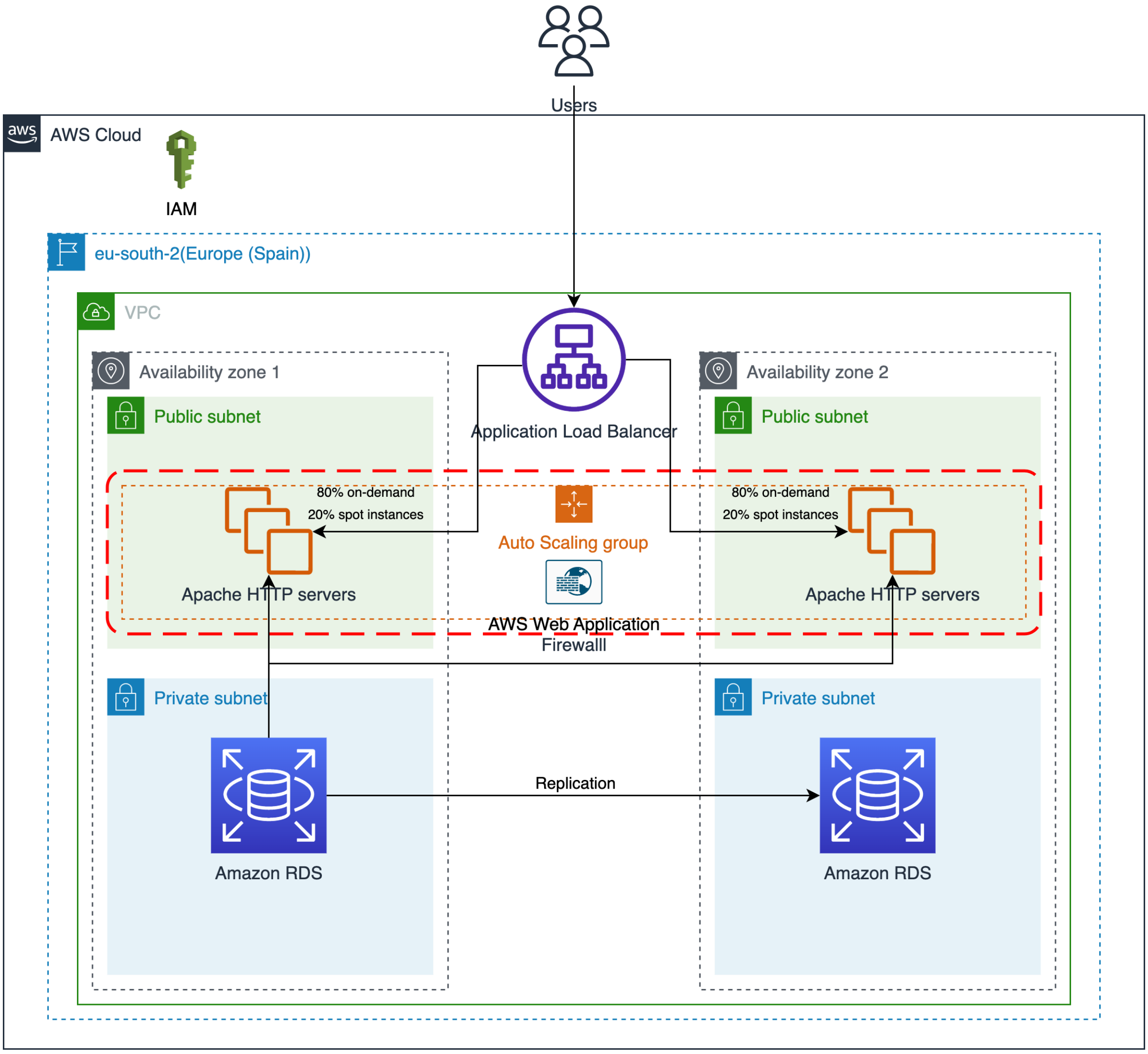
Pablo Ostos Bollmann

Cloud Solutions Architecture

Individual assignment 1

A. Architecture diagram represented in draw.io



B. Explanation of the main components and technical decisions

1. **AWS Cloud:** this will be a helpful solution for IE's problems since the on-demand delivery of IT resources provides several benefits for IE's case study. IE will be able to scale their we application to handle increased traffic and demand (Auto Scaling Group), by being present in several Availability Zones across a region redundancy and high availability will be ensured, AWS provides several security services to protect the application…

2. **Region and Availability Zones:** Assuming IE University is based in Madrid, Spain, with the aim of reducing latency, AWS eu-south-2 region (Spain) was chose. At least 2 Availability Zones have been created in the architecture to ensure high availability and redundancy (with the option of 3 AZs).

3. **Virtual Private Cloud (VPC):** with the aim of resource isolation this creates a private network environment in the cloud, which will help protect the web application and database from unwanted access and keeps the resources separated from other networks. Inside the VPC **multiple subnets** have been created, each associated with one Availability Zone, which will help organising resources, as well as providing security and access control. I chose private subnets for the RDS to enhance security and preventing exposure to the internet.

4. **Security:** To add more security mechanisms, an Amazon EC2 Security Group is created around the EC2 instances, to control inbound and outbound traffic to the instances. Also, the Web Application Firewall (WAF) is used for added security against web application threats.

5. **Elastic Load Balancing:** To distribute the incoming traffic among the different instances, an Application Load Balancer (ALB) is placed. The load balancer monitors the status of the target resources and, based on different rules, it routes the traffic to the resources in better health.

6. **Apache Web Server Instances:** Spread across different Availability Zones to ensure high availability and redundancy, EC2 instances running the Apache HTTP Server are launched.

7. **Amazon Relational Database Service (RDS):** As it was mentioned before, the database is place in a private subnet to reduce the risk of unauthorised access. Now, for PostgreSQL database, Amazon RDS ensures compatibility and an easy migration process, as well as allowing IE to continue using the same database engine. The deployment will be done in several Availability

Zones for automatic failover in case of a primary database instance failure, also automatic backups should be enabled.

8. **Auto Scaling Group:** to handle traffic spikes and ensuring high availability, auto scaling for the Apache instances should be configured. To solve this, an Auto Scaling Group is set around the EC2 instances. For the autoscaling configuration, scaling will be based on CPU utilisation, and other policies such as expected traffic and capacity should be defined to automatically scale up or down.

C. Monthly cost estimation (with justified assumptions).

***Taking into account that each user performs 100 requests a month and the number of users are 10000, the number of requests per month is 1000000.

Service	Monthly Cost	Observations
Amazon RDS for PostgreSQL db.m5.12xlarge	$7680,1	Multi-AZ, 3TB
AWS Web Application Firewall (WAF)	$7	Protect against SQL injection attacks
EC2 instances General purpose t4g.small	On-demand: (0.0168/Hora*6Horas*30dias*8instances)+(0.0168/Hora*24Horas*30dias*2instances) = 48.384/month	Considering 2 instances for rest time and 8 instances during peak time. 6 hours peak time. Historic discount for this machine type is 42% (28.063/month)
Total	**~$7735**	

D. Answer to the following questions:

I. How would you configure the autoscaling? What metric and value would you use? Why?

For IE University's case study, I would use the CPU utilisation metric for triggering the scaling actions. The reason behind this is that CPU utilisation is a good indicator of total workload and resource usage. Now, the specific CPU utilisation threshold would be around 70-80%, leaving some margin for unexpected workloads. Another threshold that can be set is the lower bound, as CPU utilisation drops below certain number, the autoscaling group can be reduced by removing instances.

II. Would we be able to meet all the requirements using only Spot instances? why?

No, using only spot instances would not be advisable in this situation since all requirements would not be met. To start with, a spot instance can be terminated with short notice and that means there is a risk of affecting the availability of the application. In other words, the requirement of guaranteed traffic handling would not be met.

III. Would you recommend the usage of Aurora instead of RDS? Why?

Both Amazon Aurora and RDS are viable options for hosting PostgreSQL database. On the one hand, Aurora might be a better option if IE expects high performance in the database, since it allows faster read and write performance compared to RDS. On the other hand, RDS allows IE to optimise costs based on workloads while Aurora tends to have a higher price. To answer the question, I would only advise IE to use Aurora if they prioritise high performance and are willing to add costs.

IV. Why do you think that your proposed architecture is resilient and scalable?

The proposed architecture is resilient and scalable due to many reasons.

Let's look at the resiliency part first. To start with, the architecture utilises several AZs within a region, which ensures redundancy by distributing the resources such as instances and Database replicas. To continue with, the databases are deployed in multiple AZs, providing high availability and automatic failover, this means that if there is a failure in the primary database, the system will switch to the replica in another AZ, ensuring continuous database operation. And last, but not least, the security measures taken to protect the system from malicious attacks. To conclude, the architecture can withstand failures in one AZ while maintaining service availability in others.

On the other side, scalability in the architecture is provided by services such as Auto Scaling Groups, which will enable the system to add resources or remove them according to the demand. Then to tie everything together, the Application Load Balancer will distribute the incoming traffic between the web server instances, allowing scalability and high availability.

V. Why have you decided to use that type of load balancer?

At the time of choosing the type of LB I was going to use I was doubting between the Elastic Load Balancer vs. The Application Load Balancer. However, once I got through the documentation of each Load Balancer, I decided to go with the Application Load Balancer since it operates at the application layer and provides advanced routing and load balancing capabilities. Also, the ALB integrates with the AWS Web Application Firewall

(WAF) for added security against web application threats. To conclude, the ALB offered more flexibility and control when managing web traffic.