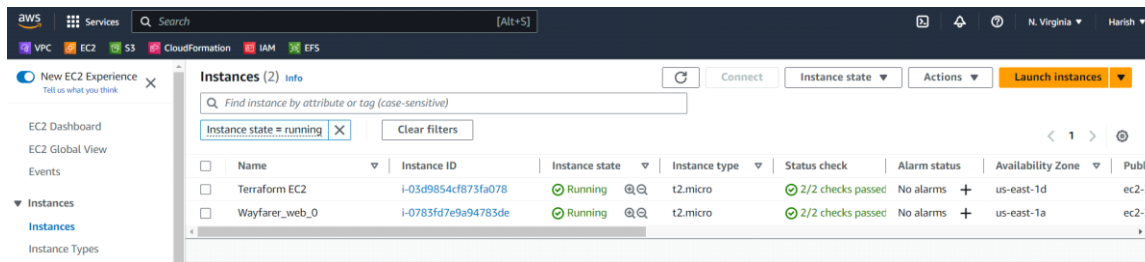
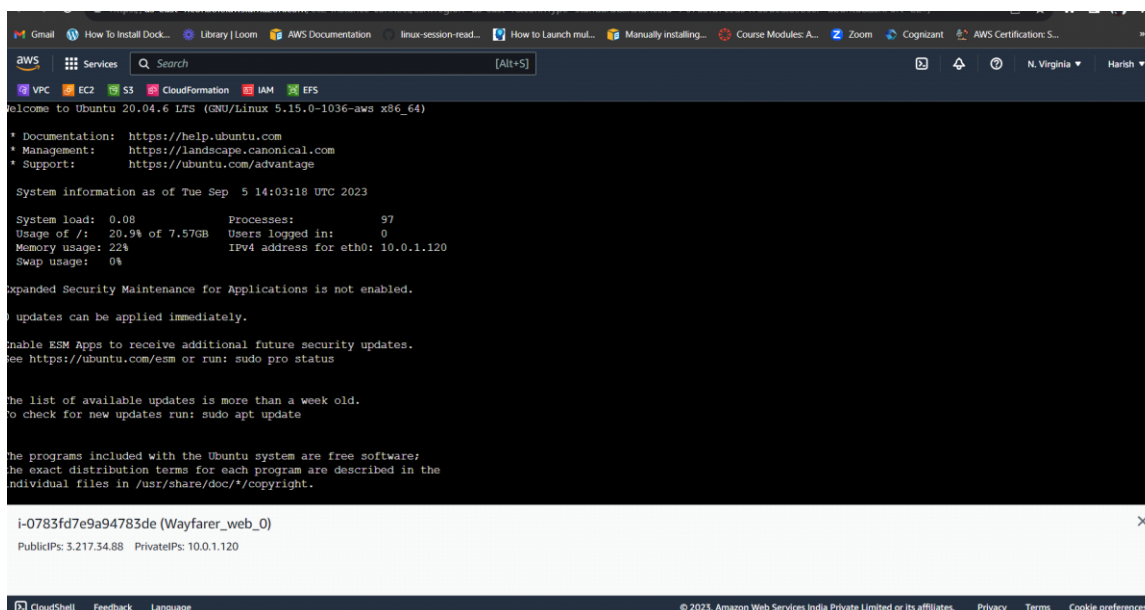


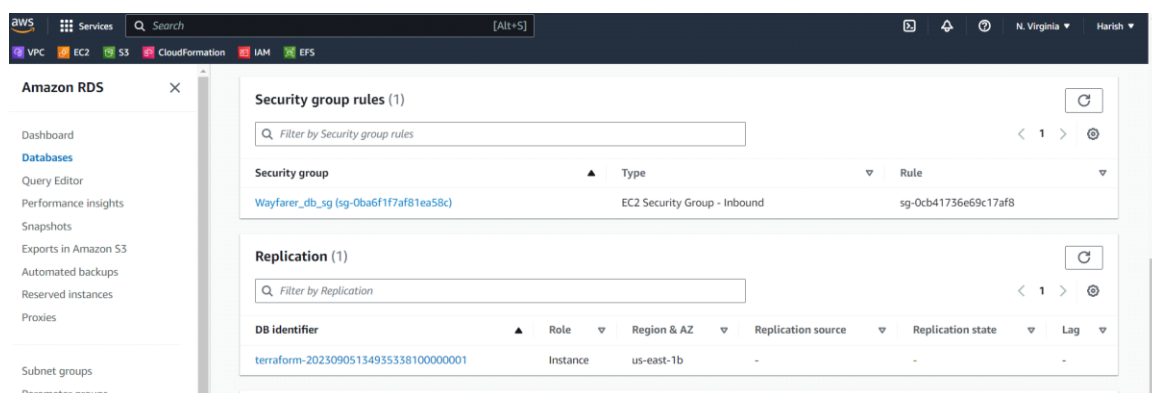
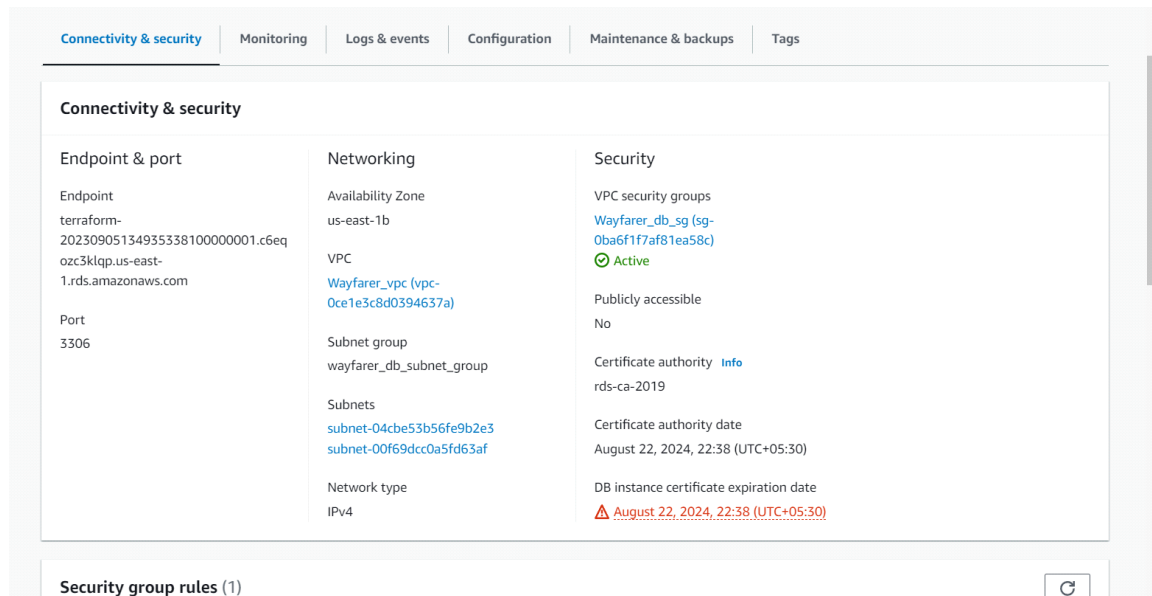
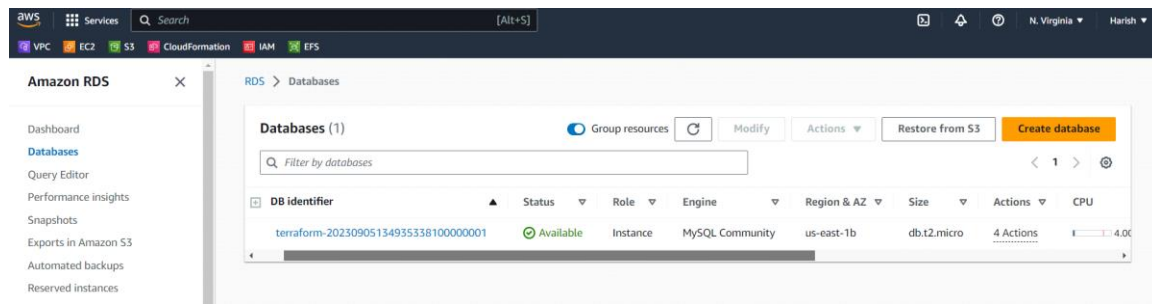
Terraform execution done after that gone to Aws console, ec2 is launched



After by connecting the ec2 instance in that we have the connect option in the ec2 service, it shows like this



After going thru the RDS the database running on it identifier as **terraform-20230905134935338100000001**



by installing a docker file with Wordpad

```
Preparing to unpack .../5-docker-compose-plugin_2.20.2-1-ubuntu.20.04-focal_amd64.deb ...
Unpacking docker-compose-plugin (2.20.2-1-ubuntu.20.04-focal) ...
Selecting previously unselected package slirp4netns.
Preparing to unpack .../6-slirp4netns_0.4.3-1_amd64.deb ...
Unpacking slirp4netns (0.4.3-1) ...
Setting up slirp4netns (0.4.3-1) ...
Setting up docker-buildx-plugin (0.11.2-1-ubuntu.20.04-focal) ...
Setting up containerd.io (1.6.22-1) ...
Setting up docker-compose-plugin (2.20.2-1-ubuntu.20.04-focal) ...
Setting up docker-ce-cli (5:24.0.5-1-ubuntu.20.04-focal) ...
Setting up docker-ce-rootless-extras (5:24.0.5-1-ubuntu.20.04-focal) ...
Setting up docker-ce (5:24.0.5-1-ubuntu.20.04-focal) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.21) ...
root@ip-10-0-1-120:~# docker --version
docker version 24.0.5, build ced0996
root@ip-10-0-1-120:~# curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-Linux-x86_64 -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 12.1M  100 12.1M    0     0  50.1M      0 --:--:-- --:--:-- --:--:--  50.8M
root@ip-10-0-1-120:~# chmod +x /usr/local/bin/docker-compose
root@ip-10-0-1-120:~# docker-compose --version
docker-compose version 1.29.2, build 5bece44d
root@ip-10-0-1-120:~# mkdir wordpress
root@ip-10-0-1-120:~# nano wordpress/docker-compose.yml
root@ip-10-0-1-120:~# cd wordpress
root@ip-10-0-1-120:~/wordpress# docker-compose up -d
Creating network "wordpress_default" with the default driver

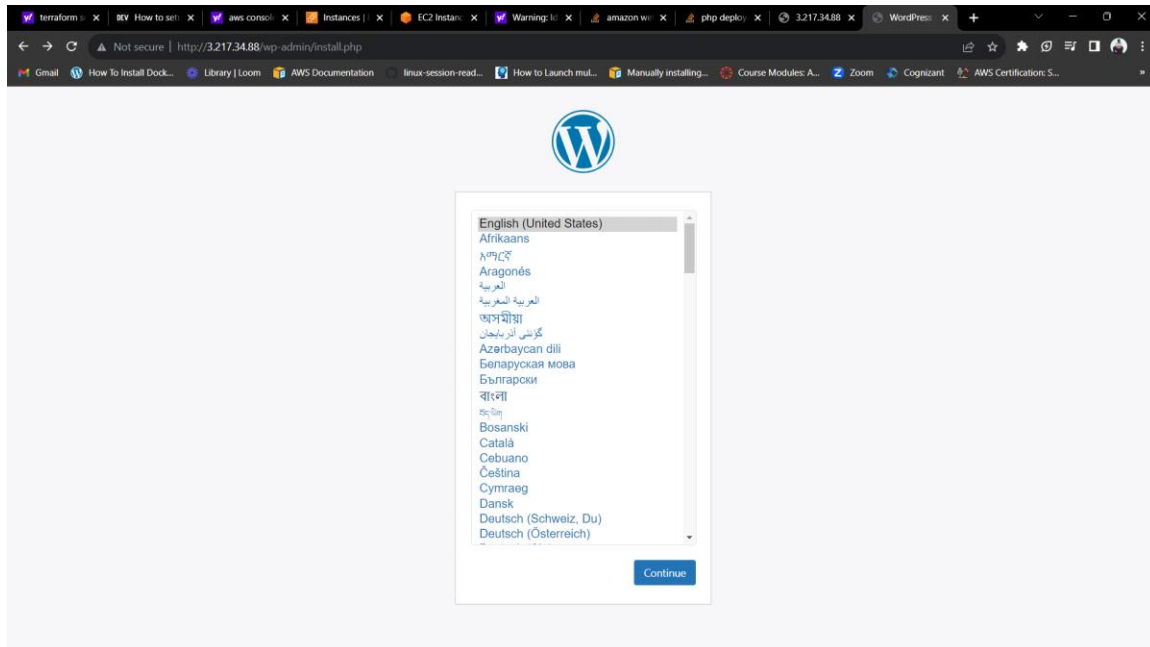
i-0783fd7e9a94783de (Wayfarer_web_0)
PublicIPs: 3.217.34.88 PrivateIPs: 10.0.1.120
```

download WordPress and MySQL image from the Docker repository starts both containers in the background and leaves them running

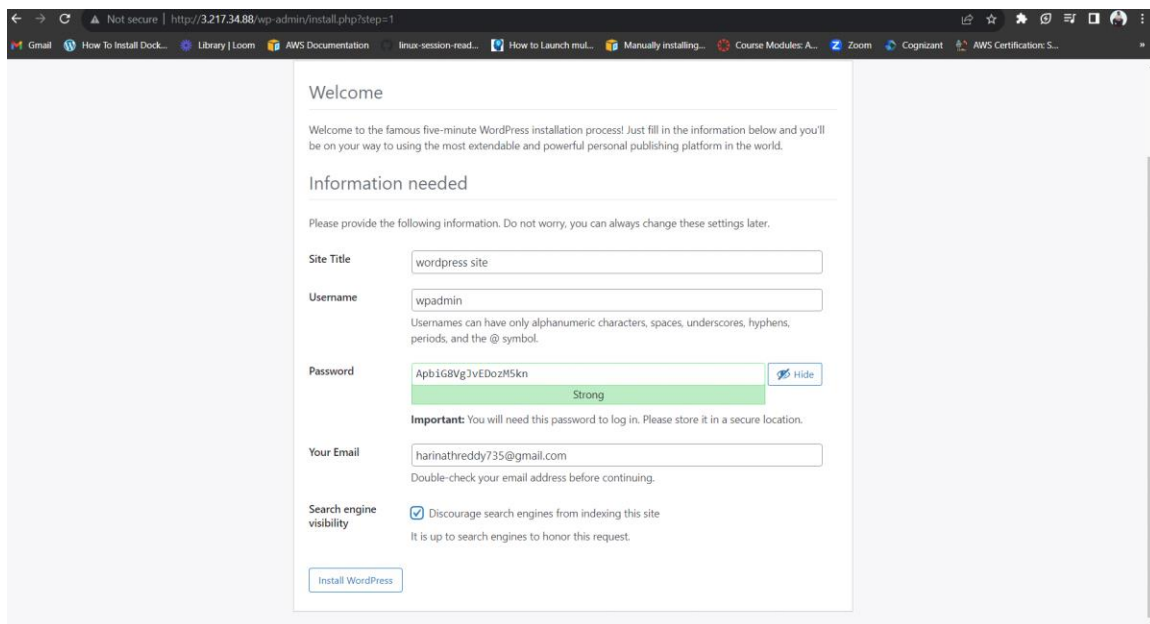
```
root@ip-10-0-1-120:~/wordpress# docker-compose images
Container      Repository      Tag      Image Id      Size
wordpress_database_1  mysql          latest   99afc808f15b  577.4 MB
wordpress_wordpress_1  wordpress     latest   efa186f6484e  665.8 MB
root@ip-10-0-1-120:~/wordpress# docker-compose ps
Name                Command              State      Ports
wordpress_database_1  docker-entrypoint.sh mysqld  Up        3306/tcp, 33060/tcp
wordpress_wordpress_1  docker-entrypoint.sh apache ...  Up        0.0.0.0:80->80/tcp, :::80->80/tcp
root@ip-10-0-1-120:~/wordpress# docker-compose logs wordpress
Attaching to wordpress_wordpress_1
wordpress_1 | WordPress not found in /var/www/html - copying now...
wordpress_1 | WARNING: /var/www/html is not empty! (copying anyhow)
wordpress_1 | Complete! WordPress has been successfully copied to /var/www/html
wordpress_1 | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied; copying 'wp-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message
wordpress_1 | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.3. Set the 'ServerName' directive globally to suppress this message
wordpress_1 | [Tue Sep 05 15:20:50.909478 2023] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.56 (Debian) PHP/8.0.30 configured -- resuming normal operations
wordpress_1 | [Tue Sep 05 15:20:50.909771 2023] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
root@ip-10-0-1-120:~/wordpress# docker-compose logs database
Attaching to wordpress_database_1
database_1 | 2023-09-05 15:20:48+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.1.0-1.el8 started.
database_1 | 2023-09-05 15:20:48+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
database_1 | 2023-09-05 15:20:48+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.1.0-1.el8 started.
database_1 | 2023-09-05 15:20:49+00:00 [Note] [Entrypoint]: Initializing database files

i-0783fd7e9a94783de (Wayfarer_web_0)
PublicIPs: 3.217.34.88 PrivateIPs: 10.0.1.120
```

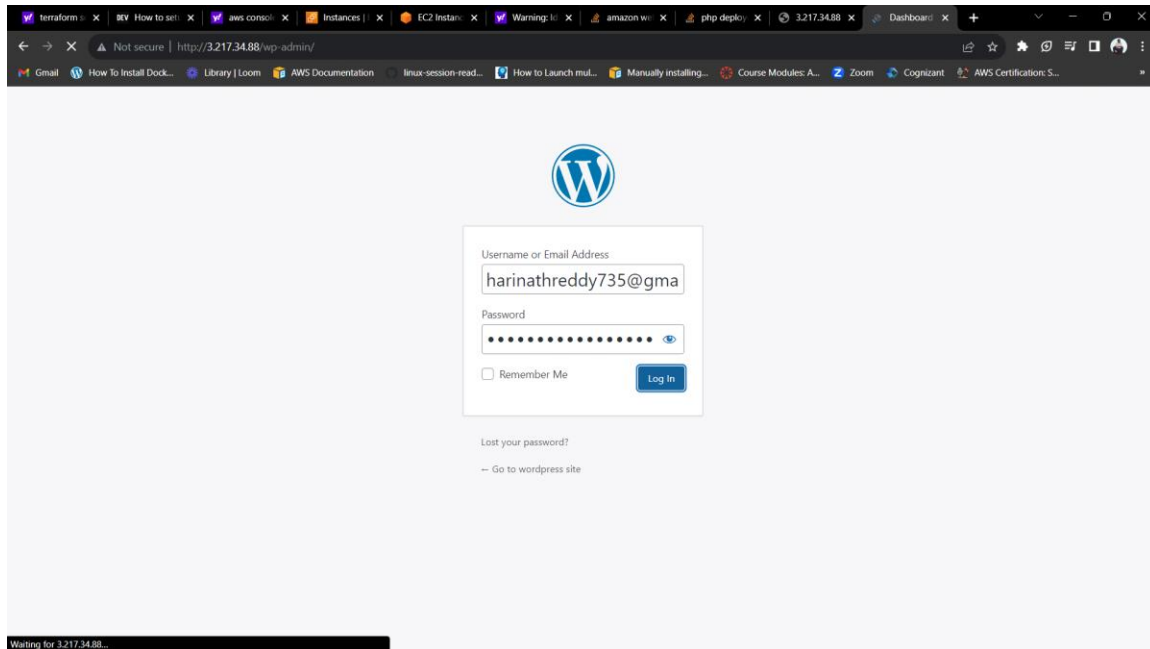
Getting Access to WordPress Website.



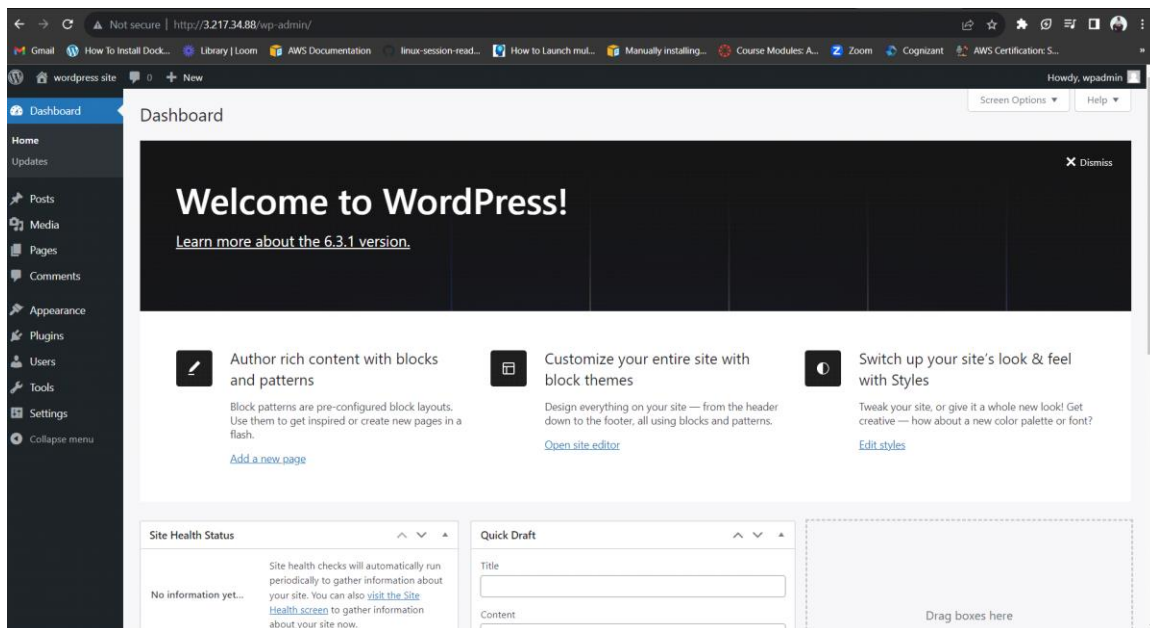
Click on the Continue button. You should see the WordPress site configuration screen



your website name, admin username, password, email, and click on Install WordPress, after that we should login by email & password.



WordPress admin username, and password and click on the Login button. You will be redirected to the WordPress dashboard it will show below screenshot



again go to the ec2 page and click on the wayfarer\_we\_0 instance. It will show the actions in that click on networking in that we click on connect rds database.

The screenshot shows the AWS Management Console for the 'us-east-1' region. The 'Instances' page is active, displaying a list of EC2 instances. The instance 'Wayfarer\_web\_0' (ID: i-0783fd7e9a94783de) is selected. The 'Actions' menu is open, showing various options for managing the instance. The instance details for 'Wayfarer\_web\_0' are visible, including its ID, state (Running), type (t2.micro), and various addresses.

Name	Instance ID	Instance state	Instance type	Status check	Actions
Terraform EC2	i-03d9854cf873fa078	Running	t2.micro	Attach network interface	
Wayfarer_web_0	i-0783fd7e9a94783de	Running	t2.micro	Detach network interface	

**Instance: i-0783fd7e9a94783de (Wayfarer\_web\_0)**

**Instance summary**

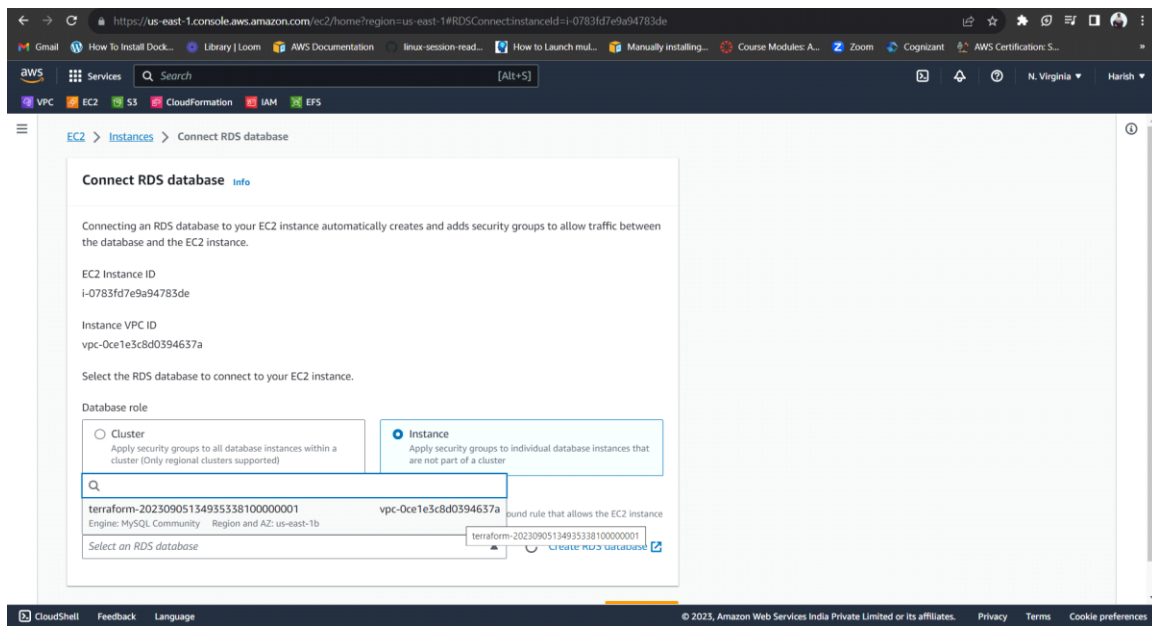
- Instance ID: i-0783fd7e9a94783de (Wayfarer\_web\_0)
- Public IPv4 address: 3.217.34.88 | [open address](#)
- Private IPv4 addresses: 10.0.1.120
- Public IPv4 DNS: ec2-3-217-34-88.compute-1.amazonaws.com | [open address](#)
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-10-0-1-120.ec2.internal
- IP name: ip-10-0-1-120.ec2.internal
- Answer private resource DNS name
- Instance type: t2.micro
- Elastic IP addresses

The screenshot shows the AWS Management Console for the 'us-east-1' region. The 'Amazon RDS' page is active, displaying the 'Databases' section. A single database instance is listed with the identifier 'terraform-20230905134935338100000001'. The instance is in an 'Available' state and is using the 'MySQL Community' engine.

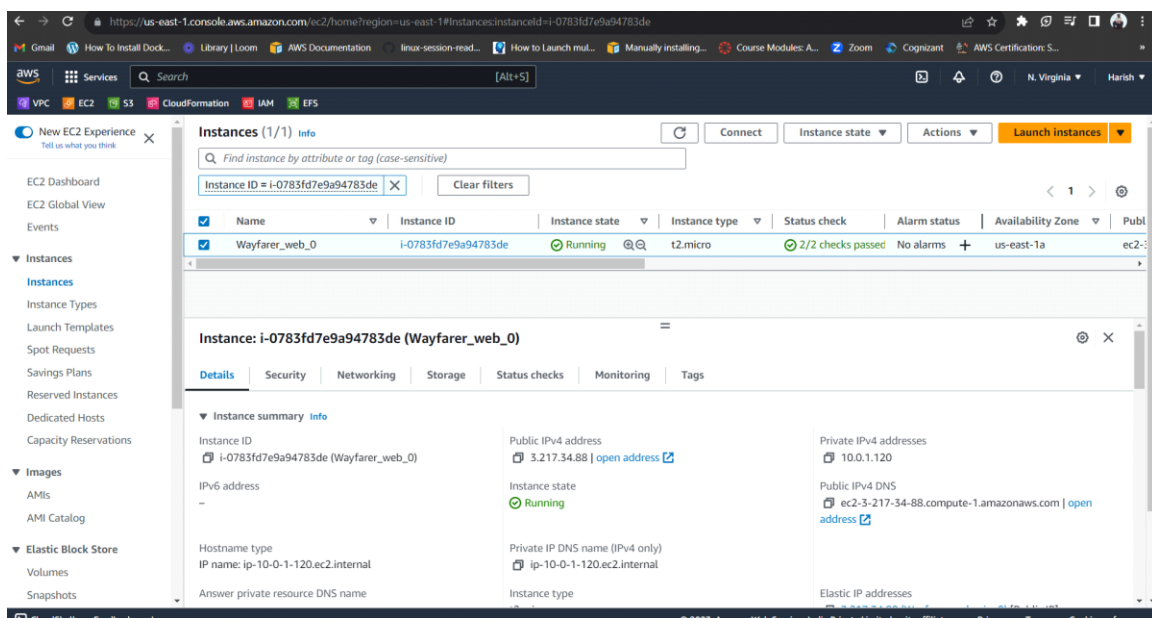
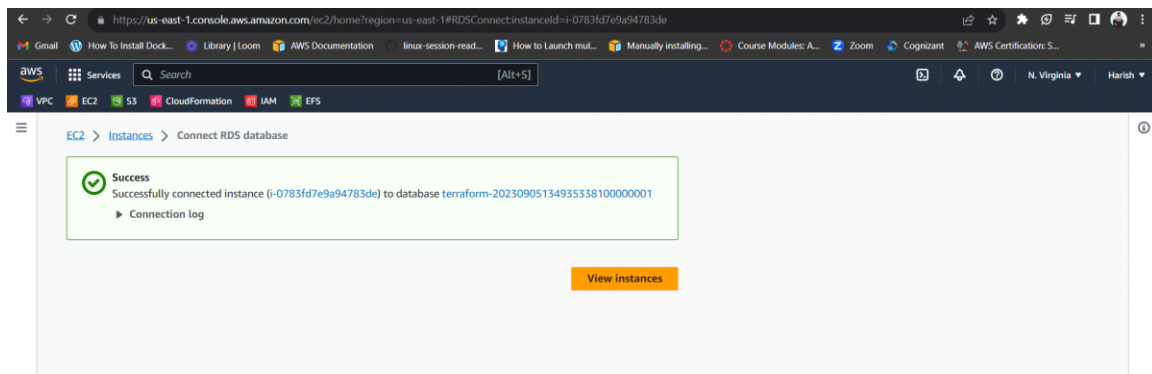
DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU
terraform-20230905134935338100000001	Available	Instance	MySQL Community	us-east-1b	db.t2.micro	4 Actions	4.0%

In that we add Database role as in instance it shows the terraform-20230905134935338100000001 click on it.





It will be connected from RDS to the EC2 instance.



IPv4

DB instance certificate expiration date  
⚠ August 22, 2024, 22:38 (UTC+05:30)

Security group rules (2)

Filter by Security group rules

< 1 > ⚙

Security group	Type	Rule
<a href="#">rds-ec2-1 (sg-0b3efbe7c652fb02e)</a>	EC2 Security Group - Inbound	sg-00a54c2f46143693a
<a href="#">Wayfarer_db_sg (sg-0ba6f1f7af81ea58c)</a>	EC2 Security Group - Inbound	sg-0cb41736e69c17af8

Replication (1)

Filter by Replication

< 1 > ⚙

DB identifier	Role	Region & AZ	Replication source	Replication state	Lag
<a href="#">terraform-20230905134935338100000001</a>	Instance	us-east-1b	-	-	-



