

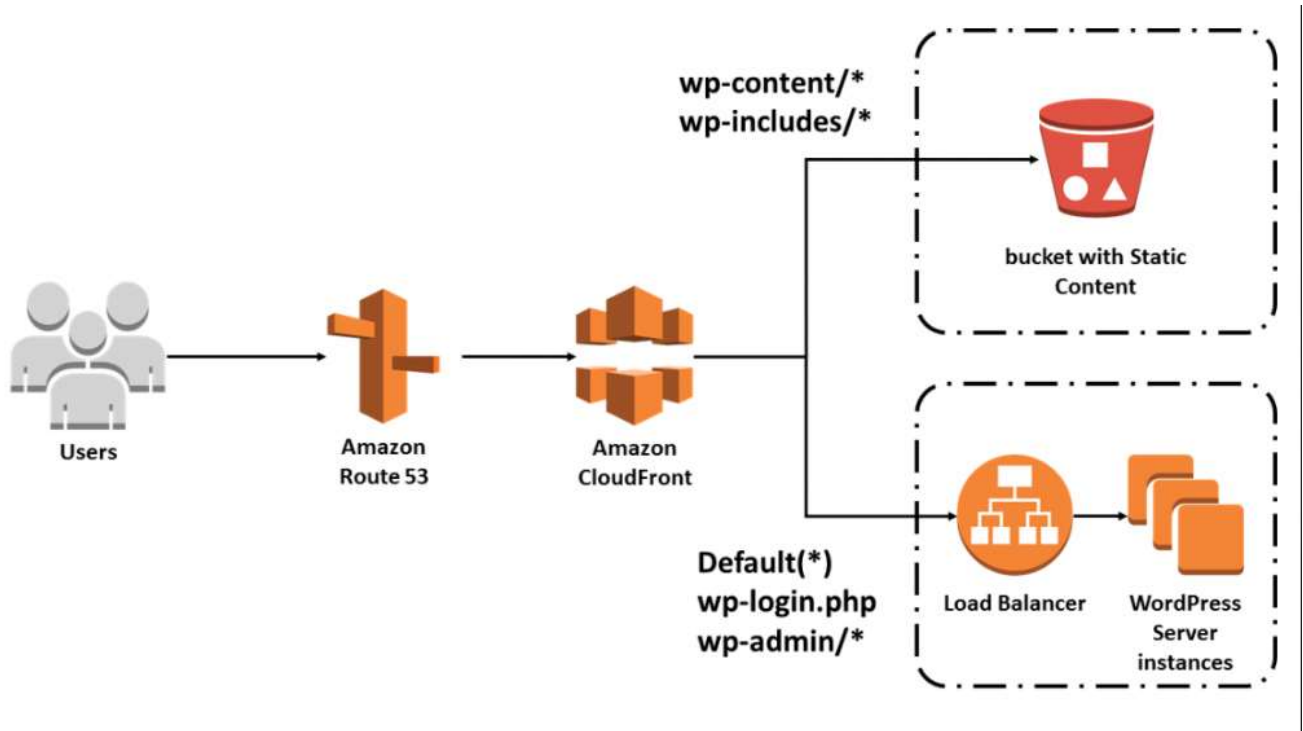
HOSTING A WORDPRESS WEBSITE

2100030958

Description:

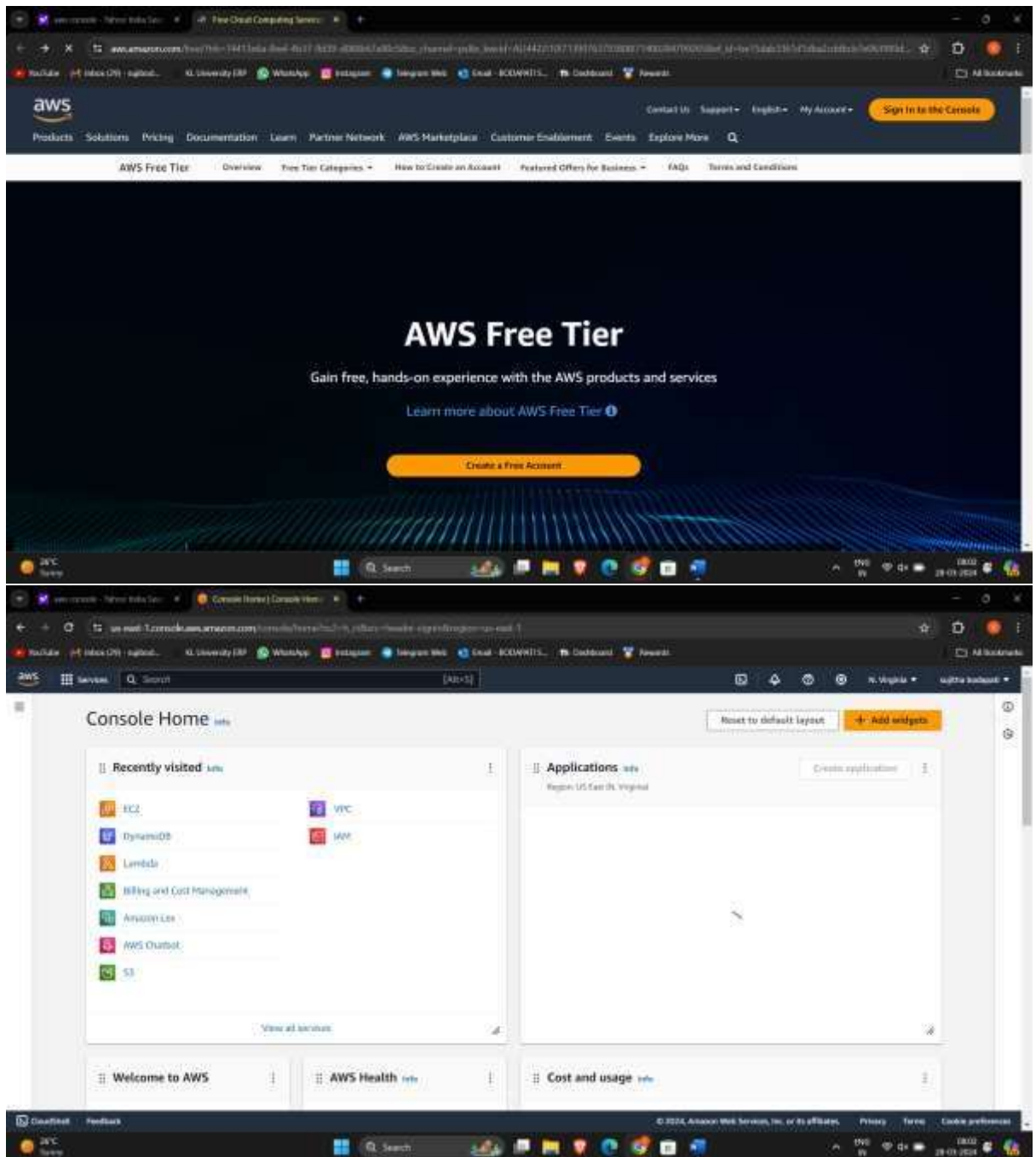
A WordPress website is a dynamic online platform built on the WordPress content management system (CMS), renowned for its user-friendly interface and extensive customization options. With an array of themes and plugins available, users can effortlessly create, design, and manage their websites to suit their unique needs and preferences. From blogs and portfolios to e-commerce stores and corporate sites, WordPress offers flexibility and scalability, catering to a diverse range of purposes. Its responsive design ensures optimal viewing experiences across devices, while its SEO-friendly features facilitate improved search engine visibility. With robust security measures and user management capabilities, WordPress empowers individuals and businesses alike to establish and maintain professional and engaging online presences with ease.

Architectures:



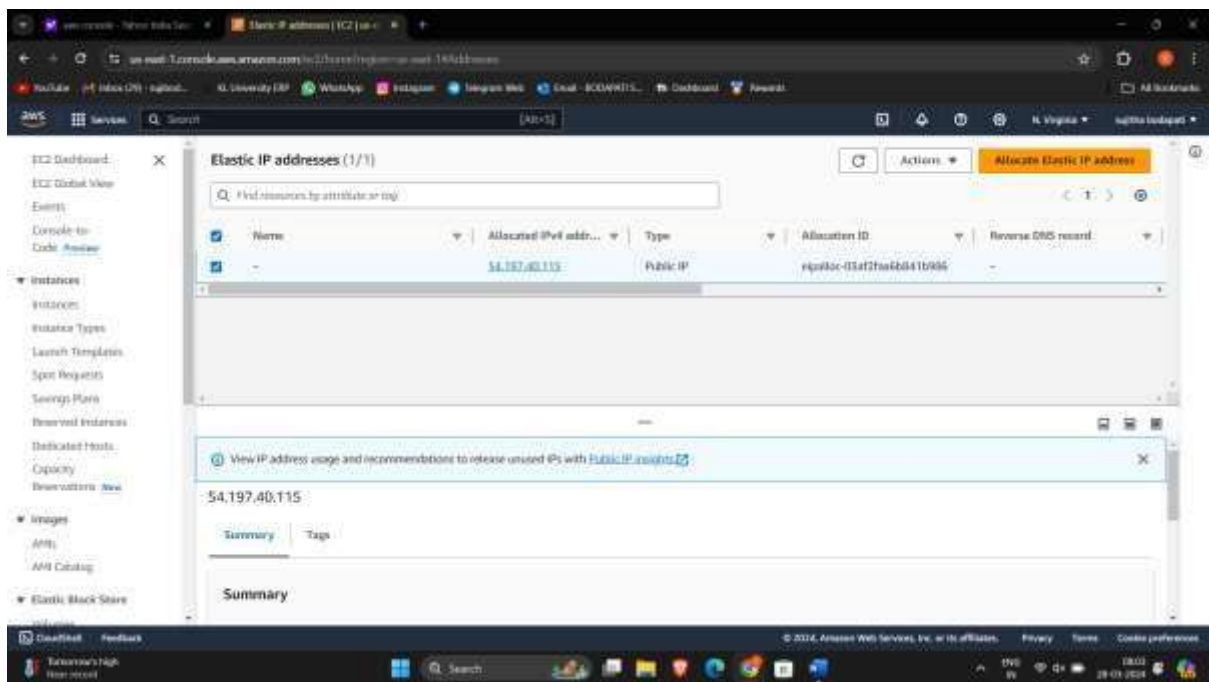
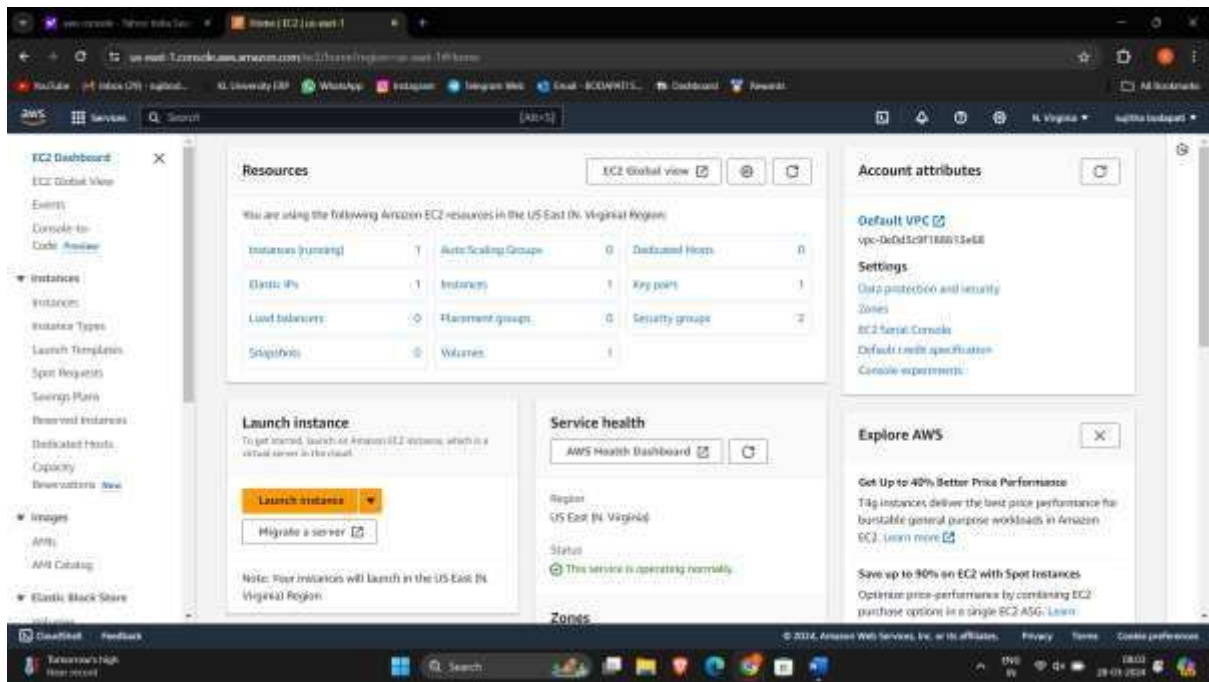
Hosting a WordPress website on AWS EC2 involves several steps. Here's a clear guide:

1. ****Sign in to AWS Console****: Log in to your AWS Management Console at <https://console.aws.amazon.com/>.



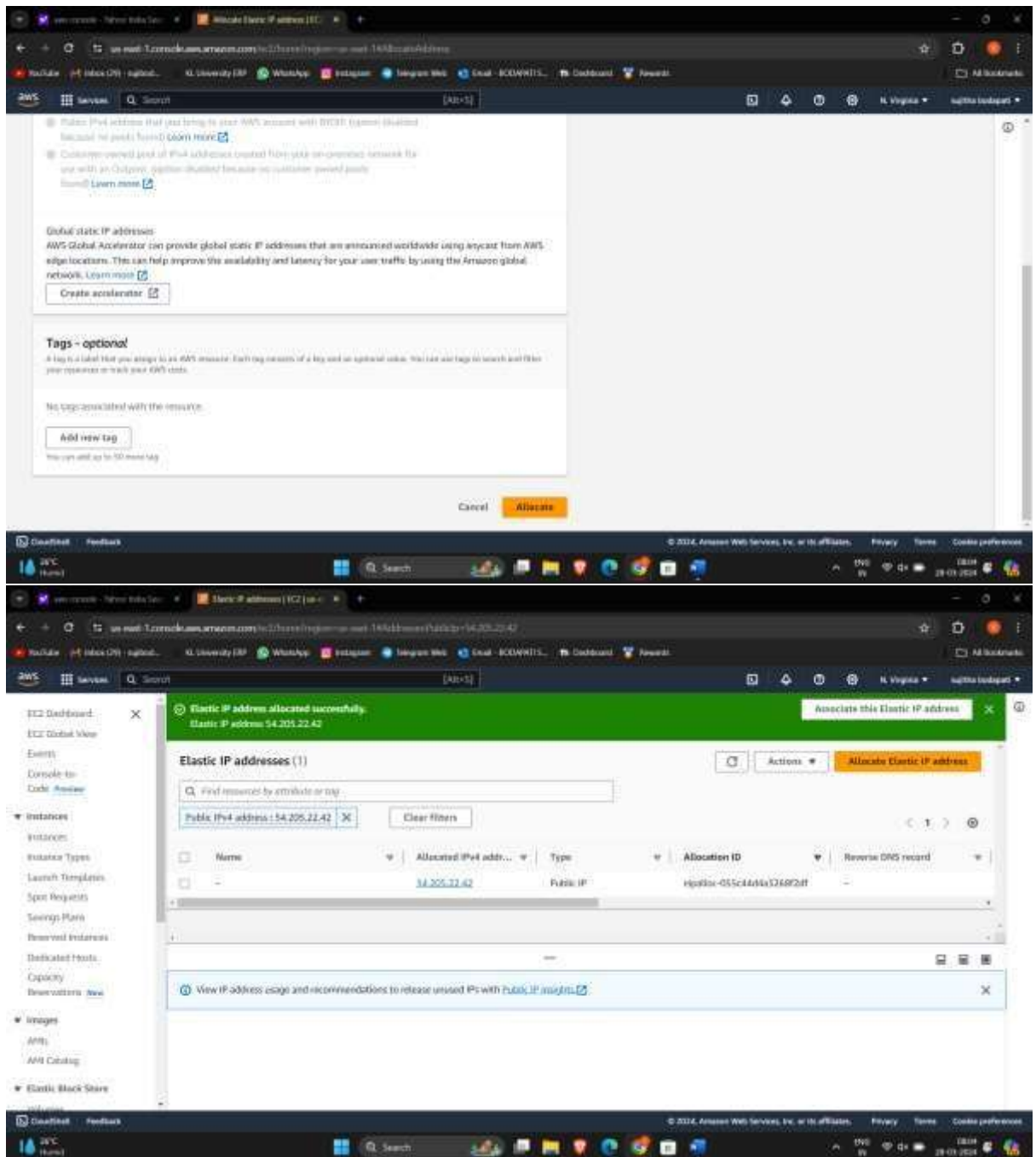
2. ****Launch an EC2 Instance****:

- Navigate to the EC2 dashboard.
- Click on "Launch Instance" to start the process.
- Choose an Amazon Machine Image (AMI). You can select an AMI that includes WordPress preinstalled, such as "Amazon Linux 2 AMI (HVM), SSD Volume Type."
- Select an instance type based on your requirements. For WordPress, a t2.micro instance should be sufficient for low traffic websites.
- Configure instance details, such as network settings, subnet, and security groups.
- Review and launch the instance.



2. ****Allocate an Elastic IP (Optional)**:** For a static IP address, allocate an Elastic IP and associate it with your EC2 instance to avoid IP changes upon instance restarts.

4.



****Connect to Your Instance**:**

- Once the instance is launched, select it from the Instances list in the EC2 dashboard.
- Click on "Connect" to get instructions on how to connect to your instance using SSH.

5. ****Install mobxterm Stack**:**

- Connect to your instance via SSH using a terminal or SSH client.
- Install the LAMP (Linux, Apache, MySQL, PHP) stack. Run the following commands:

...

```

sudo yum update -y    sudo yum install -y httpd24 php74
mysql57-server php74-mysqld    sudo service httpd start    sudo
chkconfig httpd on    sudo groupadd www    sudo usermod -a -G
www ec2-user

exit
'''

```

- Log out of the SSH session and log back in to apply group membership changes.

6. **Install WordPress**:

- Download and install WordPress. Run the following commands:

```

'''

cd /var/www/html    sudo wget
https://wordpress.org/latest.tar.gz

sudo tar -xzf latest.tar.gz

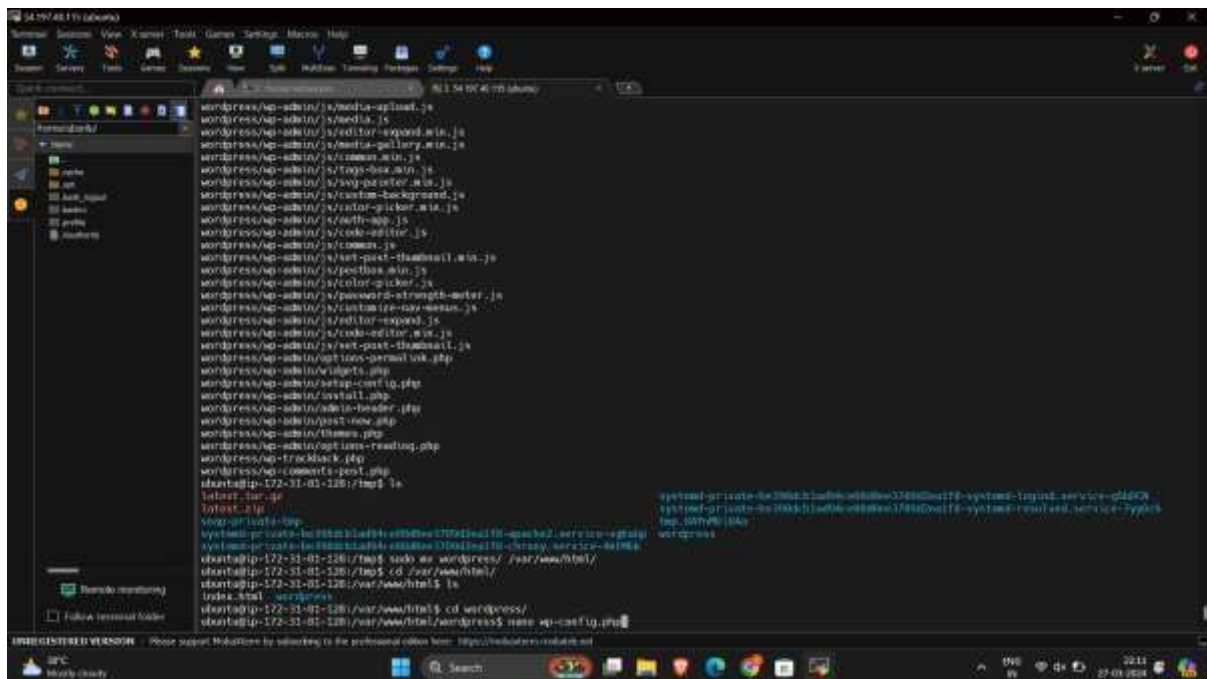
sudo mv wordpress/* .

sudo rm -rf wordpress latest.tar.gz

sudo chown -R apache:apache /var/www/html

'''

```



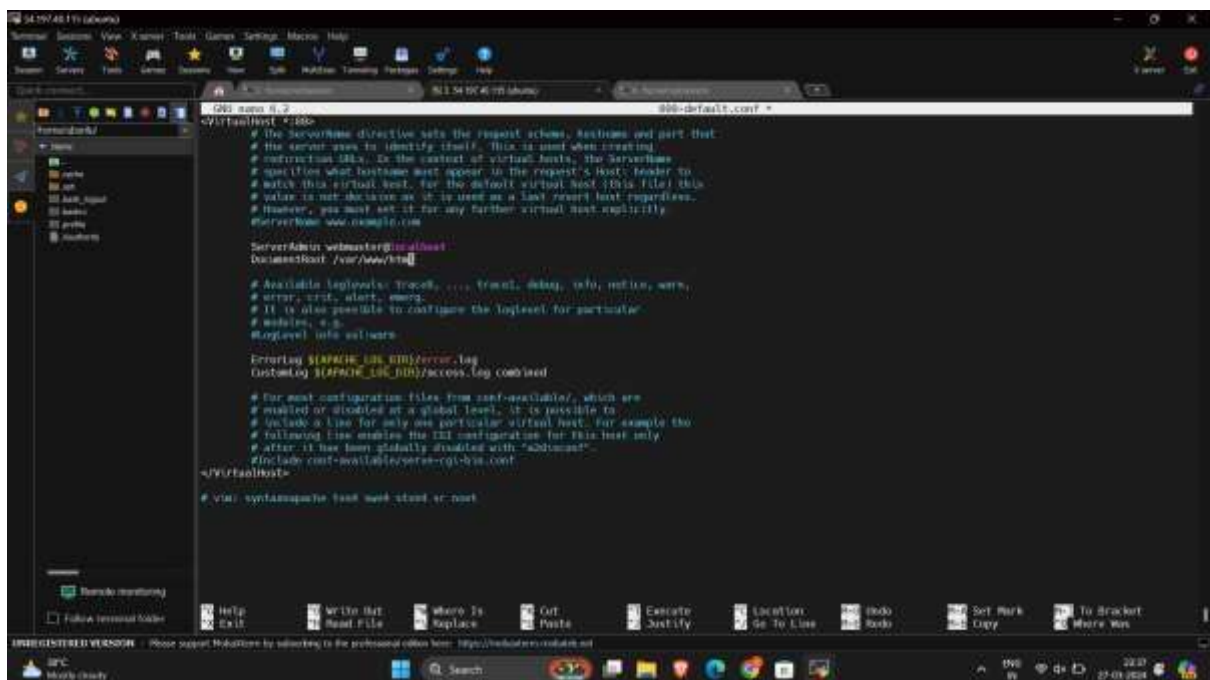
7. **Configure WordPress**:

- Set up a MySQL database and user for WordPress. Use the MySQL commands like `mysql -u root p` to access MySQL and create a new database and user.
- Navigate to <http://Your-EC2-Public-IP> in a web browser to complete the WordPress installation. Follow the on-screen instructions and enter the database information when prompted.

8. **Configure Apache**:

- Create an .htaccess file in the /var/www/html directory to configure Apache. Add the following content:

```
sudo nano /var/www/html/.htaccess
```



```
# BEGIN WordPress
#
# The RewriteEngine directive sets the Request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection rules. In the context of virtual hosts, the RewriteBase
# specifies what hostname must appear in the request's host header to
# match this virtual host. For the default virtual host (this file) this
# value is not necessary as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
# RewriteBase www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.:
# LogLevel info trace4+
#
# The following line defines the loglevel for the Rewrite module
LogLevel warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the SSL configuration for this host only.
# after it has been globally disabled with "ssl.enable"
# Include conf-available/ssl.conf

</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr=nost
```

BEGIN WordPress

RewriteEngine On

RewriteBase /

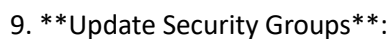
RewriteRule ^index\.php\$ - [L]

RewriteCond %{REQUEST_FILENAME} !-f

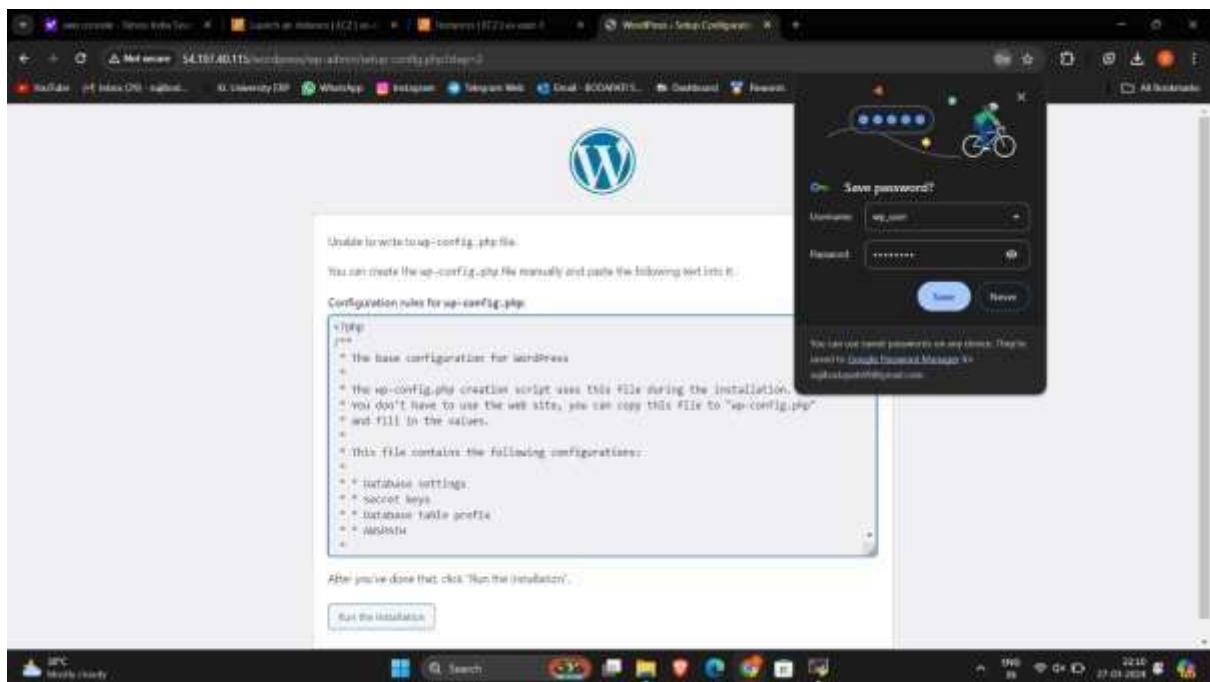
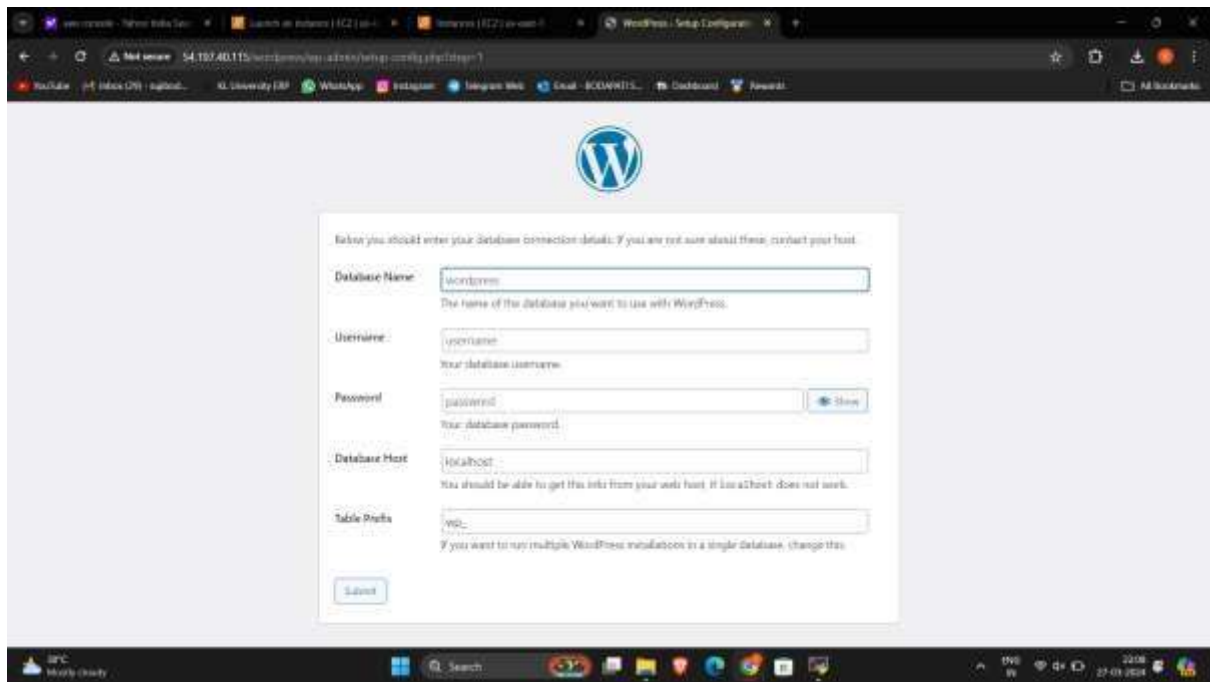
RewriteCond %{REQUEST_FILENAME} !-d

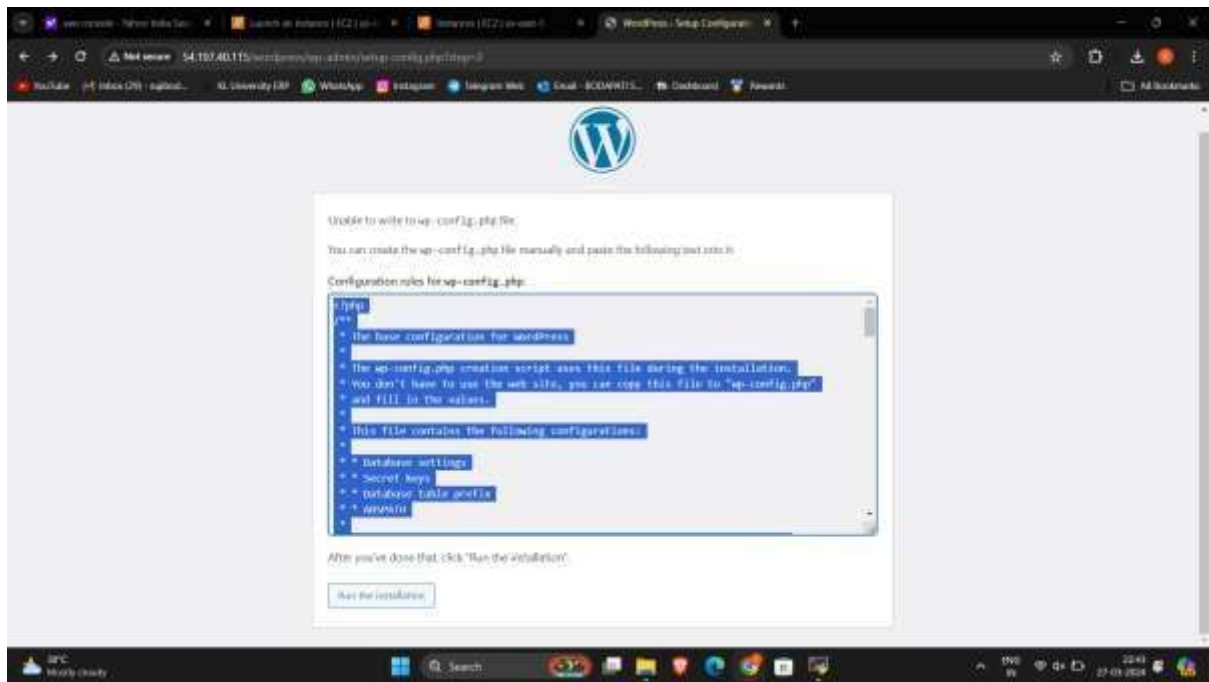
RewriteRule . /index.php [L]

...



-





10. ****Configure SSL (Optional)****:

- For HTTPS access, you can configure SSL/TLS certificates using AWS Certificate Manager (ACM) or third-party providers like Let's Encrypt.

That's it! Your WordPress website should now be up and running on AWS EC2. Remember to regularly update WordPress, themes, and plugins for security and performance improvements.

