Q. **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**.

A.

**Approach** –My approach for an ideal 3-tier setup would be dependent on 3 fundamental domains –

* Network ( Network resource group)
* Compute ( VM’s resource group)
* Application (App resource group)

Network – My network would be consisting of a virtual network having set of resources inside it. The network would consist of Application gateway load balancers. The Load balancers would act on the application layer, offering content based routing for incoming URLs.

There would be subnets based on the applications architecture (Like a separate subnet for DB, Application Virtual machines/Web servers, Front end subnet VMs). Also, we can also have a Hub and Spoke network in which we have the front end (Hub) exposed to internet (Firewall end) and applications hosting Virtual networks (Spoke) peered across each other.

1. One virtual network tied in three subnets.
2. Each subnet will have one virtual machine.
3. First virtual machine -> allow inbound traffic from internet only.
4. Second virtual machine -> entertain traffic from first virtual machine only and can reply the same virtual machine again.
5. App can connect to database and database can connect to app but database cannot connect to web.

Terraform files –

Modules to be used

1. Resource group - creating resource group
2. Networking - creating azure virtual network and required subnets
3. Security group - creating network security group, setting desired security rules and associating them to subnets
4. compute - creating availability sets, network interfaces and virtual machines
5. database - creating database server and database

├── main.tf // The primary entrypoint for terraform resources.

├── vars.tf // It contain the declarations for variables.

├── output.tf // It contain the declarations for outputs.

├── terraform.tfvars // The file to pass the terraform variables values.

Main.tf

|  |  |
| --- | --- |
|  | |
| provider "azurerm" { | |
|  | | features {} | |
|  | | } | |
|  | |  | |
|  | | module "resourcegroup" { | |
|  | | source = "./modules/resourcegroup" | |
|  | | name = var.name | |
|  | | location = var.location | |
|  | | } | |
|  | |  | |
|  | | module "networking" { | |
|  | | source = "./modules/networking" | |
|  | | location = module.resourcegroup.location\_id | |
|  | | resource\_group = module.resourcegroup.resource\_group\_name | |
|  | | vnetcidr = var.vnetcidr | |
|  | | websubnetcidr = var.websubnetcidr | |
|  | | appsubnetcidr = var.appsubnetcidr | |
|  | | dbsubnetcidr = var.dbsubnetcidr | |
|  | | } | |
|  | |  | |
|  | | module "securitygroup" { | |
|  | | source = "./modules/securitygroup" | |
|  | | location = module.resourcegroup.location\_id | |
|  | | resource\_group = module.resourcegroup.resource\_group\_name | |
|  | | web\_subnet\_id = module.networking.websubnet\_id | |
|  | | app\_subnet\_id = module.networking.appsubnet\_id | |
|  | | db\_subnet\_id = module.networking.dbsubnet\_id | |
|  | | } | |
|  | |  | |
|  | | module "compute" { | |
|  | | source = "./modules/compute" | |
|  | | location = module.resourcegroup.location\_id | |
|  | | resource\_group = module.resourcegroup.resource\_group\_name | |
|  | | web\_subnet\_id = module.networking.websubnet\_id | |
|  | | app\_subnet\_id = module.networking.appsubnet\_id | |
|  | | web\_host\_name = var.web\_host\_name | |
|  | | web\_username = var.web\_username | |
|  | | web\_os\_password = var.web\_os\_password | |
|  | | app\_host\_name = var.app\_host\_name | |
|  | | app\_username = var.app\_username | |
|  | | app\_os\_password = var.app\_os\_password | |
|  | | } | |