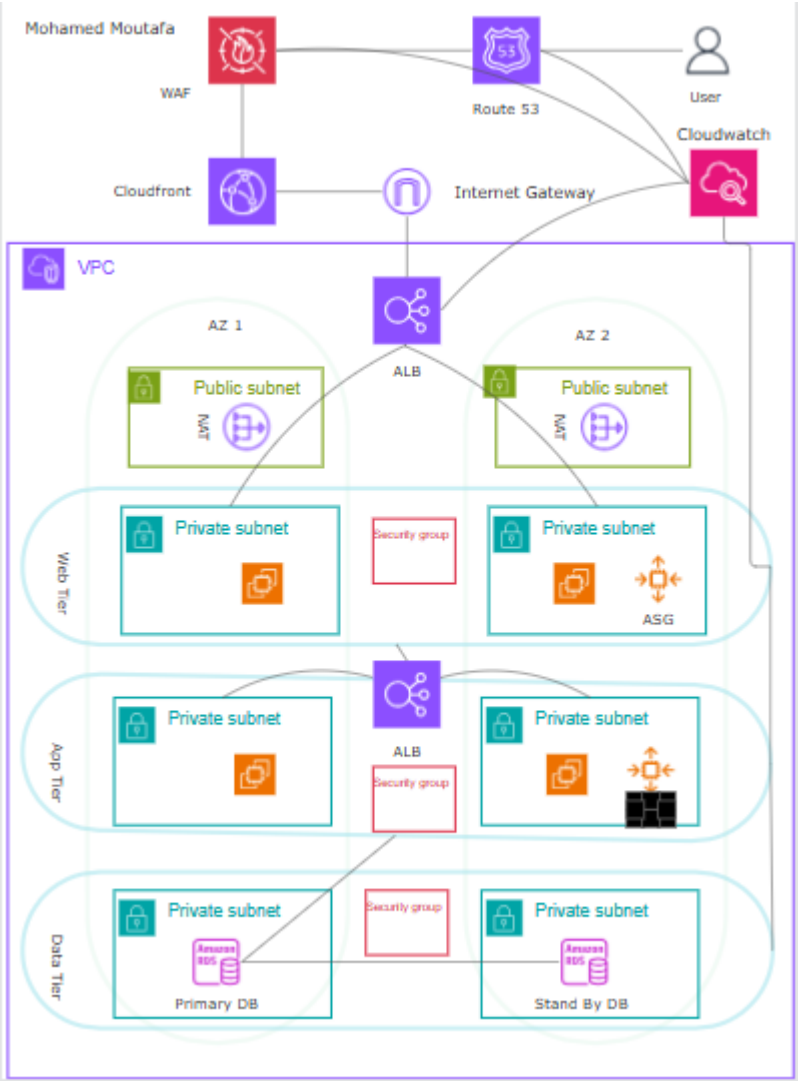
AWS Architecture Detailed Explanation

Introduction

This document provides a comprehensive explanation of the AWS architecture designed for hosting a highly available and secure application. The architecture consists of multiple AWS services integrated to ensure performance, security, scalability, and monitoring.



Architecture Overview

This architecture is built to ensure the system is always available, secure, and ready to scale as needed. It uses different AWS services that work together to create a seamless flow of data and protect the application from potential threats. The setup is divided into three main layers:

1. **Web Tier**
2. **Application Tier**
3. **Database Tier**

Network Layer

At the heart of everything is the **VPC (Virtual Private Cloud)**, which acts like a private, isolated network for the entire system. The VPC ensures that the services are only accessible to those who are supposed to access them.

To allow the system to communicate with the internet, there's an **Internet Gateway** that connects the VPC to the outside world. Whenever a user types the website's domain name, **Route 53** handles the DNS resolution, making sure the request goes to the correct resources.

To speed up content delivery globally, **CloudFront (CDN)** caches the content closer to users. At the same time, **WAF (Web Application Firewall)** acts as the first line of defense, blocking common attacks like SQL injection and cross-site scripting.

Web Tier

The **Web Tier** is the first point of contact for users. It uses an **Application Load Balancer (ALB)** to spread incoming traffic across multiple EC2 instances running in **Public Subnets** across two **Availability Zones (AZs)**. This setup makes sure that if one AZ goes down, the application stays online. Security is critical here. **Security Groups** are like virtual firewalls that only allow HTTP and HTTPS traffic to the web servers. Anything else is blocked.

Application Tier

Once the request reaches the web servers, they forward it to the **Application Tier**, which is hosted in **Private Subnets**. These subnets don't have direct internet access, adding an extra layer of security.

The backend application runs on EC2 instances managed by an **Auto Scaling Group (ASG)**. This group automatically adds or removes instances based on demand, ensuring the application performs well without wasting resources.

Communication between the Web and Application Tiers is locked down with **Security Groups**, allowing only necessary traffic.

Database Tier

The **Database Tier** stores all the application's data. It uses **Amazon RDS** with a **Primary DB** and a **Standby DB** in different Availability Zones. If the primary database fails, the standby takes over automatically.

Both databases are in **Private Subnets** with no internet access, and **Security Groups** ensure only the Application Tier can communicate with them. Data is encrypted both at rest and in transit to protect sensitive information.

Monitoring and Security

To keep everything running smoothly, **CloudWatch** continuously monitors system performance and logs any unusual activity. **IAM Roles** are used to give the right permissions to different services without hard-coding credentials.

Finally, **WAF** adds an extra layer of security, while encryption ensures that data remains safe both during transfer and when stored.

Conclusion

This architecture balances performance, security, and cost-efficiency. Every service plays a vital role, and they all work together to create a highly available, scalable, and secure environment. By following AWS best practices, the system can handle high demand, protect sensitive data, and recover quickly from failures.