**Problem Statement:** 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:**

**3 Tier Application provisioning in Azure using Terraform.**

**Tools used:**

Azure Repos(Azure DevOps) for Source code management

Azure Pipelines(Azure DevOps) for Continuous Integration & Continuous Deployment

Terraform for Cloud provisioning

Azure for Public cloud service platform

**Creation of Terraform configuration file for below Azure resources:**

Resource Group

Public VM for Web server

Private VM for App server

My SQL database

Storage Account

Virtual Network

Subnet

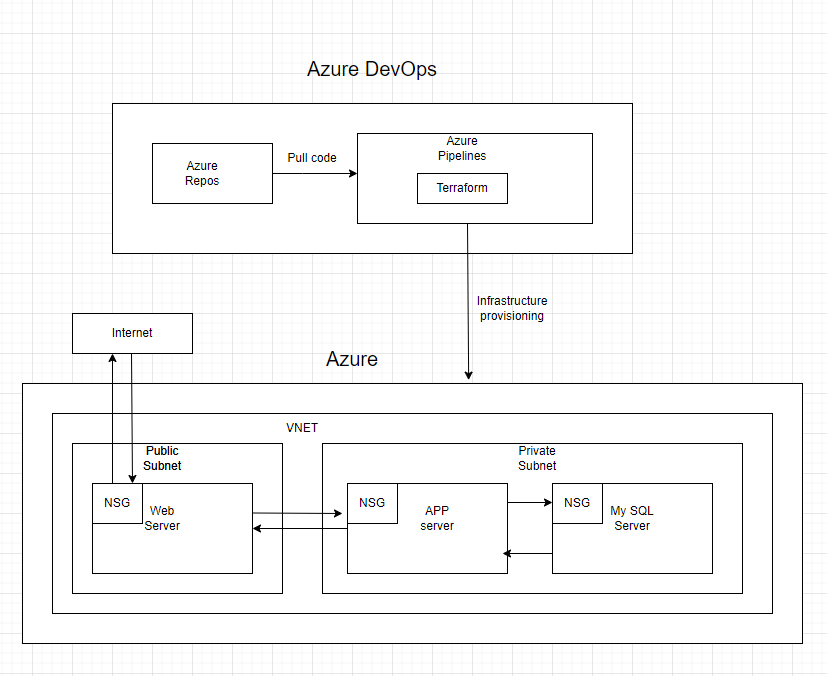
Network Security Group

Network Interface Card

**Also create separate terraform files for Provider and variables**.

\*Push Terraform configuration files into Azure Repo.

**Workflow:**



**Design Azure pipeline for below terraform actions to deploy 3 tier setup into Azure:**

Terraform installer -->To install terraform

Terraform init --> initialization of the working directory which consists of all configuration files

Terraform fmt--> ensure formatting is correct, if required it will format it in proper structure.

Terraform validate--> to ensure no syntax errors are there

Terraform plan --> to create an execution plan to reach a desired state, if required changes can be made in the configuration files to achieve desired state

Terraform apply--> it will make changes in the infrastructure as defined in the plan, and infrastructure will be in desired state.

In order to clean up infrastructure, need to use Terraform destroy action.