Task Summary: Setting Up Microservices Architecture with WordPress and MySQL

1. Overview:

We deployed a microservices architecture with two EC2 instances on AWS, one for hosting WordPress and the other for MySQL. The goal was to configure WordPress to use MySQL as its backend database.

2. Resources Used:

- Two EC2 instances:
  - One for WordPress hosting (t2.micro, Ubuntu AMI)
  - One for MySQL database (t2.micro, Ubuntu AMI)
- Security Groups:
  - Created and configured security groups for both instances to control inbound and outbound traffic.
- WordPress:
  - Installed and configured WordPress on the WordPress hosting instance.
  - Configured WordPress to use MySQL as its backend database.
- MySQL:
  - Installed and configured MySQL on the MySQL instance.
  - Adjusted MySQL configuration to allow external connections.

3. Steps Taken:

1. Launched two t2.micro EC2 instances using the Ubuntu AMI.
2. Configured security groups for both instances:
   - Allowed inbound traffic on port 22 (SSH) for administration.
   - Allowed inbound traffic on port 80 (HTTP) and 443 (HTTPS) for web traffic to WordPress.
   - Allowed inbound traffic on port 3306 (MySQL) from the WordPress instance's security group.
   - Allowed outbound traffic to all destinations for both instances.
3. Installed and configured WordPress on the WordPress instance:
   - Set up Apache web server and PHP.
   - Downloaded and configured WordPress files.
   - Configured WordPress to use MySQL database with appropriate credentials.
4. Installed and configured MySQL on the MySQL instance:
   - Installed MySQL server and client packages.

- Configured MySQL to listen on all network interfaces for external connections.
- Created a MySQL database and user for WordPress.

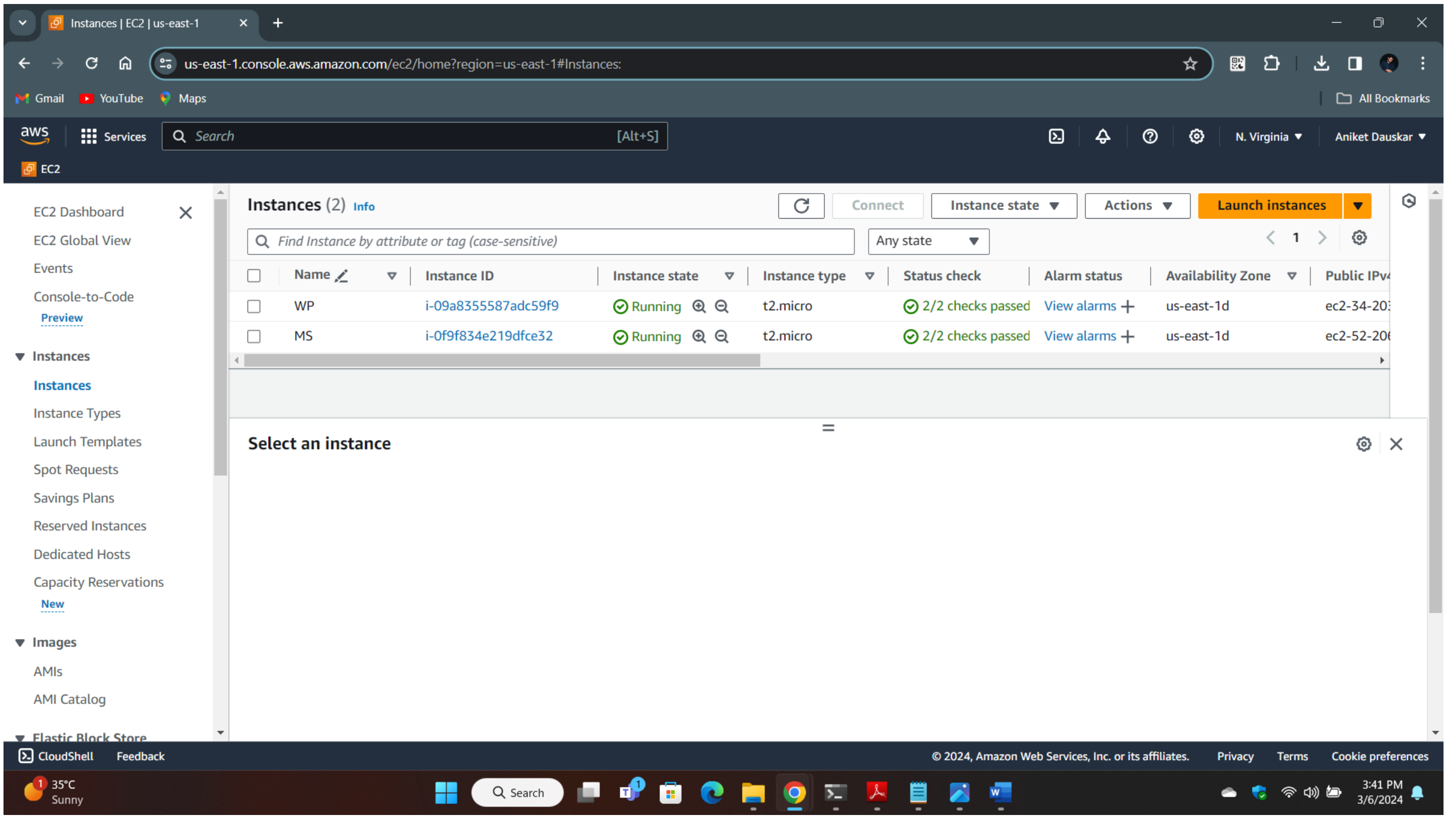
5. Tested the connectivity between the WordPress and MySQL instances to ensure they can communicate successfully.

6. Created a welcome page in WordPress to serve as the homepage of the website.

4. Outcome:

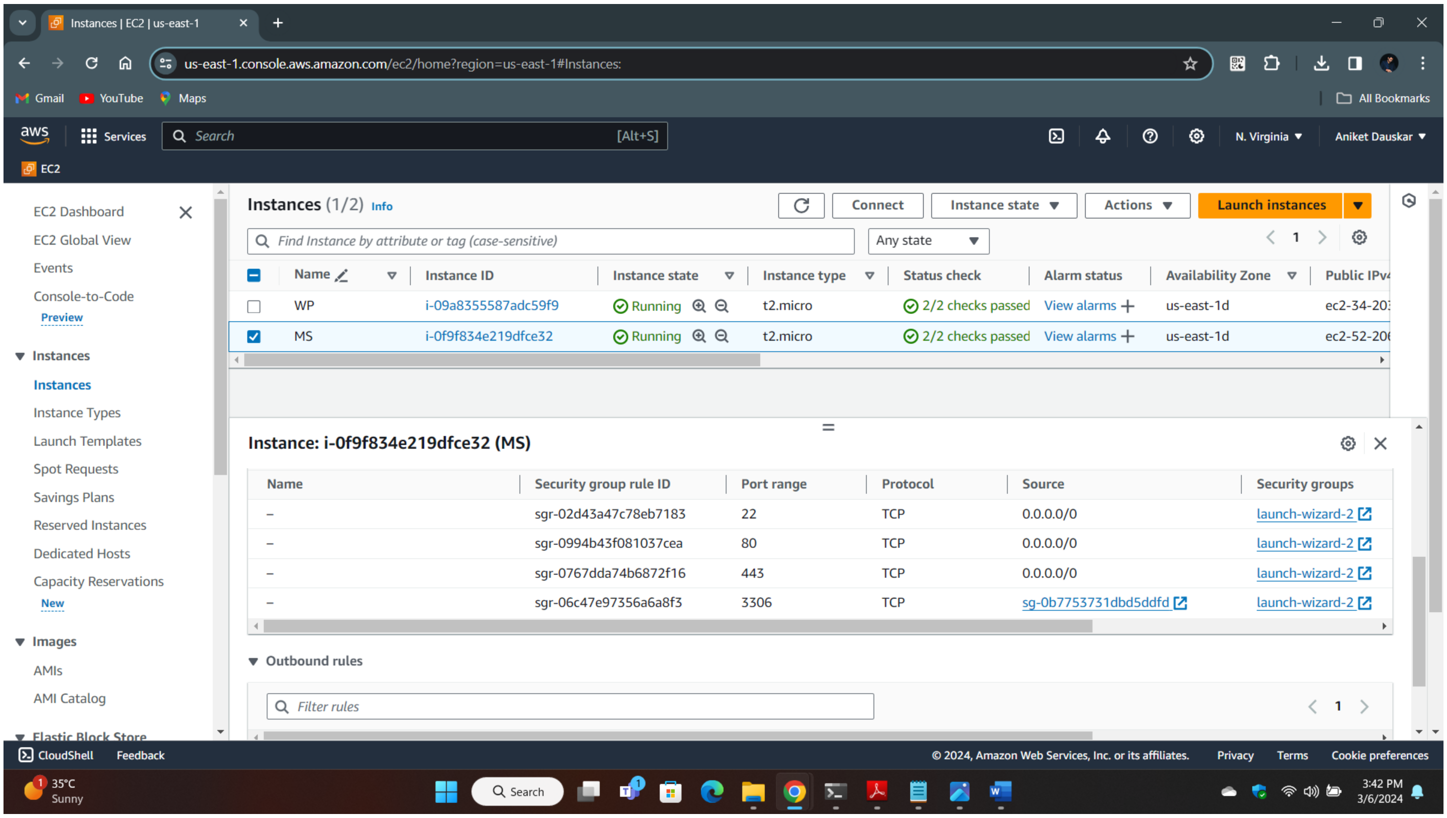
- Successfully deployed WordPress and MySQL on separate EC2 instances.
- Configured WordPress to use MySQL as its backend database.
- Established secure communication between the instances using security groups.
- Created a functional welcome page in WordPress to serve as the homepage of the website.

By following these steps, we have set up a scalable and efficient microservices architecture with WordPress and MySQL, ensuring optimal performance and security for the web application.



Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
WP	i-09a8355587adc59f9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-20
MS	i-0f9f834e219dfce32	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-52-20



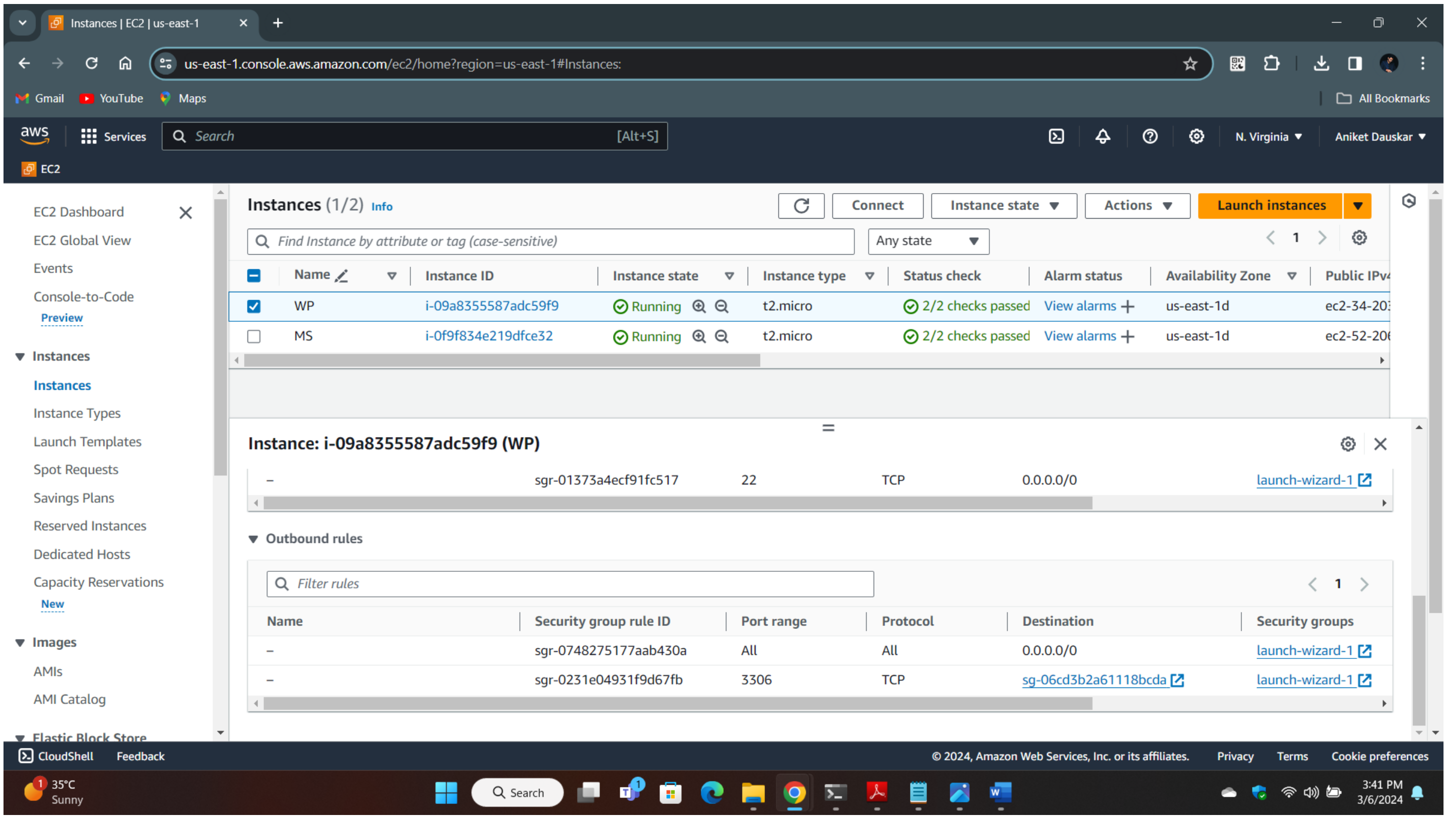
Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
□	WP	i-09a8355587adc59f9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-20
☑	MS	i-0f9f834e219dfce32	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-52-20

Instance: i-0f9f834e219dfce32 (MS)

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-02d43a47c78eb7183	22	TCP	0.0.0.0/0	launch-wizard-2
–	sgr-0994b43f081037cea	80	TCP	0.0.0.0/0	launch-wizard-2
–	sgr-0767dda74b6872f16	443	TCP	0.0.0.0/0	launch-wizard-2
–	sgr-06c47e97356a6a8f3	3306	TCP	sg-0b7753731dbd5ddfd	launch-wizard-2

Outbound rules



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	WP	i-09a8355587adc59f9	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-34-20
□	MS	i-0f9f834e219dfce32	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1d	ec2-52-20

Instance: i-09a8355587adc59f9 (WP)

–	sgr-01373a4ecf91fc517	22	TCP	0.0.0.0/0	launch-wizard-1

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
–	sgr-0748275177aab430a	All	All	0.0.0.0/0	launch-wizard-1
–	sgr-0231e04931f9d67fb	3306	TCP	sg-06cd3b2a61118bcda	launch-wizard-1