Installing Terraform by Script on MACOS:

Using Homebrew, for installing Terraform by executing the following common.
Terminal Command: $ brew install terraform

To install Terraform:

Terminal Command:
```
$ terraform
$ terraform -version
```

To install AZ CLI:

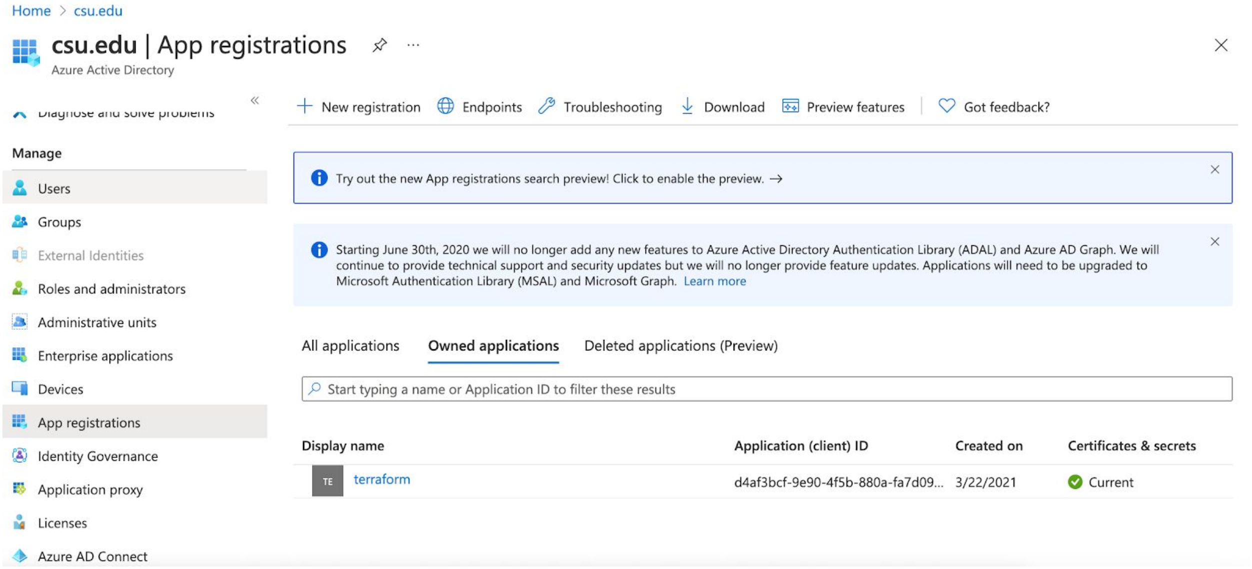
Terminal Command:
```
$ brew install azure-cli
$ az login   (To sign in azure)
$ az  (to view the installation)
```

Configuring terraform for Azure

Creating the Azure Service Principle: The following is a template az cli script that you have to run the creating an Service principal, where you have to enter your SP name, role, and scope.

Terminal Command:
```
$ az ad sp create-for-rbac --name=”terraform” --role=”contributor”
--scope=”/subscriptions/8921-1444-.....”
                    (Subscription Id)
```

The following screenshot shows the SP is created in Azure AD:

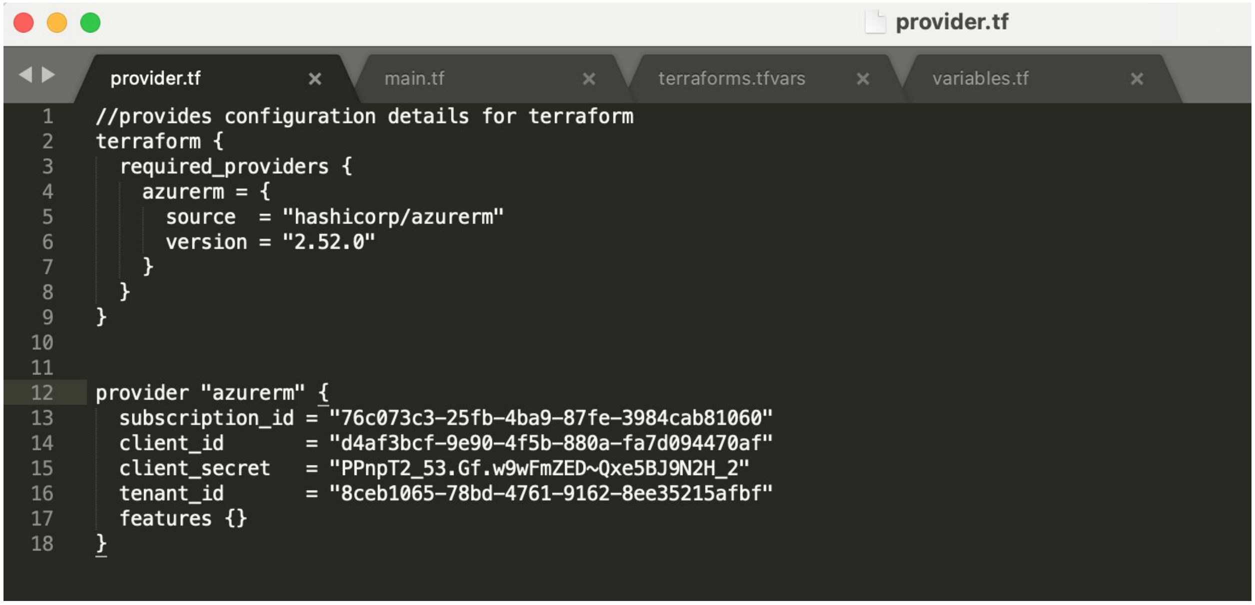


Try out the new App registrations search preview! Click to enable the preview. →

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

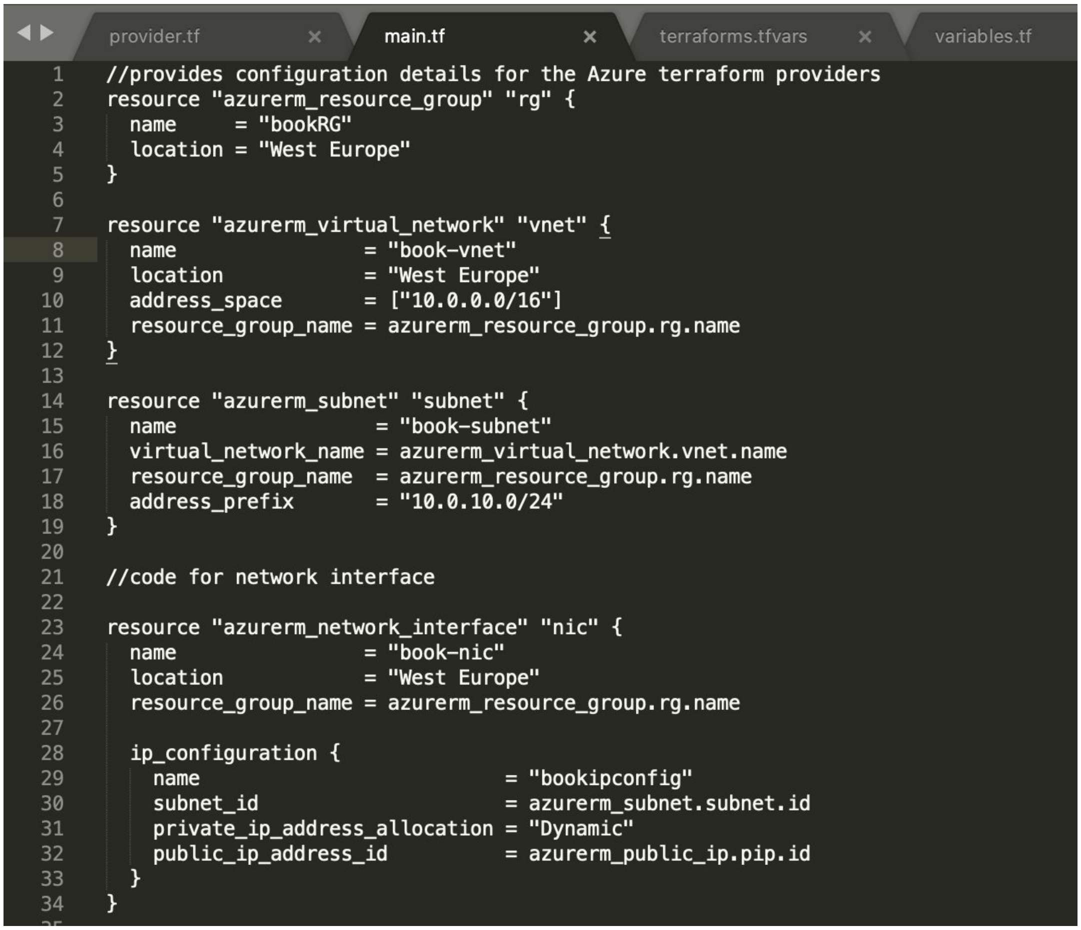
Display name	Application (client) ID	Created on	Certificates & secrets
TE terraform	d4af3bcf-9e90-4f5b-880a-fa7d09...	3/22/2021	✓ Current

Configuring the Terraform provider:

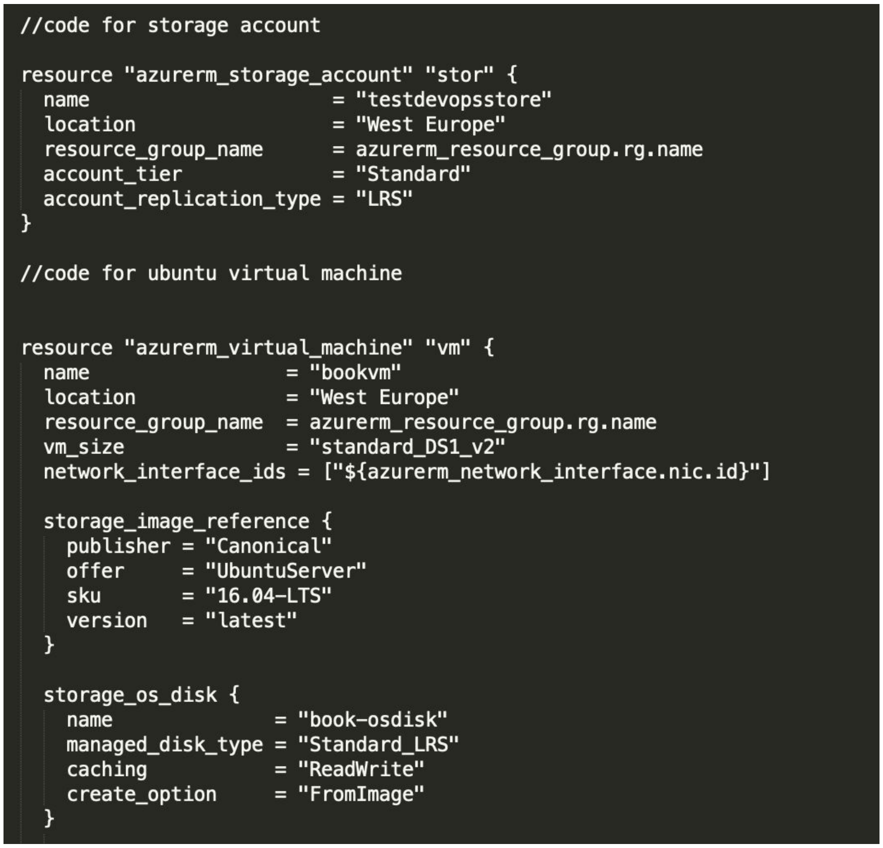


```
//provides configuration details for terraform
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "2.52.0"
    }
  }
}


provider "azurerm" {
  subscription_id = "76c073c3-25fb-4ba9-87fe-3984cab81060"
  client_id       = "d4af3bcf-9e90-4f5b-880a-fa7d094470af"
  client_secret   = "PPnpT2_53.Gf.w9wFmZED~Qxe5BJ9N2H_2"
  tenant_id       = "8ceb1065-78bd-4761-9162-8ee35215afbf"
  features {}
}
```

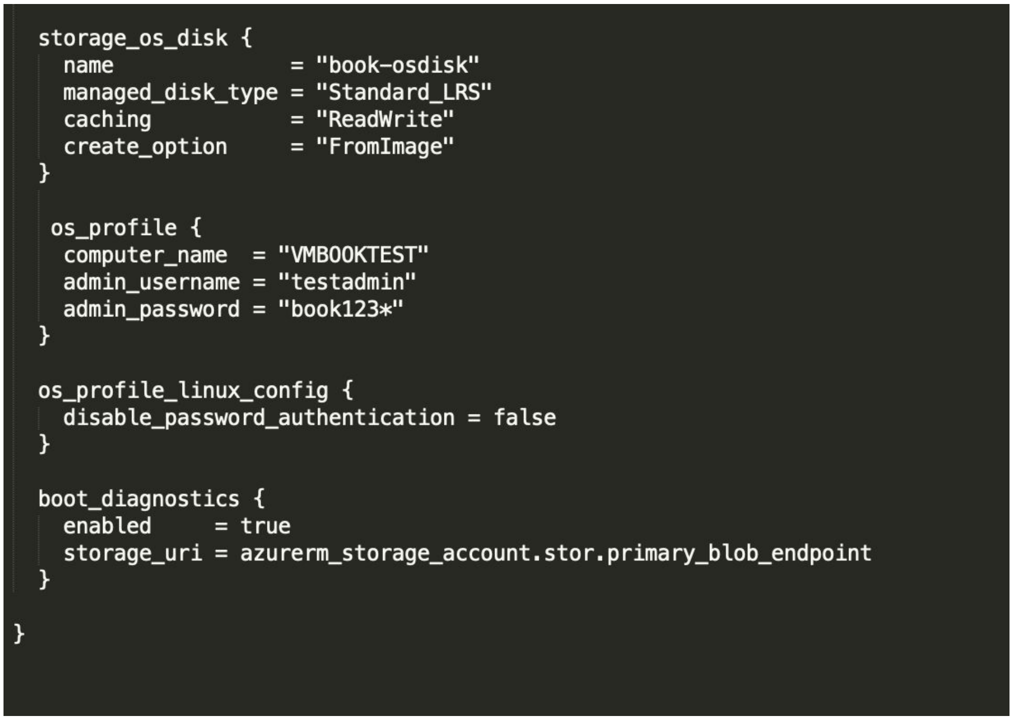
Writing a Terraform Script to deploy Azure infrastructure:



```
//provides configuration details for the Azure terraform providers
resource "azurerm_resource_group" "rg" {
  name     = "bookRG"
  location = "West Europe"
}

resource "azurerm_virtual_network" "vnet" {
  name                = "book-vnet"
  location            = "West Europe"
  address_space       = ["10.0.0.0/16"]
  resource_group_name = azurerm_resource_group.rg.name
}

resource "azurerm_subnet" "subnet" {
  name                 = "book-subnet"
  virtual_network_name = azurerm_virtual_network.vnet.name
  resource_group_name  = azurerm_resource_group.rg.name
  address_prefix       = "10.0.10.0/24"
}

//code for network interface

resource "azurerm_network_interface" "nic" {
  name                = "book-nic"
  location            = "West Europe"
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                          = "bookipconfig"
    subnet_id                     = azurerm_subnet.subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          = azurerm_public_ip.pip.id
  }
}
```

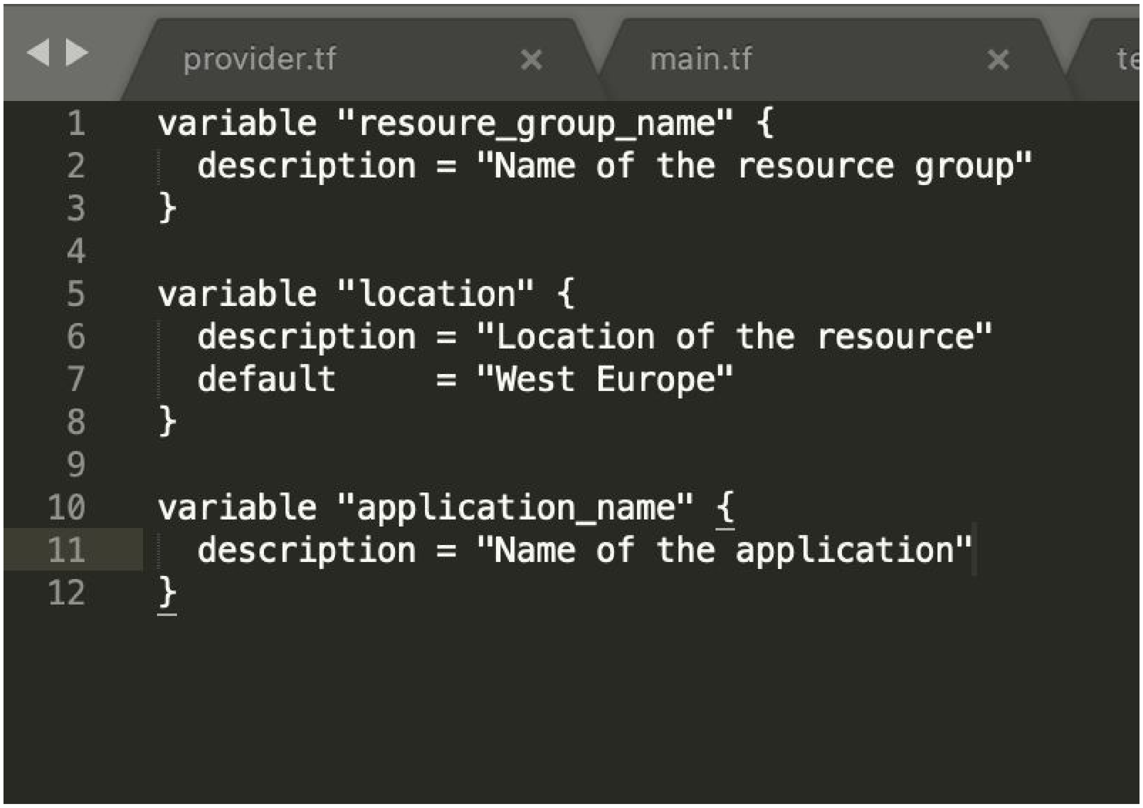


```
//code for storage account

resource "azurerm_storage_account" "stor" {
  name                     = "testdevopsstore"
  location                 = "West Europe"
  resource_group_name      = azurerm_resource_group.rg.name
  account_tier             = "Standard"
  account_replication_type = "LRS"
}

//code for ubuntu virtual machine


resource "azurerm_virtual_machine" "vm" {
  name                  = "bookvm"
  location              = "West Europe"
  resource_group_name   = azurerm_resource_group.rg.name
  vm_size               = "standard_DS1_v2"
  network_interface_ids = ["${azurerm_network_interface.nic.id}"]

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

  storage_os_disk {
    name              = "book-osdisk"
    managed_disk_type = "Standard_LRS"
    caching           = "ReadWrite"
    create_option     = "FromImage"
  }
```



```
  storage_os_disk {
    name              = "book-osdisk"
    managed_disk_type = "Standard_LRS"
    caching           = "ReadWrite"
    create_option     = "FromImage"
  }

   os_profile {
    computer_name  = "VMBOOKTEST"
    admin_username = "testadmin"
    admin_password = "book123*"
  }

  os_profile_linux_config {
    disable_password_authentication = false
  }

  boot_diagnostics {
    enabled     = true
    storage_uri = azurerm_storage_account.stor.primary_blob_endpoint
  }

}
```

Dynamizing the code with variables and interpolation functions:

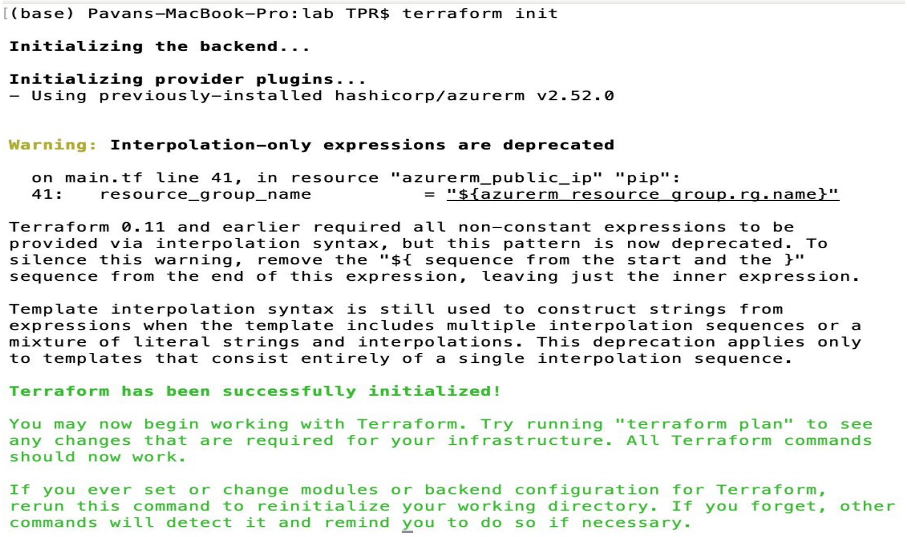


```
variable "resoure_group_name" {
  description = "Name of the resource group"
}

variable "location" {
  description = "Location of the resource"
  default     = "West Europe"
}

variable "application_name" {
  description = "Name of the application"
}
```

Initialization:

To execute the initialization, run the init command.

Terminal Command: $ terraform init



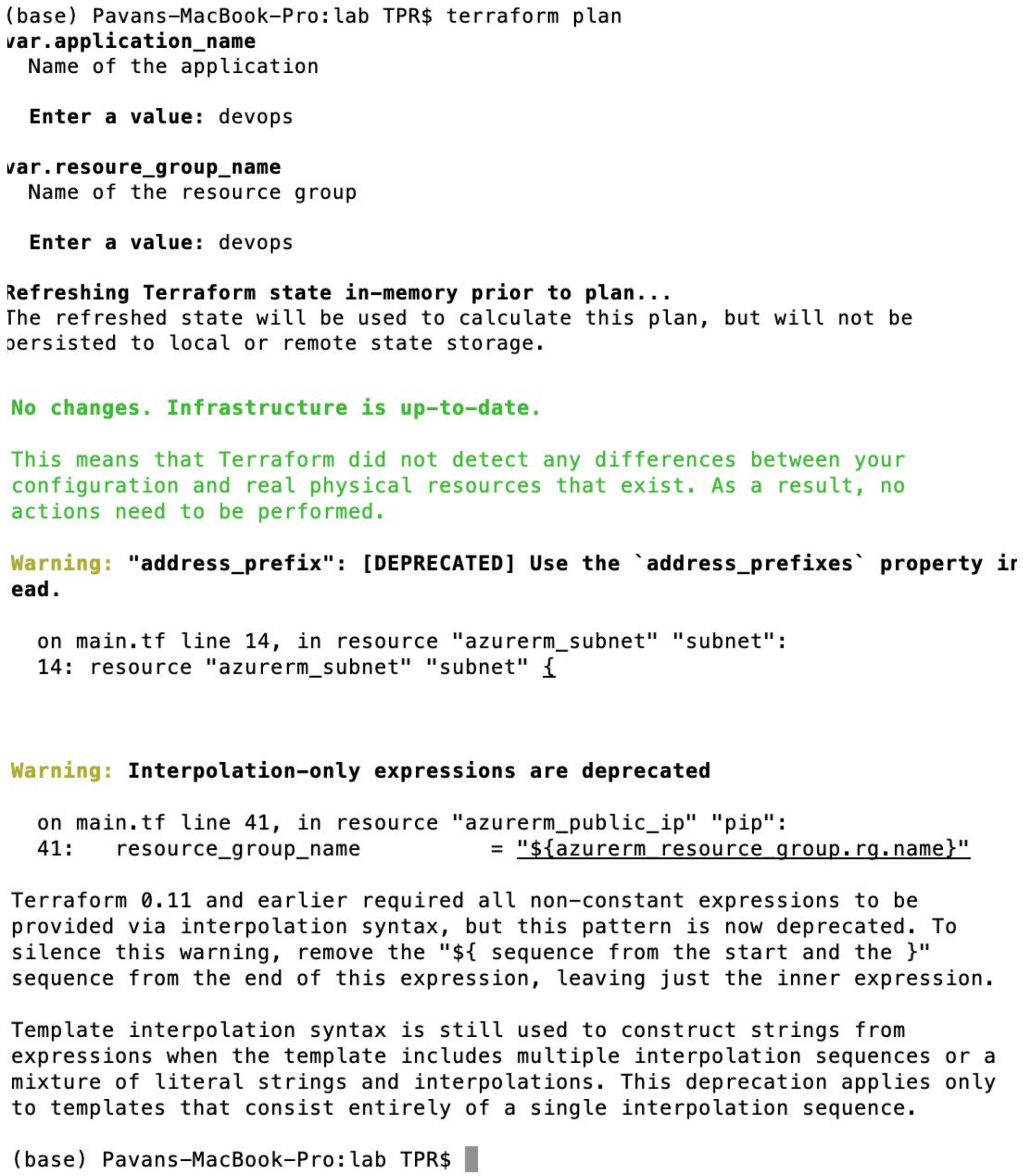
```
(base) Pavans-MacBook-Pro:lab TPR$ terraform init

Initializing the backend...

Initializing provider plugins...
- Using previously-installed hashicorp/azurerm v2.52.0


Warning: Interpolation-only expressions are deprecated

  on main.tf line 41, in resource "azurerm_public_ip" "pip":
  41:   resource_group_name          = "${azurerm_resource_group.rg.name}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Previewing changes:

The next step is the preview of the changes made to the infrastructure before apply them.

For this, run Terraform with the plan command and, when executed, the plan automatically uses the terraform.tfvars file to set the variable.

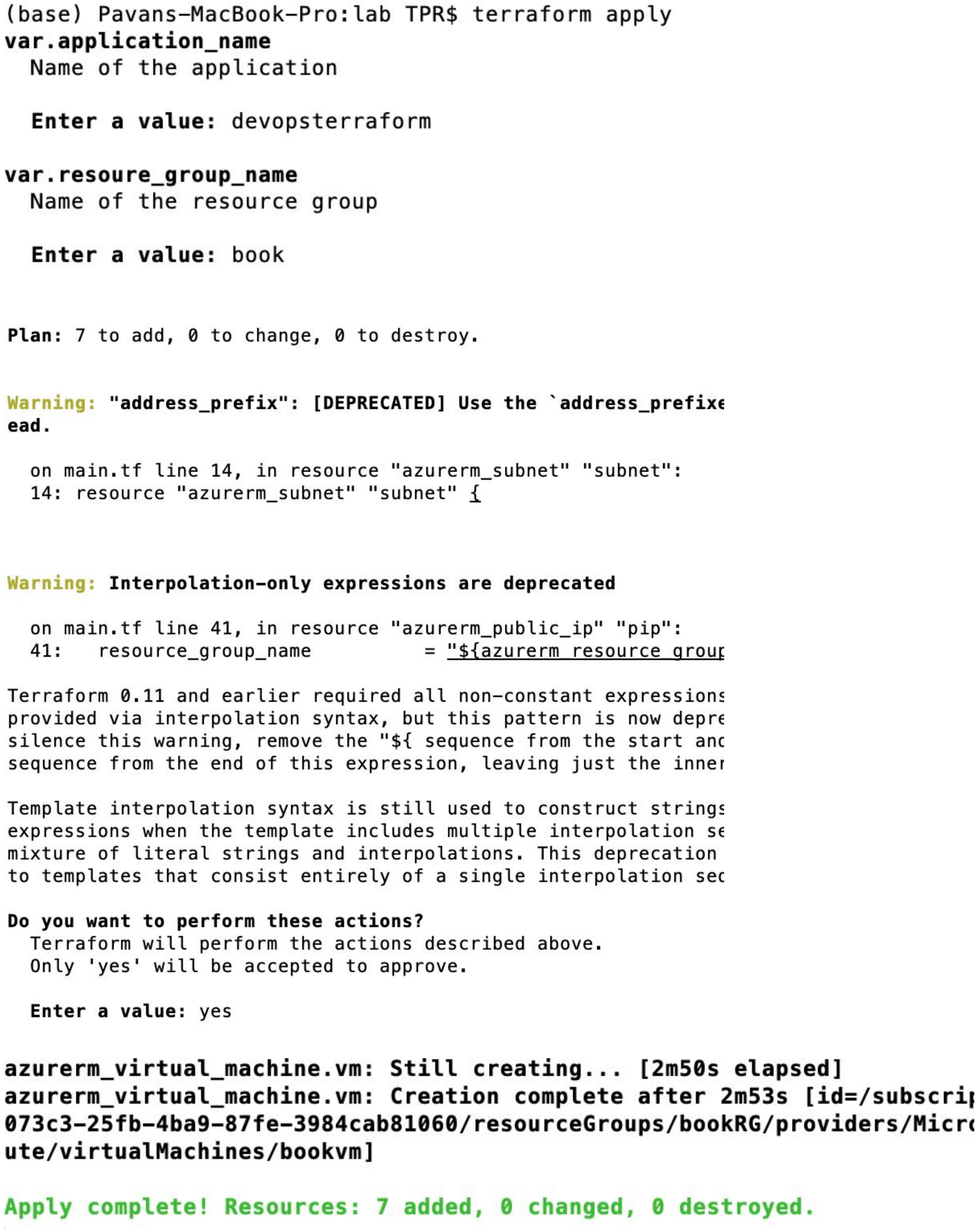
Terminal Command: $ terraform plan



```
(base) Pavans-MacBook-Pro:lab TPR$ terraform plan
var.application_name
  Name of the application

  Enter a value: devops

var.resoure_group_name
  Name of the resource group

  Enter a value: devops

Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.


No changes. Infrastructure is up-to-date.

This means that Terraform did not detect any differences between your
configuration and real physical resources that exist. As a result, no
actions need to be performed.

Warning: "address_prefix": [DEPRECATED] Use the `address_prefixes` property ir
ead.

   on main.tf line 14, in resource "azurerm_subnet" "subnet":
   14: resource "azurerm_subnet" "subnet" {



Warning: Interpolation-only expressions are deprecated

   on main.tf line 41, in resource "azurerm_public_ip" "pip":
   41:   resource_group_name          = "${azurerm_resource_group.rg.name}"

Terraform 0.11 and earlier required all non-constant expressions to be
provided via interpolation syntax, but this pattern is now deprecated. To
silence this warning, remove the "${ sequence from the start and the }"
sequence from the end of this expression, leaving just the inner expression.

Template interpolation syntax is still used to construct strings from
expressions when the template includes multiple interpolation sequences or a
mixture of literal strings and interpolations. This deprecation applies only
to templates that consist entirely of a single interpolation sequence.

(base) Pavans-MacBook-Pro:lab TPR$
```

Applying the changes: To apply to changes to infrastructure.

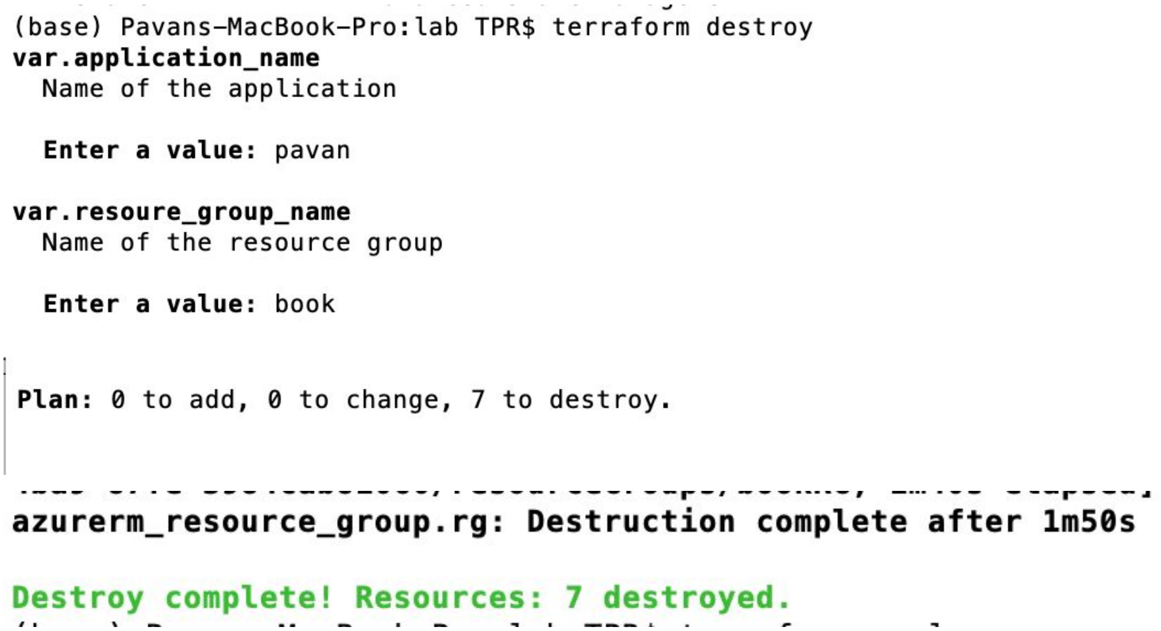
Terminal Command: $ terraform apply



```
(base) Pavans-MacBook-Pro:lab TPR$ terraform apply
var.application_name
  Name of the application

  Enter a value: devopsterraform

var.resoure_group_name
  Name of the resource group

  Enter a value: book


Plan: 7 to add, 0 to change, 0 to destroy.


Warning: "address_prefix": [DEPRECATED] Use the `address_prefixe
ead.

  on main.tf line 14, in resource "azurerm_subnet" "subnet":
  14: resource "azurerm_subnet" "subnet" {



Warning: Interpolation-only expressions are deprecated

  on main.tf line 41, in resource "azurerm_public_ip" "pip":
  41:   resource_group_name          = "${azurerm_resource_grou

Terraform 0.11 and earlier required all non-constant expressions
provided via interpolation syntax, but this pattern is now depre
silence this warning, remove the "${ sequence from the start an
sequence from the end of this expression, leaving just the inne

Template interpolation syntax is still used to construct strings
expressions when the template includes multiple interpolation se
mixture of literal strings and interpolations. This deprecation
to templates that consist entirely of a single interpolation se

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_machine.vm: Still creating... [2m50s elapsed]
azurerm_virtual_machine.vm: Creation complete after 2m53s [id=/subscri
073c3-25fb-4ba9-87fe-3984cab81060/resourceGroups/bookRG/providers/Micr
ute/virtualMachines/bookvm]

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.
```

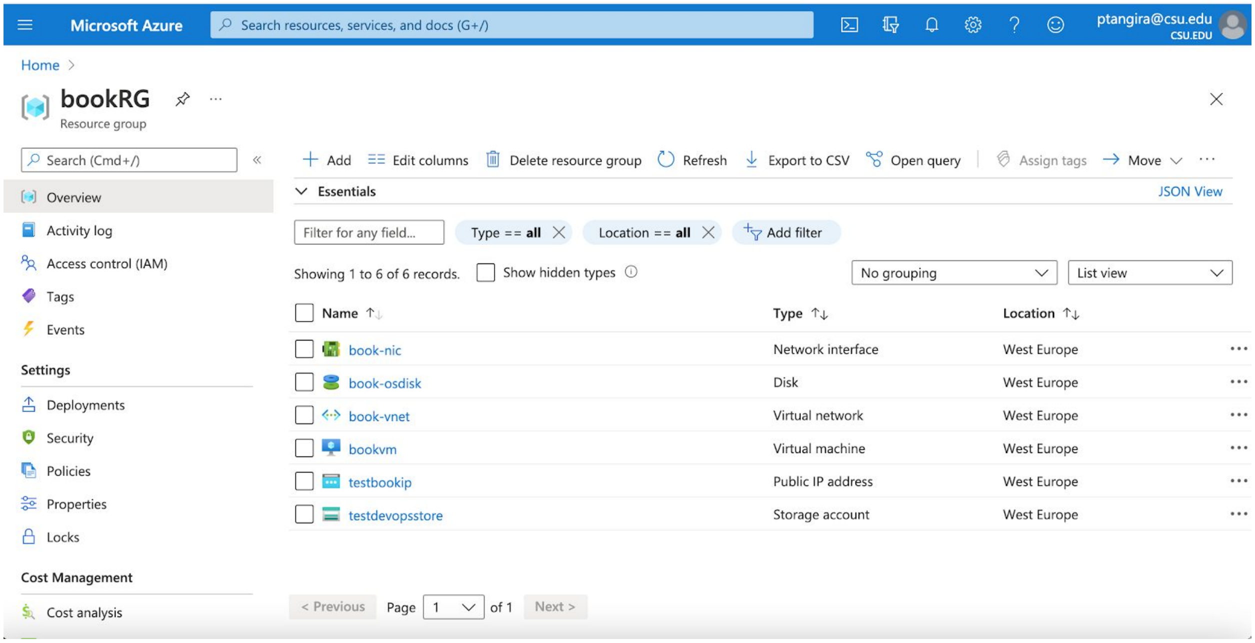
Using Destroy to better rebuild:

<u>Terminal command:</u> $ terraform destroy



```
(base) Pavans-MacBook-Pro:lab TPR$ terraform destroy
var.application_name
  Name of the application

  Enter a value: pavan

var.resoure_group_name
  Name of the resource group

  Enter a value: book


Plan: 0 to add, 0 to change, 7 to destroy.

azurerm_resource_group.rg: Destruction complete after 1m50s

Destroy complete! Resources: 7 destroyed.
```

Resources created in the Azure console:



bookRG

Resource group

Showing 1 to 6 of 6 records. □ Show hidden types

Name	Type	Location
book-nic	Network interface	West Europe
book-osdisk	Disk	West Europe
book-vnet	Virtual network	West Europe
bookvm	Virtual machine	West Europe
testbookip	Public IP address	West Europe
testdevopsstore	Storage account	West Europe