

Trabajo Práctico Integrador

Tercera entrega: Deployment de Recursos de infraestructura

Se deberá generar los archivos de infraestructura como código (**Terraform**) necesarios que permitan desplegar algunos de los recursos de la arquitectura planteada. Como mínimo se deberán presentar seis (6) de los elementos definidos en el diagrama de arquitectura de la Segunda (2da) entrega.

Tener en cuenta que algunos de los elementos que puedan elegir pueden requerir de múltiples recursos (de Terraform) para funcionar. Por ejemplo, para crear un S3 Bucket se utilizará un resource para el bucket, otro para su ACL, otro para la configuración de versioning, otro para encryption, etc.). Todos estos pertenecen al mismo elemento de su infraestructura, es decir contarán como un solo elemento.

También es importante remarcar que todo recurso cuenta como único a menos que su uso y propósito sea distinto. Es decir, los buckets del front cuentan como un elemento de S3, las distribuciones de cloudfront cuentan como un elemento de CloudFront. Adicionalmente la cátedra recomienda evitar el desarrollo de Terraform de cluster de Kubernetes debido a la complejidad y tiempo que este implicaría. No obstante, es sólo una recomendación, no una restricción.

A continuación se detallan algunos enlaces de interés que pueden ayudar.

[Terraform Best Practices for AWS users](#)

[Terraform Best Practices](#)

[Docs overview](#) | [hashicorp/aws](#) | [Terraform Registry](#)

[Best practices for using Terraform](#) | [Google Cloud](#)

Componentes de la entrega 3:

- Repositorio de código (será provisto por la cátedra) con los archivos correspondientes, debiendo cumplir con los siguientes requisitos:
 1. Uso de [módulos](#).
 2. Uso de [variables y outputs](#).
 3. Uso de al menos cuatro (4) de las siguientes [funciones](#).
 4. Uso de al menos tres (3) siguientes meta-argumentos:

| | |
|-------------------------------|------------------------------|
| a. depends_on | c. count |
| b. for_each | d. lifecycle |

Rúbrica de evaluación:

- Correcta utilización de los componentes requeridos
- Cantidad y nivel de dificultad de los componentes elegidos.
- Diseño y estructura del código
 - Naming conventions
 - Estructura y formato del proyecto
 - Principio de DRY (Don't Repeat Yourself)
 - Parametrización de módulos