Project: 1

DESCRIPTION

You are working as a database administrator for an IT firm. You have been asked to create a new database instance on AWS cloud and connect it with the employee management portal hosted on a web server.

**Background of the problem statement:**

Your organization wants to deploy a new multi-tier application. The application will take live inputs from the employees, and it will be hosted on a web server running on the AWS cloud.

The development team has asked you to set up the web server and configure it to scale automatically in cases of a traffic surge, to make the application highly available. They have also asked you to take the inputs from the employees and store them securely in the database.

Solution:

1. VPC (to process the Infrastructure in secure manner).
2. Firstly, Created VPC
3. Then created 1 Public and 2 Private Subnets(1 will be laterly used in RDS),
4. Created 2 Route Tables for private & public subnets and associated them from Subnet association.
5. Created Internet Gateway and have attached to VPC.
6. Created NAT Gateway (1 in public subnet), then have routed them from Route table as follows: Public routed to IGW & Private routed to NAT Gateway.
7. Then enabled the DNS hostname from VPC> Actions>VPC Settings.

Note: Screenshots for the above is attached to Screenshot tab by name: Project1-VPC

1. EC2 instance-webserver
2. Before creating EC2, created the 2 Security Groups for EC2 (Webserver) by allowing inbound traffic as HTTP & SSH & for RDS (DB) by allowing inbound traffic as MYSQL/Aurora.
3. Created EC2 Instance by select the network setting we created for VPC & Security group.
4. Then tested the EC2 Instance connecting to Linux instance by using Putty.
5. Firstly, converted the key file that i created for instance, from .pem to .ppk file by using Putty gen.
6. Then logged in to Putty by mentioning the Hostname (Public IP V4 address of EC2 instance) & configuring the .pem file. (Connection>SSH & upload the file)>Open. Lastly, enter the login as : ec2-user
7. Installed & tested the Apache Server is working successfully. Entered the public Domain Name System (DNS) name of EC2 instance in the address bar of a web browser.

Note: Screenshots for the above is attached to Screenshot tab by name: Project1-EC2

Commands for point D attached to Source Code -Project1- connect ec2 & install apache web server

1. Database –RDS
2. DB needs a subnet group, created the group by selecting the VPC & Private Subnets we created earlier.
3. Then, created the RDS database, selecting the VPC & subnets group by naming initial database name as: Project1SamplePage and choosing the required options.
4. Connected the Web Server to DB instance by giving command on CLI.
5. Verified the web server successfully connects to DB instance by opening a web browser and browsing to http://ec2-3-239-94-67.us-west-2.compute.amazonaws.com/ Project1SamplePage.php.

Note: Screenshots for the above is attached to Screenshot tab by name: Project1-RDS

Commands for point C attached to Source Code-Project1-Connected Apache web server to DB instance

1. Autoscaling & healthcheck using Route 53
2. Created EC2 webserver for Autoscaling and notification
3. Created AMI of EC2 instance.
4. Created Launch Template by adding user data:

#!/bin/bash

yum update -y

amazon-linux-extras install -y lamp-mariadb10.2-php7.2

yum install -y httpd

cd/var/www/html

echo>Project1SamplePage.php

systemctl start httpd

systemctl enable httpd

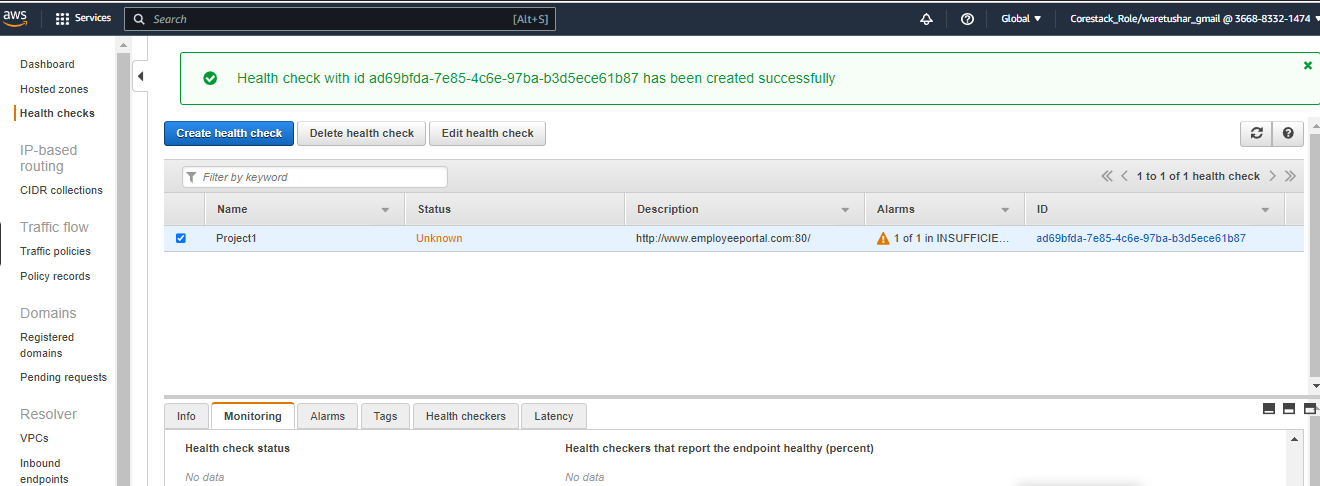
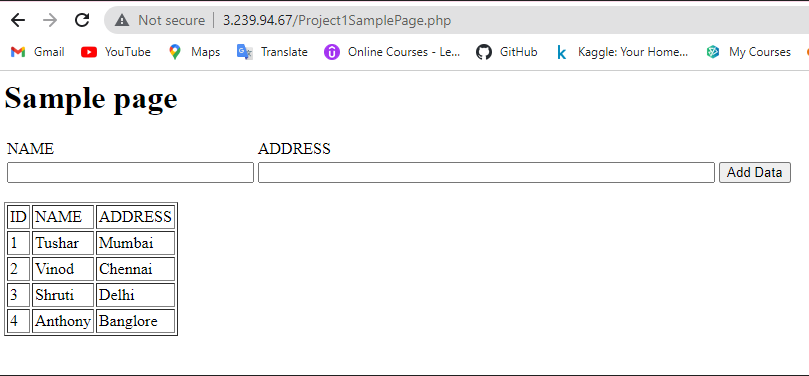
1. Then Created Autoscaling group by applying the exiting details we created for VPC & subnets. Also attaching the New Load Balancer.

1 Scaling Policy, 1 Topic, 1 Subscription, 1 Notification, 1 Load Balancer, 1 Target Group, 1 Listener created successfully. 1 Target group has been attached to ASG and Group metrics collection is enabled.

1. Lastly created the Health check using Route 53 by giving domain name as [**www.employeeportal.com**](http://www.employeeportal.com)

Note: Screenshots for the above is attached to Screenshot tab by name: Project1-Autoscaling

Final Output:



Thank You!