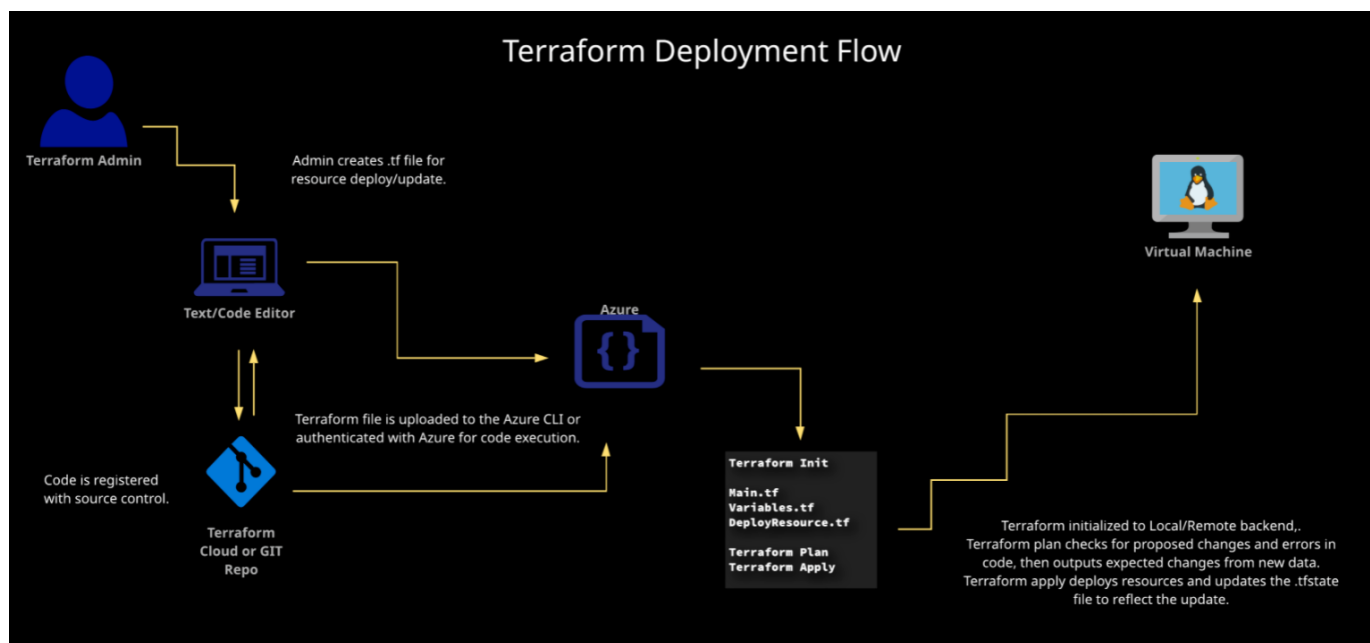


# Deploying an Azure VM with Terraform

## Introduction

Deploying VMs is the primary target for almost all attempts at automation. Terraform provides a much more human-readable syntax than that of an ARM template, and allows for the creation of any prerequisites you may want, or need without having to jump around multiple portal screens.

In this hands-on lab, we will take what we've done in the previous labs and add in the creation of a NIC and a marketplace Ubuntu VM. Additionally, we will create a Boot Diagnostics storage account to support the use of the serial console, providing a quick way to test the VM's deployment.



## Set Up the Command Line Interface (CLI)

- 1- In the Azure Portal, in the top-left corner of the page, copy the Resource group name for later use.
- 2- Click the Cloud Shell icon (>\_) in the upper right.
- 3- Select PowerShell.
- 4- Click Show advanced settings.

5- For a Storage account, select Create new and give it a globally unique name (e.g., "cloudshell" with a series of numbers at the end). Copy this name for later use.

6- For File share, select Create new and give it a name of "fileshare1".

7- Click Create storage.

## Deploy a Ubuntu VM

1- Copy the following codeblock and in a text editor update the fields using the information copied earlier:

```
provider "azurerm" {
  version = 1.38
}

# Create virtual network
resource "azurerm_virtual_network" "TFNet" {
  name                = "<UNIQUE_STORAGE_ACCOUNT_NAME>"
  address_space       = ["10.0.0.0/16"]
  location             = "East US"
  resource_group_name = "<RESOURCE_GROUP_NAME>"

  tags = {
    environment = "Terraform VNET"
  }
}

# Create subnet
resource "azurerm_subnet" "tfsubnet" {
  name                = "default"
  resource_group_name = "<RESOURCE_GROUP_NAME>"
  virtual_network_name = azurerm_virtual_network.TFNet.name
  address_prefix       = "10.0.1.0/24"
}

#Deploy Public IP
resource "azurerm_public_ip" "example" {
  name                = "pubip1"
  location             = "East US"
  resource_group_name = "<RESOURCE_GROUP_NAME>"
  allocation_method    = "Dynamic"
  sku                  = "Basic"
}

#Create NIC
resource "azurerm_network_interface" "example" {
  name                = "Enter name for this NIC"
  location             = "East US"
  resource_group_name = "<RESOURCE_GROUP_NAME>"

  ip_configuration {
```

```

        name                        = "ipconfig1"
        subnet_id                   = azurerm_subnet.tfsubnet.id
        private_ip_address_allocation = "Dynamic"
        public_ip_address_id         = azurerm_public_ip.example.id
    }
}

```

#Create Boot Diagnostic Account

```

resource "azurerm_storage_account" "sa" {
    name                        = "Enter Name for Diagnostic Account"
    resource_group_name        = "<RESOURCE_GROUP_NAME>"
    location                   = "East US"
    account_tier                = "Standard"
    account_replication_type    = "LRS"

    tags = {
        environment = "Boot Diagnostic Storage"
        CreatedBy   = "Admin"
    }
}

```

#Create Virtual Machine

```

resource "azurerm_virtual_machine" "example" {
    name                        = "Enter AzureVM Name"
    location                   = "East US"
    resource_group_name        = "Enter Resource Group Name"
    network_interface_ids      = [azurerm_network_interface.example.id]
    vm_size                    = "Standard_B1s"
    delete_os_disk_on_termination = true
    delete_data_disks_on_termination = true

    storage_image_reference {
        publisher = "Canonical"
        offer     = "UbuntuServer"
        sku       = "16.04-LTS"
        version    = "latest"
    }

    storage_os_disk {
        name            = "osdisk1"
        disk_size_gb    = "128"
        caching          = "ReadWrite"
        create_option    = "FromImage"
        managed_disk_type = "Standard_LRS"
    }

    os_profile {
        computer_name  = "Enter Server Name"
        admin_username = "vmadmin"
        admin_password = "Password12345!"
    }

    os_profile_linux_config {
        disable_password_authentication = false
    }
}

```

```
    }

    boot_diagnostics {
        enabled      = "true"
        storage_uri =
azurerm_storage_account.sa.primary_blob_endpoint
    }
}
```

**Note:** Each resource will need the *resource\_group\_name* field to be updated, and several still require a unique name. Update each "Enter Name" field with a unique name before proceeding.

2- Once updated, save the file as "test.tf".

3- In the Cloud Shell, click the Upload/Download files menu and click Upload.

4- Find the test.tf file and click Open.

5- Click the Open Editor button and review the test.tf file.

6- In the top-right corner of the Editor, click the ellipses button to open the More menu, and click Close Editor.

7- In the Cloud Shell, initialize the working directory:

```
terraform init
```

8- Create the execution plan:

```
terraform plan
```

9- Apply the execution plan:

```
terraform apply
```

10- Once complete, back in the Azure Portal, click the Refresh button and review the resources that were created.

11- Click the robot virtual machine.

12- In the left-hand menu, under Support + Troubleshooting, click Serial console.