AZURE KUBERNETES SERVICES (AKS) CLUSTER CREATION USING TERRAFORM

Created By

Ashfaque Ahmed Shaikh

Linkedin

Youtube

Contents

1. Prerequisites

1.1 Install Terraform in Windows

https://youtu.be/ERM6UKCh3Hg

1.2 Install Azure CLI

https://learn.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest

https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli

1.3 Download kubectl

https://kubernetes.io/releases/download/

1.4 Download and install VS Code

https://code.visualstudio.com/download

https://learn.microsoft.com/en-us/cli/azure/aks?view=azure-cli-latest#az-aks-install-cli

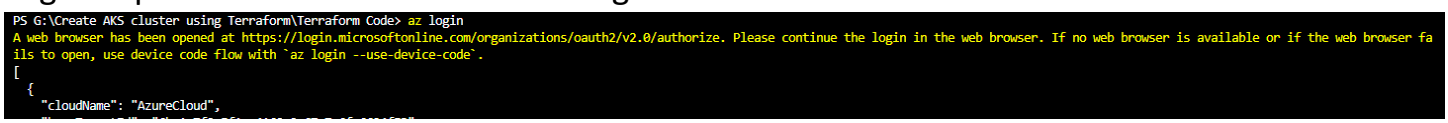
1.5 Install the Azure Terraform Visual Studio Code extension

https://learn.microsoft.com/en-us/azure/developer/terraform/configure-vs-code-extension-for-terraform?tabs=azure-cli

2. Create Directory and login to Azure CLI

2.1 Create folder “Create AKS cluster using Terraform” and inside it create folder “Terraform Code”. Open folder in VS Code. Then right-click on folder “Terraform Code” & open terminal.

2.2 Login to portal.azure.com and run “az login” on terminal.



3. Implement the Terraform code

3.1 Create file provider.tf under folder “Terraform Code” and provide below content to file, save the file.

```
terraform {
  required_version = ">=1.0"

  required_providers {
    azapi = {
      source  = "azure/azapi"
      version = "~>1.5"
    }
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>3.0"
    }
    random = {
      source  = "hashicorp/random"
      version = "~>3.0"
    }
    time = {
      source  = "hashicorp/time"
      version = "0.9.1"
    }
  }
}

provider "azurerm" {
  features {}
}
```

3.2 Create a file named ssh.tf and insert the following code, save the file.

```
resource "random_pet" "ssh_key_name" {
  prefix    = "ssh"
  separator = ""
}

resource "azapi_resource_action" "ssh_public_key_gen" {
  type        = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  resource_id = azapi_resource.ssh_public_key.id
  action      = "generateKeyPair"
  method      = "POST"

  response_export_values = ["publicKey", "privateKey"]
}

resource "azapi_resource" "ssh_public_key" {
  type      = "Microsoft.Compute/sshPublicKeys@2022-11-01"
  name      = random_pet.ssh_key_name.id
  location  = azurerm_resource_group.rg.location
  parent_id = azurerm_resource_group.rg.id
}

output "key_data" {
  value = jsondecode(azapi_resource_action.ssh_public_key_gen.output).publicKey
}
```

3.3 Create the main.tf, insert following script and save the file.

```
# Generate random resource group name
resource "random_pet" "rg_name" {
  prefix = var.resource_group_name_prefix
}

resource "azurerm_resource_group" "rg" {
  location = var.resource_group_location
  name     = random_pet.rg_name.id
}

resource "random_pet" "azurerm_kubernetes_cluster_name" {
  prefix = "cluster"
}

resource "random_pet" "azurerm_kubernetes_cluster_dns_prefix" {
  prefix = "dns"
}

resource "azurerm_kubernetes_cluster" "k8s" {
  location            = azurerm_resource_group.rg.location
  name                = random_pet.azurerm_kubernetes_cluster_name.id
  resource_group_name = azurerm_resource_group.rg.name
  dns_prefix          = random_pet.azurerm_kubernetes_cluster_dns_prefix.id
```

```
  identity {
    type = "SystemAssigned"
  }

  default_node_pool {
    name       = "agentpool"
    vm_size    = "Standard_D2_v2"
    node_count = var.node_count
  }
  linux_profile {
    admin_username = var.username

    ssh_key {
      key_data = jsondecode(azapi_resource_action.ssh_public_key_gen.output).publicKey
    }
  }
  network_profile {
    network_plugin    = "kubenet"
    load_balancer_sku = "standard"
  }
}
```

3.4 Create variables.tf, insert following code and save the file.

```
variable "resource_group_location" {
  type        = string
  default     = "eastus"
  description = "Location of the resource group."
}

variable "resource_group_name_prefix" {
  type        = string
  default     = "rg"
  description = "Prefix of the resource group name that's combined with a random ID so name is unique in your Azure subscription."
}

variable "node_count" {
  type        = number
  description = "The initial quantity of nodes for the node pool."
  default     = 2
}

variable "msi_id" {
  type        = string
  description = "The Managed Service Identity ID. Set this value if you're running this example using Managed Identity as the authentication method."
  default     = null
}

variable "username" {
  type        = string
  description = "The admin username for the new cluster."
  default     = "azureadmin"
}
```

3.5 Create a file named outputs.tf, insert the following code and save the file.

```
output "resource_group_name" {
  value = azurerm_resource_group.rg.name
}

output "kubernetes_cluster_name" {
  value = azurerm_kubernetes_cluster.k8s.name
}

output "client_certificate" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].client_certificate
  sensitive = true
}

output "client_key" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].client_key
  sensitive = true
}

output "cluster_ca_certificate" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].cluster_ca_certificate
  sensitive = true
}
```

```
output "cluster_password" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].password
  sensitive = true
}

output "cluster_username" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].username
  sensitive = true
}

output "host" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config[0].host
  sensitive = true
}

output "kube_config" {
  value     = azurerm_kubernetes_cluster.k8s.kube_config_raw
  sensitive = true
}
```

4. Create Resources using Terraform

4.1 Initialize Terraform

`terraform init -upgrade`

```
PS G:\Create AKS cluster using Terraform\Terraform Code> terraform init -upgrade

Initializing the backend...

Initializing provider plugins...
- Finding azure/azapi versions matching "~> 1.5"...
- Finding hashicorp/azurerm versions matching "~> 3.0"...
- Finding hashicorp/random versions matching "~> 3.0"...
- Finding hashicorp/time versions matching "0.9.1"...
- Installing hashicorp/azurerm v3.86.0...
- Installed hashicorp/azurerm v3.86.0 (signed by HashiCorp)
```

4.2 Create a Terraform execution plan

`terraform plan -out main.tfplan`

```
PS G:\Create AKS cluster using Terraform\Terraform Code> terraform plan -out main.tfplan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # azapi_resource.ssh_public_key will be created
  + resource "azapi_resource" "ssh_public_key" {
      + body                      = jsonencode({})
      + id                        = (known after apply)
```

4.3 Apply a Terraform execution plan

`terraform apply main.tfplan`

```
PS G:\Create AKS cluster using Terraform\Terraform Code> terraform apply main.tfplan
random_pet.azurerm_kubernetes_cluster_name: Creating...
random_pet.rg_name: Creating...
random_pet.ssh_key_name: Creating...
random_pet.azurerm_kubernetes_cluster_dns_prefix: Creating...
```

```
Apply complete! Resources: 8 added, 0 changed, 0 destroyed.

Outputs:
```

5. Verify the results

5.1 Get the Kubernetes configuration from the Terraform state and store it in a file that kubectl can read using the following command.

```
echo "$(terraform output kube_config)" > ./azurek8s
```

5.2 Verify the previous command didn't add an ASCII EOT character using the following command.

5.3 `cat ./azurek8s`

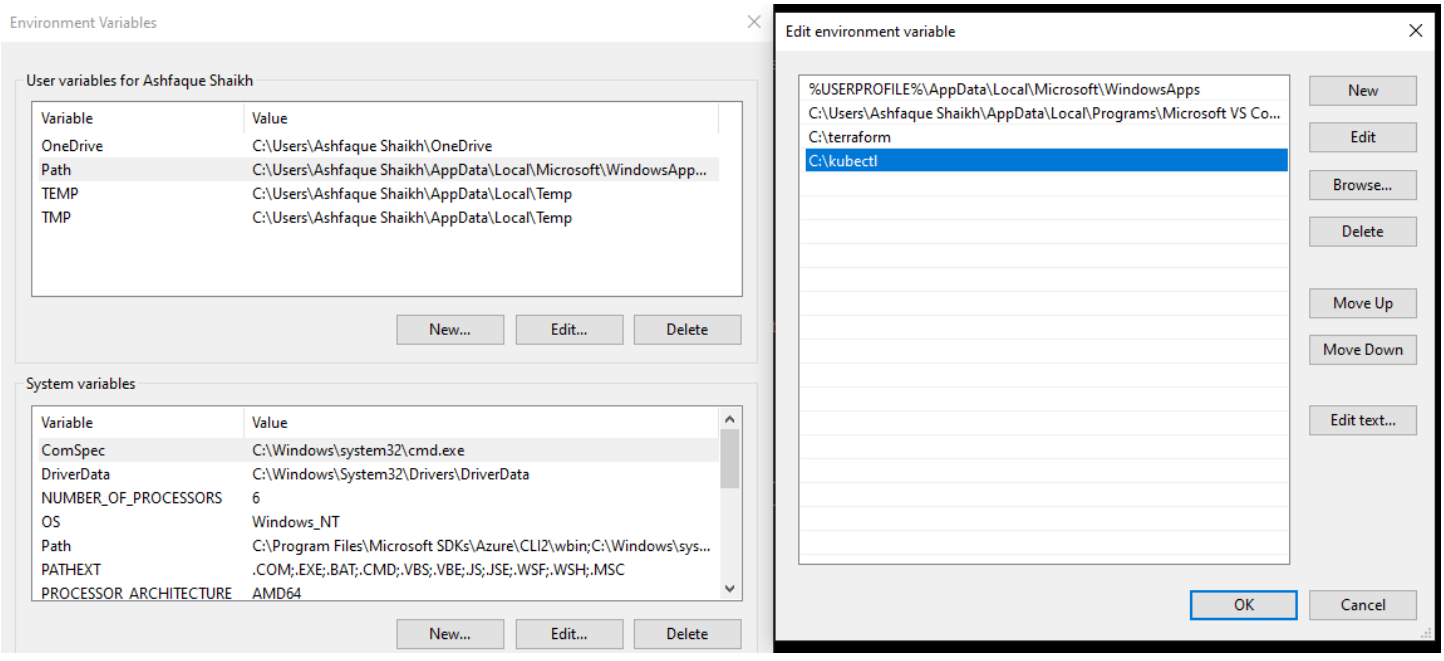
If you see `<< EOT` at the beginning and `EOT` at the end, remove these characters from the file. Otherwise, you may receive the following error message: `error: error loading config file "./azurek8s": yaml: line 2: mapping values are not allowed in this context`

5.4 Install kubectl in system

Download the kubetl.exe from below link, make folder “kubectl” under C drive and paste the downloaded Kubectl.exe in to this folder.

Go to “edit system environment variables”, “environment variables”, click on edit and add new path C:\kubectl

https://code.visualstudio.com/download



Now you will get the version of kubectl with below command

```
PS G:\Create AKS cluster using Terraform\Terraform Code> kubectl version
Client Version: v1.29.0
```

5.5 On Azure portal you can see your AKS cluster under Kubernetes services. If you click on connect, it will show you the commands to connect to your k8s cluster.

```
az login
```

```
az account set --subscription your-subscription-id
```

```
az aks get-credentials --resource-group rg-your-rg-name --name cluster-your-cluster-name
```

Note: Replace these commands with your details or copy these commands directly from portal.

5.6 Now with command kubectl get nodes you will get your node details.

```
PS G:\Create AKS cluster using Terraform\Terraform Code> kubectl get nodes
NAME                                STATUS   ROLES   AGE   VERSION
aks-agentpool-23387267-vmss000000   Ready    agent   38m   v1.27.7
aks-agentpool-23387267-vmss000001   Ready    agent   38m   v1.27.7
PS G:\Create AKS cluster using Terraform\Terraform Code>
```

6. Deploy Application

6.1 Create deployment.yml manifest file.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: swiggy-app
  labels:
    app: swiggy-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: swiggy-app
  template:
    metadata:
      labels:
        app: swiggy-app
    spec:
      terminationGracePeriodSeconds: 30
      containers:
      - name: swiggy-app
        image: ashfaque9x/swiggy-clone:latest
        imagePullPolicy: "Always"
        ports:
        - containerPort: 3000
```

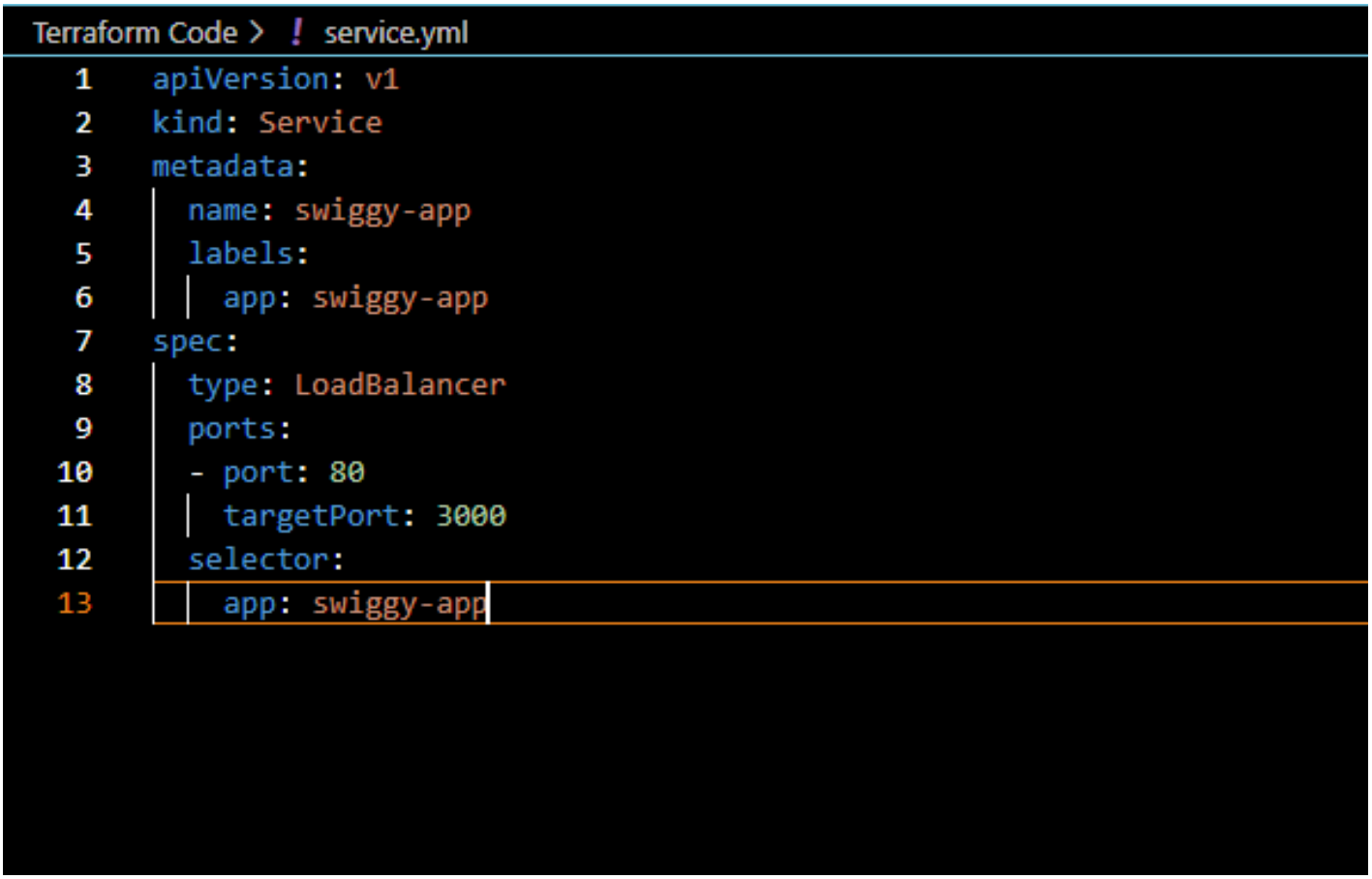
Note: It will use the dockerhub image ashfaque9x/swiggy-clone:latest

Terraform Code > ! deployment.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: swiggy-app
  labels:
    app: swiggy-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: swiggy-app
  template:
    metadata:
      labels:
        app: swiggy-app
    spec:
      terminationGracePeriodSeconds: 30
      containers:
      - name: swiggy-app
        image: ashfaque9x/swiggy-clone:latest
        imagePullPolicy: "Always"
        ports:
        - containerPort: 3000
```

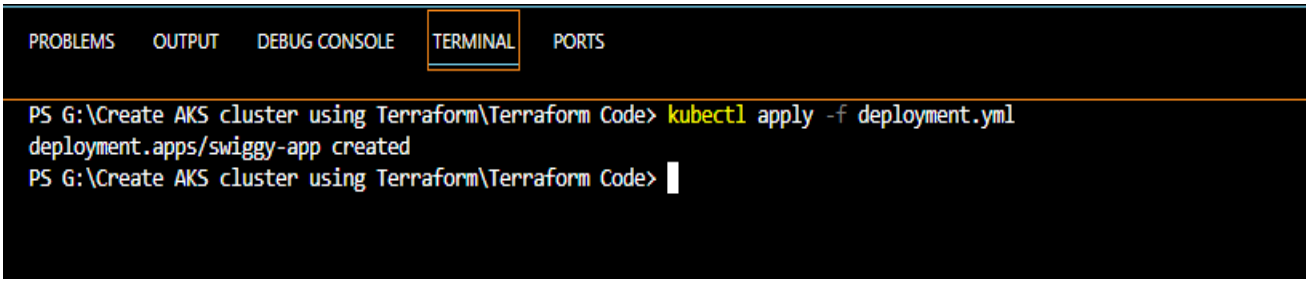
6.2 Create Service.yml file

```
apiVersion: v1
kind: Service
metadata:
  name: swiggy-app
  labels:
    app: swiggy-app
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 3000
  selector:
    app: swiggy-app
```



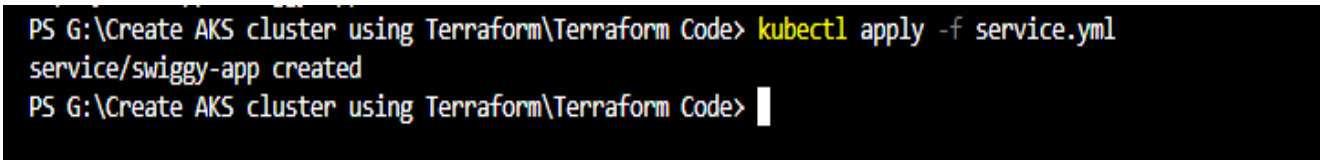
6.2 Create pod for application with below command

```
Kubectl apply –f deployment.yml
```



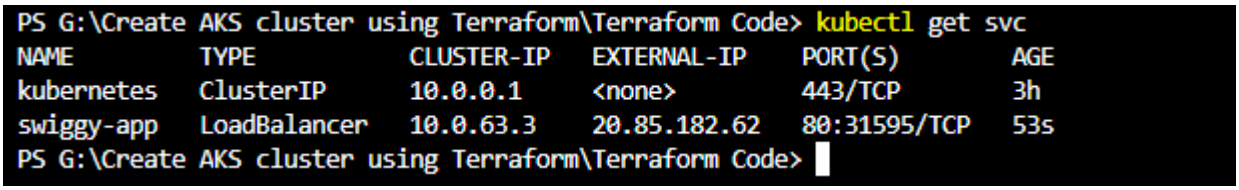
6.3 Create service for the pod with below command

```
Kubectl apply –f service.yml
```

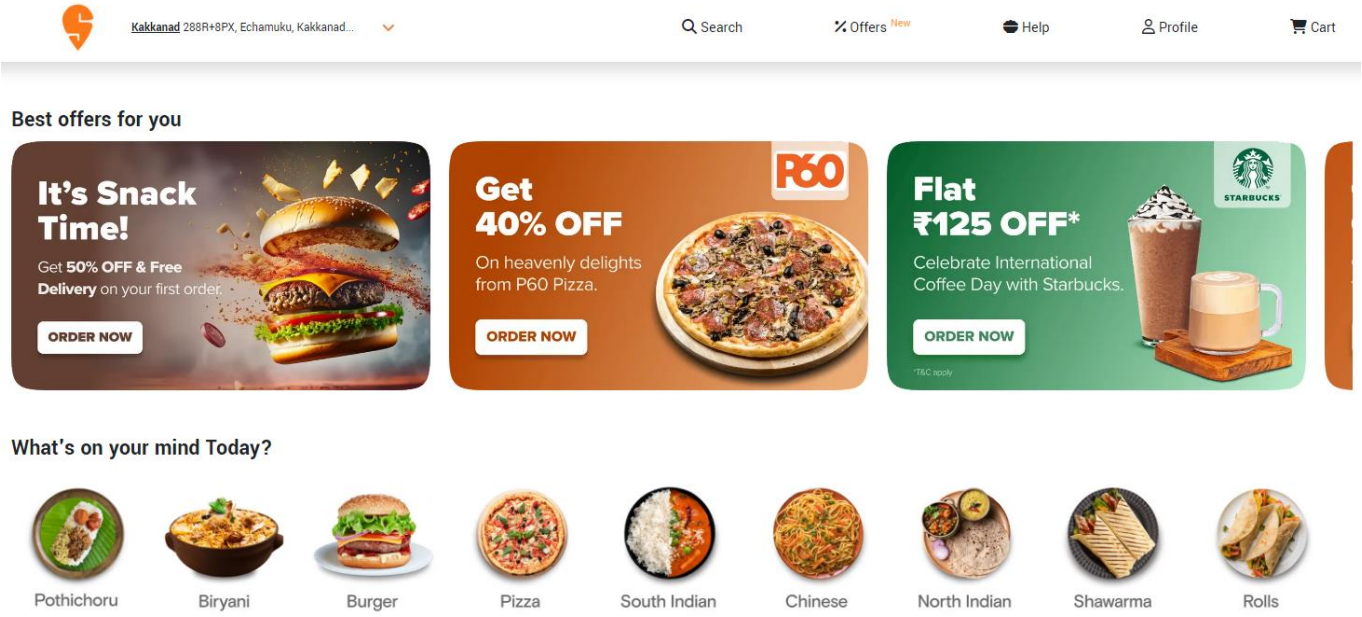


6.4 Get the external IP of the service with below command

```
Kubectl get svc
```



6.5 Browse the application with external IP of the application swiggy-app.



7. Cleanup

7.1 Run command “kubectl get all” to get deployment and service details.

```
PS G:\Create AKS cluster using Terraform\Terraform Code> kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/swiggy-app-74549cb78d-gvbvv      1/1     Running   0          10m
pod/swiggy-app-74549cb78d-v72zc      1/1     Running   0          10m

NAME                 TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)        AGE
service/kubernetes   ClusterIP      10.0.0.1      <none>         443/TCP        3h7m
service/swiggy-app   LoadBalancer   10.0.63.3     20.85.182.62   80:31595/TCP   8m

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/swiggy-app      2/2     2            2           10m
```

7.2 Delete app service with command “kubectl delete service/swiggy-app”

```
PS G:\Create AKS cluster using Terraform\Terraform Code> kubectl delete service/swiggy-app
service "swiggy-app" deleted
PS G:\Create AKS cluster using Terraform\Terraform Code>
```

7.3 Delete deployment with command “kubectl delete deployment.apps/swiggy-app”

```
PS G:\Create AKS cluster using Terraform\Terraform Code> kubectl delete deployment.apps/swiggy-app
deployment.apps "swiggy-app" deleted
PS G:\Create AKS cluster using Terraform\Terraform Code>
```

7.4 Delete AKS resources with below command

```
terraform plan -destroy -out main.destroy.tfplan

terraform apply "main.destroy.tfplan"
```

```
PS G:\Create AKS cluster using Terraform\Terraform Code> terraform plan -destroy -out main.destroy.tfplan
random_pet.rg_name: Refreshing state... [id=rg-whole-fawn]
random_pet.azurerm_kubernetes_cluster_dns_prefix: Refreshing state... [id=dns-destined-lemur]
random_pet.azurerm_kubernetes_cluster_name: Refreshing state... [id=cluster-whole-fawn]
random_pet.ssh_key_name: Refreshing state... [id=sshartisticbuzzard]
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn]
azapi_resource.ssh_public_key: Refreshing state... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn/providers/Microsoft.Compute/sshPublicKeys/sshartistic
buzzard]
azapi_resource_action.ssh_public_key_gen: Refreshing state... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn/providers/Microsoft.Compute/sshPublicKeys/
sshartisticbuzzard/generateKeyPair]
```

```
PS G:\Create AKS cluster using Terraform\Terraform Code> terraform apply "main.destroy.tfplan"
azurerm_kubernetes_cluster.k8s: Destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn/providers/Microsoft.ContainerService/managedClusters/clust
er-whole-fawn]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 10s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 20s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 30s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 40s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 50s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 1m0s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 1m10s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 1m20s elapsed]
azurerm_kubernetes_cluster.k8s: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-...ice/managedClusters/cluster-whole-fawn, 1m30s elapsed]
```

```
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn, 40s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn, 50s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn, 1m0s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn, 1m10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/f608ecc5-c7ec-4011-85fb-916089653126/resourceGroups/rg-whole-fawn, 1m20s elapsed]
azurerm_resource_group.rg: Destruction complete after 1m23s
random_pet.rg_name: Destroying... [id=rg-whole-fawn]
random_pet.rg_name: Destruction complete after 0s

Apply complete! Resources: 0 added, 0 changed, 8 destroyed.
PS G:\Create AKS cluster using Terraform\Terraform Code>
```

Created by: www.linkedin.com/in/ashfaque-ahmed-shaikh

Connect to me,

Youtube --- https://www.youtube.com/@VirtualTechBox

LinkedIn --- www.linkedin.com/in/ashfaque-ahmed-shaikh

Facebook --- https://www.facebook.com/VTechbox

Twitter --- https://twitter.com/vtechbox

Instagram --- https://www.instagram.com/vtechbox

Email --- ashfaque.s510@gmail.com

GitHub --- https://github.com/ashfaque-9x

Telegram --- https://t.me/+rgayvC_exwdlMzU1