**Challenge #1**

A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these resources. Please remember we will not be judged on the outcome but more focusing on the approach, style and reproducibility.

**Solution #1**

A three-tier architecture typically consists of:

* A frontend Load Balancer to accept with an Internet Gateway attached to receive traffic from the internet.
* Application servers at the backend which are in the private subnet to minimise the security threats.
* Database servers in the private network.

**The above three-tier architecture is deployed from Terraform on AWS cloud.** The core components of the architecture include:

**Elastic Load Balancer:**

* This serves as frontend to the public and sits in the public subnet.
* Internet Gateway is attached to it to provide a route out to the internet.
* It balances load between the application servers.

**Bastion Host:**

* This is a jump box which is created in public subnet and accepts all type of connections from the internet.
* This jump box is used to login to the application servers which are there in the private subnet.
* The main purpose of jump box is to restrict the direct access of the outside world to the application servers, thus making our environment secure.

**Internet gateway:**

* This is attached to the public subnet to allow communication from the internet.

**Subnets:**

* There are two public subnets in two different Availability Zones to ensure high availability.
* Two private subnets in different Availability Zones which consist of application servers.
* Two Management subnets which are private.
* So, total there are six subnets in the environment.

**Application servers:**

* There are two application servers sitting in the private subnet in two different Availability Zones so that whenever one Availability zone goes down, the application will be served by other Availability Zone providing high fault tolerance.
* The purpose of the application servers is to serve the apache static website.
* Apache is installed on both the application servers with the help of Terraform itself using data block in our Terraform code.
* The security group of the application servers is designed such that it will only accept traffic from the Bastion host on port 22 and from the Load Balancer on port 80. Thus, making our environment highly secure.

Steps to execute the code:

* Create a terraform workspace under the directory three\_tier. Access key and secret key are not hard-coded in the code. Rather, the code is designed to pick the credentials dynamically according to the environment depending on the workspace name.

**terraform workspace new nitu**

* Initialize the terraform. Initializing the terraform will install all the required plugins automatically necessary to build our environment.

**terraform init**

* Dry run to examine all the resources our code is capable to build.

**terraform plan**

* Finally build the resources.

**terraform apply**

Below is the output of our Terraform code:

