

---

# **Cloud & Automation Class**

IaC with Terraform & Azure

Adam Lahbib - Skander Soltane - Mouna Rouini - Mariem Cherif

4/27/2023 @ INSAT

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Overview . . . . .	3
1.2	Topics Covered . . . . .	3
1.3	Lab Walkthrough . . . . .	3
1.3.1	Task 1 - Definitions . . . . .	3
1.3.2	Task 2 – Terraform Installation on Linux . . . . .	4
1.3.3	Task 3 – Azure CLI Installation on Linux . . . . .	5
1.3.4	Task 4 - Change the infrastructure . . . . .	10
1.3.5	Task 5 - Destroy the infrastructure . . . . .	13
1.3.6	Task 6 - Terraform Input Variables . . . . .	13
1.3.7	Task 7 - Terraform Output Variables . . . . .	15

# 1 Introduction

## 1.1 Overview

In this lab, we will learn how to use Terraform to provision and manage infrastructure resources in Azure. We will start by installing Terraform and Azure CLI on a Linux machine, and then we will create a service principal in Azure to authenticate Terraform to access Azure resources. Finally, we will write a Terraform configuration to provision a virtual machine in Azure.

## 1.2 Topics Covered

- Infrastructure as Code (IaC)
- Terraform
- Azure Resource Manager
- Azure CLI
- Service Principal
- Terraform Configuration
- Terraform Commands
- Terraform State
- Terraform Variables

## 1.3 Lab Walkthrough

### 1.3.1 Task 1 - Definitions

**Define Infrastructure as a code, terraform functionalities and Azure Resource Manager (or any equivalent).**

Infrastructure as Code (IaC) is a practice of managing and provisioning infrastructure through machine-readable definition files, instead of manual configuration. This approach brings benefits such as repeatability, consistency, and version control to infrastructure management.

Terraform is a tool for building, changing, and versioning infrastructure. It uses a declarative language to define resources, dependencies, and configurations in a portable and reproducible manner. Terraform provides a range of functionalities such as provisioning, orchestration, and deployment of infrastructure.

Azure Resource Manager is a service provided by Microsoft Azure that enables the deployment and management of Azure resources in a consistent manner. It provides a unified API and management plane to create, update, and delete resources, as well as manage their lifecycle and access control. Similar services exist in other cloud providers, such as AWS CloudFormation or Google Cloud Deployment Manager.

### **Explain the approach of IaC and its benefits**

Infrastructure as Code (IaC) is an approach to infrastructure management that involves writing and managing machine-readable definition files, instead of manually configuring infrastructure components. This approach is gaining popularity due to its many benefits, including:

**Repeatability:** IaC allows for the creation of identical infrastructure environments across different stages of the development pipeline, ensuring consistency and reducing errors.

**Consistency:** Infrastructure configurations can be stored in version control, allowing teams to manage changes and collaborate on infrastructure changes in a controlled and consistent manner.

**Scalability:** With IaC, teams can easily scale infrastructure resources up or down as needed, without manual intervention or configuration.

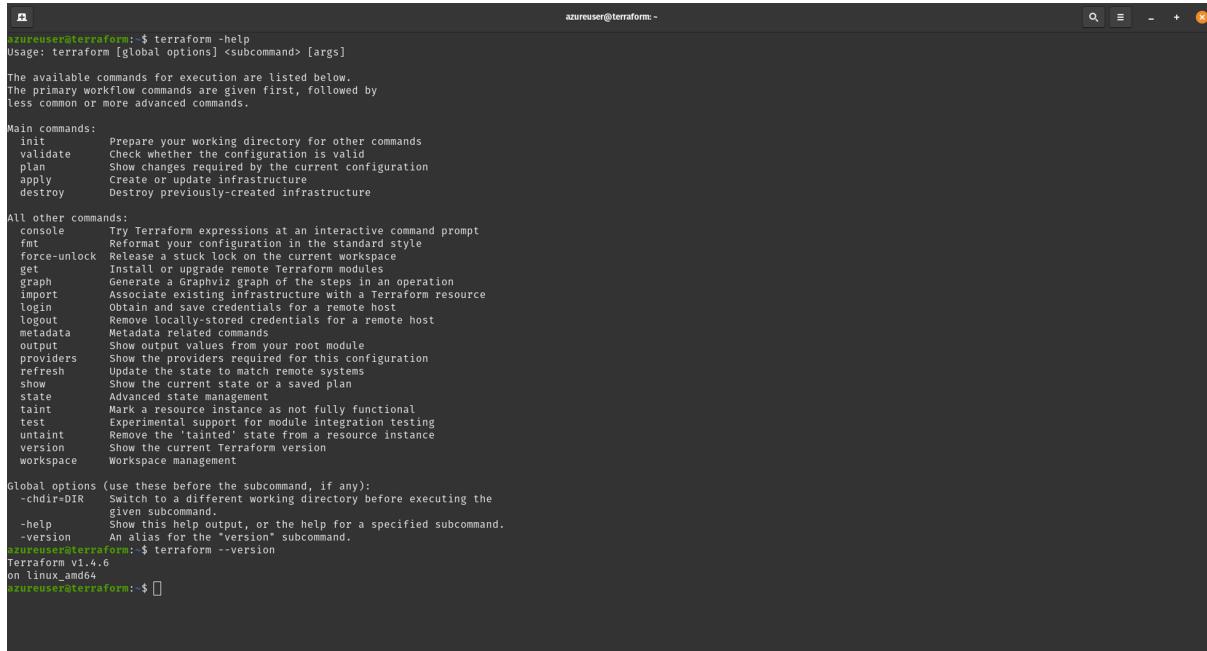
**Speed:** IaC allows teams to quickly provision and configure infrastructure resources, reducing the time needed to deploy and test applications.

**Cost-efficiency:** By automating infrastructure provisioning and configuration, teams can reduce manual labor and associated costs, while ensuring optimal resource utilization.

Overall, IaC provides a more efficient, reliable, and cost-effective way to manage infrastructure, helping teams to deliver applications faster and with greater consistency.

### **1.3.2 Task 2 – Terraform Installation on Linux**

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o
→ /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]
→ https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee
→ /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```



```
azureuser@terraform: ~
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init      Prepare your working directory for other commands
  validate   Check whether the configuration is valid
  plan      Show changes required by the current configuration
  apply     Create or update infrastructure
  destroy    Destroy previously-created infrastructure

All other commands:
  console   Try Terraform expressions at an interactive command prompt
  fmt       Reformat your configuration in the standard style
  force-unlock Release a stuck lock on the current workspace
  get       Install or upgrade remote Terraform modules
  graph     Generate a Graphviz graph of the steps in an operation
  import    Associate existing infrastructure with a Terraform resource
  logout    Logout from saved credentials for a remote host
  logout    Remove locally-stored credentials for a remote host
  metadata   Metadata related commands
  output    Show output values from your root module
  providers Show the providers required for this configuration
  refresh   Update the state to match remote systems
  show      Show the current state or a saved plan
  state     Advanced state management
  taint     Mark a resource instance as not fully functional
  test      Experimental support for module integration testing
  untaint   Remove the "tainted" state from a resource instance
  version   Show the current Terraform version
  workspace Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR  Switch to a different working directory before executing the
             given subcommand.
  -help       Show this help output, or the help for a specified subcommand.
  -version    An alias for the "version" subcommand.
azureuser@terraform: ~
Terraform v1.4.6
on linux_amd64
azureuser@terraform: ~
```

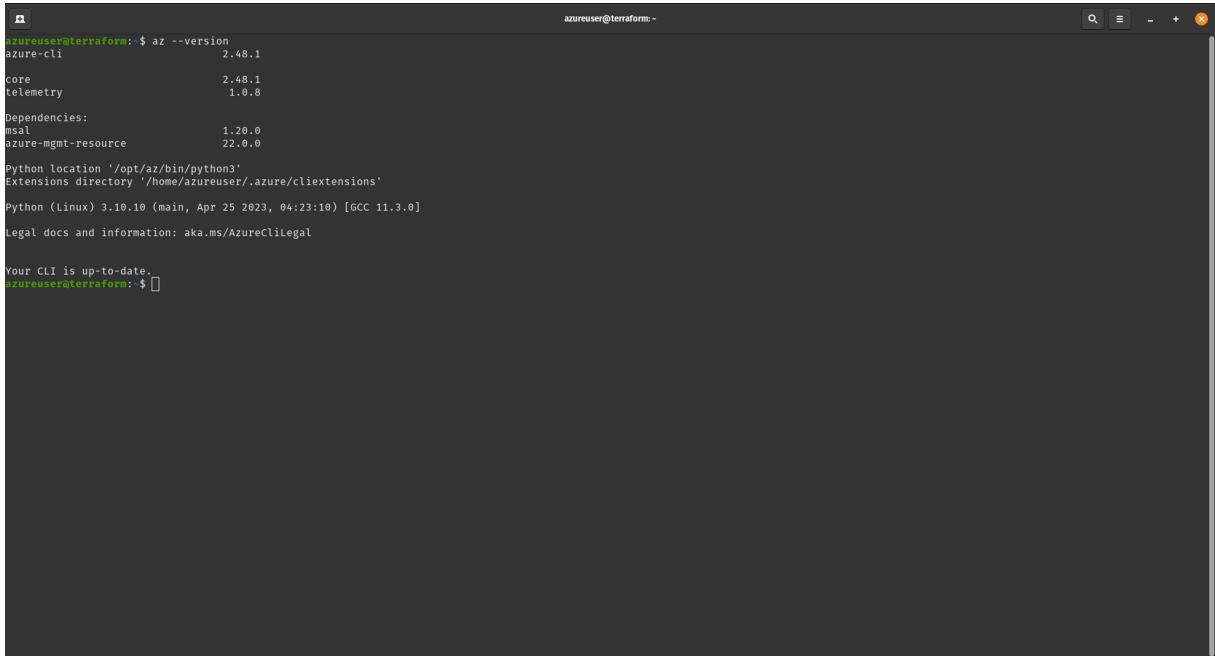
### 1.3.3 Task 3 – Azure CLI Installation on Linux

```
# Download dependencies:
sudo apt-get update
sudo apt-get install ca-certificates curl apt-transport-https lsb-release gnupg

# Download and install the Microsoft signing key:
sudo mkdir -p /etc/apt/keyrings
curl -sLS https://packages.microsoft.com/keys/microsoft.asc | 
  gpg --dearmor | 
  sudo tee /etc/apt/keyrings/microsoft.gpg > /dev/null
sudo chmod go+r /etc/apt/keyrings/microsoft.gpg

# Add the Azure CLI software repository:
AZ_REPO=$(lsb_release -cs)
echo "deb [arch=`dpkg --print-architecture` signed-by=/etc/apt/keyrings/microsoft.gpg]
  https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO main" |
  sudo tee /etc/apt/sources.list.d/azure-cli.list

# Update repository information and install the azure-cli package:
sudo apt-get update
sudo apt-get install azure-cli
```



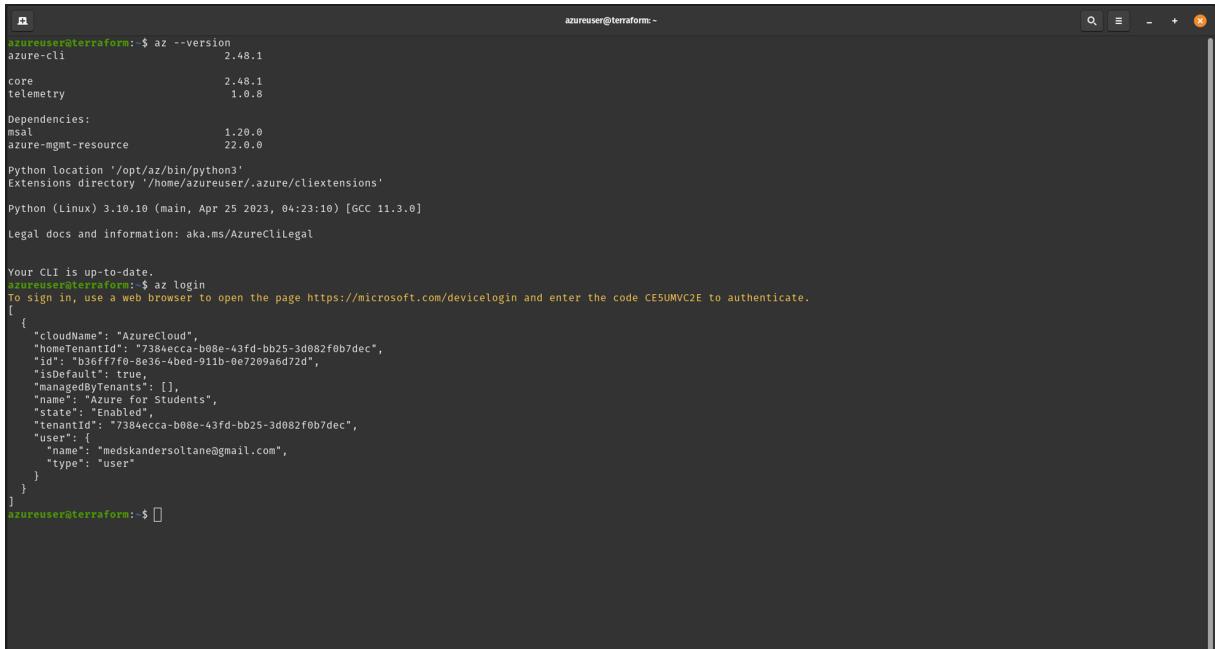
```
azureuser@terraform: $ az --version
azure-cli          2.48.1
core              2.48.1
telemetry         1.0.8
Dependencies:
msal              1.20.0
azure-mgmt-resource 22.0.0
Python location '/opt/az/bin/python3'
Extensions directory '/home/azureuser/.azure/cliextensions'

Python (Linux) 3.10.10 (main, Apr 25 2023, 04:23:10) [GCC 11.3.0]
Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.
azureuser@terraform: $ 
```

- Create a service principal

In Terraform, a service principal is used to authenticate the Azure provider to access and manage Azure resources. The service principal provides the necessary credentials to Terraform, allowing it to make changes to Azure resources on behalf of the user.



```
azureuser@terraform: $ az --version
azure-cli          2.48.1
core              2.48.1
telemetry         1.0.8
Dependencies:
msal              1.20.0
azure-mgmt-resource 22.0.0
Python location '/opt/az/bin/python3'
Extensions directory '/home/azureuser/.azure/cliextensions'

Python (Linux) 3.10.10 (main, Apr 25 2023, 04:23:10) [GCC 11.3.0]
Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.
azureuser@terraform: $ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CESUMVC2E to authenticate.
{
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "7384ecca-b08e-43fd-bb25-3d082f0b7dec",
    "id": "b3cff7f0-8e36-4bed-911b-0e7209a6d72d",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "7384ecca-b08e-43fd-bb25-3d082f0b7dec",
    "user": {
      "name": "medskandersoltane@gmail.com",
      "type": "user"
    }
  }
}
azureuser@terraform: $ 
```



```

azuser@terraforms:~ Dependencies:
msal           1.20.0
azure-mgmt-resource 22.0.0

Python location '/opt/az/bin/python3'
Extensions directory '/home/azuser/.azure/cliextensions'

Python (Linux) 3.10.10 (main, Apr 25 2023, 04:23:10) [GCC 11.3.0]

Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.

azuser@terraforms:~ $ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin and enter the code CE5UMVC2E to authenticate.

[{"cloudName": "AzureCloud",
"homeTenantId": "7384ecca-b08e-43fd-bb25-3d082f0b7dec",
"id": "b36ff7f0-8e36-4bed-911b-0e7209a6d72d",
"isDefault": true,
"managedByTenants": [],
"name": "Azure for Students",
"state": "Enabled",
"tenantId": "738",
"user": {
"email": "azuser@gmail.com",
"type": "user"
}
}
]

azuser@terraforms:~ $ az account set --subscription "b36ff7f0-8e36-4bed-911b-0e7209a6d72d"
azuser@terraforms:~ $ az ad sp create-for-rbac --role="Contributor" --scopes="/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d"
Creating a service principal assignment under scope /subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control. For more information, see https://aka.ms/azdsp-cli
{
"appId": "72ec",
"displayName": "azure-cli-2023-04-27-18-10-32",
"password": "2wM8L",
"tenant": "7384"
}

azuser@terraforms:~ $ export ARM_CLIENT_ID="72ec"
azuser@terraforms:~ $ export ARM_CLIENT_SECRET="2wM8L"
azuser@terraforms:~ $ export ARM_SUBSCRIPTION_ID="b36ff7f0-8e36-4bed-911b-0e7209a6d72d"
azuser@terraforms:~ $ export ARM_TENANT_ID="7384ec"
azuser@terraforms:~ $ 

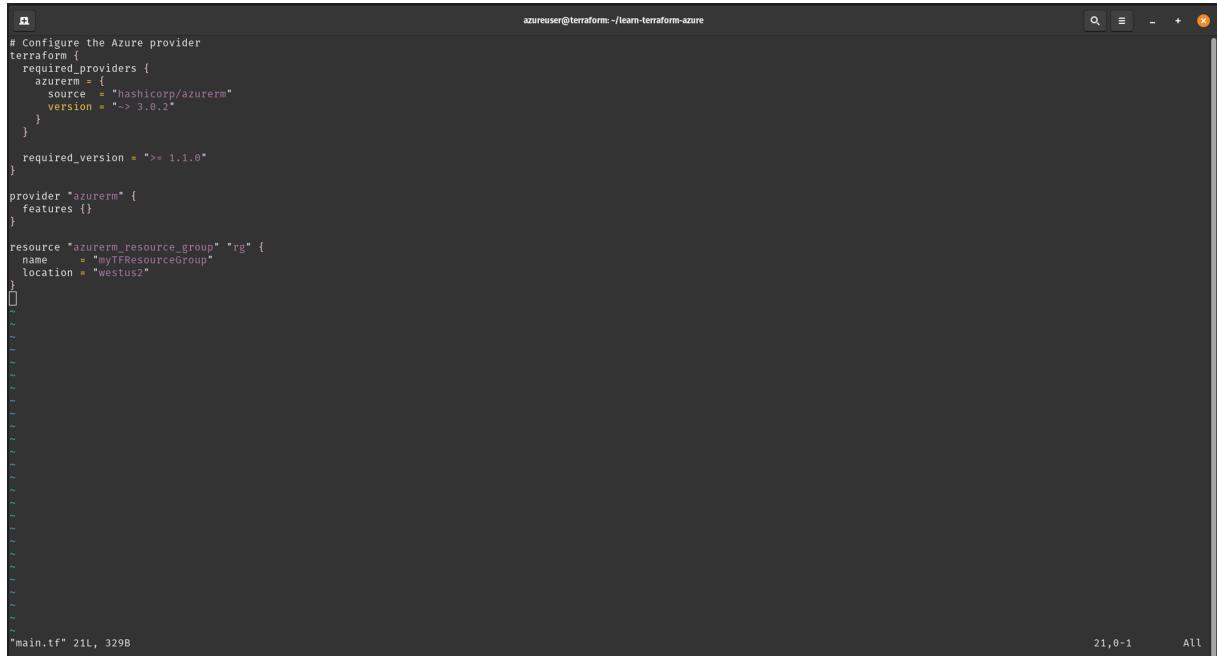
```

### What we've done: A. Creating a Service Principal:

- Log in to Azure CLI
- Set active subscription
- Run command “az ad sp create-for-rbac” to create a new service principal
- Save the output values, including the app ID and secret key, for later use

### B. Setting up Environment Variables in Terraform:

- Create a new file named “.env”
- Add the following environment variables: ARM\_CLIENT\_ID, ARM\_CLIENT\_SECRET, ARM\_SUBSCRIPTION\_ID, ARM\_TENANT\_ID
- Save the file with the actual values obtained from creating the service principal in Azure
- Run Terraform commands, which will automatically load the variables from the “.env” file.
- Write Configuration

A screenshot of a terminal window titled "azuser@terraform: ~/learn-terraform-azure". The window displays a portion of a Terraform configuration file named "main.tf". The code includes provider declarations for "azurerm" and "azurerm\_resource\_group", and a resource block for creating a resource group named "myTFResourceGroup" in the "westus2" location. The terminal shows the file has 211 lines and 3298 bytes.

```
# Configure the Azure provider
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0.2"
    }
  }
  required_version = ">= 1.1.0"
}

provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "rg" {
  name     = "myTFResourceGroup"
  location = "westus2"
}
```

```
# Configure the Azure provider
terraform {
  # Declare the required providers for this configuration
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"  # The source of the provider (in this case,
      version = "~> 3.0.2"          # The version constraint for the provider (a
    }
  }

  # Specify the minimum required version of Terraform
  required_version = ">= 1.1.0"
}

# Configure the Azure provider and its features
provider "azurerm" {
  features {}  # No features enabled in this example
}

# Define an Azure resource group
resource "azurerm_resource_group" "rg" {
  name     = "myTFResourceGroup"  # The name of the resource group to create
```

```
location = "westus2"          # The Azure region in which to create the resource group
}
```

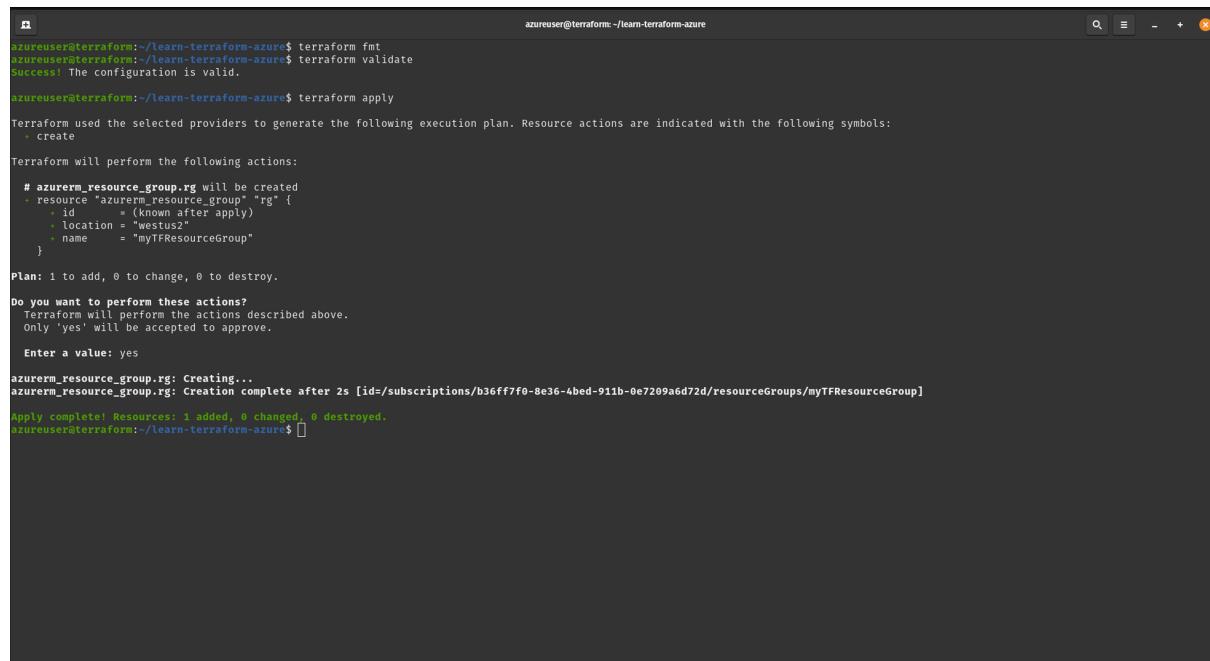
- Terraform init, format, validate, and apply

Terraform init initializes the working directory, downloading and installing the Azure provider and any other providers declared in the configuration.

Terraform format automatically updates configurations in the current directory for readability and consistency.

Terraform validate checks the configuration for errors, ensuring that it is valid and internally consistent.

Terraform apply creates or updates the resources defined in the configuration, in this case the resource group.



```
azureuser@terraform:~/learn-terraform-azure$ terraform fmt
azureuser@terraform:~/learn-terraform-azure$ terraform validate
Success! The configuration is valid.

azureuser@terraform:~/learn-terraform-azure$ terraform apply
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + Create
  - Destroy
  ~ Update as part of another resource

Terraform will perform the following actions:

  # azurerm_resource_group.rg will be created
  + resource "azurerm_resource_group" "rg" {
      + id      = (known after apply)
      + location = "westus2"
      + name    = "myTFResourceGroup"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Creation complete after 2s [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
azureuser@terraform:~/learn-terraform-azure$
```

- Inspect your state

Terraform show displays the current state of the infrastructure managed by Terraform.

Terraform state list lists all the resources managed by Terraform.

```
azureuser@terraform:~/learn-terraform-azure$ terraform show
# azurerm_resource_group:
resource "azurerm_resource_group" "rg" {
  id       = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup"
  location = "westus2"
  name     = "myTFResourceGroup"
}
azurermuser@terraform:~/learn-terraform-azure$ terraform state list
azurerm_resource_group.rg
azurermuser@terraform:~/learn-terraform-azure$
```

### 1.3.4 Task 4 - Change the infrastructure

- Create a new resource

```
# Configure the Azure provider
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0.2"
    }
  }
  required_version = ">= 1.1.0"
}

provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "rg" {
  name     = "myTFResourceGroup"
  location = "westus2"
}

# Create a virtual network
resource "azurerm_virtual_network" "vnet" {
  name            = "myTFVnet"
  address_space   = ["10.0.0.0/16"]
  location        = "westus2"
  resource_group_name = azurerm_resource_group.rg.name
}
```

```

azuser@terraform:~/learn-terraform-azure$ vim main.tf
azuser@terraform:~/learn-terraform-azure$ terraform apply
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + Create

Terraform will perform the following actions:

  # azurerm_virtual_network.vnet will be created
+ resource "azurerm_virtual_network" "vnet" {
    + address_space     = [
        + "10.0.0.0/16",
    ]
    + dns_servers      = (known after apply)
    + id               = (known after apply)
    + location         = "westus2"
    + name             = "myTFVnet"
    + resource_group_name = "myTFResourceGroup"
    + subnet           = (known after apply)
}

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_virtual_network.vnet: Creating...
azurerm_virtual_network.vnet: Still creating... [10s elapsed]
azurerm_virtual_network.vnet: Creation complete after 12s [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
azuser@terraform:~/learn-terraform-azure$ 

```

A new resource was added!

- Modify an existing resource

```

# Configure the Azure provider
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "*~> 3.0.2"
    }
  }
  required_version = ">= 1.1.0"
}

provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "rg" {
  name       = "myTFResourceGroup"
  location   = "westus2"
  tags       = [
    Environment = "Terraform Getting Started"
    Team        = "DevOps"
  ]
}

resource "azurerm_virtual_network" "vnet" {
  name           = "myTFVnet"
  address_space  = ["10.0.0.0/16"]
  location       = "westus2"
  resource_group_name = azurerm_resource_group.rg.name
}

```

```

azuser@terraform:~/learn-terraform-azure$ vim main.tf
azuser@terraform:~/learn-terraform-azure$ terraform apply
azurerm_resource_group: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]
azurerm_virtual_network.vnet: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

# azurerm_resource_group.rg will be updated in-place
~ resource "azurerm_resource_group" "rg" {
  id          = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup"
  name        = "myTFResourceGroup"
  ~ tags      = {
    "Environment" = "Terraform Getting Started"
    "Team"        = "DevOps"
  }
  # (1 unchanged attribute hidden)
}

Plan: 0 to add, 1 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_resource_group.rg: Modifying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]
azurerm_resource_group.rg: Modifications complete after 3s [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]

apply complete! Resources: 0 added, 1 changed, 0 destroyed.
azuser@terraform:~/learn-terraform-azure$ 

```

The resource was changed!

- Review updates to state

```

azuser@terraform:~/learn-terraform-azure$ terraform show
# azurerm_resource_group.rg:
resource "azurerm_resource_group" "rg" {
  id          = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup"
  location    = "westus2"
  name        = "myTFResourceGroup"
  ~ tags      = {
    "Environment" = "Terraform Getting Started"
    "Team"        = "DevOps"
  }
}

# azurerm_virtual_network.vnet:
resource "azurerm_virtual_network" "vnet" {
  address_space     = [
    "10.0.0.0/16",
  ]
  dns_servers       = []
  flow_timeout_in_minutes = 0
  guid              = "af78e8ca-92cc-49fc-b266-a7ad5223d303"
  id                = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet"
  location          = "westus2"
  name              = "myTFVnet"
  resource_group_name = "myTFResourceGroup"
  subnet            = []
  ~ tags      = {}
}

azuser@terraform:~/learn-terraform-azure$ terraform state list
azurerm_resource_group.rg
azurerm_virtual_network.vnet
azuser@terraform:~/learn-terraform-azure$ 

```

### 1.3.5 Task 5 - Destroy the infrastructure

```

azureuser@terraforms:~/learn-terraform-azure$ terraform destroy
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]
azurerm_virtual_network.vnet: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- : destroy

Terraform will perform the following actions:

# azurerm_resource_group.rg will be destroyed
resource "azurerm_resource_group" "rg" {
  id           = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup"
  location     = "westus2"
  name         = "myTFResourceGroup"
  tags         = [
    {
      "Environment" = "Terraform Getting Started"
      "Team"        = "DevOps"
    }
  ]
}

# azurerm_virtual_network.vnet will be destroyed
resource "azurerm_virtual_network" "vnet" {
  address_space     = [
    "10.0.0.0/16"
  ]
  dns_servers       = []
  flow_timeout_in_minutes = 0
  guid              = "af78e8ca-92cc-49fc-b266-a7ad5223d303"
  id                = "/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet"
  location          = "westus2"
  name              = "myTFVnet"
  resource_group_name = "myTFResourceGroup"
  subnet            = []
  tags              = []
}

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

azurerm_virtual_network.vnet: Destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]
azurerm_virtual_network.vnet: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet, 10s elapsed]
azurerm_virtual_network.vnet: Destruction complete after 12s

azurerm_resource_group.rg: Destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 20s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 30s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 40s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 50s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 1m0s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFResourceGroup, 1m10s elapsed]
azurerm_resource_group.rg: Destruction complete after 1m0s

destroy complete! Resources: 2 destroyed.
azureuser@terraforms:~/learn-terraform-azure$ 

```

Terraform destroy destroys all the resources defined in the configuration.

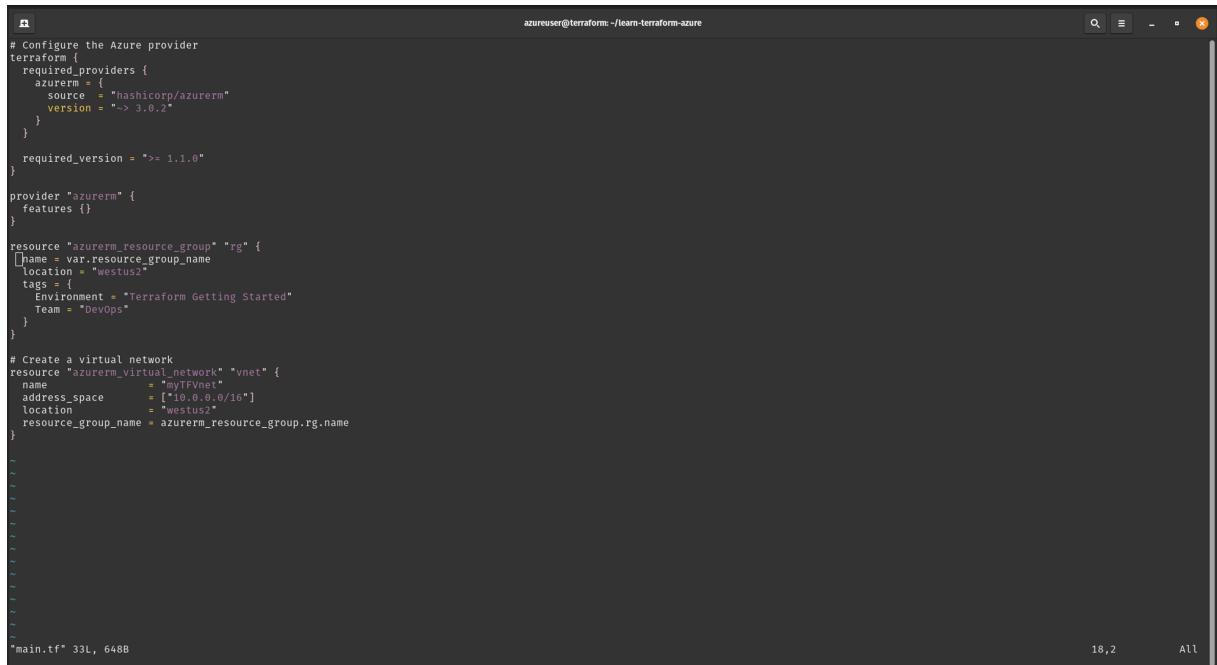
### 1.3.6 Task 6 - Terraform Input Variables

- Create a variable



```
variable "resource_group_name" {
  default = "myTFResourcegroup"
}
```

- Use a variable in main.tf



```
# Configure the Azure provider
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0.2"
    }
  }
  required_version = ">= 1.1.0"
}

provider "azurerm" {
  features {}
}

resource "azurerm_resource_group" "rg" {
  name        = var.resource_group_name
  location    = "westus2"
  tags        = {
    Environment = "Terraform Getting Started"
    Team       = "Devops"
  }
}

# Create a virtual network
resource "azurerm_virtual_network" "vnet" {
  name                = "myTFvnet"
  address_space       = "10.0.0.0/16"
  location            = "westus2"
  resource_group_name = azurerm_resource_group.rg.name
}

main.tf 33L, 648B
```

- Terraform apply

- Forcing a variable

```
azureresource@terrafrom-azure:~/learn-terraform-azure$ terraform apply -var "resource_group_name=myNewResourceGroup"
azurerm_resource_group.rg: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup]
azurerm_virtual_network.vnet: Refreshing state... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  # destroy and then create replacement

Terraform will perform the following actions:

# azurerm_resource_group.rg must be replaced
-/ resource "azurerm_resource_group" "rg" {
  + id          = "/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myNewResourceGroup" > (known after apply)
  - name        = "myTFRResourceGroup" > "myNewResourceGroupName" # forces replacement
  - tags       = [
      + "environment" = "Terraform Getting Started"
      + "team"        = "DevOps"
    ]
  # (1 unchanged attribute hidden)
}

# azurerm_virtual_network.vnet must be replaced
-/ resource "azurerm_virtual_network" "vnet" {
  + id          = "/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet" > (known after apply)
  - address_space {
      + cidr_blocks = [
          + "10.0.0.0/16"
        ]
    }
  - flow_logs {
      + enabled = true
      + log_analytics_workspace_id = "/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Insights/workspaces/TFLog"
      + log_retention_in_minutes = 0
    }
  - guid          = "b322f2f05-ce8a-4e8a-bcd4-e8e591dc920a" > (known after apply)
  - id            = "/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet" > (known after apply)
  - name          = "myTFVnet"
  - resource_group_name = "myTFRResourceGroup" > "myNewResourceGroupName" # forces replacement
  - subnets {
      + id          = "/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet/subnets/1" > (known after apply)
      + name        = "Subnet1"
      + tags       = [
          + "key": "value"
        ]
    }
  - tags       = [
      + "key": "value"
    ]
  # (2 unchanged attributes hidden)
}

Terraform will destroy approximately 2 objects.

Do you want to perform these actions?
  Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_virtual_network.vnet: Destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]
azurerm_virtual_network.vnet: Still destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet, 10s elapsed]
azurerm_virtual_network.vnet: Destruction complete after 13s

azurerm_resource_group.rg: Destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 10s elapsed]
azurerm_resource_group.rg: Destroyed [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 20s elapsed]
azurerm_resource_group.rg: Destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 40s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 49s elapsed]
azurerm_resource_group.rg: Destroyed [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 59s elapsed]
azurerm_resource_group.rg: Destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 1m10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 1m20s elapsed]
azurerm_resource_group.rg: Destroyed [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myTFRResourceGroup, 1m20s elapsed]
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Still creating... [0s elapsed]
azurerm_resource_group.rg: Creation complete after 1s [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myNewResourceGroup]
azurerm_virtual_network.vnet: Still creating... [0s elapsed]
azurerm_virtual_network.vnet: Creation complete after 11s [id=/subscriptions/b36ff7f0-8e36-abed-911b-0e7209ad72d/resourceGroups/myNewResourceGroup/providers/Microsoft.Network/virtualNetworks/myTFVnet]

apply complete! Resources: 2 added, 0 changed, 2 destroyed.
azureresource@terrafrom-azure:~$
```

### **1.3.7 Task 7 - Terraform Output Variables**

- Create an output variable

```
azurerm_resource_group.rg {
  name = "myNewResourceGroupName"
}

output "resource_group_id" {
  value = azurerm_resource_group.rg.id
}
```

outputs.tf 4L, 71B

- Getting the output value

ID of the resource group is now available as an output variable.

```
# azurerm_virtual_network.vnet must be replaced
~/resource "azurerm_virtual_network" "vnet" {
  ~ dns_servers           = [] -> (known after apply)
  ~ flow_timeout_in_minutes = 0 -> null
  ~ guid                  = "441d4ff23-9b7f-431c-a713-92699210609d" -> (known after apply)
  ~ id                    = "/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myNewResourceGroupName/providers/Microsoft.Network/virtualNetworks/myTFVnet" -> (known after apply)
  ~ name                  = "myTFVnet"
  ~ resource_group_name   = "myNewResourceGroupName" -> "myTFRG" # forces replacement
  ~ subnet                = [] -> (known after apply)
  ~ tags                  = {} -> null
  # (2 unchanged attributes hidden)
}

Plan: 2 to add, 0 to change, 2 to destroy.

Changes to Outputs:
  resource_group_id = (known after apply)

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_virtual_network.vnet: Destroying... [id=/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myNewResourceGroupName/providers/Microsoft.Network/virtualNetworks/myTFVnet]
azurerm_virtual_network.vnet: Still destroying... [id=/subscriptions/b36fff7f0-8e36-4bed-911b-...osoft.Network/virtualNetworks/myTFVnet, 10s elapsed]
azurerm_virtual_network.vnet: Destruction complete after 13s
azurerm_resource_group.rg: Destroying... [id=/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myNewResourceGroupName]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36fff7f0-8e36-4bed-911b-.../resourceGroups/myNewResourceGroupName, 10s elapsed]
azurerm_resource_group.rg: Still destroying... [id=/subscriptions/b36fff7f0-8e36-4bed-911b-.../resourceGroups/myNewResourceGroupName, 20s elapsed]
azurerm_resource_group.rg: Destruction complete after 21s
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Creation complete after 1s [id=/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFRG]
azurerm_virtual_network.vnet: Creating...
azurerm_virtual_network.vnet: Still creating... [10s elapsed]
azurerm_virtual_network.vnet: Creation complete after 10s [id=/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFRG/providers/Microsoft.Network/virtualNetworks/myTFVnet]

Apply complete! Resources: 2 added, 0 changed, 2 destroyed.

Outputs:

resource_group_id = "/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFRG"
azureuser@terraform: ~/learn-terraform-azure$ terraform output resource_group_id
"/subscriptions/b36fff7f0-8e36-4bed-911b-0e7209a6d72d/resourceGroups/myTFRG"
azureuser@terraform: ~/learn-terraform-azure$ 
```