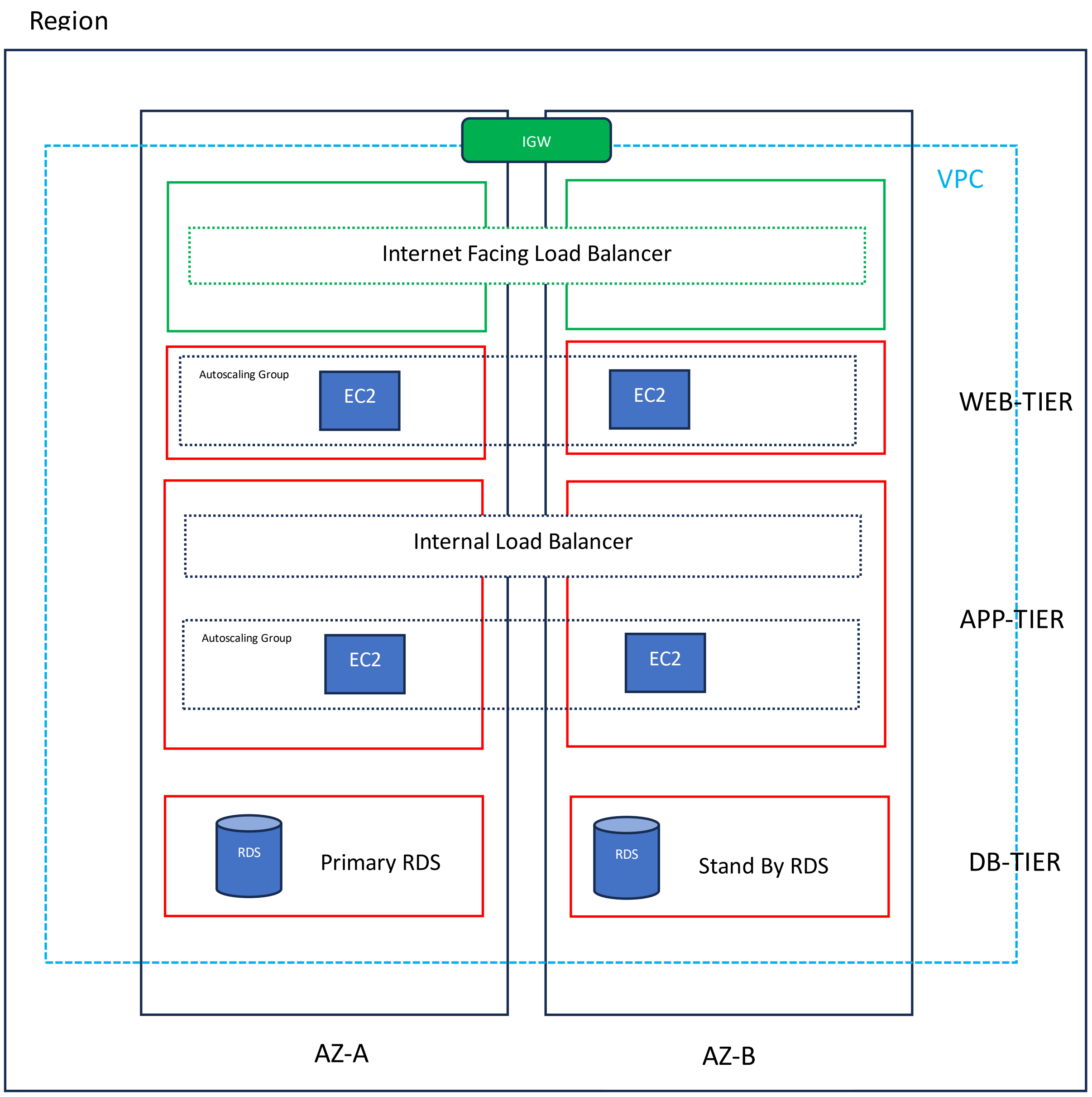
Three Tier Architecture



STEPS :

1) Create a Custom VPC

AWS VPC Dashboard → Click Create VPC → choose VPC and more → Enter VPC Name and IPv4 CIDR Block (e.g., 10.0.0.0/16) → Click Create VPC.

2) Create an Internet Gateway (IGW)

AWS VPC Dashboard → Internet Gateways → Click Create Internet Gateway → Enter a name → Click Create → Select IGW → Attach to VPC →Select the previously created VPC and attach.

3) Create Subnets

AWS VPC Dashboard → Subnets → Click Create Subnet → Select custom VPC → Enter Subnet Name → choose AZ A → set CIDR → Click add subnet → add next subnet details.

Likewise create 8 subnets = 6 private(3 in AZ A and 3 in AZ B) and 2 public(one each).

4) Create Route Tables

Public Route Table

VPC Dashboard → Route Tables → Click Create Route Table → Enter a Name → select VPC → create.
Select Route Table → go to Routes →add: Destination: 0.0.0.0/0 → Target: Internet Gateway.
Associate this Route Table with Public Subnets.

Private Route Table

VPC Dashboard → Route Tables → Click Create Route Table → Enter a Name → select VPC → create.
Associate this Route Table with Private Subnets.

5) Create an RDS Instance

1. RDS Dashboard → Subnet Groups → Click Create DB Subnet Group → Enter a Name and Description → Select custom VPC → Add Private Subnets from multiple Availability Zones (AZ A and AZ B) → Click Create DB Subnet Group.

2. AWS RDS Dashboard → Create Database → Select Standard Create → choose MySQL →Select Dev/Test template → Master Username, and Password → choose db subnet group → Select Security Group to allow MySQL connections → allow stand by → Click Create Database.

6) Launch a APP TIER Development Server in Public Subnet

AWS EC2 Dashboard → Launch Instance → Select Amazon Linux 2023 AMI → Choose t2.micro instance type → Select Public Subnet A or B → enable Public IP → Attach Security Group allowing SSH (22) and HTTP (80)→ Click Launch and connect via SSH.

7) Install required service and setup in app tier development sever

- Install php , php-fpm , php8.3-mysqlnd , mariadb105-server

  ```
  sudo yum install -y php php-mysqlnd mariadb105-server
  ```

- Connect to RDS and perform required queries
  mysql -u username -p -h rds endpoint

- start and enable php-fpm
  sudo service php-fpm start
  sudo systemctl enable php-fpm.service

- Add PHP Code at /usr/share/nginx/html location

8) Create an AMI Image of APP TIER development server

EC2 Dashboard → Select the app tier development instance → Click Actions → Image and Templates → Create Image → Provide Image Name and Description → create.

9) Create an Auto Scaling Group for APP TIER (with template and load balancer)

EC2 Dashboard → Auto Scaling Groups → Click Create Auto Scaling Group → give group name → create launch template → give template name → select app tier development server ami → give instance type → give key pair → create security group → give name → select custom vpc → launch template → back → select template → next → select custom VPC → select app tier private subnets → next → select Attach to new load balancer → select load balancer type ALB → Give name → internal → keep subnets same as instances →create target group → give name → allow health checks for EBS and instances → next → give min , max and desire instances count → give increment policy → allow cloudwatch monitoring →next → next →next→ review → Create

10) Launch a WEB TIER Development Server in Public Subnet

AWS EC2 Dashboard → Launch Instance → Select Amazon Linux 2023 AMI → Choose t2.micro instance type → Select Public Subnet A or B → enable Public IP → Attach Security Group allowing SSH (22) and HTTP (80)→ Click Launch and connect via SSH.

11) Install required service and setup in web tier development sever

- Install nginx
  sudo yum install nginx -y

- Configure for php – send php requests to internal load balancer

```
Sudo nano /etc/nginx/nginx.conf
```

Add

```
Location ~ \.php$  {
  proxy_pass http://internal_load_balancer_dns;
}
```

- start and enable nginx

  ```
  sudo service nginx start
  sudo systemctl enable nginx.service
  ```

- Add html Code at /usr/share/nginx/html location

12) Create an AMI Image of WEB TIER development server

EC2 Dashboard → Select the web tier development instance → Click Actions → Image and Templates → Create Image → Provide Image Name and Description → create.

13) Create an Auto Scaling Group for APP TIER

EC2 Dashboard → Auto Scaling Groups → Click Create Auto Scaling Group → give group name → create launch template → give template name → select web tier development server ami → give instance type → give key pair → create security group → give name → select custom vpc → launch template → back → select template → next → select custom VPC → select web tier private subnets → next → select Attach to new load balancer → select load balancer type ALB → Give name → internet-facing → select public subnets →create target group → give name → allow health checks for EBS and instances → next → give min , max and desire instances count → give increment policy → allow cloudwatch monitoring →next → next →next→ review → Create.

14) Build security group chain from top to bottom in architecture

Select security group → Edit inbound rule

1. Internet Facing Load Balancer Security Group –

Type	Port	Source
HTTP	80	0.0.0.0/0

2. WEB TIER instances security group -

Type	Port	Source
HTTP	80	Internet-facing load balancer Security group id

3. Internal Load Balancer Security Group –

Type	Port	Source
HTTP	80	WEB TIER instances Security group id

4. APP TIER instances security group –

Type	Port	Source
HTTP	80	Internal load balancer Security group id

5. DB TIER instances security group –

Type	Port	Source
MYSQL/Aurora	3306	APP TIER instances Security group id

15) Run website using internet-facing load balancer DNS name