Step by Step creating Resource Group, VM, Storage Account with Terraform in Azure.

## 1. Resource Group

- Open VS Code and create new folder (ctrl+O),
- Create new file and name it main.tf

```
• In the file past below script,
   # We strongly recommend using the required providers block to set the
   # Azure Provider source and version being used
   terraform {
    required_providers {
      azurerm = {
      source = "hashicorp/azurerm"
      version = "=3.0.0"
     }
    }
   }
   # Configure the Microsoft Azure Provider
    provider "azurerm" {
    features {}
   }
   # Create a resource group
    resource "azurerm_resource_group" "akin123" {
    name = "akin123"
    location = "West us"
    tags = {
```

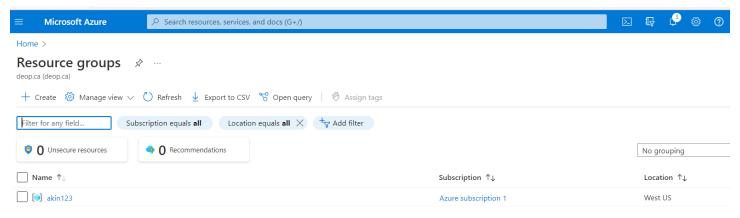
- from the terminal, first, login your az portal with using "az login" command.
- After login your azure portal, provision below commands:

terraform init terraform plan terraform apply

}

"terraform" = "value"

```
$ terraform apply
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_resource_group.akin123 will be created
  + resource "azurerm_resource_group" "akin123" {
                = (known after apply)
      + id
      + location = "westus"
                 = "akin123"
     + name
      + tags
                 = {
          + "terraform" = "value"
azurerm_resource_group.akin123: Creating...
azurerm_resource_group.akin123: Creation complete after 1s [id=/subscriptions/658b7aa8-6d79-4894-a694-a6e02
20fdcae/resourceGroups/akin123]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



#### 2. VM

- To get VM first we should find Windows virtual machine script from terraform web site; https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/windows virtual machine
- As a second step copy all code from webpage to VS code:

```
resource "azurerm_resource_group" "example" {
  name = "example-resources"
  location = "West Europe"
}

resource "azurerm_virtual_network" "example" {
  name = "example-network"
  address_space = ["10.0.0.0/16"]
  location = azurerm_resource_group.example.location
```

```
resource_group_name = azurerm_resource_group.example.name
}
resource "azurerm_subnet" "example" {
              = "internal"
 name
resource_group_name = azurerm_resource_group.example.name
virtual_network_name = azurerm_virtual_network.example.name
address_prefixes = ["10.0.2.0/24"]
}
resource "azurerm_network_interface" "example" {
 name
              = "example-nic"
              = azurerm_resource_group.example.location
location
 resource_group_name = azurerm_resource_group.example.name
 ip configuration {
                    = "internal"
  name
  subnet id
                     = azurerm_subnet.example.id
  private_ip_address_allocation = "Dynamic"
 }
}
resource "azurerm_windows_virtual_machine" "example" {
 name
              = "example-machine"
 resource_group_name = azurerm_resource_group.example.name
 location
              = azurerm_resource_group.example.location
            = "Standard F2"
 size
                   = "adminuser"
 admin username
                   = "P@$$w0rd1234!"
 admin password
 network_interface_ids = [
  azurerm network interface.example.id,
]
```

```
os_disk {
  caching = "ReadWrite"
  storage_account_type = "Standard_LRS"
}
 source_image_reference {
  publisher = "MicrosoftWindowsServer"
  offer = "WindowsServer"
  sku = "2016-Datacenter"
 version = "latest"
}
}
  To create VM we should create.
   RG
   Virtual network
   Subnet
   NIC
   ΙP
   User ID and password
   OS
```

### Modified ones:

```
resource "azurerm_virtual_network" "vnet123" {
                   = "vnet123"
 name
 address_space = ["10.0.0.0/16"]
                   = azurerm_resource_group.akin123.location
 location
  resource_group_name = azurerm_resource_group.akin123.name
resource "azurerm_subnet" "subnet123" {
                      = "subnet123"
  name
  resource_group_name = azurerm_resource_group.akin123.name
 virtual_network_name = azurerm_virtual_network.vnet123.name
  address_prefixes = ["10.0.2.0/24"]
resource "azurerm_network_interface" "nic123" {
                    = "akin123-nic"
  name
                    = azurerm_resource_group.akin123.location
  location
  resource_group_name = azurerm_resource_group.akin123.name
```

```
ip_configuration {
                                  = "internal"
   name
   subnet id
                                  = azurerm subnet.subnet123.id
   private_ip_address_allocation = "Dynamic"
resource "azurerm windows virtual machine" "vm123" {
                     = "vm123"
 resource_group_name = azurerm_resource_group.akin123.name
              = azurerm_resource_group.akin123.location
 location
                    = "Standard F2"
 size
 admin_username = "xxxxxx"
admin_password = "xxxxxx"
 network_interface_ids = [
   azurerm_network_interface.vnet123.id,
 os_disk {
   caching
                        = "ReadWrite"
   storage_account_type = "Standard_LRS"
 source_image_reference {
   publisher = "MicrosoftWindowsServer"
   offer = "WindowsServer"
   sku = "2016-Datacenter"
   version = "latest"
```

```
Akin@worker MINGW64 ~/Desktop/Git_Terraform/Akin
$ terraform plan
azurerm_resource_group.akin123: Refreshing state... [id=/subscriptions/658b7aa8-6d79-4894-a6e0220fdcae
/resourceGroups/akin123]
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 network interface.nic123 will be created
  + resource "azurerm_network_interface" "nic123" {
     + applied_dns_servers = (known after apply)
                                   = (known after apply)
     + dns_servers
     + enable_accelerated_networking = false
     + enable_ip_forwarding = false
     + internal_dns_name_label = (known_after apply)
+ internal_demoi
     + internal_domain_name_suffix = (known after apply)
+ location = "westus"
                                   = (known after apply)
     + mac_address
     = "nic123"
                                   = (known after apply)
                                  = (known after apply)
                                   = "akin123"
                                   = (known after apply)
```

```
+ ip_configuration {
        + gateway_load_balancer_frontend_ip_configuration_id = (known after apply)
                                                              = "internal"
        + name
        + primary
                                                              = (known after apply)
                                                              = (known after apply)
        + private ip address
                                                              = "Dynamic"
        + private_ip_address_allocation
                                                              = "IPv4"
       + private ip address version
        + subnet_id
                                                              = (known after apply)
      }
  }
# azurerm_subnet.subnet123 will be created
 resource "azurerm_subnet" "subnet123" {
   + address prefixes
                                                      = [
        + "10.0.2.0/24",
    + enforce_private_link_endpoint_network_policies = false
    + enforce_private_link_service_network_policies = false
    + id
                                                      = (known after apply)
                                                      = "subnet123"
   + name
                                                      = "akin123"
    + resource group name
                                                      = "vnet123"
    + virtual network name
```

```
# azurerm_virtual_network.vnet123 will be created
+ resource "azurerm virtual network" "vnet123" {
   + address space
        + "10.0.0.0/16",
    + dns_servers = (known after apply)
   + guid = (known after apply)
+ id = (known after apply)
+ location = "westus"
+ name = "vnet123"
   + resource_group_name = "akin123"
    + subnet = (known after apply)
  }
# azurerm_windows_virtual_machine.vm123 will be created
+ resource "azurerm_windows_virtual_machine" "vm123" {
   + admin_password = (sensitive value)
+ admin_username = "akin3540"
    + allow_extension_operations = true
   + computer_name = (known after apply)
   + enable_automatic_updates = true
   + extensions_time_budget = "PT1H30M"
+ hotpatching_enabled = false
    + hotpatching_enabled
   + id
                                   = (known after apply)
```

```
# azurerm_windows_virtual_machine.vm123 will be created
+ resource "azurerm_windows_virtual_machine" "vm123" {
   + admin_password = (sensitive value)
+ admin_username = "akin3540"
                                 = "akin3540"
   + admin_username
   + allow_extension_operations = true
   + computer_name = (known after apply)
    + enable_automatic_updates = true
   + extensions_time_budget = "PT1H30M"
   + hotpatching_enabled
                                 = false
   + id
                                 = (known after apply)
                                 = "westus"
   + location
   + max_bid_price
                                 = -1
                                 = "vm123"
    + name
   + network_interface_ids = (known after apply)
   + patch_mode
                                 = "AutomaticByOS"
    + platform_fault_domain = -1
    + priority
                                 = "Regular"
   + private_ip_address = (known after apply)
+ private_ip_addresses = (known after apply)
+ provision_vm_agent = true
    + public_ip_address
                                 = (known after apply)
   + public_ip_addresses = (known after apply)
+ resource_group_name = "akin123"
                                 = "Standard F2"
   + size
    + virtual_machine_id
                                  = (known after apply)
```

```
Plan: 4 to add, 0 to change, 0 to destroy.

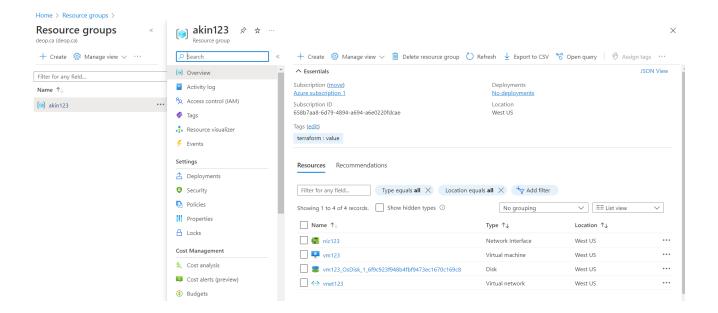
Do you want to perform these actions?

Terraform will perform the actions described above.

Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_virtual_network.vnet123: Creating...
```



## 3. Storage Account:

• On the left pane of the terraform website find azurerm storage account

https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage\_account

copy and paste entire template into VS Code

and follow the steps as part one.
 (you can see in the last picture on azure portal which is storage account created from terraform.

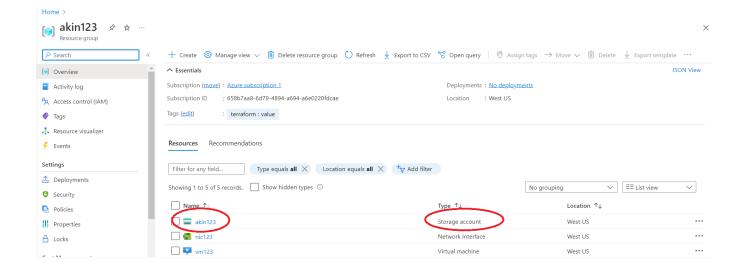
```
Do you want to perform these actions?

Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

azurerm_storage_account.storage_account: Creating...
azurerm_storage_account.storage_account: Still creating... [10s elapsed]
azurerm_storage_account.storage_account: Still creating... [20s elapsed]
azurerm_storage_account.storage_account: Creation complete after 27s [id=/subscriptions/658b7aa8-6d79-4894-a694-a6e0220fdcae/resourceGroups/akin123/providers/Microsoft.Storage/storageAccounts/akin123]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```



# Thank you!