

Project Documentation

1. Overview

The objective of this project is to design and deploy a highly available, scalable, and secure cloud-based environment for hosting a web application using AWS services. This architecture leverages **Infrastructure as Code (IaC)**, automatic monitoring, and remediation, ensuring disaster recovery (DR) is in place.

2. Architecture Overview

The environment will be designed with the following AWS services and components:

- **Compute:** Amazon EC2 (Elastic Compute Cloud)
- **Storage:** Amazon S3 (Simple Storage Service) for static content
- **Monitoring:** Amazon CloudWatch and AWS SNS for notifications
- **Networking:** Amazon VPC, Load Balancer, NAT Gateway
- **Automation:** Amazon EC2 autoscaling integrated by CloudWatch and SNS.
- **Security:** IAM roles and policies, Security Groups, HTTPS encryption
- **IaC Tool:** AWS CloudFormation

3. Infrastructure Components

3.1 Compute Layer

- **Amazon EC2 Instances:** Host the web application. Deployed in an Auto Scaling group across multiple Availability Zones (AZs) to ensure high availability.

3.2 Storage Layer

- **Amazon S3:** Store static content such as images, HTML files, and CSS. This offloads storage from the EC2 instances.

3.3 Network Layer

- **Virtual Private Cloud (VPC):** Create a custom VPC with public and private subnets. EC2 instances are deployed in private subnets, while Load Balancer and Nat Gateway sits in public subnets.
- **Application Load Balancer (ALB):** Directs HTTP and HTTPS traffic to the EC2 instances. Set up redirection from HTTP to HTTPS for secure access.

3.4 Security Layer

- **IAM Roles:** Grant EC2 instances and other AWS services appropriate access permissions.
- **Security Groups:** Define inbound/outbound traffic rules for EC2 and load balancers.

4. Monitoring and Automatic Remediation

4.1 Monitoring

- **Amazon CloudWatch:** Monitor CPU utilization, memory usage, and network traffic on EC2 instances. CloudWatch Alarms are configured to trigger notifications when specific thresholds are exceeded.
- **AWS SNS (Simple Notification Service):** Sends notifications via email or SMS if performance issues are detected or if an instance goes down.

4.2 Remediation

- **Automation:** Amazon EC2 autoscaling integrated by CloudWatch and SNS.

5. Disaster Recovery (DR)

- **Multi-AZ Deployment:** Ensure that the application is distributed across multiple AZs to handle failover scenarios.
- **S3 Backup:** Use S3 versioning and cross-region replication to ensure that static content is replicated across regions.

6. Additional Features

6.1 HTTP to HTTPS Redirection

- Configure the **Application Load Balancer** to redirect all HTTP traffic to HTTPS, ensuring all communications are encrypted.

6.2 Mount S3 on EC2

- **SSH:** Connect to the EC2 instance using SSH, install **S3FS**, and mount the S3 bucket to a local directory with the command `s3fs <bucket-name> <mount-point>`.
- **Session Manager:** Use AWS Systems Manager **Session Manager** to access the EC2 instance, install **S3FS**, and execute the same mounting command without the need for direct SSH access.

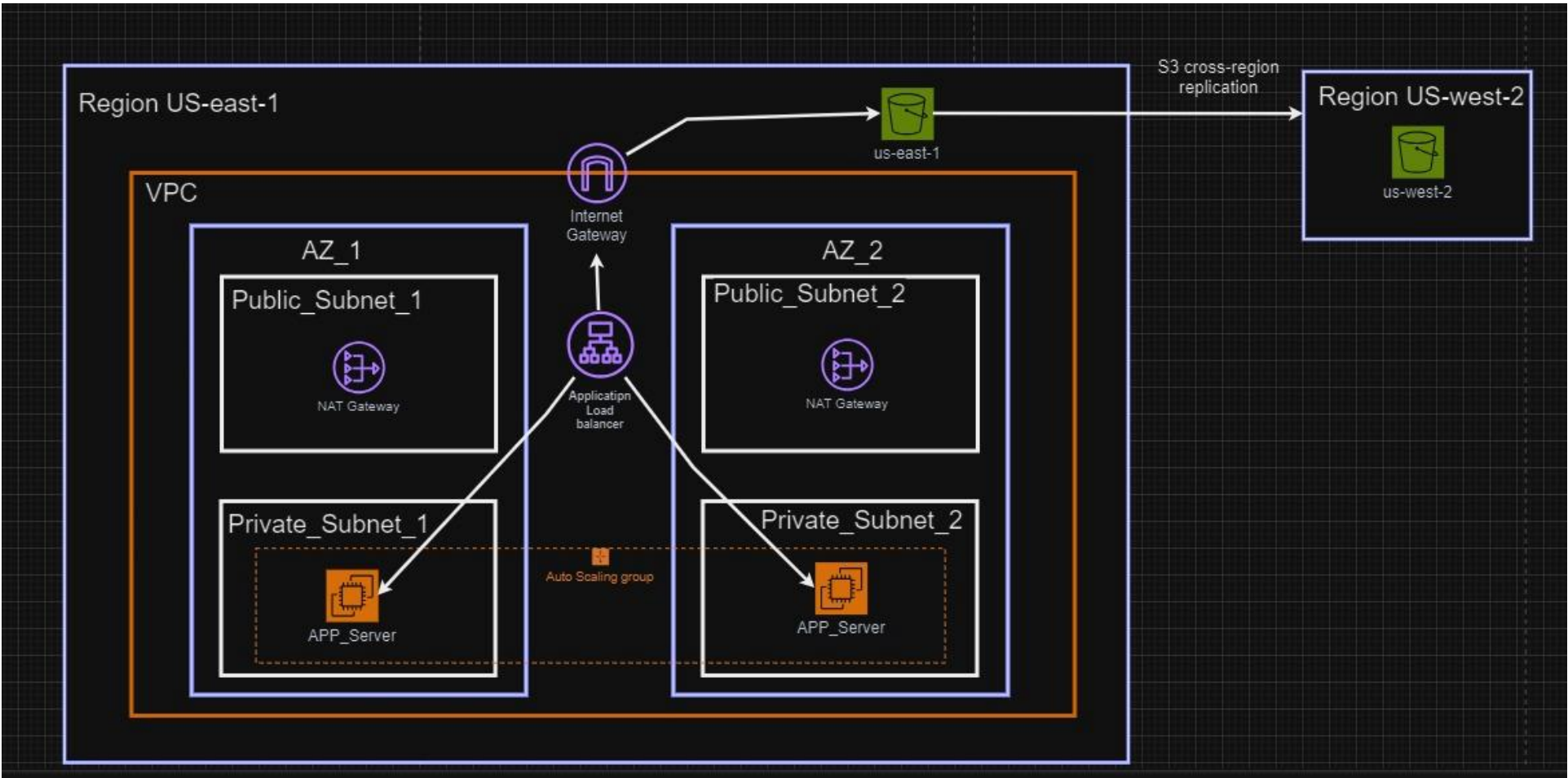
6.3 API Gateway to Access Images in S3

- **Amazon API Gateway:** Create an API endpoint that allows users to fetch images from S3. API Gateway integrates with S3, ensuring that images are retrieved securely via signed URLs.

7. Infrastructure as Code (IaC)

Use **AWS CloudFormation** to automate the provisioning of all AWS resources. This ensures consistency, scalability, and easier management of the environment.

8. Architecture (Draw.io):



9. Conclusion

This architecture ensures that the web application is secure, highly available, and scalable. Monitoring, disaster recovery, and automatic remediation mechanisms are built in to maintain operational continuity.