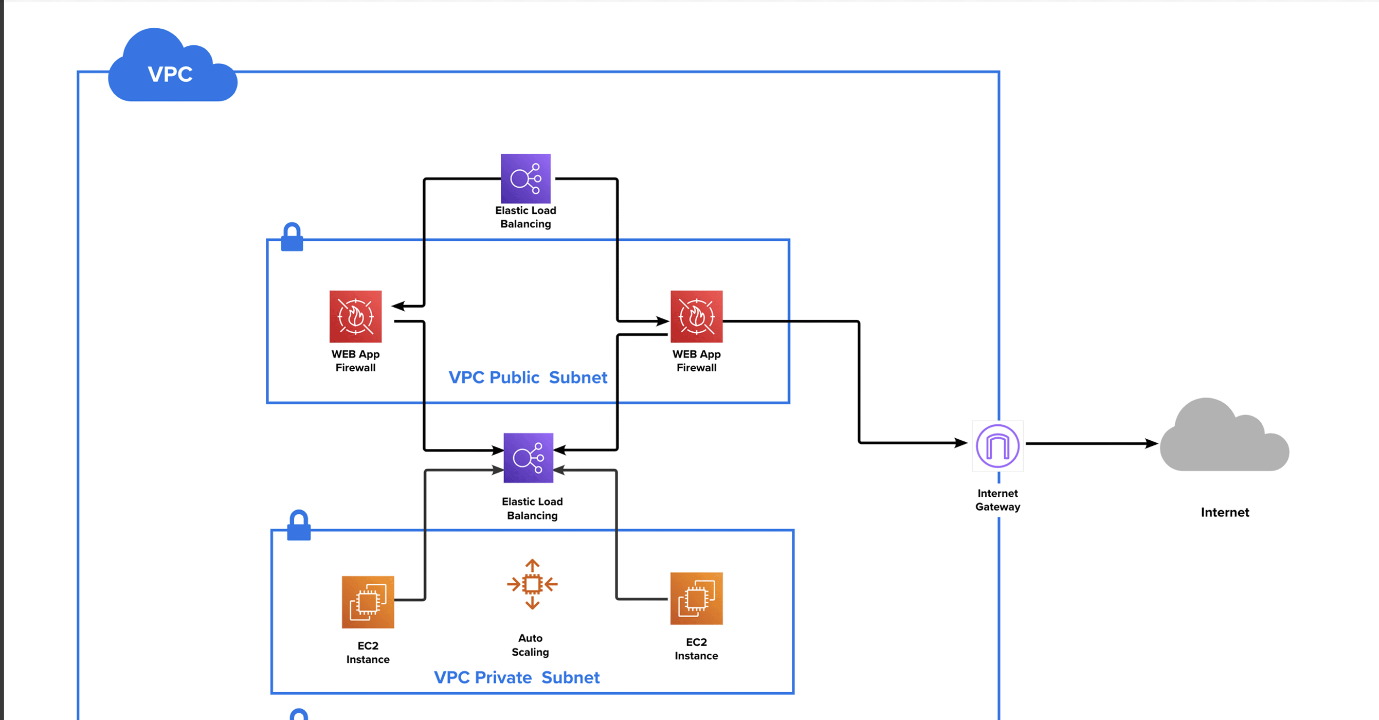
**Architecting a Highly Available Web Service Infrastructure with VPC with Public and Private subnets, Auto Scaling and Load Balancers.**

**Aws Architecture Diagram**

****

**Resources we need to know before implementing AWS Architecture:**

vpc,subnet, internet gateway, route-table, aws-route-table\_association, security group, aws\_load\_balancer, aws\_lb\_target\_group, and target\_attachment, aws\_lb\_listener.

->process to create Aws Architecture:

🡪In AWS Cloud

Create VPC -> then 2 Subnets(In different Availability Zone)

1. We need to create instances inside subnets.
2. We need Internet-Gateway to connect VPC**,** as it enables resources in your public subnets (such as EC2 instances) to connect to the internet, if the resource has a public IPv4 address or an IPv6 address.
3. We need to have route\_table

-->At first, we need route table ,if we want to access internet through internet gateway. In interent gateway we have destination and target, target specify through which we want to access.

-->second, now we linked vpc to route table, so we need instances and subnet's to assosiate with route table to access.

4. Security Group – A security group controls the traffic that is allowed to reach and leave the resources that it is associated with.

5. Application load balancer-->we link load\_balancer with subnets . so that it distributes traffic to instances that is present in these subnets and checks the health.

6. aws\_lb\_target\_group-->A target group tells a load balancer where to direct traffic to : EC2 instances, fixed IP addresses; or AWS Lambda functions, amongst others. When creating a load balancer, you create one or more listeners and configure listener rules to direct the traffic to one target group.

**Terraform**

HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.

Terraform allows you to configure your infrastructure over multiple cloud providers like AWS, Azure, etc. by using the same workflow. This makes management and orchestration for large, multi-cloud infrastructures simpler.