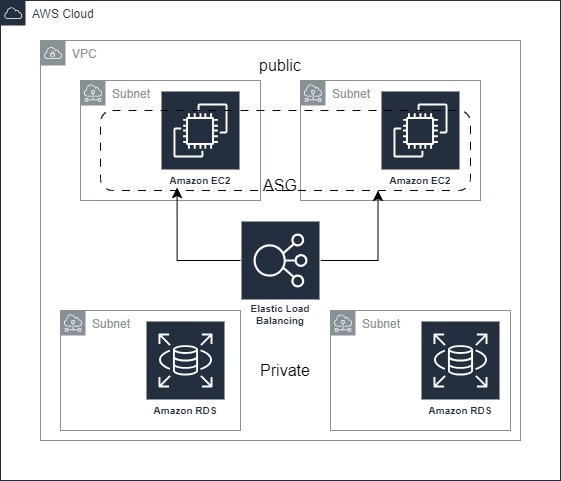
WordPress Application Deployment on AWS

Objective: Deploy and host a highly available WordPress application using AWS services such as EC2, RDS, Auto Scaling Groups (ASG), and VPC to ensure scalability and fault tolerance across multiple Availability Zones.

**The project involved the following components:**

* Amazon EC2: For web servers.
* Amazon RDS: For the MySQL database with automated backups.
* Auto Scaling Groups (ASG): For dynamic resource management.
* Amazon VPC: For network isolation and security.

# Architecture:



# Project Steps:

**Set Up VPC:**

* Create a Virtual Private Cloud (VPC) to logically isolate your AWS resources.
* Configure public and private subnets across multiple Availability Zones for high availability.

**Launch EC2 Instances:**

* Launch EC2 instances in public subnets to serve as the web servers for the WordPress application.
* Install and configure the necessary software (Apache, PHP, WordPress) on the EC2 instances.

**Configure RDS for WordPress Database:**

* Create an Amazon RDS instance using MySQL for the WordPress database.
* Place the RDS instance in private subnets to enhance security.
* Configure automated backups, snapshots, and Multi-AZ deployment for high availability and disaster recovery.

**Set Up Auto Scaling Groups (ASG):**

* Create an Auto Scaling Group to manage the EC2 instances.
* Configure scaling policies to automatically add or remove instances based on traffic load.
* Ensure instances are spread across multiple Availability Zones to maintain fault tolerance.

**Configure Load Balancer:**

* Set up an Elastic Load Balancer (ELB) to distribute incoming traffic across the EC2 instances.
* Configure health checks to monitor the status of the EC2 instances and ensure traffic is only sent to healthy instances.

**Set Up S3 for Media Storage:**

* Use Amazon S3 to store WordPress media files.
* Configure WordPress to offload media uploads to S3, reducing the load on the web servers.

**Implement Security Best Practices:**

* Set up security groups and network ACLs to control inbound and outbound traffic.
* Enable HTTPS using SSL/TLS certificates for secure communication.

**Monitor and Optimize:**

* Use Amazon CloudWatch to monitor the performance and health of the application.
* Set up alarms to notify of any issues or unusual activity.
* Optimize instance types and database configurations based on performance metrics.