Troubleshooting Task

By Keerthi Konda and Omer Ali Alshiekeh

First step we undertook was to check if the EC2 instance was connecting to the internet. Upon checking, we noticed that the EC2 instance did not connect. To troubleshoot the issue we investigated the components that could be responsible for the issue. Upon inspection, we realised that the security group attached to the ELB did not have an inbound rule to allow traffic to the correct route, as well as the security group attached to the EC2 instance so we added the inbound rules for those security groups. Subsequently, the load balancer was not in the same availability zone as the EC2 instance. The load balancer was attached to a public subnet in us-east1b, whereas the EC2 instance was in availability zone us-east1a. Once the load balancer was in the same availability zone as the instance, we ran a health check to confirm whether it was working, which it was. To check whether the changes we made worked, we went back to the management console and navigated to the CloudFormation section and clicked on the customers CloudFormation template. We navigated to the Outputs panel and clicked the link for the LoadBalancerDNSName and the website displayed correctly now, when it did not before. Furthermore, we migrated the existing Classic Load Balancer into an Application Load balancer because ALB are newer generations and Application load balancer serves as the single point of contact for clients. The load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. Also, the Application Load Balancer can address more complex load-balancing needs by managing traffic at the application level (HTTP, HTTPS).

To summarise the troubleshooting steps:

* Configure inbound rules for the instance and ELB security groups to allow traffic to HTTP port 80 from the internet.
* Change the Availability Zone of the Load Balancer to match that of the EC2 instance.
* Run a health check to ensure the instance is configured and running fine.
* Migrate the Classic Load Balancer to the new generation Application Load Balancer.
* Create a Target Group and register the Target Group to the Load Balancer.
* Confirm that the website is running by clicking on the LoadBalancerDNSName link in CloudFormation under Outputs.

Diagram

Description automatically generated

Fig.1

As shown in Fig.1, this is the current architectural design of the CloudFormation template provided by the customer.

There are alternative architectural changes that could be implemented to the customers template to ensure higher security, availability, scalability and reliability. We would advise to leave Classic Load balancers behind in the future as they are not recommended by AWS. Instead, use an Application Load Balancer for their design because it distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. This increases the availability of your application. Also, the Application Load Balancer can address more complex load-balancing needs. For higher availability, we recommend launching the EC2 instances in multiple availability zones to guarantee higher availability in the case of one availability zone failing. Additionally, we would advise to couple the design with Auto Scaling Groups in the future in the case of the application becoming more successful and resulting in an increase of demand on the application. The auto scaling groups would scale up or down automatically in regard to the amount of requests received. This would ensure higher scalability. To increase the security of the application, we would advise to implement NAT Gateways. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances. This would avoid having to go directly through the internet and add an extra layer of security. Below we have included an architectural design of what it would look like.

Diagram

Description automatically generated

Fig.2

Summary:

To summarise, the customers application did not run at first because the security group inbound rules for the instance and the Load Balancer were not configured correctly. The customer also did not ensure that the Load balancer and the EC2 instance were in the same Availability Zone. After troubleshooting those issues, the LoadBalancerDNSName link started working. Furthermore, to improve the customers application certain recommendations were given to strengthen the security, availability, scalability and reliability of the application. The recommendations given to the customer were to avoid using a Classic Load Balancer and instead opt for an Application Load Balancer as the latter is the new generation of Load Balancers and Classic Loas Balancers are no longer recommended to use by AWS. Additionally, recommendations to implement Auto Scaling Groups to increase scalability for the future in case the application gains popularity to manage the increased workload, as well as adding NAT Gateways to add extra security by avoiding having their application come in contact with the internet directly.