Documentation: Creating an AKS cluster with IaC

Tasks:

1. Complete the provisioning of an AKS cluster using Terraform as well as the previously created network and aks-cluster modules by initializing and applying the Terraform projects to create and aks cluster.
2. Access the cluster and test its functionality to ensure a successful delivery.

Steps:

1. Main.tf : this is the main project configuration file created in the main project directory (aks-terraform-main). Within this file, first thing to be defined was the Azure provider block to enable authentication to Azure using the service principal credentials. The Azure provider is Azurerm. The provider block contains a reference to authentication credentials in the variables.tf file.
2. Variables.tf : defined variables for the client_id and client_secret, subscription_id and tenant_id arguments in a variables.tf file, and then created equivalent variables in a secrets.tfvars file to store the values without exposing the credentials.
3. Referenced the secrets.tfvars file in the .gitignore file.
4. Next, after provisioning the provider block, integrate the network module in the project's main configuration file. This integration will ensure that the networking resources previously defined in their respective module are included, and therefore accessible in the main project. Make sure to provide the following input variables when calling the module:
    a. Set resource_group_name to a descriptive name, such as "networking-resource-group".
    b. Set location to an Azure region that is geographically close to you to improve latency (e.g."UK South").
    c. Set vnet_address_space to ["10.0.0.0/16"].
5. Then, integrated the cluster module in the main project configuration file. This step connects the AKS cluster specifications to the main project, as well as allowing you to provision the cluster within the previously defined networking infrastructure. Provide the following input variables when calling the module:
    a. Set cluster_name to "terraform-aks-cluster".
    b. Set location to an Azure region that is geographically close to you to improve latency (e.g."UK South").
    c. Set dns_prefix to "myaks-project".
    d. Set kubernetes_version to a Kubernetes version supported by AKS, set to "1.27.7".
    e. Set service_principal_client_id and service_principal_secret to your service principal credentials.
6. Used variables referencing the output variables from the networking module for the other input variables required by the cluster module such as:
    a. resource_group_name,
    b. vnet_id,
    c. control_plane_subnet_id,
    d. worker_node_subnet_id
    e. aks_nsg_id
7. The Terraform project was initialized within the main project directory. Then a plan of the project was issued with the “terraform plan” command. Then the Terraform configuration

was applied using the “terraform apply” command. This initiated the creation of the defined infrastructure, including the networking resources and AKS cluster.

8. Note that the Terraform plan and Terraform apply commands were issued using the .tfvars file this way: terraform plan -var-file="secrets.tfvars" and terraform apply -var-file="secrets.tfvars".
9. Added the resultant state file to the .gitignore to avoid exposing any secrets.
10. Retrieved the kubeconfig file once the AKS cluster had been provisioned. This configuration file allows you to connect to the AKS cluster securely. Connect to the newly created cluster to ensure that the provisioning process was successful, and the cluster is operational.
11. Finally, pushed the latest IaC file to github. First while on the aks-cluster 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.