**Step-by-Step Guide**

1. **Initialize Terraform Configuration**:
   * Define the required providers.
   * Create resources (Resource Group and Virtual Network).
   * Import existing resources into Terraform.

**1. Define Providers**

Create a file named main.tf and configure the Azure provider.

provider "azurerm" {

features {}

}

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = ">= 3.0.0"

}

}

}

**2. Create Resources**

Add the resource definitions for the Resource Group and the Virtual Network.

resource "azurerm\_resource\_group" "example" {

name = "example-resources"

location = "East US"

}

resource "azurerm\_virtual\_network" "example" {

name = "example-vnet"

location = azurerm\_resource\_group.example.location

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

address\_space = ["10.0.0.0/16"]

}

**3. Import Existing Resources**

To import an existing resource, you'll need its resource ID. For example, if you have an existing resource group named existing-resources, you can import it into the Terraform state.

**Step-by-Step to Import**

1. **Identify Resource ID**:
   * For a resource group, the ID format is /subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}.
   * For a virtual network, the ID format is /subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Network/virtualNetworks/{vnetName}.
2. **Import Resource**:
   * Use the Terraform import command to bring existing resources under Terraform management.

First, define the resources you intend to import in your configuration file:

resource "azurerm\_resource\_group" "existing" {

name = "existing-resources"

location = "East US"

}

resource "azurerm\_virtual\_network" "existing" {

name = "existing-vnet"

location = azurerm\_resource\_group.existing.location

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

address\_space = ["10.0.0.0/16"]

}

Now, import these resources using the Terraform CLI:

# Replace {subscriptionId} with your Azure Subscription ID

**terraform import azurerm\_resource\_group.existing** **/subscriptions/{subscriptionId}/resourceGroups/existing-resources**

# Replace {subscriptionId} with your Azure Subscription ID and {vnetName} with your VNet name

**terraform import azurerm\_virtual\_network.existing /subscriptions/{subscriptionId}/resourceGroups/existing-resources/providers/Microsoft.Network/virtualNetworks/{vnetName}**

**Complete main.tf File**

Here's the complete main.tf file with both new and existing resources:

provider "azurerm" {

features {}

}

terraform {

required\_providers {

azurerm = {

source = "hashicorp/azurerm"

version = ">= 3.0.0"

}

}

}

resource "azurerm\_resource\_group" "example" {

name = "example-resources"

location = "East US"

}

resource "azurerm\_virtual\_network" "example" {

name = "example-vnet"

location = azurerm\_resource\_group.example.location

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

address\_space = ["10.0.0.0/16"]

}

resource "azurerm\_resource\_group" "existing" {

name = "existing-resources"

location = "East US"

}

resource "azurerm\_virtual\_network" "existing" {

name = "existing-vnet"

location = azurerm\_resource\_group.existing.location

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

address\_space = ["10.0.0.0/16"]

}

**Running Terraform Commands**

1. **Initialize Terraform**:

terraform init

1. **Validate the Configuration**:

terraform validate

1. **Plan the Deployment**:

terraform plan

1. **Apply the Configuration**:

terraform apply

By following these steps, you will create new resources and import existing ones into your Terraform state for management. This process ensures that both new and existing Azure resources are handled consistently within your infrastructure as code practices