Invented for life

04. Create Sample environment using Terraform

inside.Docupedia Export

Author: Shanmugapriya Manickaraj (RBEI/BSF6)

Date: 17-Jul-2019 12:29

Table of Contents

This is a sample demo of how to create a sample environment in the Terraform

Note: There are certain Azure services which are cost incurring Request to go through the pricing and understand clearly and run the code.

https://azure.microsoft.com/en-in/pricing/calculator/

Now I am going to create the environment with the following TF files in the azure

- Resource Group
- Networking
- Security
- Virtual machines

Resource Group:

Resource Group Name	Location
WCMSTEST	Southeast asia

here's the sample code
Save the file as "Resourcegroup.tf"

```
resource "azurerm_resource_group" "main" {
  name     = "POCTEST"    #Name of the Resource group
  location = "Southeast Asia" #Location
  tags {
    environment = "Production"
  }
}
```

Networking

VNET Name: POCVNET

Address Space: 192.168.0.0/16

Subnets: Websubnet (192.168.0.0/24), DBSubnet (192.168.1.0/24), DMZsubnet (!92.168.2.0/24)

Network Interface: POCWEBNIC and POCDBNIC

Save the file as "Networking.tf"

here's the sample code

```
resource "azurerm_virtual_network" "main" {
  name                = "POCVNET"  #Name of the VNET
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
  address_space       = ["192.168.0.0/16"]  #Address space
  tags {
    environment = "Production"
  }

}
resource "azurerm_subnet" "websubnet" {
    name           = "WebSubet"     #Name of the subnet
    address_prefix = "192.168.0.0/24" #CIDR Notation
  resource_group_name  = "${azurerm_resource_group.main.name}"
   virtual_network_name = "${azurerm_virtual_network.main.name}"
   }

resource "azurerm_subnet" "dbsubnet" {
    name           = "DBSubnet"
    address_prefix = "192.168.1.0/24"
  resource_group_name  = "${azurerm_resource_group.main.name}"
   virtual_network_name = "${azurerm_virtual_network.main.name}"
   }

resource "azurerm_subnet" "dmzsubnet" {
    name           = "dmzSubnet"
    address_prefix = "192.168.2.0/24"
  resource_group_name  = "${azurerm_resource_group.main.name}"
   virtual_network_name = "${azurerm_virtual_network.main.name}"
   }


 resource "azurerm_network_interface" "wcmsweb" {
  name                = "POCWEBNIC"   #Name of the network inferface
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
network_security_group_id="${azurerm_network_security_group.webnsg.id}"  #Name of the security group

  ip_configuration {
    name                          = "POCWEBIP"
    subnet_id                      = "${azurerm_subnet.websubnet.id}"
    private_ip_address_allocation = "Dynamic"
  }

}

resource "azurerm_network_interface" "wcmsdb" {
  name                = "POCDBNIC"
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
network_security_group_id="${azurerm_network_security_group.dbnsg.id}"

  ip_configuration {
    name                          = "POCDBIP"
    subnet_id                      = "${azurerm_subnet.dbsubnet.id}"
    private_ip_address_allocation = "Dynamic"
  }
}
```

Security

Network Security Group Name: Webnsg and DBnsg

Rule1: "Allow 443 and 80 for "Webnsg"

Save the file a Security.tf

Here's the sample code.

```
resource "azurerm_network_security_group" "webnsg" {
  name                = "Webnsg" #Name of the nsg
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
}


resource "azurerm_network_security_rule" "allow443" {
  name                        = "allow443"
  priority                    = 100
  direction                   = "Inbound"
  access                      = "Allow"
  protocol                    = "Tcp"
  source_port_range           = "*"
  destination_port_range      = "443"
  source_address_prefix       = "*"
  destination_address_prefix  = "*"
  resource_group_name         = "${azurerm_resource_group.main.name}"
  network_security_group_name = "${azurerm_network_security_group.webnsg.name}"
}

resource "azurerm_network_security_rule" "allow80" {
  name                        = "allow80"
  priority                    = 101
  direction                   = "Inbound"
  access                      = "Allow"
  protocol                    = "Tcp"
  source_port_range           = "*"
  destination_port_range      = "80"
  source_address_prefix       = "*"
  destination_address_prefix  = "*"
  resource_group_name         = "${azurerm_resource_group.main.name}"
  network_security_group_name = "${azurerm_network_security_group.webnsg.name}"
}
resource "azurerm_network_security_group" "dbnsg" {
  name                = "dbnsg"
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
}
```

Virtual Machines

Availability set Name: NAVWEB1AV and SQLDB01AV

Virtual machine name: NAVWEB1 and SQLDB01

Save the file as Virtualmachine.tf

```
resource "azurerm_availability_set" "navweb1" {
  name                = "NAVWEB1AV"
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
}


resource "azurerm_availability_set" "sqldb01" {
  name                = "SQLDB01AV"
  location            = "${azurerm_resource_group.main.location}"
  resource_group_name = "${azurerm_resource_group.main.name}"
}

resource "azurerm_virtual_machine" "wcmsweb01" {
  name                  = "NAVWEB1"
  location              = "${azurerm_resource_group.main.location}"
  resource_group_name   = "${azurerm_resource_group.main.name}"
  network_interface_ids = ["${azurerm_network_interface.wcmsweb01.id}"]
  availability_set_id = ["${azurerm_availability_set.navweb1.id}"]

  vm_size               = "Basic_A0"


  # Uncomment this line to delete the OS disk automatically when deleting the VM
  delete_os_disk_on_termination = true


  # Uncomment this line to delete the data disks automatically when deleting the VM
 delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
  storage_os_disk {
    name              = "NAVWEB1OS"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  os_profile {
    computer_name  = "NAVWEB1"
    admin_username = "wcmsroot"
    admin_password = "WCMS$$@123aa"
  }
  os_profile_linux_config {
    disable_password_authentication = false
  }
  tags {
    environment = "ProductionNAV01"
  }
}



resource "azurerm_virtual_machine" "wcmsdb" {
  name                  = "SQLDB01"
  location              = "${azurerm_resource_group.main.location}"
  resource_group_name   = "${azurerm_resource_group.main.name}"
  network_interface_ids = ["${azurerm_network_interface.wcmsdb.id}"]
  availability_set_id = ["${azurerm_availability_set.sqldb01.id}"]
  vm_size               = "Basic_A0"


  # Uncomment this line to delete the OS disk automatically when deleting the VM
  delete_os_disk_on_termination = true


  # Uncomment this line to delete the data disks automatically when deleting the VM
 delete_data_disks_on_termination = true

  storage_image_reference {
    publisher = "Canonical"
    offer     = "UbuntuServer"
    sku       = "16.04-LTS"
    version   = "latest"
  }
  storage_os_disk {
    name              = "SQLDB01OS"
    caching           = "ReadWrite"
    create_option     = "FromImage"
    managed_disk_type = "Standard_LRS"
  }
  os_profile {
    computer_name  = "SQLDB01"
    admin_username = "wcmsroot"
    admin_password = "WCMS$$@123aa"
  }
  os_profile_linux_config {
    disable_password_authentication = false
  }
  tags {
    environment = "ProductionDB"
  }
}
```

Follow the deployment steps provided in the below link

05. Terraform Deployment Sequence

Also visit how to get Visual studio subscription

01. How to get Visual Studio Subscription