Lab 3: Creating Virtual Network (VNet) and Subnet in Microsoft Azure using Terraform

Author: Dr. Sandeep Kumar Sharma
Level: Beginner
Platform: Ubuntu Linux + Microsoft Azure
Prerequisite: Lab 1 (Setup) + Lab 2 (Resource Group)

Learning Objective

By the end of this lab, participants will be able to:

- Understand what a Virtual Network (VNet) is
- Understand what a Subnet is
- Learn why networking is important in cloud
- Create a Virtual Network using Terraform
- Create a Subnet inside a Virtual Network
- Understand resource dependency in Terraform

Learning Outcome

After completing this lab, participants will:

- Clearly understand Azure networking basics
- Be able to create network infrastructure using Terraform
- Understand parent-child resource relationships
- Build confidence in cloud networking concepts

First Important Concepts

What is a Virtual Network (VNet)?

A **Virtual Network (VNet)** is a private network in Microsoft Azure.

It is used to connect Azure resources like: - Virtual Machines - Databases - Containers - Load Balancers

Simple Meaning:

A VNet is like a **private network inside Azure**, just like WiFi network in your home.

What is a Subnet?

A **Subnet** is a small network inside a Virtual Network.

Simple Meaning:

If VNet is a **big land**, then Subnet is a **plot inside that land**.

Real-Life Example

Real World	Azure
City	Virtual Network (VNet)
Society	Subnet
Houses	Virtual Machines

Hands-On Lab

Step 1: Go to Terraform Directory

```
cd terraform-azure-lab
```

Step 2: Create Terraform File for Network

```
touch network.tf
```

Step 3: Open File

```
nano network.tf
```

Step 4: Write Terraform Code for VNet and Subnet

Paste the following code:

```
resource "azurerm_virtual_network" "vnet1" {
  name                = "vnet-terraform-lab"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.rg1.location
  resource_group_name = azurerm_resource_group.rg1.name
}

resource "azurerm_subnet" "subnet1" {
  name                  = "subnet-terraform-lab"
  resource_group_name   = azurerm_resource_group.rg1.name
  virtual_network_name = azurerm_virtual_network.vnet1.name
  address_prefixes      = ["10.0.1.0/24"]
}
```

Step 5: Understand the Code (Beginner Friendly)

Virtual Network Block

```
resource "azurerm_virtual_network" "vnet1"
```

- Creates Azure Virtual Network

```
address_space = ["10.0.0.0/16"]
```

- IP range for network

```
location = azurerm_resource_group.rg1.location
```

- Same region as Resource Group

```
resource_group_name = azurerm_resource_group.rg1.name
```

- Attach VNet to Resource Group

Subnet Block

```
resource "azurerm_subnet" "subnet1"
```

- Creates subnet

```
virtual_network_name = azurerm_virtual_network.vnet1.name
```

- Subnet inside VNet

```
address_prefixes = ["10.0.1.0/24"]
```

- Subnet IP range

Dependency Concept

Terraform automatically understands:

VNet must be created before Subnet

This is called **resource dependency**.

Step 6: Initialize Terraform

```
terraform init
```

Step 7: Preview Changes

```
terraform plan
```

Step 8: Apply Configuration

```
terraform apply
```

Type:

```
yes
```

Step 9: Verify in Azure Portal

Go to:

Resource Group → `rg-terraform-lab`

You will see: - Virtual Network: `vnet-terraform-lab` - Subnet: `subnet-terraform-lab`

Step 10: Verify using Azure CLI

```
az network vnet list -o table
```

```
az network vnet subnet list --resource-group rg-terraform-lab --vnet-name vnet-
terraform-lab -o table
```

Beginner Summary

- VNet = Private network in Azure
- Subnet = Small network inside VNet
- VNet is parent, Subnet is child
- Terraform handles dependencies automatically
- Networking is foundation of cloud

Cloud Foundation Concept

Without networking: - No VM communication - No internet access - No application connectivity

Next Lab Preview

Lab 4: - Create Network Security Group (NSG) - Create Security Rules - Attach NSG to Subnet

Trainer Note:
This lab builds cloud networking foundation. Participants must clearly understand VNet and Subnet before moving to VM creation.