# STARTING WITH TERRAFORM:

* Before starting it we need active subscription.
* And making app registration that has to be given access according to its requirements.

Code is must be in a main.tf:

*provider* "azurerm" {

    version = "=2.0.0"

  features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

*resource* "azurerm\_resource\_group" "resource\_gp" {

name = "hello"

location = "eastus2"

}

## [»](https://learn.hashicorp.com/terraform/azure/configure_az#configuration-file-format)Configuration file format

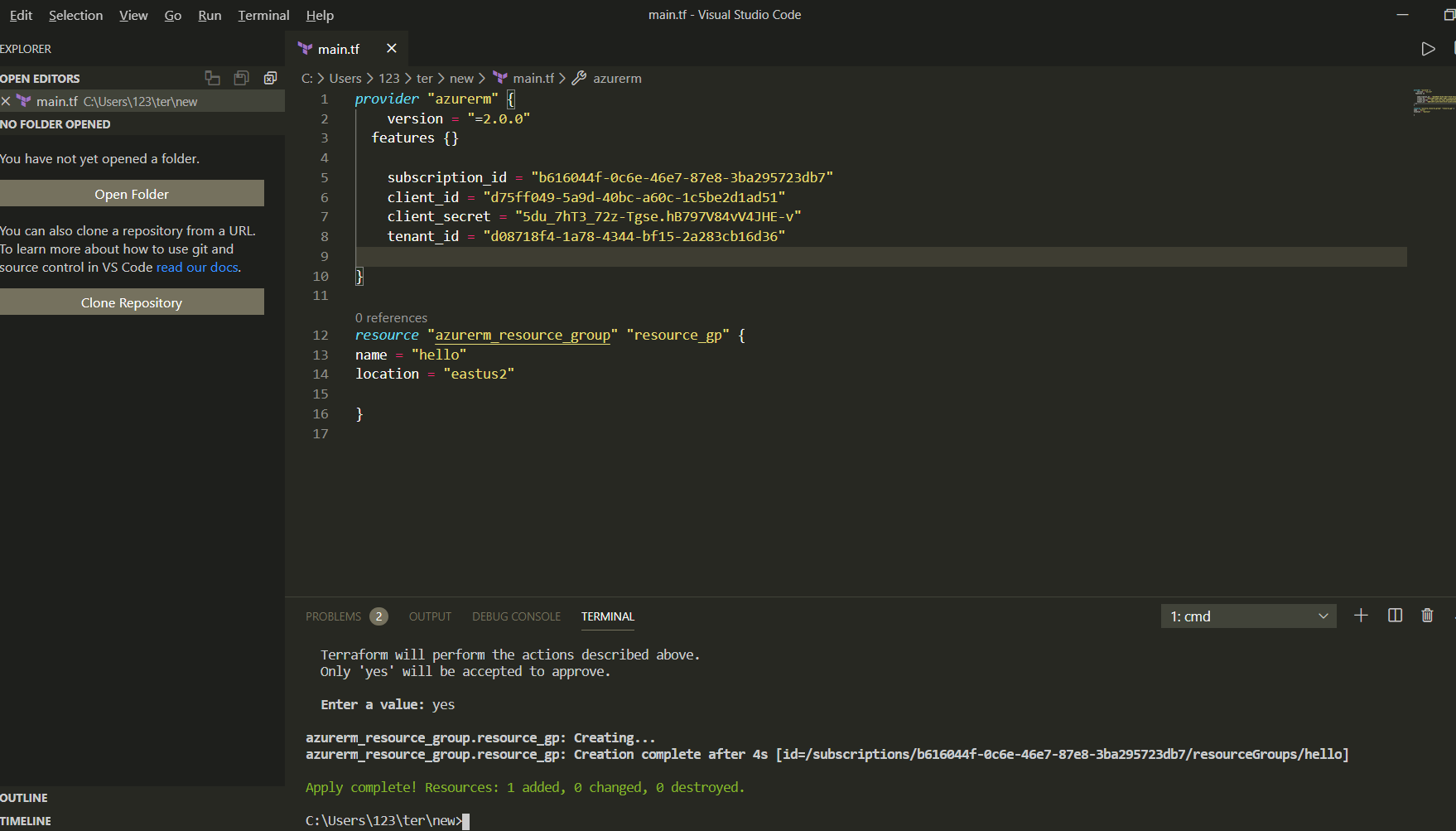
Configuration files can be in either of two formats: HashiCorp Configuration Language (HCL), or JSON. HCL is a structured language created with DevOps in mind; it is machine-friendly yet easy for humans to read, and it supports comments. HCL format files have a .tf extension. JSON is sometimes preferable when configurations are generated by a machine. JSON files have a .tf.json extension. A configuration can be composed of both .tf and .tf.json files. In general, we recommend that you work with HCL.

CODE SUCESSFULLY RUN:

COMMANDS ARE:

* Terraform init
* Terraform plan
* Terraform apply

SCREENSHOTS FOR DEPLOYMNETS:



TERRAFORM SCRIPT FOR VNET WITH MULTIPLE SUBNETS AND WITH NSG CONFIGURED:

*variable* "vnet\_name" {

  description = "Name of the vnet to create"

  default     = "acctvnet"

}

*variable* "resource\_group\_name" {

  description = "Name of the resource group to be imported."

}

*variable* "address\_space" {

  type        = list(*string*)

  description = "The address space that is used by the virtual network."

  default     = ["10.0.0.0/16"]

}

# If no values specified, this defaults to Azure DNS

*variable* "dns\_servers" {

  description = "The DNS servers to be used with vNet."

  default     = []

}

*variable* "subnet\_prefixes" {

  description = "The address prefix to use for the subnet."

  default     = ["10.0.1.0/24" , "10.0.2.0/24" , "10.0.3.0/24"]

}

*variable* "subnet\_names" {

  description = "A list of public subnets inside the vNet."

  default     = ["subnet1", "subnet2", "subnet3"]

}

*variable* "nsg\_ids" {

  description = "A map of subnet name to Network Security Group IDs"

  type        = map(*string*)

  default = {

  }

}

*variable* "tags" {

  description = "The tags to associate with your network and subnets."

  type        = map(*string*)

  default = {

    ENV = "test"

  }

}

*provider* "azurerm" {

    version = "=2.0.0"

  features {}

    subscription\_id = "b616044f-0c6e-46e7-87e8-3ba295723db7"

    client\_id = "d75ff049-5a9d-40bc-a60c-1c5be2d1ad51"

    client\_secret = "5du\_7hT3\_72z-Tgse.hB797V84vV4JHE-v"

    tenant\_id = "d08718f4-1a78-4344-bf15-2a283cb16d36"

}

#Azure Generic vNet Module

*data* azurerm\_resource\_group "vnet" {

  name = var.resource\_group\_name

}

*resource* azurerm\_virtual\_network "vnet" {

  name                = var.vnet\_name

  resource\_group\_name = data.azurerm\_resource\_group.vnet.name

  location            = data.azurerm\_resource\_group.vnet.location

  address\_space       = var.address\_space

  dns\_servers         = var.dns\_servers

  tags                = var.tags

}

*resource* "azurerm\_subnet" "subnet" {

  count                = length(var.subnet\_names)

  name                 = var.subnet\_names[count.index]

  resource\_group\_name  = data.azurerm\_resource\_group.vnet.name

  virtual\_network\_name = azurerm\_virtual\_network.vnet.name

  address\_prefix       = var.subnet\_prefixes[count.index]

}

*data* "azurerm\_subnet" "import" {

  for\_each             = var.nsg\_ids

  name                 = each.key

  resource\_group\_name  = data.azurerm\_resource\_group.vnet.name

  virtual\_network\_name = azurerm\_virtual\_network.vnet.name

  depends\_on = ["azurerm\_subnet.subnet"]

}

*resource* "azurerm\_subnet\_network\_security\_group\_association" "vnet" {

  for\_each                  = var.nsg\_ids

  subnet\_id                 = data.azurerm\_subnet.import[each.key].id

  network\_security\_group\_id = each.value

  depends\_on = ["data.azurerm\_subnet.import"]

}

SCREENSHOTS FOR DEPLOYMENTS:

