Assignment: SetUp End to End Web Application

- Docker Installation / Docker Desktop SetUp
- Web Application - Front-End + Back-End Application
  - MariaDB - Back-End
  - Wordpress - Front-End

1. Setting Up Docker on Local or EC2 Instance

a. On Local Machine

1. **Install Docker:**

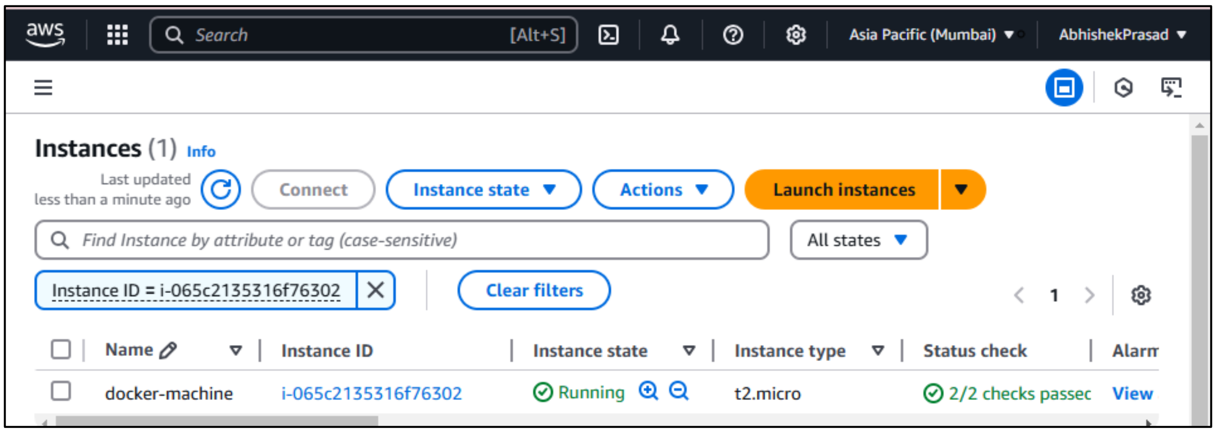
   - For Linux: Commands
     - sudo apt-get update
     - sudo apt-get install docker.io -y
     - sudo systemctl start docker
     - sudo systemctl enable docker
     - sudo usermod -aG docker $USER

2. **Verify Installations:**
     - docker –version

b. On EC2 Instance

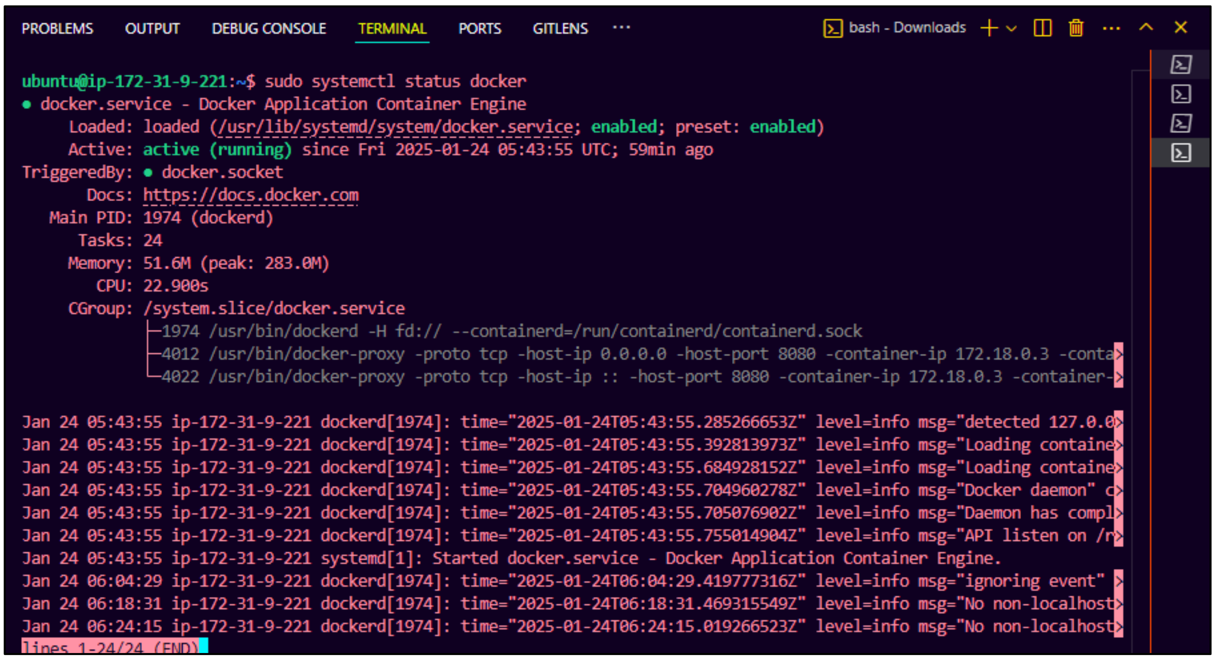
- **Launch an EC2 Instance:**
  - Use an Ubuntu 20.04/22.04 AMI.
  - Allow ports 22, 80, and 443 in the Security Group
- **Install Docker:**
  - sudo apt-get update
  - sudo apt-get install docker.io -y
  - sudo systemctl start docker
  - sudo systemctl enable docker
  - sudo usermod -aG docker ubuntu
- **Install Docker Compose**: (optionally)
  - 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
  - docker-compose --version

- Create a EC2 instance



Name	Instance ID	Instance state	Instance type	Status check	Alarm
docker-machine	i-065c2135316f76302	Running	t2.micro	2/2 checks passec	View

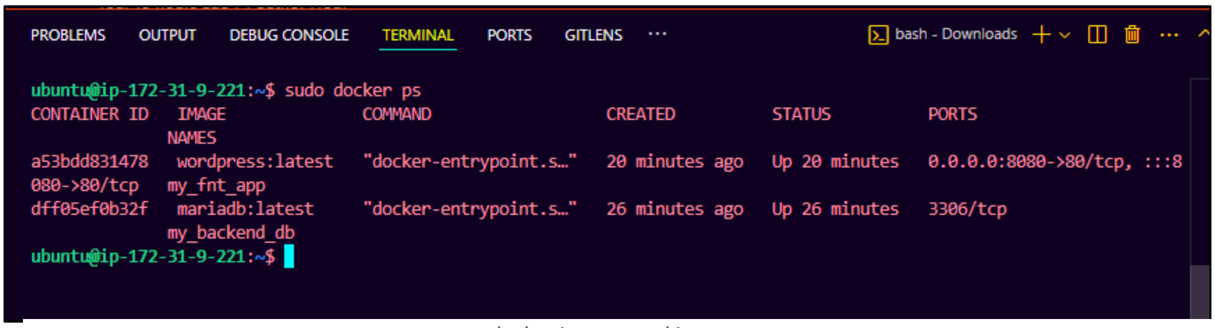
- Docker installation



```
ubuntu@ip-172-31-9-221:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-01-24 05:43:55 UTC; 59min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1974 (dockerd)
      Tasks: 24
     Memory: 51.6M (peak: 283.0M)
        CPU: 22.900s
     CGroup: /system.slice/docker.service
             ├─1974 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
             ├─4012 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 8080 -container-ip 172.18.0.3 -conta
             └─4022 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 8080 -container-ip 172.18.0.3 -container-

Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.285266653Z" level=info msg="detected 127.0.0
Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.392813973Z" level=info msg="Loading containe
Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.684928152Z" level=info msg="Loading containe
Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.704960278Z" level=info msg="Docker daemon" c
Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.705076902Z" level=info msg="Daemon has compl
Jan 24 05:43:55 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T05:43:55.755014904Z" level=info msg="API listen on /r
Jan 24 05:43:55 ip-172-31-9-221 systemd[1]: Started docker.service - Docker Application Container Engine.
Jan 24 06:04:29 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T06:04:29.419777316Z" level=info msg="ignoring event"
Jan 24 06:18:31 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T06:18:31.469315549Z" level=info msg="No non-localhost
Jan 24 06:24:15 ip-172-31-9-221 dockerd[1974]: time="2025-01-24T06:24:15.019266523Z" level=info msg="No non-localhost
lines 1-24/24 (END)
```

- Sudo docker ps



```
ubuntu@ip-172-31-9-221:~$ sudo docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
              NAMES
a53bdd831478   wordpress:latest   "docker-entrypoint.s…"   20 minutes ago   Up 20 minutes   0.0.0.0:8080->80/tcp, :::8
080->80/tcp   my_fnt_app
dff05ef0b32f   mariadb:latest     "docker-entrypoint.s…"   26 minutes ago   Up 26 minutes   3306/tcp
              my_backend_db
ubuntu@ip-172-31-9-221:~$
```

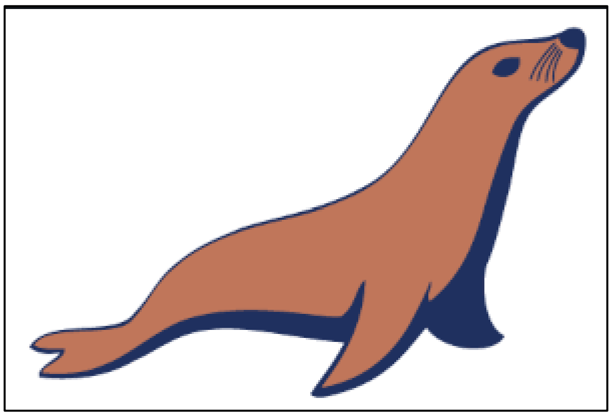
docker is now working stage

➢ Set Up Step by Step –

- ➢ **Execute MariaDB Container First.**
  - Make the Container name Static like - my-mariadb-container
  - Set the DB Root password using Env Variable - MARIADB_ROOT_PASSWORD
  - Set the DB Username using Env Variable - MARIADB_USER
  - Set the DB password using Env Variable - MARIADB_PASSWORD
  - Set the DB Name using Env Variable - MARIADB_DATABASE

1. What is MariaDB?

- MariaDB Server is a high performing open-source relational database, forked from MySQL.
- MariaDB Server is one of the most popular database servers in the world. It's made by the original developers of MySQL and guaranteed to stay open source. Notable users include Wikipedia, DBS Bank, and ServiceNow.
- The intent is also to maintain high compatibility with MySQL, ensuring a library binary equivalency and exact matching with MySQL APIs and commands. MariaDB developers continue to develop new features and improve performance to better serve its users.



How to use this image

The mariadb has a number of tags, and of note is latest, as the latest stable version, and lts, as the last long term support release.

Running the container

Starting using a minimal configuration

The environment variables required to use this image involves the setting of the root user password:

- $ docker run --detach --name some-mariadb --env MARIADB_ROOT_PASSWORD=my-secret-pw mariadb:latest
- $ docker run --detach --name some-mariadb --env MARIADB_ALLOW_EMPTY_ROOT_PASSWORD=1 mariadb:latest
- $ docker run --detach --name some-mariadb --env MARIADB_RANDOM_ROOT_PASSWORD=1 mariadb:latest

... where the container logs will contain the generated root password.

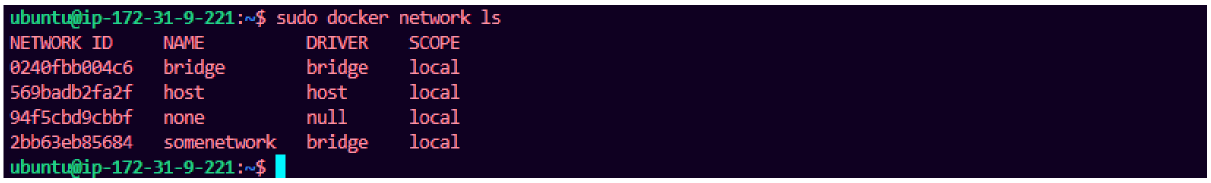
Create a Custom Docker Network

$ docker network create **some-network**

$ docker network ls

Check if your container is running and connected to the desired network:

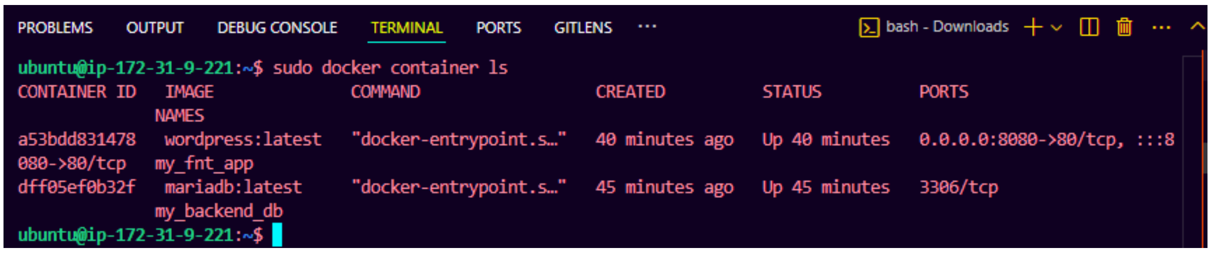
Use the Default bridge Network (If You Don't Need a Custom Network)



```
ubuntu@ip-172-31-9-221:~$ sudo docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
0240fbb004c6   bridge        bridge    local
569badb2fa2f   host          host      local
94f5cbd9cbbf   none          null      local
2bb63eb85684   somenetwork   bridge    local
ubuntu@ip-172-31-9-221:~$
```

- -$ docker run --detach --name some-mariadb --env MARIADB_USER=example-user --env MARIADB_PASSWORD=my_cool_secret --env MARIADB_DATABASE=exmple-database --env MARIADB_ROOT_PASSWORD=my-secret-pw mariadb:latest

- $ sudo docker container run -d -e MARIADB_USER=db_user -e MARIADB_PASSWORD=db_pass -e MARIADB_DATABASE=frontend_app -e MARIADB_ROOT_PASSWORD=root_pass --network somenetwork --name my_backend_db mariadb:latest



```
ubuntu@ip-172-31-9-221:~$ sudo docker container ls
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
              NAMES
a53bdd831478   wordpress:latest   "docker-entrypoint.s…"   40 minutes ago   Up 40 minutes   0.0.0.0:8080->80/tcp, :::8
080->80/tcp   my_fnt_app
dff05ef0b32f   mariadb:latest     "docker-entrypoint.s…"   45 minutes ago   Up 45 minutes   3306/tcp
              my_backend_db
ubuntu@ip-172-31-9-221:~$
```

See the container is currently running over the docker

➢ Set Up Step by Step – WordPress Container

➢ **Execute Wordpress Container –**

- Make the Container name Static like - wordpress-container
- Set the DB Container Name in Env Variable - WORDPRESS_DB_HOST
- Set the DB Name in Env Variable - WORDPRESS_DB_NAME
- Set the DB User in Env Variable - WORDPRESS_DB_USER
- Set the DB password in Env Variable - WORDPRESS_DB_PASSWORD
- Expose Front-End Container on port 8080/80
- Access WebSite on LocalHost/HostIP:PORT

2. Word press

- The WordPress rich content management system can utilize plugins, widgets, and themes.

```
$ docker pull wordpress
```

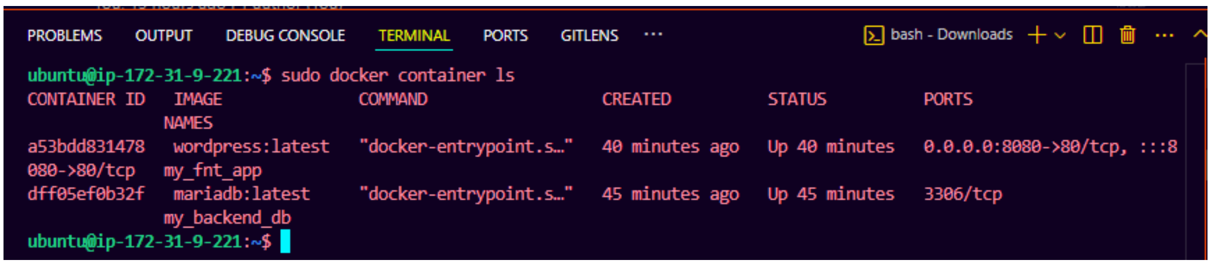
What is wordpress?

WordPress is a free and open-source blogging tool and a content management system (CMS) based on PHP and MySQL, which runs on a web hosting service. Features include a plugin architecture and a template system. WordPress is used by more than 22.0% of the top 10 million websites as of August 2013. WordPress is the most popular blogging system in use on the Web, at more than 60 million websites. The most popular languages used are English, Spanish and Bahasa Indonesia.

How to use this image

```
$ docker run --name some-wordpress --network some-network -d wordpress
-e WORDPRESS_DB_HOST=...
-e WORDPRESS_DB_USER=...
-e WORDPRESS_DB_PASSWORD=...
-e WORDPRESS_DB_NAME=...
-e WORDPRESS_TABLE_PREFIX=...
$ docker run --name some-wordpress -p 8080:80 -d wordpress
```

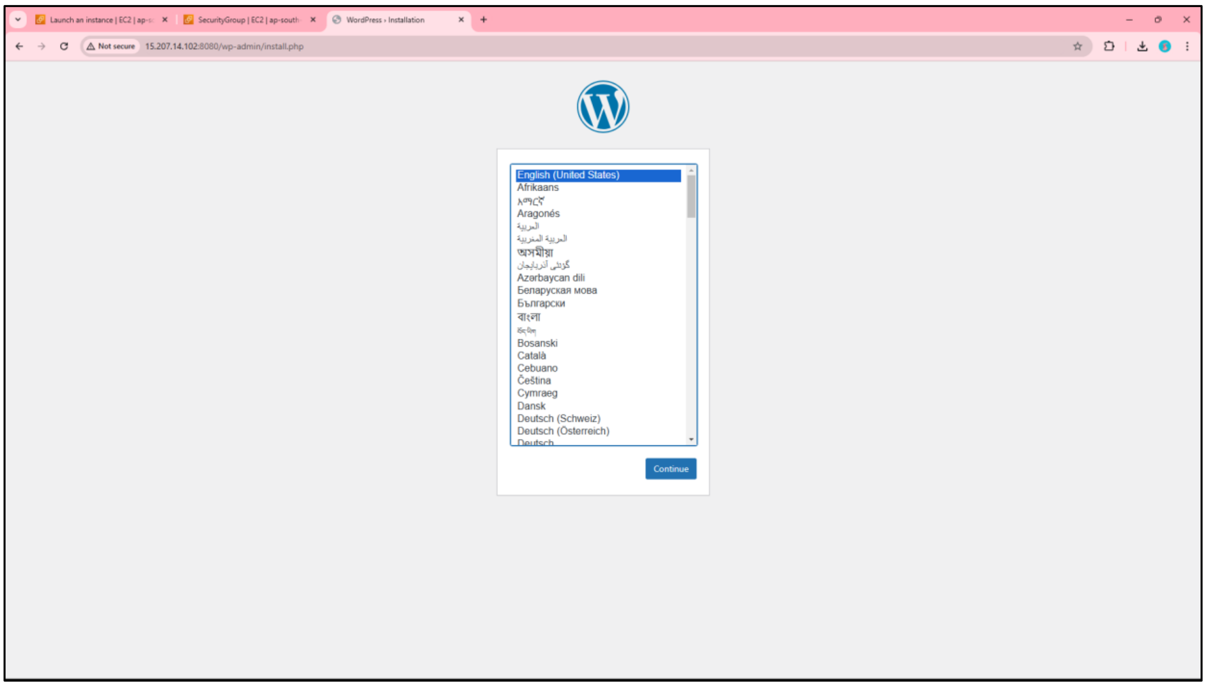
docker container run -d -e WORDPRESS_DB_HOST=**my_backend_db** -e WORDPRESS_DB_USER=**db_user** -e WORDPRESS_DB_PASSWORD=**db_pass** -e WORDPRESS_DB_NAME=**frontend_app** -p **8080:80** --network **somenetwork** --name **my_fnt_app** **wordpress**:latest



```
ubuntu@ip-172-31-9-221:~$ sudo docker container ls
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
              NAMES
a53bdd831478   wordpress:latest   "docker-entrypoint.s…"   40 minutes ago   Up 40 minutes   0.0.0.0:8080->80/tcp, :::8
080->80/tcp   my_fnt_app
dff05ef0b32f   mariadb:latest     "docker-entrypoint.s…"   45 minutes ago   Up 45 minutes   3306/tcp
              my_backend_db
ubuntu@ip-172-31-9-221:~$
```

Now my both containers are running 1. Mariadb | 2. Wordpress

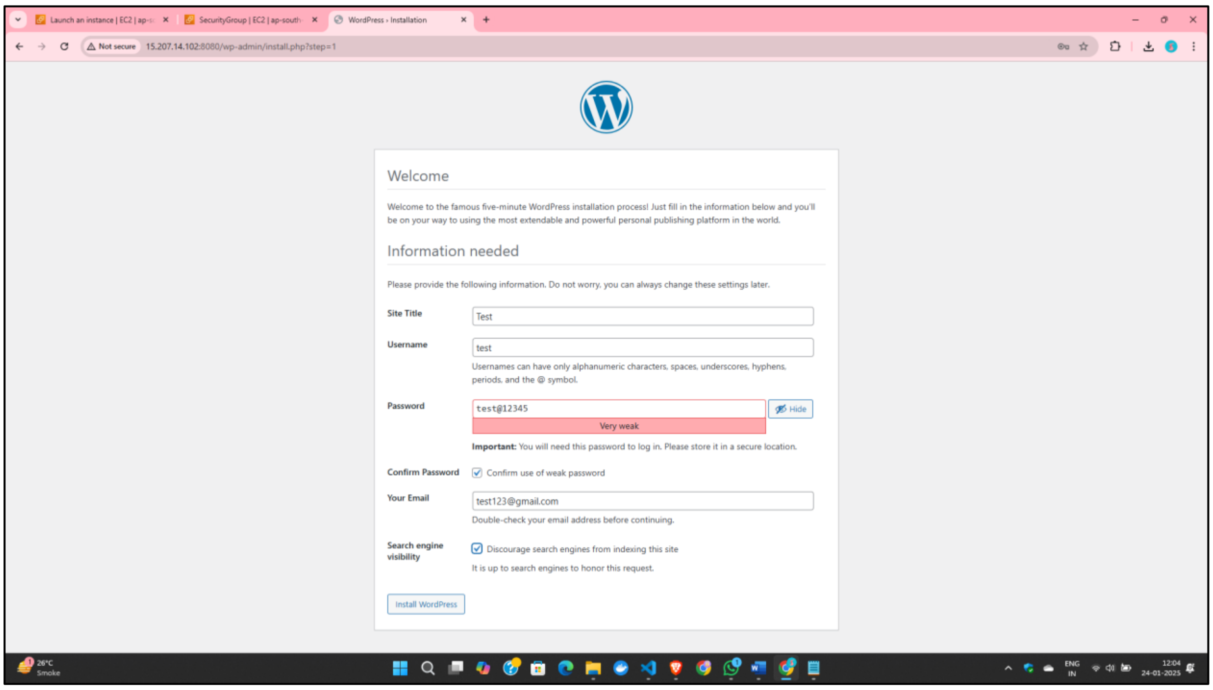
Now I am going to a browser and hit IP address over the search bar then the browser response me.



15.207.14.102:8080/wp-admin/install.php

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

Continue



15.207.14.102:8080/wp-admin/install.php?step=1

Welcome

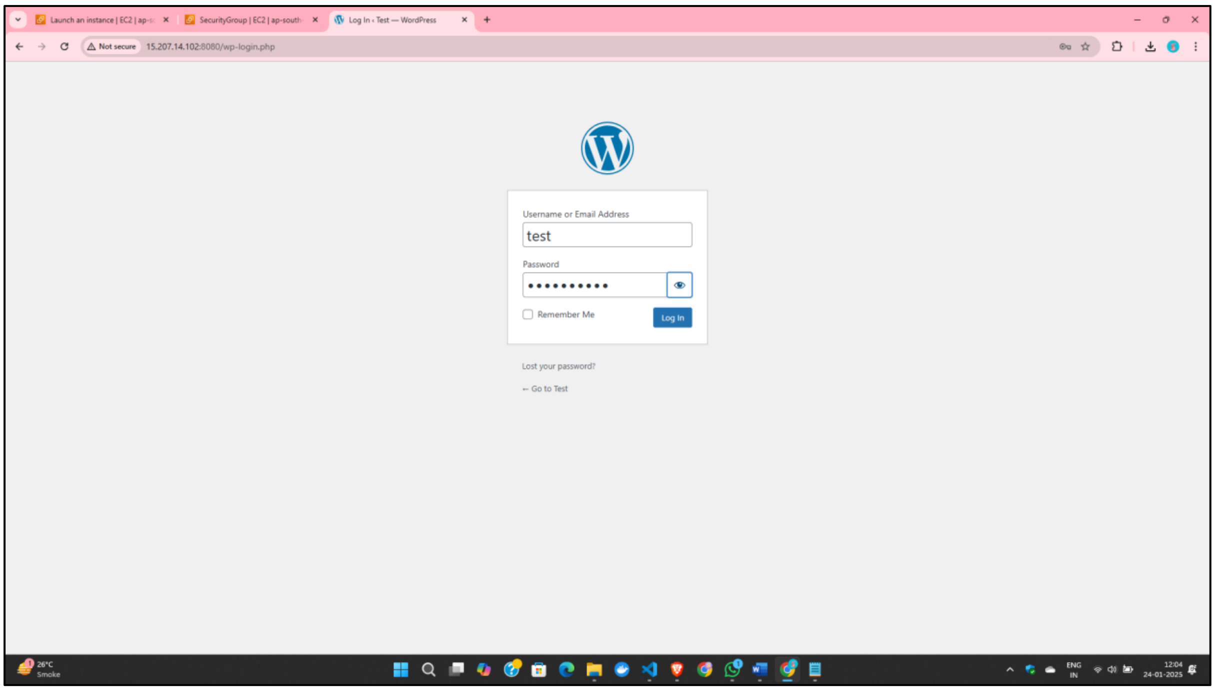
Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

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

Site Title	Test
Username	test Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.
Password	test@12345 Hide Very weak **Important:** You will need this password to log in. Please store it in a secure location.
Confirm Password	✓ Confirm use of weak password
Your Email	test123@gmail.com Double-check your email address before continuing.
Search engine visibility	✓ Discourage search engines from indexing this site It is up to search engines to honor this request.

Install WordPress


Launch an instance | EC2 | ap-s
SecurityGroup | EC2 | ap-south-
Log In ‹ Test — WordPress
Not secure
15.207.14.102:8080/wp-login.php

Username or Email Address

test

Password

••••••••••

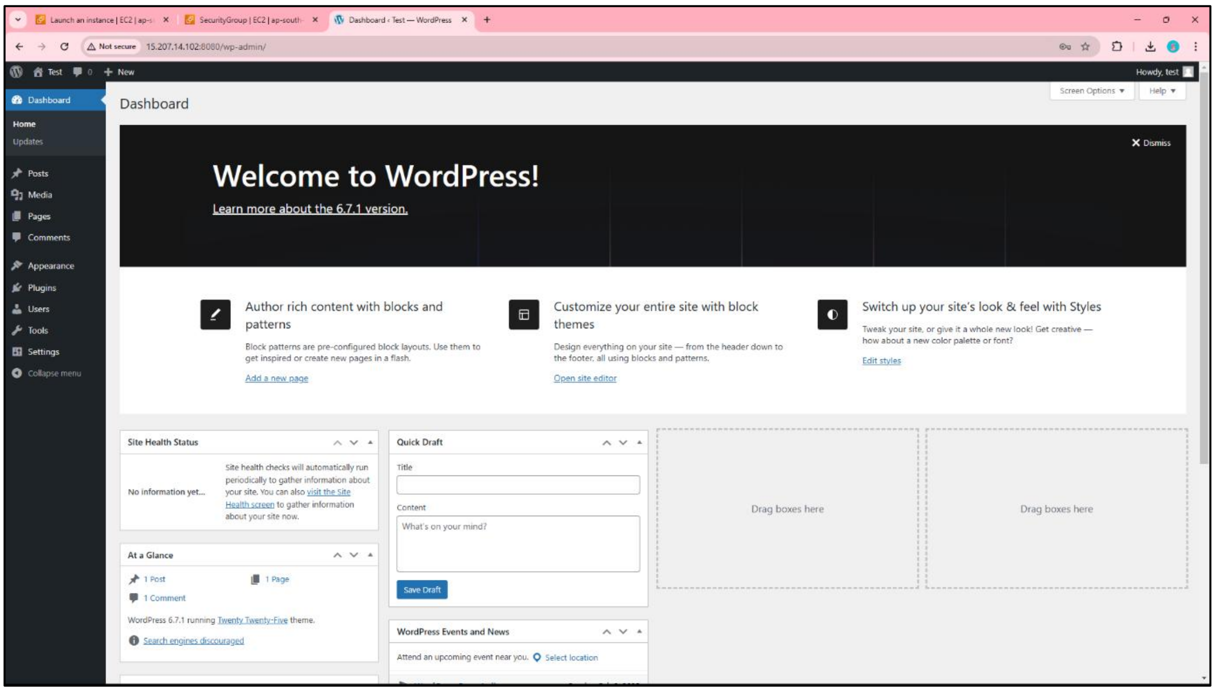
Remember Me

Log In

Lost your password?

← Go to Test

26°C
Smoke
ENG
IN
12:04
24-01-2025



Dashboard

Welcome to WordPress!

Learn more about the 6.7.1 version.

Author rich content with blocks and patterns

Block patterns are pre-configured block layouts. Use them to get inspired or create new pages in a flash.

Add a new page

Customize your entire site with block themes

Design everything on your site — from the header down to the footer, all using blocks and patterns.

Open site editor

Switch up your site's look & feel with Styles

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

Edit styles

Site Health Status

No information yet...

Site health checks will automatically run periodically to gather information about your site. You can also visit the Site Health screen to gather information about your site now.

At a Glance

1 Post

1 Page

1 Comment

WordPress 6.7.1 running Twenty Twenty-Five theme.

Search engines discouraged

Quick Draft

Title

Content

What's on your mind?

Save Draft

WordPress Events and News

Attend an upcoming event near you. Select location

Drag boxes here

Drag boxes here

This Assignment are Done :Setup End to End Web Application.