Deploying wordpress on AWS/Azure with RDS using Terraform, and Docker.

- Create a Terraform script to deploy a VM. The VM should be in public subnet.
- In the same VM Create a Dockerfile to deploy Apache webserver + PHP + Wordpress. Use COPY, ARG, RUN, ENTRYPOINT, CMD, WORKDIR in Dockerfile.
- Build the docker image, tag the image and push the docker image to Docker Hub and run the container.
- Deploy the RDS on AWS/Azure and it should be on private subnet.
- Connect your Wordpress container with RDS database.
- Setup the Wordpress and attach all the screenshot and code to Github.

Note- Please add the GitHub link while submitting the assignment.

—----------

Step 1: Setting Up the Local Environment

- Installed Terraform on your local machine.
- Installed the AWS CLI (Command Line Interface).
- Configured my AWS account using the aws configure command to set your access key, secret key, default region, and output format.

Step 2: Terraform Infrastructure as Code

- Wrote a Terraform script (main.tf) to define AWS resources for the project. The script included the following components:
    - Provider configuration
    - VPC (Virtual Private Cloud) declaration
    - Public subnet
    - Private subnet
    - Internet Gateway (IGW)
    - Network Address Translation (NAT) Gateway
    - Route tables
    - Security groups
    - EC2 instance (placed in the public subnet)
- After completing the Terraform script, ran the following Terraform commands:
    - terraform init: Initialised your working directory and prepared it for creating and managing the infrastructure.
    - terraform plan: Reviewed the execution plan to ensure it matched your intended changes.
    - terraform apply -auto-approve: Applied the changes to create the infrastructure. Used the -auto-approve flag for a non-interactive

deployment. This command deployed a virtual machine (EC2 instance) in the public subnet.

Step 3: Deploying Docker Container

- Wrote a Dockerfile on the deployed EC2 instance created using the Terraform script to customize the Docker image.
- Built the customized Docker image using the docker build command.
- Logged into your Docker Hub account using docker login.
- Pushed the Docker image to Docker Hub using the docker push command.
- Ran the Docker container on the EC2 instance in detached mode (-d) using the docker run command.

Step 4: Setting Up RDS (Relational Database Service)

- Set up an RDS instance using the AWS Management Console in a private subnet.

Step 5: Connecting the WordPress Application

- Connected the WordPress application to the Docker container hosting the RDS database, ensuring that the WordPress configuration was correctly set to use the RDS database endpoint.

Screenshots (screenshot and source code has been shared to github account)

screenshots for each of the following steps to accompany my documentation:

AWS CLI configuration.

```
diwakarrajanna@Diwakars-MacBook-Pro ~ % aws configure
AWS Access Key ID [None]: AKIA2LJG3WMFI45FAQGP
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Terraform script (main.tf) showing the defined resources.

Wrote it in a visual studio code

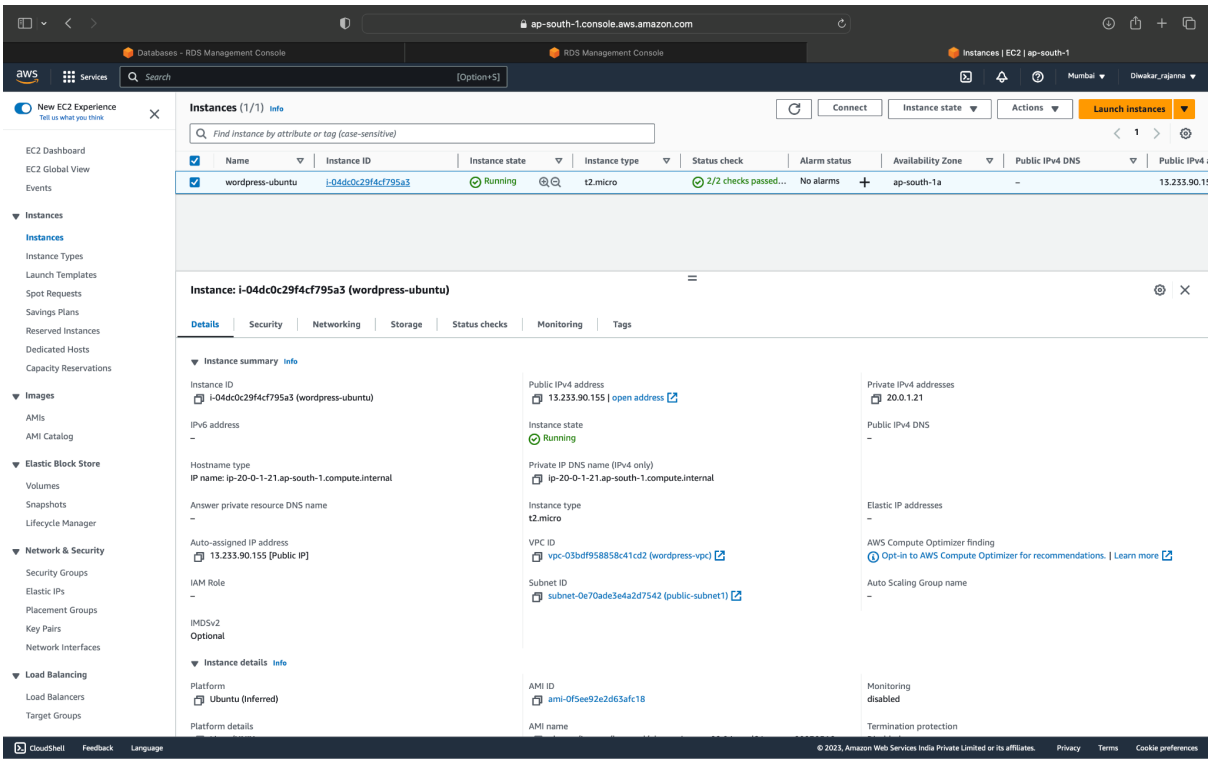
main.tf | DOCKERFILE | apache-config.conf

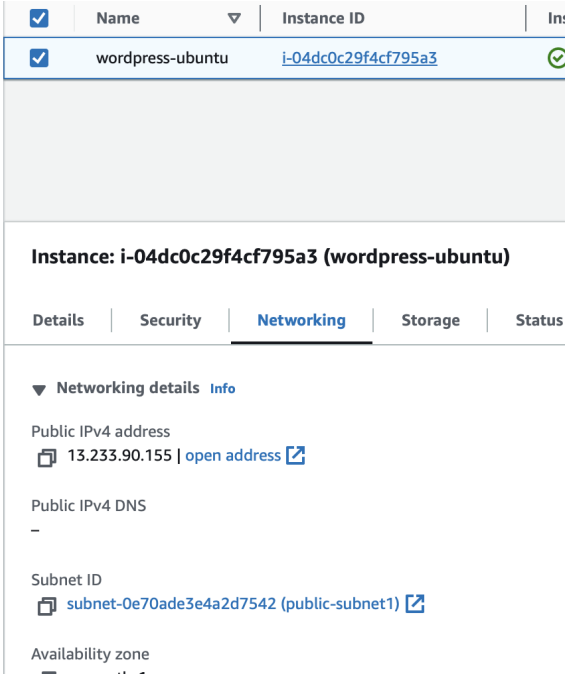
main.tf

```
provider "aws" {
  region = "ap-south-1"
}

# VPC declaration
resource "aws_vpc" "my-wordpress-vpc" {
  cidr_block = "20.0.0.0/16"
  tags = {
    Name = "wordpress-vpc"
  }
}

# Public subnet
resource "aws_subnet" "public1" {
  vpc_id            = aws_vpc.my-wordpress-vpc.id
  cidr_block        = "20.0.1.0/24"  # Adjusted CIDR block
  availability_zone = "ap-south-1a"

  tags = {
    Name = "public-subnet1"
  }
}

# Private subnets
resource "aws_subnet" "private1" {
  vpc_id            = aws_vpc.my-wordpress-vpc.id
  cidr_block        = "20.0.2.0/24"  # Adjusted CIDR block
  availability_zone = "ap-south-1a"

  tags = {
    Name = "private-subnet1"
  }
}

# Internet Gateway
resource "aws_internet_gateway" "igw" {
  vpc_id = aws_vpc.my-wordpress-vpc.id

  tags = {
    Name = "IGW"
  }
}

# Elastic IP
resource "aws_eip" "eip" {
  domain = "vpc"
}

# NAT Gateway
resource "aws_nat_gateway" "nat" {
  allocation_id = aws_eip.eip.id
  subnet_id     = aws_subnet.public1.id

  tags = {
```

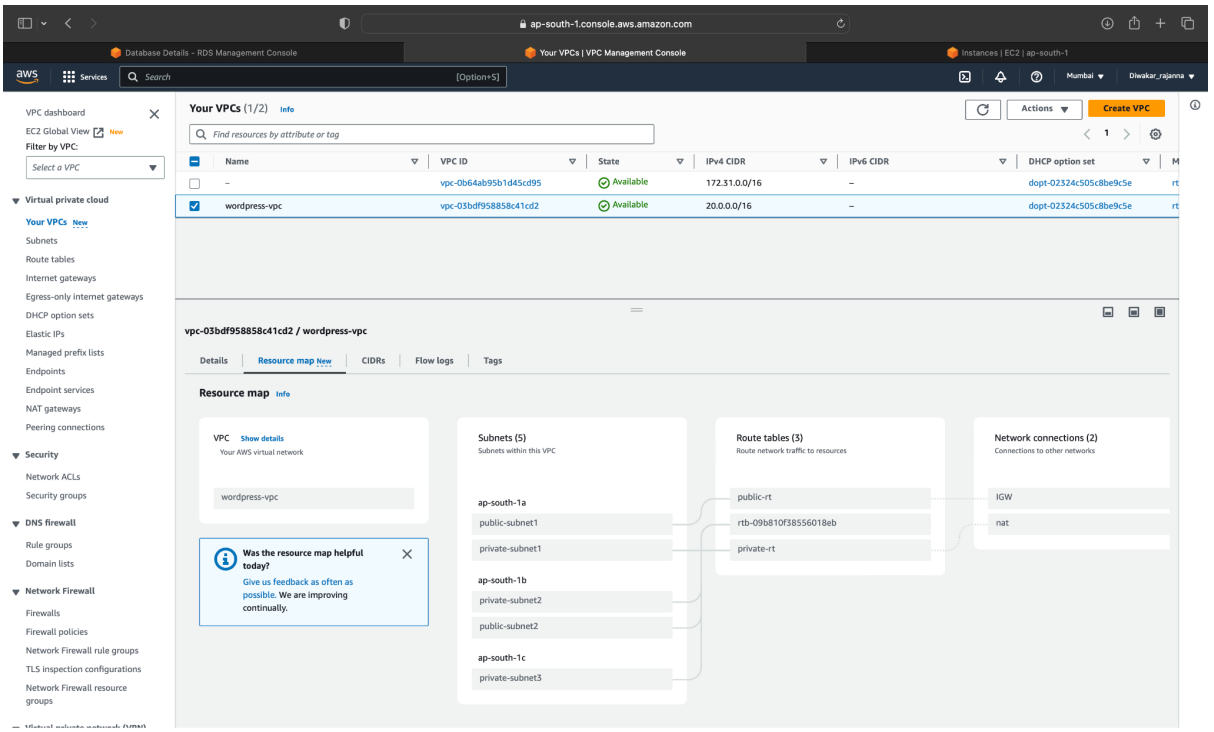
Terraform apply command output.

```
Plan: 12 to add, 0 to change, 0 to destroy.
aws_eip.eip: Creating...
aws_vpc.my-wordpress-vpc: Creating...
aws_eip.eip: Creation complete after 1s [id=eipalloc-0291f0ee53c3aba0d]
aws_vpc.my-wordpress-vpc: Creation complete after 2s [id=vpc-03bdf958858c41cd2]
aws_internet_gateway.igw: Creating...
aws_subnet.private1: Creating...
aws_subnet.public1: Creating...
aws_security_group.allow_ssh_http: Creating...
aws_internet_gateway.igw: Creation complete after 0s [id=igw-0f6c5c51e9dee4d1b]
aws_route_table.public: Creating...
aws_subnet.public1: Creation complete after 0s [id=subnet-0e70ade3e4a2d7542]
aws_subnet.private1: Creation complete after 0s [id=subnet-0934e913febbcf2eb]
aws_nat_gateway.nat: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-0e4c33662217a5565]
aws_route_table_association.public: Creating...
aws_route_table_association.public: Creation complete after 0s [id=rtbassoc-0ffc93e28039a8c86]
aws_security_group.allow_ssh_http: Creation complete after 2s [id=sg-0990eb5dd26abd8f8]
aws_instance.wordpress: Creating...
aws_nat_gateway.nat: Still creating... [10s elapsed]
aws_instance.wordpress: Still creating... [10s elapsed]
aws_nat_gateway.nat: Still creating... [20s elapsed]
aws_instance.wordpress: Still creating... [20s elapsed]
aws_nat_gateway.nat: Still creating... [30s elapsed]
aws_instance.wordpress: Still creating... [30s elapsed]
aws_instance.wordpress: Creation complete after 32s [id=i-04dc0c29f4cf795a3]
aws_nat_gateway.nat: Still creating... [40s elapsed]
aws_nat_gateway.nat: Still creating... [50s elapsed]
aws_nat_gateway.nat: Still creating... [1m0s elapsed]
aws_nat_gateway.nat: Still creating... [1m10s elapsed]
aws_nat_gateway.nat: Still creating... [1m20s elapsed]
aws_nat_gateway.nat: Still creating... [1m30s elapsed]
aws_nat_gateway.nat: Still creating... [1m40s elapsed]
aws_nat_gateway.nat: Creation complete after 1m45s [id=nat-05eee49fca7eec884]
aws_route_table.private: Creating...
aws_route_table.private: Creation complete after 2s [id=rtb-0fb3710dfdf48fa7d]
aws_route_table_association.private: Creating...
aws_route_table_association.private: Creation complete after 0s [id=rtbassoc-03f58d44204573f90]

Apply complete! Resources: 12 added, 0 changed, 0 destroyed.
```

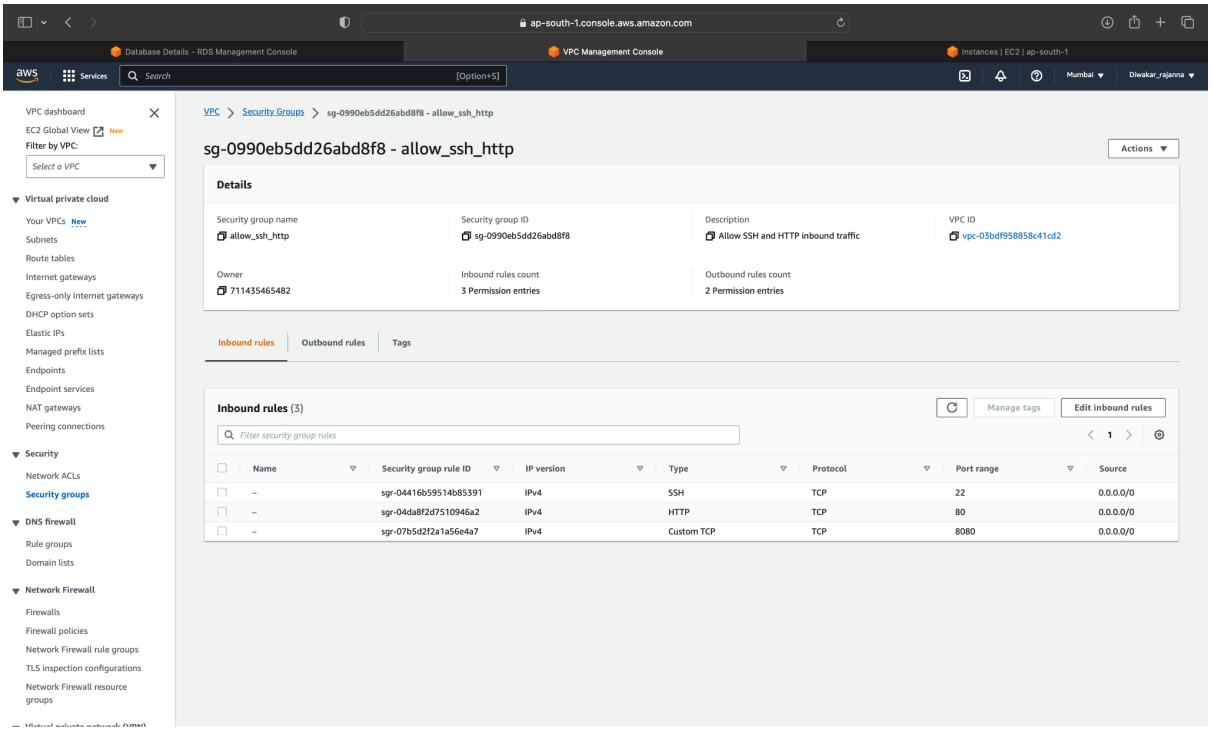



Name
Instance ID
wordpress-ubuntu
i-04dc0c29f4cf795a3
Instance: i-04dc0c29f4cf795a3 (wordpress-ubuntu)
Details
Security
Networking
Storage
Status
Networking details Info
Public IPv4 address
13.233.90.155 | open address
Public IPv4 DNS
–
Subnet ID
subnet-0e70ade3e4a2d7542 (public-subnet1)
Availability zone

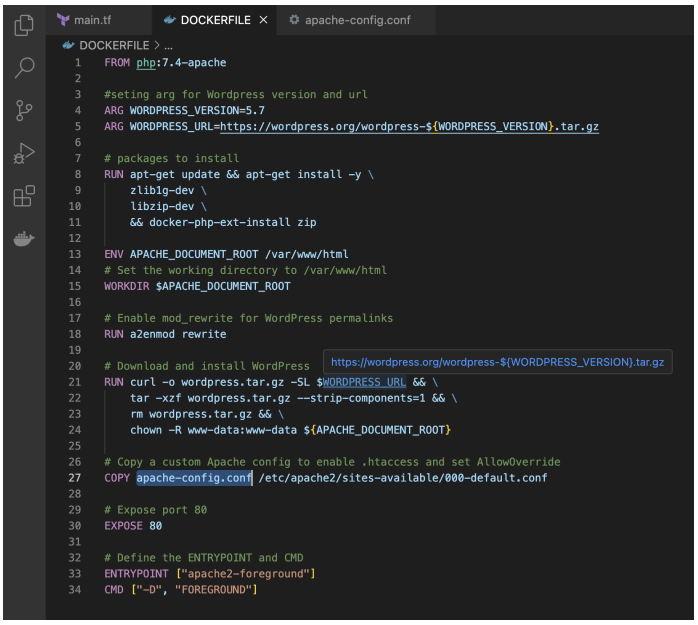
Vpc


ap-south-1.console.aws.amazon.com
Your VPCs | VPC Management Console
Your VPCs (1/2) Info
Actions
Create VPC
Name
VPC ID
State
IPv4 CIDR
IPv6 CIDR
DHCP option set
wordpress-vpc
vpc-0b64ab95b1d45cd95
vpc-03bdf958858c41cd2
Available
172.31.0.0/16
20.0.0.0/16
dopt-02324c505c8be9c5e
vpc-03bdf958858c41cd2 / wordpress-vpc
Details
Resource map New
CIDRs
Flow logs
Tags
Resource map Info
VPC
Subnets (5)
Route tables (3)
Network connections (2)
wordpress-vpc
ap-south-1a
public-subnet1
private-subnet1
ap-south-1b
private-subnet2
public-subnet2
ap-south-1c
private-subnet3
public-rt
rtb-09b810f38556018eb
private-rt
IGW
nat

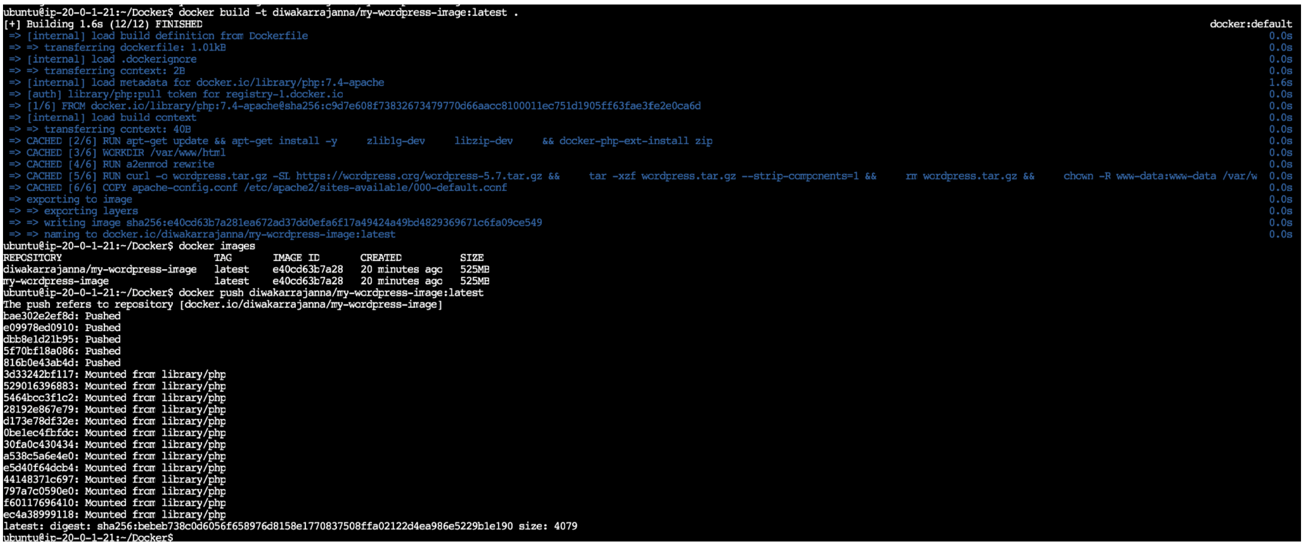
Security group



Dockerfile.

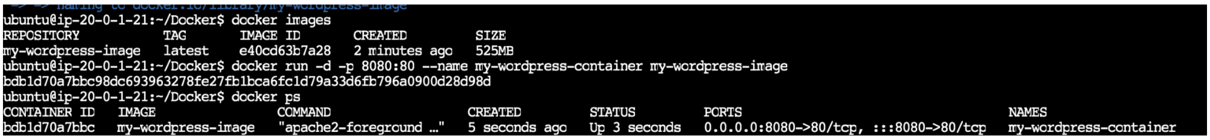


Docker image build and push commands.



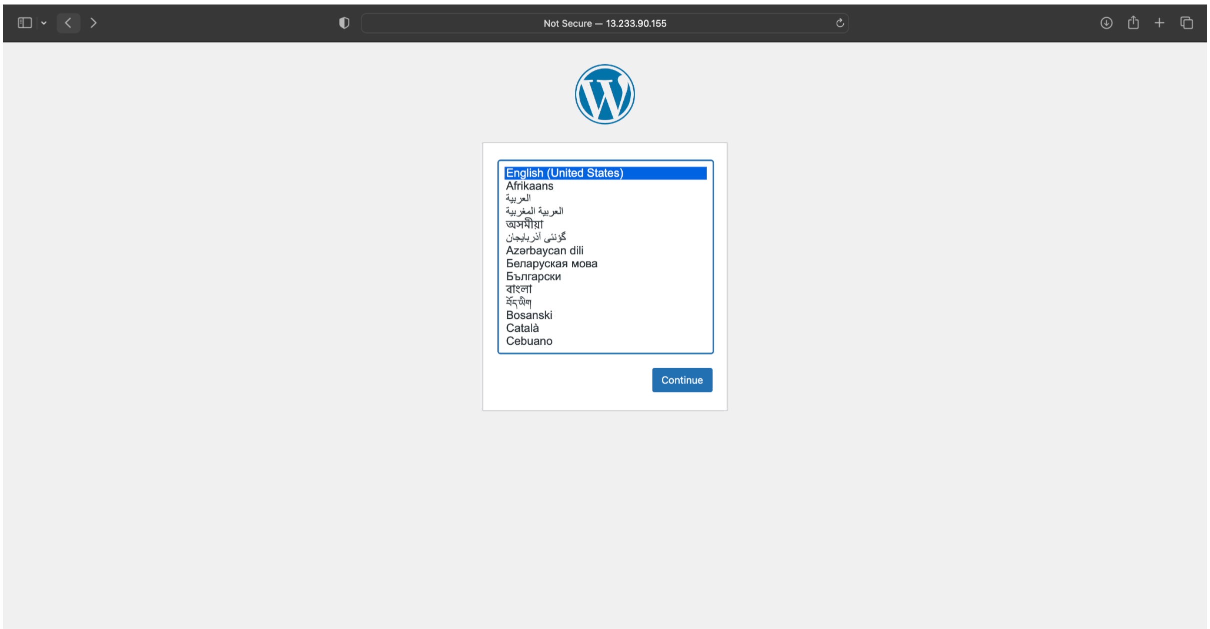
```
ubuntu@ip-20-0-1-21:~/Docker$ docker build -t diwakarrajanna/my-wordpress-image:latest .
[+] Building 1.6s (12/12) FINISHED                                                                                                                                                                                                     docker:default
 => [internal] load build definition from Dockerfile                                                                                                                                                                                              0.0s
 => => transferring dockerfile: 1.01kB                                                                                                                                                                                                            0.0s
 => [internal] load .dockerignore                                                                                                                                                                                                                 0.0s
 => => transferring context: 2B                                                                                                                                                                                                                   0.0s
 => [internal] load metadata for docker.io/library/php:7.4-apache                                                                                                                                                                                 1.6s
 => [auth] library/php:pull token for registry-1.docker.io                                                                                                                                                                                        0.0s
 => [1/6] FROM docker.io/library/php:7.4-apache@sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905ff63fae3fe2e0ca6d                                                                                                                          0.0s
 => [internal] load build context                                                                                                                                                                                                                 0.0s
 => => transferring context: 40B                                                                                                                                                                                                                  0.0s
 => CACHED [2/6] RUN apt-get update && apt-get install -y     zlib1g-dev     libzip-dev     && docker-php-ext-install zip                                                                                                                         0.0s
 => CACHED [3/6] WORKDIR /var/www/html                                                                                                                                                                                                            0.0s
 => CACHED [4/6] RUN a2enmod rewrite                                                                                                                                                                                                              0.0s
 => CACHED [5/6] RUN curl -o wordpress.tar.gz -SL https://wordpress.org/wordpress-5.7.tar.gz &&     tar -xzf wordpress.tar.gz --strip-components=1 &&     rm wordpress.tar.gz &&     chown -R www-data:www-data /var/w  0.0s
 => CACHED [6/6] COPY apache-config.conf /etc/apache2/sites-available/000-default.conf                                                                                                                                                            0.0s
 => exporting to image                                                                                                                                                                                                                            0.0s
 => => exporting layers                                                                                                                                                                                                                           0.0s
 => => writing image sha256:e40cd63b7a281ea672ad37dd0efa6f17a49424a49bd4829369671c6fa09ce549                                                                                                                                                      0.0s
 => => naming to docker.io/diwakarrajanna/my-wordpress-image:latest                                                                                                                                                                               0.0s
ubuntu@ip-20-0-1-21:~/Docker$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
diwakarrajanna/my-wordpress-image   latest    e40cd63b7a28   20 minutes ago   525MB
my-wordpress-image                  latest    e40cd63b7a28   20 minutes ago   525MB
ubuntu@ip-20-0-1-21:~/Docker$ docker push diwakarrajanna/my-wordpress-image:latest
The push refers to repository [docker.io/diwakarrajanna/my-wordpress-image]
bae302e2ef8d: Pushed
e09978ed0910: Pushed
dbb8e1d21b95: Pushed
5f70bf18a086: Pushed
816b0e43ab4d: Pushed
3d33242bf117: Mounted from library/php
529016396883: Mounted from library/php
5464bcc3f1c2: Mounted from library/php
28192e867e79: Mounted from library/php
d173e78df32e: Mounted from library/php
0be1ec4fbfdc: Mounted from library/php
30fa0c430434: Mounted from library/php
a538c5a6e4e0: Mounted from library/php
e5d40f64dcb4: Mounted from library/php
44148371c697: Mounted from library/php
797a7c0590e0: Mounted from library/php
f60117696410: Mounted from library/php
ec4a38999118: Mounted from library/php
latest: digest: sha256:bebeb738c0d6056f658976d8158e1770837508ffa02122d4ea986e5229b1e190 size: 4079
ubuntu@ip-20-0-1-21:~/Docker$
```

Docker container running on the EC2 instance.

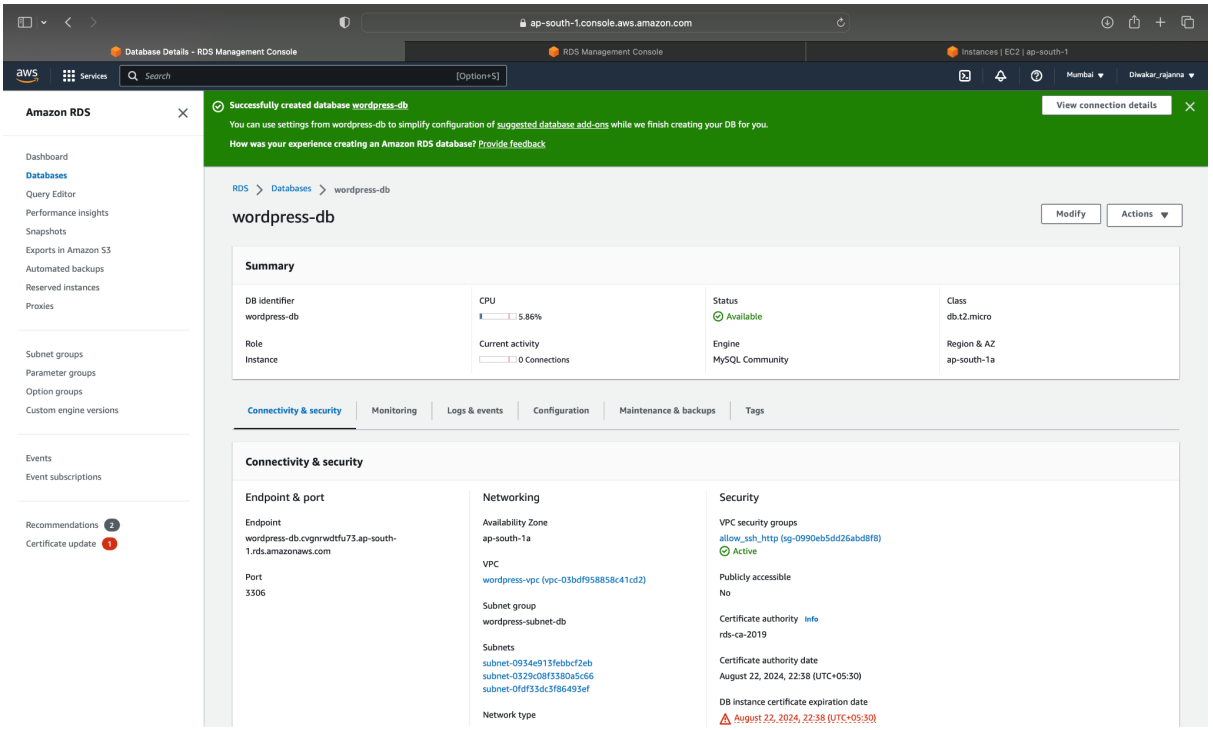


```
 => => naming to docker.io/library/my-wordpress-image
ubuntu@ip-20-0-1-21:~/Docker$ docker images
REPOSITORY           TAG       IMAGE ID       CREATED         SIZE
my-wordpress-image   latest    e40cd63b7a28   2 minutes ago   525MB
ubuntu@ip-20-0-1-21:~/Docker$ docker run -d -p 8080:80 --name my-wordpress-container my-wordpress-image
bdb1d70a7bbc98dc693963278fe27fb1bca6fc1d79a33d6fb796a0900d28d98d
ubuntu@ip-20-0-1-21:~/Docker$ docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED         STATUS         PORTS                                   NAMES
bdb1d70a7bbc   my-wordpress-image   "apache2-foreground …"   5 seconds ago   Up 3 seconds   0.0.0.0:8080->80/tcp, :::8080->80/tcp   my-wordpress-container
```

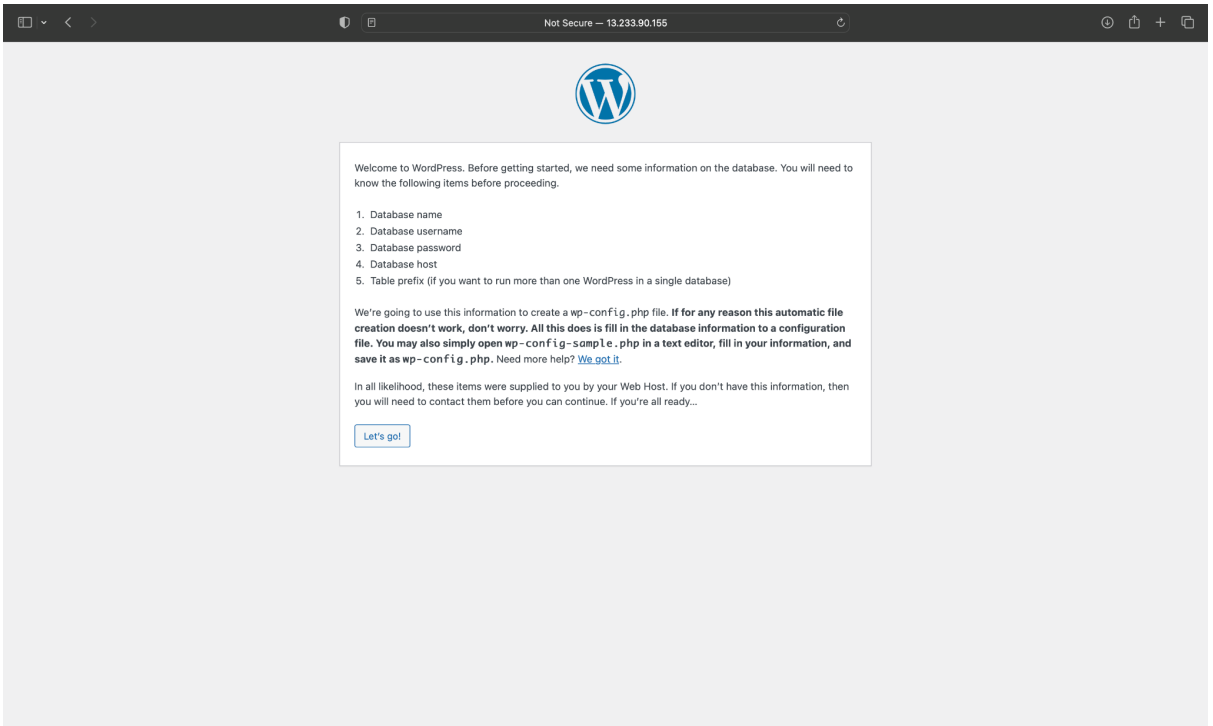
Wordpress webapplication



RDS setup in the AWS Management Console.



WordPress application successfully connected to the RDS database.



Not Secure — 13.233.90.155

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	wordpress	The name of the database you want to use with WordPress.
Username	username	Your database username.
Password	password	Your database password.
Database Host	localhost	You should be able to get this info from your web host, if `localhost` doesn't work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

Submit