

## Automatización de despliegues en entornos Cloud

Asignatura	Datos del alumno	Fecha
Experto Universitario en DevOps y Cloud	Apellidos ,Nombre: SOLIS LOPEZ ,RAFAEL	22/07/2022

### 1. Introducción

#### 1.1 Objetivo

El objetivo de esta tarea es la realización de la creación de una determinada infraestructura de software basada en el paradigma de programación denominado “Infraestructura como Código” a través de dos de los softwares más conocidos actualmente en el mercado que nos permiten definir y configurar recursos de la plataforma IaaS (Infraestructura como Servicio) de diferentes proveedores cloud (Azure, AWS, GCP) empleando para ello el lenguaje de alto nivel DSL (Domain Specific Language), como son:

- **Terraform.**
- **Asinble.**

#### 1.2 Breve descripción Terraform y Asinble

Ambas tecnología permiten poseer características similares:

- Permiten desarrollar IaC (Infraestructura como Servicio).
- No necesitan de servidor central para conservar el estado de la infraestructura.
- Son de código abierto.

En el siguiente cuadro comparativo podemos apreciar sus principales diferencias:

Criterios	Ansible	Terraform
Fundado en	2012	2014
Desarrollado por	Ansible Inc./Red Hat Inc.	HashiCorp
Escrito en	Python	Go
Escribe	Herramienta de gestión de la configuración	Herramienta de orquestación
Idioma	Utiliza lenguajes procedimentales y declarativos	Utiliza un lenguaje declarativo
Infraestructura	Admite infraestructura mutable	Admite infraestructura inmutable
Gestión del ciclo de vida	No	sí
Empaque y Plantillas	Totalmente compatible	Parcial
Redes y aprovisionamiento de máquinas virtuales	Parcial	Totalmente compatible

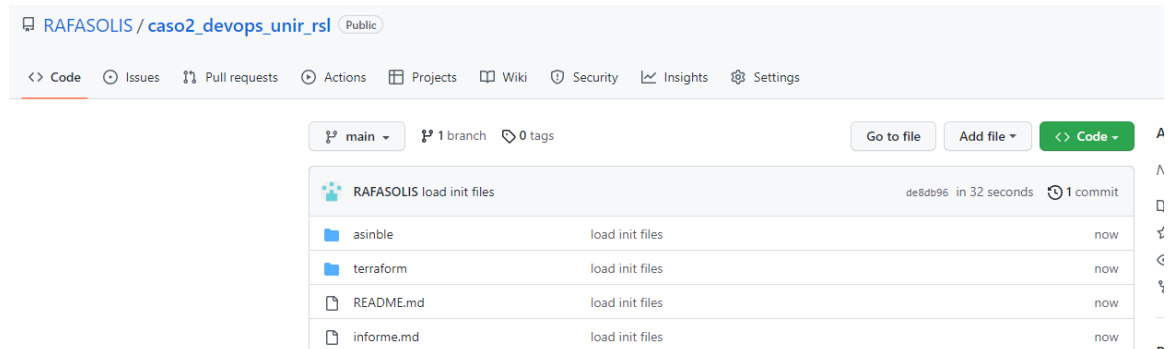
Fuente: <https://blogging-techies.com/descripcion-de-las-herramientas-devops-ansible-y-terraform/>

Terraform sería más recomendado para casos en los que la infraestructura a desarrollar va a permanecer estática, siendo Asinble más recomendado en los casos en los que la infraestructura esté en constante evolución.

## 2. Creación repositorio Github.

URL: [https://github.com/RAFASOLIS/caso2\\_devops\\_unir\\_rsl.git](https://github.com/RAFASOLIS/caso2_devops_unir_rsl.git)

Estructura:




## 3. Creación cuenta Azure y service principal.

Creamos una cuenta gratuita en Azure con nuestro correo del curso de experto ([rafael.solis886@comunidadunir.net](mailto:rafael.solis886@comunidadunir.net)).

Debido a una incidencia no tenemos acceso a AD de Azure con la suscripción creada ni tenemos posibilidad de modificar el directorio.

No access

Summary 	
Session ID	Resource ID
8b09013c08f2438692857efd1890d461	Not available
Extension	Content
Microsoft_AAD_IAM	ActiveDirectoryMenuBlade
Error code	
403	

# Change the directory



Changing the directory doesn't change billing ownership for the subscription. You won't be able to delete the original directory until billing ownership is transferred to someone else. [Learn more](#)

From

UNIR (www.comunidadunir.net)

To

Select a directory

Cannot change directory because you only have access to one directory.

## 4. Descarga y configuración de la VM

- Creamos una Instancia Centos en Azure para la gestión de la tarea.

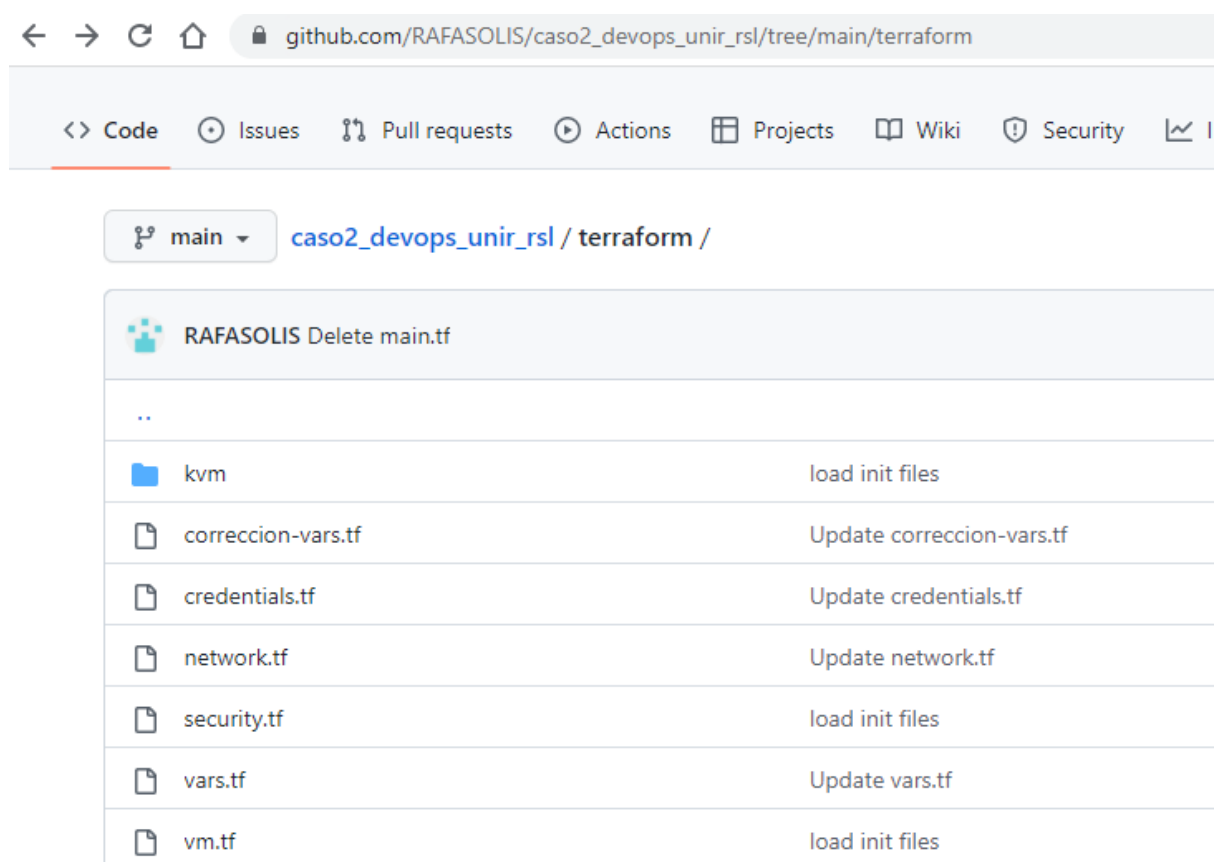
- Instalamos Azure CL.
- Instalamos Terraform CLI.
  - `mv ~/Downloads/terraform /usr/local/bin/`
  - `touch ~/.bashrc`
  - `terraform -install-autocomplete`
- Instalamos Python 3 y Asinble ([https://docs.ansible.com/ansible/latest/installation\\_guide/intro\\_installation.html#installing-devel-from-github-with-pip](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-devel-from-github-with-pip))
- Instalar cloud-init.
- Actualizamos los paquete.

## 5. Desarrollo plan Terraform.

A través de los diferentes ficheros de configuración que a continuación se muestran se declaran los diferentes componentes que serán necesarios para desplegar la infraestructura final.

Lanzamos el comando en la consola: `terraform init` para iniciar terraform en nuestro servidor y posteriormente `terraform plan` para crear un plan Terraform en el que definiremos todos valores para configurar la infraestructura a montar, en este caso en el cloud de Azure. L

- **Correcction-var.tf.** El nombre de la cuenta de almacenamiento ,usuario y claves SSH pública para la conexión con los recursos cloud.
- **Credentials.tf.** Contiene el ID de la suscripción así como el nombre y clave del cliente para la conexión con nuestros recursos cloud.
- **network.tf.** Contendrá la configuración de la red y subred virtual creada para el grupo de recursos del desarrollo.
- **var.tf.** Indicamos el nombre de la instancia de VM a desplegar en Azure.
- **vm.tf.** Nos permite indicar todos los parámetros para crear las diferentes instancias de VM desde nuestra suscripción y grupo de recursos de AZ.

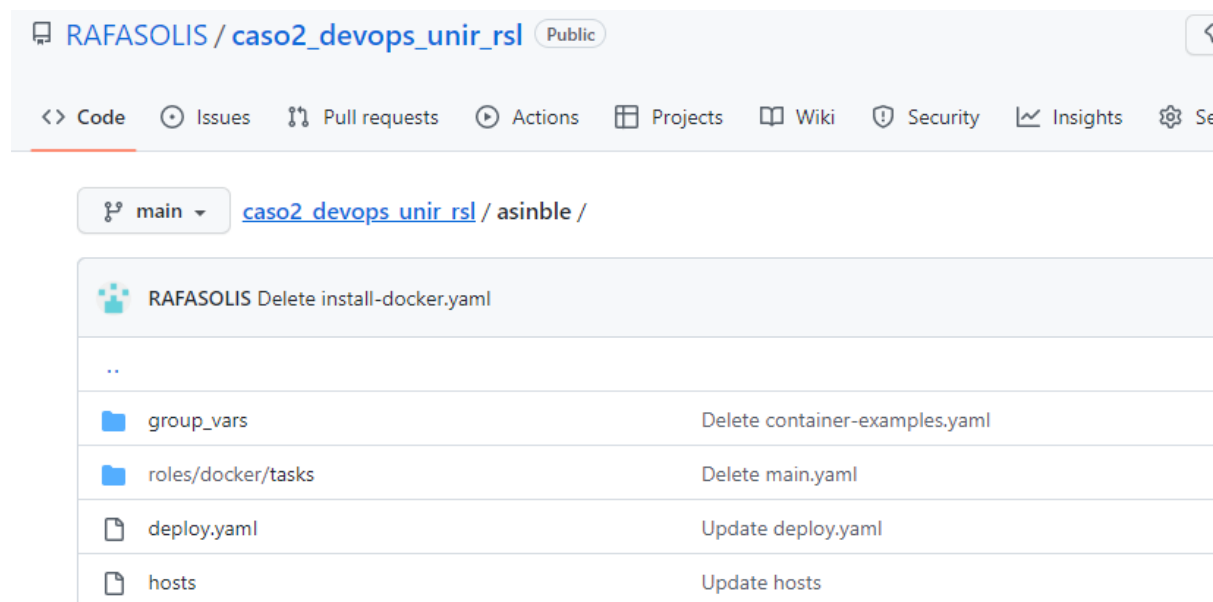


A través de los siguientes comandos Terraform podemos gestionar la creación de la infraestructura a crear.

Comando Terraform	Acción realizada
>> terraform init	inicia el directorio en el servidor para que Terraform pueda ser ejecutado
>> terraform plan	Crea plan Terraform
>> terraform apply	Escanea el directorio creado y realiza las siguientes acciones
	Lee el estado de los recursos definidos en el fichero .tfstate
	Obtiene los datos definidos tfstate del Provider
	Genera la configuración a aplicar leyendo todos los ficheros .tf
	Aplica la nueva configuración para crear la infraestructura

## 6. Desarrollo Playbooks Asinble.

- **deploy.yaml.** En este playbook se llama a los diferentes
- **host.yaml.** Este fichero se usa de inventario para rastrear todos los componentes que el fichero deploy debe lanzar a través de la api Asinble.
- 



## 7. Diagrama de la arquitectura.

En el siguiente diagrama se muestra como partiendo de la definición de unos ficheros .tf definimos los componentes de arquitectura que posteriormente creará la API Terraform en nuestra cuenta de cloud Azure. La clave SSH permitirá conectar Terraform y Asinble. Asinble, a través de los playbooks y rastreando los componentes definidos en el fichero de inventario gestionará la configuración de la infraestructura creada, en este caso las 2 máquinas virtuales con el sistema operativo Centos 8 y la configuración especificada en el proceso.

