Report: Deployment of WordPress on AWS EC2 Instance

Introduction:

In this report, I will outline the steps taken to deploy WordPress on an AWS EC2 instance. The deployment was done using the monolithic architecture approach, where WordPress and MySQL are hosted on the same EC2 instance.

Steps Taken:

- Setting up the AWS Environment:
- Created a free-tier AWS account.
- Navigated to the EC2 service in the AWS Management Console.
- Launching the EC2 Instance:
- Clicked on "Launch Instance" and selected an Ubuntu AMI (Amazon Machine Image).
- Choose the "t2.micro" instance type.
- Configured the instance details and security group settings as per the requirements.
- Accessing the EC2 Instance:
- Connected to the EC2 instance using SSH.
- Used the private key file (.pem) for authentication.
- Installing WordPress and MySQL:
- Updated the package index with sudo apt update.
- Installed Apache, MySQL, PHP, and other required packages with sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql.
- Downloaded and extracted the WordPress files into the /var/www/html/ directory.
- Created a MySQL database and user for WordPress.
- Configured WordPress to use the MySQL database.
- Configuring WordPress:
- Copied the sample configuration file to wp-config.php.
- Updated the database connection settings in wp-config.php.
- Testing and Verification:
- Accessed the public IP address of the EC2 instance in a web browser.
- Completed the WordPress setup wizard to finalize the installation.
- Verified that the WordPress site was up and running.

Resources Referred to:

- **AWS Documentation:** Utilized various AWS documentation guides for EC2 instance setup, security group configuration, and SSH connection.

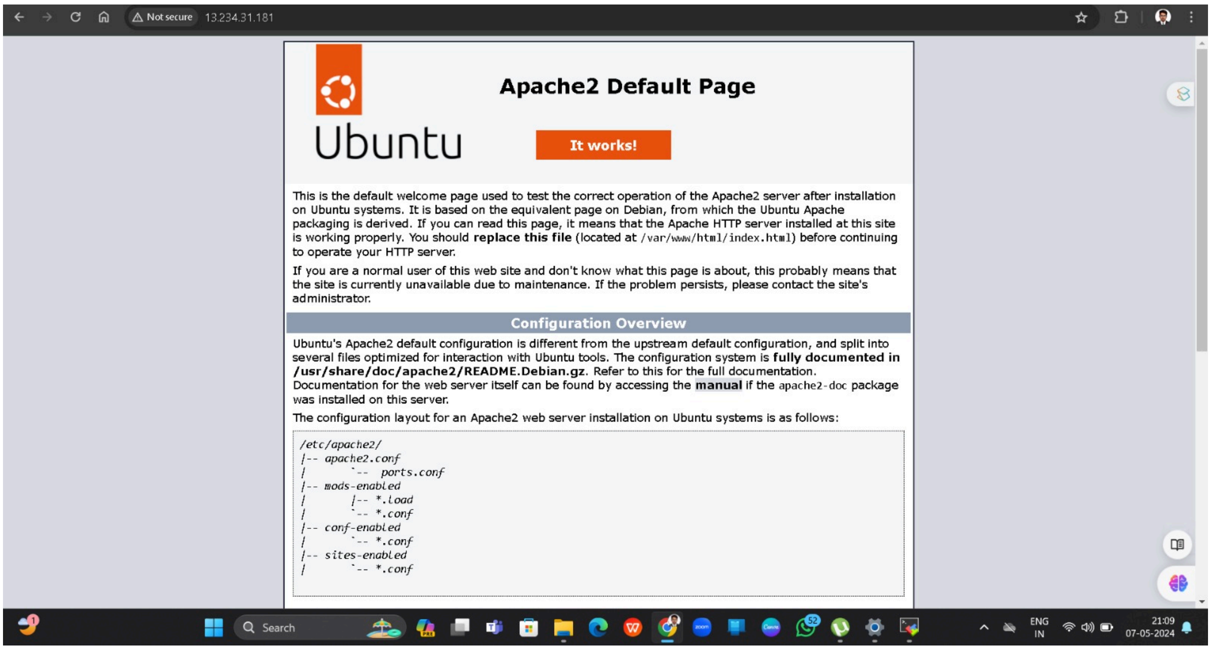
- **WordPress Installation Guide:** Followed the official WordPress installation guide for downloading, installing, and configuring WordPress.

- **Online tutorials and forums:** Referred to various online tutorials and forums for troubleshooting specific issues and clarifications on certain steps.

Conclusion:

The deployment of WordPress on an AWS EC2 instance using the monolithic architecture approach was successfully completed. By following a systematic approach and referring to relevant resources, the task was accomplished efficiently. The WordPress site is now ready for further customization and content creation.

Images:



Apache2 Default Page

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.

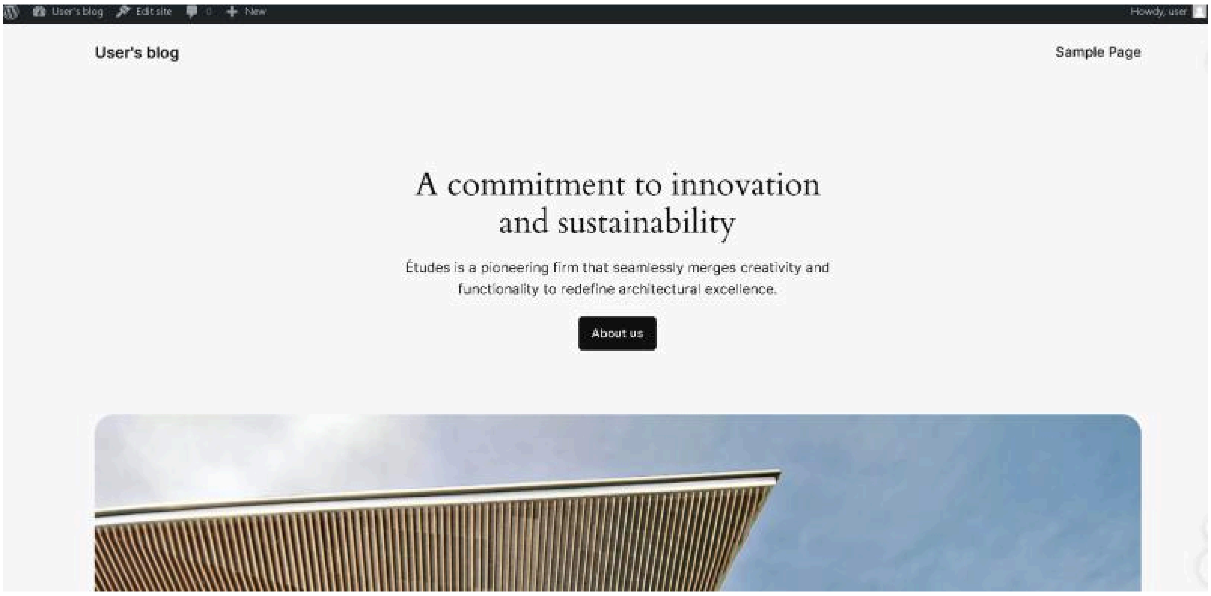
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

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
```



A commitment to innovation and sustainability

Études is a pioneering firm that seamlessly merges creativity and functionality to redefine architectural excellence.

Commands used in this Project:

1. Connect to EC2 Instance via SSH:

```
ssh -i your-key.pem ubuntu@your-instance-ip
```

This command establishes an SSH connection to your EC2 instance using your private key (your-key.pem) and the instance's public IP address (your-instance-ip).

2. Update Package Index:

```
sudo apt update
```

This command updates the package index on your EC2 instance, ensuring that you have access to the latest versions of packages.

3. Install Required Packages:

```
sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql
```

This command installs Apache web server, MySQL database server, PHP, and necessary PHP modules to run WordPress on your EC2 instance.

4. Download WordPress:

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

This command downloads the latest version of WordPress in a .tar.gz archive format to your EC2 instance.

5. Extract WordPress Files:

```
tar -xzvf latest.tar.gz
```

This command extracts the WordPress files from the downloaded archive.

6. Move WordPress Files to Apache Document Root:

```
sudo mv wordpress/* /var/www/html/
```

This command moves the extracted WordPress files to the Apache document root directory, which is where your website files are served from.

7. Create MySQL Database and User for WordPress:

```
sudo mysql -u root -p
```

This command logs into the MySQL shell using the root user.

SQL

```
CREATE DATABASE wordpress;
CREATE USER 'wordpressuser'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON wordpress.* TO 'wordpressuser'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

These SQL commands create a MySQL database named wordpress, create a MySQL user named wordpressuser with the password password, grant all privileges on the wordpress database to the wordpressuser, and then flush privileges to apply the changes.

8. Configure WordPress:

```
sudo cp /var/www/html/wp-config-sample.php /var/www/html/wp-config.php
sudo nano /var/www/html/wp-config.php
```

The first command copies the WordPress sample configuration file to a new file named wp-config.php.The second command opens the wp-config.php file in the Nano text editor, allowing you to edit it.

In the wp-config.php file, you'll need to update the database connection settings with the database name, username, password, and host.

These commands are the fundamental steps involved in setting up WordPress and MySQL on an AWS EC2 instance. Adjustments may be needed depending on your specific setup or requirements.