

Scalable Web Application Architecture on AWS

Table of Contents

- Introduction
- Project Overview
- Architecture Overview
- AWS Services Used
- Implementation Steps
- Backend Services Integration
- Deployment Process
- Conclusion

Introduction

This document provides a comprehensive overview of the implementation of a scalable web application architecture on AWS. The project aims to leverage AWS services such as Auto Scaling, Load Balancing, and backend service integration to ensure high availability, scalability, and reliability of the web application.

Project Overview

The project involves designing and implementing a web application architecture on AWS to support a scalable and resilient application. Key objectives include:

- Implementing load balancing to distribute traffic across multiple instances.
- Setting up Auto Scaling to dynamically adjust the number of instances based on demand.
- Integrating backend services such as Memcached, RabbitMQ, and MySQL for caching, messaging, and data storage

Architecture Overview

The architecture consists of:

- Application Load Balancer to distribute incoming traffic.
- Auto Scaling group to manage instances dynamically.
- Backend services including Memcached, RabbitMQ, and MySQL
- Integration with Route 53 for DNS management
- SSL certificate from AWS Certificate Manager for secure communication

AWS Services Used

- EC2: Virtual servers for running the web application instances.
- ELB: Application Load Balancer for distributing traffic.
- Auto Scaling: Dynamically adjusting the number of instances based on demand.
- S3: Storing deployment artifacts and other resources.
- Route 53: DNS management for domain registration.
- AWS Certificate Manager: SSL certificate for secure communication.
- Memcached: In-memory caching for improved performance.
- RabbitMQ: Message broker for communication between components.
- MySQL: Backend database for storing application data.

Implementation Steps

1. Set up AWS account and configure IAM roles and permissions.
2. Launch EC2 instances for the web application and backend services.
3. Configure security groups, IAM roles, and instance profiles.
4. Create an Application Load Balancer and set up target groups.
5. Configure Auto Scaling policies and scaling rules.
6. Integrate backend services with the web application instances.
7. Configure Route 53 for domain registration and SSL certificate.
8. Deploy application artifacts to S3 bucket.
9. Test and monitor the deployment.

Backend Services Integration

- Memcached: Used for caching frequently accessed data to improve performance.
- RabbitMQ: Used as a message broker for asynchronous communication between components.
- MySQL: Used as the backend database for storing application data.

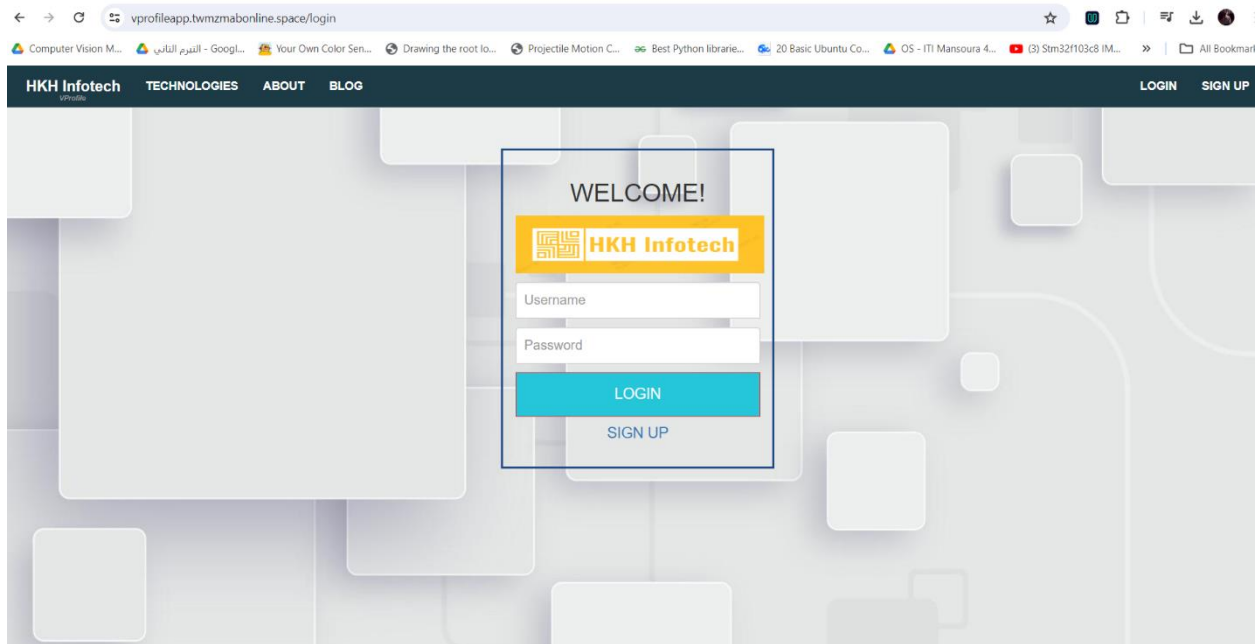
Deployment Process

1. Build application artifacts using Maven.
2. Upload artifacts to S3 bucket.
3. Monitor deployment status and performance metrics using CloudWatch.
4. Test application functionality and scalability under various load conditions.

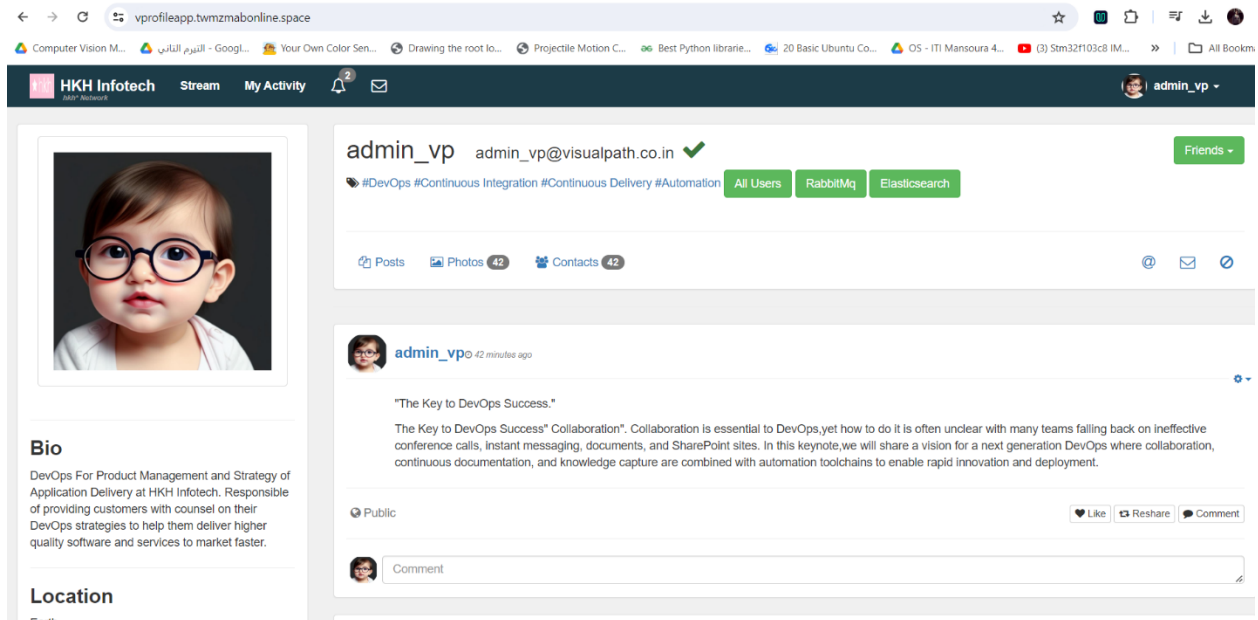
Conclusion

The implementation of the scalable web application architecture on AWS has been successful in achieving the project objectives of high availability, scalability, and reliability. The use of AWS services such as Auto Scaling, Load Balancing, and backend service integration has enabled the deployment of a resilient and efficient web application.

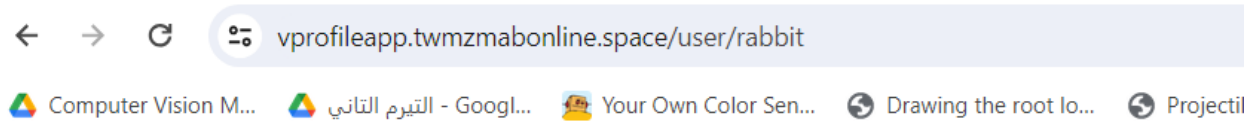
Web App Login Page



After logging in, the application is successfully connecting to the database.



Testing RabbitMQ functionality.



Rabbitmq initiated

Generated 2 Connections

6 Chanel 1 Exchage and 2 Que

Testing Mem-cache functionality

A screenshot of a web application interface. The header is dark blue with white text for 'VISUAL PATH' and navigation links: 'TECHNOLOGIES', 'ABOUT', 'CONTACT', 'BLOG', and 'LOG'. A red message box at the top center contains the text '[Data is From DB and Data Inserted In Cache !!]' and a 'Back' button. Below the message, there are two tables. The first table, 'User Primary Details', has columns: Id, Name, Father's Name, Mother's Name, Email, and Phone Number. The second table, 'User Extra Details', has columns: Date Of Birth, Gender, Marital Status, Permanent Address, Temporary Address, Primary Occupation, Secondary Occupation, Skills, Secondary PhoneNumber, Nationality, Language, and Working Experience.

Id	Name	Father's Name	Mother's Name	Email	Phone Number
4	Hibo Prince	Abara	Queen	hibo.prince@gmail.com	9146389863

Date Of Birth	Gender	Marital Status	Permanent Address	Temporary Address	Primary Occupation	Secondary Occupation	Skills	Secondary PhoneNumber	Nationality	Language	Working Experience
6/09/2000	male	unMarried	Electronic City,UAE	Electronic City,UAE	Tester	Freelancing	Python PHP	9146389871	Indian	hindi	3