

## Parcial Infraestructura II - TEMA 2



XXXXXXXXXX.com (no se comparten)

[Cambiar cuenta](#)



### Preguntas Multiple Choice

Indicá etapas correctas del ciclo de vida de DevOps

- ☒ Pruebas
- ☒ Lanzamiento
- ☐ Debug
- ☐ Desplazamiento

¿Que perfiles encontramos en ecosistema DevOps?

- ☐ Analista de hardware
- ☒ Desarrolladores de aplicaciones
- ☒ Analistas de CI/CD
- ☐ Experto en telecomunicaciones

La Infraestructura como código nos propone..

- ☒ Gestionar la configuración de la infraestructura.
- ☐ Implementar servidores virtuales más fácilmente.
- ☐ Publicar una aplicación de manera automática.

¿Cuales son los beneficios de la IaC?

- ☐ Reducir el error de los sistemas.
- ☒ Reducción de costos.
- ☒ Estandarización de la configuración.
- ☐ Reduccion de los testeos.



Ansible es una herramienta IaC que funciona...

- ☐ solo en AWS
- ☐ solo en Infraestructura On Premise
- ☐ solo en maquinas virtuales
- ☒ todas las anteriores

Borrar la selección

Si utilizo AWS y Azure, ¿que herramienta me recomendás?

- ☐ CloudFormation
- ☐ Ansible
- ☒ Terraform

Borrar la selección

El Principio de Idempotencia se define como

- ☐ Posibilidad de hacer despliegues continuos, obteniendo cambios permanentes en la Infraestructura
- ☐ Posibilidad de reduccion de costos, manteniendo la Infraestructura al minimo
- ☒ Automatización "n" cantidad de veces obteniendo siempre y en todos los casos el mismo resultado

Borrar la selección

Ansible nacio como una herramienta de Configuration Managment y luego se extendió a Infrastructure as Code

- ☒ Verdadero
- ☐ Falso

Borrar la selección

¿Cual característica define a los modulos de Terraform?

- ☐ Son estaticos
- ☒ Son reutilizables
- ☐ Son inmutables

Borrar la selección

Terraform y Ansible destruyen la infraestructura utilizando el mismo codigo con la que se creo

- ☐ Si, ambas herramientas lo haces
- ☒ Solo lo hace Terraform
- ☐ Solo lo hace Ansible

Borrar la selección

Atrás

Siguiente

Borrar formulario

Nunca envíes contraseñas a través de Formularios de Google.

El formulario se creó en Digital House. [Denunciar abuso](#)

## Parcial Infraestructura II - TEMA 2



**[Redacted Name]** (no se comparten)

[Cambiar cuenta](#)



Se guardó el borrador

### Preguntas a desarrollar

Describi paso a paso cómo funciona Terraform.

Para automatizar los despliegues y cambios Terraform tiene dos entradas en su arquitectura. Una es la configuración concreta que hayamos escrito en un archivo tf, la otra es el estado que almacena Terraform con la infraestructura ya desplegada. Con eso es que puede crear el plan comparando el estado actual con la configuración.

1. Instalamos (si es que aún no lo hemos hecho) terraform
2. Creamos un archivo con extensión .tf. En este archivo vamos a escribir de manera declarativa utilizando HCL lo que queremos configurar.
  - a. Indicamos el proveedor que vamos a usar (nosotros venimos utilizando AWS) y declaramos las configuraciones que queramos hacerle. Algunas básicas podrían ser indicar la ubicación de nuestras credenciales o la región que vamos a usar
  - b. Declaramos los módulos con sus respectivas configuraciones
3. En nuestro caso al utilizar cuentas de aws educate deberíamos ir a la consola de AWS para obtener nuestras credenciales. Una vez obtenidas, las configuramos: esto lo podemos hacer creando el archivo .aws/credentials manualmente, o a través del comando aws configure. Si en el item 2a pusimos otra ubicación para las credenciales, debemos asegurarnos de crear ahí el archivo. Luego debemos agregarlas al PATH, cosa que en linux se hace con el comando export.
4. Con el archivo y las credenciales listas, nos ubicamos en la carpeta donde esté nuestro archivo .tf y ejecutamos el comando terraform init. Este comando va a descargarnos todos los módulos que hayamos indicado en nuestro archivo
5. Ejecutamos el comando terraform plan para ver el plan de acción de terraform
6. Si el plan es correcto, hacemos los cambios efectivos con terraform apply

Describe algunas características del tipo de Analista que usa Ansible

- Gestiona y administra servidores
- Le gusta el software open source
- Usa YAML
- Prefiere hacer las cosas de manera declarativa

Describe con tus palabras las ventajas de usar IaC frente al enfoque tradicional de la creación / administración de Infraestructura

Creo que uno de los principales beneficios es ahorramos el tener que darle las instrucciones sobre los pasos a seguir para aprovisionar o configurar un servidor. Podemos simplemente declarar el estado en el que queremos encontrarnos el servidor y listo. De acá se desprende la automatización de procesos que antes eran manuales, la reducción del error humano, la reducción de tiempos y costos. Además al tener el proceso automatizado podemos replicarlo las veces que queramos sabiendo que siempre vamos a obtener el mismo resultado.

Otras ventajas son:

- El control de versiones, porque al plasmar nuestra configuración en templates podemos versionarlas junto con el código fuente.
- Poder hacer pruebas en cualquier entorno desde el principio del ciclo de desarrollo

[Atrás](#)

[Siguiente](#)

[Borrar formulario](#)

## Parcial Infraestructura II - TEMA 2



[redacted] (no se comparten)

[Cambiar cuenta](#)



Se guardó el borrador

### Corregir una laC

Asumiendo que tengo una llave .pem denominada "millave", una vpc id:vpc-7a117c07, una subnet id:subnet-82bceedd, quiero crear 1 instancia Linux y poder conectarme a la misma via SSH, revisamos y corregimos el playbook de Ansible que esta en [https://github.com/repoinfradh/Infra2/blob/main/parcial/parcial\\_t2.yml](https://github.com/repoinfradh/Infra2/blob/main/parcial/parcial_t2.yml) . Debo marcar errores/faltantes y proponer la solucion al mismo.

Corrijo el codigo.

```
---
- hosts: localhost
  connection: local
  tasks:
    - name: Crear grupo de seguridad con HTTPS, HTTP y SSH
      ec2_group:
        name: sg_profe
        vpc_id: vpc-7a117c07
        description: sg con las reglas
        region: us-east-1
        rules:
          - proto: tcp
            ports:
              - 443
              - 80
              - 22
              - 8080
            cidr_ip: 0.0.0.0/0
            rule_desc: Acepto todo el trafico
    - name: Creamos nuestro servidor
      ec2:
        region: us-east-1
        instance_type: t2.micro
        image: ami-0c2b8ca1dad447f8a
        instance_tags:
          Name: Instancia_Profe
        wait: yes
        wait_timeout: 500
        # el nombre del grupo declarado en la linea 7 es sg_profe, no grupo_creado
        group: sg_profe
        volumes:
          - device_name: /dev/xvda
            volume_type: gp2
            volume_size: 8
        vpc_subnet_id: subnet-82bceedd
        # debería asignar una ip pública para conectarnos vía ssh
```

```
assign_public_ip: yes
# faltaba declarar la llave
key_name: millave
register: info
- name: DNS Publico de nuestro servidor
  debug:
    msg: "La ip publica es {{ info.instances[0].public_ip }} y su DNS es {{
info.instances[0].public_dns_name }}"
```

[Atrás](#)

[Enviar](#)

[Borrar formulario](#)

Nunca envíes contraseñas a través de Formularios de Google.

El formulario se creó en Digital House. [Denunciar abuso](#)

Google Formularios