**Creating Network Resources in Azure with Terraform**

**Introduction**

 1. First, we will log into the Azure Portal and configure the Cloud Shell, and then download and run the lab setup script to setup the lab.  1. Second, we will import the resource group.  1. Third, we will add the virtual network, subnet, and a network security group.  1. And for the fourth objective, we will add a load balancer.

These objectives will build the foundation for VMs and other resources.

**Solution**

**Set Up Cloud Shell and the Lab Environment**

1. In the Azure Portal, click the Cloud Shell icon (>\_) in the upper right. Take note of the **Location** for the lab environment.
2. Select **Bash**.
3. Click **Show advanced settings**.
4. If needed, change the **Cloud Shell region** to match the **Location** of the lab environment.
5. For *Storage account*, select **Use existing**.
6. For **File share**, select **Create new** and give it a name of *terraform*.
7. Click **Create storage**.

**Import the Resource Group into Terraform**

1. Using az group list, find and copy everything within the quotes for the subscription id:

az group list

1. Using the resource name and label from the networking.tf file, as well as the subscription id copied earlier, import the Azure resource group:

terraform import azurerm\_resource\_group.comics <SUBSCRIPTION\_ID>

**Add the Networking Configuration**

1. Open the networking.tf file:

vim networking.tf

1. In the Azure Portal, copy the **Resource group** name.
2. In the Cloud Shell, on line 2 of the networking.tf file, uncomment the line by deleting the # and replace <RESOURCE\_GROUP\_NAME> with the **Resource group** name that was just copied.
3. On line 3, uncomment the line by deleting the # and replace <LOCATION> with the location for the lab environment, using westus for West US, centralus for Central US, and so on.
4. Starting on line 6, add a new azurerm\_virtual\_network resource using the following configuration:

* *label:* comics
* *name:* vnet
* *address\_space:* 10.0.0.0/16
* *resource\_group\_name:* azurerm\_resource\_group.comics.name
* *location:* azurerm\_resource\_group.comics.location

1. The resource should look like the example below:

resource "azurerm\_virtual\_network" "comics" {

name = "vnet"

address\_space = ["10.0.0.0/16"]

resource\_group\_name = azurerm\_resource\_group.comics.name

location = azurerm\_resource\_group.comics.location

}

1. Starting on line 13, add a new azurerm\_subnet resource using the following configuration:

* *label:* comics
* *name:* subnet
* *resource\_group\_name:* azurerm\_resource\_group.comics.name
* *virtual\_network\_name:* azurerm\_virtual\_network.comics.name
* *address\_prefix:* 10.0.2.0/24

1. The resource should look like the example below:

resource "azurerm\_subnet" "comics" {

name = "subnet"

resource\_group\_name = azurerm\_resource\_group.comics.name

virtual\_network\_name = azurerm\_virtual\_network.comics.name

address\_prefixes = ["10.0.0.0/24"]

}

1. Starting on line 20, add an azurerm\_public\_ip resource using the following configuration:

* *label:* comics
* *name:* publicIpForLB
* *resource\_group\_name:* azurerm\_resource\_group.comics.name
* *location:* azurerm\_resource\_group.comics.location
* *allocation\_method:* Static

1. The resource should look like the example below:

resource "azurerm\_public\_ip" "comics" {

name = "publicIpForLB"

resource\_group\_name = azurerm\_resource\_group.comics.name

location = azurerm\_resource\_group.comics.location

allocation\_method = "Static"

}

1. Save and exit the file:

ESC :wq!

**Add the Load Balancer Configuration**

1. Create a new file named lb.tf:

vim lb.tf

1. At the top of the file, add an azurerm\_lb load balancer resource using the following configuration:

* *label:* comics
* *name:* loadBalancer
* *resource\_group\_name:* azurerm\_resource\_group.comics.name
* *location:* azurerm\_resource\_group.comics.location

1. The resource should also include a frontend\_ip\_configuration using the public\_ip\_address\_id created in the networking.tf file.
2. The resource should look like the example below:

resource "azurerm\_lb" "comics" {

name = "loadBalancer"

resource\_group\_name = azurerm\_resource\_group.comics.name

location = azurerm\_resource\_group.comics.location

frontend\_ip\_configuration {

name = "publicIPAddress"

public\_ip\_address\_id = azurerm\_public\_ip.comics.id

}

}

1. Starting on line 12, add an azurerm\_lb\_backend\_address\_pool resource using the following configuration:

* *label:* comics
* *loadbalancer\_id:* azurerm\_lb.comics.id
* *name:* BackendAddressPool

1. The resource should look like the example below:

resource "azurerm\_lb\_backend\_address\_pool" "comics" {

loadbalancer\_id = azurerm\_lb.comics.id

name = "BackendAddressPool"

}

1. Save and exit the file:

ESC :wq!

1. Create a new file named output.tf:

vim output.tf

1. In the file, add the following outputs:

output "resource\_group\_name" {

value = azurerm\_resource\_group.comics.name

}

output "vnet\_address\_space" {

value = azurerm\_virtual\_network.comics.address\_space[0]

}

output "subnet\_prefixes" {

value = azurerm\_subnet.comics.address\_prefixes[0]

}

output "public\_ip\_address" {

value = azurerm\_public\_ip.comics.ip\_address

}

output "location" {

value = azurerm\_resource\_group.comics.location

}

1. Save and exit the file:

ESC :wq!

**Apply the Configuration**

1. Verify the files are formatted correctly:

terraform fmt

1. Validate the files:

terraform validate

1. Do a dry run of the configuration:

terraform plan

1. Apply the configuration:

terraform apply

1. When prompted, type yes to confirm.
2. Once the deployment is complete, use terraform state list to confirm that the resources were deployed successfully.

**Conclusion**