Task Report for Internship: Using Monolithic and Microservices Architectures to Deploy WordPress

Overview

The procedures used to install WordPress utilizing are described in this report.

Microservices and monolithic architectures are two distinct styles.

The work include installing WordPress and MySQL, configuring EC2 instances on AWS, and creating a welcome page that serves as the homepage.

Overview of Architectures

Architecture in Monoliths

All of the application's components operate on a single server in a monolithic design. We set up WordPress and MySQL on a single EC2 machine for this operation.

Architecture of Microservices

A microservices design divides the application's many components into independent services. We set up WordPress on one EC2 instance and MySQL on another for this operation.

EC2 Instances Configuration

Instance Details

 Instance Type: t3-micro

 AMI: Ubuntu 20.04 LTS (ami-0a91cd140a1fc148a)

Security Groups

 Allow HTTP (port 80) for WordPress.

 Allow MySQL (port 3306) only from the WordPress

instance for enhanced security.

 Allow SSH (port 22) for administration purposes.

Deployment Steps

Monolithic Architecture

1. Launch EC2 Instance

 Create a t3-micro EC2 instance with the specified

AMI (Ubuntu 20.04 LTS).

 Assign a security group allowing HTTP, MySQL, and

SSH access.

2. Install and Configure Apache

->sudo apt update

->sudo apt install apache2

->sudo systemctl start apache2

->sudo systemctl enable apache2

3. Install and Configure MySQL

->sudo apt install mysql-server

->sudo mysql_secure_installaƟon

->sudo mysql

->CREATE DATABASE wordpress;

->CREATE USER 'wp_user'@'localhost' IDENTIFIED BY

'password';

->GRANT ALL PRIVILEGES ON wordpress.* TO

'wp_user'@'localhost';

>FLUSH PRIVILEGES;

4. Install PHP and WordPress

->sudo apt install php libapache2-mod-php php-mysql

->cd /tmp

->wget hΣps://wordpress.org/latest.tar.gz

->tar -xzvf latest.tar.gz

->sudo mv wordpress /var/www/html/

->sudo chown -R www-data:www-data

/var/www/html/wordpress

->sudo chmod -R 755 /var/www/html/wordpress

->sudo systemctl restart apache2

5. Configure WordPress

 Navigate to hΣp://your_instance_ip/wordpress in a

web browser.

 Complete the WordPress setup wizard, connecƟng

it to the MySQL database created earlier.

 Create a welcome page and set it as the homepage.

Microservices Architecture

1. Launch EC2 Instances

 Create two t3-micro EC2 instances with the specified AMI (Ubuntu 20.04 LTS).

 One instance for WordPress (WP-Instance) and another for MySQL (DB-Instance).

 Assign appropriate security groups.

2. Configure MySQL on DB-Instance

->sudo apt update

->sudo apt install mysql-server

->sudo mysql_secure_installaϴon

-> sudo mysql

->CREATE DATABASE wordpress;

->CREATE USER 'wp_user'@'%' IDENTIFIED BY 'password';

->GRANT ALL PRIVILEGES ON wordpress.* TO 'wp_user'@'%';

-> FLUSH PRIVILEGES;

->sudo ufw allow 3306

3. Install and Configure Apache and WordPress on WPInstance

->sudo apt update

->sudo apt install apache2 php libapache2-mod-php phpmysql

->cd /tmp

>wget hΣps://wordpress.org/latest.tar.gz

->tar -xzvf latest.tar.gz

->sudo mv wordpress /var/www/html/

->sudo chown -R www-data:www-data /var/www/html/wordpress

->sudo chmod -R 755 /var/www/html/wordpress

->sudo systemctl start apache2

->sudo systemctl enable apache2

4. Configure WordPress to Use Remote MySQL Database

 -Update the WordPress configuraϴon to point to the DB-Instance.

-> // In /var/www/html/wordpress/wp-config.php

->define('DB_NAME', 'wordpress');

->define('DB_USER', 'wp_user');

->define('DB_PASSWORD', 'password');

->define('DB_HOST', 'DB-Instance_IP');

5. Complete WordPress Setup

 Navigate to ` hΣp://WP-Instance_IP/wordpress` in a

web browser.

 Complete the WordPress setup wizard, connecting it to

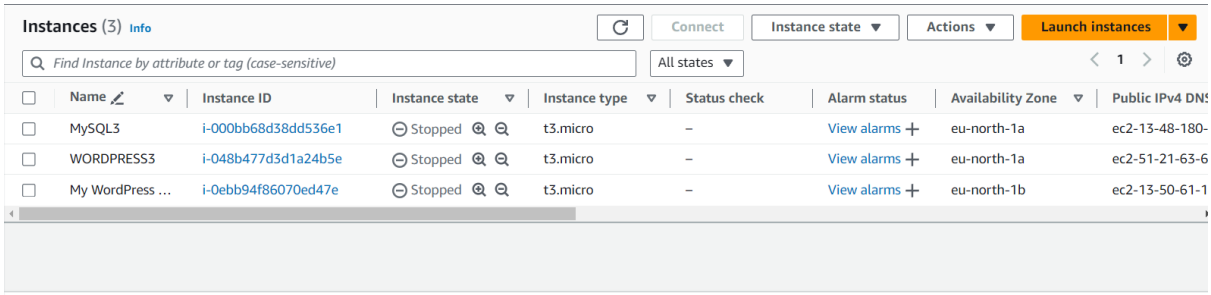
the remote MySQL database.

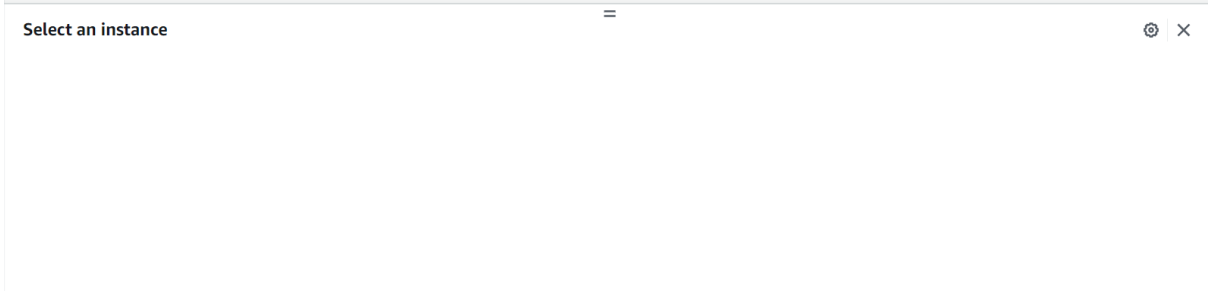
 Create a welcome page and set it as the homepage

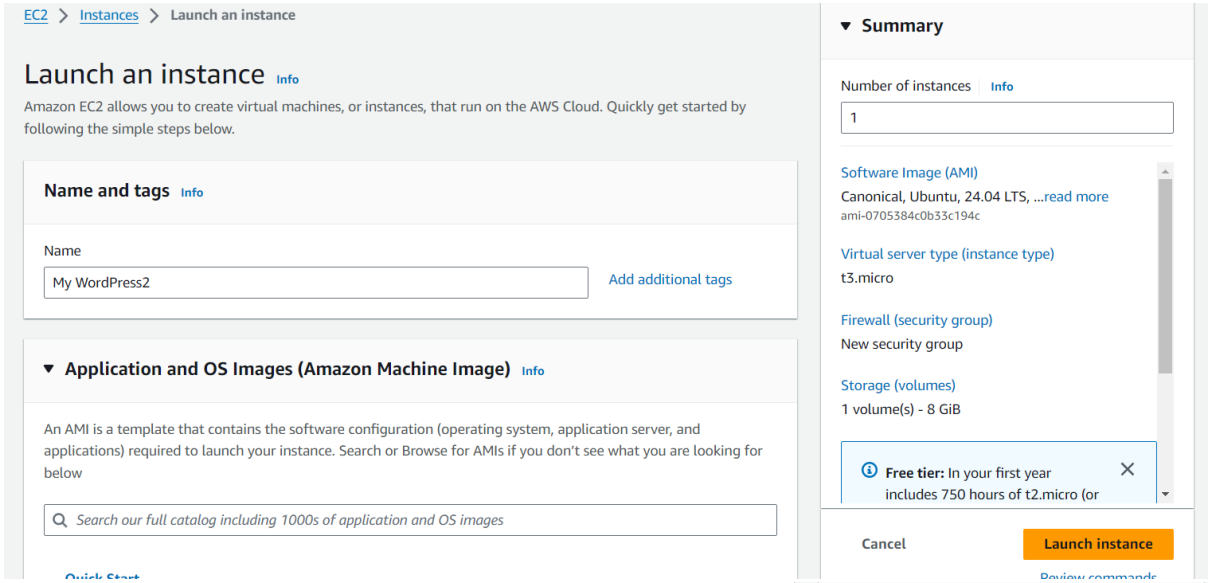
Conclusion

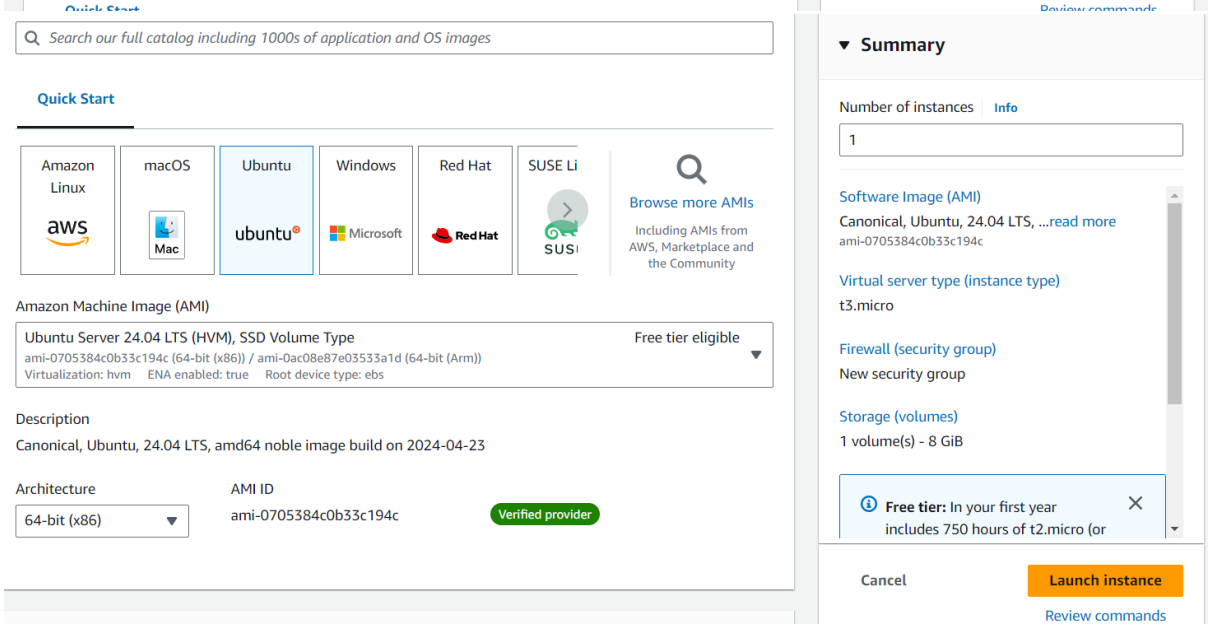
Different approaches to application design are demonstrated via the deployment of WordPress utilizing both microservices and monolithic structures. All services are combined into a single instance in a monolithic arrangement, which simplifies management while maximizing scalability. The application server and database are kept apart by the microservices architecture, which increases fault isolation and scalability at the expense of complexity. After configuring both configurations to guarantee functioning and security, a WordPress welcome page was created. This job prepared for more complex deployment situations in real-world environments by offering insightful information about the advantages and disadvantages of various architectural styles.

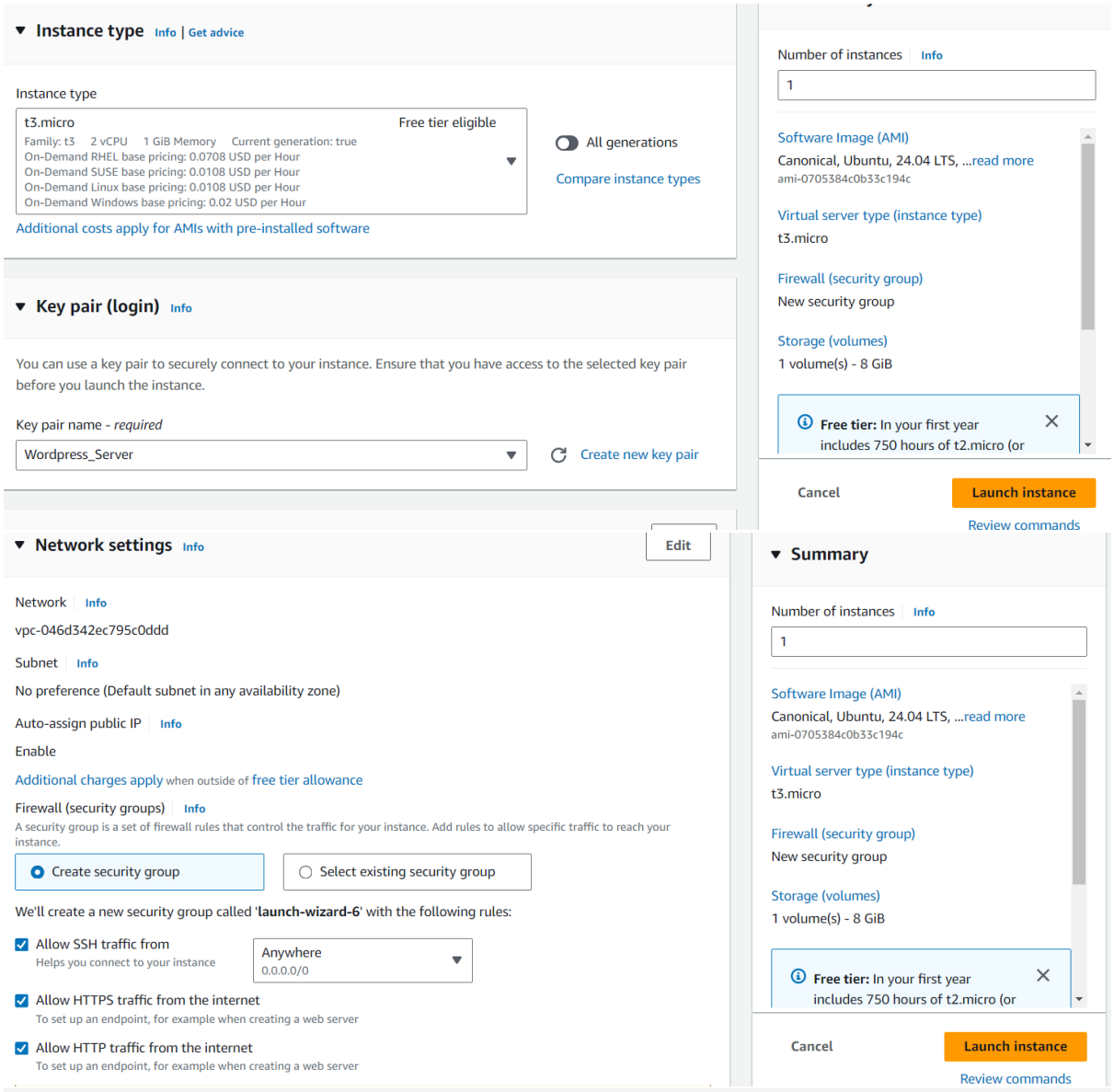
Screenshots

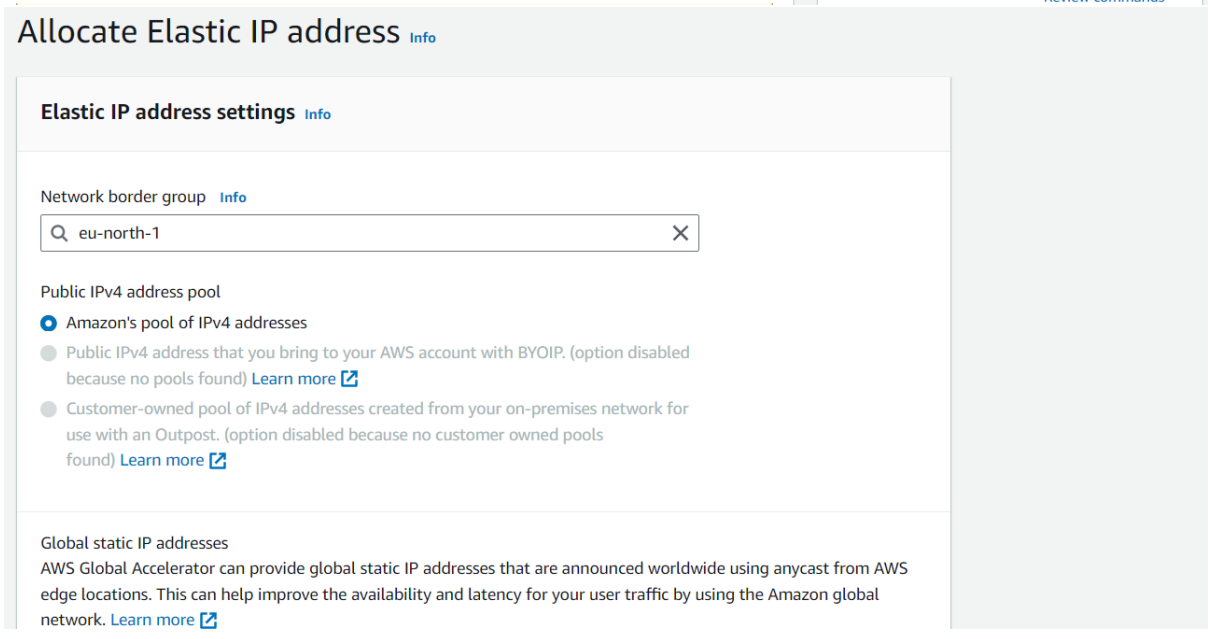

Instances (3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
MySQL3
i-000bb68d38dd536e1
Stopped
t3.micro
–
View alarms
eu-north-1a
ec2-13-48-180-
WORDPRESS3
i-048b477d3d1a24b5e
Stopped
t3.micro
–
View alarms
eu-north-1a
ec2-51-21-63-6
My WordPress ...
i-0ebb94f86070ed47e
Stopped
t3.micro
–
View alarms
eu-north-1b
ec2-13-50-61-1

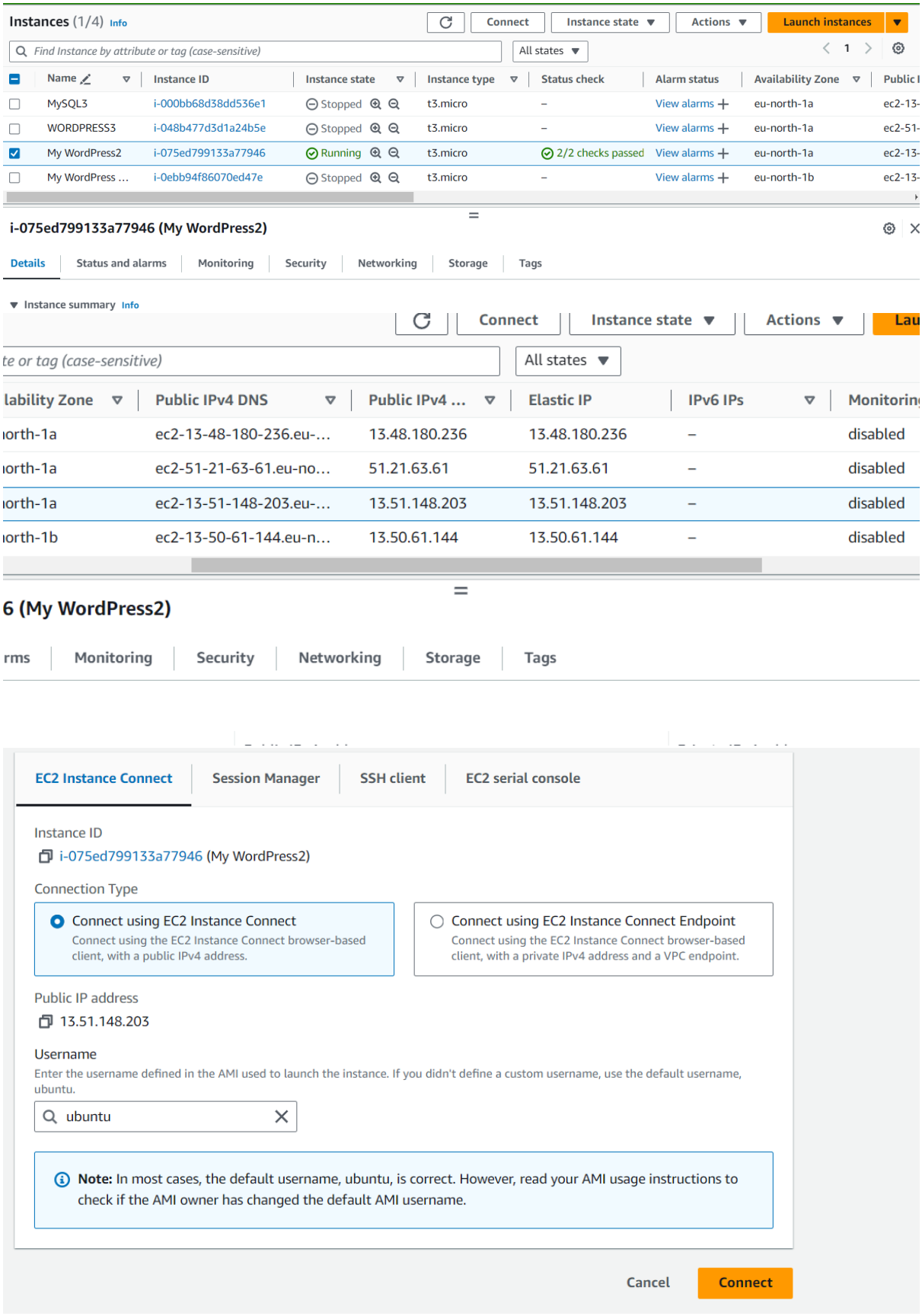

Select an instance


EC2
Instances
Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
My WordPress2
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below
Search our full catalog including 1000s of application and OS images
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0705384c0b33c194c
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Launch instance


Search our full catalog including 1000s of application and OS images
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Li
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0705384c0b33c194c (64-bit (x86)) / ami-0ac08e87e03533a1d (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Canonical, Ubuntu, 24.04 LTS, amd64 noble image build on 2024-04-23
Architecture
64-bit (x86)
AMI ID
ami-0705384c0b33c194c
Verified provider
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0705384c0b33c194c
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Launch instance
Review commands


Instance type Info | Get advice
Instance type
t3.micro
Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand RHEL base pricing: 0.0708 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour
On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour
All generations
Compare instance types
Additional costs apply for AMIs with pre-installed software
Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Wordpress_Server
Create new key pair
Network settings Info
Edit
Network Info
vpc-046d342ec795c0ddd
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-6' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0705384c0b33c194c
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Launch instance
Review commands
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0705384c0b33c194c
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or
Cancel
Launch instance
Review commands


Allocate Elastic IP address Info
Elastic IP address settings Info
Network border group Info
eu-north-1
Public IPv4 address pool
Amazon's pool of IPv4 addresses
Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) Learn more
Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) Learn more
Global static IP addresses
AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. Learn more


Instances (1/4) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
MySQL3 i-000bb68d38dd536e1 Stopped t3.micro – View alarms eu-north-1a
WORDPRESS3 i-048b477d3d1a24b5e Stopped t3.micro – View alarms eu-north-1a
My WordPress2 i-075ed799133a77946 Running t3.micro 2/2 checks passed View alarms eu-north-1a
My WordPress ... i-0ebb94f86070ed47e Stopped t3.micro – View alarms eu-north-1b
i-075ed799133a77946 (My WordPress2)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Connect
Instance state
Actions
All states
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monitoring
ec2-13-48-180-236.eu-... 13.48.180.236 13.48.180.236 – disabled
ec2-51-21-63-61.eu-no... 51.21.63.61 51.21.63.61 – disabled
ec2-13-51-148-203.eu-... 13.51.148.203 13.51.148.203 – disabled
ec2-13-50-61-144.eu-n... 13.50.61.144 13.50.61.144 – disabled
6 (My WordPress2)
Monitoring
Security
Networking
Storage
Tags
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-075ed799133a77946 (My WordPress2)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
13.51.148.203
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect

```
ubuntu@ip-172-31-21-20:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2081 kB of archives.
After this operation, 8087 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-21-20:~$ sudo apt install php libapache2-mod-php php-mysql
```

```
apache2_invoke: Enable module php8.3
Setting up php8.3 (8.3.6-0maysync1) ...
Setting up libapache2-mod-php (2:8.3+93ubuntu2) ...
Setting up php (2:8.3+93ubuntu2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for php8.3-cli (8.3.6-0maysync1) ...
Processing triggers for libapache2-mod-php8.3 (8.3.6-0maysync1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-21-20:~$ sudo apt install mysql-server
```

i-075ed799133a77946 (My WordPress2)

```
Setting up libhtml-template-perl (2.97-2) ...
Setting up libcgi-fast-perl (1:2.17-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-21-20:~$ sudo mysql -u root
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Vivu172469';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'Vivu_user'@localhost IDENTIFIED BY 'Vivu172469';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE DATABASE Vivu_db;
Query OK, 1 row affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON Vivu_db.* TO 'Vivu_user'@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
ubuntu@ip-172-31-21-20:~$
```

```
ubuntu@ip-172-31-21-20:~$ cd /tmp
ubuntu@ip-172-31-21-20:/tmp$ wget https://wordpress.org/latest.tar.gz
--2024-06-08 10:50:51--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24696852 (24M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                       100%[===================================================================>]  23.55M  5.89MB/s    in 6.9s

2024-06-08 10:50:59 (3.42 MB/s) - 'latest.tar.gz' saved [24696852/24696852]

ubuntu@ip-172-31-21-20:/tmp$
```

```
HTTP request sent, awaiting response... 200 OK
Length: 24696852 (24M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                          100%[=========================================

2024-06-08 10:50:59 (3.42 MB/s) - 'latest.tar.gz' saved [24696852/24696852]

ubuntu@ip-172-31-21-20:/tmp$ tar -xvf latest.tar.gz
```

```
wordpress/wp-admin/setup-config.php
wordpress/wp-admin/install.php
wordpress/wp-admin/admin-header.php
wordpress/wp-admin/post-new.php
wordpress/wp-admin/themes.php
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
ubuntu@ip-172-31-21-20:/tmp$ sudo mv wordpress/ /var/www/html
ubuntu@ip-172-31-21-20:/tmp$ cd /var/www/html/
ubuntu@ip-172-31-21-20:/var/www/html$ cd wordpress
ubuntu@ip-172-31-21-20:/var/www/html/wordpress$
```

i-075ed799133a77946 (My WordPress2)

PublicIPs: 13.51.148.203 PrivateIPs: 172.31.21.20

Not secure 13.51.148.203

Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

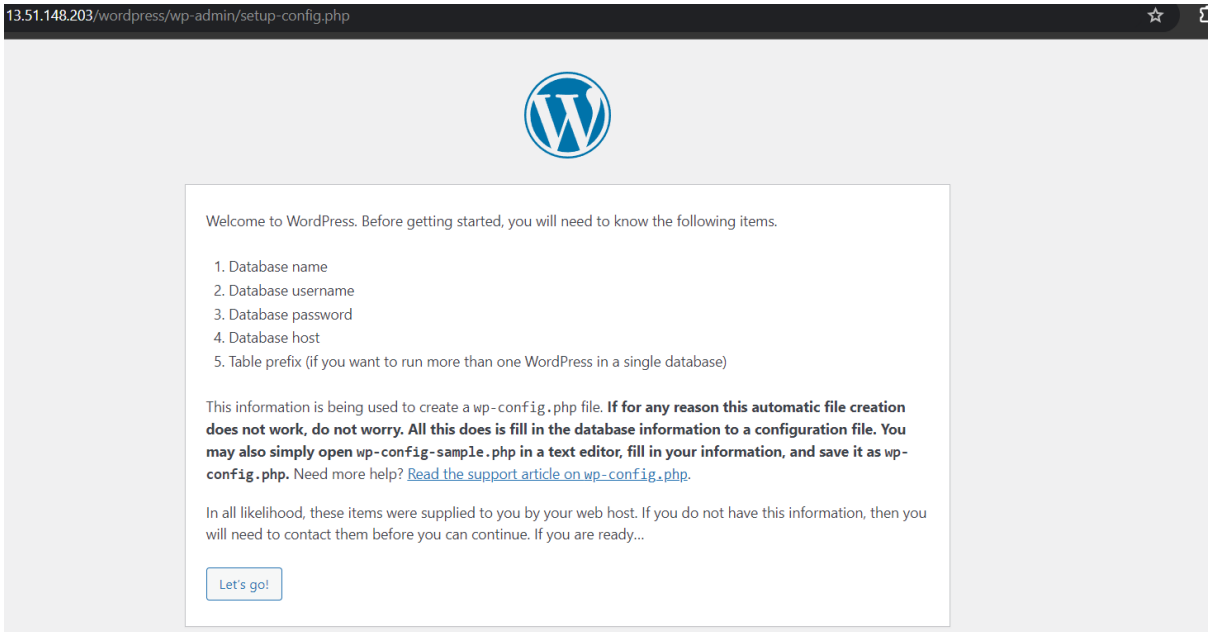
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

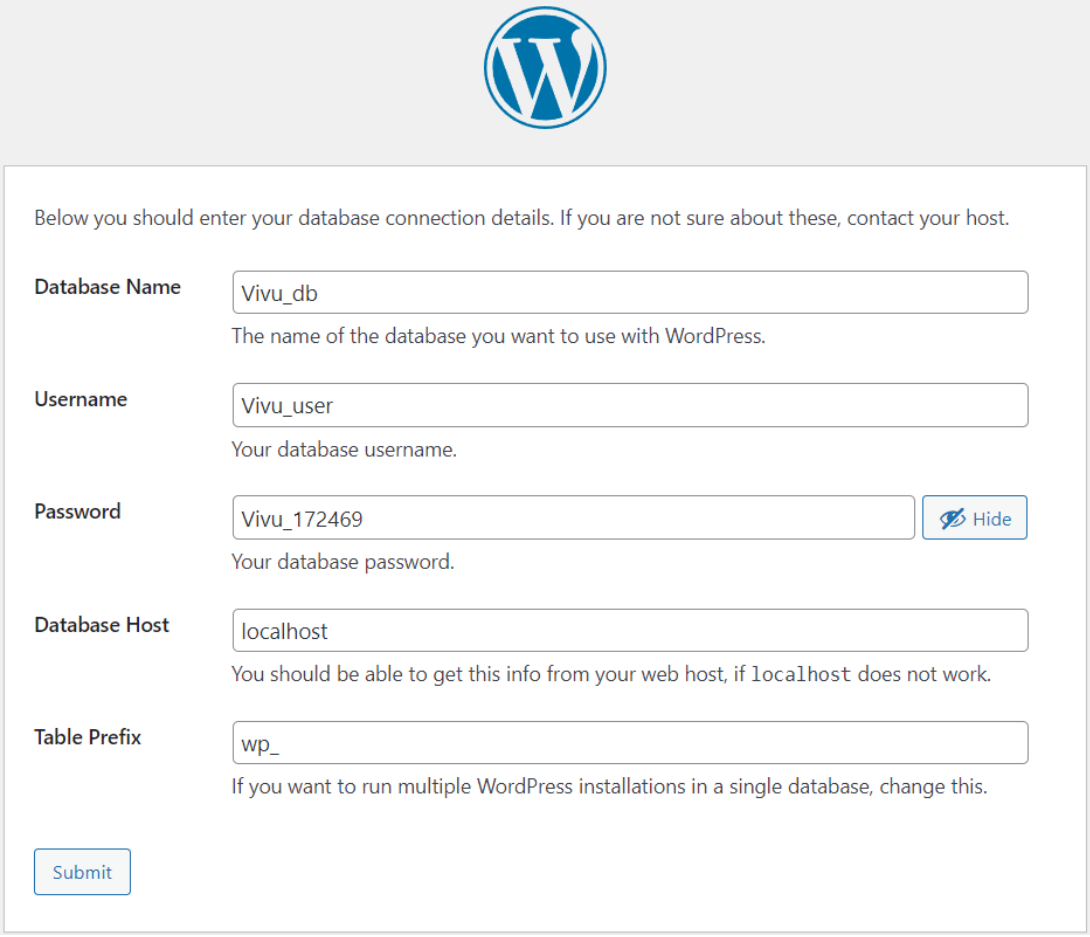
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```


13.51.148.203/wordpress/wp-admin/setup-config.php
Welcome to WordPress. Before getting started, you will need to know the following items.
1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)
This information is being used to create a wp-config.php file. If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open wp-config-sample.php in a text editor, fill in your information, and save it as wp-config.php. Need more help? Read the support article on wp-config.php.
In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...
Let's go!


Below you should enter your database connection details. If you are not sure about these, contact your host.
Database Name
Vivu_db
The name of the database you want to use with WordPress.
Username
Vivu_user
Your database username.
Password
Vivu_172469
Hide
Your database password.
Database Host
localhost
You should be able to get this info from your web host, if localhost does not work.
Table Prefix
wp_
If you want to run multiple WordPress installations in a single database, change this.
Submit

Unable to write to `wp-config.php` file.

You can create the `wp-config.php` file manually and paste the following text into it.

Configuration rules for `wp-config.php`:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
```

After you've done that, click "Run the installation".

Run the installation

⚠ Not secure 13.51.148.203/wordpress/wp-admin/setup-config.php?step=2

Unable to write to `wp-config.php` file.

You can create the `wp-config.php` file manually and paste the following text into it.

Configuration rules for `wp-config.php`:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
```

After you've done that, click "Run the installation".

Run the installation

```
wordpress/wp-comments-post.php
ubuntu@ip-172-31-21-20:/tmp$ sudo mv wordpress/ /var/www/html
ubuntu@ip-172-31-21-20:/tmp$ cd /var/www/html/
ubuntu@ip-172-31-21-20:/var/www/html$ cd wordpress
ubuntu@ip-172-31-21-20:/var/www/html/wordpress$ cd wp-config.php
-bash: cd: wp-config.php: No such file or directory
ubuntu@ip-172-31-21-20:/var/www/html/wordpress$ ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
ubuntu@ip-172-31-21-20:/var/www/html/wordpress$ nano wp-config.php
```

i-075ed799133a77946 (My WordPress2)

PublicIPs: 13.51.148.203 PrivateIPs: 172.31.21.20

```
 GNU nano 7.2                                  wp-config.php *
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/documentation/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** Database settings - You can get this info from your web host ** //

^G Help      ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo      M-A Set Mark   M-] To Bracke
^X Exit      ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo      M-6 Copy       ^Q Where Was
```

i-075ed799133a77946 (My WordPress2)

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 MY SAMPLE WORDPRESS 2

Username VIVAAN_JOHRI

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

Password @Vivaan_Johri_1724@ Hide

Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email vivaanjohri2003@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

Success!

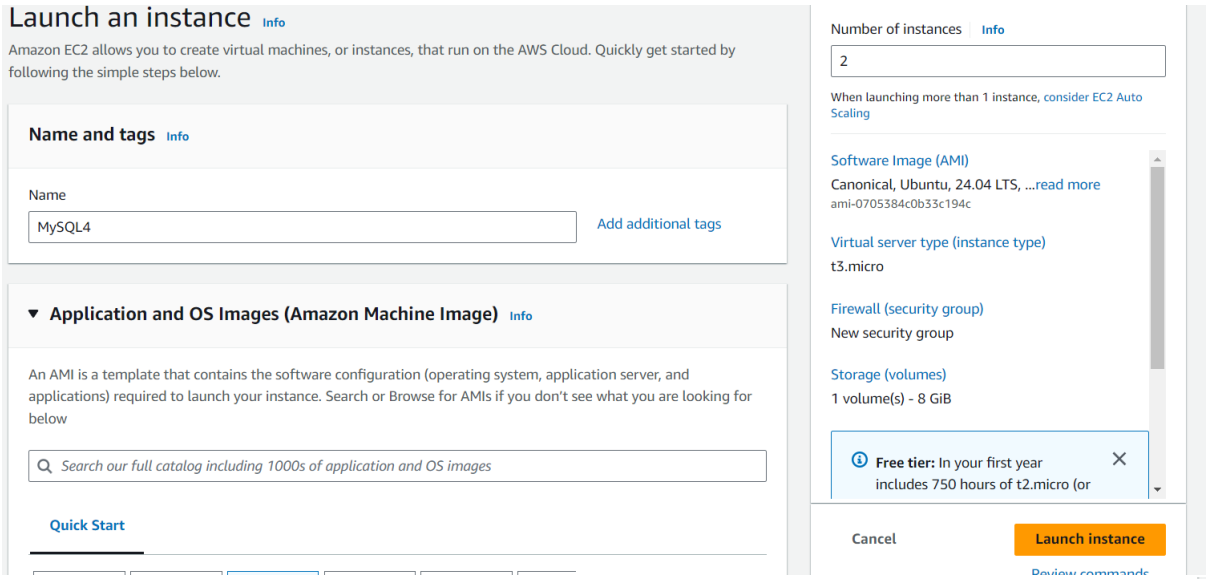
WordPress has been installed. Thank you, and enjoy!

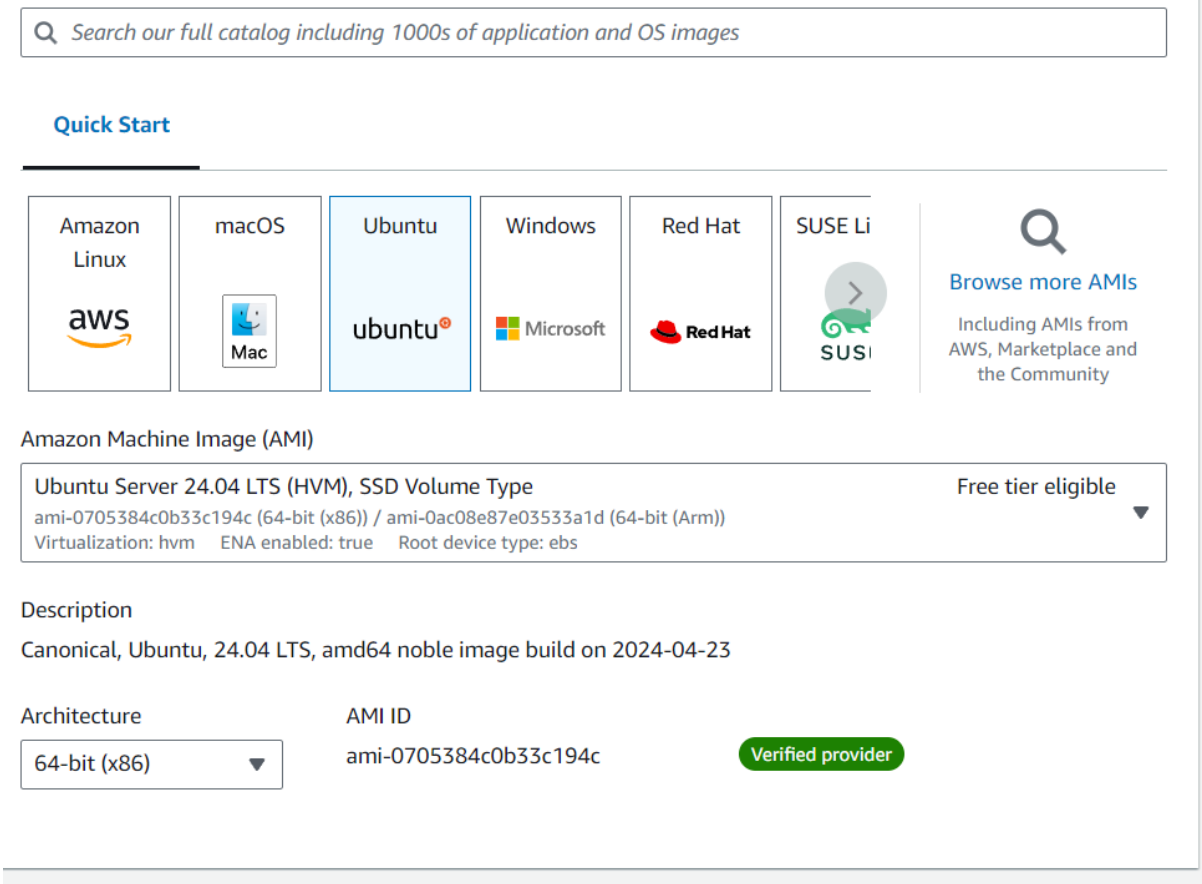
Username Vivaan_Johri

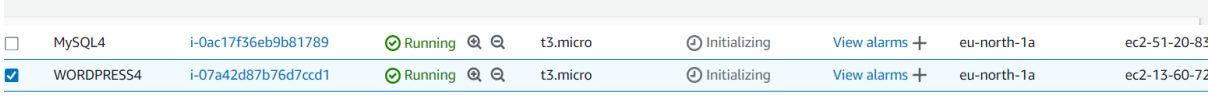
Password *Your chosen password.*

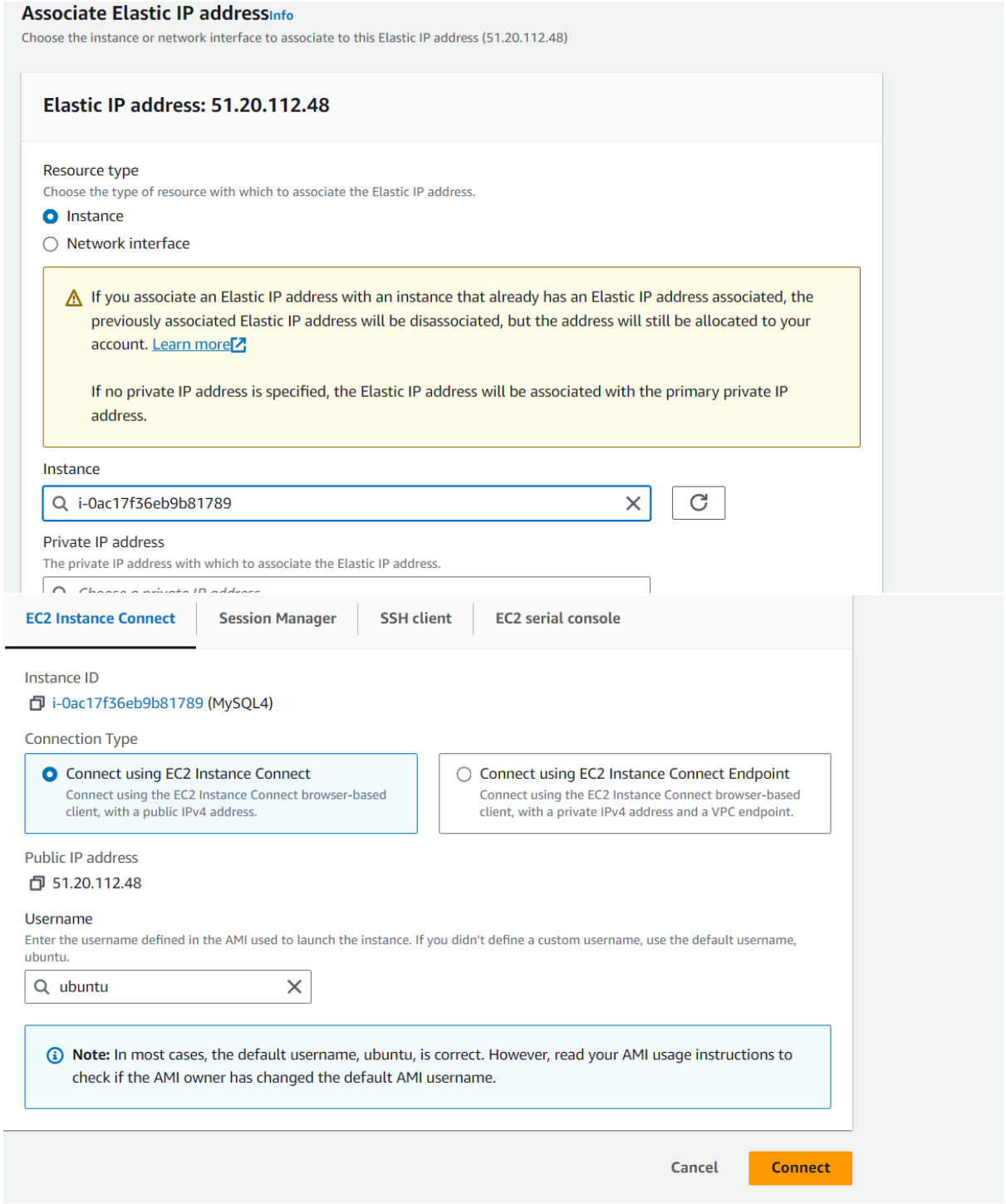
Log In

Microservice Screenshots








Associate Elastic IP address Info
Choose the instance or network interface to associate to this Elastic IP address (51.20.112.48)
Elastic IP address: 51.20.112.48
Resource type
Choose the type of resource with which to associate the Elastic IP address.
Instance
Network interface
If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. Learn more
If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.
Instance
i-0ac17f36eb9b81789
Private IP address
The private IP address with which to associate the Elastic IP address.
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0ac17f36eb9b81789 (MySQL4)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
51.20.112.48
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect

```
The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-26-49:~$ sudo -i
```

i-0ac17f36eb9b81789 (MySQL4)

PublicIPs: 51.20.112.48 PrivateIPs: 172.31.26.49

```
ubuntu@ip-172-31-26-49:~$ sudo -i
root@ip-172-31-26-49:~# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64
  libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0
  mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64
  libtimedate-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.5 MB of archives.
```

```
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-26-49:~# sudo mysql -u root
```

i-0ac17f36eb9b81789 (MySQL4)

PublicIPs: 51.20.112.48 PrivateIPs: 172.31.26.49

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Vivi1724';
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE DATABASE Vivi_db;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'Vivi_user'@'%' IDENTIFIED BY 'Vivi1724';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON Vivi_db.* TO 'Vivi_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@ip-172-31-26-49:~#
```

i-0ac17f36eb9b81789 (MySQL4)

PublicIPs: 51.20.112.48 PrivateIPs: 172.31.26.49

```
mysql> CREATE DATABASE Vivi_db;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'Vivi_user'@'%' IDENTIFIED BY 'Vivi1724';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES ON Vivi_db.* TO 'Vivi_user'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@ip-172-31-26-49:~# sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

i-0ac17f36eb9b81789 (MySQL4)

PublicIPs: 51.20.112.48 PrivateIPs: 172.31.26.49

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 127.0.0.1

# * Fine Tuning

key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location      M-U Undo         M-A Set Mark     M-] To Bracket   M-Q
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-E Redo         M-6 Copy         ^Q Where Was     M-W
```

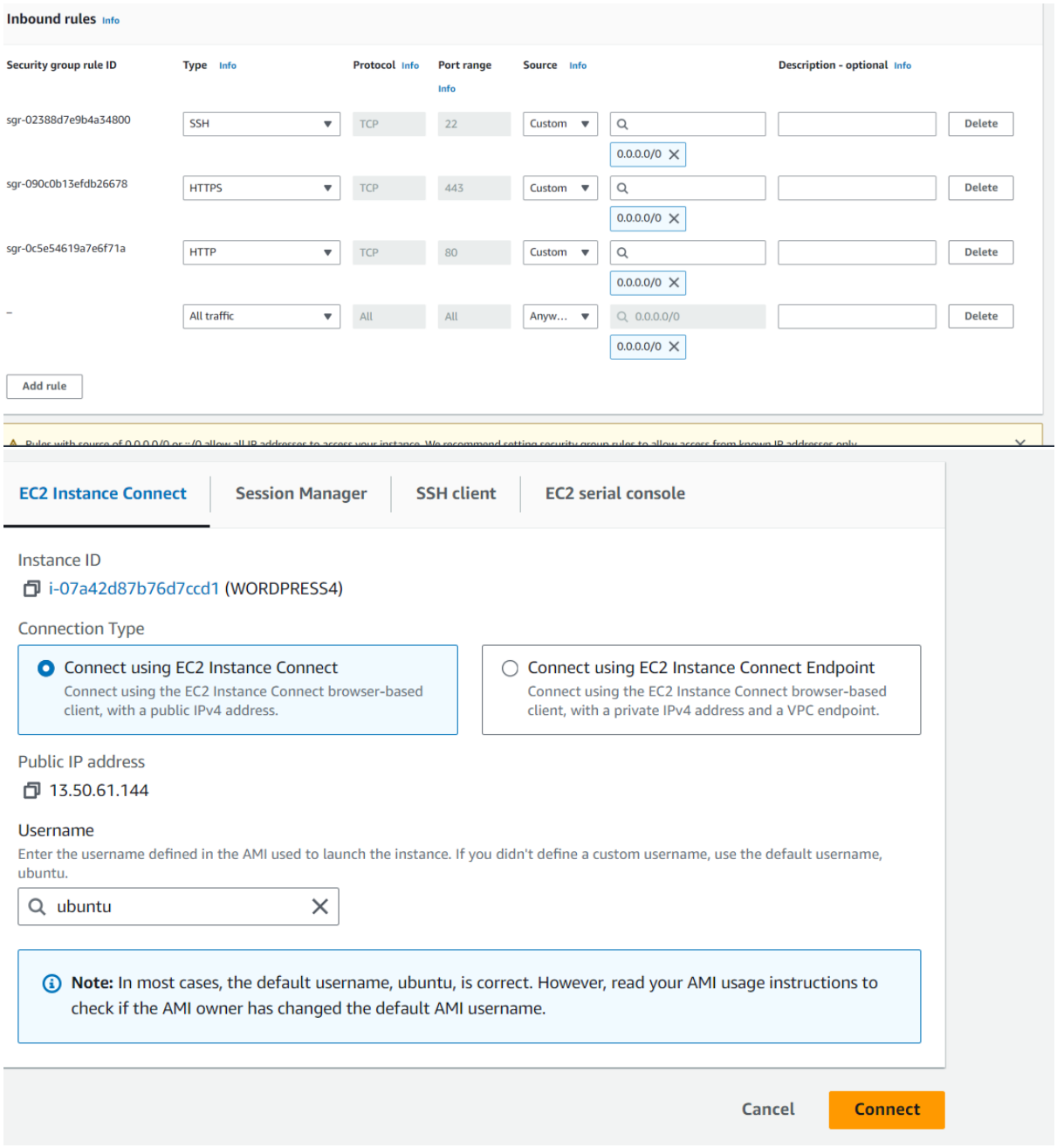
i-0ac17f36eb9b81789 (MySQL4)

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye
root@ip-172-31-26-49:~# sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
root@ip-172-31-26-49:~# systemctl restart mysql
root@ip-172-31-26-49:~#
```

i-0ac17f36eb9b81789 (MySQL4)

PublicIPs: 51.20.112.48 PrivateIPs: 172.31.26.49


Inbound rules Info
Security group rule ID
Type Info
Protocol Info
Port range Info
Source Info
Description - optional Info
sgr-02388d7e9b4a34800
SSH
TCP
22
Custom
0.0.0.0/0
Delete
sgr-090c0b13efdb26678
HTTPS
TCP
443
Custom
0.0.0.0/0
Delete
sgr-0c5e54619a7e6f71a
HTTP
TCP
80
Custom
0.0.0.0/0
Delete
-
All traffic
All
All
Anyw...
0.0.0.0/0
0.0.0.0/0
Delete
Add rule
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-07a42d87b76d7ccd1 (WORDPRESS4)
Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.
Public IP address
13.50.61.144
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.
ubuntu
Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-79:~$ sudo -i
root@ip-172-31-27-79:~# sudo apt update -y
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
Get:42 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64
Get:43 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe am
Get:44 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Tr
Get:45 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe am
Get:46 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe am
Get:47 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted
Get:48 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted
Get:49 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse
Get:50 http://eu-north-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse
Fetched 29.0 MB in 6s (4773 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
75 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-27-79:~# sudo apt install apache2
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-27-79:~# sudo apt install php libapache2-mod-php php-mysql
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-27-79:~# wget https://wordpress.org/latest.tar.gz
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
HTTP request sent, awaiting response... 200 OK
Length: 24696852 (24M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz                       100%[===================================================================================>]  23.55M  6.90MB/s    in 6.0s

2024-06-08 17:35:58 (3.96 MB/s) - 'latest.tar.gz' saved [24696852/24696852]

root@ip-172-31-27-79:~# tar -xvf latest.tar.gz
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

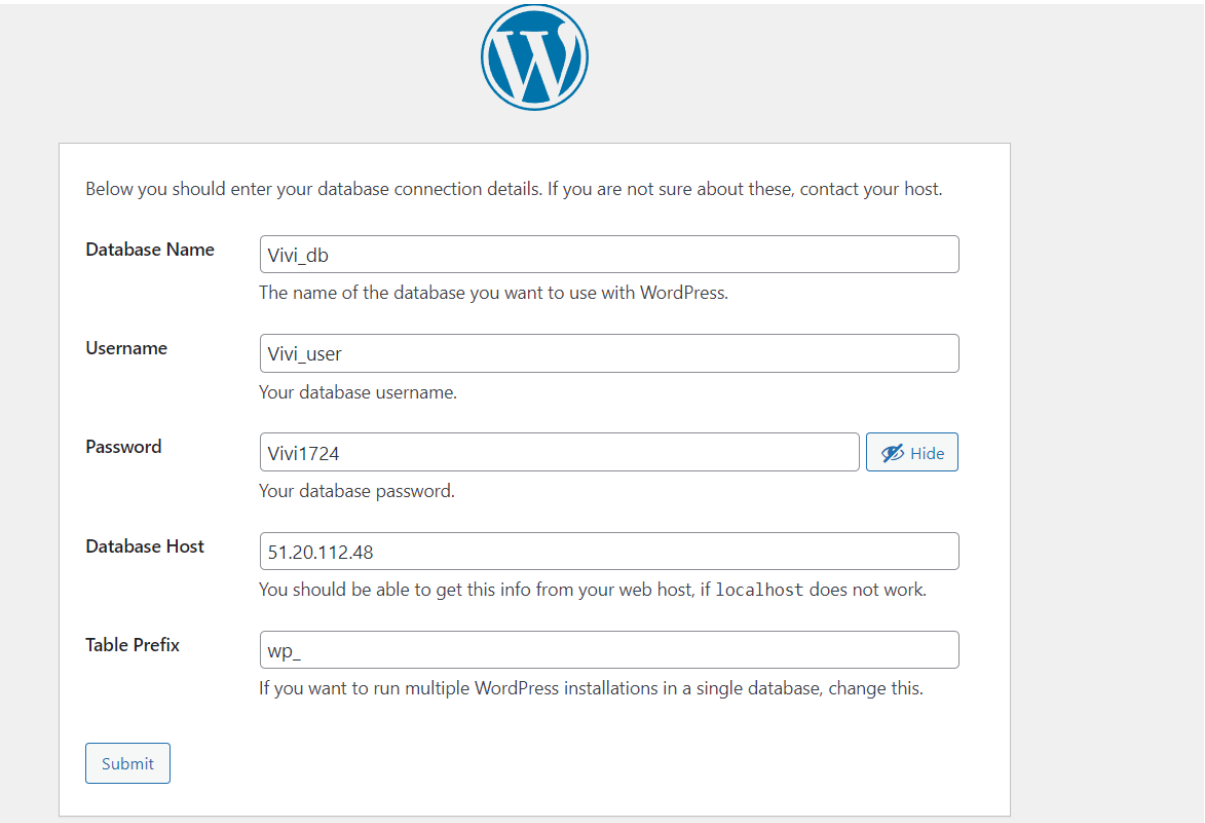
Welcome to WordPress. Before getting started, you will need to know the following items.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

This information is being used to create a `wp-config.php` file. **If for any reason this automatic file creation does not work, do not worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? Read the support article on `wp-config.php`.

In all likelihood, these items were supplied to you by your web host. If you do not have this information, then you will need to contact them before you can continue. If you are ready...

Let's go!



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

Database Name
Vivi_db
The name of the database you want to use with WordPress.

Username
Vivi_user
Your database username.

Password
Vivi1724 Hide
Your database password.

Database Host
51.20.112.48
You should be able to get this info from your web host, if `localhost` does not work.

Table Prefix
wp_
If you want to run multiple WordPress installations in a single database, change this.

Submit

Unable to write to `wp-config.php` file.

You can create the `wp-config.php` file manually and paste the following text into it.

Configuration rules for `wp-config.php`:

```
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the website, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
```

After you've done that, click "Run the installation".

Run the installation

```
ordpress/wp-admin/admin-header.php
ordpress/wp-admin/post-new.php
ordpress/wp-admin/themes.php
ordpress/wp-admin/options-reading.php
ordpress/wp-trackback.php
ordpress/wp-comments-post.php
oot@ip-172-31-27-79:~# sudo mv wordpress/ /var/www/html
oot@ip-172-31-27-79:~# cd /var/www/html
oot@ip-172-31-27-79:/var/www/html# cd wordpress
oot@ip-172-31-27-79:/var/www/html/wordpress#
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
 *
 * For information on other constants that can be used for debugging,
 * visit the documentation.
 *
 * @link https://wordpress.org/documentation/article/debugging-in-wordpress/
 */
define( 'WP_DEBUG', false );

/* Add any custom values between this line and the "stop editing" line. */



/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
        define( 'ABSPATH', __DIR__ . '/' );
}

/** Sets up WordPress vars and included files. */
require_once ABSPATH . 'wp-settings.php';

:wq
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

```
wordpress/wp-admin/options-reading.php
wordpress/wp-trackback.php
wordpress/wp-comments-post.php
root@ip-172-31-27-79:~# sudo mv wordpress/ /var/www/html
root@ip-172-31-27-79:~# cd /var/www/html
root@ip-172-31-27-79:/var/www/html# cd wordpress
root@ip-172-31-27-79:/var/www/html/wordpress# vi wp-config.php
root@ip-172-31-27-79:/var/www/html/wordpress# systemctl restart apache2
root@ip-172-31-27-79:/var/www/html/wordpress#
```

i-07a42d87b76d7ccd1 (WORDPRESS4)

PublicIPs: 13.50.61.144 PrivateIPs: 172.31.27.79

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 Techmplement

Username VIVAAN_JOHRI

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 vivaanjohri2003@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