

Terraform, getting started

v11 VS v12

- **terraform.tfvars**

- no necesidad entrecomillar las variables

<pre>9.1 V12 Syntax changes > 11 > terraform.tfvars 1 # Version 11 Syntax 2 3 "terraform_script_version" = "1.00" 4 "web_server_subnets" = ["1.0.1.0/24","1.0.2.0/24"]</pre>	<pre>9.1 V12 Syntax changes > 12 > terraform.tfvars 1 # Version 12 Syntax 2 3 terraform_script_version = "1.00" 4 web_server_subnets = ["1.0.1.0/24","1.0.2.0/24"]</pre>
--	---

- ***.tf**

- no interpolation para el uso de variables
- si aún en expresiones dentro de strings
- en el caso de azure, para usar versión 12, la versión del **provider** ha de ser ≥ 1.27
- los bloques dentro de un **resource** han/pueden usar **=** (línea 25 de la sintaxis para v12)

<pre>main.tf x terraform.tfvars 9.1 V12 Syntax changes > 11 > main.tf 1 # Version 11 Syntax 2 3 variable "subscription_id" {} 4 variable "client_id" {} 5 variable "client_secret" {} 6 variable "tenant_id" {} 7 8 variable "terraform_script_version" {} 9 variable "web_server_subnets" { 10 type = "list" 11 } 12 13 provider "azurerm" { 14 version = "1.16" 15 tenant_id = "\${var.tenant_id}" 16 subscription_id = "\${var.subscription_id}" 17 client_id = "\${var.client_id}" 18 client_secret = "\${var.client_secret}" 19 } 20 21 resource "azurerm_resource_group" "web_server_rg" { 22 name = "web-rg" 23 location = "westus2" 24 25 tags { 26 build-version = "\${var.terraform_script_version}" 27 interpolation = "\${var.terraform_script_version}-v11-\${length(var.web_server_subnets)}" 28 } 29 }</pre>	<pre>main.tf x terraform.tfvars 9.1 V12 Syntax changes > 12 > main.tf 1 # Version 12 Syntax 2 3 variable "subscription_id" {} 4 variable "client_id" {} 5 variable "client_secret" {} 6 variable "tenant_id" {} 7 8 variable "terraform_script_version" {} 9 variable "web_server_subnets" { 10 type = "list" 11 } 12 13 provider "azurerm" { 14 version = "1.36" 15 tenant_id = var.tenant_id 16 subscription_id = var.subscription_id 17 client_id = var.client_id 18 client_secret = var.client_secret 19 } 20 21 resource "azurerm_resource_group" "web_server_rg" { 22 name = "web-rg" 23 location = "westus2" 24 25 tags = { 26 build-version = var.terraform_script_version 27 interpolation = "\${var.terraform_script_version}-v12-\${length(var.web_server_subnets)}" 28 } 29 }</pre>
---	---

installs

- visual studio code + plugin terraform
- git
- azure cli `curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash:`

```
sudo apt-get update
sudo apt-get install ca-certificates curl apt-transport-https lsb-release
gnupg
curl -sL https://packages.microsoft.com/keys/microsoft.asc |
  gpg --dearmor |
  sudo tee /etc/apt/trusted.gpg.d/microsoft.asc.gpg > /dev/null
AZ_REPO=$(lsb_release -cs)
echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/
$AZ_REPO main" |
  sudo tee /etc/apt/sources.list.d/azure-cli.list
sudo apt-get update
```

```
sudo apt-get install azure-cli
```

- <https://docs.microsoft.com/ca-es/cli/azure/?view=azure-cli-latest>

azure SPN

- **Tenant:** organización. Instancia dedicada de tu AD
- **Azure AD:** AD para la gestión de permisos en el cloud de Azure
- **SPN** = Service Principal Name. Cuenta de aplicación para autenticación y autorización para servicios automatizados
- **Subscriptions:**
 - ligadas a un AD (nuestro AD), puede haber varias de ellas
 - se puede autorizar a un SPN a una o varias subscripciones
- **SPK**
 - portal → Azure Active Directory → App registrations → ADD
 - valores por defecto
 - guardar los siguientes valores:
 - client ID¹⁾: 🔒 🔒
 - client_secret: 🔒 🔒
 - en Certificates & secrets → Add a new client secret
 - tenant ID: 🔒 🔒
 - Portal → Subscriptions → Evaluación Gratuita (Pay-As-You-Go)
 - subscription ID: 🔒 🔒
 - Autorizar a nuestro SPN **terraform**
 - portal → subscriptions → Pay-As-You-Go → IAM → Add a role assignment
 - Role = contributor, Assign access to = Azure AD user, group o service principal, terraform

Providers

- En terraform, los **providers** es la manera de hablar con los diferentes servicios (en cloud)
- el lenguaje de terraform está pensado para ser inteligible para los humanos
- creamos el siguiente script terraform:

```
provider "azurerm" {  
  version      = ">=1.27"  
  client_id    = "6c609608-a350-476f-85ff-eaff7e82f981"  
  client_secret = "qW0S]tToc]=]ub8HM7yciLSnw0Wh1emD"  
  tenant_id    = "c8537169-fe81-494a-a2dd-b4ea067073a1"  
  subscription_id = "2eaf2655-54e9-4b81-843e-86b09fd71ab1"  
  features {} # terraform v12  
}
```

Running terraform

- **terraform init:** inicialización, descarga según el proveedor, mira los módulos...
- **terraform plan:** mira los recursos definidos en tu código y los compara con los que existen e indica que acciones se tomarán
- añadimos un **resource** a nuestro código:

```
resource "azurerm_resource_group" "web_server_rg" {  
  name      = "web-rg"  
  location = "West Europe"  
}
```

◦ <https://azure.microsoft.com/en-us/global-infrastructure/locations/>

- terraform init, terraform plan, terraform apply, terraform destroy

1)

2s

From:

<https://matebcn.eu/> - miguel angel torres egea

Permanent link:

<https://matebcn.eu/info:cursos:udemy:terraform-azure:getting-started>

Last update: **29/03/2020 19:22**

