Deploy Wordpress With Amazon RDS

Overview

- In this hands-on, we host a wordpress site using Amazon EC2 instance and then later we install and host the WordPress application .
- And Amazon RDS for MySQL database to store our WordPress data.
- WordPress is a flexible content management system for building blogs, e-commerce sites, personal websites, and more.
- WordPress is the easiest and most powerful blogging and website builder. In a blog, it will be our blog posts, comments, and images. If you raise an e-commerce site, it will be our product catalog and user accounts.
- All this content needs to be permanently stored somewhere. No matter what type of website we choose to assemble, there will be the need to store the content.
- So, in this hand-on, we use RDS for MySQL to host the database of WordPress site.

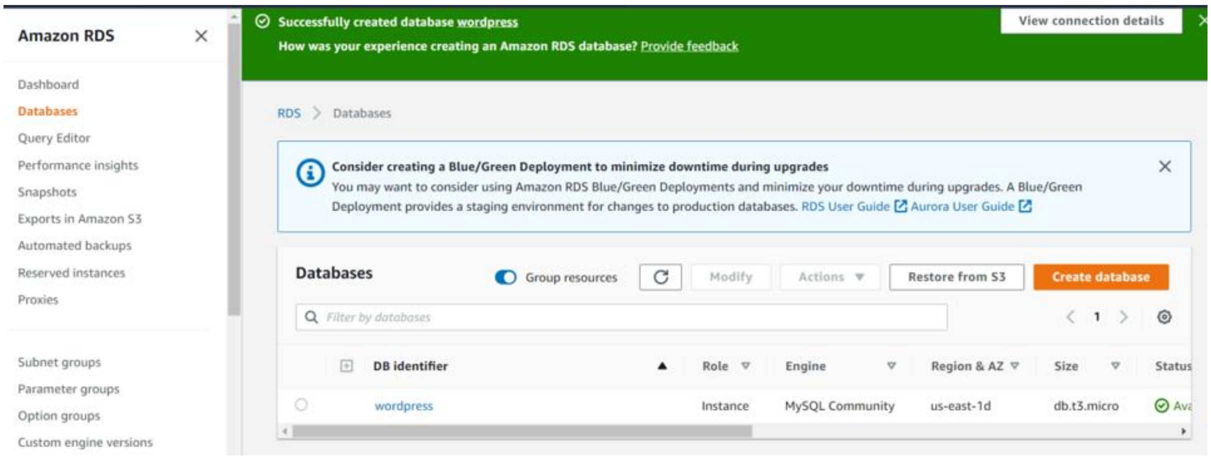
RDS

- Amazon Relational Database Service (RDS) is a managed SQL database service provided by Amazon Web Services (AWS). Amazon RDS supports an array of database engines to store and organize data.
- Amazon RDS facilitates the deployment and maintenance of relational databases in the cloud. Amazon RDS is not itself a database; it is a service used to manage relational databases.

MySQL

- MySQL is the most popular open source relational database and Amazon RDS makes it easier to set up, operate, and scale MySQL deployments in the cloud. With Amazon RDS, we can deploy scalable MySQL servers .
- WordPress stores data at the backend on a MySQL database server. Therefore, we need to setup a MySQL server using the AWS RDS service .
- And later provide the endpoint/connection string to the WordPress application to make it work.
- On the AWS management console, under services we will find RDS in that.

- The first step is to choose the database engine we want to use .Amazon RDS supports six different engines. Wordpress uses the mysql engine.
- In the settings section, I configured the DB instance identifier as WordPress and configured credentials settings, master username as admin and master password for login purposes, and in the instance configuration,
- I chose db.t3.micro and in storage type, I chose general purpose ssd and allocated storage 20 . And configured connectivity and network settings as default and created Amazon RDS Database.



Successfully created database wordpress

How was your experience creating an Amazon RDS database? Provide feedback

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. RDS User Guide Aurora User Guide

Databases

DB identifier	Role	Engine	Region & AZ	Size	Status
wordpress	Instance	MySQL Community	us-east-1d	db.t3.micro	Ava

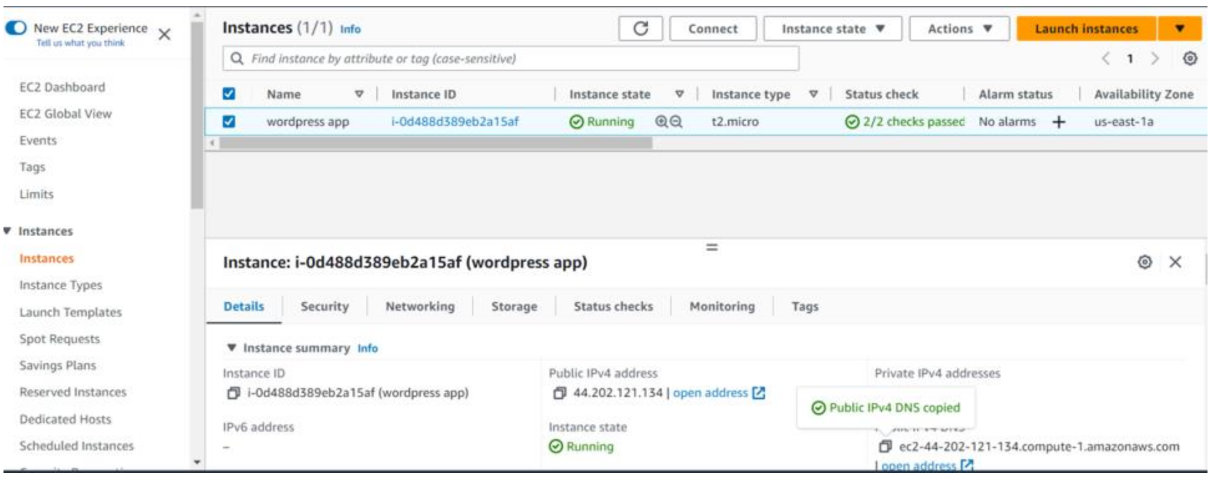
Creating an EC2 instance

We created an Amazon EC2 instance to run our wordpress site

Amazon EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web-based service that allows people to run application programs in AWS and Amazon EC2 provides highly configurable server instances on-demand. For an EC2 instance, we run a WordPress site that will be accessible by users anywhere.
- To create an Amazon EC2 instance on the AWS Management console and choose the launch instance.
- Configured the instance name as a wordpress app and chose Amazon Linux 2 AMI(HVM) as machine image and selected t2 micro for instance type and created a new key pair for the Linux instance to ssh into the Linux instance and configure the security group.

- And added new inbound rules in security to allow ssh traffic from my Ip and allowed HTTP traffic in the security group and launched EC2 instance.



New EC2 Experience
Tell us what you think

EC2 Dashboard
EC2 Global View
Events
Tags
Limits
▼ Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances

Instances (1/1) Info

Connect | Instance state ▼ | Actions ▼ | Launch instances ▼

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
✓	wordpress app	i-0d488d389eb2a15af	Running	t2.micro	2/2 checks passed	No alarms +	us-east-1a

Instance: i-0d488d389eb2a15af (wordpress app)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID
i-0d488d389eb2a15af (wordpress app)

Public IPv4 address
44.202.121.134 | open address

Private IPv4 addresses

Public IPv4 DNS copied

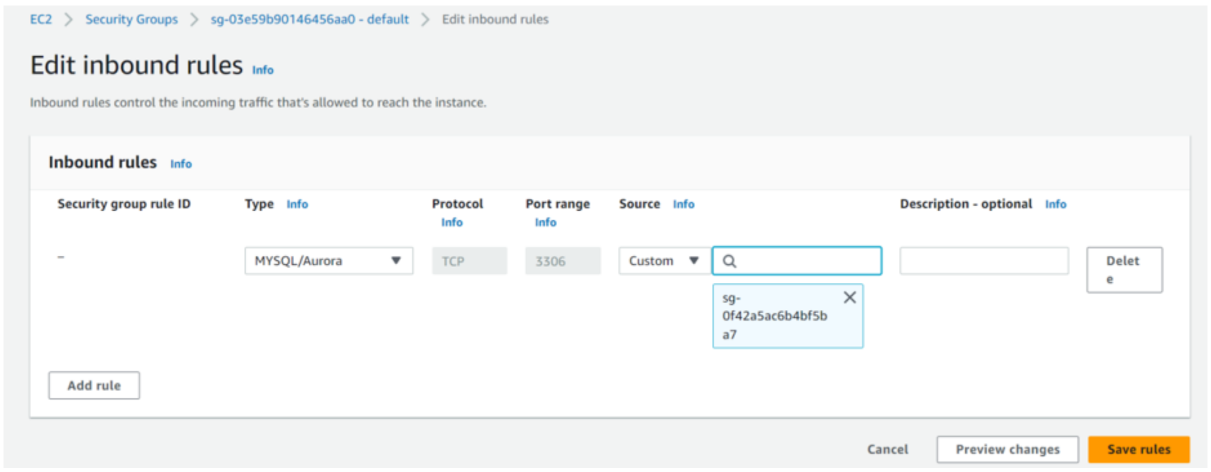
IPv6 address
–

Instance state
Running

ec2-44-202-121-134.compute-1.amazonaws.com
| open address

Configuring Amazon RDS Database

- In this, we configure the Amazon RDS database to allow access to specific entities.
- In the Amazon RDS database instance, in that instance, the security group we had to configure it to allow network access from the EC2 instance.
- In that security group type, I chose MYSQL/Aurora and, in source, I selected the security group that I created for an EC2 instance



EC2 > Security Groups > sg-03e59b90146456aa0 - default > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
–	MYSQL/Aurora	TCP	3306	Custom	sg-0f42a5ac6b4bf5ba7		Delete

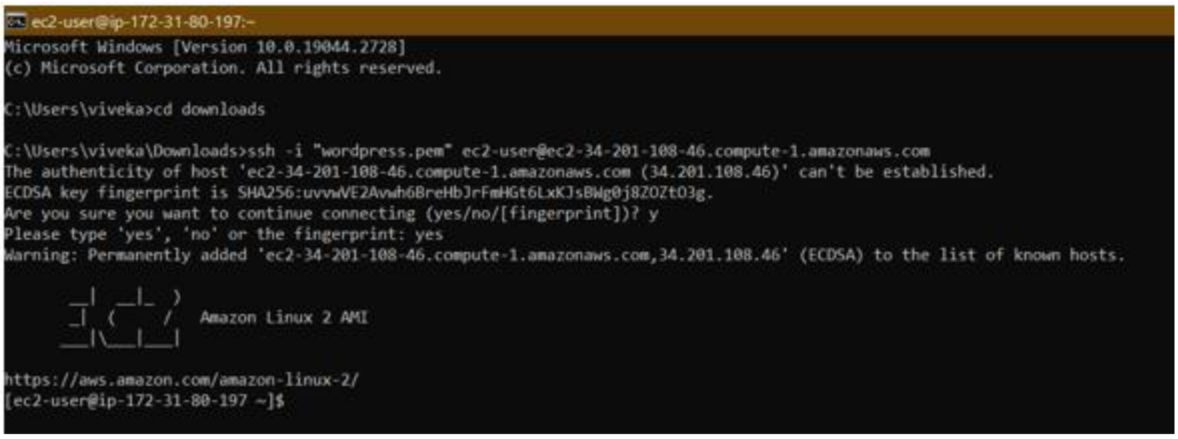
Add rule

Cancel | Preview changes | Save rules

SSH into our EC2 instance

- Now ec2 instance has access to our amazon RDS database

- Using an ssh key I connected my ec2 instance in my local window terminal.

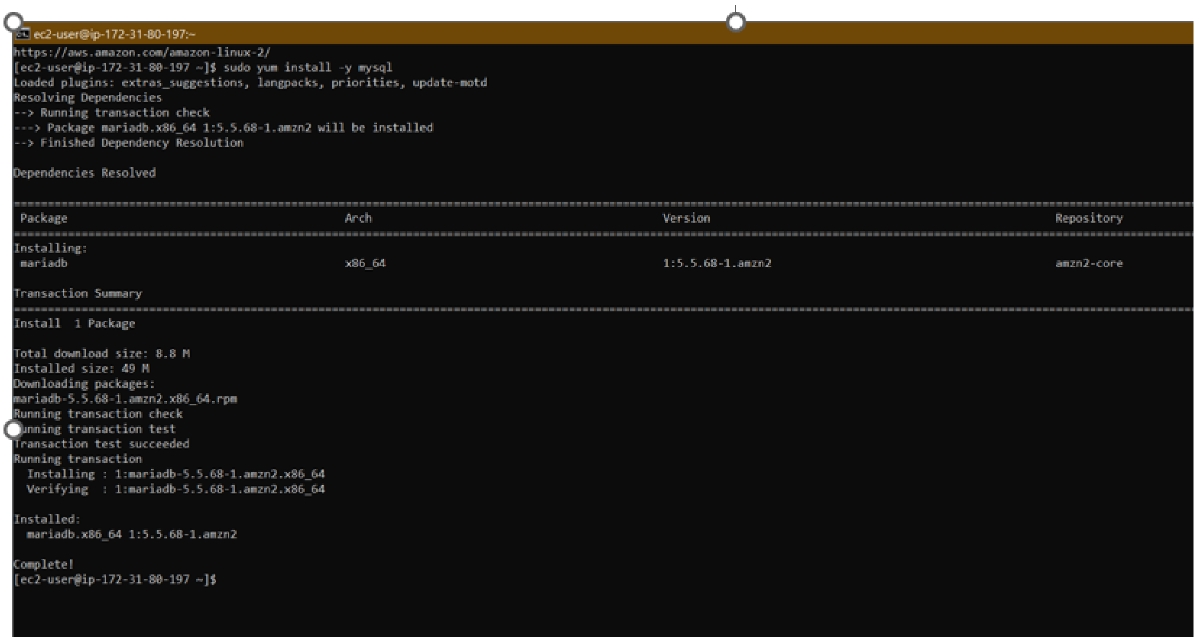


```
ec2-user@ip-172-31-80-197:~
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\viveka>cd downloads

C:\Users\viveka\Downloads>ssh -i "wordpress.pem" ec2-user@ec2-34-201-108-46.compute-1.amazonaws.com
The authenticity of host 'ec2-34-201-108-46.compute-1.amazonaws.com (34.201.108.46)' can't be established.
ECDSA key fingerprint is SHA256:uvvwVE2Avwh6BreHbJrFmHGt6LxKJsBWg0j8Z0Zt03g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added 'ec2-34-201-108-46.compute-1.amazonaws.com,34.201.108.46' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-80-197 ~]$
```

Create Database user

- And in my window terminal to interact with the database, I installed a MySQL client using the following cmd

sudo yum install -y mysql



```
ec2-user@ip-172-31-80-197:~
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-80-197 ~]$ sudo yum install -y mysql
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package                        Arch                        Version                        Repository
================================================================================================================
Installing:
 mariadb                        x86_64                      1:5.5.68-1.amzn2               amzn2-core

Transaction Summary
================================================================================================================
Install  1 Package

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

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

Complete!
[ec2-user@ip-172-31-80-197 ~]$
```

- Next to set an environment variable for MySQL host with hostname of RDS instance using this cmd

export MYSQL_HOST=<your-endpoint>

- And configured user name and password in my terminal which I created in earlier step while creating Amazon RDS

mysql --user=<user> --password=<password> wordpress

and gave the user permission to access the MySQL Database using the following cmd

CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';

GRANT ALL PRIVILEGES ON wordpress.* TO wordpress;

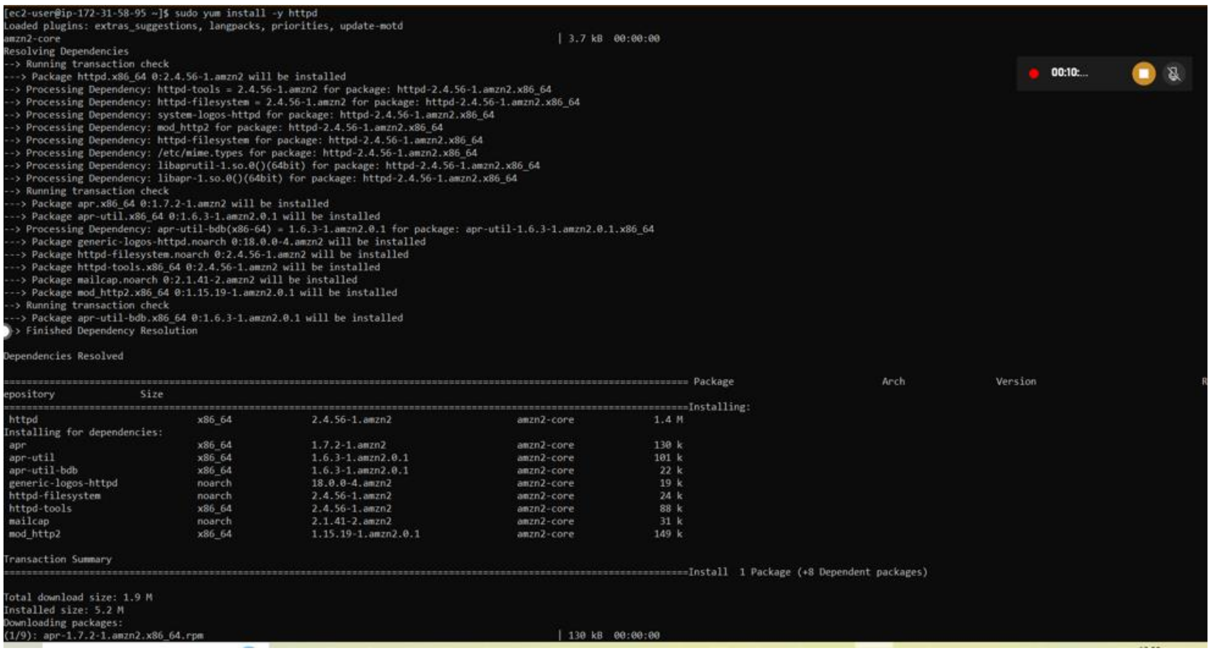
FLUSH PRIVILEGES;

Exit

Configuring Wordpress On EC2

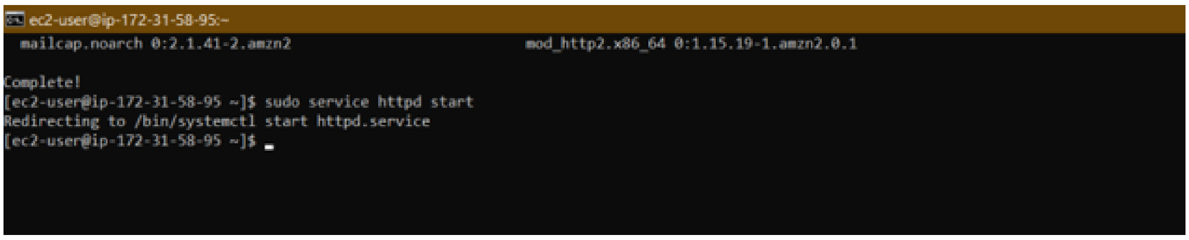
- We will install the wordpress application and dependencies on the EC2 instance.
- To run wordpress we need to run web server on EC2 instance
- So we install apache on our EC2 instance using this following command in our terminal

sudo yum install -y httpd



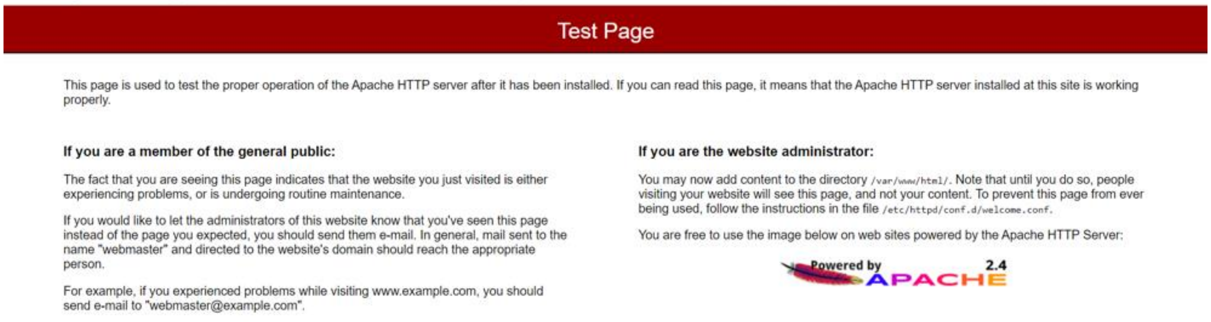
```
[ec2-user@ip-172-31-58-95 ~]$ sudo yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.4.56-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.56-1.amzn2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.56-1.amzn2.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.7.2-1.amzn2 will be installed
---> Package apr-util.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.3-1.amzn2.0.1 for package: apr-util-1.6.3-1.amzn2.0.1.x86_64
---> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
---> Package httpd-filesystem.noarch 0:2.4.56-1.amzn2 will be installed
---> Package httpd-tools.x86_64 0:2.4.56-1.amzn2 will be installed
---> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
---> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
---> Package apr-util-bdb.x86_64 0:1.6.3-1.amzn2.0.1 will be installed
 -> Finished Dependency Resolution

Dependencies Resolved

========================================================================================================================= Package                    Arch             Version                     R
epository              Size
=========================================================================================================================Installing:
 httpd                          x86_64             2.4.56-1.amzn2                  amzn2-core            1.4 M
Installing for dependencies:
 apr                            x86_64             1.7.2-1.amzn2                   amzn2-core            130 k
 apr-util                       x86_64             1.6.3-1.amzn2.0.1               amzn2-core            101 k
 apr-util-bdb                   x86_64             1.6.3-1.amzn2.0.1               amzn2-core             22 k
 generic-logos-httpd            noarch             18.0.0-4.amzn2                  amzn2-core             19 k
 httpd-filesystem               noarch             2.4.56-1.amzn2                  amzn2-core             24 k
 httpd-tools                    x86_64             2.4.56-1.amzn2                  amzn2-core             88 k
 mailcap                        noarch             2.1.41-2.amzn2                  amzn2-core             31 k
 mod_http2                      x86_64             1.15.19-1.amzn2.0.1             amzn2-core            149 k

Transaction Summary
=========================================================================================================================Install  1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Downloading packages:
(1/9): apr-1.7.2-1.amzn2.x86_64.rpm                                      | 130 kB  00:00:00
```

- We can see the following packages being installed
- After this to start the apache web server using this cmd
- sudo service httpd start



```
ec2-user@ip-172-31-58-95:~
  mailcap.noarch 0:2.1.41-2.amzn2                          mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-58-95 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[ec2-user@ip-172-31-58-95 ~]$ _
```

- To see the apache web server is working or not by visiting the public DNS of our EC2 instance and copying it and paste it into our web browser.



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:

- And next we will download the wordpress software and set up the configuration
- First we download and uncompress the software by running the following cmd

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

and if we run ls to view the content of our directory we will see tar file and a directory called wordpress with uncompressed contents $ ls

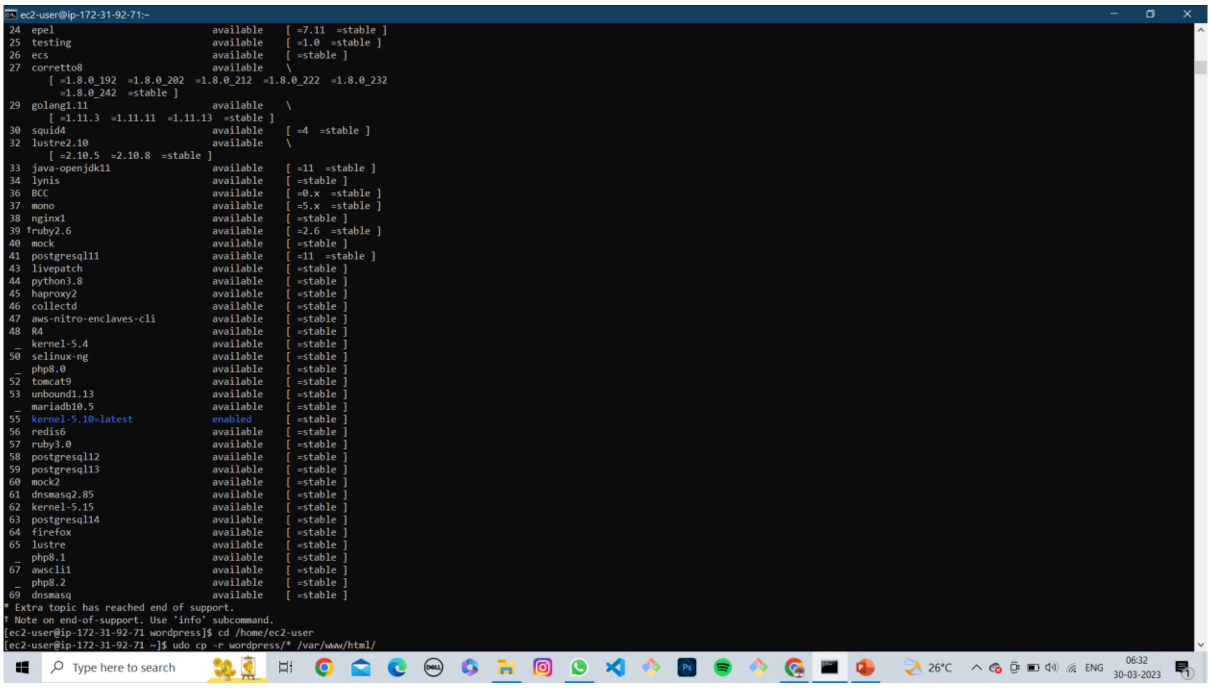
latest.tar.gz wordpress

And next we change the directory to the wordpress directory and create a copy of the default config using

cd wordpress

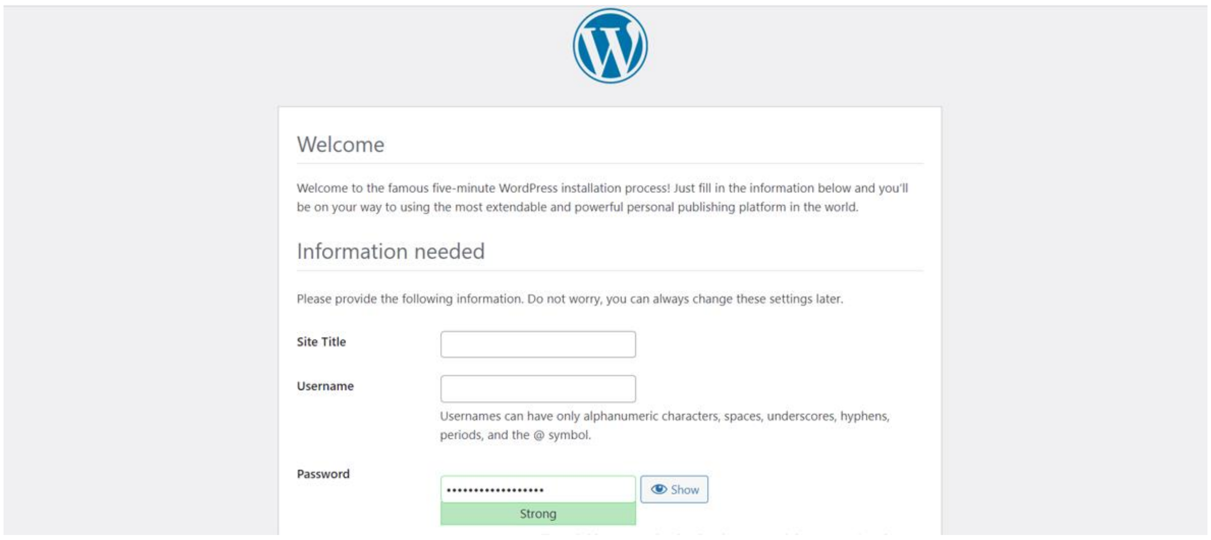
cp wp-config-sample.php wp-config.php

- And finally we edit the database configuration name , user, password, host in nano editor
- Next we deploy our wordpress first install the application dependencies we need for wordpress and we change the proper directory to copy our wordpress application files



```
ec2-user@ip-172-31-92-71:~
24  epel                       available    [ =7.11  =stable ]
25  testing                    available    [ =1.0  =stable ]
26  ecs                        available    [ =stable ]
27  corretto8                  available    \
      [ =1.8.0_192  =1.8.0_202  =1.8.0_212  =1.8.0_222  =1.8.0_232
        =1.8.0_242  =stable ]
29  golang1.11                 available    \
      [ =1.11.3  =1.11.11  =1.11.13  =stable ]
30  squid4                     available    [ =4  =stable ]
32  lustre2.10                 available    \
      [ =2.10.5  =2.10.8  =stable ]
33  java-openjdk11             available    [ =11  =stable ]
34  lynis                      available    [ =stable ]
36  BCC                        available    [ =0.x  =stable ]
37  mono                       available    [ =5.x  =stable ]
38  nginx1                     available    [ =stable ]
39 †ruby2.6                    available    [ =2.6  =stable ]
40  mock                       available    [ =stable ]
41  postgresql11               available    [ =11  =stable ]
43  livepatch                  available    [ =stable ]
44  python3.8                  available    [ =stable ]
45  haproxy2                   available    [ =stable ]
46  collectd                   available    [ =stable ]
47  aws-nitro-enclaves-cli     available    [ =stable ]
48  R4                         available    [ =stable ]
 _  kernel-5.4                 available    [ =stable ]
50  selinux-ng                 available    [ =stable ]
 _  php8.0                     available    [ =stable ]
52  tomcat9                    available    [ =stable ]
53  unbound1.13                available    [ =stable ]
 _  mariadb10.5                available    [ =stable ]
55  kernel-5.10=latest         enabled      [ =stable ]
56  redis6                     available    [ =stable ]
57  ruby3.0                    available    [ =stable ]
58  postgresql12               available    [ =stable ]
59  postgresql13               available    [ =stable ]
60  mock2                      available    [ =stable ]
61  dnsmasq2.85                available    [ =stable ]
62  kernel-5.15                available    [ =stable ]
63  postgresql14               available    [ =stable ]
64  firefox                    available    [ =stable ]
65  lustre                     available    [ =stable ]
 _  php8.1                     available    [ =stable ]
67  awscli1                    available    [ =stable ]
 _  php8.2                     available    [ =stable ]
69  dnsmasq                    available    [ =stable ]
* Extra topic has reached end of support.
† Note on end-of-support. Use 'info' subcommand.
[ec2-user@ip-172-31-92-71 wordpress]$ cd /home/ec2-user
[ec2-user@ip-172-31-92-71 ~]$ udo cp -r wordpress/* /var/www/html/
```

- Finally we restart the apache web server to pick up the changes
- We can see the wordpress welcome page



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

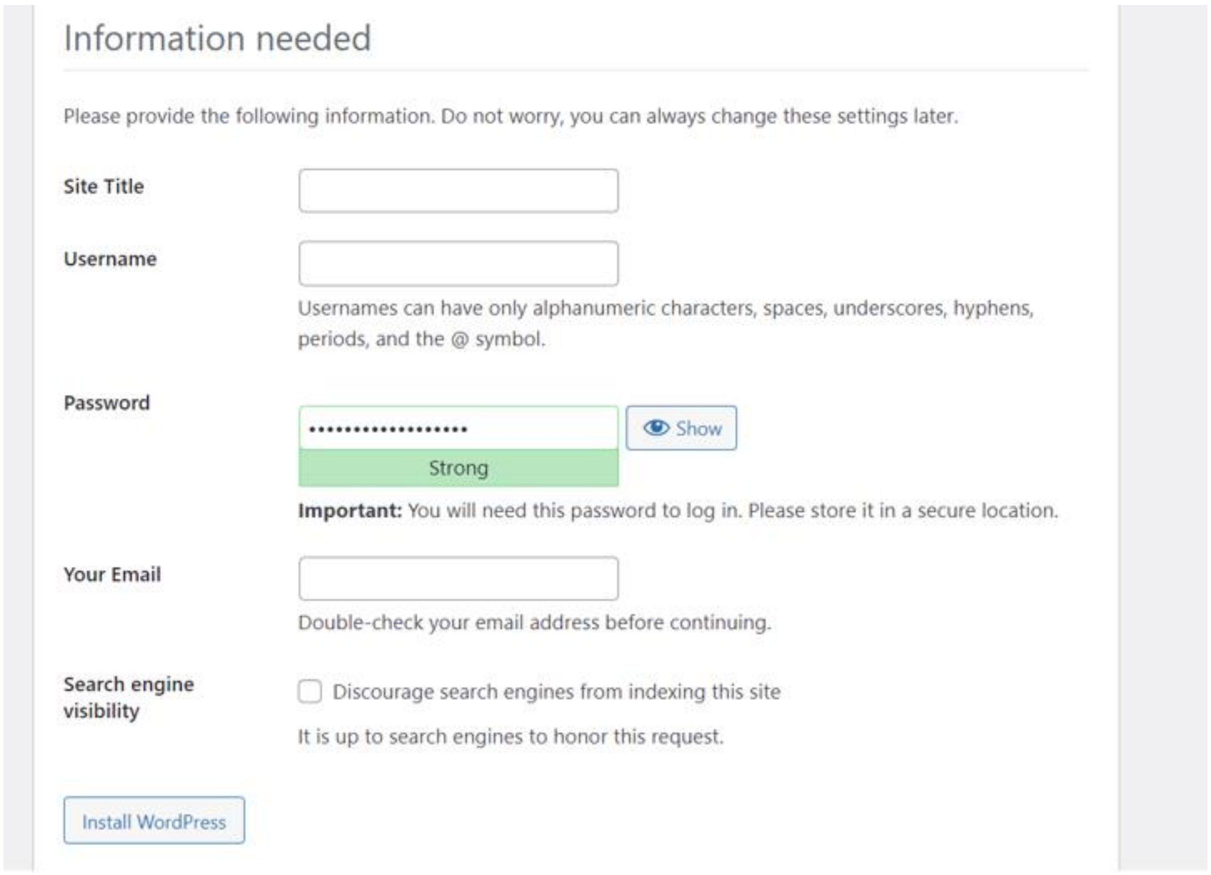
Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

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

Password ••••••••••••••••••



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	•••••••••••••••••• Show Strong **Important:** You will need this password to log in. Please store it in a secure location.
Your Email	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.