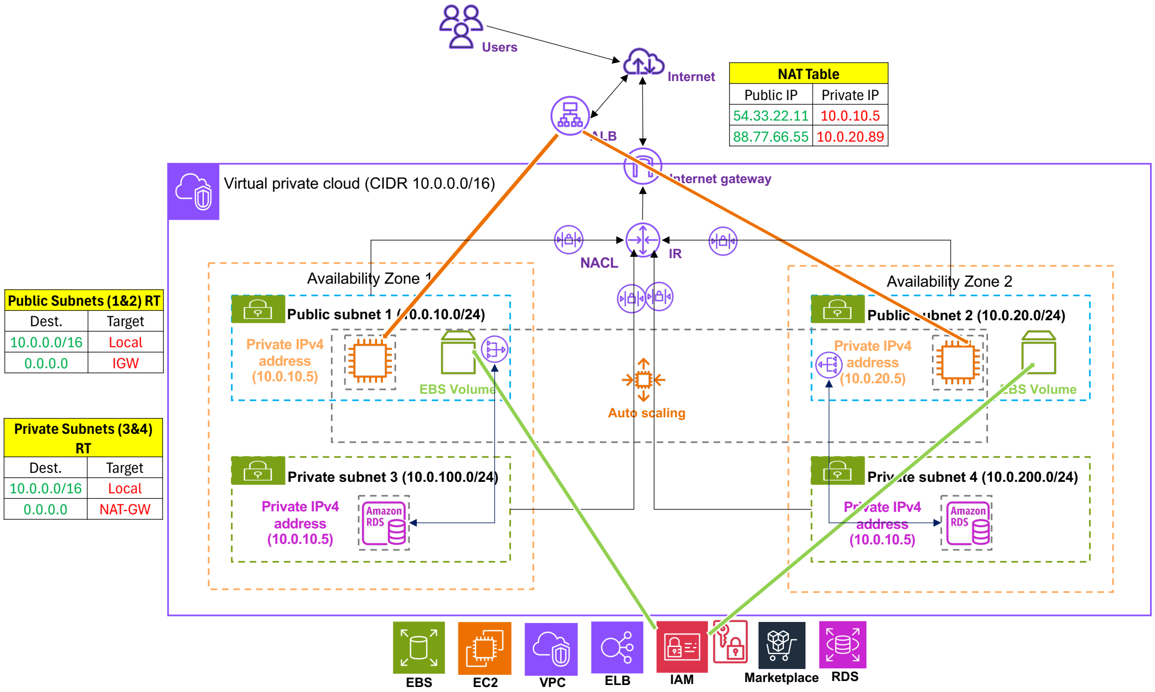
In this project, I will create the IaaS and PaaS infrastructure required to host a multi-tier web application in AWS VPC.


Users
Internet
ALB
Internet gateway
Virtual private cloud (CIDR 10.0.0.0/16)
NACL
IR
Availability Zone 1
Availability Zone 2
Public subnet 1 (10.0.10.0/24)
Public subnet 2 (10.0.20.0/24)
Private IPv4 address (10.0.10.5)
Private IPv4 address (10.0.20.5)
EBS Volume
EBS Volume
Auto scaling
Private subnet 3 (10.0.100.0/24)
Private subnet 4 (10.0.200.0/24)
Private IPv4 address (10.0.10.5)
Private IPv4 address (10.0.10.5)
Amazon RDS
Amazon RDS
EBS
EC2
VPC
ELB
IAM
Marketplace
RDS

NAT Table	
Public IP	Private IP
54.33.22.11	10.0.10.5
88.77.66.55	10.0.20.89

Public Subnets (1&2) RT	
Dest.	Target
10.0.0.0/16	Local
0.0.0.0	IGW

Private Subnets (3&4) RT	
Dest.	Target
10.0.0.0/16	Local
0.0.0.0	NAT-GW

The requirements are:

Design a solution for a multi-tier web application that will be deployed in a custom VPC.

1. Create a custom VPC with CIDR block 10.0.0.0/16 with:

   - 2 Public subnets in two different AZs US-east-1a and us-east-1b in US-east-1 region. Use 10.0.10.0/24 and 10.0.20.0/24 ranges for these two subnets.

   - Please make sure the subnet settings are such that public IP addresses are automatically assigned to EC2 instances launched in these public subnets.

   - 2 Private Subnets in the same AZs as above. Use 10.0.100.0/24 and 10.0.200.0/24. Create a separate route tables for the private subnets.

2. Launch two EBS-backed EC2 Instances, one in each of the two private subnets above (10.0.100.0/24 and 10.0.200.0/24).

   - The instances will serve as the web and application tiers.

   - Ensure that the EBS volumes of these instances are encrypted at rest.

- The Instances will have the following user data script run at launch time.

Bash script 1

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd; systemctl enable httpd
systemctl enable httpd@ enable httpd to auto-start at system boot
echo "This is server "1" in AWS Region US-EAST-1 in AZ US-EAST-1A " > /var/www/html/index.html
```

Bash script 2

```
#!/bin/bash
yum update -y
yum install httpd -y
systemctl start httpd; starts httpd service
systemctl enable httpd@ enable httpd to auto-start at system boot
echo "This is server "2" in AWS Region US-EAST-1 in AZ US-EAST-1B " > /var/www/html/index.html
```

- Launch a NAT gateway in each of the two availability zones above to allow the two Instances to access the Internet for updates.

- Adjust the private subnets route table(s) to route the update traffic through the NAT Gateway.

- The security group assigned to the instances should use the name webSG and must allow ports ssh (22), http (80) and https (443) in the Inbound direction.

3. create a target group with the name webTG and add the two application instances as targets.

- The target group will use the port 80 (HTTP) for traffic forwarding and health checks.

4. Launch an application load balancer that will load balance to these two Instances using HTTP. The application load balancer must be enabled in the two public subnets you have configured in step 1.

    - Adjust the security group of the web/app instances to allow inbound application load balancer security group (ALBSG) as source only for port 80.

    - ALB security group (ALBSG) must allow outbound http to the web/app security group (webSG).

    - The ALBSG should allow inbound traffic from the Internet on port 80.

5. Configure a target tracking auto scaling group for the web/app instances that will ensure elasticity and high availability for the site. The Auto Scaling group should monitor the two instances and be able to add lost instances on demand and replaced failed instances.

6. Launch a Multi AZ RDS database and ensure that its security group will only allow access from the webSG (step 2) only above.

7. Test to ensure that you can get to the index.html message on the instances through the load balancer. If it works, congratulations, you have completed the assignment on AWS.