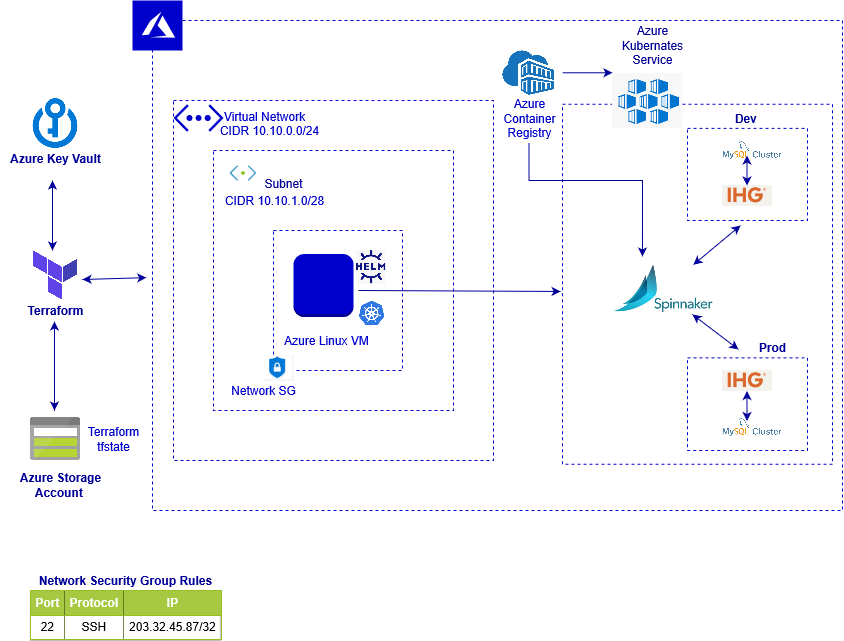
Infrastructure as code (IaC) is managing and provisioning of infrastructure through code instead of through manual processes. IaC ensures that we provision the same environment every time. Deploying our infrastructure as code also means that you can divide your infrastructure into modular components that can combined in different ways through automation.

There are many tools that can help automate our infrastructure. We are using Terraform for to deploy the entire underlying infrastructure.

**Infrastructure Diagram:**



**Diagram Description:**

* Create azure kubernetes service.
* Azure container registry for store applications docker images.
* Azure linux vm for accessing kubernetes cluster.
* Required azure key vault for encrypt our credentials.
* Azure storage account for used it as terraform backend.
* Azure network security group for control vm access.

**Access Required:**

* Azure account
* Azure account user with azure cli access

**Environment setup:**

* Install terraform
* Install azure cli
* Create azure user for terraform
* Enable azure vault service (one time activity)
* Create azure secret (one time activity)
* Create azure storage account and storage container for terraform tfstate (one time activity)
* Install git

**Install Terraform (In linux):**

# wget https://releases.hashicorp.com/terraform/0.12.4/terraform\_0.12.4\_linux\_386.zip

# unzip terraform\_0.12.4\_linux\_386.zip

# mv terraform /usr/local/bin/terraform

# terraform --version

**Install azure cli:**

# curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

# az --version

**Create azure user for terraform:**

# az login ( Follow instruction)

# az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/SUBSCRIPTION\_ID"

Copy appId, password, tenant and subscription Id for create vault secret.

Create azure storage account and storage container:

# az group create -l eastus -n <YOUR\_RESOURCE\_GROUP\_NAME>

# az storage account create -n <YOUR\_STORGE\_ACC\_NAME> -g <YOUR\_RESOURCE\_GROUP\_NAME> -l eastus

# az storage container create -n <YOUR\_CONTAINER\_NAME>

Collect storage container access key from azure portal.

**Create azure vault secret:**

# az provider register -n Microsoft.KeyVault

# az keyvault create --name "<azure\_vault\_name>" --resource-group "<resourece\_group\_name>" --location <location>

# az keyvault secret set --name "ARM-SUBSCRIPTION-ID" --vault-name "<azure\_vault\_name>" --value "<replace\_subcription\_id>"

# az keyvault secret set --name "ARM-CLIENT-ID" --vault-name "<azure\_vault\_name>" --value "<client\_id>"

# az keyvault secret set --name "ARM-CLIENT-SECRET" --vault-name "<azure\_vault\_name>" --value "<client\_secret>"

# az keyvault secret set --name "ARM-TENANT-ID" --vault-name "<azure\_vault\_name>" --value "<tenant\_id>"

# az keyvault secret set --name "STORAGE-ACCOUNT-KEY" --vault-name "<azure\_vault\_name>" --value "<storage\_container\_access\_key>"

Give terraform to azure vault read access.

Follow this link:- <https://docs.microsoft.com/en-us/azure/key-vault/general/managed-identity>

Source Code:

Azure repo link----

**Configuration for execution:**

# git clone azure\_repo\_link

# cd folder\_name

Replace vault name in vault.sh:

export ARM\_SUBSCRIPTION\_ID=$(az keyvault secret show --name ARM-SUBSCRIPTION-ID --vault-name <azure\_vault\_name> --query value -o tsv)

export ARM\_CLIENT\_ID=$(az keyvault secret show --name ARM-CLIENT-ID --vault-name <azure\_vault\_name> --query value -o tsv)

export ARM\_CLIENT\_SECRET=$(az keyvault secret show --name ARM-CLIENT-SECRET --vault-name <azure\_vault\_name> --query value -o tsv)

export ARM\_TENANT\_ID=$(az keyvault secret show --name ARM-TENANT-ID --vault-name <azure\_vault\_name> --query value -o tsv)

export ARM\_ACCESS\_KEY=$(az keyvault secret show --name STORAGE-ACCOUNT-KEY --vault-name <azure\_vault\_name> --query value -o tsv)

**Source vault.sh:**

# source vault.sh

**Execution:**

For create azure kubernetes service

# cd aks

Replace your required value in var.tf.

# terraform init

# terraform plan

# terraform apply

For create azure vm for access aks

# cd ../vm

Replace your required value in var.tf.

# terraform init

# terraform plan

# terraform apply

**Validation:**