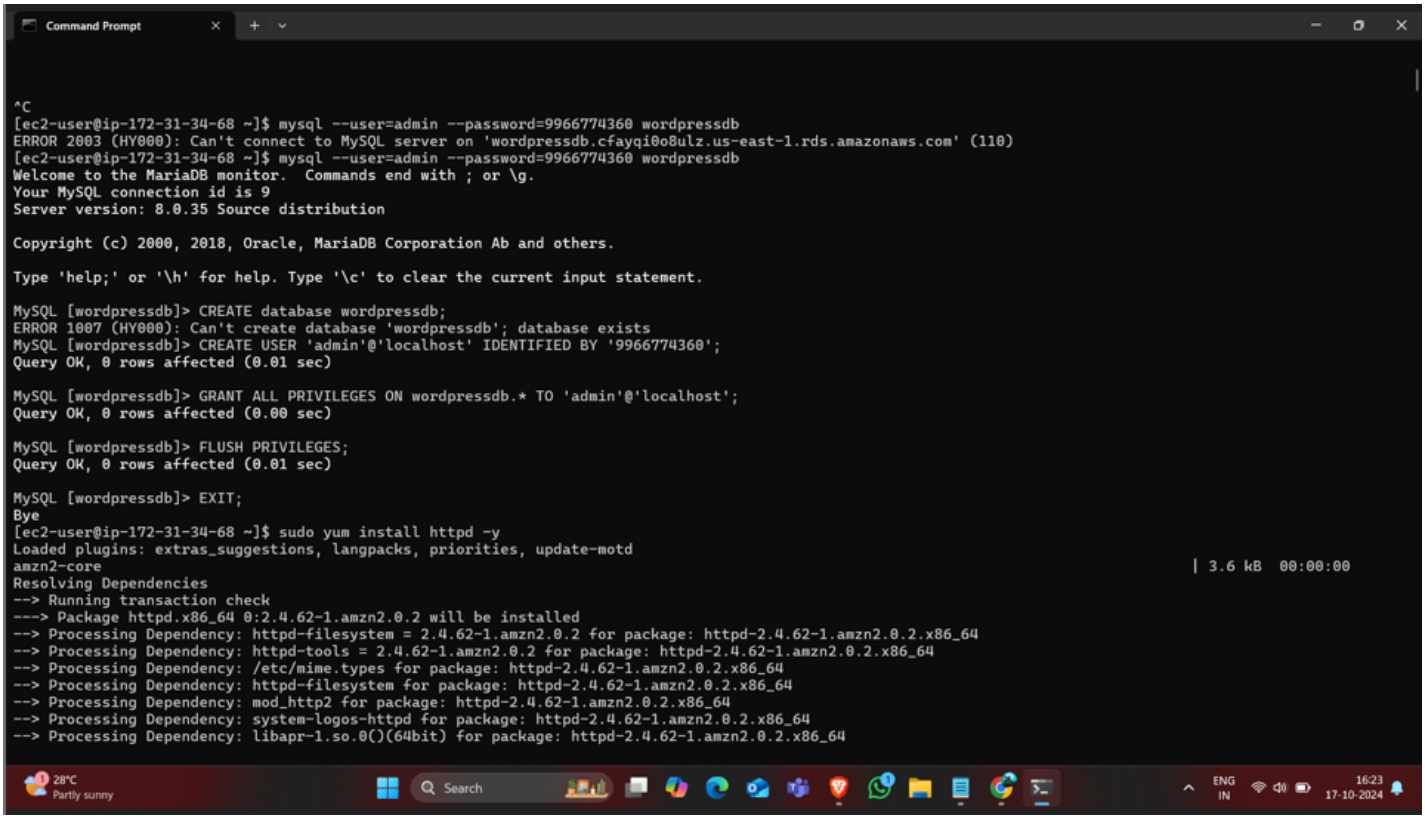
- Now access the MySQL database by using the command as “export MYSQL_HOST <endpoint address>” for that go inside the created database and go to the endpoint address and select and copy the endpoint address
- Now give the credentials of database by giving the command as “MySQL –user=<username> -password=<password> database=”name”
- There is another method command to access your database as “MySQL -h <endpoint address> -u <user>

-p” press enter button it asks the password to enter it, and press enter button

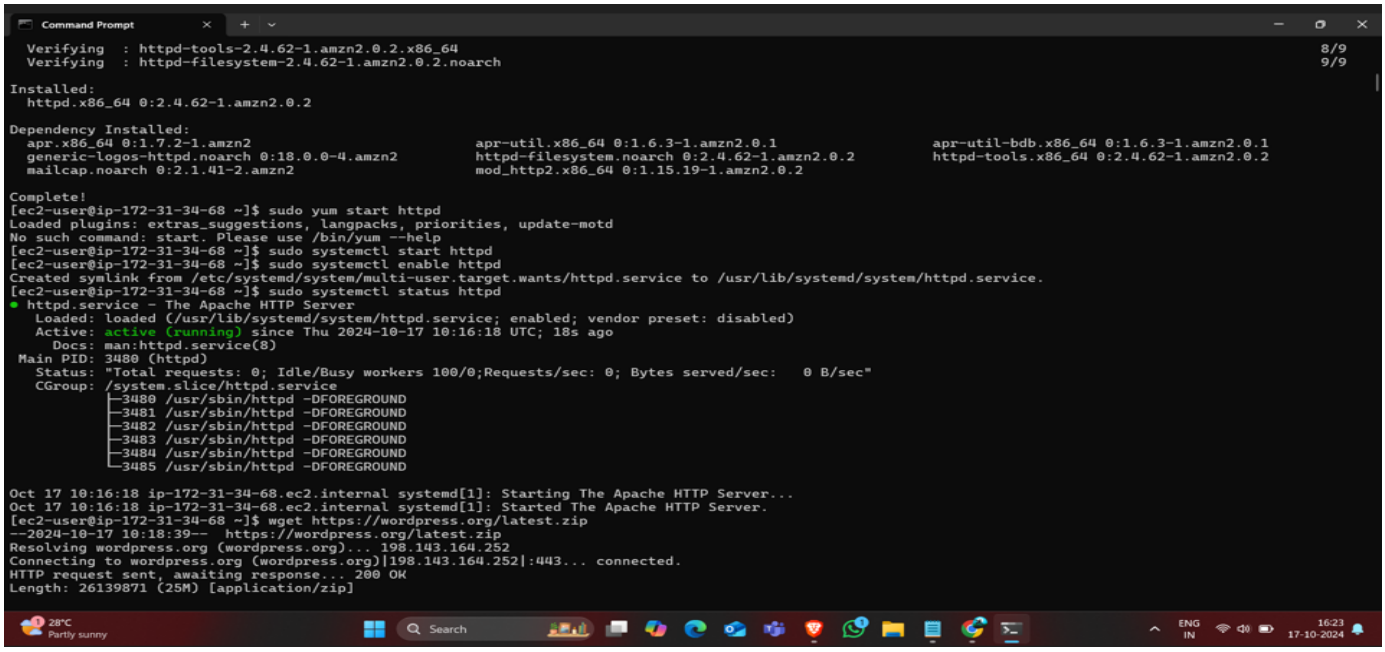
Now create a database user for your WordPress application and give it permission to access the

WordPress database. By using these commands as

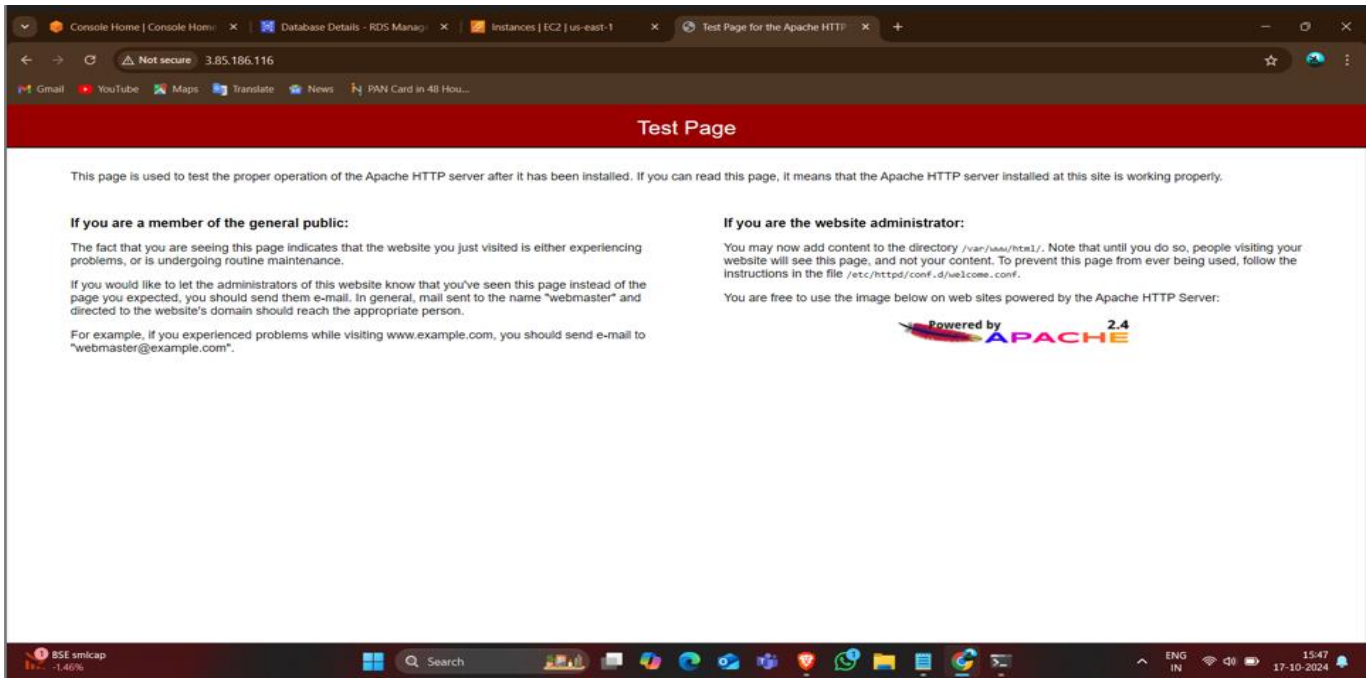
a. CREATE USER WordPress IDENTIFIED BY ‘WordPress-pass’
b. GRANT ALL PRIVILEGE ON WordPress. * TO WordPress
c. FLUSH PRIVILEGES
d. EXIT



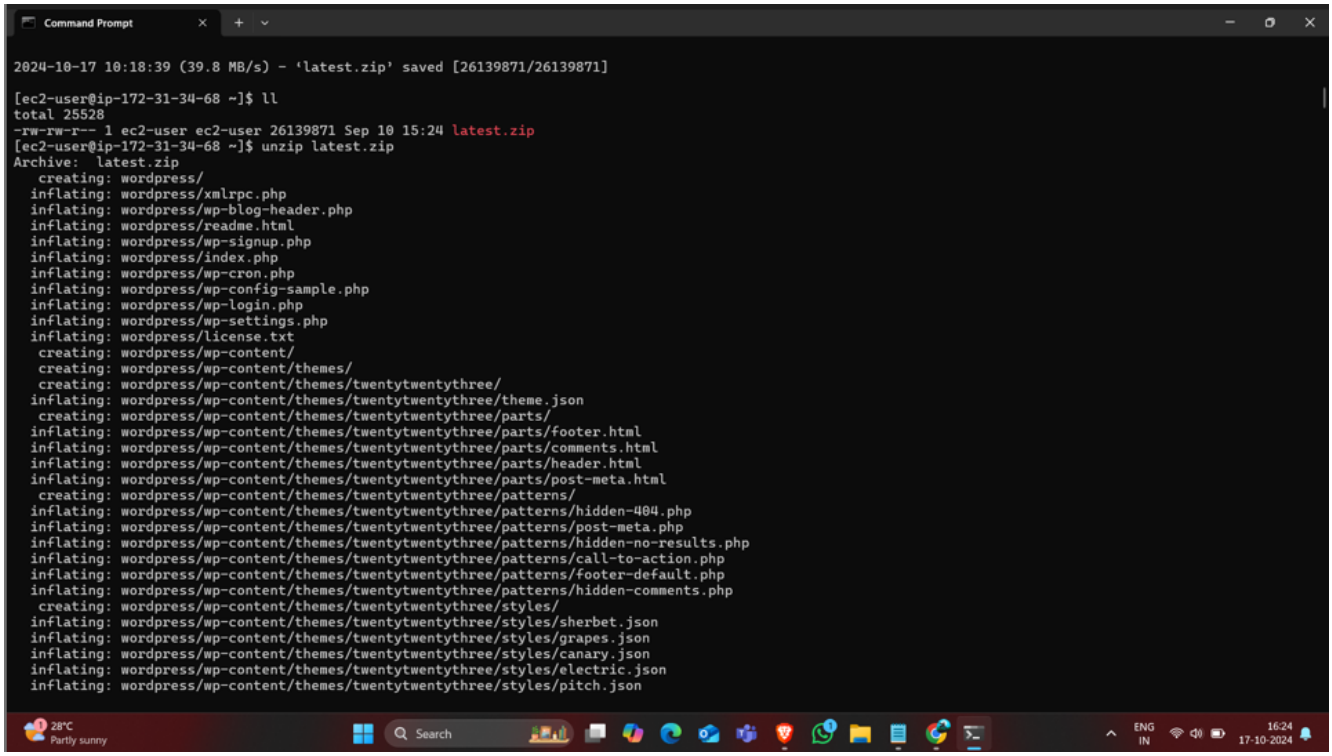
- To host WordPress application need a Apache web server httpd install that by giving command as <sudo yum install httpd -y> and start and enable the httpd service by giving the command as (a). “sudo service httpd start (or) sudo systemctl start httpd (b). Sudo chkconfig httpd on (or) sudo systemctl enable httpd



- Now go to ec2 instance and copy the public ipv4 it on Google. Browse it and check the official page of httpd is displayed.

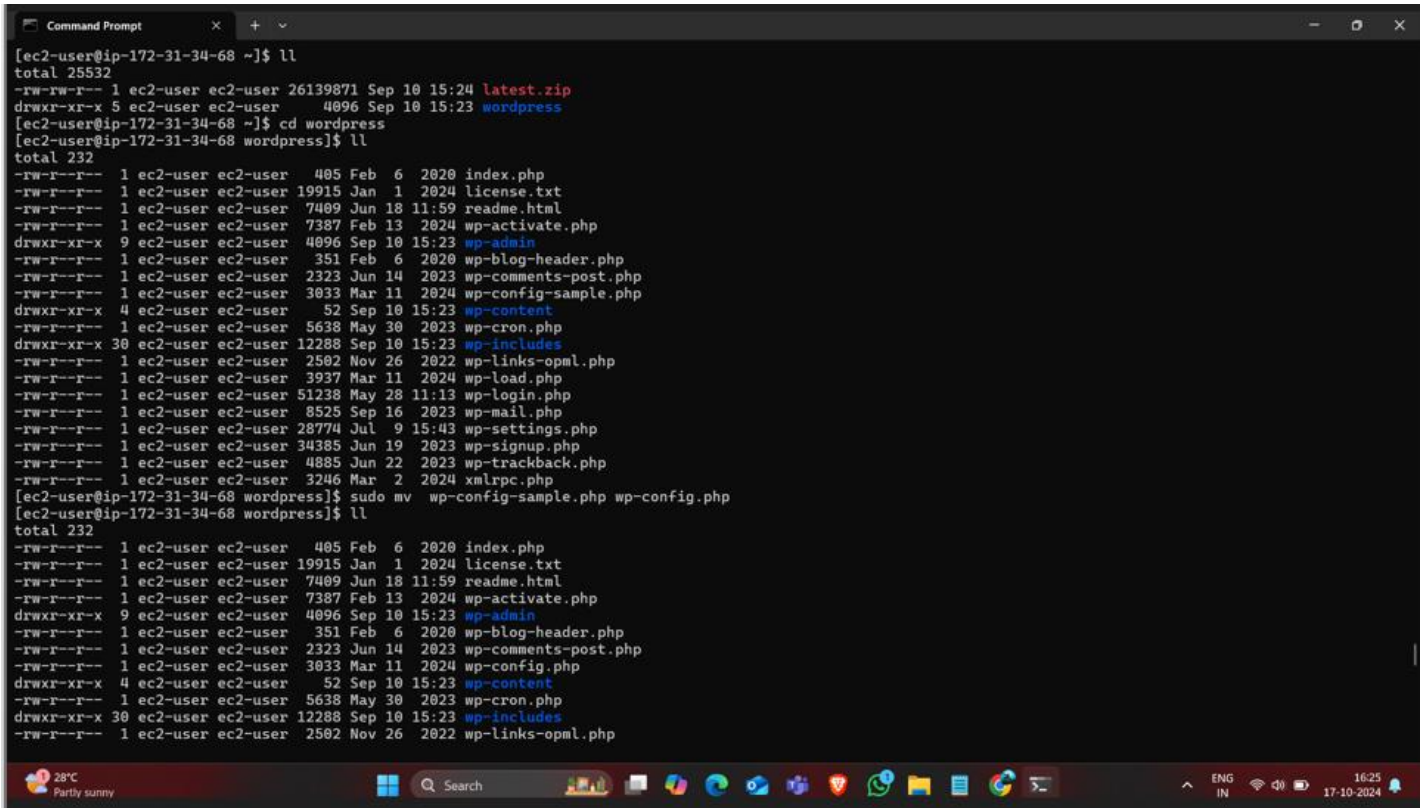


- Now go to browse and select a download WordPress and click on proper link and select and copy the address link of WordPress. Download the file and past that along with wget command in gitbash as “wget <address. Link>”
- It gives the zip file to unzip that file by using a command as “unzip <zip file>”

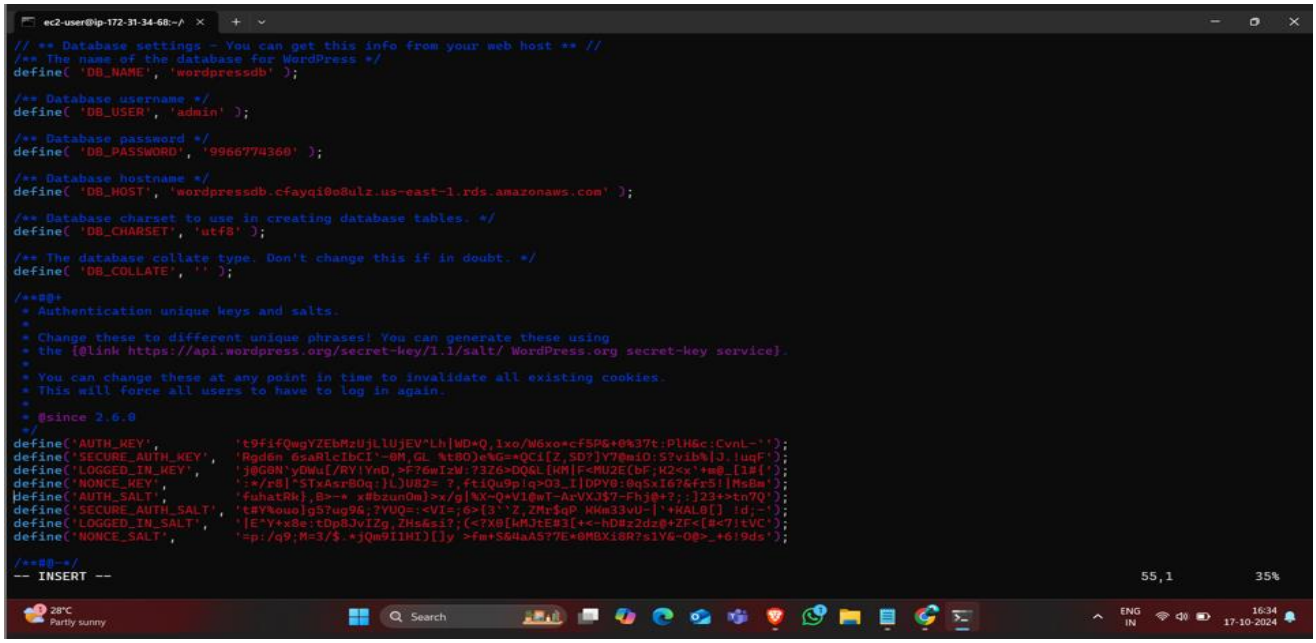


- To run WordPress web application, you must install of WordPress web application as php language with the following commands <sudo Amazon-linux-extras install -y lamp-mariadb10.2-php7.2.php7.2> otherwise update ec2 instance with the following commands <sudo yum update -y>
- Now go inside the unzip directory by using command as “cd <unzip directory> and change the WordPress configuration file by giving command as “sudo mv wp-config-sample.php wp-config.php”

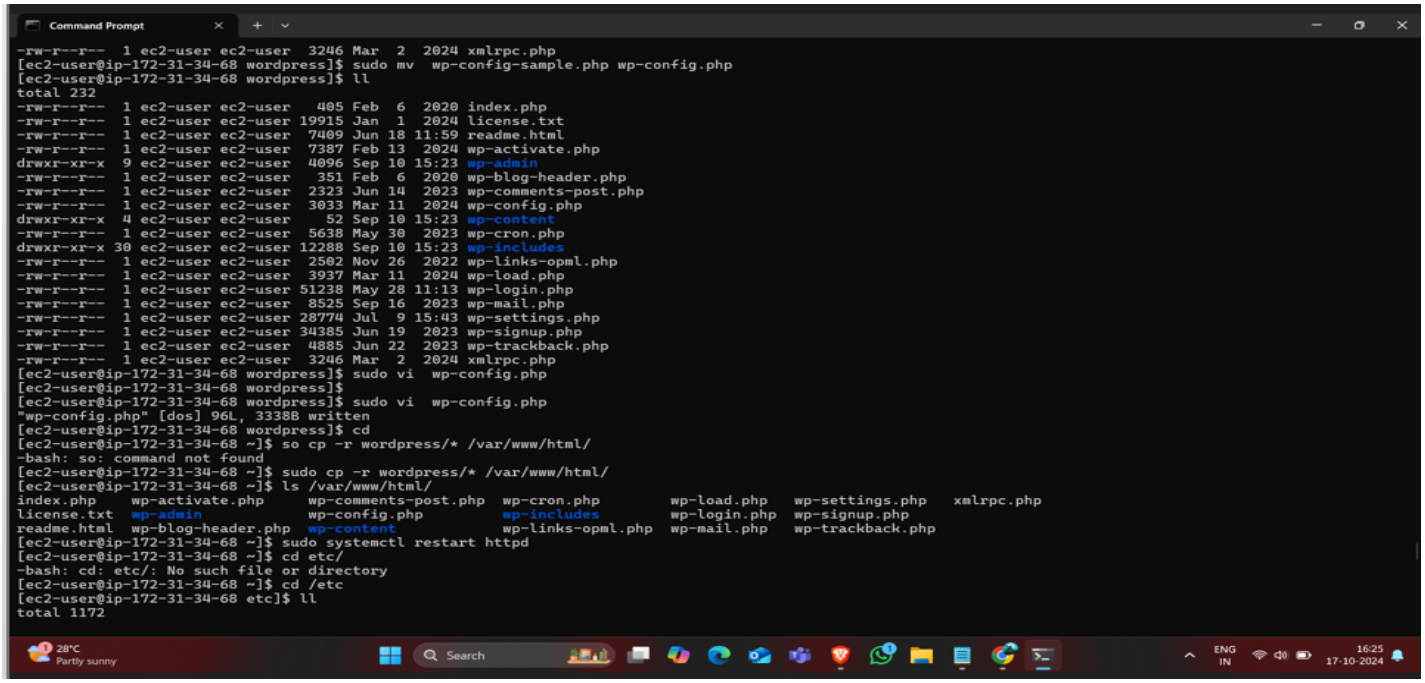
•



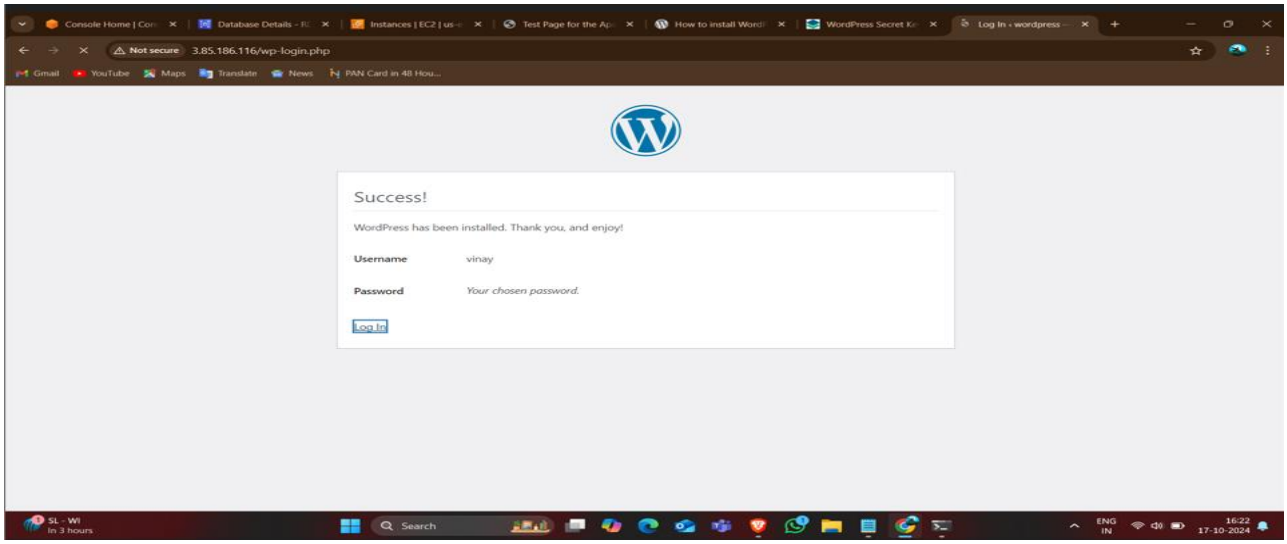
- Now do some configurations in the WordPress configuration file by giving database name, username, password and hostname for that execute as sudo vi wp-config.php" .along with these change the WordPress secret key. To get the secret key browse the Google generate WordPress secret key.



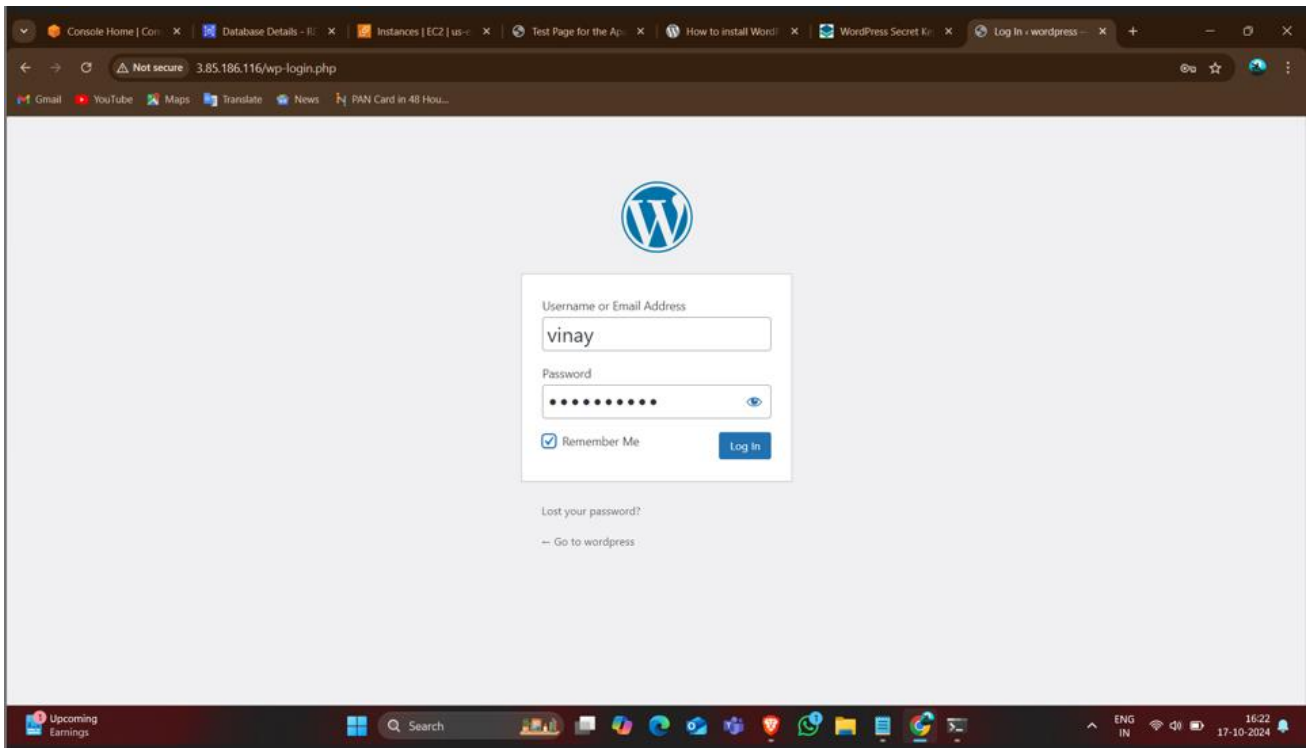
- Now copy the WordPress directory to the document root directory to host the web application of
- WordPress by giving as command as
- Sudo cp -I <unzip file>/* /var/www/html/
- To restart the httpd server by using command sudo systemctl restart the httpd

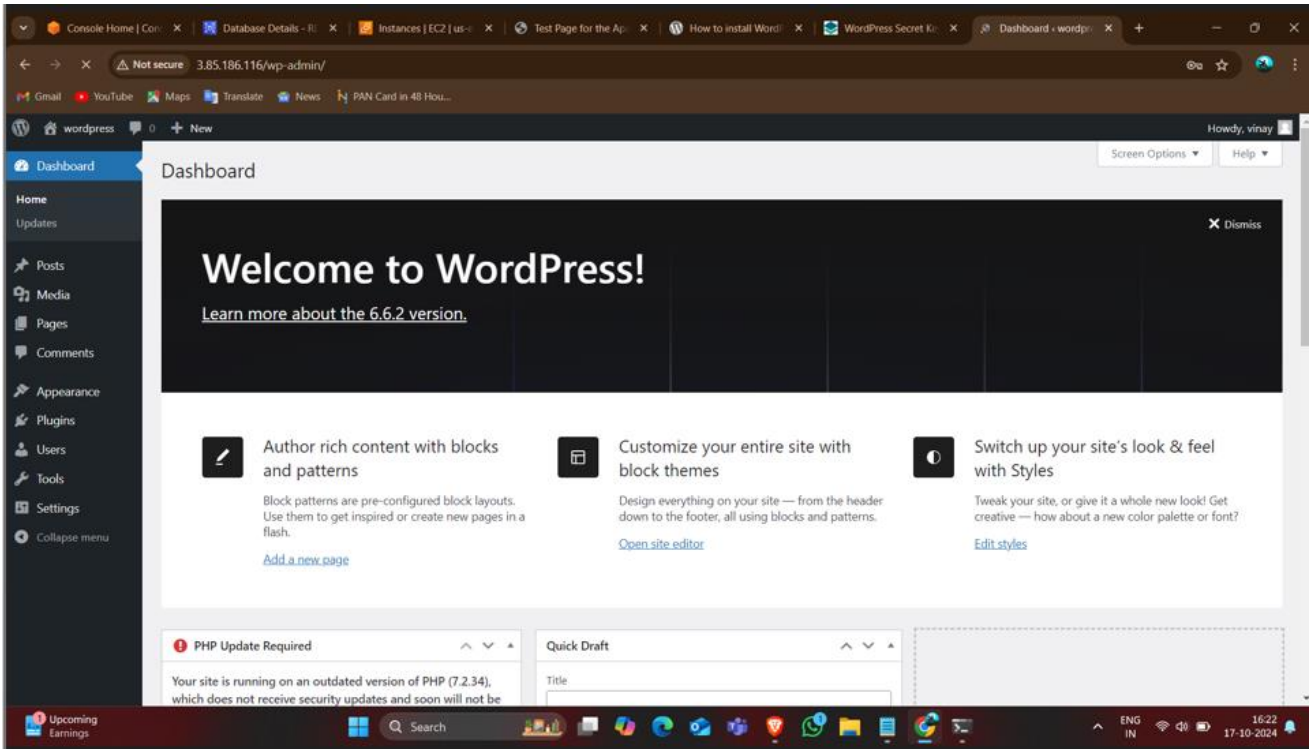


- Now go to ec2 instance and copy public ipv4 and paste it on google browse it and check the official page of WordPress it displays.Than register and login.



•


Not secure 3.85.186.116/wp-login.php
Username or Email Address
vinay
Password
Remember Me
Log In
Lost your password?
← Go to wordpress

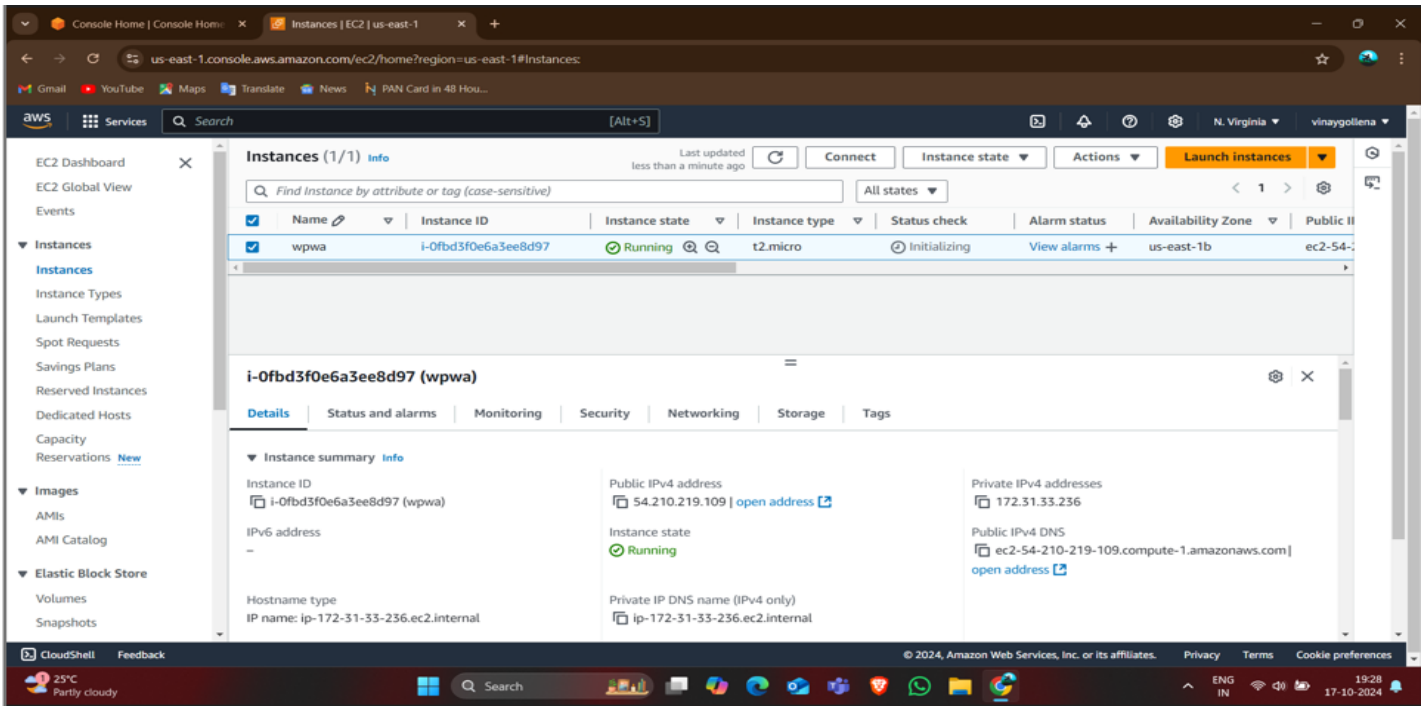

Not secure 3.85.186.116/wp-admin/
wordpress
New
Howdy, vinay
Dashboard
Screen Options
Help
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Welcome to WordPress!
Learn more about the 6.6.2 version.
Dismiss
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
PHP Update Required
Your site is running on an outdated version of PHP (7.2.34), which does not receive security updates and soon will not be
Quick Draft
Title

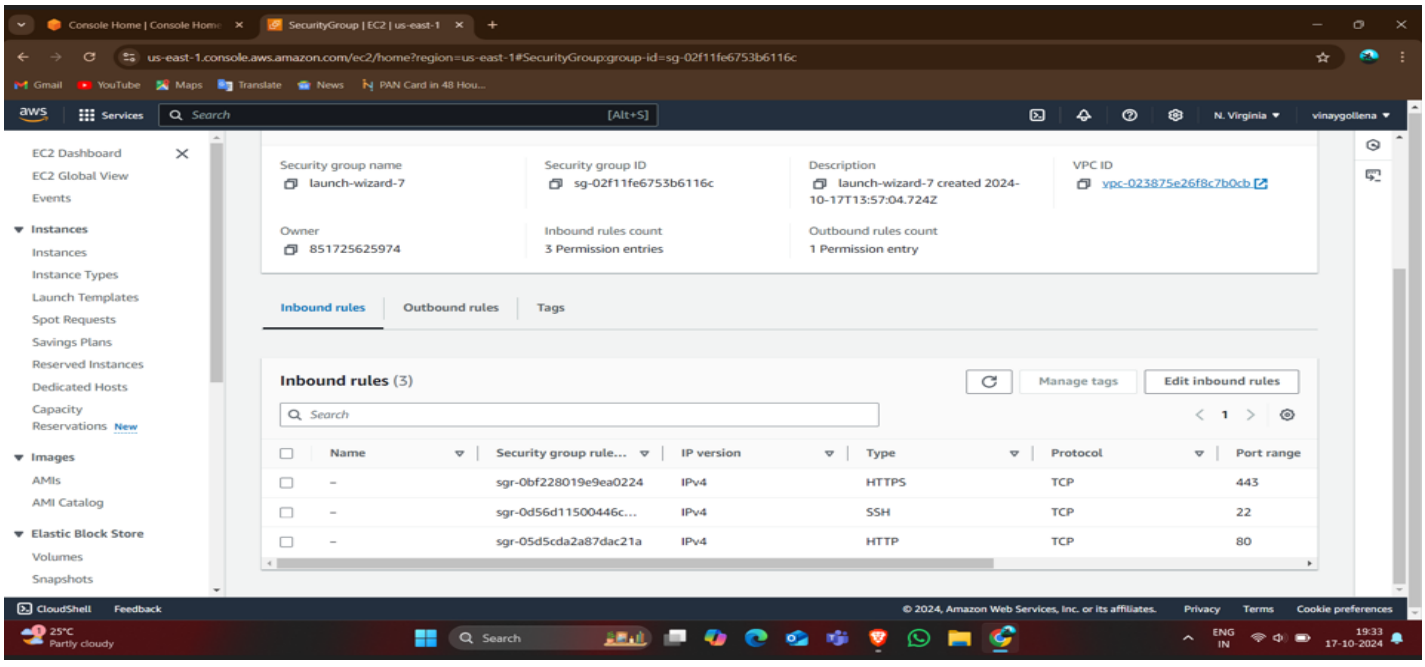
•

Deploying the Wordpress web application by using Docker

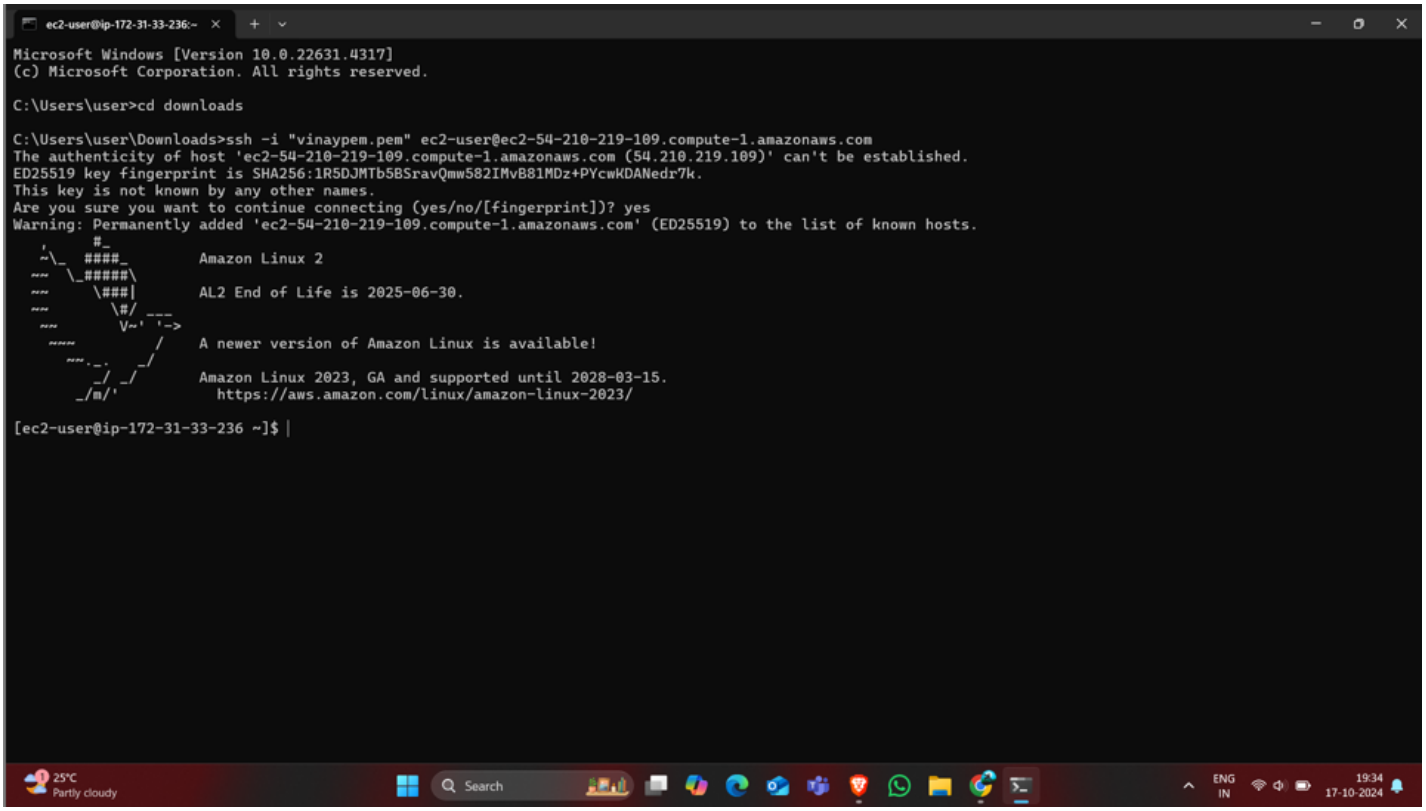
Module-1:

- Login to the AWS consol.
- Launch the EC2 instance.
- Use AMI amazon-linux 2,t2.micro,keypair.
- Port 80(HTTP),443(HTTPS),& 22(SSH).
- Connect to the terminal.



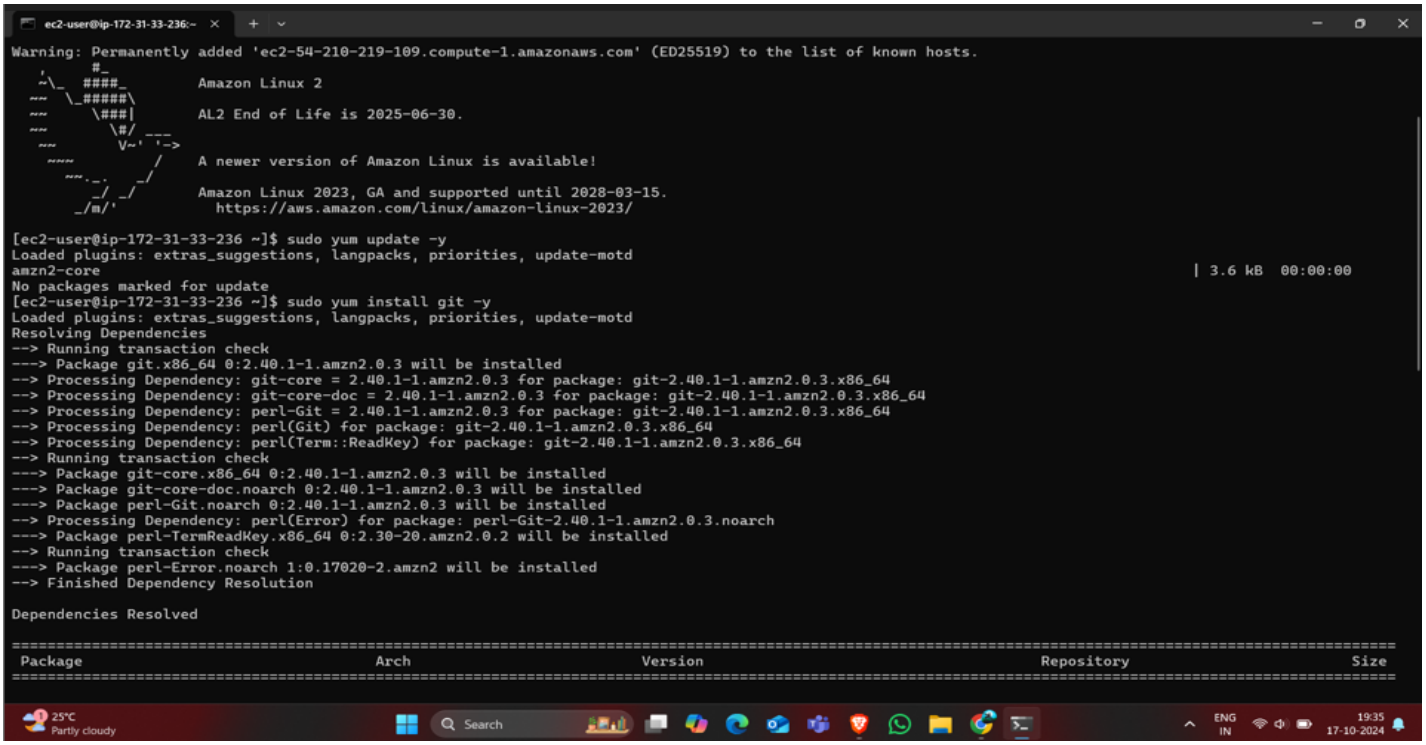


•

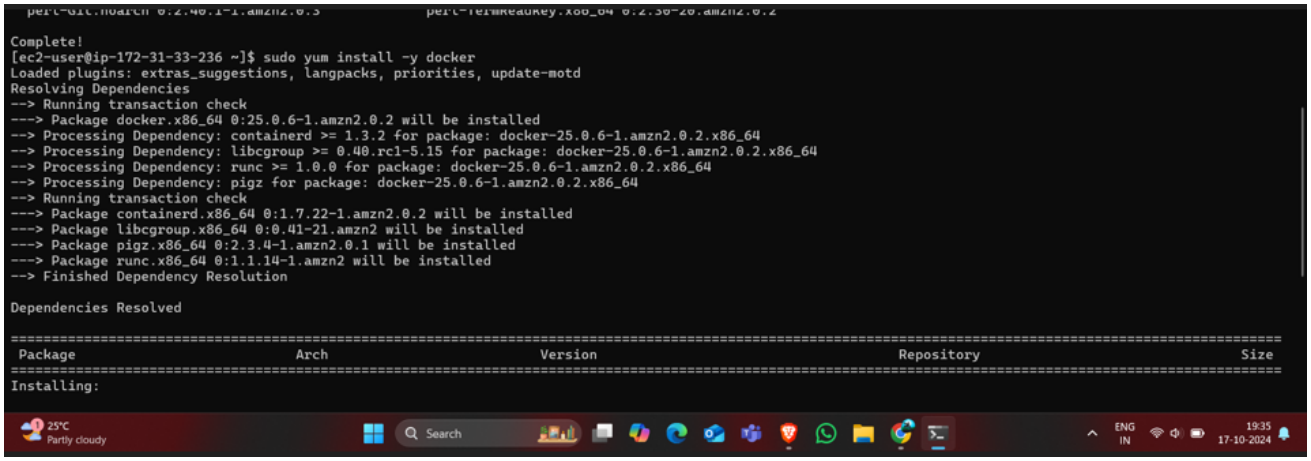


Module-2:

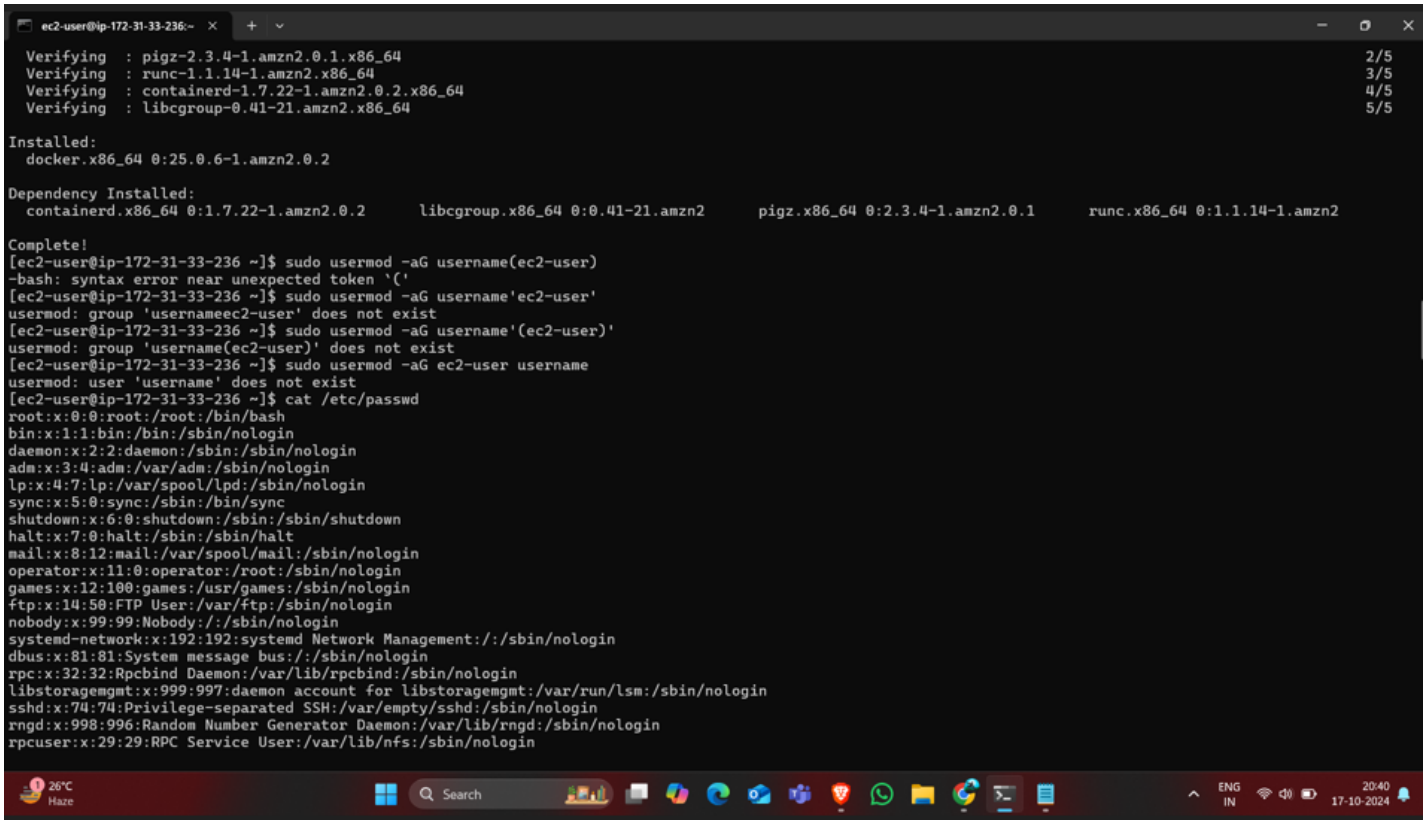
- Install git “sudo yum –y install git”



- Install Docker “sudo yum install docker –y”

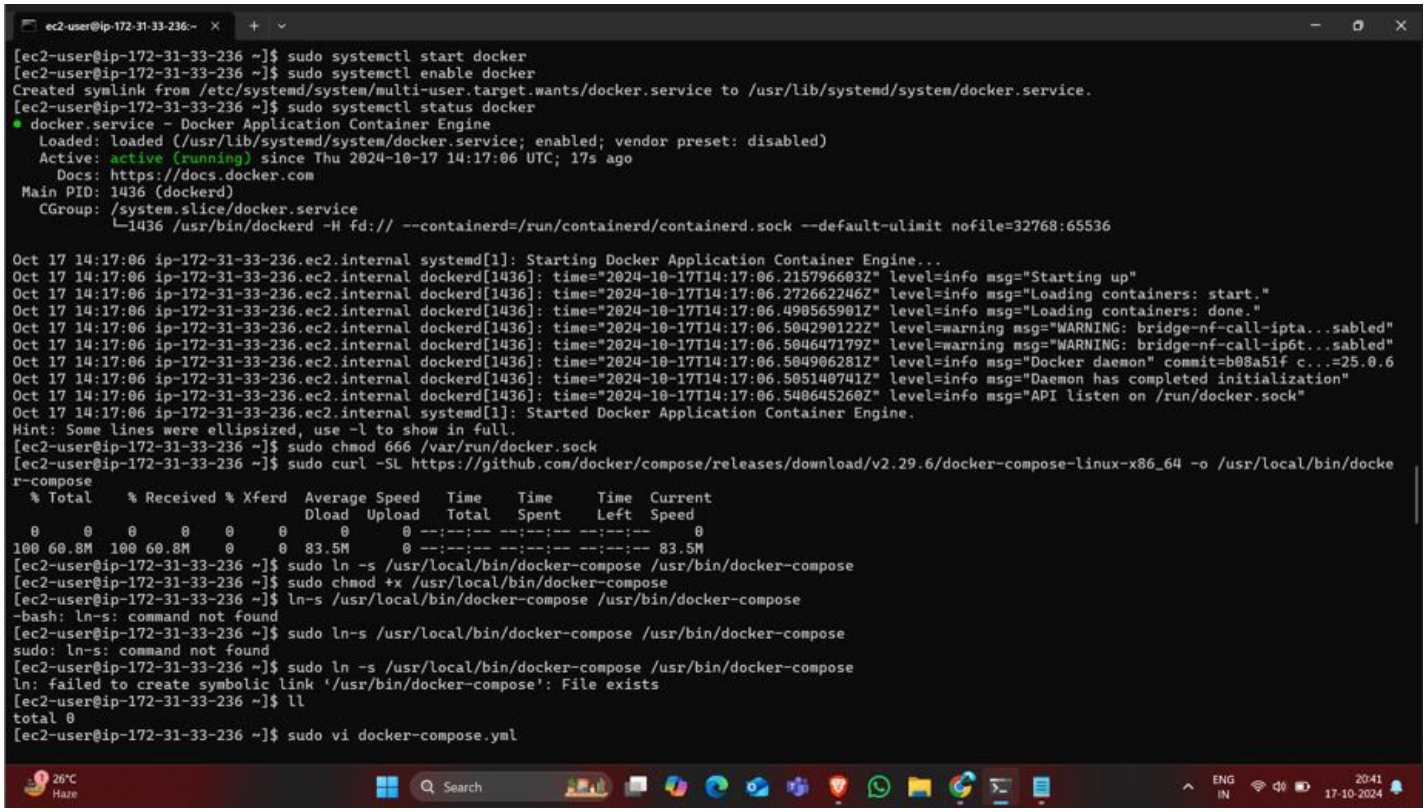


- Now give the permissions to add limited user account to docker group “sudo usermod –aG username” or “sudo chmod 666 /var/run/dokcer.sock”

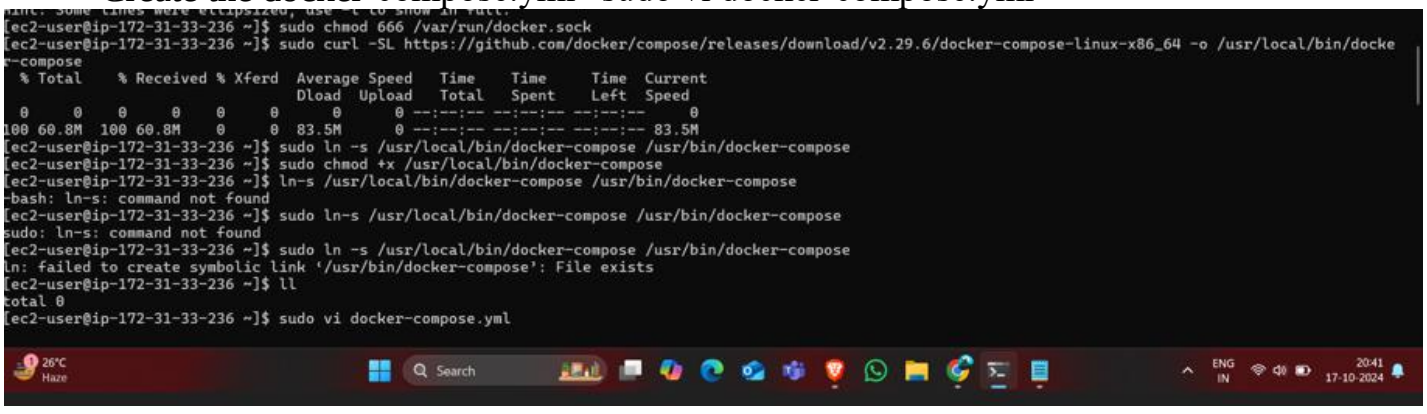


- Now start and enable the docker
  “sudo systemctl start docker”
  “sudo systemctl enable docker”
  “sudo systemctl status docker”

•

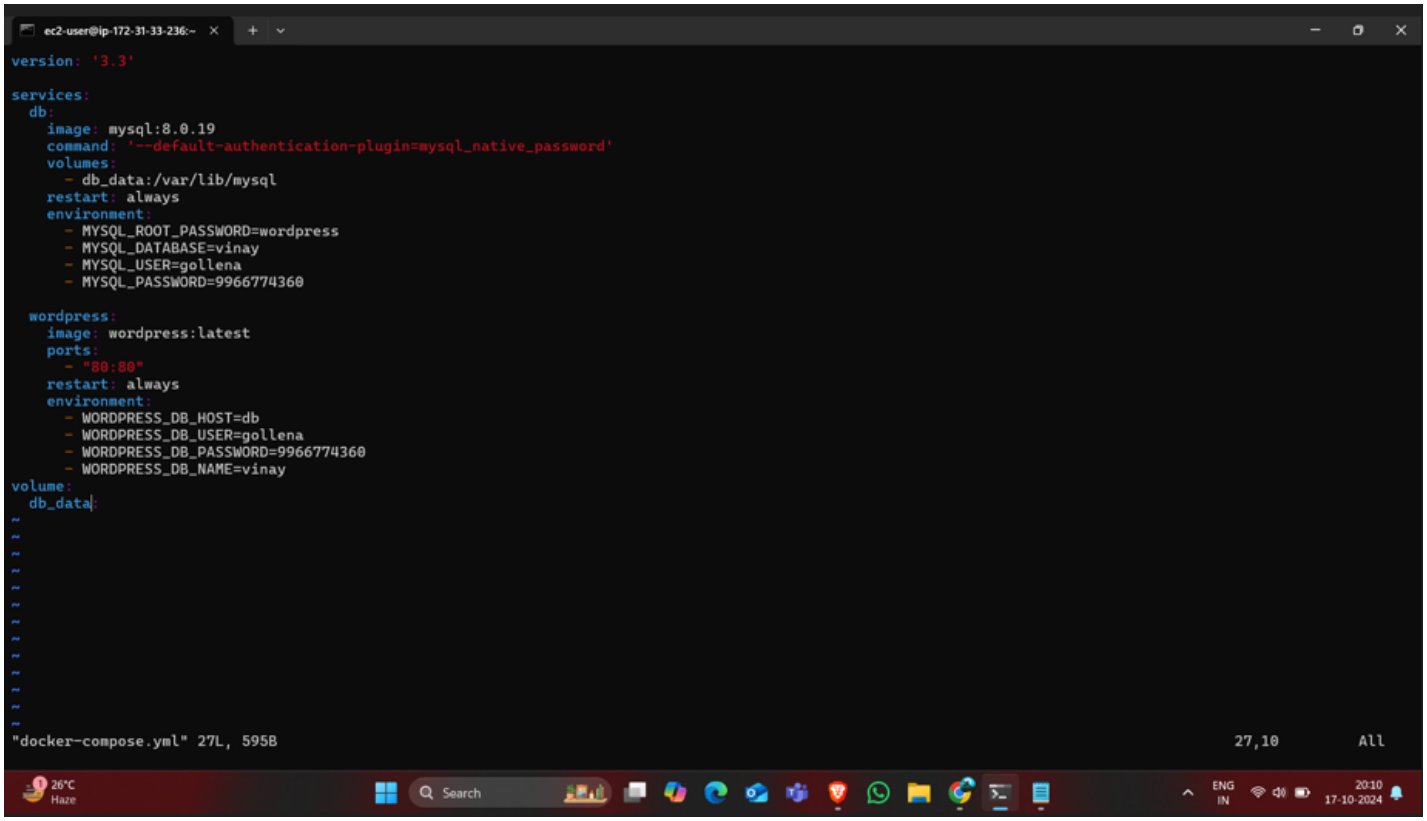


- Now install the docker compose by searching the commands in google (install docker compose).
- Now give the executable permissions to the binary by using “sudo chmod +x /usr/local/bin/docker-compose”
- Create the system link “sudo ln –s /usr/local/bin/docker-compose /usr/bin/docker-compose”.
- Create the docker-compose.yml “sudo vi docker-compose.yml”

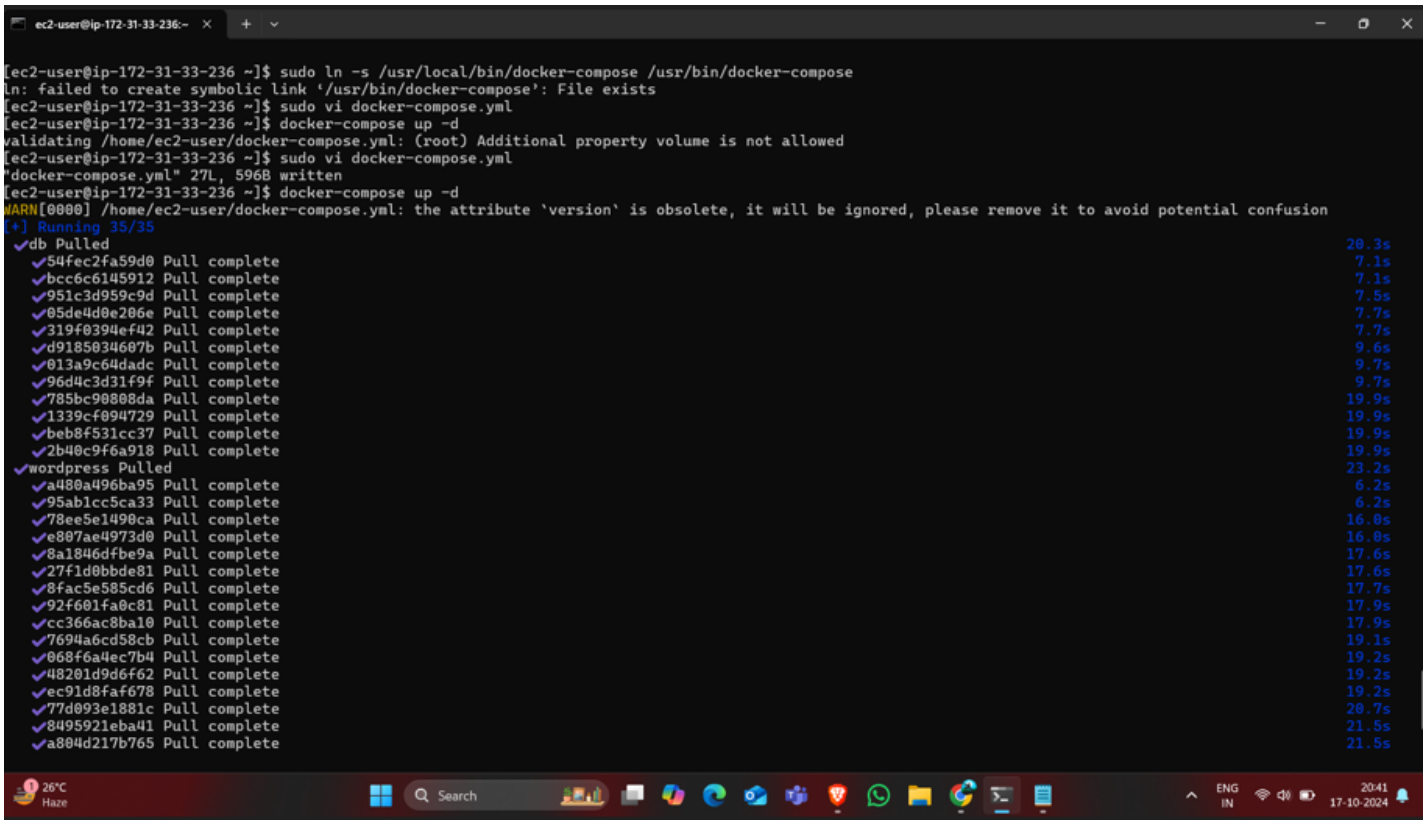


- Now writhe yaml code to run the wordpress web application.

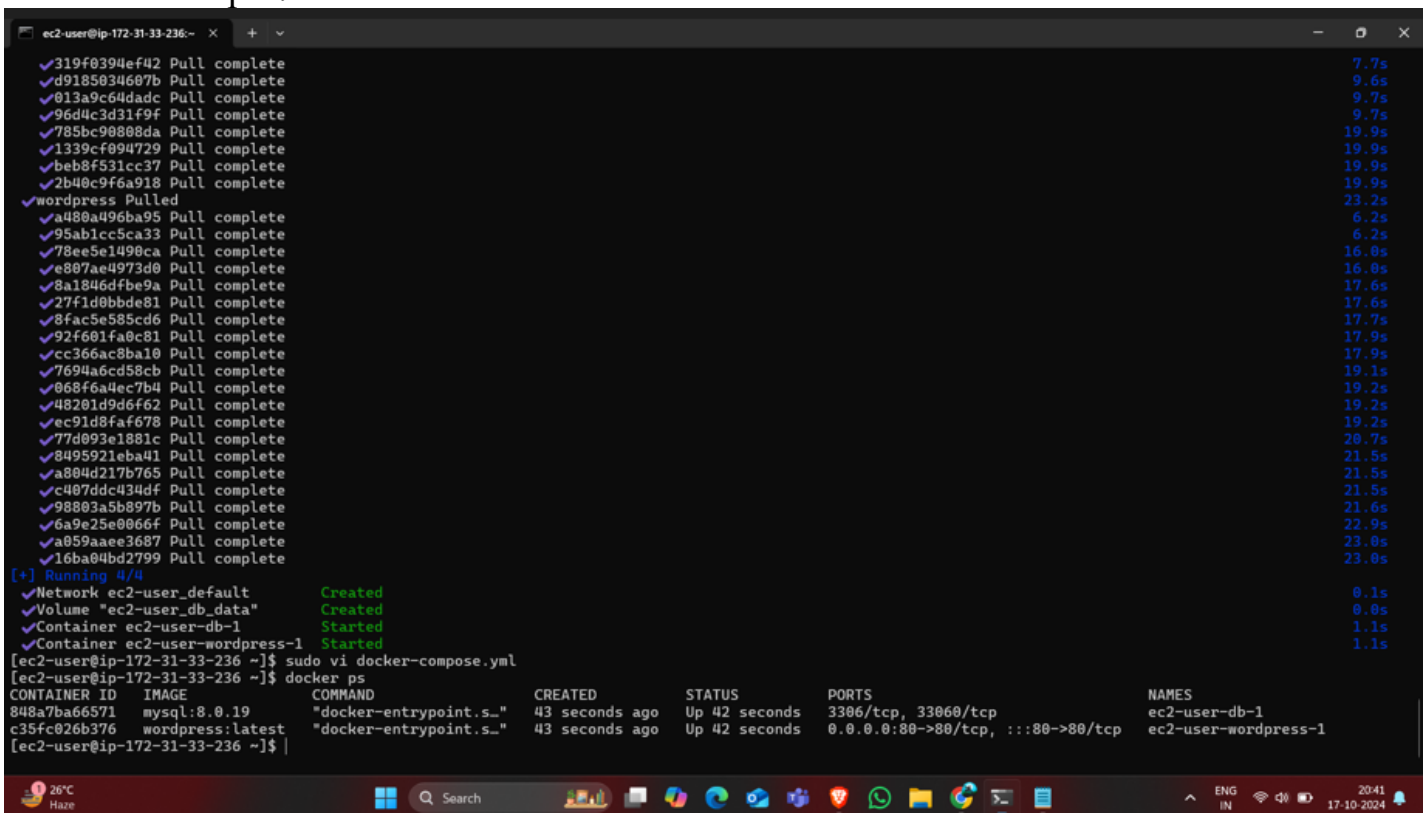
•



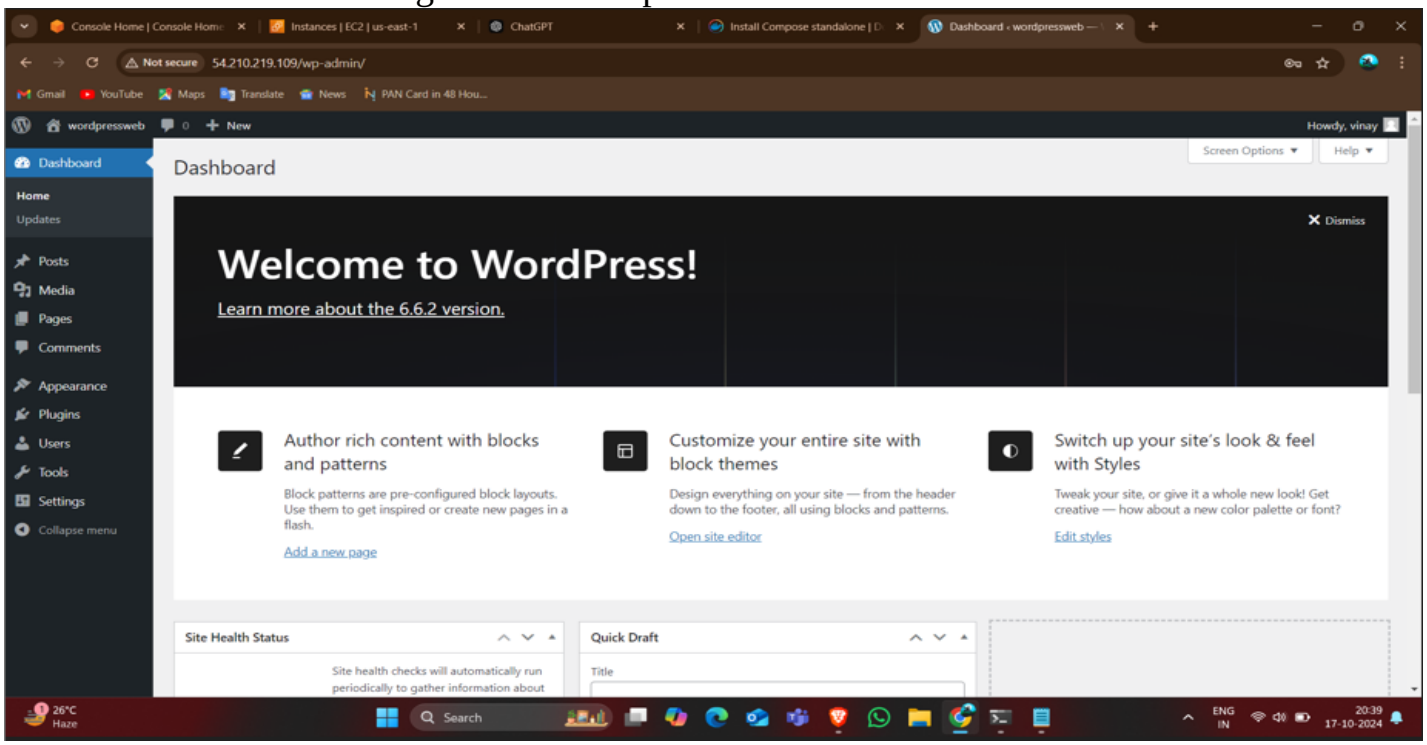
- Now pull file “docker-compose up –d”



- Now see the containers of the pulled images of MYSQL and Wordpress by using “docker ps”.



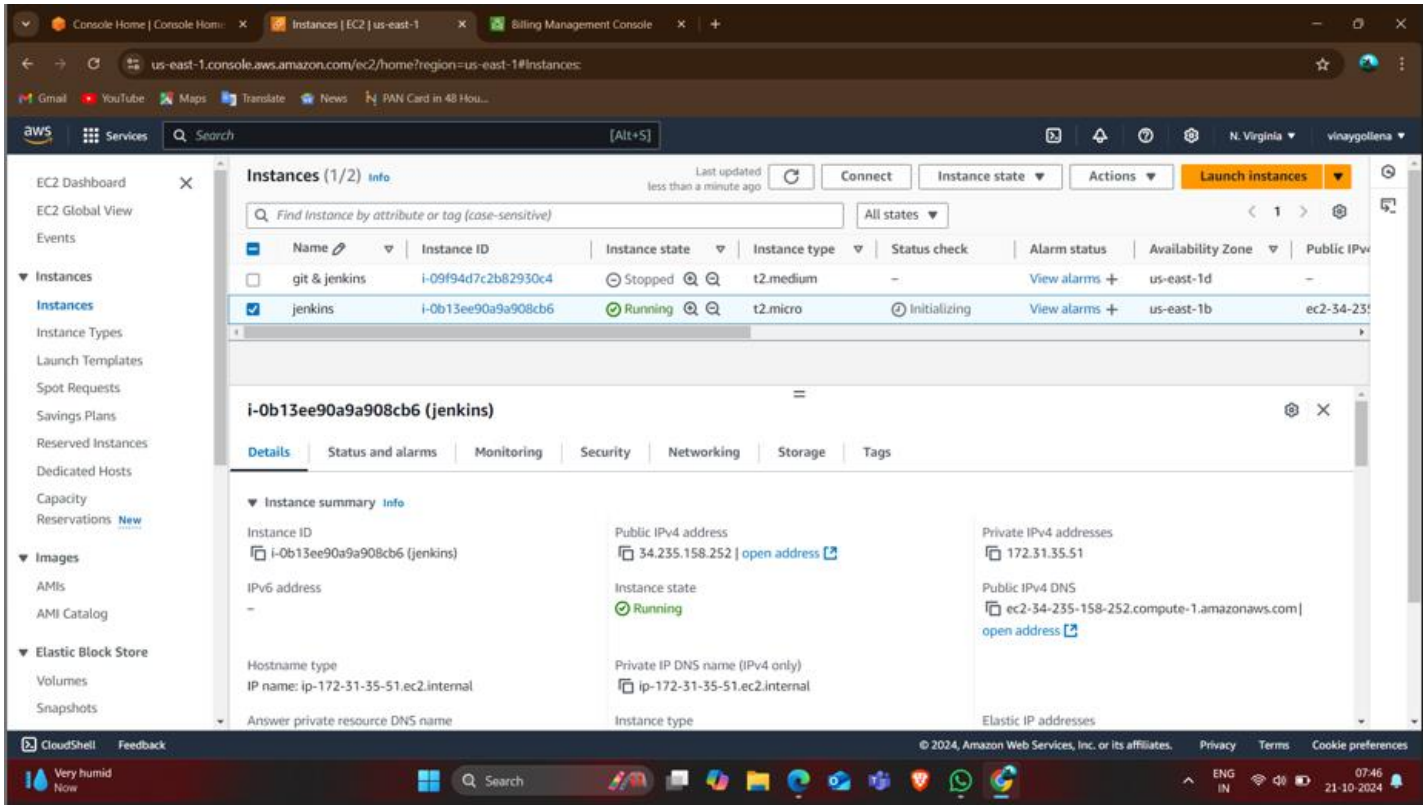
- Now copy the static public ip of your ec2 instance and paste it in the browser with ip:80 and search.
- Fill the details and login to the wordpress.



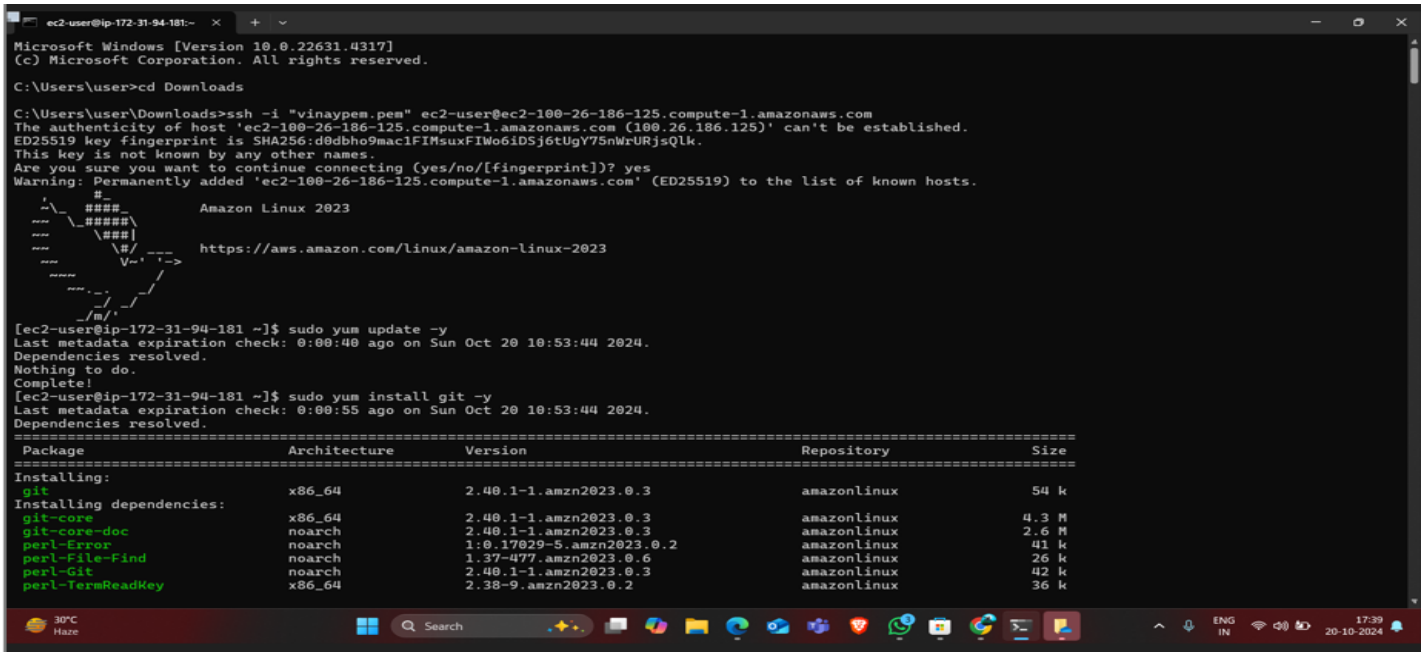
•

Deploy WordPress web application by using git and Jenkins

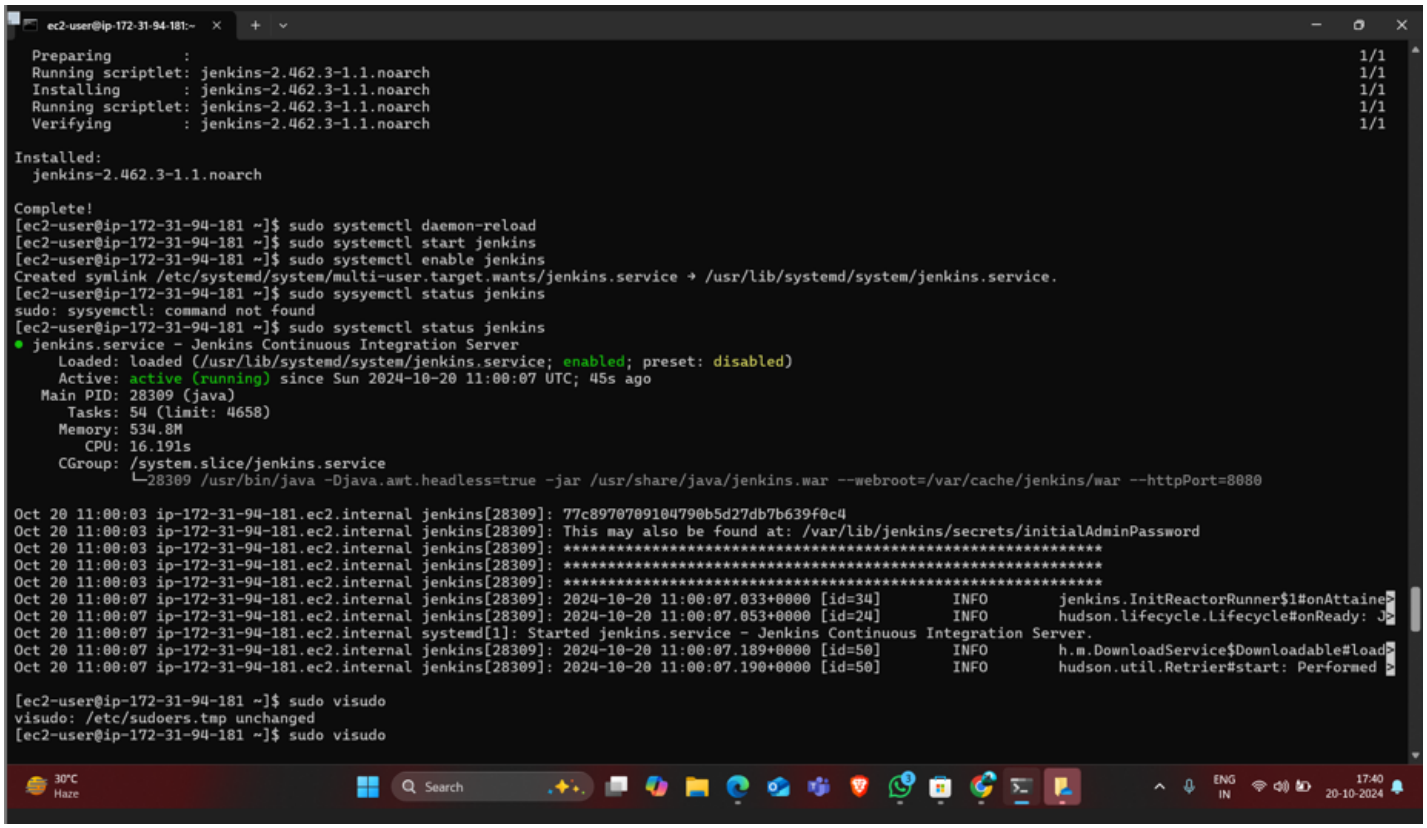
- Create an EC2 instance with t2.micro and connect it to the terminal



- Update “sudo yum update –y”
- And install git “sudo yum install git –y”

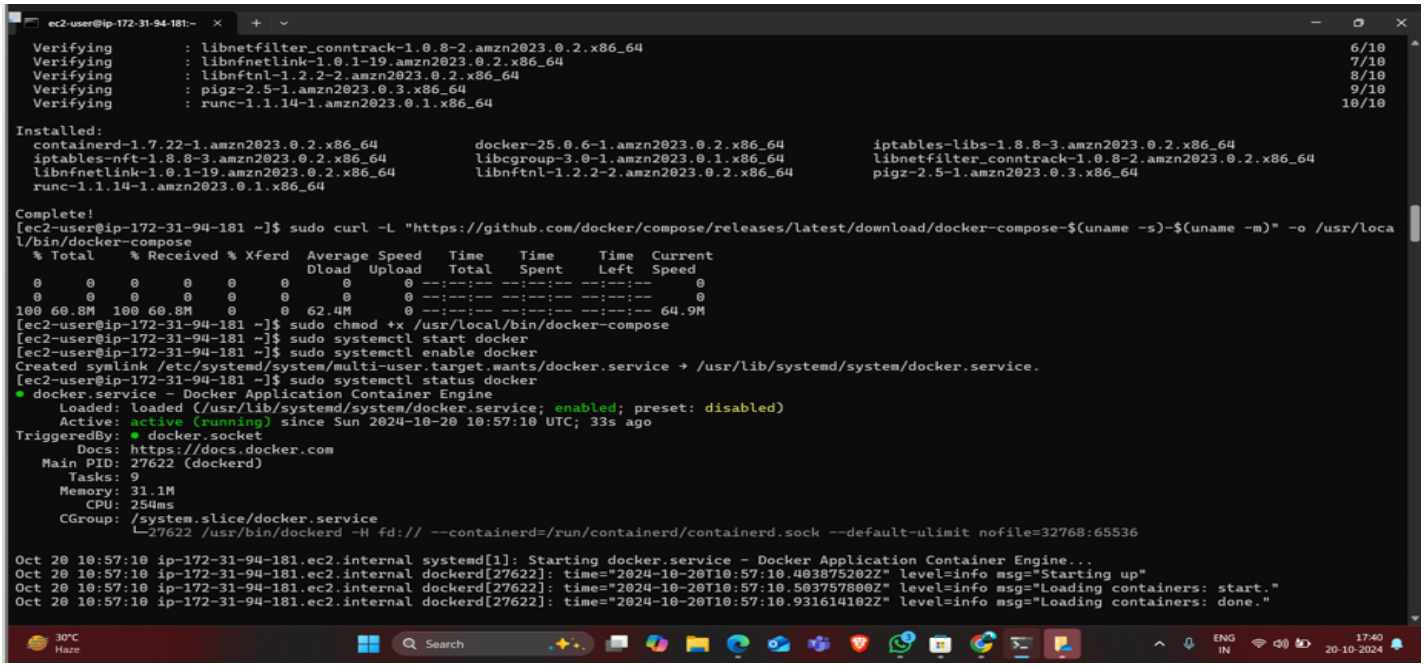


- Install Jenkins by using google (install Jenkins) select the amazon linux or Ubuntu as your wish then copy from it and paste in the terminal.
- Than start and enable the Jenkins and check the status.

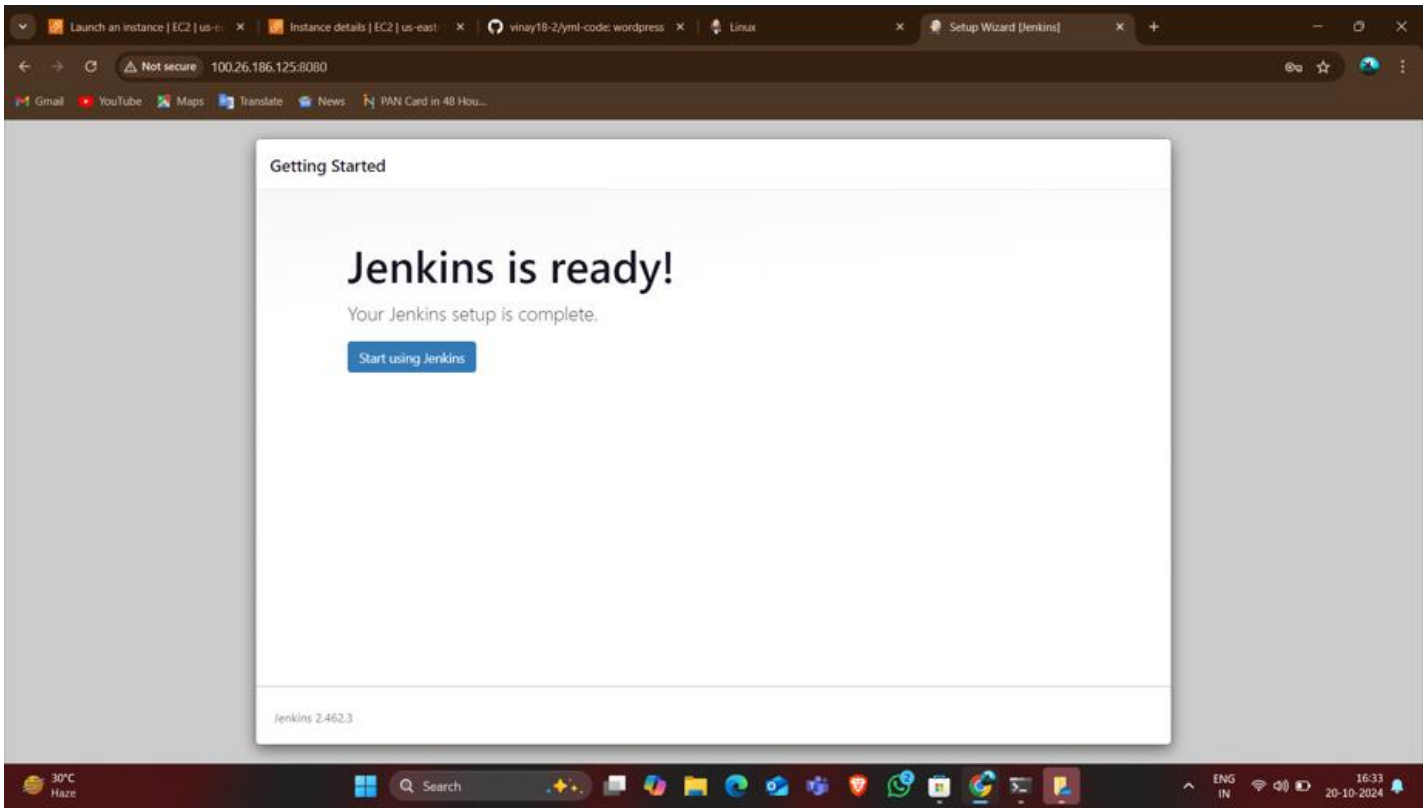


- Than install the docker and docker-compose. Than start the docker and give the limited docker usage permissions.
- Give the command “docker-compose up –d” to pull the image.

•

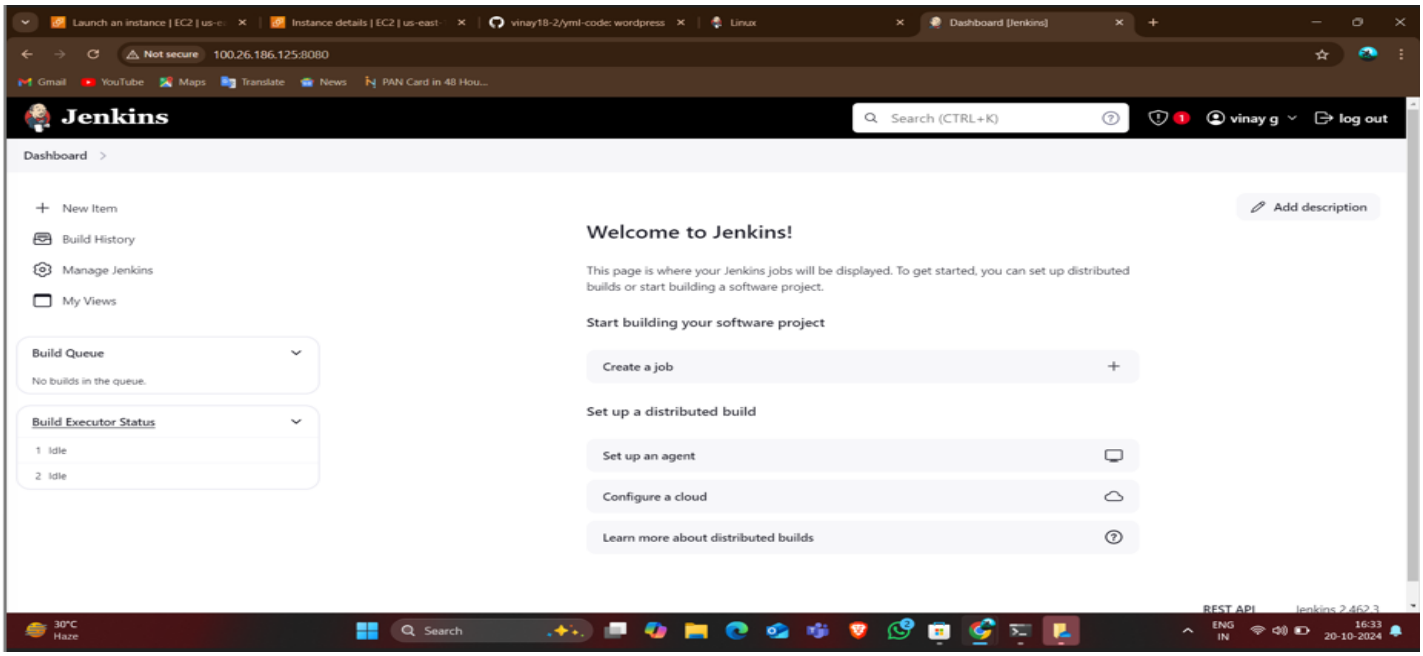


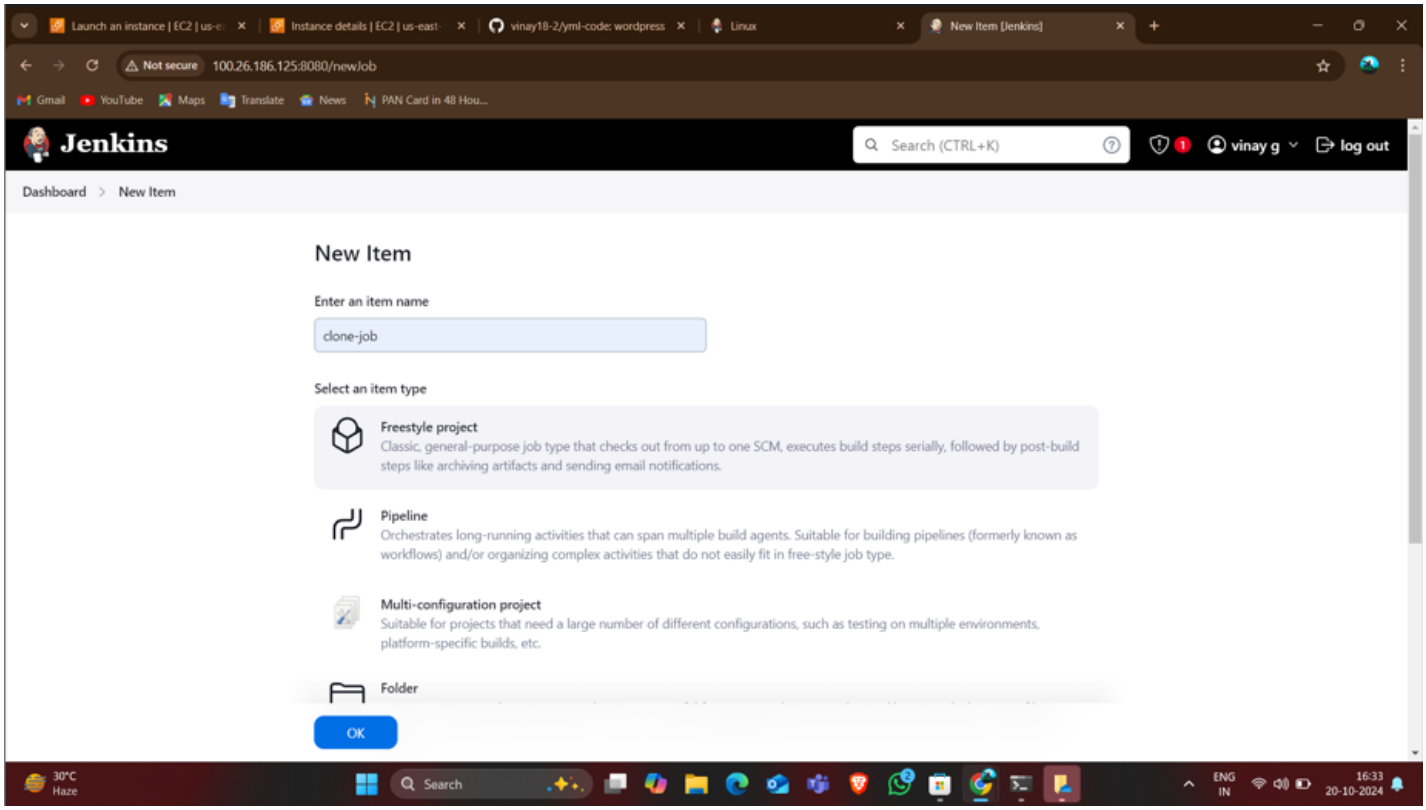
- Copy the ip address with port number 8080 and browse it and login to the Jenkins page by using secret key.



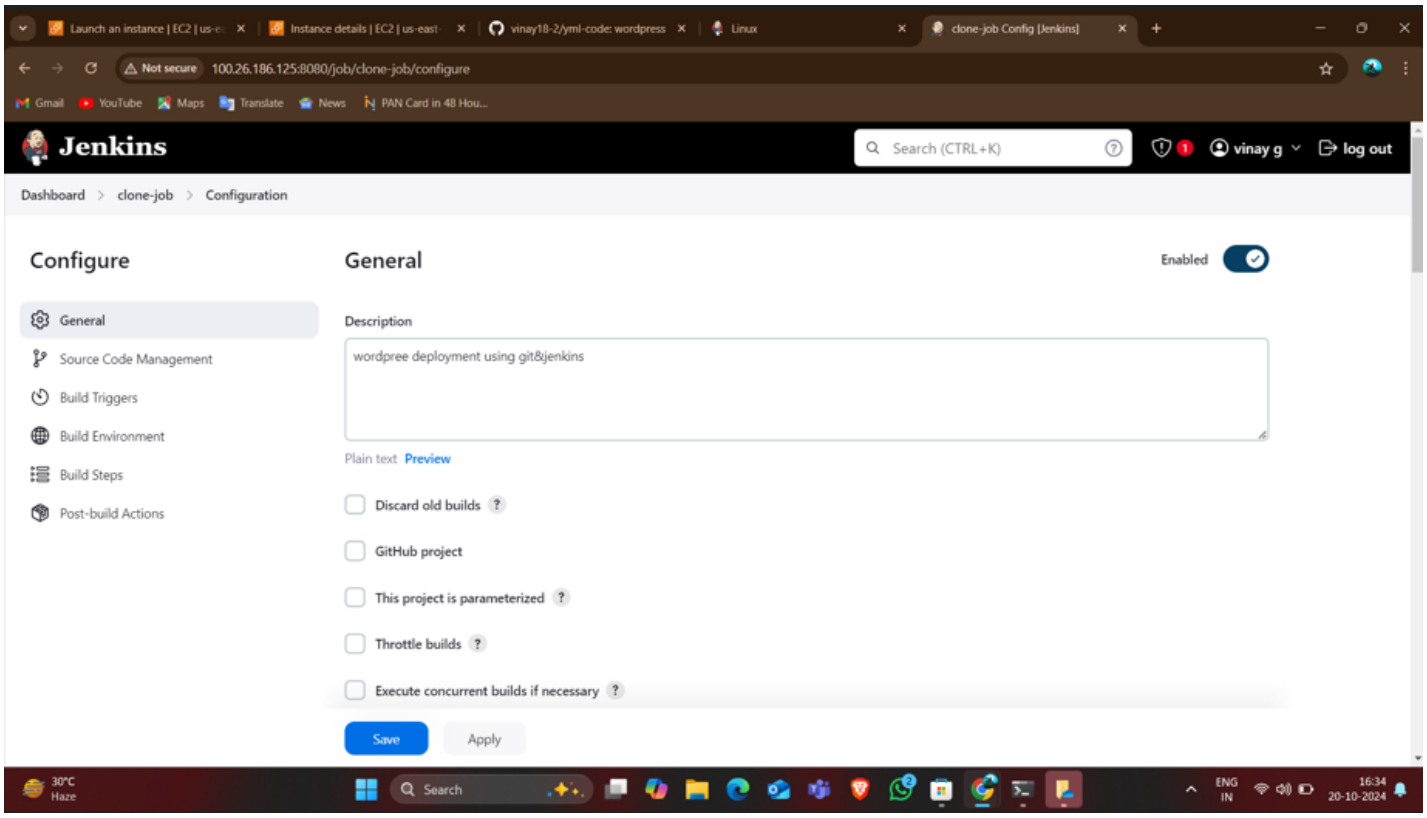
-

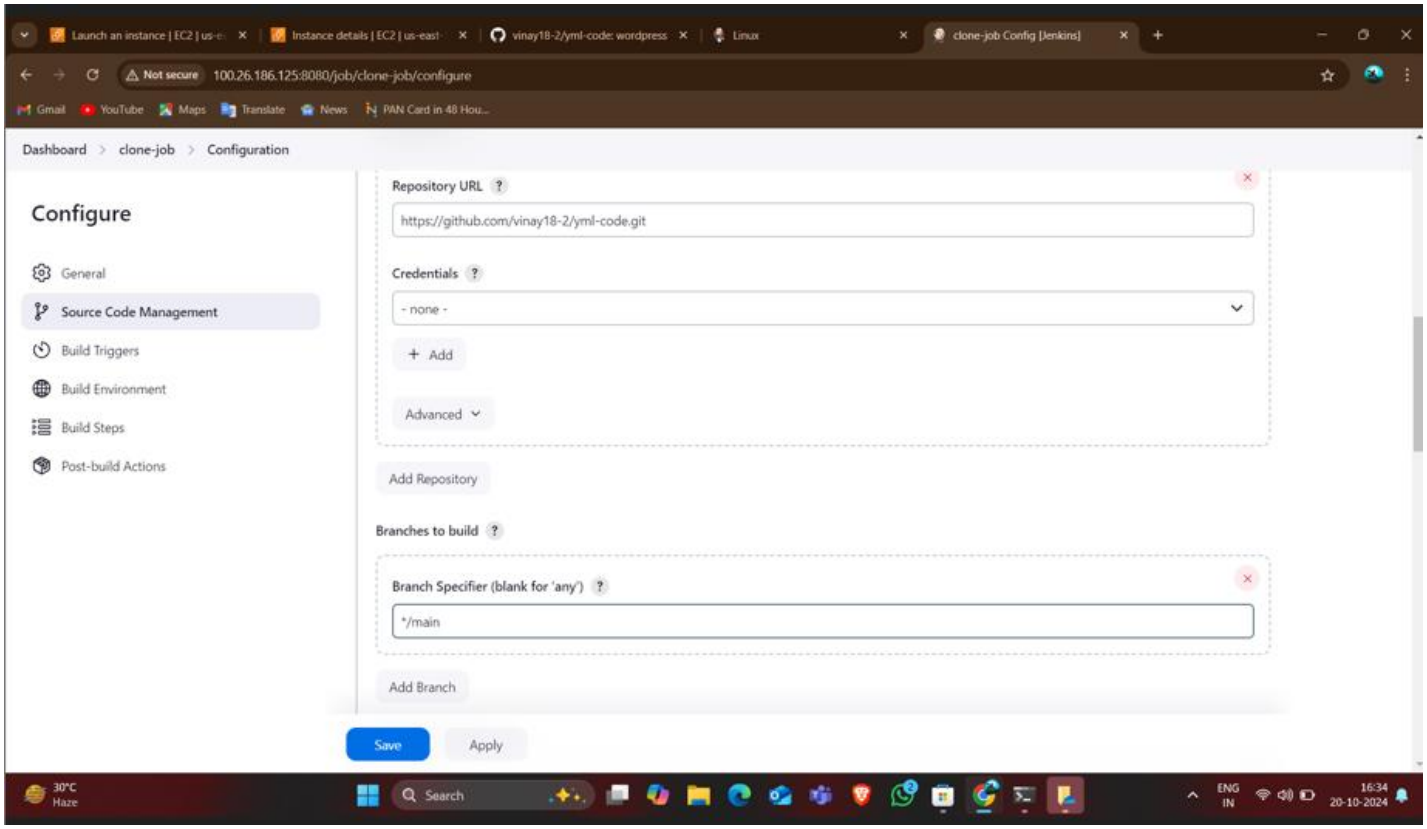
- Than create new item that is clone job with freestyle project and select the source is Git and paste the git repo link and choose the branch.
- Save and apply than build now.
- After that copy the ip address of instace and browse it in google and observe the output.



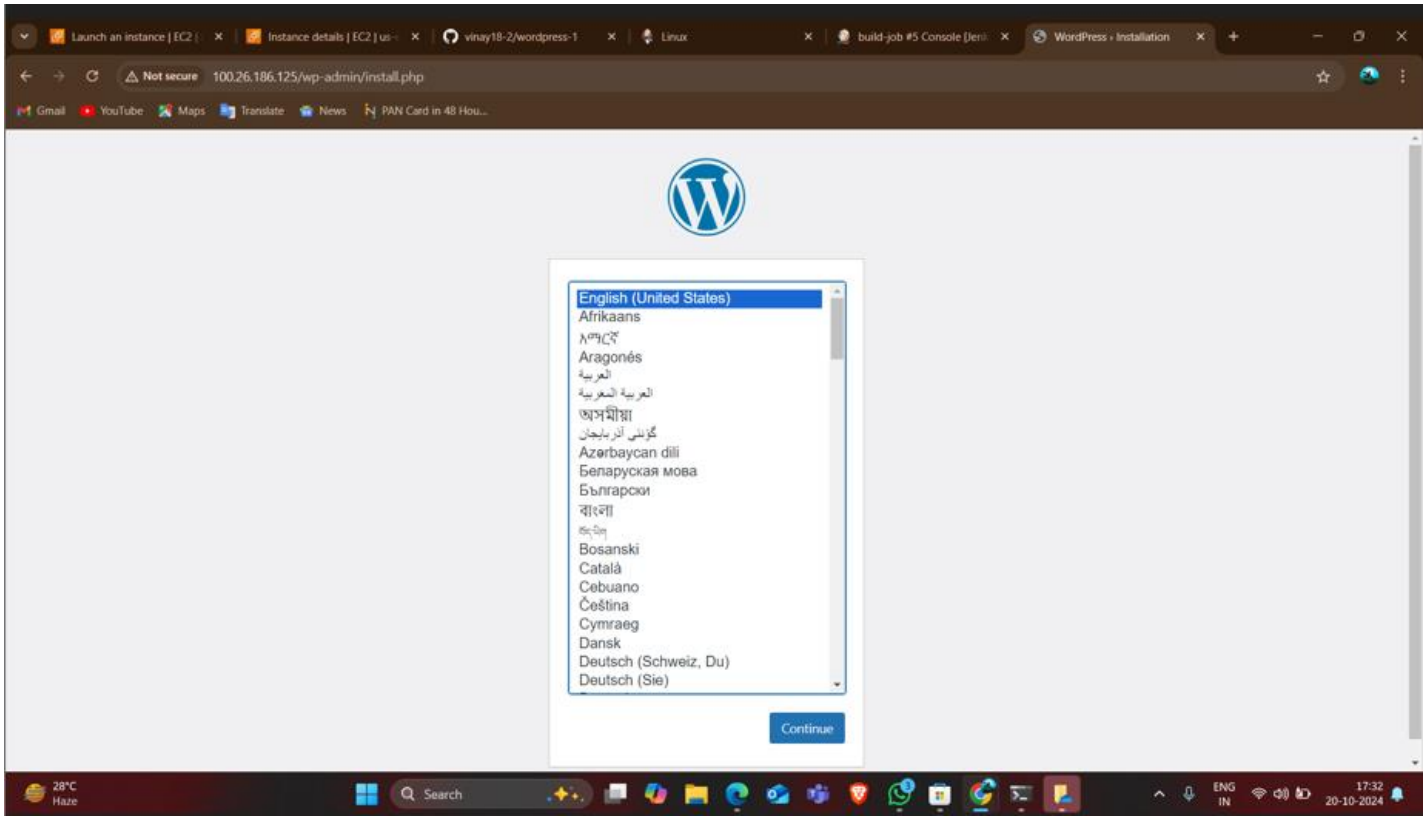


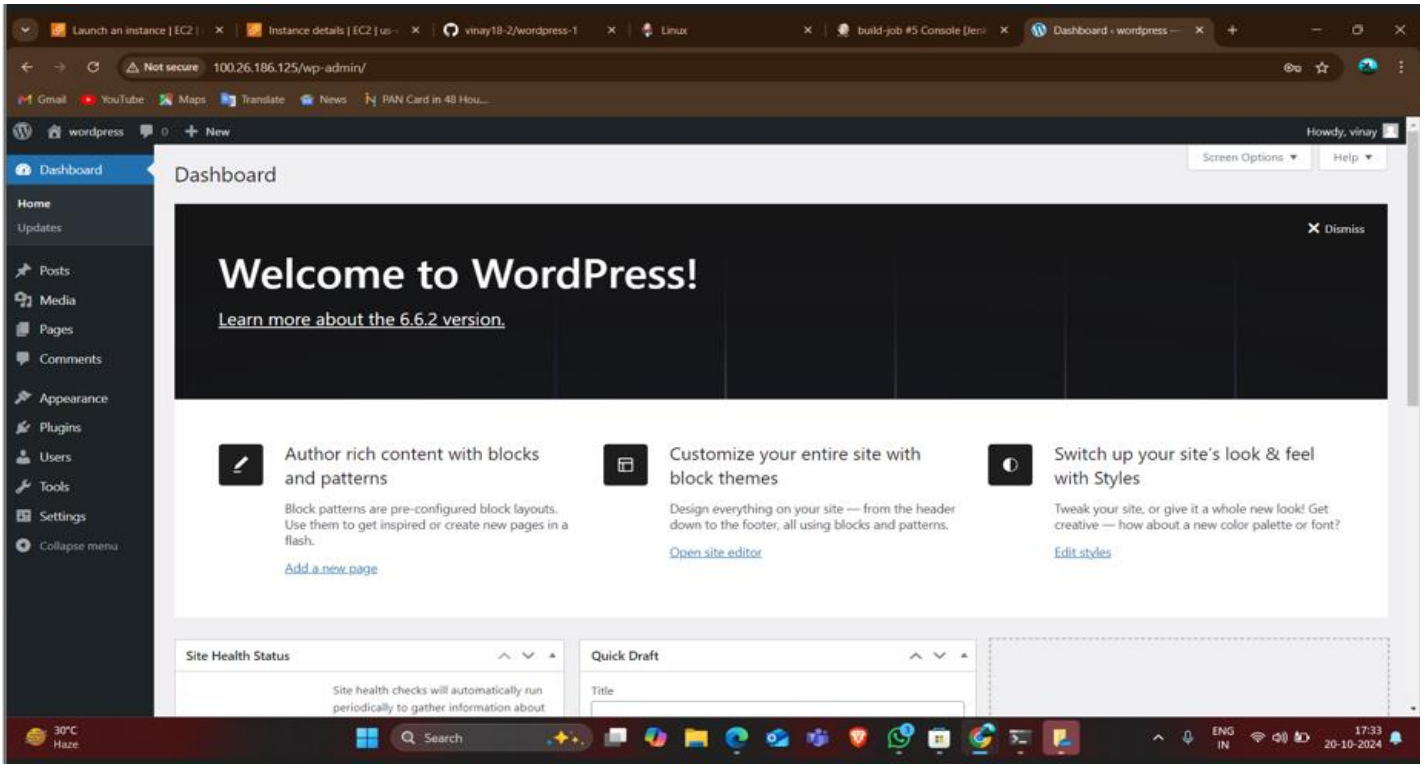
•


Launch an instance | EC2 | us-e
Instance details | EC2 | us-east
vinay18-2/yml-code: wordpress
Linux
clone-job Config [Jenkins]
Not secure
100.26.186.125:8080/job/clone-job/configure
Gmail
YouTube
Maps
Translate
News
PAN Card in 48 Hou...
Jenkins
Search (CTRL+K)
vinay g
log out
Dashboard
clone-job
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
wordpree deployment using git&jenkins
Plain text
Preview
Discard old builds
GitHub project
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Save
Apply
30°C
Haze
Search
ENG
IN
16:34
20-10-2024


Launch an instance | EC2 | us-e
Instance details | EC2 | us-east
vinay18-2/yml-code: wordpress
Linux
clone-job Config [Jenkins]
Not secure
100.26.186.125:8080/job/clone-job/configure
Gmail
YouTube
Maps
Translate
News
PAN Card in 48 Hou...
Dashboard
clone-job
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Repository URL
https://github.com/vinay18-2/yml-code.git
Credentials
- none -
+ Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Add Branch
Save
Apply
30°C
Haze
Search
ENG
IN
16:34
20-10-2024

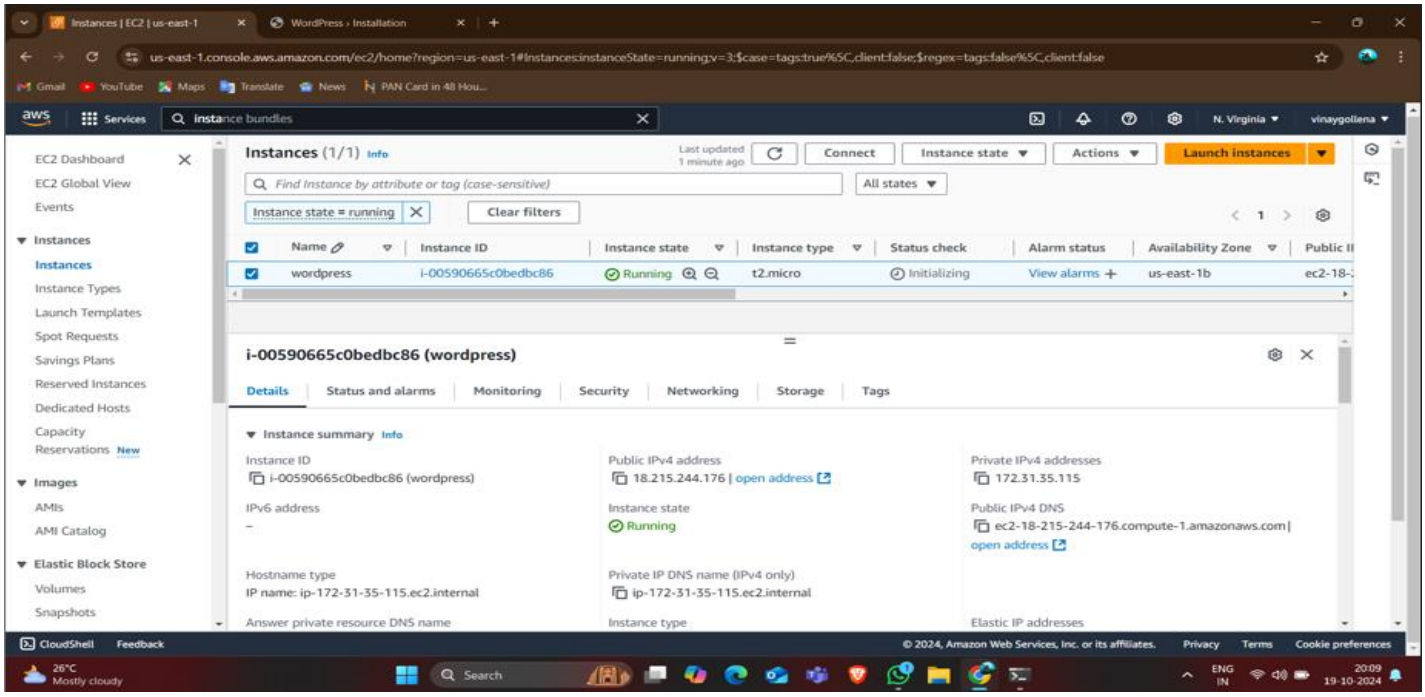
•

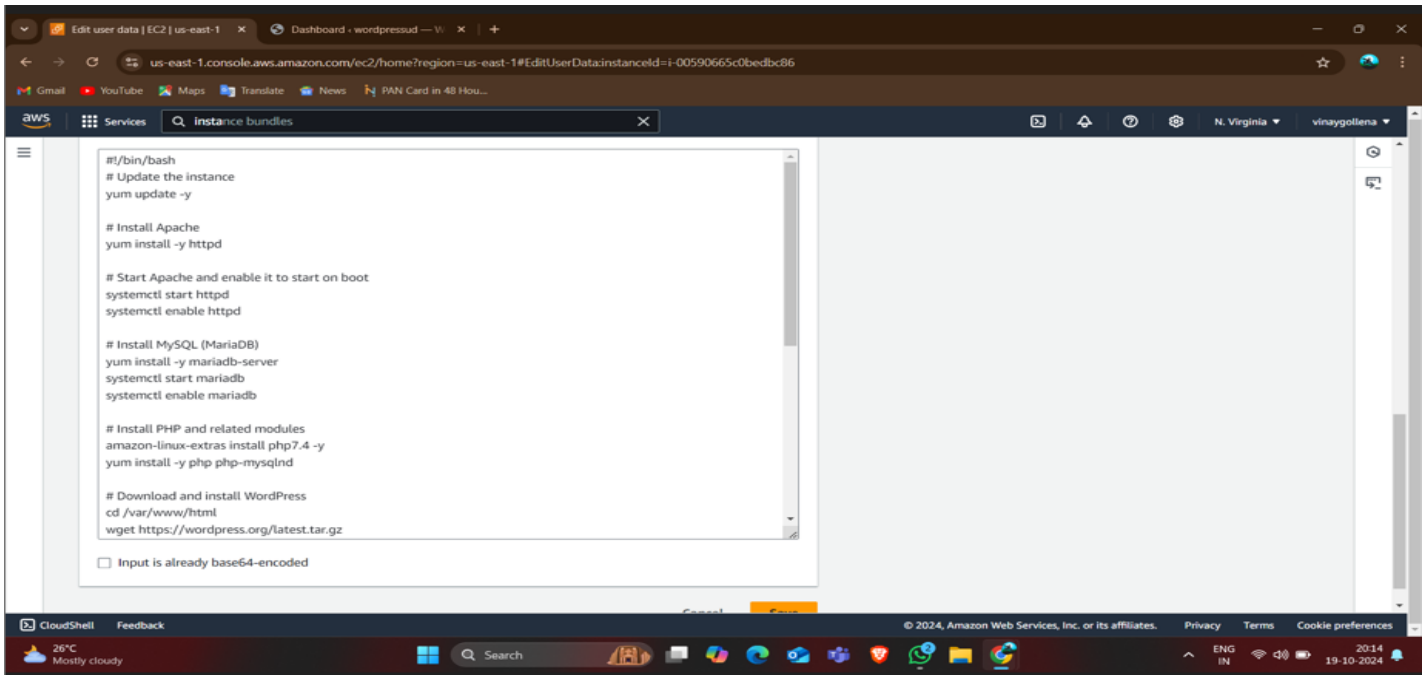

Launch an instance | EC2
Instance details | EC2 | us-
vinay18-2/wordpress-1
Linux
build-job #5 Console [Jen
WordPress › Installation
Not secure
100.26.186.125/wp-admin/install.php
Gmail
YouTube
Maps
Translate
News
PAN Card in 48 Hou...
English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz, Du)
Deutsch (Sie)
Continue
28°C
Haze
Search
ENG
IN
17:32
20-10-2024


Launch an instance | EC2
Instance details | EC2 | us-
vinay18-2/wordpress-1
Linux
build-job #5 Console [Jen
Dashboard ‹ wordpress —
Not secure
100.26.186.125/wp-admin/
Gmail
YouTube
Maps
Translate
News
PAN Card in 48 Hou...
wordpress
0
New
Howdy, vinay
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Dashboard
Screen Options
Help
Dismiss
Welcome to WordPress!
Learn more about the 6.6.2 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
Site health checks will automatically run periodically to gather information about
Quick Draft
Title
30°C
Haze
Search
ENG
IN
17:33
20-10-2024

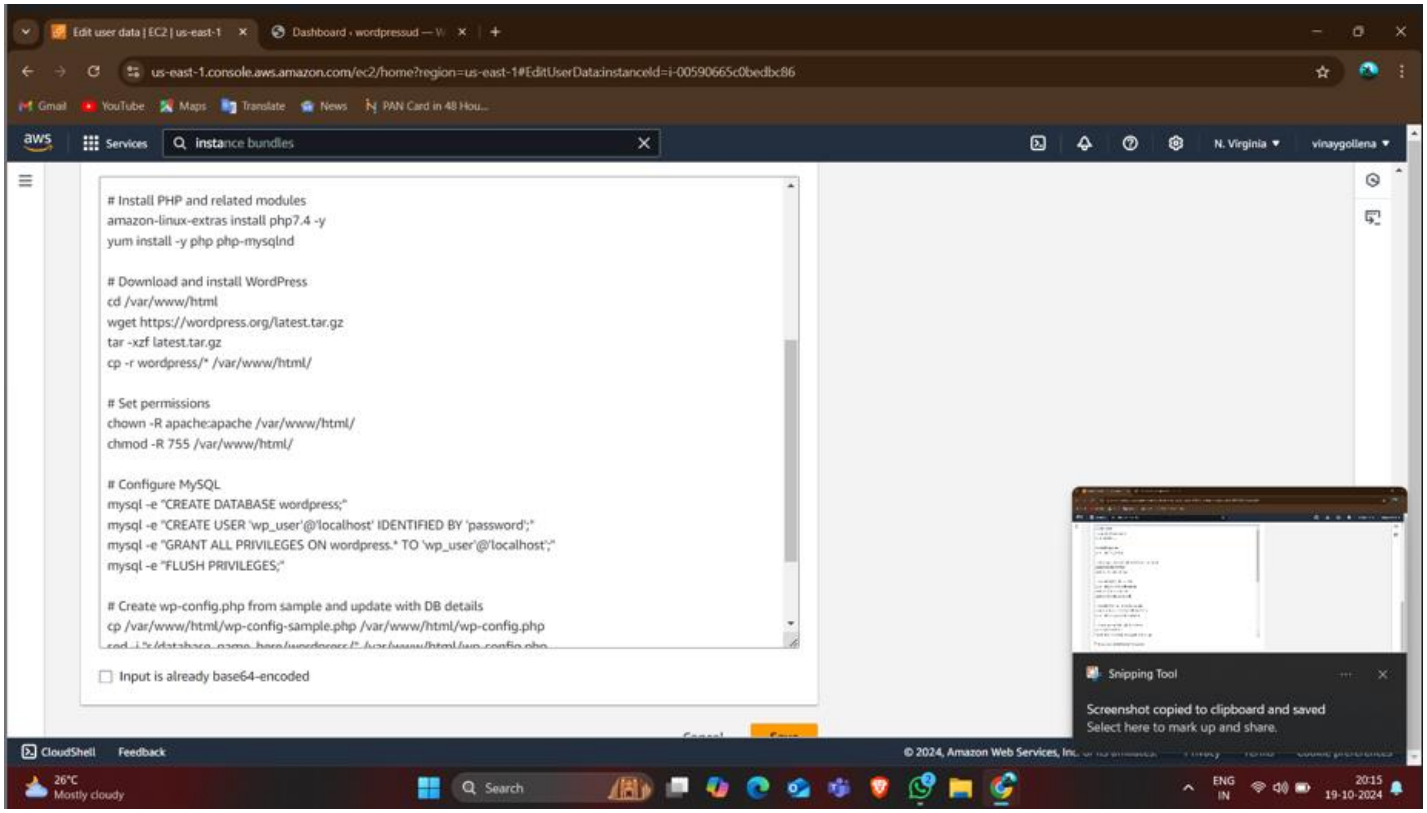
Deploy WordPress web application by using userdata of EC2 instance

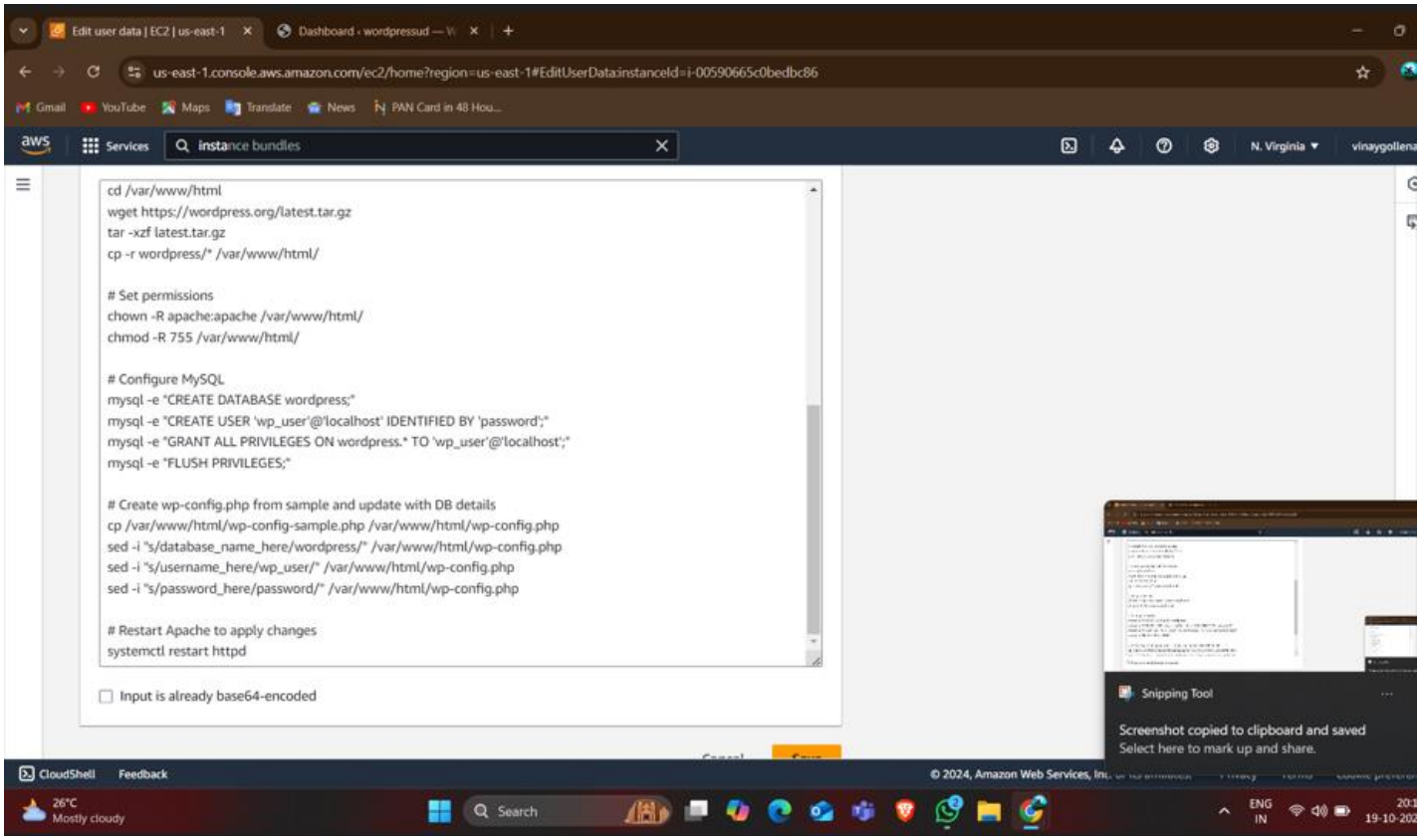
- Create an EC2 instance with t3.micro,amazon-linux 2.
- Select the security groups HTTP,HTTPS and SSH with port numbers 80,443&22.
- While creating the instances we should write the bash script to install MYSQL and HTTPD and wordpress in the additional settings and save it.



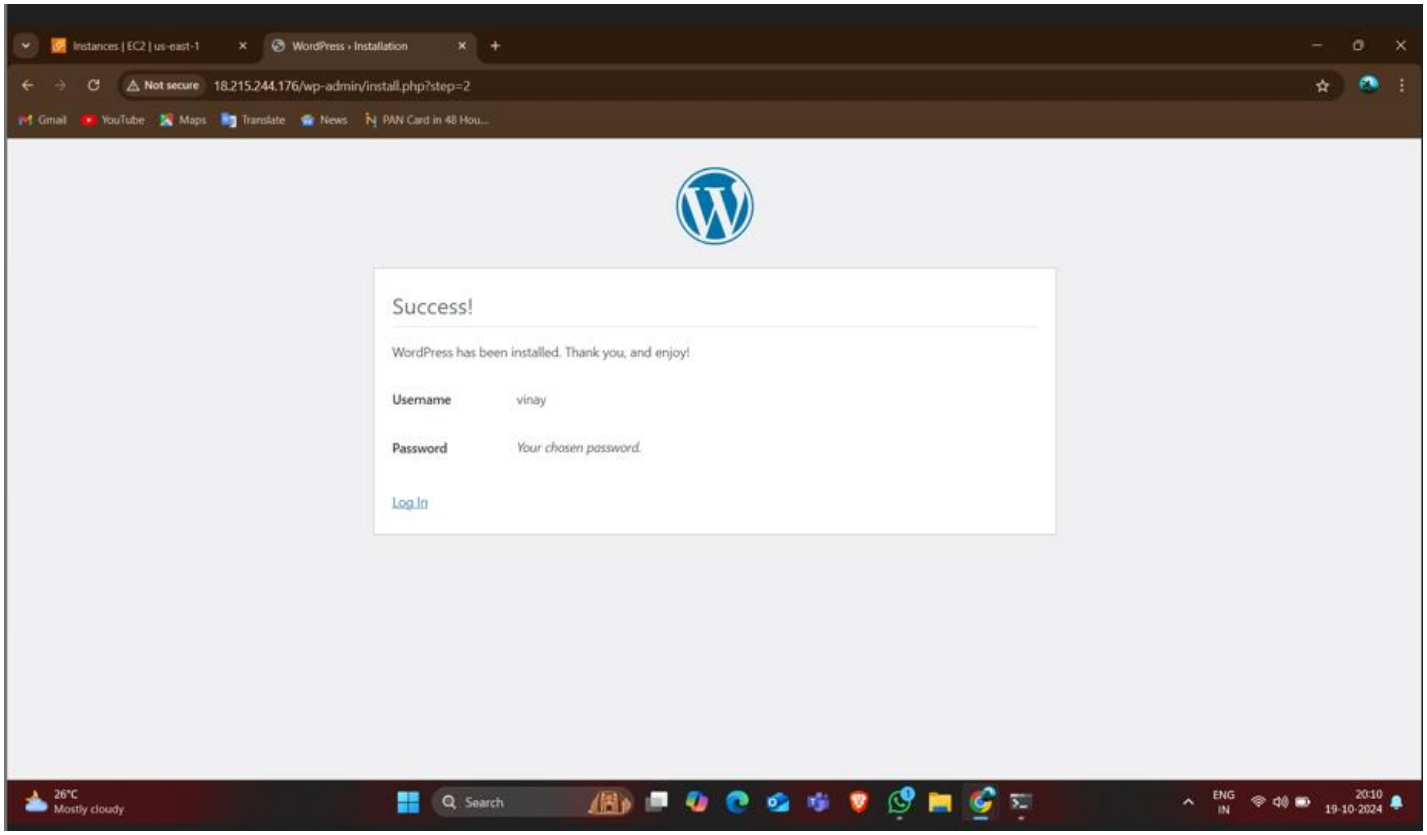


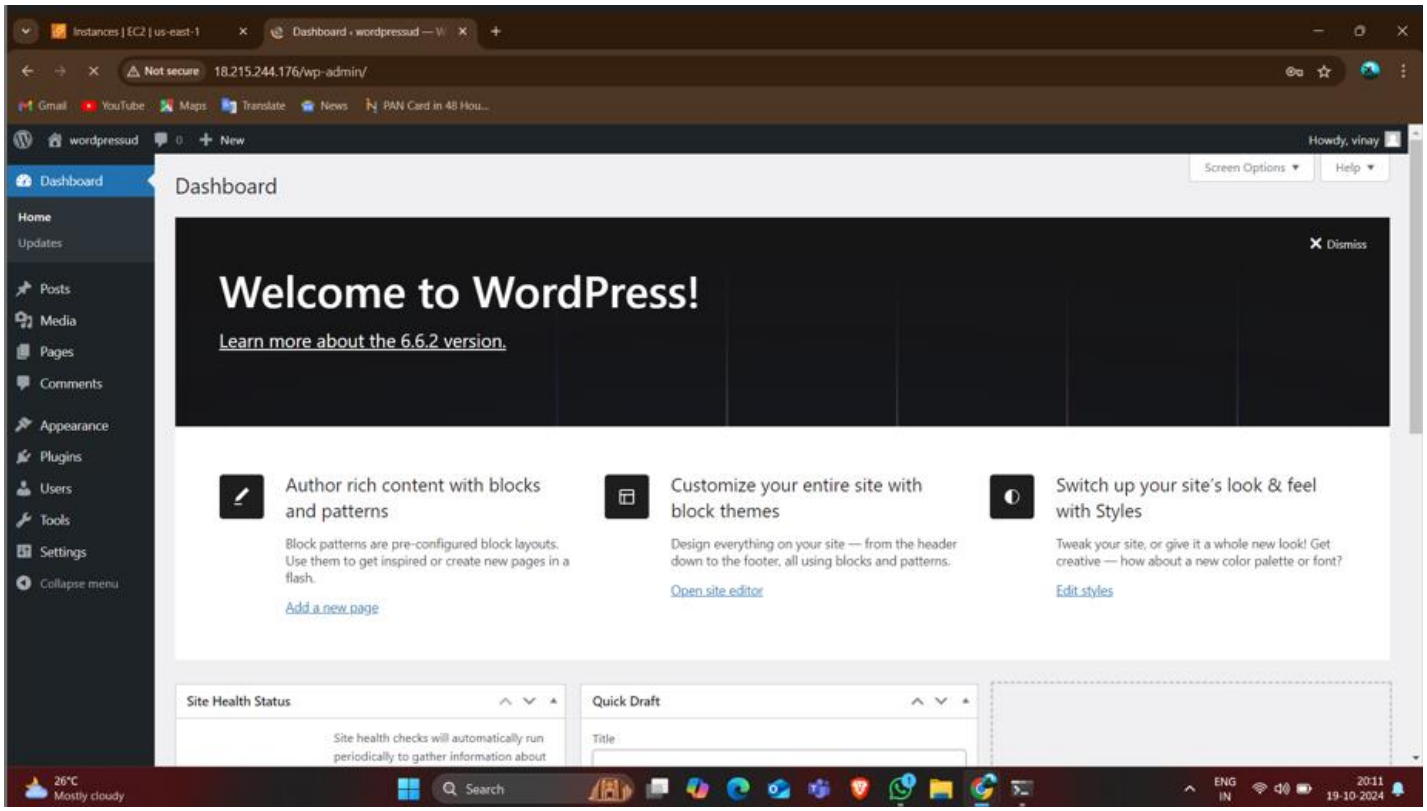
•


Edit user data | EC2 | us-east-1
Dashboard ‹ wordpressud
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#EditUserData:instanceId=i-00590665c0bedbc86
Gmail YouTube Maps Translate News PAN Card in 48 Hou...
aws Services instance bundles N. Virginia vinaygollena
Install PHP and related modules
amazon-linux-extras install php7.4 -y
yum install -y php php-mysqlnd
Download and install WordPress
cd /var/www/html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
Set permissions
chown -R apache:apache /var/www/html/
chmod -R 755 /var/www/html/
Configure MySQL
mysql -e "CREATE DATABASE wordpress;"
mysql -e "CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';"
mysql -e "GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';"
mysql -e "FLUSH PRIVILEGES;"
Create wp-config.php from sample and update with DB details
cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
Input is already base64-encoded
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.
CloudShell Feedback
© 2024, Amazon Web Services, Inc.
26°C Mostly cloudy
Search
ENG IN
20:15 19-10-2024


Edit user data | EC2 | us-east-1
Dashboard ‹ wordpressud
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#EditUserData:instanceId=i-00590665c0bedbc86
Gmail YouTube Maps Translate News PAN Card in 48 Hou...
aws Services instance bundles N. Virginia vinaygollena
cd /var/www/html
wget https://wordpress.org/latest.tar.gz
tar -xzf latest.tar.gz
cp -r wordpress/* /var/www/html/
Set permissions
chown -R apache:apache /var/www/html/
chmod -R 755 /var/www/html/
Configure MySQL
mysql -e "CREATE DATABASE wordpress;"
mysql -e "CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';"
mysql -e "GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'localhost';"
mysql -e "FLUSH PRIVILEGES;"
Create wp-config.php from sample and update with DB details
cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
sed -i "s/database_name_here/wordpress/" /var/www/html/wp-config.php
sed -i "s/username_here/wp_user/" /var/www/html/wp-config.php
sed -i "s/password_here/password/" /var/www/html/wp-config.php
Restart Apache to apply changes
systemctl restart httpd
Input is already base64-encoded
Snipping Tool
Screenshot copied to clipboard and saved
Select here to mark up and share.
CloudShell Feedback
© 2024, Amazon Web Services, Inc.
26°C Mostly cloudy
Search
ENG IN

•

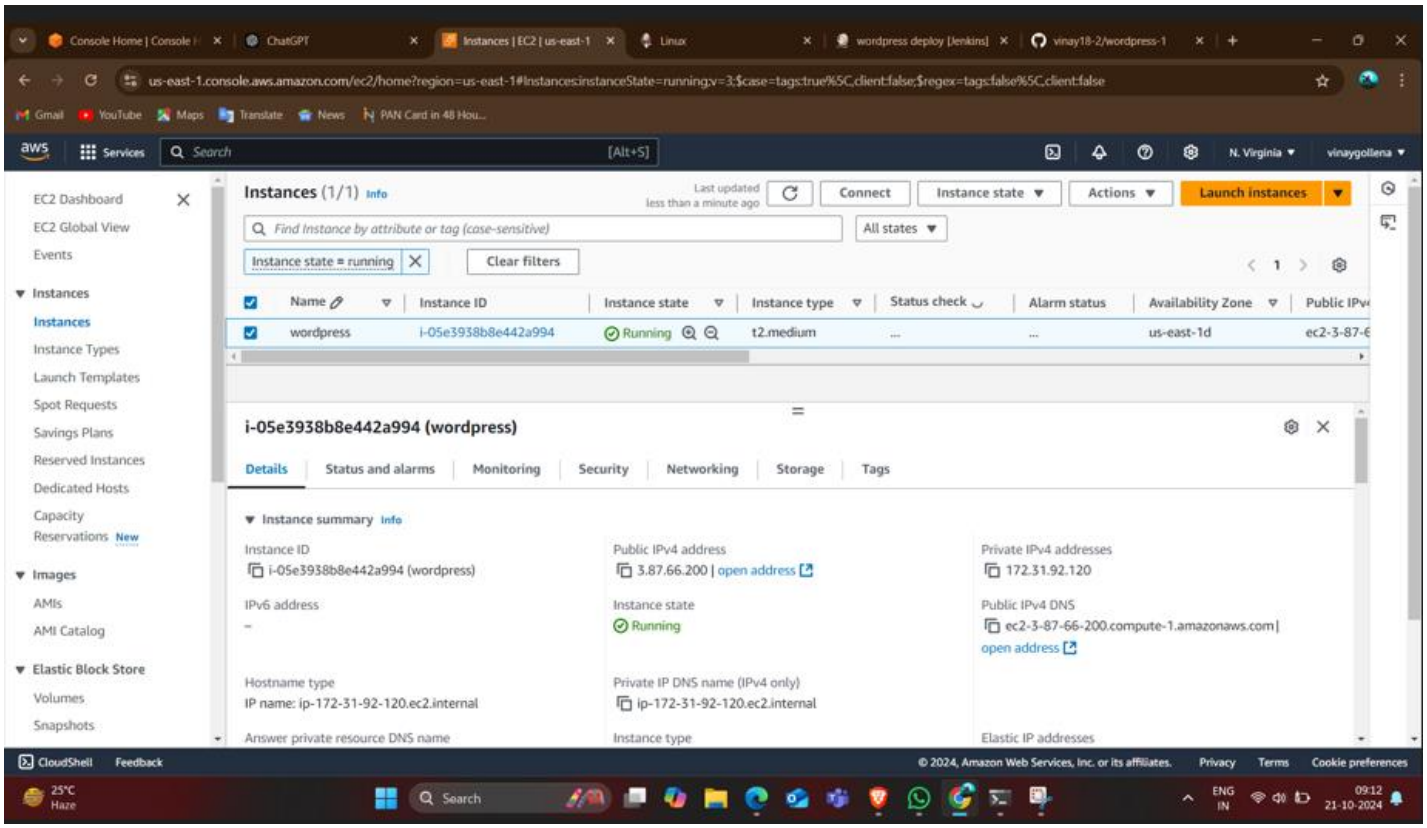

Instances | EC2 | us-east-1
WordPress › Installation
Not secure 18.215.244.176/wp-admin/install.php?step=2
Gmail YouTube Maps Translate News PAN Card in 48 Hou...
Success!
WordPress has been installed. Thank you, and enjoy!
Username vinay
Password Your chosen password.
Log In
26°C Mostly cloudy
Search
ENG IN
20:10 19-10-2024


Instances | EC2 | us-east-1
Dashboard ‹ wordpressud — W
Not secure 18.215.244.176/wp-admin/
Gmail YouTube Maps Translate News PAN Card in 48 Hou...
wordpressud 0 New
Howdy, vinay
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Dashboard
Screen Options
Help
Dismiss
Welcome to WordPress!
Learn more about the 6.6.2 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
Site health checks will automatically run periodically to gather information about
Quick Draft
Title
26°C Mostly cloudy
Search
ENG IN
20:11 19-10-2024

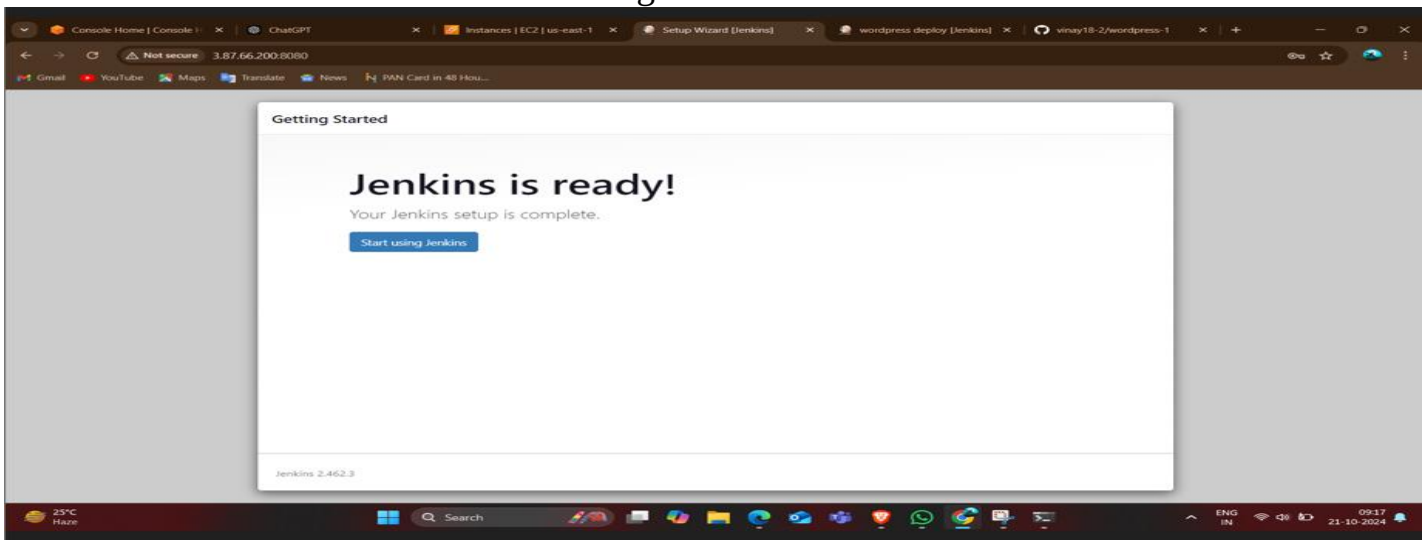
•

Deploy WordPress web application by using git and jenkins execute shell (bash script)

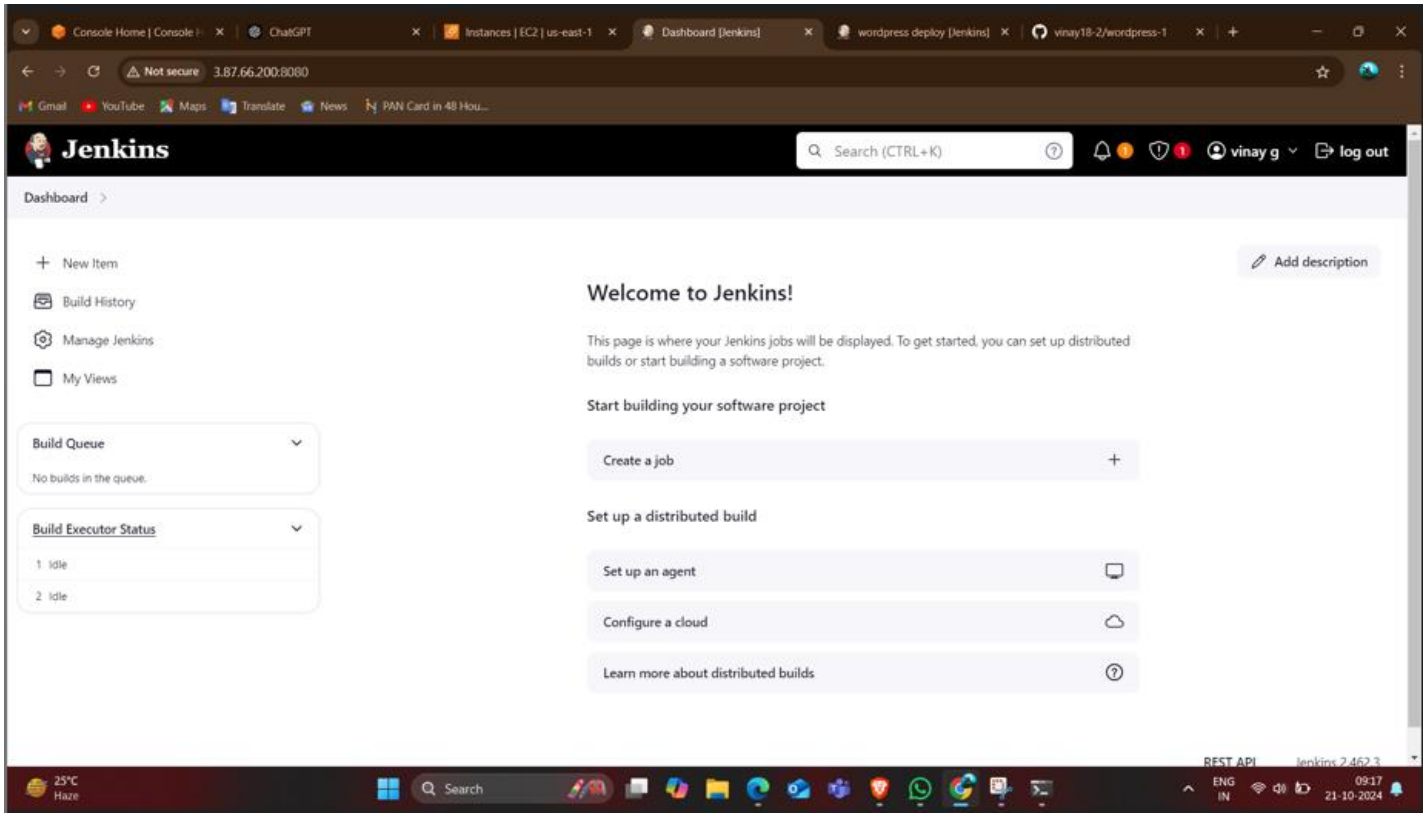
- Launch the instance with t2.medium by allowing the security groups SSH,HTTP and All traffic , or provide the port 8080 which runs the Jenkins.

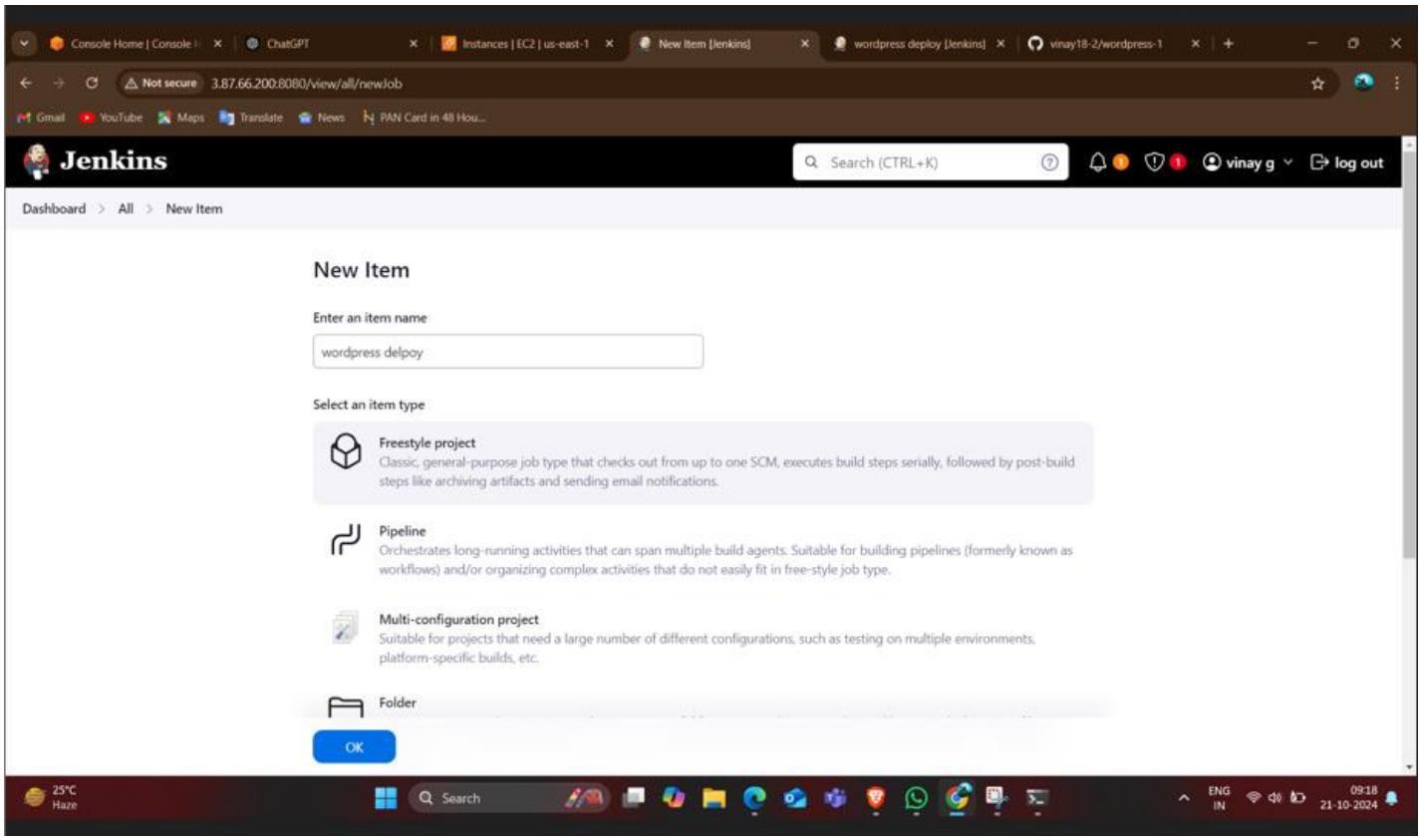


- Connect to the terminal and install the Jenkins as same as the 3rd method.
- Start and enable the Jenkins and log in to the Jenkins.

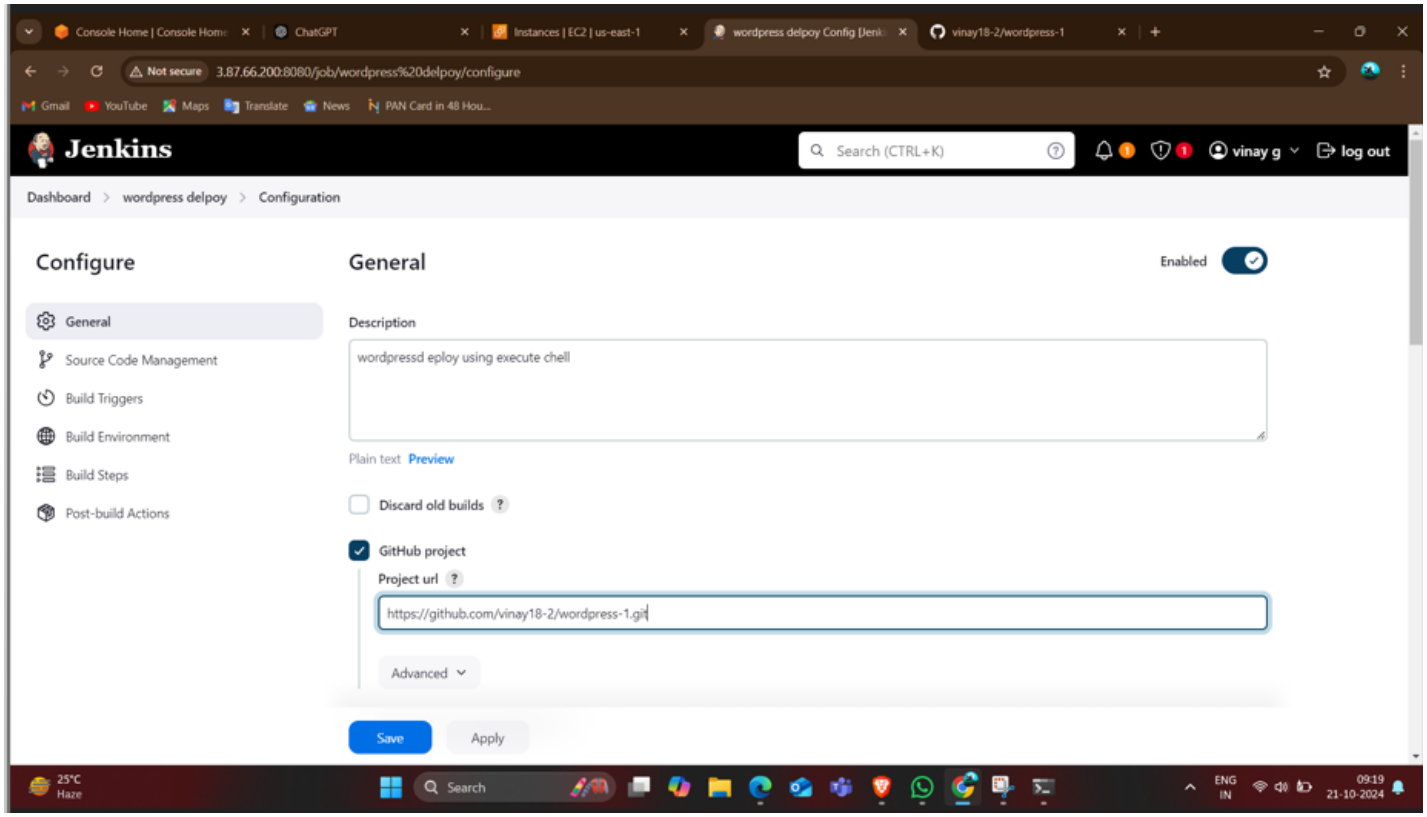


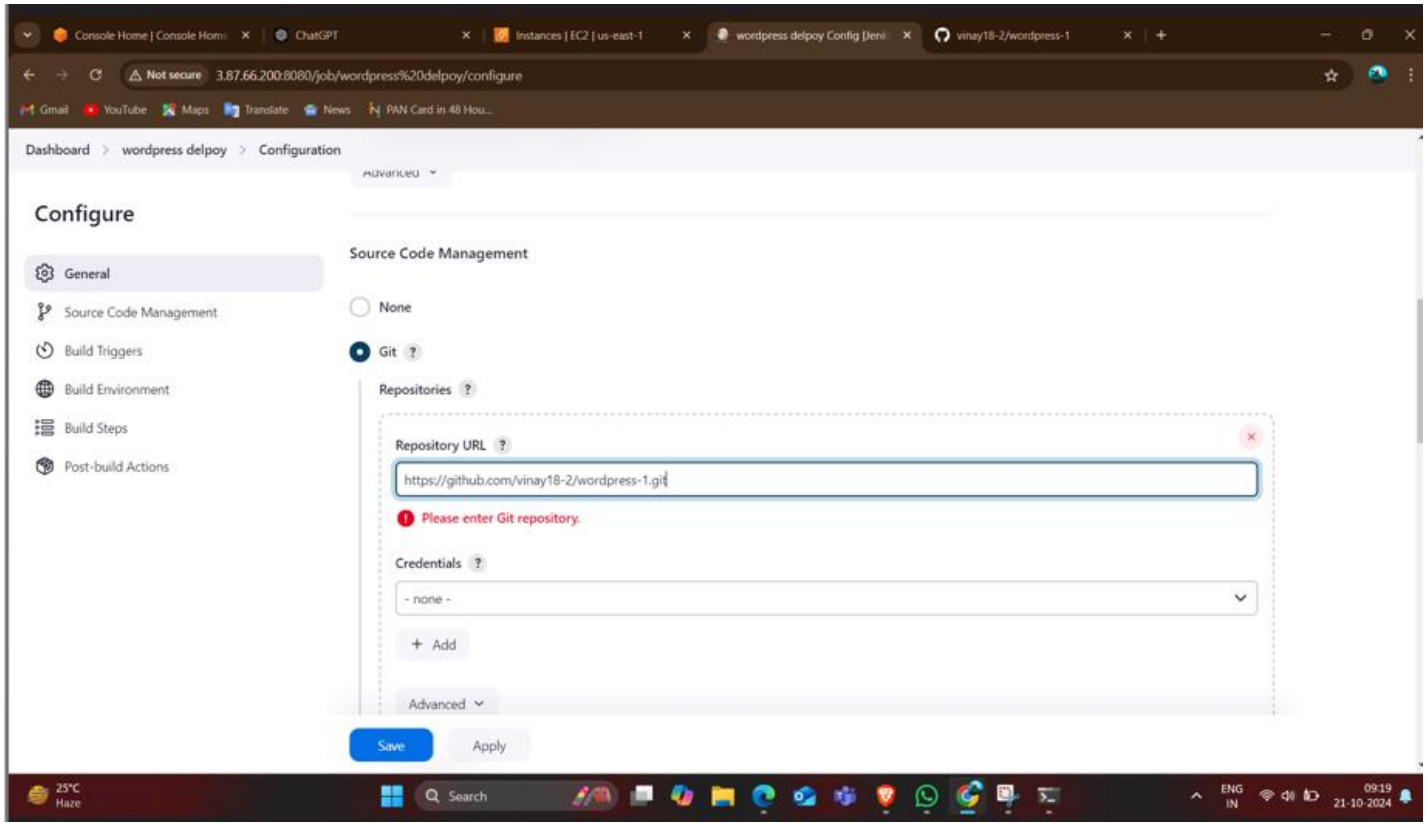
-
    - Than create the new job and select the freestyle project.



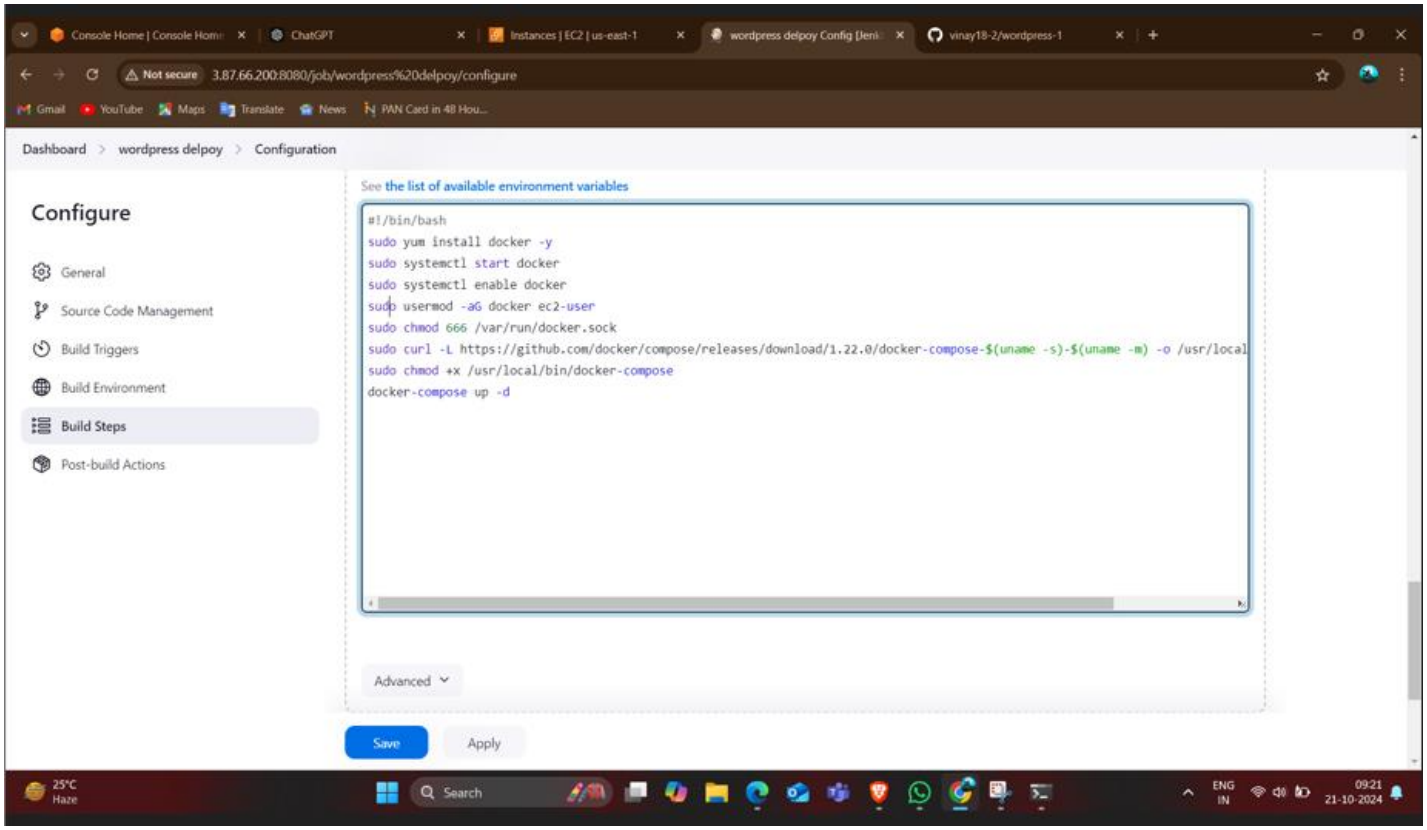


-
    - Provide the description and select the git in source code management change the branch.

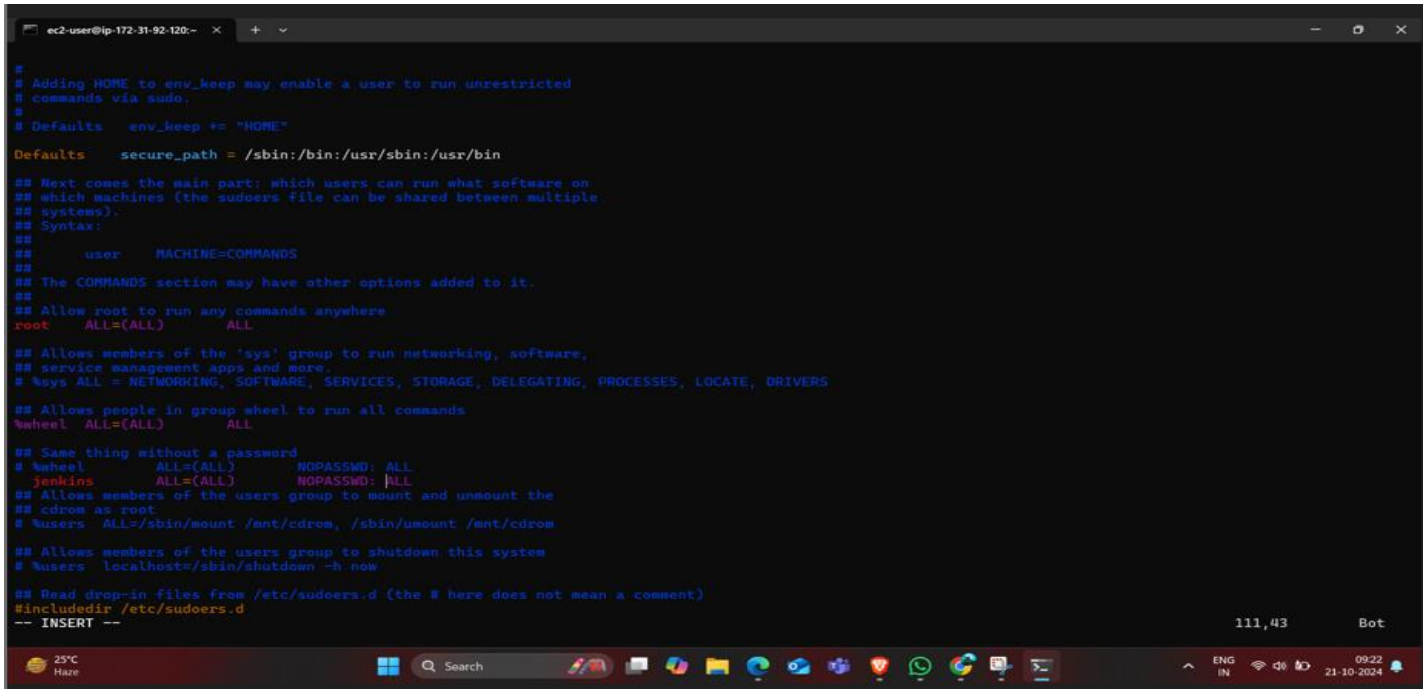




- Write the code in the excute shell and save it.

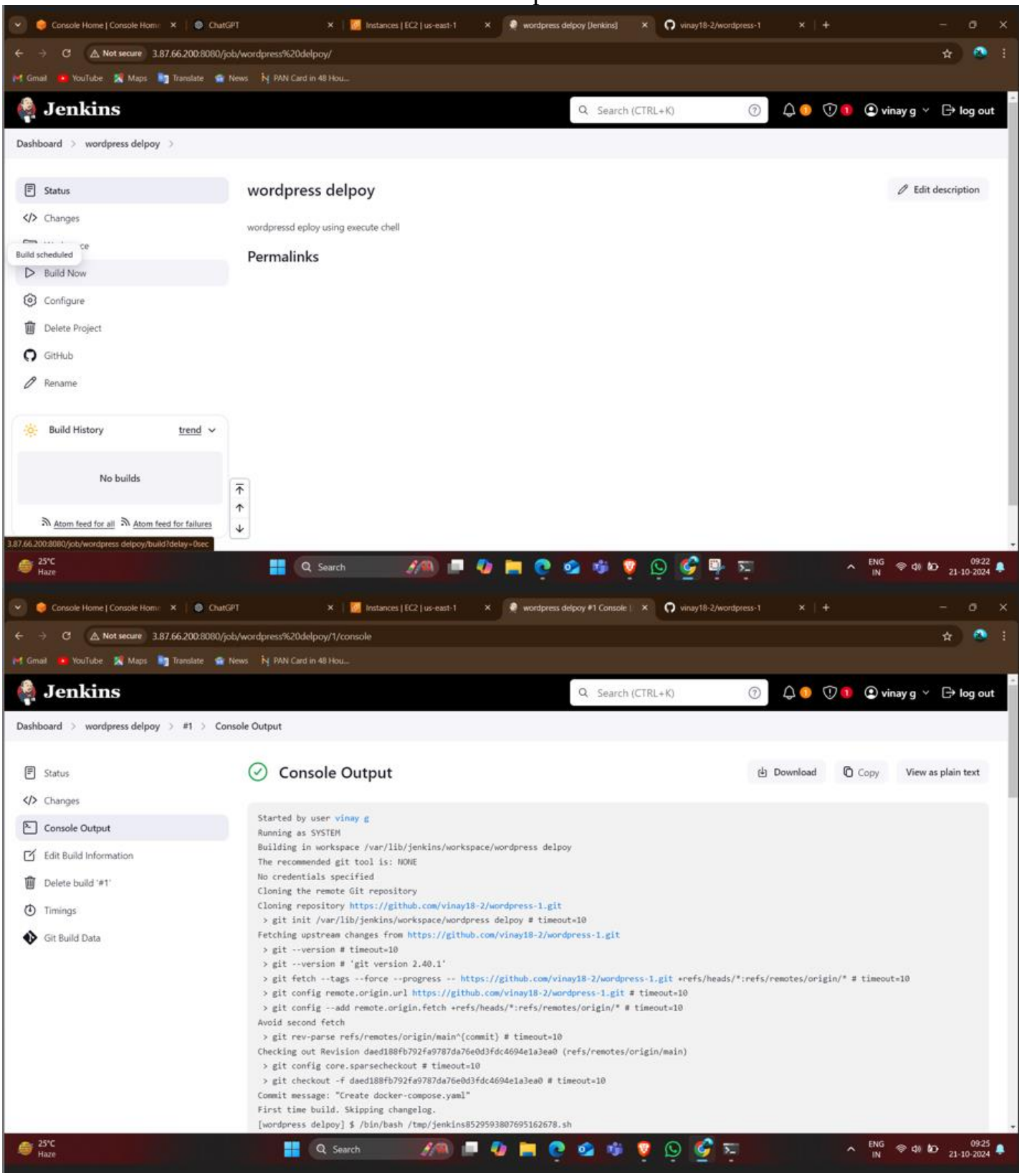


- We have to made some changes in visudo file by using “sudo visudo” Jenkins ALL=(ALL) NOPASSWD:ALL”



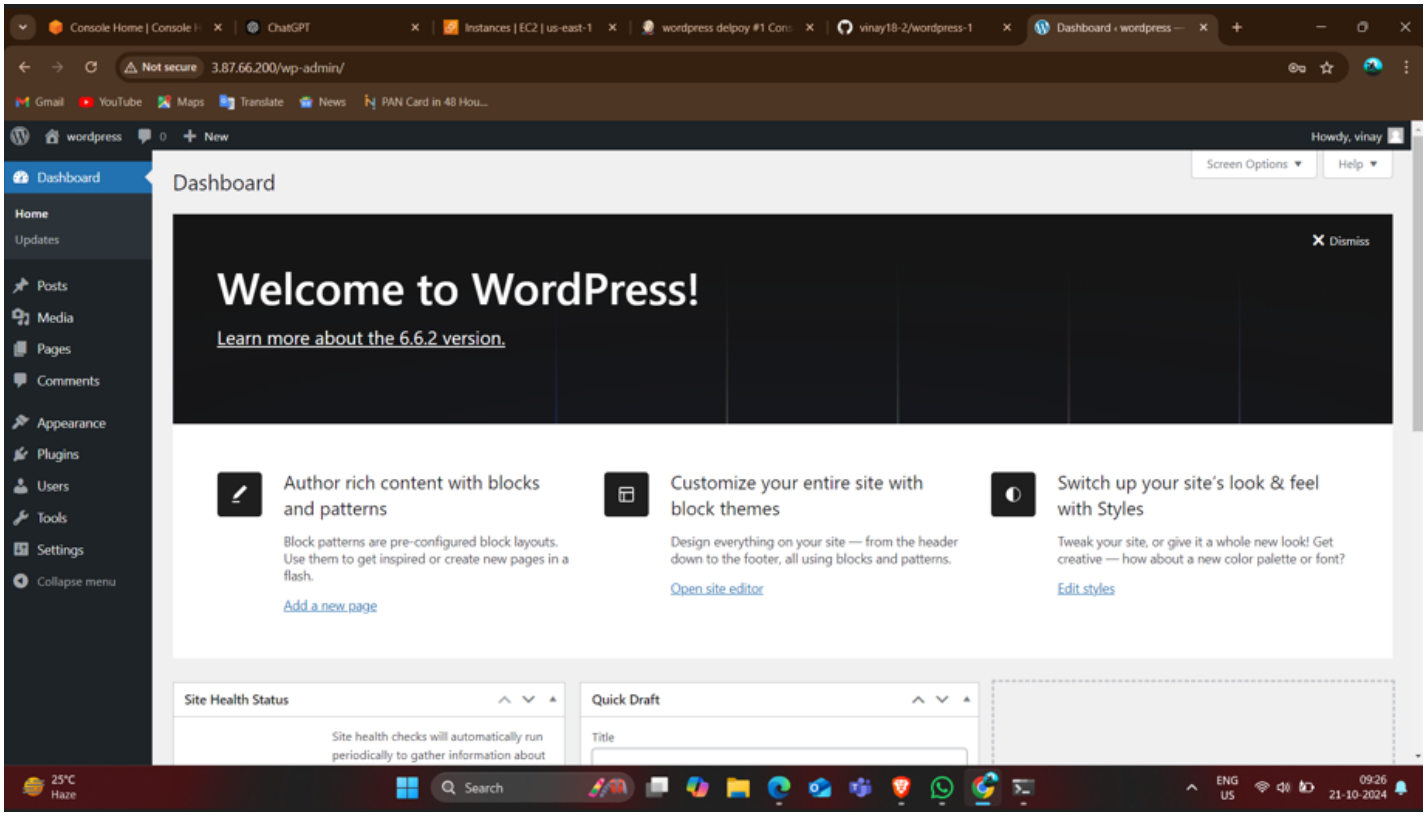
-

- Than build now and observe the consol output.



- Than copy the ip address and paste it in google with port 80 and browse it.

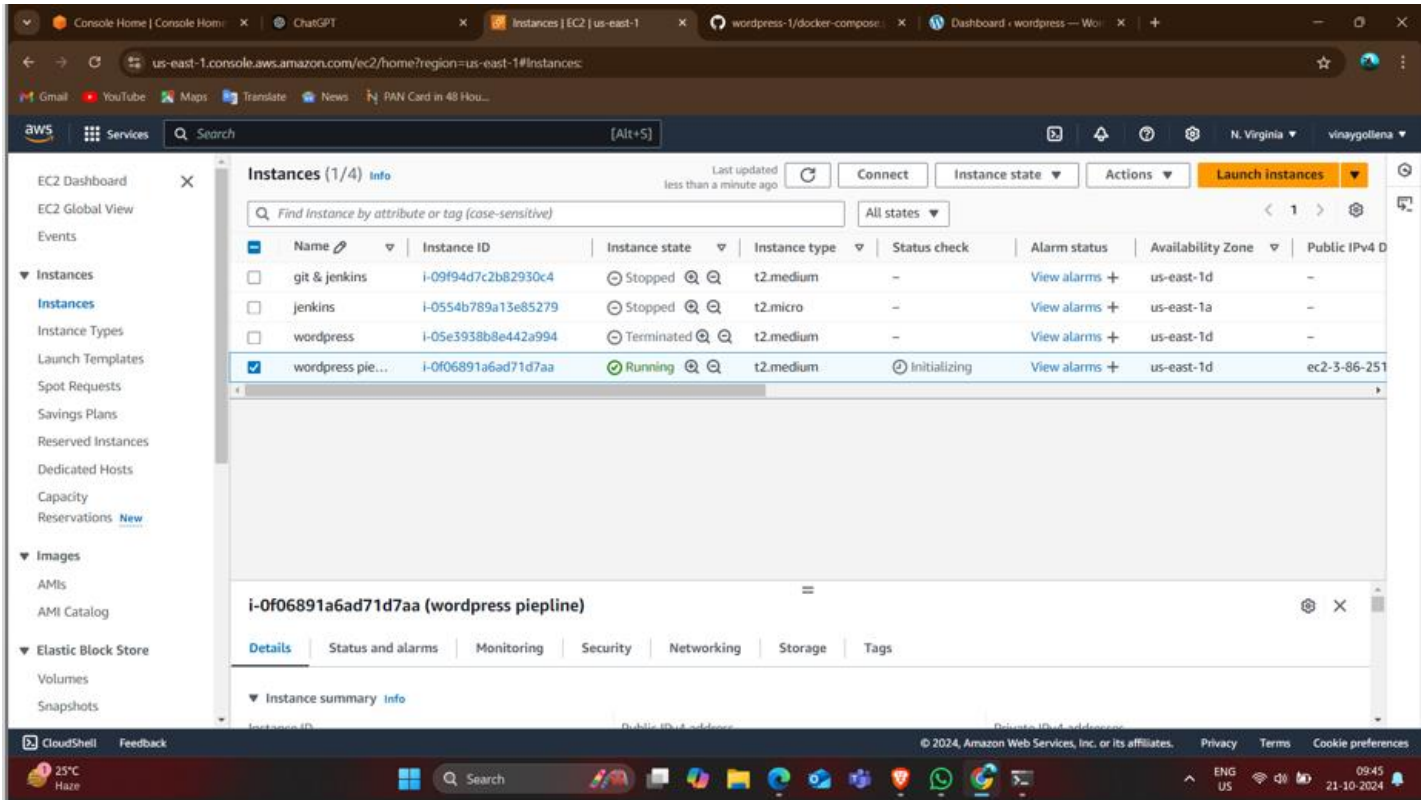
•


Not secure 3.87.66.200/wp-admin/
wordpress
New
Howdy, vinay
Dashboard
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Screen Options
Help
Dismiss
Welcome to WordPress!
Learn more about the 6.6.2 version.
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
Site health checks will automatically run periodically to gather information about
Quick Draft
Title
25°C Haze
Search
ENG US
09:26 21-10-2024

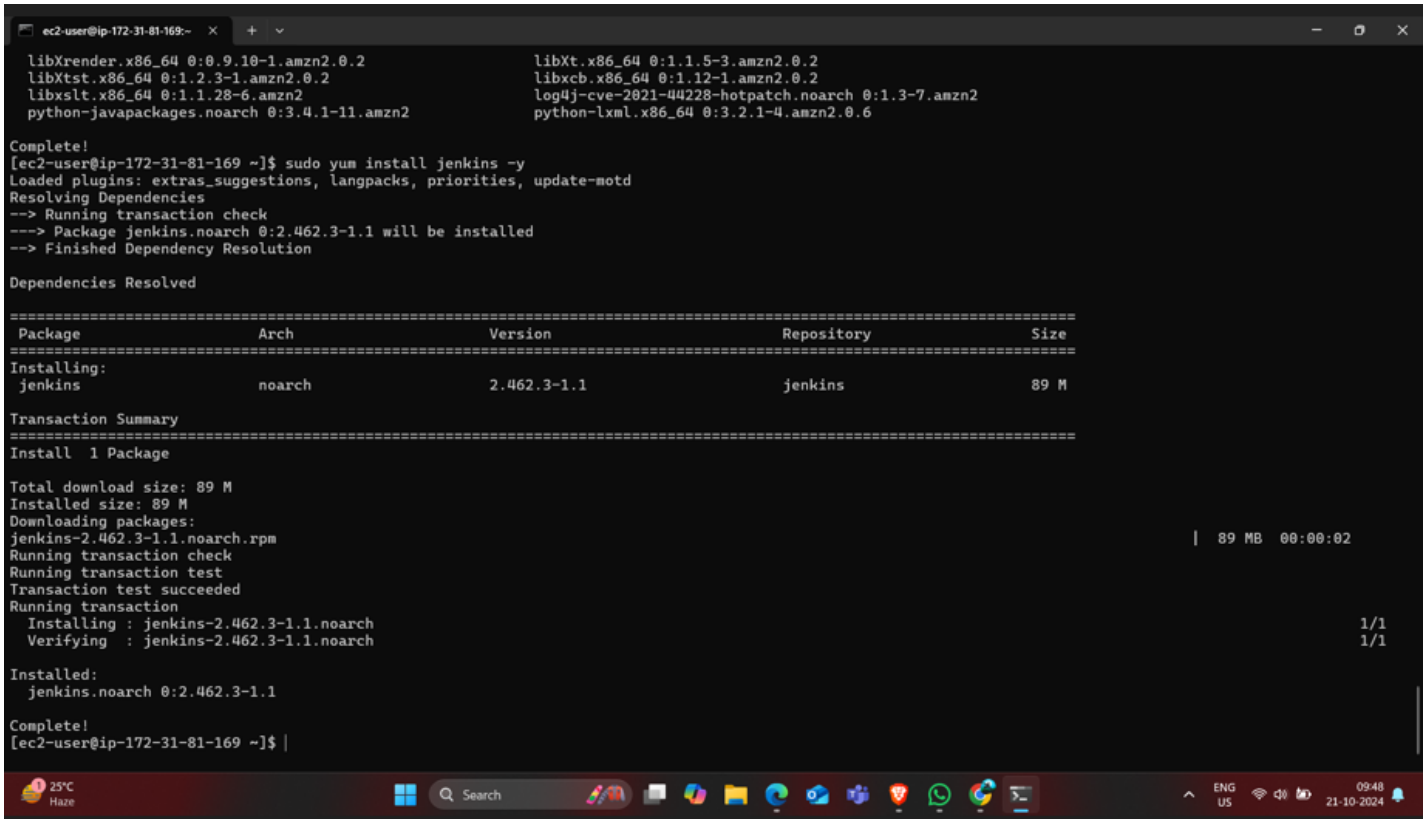
•

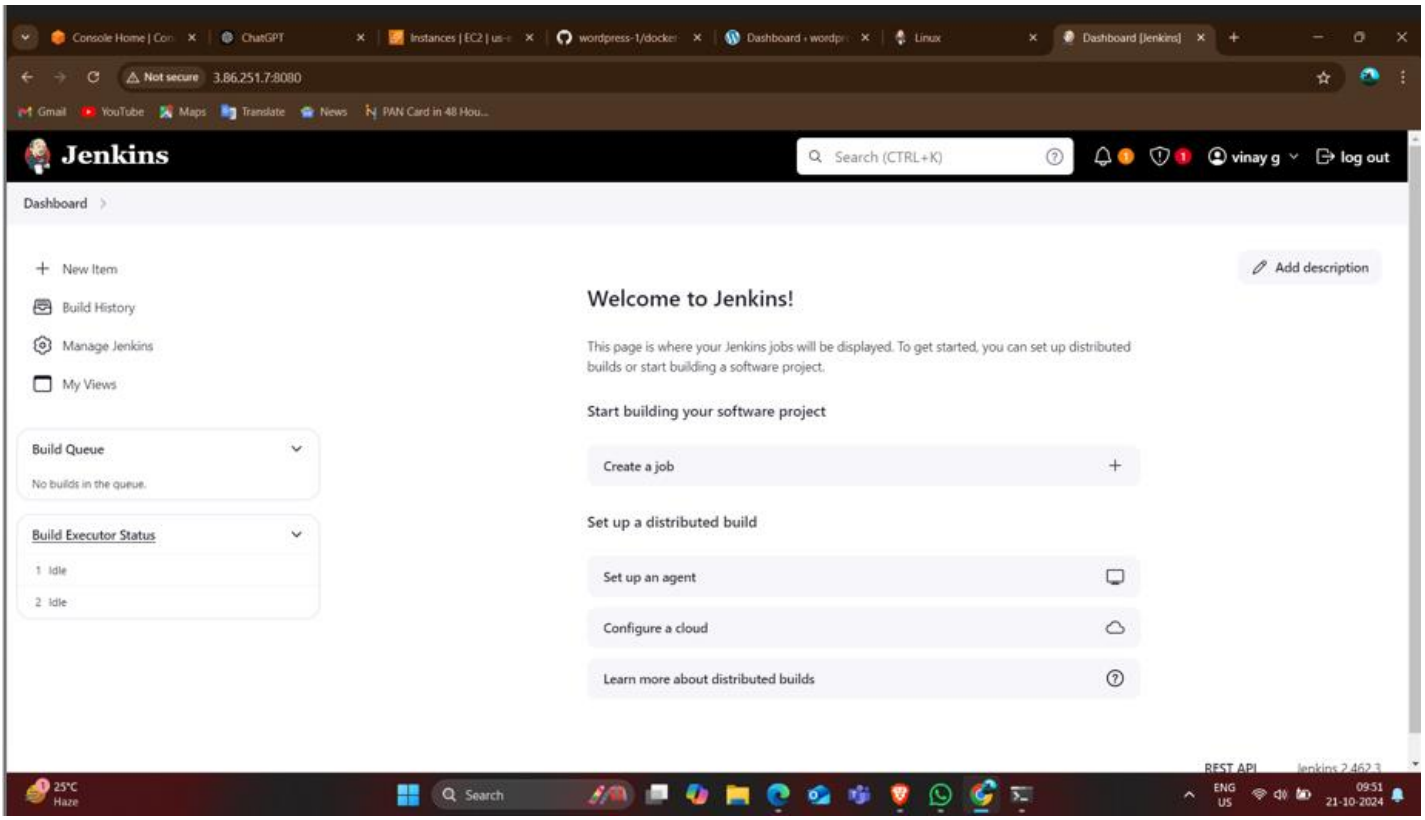
Deploy WordPress web application by using git and jenkins execute shell (bash script) create jenkins pipeline add build periodically and poll scm to initial job of pipeline and check the changes happened or not which are made in github repo

- Do the same as the method 6 or above method.
- Connect to the terminal and install Jenkins and git.
- Than start the Jenkins and login.

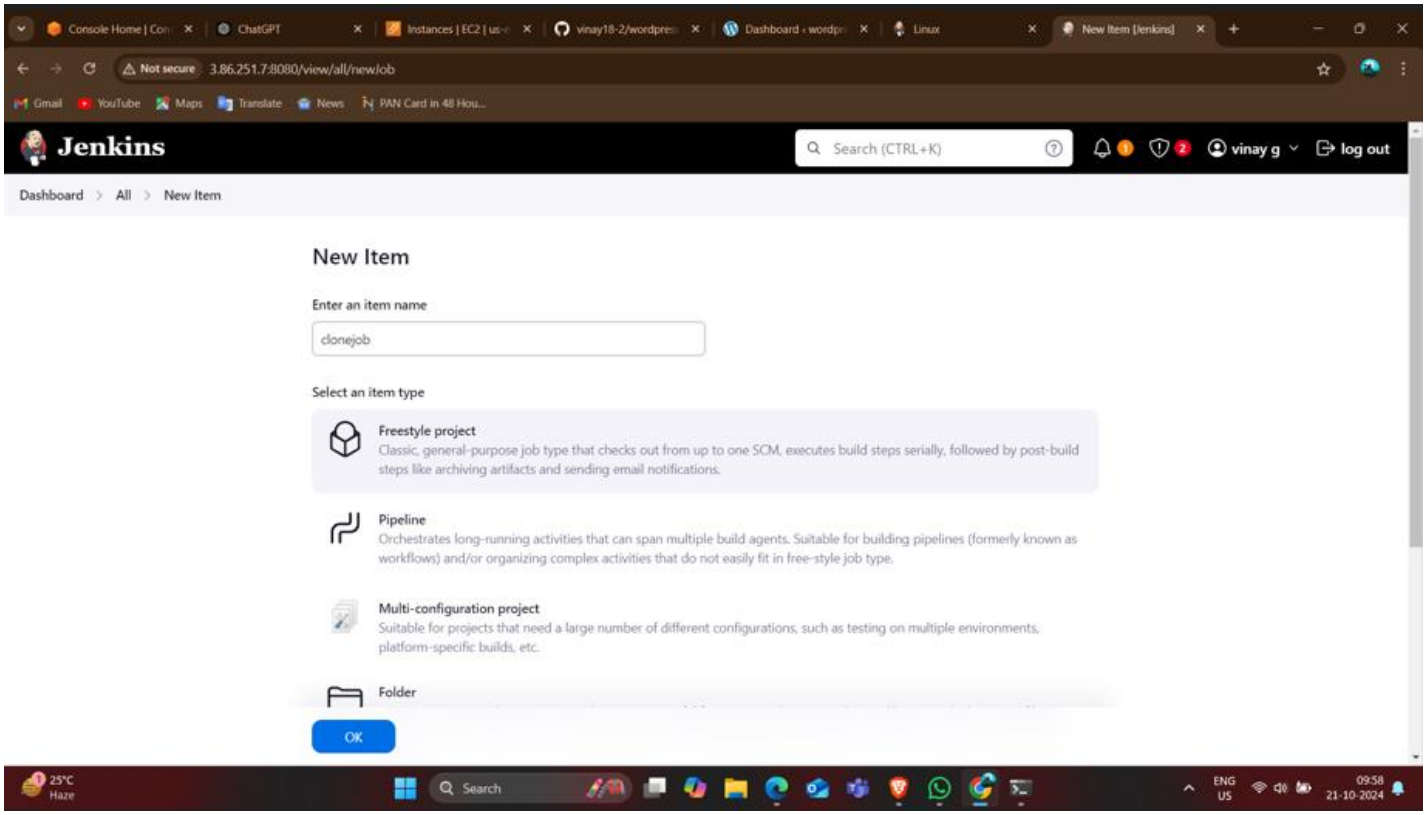


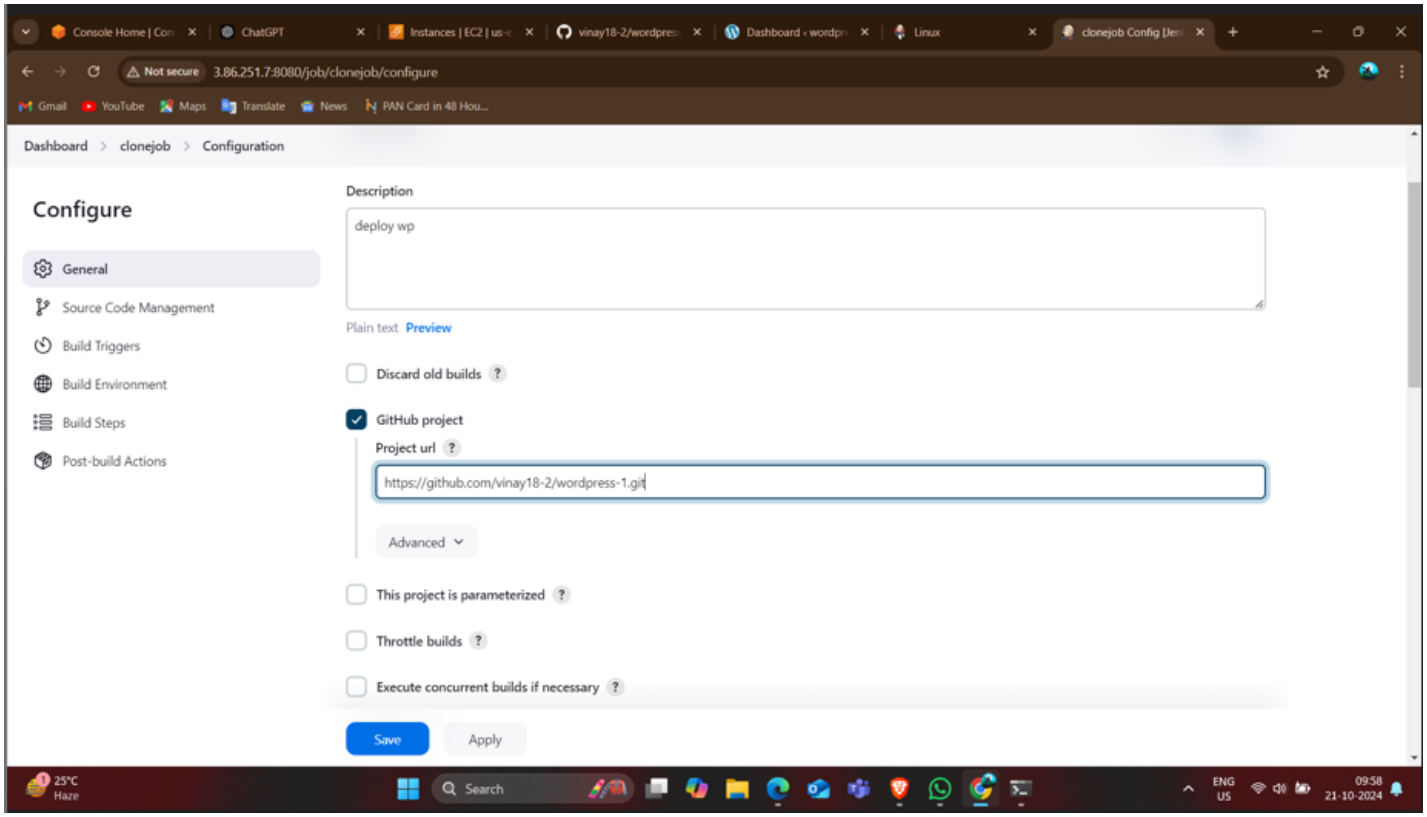
•


libXrender.x86_64 0:0.9.10-1.amzn2.0.2
libXtst.x86_64 0:1.2.3-1.amzn2.0.2
libxslt.x86_64 0:1.1.28-6.amzn2
python-javapackages.noarch 0:3.4.1-11.amzn2
libXt.x86_64 0:1.1.5-3.amzn2.0.2
libxcb.x86_64 0:1.12-1.amzn2.0.2
log4j-cve-2021-44228-hotpatch.noarch 0:1.3-7.amzn2
python-lxml.x86_64 0:3.2.1-4.amzn2.0.6
Complete!
[ec2-user@ip-172-31-81-169 ~]$ sudo yum install jenkins -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package jenkins.noarch 0:2.462.3-1.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
jenkins noarch 2.462.3-1.1 jenkins 89 M
Transaction Summary
Install 1 Package
Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.462.3-1.1.noarch.rpm | 89 MB 00:00:02
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : jenkins-2.462.3-1.1.noarch 1/1
Verifying : jenkins-2.462.3-1.1.noarch 1/1
Installed:
jenkins.noarch 0:2.462.3-1.1
Complete!
[ec2-user@ip-172-31-81-169 ~]$


Not secure 3.86.251.7:8080
Jenkins
Search (CTRL+K)
vinay g
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds
REST API
Jenkins 2.462.3

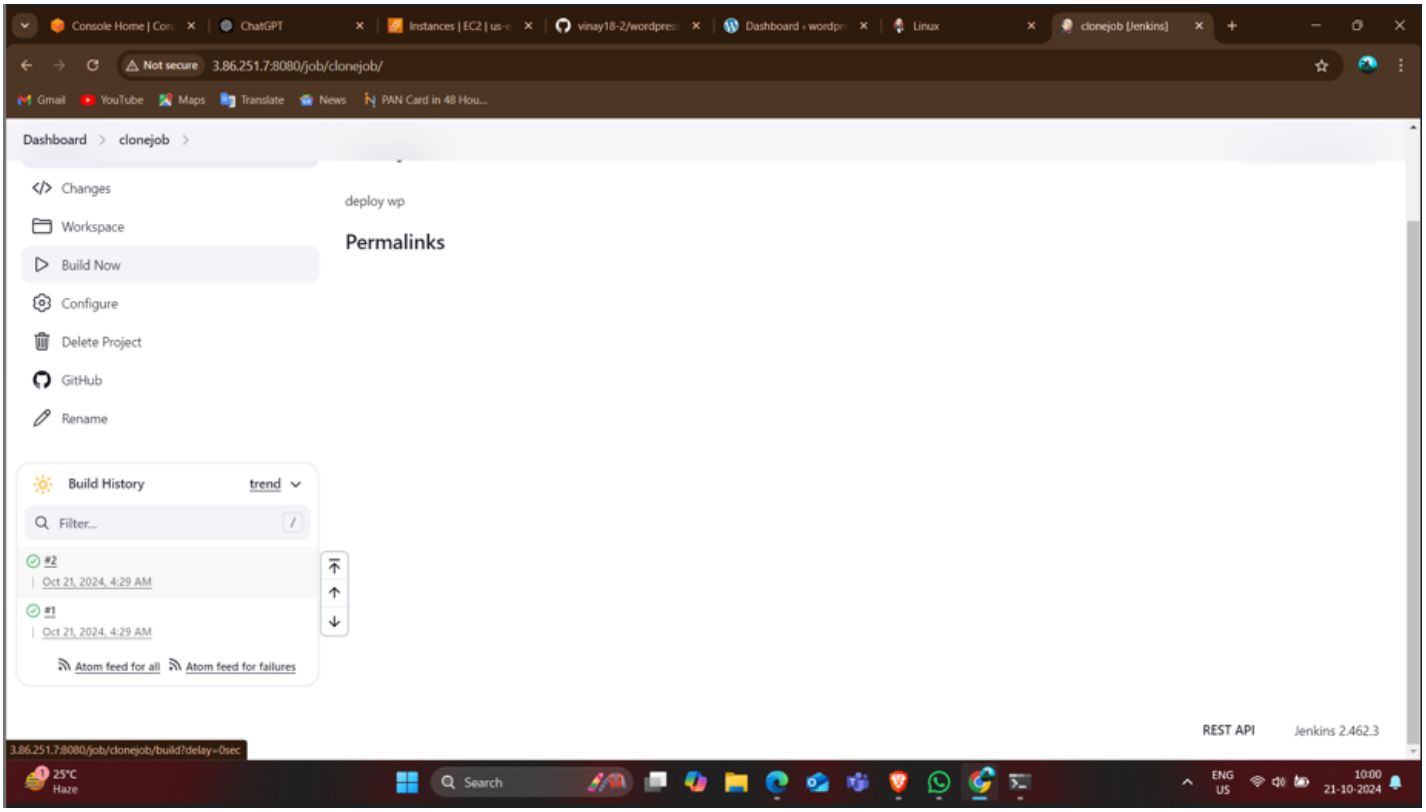
-
  - Create the new project clonejob.



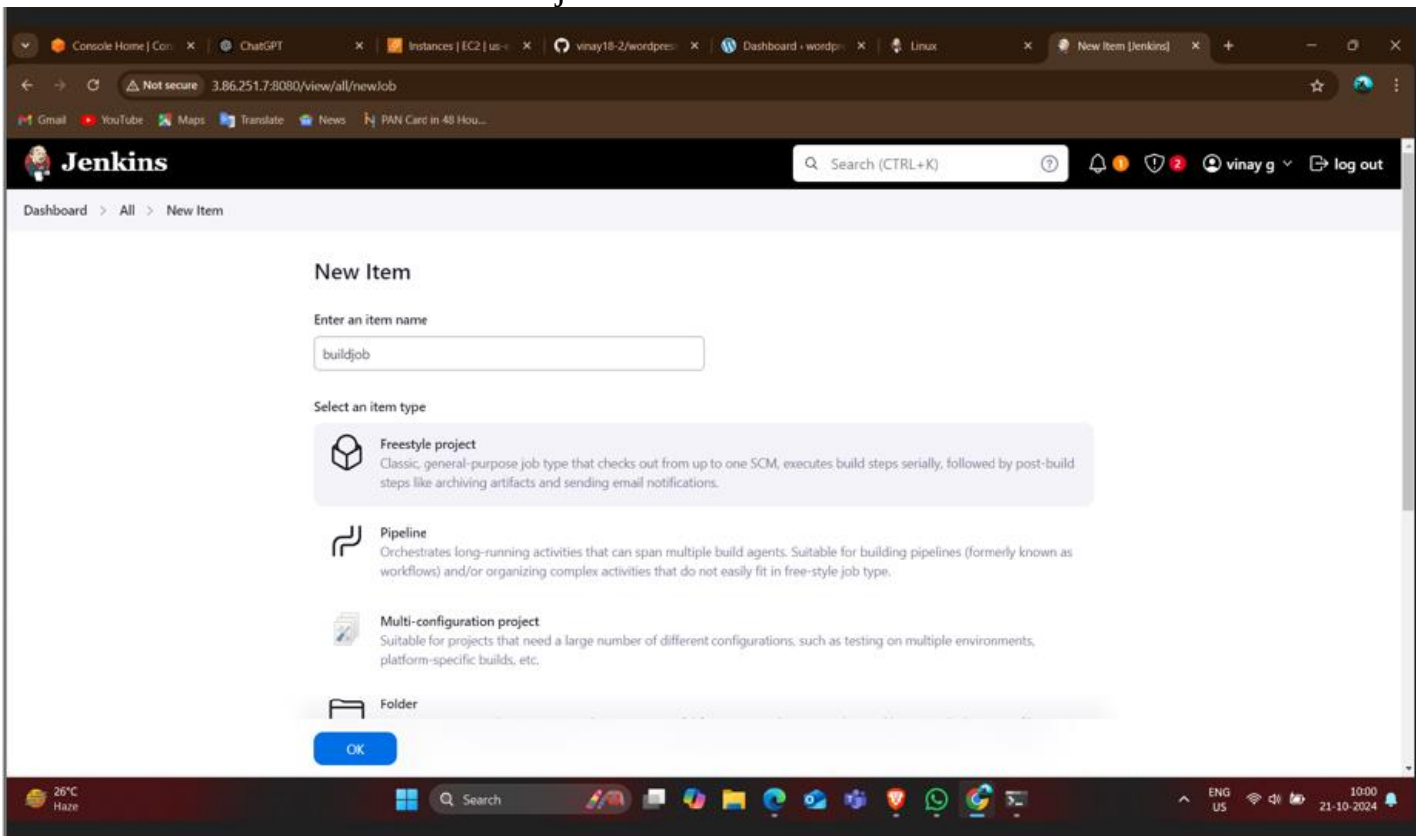


-

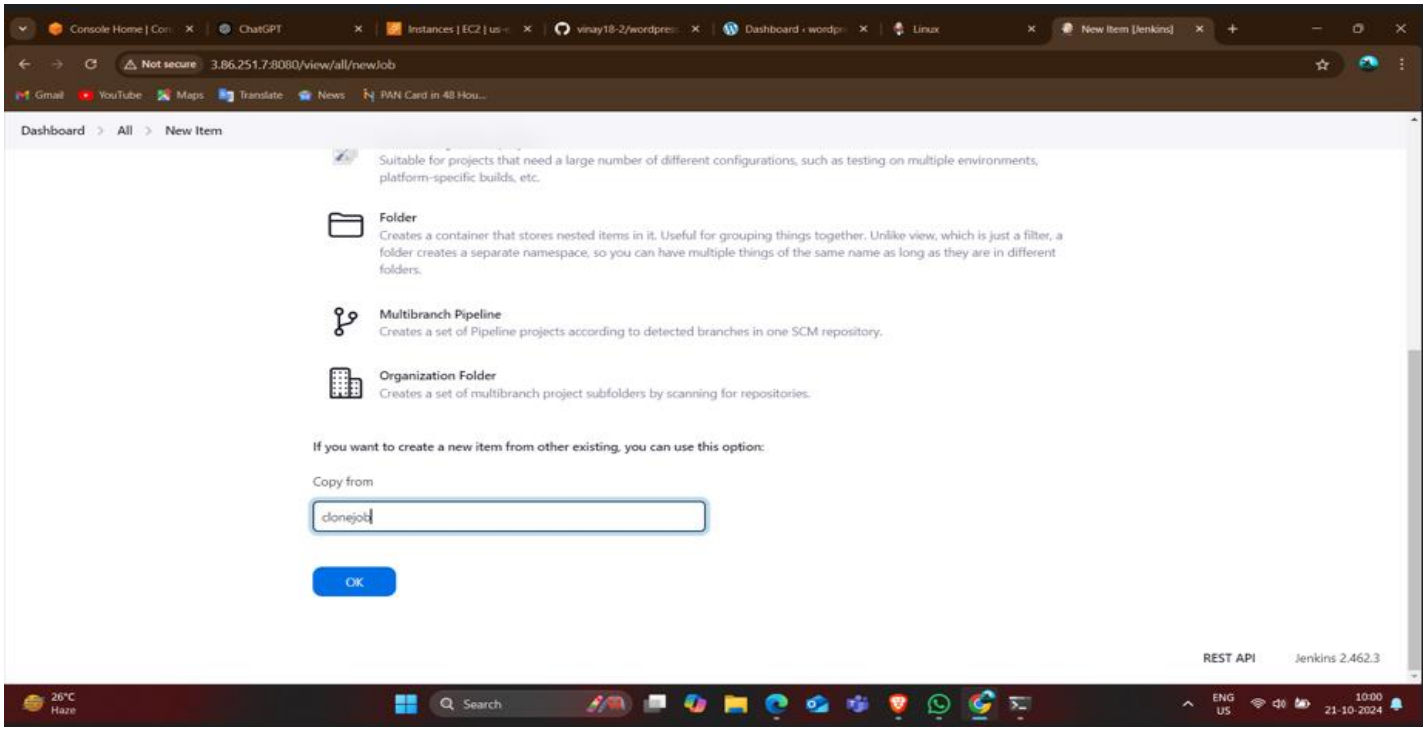
- Save it and build now.



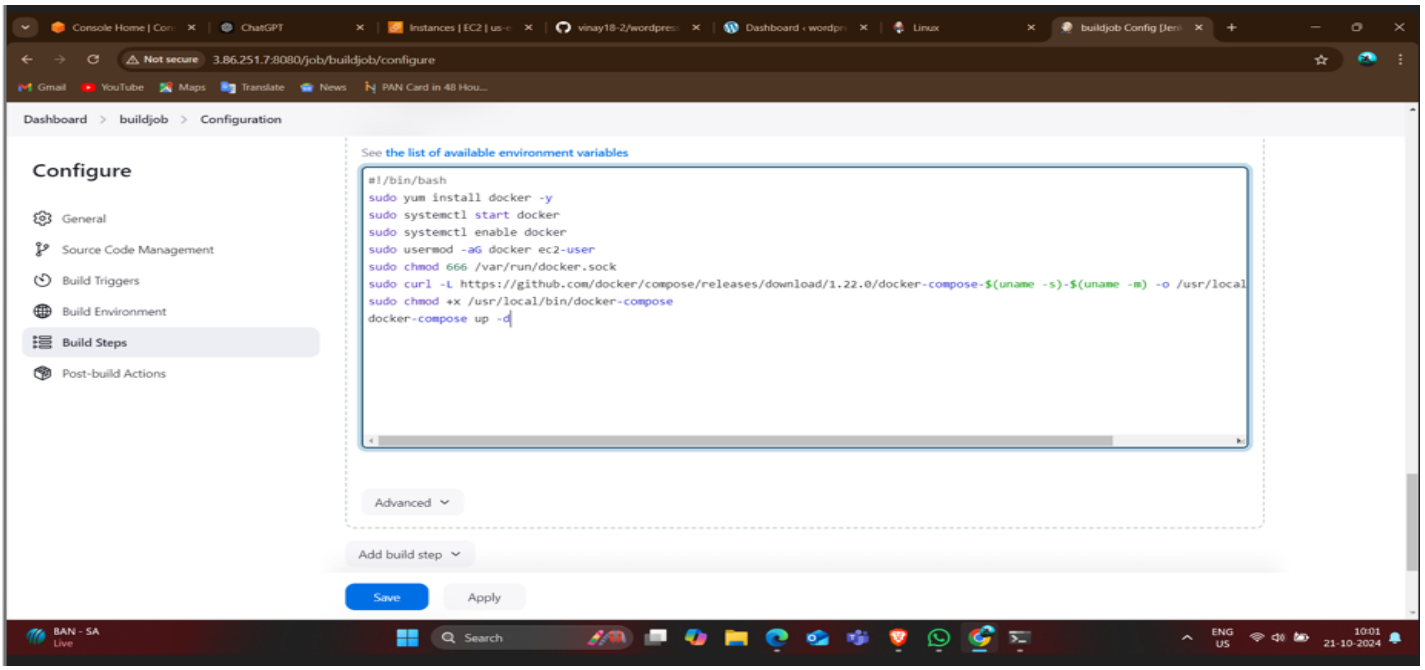
- Next create the new item buildjob.



-
  - Copy the exixting file to clone git and clonejob.

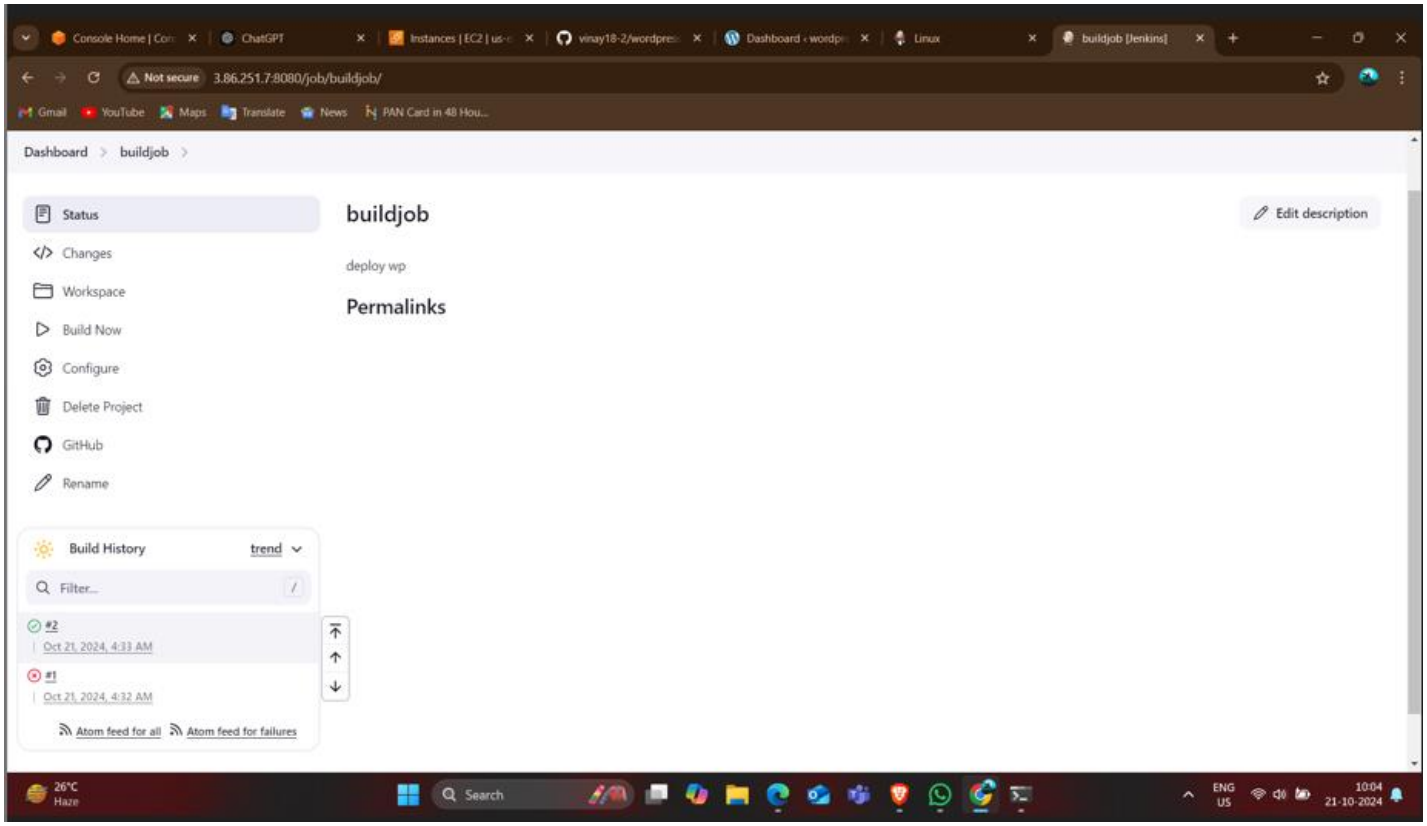


  - Write the bash script in the execute shell to install docker and docker-compose and clone the git repo install git also and run the docker than start the docker enable it,all the commands should be in the execute shell.

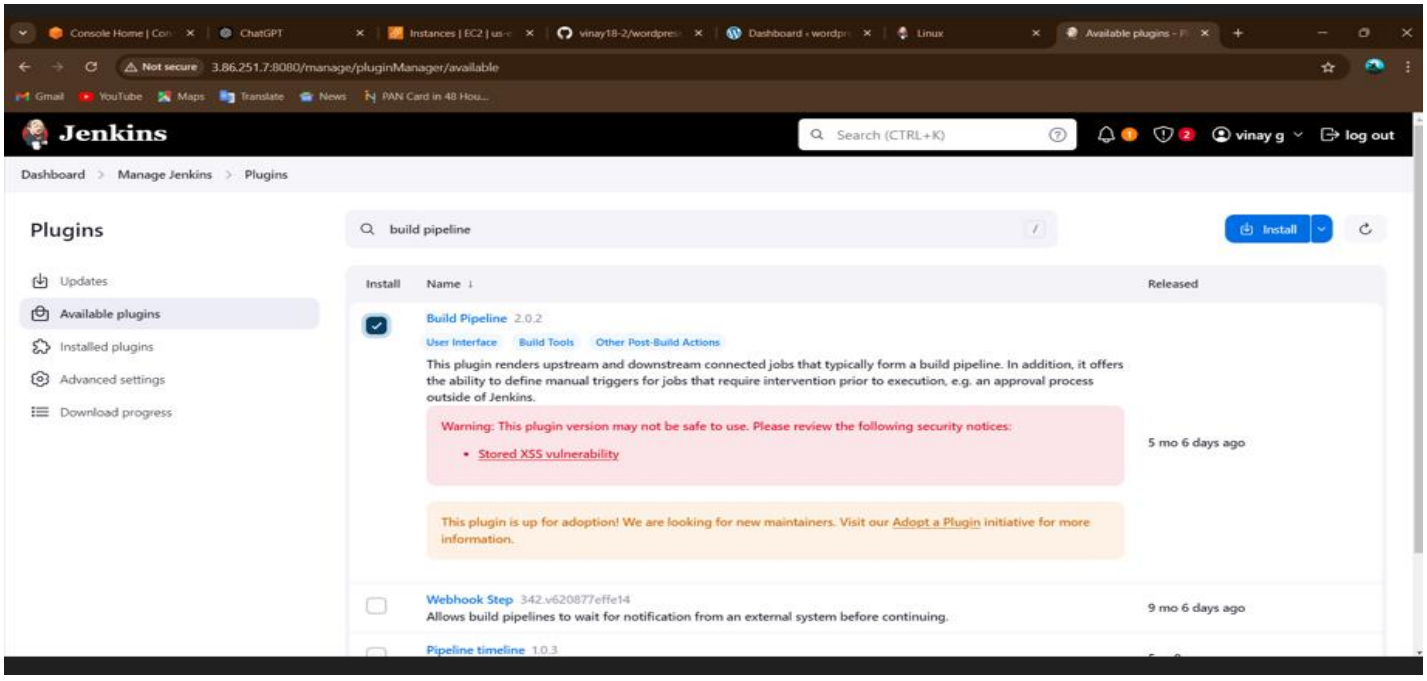


•

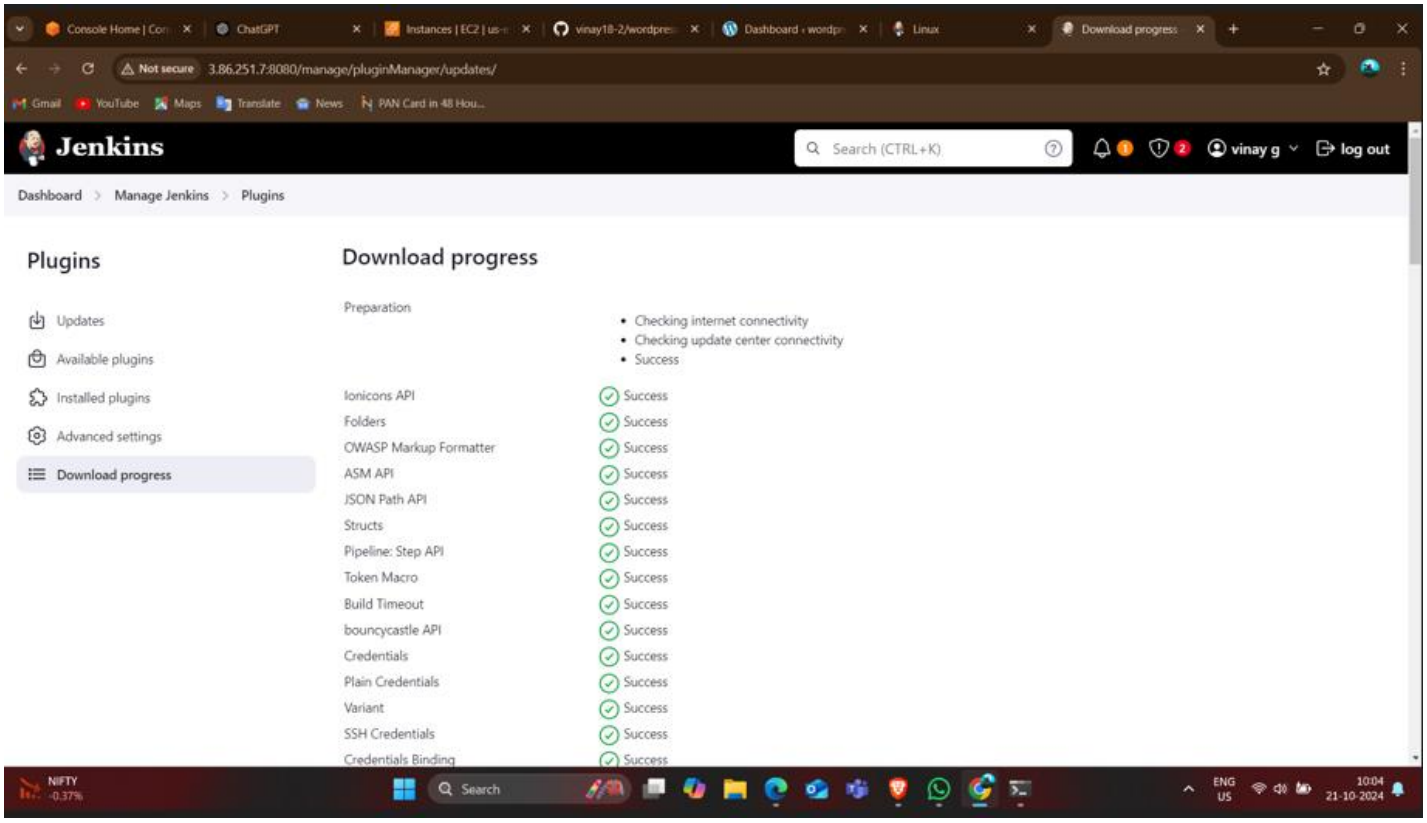
- Than save it and to run build it and observe it.



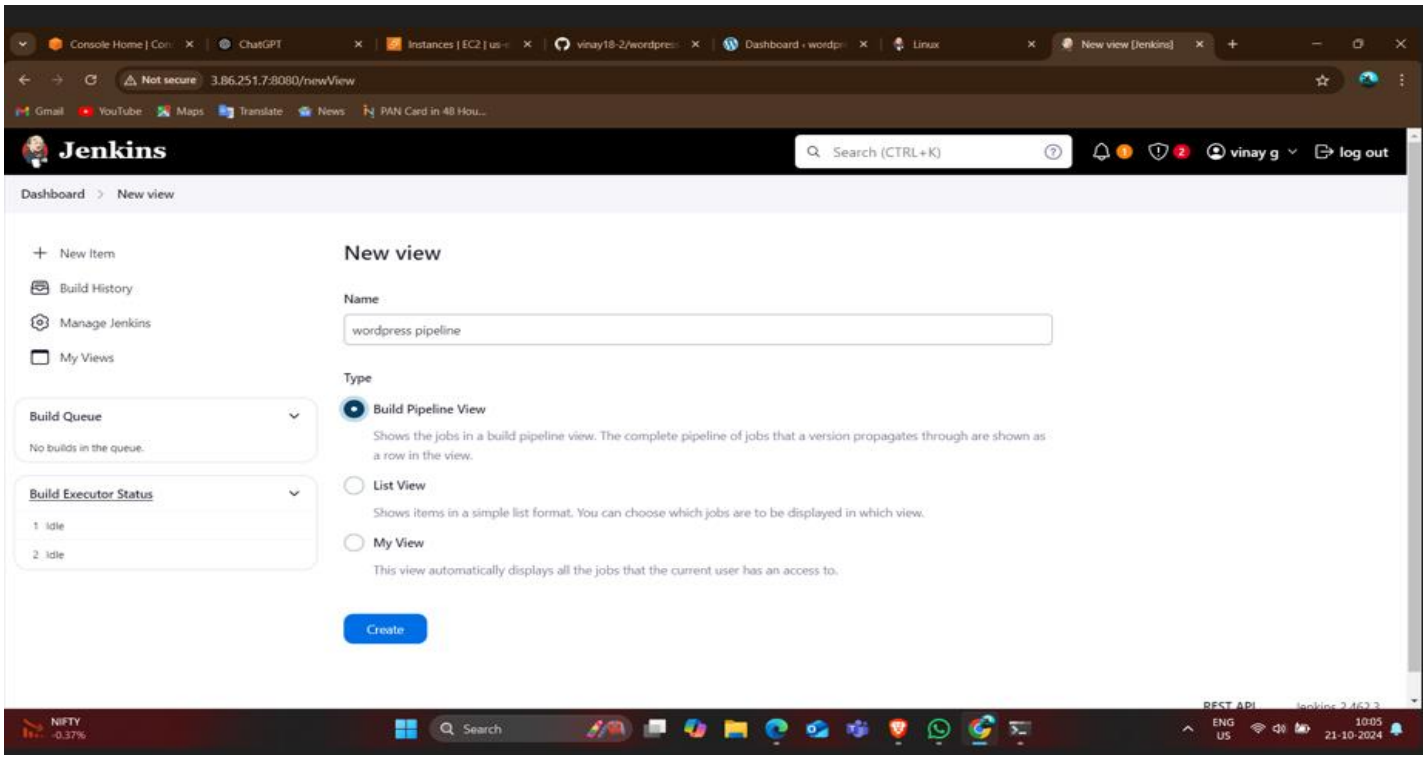
- In order to build pipeline we have to install the plugin.Go to dashboard,select the manage Jenkins in that we have the plugins, select it and go to available plugins and search for the build pipeline plugin and install it.



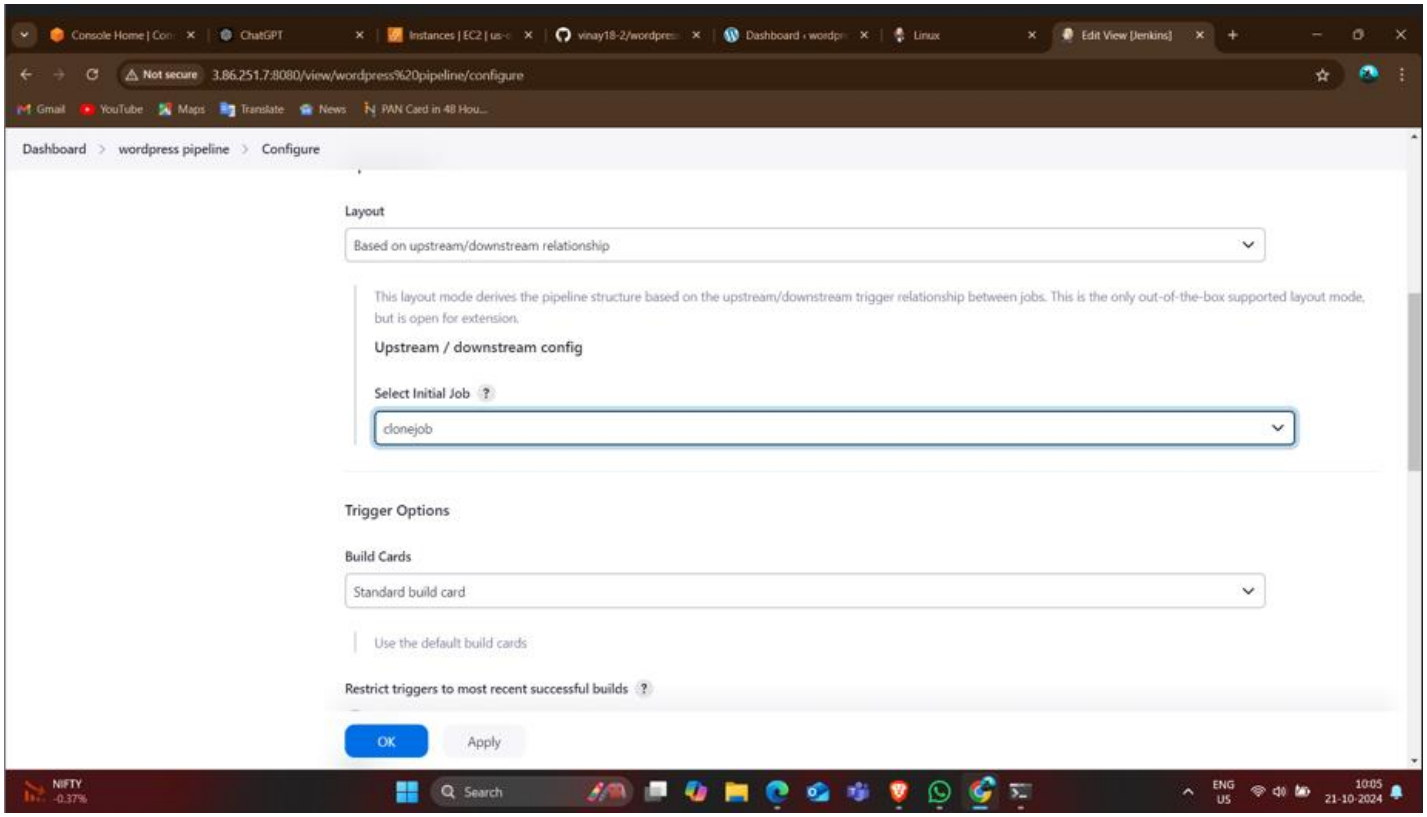
•



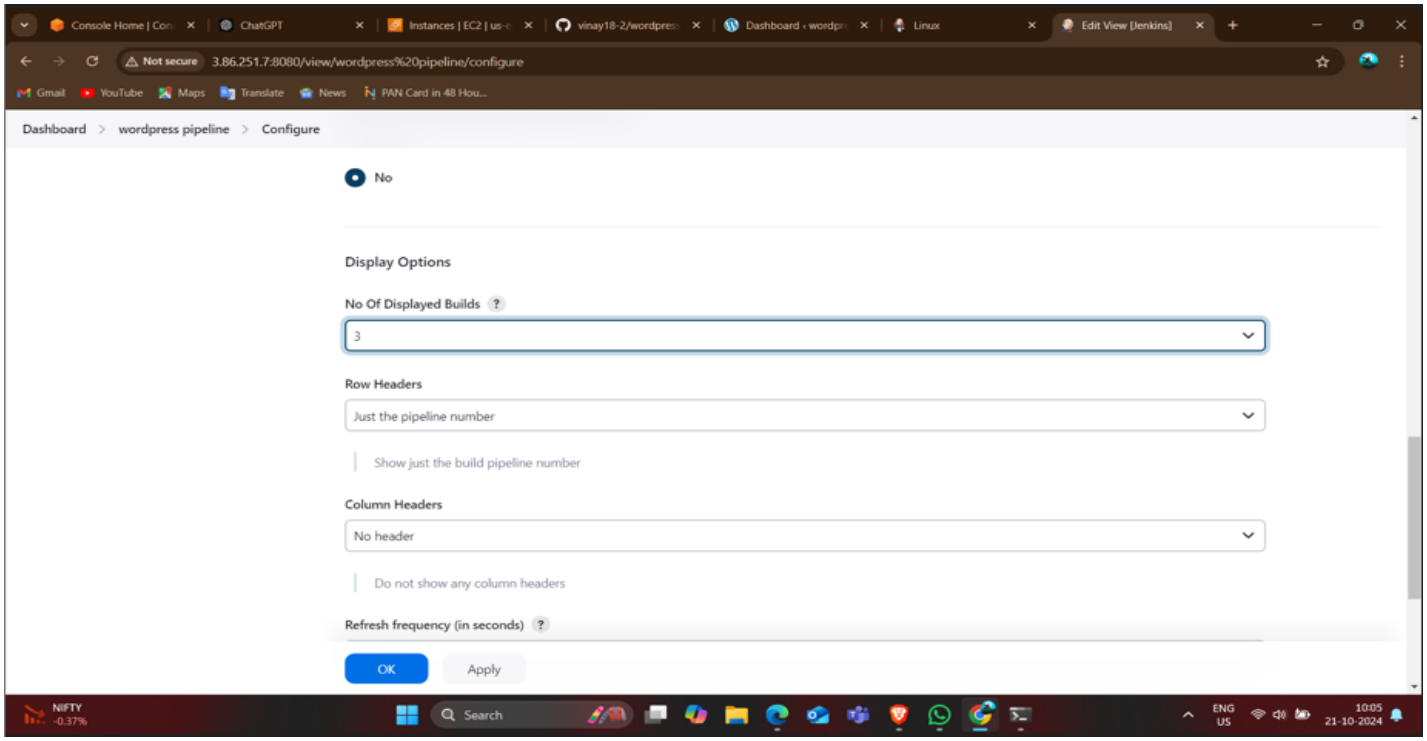
- Create the pipeline.



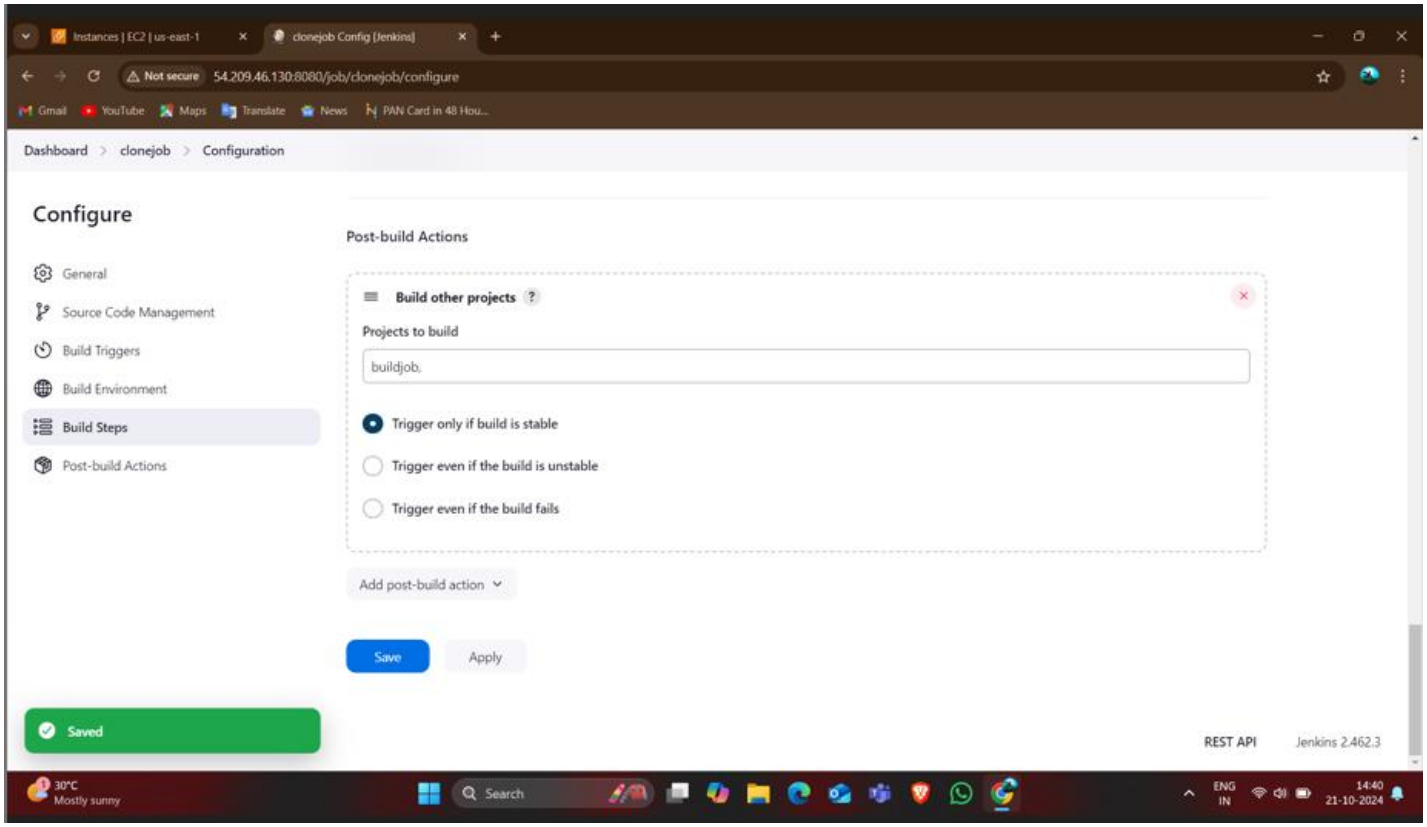
•



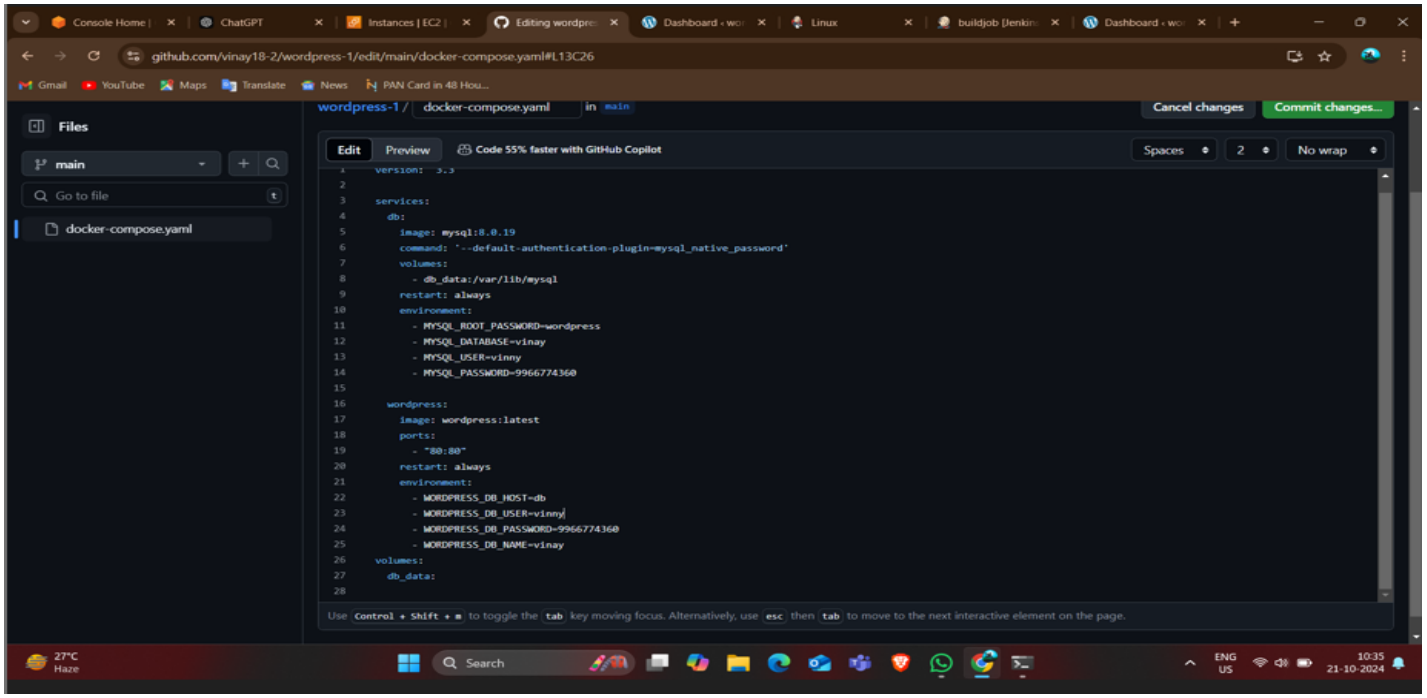
- Here the no.of displayed builds is used to displayes the how many builds we need that much count we have to provide.



- Select the post-build action, In that we have to select the buildjob to build the project.Than save and apply.

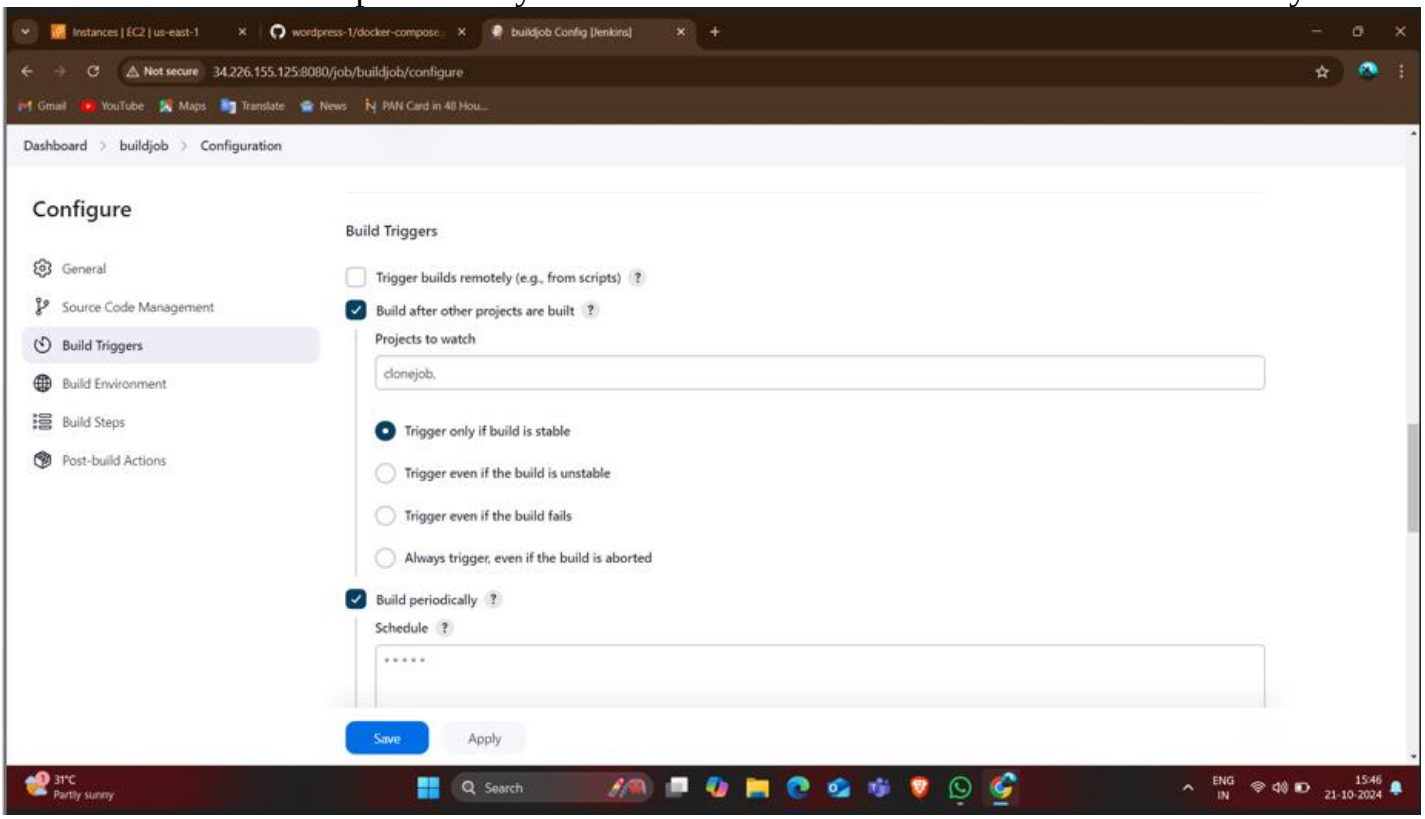


- Here the yaml code for build periodically.

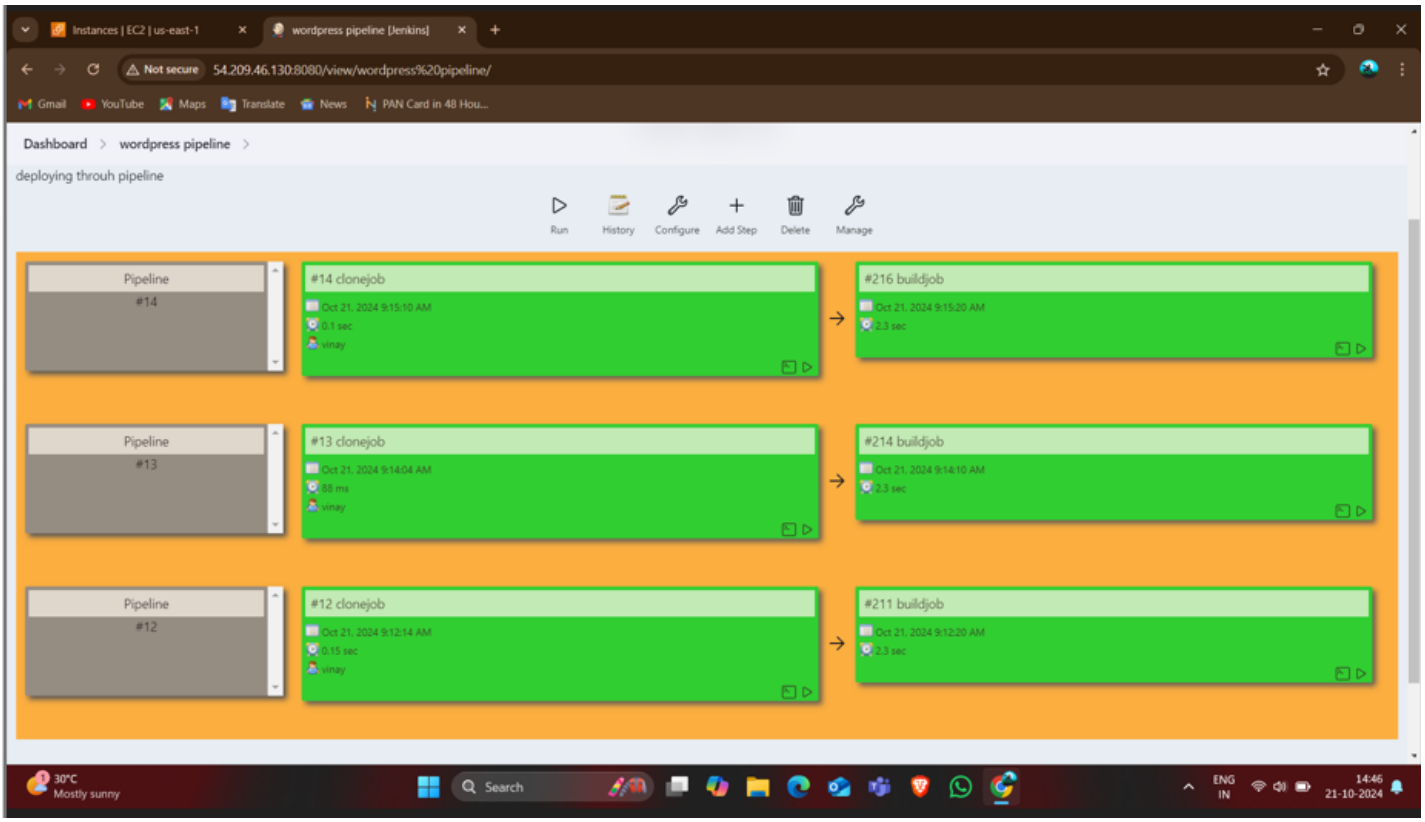


-

- This os for build periodically.here * * * * * indicates that the code runs for every time.

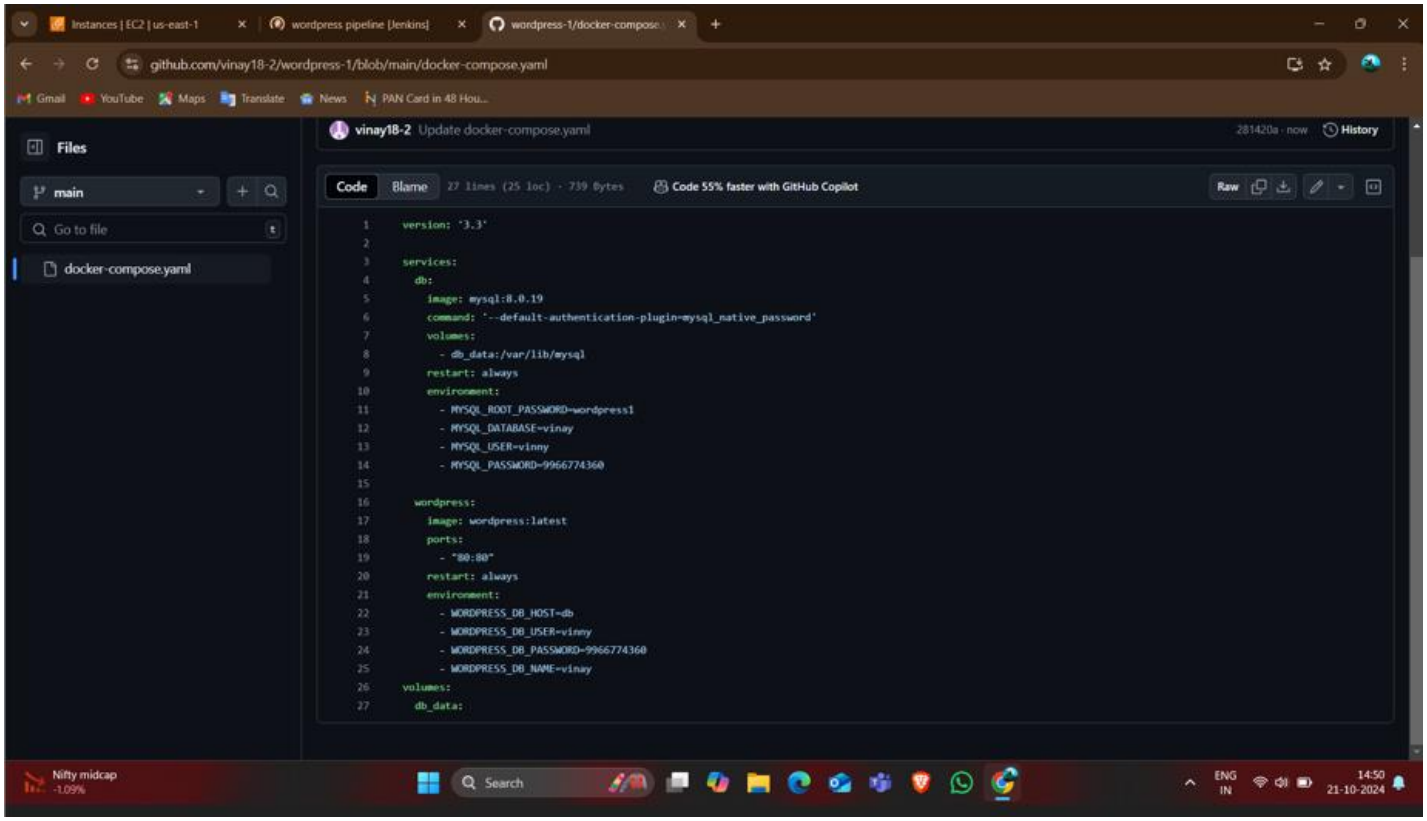


- Output for build periodically.

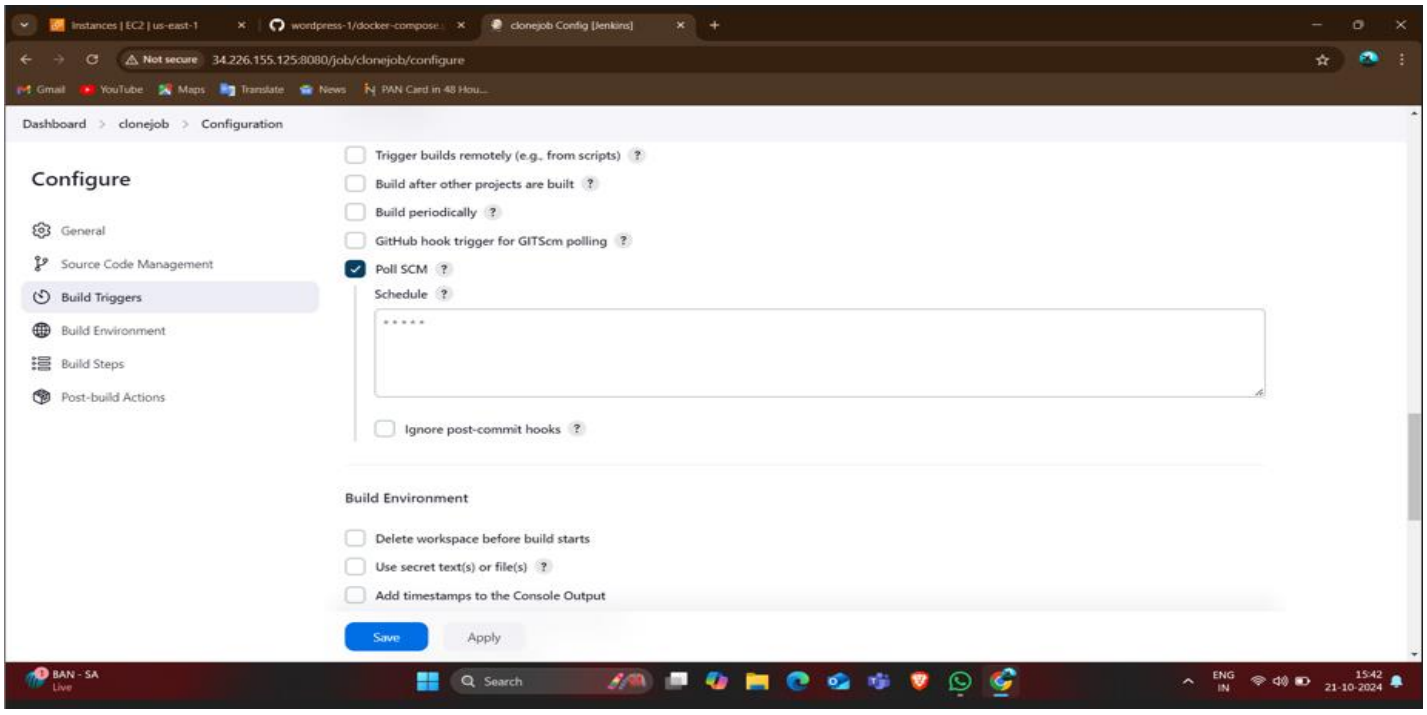


-

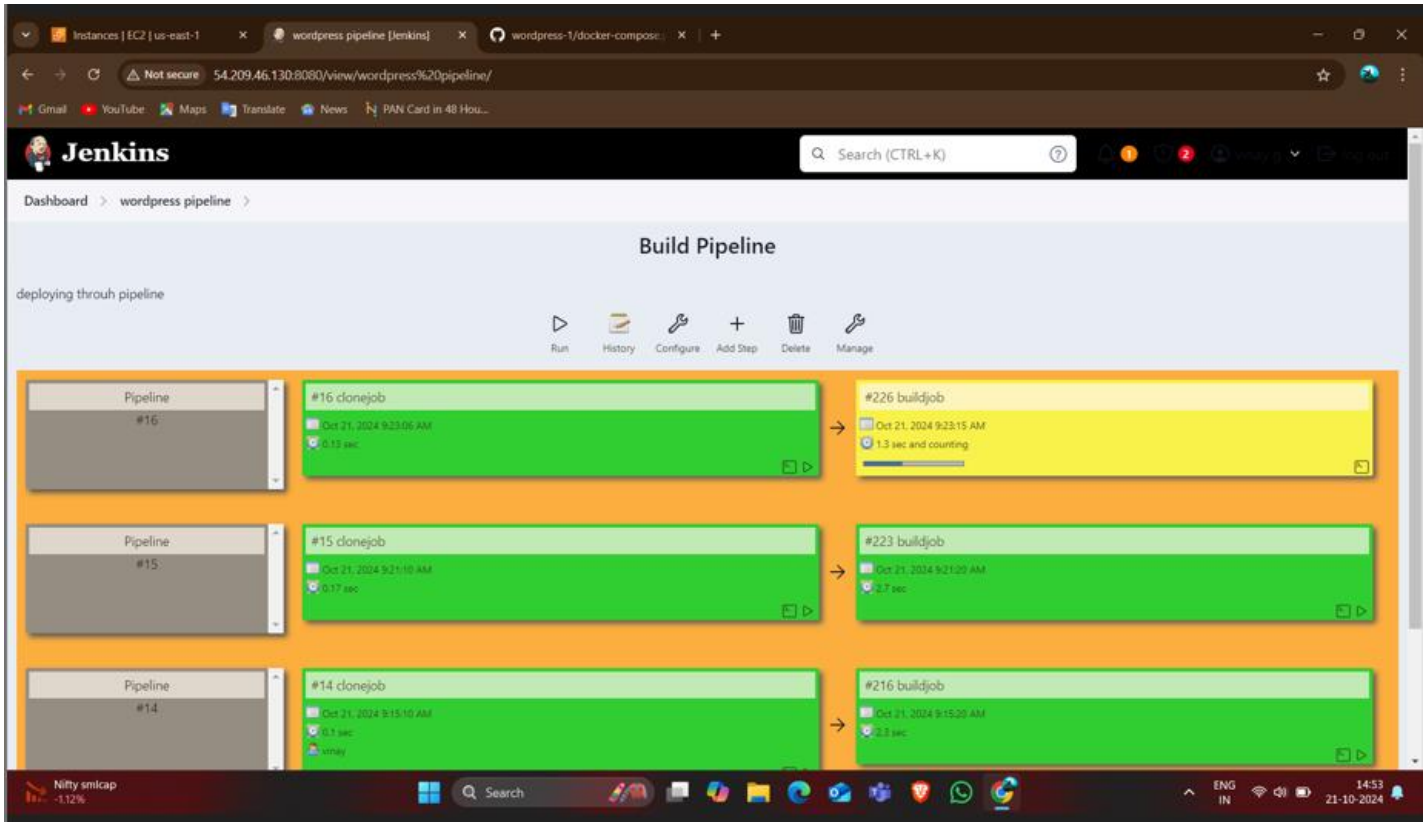
  - Modified yaml code for poll scm.



  - This is for poll scm .Here the schedule is * * * * * it indicates runs every minute.

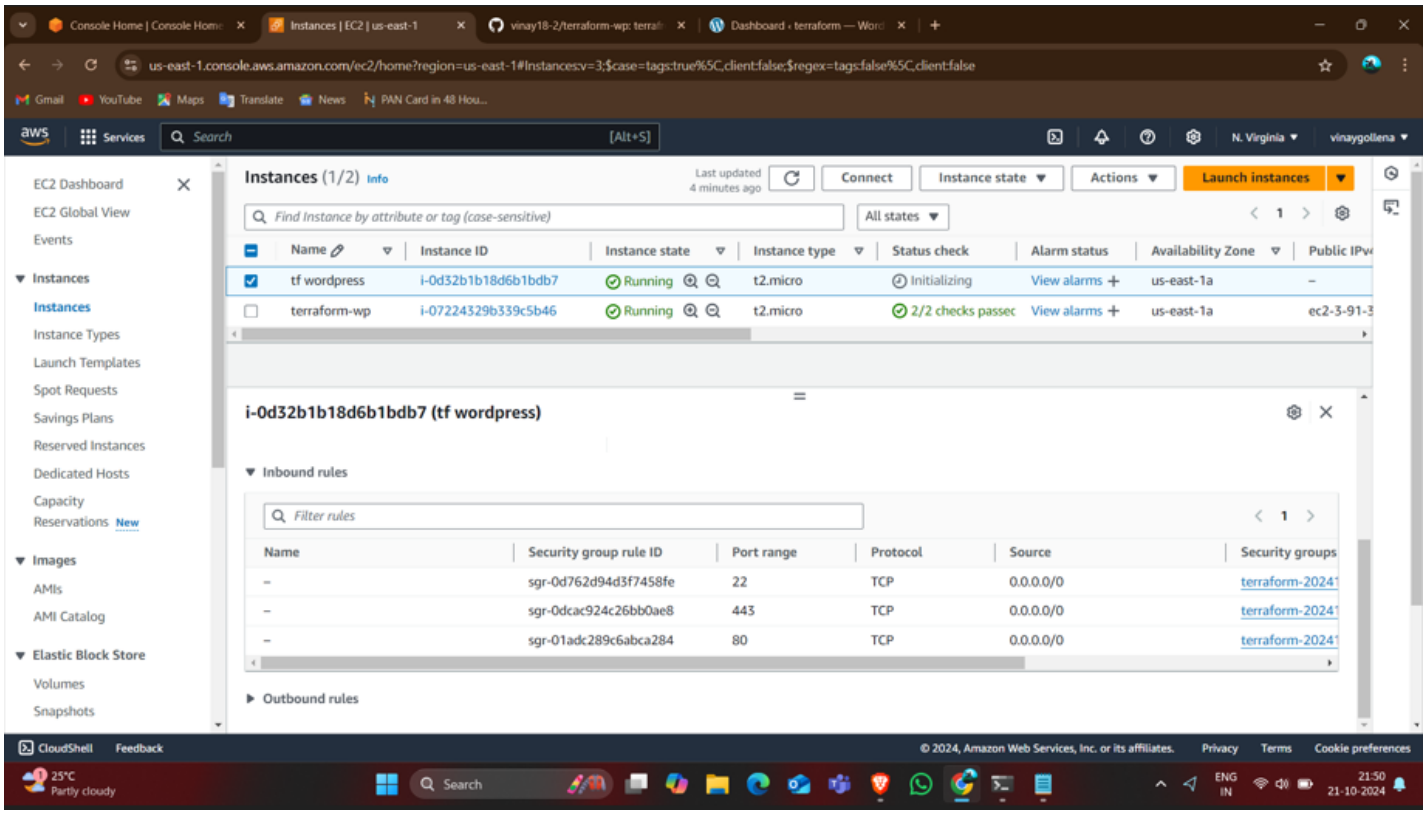


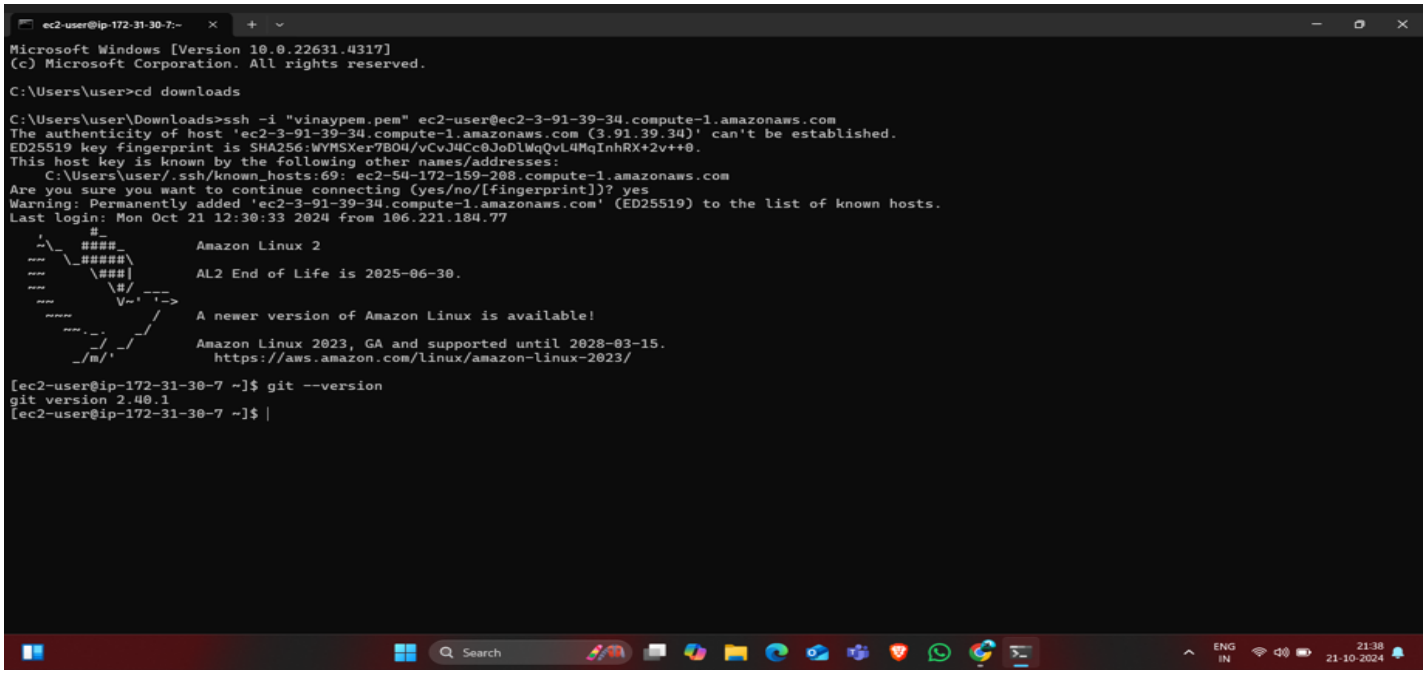
-
  - Output for poll scm.



Deploy WordPress web application by using terraform

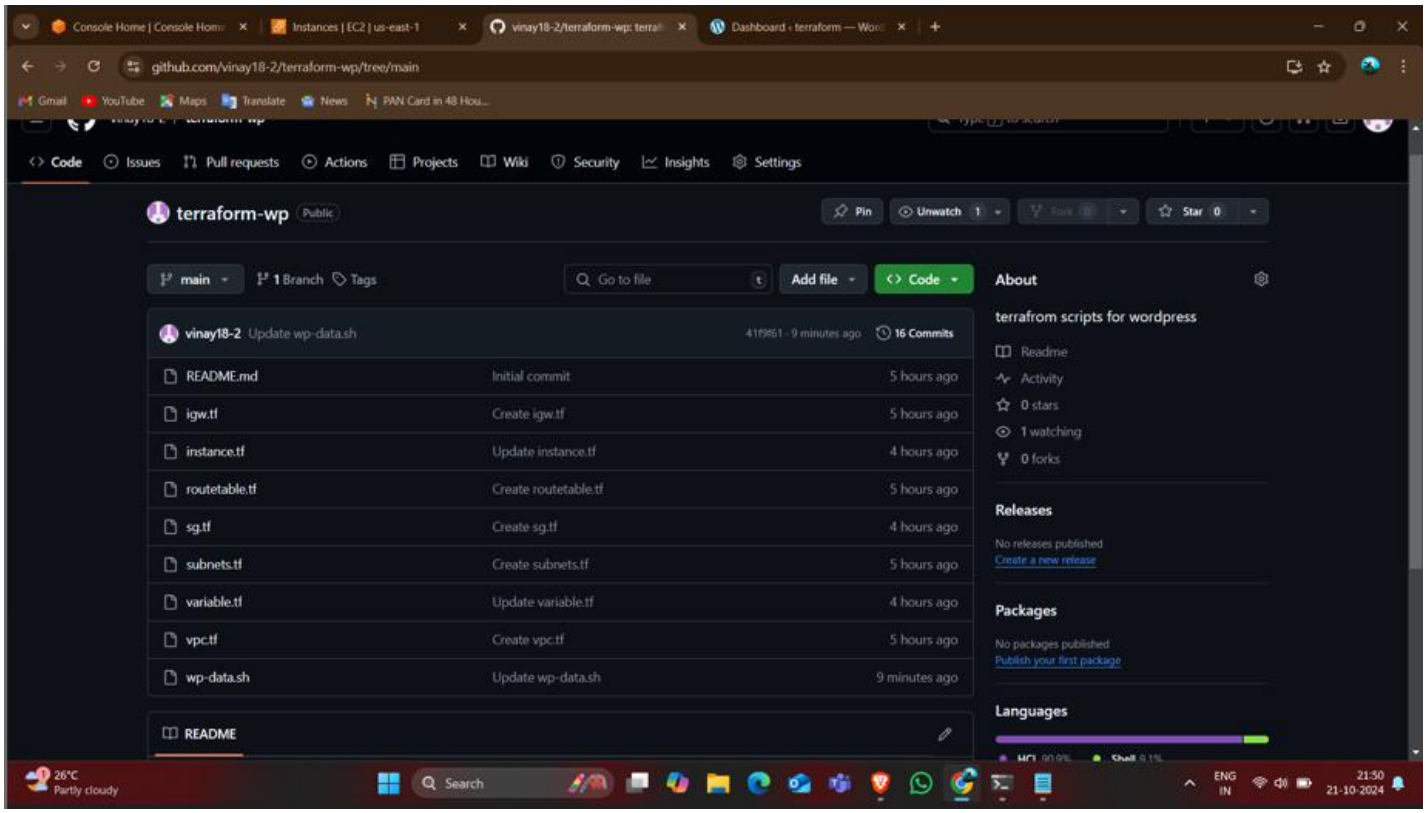
- Launch the EC2 instance with t2.medium.
- Connect to the terminal and instll git “sudo yum install git –y”.





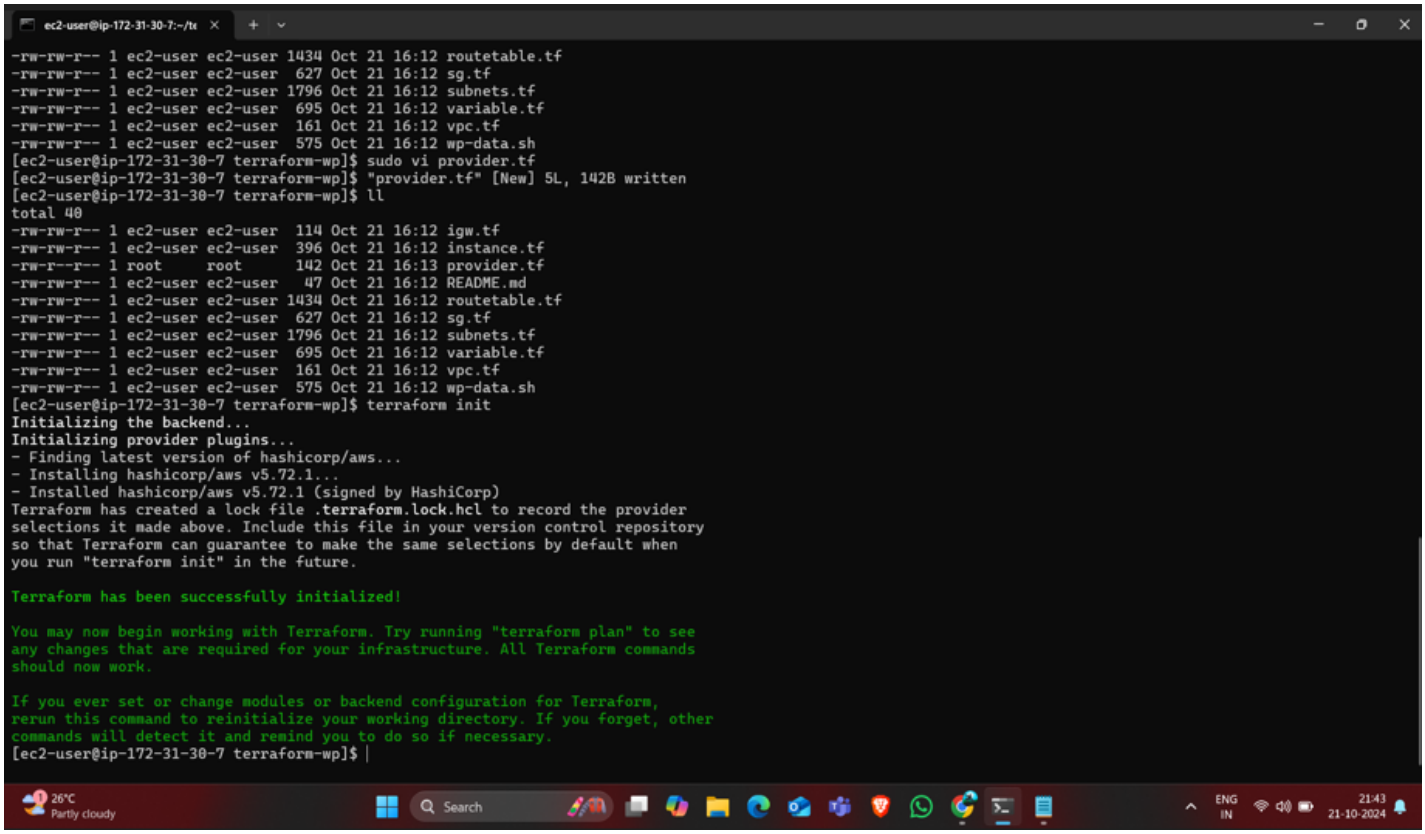
•

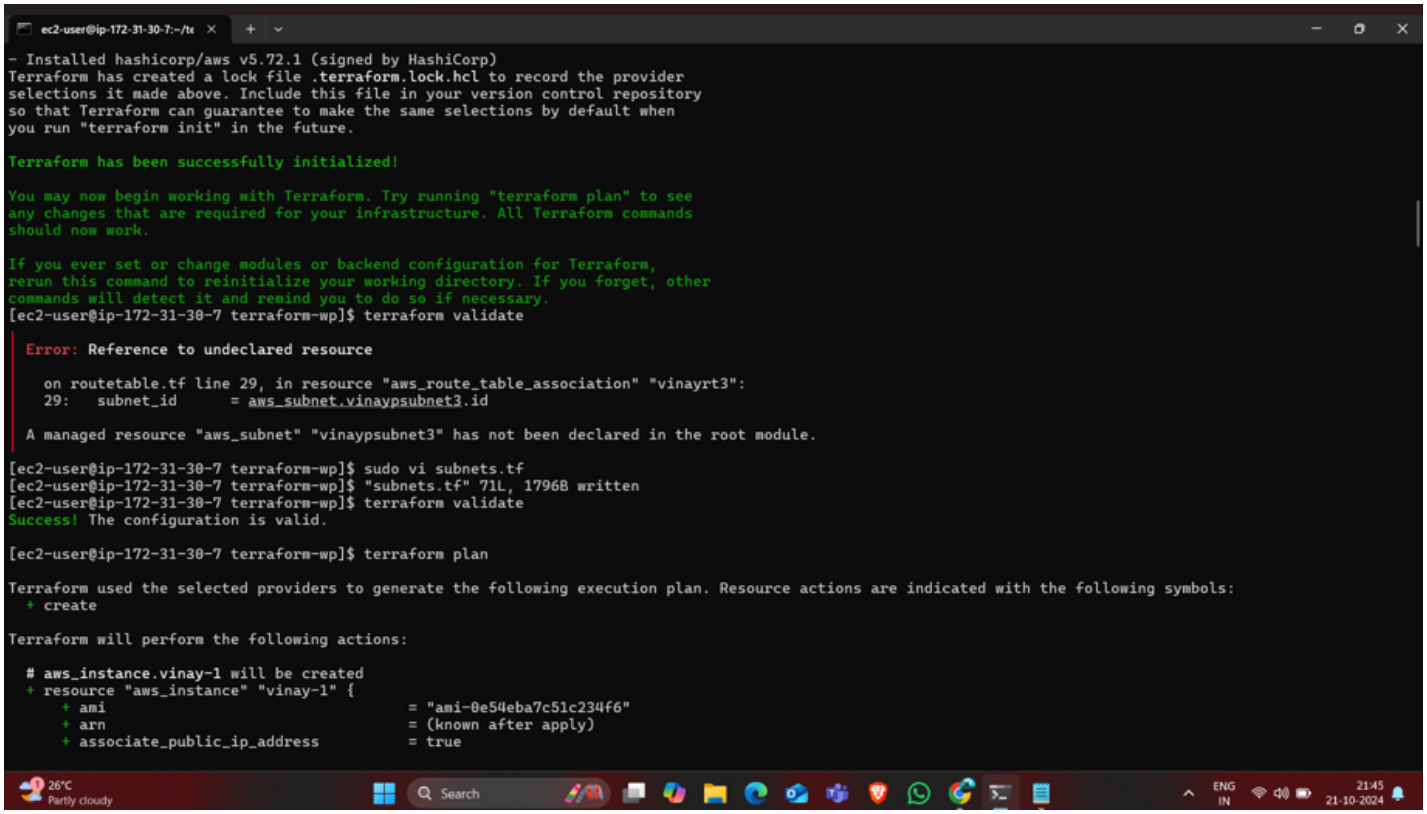
- Install the terrfaorm in the terminal.
- Than clone the git repositories of terraform scripts from git hub.
- We have to create the Instance with data.sh file,VPC,Subnets,Internet gateway,Route tables,Security groups and provider file.



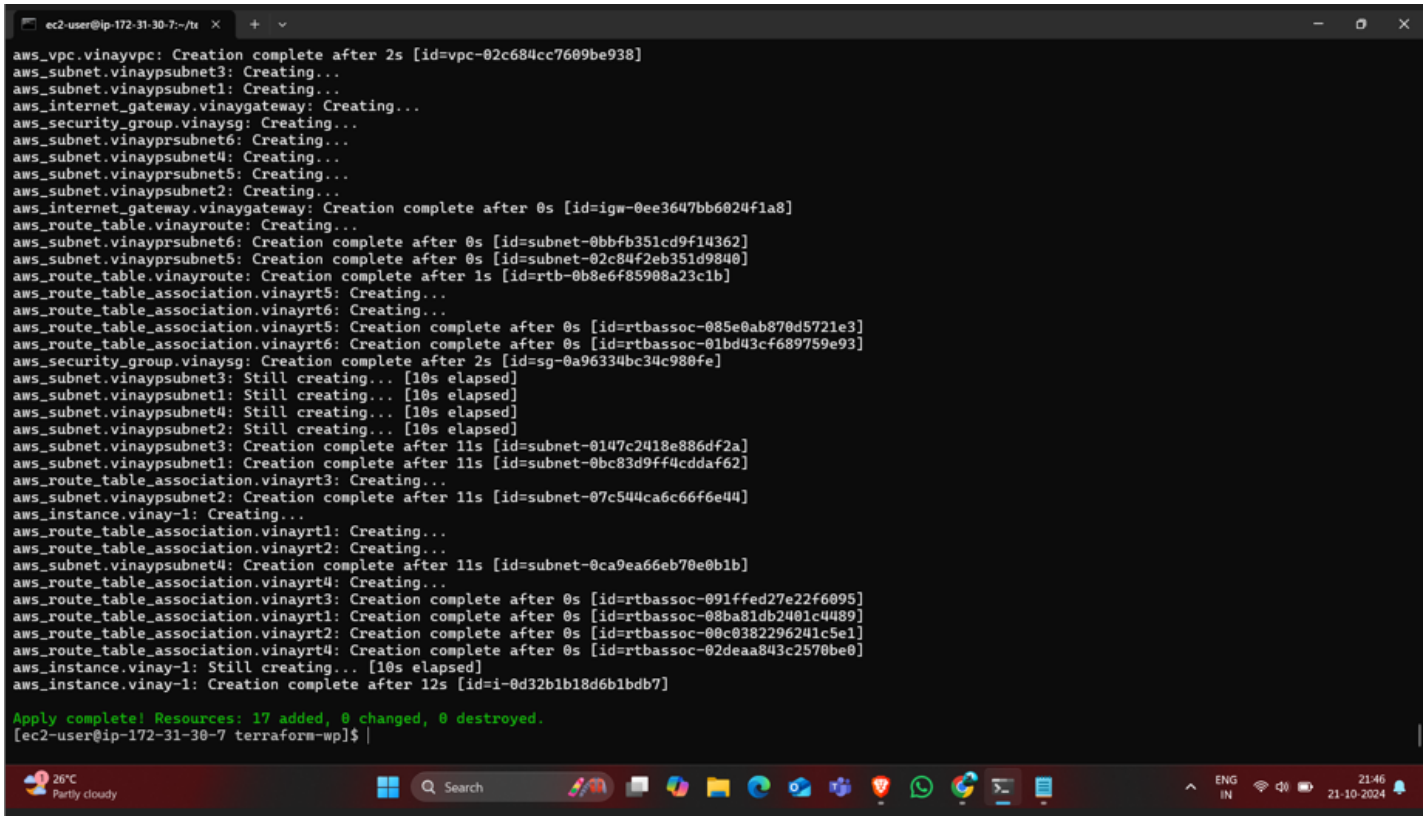
- After cloning that execute the terraform commands:
  “terraform init”
  “terraform validate”
  “terraform plan”
  “terrafrom apply”
  “terraform destroy”

•

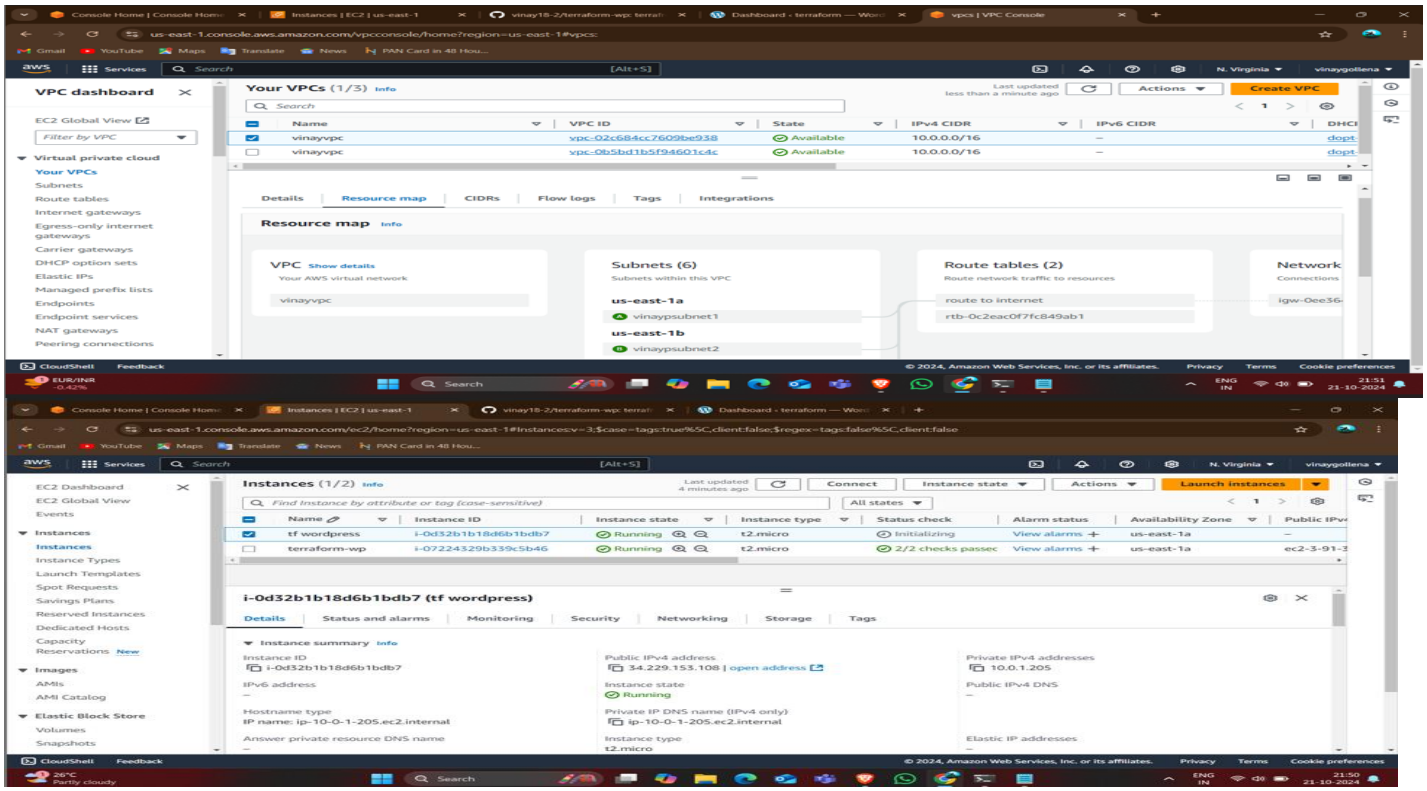

ec2-user@ip-172-31-30-7:~/te
-rw-rw-r-- 1 ec2-user ec2-user 1434 Oct 21 16:12 routetable.tf
-rw-rw-r-- 1 ec2-user ec2-user 627 Oct 21 16:12 sg.tf
-rw-rw-r-- 1 ec2-user ec2-user 1796 Oct 21 16:12 subnets.tf
-rw-rw-r-- 1 ec2-user ec2-user 695 Oct 21 16:12 variable.tf
-rw-rw-r-- 1 ec2-user ec2-user 161 Oct 21 16:12 vpc.tf
-rw-rw-r-- 1 ec2-user ec2-user 575 Oct 21 16:12 wp-data.sh
[ec2-user@ip-172-31-30-7 terraform-wp]$ sudo vi provider.tf
[ec2-user@ip-172-31-30-7 terraform-wp]$ "provider.tf" [New] 5L, 142B written
[ec2-user@ip-172-31-30-7 terraform-wp]$ ll
total 40
-rw-rw-r-- 1 ec2-user ec2-user 114 Oct 21 16:12 igw.tf
-rw-rw-r-- 1 ec2-user ec2-user 396 Oct 21 16:12 instance.tf
-rw-r--r-- 1 root root 142 Oct 21 16:13 provider.tf
-rw-rw-r-- 1 ec2-user ec2-user 47 Oct 21 16:12 README.md
-rw-rw-r-- 1 ec2-user ec2-user 1434 Oct 21 16:12 routetable.tf
-rw-rw-r-- 1 ec2-user ec2-user 627 Oct 21 16:12 sg.tf
-rw-rw-r-- 1 ec2-user ec2-user 1796 Oct 21 16:12 subnets.tf
-rw-rw-r-- 1 ec2-user ec2-user 695 Oct 21 16:12 variable.tf
-rw-rw-r-- 1 ec2-user ec2-user 161 Oct 21 16:12 vpc.tf
-rw-rw-r-- 1 ec2-user ec2-user 575 Oct 21 16:12 wp-data.sh
[ec2-user@ip-172-31-30-7 terraform-wp]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-30-7 terraform-wp]$
26°C
Partly cloudy
Search
ENG
IN
21:43
21-10-2024


ec2-user@ip-172-31-30-7:~/te
- Installed hashicorp/aws v5.72.1 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-30-7 terraform-wp]$ terraform validate
Error: Reference to undeclared resource
on routetable.tf line 29, in resource "aws_route_table_association" "vinayrt3":
29: subnet_id = aws_subnet.vinaypsubnet3.id
A managed resource "aws_subnet" "vinaypsubnet3" has not been declared in the root module.
[ec2-user@ip-172-31-30-7 terraform-wp]$ sudo vi subnets.tf
[ec2-user@ip-172-31-30-7 terraform-wp]$ "subnets.tf" 71L, 1796B written
[ec2-user@ip-172-31-30-7 terraform-wp]$ terraform validate
Success! The configuration is valid.
[ec2-user@ip-172-31-30-7 terraform-wp]$ terraform plan
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.vinay-1 will be created
+ resource "aws_instance" "vinay-1" {
+ ami = "ami-0e54eba7c51c234f6"
+ arn = (known after apply)
+ associate_public_ip_address = true
26°C
Partly cloudy
Search
ENG
IN
21:45
21-10-2024

•

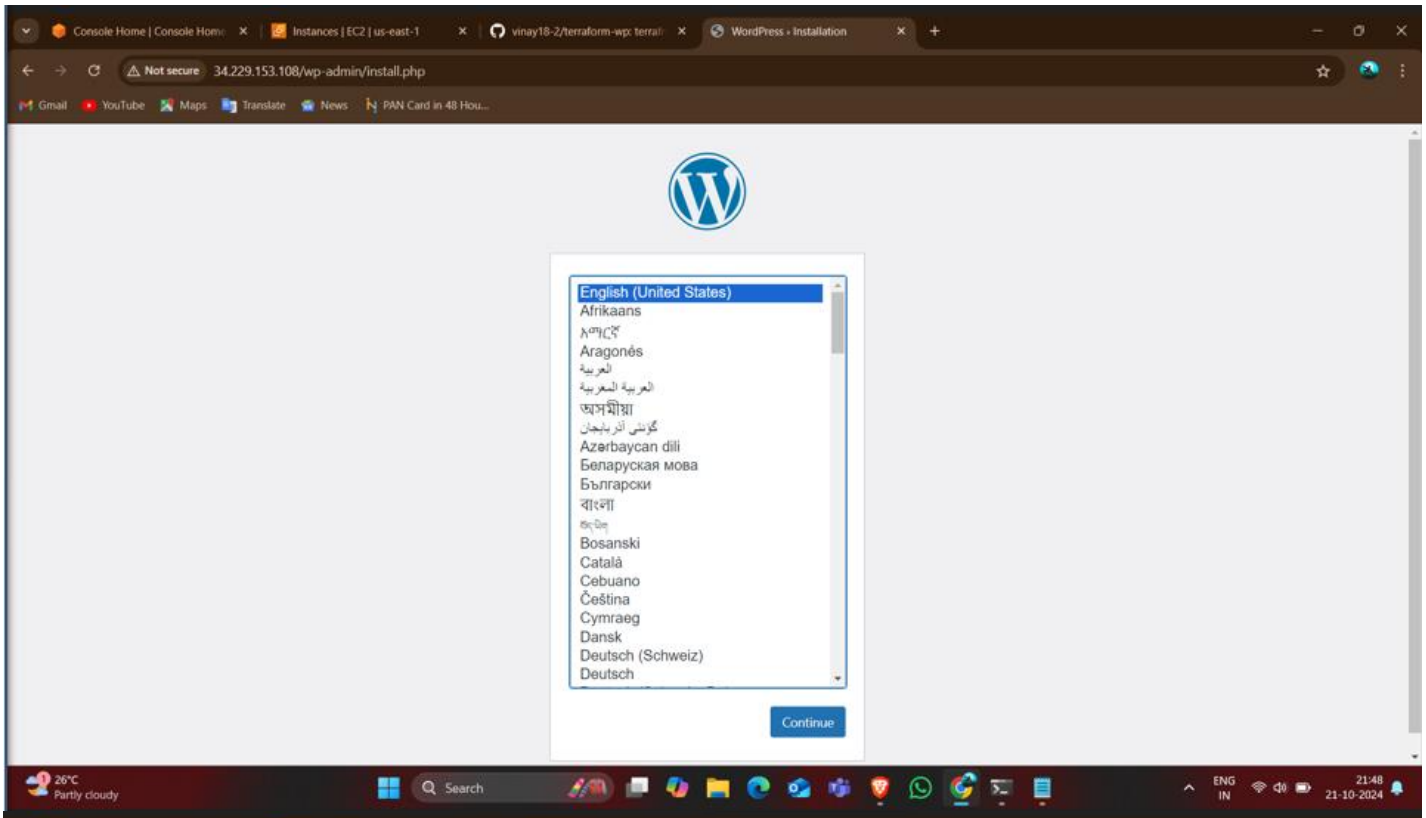


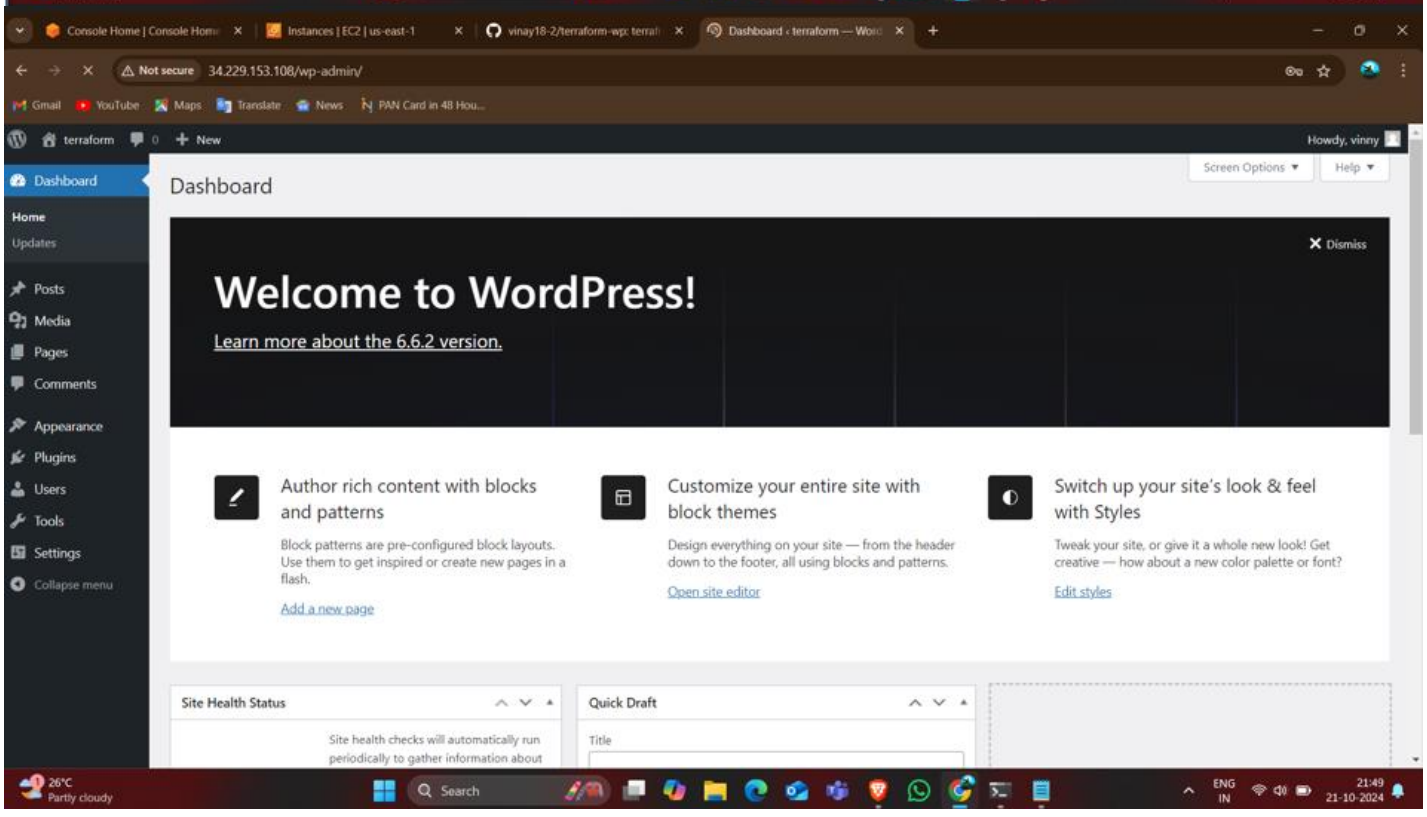
- Here the results of terraform apply applying.



•

- Outputs.
- Copy the created instance ip and browse it in google.

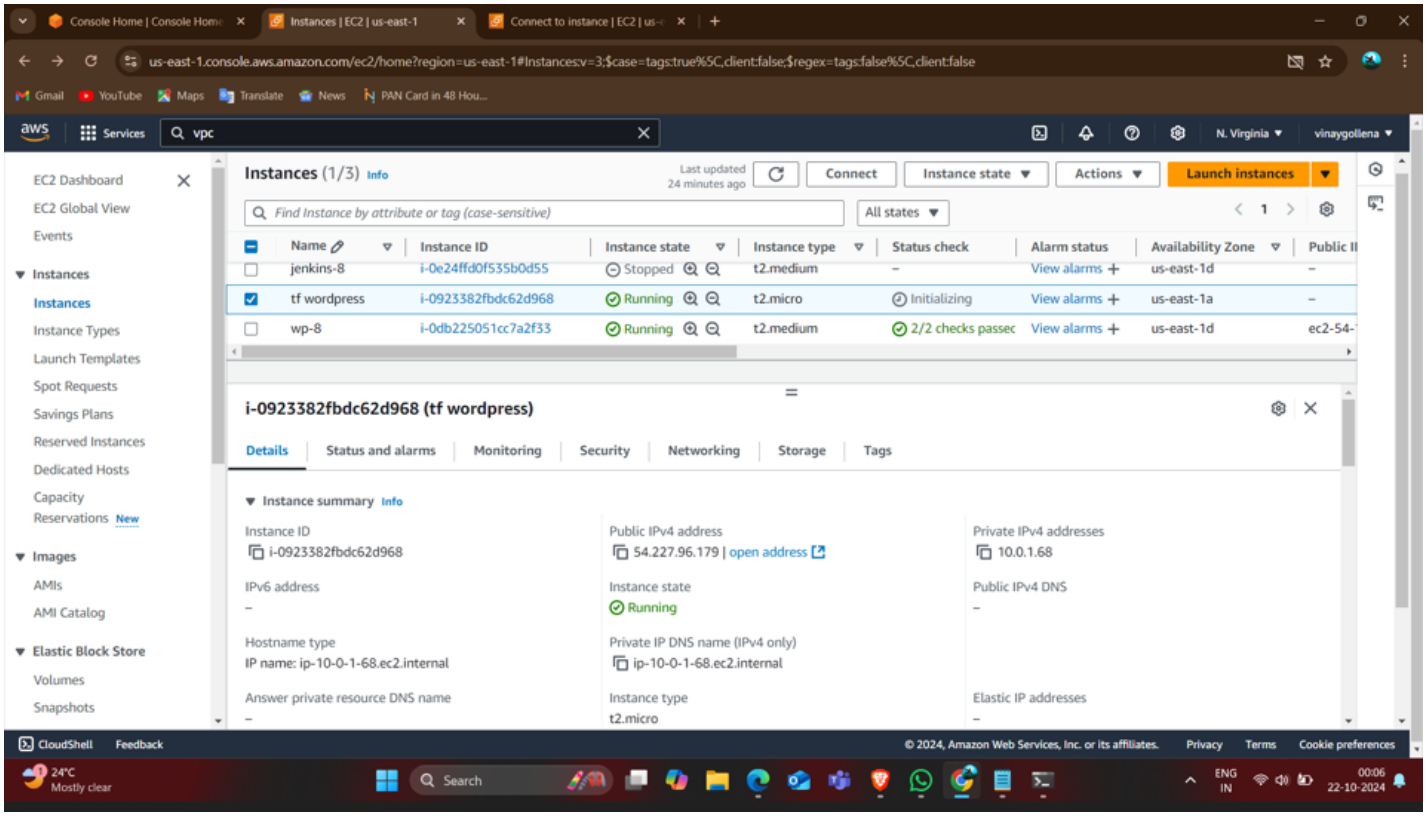




•

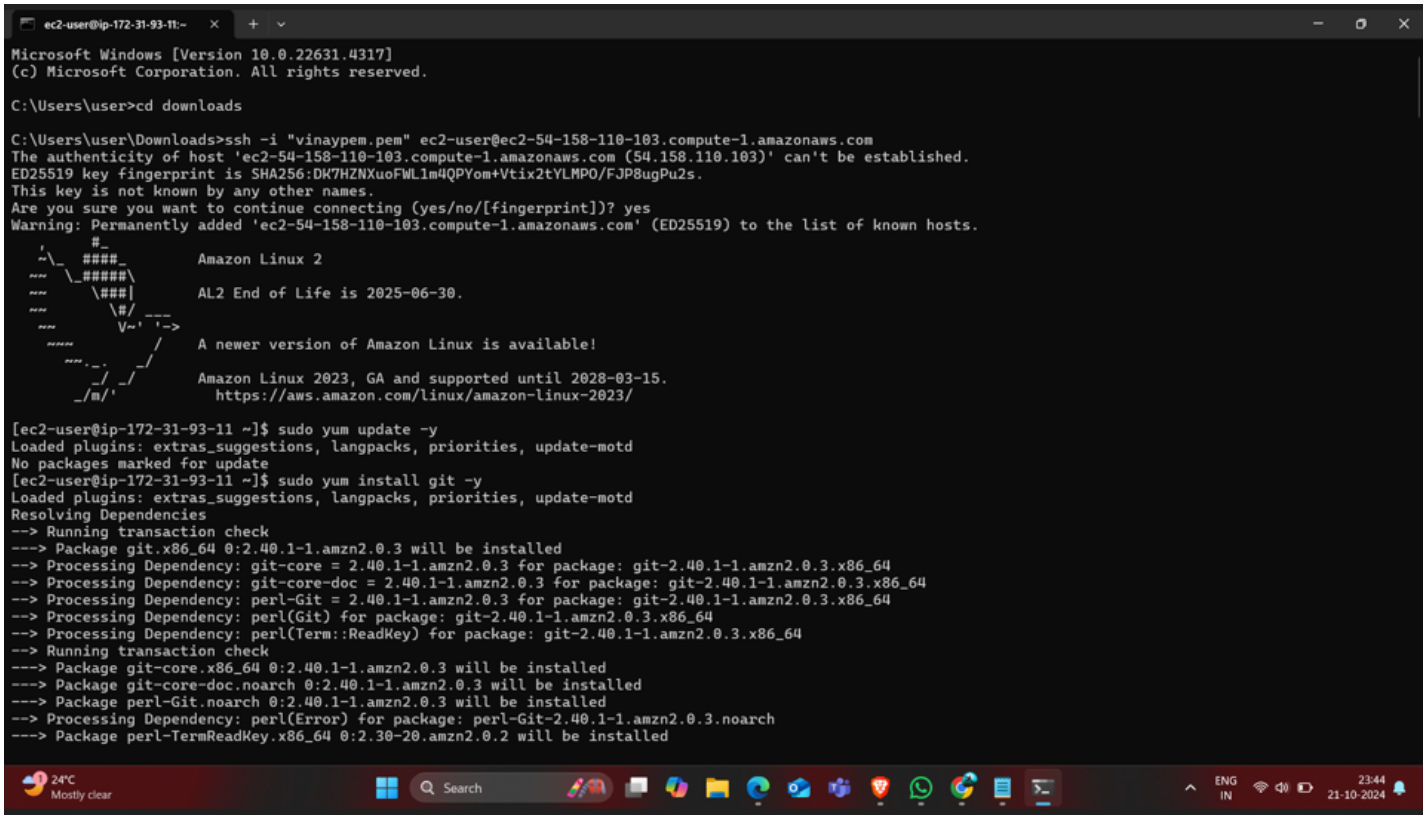
Deploy WordPress web application by using git (clone terraform script which helps to deploy WordPress web application), jenkins (in execute shell install terraform, init, fmt, validate and apply with automatic command as terraform apply --auto-approve) and terraform.

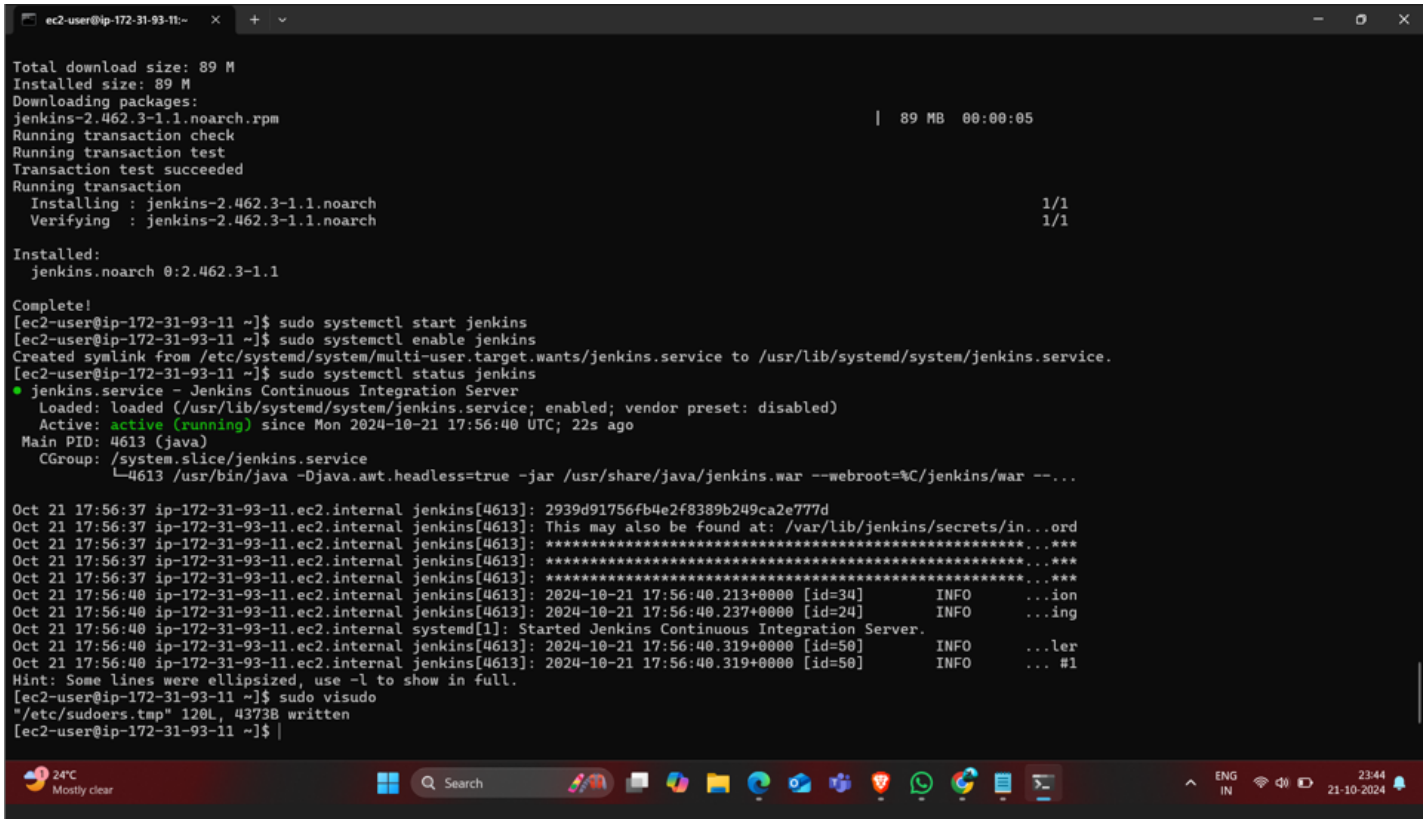
- Launch instance with t2.medium
- Allow security groups ssh.http and all traffic.
- Connect it to the terminal.



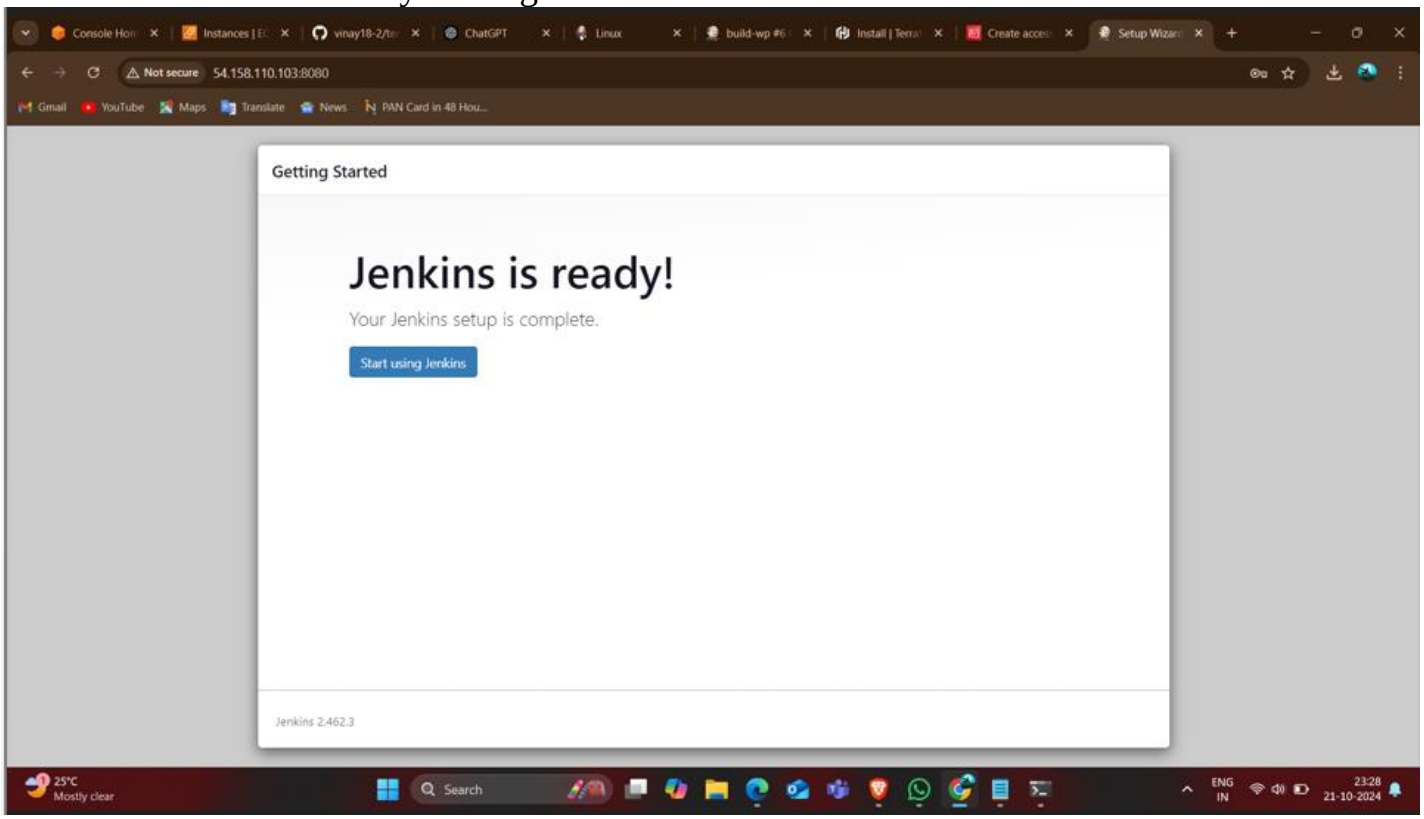
- After connected to the terminal update it.
- Install git.
- Install Jenkins.
- Start and enable the Jenkins and check the status of the Jenkins.

•

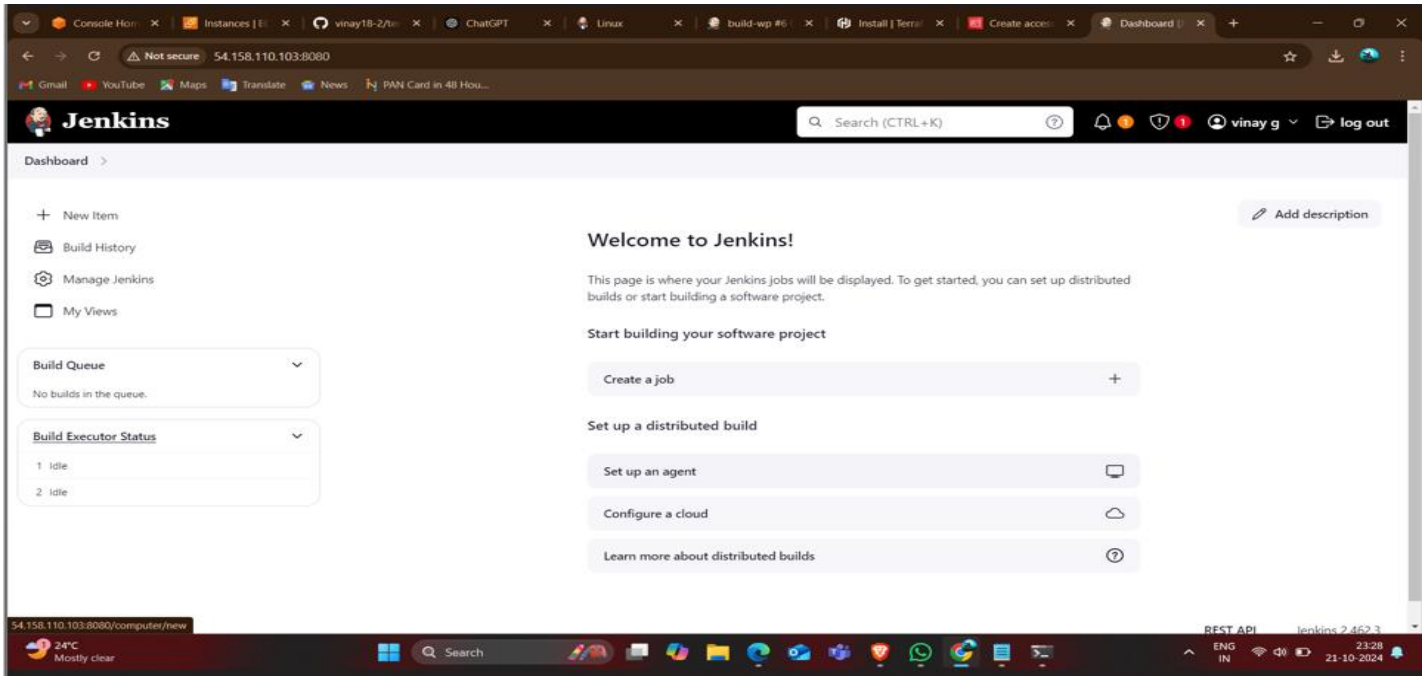

ec2-user@ip-172-31-93-11:~
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.
C:\Users\user>cd downloads
C:\Users\user\Downloads>ssh -i "vinaypem.pem" ec2-user@ec2-54-158-110-103.compute-1.amazonaws.com
The authenticity of host 'ec2-54-158-110-103.compute-1.amazonaws.com (54.158.110.103)' can't be established.
ED25519 key fingerprint is SHA256:DK7HZNXuoFWL1m4QPYom+Vtix2tYLMPO/FJP8ugPu2s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-158-110-103.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Amazon Linux 2
AL2 End of Life is 2025-06-30.
A newer version of Amazon Linux is available!
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-93-11 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-93-11 ~]$ sudo yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Running transaction check
---> Package git-core.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
---> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.3 will be installed
---> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.3.noarch
---> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
24°C
Mostly clear
Search
ENG
IN
23:44
21-10-2024


ec2-user@ip-172-31-93-11:~
Total download size: 89 M
Installed size: 89 M
Downloading packages:
jenkins-2.462.3-1.1.noarch.rpm | 89 MB 00:00:05
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : jenkins-2.462.3-1.1.noarch 1/1
Verifying : jenkins-2.462.3-1.1.noarch 1/1
Installed:
jenkins.noarch 0:2.462.3-1.1
Complete!
[ec2-user@ip-172-31-93-11 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-93-11 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-93-11 ~]$ sudo systemctl status jenkins
jenkins.service - Jenkins Continuous Integration Server
Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
Active: active (running) since Mon 2024-10-21 17:56:40 UTC; 22s ago
Main PID: 4613 (java)
CGroup: /system.slice/jenkins.service
└4613 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --...
Oct 21 17:56:37 ip-172-31-93-11.ec2.internal jenkins[4613]: 2939d91756fb4e2f8389b249ca2e777d
Oct 21 17:56:37 ip-172-31-93-11.ec2.internal jenkins[4613]: This may also be found at: /var/lib/jenkins/secrets/in...ord
Oct 21 17:56:37 ip-172-31-93-11.ec2.internal jenkins[4613]: ...***
Oct 21 17:56:37 ip-172-31-93-11.ec2.internal jenkins[4613]: ...***
Oct 21 17:56:37 ip-172-31-93-11.ec2.internal jenkins[4613]: ...***
Oct 21 17:56:40 ip-172-31-93-11.ec2.internal jenkins[4613]: 2024-10-21 17:56:40.213+0000 [id=34] INFO ...ion
Oct 21 17:56:40 ip-172-31-93-11.ec2.internal jenkins[4613]: 2024-10-21 17:56:40.237+0000 [id=24] INFO ...ing
Oct 21 17:56:40 ip-172-31-93-11.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Oct 21 17:56:40 ip-172-31-93-11.ec2.internal jenkins[4613]: 2024-10-21 17:56:40.319+0000 [id=50] INFO ...ler
Oct 21 17:56:40 ip-172-31-93-11.ec2.internal jenkins[4613]: 2024-10-21 17:56:40.319+0000 [id=50] INFO ... #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-93-11 ~]$ sudo visudo
"/etc/sudoers.tmp" 120L, 4373B written
[ec2-user@ip-172-31-93-11 ~]$
24°C
Mostly clear
Search
ENG
IN
23:44
21-10-2024

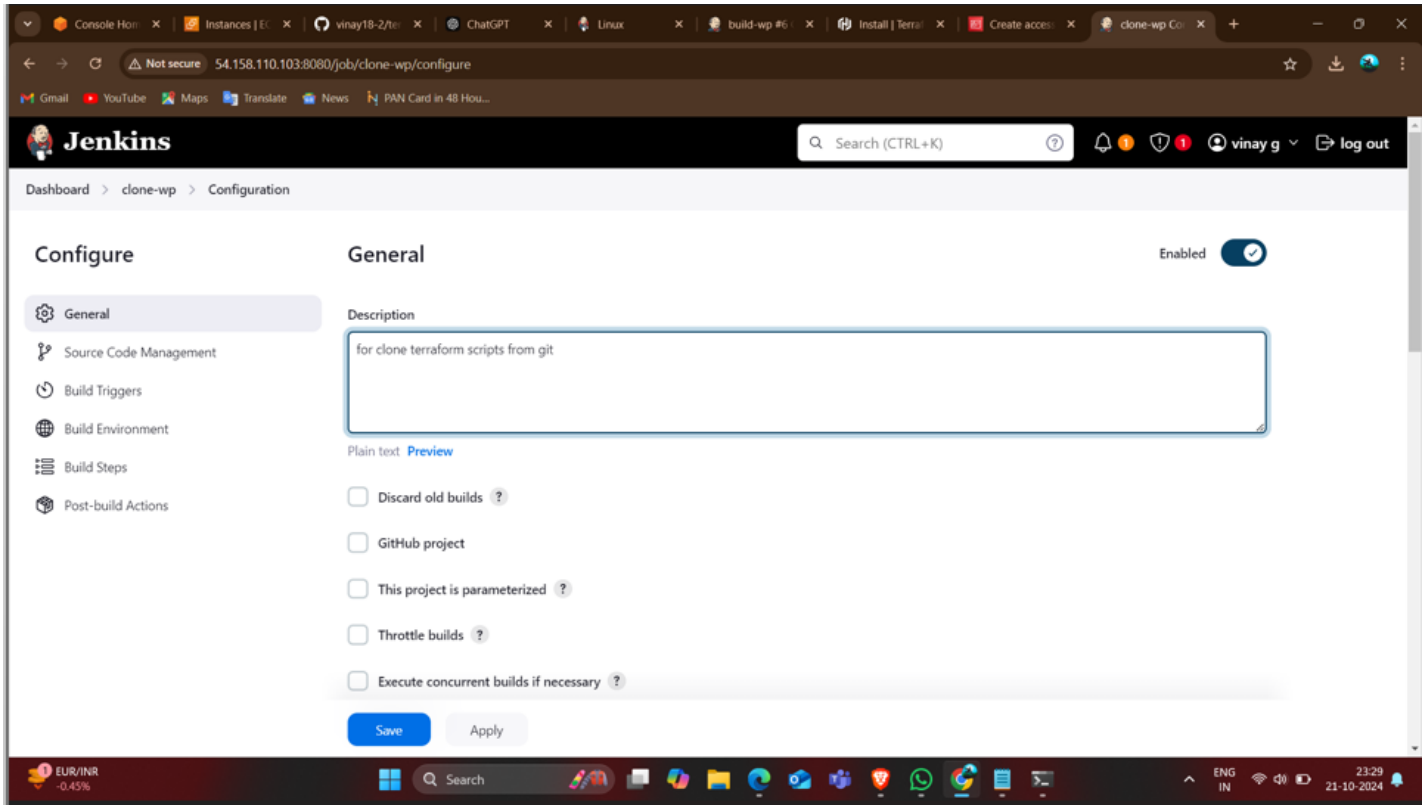
- Here Jenkins is ready and login to the Jenkins dashboard.

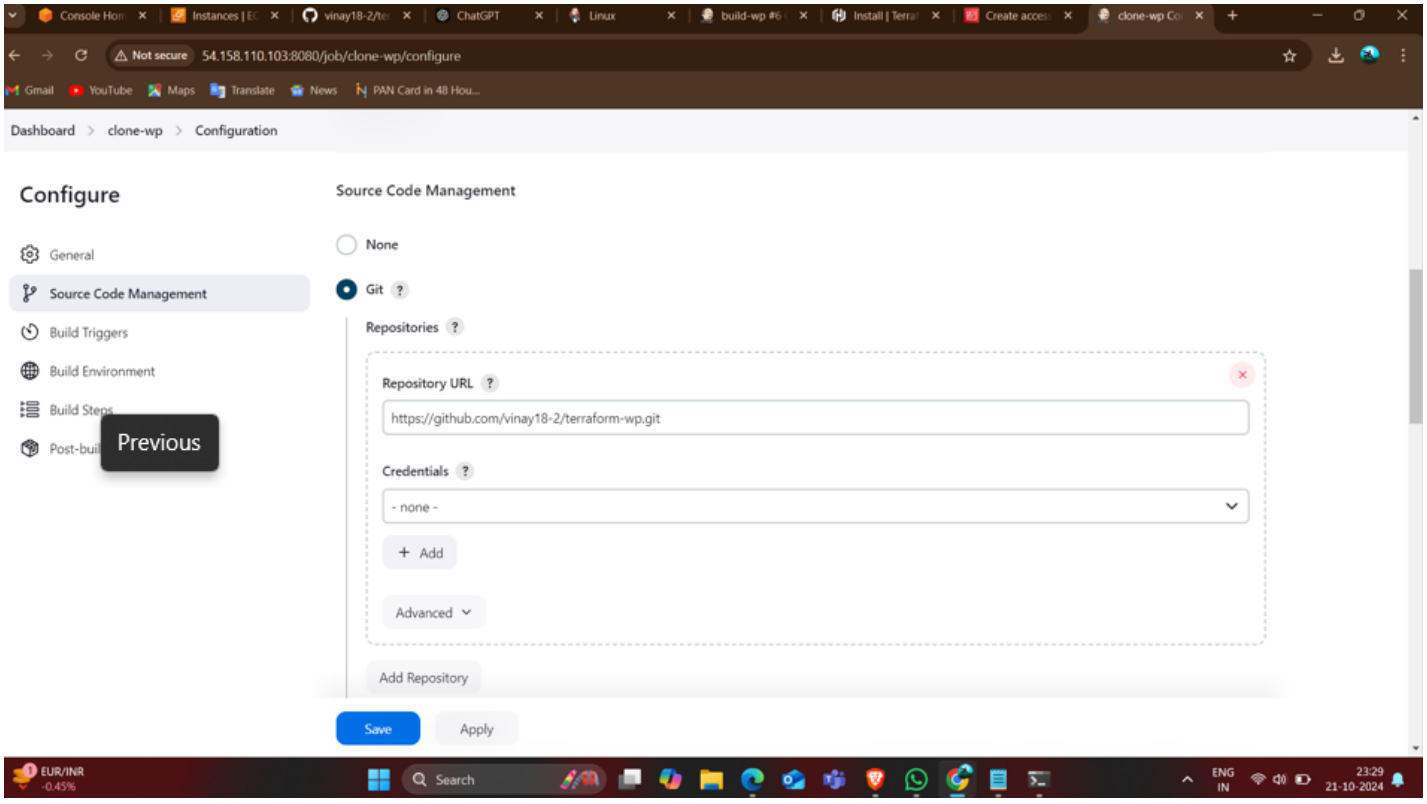


- Create the clone job to clone the repo from the git and follow the below steps to complete the clone job.

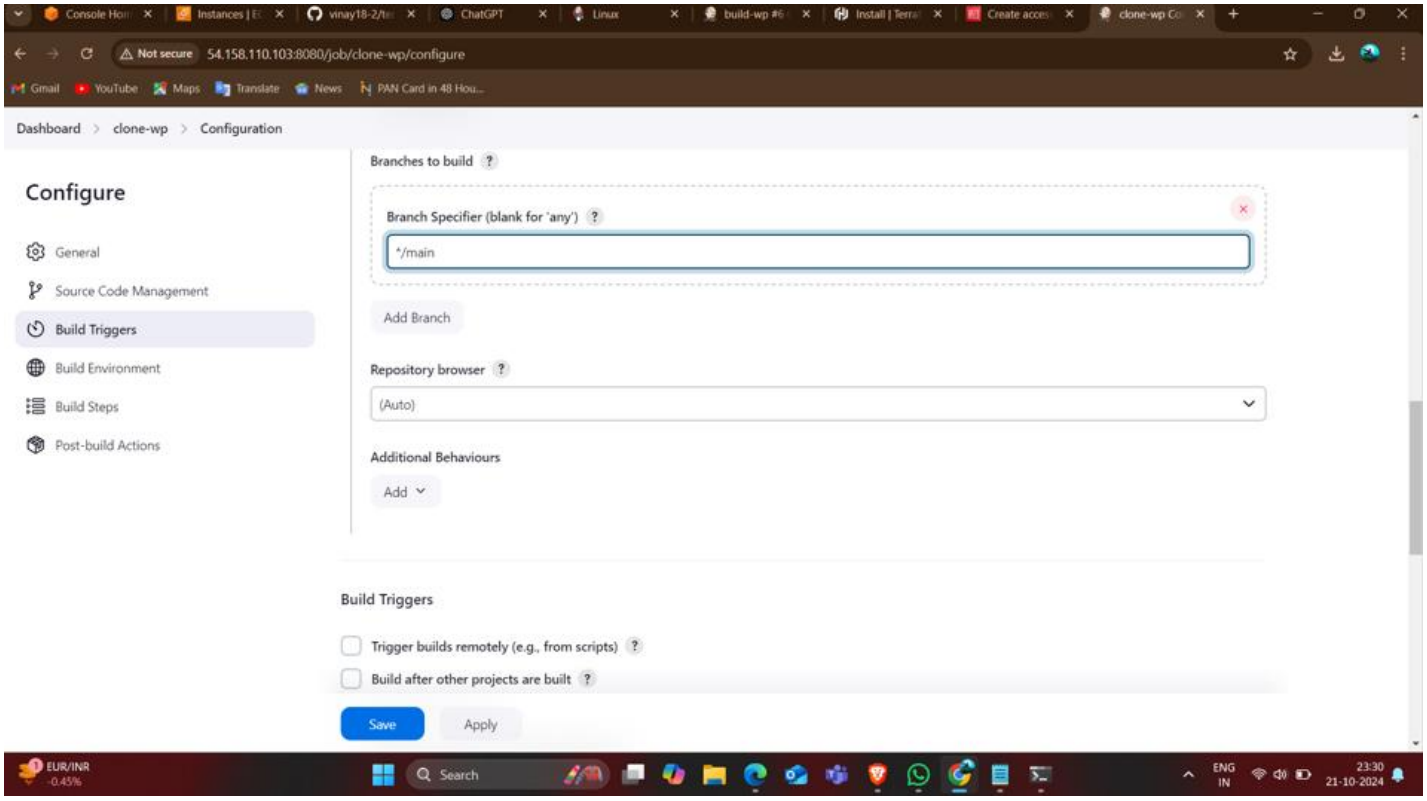


•

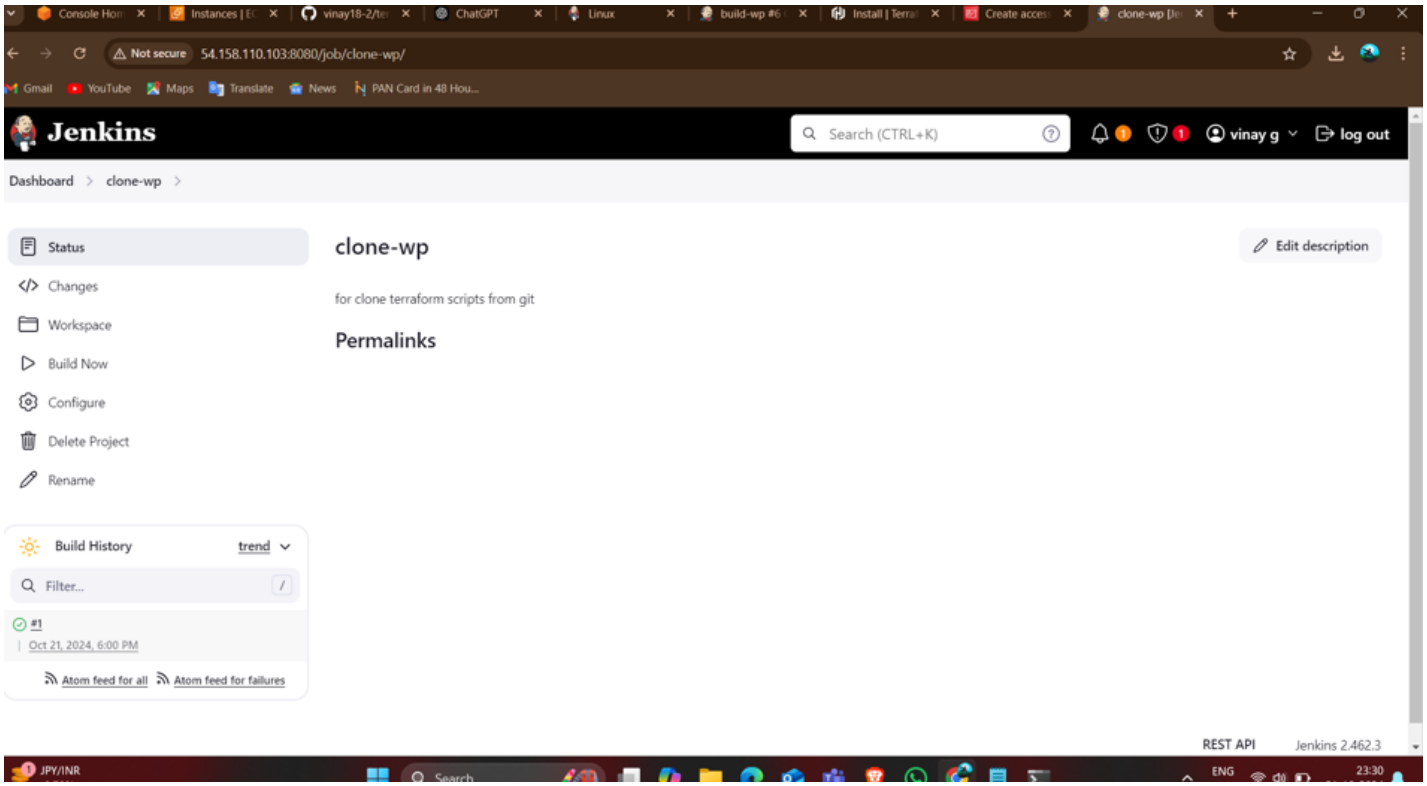

Jenkins
Search (CTRL+K)
vinay g
log out
Dashboard > clone-wp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
for clone terraform scripts from git
Plain text Preview
Discard old builds
GitHub project
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Save
Apply


Dashboard > clone-wp > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Previous
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/vinay18-2/terraform-wp.git
Credentials
- none -
+ Add
Advanced
Add Repository
Save
Apply

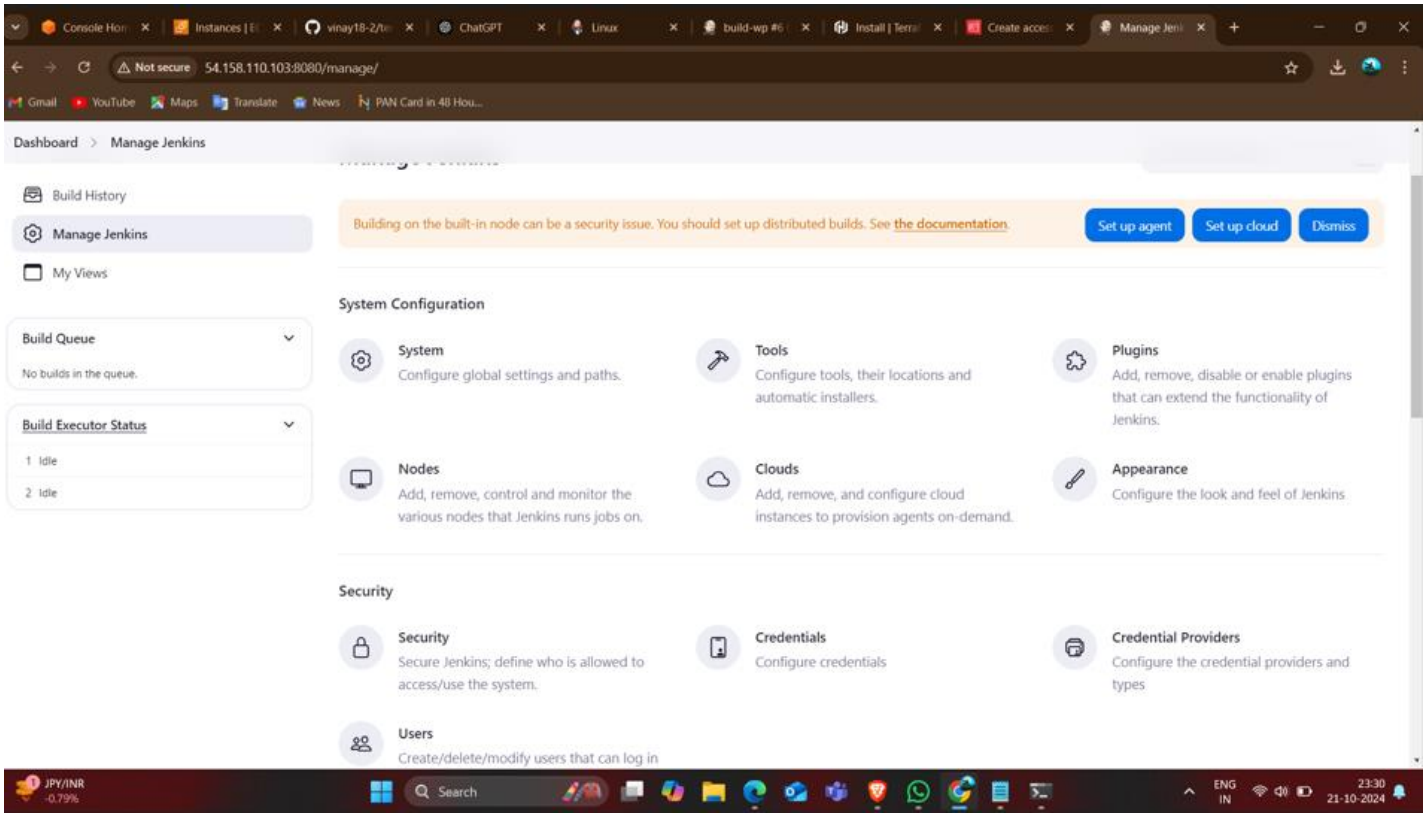
•

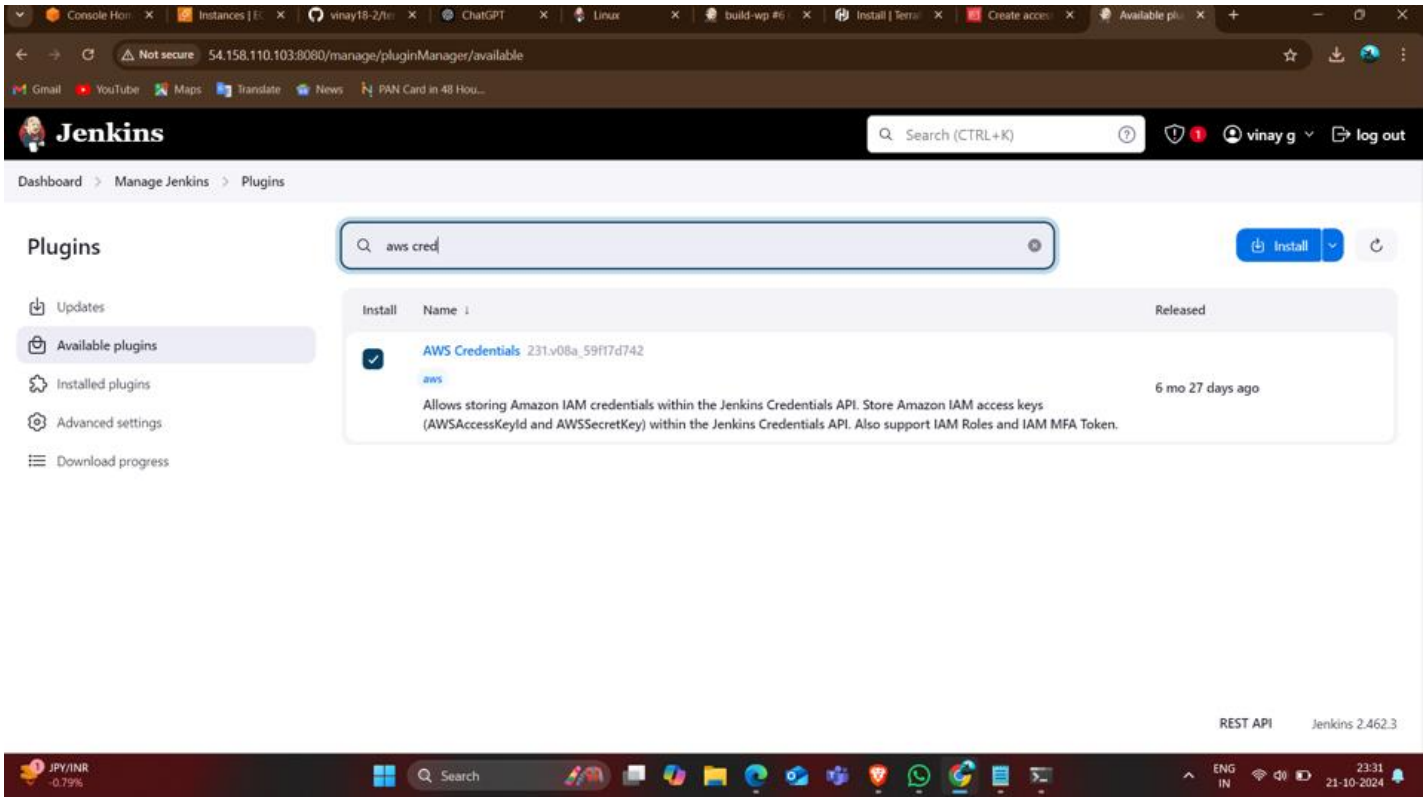


- After save and apply than to run select the build now.

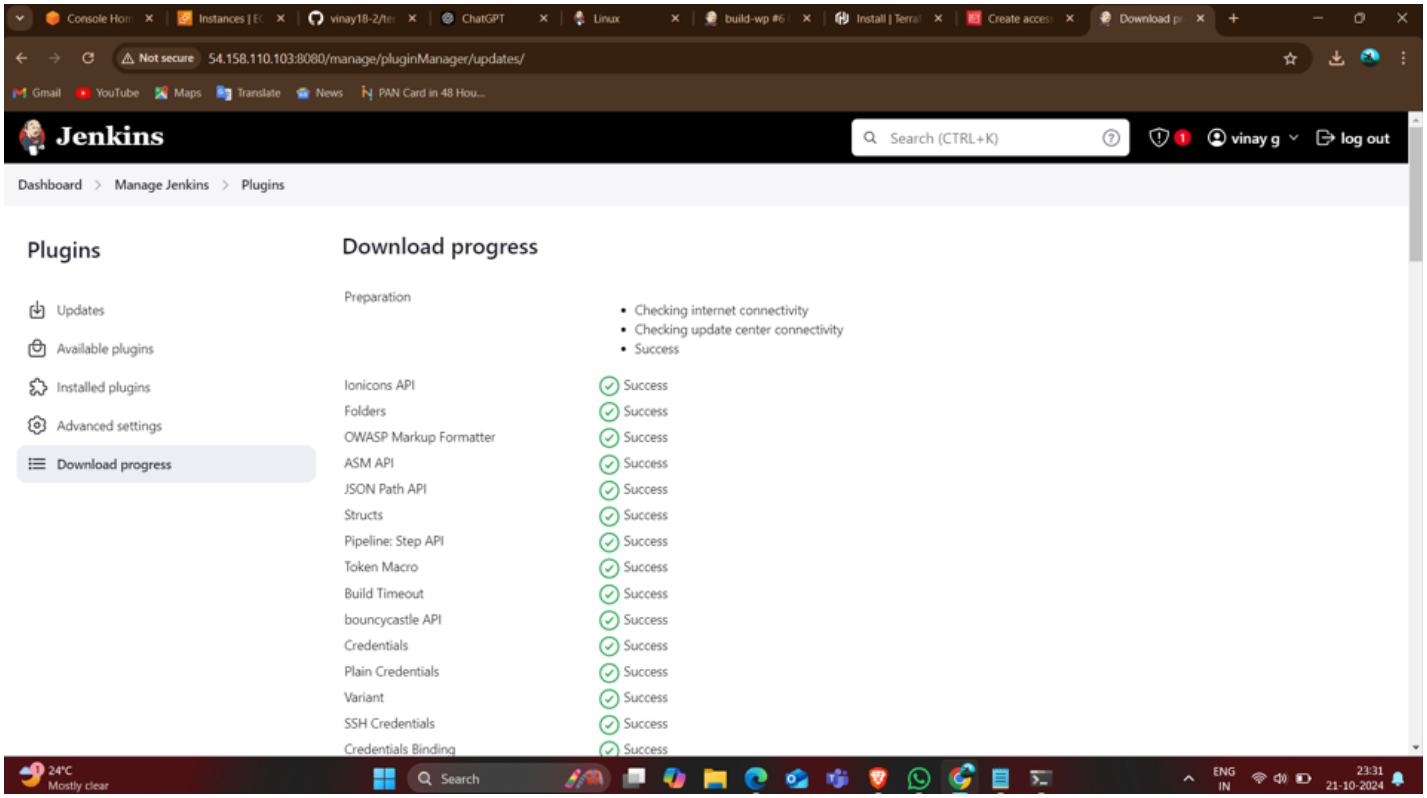


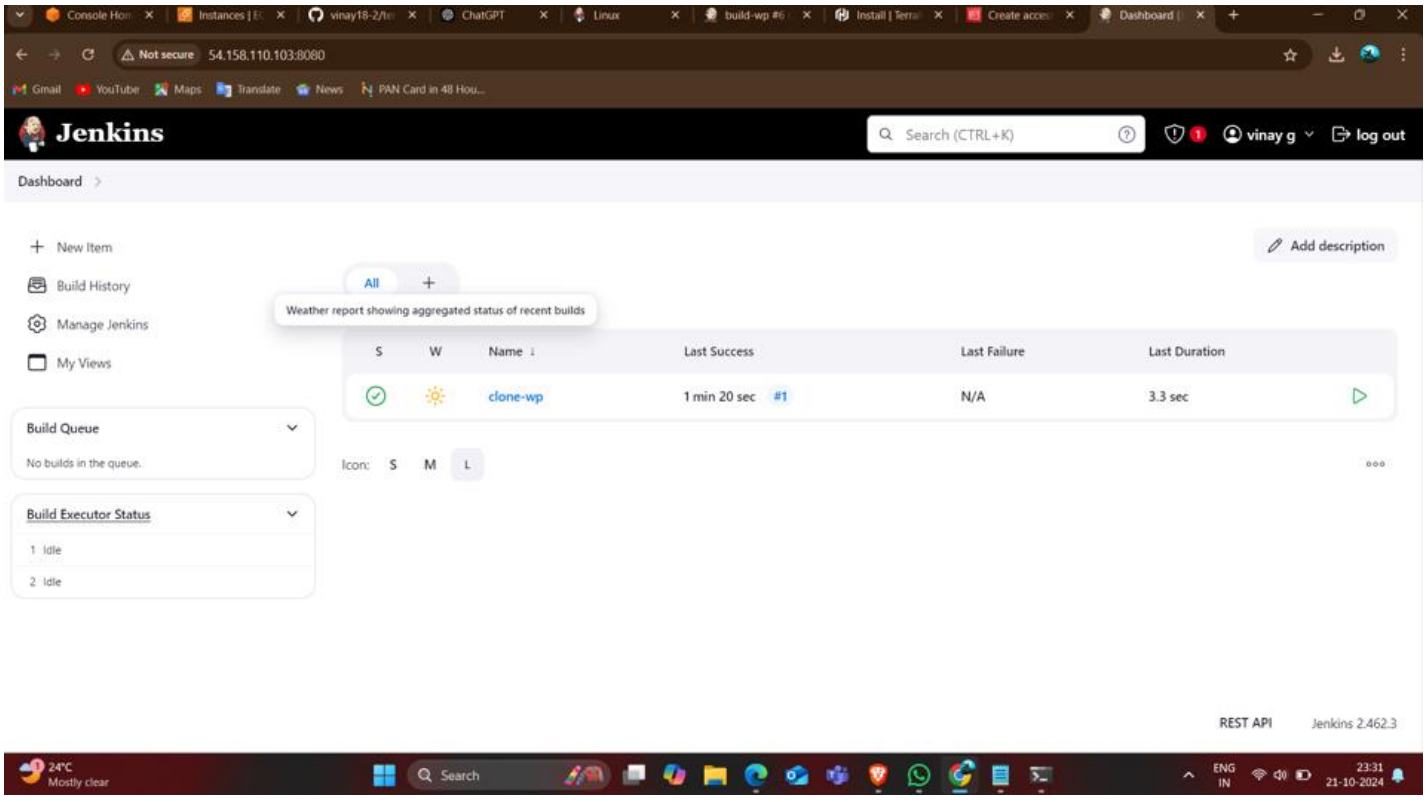
-
  - Than install AWS credentials plugin to store the aws access keys and secret keys.





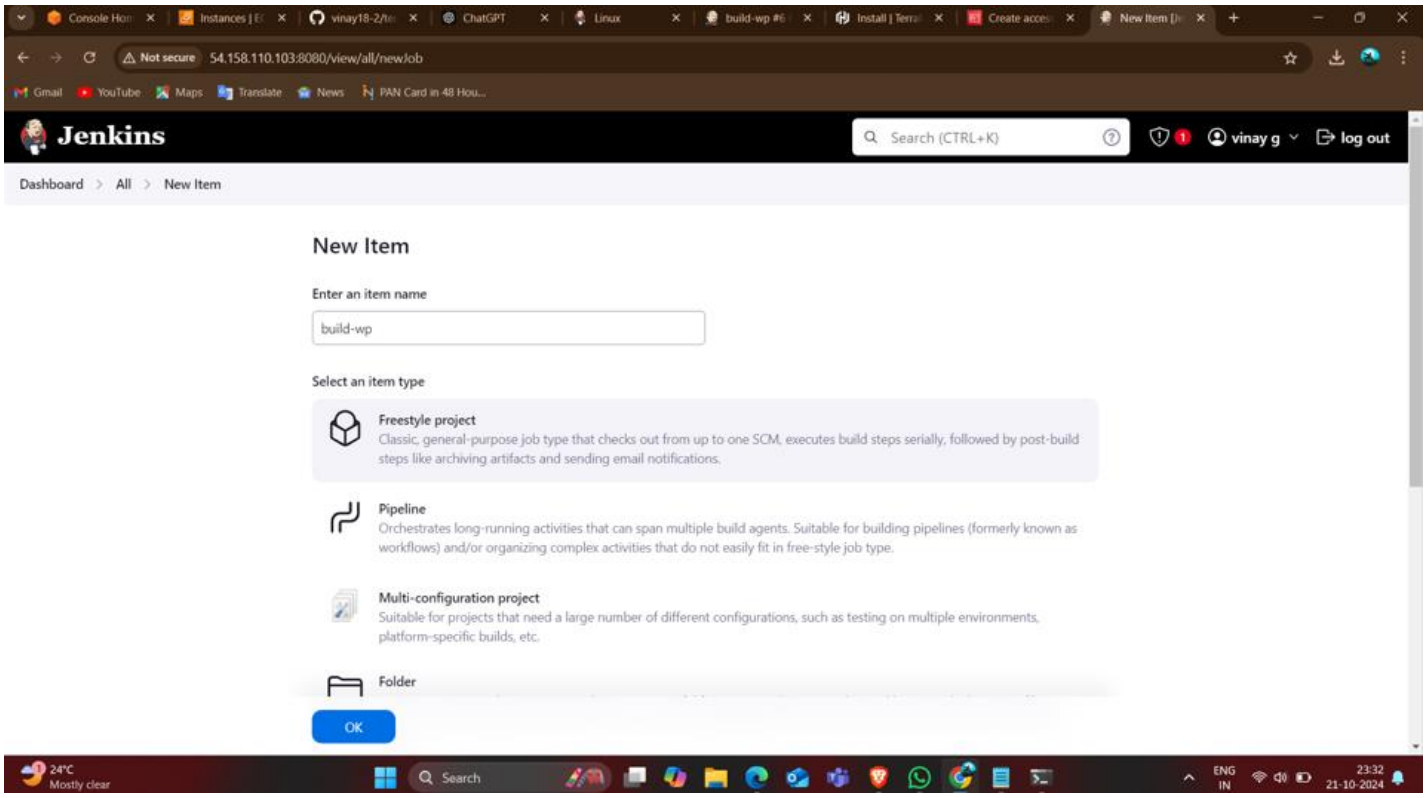
•

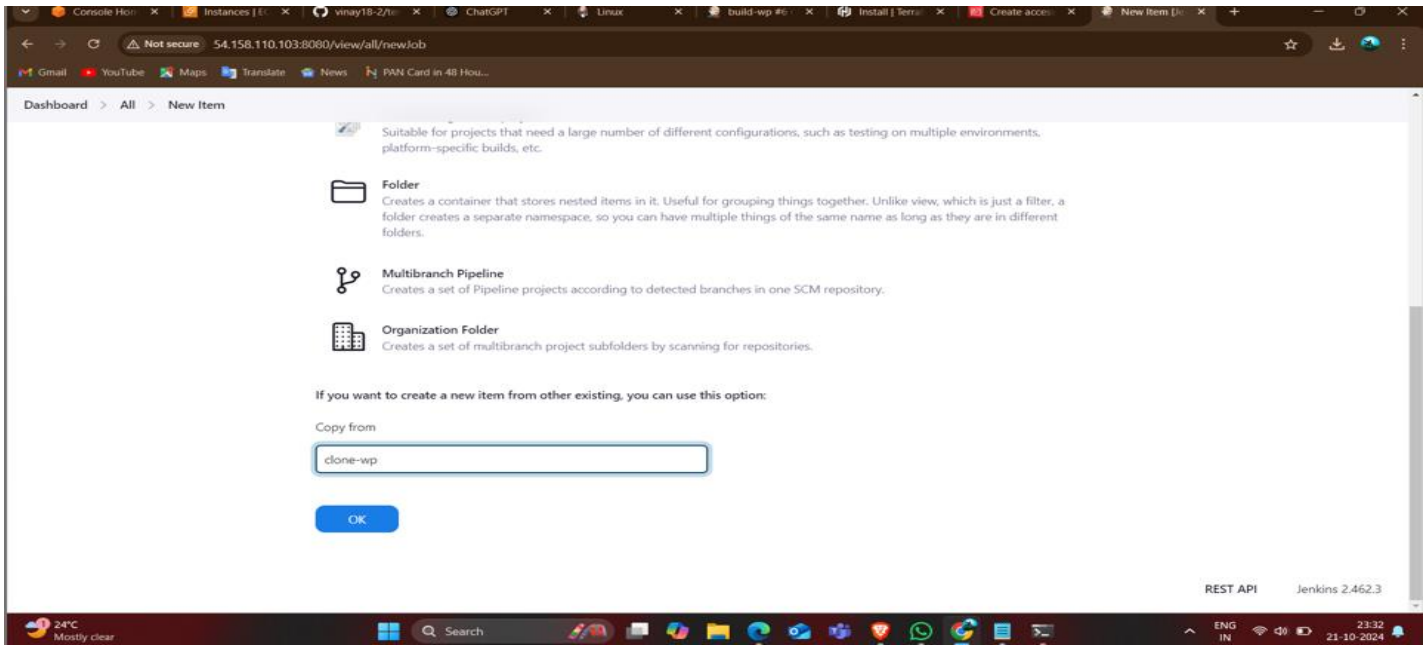

Not secure 54.158.110.103:8080/manage/pluginManager/updates/
Jenkins
Search (CTRL+K)
vinay g
log out
Dashboard > Manage Jenkins > Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Download progress
Preparation
Checking internet connectivity
Checking update 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 54.158.110.103:8080
Jenkins
Search (CTRL+K)
vinay g
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Add description
All
Weather report showing aggregated status of recent builds
S
W
Name
Last Success
Last Failure
Last Duration
clone-wp
1 min 20 sec #1
N/A
3.3 sec
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Icon: S M L
REST API
Jenkins 2.462.3

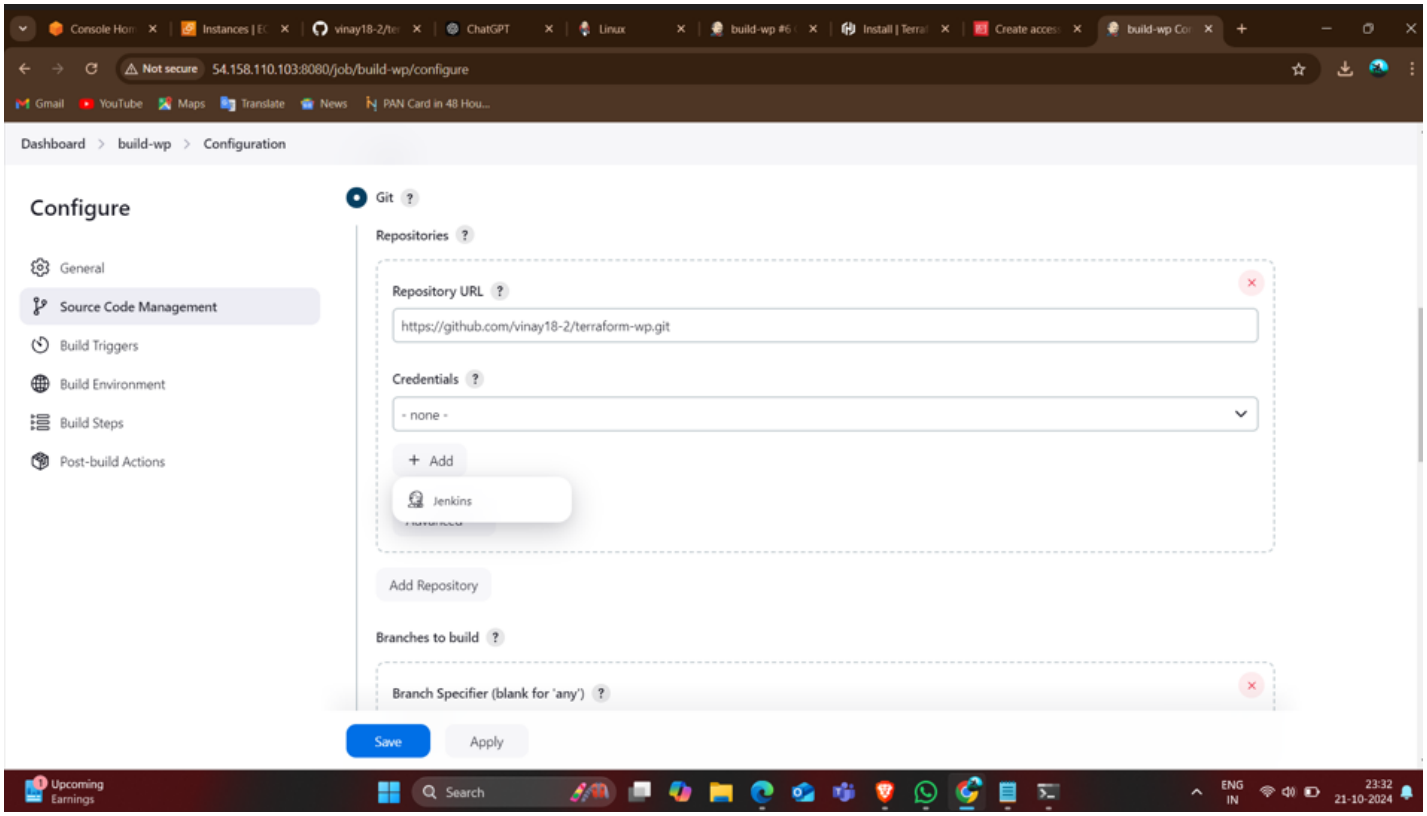
-

  - After completion of clone job than we shoul create new project for build job from copying of existing project i.e clone job.

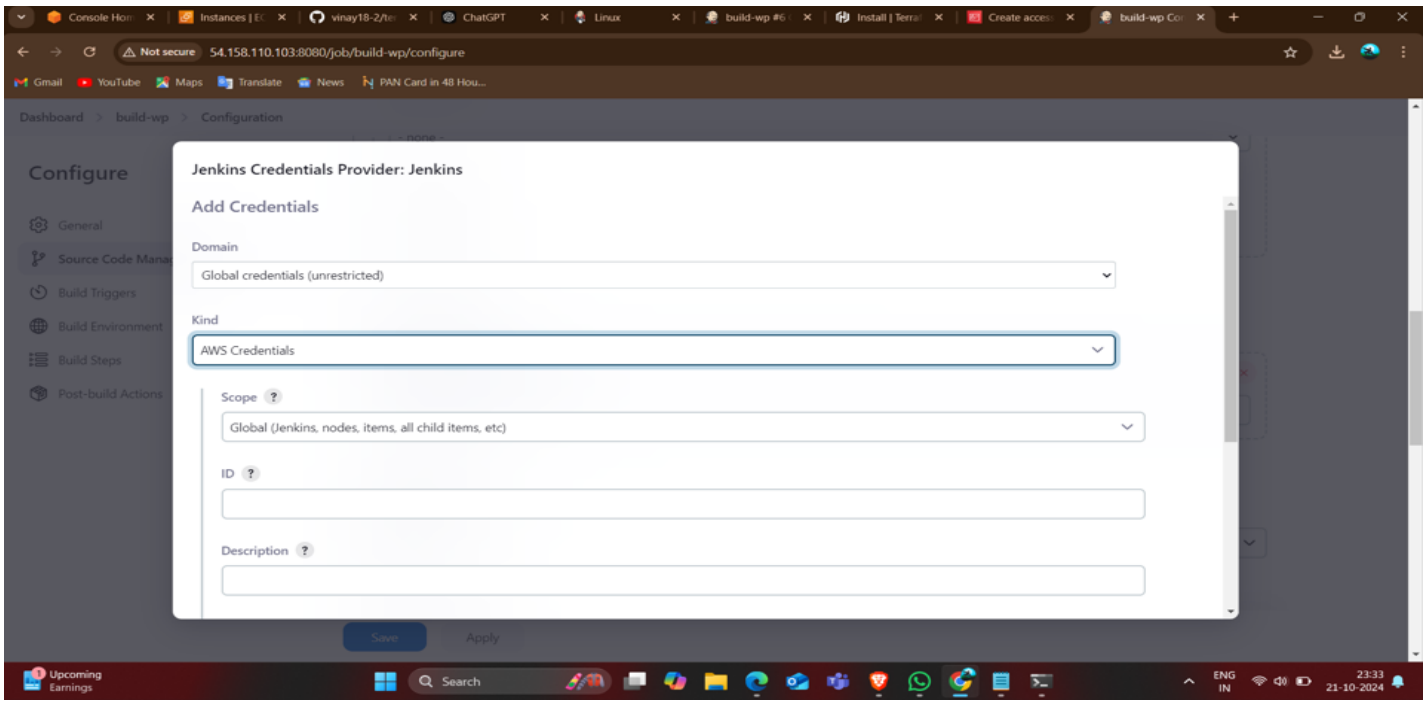




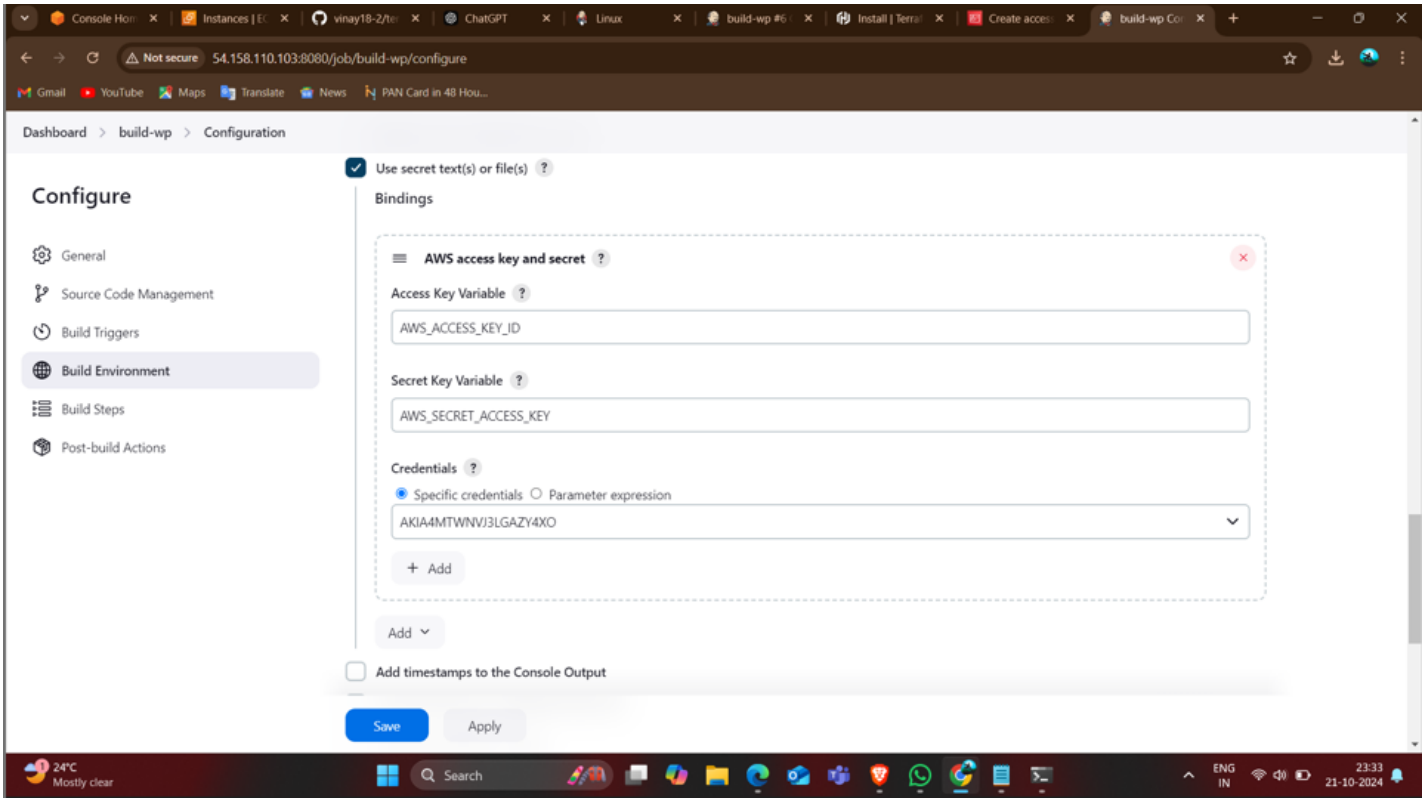
-
  - After providing the repo link than select the Credentials.



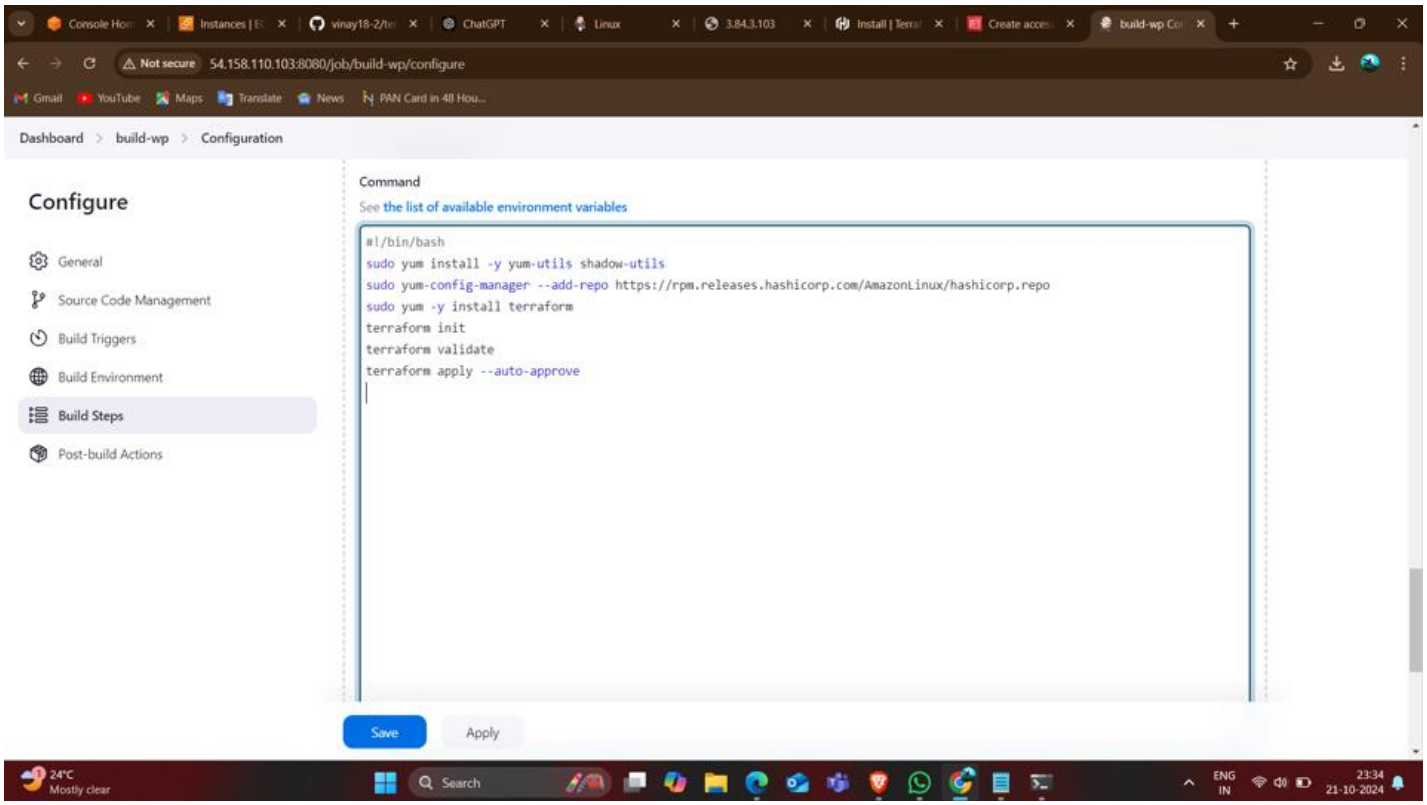
  - Add Jenkins credentials provider,than select tha kind AWS credentials. Add the access keys and secret keys.



•

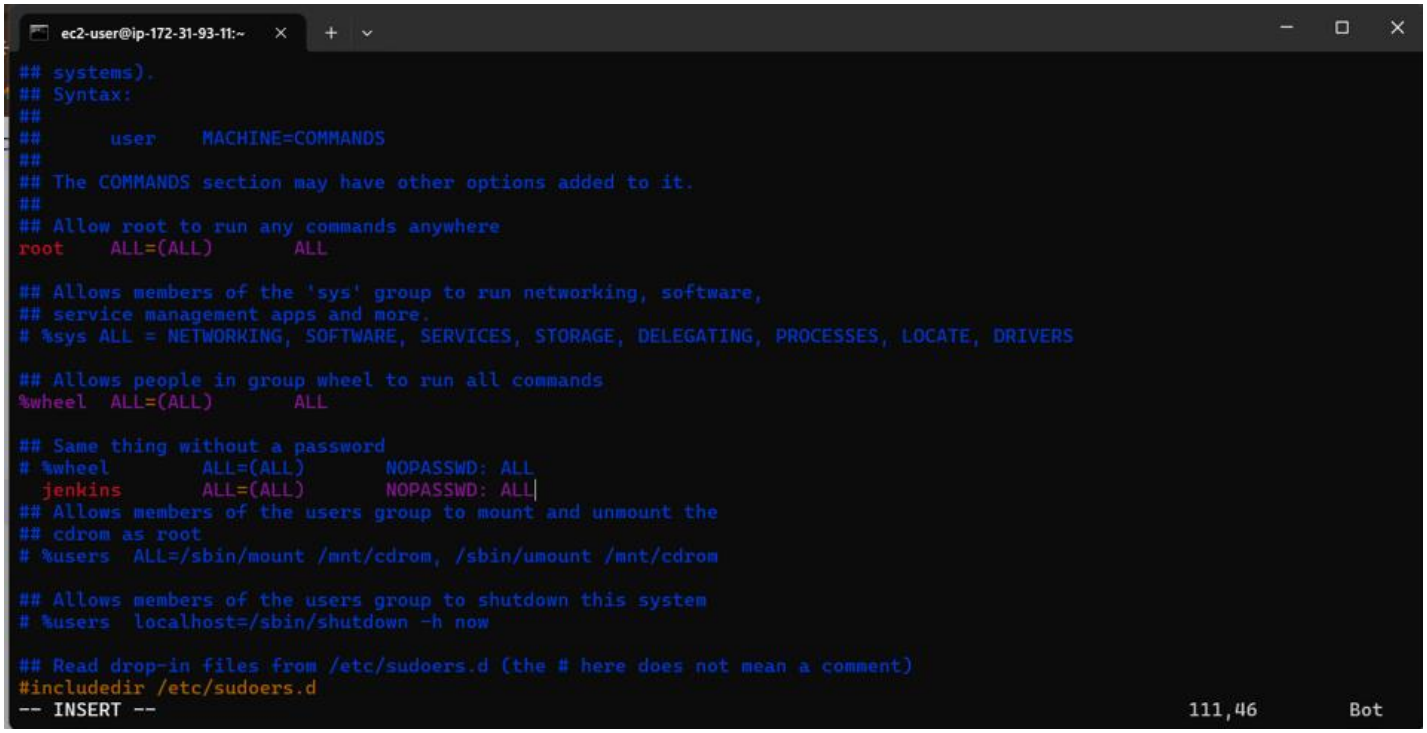


- Provide the terraform commands and installation commands in execute shell and save it and apply.

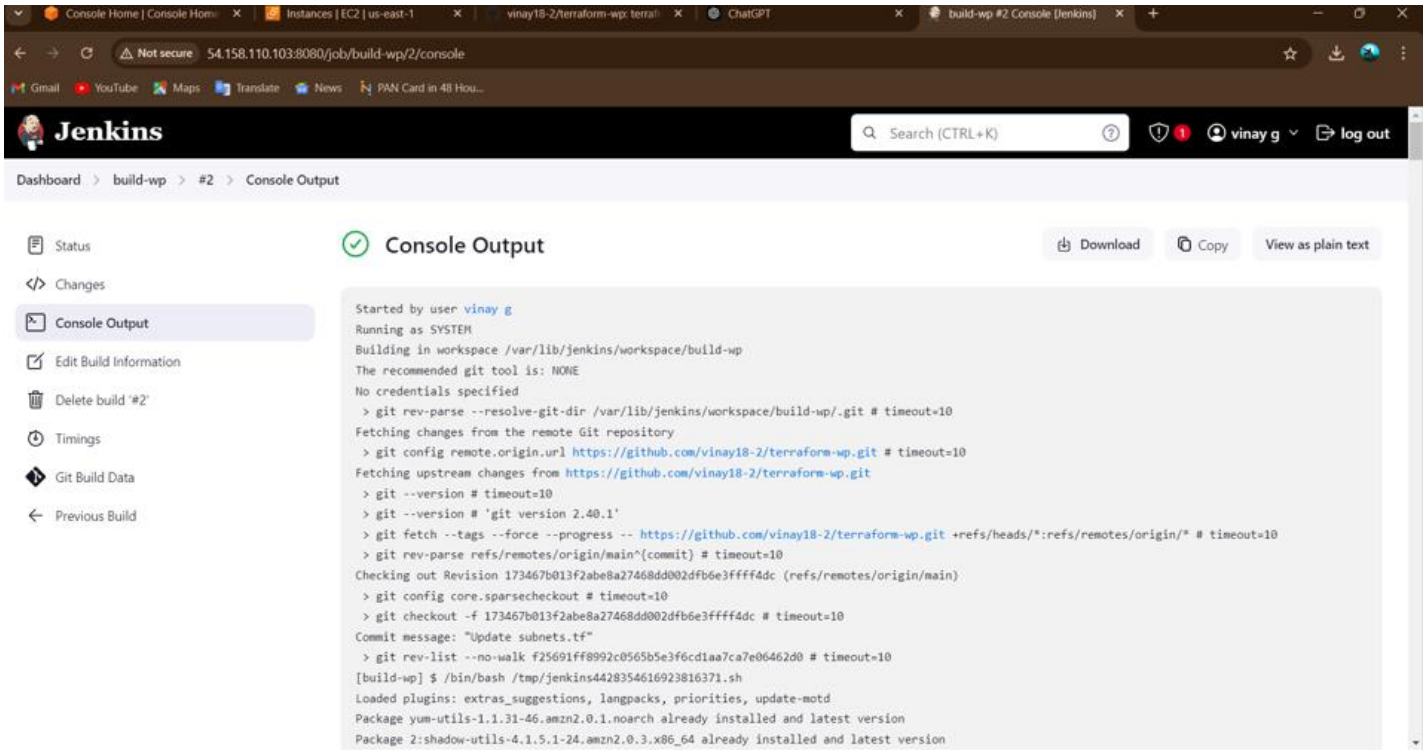


-

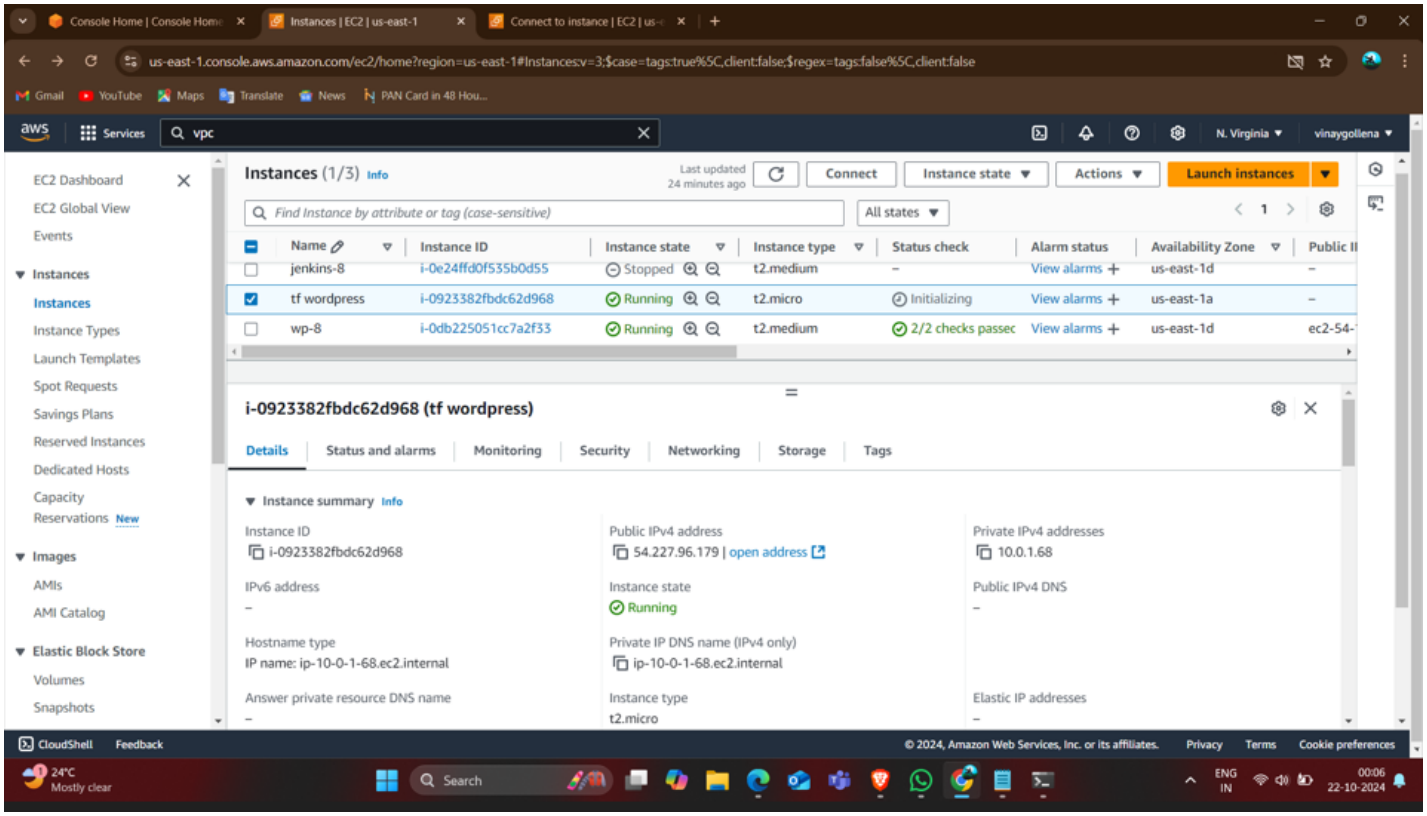
  - To prevent the errors we have to edit the visudo file.



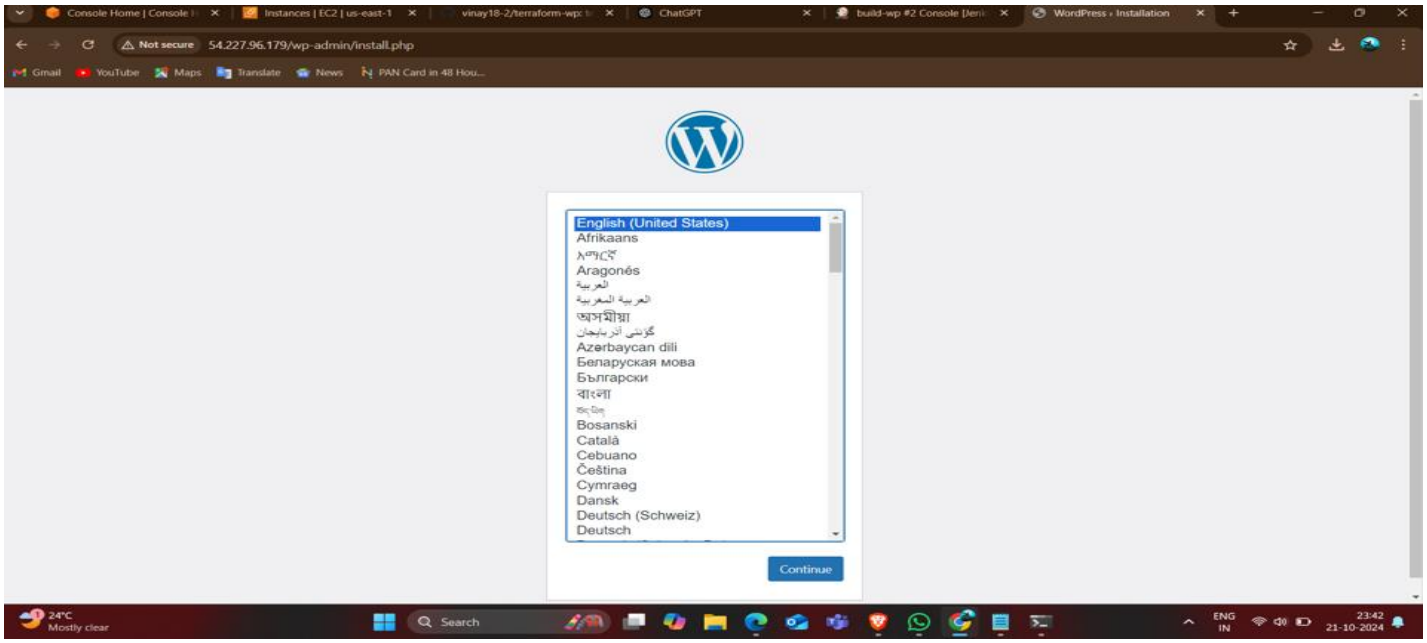
  - Observe the console output after completion of build.we can get to know the build is executed or get any errors and where the errors have been occurred.



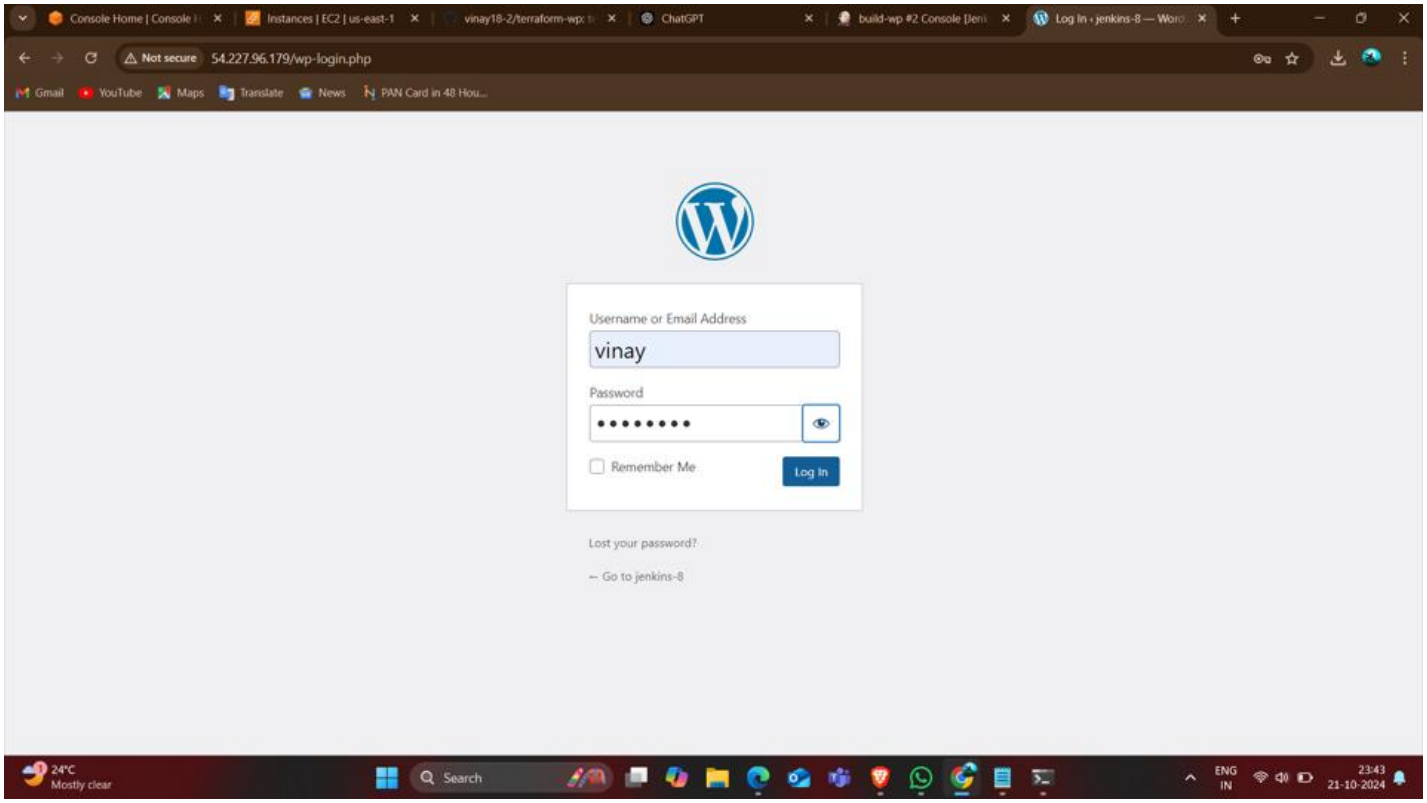
- After executed than observe in the aws console that the services we are provided in the git has been cloned and created with the help of Jenkins or not.
- Here we can observe that the services has been created and we can observe the output by copying the ip address of created instance.

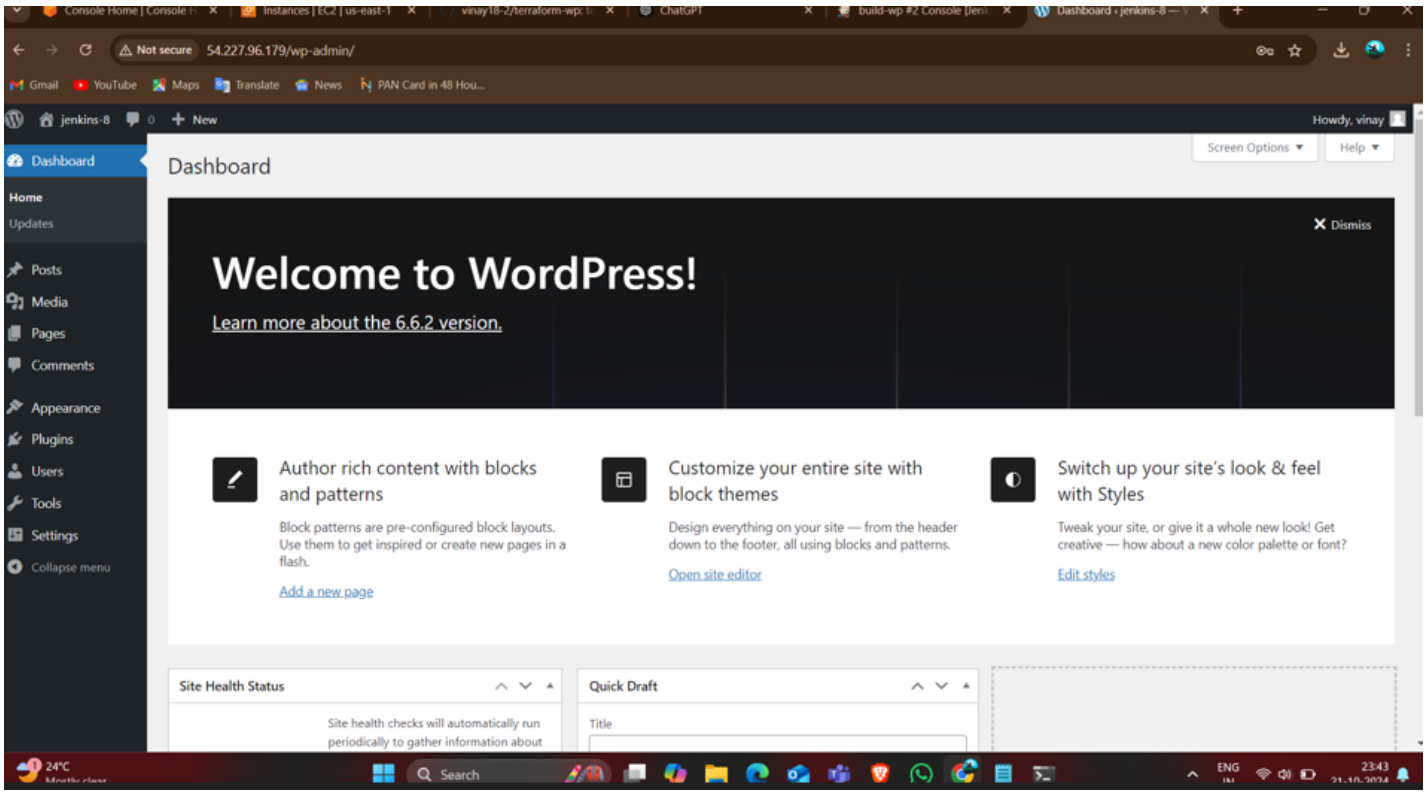


- Here the output.



•


Not secure 54.227.96.179/wp-login.php
Username or Email Address
vinay
Password
Remember Me
Log In
Lost your password?
← Go to jenkins-8

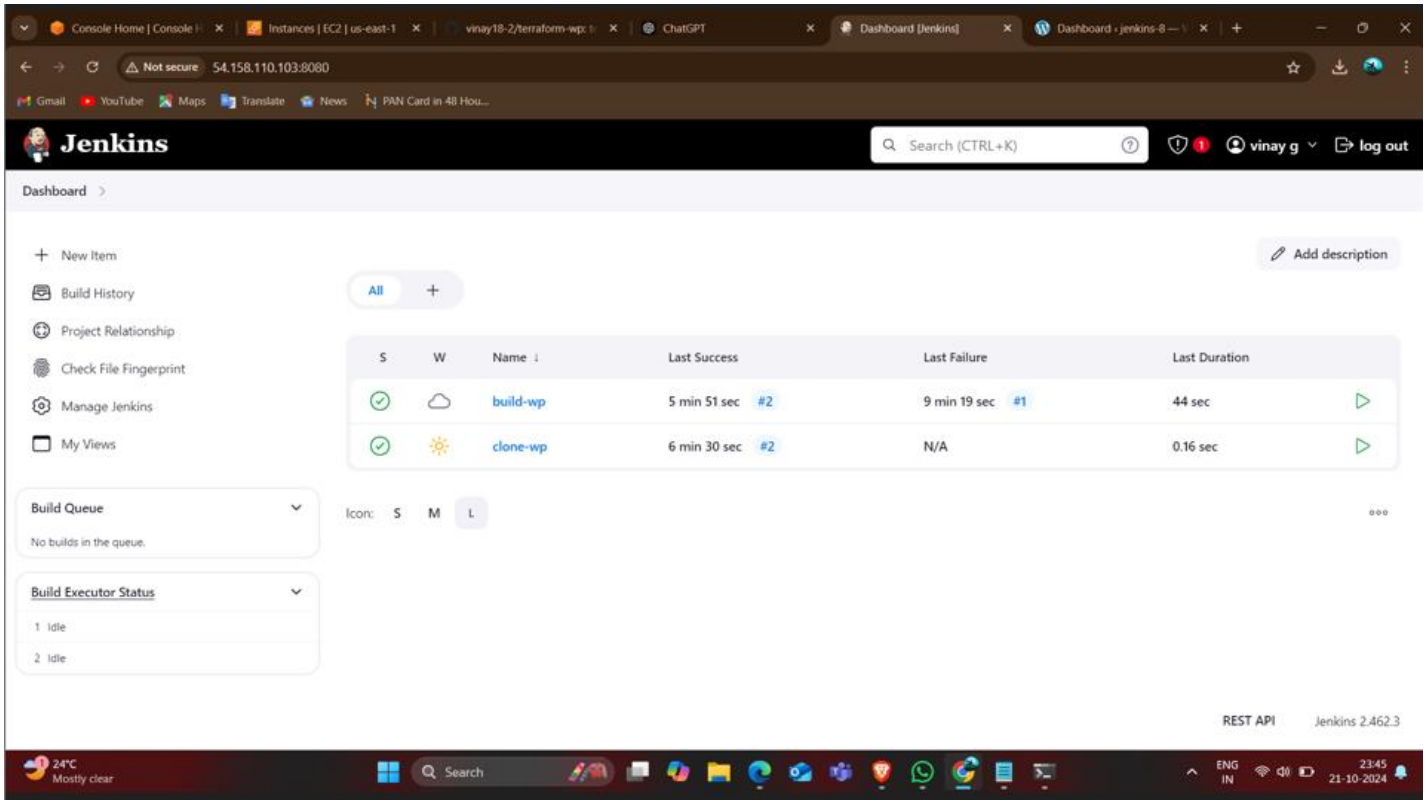

Not secure 54.227.96.179/wp-admin/
jenkins-8
New
Howdy, vinay
Dashboard
Screen Options
Help
Home
Updates
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Collapse menu
Welcome to WordPress!
Learn more about the 6.6.2 version.
Dismiss
Author rich content with blocks and patterns
Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.
Add a new page
Customize your entire site with block themes
Design everything on your site — from the header down to the footer, all using blocks and patterns.
Open site editor
Switch up your site's look & feel with Styles
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?
Edit styles
Site Health Status
Site health checks will automatically run periodically to gather information about
Quick Draft
Title

•

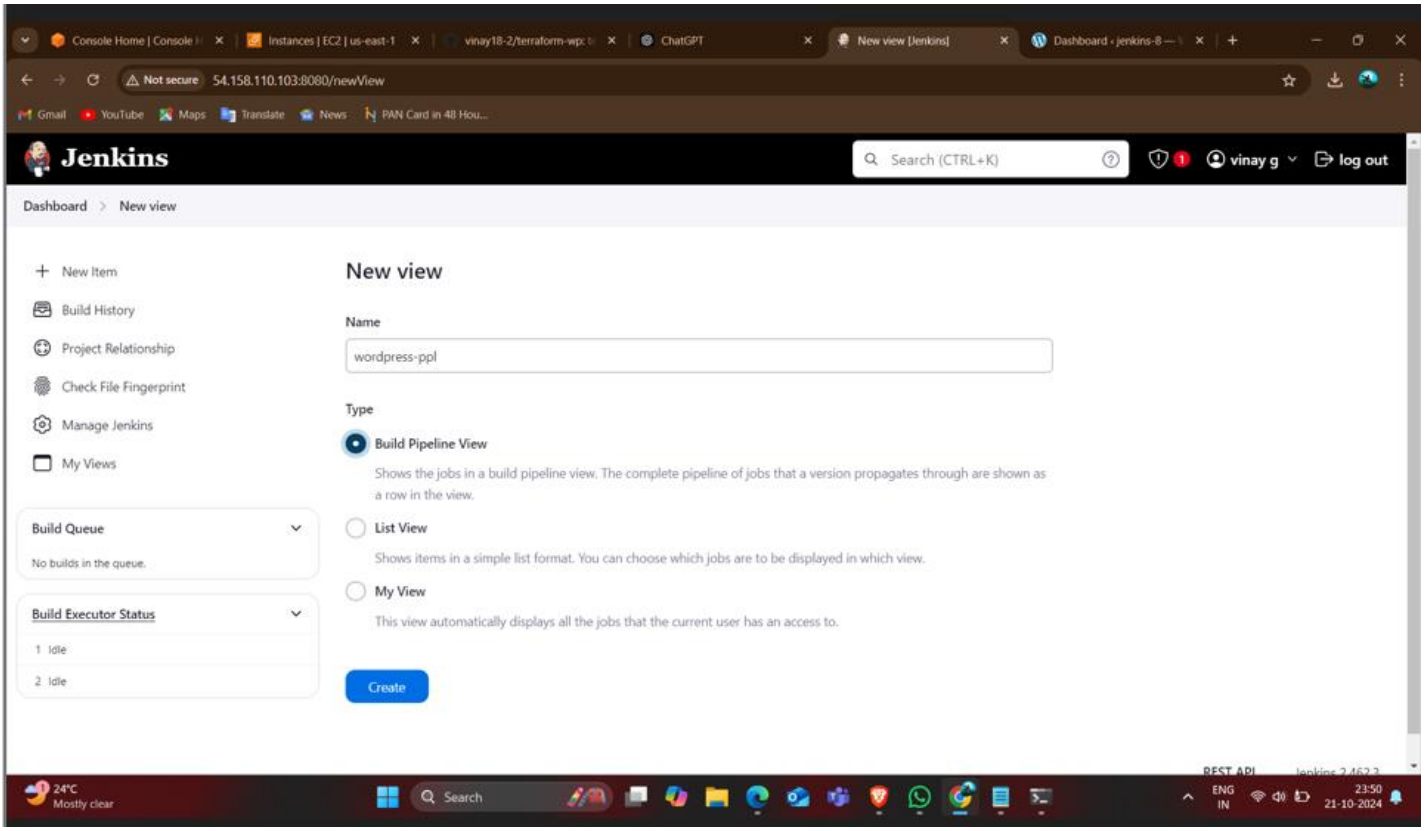
Deploy WordPress web application by using git (clone terraform script which helps to deploy WordPress web application), jenkins (in execute shell install terraform, init, fmt, validate and apply with automatic command as terraform apply --auto-approve) and terraform and create jenkins pipeline and add build periodically and poll scm to initial job of pipeline and check the changes happened or not which are made in github repo.

- For this method the procedure is the same as the above method.
- Launching the instance.
- Connect to the terminal.
- Update it.
- Install git.
- Install Jenkins and start Jenkins.
- Enable Jenkins.
- Check the status and login to the Jenkins.
- Create the clone job and build job.
- Than install the build pipeline plugin.
- Than connect bothe jobs through pipeline.

•

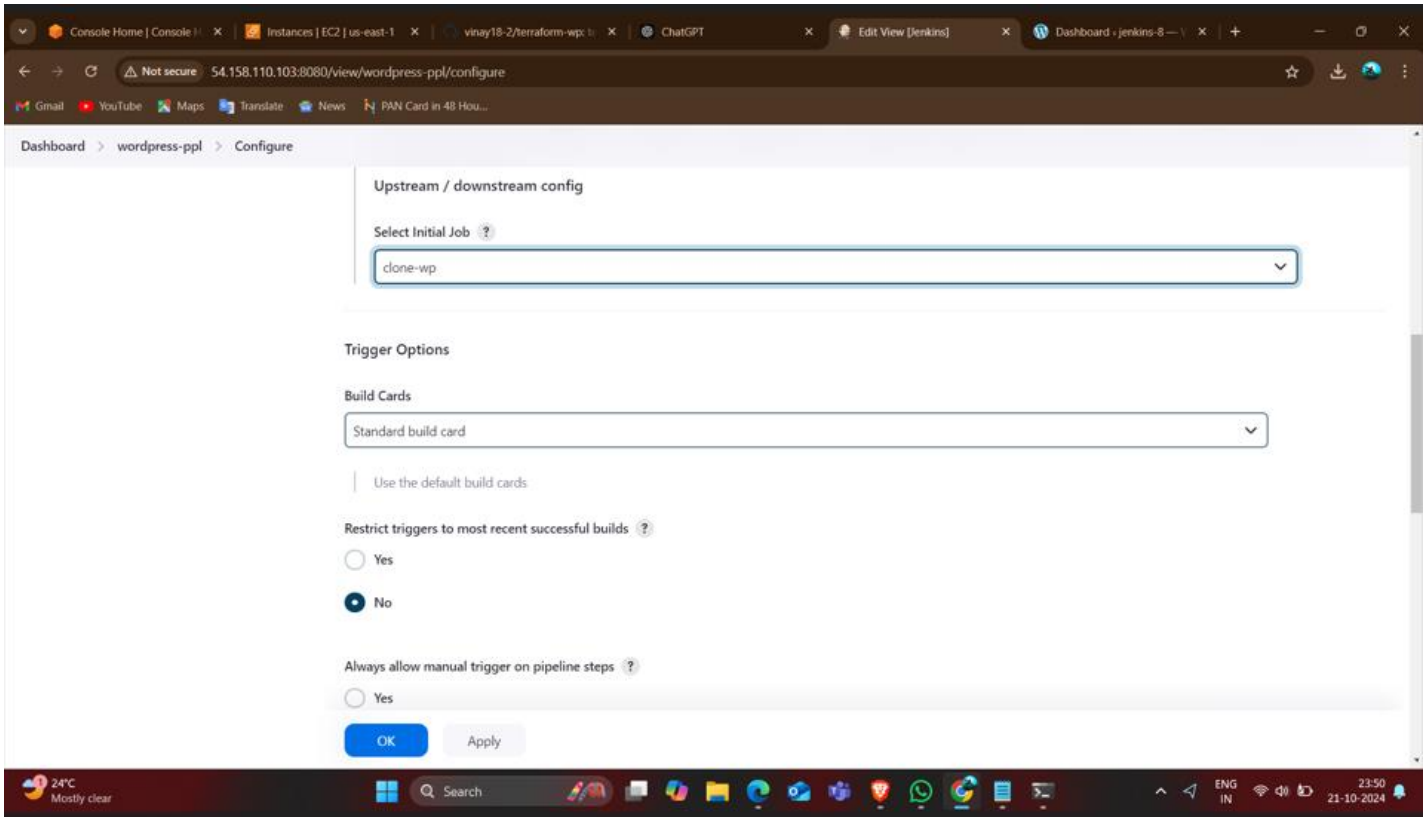


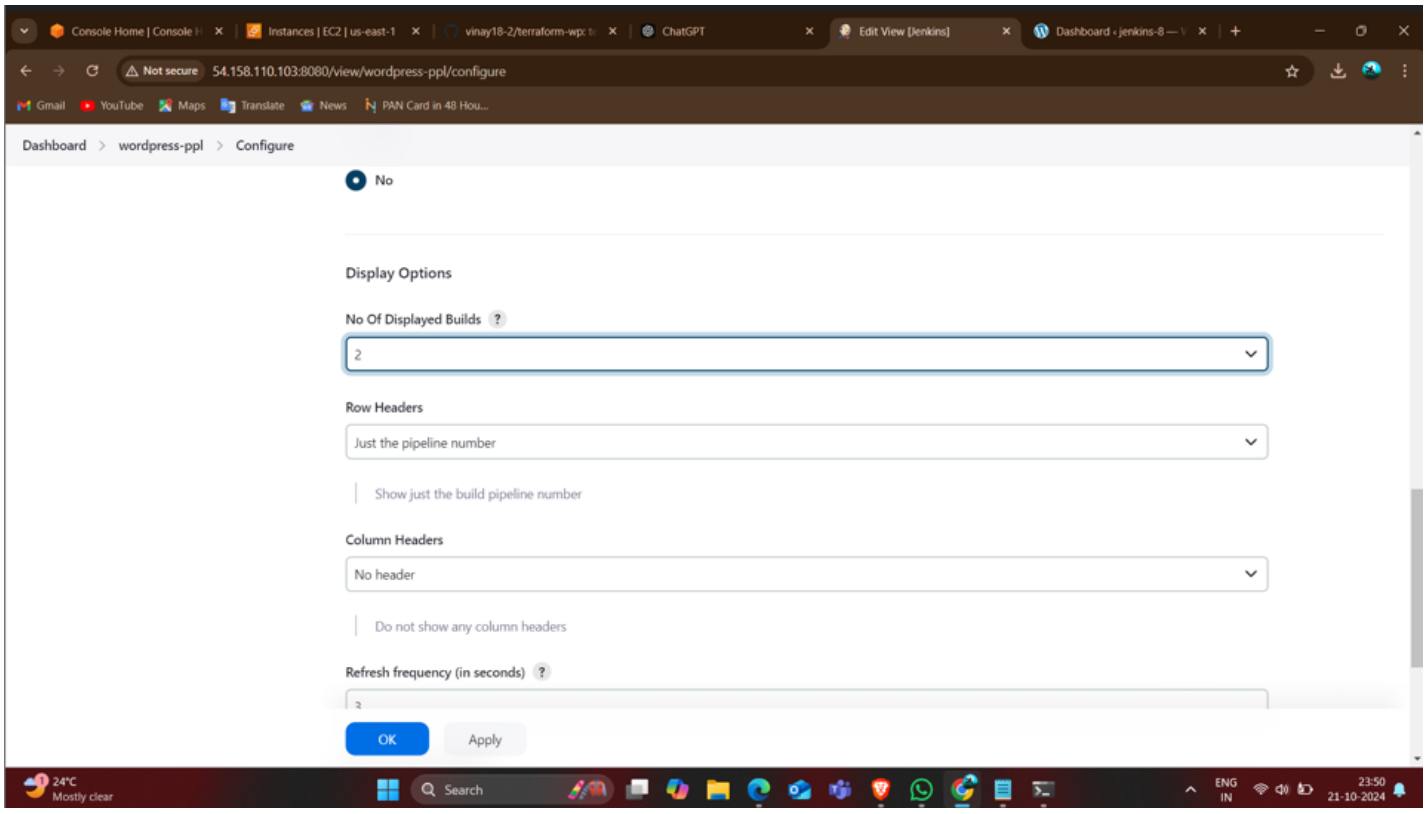
- Create a new pipeline view.



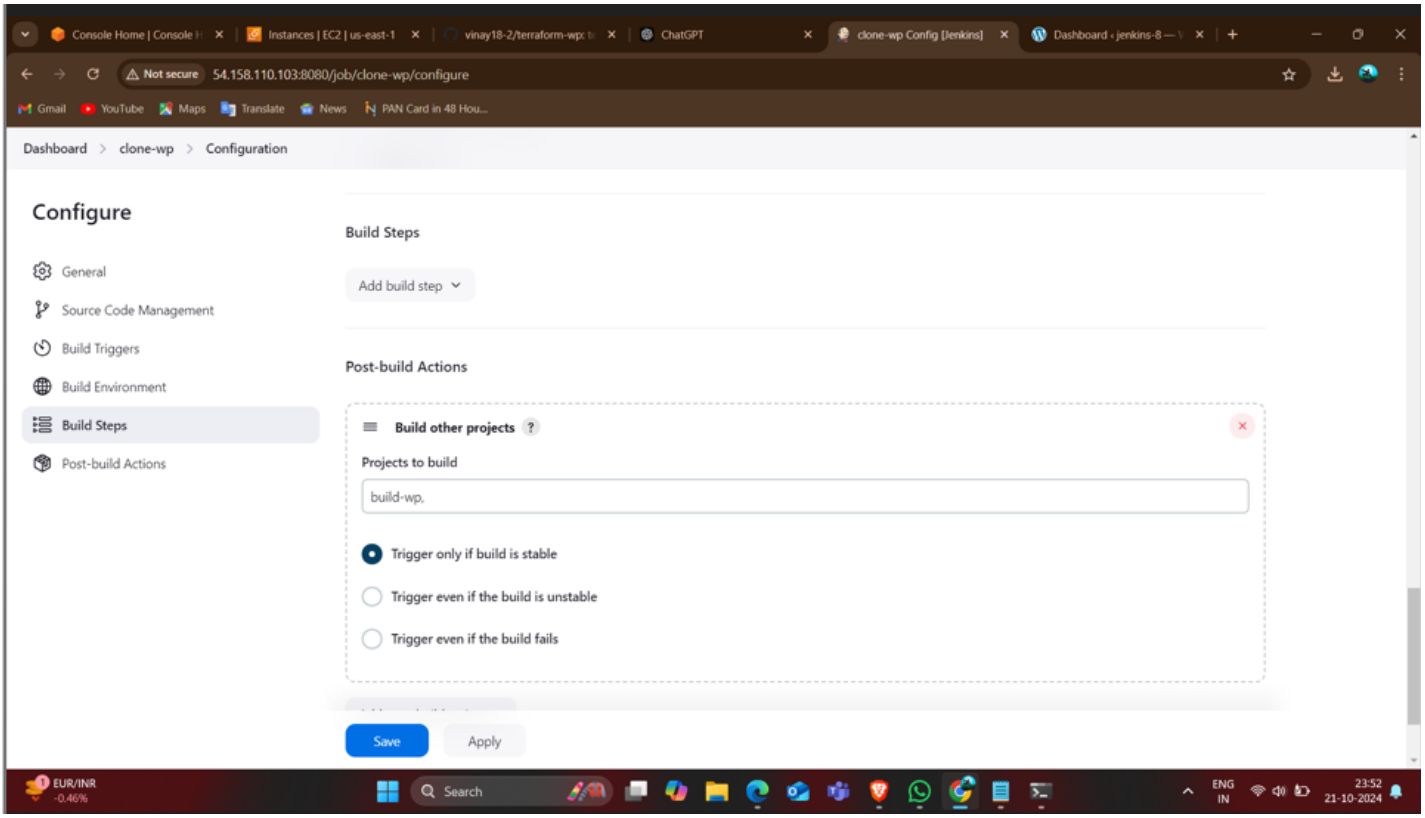
-

- Configure it as shown in below pictures.

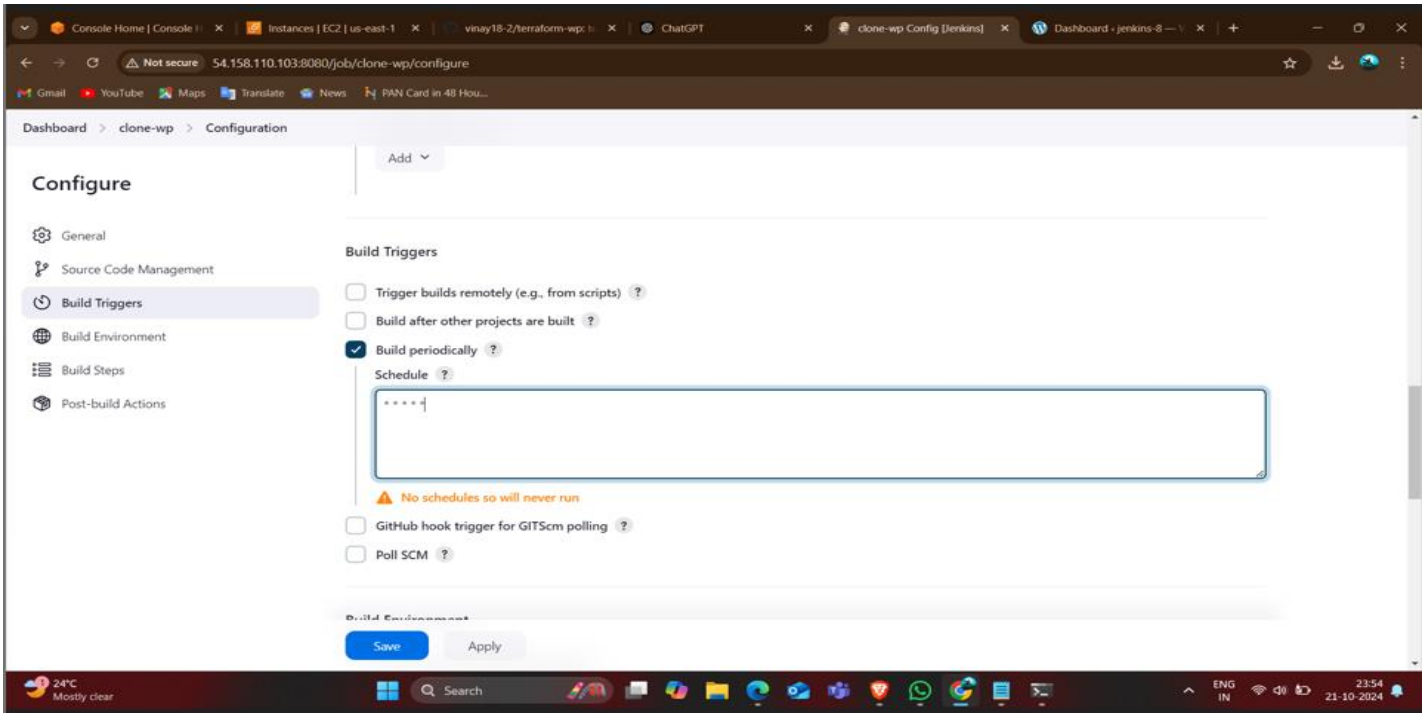




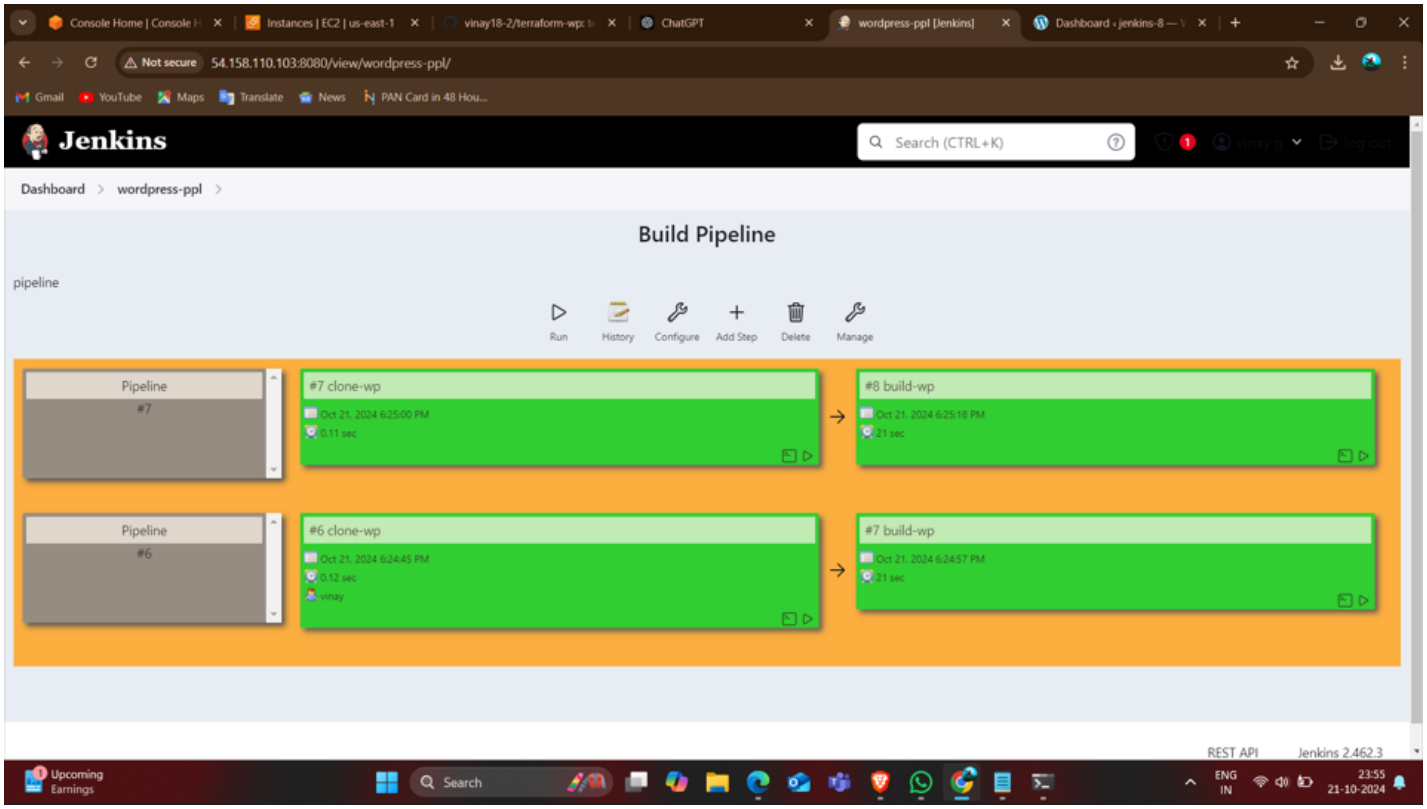
- Configure the clone job by adding post build action as a build job.



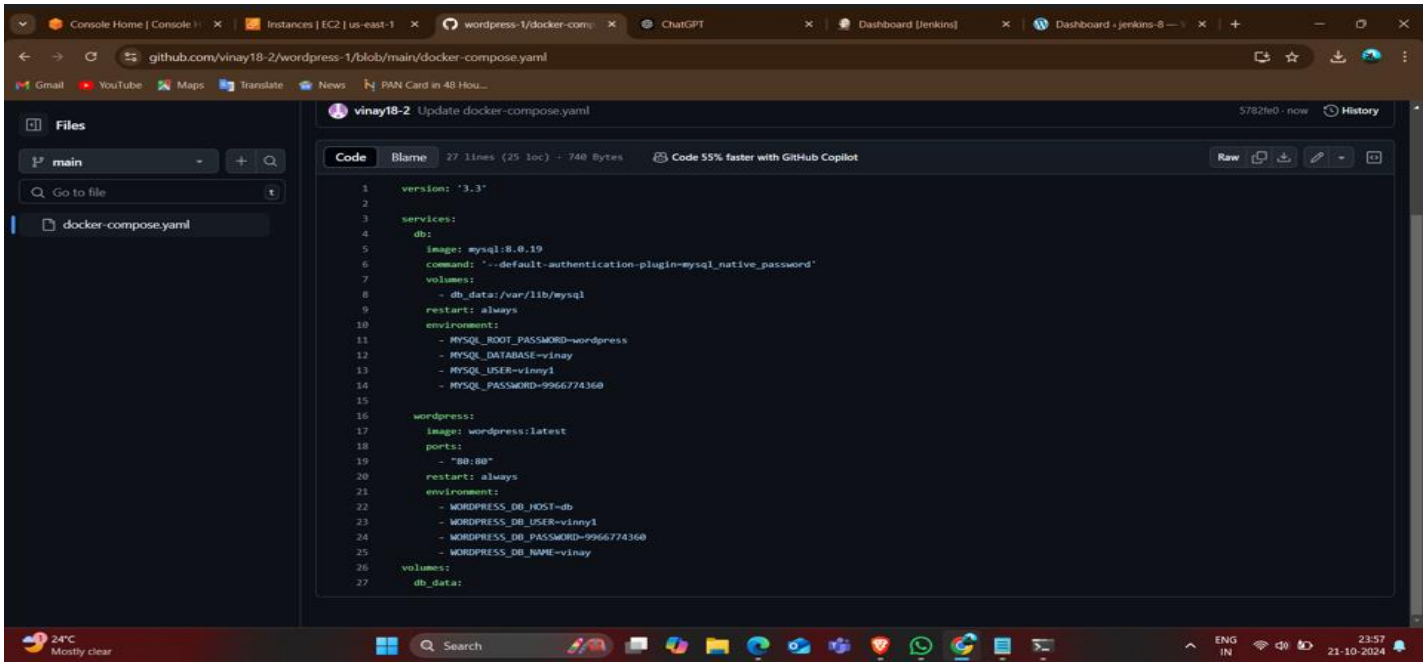
- Provide the schedule for build periodically.



-
  - Output for build periodically.

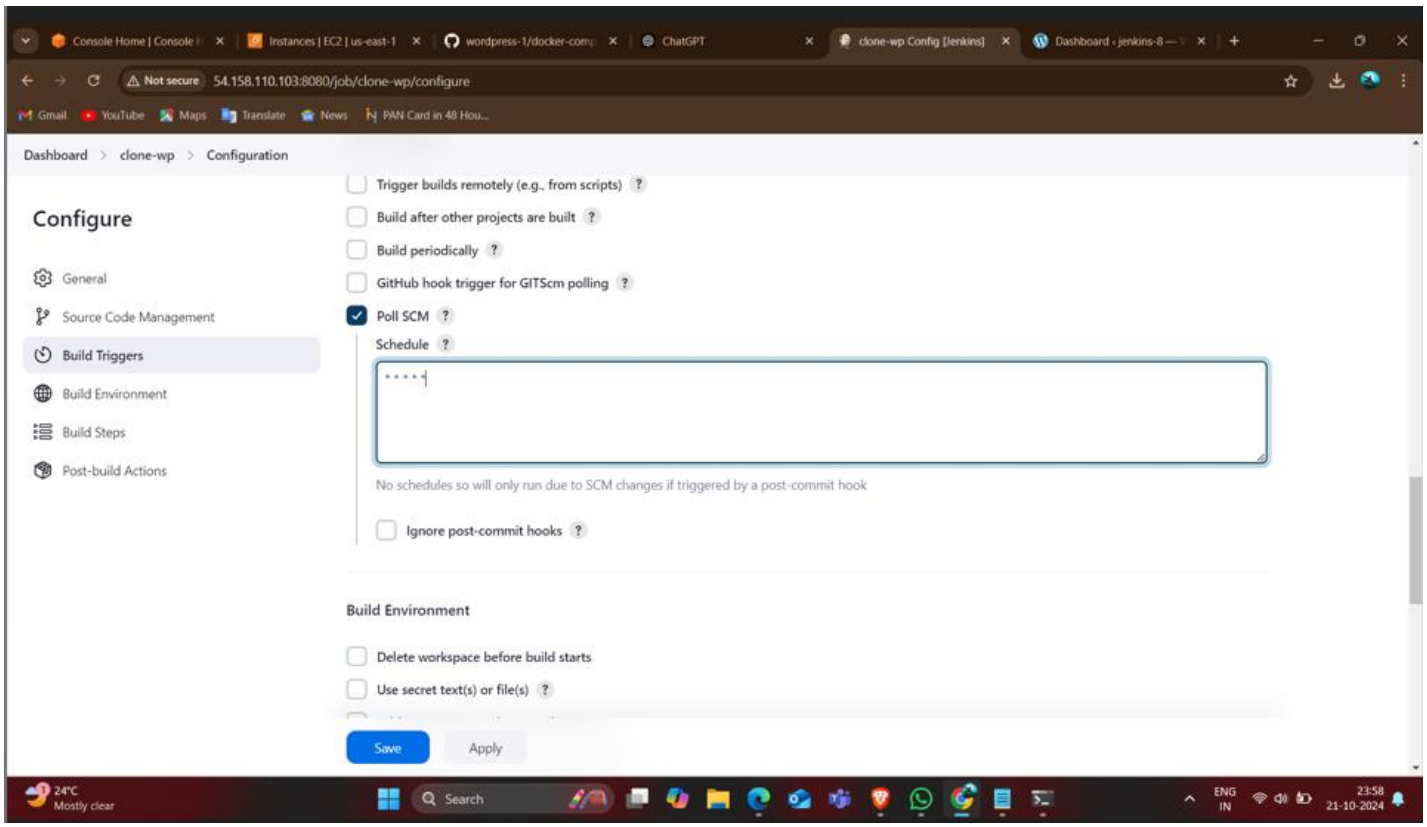


  - After that change the script or yaml code and buil woth the poll scm and observe.

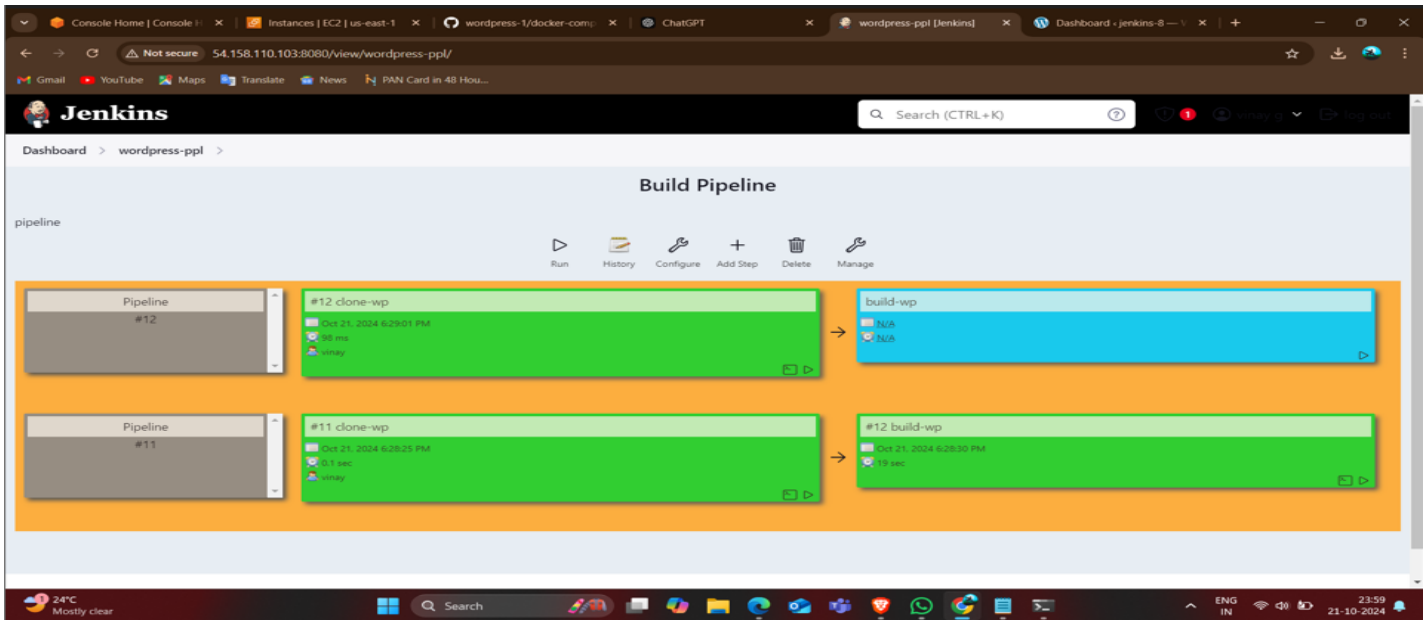


-

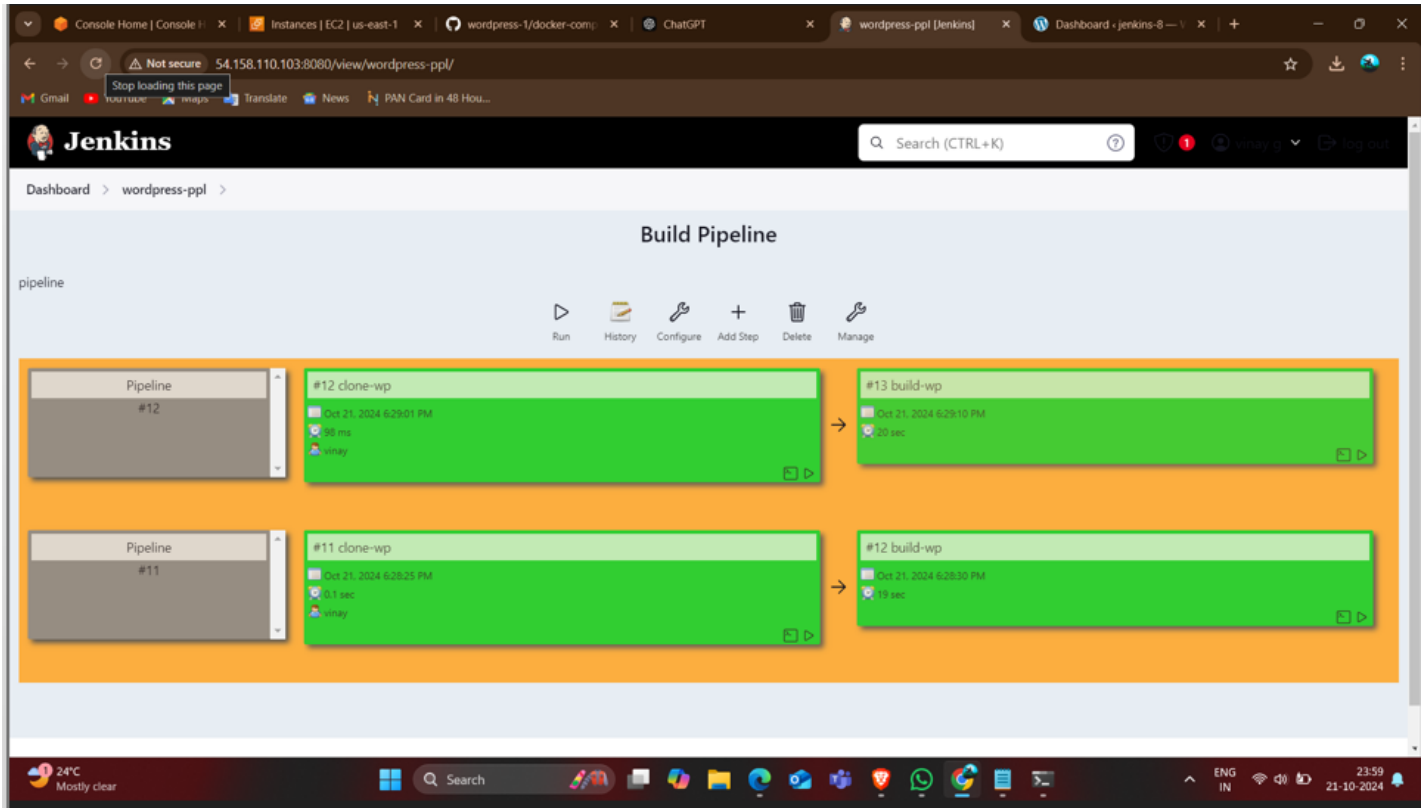
  - Provide the schedule for poll scm and save it and apply it. Than build now.



  - Output.



•

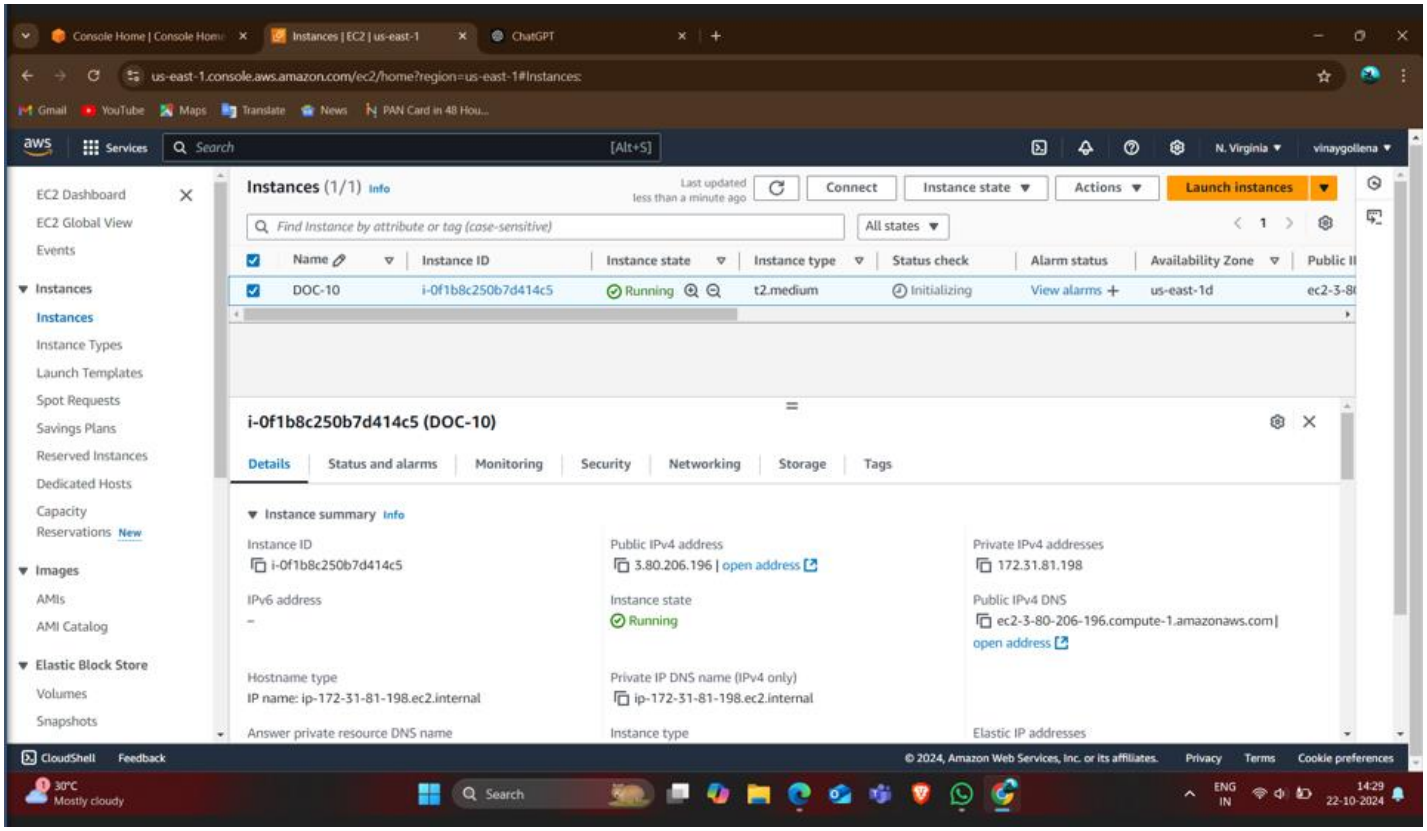

Jenkins
Search (CTRL+K)
Dashboard
wordpress-ppl
Build Pipeline
pipeline
Run
History
Configure
Add Step
Delete
Manage
Pipeline
#12
#12 clone-wp
#13 build-wp
Pipeline
#11
#11 clone-wp
#12 build-wp

•

Deploy WordPress web application by using k8's (Declarative manifest method) with the help of docker hub images

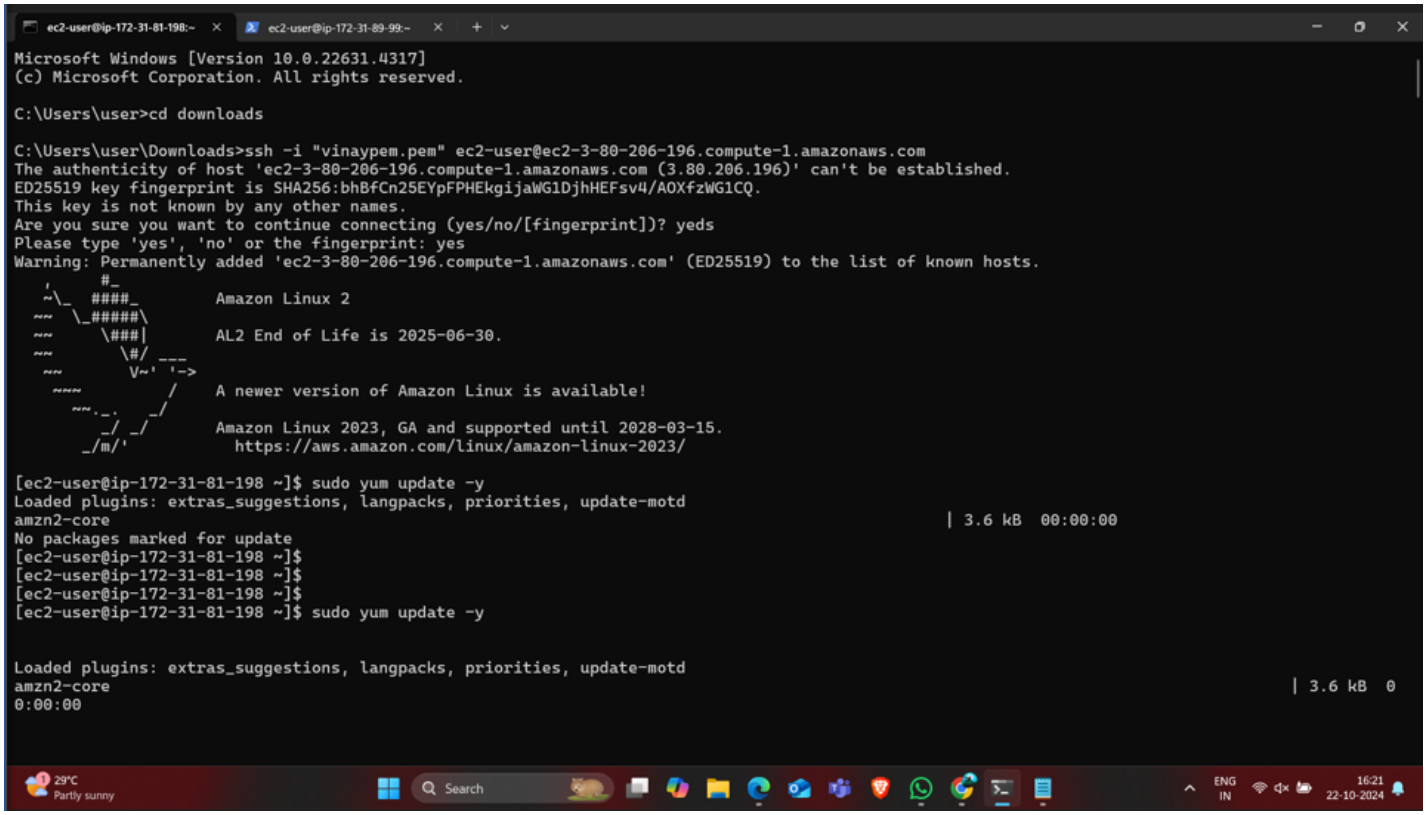
- Launch the 2 EC2 instances.
- One for docker and another for kuburnetes.

- Create the instance for Kuburnetes.



- Connect it to the terminal.
- Update it.

•



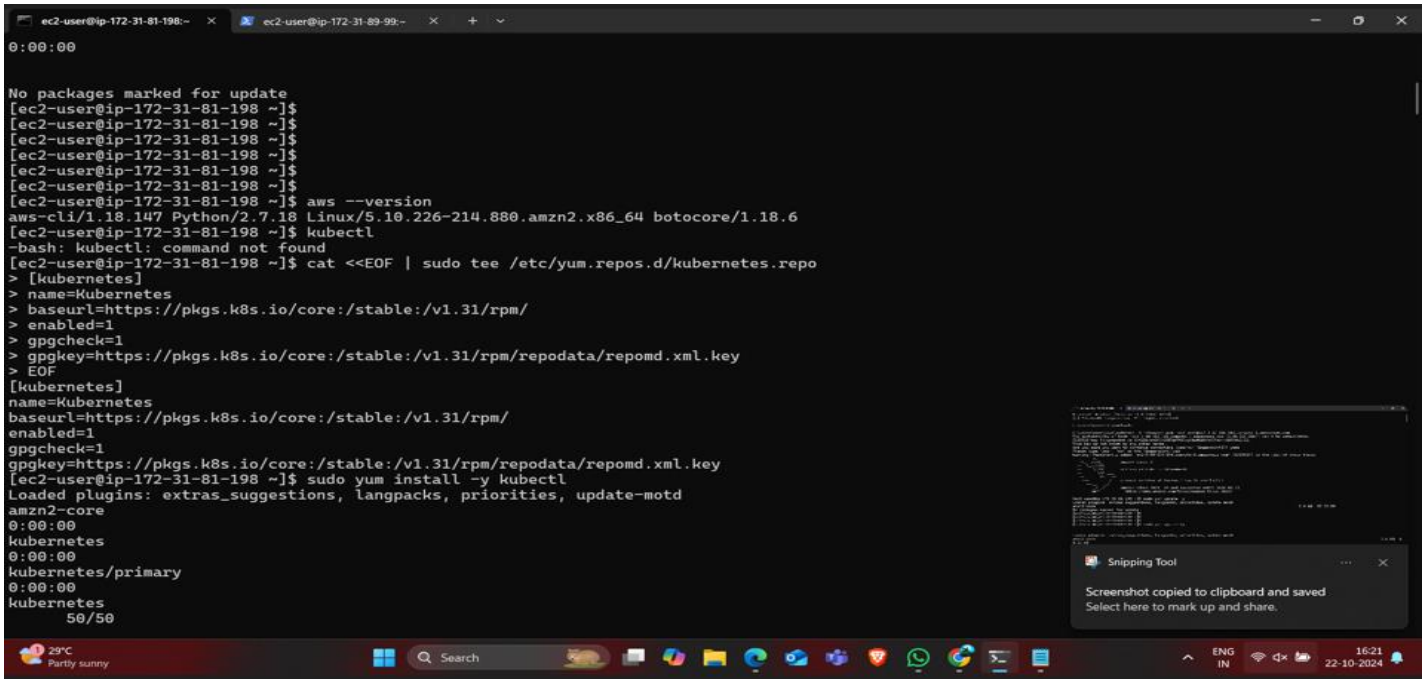
- Check the version of aws.
- Install the kubernetes from the google.
- Install kubectl by using command “sudo yum install kubectl –y”
- Install kops

curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d "" -f 4)/kops-linux-amd64

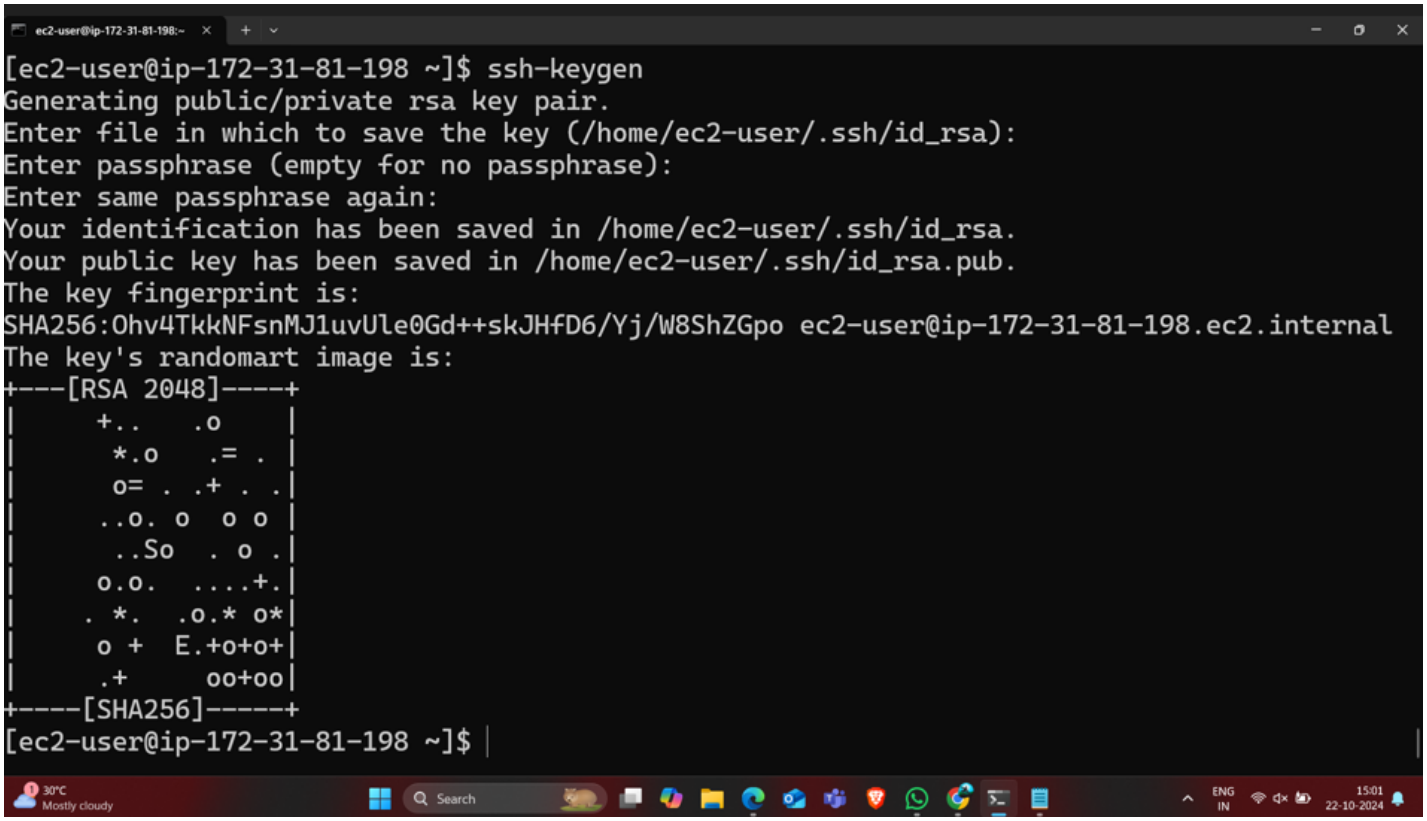
chmod +x kops

sudo mv kops /usr/local/bin/kops

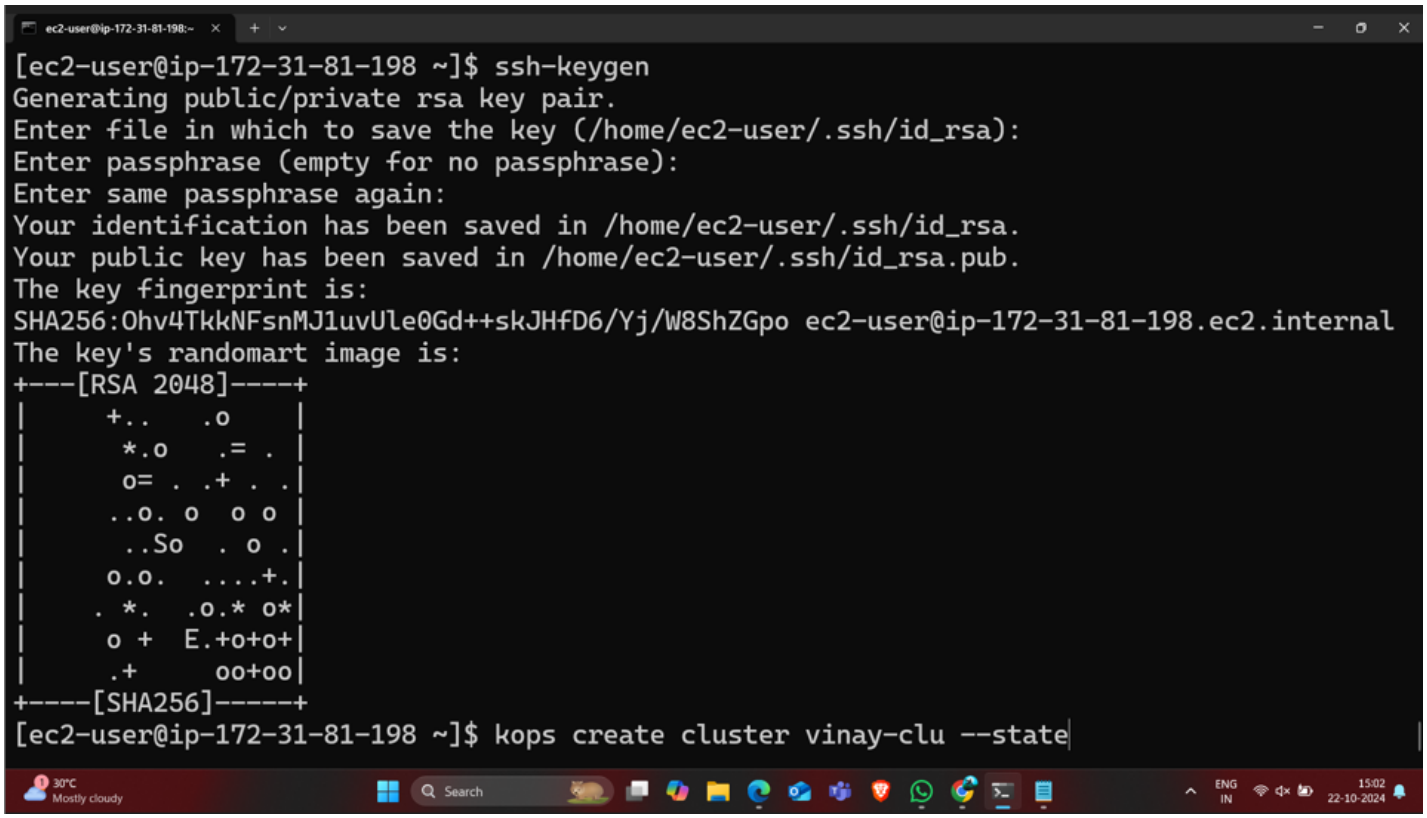
•



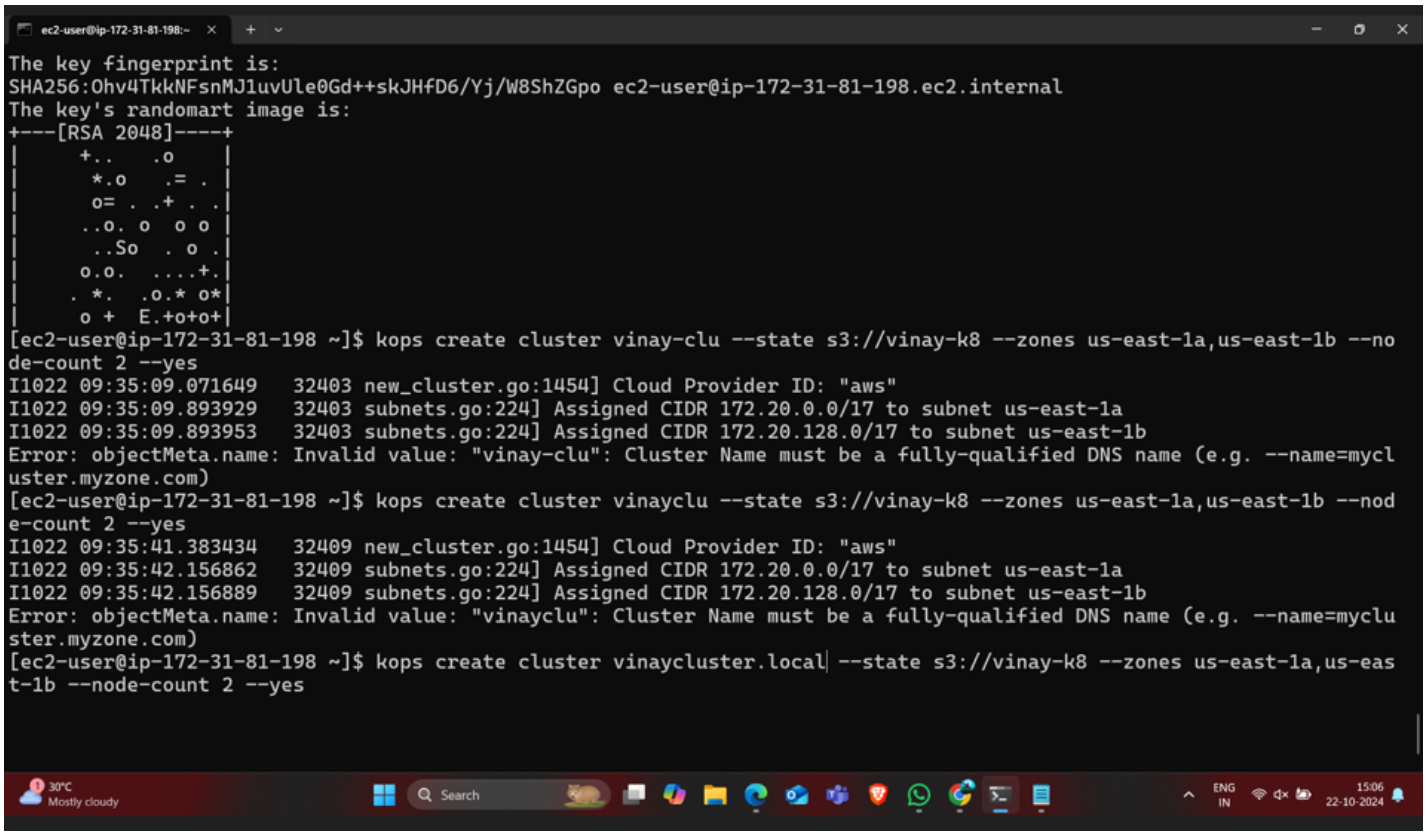
- Configure the AWS and provide the access key and secret key
- Create the s3 bucket by using “aws s3 mb s3://bucket name(vinay-k8)”
- Export the bucket to kops and generate ssh key.



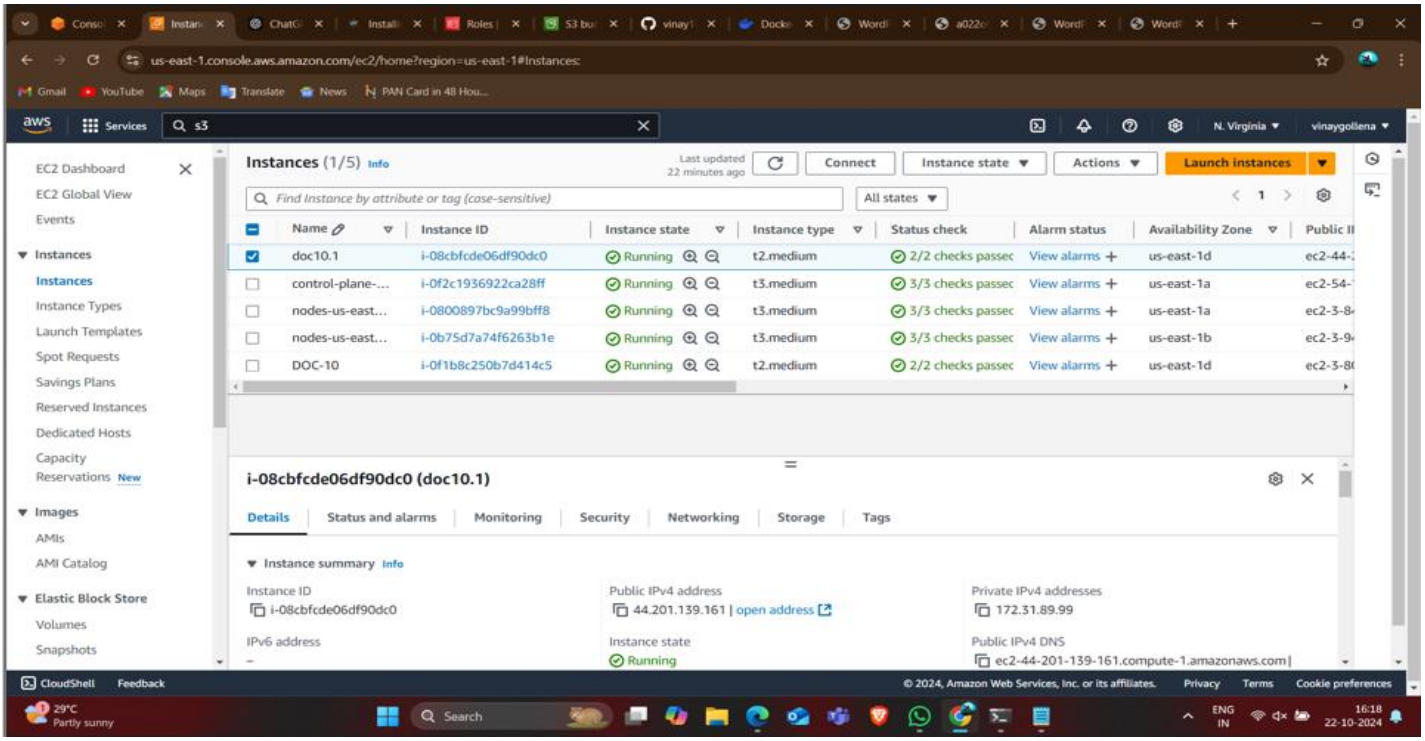
•



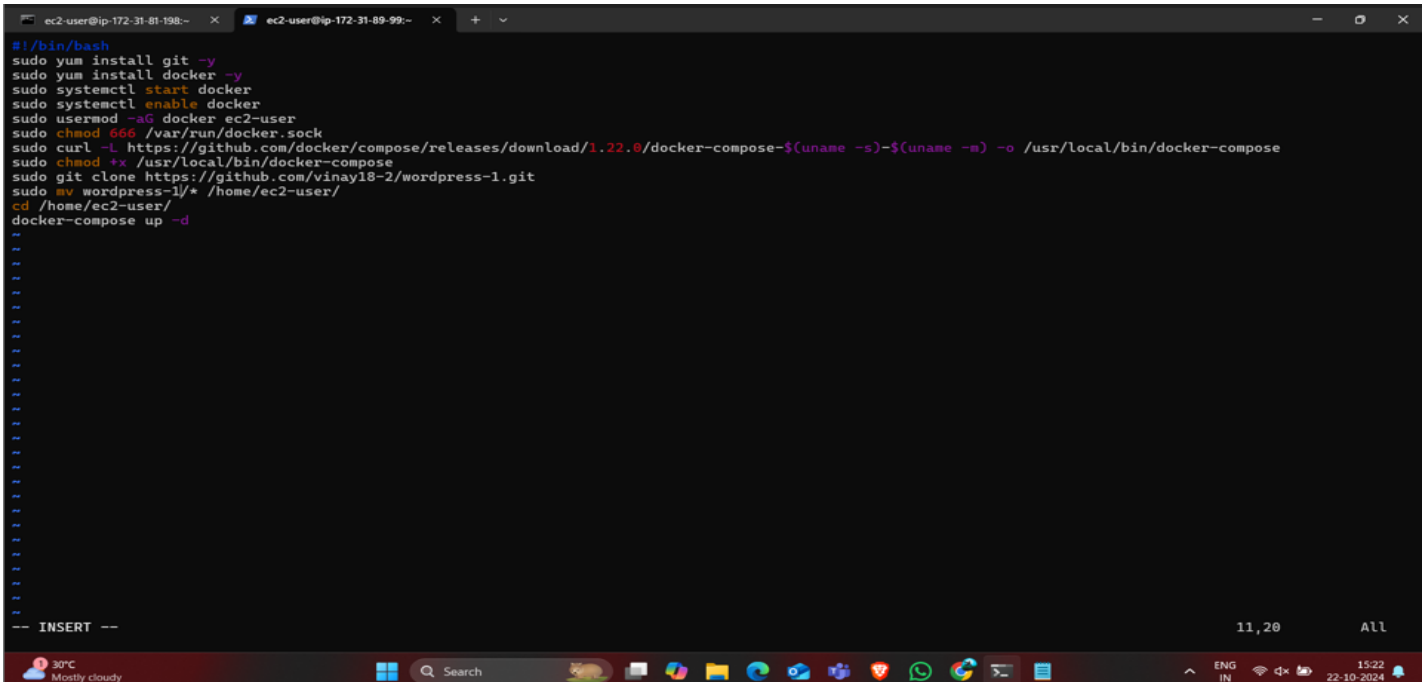
- Create the cluster.



- Creating the 2nd instance for docker.

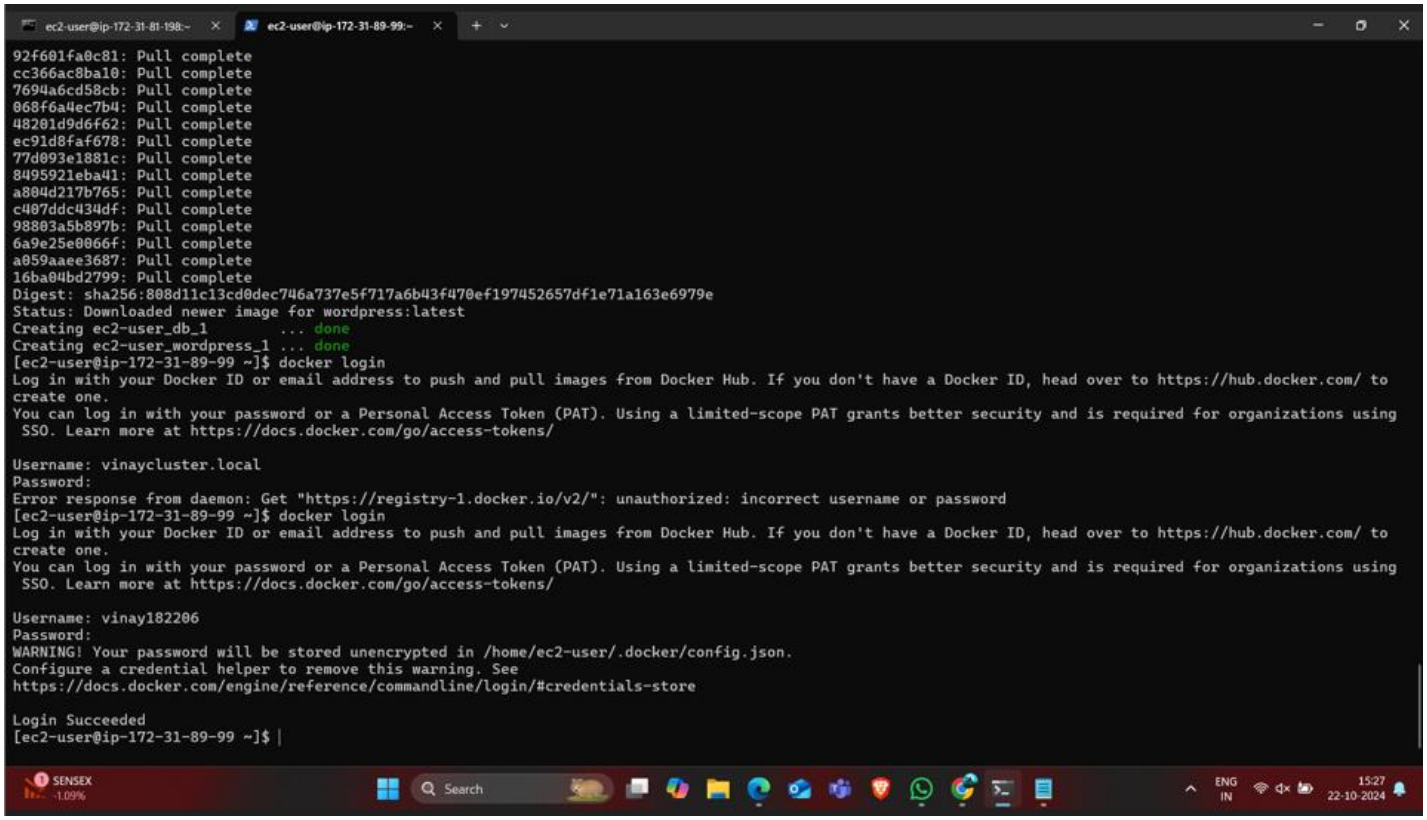


- Connect to the terminal by using the ssh command,and update it.



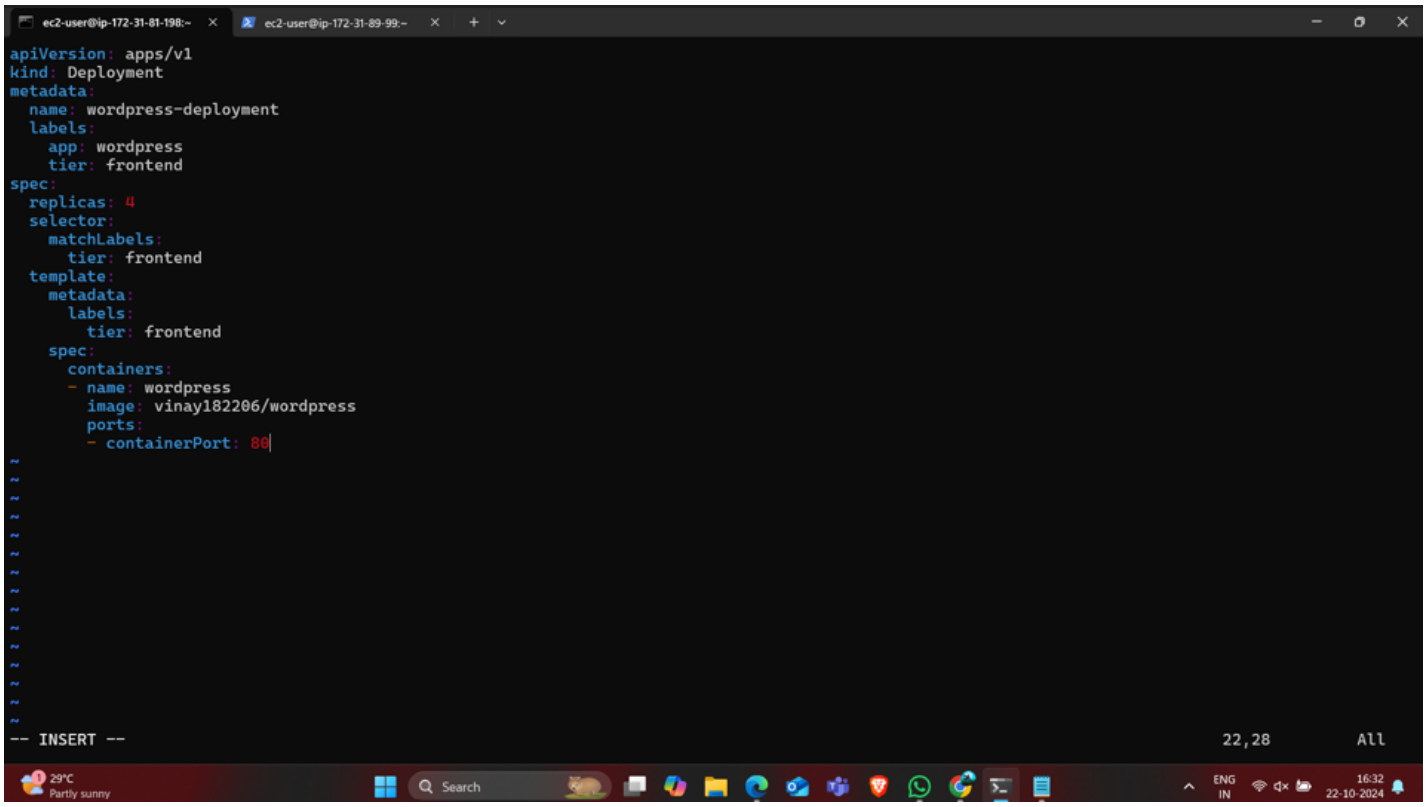
•

- Run the bash script by using “./script_name.sh”
- Create an image and tag
  “docker build -t vinay182206/wordpress” or” docker tag wordpress:latest vinay182206/wordpress”
  docker login
  docker images
  docker push vinay182206/wordpress
- Run the command “docker login”
- Provide the username and password.

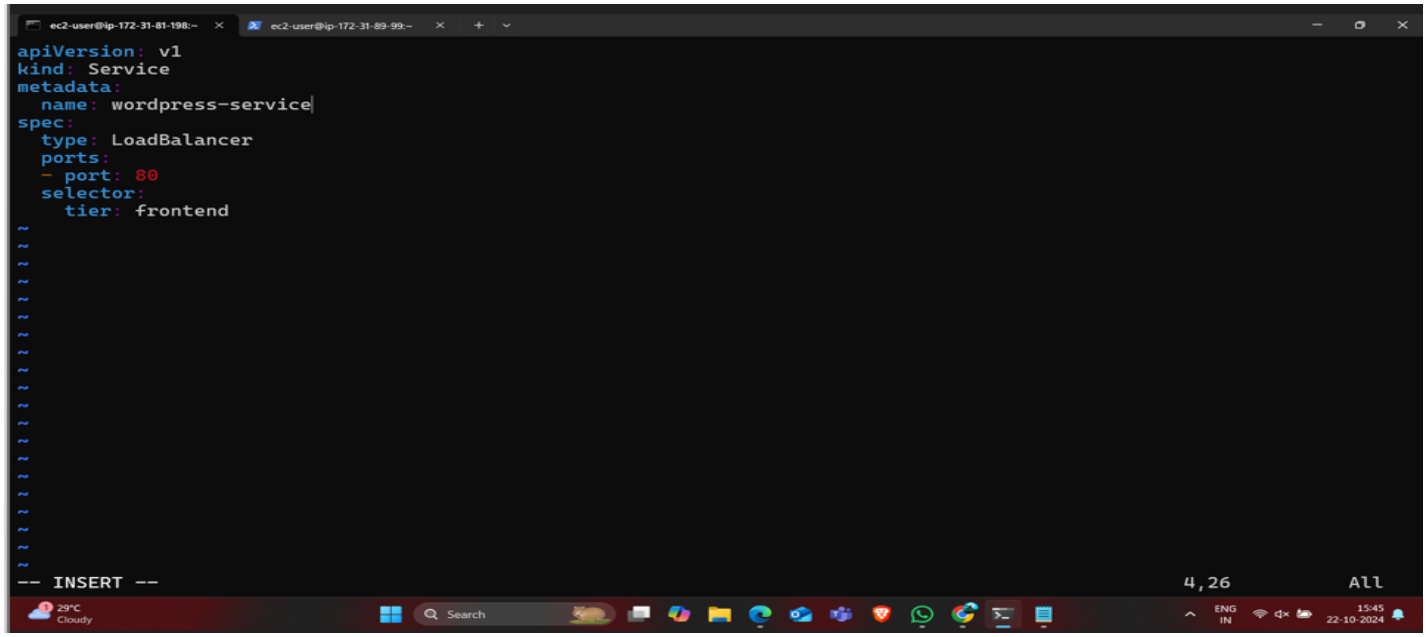


- Write the deployment file and service file in yaml script in the terminal where the kubernetes installed.
- In that provide the image name and username correctly.
- And save it than apply the commands
  kubectl apply deployment.yml
  Kubectl apply service.yml

•

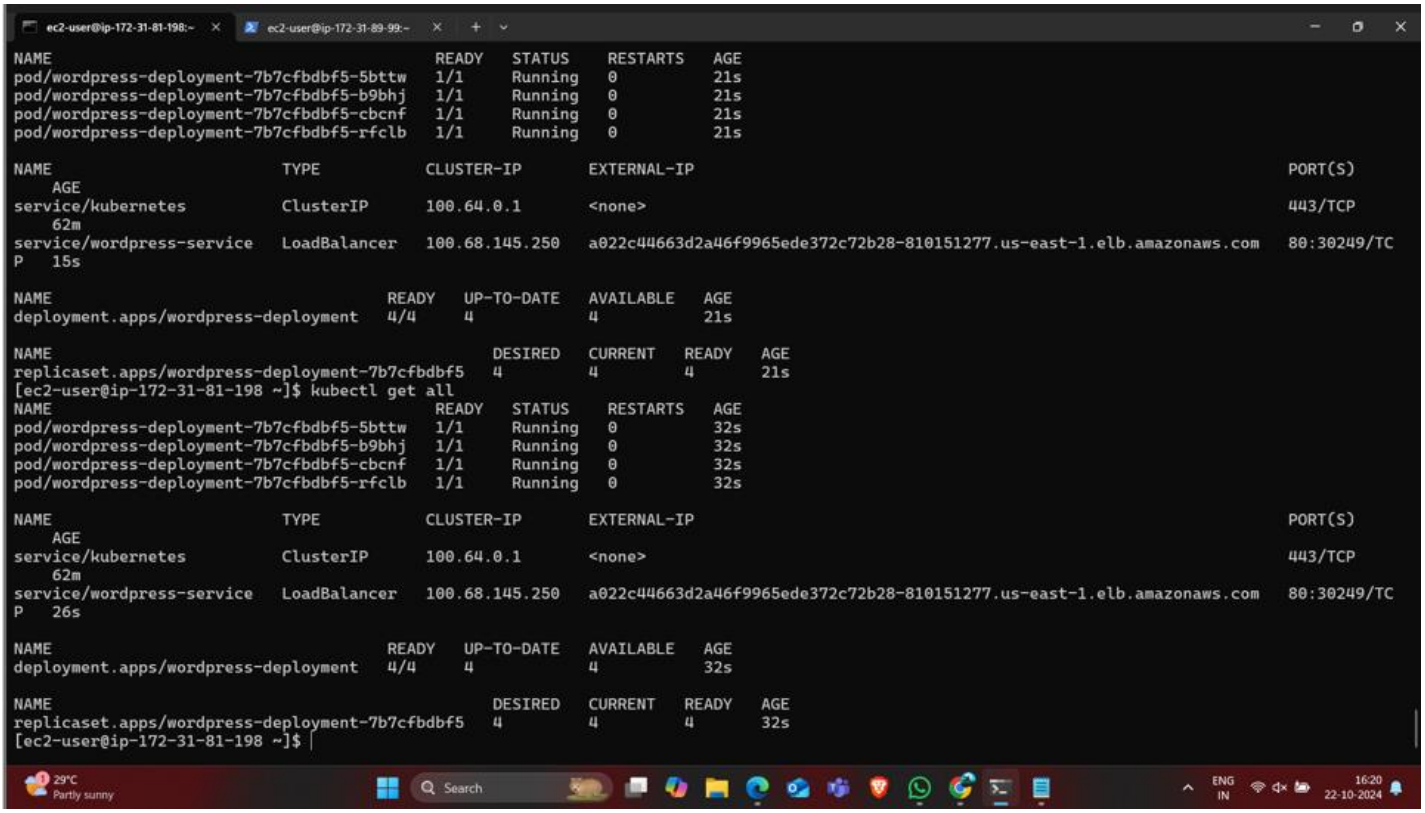


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
  labels:
    app: wordpress
    tier: frontend
spec:
  replicas: 4
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
      - name: wordpress
        image: vinay182206/wordpress
        ports:
        - containerPort: 80
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    22,28         All
```

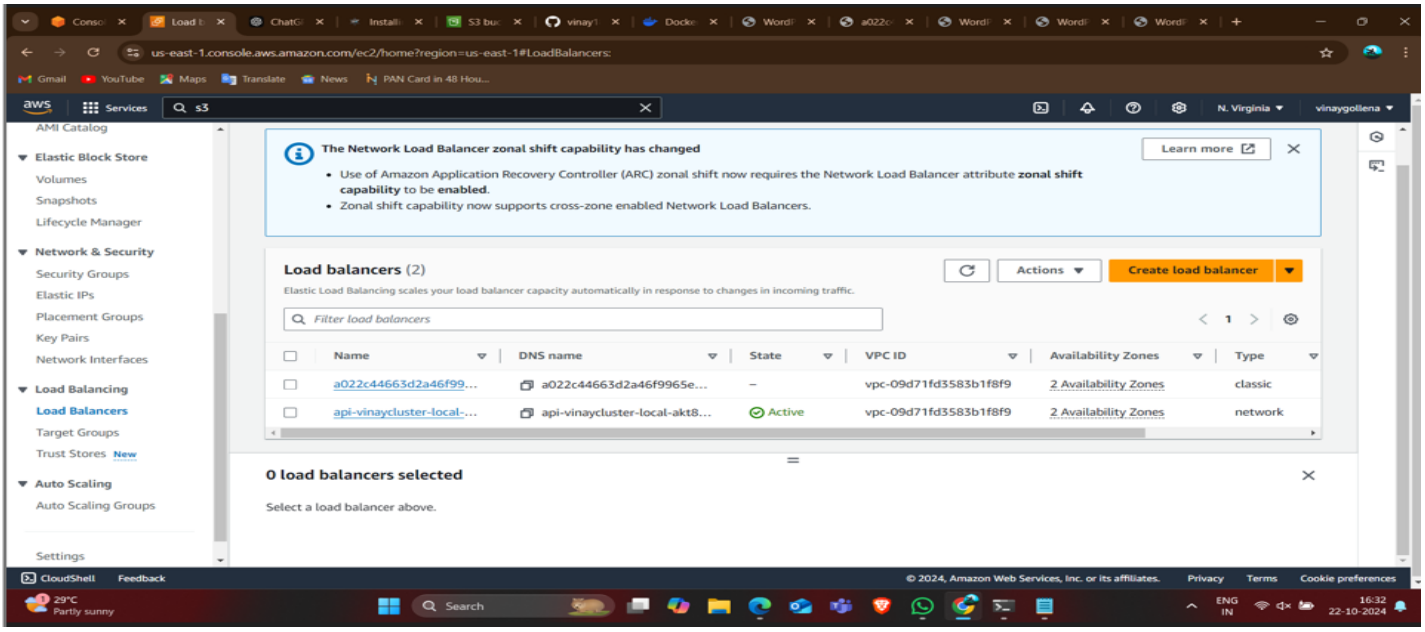


```
apiVersion: v1
kind: Service
metadata:
  name: wordpress-service
spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    tier: frontend
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                                    4,26          All
```

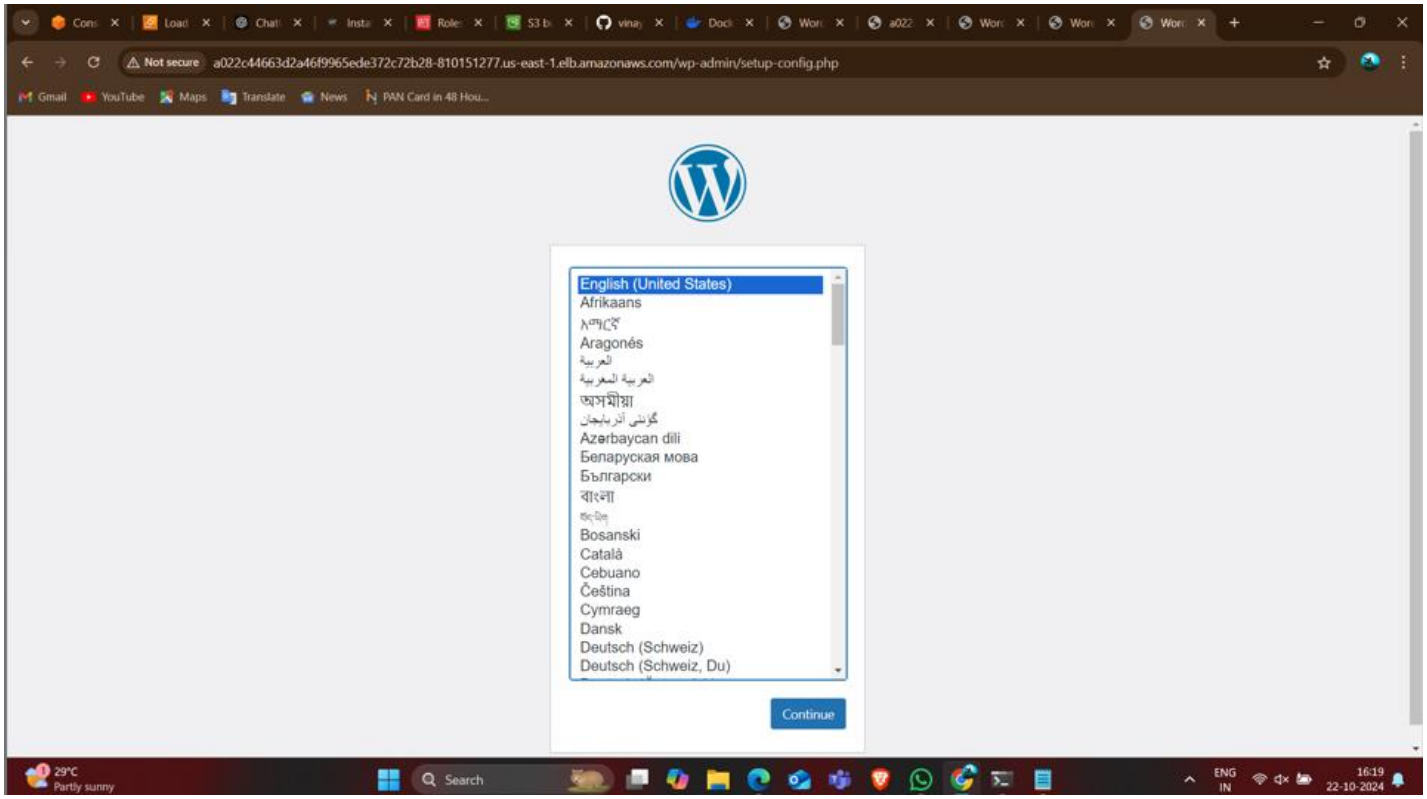
- After applying the kubectl aommands we will get this type and here we can get the end point address of the loadbalancer.
- Copy it and browse in the google we will get the output.



- Or else we can go to aws concole and copythe loadbalancer end point url and browse it.



-
  - Output.


Not secure
a022c44663d2a46f9965ede372c72b28-810151277.us-east-1.elb.amazonaws.com/wp-admin/setup-config.php
English (United States)
Afrikaans
Aragonés
Azərbaycan dili
Беларуская мова
Български
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Schweiz, Du)
Continue
Search