

# Report: Deploying Application in Monolithic and Microservices Architectures

## Task Overview

The objective of this task is to deploy an application using both monolithic and microservices architectures on AWS. The setup involves deploying WordPress and MySQL on EC2 instances configured as per the architecture requirements.

## Architecture Description

1. **Monolithic Architecture:**
  - **Setup:** WordPress and MySQL are deployed on a single EC2 instance.
  - **Instance Type:** `t2-micro`
  - **AMI:** Ubuntu
  - **Security Group Configuration:** Configured to allow HTTP, HTTPS, and MySQL traffic, as well as SSH for instance management.
2. **Microservices Architecture:**
  - **Setup:** Two separate EC2 instances are used—one for WordPress and another for MySQL.
  - **Instance Type:** `t2-micro` for both instances
  - **AMI:** Ubuntu
  - **Security Group Configuration:** Separate security groups for each service to ensure controlled access between WordPress and MySQL.

## Step-by-Step Process

### Monolithic Architecture Deployment

1. **Launch EC2 Instance:**
  - Choose `t2-micro` as the instance type.
  - Select Ubuntu as the AMI.
  - Configure a security group allowing SSH (port 22), HTTP (port 80), and MySQL (port 3306).
2. **Install WordPress and MySQL:**
  - Update package manager and install Apache, PHP, MySQL, and WordPress.
  - Configure MySQL, set up the database, and update WordPress configuration to connect with MySQL.

- Start services and verify that WordPress is accessible.
- 3. **Create Welcome Page:**
  - Log in to the WordPress dashboard and create a welcome page.
  - Set the welcome page as the homepage.

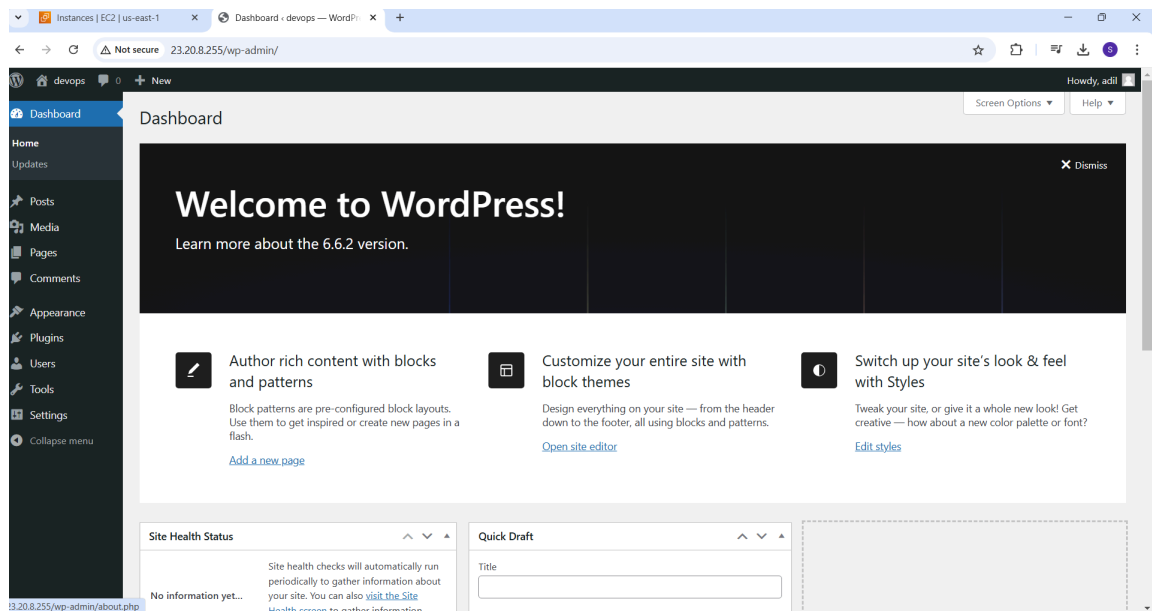
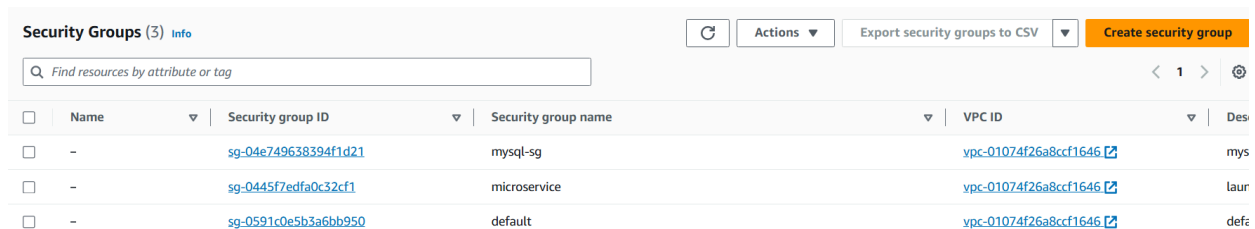
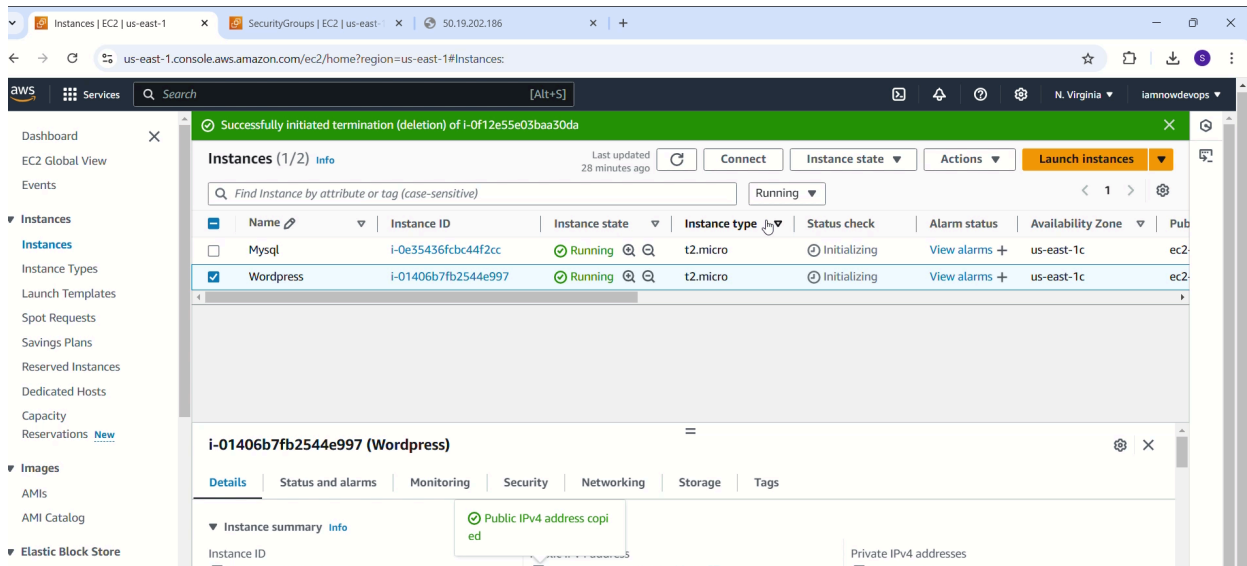
## Microservices Architecture Deployment

1. **Launch Two EC2 Instances:**
  - Instance 1 for WordPress with `t2-micro` and Ubuntu AMI.
  - Instance 2 for MySQL with `t2-micro` and Ubuntu AMI.
2. **Configure Security Groups:**
  - Configure security group for the WordPress instance to allow HTTP and SSH access.
  - Configure security group for MySQL to allow access only from the WordPress instance.
3. **Install and Configure Services:**
  - Install MySQL on the MySQL instance and configure the database.
  - Install WordPress on the WordPress instance and configure it to connect to the MySQL instance's private IP.
4. **Create Welcome Page:**
  - Log in to WordPress and set up the welcome page as in the monolithic architecture.

## Resources and References

- **AWS EC2 Documentation:** [EC2 Instance Launch Guide](#)
- **WordPress Installation Guide:** WordPress Setup
- **MySQL Setup Guide:** [MySQL on Ubuntu](#)

## Screenshots



## Conclusion

In this task, I deployed WordPress and MySQL using both monolithic and microservices approaches, comparing how each architecture manages resources and service isolation. This task provided insights into architectural differences and security configurations essential for scalable deployments.