ScalableEmployeeManager Project Documentation

PuTTY and SSH

PuTTY is a widely used open-source SSH and Telnet client for Windows operating systems. It allows users to securely connect to remote systems like Linux-based servers or Amazon EC2 instances.

- **SSH (Secure Shell)**: SSH is a cryptographic network protocol that provides a secure way to access and manage remote systems. It encrypts data transmitted over the network, ensuring confidentiality and integrity.
- **Key Pair**: In AWS, when you create an EC2 instance, you generate a key pair consisting of a public key (`.pem` file) for AWS to install on the instance and a private key (`.pem` file) for you to download. PuTTY requires the private key to be converted into a `.ppk` format for authentication.

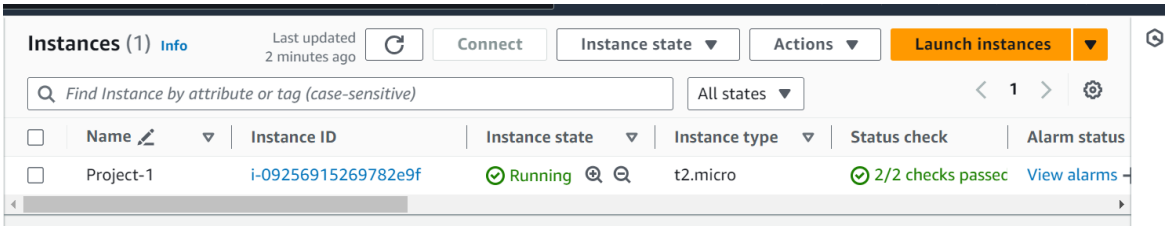
Amazon EC2 (Elastic Compute Cloud)

Amazon EC2 provides resizable compute capacity in the cloud. It allows you to quickly scale up or down to handle changes in requirements and traffic.

- **Instance Types**: EC2 offers various instance types optimized for different use cases, such as compute-optimized, memory-optimized, and storage-optimized instances. Choosing the right instance type depends on factors like CPU, memory, storage, and networking requirements.
- **Security Groups**: Security groups act as virtual firewalls that control inbound and outbound traffic for EC2 instances. They specify which traffic is allowed to reach the instance based on rules defined by protocol, port number, and source/destination IP address.

1. Preparing Your EC2 Instance:

- Sign in to the AWS Management Console and navigate to the EC2 service.
- Choose a region and proceed to launch a new EC2 instance. Select an appropriate Amazon Machine Image (AMI) and instance type, ensuring it fits your needs.
- Configure instance details such as storage, tags, security groups, and review before launching the instance. Create and download a key pair for SSH access.

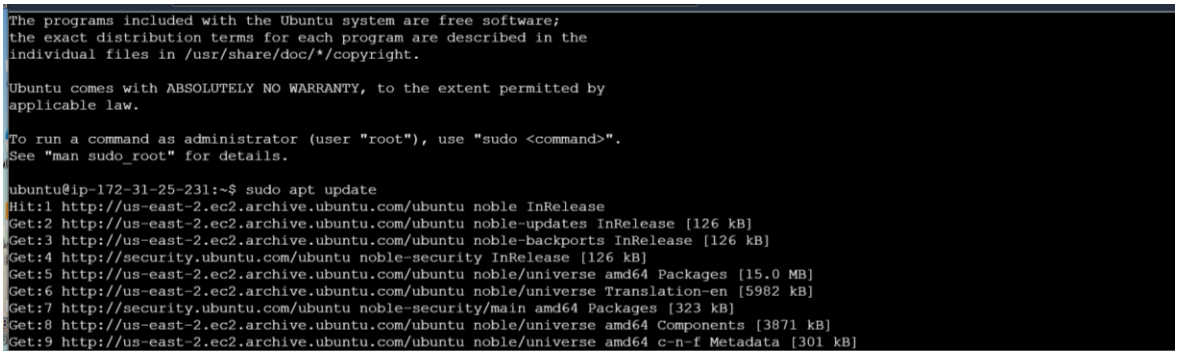


2. **Converting Key Pair to PuTTY Format:**

- After downloading the key pair in `.pem` format, convert it to `.ppk` format using PuTTYgen. Load the `.pem` file in PuTTYgen, then save the private key as a `.ppk` file.

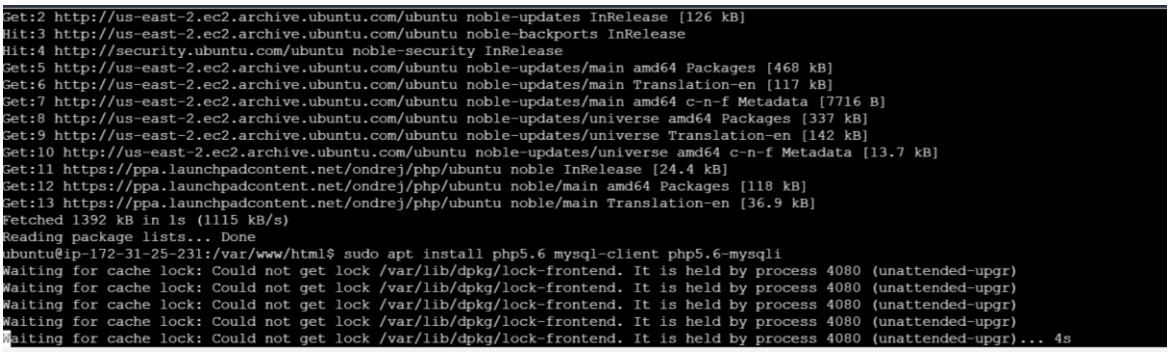
3. **Connecting to EC2 Instance Using PuTTY:**

- Open PuTTY and enter the instance's Public IP address in the Host Name field under Session.
- Expand SSH in the category list, click on Auth, and browse to select the `.ppk` file you saved.
- Click Open to establish the SSH connection. Login as the default user (`ubuntu` in case of Ubuntu instances).



4. **Updating and Installing Software:**

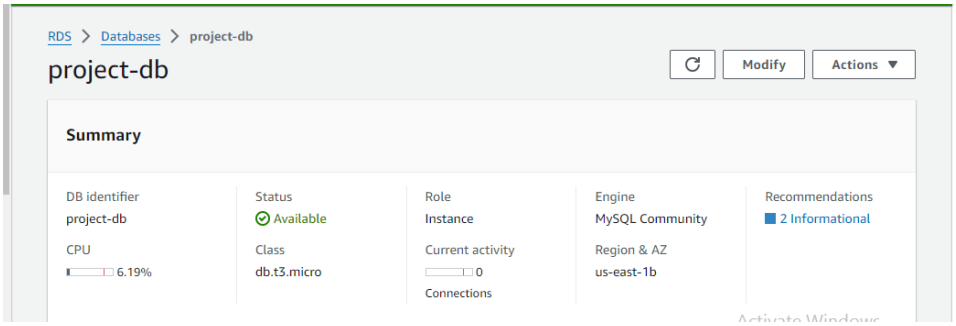
- Once connected via SSH, update the system using `sudo apt-get update`.
- Install required software packages like Apache2 and PHP using commands such as `sudo apt-get install apache2` and `sudo apt-get install php5.6`.



Amazon RDS (Relational Database Service)

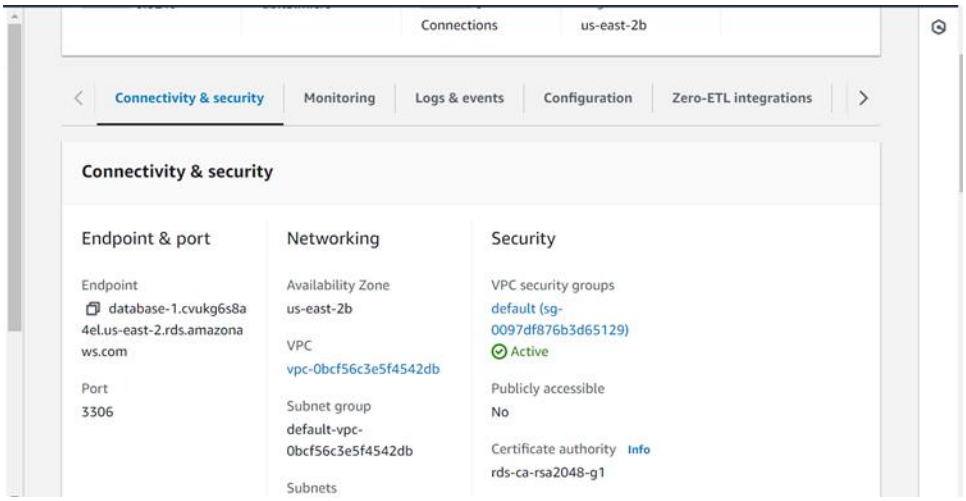
Amazon RDS is a managed relational database service that makes it easier to set up, operate, and scale a relational database in the cloud.

- **Database Engines**: RDS supports various database engines including MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora. Each engine has its strengths and is suitable for different types of applications.
- **Multi-AZ Deployments**: Multi-AZ deployments provide high availability by automatically replicating databases across multiple Availability Zones (AZs). In case of a failure, RDS automatically fails over to the standby instance to minimize downtime.



5. Connecting to RDS (Relational Database Service):

- In the AWS Management Console, navigate to RDS and create a MySQL database instance.
- Configure database details, including choosing appropriate instance classes and security group settings. Note down the endpoint and credentials for future use.



```
ubuntu@ip-172-31-82-232:/var/www/html$ sudo nano index.php
ubuntu@ip-172-31-82-232:/var/www/html$ mysql -h project-db.ctksiw2u20jf.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> create database intel;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| intel              |
```

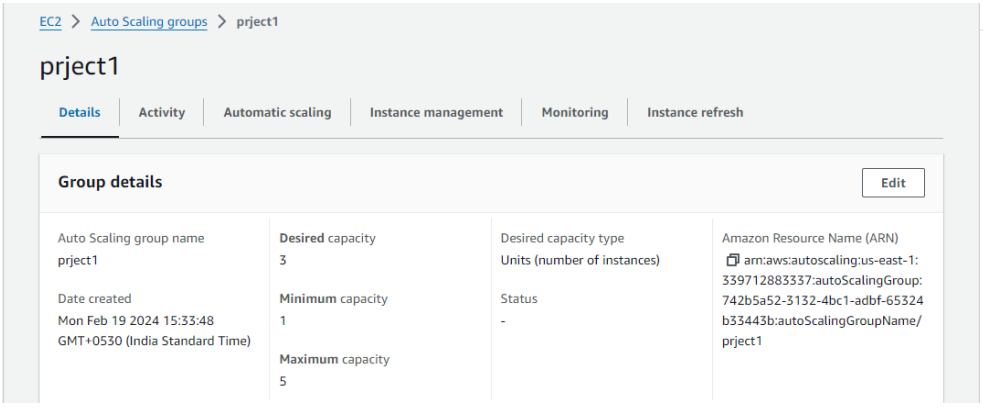
6. **Configuring Security Groups:**

- Adjust security group settings to allow inbound traffic from your EC2 instance to the RDS instance, ensuring MySQL/Aurora port access.

Auto Scaling and Load Balancing

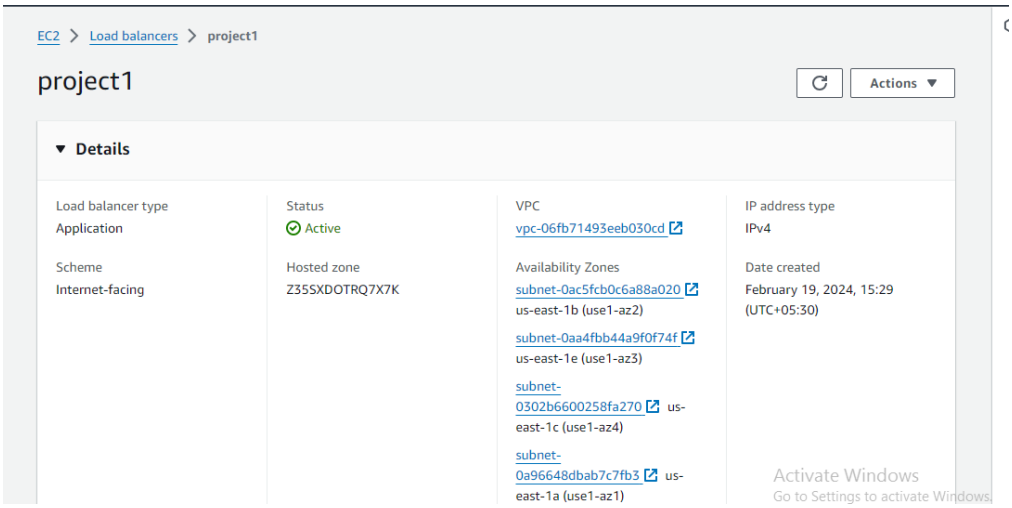
Auto Scaling and **Load Balancing** are essential for ensuring your application can handle varying levels of traffic and maintain high availability.

- **Auto Scaling**: Auto Scaling automatically adjusts the number of EC2 instances in a group based on traffic demand or other metrics defined by you. It helps ensure that you have the right amount of compute capacity at any given time without manual intervention.



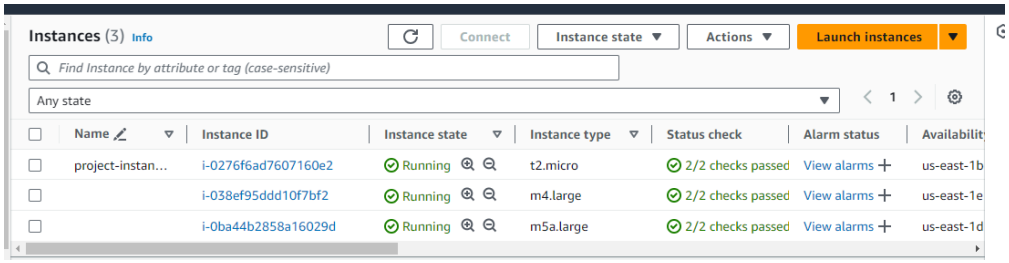
Load Balancing: Load Balancers distribute incoming application or network traffic across multiple targets, such as EC2 instances. This ensures no single instance becomes overwhelmed and helps achieve fault tolerance by routing traffic only to healthy • **Website Deployment:**

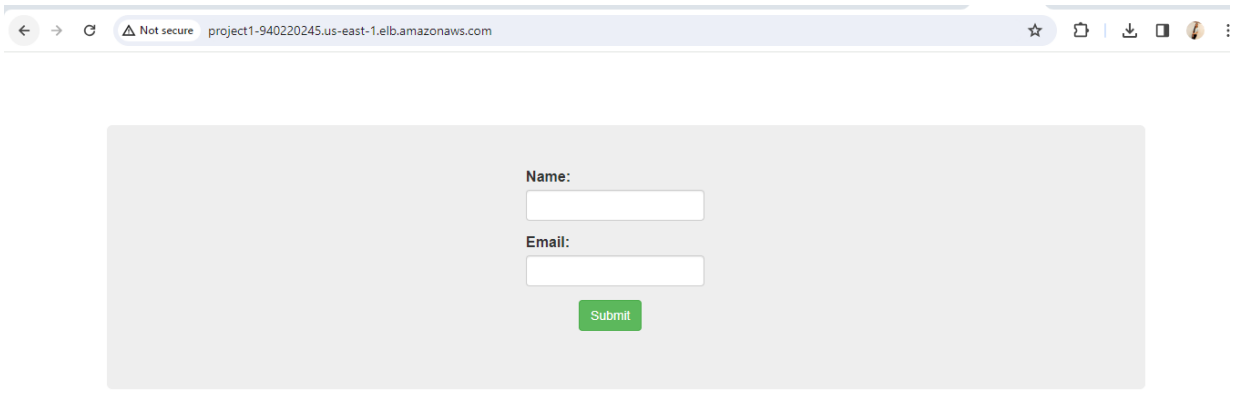
- Modify website configurations to use the RDS endpoint, username, password, and database name in your PHP files.
- Test the website functionality by accessing the EC2 instance's Public IP address in a web browser.

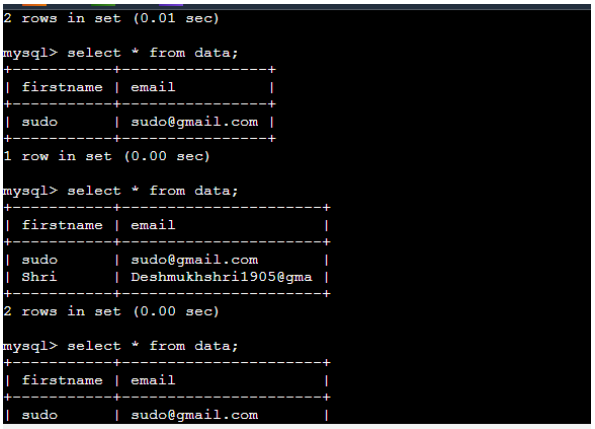


7. Implementing AutoScaling:

- Create an Amazon Machine Image (AMI) of your EC2 instance in the AWS Management Console.
- Configure Auto Scaling and set up a Classic Load Balancer to distribute incoming traffic among multiple instances of your application.







File Transfer with FileZilla

FileZilla is a popular open-source FTP (File Transfer Protocol) solution for transferring files between local systems and remote servers.

- **FTP vs. SFTP**: While FileZilla primarily supports FTP, for secure file transfer with AWS EC2 instances, it's recommended to use SFTP (SSH File Transfer Protocol) which provides encryption of both commands and data.

8. File Transfer with FileZilla:

- Install and open FileZilla on your local system. Enter the EC2 instance's endpoint as the hostname, and use the username `ubuntu` for authentication (no password needed).
- Transfer website files to the EC2 instance's `/home/ubuntu` directory using FileZilla.

Practical Application

- **Website Deployment**: Deploying a website on EC2 involves configuring web servers like Apache or Nginx, setting up databases, and ensuring security measures such as using HTTPS (SSL/TLS certificates) for secure communication.
- **Scalability**: Leveraging Auto Scaling and Load Balancing ensures your application remains responsive under varying workloads, providing a seamless user experience even during peak traffic periods.

9. **Website Deployment:**

- Modify website configurations to use the RDS endpoint, username, password, and database name in your PHP files.
- Test the website functionality by accessing the EC2 instance's Public IP address in a web browser.

By understanding these concepts and leveraging AWS services effectively, we have deployed robust, scalable, and secure applications on Amazon EC2 while optimizing performance and cost efficiency.