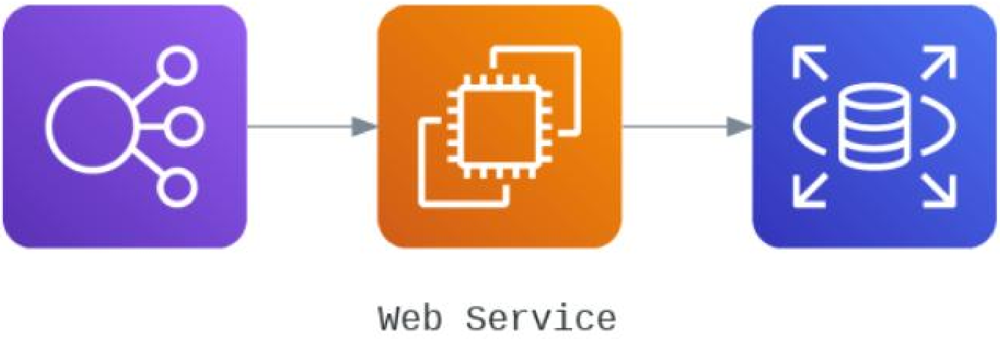
Assignment 6

Ques:Complete the below task:

1. Explain the below AWS Architecture

2. Implement the same in the AWS(only do a proper connection between service)

NOTE:Submission can be done by sharing the properscreenshots of implementation and docfor explanation!



In this architecture, the Load Balancer distributes incoming traffic to multiple EC2 instances that are running your application. The EC2 instances are responsible for serving your application to the end user. The RDS instance is used as a database server for your application.

The Load Balancer monitors the health of the EC2 instances and only sends traffic to instances that are healthy. This ensures that if one EC2 instance fails, the Load Balancer will automatically route traffic to other healthy instances. Additionally, the Load Balancer can automatically scale the number of EC2 instances up or down based on traffic patterns.

1. Create an RDS Instance:

- Go to the RDS console and create a new database instance.

- Choose a database engine and instance size, and configure the settings according to your

needs.

- Make sure to note the endpoint and credentials for the RDS instance, as we will need them

later.

2. Create an EC2 Instance:

- Go to the EC2 console and launch a new EC2 instance.

- Choose an Amazon Machine Image (AMI) that includes a web server, such as Amazon Linux.

- Configure the instance settings according to your needs.
- Make sure to add a security group that allows inbound traffic on port 80.

3. Install and Configure Web Server:

- Once the EC2 instance is running, connect to it via SSH.
- Install a web server, such as Apache or Nginx, and configure it to serve your application.
- Test that the web server is serving pages correctly.

4. Create a Load Balancer:

- Go to the EC2 console and create a new Application Load Balancer.
- Configure the Load Balancer settings according to your needs, including adding the EC2 instance we created in step 2 as a target.
- Make sure to add a security group that allows inbound traffic on port 80.

5. Configure Load Balancer Listener:

- Configure the Load Balancer listener to forward incoming traffic to the EC2 instance.
- Test that the Load Balancer is forwarding traffic correctly.

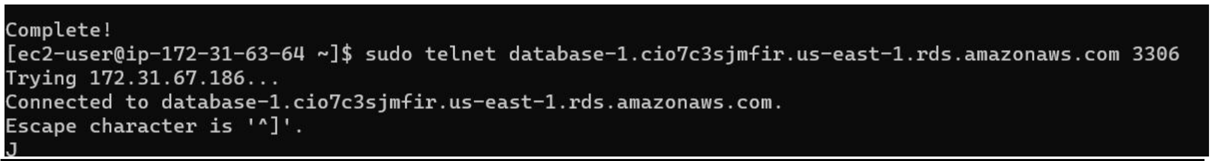
6. Update Application Settings to use RDS:

- Update the application code on the EC2 instance to use the RDS instance as its database.
- Use the endpoint and credentials from step 1 to connect to the RDS instance.

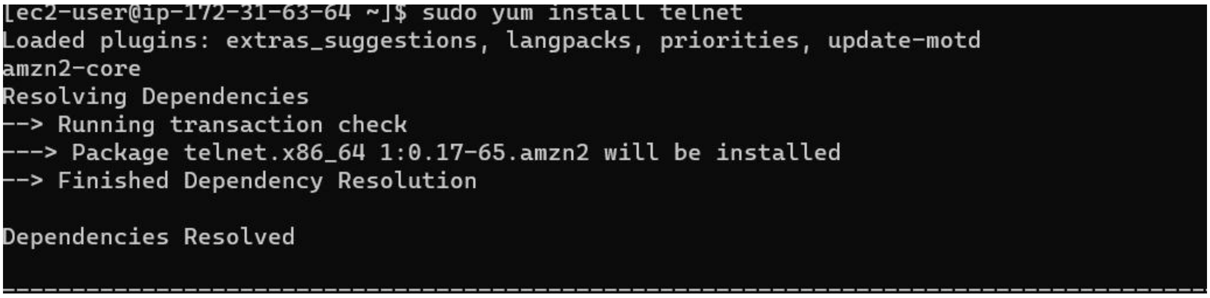
7. Test the Architecture:

- Test that the application is working correctly by accessing it through the Load Balancer's DNS name.
- Test that the application is correctly retrieving and storing data from the RDS instance.

By using this architecture, you can improve the availability and scalability of your application while also ensuring that your database is properly managed by RDS.



```
Complete!
[ec2-user@ip-172-31-63-64 ~]$ sudo telnet database-1.cio7c3sjmfir.us-east-1.rds.amazonaws.com 3306
Trying 172.31.67.186...
Connected to database-1.cio7c3sjmfir.us-east-1.rds.amazonaws.com.
Escape character is '^]'.
J
```



```
[ec2-user@ip-172-31-63-64 ~]$ sudo yum install telnet
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
---> Package telnet.x86_64 1:0.17-65.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

==========================================================================================
```