Documentation: Defining network services with IaC

Tasks:

1. Deploy the containerised application on to a Kubernetes cluster to ensure application scalability using terraform to implement infrastructure as code (IaC).
2. Before the Kubernetes can be deployed, first the network services are to be provisioned or setup in Azure to ensure the cluster's seamless operations and secure communication.

Steps:

1. Created a terraform project named "aks-terraform-main" to serve as a foundation for the Kubernetes cluster.
2. Created two terraform module directories namely, network-module and aks-cluster-module. The network-module is for provisioning the necessary network services for an AKS services. The aks-cluster-module is for provisioning the Kubernetes cluster.
3. Three files are created inside the network-module namely, variables.tf, main.tf and outputs.tf.
4. The variables.tf file is used to define the input variables for the network module. These variables are critical components of the terraform project allowing the configuration and customisation of the network services based on specific requirements. The following variables were created in the variables.tf file, adding a description, type and default value:
    a. Resource-group-name: this variable will represent the name of the Azure Resources group where the networking resources would be deployed in.
    b. Location: this variable is used to specify the Azure region where the networking resources will be deployed to.
    c. vnet_address_space: this variable specifies the address space for the Virtual Network (VNet) to be created in the main configuration file of this network-module.
5. The main.tf file within the network-module defines the essential networking resources for the AKS cluster. This includes creating an Azure Resource Group, a VNet, two subnets (for the control plane and worker nodes) and a Network Security Group (NSG). These resources are named as follows:
    a. Azure Resource Group: Name this resource by referencing the resource_group_name variable created in the variables.tf file.
    b. Virtual Network (VNet): aks-vnet
    c. Control Plane Subnet: control-plane-subnet
    d. Worker Node Subnet: worker-node-subnet
    e. Network Security Group (NSG): aks-nsg - Within the NSG, two inbound rules were defined: one to allow traffic to the kube-apiserver (named kube-apiserver-rule) and the other to allow inbound SSH traffic (named ssh-rule). Both rules would only allow inbound traffic from my public IP address.
6. The outputs.tf configuration file, is used to define the output variables of the module. Output variables enable one to access and utilize information from the networking module. Specifically, these variables will be used to provision the networking services used by the AKS cluster later on when cluster module is provisioned. The following output variables were defined:
    a. vnet_id: variable that stores the ID of the previously created VNet. This will be used within the cluster module to connect the cluster to the defined VNet.

  b. control_plane_subnet_id: variable that holds the ID of the control plane subnet within the VNet. This will be used to specify the subnet where the control plane components of the AKS cluster will be deployed to.
  c. worker_node_subnet_id: variable that stores the ID of the worker node subnet within the VNet. This will be used to specify the subnet where the worker nodes of the AKS cluster will be deployed to.
  d. networking_resource_group_name: variable that provides the name of the Azure Resource Group where the networking resources were provisioned in. This will be used to ensure the cluster module resources are provisioned within the same resource group.
  e. aks_nsg_id: variable that stores the ID of the Network Security Group (NSG). This will be used to associate the NSG with the AKS cluster for security rule enforcement and traffic filtering.

7. Initialized the networking module to make it ready to use within the main project. To do this made sure to be in the correct directory (network-module) before running the initialization command (terraform init).
8. Finally, pushed the latest IaC file to github. First while on the network module directory add the files to git (git add <file name>), then commit the files (git commit -m <"description of action">). And git push command to push to git hub.