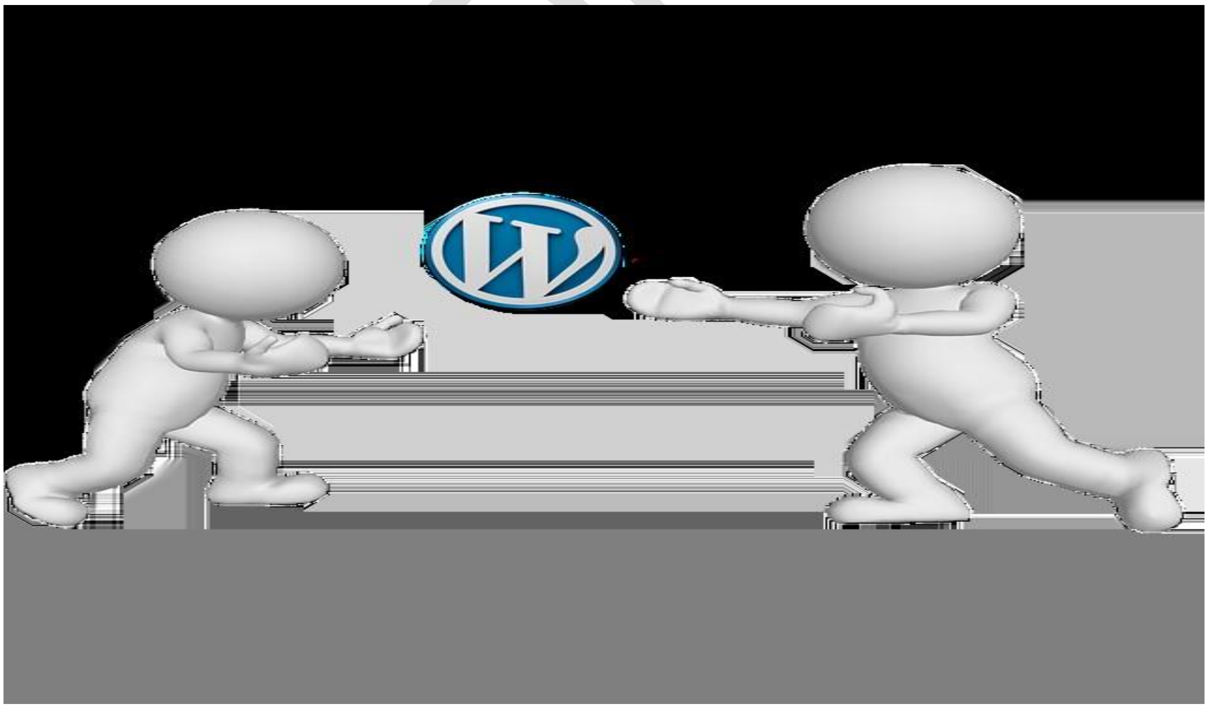
Word Press

WordPress (also known as WP or WordPress.org) is a web content management system. It was originally created a tool to publish blogs but has evolved to support publishing other web content, including more traditional websites, mailing lists and Internet forum, media galleries, membership sites, learning management systems and online stores. Available as free and open-source software, WordPress is among the most popular content management systems – it was used by 43.1% of the top 10 million websites as of December 2023.



WordPress is written in PHP hypertext preprocessor language and paired with a MySQL or MariaDB database.

Features include a plugin architecture and a template system, referred to within WordPress as "Themes".

WordPress has to be installed on a web server, either as part of an Internet hosting service or on a computer running the WordPress software package.

WordPress was released on May 27, 2003, by its founders, American developer Matt Mullenwegand English developer Mike Little. WordPress Foundation owns WordPress, WordPress projects, and other related trademarks.

What is Wordpress used for?

WordPress is a content management system (CMS) that allows you to host and build websites. WordPress contains plugin architecture and a template system, so you can customize any website to fit your business, blog, portfolio, or online store.

Features:

- Flexibility.
- User-friendliness.
- Media management.
- Quick installation and upgrade.
- WordPress language.
- User management.
- Simplicity of operations.

- Easy theme system.

How many types of wordpress are there?

- There are two types of WordPress websites:

1. There is WordPress.com, which is a web hosting company.

2. There is WordPress.org, also known as self-hosted WordPress.

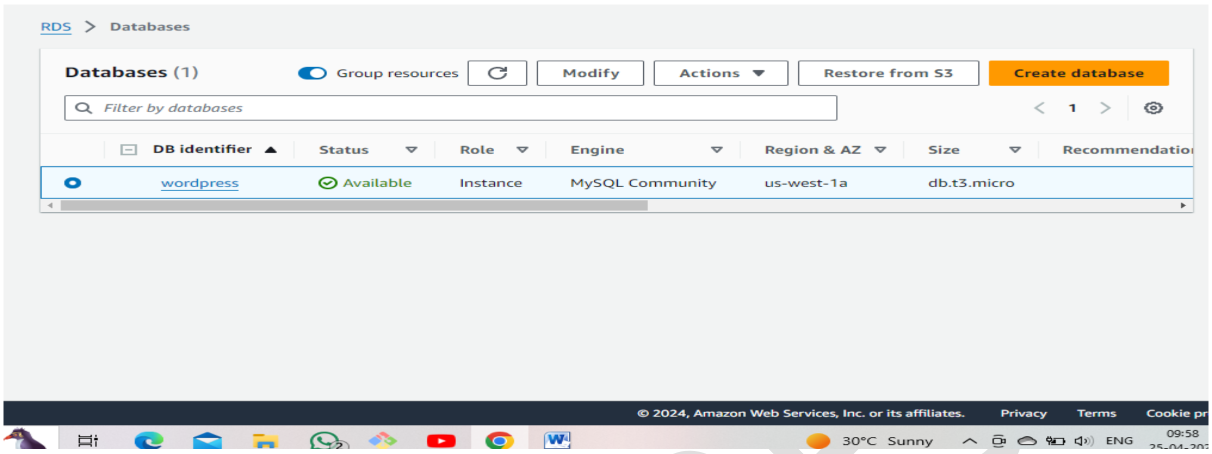
What is the most popular website on Wordpress?

- Blog or Personal Website
- Business Website
- Ecommerce Website / Online Store
- Membership Website
- Online Courses Website
- Online Marketplace Website
- Pay-Per-View Website
- Podcast Website

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

- First go to aws account and login with credentials. After go to RDS service and open that service.
- Now create a mysql database by using RDS service for that go into the RDS services and click on create database.
- Now select the database creation method I selected here standard create method.
- Now select the database engine with version but I select the mysql database engine.
- Now select the template as free tier and by selecting this free tier.
- Now give the some name to your database and give username and passwords as credentials for your database access.
- Now select the storage type as General purpose SSD (gp2) and enter the storage value as (1000GB) minimum (20GB).
- Now select created VPC or select default VPC and it automatically select the database subnet group.
- Now give the name of the database which you give at the stage of DB instance identifier enter the same name here.

- **Now click on create database button and it will create the mysql database.**



RDS > Databases

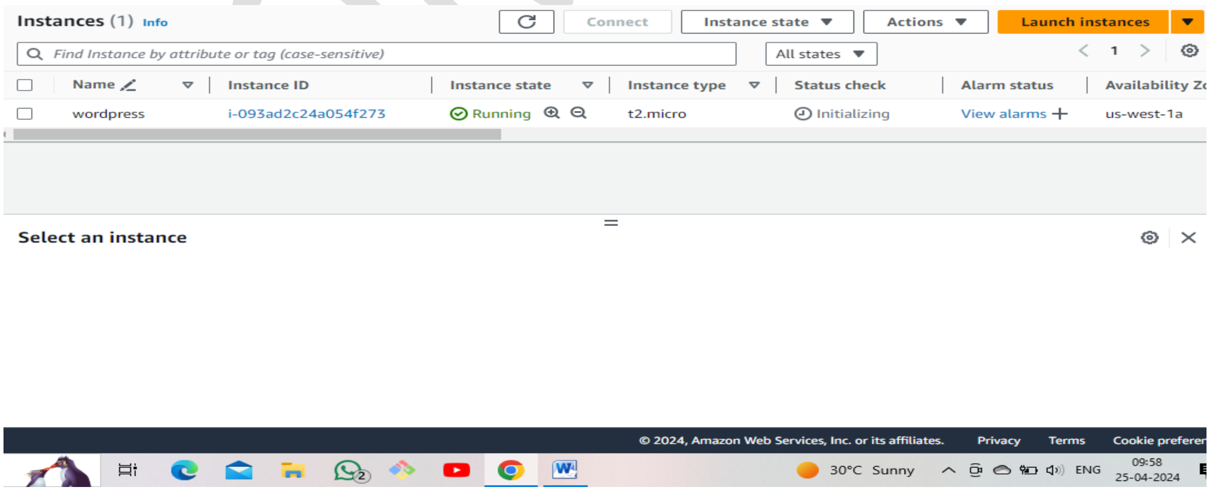
Databases (1) Group resources Modify Actions Restore from S3 Create database

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations
wordpress	Available	Instance	MySQL Community	us-west-1a	db.t3.micro	

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

- **Now create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and giving security group with SSH (22) and HTTP (80).**



Instances (1) Info Connect Instance state Actions Launch instances

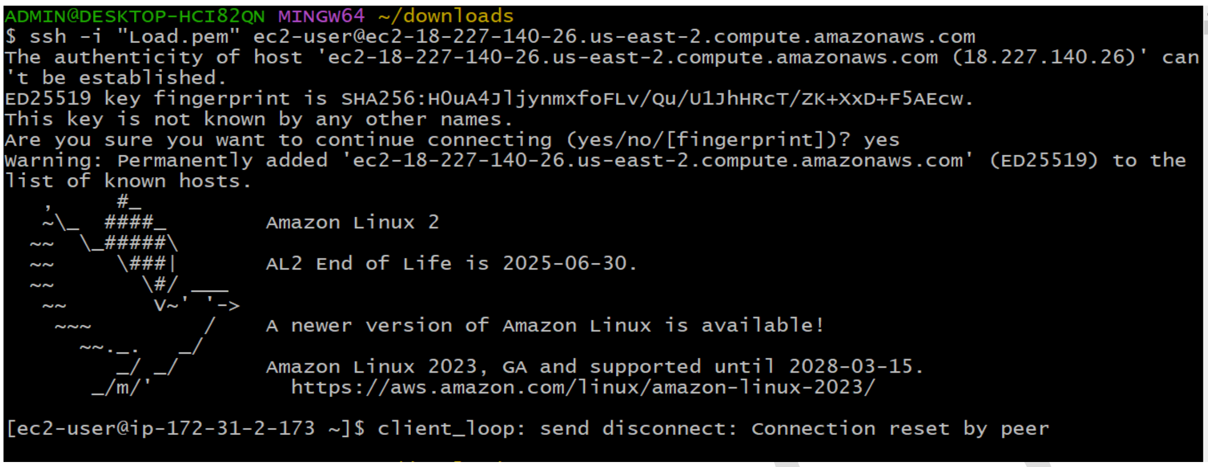
Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
wordpress	i-093ad2c24a054f273	Running	t2.micro	Initializing	View alarms +	us-west-1a

Select an instance

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

➢ Now connect the virtual server through the GitBash.

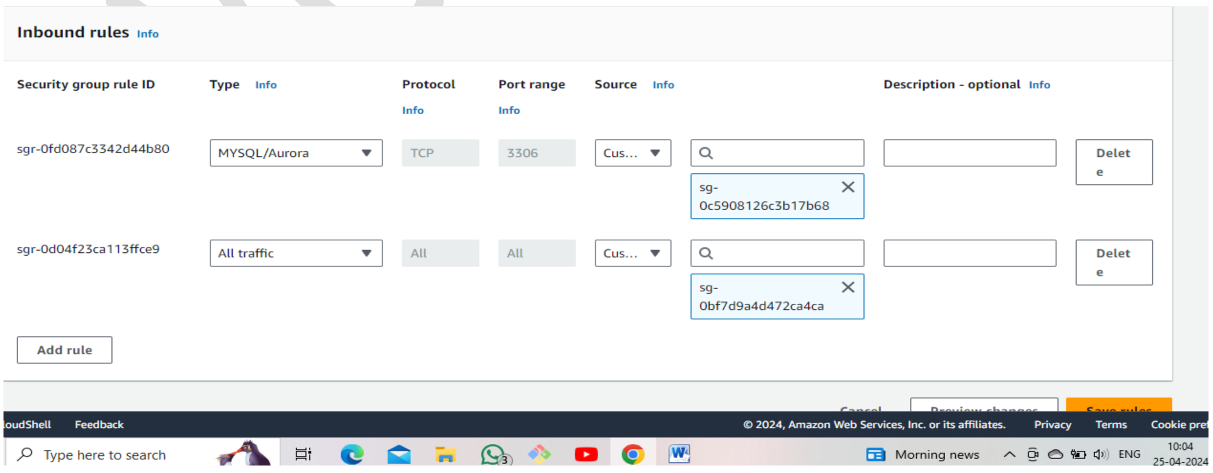


```
ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "Load.pem" ec2-user@ec2-18-227-140-26.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-227-140-26.us-east-2.compute.amazonaws.com (18.227.140.26)' can
't be established.
ED25519 key fingerprint is SHA256:HOuA4JljynmxfoFLv/Qu/U1JhHRcT/ZK+XxD+F5AEcw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-227-140-26.us-east-2.compute.amazonaws.com' (ED25519) to the
list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-2-173 ~]$ client_loop: send disconnect: Connection reset by peer
```

➢ Now update the linux version by using command as
< sudo yum –y update >

➢ Now go inside the created database and go to the security under this option there is a security group id click on that.

➢ Now go to inbound rules and click on the edit inbound rules and select the EC2 instance security group id and click save rules.



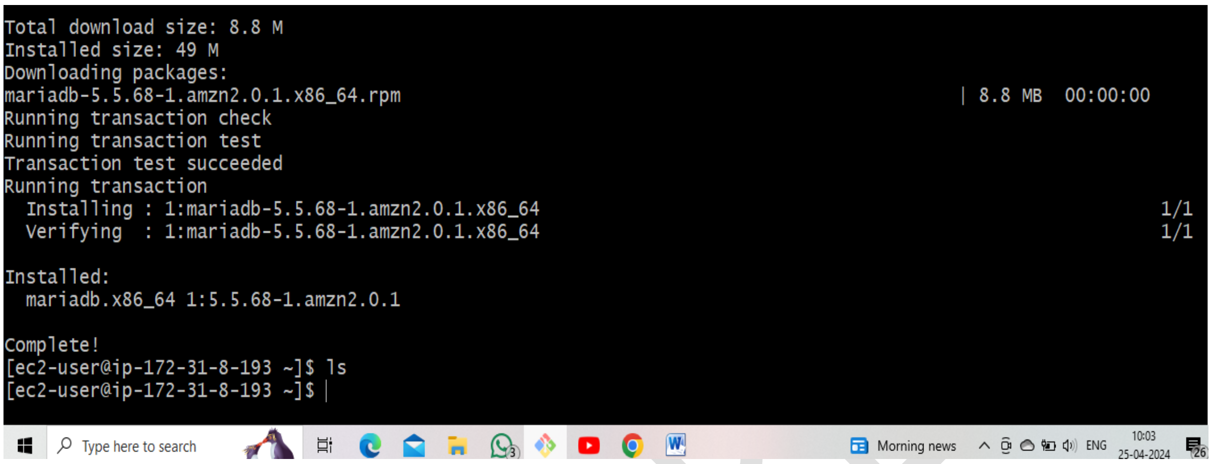
Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0fd087c3342d44b80	MYSQL/Aurora	TCP	3306	Cus...	sg-0c5908126c3b17b68		Delete
sgr-0d04f23ca113ffce9	All traffic	All	All	Cus...	sg-0bf7d9a4d472ca4ca		Delete

Add rule

➢ **Now access the mysql database by using the command as**

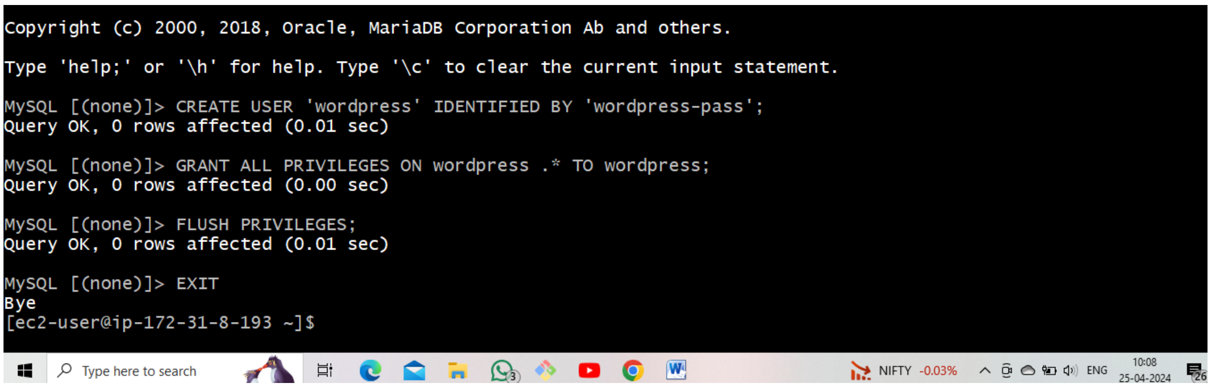
< sudo mysql –h (database endpoint address) –u (database user) –p >



```
Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm                                             | 8.8 MB   00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64                                                   1/1
  Verifying  : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64                                                   1/1

Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-8-193 ~]$ ls
[ec2-user@ip-172-31-8-193 ~]$ |
```

➢ **Now create a database user for wordpress application and give it permissions to access the "wordpress" database.**

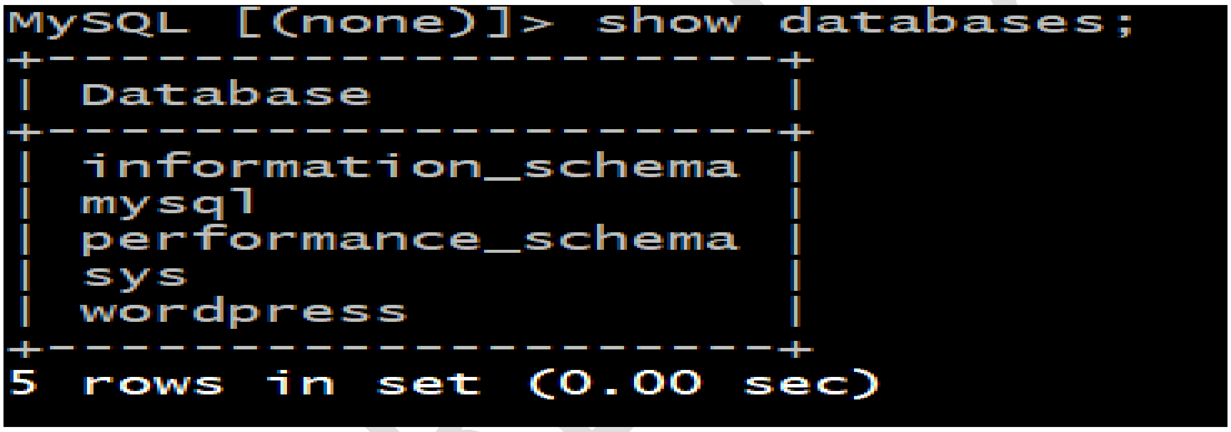
- ✓ **CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';**
- ✓ **GRANT ALL PRIVILEGES ON wordpress .* TO wordpress.**
- ✓ **FLUSH PRIVILEGES**
- ✓ **EXIT**



```
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> GRANT ALL PRIVILEGES ON wordpress .* TO wordpress;
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> EXIT
Bye
[ec2-user@ip-172-31-8-193 ~]$
```

- Show the databases using the command as

  < show databases; >



```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
+--------------------+
5 rows in set (0.00 sec)
```

- **Now install httpd server using this command as**

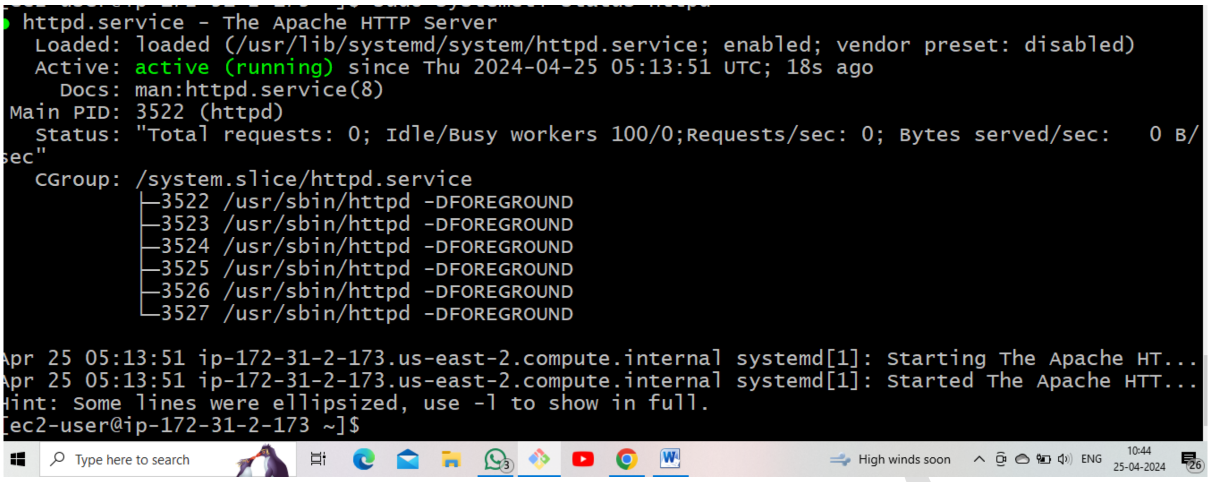
  < sudo yum –y install httpd >

- **Now start,enable,status the HTTPD service by giving the commands as**

  < sudo systemctl start httpd >

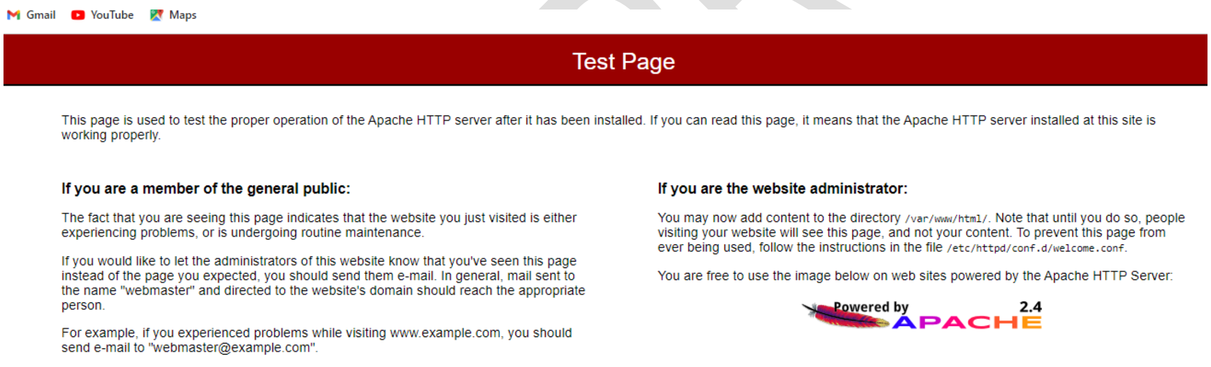
  < sudo systemctl enable httpd >

  < sudo systemctl status httpd >



```
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-04-25 05:13:51 UTC; 18s ago
     Docs: man:httpd.service(8)
 Main PID: 3522 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/
sec"
   CGroup: /system.slice/httpd.service
           ├─3522 /usr/sbin/httpd -DFOREGROUND
           ├─3523 /usr/sbin/httpd -DFOREGROUND
           ├─3524 /usr/sbin/httpd -DFOREGROUND
           ├─3525 /usr/sbin/httpd -DFOREGROUND
           ├─3526 /usr/sbin/httpd -DFOREGROUND
           └─3527 /usr/sbin/httpd -DFOREGROUND

Apr 25 05:13:51 ip-172-31-2-173.us-east-2.compute.internal systemd[1]: Starting The Apache HT...
Apr 25 05:13:51 ip-172-31-2-173.us-east-2.compute.internal systemd[1]: Started The Apache HTT...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-2-173 ~]$
```

- ➢ **Now go to EC2 instance and copy public ip and paste it on Google browse it and check the official page of HTTPD is displays or not.**



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

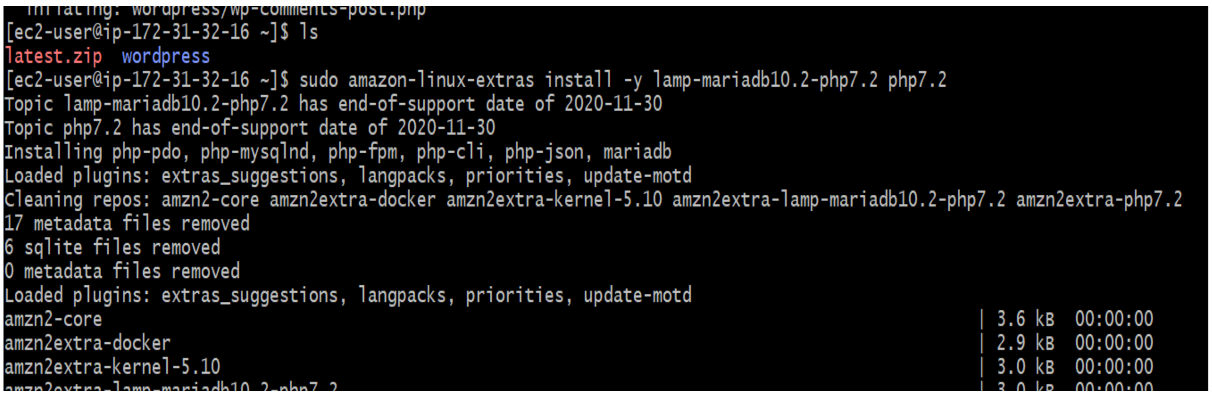
You are free to use the image below on web sites powered by the Apache HTTP Server:

- ➢ **Now go to browser and search as download Wordpress.**

  < sudo wget (copy link address) >

- ➢ **It gives the zip file to unzip that file by using a command as**

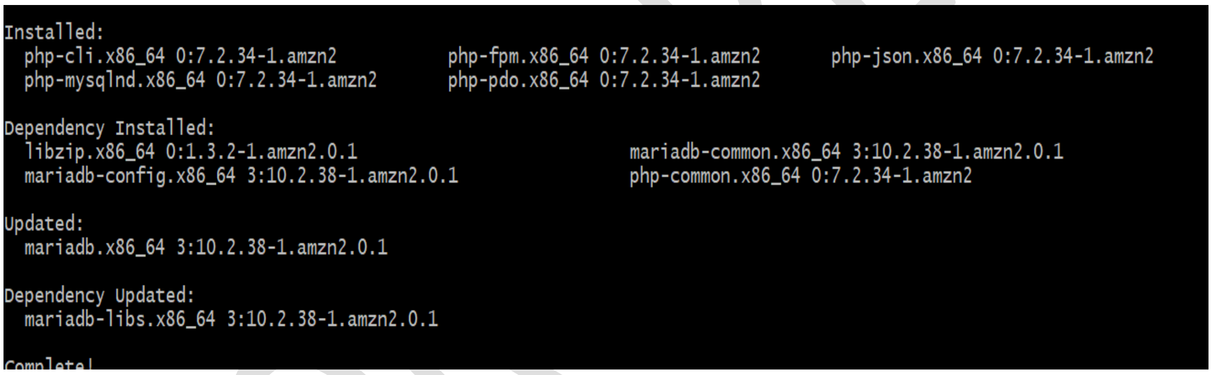
  < unzip (zip file) >



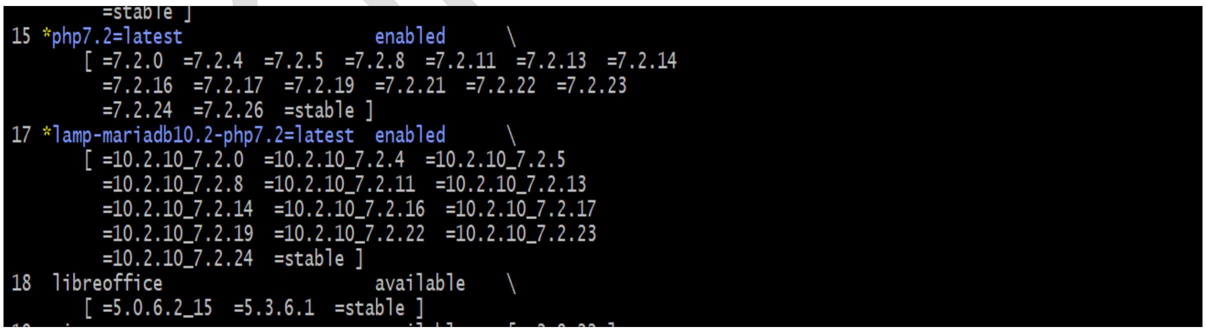
```
  inflating: wordpress/wp-comments-post.php
[ec2-user@ip-172-31-32-16 ~]$ ls
latest.zip  wordpress
[ec2-user@ip-172-31-32-16 ~]$ sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
Topic lamp-mariadb10.2-php7.2 has end-of-support date of 2020-11-30
Topic php7.2 has end-of-support date of 2020-11-30
Installing php-pdo, php-mysqlnd, php-fpm, php-cli, php-json, mariadb
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker amzn2extra-kernel-5.10 amzn2extra-lamp-mariadb10.2-php7.2 amzn2extra-php7.2
17 metadata files removed
6 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                                   | 3.6 kB  00:00:00
amzn2extra-docker                                                            | 2.9 kB  00:00:00
amzn2extra-kernel-5.10                                                       | 3.0 kB  00:00:00
amzn2extra-lamp-mariadb10.2-php7.2                                           | 3.0 kB  00:00:00
```

- **Now download the following command as**

  < sudo amazon-linux-extras install –y lamp-mariadb10.2-php7.2 php7.2 >



```
Installed:
  php-cli.x86_64 0:7.2.34-1.amzn2          php-fpm.x86_64 0:7.2.34-1.amzn2       php-json.x86_64 0:7.2.34-1.amzn2
  php-mysqlnd.x86_64 0:7.2.34-1.amzn2      php-pdo.x86_64 0:7.2.34-1.amzn2

Dependency Installed:
  libzip.x86_64 0:1.3.2-1.amzn2.0.1                          mariadb-common.x86_64 3:10.2.38-1.amzn2.0.1
  mariadb-config.x86_64 3:10.2.38-1.amzn2.0.1                php-common.x86_64 0:7.2.34-1.amzn2

Updated:
  mariadb.x86_64 3:10.2.38-1.amzn2.0.1

Dependency Updated:
  mariadb-libs.x86_64 3:10.2.38-1.amzn2.0.1

Complete!
```

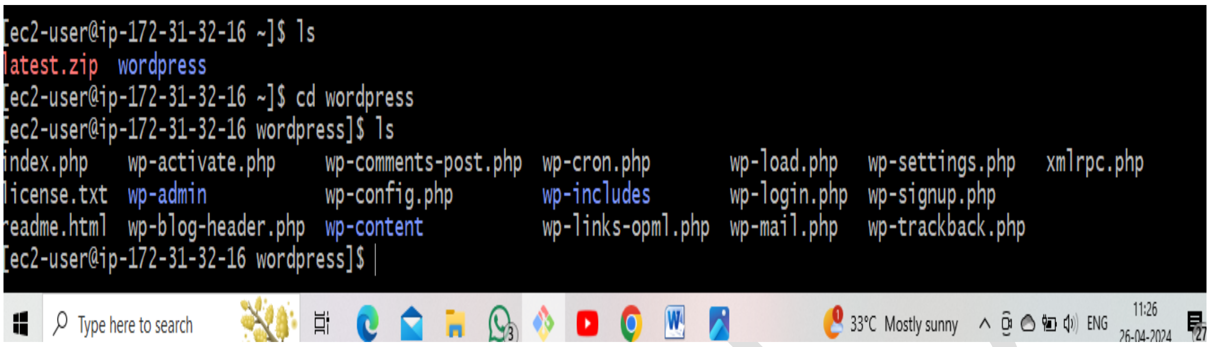


```
          =stable ]
 15 *php7.2=latest               enabled      \
        [ =7.2.0  =7.2.4  =7.2.5  =7.2.8  =7.2.11  =7.2.13  =7.2.14
          =7.2.16  =7.2.17  =7.2.19  =7.2.21  =7.2.22  =7.2.23
          =7.2.24  =7.2.26  =stable ]
 17 *lamp-mariadb10.2-php7.2=latest  enabled      \
        [ =10.2.10_7.2.0  =10.2.10_7.2.4  =10.2.10_7.2.5
          =10.2.10_7.2.8  =10.2.10_7.2.11  =10.2.10_7.2.13
          =10.2.10_7.2.14  =10.2.10_7.2.16  =10.2.10_7.2.17
          =10.2.10_7.2.19  =10.2.10_7.2.22  =10.2.10_7.2.23
          =10.2.10_7.2.24  =stable ]
 18  libreoffice                  available    \
        [ =5.0.6.2_15  =5.3.6.1  =stable ]
```

- **Now go inside the unzip directory by using command as**

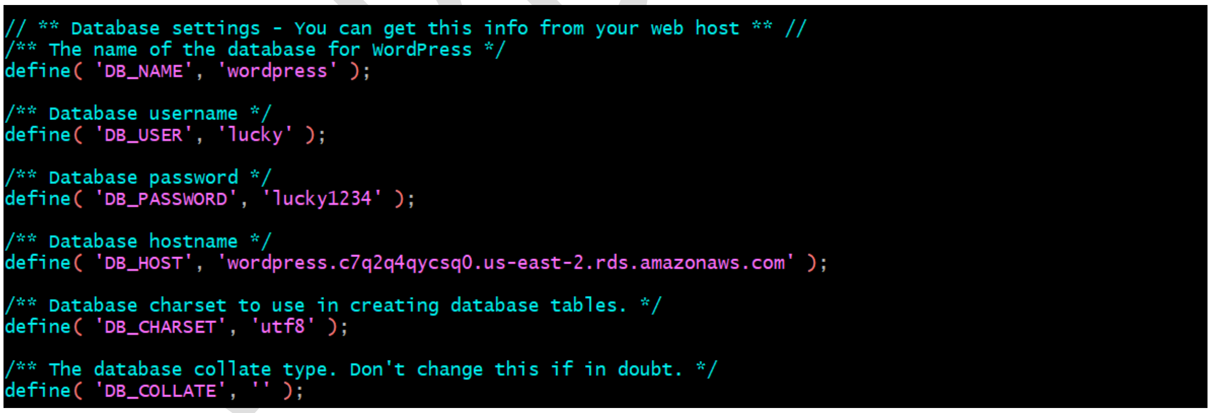
  < cd (unzip directory) >

- **Now change the wordpress configuration file by giving command as**
  < sudo mv wp-config-sample.php wp-config.php>



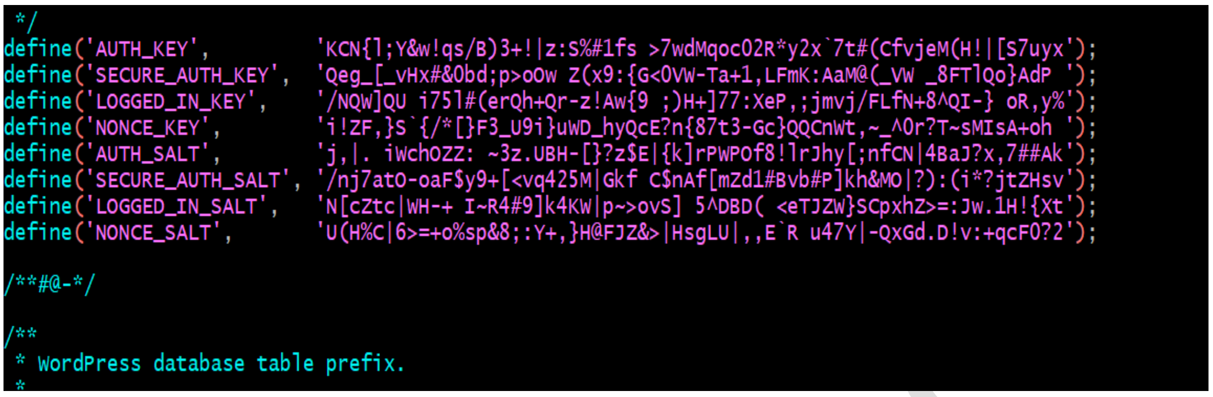
```
[ec2-user@ip-172-31-32-16 ~]$ ls
latest.zip  wordpress
[ec2-user@ip-172-31-32-16 ~]$ cd wordpress
[ec2-user@ip-172-31-32-16 wordpress]$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config.php         wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
[ec2-user@ip-172-31-32-16 wordpress]$
```

- **Now some configurations in wordpress configuration file as by giving database name, username, password and host name and wordpress keys.**
  < Sudo vi wp-config.php>



```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'lucky' );

/** Database password */
define( 'DB_PASSWORD', 'lucky1234' );

/** Database hostname */
define( 'DB_HOST', 'wordpress.c7q2q4qycsq0.us-east-2.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

- **Go to Google searching wordpress keys.**



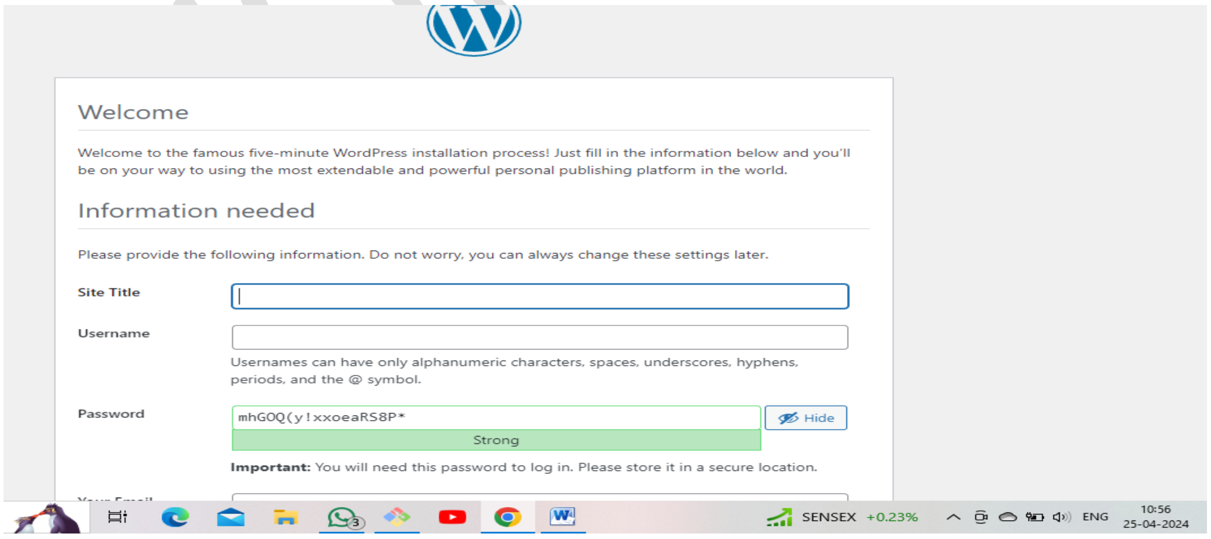
```
 */
define('AUTH_KEY',         'KCN{l;Y&w!qs/B)3+!|z:S%#1fs >7wdMqoc02R*y2x`7t#(CfvjeM(H!|[S7uyx');
define('SECURE_AUTH_KEY',  'Qeg_[_vHx#&0bd;p>oOw Z(x9:{G<0VW-Ta+1,LFmK:AaM@(_VW _8FTlQo}AdP ');
define('LOGGED_IN_KEY',    '/NQW]QU i75l#(erQh+Qr-z!Aw{9 ;)H+]77:XeP,;jmvj/FLfN+8^QI-} oR,y%');
define('NONCE_KEY',        'i!ZF,}S`{/*[}F3_U9i}uWD_hyQcE?n{87t3-Gc}QQCnWt,~_^0r?T~sMIsA+oh ');
define('AUTH_SALT',        'j,|. iWchOZZ: ~3z.UBH-[}?z$E|{k]rPWPOf8!lrJhy[;nfCN|4BaJ?x,7##Ak');
define('SECURE_AUTH_SALT', '/nj7atO-oaF$y9+[<vq425M|Gkf C$nAf[mZd1#Bvb#P]kh&MO|?):(i*?jtZHsv');
define('LOGGED_IN_SALT',   'N[cZtc|WH-+ I~R4#9]k4KW|p~>ovS] 5^DBD( <eTJZW}SCpxhZ>=:Jw.1H!{Xt');
define('NONCE_SALT',       'U(H%C|6>=+o%sp&8;:Y+,}H@FJZ&>|HsgLU|,,E`R u47Y|-QxGd.D!v:+qcF0?2');

/**#@-*/

/**
 * WordPress database table prefix.
 *
```

- **Now copy this wordpress directory to the document root directory to host web application of wordpress by giving a command as**

  < sudo cp –r (wordpress or unzip directory)/* /var/www/html/ >

- **Restart the httpd by giving a command as**

  < sudo systemctl restart httpd >



Welcome

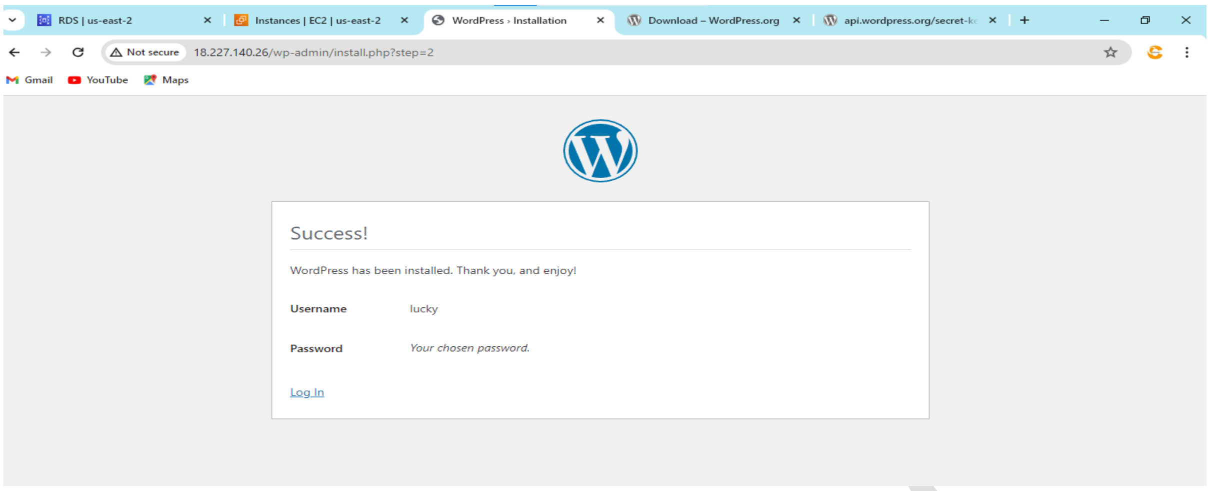
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title	
Username	Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	mhGOQ(y!xxoeaRS8P* Strong

Important: You will need this password to log in. Please store it in a secure location.



18.227.140.26/wp-admin/install.php?step=2

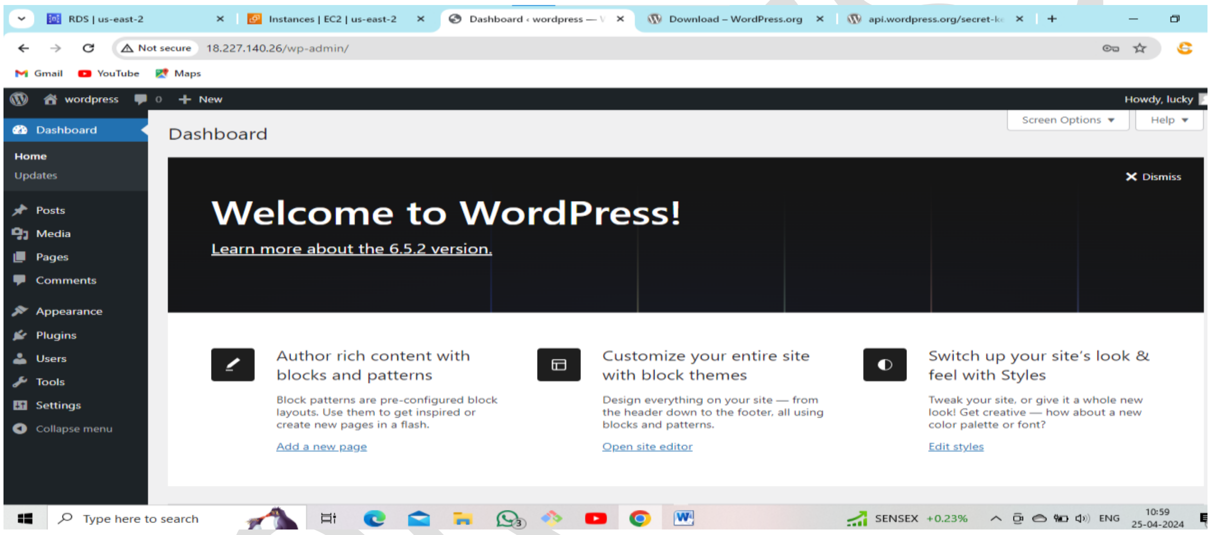
Success!

WordPress has been installed. Thank you, and enjoy!

Username lucky

Password *Your chosen password.*

Log In



18.227.140.26/wp-admin/

Dashboard

Welcome to WordPress!

Learn more about the 6.5.2 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

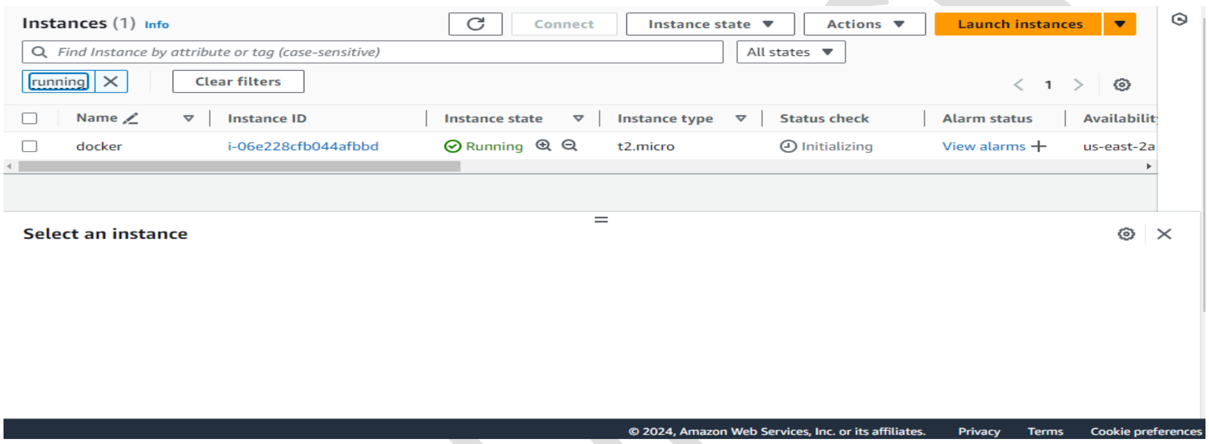
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

****

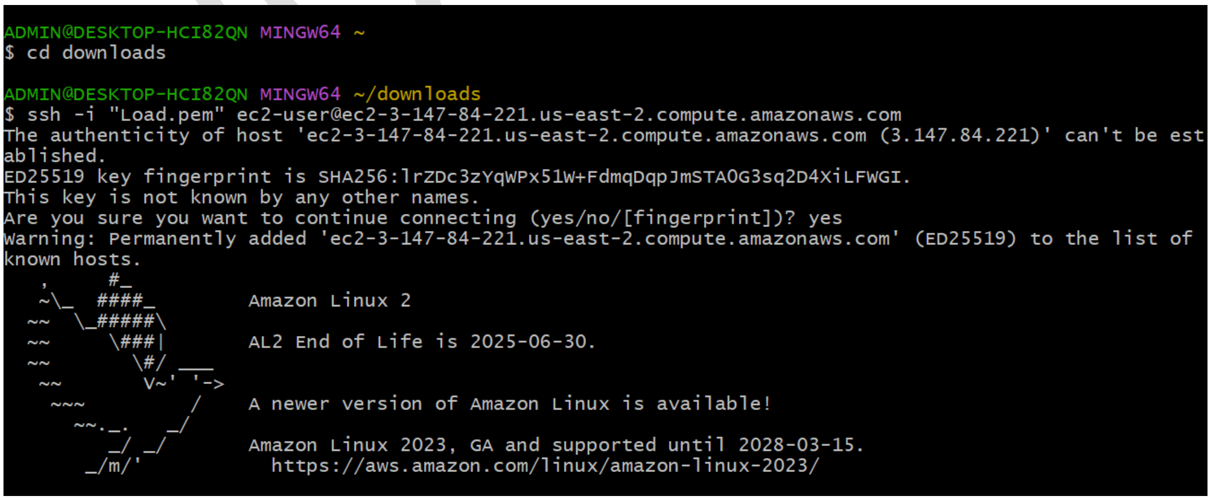
2. Deploy WordPress web application by using docker compose file?

- Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.mediam and giving security group HTTP (80) and security group mysql aurora and port 8080.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabilit
docker	i-06e228cfb044afbbd	Running	t2.micro	Initializing	View alarms +	us-east-2a

- Now connect the virtual server through the GitBash as shown in below.



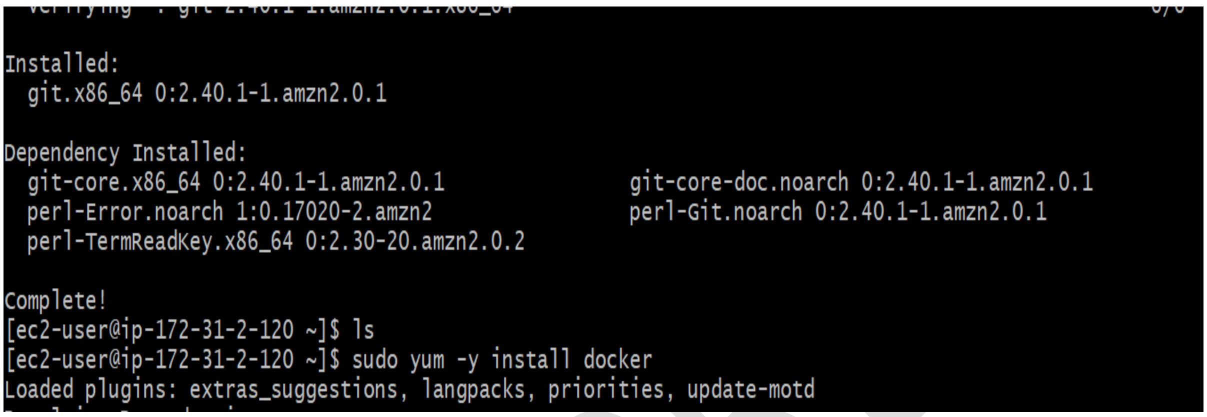
```
ADMIN@DESKTOP-HCI82QN MINGW64 ~
$ cd downloads

ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "Load.pem" ec2-user@ec2-3-147-84-221.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-147-84-221.us-east-2.compute.amazonaws.com (3.147.84.221)' can't be est
ablished.
ED25519 key fingerprint is SHA256:lrZDc3zYqWPx51W+FdmqDqpJmSTA0G3sq2D4XiLFWGI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-147-84-221.us-east-2.compute.amazonaws.com' (ED25519) to the list of
known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

- Run these following command.

  < sudo yum –y update >

  Now installing git using following command as

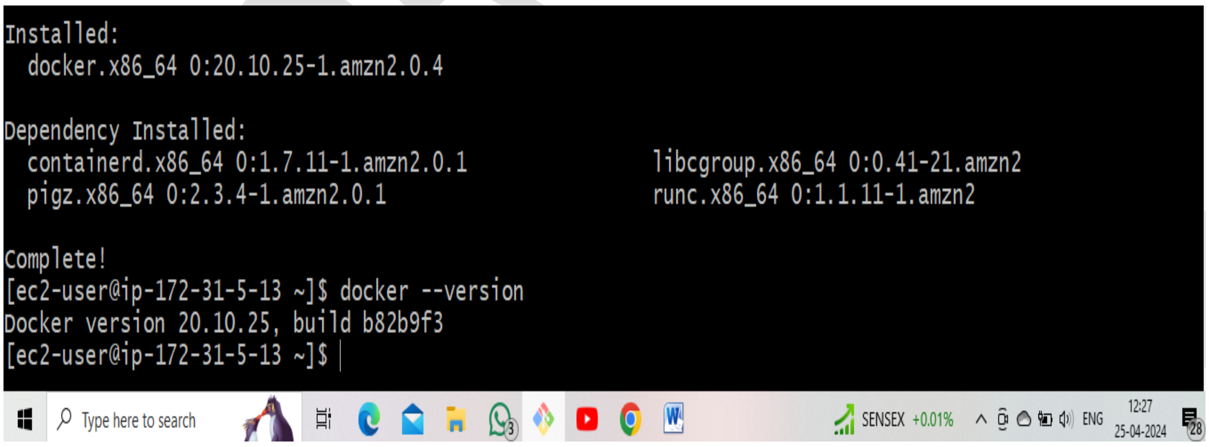
  < sudo yum –y install git >



```
Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.1

Dependency Installed:
  git-core.x86_64 0:2.40.1-1.amzn2.0.1          git-core-doc.noarch 0:2.40.1-1.amzn2.0.1
  perl-Error.noarch 1:0.17020-2.amzn2           perl-Git.noarch 0:2.40.1-1.amzn2.0.1
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-2-120 ~]$ ls
[ec2-user@ip-172-31-2-120 ~]$ sudo yum -y install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

- Now installing Docker using these following command.

  < sudo yum –y install docker >



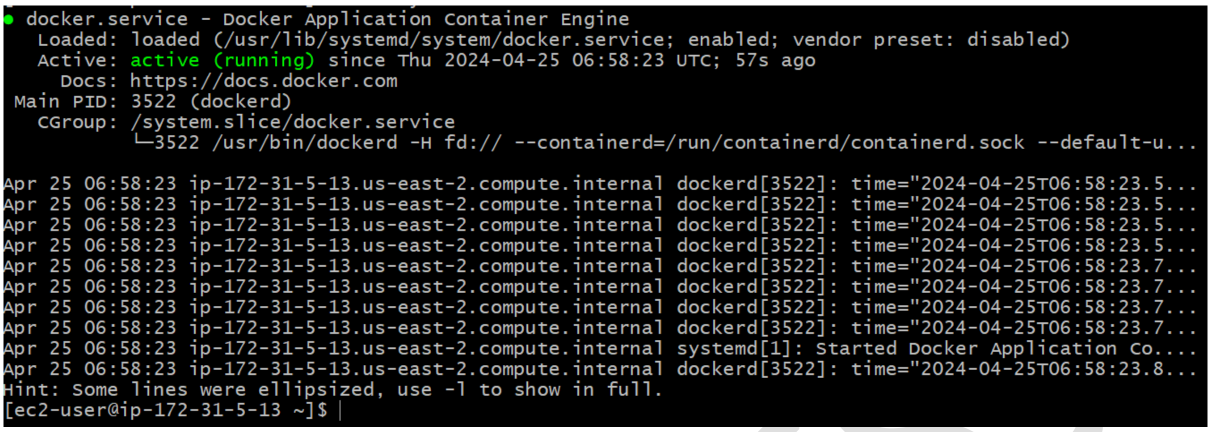
```
Installed:
  docker.x86_64 0:20.10.25-1.amzn2.0.4

Dependency Installed:
  containerd.x86_64 0:1.7.11-1.amzn2.0.1          libcgroup.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1                 runc.x86_64 0:1.1.11-1.amzn2

Complete!
[ec2-user@ip-172-31-5-13 ~]$ docker --version
Docker version 20.10.25, build b82b9f3
[ec2-user@ip-172-31-5-13 ~]$ |
```

- Run these following commands as Docker start,enable and status.

  < sudo systemctl start docker >

< sudo systemctl enable docker >

< sudo systemctl status docker >



```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-04-25 06:58:23 UTC; 57s ago
     Docs: https://docs.docker.com
 Main PID: 3522 (dockerd)
   CGroup: /system.slice/docker.service
           └─3522 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-u...

Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.5...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.5...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.5...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.5...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.7...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.7...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.7...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.7...
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal systemd[1]: Started Docker Application Co....
Apr 25 06:58:23 ip-172-31-5-13.us-east-2.compute.internal dockerd[3522]: time="2024-04-25T06:58:23.8...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-5-13 ~]$ |
```

- **Now give permissions to add a limited linux user account to docker group by using a command as**

  < sudo chmod 666 /var/run/docker.sock >

- **Now install the docker-compose file by using these command as**

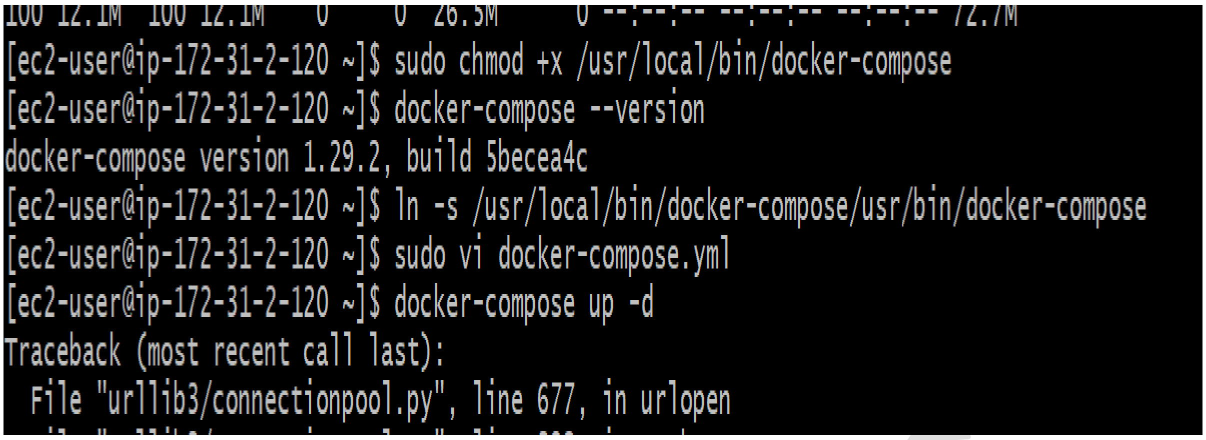
  < sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose >

- **Set execution permissions using these command as**

  < sudo chmod +x /usr/local/bin/docker-compose >

- **Verify Installation using these command as**

  < docker-compose –version >



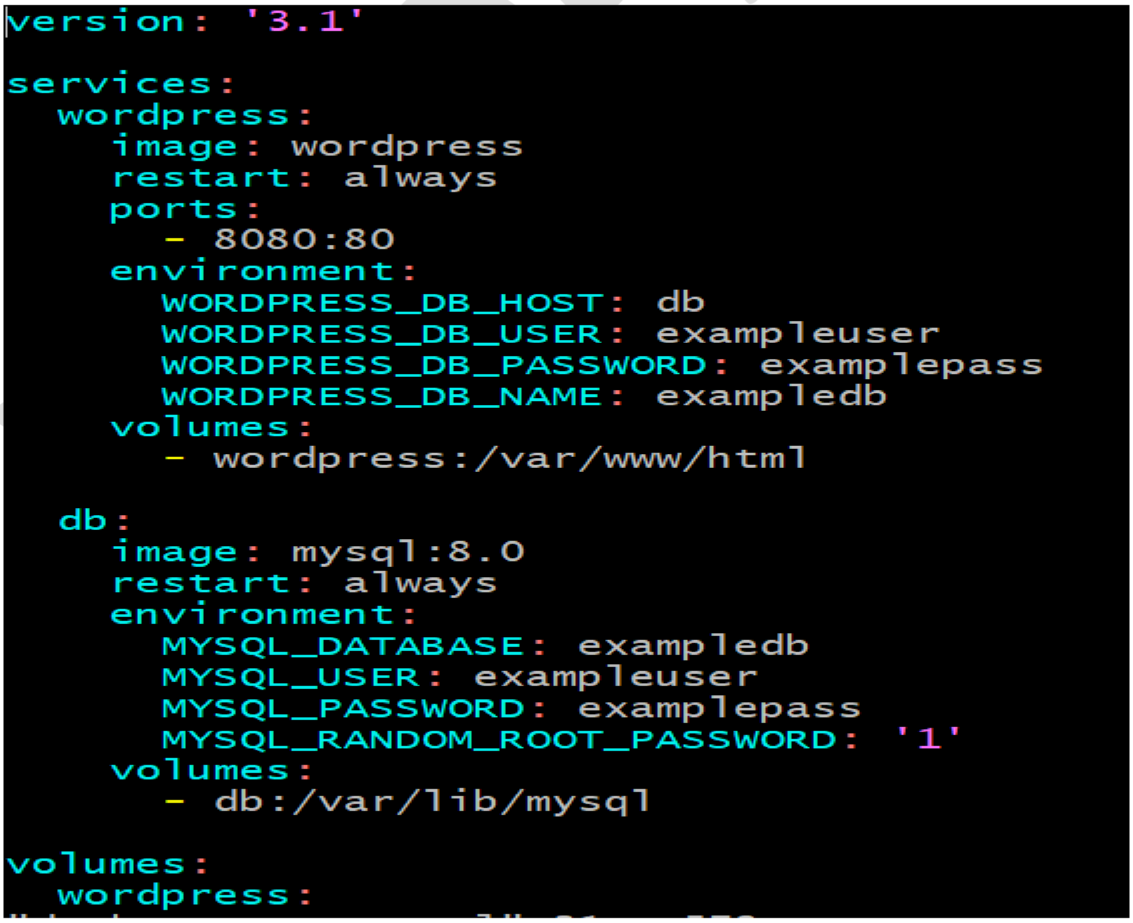
```
[ec2-user@ip-172-31-2-120 ~]$ sudo chmod +x /usr/local/bin/docker-compose
[ec2-user@ip-172-31-2-120 ~]$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
[ec2-user@ip-172-31-2-120 ~]$ ln -s /usr/local/bin/docker-compose/usr/bin/docker-compose
[ec2-user@ip-172-31-2-120 ~]$ sudo vi docker-compose.yml
[ec2-user@ip-172-31-2-120 ~]$ docker-compose up -d
Traceback (most recent call last):
  File "urllib3/connectionpool.py", line 677, in urlopen
```

- Now create the symbolic link by using command as

  < ln-s /usr/local/bin/docker-compose/usr/bin/docker-compose >

- Now to create a docker-compose.yml file in vi mode as

  < sudo vi docker-compose.yml >



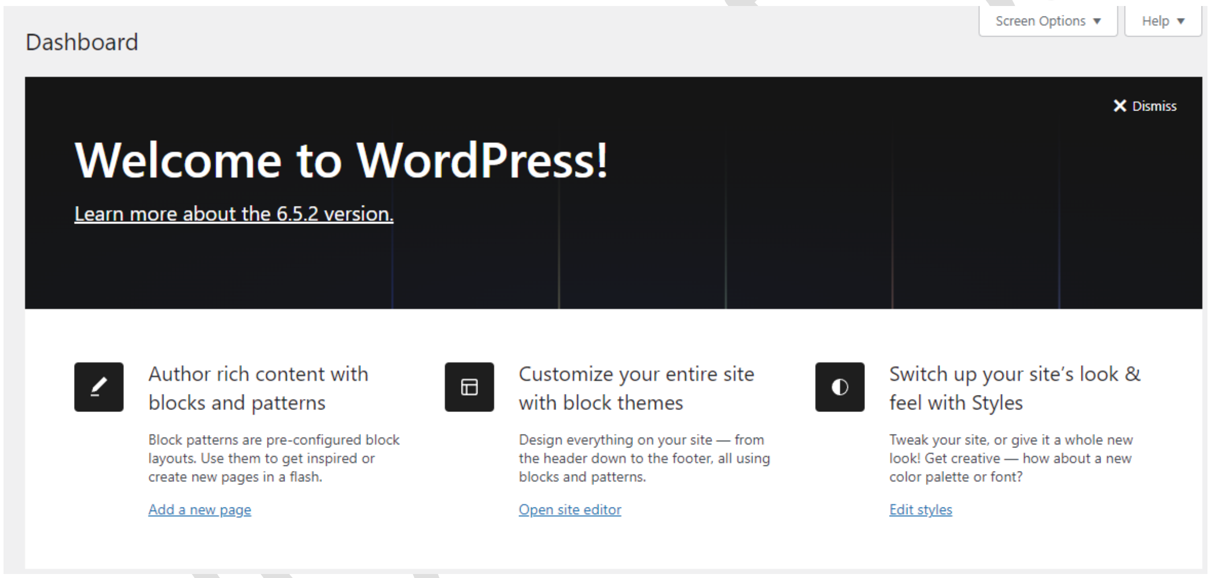
```
version: '3.1'

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:8.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
```

- Now **docker execute** a command within a running docker container as

  < docker-compose up –d >
- Now once again the **docker restart** command as

  < sudo service docker restart >
- Go to EC2 instance **copy the public ip** and paste with Google.

  < public ip:8080 (port) >



Welcome to WordPress!

Learn more about the 6.5.2 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

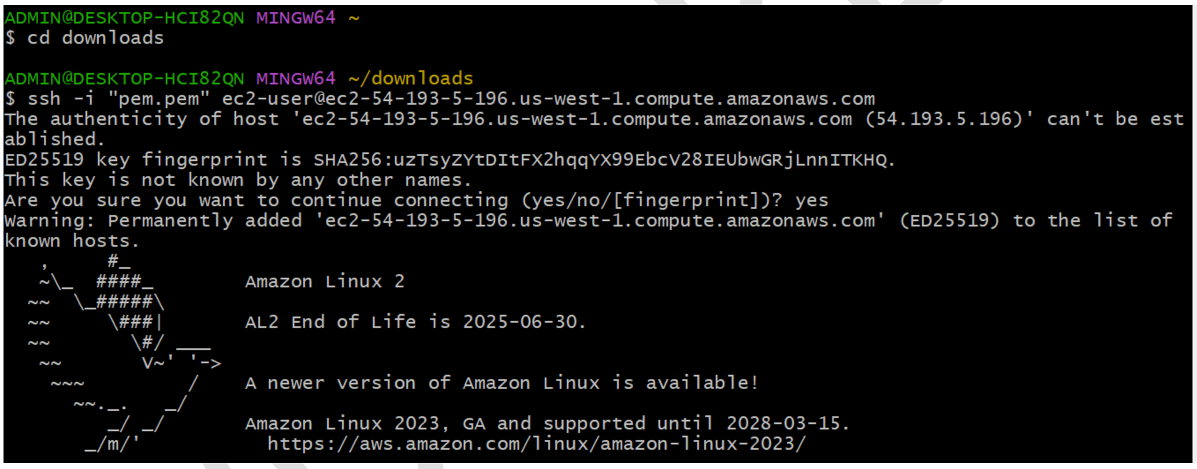
Tweak your site, or give it a whole new look! Get creative — how about a new color palette or font?

Edit styles

****

3. Deploy WordPress web application by using git and jenkins?

- **Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.mediam and giving security group HTTP (80) and security group mysql aurora and port 8080 and 8081.**
- **Now connect the virtual server through the GitBash as shown in below.**

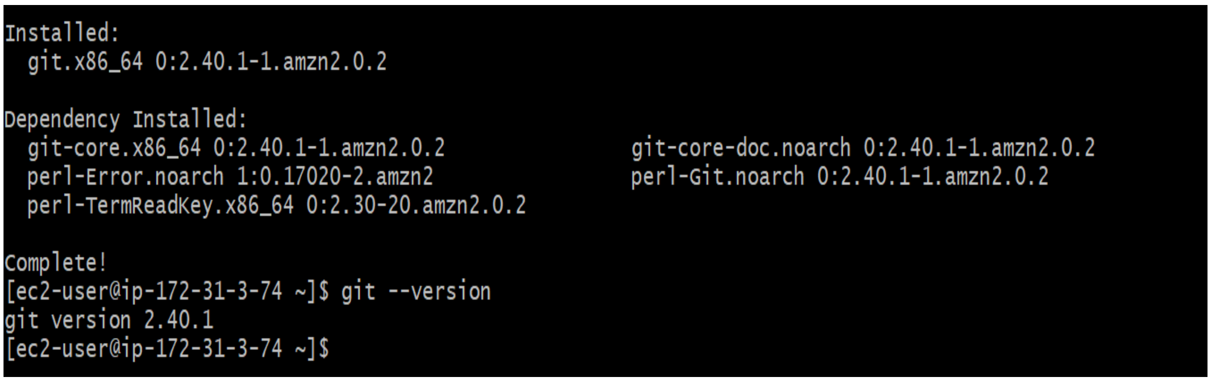


```
ADMIN@DESKTOP-HCI82QN MINGW64 ~
$ cd downloads

ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "pem.pem" ec2-user@ec2-54-193-5-196.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-54-193-5-196.us-west-1.compute.amazonaws.com (54.193.5.196)' can't be est
ablished.
ED25519 key fingerprint is SHA256:uzTsyZYtDItFX2hqqYX99EbcV28IEUbwGRjLnnITKHQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-193-5-196.us-west-1.compute.amazonaws.com' (ED25519) to the list of
known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

- **Run these following command.**

  < sudo yum –y update >
- **Now installing git using following command as**

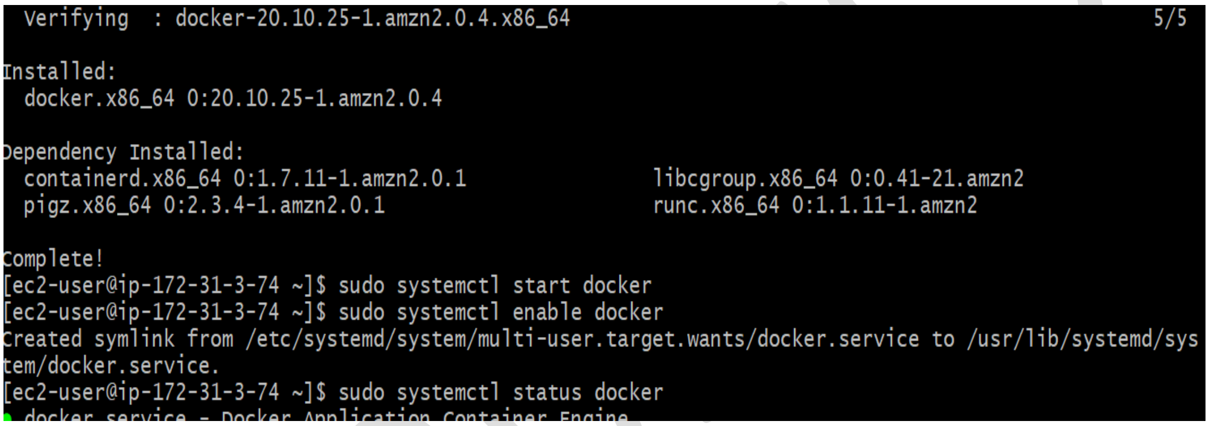
  < sudo yum –y install git >



```
Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.2

Dependency Installed:
  git-core.x86_64 0:2.40.1-1.amzn2.0.2                  git-core-doc.noarch 0:2.40.1-1.amzn2.0.2
  perl-Error.noarch 1:0.17020-2.amzn2                   perl-Git.noarch 0:2.40.1-1.amzn2.0.2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-3-74 ~]$ git --version
git version 2.40.1
[ec2-user@ip-172-31-3-74 ~]$
```

- **Now install the docker.**

  < sudo yum -y install docker >



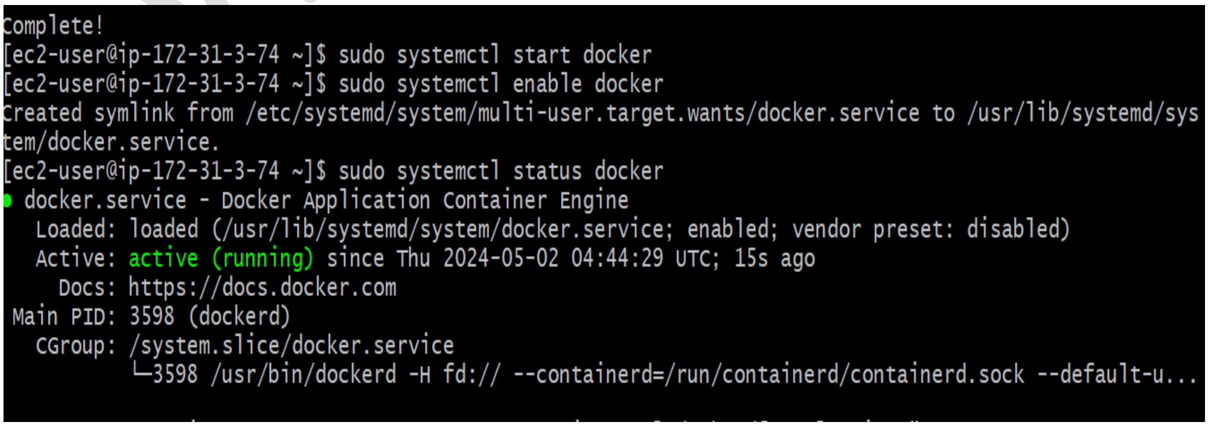
```
  Verifying  : docker-20.10.25-1.amzn2.0.4.x86_64                                                      5/5

Installed:
  docker.x86_64 0:20.10.25-1.amzn2.0.4

Dependency Installed:
  containerd.x86_64 0:1.7.11-1.amzn2.0.1                   libcgroup.x86_64 0:0.41-21.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1                          runc.x86_64 0:1.1.11-1.amzn2

Complete!
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/sys
tem/docker.service.
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
```

- **Run these following commands as Docker start,enable and status.**

  < sudo systemctl start docker >

  < sudo systemctl enable docker >

  < sudo systemctl status docker >



```
Complete!
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/sys
tem/docker.service.
[ec2-user@ip-172-31-3-74 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2024-05-02 04:44:29 UTC; 15s ago
     Docs: https://docs.docker.com
 Main PID: 3598 (dockerd)
   CGroup: /system.slice/docker.service
           └─3598 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-u...
```

- Now install the docker-compose file by using these command as

  < sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose>

- Set execution permissions using these command as

  < sudo chmod +x /usr/local/bin/docker-compose>

- verify installation using these command as

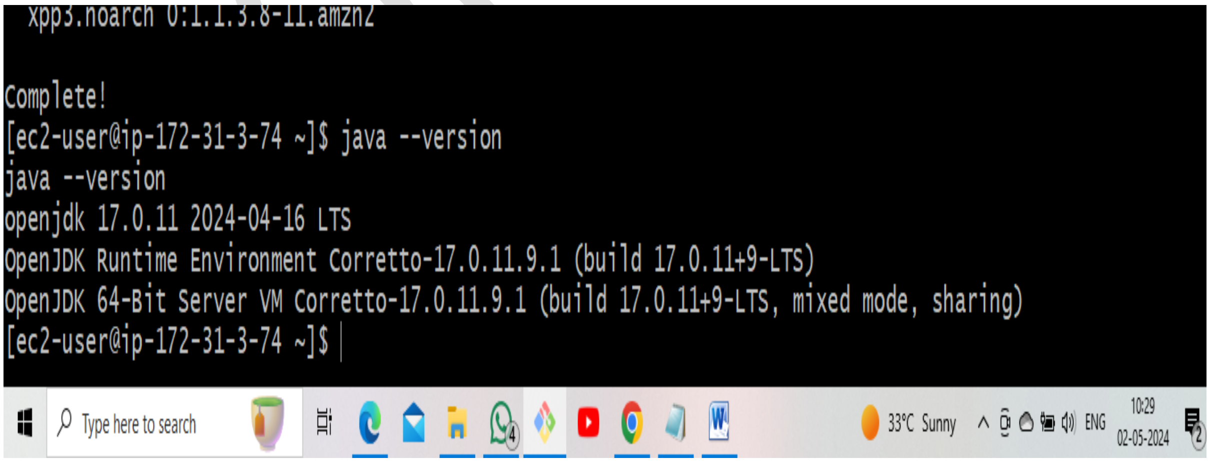
  < docker-compose –version >

- Now give permissions to add a limited linux user account to docker group by using a command as

  < sudo chmod 666 /var/run/docker.sock >

- Now installing java using following command as

  < sudo yum –y install java* >



```
 xpp3.noarch 0:1.1.3.8-11.amzn2

Complete!
[ec2-user@ip-172-31-3-74 ~]$ java --version
java --version
openjdk 17.0.11 2024-04-16 LTS
OpenJDK Runtime Environment Corretto-17.0.11.9.1 (build 17.0.11+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.11.9.1 (build 17.0.11+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-3-74 ~]$ 
```

- Now installing Jenkins following commands as

  < sudo wget -O /etc/yum.repos.d/jenkins.repo \
  https://pkg.jenkins.io/redhat-stable/jenkins.repo >

< sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key >

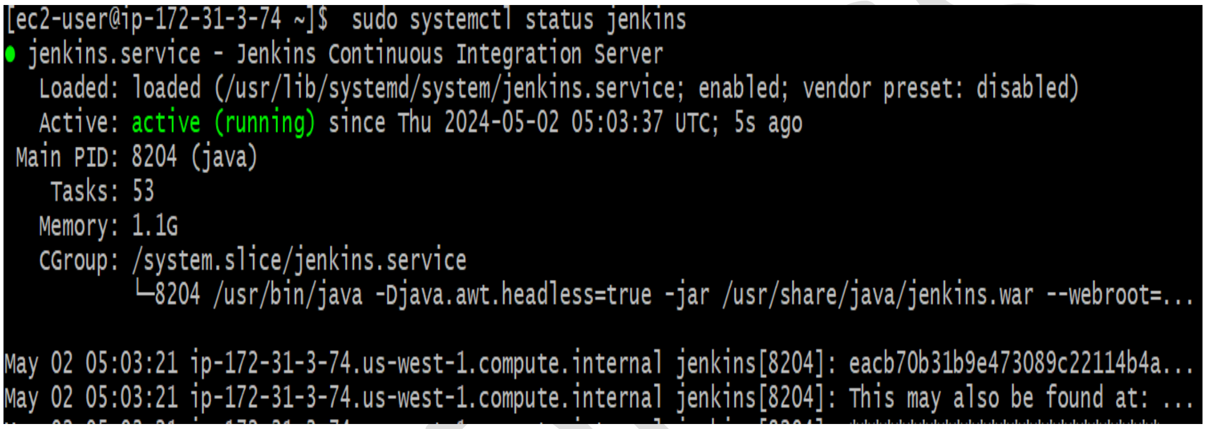
< sudo yum upgrade >

< sudo yum install jenkins –y >

< sudo systemctl start Jenkins >

< sudo systemctl enable Jenkins >

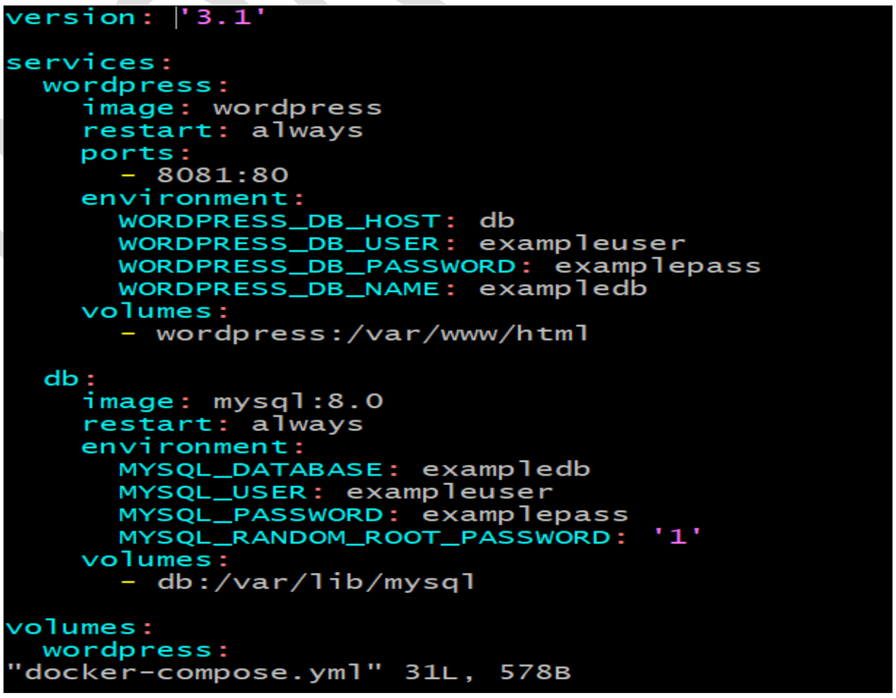
< sudo systemctl status Jenkins>



```
[ec2-user@ip-172-31-3-74 ~]$  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 Thu 2024-05-02 05:03:37 UTC; 5s ago
 Main PID: 8204 (java)
    Tasks: 53
   Memory: 1.1G
   CGroup: /system.slice/jenkins.service
           └─8204 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=...

May 02 05:03:21 ip-172-31-3-74.us-west-1.compute.internal jenkins[8204]: eacb70b31b9e473089c22114b4a...
May 02 05:03:21 ip-172-31-3-74.us-west-1.compute.internal jenkins[8204]: This may also be found at: ...
```

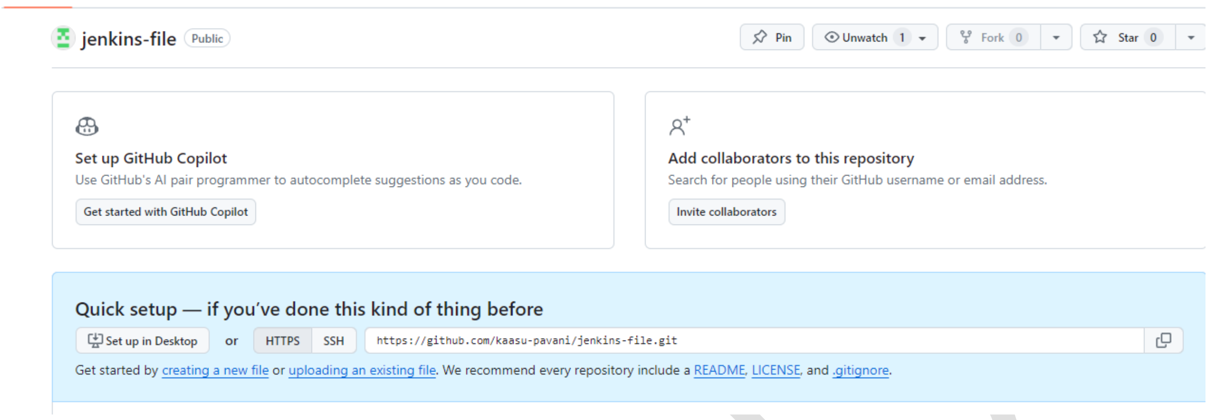
- Now to create a docker-compose.yml file in vi mode as

  < sudo vi docker-compose.yml >



```
version: '3.1'

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8081:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:8.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
"docker-compose.yml" 31L, 578B
```

- Now open the git hub.
- Create a new repository.



Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository
Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

```
https://github.com/kaasu-pavani/jenkins-file.git
```

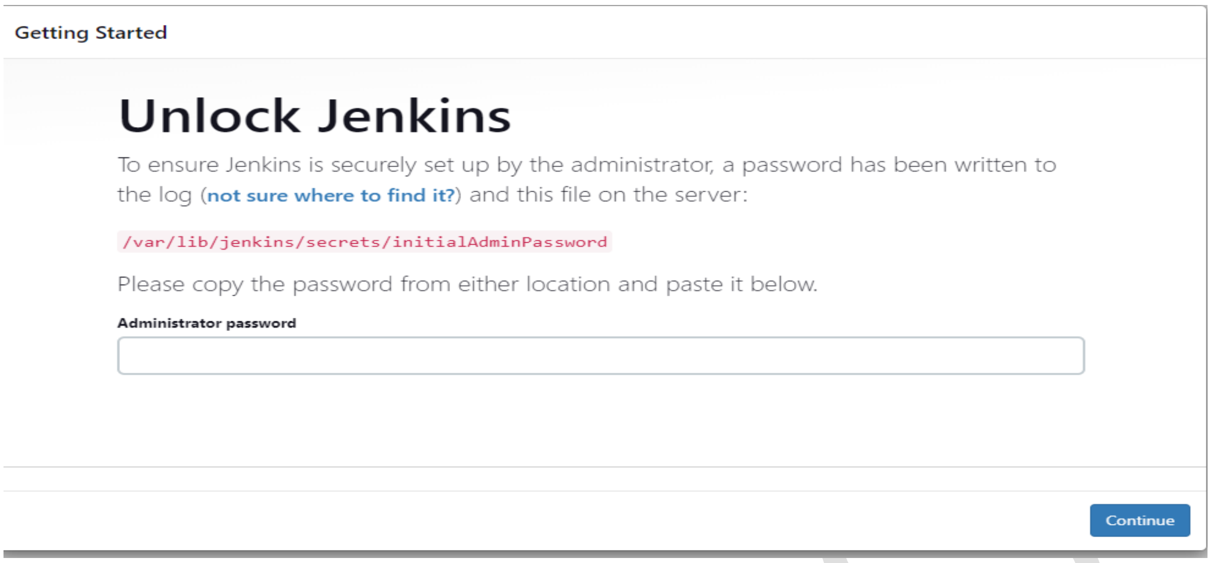
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

- Run these following commands as git Bash

< git init (git repository name) >

< sudo cp * docker-compose.yml (git repository name) >

< cd (git repository name) >
< git status >
< git add . >
< git commit –m “hello” docker-compose.yml >
< git remote add origin https://github.com/kaasu-pavani/jenkins-file.git >
< git push –all >
< Username: kaasu-pavani
Password:
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 (Personal access token) >



Getting Started

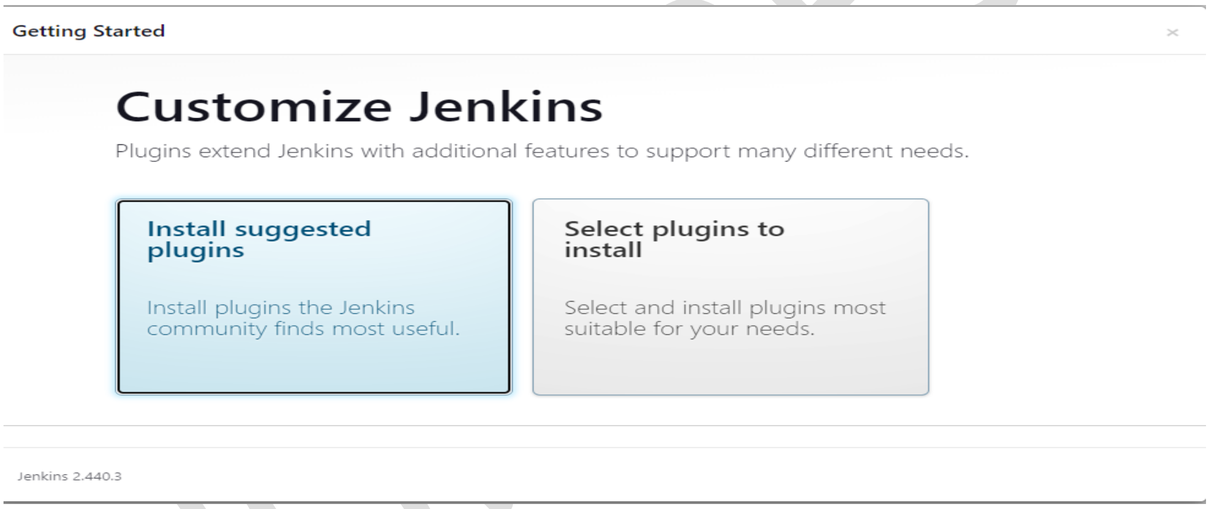
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

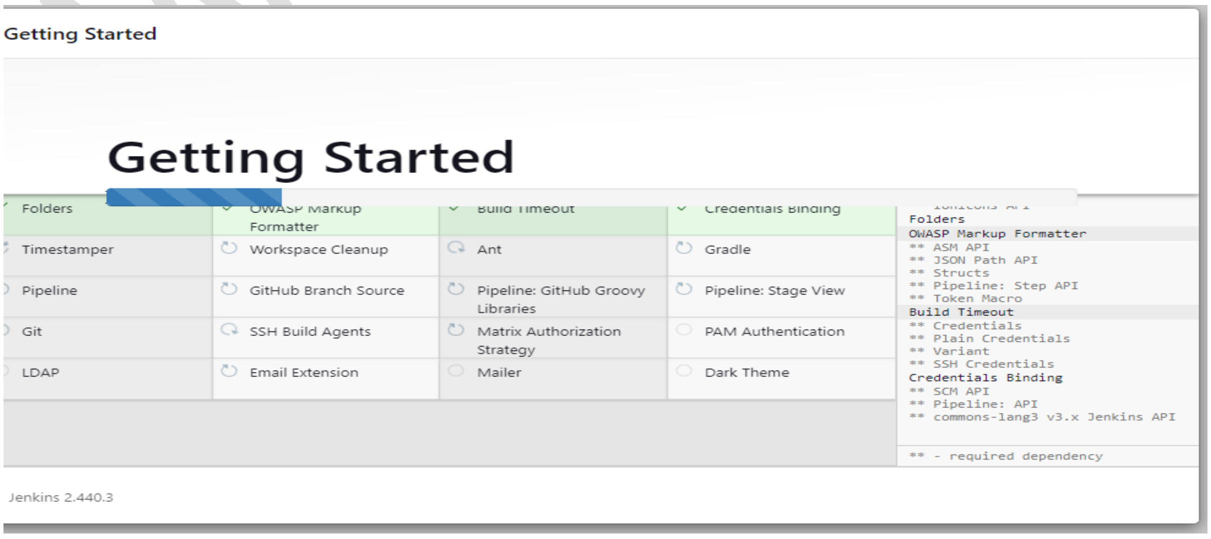
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.440.3



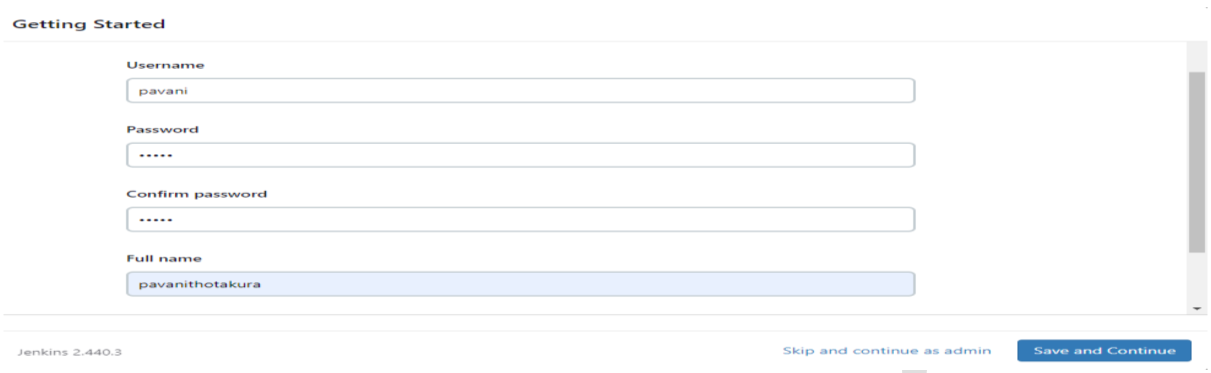
Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	Dark Theme

```
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** Credentials
** Plain Credentials
** Variant
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API

** - required dependency
```

Jenkins 2.440.3


Getting Started

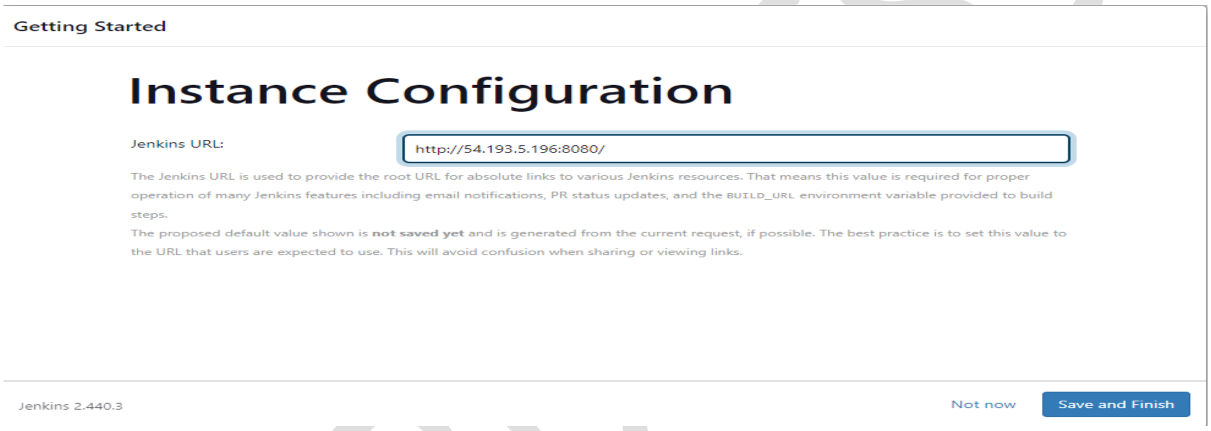
Username: pavani

Password: •••••

Confirm password: •••••

Full name: pavanithotakura

Jenkins 2.440.3


Getting Started

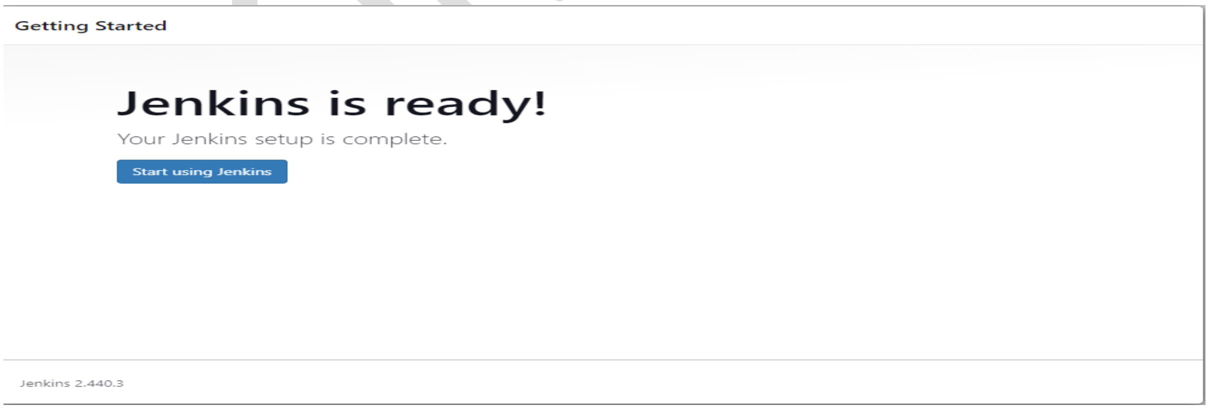
Instance Configuration

Jenkins URL: http://54.193.5.196:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

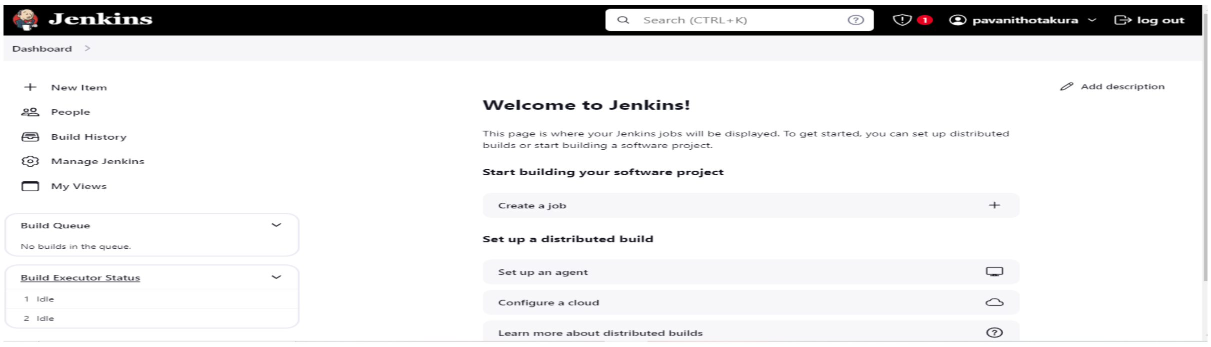
Jenkins 2.440.3


Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Jenkins 2.440.3



Jenkins

Search (CTRL+K) | pavanithotakura | log out

Dashboard >

- New Item
- People
- 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

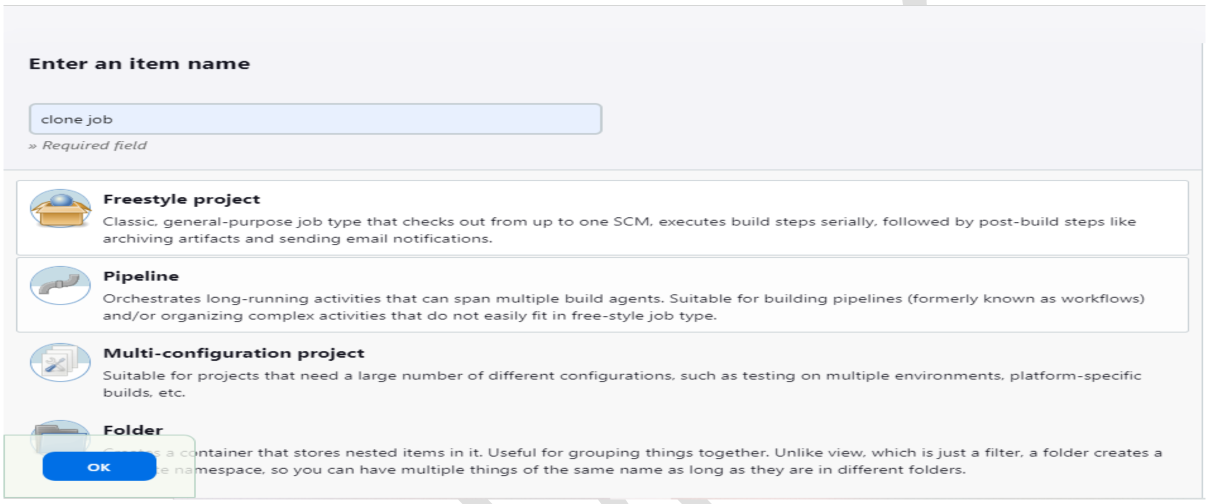
Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds



Enter an item name

clone job

» Required field

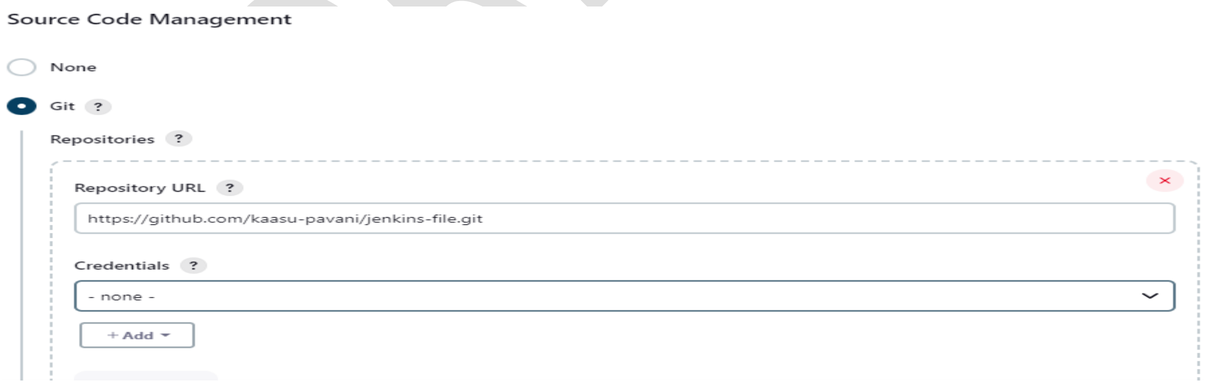
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK



Source Code Management

None [✓ Git]

Repositories

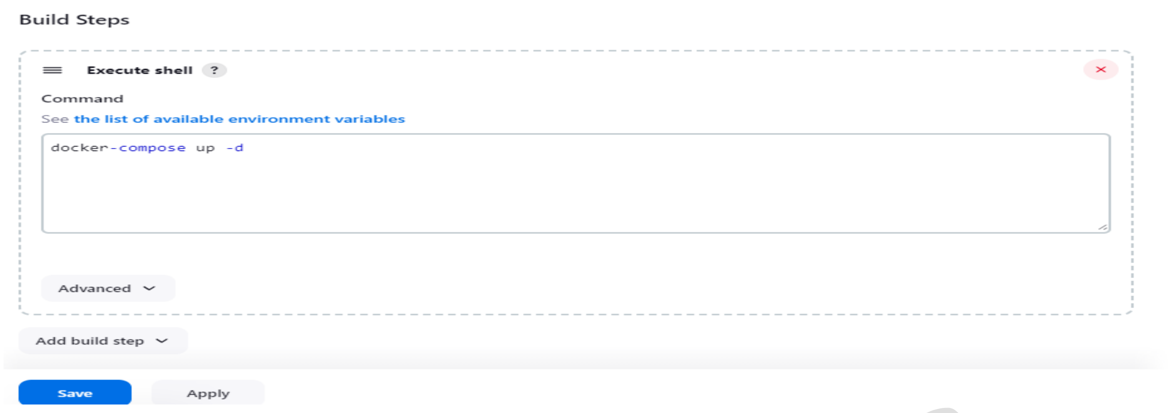
Repository URL

https://github.com/kaasu-pavani/jenkins-file.git

Credentials

- none -

+ Add



```
docker-compose up -d
```

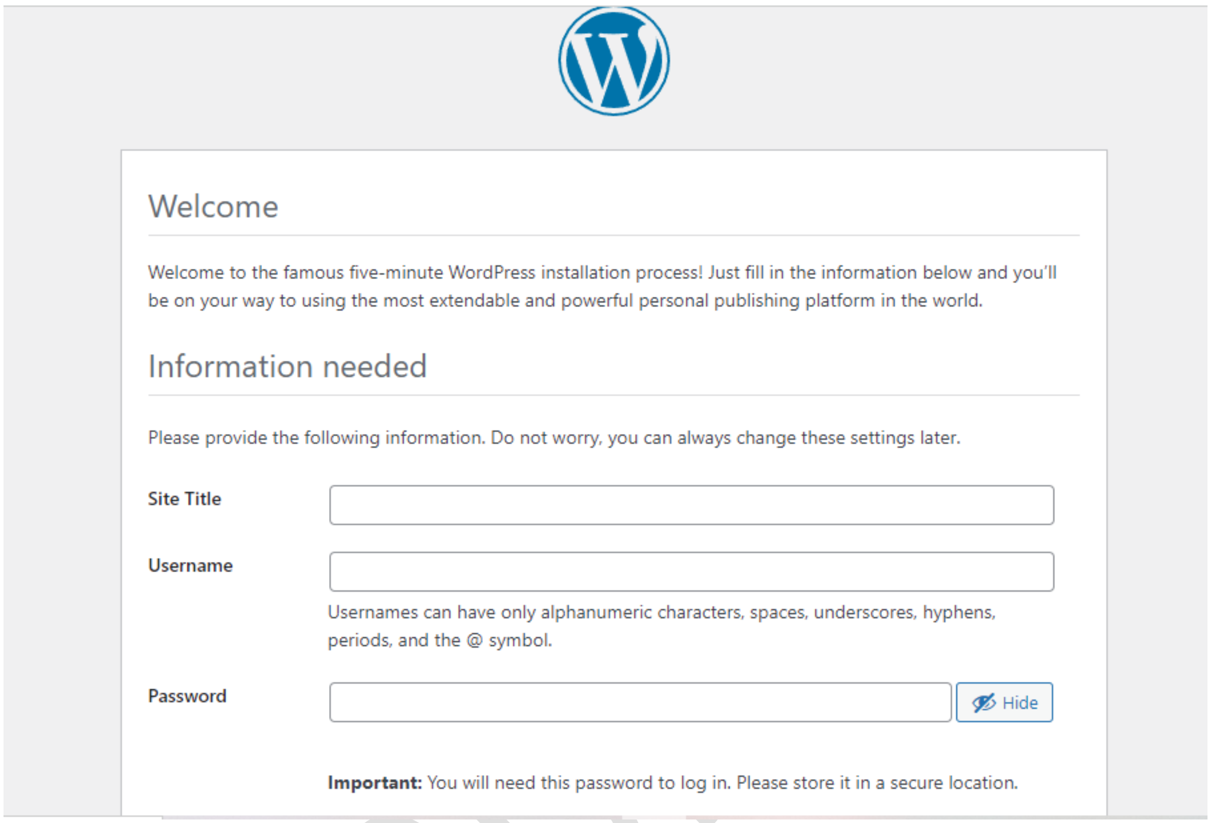
****

4. Deploy WordPress web application by using userdata of EC2 instance?

- Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.micro and giving security group HTTP (80) and HTTPs.
- Go to advanced details.
- Upload file in user data these following script.

```
#!/bin/bash

sudo yum -y install git docker

sudo systemctl start docker

sudo systemctl enable docker

sudo chmod 666 /var/run/docker.sock

sudo usermod -a -G docker ec2-user

sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$ (uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

git clone https://github.com/Hemayuva/wordpress.git

cd wordpress

docker-compose up -d
```

- **Copy public ip and paste to Google.**
- **Now open wordpress application.**



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

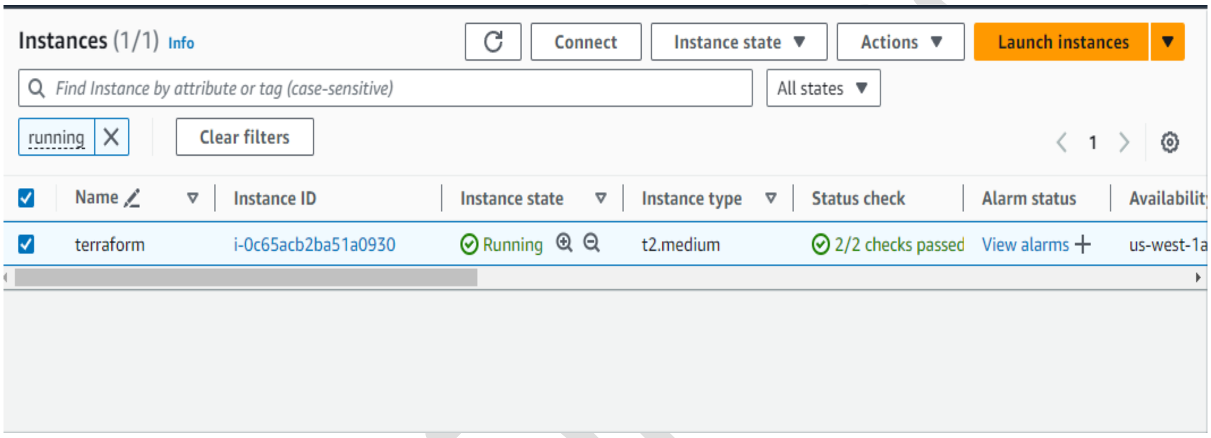
Password

Important: You will need this password to log in. Please store it in a secure location.

****

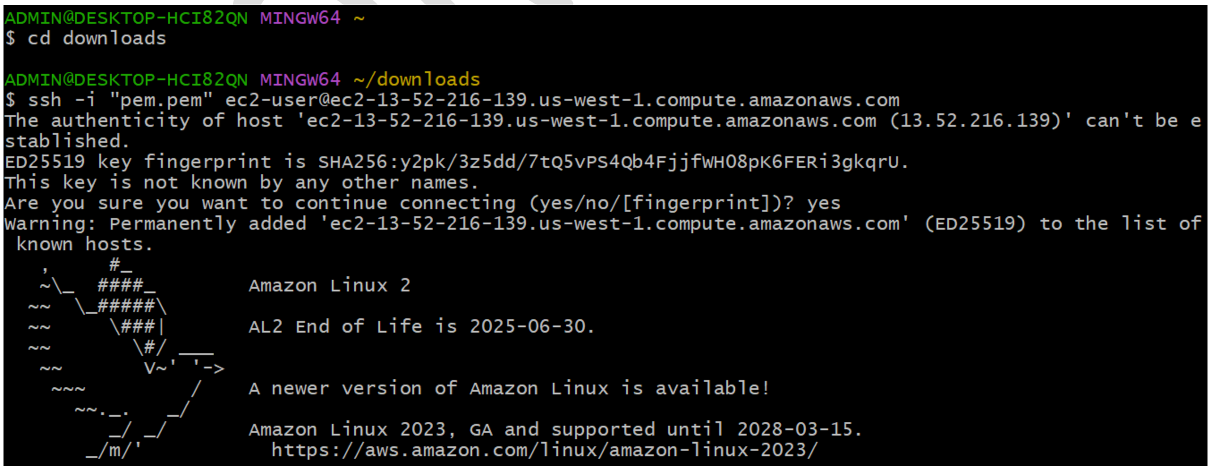
5. Deploy WordPress web application by using git and jenkins execute shell (bash script)?

- **Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.mediam and giving security group HTTP (80) and security group mysql aurora and port 8080 and 8081.**



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabilit
✓	terraform	i-0c65acb2ba51a0930	Running	t2.medium	2/2 checks passed	View alarms +	us-west-1a

- **Now connect the virtual server through the GitBash as shown in below.**



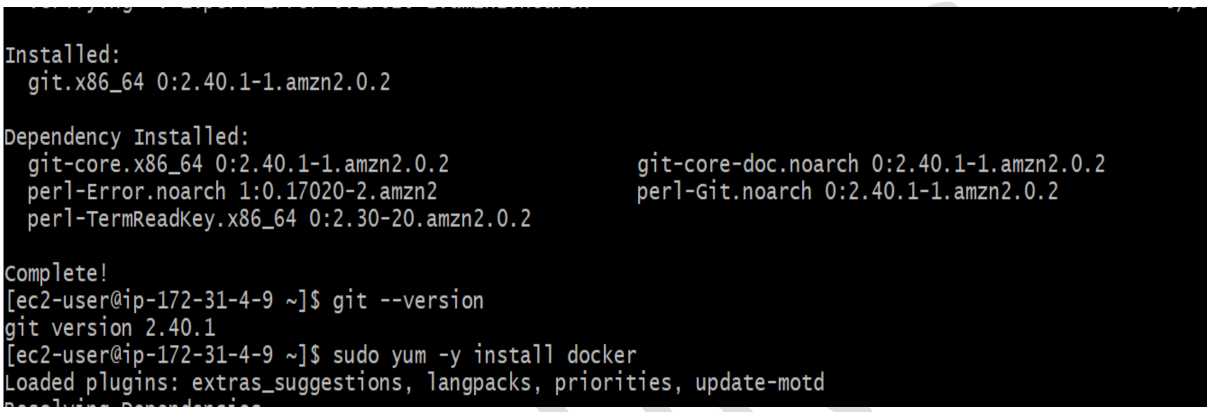
```
ADMIN@DESKTOP-HCI82QN MINGW64 ~
$ cd downloads

ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "pem.pem" ec2-user@ec2-13-52-216-139.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-13-52-216-139.us-west-1.compute.amazonaws.com (13.52.216.139)' can't be e
stablished.
ED25519 key fingerprint is SHA256:y2pk/3z5dd/7tQ5vPS4Qb4FjjfWHO8pK6FERi3gkqrU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-52-216-139.us-west-1.compute.amazonaws.com' (ED25519) to the list of
 known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

- **Run these following command as**

  < sudo yum –y update >

- **Now installing git using following command as**

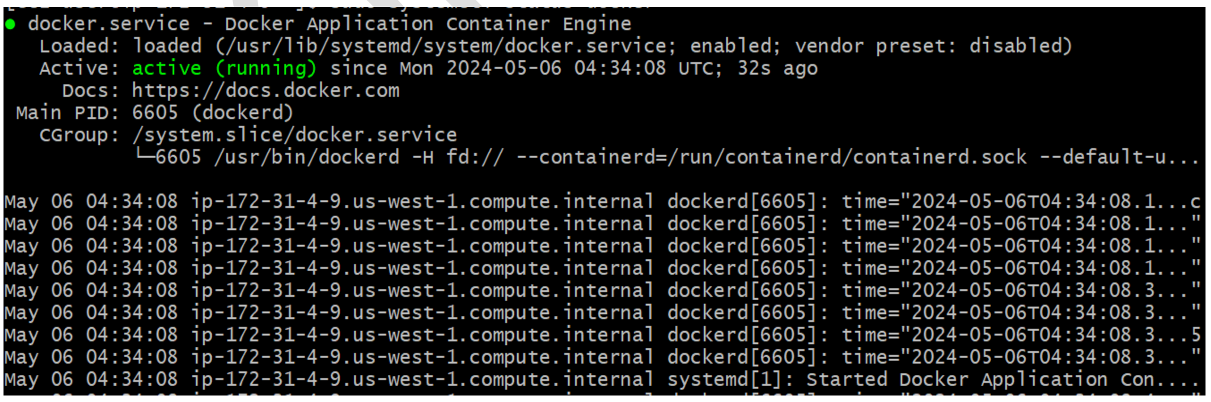
  < sudo yum –y install git >



```
Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.2

Dependency Installed:
  git-core.x86_64 0:2.40.1-1.amzn2.0.2              git-core-doc.noarch 0:2.40.1-1.amzn2.0.2
  perl-Error.noarch 1:0.17020-2.amzn2               perl-Git.noarch 0:2.40.1-1.amzn2.0.2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-4-9 ~]$ git --version
git version 2.40.1
[ec2-user@ip-172-31-4-9 ~]$ sudo yum -y install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

- **Now install the docker.**

  < sudo yum -y install docker >



```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2024-05-06 04:34:08 UTC; 32s ago
     Docs: https://docs.docker.com
 Main PID: 6605 (dockerd)
   CGroup: /system.slice/docker.service
           └─6605 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-u...

May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.1...c
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.1..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.1..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.1..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.3..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.3..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.3...5
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal dockerd[6605]: time="2024-05-06T04:34:08.3..."
May 06 04:34:08 ip-172-31-4-9.us-west-1.compute.internal systemd[1]: Started Docker Application Con....
```

- **Now install the docker-compose file by using these command as**

< sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$ (uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose >

- Set execution permissions using these command as

  < sudo chmod +x /usr/local/bin/docker-compose>

- Verify Installation using these command as

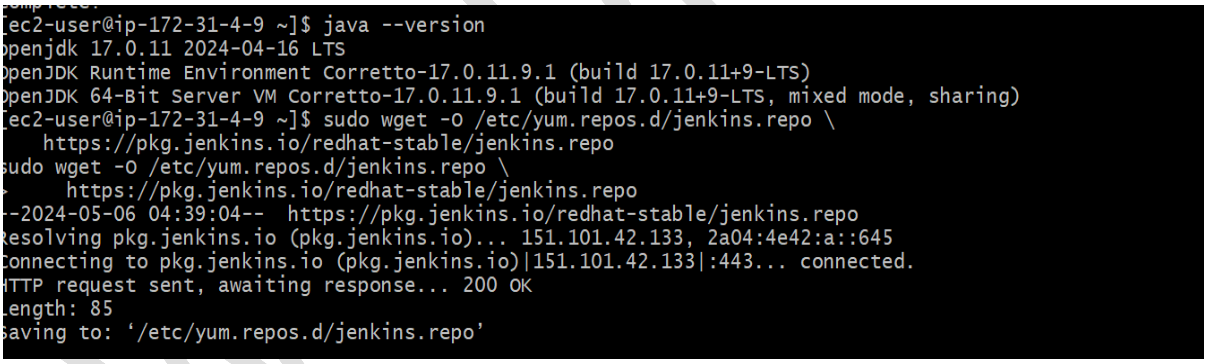
  < docker-compose –version >

- Now give permissions to add a limited linux user account to docker group by using a command as

  < sudo chmod 666 /var/run/docker.sock >

- Now installing java using following command as

  < sudo yum –y install java* >



```
[ec2-user@ip-172-31-4-9 ~]$ java --version
openjdk 17.0.11 2024-04-16 LTS
OpenJDK Runtime Environment Corretto-17.0.11.9.1 (build 17.0.11+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.11.9.1 (build 17.0.11+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-4-9 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo wget -O /etc/yum.repos.d/jenkins.repo \
>     https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-05-06 04:39:04--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.42.133, 2a04:4e42:a::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.42.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'
```

- Now installing Jenkins following commands as

  < sudo wget -O /etc/yum.repos.d/jenkins.repo \ https://pkg.jenkins.io/redhat-stable/jenkins.repo >

  < sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key >

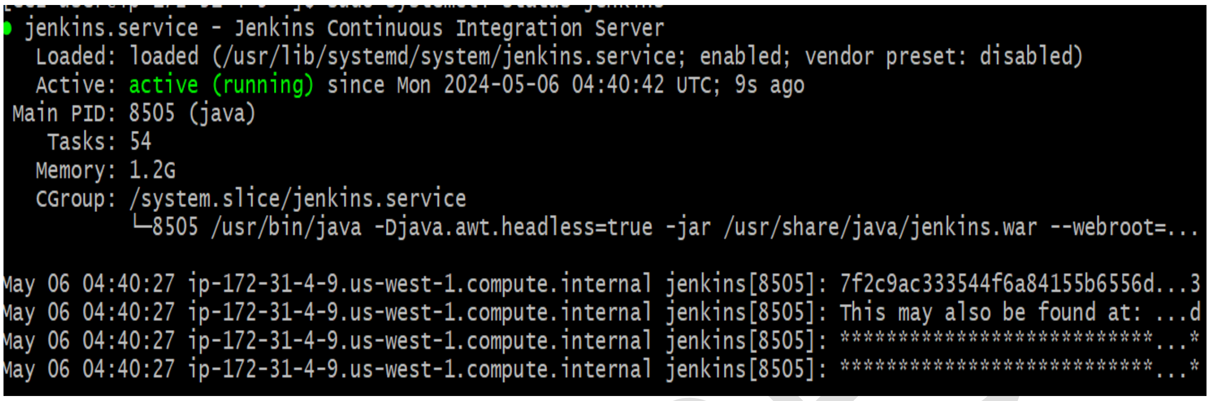
  < sudo yum upgrade >

  < sudo yum install jenkins -y >

< sudo systemctl start Jenkins >

< sudo systemctl enable Jenkins >

< 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 2024-05-06 04:40:42 UTC; 9s ago
 Main PID: 8505 (java)
    Tasks: 54
   Memory: 1.2G
   CGroup: /system.slice/jenkins.service
           └─8505 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=...

May 06 04:40:27 ip-172-31-4-9.us-west-1.compute.internal jenkins[8505]: 7f2c9ac333544f6a84155b6556d...3
May 06 04:40:27 ip-172-31-4-9.us-west-1.compute.internal jenkins[8505]: This may also be found at: ...d
May 06 04:40:27 ip-172-31-4-9.us-west-1.compute.internal jenkins[8505]: ***************************...*
May 06 04:40:27 ip-172-31-4-9.us-west-1.compute.internal jenkins[8505]: ***************************...*
```

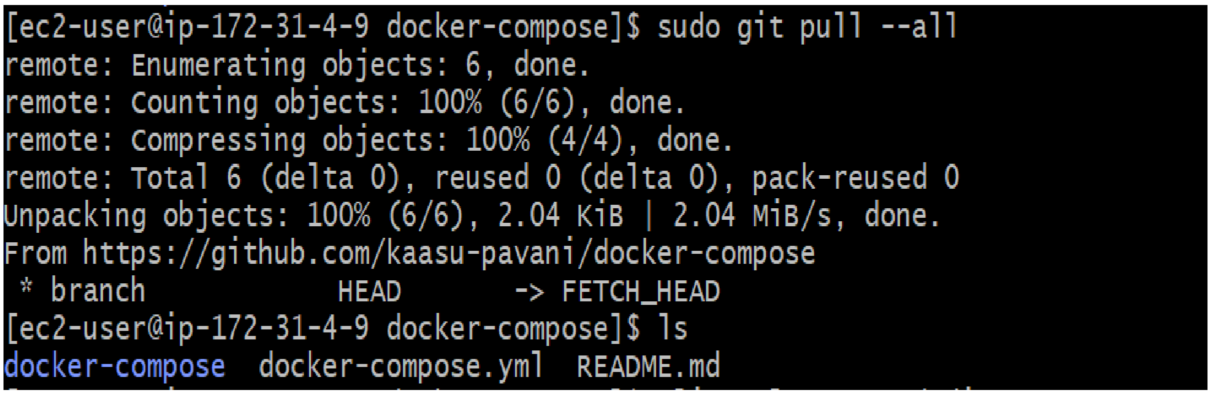
- ➢ **Now open the github repository.**
- ➢ **Create one repo and upload these following code.**

```
version: '3.1'
services:
wordpress:
image: wordpress
restart: always
ports:
- 80:80
environment:
WORDPRESS_DB_HOST: db
WORDPRESS_DB_USER: exampleuser
WORDPRESS_DB_PASSWORD: examplepass
WORDPRESS_DB_NAME: exampledb
 volumes:
 - wordpress:/var/www/html
```

```
        db:
        image: mysql:8.0
        restart: always
       environment:
       MYSQL_DATABASE: exampledb
       MYSQL_USER: exampleuser
       MYSQL_PASSWORD: examplepass
       MYSQL_RANDOM_ROOT_PASSWORD: 1 #
   removed quotes here
       volumes:
       - db:/var/lib/mysql
       expose:
       - 3306
       - 33060
       volumes:
      wordpress:
      db:
```

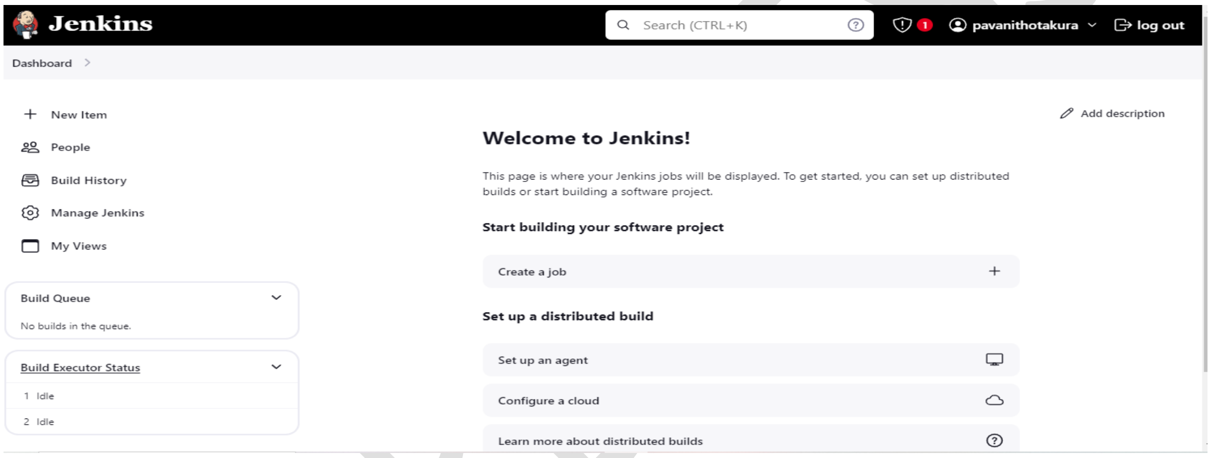
- Go to GitBash , run these following commands

```
     <sudo git init docker-compose>
     <cd docker-compose>
     < git add .>
    < git remote add origin
https://github.com/kaasu-pavani/docker-
compose.git>
    < git clone https://github.com/kaasu-pavani/docker-
 compose.git >
    < Sudo Git pull –all >
    < ls >
```



```
[ec2-user@ip-172-31-4-9 docker-compose]$ sudo git pull --all
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 2.04 KiB | 2.04 MiB/s, done.
From https://github.com/kaasu-pavani/docker-compose
 * branch            HEAD       -> FETCH_HEAD
[ec2-user@ip-172-31-4-9 docker-compose]$ ls
docker-compose  docker-compose.yml  README.md
```

➢ **Now sign in Jenkins**



No builds in the queue.

1 Idle

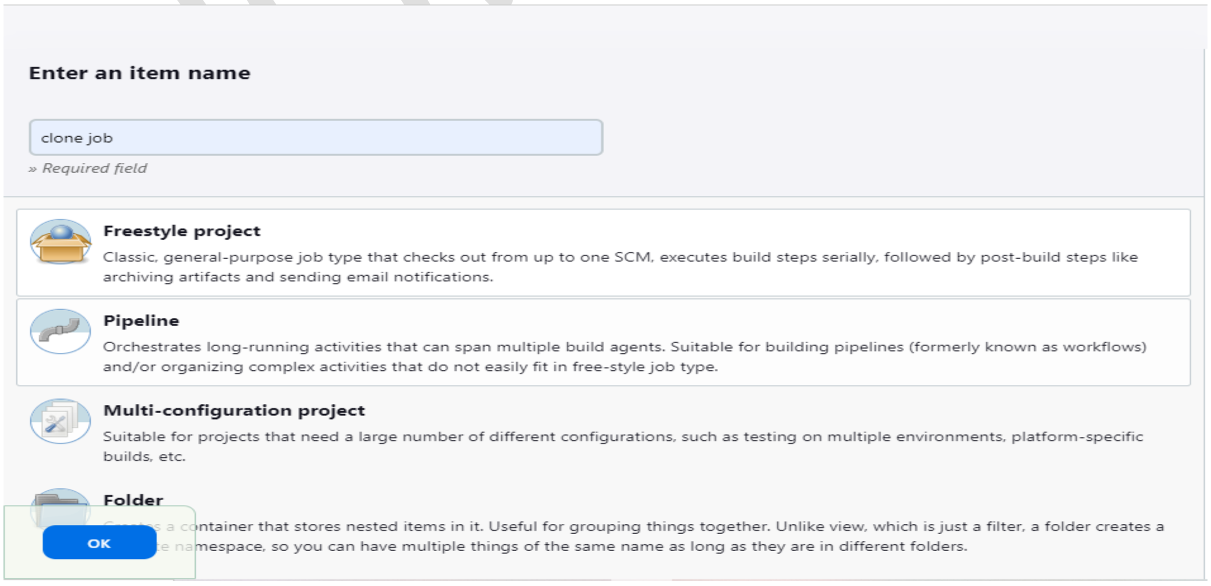
2 Idle

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

Set up a distributed build



Enter an item name

clone job

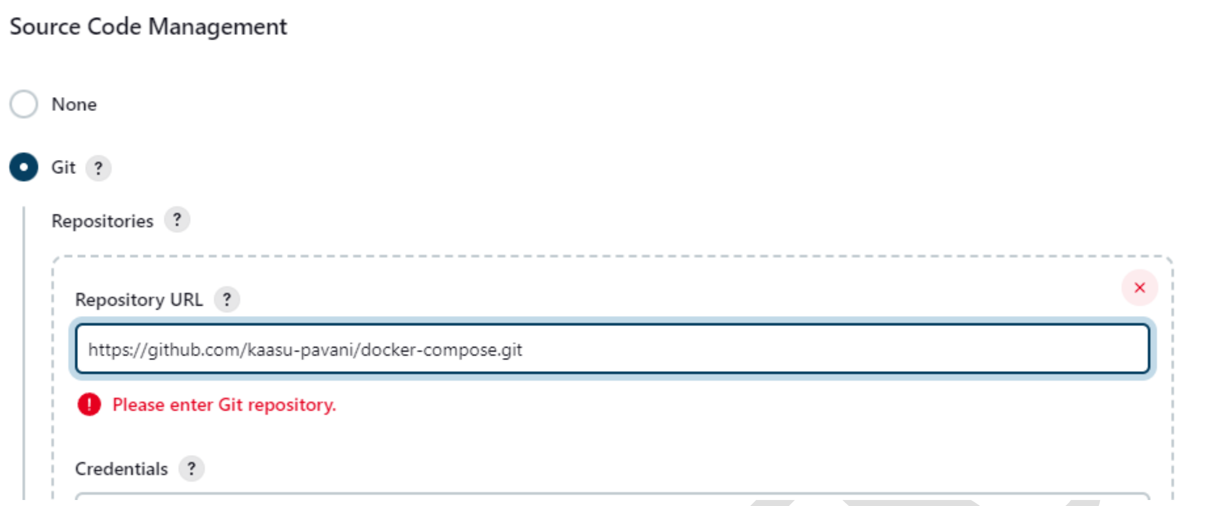
» Required field

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Source Code Management

None

[✓ Git]

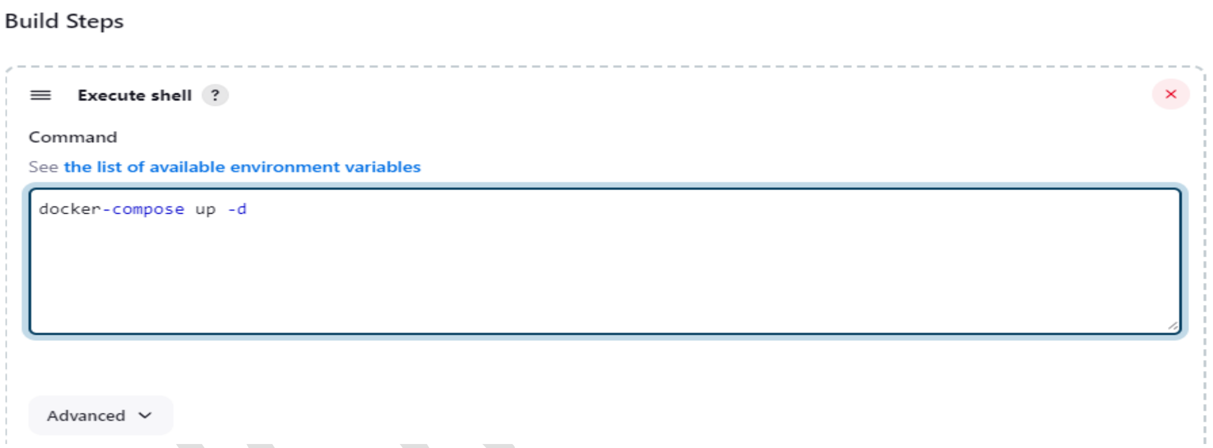
Repositories

Repository URL

https://github.com/kaasu-pavani/docker-compose.git

Please enter Git repository.

Credentials



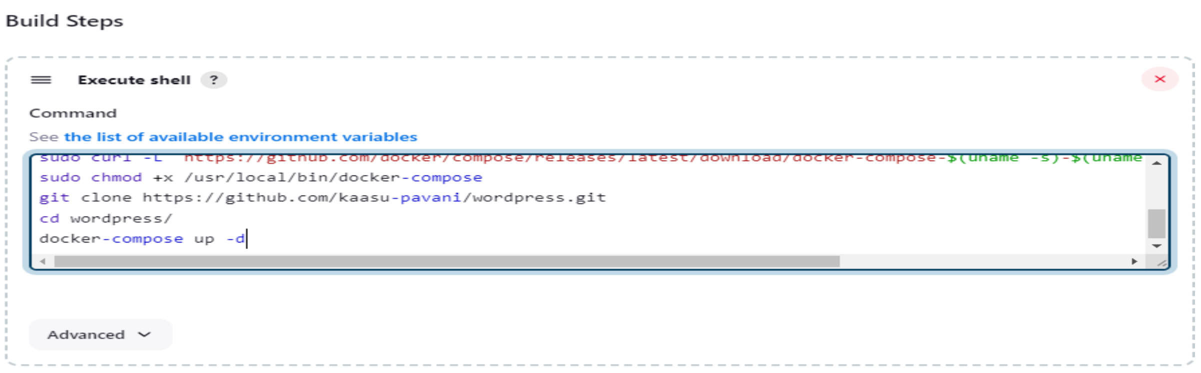
Build Steps

Execute shell

Command

See the list of available environment variables

```
docker-compose up -d
```



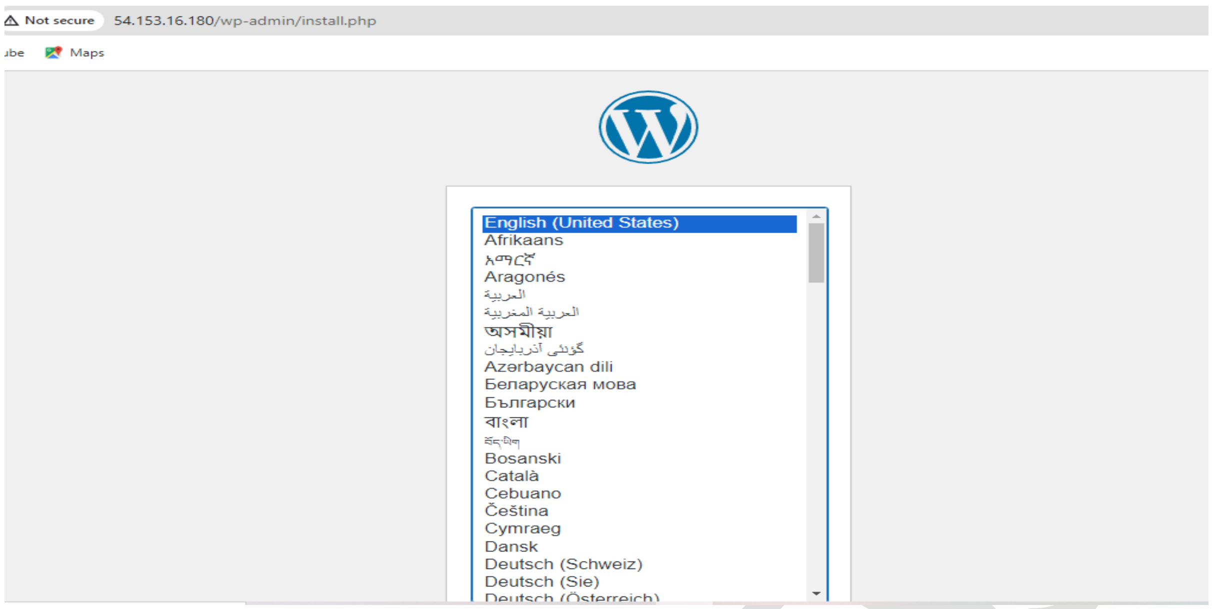
Build Steps

Execute shell

Command

See the list of available environment variables

```
sudo curl -L  https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname
sudo chmod +x /usr/local/bin/docker-compose
git clone https://github.com/kaasu-pavani/wordpress.git
cd wordpress/
docker-compose up -d
```


Not secure 54.153.16.180/wp-admin/install.php

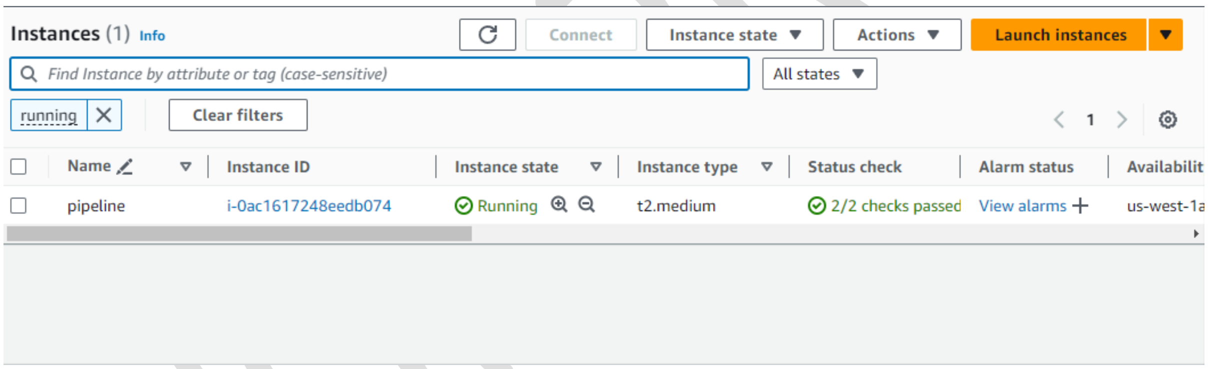
ube Maps

English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Sie)
Deutsch (Österreich)

****

6. 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?

- **Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.mediam and giving security group HTTP (80) and HTTPS and security group and port 8080.**

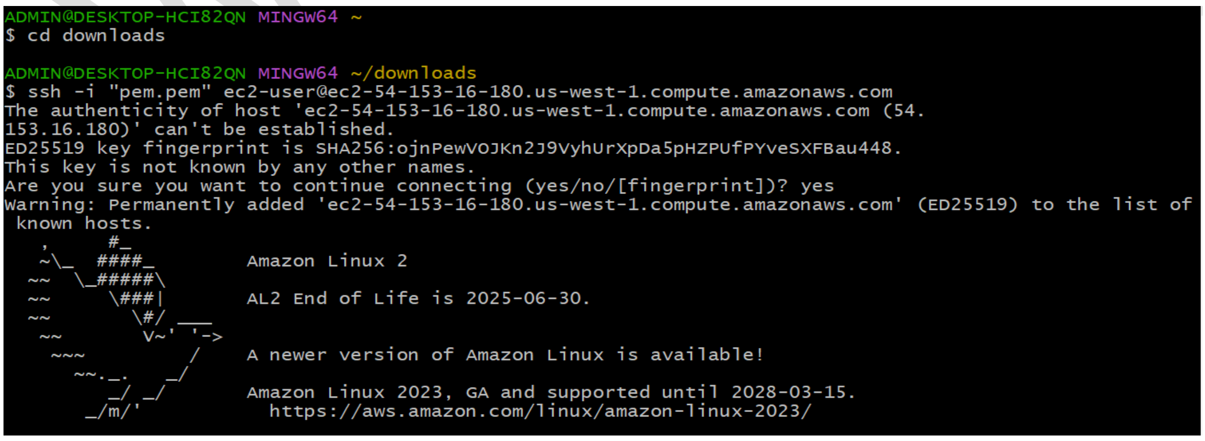


Instances (1) Info

running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabilit
pipeline	i-0ac1617248eedb074	Running	t2.medium	2/2 checks passed	View alarms	us-west-1a

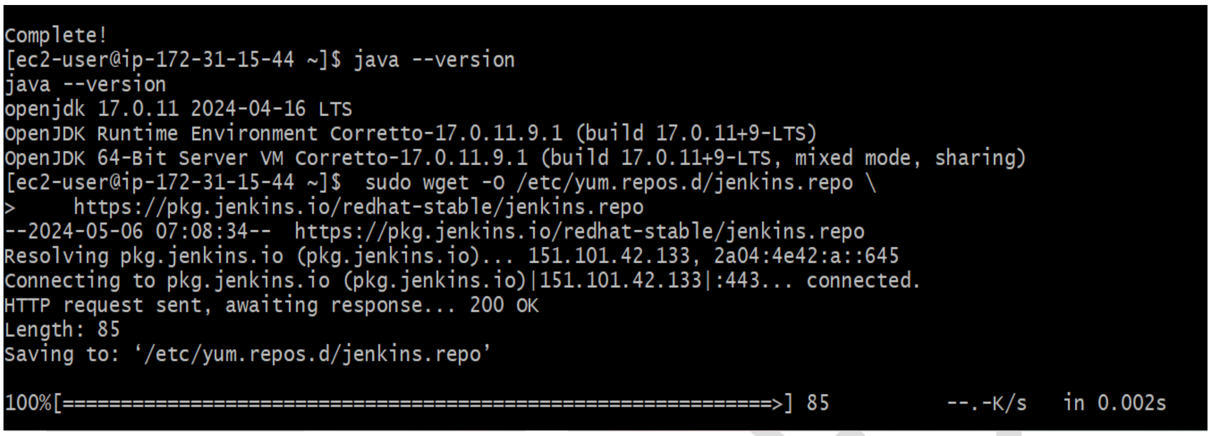
- **Go to Git Bash terminal.**



```
ADMIN@DESKTOP-HCI82QN MINGW64 ~
$ cd downloads

ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "pem.pem" ec2-user@ec2-54-153-16-180.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-54-153-16-180.us-west-1.compute.amazonaws.com (54.
153.16.180)' can't be established.
ED25519 key fingerprint is SHA256:ojnPewVOJKn2J9VyhUrXpDa5pHZPUfPYveSXFBau448.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-153-16-180.us-west-1.compute.amazonaws.com' (ED25519) to the list of
 known hosts.
         #_
   ,     ####_        Amazon Linux 2
   ~\_  #####\
  ~~  \_#####\        AL2 End of Life is 2025-06-30.
  ~~     \###|
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /     A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/        Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'            https://aws.amazon.com/linux/amazon-linux-2023/
```

➢ **Now run these following command as install java software**

< sudo yum –y install java* >



```
Complete!
[ec2-user@ip-172-31-15-44 ~]$ java --version
java --version
openjdk 17.0.11 2024-04-16 LTS
OpenJDK Runtime Environment Corretto-17.0.11.9.1 (build 17.0.11+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.11.9.1 (build 17.0.11+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-15-44 ~]$  sudo wget -O /etc/yum.repos.d/jenkins.repo \
>     https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-05-06 07:08:34--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.42.133, 2a04:4e42:a::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.42.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

100%[=============================================================>] 85          --.-K/s    in 0.002s
```

➢ **Now installing the Jenkins following below commands**

< sudo wget -O /etc/yum.repos.d/jenkins.repo \ https://pkg.jenkins.io/redhat-stable/jenkins.repo >

< sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key >

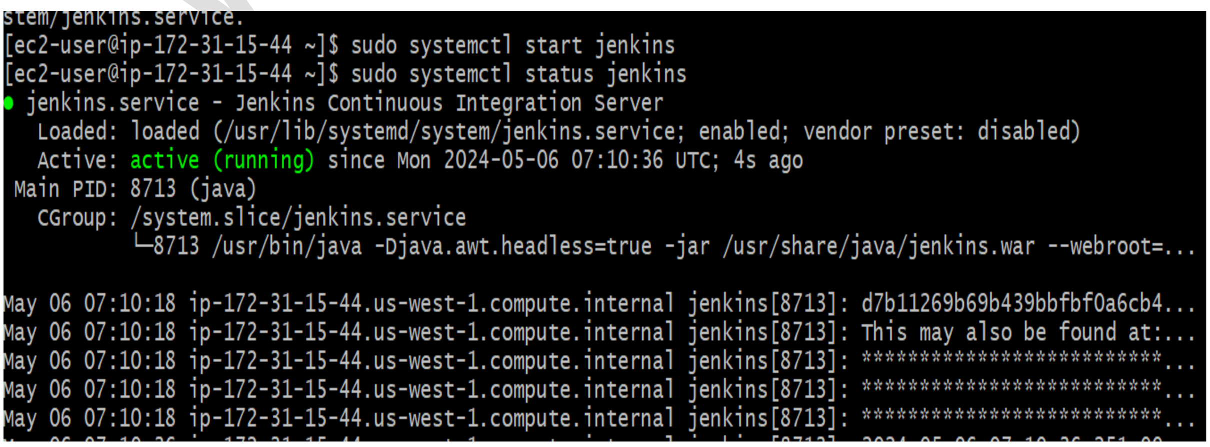
< sudo yum upgrade >

< sudo yum install jenkins -y >

< sudo systemctl start Jenkins >

< sudo systemctl enable Jenkins >

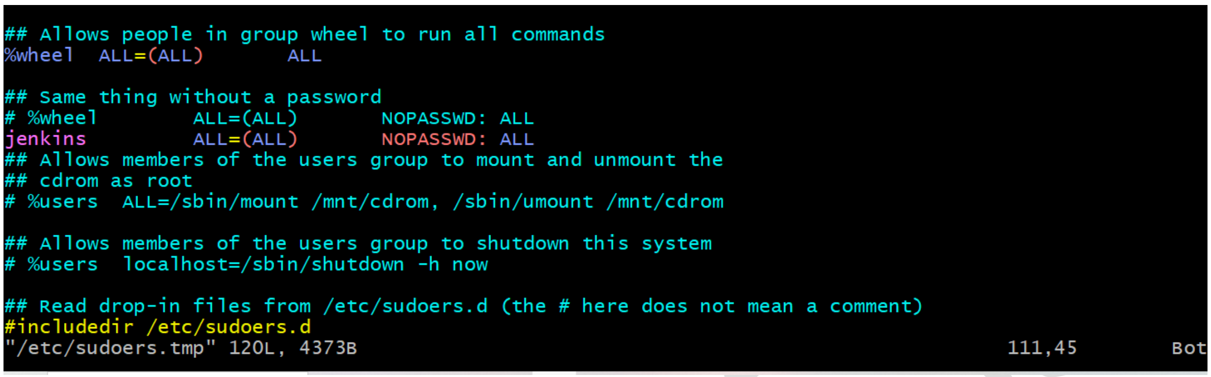
< sudo systemctl status Jenkins >



```
stem/jenkins.service.
[ec2-user@ip-172-31-15-44 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-15-44 ~]$ 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 2024-05-06 07:10:36 UTC; 4s ago
 Main PID: 8713 (java)
   CGroup: /system.slice/jenkins.service
           └─8713 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=...

May 06 07:10:18 ip-172-31-15-44.us-west-1.compute.internal jenkins[8713]: d7b11269b69b439bbfbf0a6cb4...
May 06 07:10:18 ip-172-31-15-44.us-west-1.compute.internal jenkins[8713]: This may also be found at:...
May 06 07:10:18 ip-172-31-15-44.us-west-1.compute.internal jenkins[8713]: *************************...
May 06 07:10:18 ip-172-31-15-44.us-west-1.compute.internal jenkins[8713]: *************************...
May 06 07:10:18 ip-172-31-15-44.us-west-1.compute.internal jenkins[8713]: *************************...
```

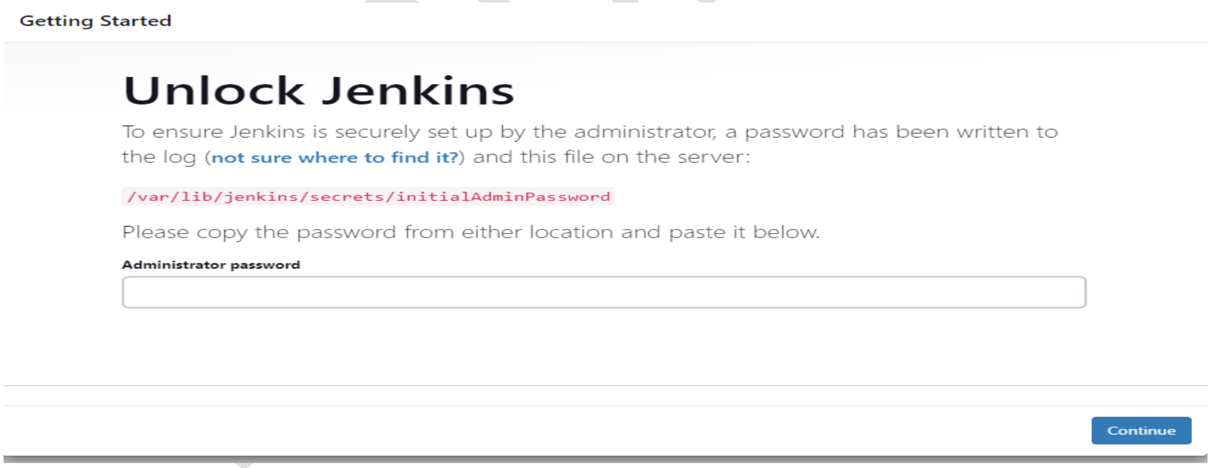
- **Now give the permissions as**

  < sudo visudo >



```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
jenkins         ALL=(ALL)       NOPASSWD: ALL
## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
"/etc/sudoers.tmp" 120L, 4373B                                    111,45        Bot
```

- **Now go to EC2 instance, copy the public ip and host to Google.**
- **Now open the Jenkins windows.**



Getting Started

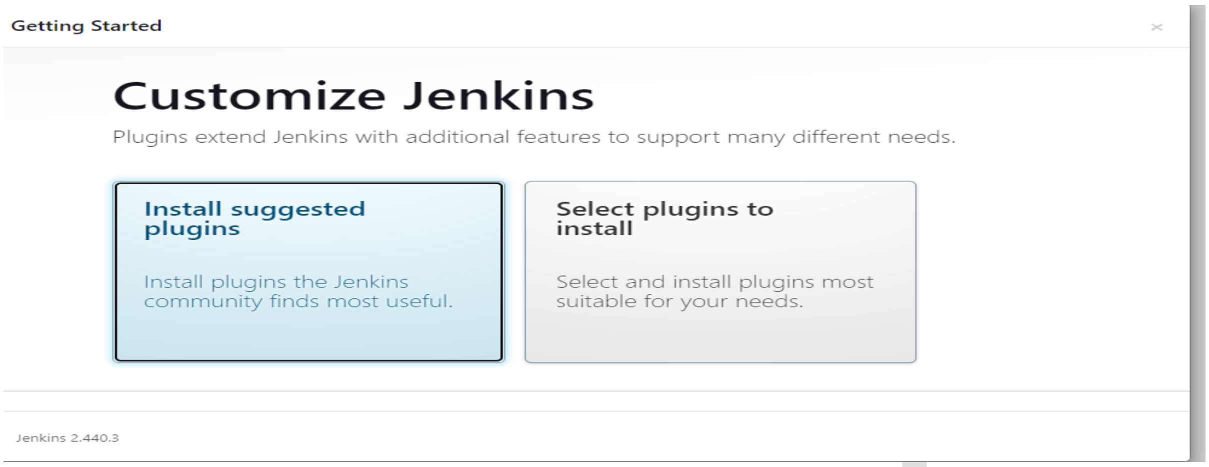
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

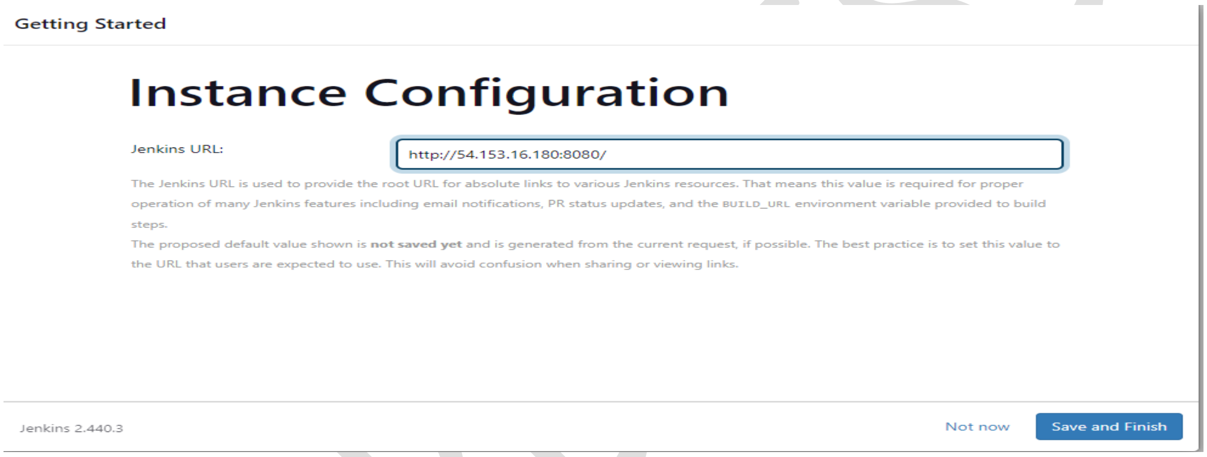
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.440.3


Getting Started

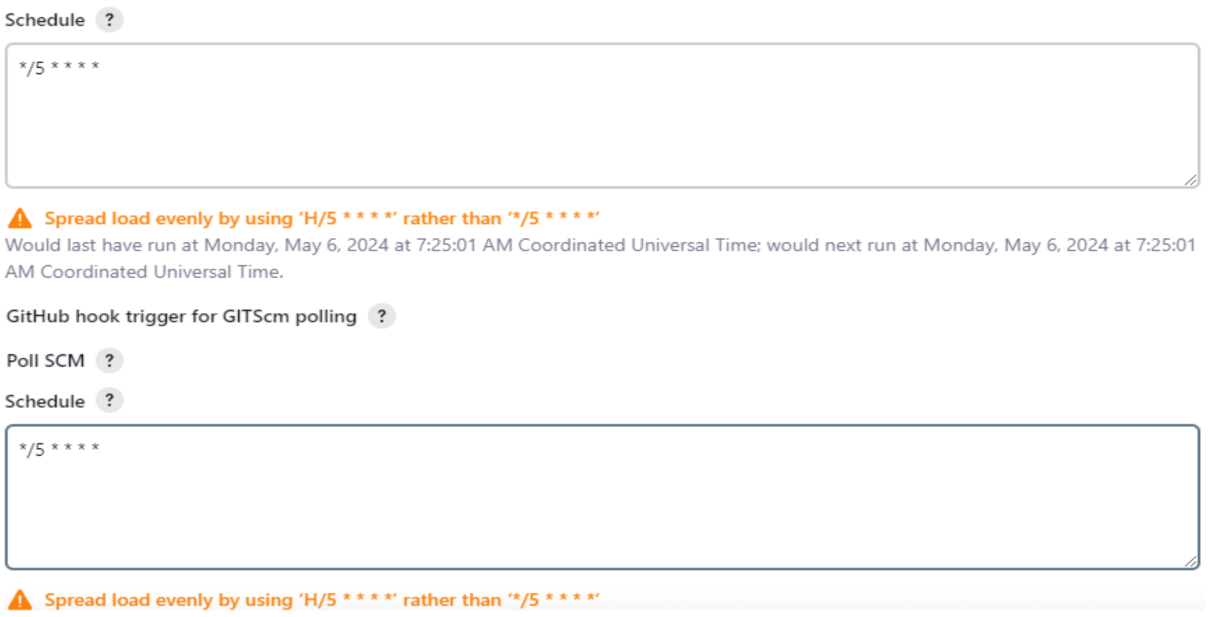
Instance Configuration

Jenkins URL: http://54.153.16.180:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.440.3


Schedule

```
*/5 * * * *
```

Spread load evenly by using 'H/5 * * * *' rather than '*/5 * * * *'

Would last have run at Monday, May 6, 2024 at 7:25:01 AM Coordinated Universal Time; would next run at Monday, May 6, 2024 at 7:25:01 AM Coordinated Universal Time.

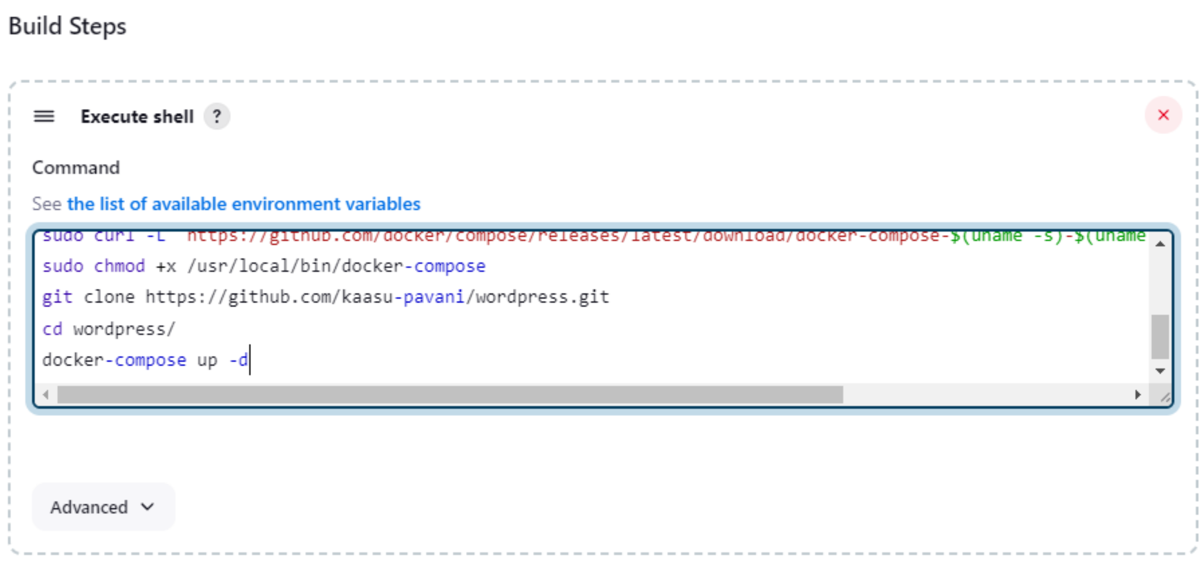
GitHub hook trigger for GITScm polling

Poll SCM

Schedule

```
*/5 * * * *
```

Spread load evenly by using 'H/5 * * * *' rather than '*/5 * * * *'



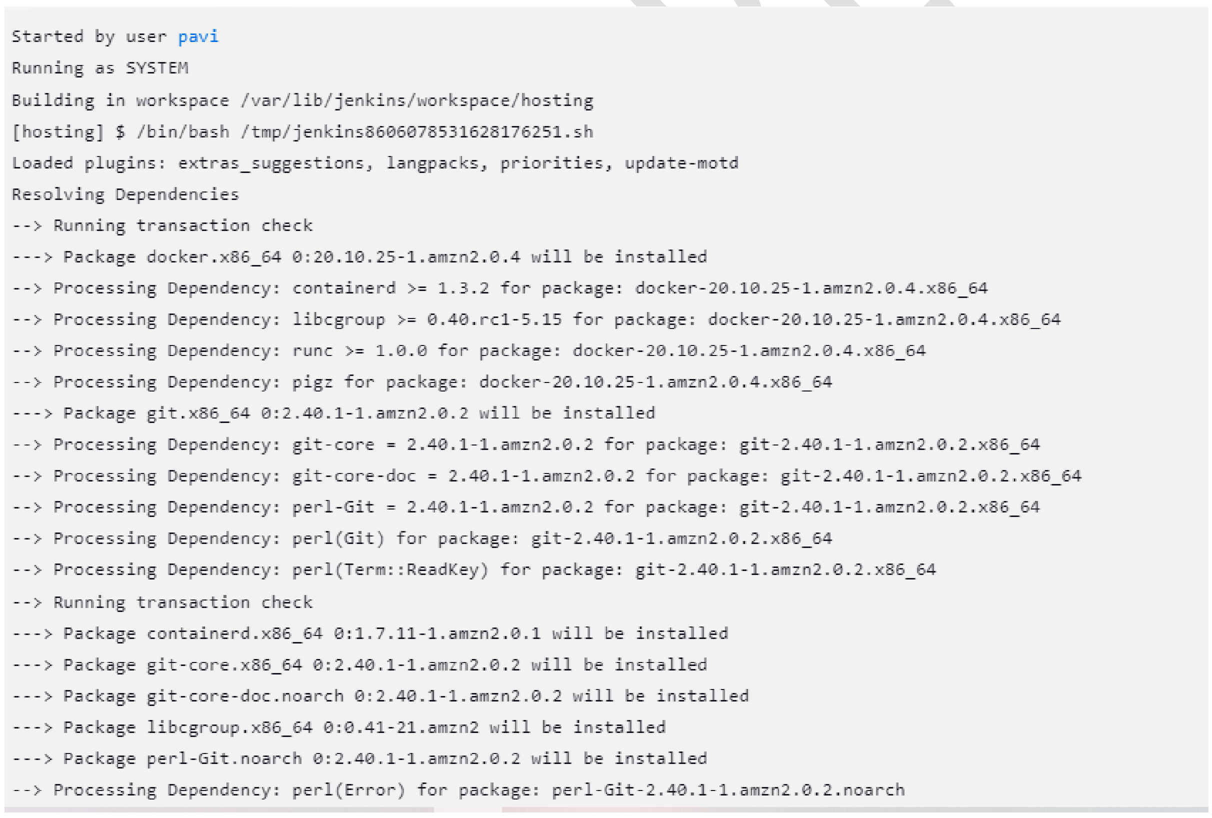
Build Steps

Execute shell

Command

See the list of available environment variables

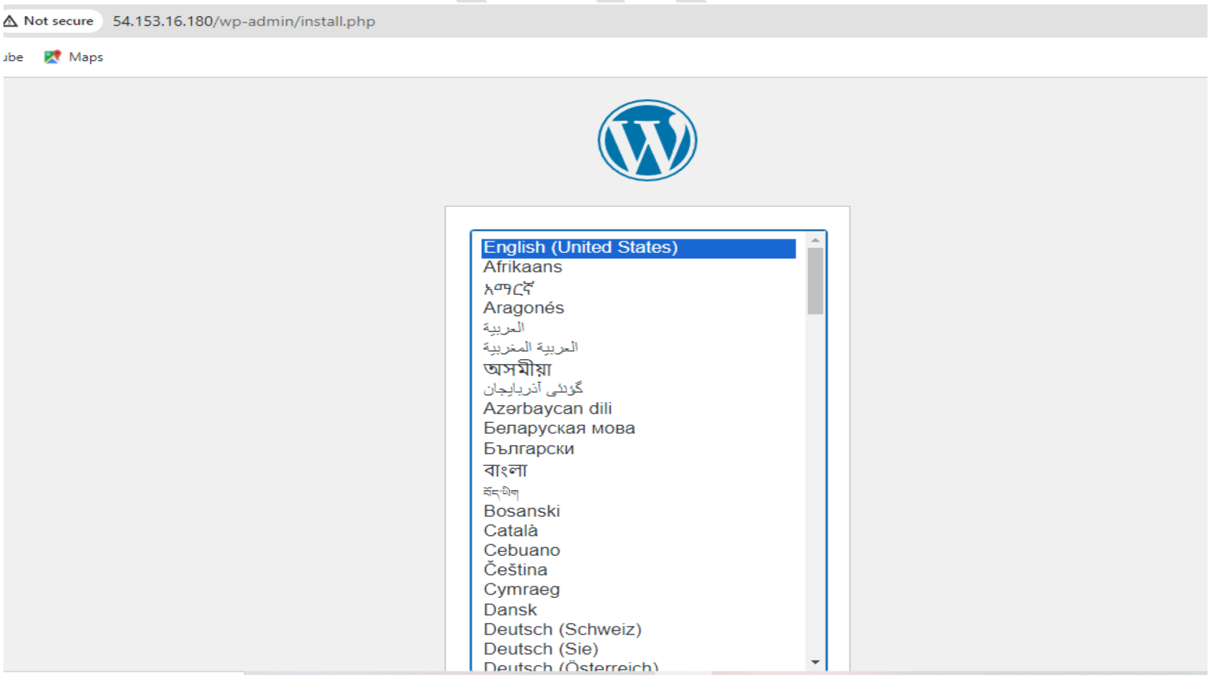
```
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname
sudo chmod +x /usr/local/bin/docker-compose
git clone https://github.com/kaasu-pavani/wordpress.git
cd wordpress/
docker-compose up -d
```



```
Started by user pavi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/hosting
[hosting] $ /bin/bash /tmp/jenkins8606078531628176251.sh
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package docker.x86_64 0:20.10.25-1.amzn2.0.4 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.4.x86_64
---> Package git.x86_64 0:2.40.1-1.amzn2.0.2 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.2 for package: git-2.40.1-1.amzn2.0.2.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.2 for package: git-2.40.1-1.amzn2.0.2.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.2 for package: git-2.40.1-1.amzn2.0.2.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.2.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.2.x86_64
--> Running transaction check
---> Package containerd.x86_64 0:1.7.11-1.amzn2.0.1 will be installed
---> Package git-core.x86_64 0:2.40.1-1.amzn2.0.2 will be installed
---> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.2 will be installed
---> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
---> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.2 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.2.noarch
```

```
 59fe2989bce1 Extracting [==================================================>]  1.727kB/1.727kB
 59fe2989bce1 Extracting [==================================================>]  1.727kB/1.727kB
 59fe2989bce1 Pull complete
 wordpress Pulled
 Network wordpress_default  Creating
 Network wordpress_default  Created
 Volume "wordpress_wordpress"  Creating
 Volume "wordpress_wordpress"  Created
 Volume "wordpress_db"  Creating
 Volume "wordpress_db"  Created
 Container wordpress-db-1  Creating
 Container wordpress-wordpress-1  Creating
 Container wordpress-db-1  Created
 Container wordpress-wordpress-1  Created
 Container wordpress-wordpress-1  Starting
 Container wordpress-db-1  Starting
 Container wordpress-db-1  Started
 Container wordpress-wordpress-1  Started
Finished: SUCCESS
```

- **Now go to EC2 instance, copy the public ip and host to Google.**
- **Now open the wordpress window.**

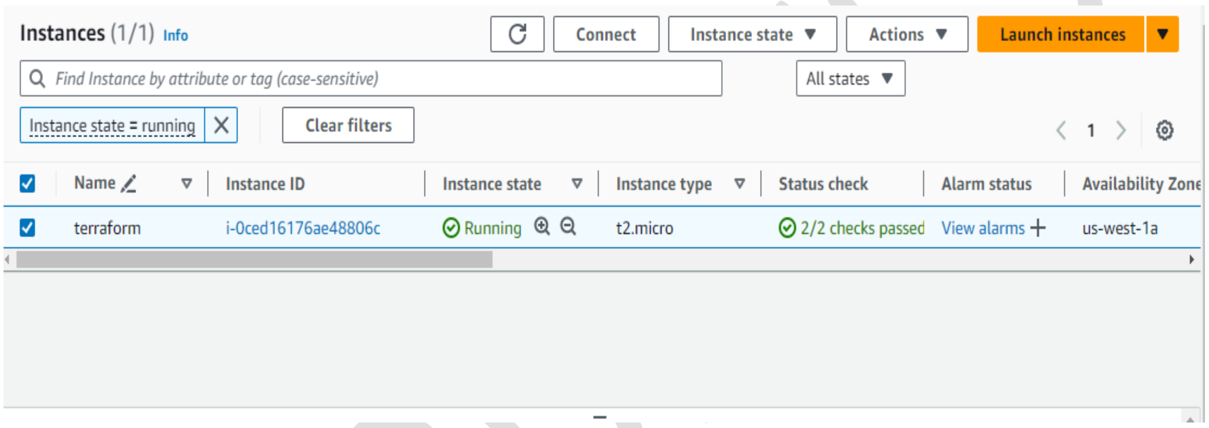


English (United States)
Afrikaans
አማርኛ
Aragonés
العربية
العربية المغربية
অসমীয়া
گؤنئی آذربایجان
Azərbaycan dili
Беларуская мова
Български
বাংলা
བོད་ཡིག
Bosanski
Català
Cebuano
Čeština
Cymraeg
Dansk
Deutsch (Schweiz)
Deutsch (Sie)
Deutsch (Österreich)

****

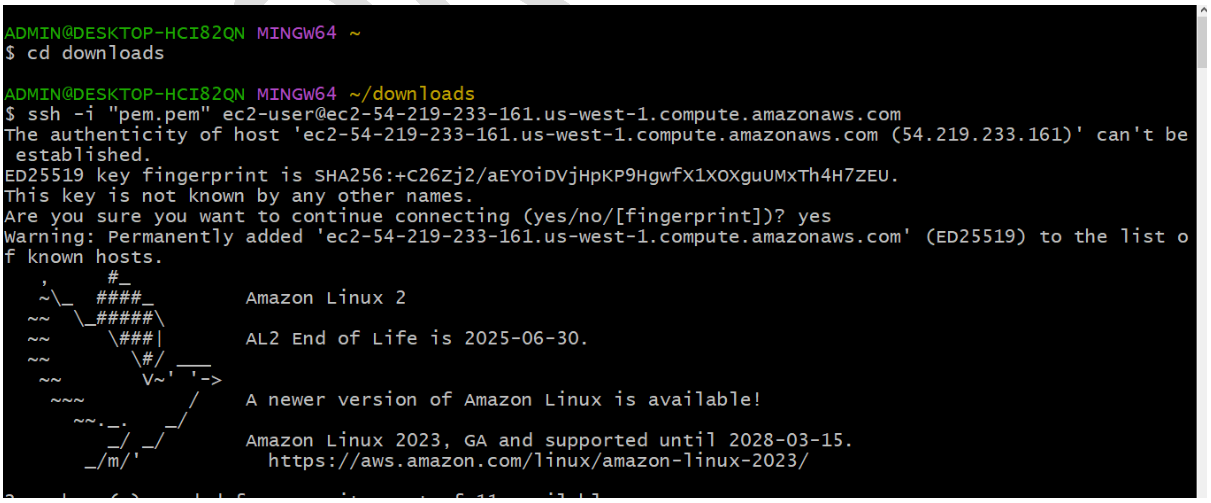
7. Deploy WordPress web application by using terraform (create Ec2 instance along with userdata .sh file)?

- Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t2.micro and giving security group HTTP (80) and HTTPS and security group and port 8080.



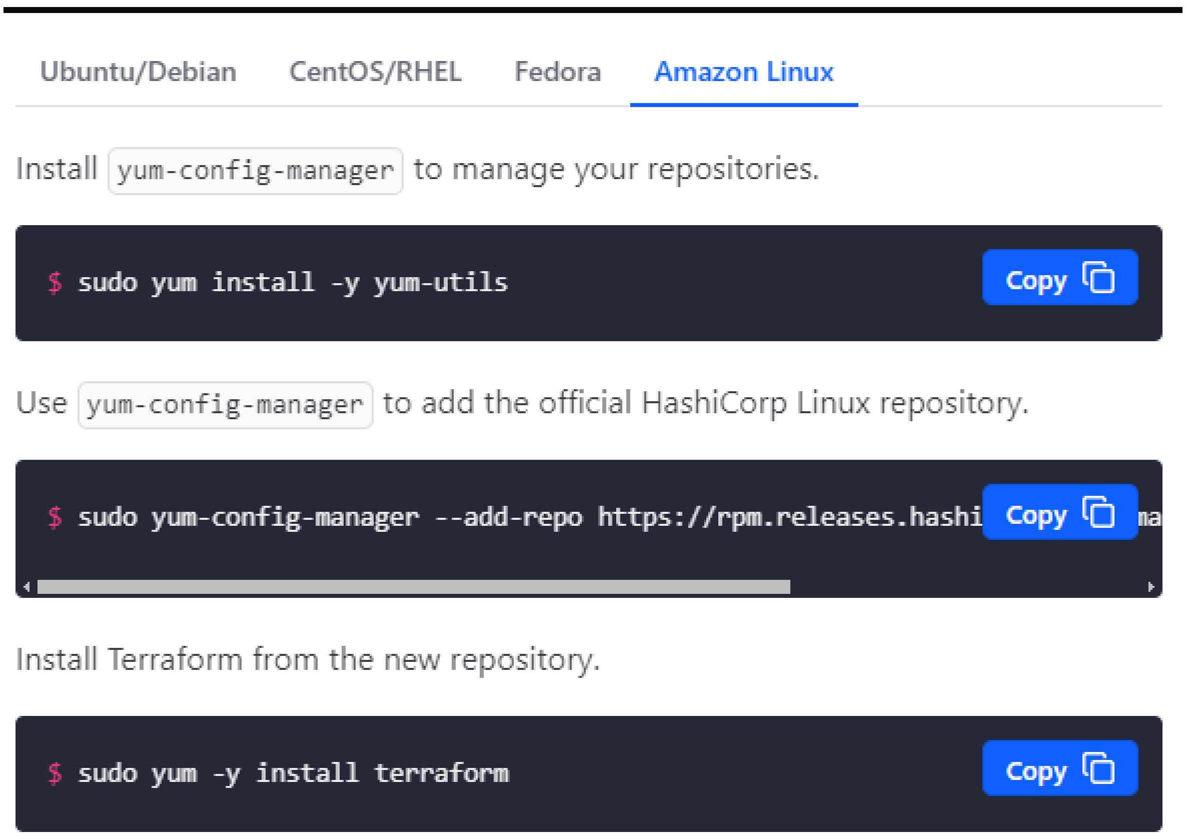
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
terraform	i-0ced16176ae48806c	Running	t2.micro	2/2 checks passed	View alarms	us-west-1a

- Go to Git Bash terminal.



```
ADMIN@DESKTOP-HCI82QN MINGW64 ~
$ cd downloads

ADMIN@DESKTOP-HCI82QN MINGW64 ~/downloads
$ ssh -i "pem.pem" ec2-user@ec2-54-219-233-161.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-54-219-233-161.us-west-1.compute.amazonaws.com (54.219.233.161)' can't be
 established.
ED25519 key fingerprint is SHA256:+C26Zj2/aEYOiDVjHpKP9HgwfX1XOXguUMxTh4H7ZEU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-219-233-161.us-west-1.compute.amazonaws.com' (ED25519) to the list o
f known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/
```

- Now install the terraform.
- Go to Google, copy Teraform commands and paste to Git Bash terminal.



Install `yum-config-manager` to manage your repositories.

```
$ sudo yum install -y yum-utils
```

Use `yum-config-manager` to add the official HashiCorp Linux repository.

```
$ sudo yum-config-manager --add-repo https://rpm.releases.hashi
```

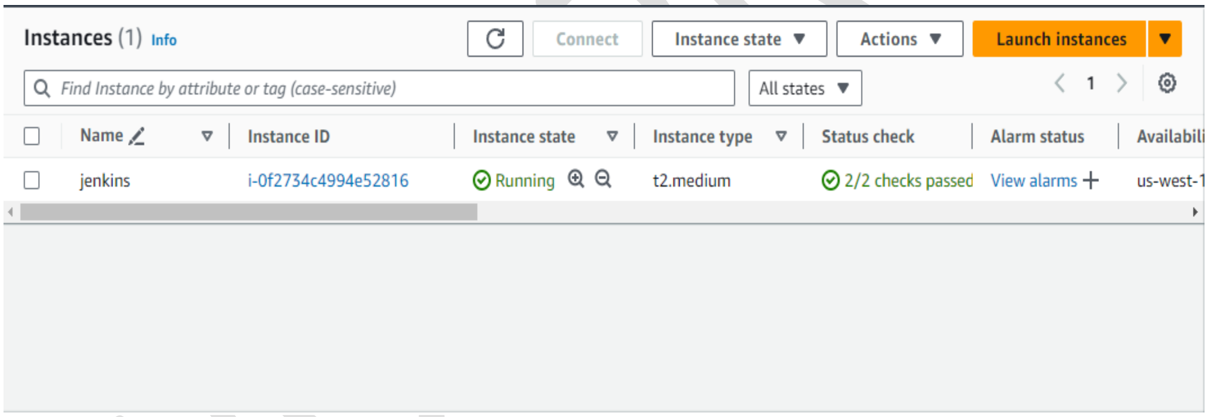
Install Terraform from the new repository.

```
$ sudo yum -y install terraform
```

- **Now create the file following these command as**
  < vi main.tf >
- **Now create the another file following these command as**
  < vi user.sh >
- **Now run the terraform execution following these commands as**
  - **Terraform init**
  - **Terraform fmt**
  - **Terraform validate**
  - **Terraform plan**
  - **Terraform apply**

8. 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.

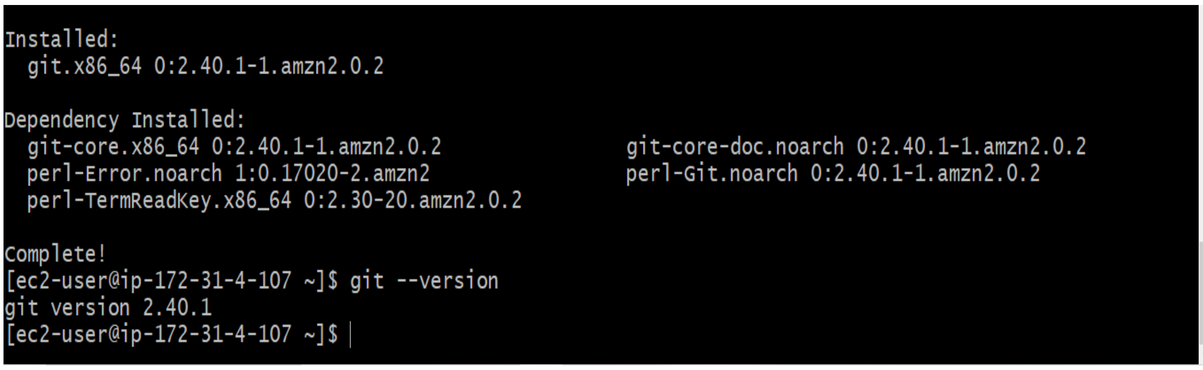
- **Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t3.mediam and giving security group HTTP (80) and HTTPS and security group and port 8080.**



	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili
☐	jenkins	i-0f2734c4994e52816	Running	t2.medium	2/2 checks passed	View alarms +	us-west-1

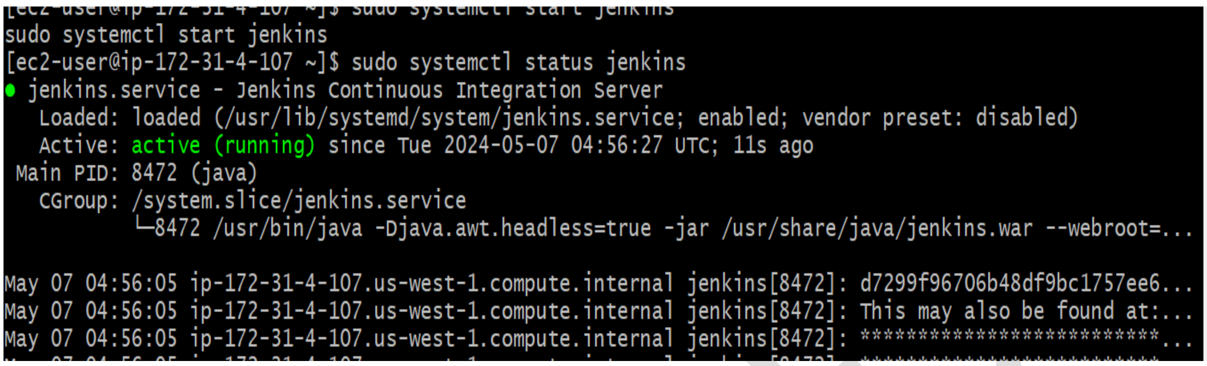
- **Go to Gitbash.**
- **Now install Git using these command as**

  < sudo yum –y install git >



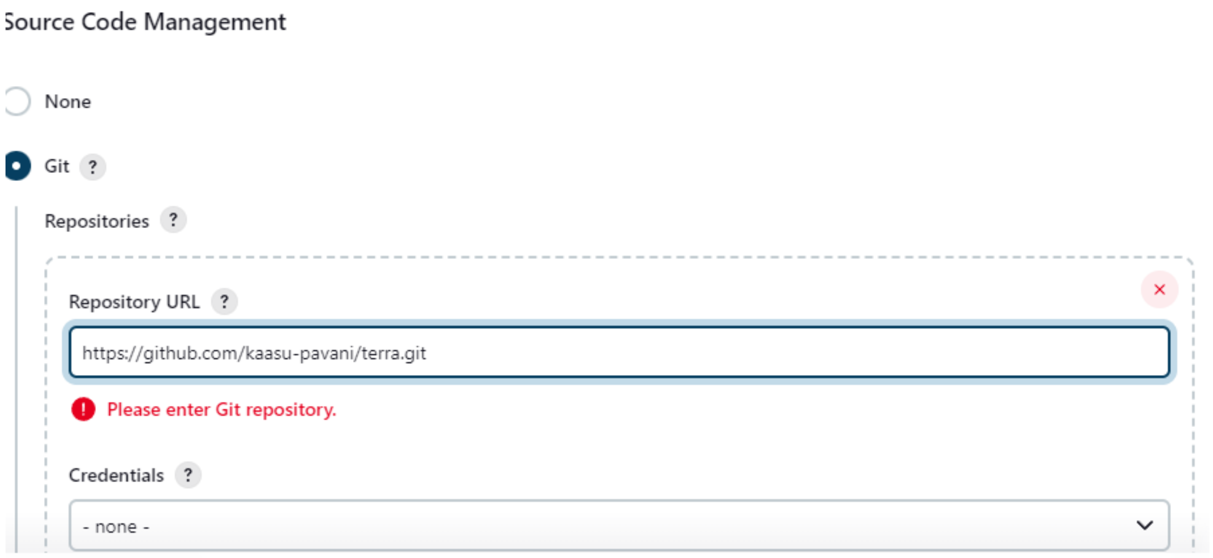
```
Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.2

Dependency Installed:
  git-core.x86_64 0:2.40.1-1.amzn2.0.2                git-core-doc.noarch 0:2.40.1-1.amzn2.0.2
  perl-Error.noarch 1:0.17020-2.amzn2                 perl-Git.noarch 0:2.40.1-1.amzn2.0.2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-4-107 ~]$ git --version
git version 2.40.1
[ec2-user@ip-172-31-4-107 ~]$ 
```

- **Now install the java using these following command as**
  < sudo yum –y install java* >
- **Now install the Jenkins (follow the above steps Jenkins installation).**



```
[ec2-user@ip-172-31-4-107 ~]$ sudo systemctl start jenkins
sudo systemctl start jenkins
[ec2-user@ip-172-31-4-107 ~]$ 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 Tue 2024-05-07 04:56:27 UTC; 11s ago
 Main PID: 8472 (java)
   CGroup: /system.slice/jenkins.service
           └─8472 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=...

May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: d7299f96706b48df9bc1757ee6...
May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: This may also be found at:...
May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: *************************...
```

- **Go to Jenkins window.**
- **Username, password, full name and Email entered after that start using Jenkins.**
- **Create job.**
- **Go to source code management, select Git.**
- **Go to Git repo URL (copy Git URL from Git hub). [Git hub repo (main.tf and user.sh) these data].**



Source Code Management

None

[✓ Git] ?

Repositories ?

Repository URL ?

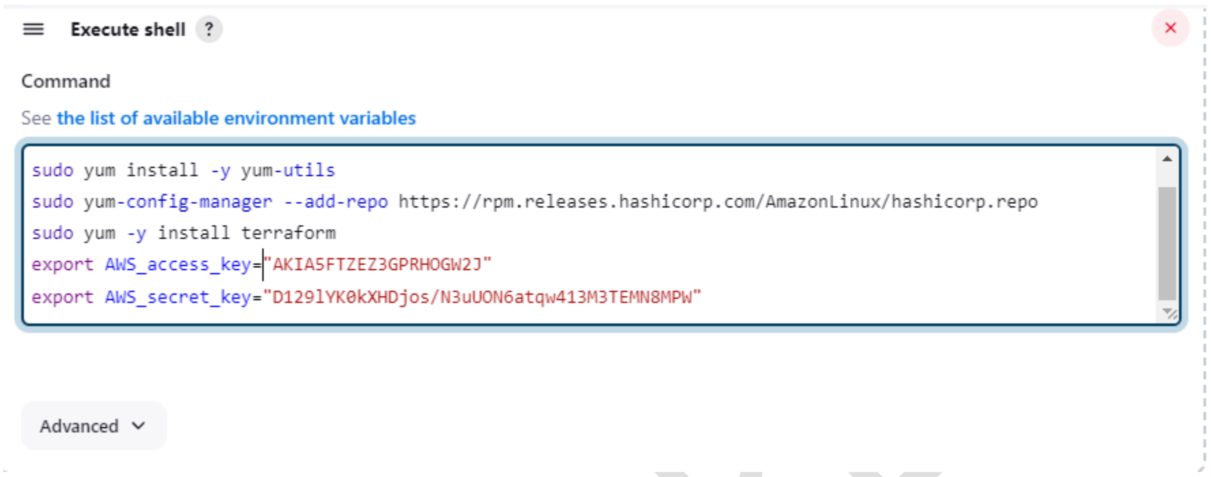
https://github.com/kaasu-pavani/terra.git

Please enter Git repository.

Credentials ?

- none -

➢ **Go to Execute shell enter the Terraform installation commands and IAM user (access and secret key).**



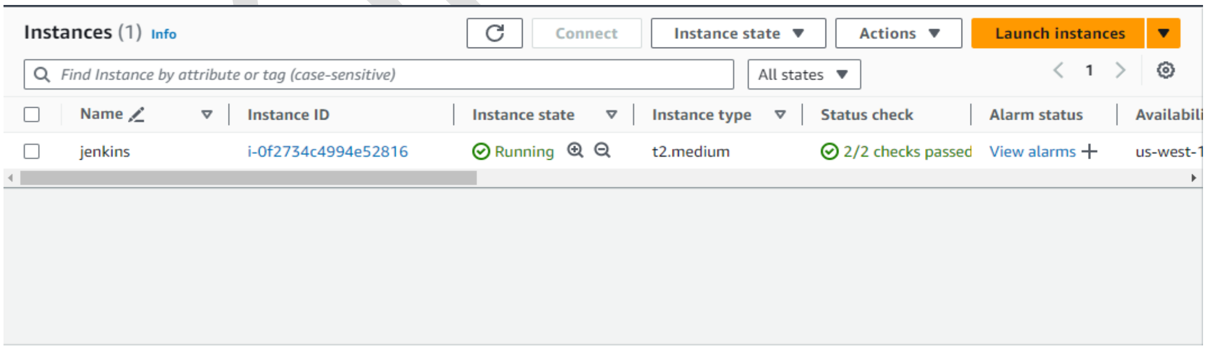
```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum -y install terraform
export AWS_access_key="AKIA5FTZEZ3GPRHOGW2J"
export AWS_secret_key="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"
```

➢ **After that click apply & save and Build now.**

****

9. 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.

- Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t3.mediam and giving security group HTTP (80) and HTTPS and security group and port 8080.



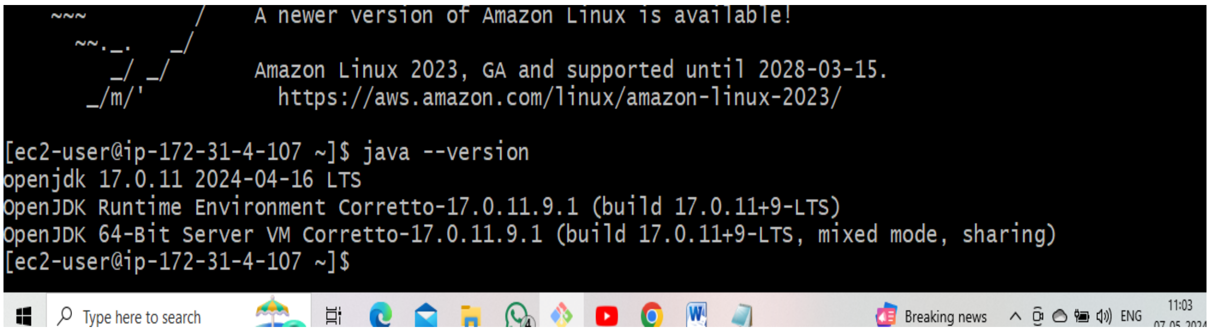
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabili
☐	jenkins	i-0f2734c4994e52816	Running	t2.medium	2/2 checks passed	View alarms	us-west-1

- Go to Git Bash terminal.
- Now install the Git command as below

< sudo yum –y install git >

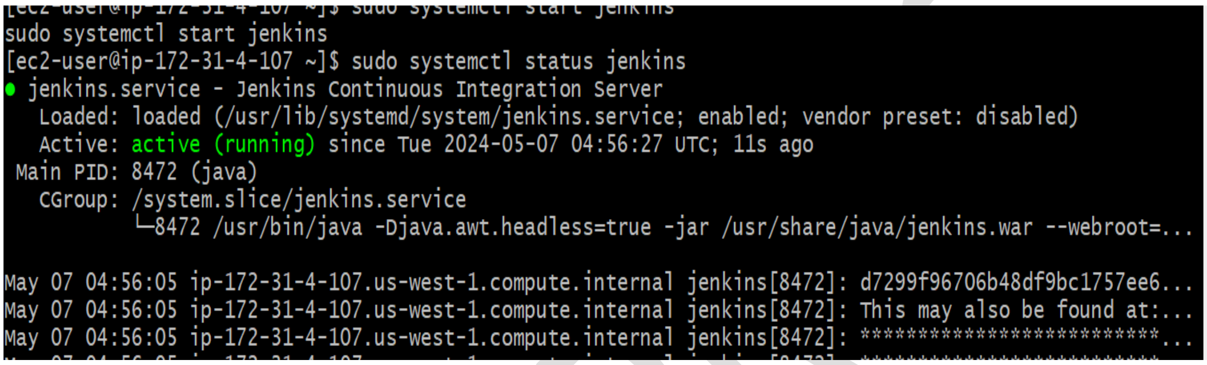
- Now install the java command as

< sudo yum –y install java* >



```
  ~~~          /     A newer version of Amazon Linux is available!
    ~~._.   _/
       _/ _/         Amazon Linux 2023, GA and supported until 2028-03-15.
     _/m/'             https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-4-107 ~]$ java --version
openjdk 17.0.11 2024-04-16 LTS
OpenJDK Runtime Environment Corretto-17.0.11.9.1 (build 17.0.11+9-LTS)
OpenJDK 64-Bit Server VM Corretto-17.0.11.9.1 (build 17.0.11+9-LTS, mixed mode, sharing)
[ec2-user@ip-172-31-4-107 ~]$
```

➢ **Now install the Jenkins follow the above commands.**



```
sudo systemctl start jenkins
[ec2-user@ip-172-31-4-107 ~]$ 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 Tue 2024-05-07 04:56:27 UTC; 11s ago
 Main PID: 8472 (java)
   CGroup: /system.slice/jenkins.service
           └─8472 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=...

May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: d7299f96706b48df9bc1757ee6...
May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: This may also be found at:...
May 07 04:56:05 ip-172-31-4-107.us-west-1.compute.internal jenkins[8472]: *************************...
```

- ➢ **Now create the job.**
- ➢ **After that select SCM (Git).**
- ➢ **Copy Git URL from Git hub repo and paste to Git URL.**
- ➢ **Go to execute shell enter the (Terraform installation commands).**
- ➢ **Click the apply & save and Build now.**

****

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

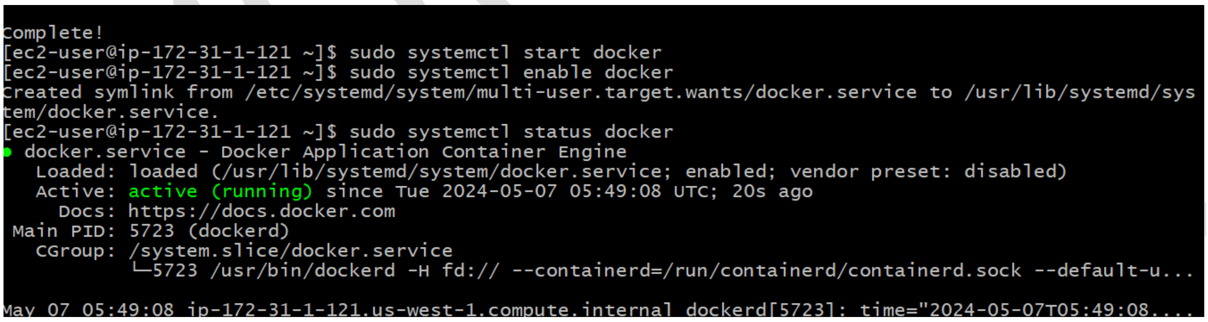
- Create the EC2 instance by selecting EC2 services and launch the instance by selecting Amazon Linux-2 version and t3.mediam and giving security group HTTP (80) and security group and port 8080.
- Go to Git Bash terminal.
- Now install the docker command as

  < sudo yum –y install docker >
- Now docker (start, enable, status) commands as

  < sudo systemctl start docker >

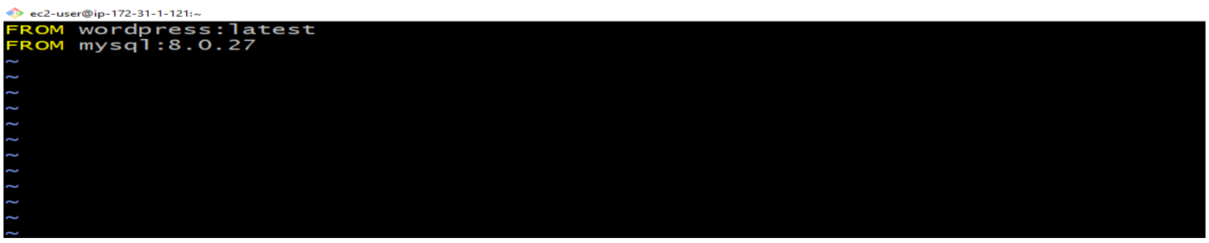
  < sudo systemctl enable docker >

  < sudo systemctl start docker >
- Now install the docker-compose file (command as Google).



```
Complete!
[ec2-user@ip-172-31-1-121 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-1-121 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/sys
tem/docker.service.
[ec2-user@ip-172-31-1-121 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-05-07 05:49:08 UTC; 20s ago
     Docs: https://docs.docker.com
 Main PID: 5723 (dockerd)
   CGroup: /system.slice/docker.service
           └─5723 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-u...

May 07 05:49:08 ip-172-31-1-121.us-west-1.compute.internal dockerd[5723]: time="2024-05-07T05:49:08....
```

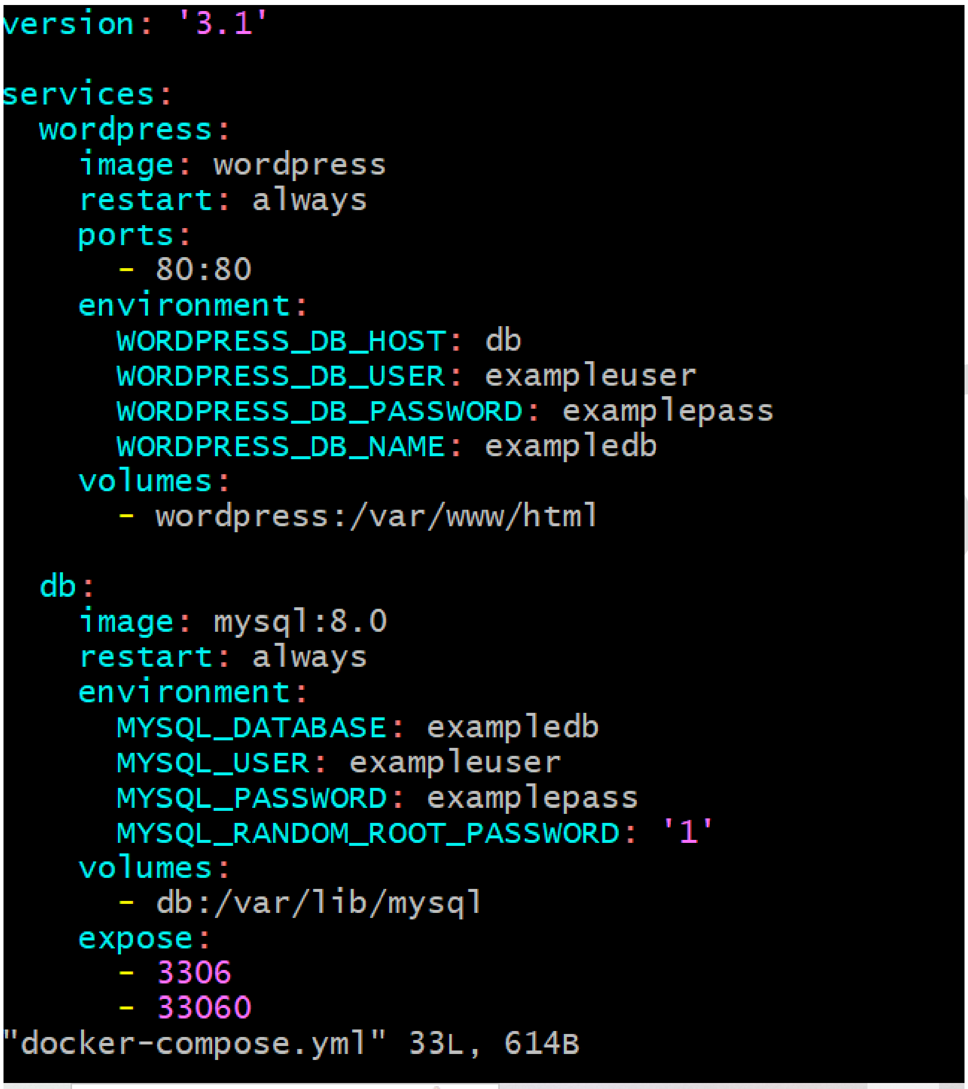
- Now create file docker file.



```
ec2-user@ip-172-31-1-121:~
FROM wordpress:latest
FROM mysql:8.0.27
~
~
~
~
~
~
~
~
~
~
~
```

- Now create the docker-compose file.

  < sudo vi docker-compose.yml >



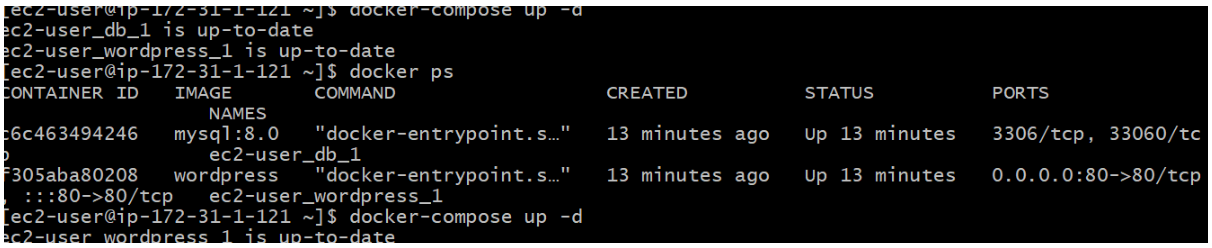
```
version: '3.1'

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 80:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:8.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql
    expose:
      - 3306
      - 33060
"docker-compose.yml" 33L, 614B
```

- Now execute the docker-compose file command as

  < docker-compose up –d >

- Now docker ps command as

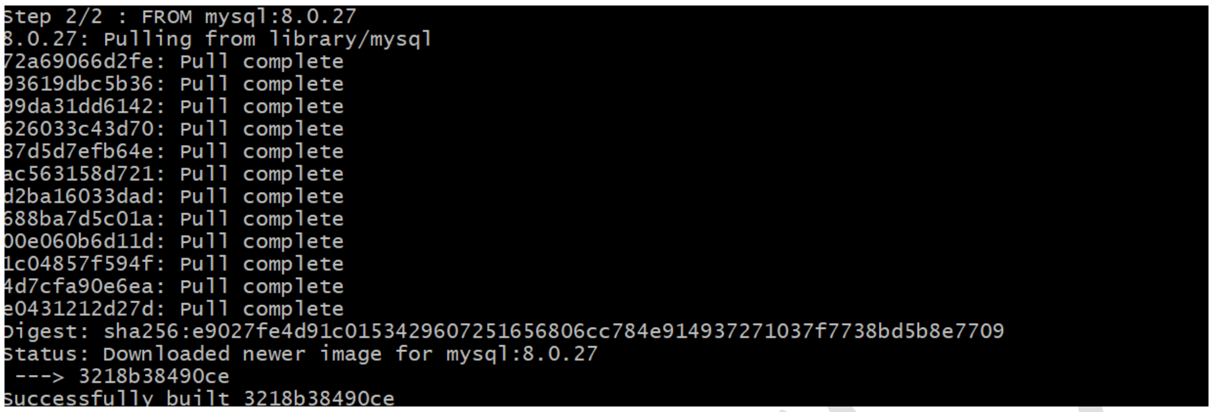
  < docker ps >



```
[ec2-user@ip-172-31-1-121 ~]$ docker-compose up -d
ec2-user_db_1 is up-to-date
ec2-user_wordpress_1 is up-to-date
[ec2-user@ip-172-31-1-121 ~]$ docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED          STATUS          PORTS
                NAMES
c6c463494246   mysql:8.0   "docker-entrypoint.s…"   13 minutes ago   Up 13 minutes   3306/tcp, 33060/tc
p               ec2-user_db_1
f305aba80208   wordpress   "docker-entrypoint.s…"   13 minutes ago   Up 13 minutes   0.0.0.0:80->80/tcp
, :::80->80/tcp   ec2-user_wordpress_1
[ec2-user@ip-172-31-1-121 ~]$ docker-compose up -d
ec2-user_wordpress_1 is up-to-date
```

- **Create new file using these command as**

  < docker build –t new . >



```
Step 2/2 : FROM mysql:8.0.27
8.0.27: Pulling from library/mysql
72a69066d2fe: Pull complete
93619dbc5b36: Pull complete
99da31dd6142: Pull complete
626033c43d70: Pull complete
37d5d7efb64e: Pull complete
ac563158d721: Pull complete
d2ba16033dad: Pull complete
688ba7d5c01a: Pull complete
00e060b6d11d: Pull complete
1c04857f594f: Pull complete
4d7cfa90e6ea: Pull complete
e0431212d27d: Pull complete
Digest: sha256:e9027fe4d91c0153429607251656806cc784e914937271037f7738bd5b8e7709
Status: Downloaded newer image for mysql:8.0.27
 ---> 3218b38490ce
Successfully built 3218b38490ce
```

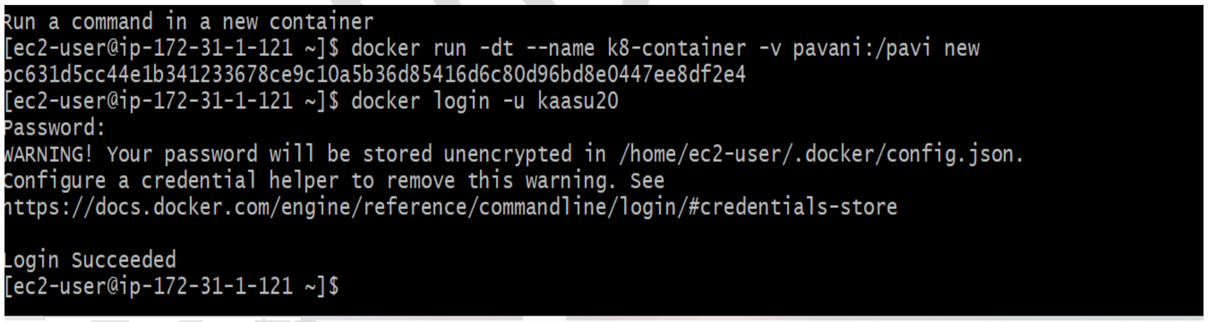
- **Now newly created images follow the command as**

  < docker run –dt –name k8-container –v pavani:/pavi new >

- **Now login the docker account using these command as**

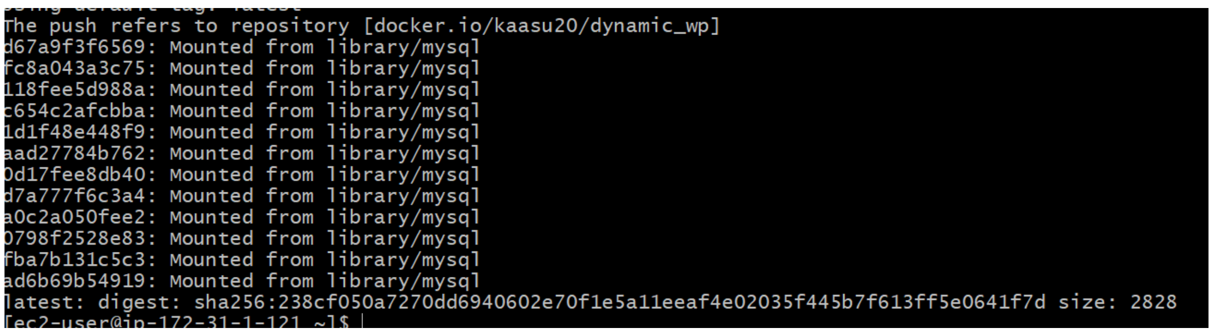
  < docker login –u kaasu20 >

- **After that entered password.**



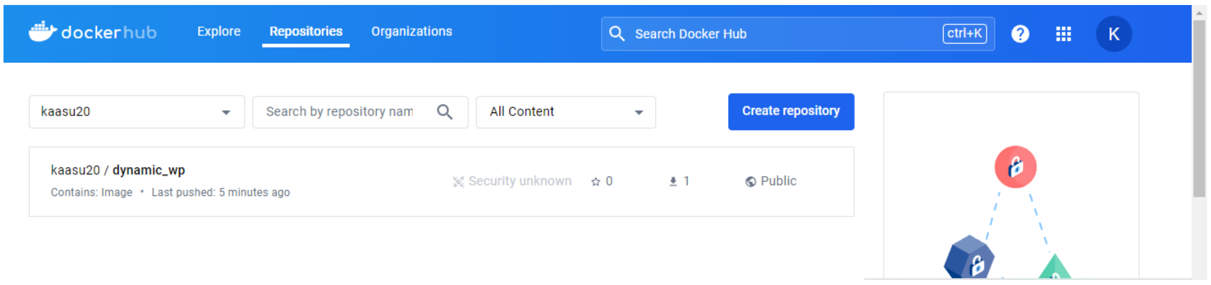
```
Run a command in a new container
[ec2-user@ip-172-31-1-121 ~]$ docker run -dt --name k8-container -v pavani:/pavi new
bc631d5cc44e1b341233678ce9c10a5b36d85416d6c80d96bd8e0447ee8df2e4
[ec2-user@ip-172-31-1-121 ~]$ docker login -u kaasu20
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-1-121 ~]$
```

- **Now push the images to docker hub.**



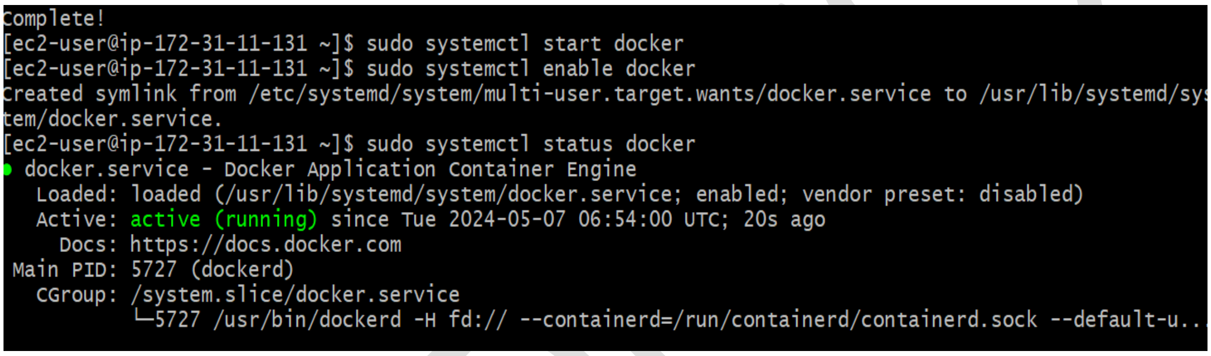
```
The push refers to repository [docker.io/kaasu20/dynamic_wp]
d67a9f3f6569: Mounted from library/mysql
fc8a043a3c75: Mounted from library/mysql
118fee5d988a: Mounted from library/mysql
c654c2afcbba: Mounted from library/mysql
1d1f48e448f9: Mounted from library/mysql
aad27784b762: Mounted from library/mysql
0d17fee8db40: Mounted from library/mysql
d7a777f6c3a4: Mounted from library/mysql
a0c2a050fee2: Mounted from library/mysql
0798f2528e83: Mounted from library/mysql
fba7b131c5c3: Mounted from library/mysql
ad6b69b54919: Mounted from library/mysql
latest: digest: sha256:238cf050a7270dd6940602e70f1e5a11eeaf4e02035f445b7f613ff5e0641f7d size: 2828
[ec2-user@ip-172-31-1-121 ~]$
```

➢ **Now see the docker hub.**



➢ **Now create the new instance.**

➢ **Install the docker.**



```
Complete!
[ec2-user@ip-172-31-11-131 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-11-131 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/sys
tem/docker.service.
[ec2-user@ip-172-31-11-131 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-05-07 06:54:00 UTC; 20s ago
     Docs: https://docs.docker.com
 Main PID: 5727 (dockerd)
   CGroup: /system.slice/docker.service
           └─5727 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-u..
```

➢ **Now give the permissions docker.**

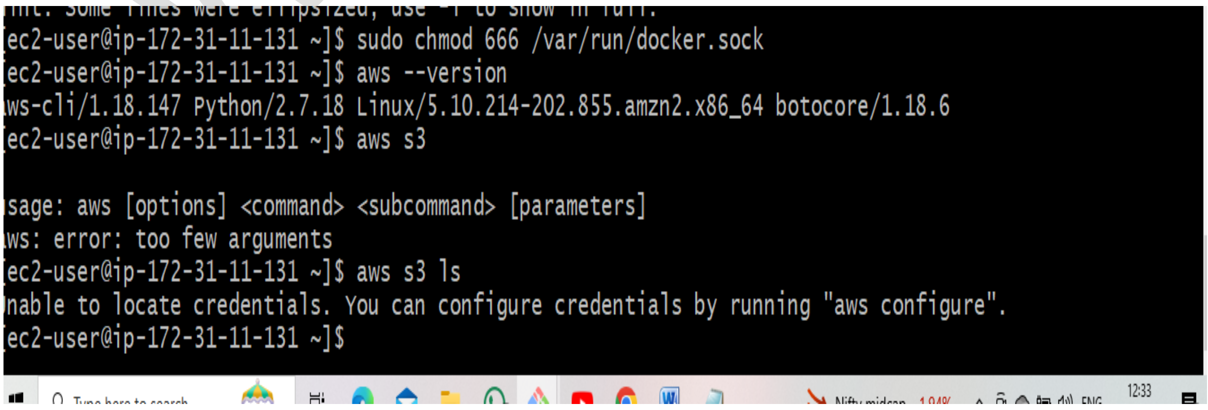
< sudo chmod 666 /var/run/docker.sock >

➢ **Now check the AWS version**

< aws –version >

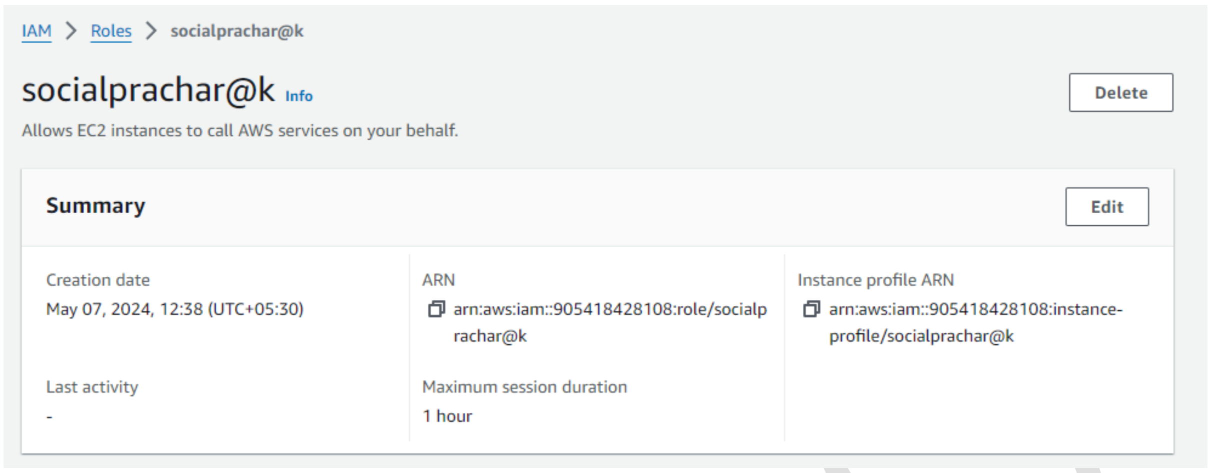
➢ **Now check for s3 bucket.**

< aws s3 ls >



```
int: Some files were ellipsized, use -l to show in full.
ec2-user@ip-172-31-11-131 ~]$ sudo chmod 666 /var/run/docker.sock
ec2-user@ip-172-31-11-131 ~]$ aws --version
ws-cli/1.18.147 Python/2.7.18 Linux/5.10.214-202.855.amzn2.x86_64 botocore/1.18.6
ec2-user@ip-172-31-11-131 ~]$ aws s3

sage: aws [options] <command> <subcommand> [parameters]
ws: error: too few arguments
ec2-user@ip-172-31-11-131 ~]$ aws s3 ls
nable to locate credentials. You can configure credentials by running "aws configure".
ec2-user@ip-172-31-11-131 ~]$
```

- **Now create the IAM role.**



IAM > Roles > socialprachar@k

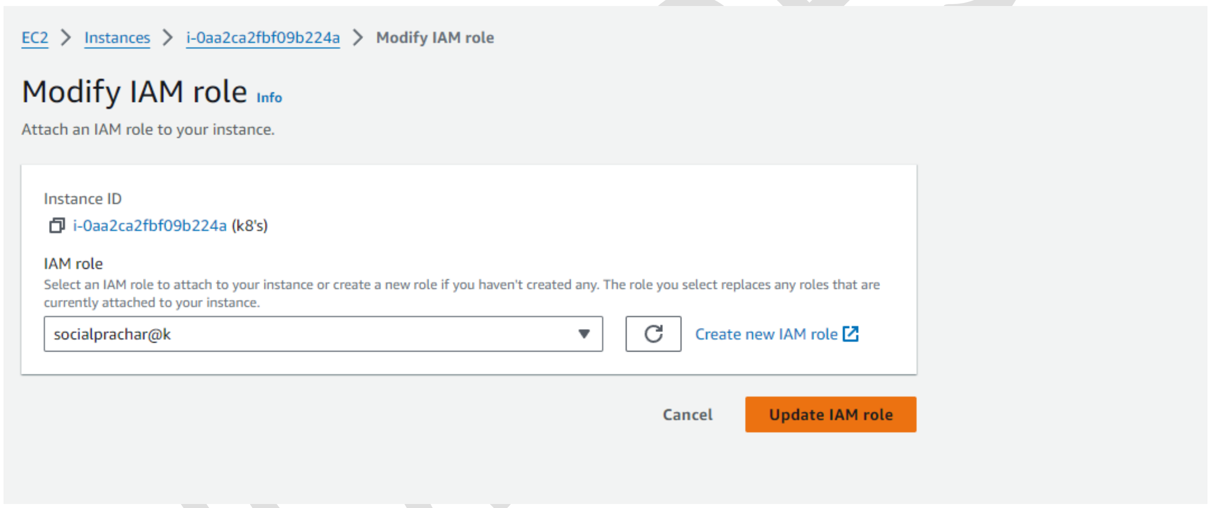
socialprachar@k

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date May 07, 2024, 12:38 (UTC+05:30)	ARN arn:aws:iam::905418428108:role/socialprachar@k	Instance profile ARN arn:aws:iam::905418428108:instance-profile/socialprachar@k
Last activity -	Maximum session duration 1 hour	

- **After that attach the EC2 instance.**



EC2 > Instances > i-0aa2ca2fbf09b224a > Modify IAM role

Modify IAM role

Attach an IAM role to your instance.

Instance ID
i-0aa2ca2fbf09b224a (k8's)

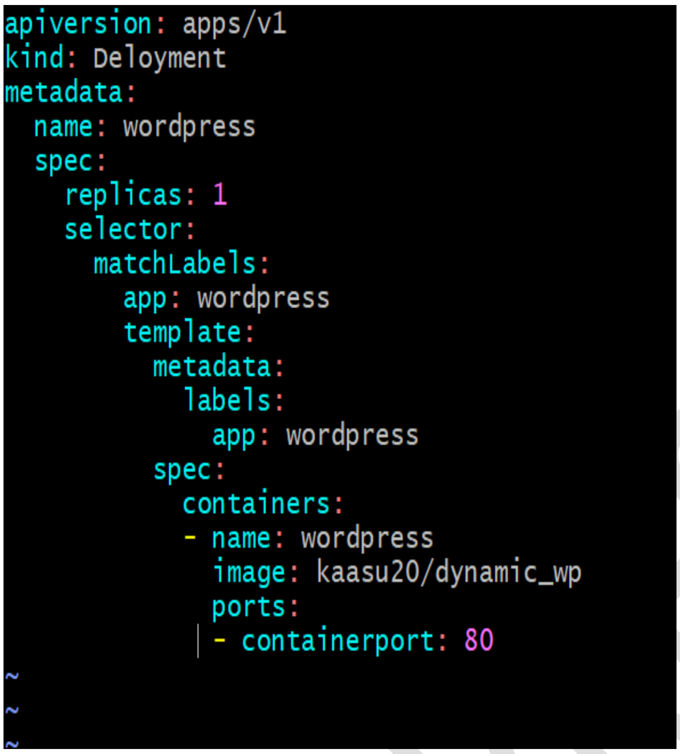
IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

socialprachar@k

- **After that create S3 bucket.**
- **After that check the following command as**

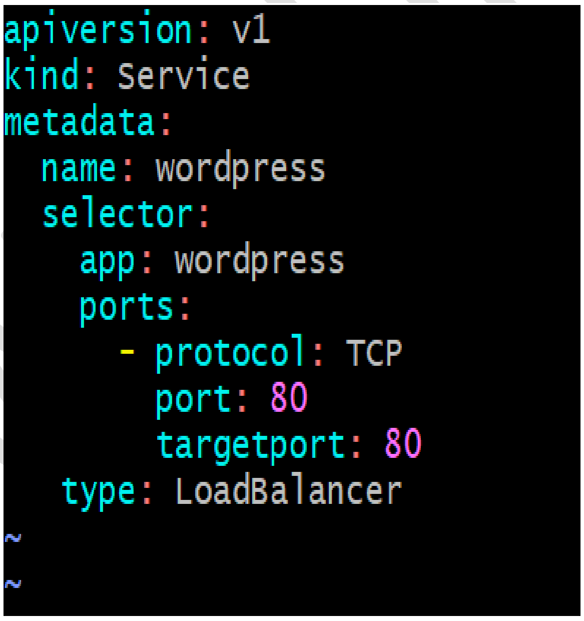
  < aws s3 ls >
- **After that install kops and kubectl (Google).**

➢ **Create one file deployment.yml.**



```
apiversion: apps/v1
kind: Deloyment
metadata:
  name: wordpress
  spec:
    replicas: 1
    selector:
      matchLabels:
        app: wordpress
        template:
          metadata:
            labels:
              app: wordpress
          spec:
            containers:
            - name: wordpress
              image: kaasu20/dynamic_wp
              ports:
              - containerport: 80
~
~
~
```

➢ **Create another file service.yml.**



```
apiversion: v1
kind: Service
metadata:
  name: wordpress
  selector:
    app: wordpress
    ports:
      - protocol: TCP
        port: 80
        targetport: 80
   type: LoadBalancer
~
~
```

➢ **Now enter the command as kops create cluster**

➢ **After that enter the command as Kops validate cluster**

➢ **Now enter the Kubectl apply –f deployment.yml**

➢ **And Kubectl apply –f service.yml**

- **After that Kubectl get svc wordpress**

****