

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 proper screenshots of implementation and doc for explanation!



Create ELB to route traffic to EC2 instances and Connect to AWS RDS from AWS EC2

Elastic Load Balancer-

Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple targets such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. It monitors the health of its registered targets, and routes traffic only to the healthy targets.

EC2 > Load balancers

Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter by property or value

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
DemoELB	DemoELB-2019349693.us-east-1.elb.amazonaws.com	Active	vpc-09878b2678e2329fe	2 Availability Zones	application	December 21, 2022, 01:02 (UTC+05:30)

EC2 > Target groups

Target groups (1)

Search or filter target groups

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
DemoTG	arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/DemoTG/vpc-09878b2678e2329fe	80	HTTP	Instance	DemoELB	vpc-09878b2678e2329fe

EC2 > Target groups > DemoTG

DemoTG

Actions

Details

arn:aws:elasticloadbalancing:us-east-1:654568523680:targetgroup/DemoTG/902298a46c1ca3c

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-09878b2678e2329fe
IP address type IPv4	Load balancer DemoELB		

Total targets 2	Healthy 2	Unhealthy 0	Unused 0	Initial 0	Draining 0
--------------------	--------------	----------------	-------------	--------------	---------------

Targets Monitoring Health checks Attributes Tags

Registered targets (2)

Filter resources by property or value

Instance ID	Name	Port	Zone	Health status	Health status details
i-0dcea57add81c26b9	DemoEC2_2	80	us-east-1b	healthy	
i-01ae85bca49597d83	DemoEC2_1	80	us-east-1a	healthy	

EC2 Instance-

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. It also allows the user to configure their instances as per their requirements i.e. allocate the RAM, ROM, and storage according to the need of the current task. EC2 offers security, reliability, high-performance and cost-effective infrastructure so as to meet the demanding business needs.

Instances (1/2) Info

Find instance by attribute or tag (case-sensitive)

Instance state: [running](#) Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input checked="" type="checkbox"/> DemoEC2_1	i-01ae85bca49597d83	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-196-250-219.co...	54.196.250.219	-
<input type="checkbox"/> DemoEC2_2	i-0dcea57add81c26b9	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-44-198-189-181.co...	44.198.189.181	-

EC2 Management Console x EC2 Management Console x Target groups | EC2 Managemen x Target groups | EC2 Managemen x demoelb-2019349693.us-east-1.x

Not secure | demoelb-2019349693.us-east-1.elb.amazonaws.com

Gmail YouTube namaste-javascript... PDI SRE Interview Prep... TCS Alumni Portal Striver's SDE Sheet... Docker Cheetsheet... Linux CSA Exercise - Create

Hello World from ip-172-31-45-63.ec2.internal

EC2 Management Console x EC2 Management Console x Target groups | EC2 Managemen x Target groups | EC2 Managemen x demoelb-2019349693.us-east-1.x

Not secure | demoelb-2019349693.us-east-1.elb.amazonaws.com

Gmail YouTube namaste-javascript... PDI SRE Interview Prep... TCS Alumni Portal Striver's SDE Sheet... Docker Cheetsheet... Linux CSA Exercise - Create

Hello World from ip-172-31-4-139.ec2.internal

RDS Database-

Amazon Relational Database Service (RDS) is a managed SQL database service provided by Amazon Web Services (AWS). It's a managed DB service for DB use SQL as a query language. RDS is a managed service provide automated provisioning, OS patching, continuous backup and restore to specific time stamp. Its increase read performance by implementing read replicas. Perform disaster recovery using Multi-AZ setup.

RDS > Databases

Consider creating a Blue/Green Deployment to minimize downtime during upgrades

You may want to consider using Amazon RDS Blue/Green Deployments and minimize your downtime during upgrades. A Blue/Green Deployment provides a staging environment for changes to production databases. [RDS User Guide](#)

[Guide](#) [Aurora User Guide](#)

Databases

Group resources

Modify

Actions

Restore from S3

Create database

Filter by databases

< 1 >

DB identifier

▲

Role

Engine

Region & AZ

Size

Status

CPU

Current activity

Maintenance

VPC

database-1

Instance

MariaDB

us-east-1a

db.t3.micro

Available

1.23%

0 Connections

none

vpc-09878

EC2 > Security Groups > sg-044042ae2ec96ea1f - VPC SG

sg-044042ae2ec96ea1f - VPC SG

Actions

Details

Security group name

VPC SG

Security group ID

sg-044042ae2ec96ea1f

Description

Created by RDS management console

VPC ID

vpc-09878b2678e2329fe

Owner

654568523680

Inbound rules count

1 Permission entry

Outbound rules count

1 Permission entry

Inbound rules

Outbound rules

Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (1/1)

Manage tags

Edit inbound rules

Filter security group rules

< 1 >

Name

Security group rule...

IP version

Type

Protocol

Port range

Source

Description

-

sgr-07196940e2a5ed8...

IPv4

MySQL/Aurora

TCP

3306

103.170.177.103/32

-

RDS > Databases > database-1

database-1

Modify

Actions

Summary

DB identifier

database-1

CPU

1.23%

Status

Available

Class

db.t3.micro

Role

Instance

Current activity

0 Connections

Engine

MariaDB

Region & AZ

us-east-1a

Connectivity & security

Monitoring

Logs & events

Configuration

Maintenance & backups

Tags

Connectivity & security

Endpoint & port

Endpoint

database-1.cxmh0a2kraq.us-east-1.rds.amazonaws.com

Port

3306

Networking

Availability Zone

us-east-1a

VPC

Default (vpc-09878b2678e2329fe)

Subnet group

default-vpc-09878b2678e2329fe

Subnets

Security

VPC security groups

VPC SG (sg-044042ae2ec96ea1f)

Active

Publicly accessible

No

Certificate authority

rds-ca-2019



```
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-45-63 ~]$ sudo su -
[root@ip-172-31-45-63 ~]# yum -y install mariadb-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package mariadb-server.x86_64 1:5.5.68-1.amzn2 will be installed
--> Processing Dependency: mariadb(x86-64) = 1:5.5.68-1.amzn2 for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl-DBI for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl(Data::Dumper) for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Processing Dependency: perl(DBI) for package: 1:mariadb-server-5.5.68-1.amzn2.x86_64
--> Running transaction check
--> Package mariadb.x86_64 1:5.5.68-1.amzn2 will be installed
--> Package perl-DBD-MySQL.x86_64 0:4.023-6.amzn2 will be installed
--> Package perl-DBI.x86_64 0:1.627-4.amzn2.0.2 will be installed
--> Processing Dependency: perl(RPC::PlServer) >= 0.2001 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Processing Dependency: perl(RPC::PlClient) >= 0.2000 for package: perl-DBI-1.627-4.amzn2.0.2.x86_64
--> Package perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-PlRPC.noarch 0:0.2020-14.amzn2 will be installed
--> Processing Dependency: perl(Net::Daemon) >= 0.13 for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Test) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Net::Daemon::Log) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Processing Dependency: perl(Compress::Zlib) for package: perl-PlRPC-0.2020-14.amzn2.noarch
--> Running transaction check
--> Package perl-IO-Compress.noarch 0:2.061-2.amzn2 will be installed
--> Processing Dependency: perl(Compress::Raw::Zlib) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Processing Dependency: perl(Compress::Raw::Bzip2) >= 2.061 for package: perl-IO-Compress-2.061-2.amzn2.noarch
--> Package perl-Net-Daemon.noarch 0:0.48-5.amzn2 will be installed
--> Running transaction check
--> Package perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.amzn2.0.2 will be installed
--> Package perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
Dependency Installed:
  mariadb.x86_64 1:5.5.68-1.amzn2                perl-Compress-Raw-Bzip2.x86_64 0:2.061-3.amzn2.0.2    perl-Compress-Raw-Zlib.x86_64 1:2.061-4.amzn2.0.2    perl-DBI-MySQL.x86_64 0:4.023-6.amzn2
  perl-DBI.x86_64 0:1.627-4.amzn2.0.2            perl-Data-Dumper.x86_64 0:2.145-3.amzn2.0.2            perl-IO-Compress.noarch 0:2.061-2.amzn2                perl-Net-Daemon.noarch 0:0.48-5.amzn2
  perl-PlRPC.noarch 0:0.2020-14.amzn2

Complete!
[root@ip-172-31-45-63 ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[root@ip-172-31-45-63 ~]# systemctl start mariadb
[root@ip-172-31-45-63 ~]# mysql -h database-1.coxnoha2kraq-us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
^C
[root@ip-172-31-45-63 ~]# mysql -h database-1.coxnoha2kraq-us-east-1.rds.amazonaws.com -P 3306 -u admin -p
```