

Techplement Cloud (AWS) Week 1 Task Report

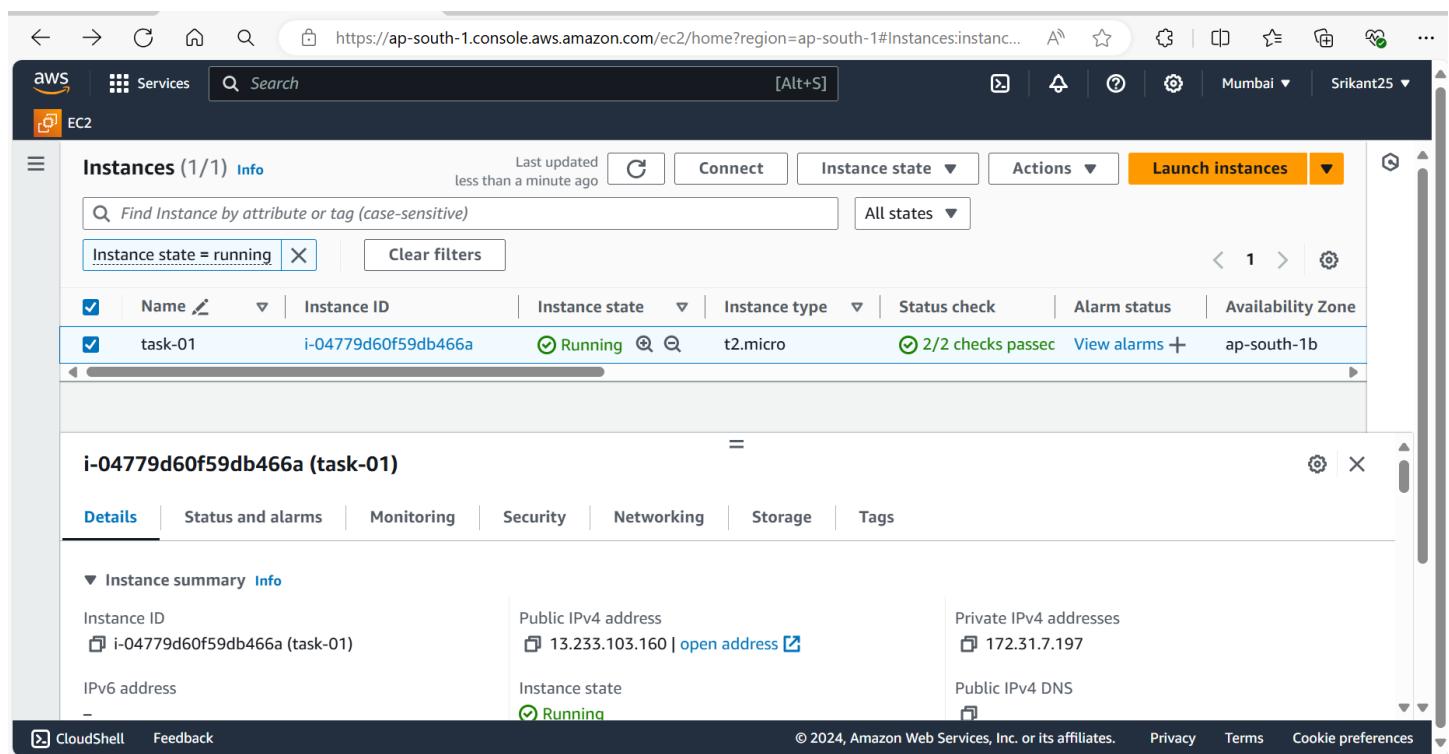
Overview:

This report details the process of deploying WordPress in both *monolithic* and *microservices* architectures using Amazon Web Services (AWS). The task involves setting up a WordPress instance and MySQL database on a single **EC2** instance for the monolithic approach and deploying WordPress and MySQL on separate **EC2** instances for the microservices architecture.

Monolithic Architecture

Step 1: Setting up EC2 Instance

- For the monolithic architecture, I launched a single **EC2** instance using the `t2.micro` instance type with an Ubuntu AMI (`ubuntu`).



The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a 'Launch instances' button. Below it, a table lists one instance: 'task-01' (Instance ID: i-04779d60f59db466a), which is 'Running' (Status check: 2/2 checks passed) and has an 't2.micro' instance type. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability Zone. Below the table, the instance details for 'task-01' are shown, including its Public IPv4 address (13.233.103.160) and Private IPv4 address (172.31.7.197). The 'Details' tab is selected in the navigation bar.

Step 2: Installing Apache, PHP, and MySQL

- Updated the instance and installed Apache

```
sudo apt update  
sudo apt install apache2 -y
```

- Installed PHP and other necessary packages:

```
sudo apt install php libapache2-mod-php php-mysql -y
```

- Installed MySQL on the same instance and secured it with a root password:

```
sudo apt install mysql-server -y  
sudo mysql_secure_installation
```

Step 3: Configuring WordPress

- Downloaded and extracted WordPress files in the Apache root directory:

```
wget https://wordpress.org/latest.tar.gz  
tar -xvzf latest.tar.gz  
sudo mv wordpress /var/www/html/
```

- Configured the WordPress database and set permissions .
- Edited the `wp-config.php` file with database credentials  .

```
ubuntu@ip-172-31-8-248: /var/www/html/wordpress
ubuntu@ip-172-31-8-248:~$ cd /var/www/html/wordpress
ubuntu@ip-172-31-8-248:/var/www/html/wordpress$ nano wp-config.php
```

```
ubuntu@ip-172-31-8-248: /var/www/html/wordpress
GNU nano 7.2          wp-config.php *
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */

define('DB_NAME', 'wordpress_db');

/** MySQL database username */
define('DB_USER', 'wordpress_user');

/** MySQL database password */
define('DB_PASSWORD', 'password_dh');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secr>
 *
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste    ^J Justify   ^/ Go To Line
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

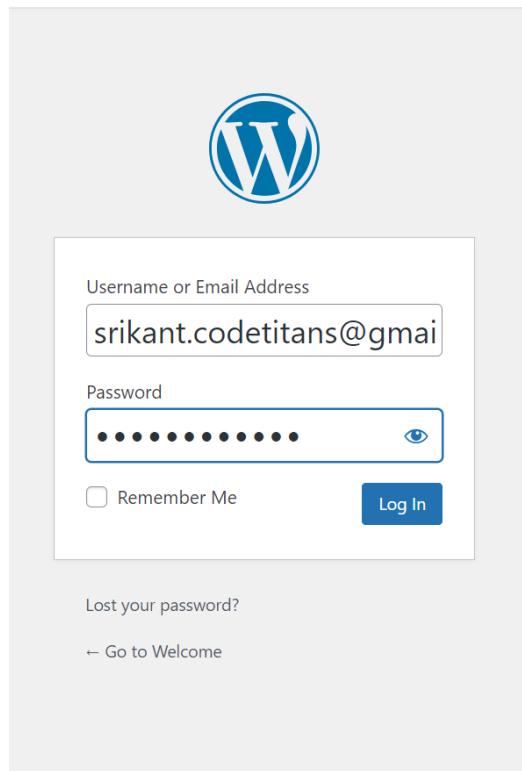
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql               |
| performance_schema |
| sys                |
| wordpress_db       |
+--------------------+
5 rows in set (0.01 sec)
```

Step 4: Completing the Setup

- After completing the configuration, I accessed the WordPress site via the instance's public IP, followed the setup process, and created a welcome page.
- The security groups were configured to allow HTTP, HTTPS, and SSH traffic.

The screenshot shows the AWS EC2 instance details page. The instance has an IAM Role assigned and is part of a security group named 'sg-0534496dd361a9391'. The inbound rules table lists three rules allowing TCP traffic on ports 80, 443, and 22 from 0.0.0.0/0.

Name	Security group rule ID	Port range	Protocol	Source
-	sgr-091a0a8f4604f5767	80	TCP	0.0.0.0/0
-	sgr-09d6add99d450f9fe	443	TCP	0.0.0.0/0
-	sgr-019240c0b7f6773e9	22	TCP	0.0.0.0/0



<https://www.linkedin.com/in/srikantsnehit/>

The screenshot shows a Microsoft Edge browser window with three tabs open:

- Launch an instance | EC2 | ap-sou...
- Instance details | EC2 | ap-south-1 |
- WordPress > Installation

The active tab is "WordPress > Installation" at the URL 43.204.237.213/wordpress/wp-admin/install.php?step=1. The page displays the classic blue "W" WordPress logo at the top. Below it, a large "Welcome" heading is followed by a sub-headline: "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." A section titled "Information needed" asks for "Site Title", "Username", and "Password". A note specifies that usernames can only contain alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol. The bottom of the page includes a standard Windows taskbar with various pinned icons like File Explorer, Task View, and Control Panel.

The screenshot shows a Microsoft Edge browser window with the following details:

- Address bar: Not secure | 15.206.147.101/wordpress/
- User info: Howdy, srikant_25
- Toolbar: Home, Edit site, New

The main content area features a dark-themed "Welcome ..." page. It includes a small decorative graphic of a bird in flight. To the right, a quote by Winston Churchill is displayed: "Success is not final, failure is not fatal: It is the courage to continue that counts." Below the quote is a teal button with the text "Click here!". At the bottom, the author's name "– Winston Churchill" is cited.

Microservices Architecture

In the microservices architecture, WordPress and MySQL were deployed on separate **EC2** instances, providing scalability and better management.

Step 1: Launching Two EC2 Instances

MySQL Instance:

- Launched a `t2.micro` instance for the MySQL database.
- Configured security groups to allow inbound MySQL (port `3306`) traffic from the WordPress instance.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays the 'Instance summary for i-0fcc4b0d5477af2e3 (MySQL-Instance)'. The instance is running and has a public IPv4 address of 13.127.214.157 and a private IPv4 address of 172.31.13.34. It also has a public IPv4 DNS name of ec2-13-127-214-157.ap-south-1.compute.amazonaws.com and a private IP DNS name of ip-172-31-13-34.ap-south-1.compute.internal. The instance type is t2.micro. The page includes standard AWS navigation and search tools.

WordPress Instance:

- Launched a separate `t2.micro` instance for WordPress.
- Configured the security group to allow HTTP (port `80`), HTTPS (port `443`), and SSH (port `22`) traffic.

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main content area displays the 'Instance summary for i-05912ea27c2c8d80a (WordPress-Instance)'. The instance is running and has a public IPv4 address of 13.201.186.194 and a private IPv4 address of 172.31.1.232. It also has a public IPv4 DNS name of ec2-13-201-186-194.ap-south-1.compute.amazonaws.com and a private IP DNS name of ip-172-31-1-232.ap-south-1.compute.internal. The instance type is t2.micro. The page includes standard AWS navigation and search tools.

Step 2: Setting Up MySQL Instance

- Installed MySQL and configured it to allow remote connections by editing the `mysqld.cnf` file:

```
bind-address = 0.0.0.0
```

- Created a new database and user for WordPress:

```
CREATE DATABASE wordpress_db;
CREATE USER 'wordpress_user'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON wordpress_db.* TO 'wordpress_user'@'%';
FLUSH PRIVILEGES;
```

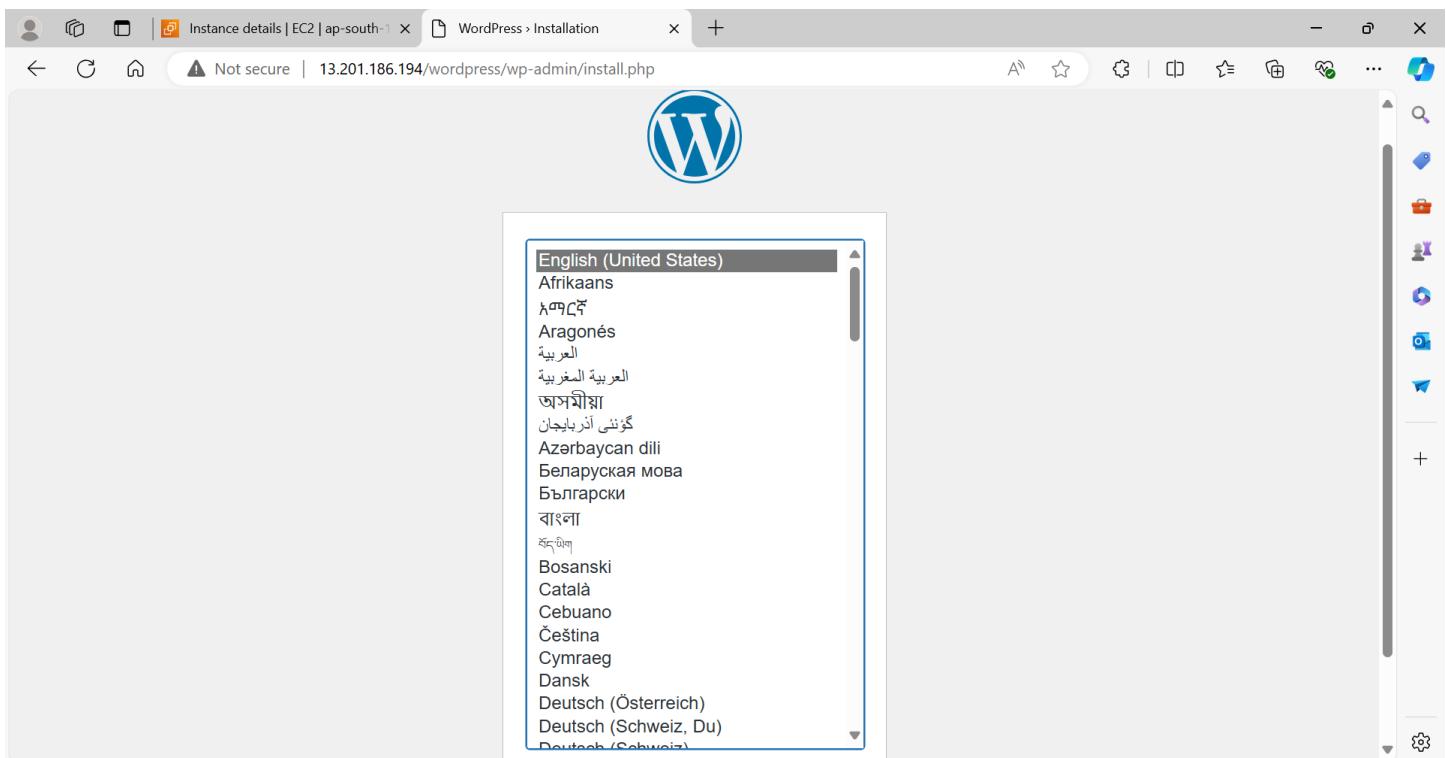
Step 3: Setting Up WordPress Instance

- Installed Apache, PHP, and WordPress as in the monolithic setup.
- Configured the `wp-config.php` file to point to the MySQL instance's private IP for the database connection:

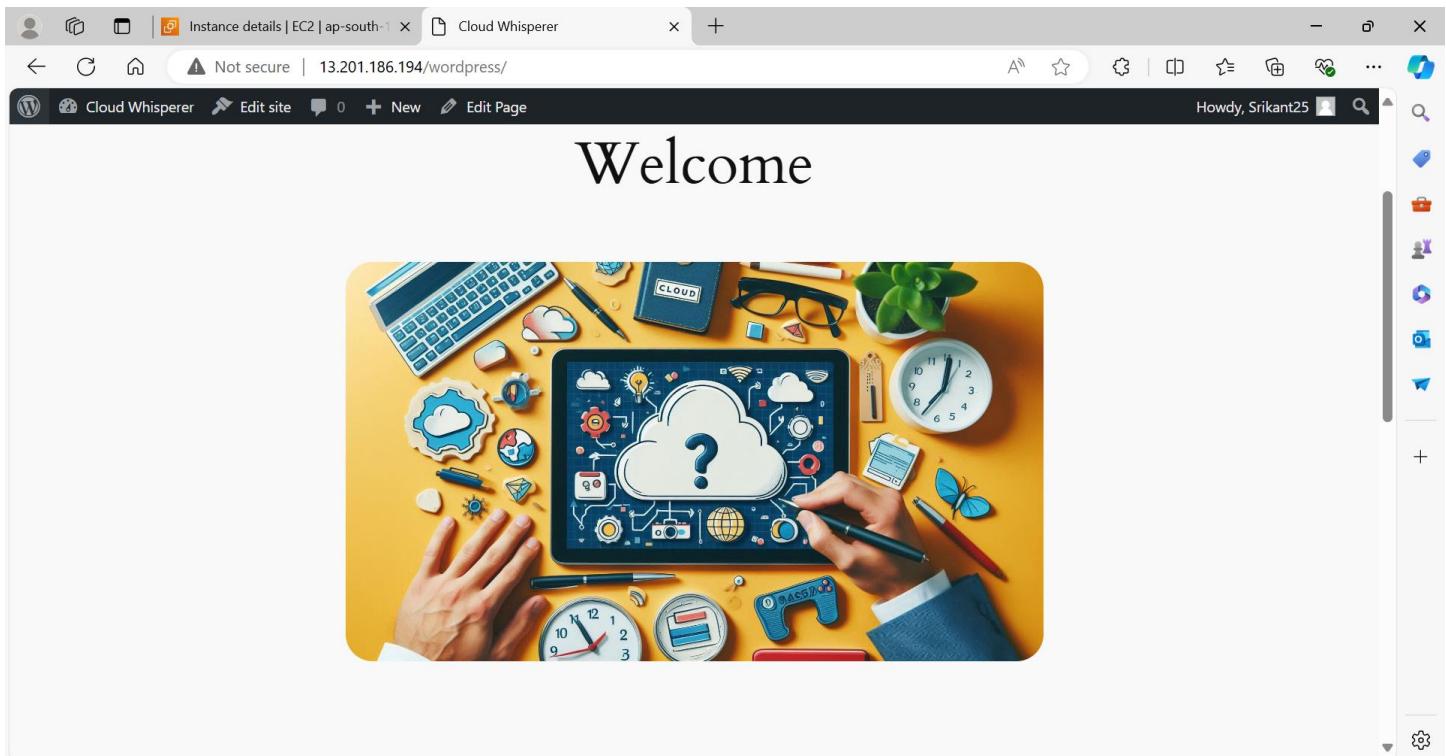
```
define('DB_HOST', '<mysql-instance-private-ip>');
```

Step 4: Finalizing WordPress Setup

Accessed the WordPress site via the WordPress instance's public IP and completed the setup. The Welcome Page was recreated, and the homepage was set to display this page.



The screenshot shows the WordPress dashboard. At the top, there is a header with the user "Howdy, Srikant25" and links for "Screen Options" and "Help". The main area features a large "Welcome to WordPress!" message with a link to "Learn more about the 6.6.1 version.". Below this, there are three main call-to-action boxes: "Author rich content with blocks and patterns", "Customize your entire site with block themes", and "Switch up your site's look & feel with Styles". Each box contains a brief description and a "Dismiss" button. On the left side, there is a sidebar with links for "Dashboard", "Home", "Updates", "Posts", "Media", "Pages", "Comments", "Appearance", "Plugins", "Users", "Tools", "Settings", and a "Collapse menu" option. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.



Challenges and Solutions

Database Connection Issue : I initially encountered an error establishing a connection to the MySQL database. This was resolved by updating the wp-config.php file with the correct database credentials and MySQL instance IP address.

New Public IP on EC2 Restart : After restarting EC2 instances, the public IP changed. I had to use the new IP to access the WordPress site.

Resources

Throughout the task, I referred to the following resources:

1. AWS EC2 Documentation: <https://docs.aws.amazon.com/ec2>
2. WordPress Installation Guide: <https://wordpress.org/support/article/how-to-install-wordpress>
3. MySQL Documentation: <https://dev.mysql.com/doc>
4. AWS Security Groups Best Practices: <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-security-group-best-practices/>
5. Microservices Architecture: <https://d1.awsstatic.com/whitepapers/microservices-on-aws.pdf>



More content coming soon!

Stay tuned ☀️.

Follow me on GitHub: <https://github.com/srikxcipher>

Follow me on LinkedIn: <https://www.linkedin.com/in/srikantsnehit/>