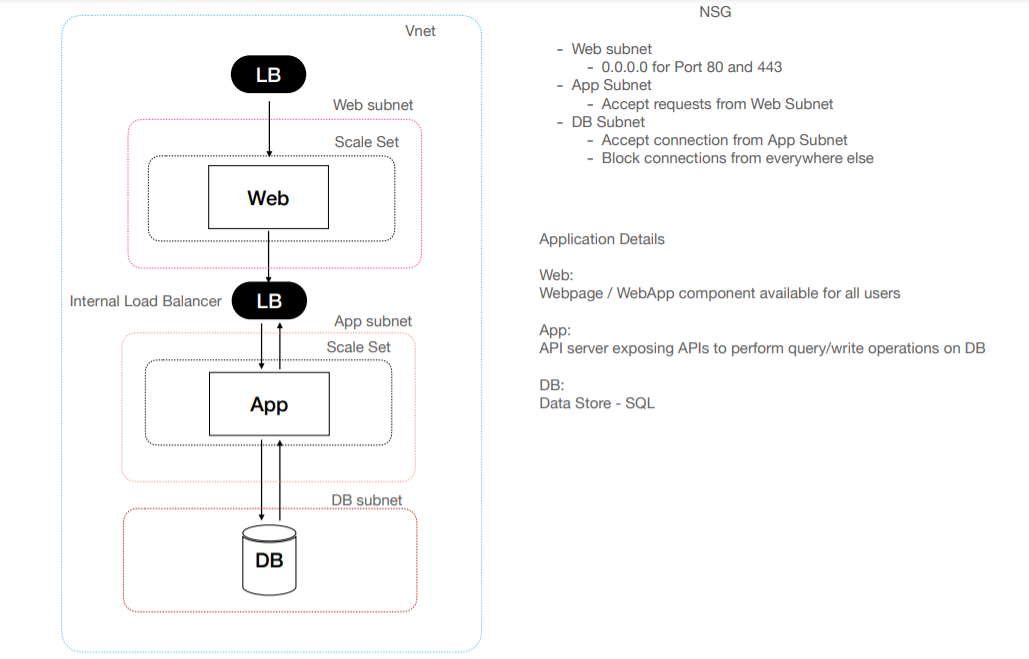
**Challenge #1**

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.Designed the below architecture for a 3-tier app on azure



2. The architecture has the following components

* Virtual network in a resource group
* Three different subnets for webapp, appserver & database
* There is public facing load balacer to accept traffic and backend on for the appserver
* Avaliblity set to handle the scaling part
* Database at the backend which is a sql server

3. Created the below tf files or manifests to achieve the desired state

* main.tf : The main file to govern all the modules
* vars.tf : Variables declared
* terraform.tfvars : passed the terraform variables values

4. Added the below mentioned Modules for organizing resources and reusability.

* Resourcegroup : Container for all resources
* Networksetup : To setup the network
* networksecuriygroup : To govern the traffic flowing
* loadbalancers : To route the traffic based on load
* workload : workernodes to process the code and provide the result
* backendatabase : To host the database

5. Deployment of resources

Steps

* terraform init : Initialize a working directory for Terraform configuration files
* terraform plan : check the desired state and plan the execution
* terraform validate: validate the configuration files in a directory, not accessing any remote services such as remote state, provider APIs.
* terraform apply :used to apply the desired changes required to reach the state of the configuration