Terraform for Azure | Hands-On

Version 24.12

Ahmed Abdelwahed
ahmed@abdelwahed.me
www.abdelwahed.me
LinkedIn
GitHub

Setting up Azure on Rockey Linux

Install Azure CLI

1. Import the Microsoft repository key.

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

2. For RHEL 9 or CentOS Stream, add packages-microsoft-com-prod repository:

```
sudo dnf install -y https://packages.microsoft.com/config/rhel/9.0/packages-microsoft-prod.rpm
```

3. Install with the dnf install command.

```
sudo dnf install azure-cli
```

4. Log in to Azure using Azure CLI

```
az login --use-device-code
```

5. Change the Tenant

```
az login --tenant "Your-Tenant-ID"
```

Install Terraform on Rockey Linux

```
sudo yum install -y yum-utils

sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo

sudo yum -y install terraform

terraform -v #verify the installation

#Enable Shell Autocompletion

terraform -install-autocomplete

exec $SHELL
```

Lab1: Create an Azure Resource Group with Terraform

Within new directory do the following

1. Prepare the Terraform Configuration File

Create a `main.tf` file with the following configuration:

```
# Configure the Azure provider
provider "azurerm" {
  features {}
  subscription_id = "b15d7xxx-xxxx-xxxxxx-xxxx-xxxxxxx"
  tenant_id = "0616xxxx-xxxx-xxxxx-xxxxxx-xxxxxxxxx"
}

# Create a Resource Group
resource "azurerm_resource_group" "test" {
  name     = "Terraform-Test-RG"
  location = "East US" # Choose your desired Azure region
}
```

This configuration:

- Connects to Azure using the azurerm provider.
- Creates a resource group named **Terraform-Test-RG** in the **East US** region.

2. Initialize Terraform

Run the following command to initialize Terraform and download the necessary provider plugins:

```
terraform init
```

3. Validate the Configuration

Ensure there are no syntax errors or issues with the configuration:

```
terraform validate
```

4. Generate an Execution Plan

Preview what Terraform will do without making changes yet:

```
terraform plan
```

This will display a plan showing that Terraform intends to create a resource group.

5. Apply the Configuration

Apply the changes to create the resource group in Azure:

```
terraform apply
```

When prompted, type yes to confirm the execution.

6. Verify in the Azure Portal

Log in to the Azure portal and navigate to **Resource Groups**. You should see the newly created resource group named **Terraform-Test-RG** in the selected location.

7. Clean Up (Optional)

To delete the resource group and clean up Azure resources created by Terraform, run:

```
terraform destroy
```

Type **yes** to confirm.

Lab2: Creating Azure Resources (RG, VNET, Subnet, PublicIP, VM)

Terraform Configuration File

1. Create the main.tf File

```
# Configure the Azure provider
provider "azurerm" {
  features {}

  subscription_id = "b15xxxx-xxxx-xxxx-xxxx-xxx" # Replace with your subscription ID
  tenant_id       = "0616xxx-xxxx-xxxxx-xxxx-xxxx"       # Replace with your tenant ID
}

# Create a Resource Group
resource "azurerm_resource_group" "linux_rg" {
  name     = "LinuxRG"
  location = "East US" # Replace with your preferred Azure region
}

# Create a Virtual Network
resource "azurerm_virtual_network" "linux_vnet" {
  name                = "LinuxVNet"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  address_space       = ["10.0.0.0/16"]
}

# Create a Subnet
resource "azurerm_subnet" "linux_subnet" {
  name                 = "LinuxSubnet"
  resource_group_name  = azurerm_resource_group.linux_rg.name
  virtual_network_name = azurerm_virtual_network.linux_vnet.name
  address_prefixes     = ["10.0.1.0/24"]
}

# Create a Public IP with Static Allocation
resource "azurerm_public_ip" "linux_public_ip" {
  name                = "LinuxPublicIP"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  allocation_method   = "Static" # Change from Dynamic to Static
  sku                 = "Standard" # Ensure SKU is Standard

}

# Create a Network Interface
```

```
resource "azurerm_network_interface" "linux_nic" {
  name                = "LinuxNIC"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name

  ip_configuration {
    name                          = "LinuxIPConfig"
    subnet_id                     = azurerm_subnet.linux_subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          = azurerm_public_ip.linux_public_ip.id
  }
}

# Create a Linux Virtual Machine
resource "azurerm_linux_virtual_machine" "Ubuntu_linux_vm" {
  name                = "UbuntuLinuxVM"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  size                = "Standard_B1ls" # Choose the desired size

  admin_username      = "azureuser"
  admin_password      = "SecurePassword123!" # Replace with your secure password

  network_interface_ids = [
    azurerm_network_interface.linux_nic.id,
  ]

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  source_image_reference {
    publisher = "Canonical"
    offer     = "ubuntu-24_04-lts"
    sku       = "ubuntu-pro"
    version   = "latest"
  }

  # Disable SSH key authentication
  disable_password_authentication = false
}
```

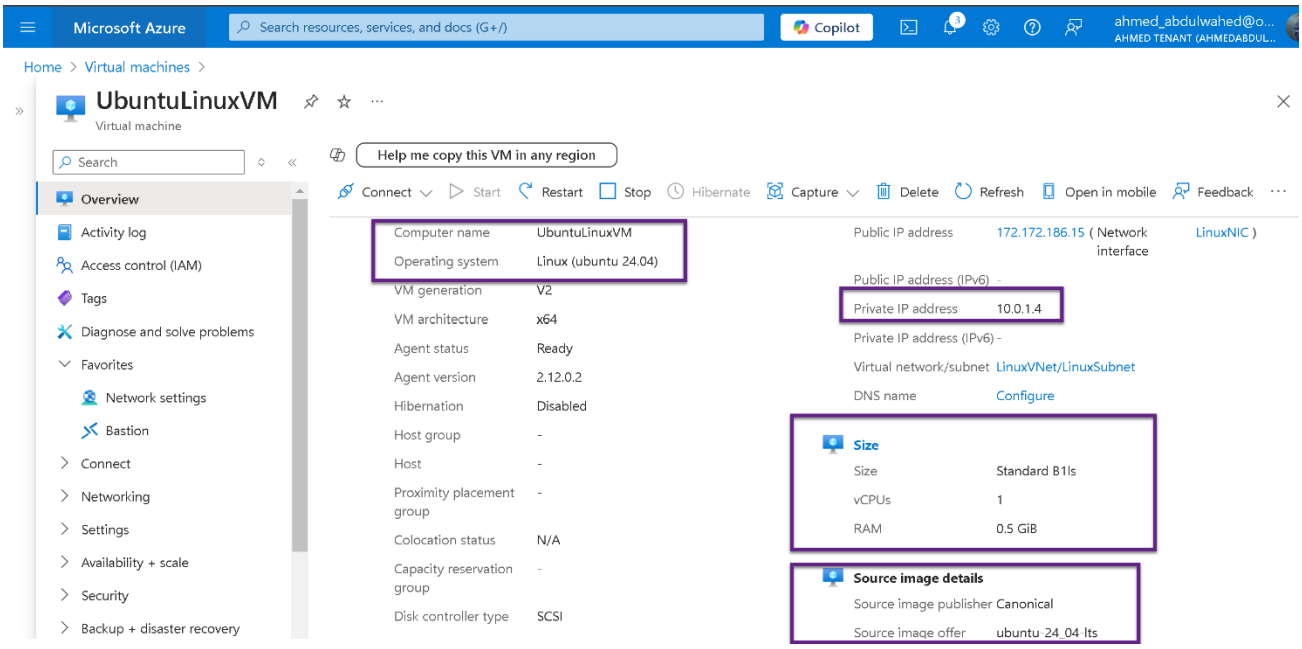
Steps to Deploy

1. **Initialize Terraform:**
   `terraform init`
2. **Validate the Configuration:**
   `terraform validate`
3. **Preview the Execution Plan:**
   `terraform plan -out=terraform-plan.out` # Save the plan to ensure consistency between 'plan' and 'apply'
4. **Apply the Configuration:**
   `terraform apply "terraform-plan.out"` # Apply the previously saved plan to execute the exact changes

Type yes to confirm the deployment.

What Will Be Created?

- A **Resource Group** named LinuxRG.
- A **Virtual Network** and **Subnet** for the VM.
- A **Public IP** and **Network Interface** for the VM.
- A **Ubuntu Linux VM** with default settings.



LAB3: Create 5 Linux VMs

Terraform Configuration File

Create the main.tf File

```
# Configure the Azure provider

provider "azurerm" {

  features {}

  subscription_id = "b15xxx-xxxxx-xxxx-xxxxxx-xxxxxxxxx" # Replace with your subscription
ID

  tenant_id       = "061xxx-xxxxx-xxxx-xxxxxx-xxxxxxxxx "       # Replace with your
tenant ID

}

# Create a Resource Group

resource "azurerm_resource_group" "linux_rg" {

  name     = "LinuxRG"

  location = "East US" # Replace with your preferred Azure region

}

# Create a Virtual Network

resource "azurerm_virtual_network" "linux_vnet" {

  name                = "LinuxVNet"

  location            = azurerm_resource_group.linux_rg.location

  resource_group_name = azurerm_resource_group.linux_rg.name

  address_space       = ["10.0.0.0/16"]

}

resource "azurerm_network_security_group" "linux_nsg" {

  name                = "LinuxNSG"

  location            = azurerm_resource_group.linux_rg.location

  resource_group_name = azurerm_resource_group.linux_rg.name

  security_rule {

    name                       = "AllowHTTP"

    priority                   = 100

    direction                  = "Inbound"
```

```
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "80"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }
  security_rule {
    name                       = "AllowSSH"
    priority                   = 110
    direction                  = "Inbound"
    access                     = "Allow"
    protocol                   = "Tcp"
    source_port_range          = "*"
    destination_port_range     = "22"
    source_address_prefix      = "*"
    destination_address_prefix = "*"
  }
}
# Create a Subnet
resource "azurerm_subnet" "linux_subnet" {
  name                 = "LinuxSubnet"
  resource_group_name  = azurerm_resource_group.linux_rg.name
  virtual_network_name = azurerm_virtual_network.linux_vnet.name
  address_prefixes     = ["10.0.1.0/24"]
}
# Associate the Subnet with the Network Security Group
resource "azurerm_subnet_network_security_group_association"
"linux_subnet_nsg_association" {
  subnet_id                 = azurerm_subnet.linux_subnet.id
```

```
  network_security_group_id = azurerm_network_security_group.linux_nsg.id
}
# Create Public IPs for the VMs
resource "azurerm_public_ip" "linux_public_ip" {
  count                = 5
  name                 = "LinuxPublicIP-${count.index + 1}"
  location             = azurerm_resource_group.linux_rg.location
  resource_group_name  = azurerm_resource_group.linux_rg.name
  allocation_method    = "Static"
  sku                  = "Standard"
}
# Create Network Interfaces for the VMs
resource "azurerm_network_interface" "linux_nic" {
  count               = 5
  name                = "LinuxNIC-${count.index + 1}"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  ip_configuration {
    name                          = "LinuxIPConfig"
    subnet_id                     = azurerm_subnet.linux_subnet.id
    private_ip_address_allocation = "Static"
    private_ip_address            = "10.0.1.${count.index + 5}"
    public_ip_address_id          = azurerm_public_ip.linux_public_ip[count.index].id
  }
}
resource "azurerm_network_interface_security_group_association" "nic_nsg_association" {
  count                     = 5
  network_interface_id      = azurerm_network_interface.linux_nic[count.index].id
  network_security_group_id = azurerm_network_security_group.linux_nsg.id
}
```

```
resource "azurerm_linux_virtual_machine" "linux_vm" {
  name                = "DebianVM-${count.index + 1}"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  size                = "Standard_B1ls"
  count               = 5
  admin_username = "azureuser"
  admin_password = "SecurePassword123!"
  network_interface_ids = [
    azurerm_network_interface.linux_nic[count.index].id,
  ]
  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }
  source_image_reference {
    publisher = "Debian"
    offer     = "debian-11"
    sku       = "11-backports-gen2"
    version   = "latest"
  }
  disable_password_authentication = false
  custom_data = base64encode(<<EOT
#cloud-config
package_update: true
packages:
  - apache2
  - ufw
runcmd:
  - ufw allow 80/tcp
```

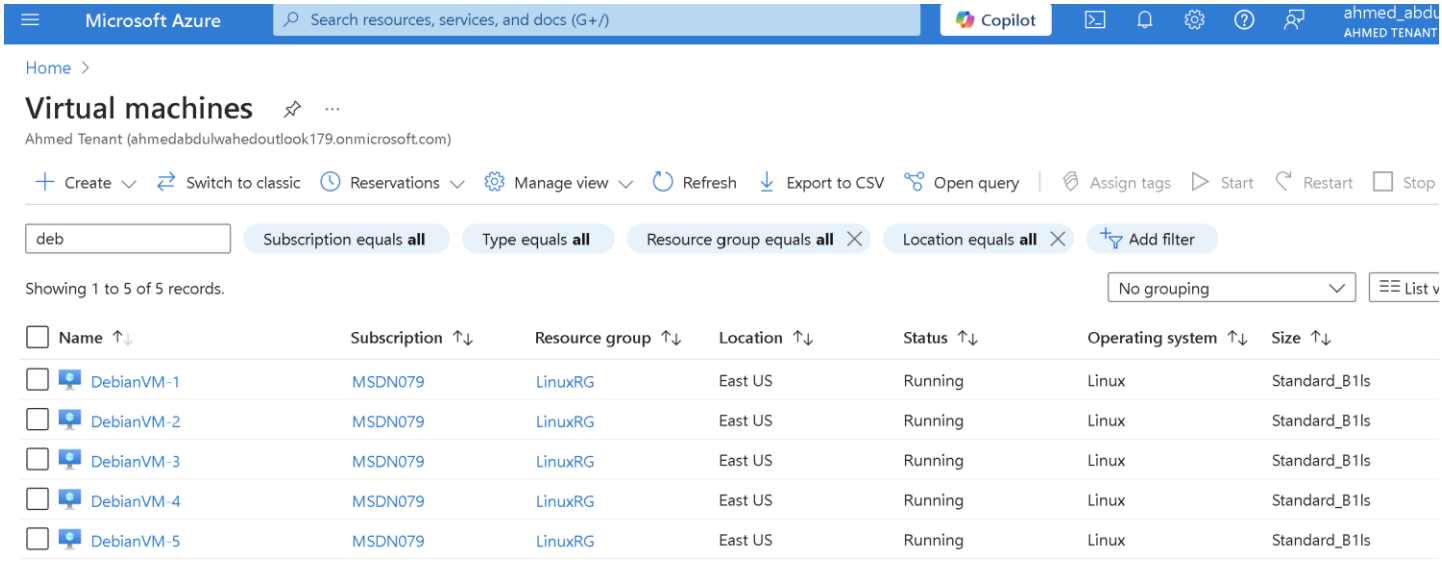
```
  - ufw --force enable
  - echo "<h1>Welcome to $(hostname)</h1>" > /var/www/html/index.html
  - systemctl restart apache2
EOT
  )
}
```

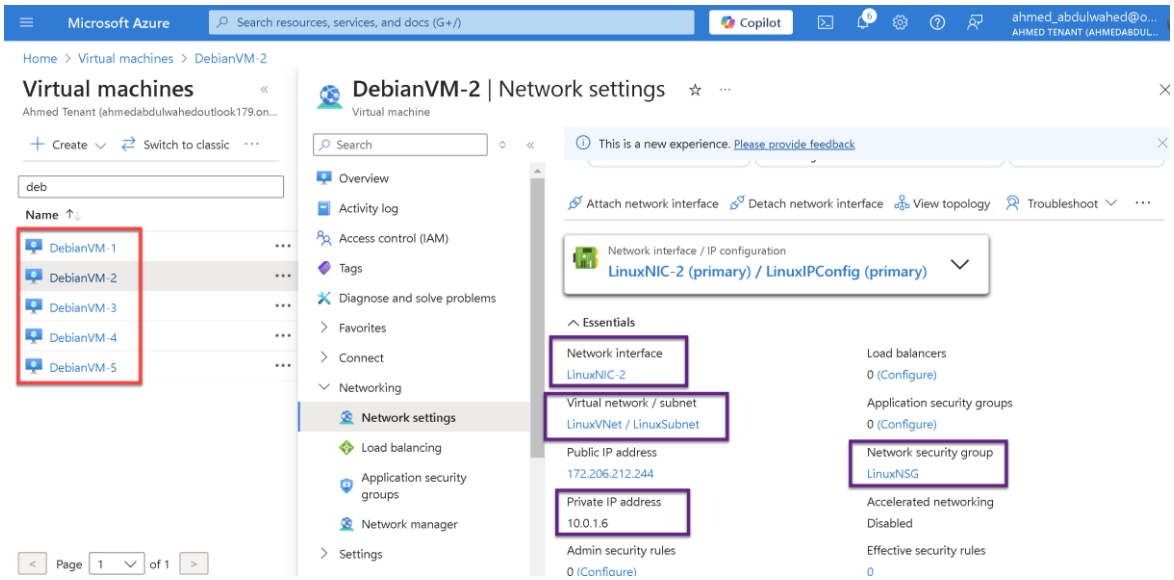
Steps to Deploy

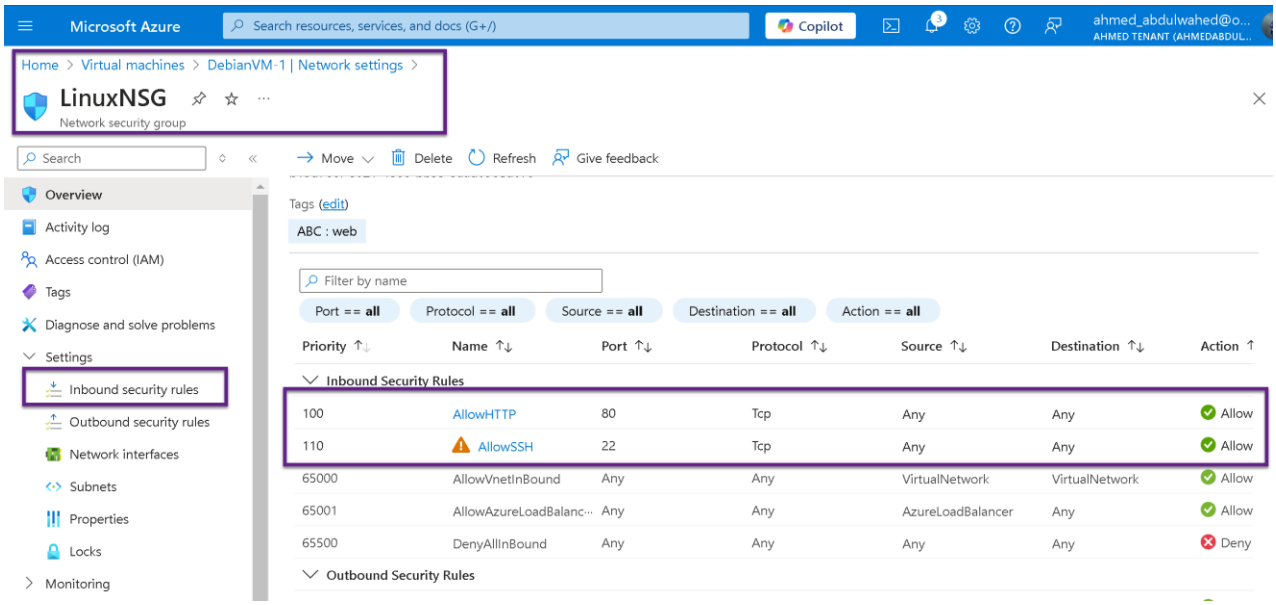
1. **Initialize Terraform:**
   `terraform init`
2. **Validate the Configuration:**
   `terraform validate`
3. **Preview the Execution Plan:**
   `terraform plan -out=terraform-plan.out` # Save the plan to ensure consistency between 'plan' and 'apply'
4. **Apply the Configuration:**
   `terraform apply "terraform-plan.out"` # Apply the previously saved plan to execute the exact changes

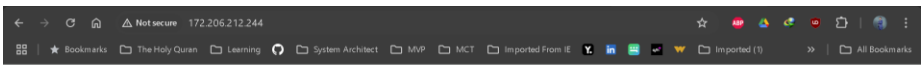
```
azurerm_linux_virtual_machine.linux_vm[2]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[4]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[2]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[1]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[3]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[4]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[0]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[3]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[4]: Creation complete after 21s [id=/subscriptions/b1
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-5]
azurerm_linux_virtual_machine.linux_vm[1]: Creation complete after 22s [id=/subscriptions/b1
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-2]
azurerm_linux_virtual_machine.linux_vm[0]: Creation complete after 23s [id=/subscriptions/b1
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-1]
azurerm_network_interface_security_group_association.nic_nsg_association[2]: Creation complete after 27s [id=/subscriptions/b1
-                    )/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-3|/subscriptions/b1
-                    )/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]
azurerm_linux_virtual_machine.linux_vm[2]: Creation complete after 28s [id=/subscriptions/b15d766f-8021-4866-bb33-5aad096ed079/resc
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-3]
azurerm_linux_virtual_machine.linux_vm[3]: Still creating... [30s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[4]: Still creating... [30s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[3]: Still creating... [30s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[4]: Creation complete after 38s [id=/subscriptions/b
                    79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-5|/subscriptions/b
                    79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]
azurerm_linux_virtual_machine.linux_vm[3]: Still creating... [40s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[3]: Still creating... [40s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[3]: Creation complete after 46s [id=/subscriptions/b
-                    79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-4|/subscriptions/b
-                    79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]
azurerm_linux_virtual_machine.linux_vm[3]: Still creating... [50s elapsed]
azurerm_linux_virtual_machine.linux_vm[3]: Creation complete after 59s [id=/subscriptions/b15d766f-8021-4866-bb33-5aad096ed079/resc
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-4]

Apply complete! Resources: 25 added, 0 changed, 0 destroyed.
```









Welcome to DebianVM-2



Lab 4: Create 5 Debian VMs with Apache and Azure Loadbalancer

Terraform Configuration File

Create the main.tf File

```
# Configure the Azure provider
provider "azurerm" {
  features {}
  subscription_id = "b15xxxxx-xxxxx-xxxx-xxxxxx-xxxxxxxxx " # Replace with your subscription ID
  tenant_id       = "061xxx-xxxxx-xxxx-xxxxxx-xxxxxxxxx " # Replace with your tenant ID
}
# Create a Resource Group
resource "azurerm_resource_group" "linux_rg" {
  name     = "LinuxRG"
  location = "East US" # Replace with your preferred Azure region
}
# Create a Virtual Network
resource "azurerm_virtual_network" "linux_vnet" {
  name                = "LinuxVNet"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  address_space       = ["10.0.0.0/16"]
}
# Network Security Group
resource "azurerm_network_security_group" "linux_nsg" {
  name                = "LinuxNSG"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  security_rule {
    name                       = "AllowHTTP"
```

```
    priority                    = 100
    direction                   = "Inbound"
    access                      = "Allow"
    protocol                    = "Tcp"
    source_port_range           = "*"
    destination_port_range      = "80"
    source_address_prefix       = "*"
    destination_address_prefix = "*"
  }
  security_rule {
    name                        = "AllowSSH"
    priority                    = 110
    direction                   = "Inbound"
    access                      = "Allow"
    protocol                    = "Tcp"
    source_port_range           = "*"
    destination_port_range      = "22"
    source_address_prefix       = "*"
    destination_address_prefix = "*"
  }
}
# Create a Subnet
resource "azurerm_subnet" "linux_subnet" {
  name                 = "LinuxSubnet"
  resource_group_name  = azurerm_resource_group.linux_rg.name
  virtual_network_name = azurerm_virtual_network.linux_vnet.name
  address_prefixes     = ["10.0.1.0/24"]
}
# Associate the Subnet with the Network Security Group
```

```
resource "azurerm_subnet_network_security_group_association"
"linux_subnet_nsg_association" {
  subnet_id                 = azurerm_subnet.linux_subnet.id
  network_security_group_id = azurerm_network_security_group.linux_nsg.id
}
# Create Public IPs for the VMs
resource "azurerm_public_ip" "linux_public_ip" {
  count                = 5
  name                 = "LinuxPublicIP-${count.index + 1}"
  location             = azurerm_resource_group.linux_rg.location
  resource_group_name  = azurerm_resource_group.linux_rg.name
  allocation_method    = "Static"
  sku                  = "Standard"
}
# Create Network Interfaces for the VMs
resource "azurerm_network_interface" "linux_nic" {
  count               = 5
  name                = "LinuxNIC-${count.index + 1}"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  ip_configuration {
    name                          = "LinuxIPConfig"
    subnet_id                     = azurerm_subnet.linux_subnet.id
    private_ip_address_allocation = "Static"
    private_ip_address            = "10.0.1.${count.index + 5}"
    public_ip_address_id          = azurerm_public_ip.linux_public_ip[count.index].id
  }
}
resource "azurerm_network_interface_security_group_association" "nic_nsg_association" {
  count                     = 5
```

```
  network_interface_id      = azurerm_network_interface.linux_nic[count.index].id
  network_security_group_id = azurerm_network_security_group.linux_nsg.id
}
# Create Linux Virtual Machines
resource "azurerm_linux_virtual_machine" "linux_vm" {
  count               = 5
  name                = "DebianVM-${count.index + 1}"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  size                = "Standard_B1ls"
  admin_username = "azureuser"
  admin_password = "SecurePassword123!"
  network_interface_ids = [
    azurerm_network_interface.linux_nic[count.index].id,
  ]
  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }
  source_image_reference {
    publisher = "Debian"
    offer     = "debian-11"
    sku       = "11-backports-gen2"
    version   = "latest"
  }
  disable_password_authentication = false
  custom_data = base64encode(<<EOT
#cloud-config
package_update: true
packages:
```

```
  - apache2
  - ufw
runcmd:
  - ufw allow 80/tcp
  - ufw --force enable
  - echo "<h1>Welcome to $(hostname)</h1>" > /var/www/html/index.html
  - systemctl restart apache2
EOT
  )
}
# Create the Load Balancer Public IP
resource "azurerm_public_ip" "lb_public_ip" {
  name                = "LoadBalancerPublicIP"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  allocation_method   = "Static"
  sku                 = "Standard"
}
# Create the Load Balancer
resource "azurerm_lb" "linux_lb" {
  name                = "LinuxLoadBalancer"
  location            = azurerm_resource_group.linux_rg.location
  resource_group_name = azurerm_resource_group.linux_rg.name
  sku                 = "Standard"
  frontend_ip_configuration {
    name                 = "LoadBalancerFrontEnd"
    public_ip_address_id = azurerm_public_ip.lb_public_ip.id
  }
}
# Backend Pool for the Load Balancer
```

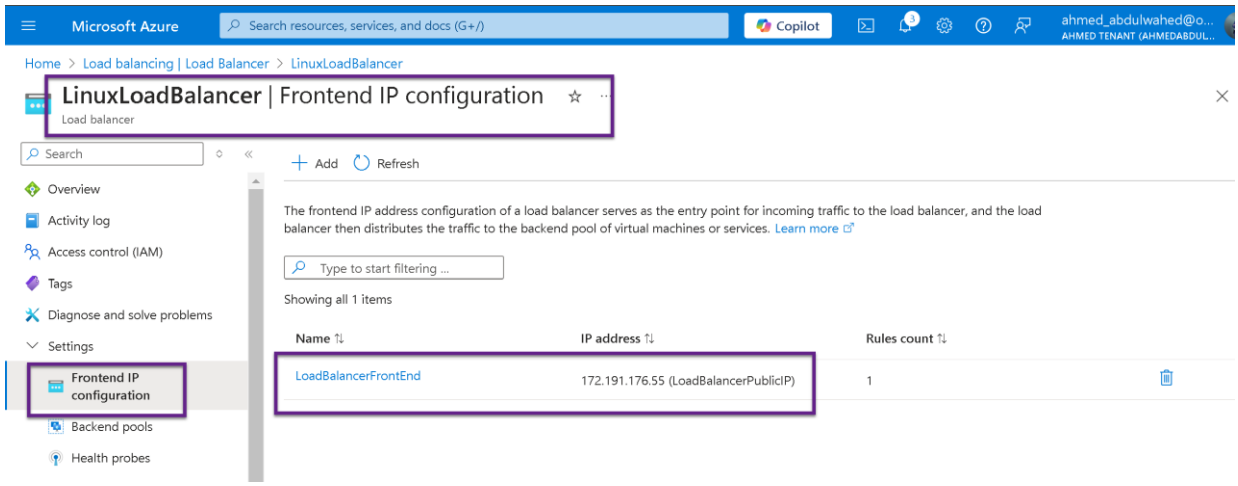
```
resource "azurerm_lb_backend_address_pool" "lb_backend_pool" {
  name            = "LoadBalancerBackendPool"
  loadbalancer_id = azurerm_lb.linux_lb.id
}
# Associate NICs with Load Balancer Backend Pool
resource "azurerm_network_interface_backend_address_pool_association"
"nic_backend_pool_association" {
  count                     = 5
  network_interface_id      = azurerm_network_interface.linux_nic[count.index].id
  backend_address_pool_id   = azurerm_lb_backend_address_pool.lb_backend_pool.id
  ip_configuration_name     = "LinuxIPConfig" # Add this line to define the IP
configuration name
}
# Load Balancer Rule
resource "azurerm_lb_rule" "http_lb_rule" {
  name                           = "HTTPRule"
  loadbalancer_id                = azurerm_lb.linux_lb.id
  frontend_ip_configuration_name = "LoadBalancerFrontEnd"
  backend_address_pool_ids       = [azurerm_lb_backend_address_pool.lb_backend_pool.id]
  protocol                       = "Tcp"
  frontend_port                  = 80
  backend_port                   = 80
  enable_tcp_reset               = true
  idle_timeout_in_minutes        = 4
}
```

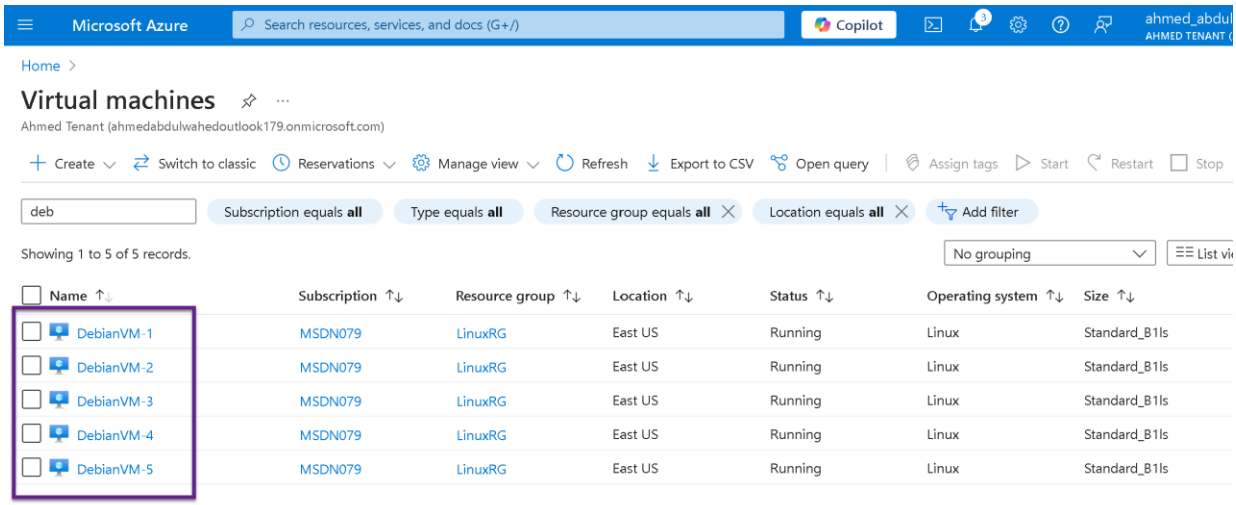
Steps to Deploy

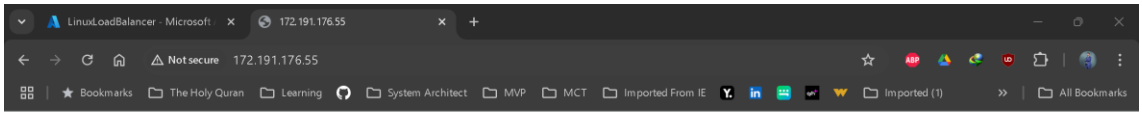
1. **Initialize Terraform:**
   `terraform init`
2. **Validate the Configuration:**
   `terraform validate`
3. **Preview the Execution Plan:**
   `terraform plan -out=terraform-plan.out` # Save the plan to ensure consistency between 'plan' and 'apply'
4. **Apply the Configuration:**
   `terraform apply "terraform-plan.out"` # Apply

```
azurerm_linux_virtual_machine.linux_vm[3]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[0]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[2]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[2]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[2]: Creation complete after 21s [id=/subscriptions/b15
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-3]
azurerm_linux_virtual_machine.linux_vm[4]: Creation complete after 22s [id=/subscriptions/b15
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-5]
azurerm_linux_virtual_machine.linux_vm[0]: Creation complete after 23s [id=/subscriptions/b15
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-1]
azurerm_linux_virtual_machine.linux_vm[3]: Creation complete after 23s [id=/subscriptions/b15
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-4]
azurerm_network_interface_backend_address_pool_association.nic_backend_pool_association[4]: Creation complete after 16s [id=/subs
ptions/b1
iders/Microsoft.Network/networkInterfaces/LinuxNIC-5/ipCon
urations/
079/resourceGroups/LinuxRG/providers/Microsoft.Network/loa
lancers/LinuxLoadBalancer/backendAddressPools/LoadBalancerBackendPool]
azurerm_network_interface_security_group_association.nic_nsg_association[0]: Still creating... [10s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[2]: Creation complete after 27s [id=/subscrip
-8021-4866-bb33-5aad096ed079/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-3|/subscrip
-8021-4866-bb33-5aad096ed079/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]
azurerm_linux_virtual_machine.linux_vm[1]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[1]: Still creating... [20s elapsed]
azurerm_network_interface_security_group_association.nic_nsg_association[1]: Creation complete after 25s [id=/subscrip
-8021-4866-bb33-5aad096ed079/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-2|/subscrip
-8021-4866-bb33-5aad096ed079/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]
azurerm_network_interface_security_group_association.nic_nsg_association[0]: Still creating... [20s elapsed]
azurerm_linux_virtual_machine.linux_vm[1]: Creation complete after 30s [id=/subscriptions/b15d766f-8021-4866-bb33-5aad
rceGroups/LinuxRG/providers/Microsoft.Compute/virtualMachines/DebianVM-2]
azurerm_network_interface_security_group_association.nic_nsg_association[0]: Creation complete after 27s [id=/subscrip
-80
79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkInterfaces/LinuxNIC-1|/subscrip
-80
79/resourceGroups/LinuxRG/providers/Microsoft.Network/networkSecurityGroups/LinuxNSG]

Apply complete! Resources: 34 added, 0 changed, 0 destroyed.
PS D:\TerraformProject\loadbalancer>
```







Welcome to DebianVM-2

