

---

---

# Terraform 101

Guía Inicial  
Jesús A. Suárez

---

---

# Prerrequisitos

- Crear cuenta en AWS (free tier)
  - Tener instalado WSL2 con Ubuntu 20.04
  - Tener instalado AWS CLI
  - Tener un usuario administrativo
  - Crear accesos programáticos al usuario administrativo
  - Configurar el AWS CLI con los accesos programáticos y autenticar en AWS desde el AWS-CLI
  - Visual Studio Code con plug-in de Hashicorp Terraform
-

---

# Instalar la última versión de Terraform

## Configurar repositorio:

```
sudo apt update  
sudo apt upgrade  
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -  
sudo apt-add-repository "deb [arch=$(dpkg --print-architecture)] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
```

## Instalar terraform:

```
sudo apt install terraform
```

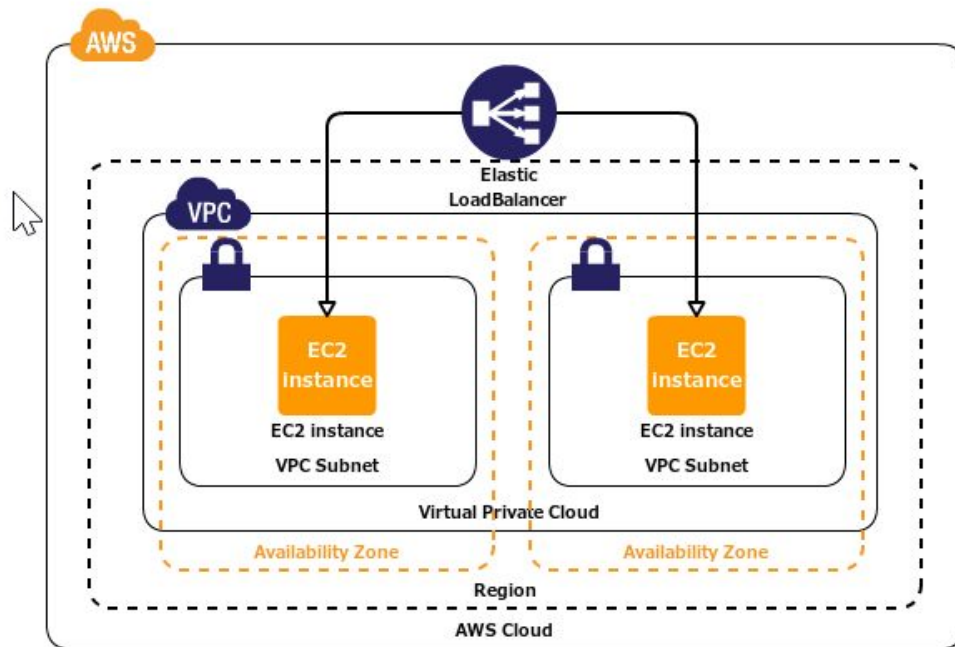
## Validar instalación:

```
terraform version
```

---

# ¿Qué vamos a implementar?

- VPC
- Subnets
- Route Tables
- Internet Gateway
- Instances
- ELB
- etc.



---

# Forma de trabajo

- Terraform se declara la configuración en archivos con extensión .tf
  - Hay dos tipos de formatos que terraform usa, una para humanos y otra para máquinas.
  - Para humanos se utiliza el formato Hashicorp Configuration Language (HCL), y para máquinas se genera archivos json a partir del HCL. Estos últimos con extensión .tf.json. Si lo prefiere puede usar los json directamente.
  - Inicialización con **terraform init**
  - Luego de ir agregando recursos a los .tf, puede ir validando el estado con **terraform plan**. Corregir si es necesario.
  - Para aplicar los cambios en el provider ejecutar **terraform apply**
-

---

# Forma de trabajo

- Terraform crea los recursos y mantiene un estado en el archi **terraform.tfstate**
  - En caso de despliegues por un equipo se estila compartir el estado en un storage S3.
  - Puede validar el estado se utiliza **terraform show**
  - Es posible exportar un graph de la infraestructura con **terraform graph** y exportar a un archivo DOT, requiere graphviz
  - Y finalmente para eliminar todos los recursos **terraform destroy**
-

---

# Provider

- El provider terraform es plug-in que permite conectarse y administrar recursos con diferentes proveedores de nube, SaaS, PaaS, etc. Como AWS, Azure, Google, Docker, Kubernetes, Github, Heroku, Consul, Ansible, etc.

Component      Name



```
provider "aws" {  
  access_key = "abc123"  
  secret_key = "abc123"  
  region     = "us-east-1"  
}
```

---

# Resources

Component



Provider



Type



Name



```
resource "aws_instance" "base" {  
  ami = "ami-408c7f28"  
  instance_type = "t1.micro"  
}
```



---

# Variables

- Variables de tipo string

```
variable "region" {  
  description = "The AWS region."  
  default     = "us-east-1"  
}
```

```
provider "aws" {  
  access_key = "${var.access_key}"  
  secret_key = "${var.secret_key}"  
  region     = "${var.region}"  
}
```

- De tipo list

```
variable "security_group_ids" {  
  type      = "list"  
  description = "List of security group IDs."  
  default = ["sg-4f713c35", "sg-4f713c35", "sg-4f713c35"]  
}
```

```
vpc_security_group_ids = "${var.security_group_ids[1]}"
```

- De tipo map

```
variable "ami" {  
  type = "map"  
  default = {  
    us-east-1 = "ami-0d729a60"  
    us-west-1 = "ami-7c4b331c"  
  }  
  description = "The AMIs to use."  
}
```

```
resource "aws_instance" "base" {  
  ami           = "${lookup(var.ami, var.region)}"  
  instance_type = "t2.micro"  
}
```

`"${lookup(map, key)}"`

---

---

# outputs

Las salidas se pueden ver en las salidas del apply y ejecutando:  
**terraform output**

```
output "public_subnet_id" {  
  value = "${aws_subnet.public.id}"  
}
```

```
output "vpc_id" {  
  value = "${aws_vpc.tfb.id}"  
}
```

---

---

# Usando GIT

- Antes de usar git, hay que evitar subir al repositorio el archivo con el estado, y toda la carpeta .terraform
  - Crear archivo .gitignore con las siguientes líneas y luego ejecutar git add .gitignore

```
▶ .gitignore
1   .terraform/
2   terraform.tfstate*
3   .terraform.lock.hcl
4
```