ISC	**Infraestructuras y Servicios Cloud**
25/26	Redes y Conectividad Avanzada y Contenedores
GIIA	**ACLARACIONES**

Práctica 4.b:

A raíz de las cuestiones planteadas en clase, hemos realizado este documento para aclarar y corregir algunas cuestiones de la práctica 4.

Prueba de la práctica

Presentamos dos posibles vías para probar la práctica de forma sencilla.

Protocolo de Prueba Simplificado (Curl)

El siguiente comando permite verificar la funcionalidad del Despliegue Canario (ruteo 90/10) sin necesidad de un *front-end* web, simulando una aplicación cliente que envía una petición POST al Balanceador de Carga.
Este método es la forma más rápida de validar que el **ALB**, los **Target Groups** (TG-V1 y TG-V2), y el **ruteo basado en peso (90/10)** están configurados correctamente.

---

1. Obtención del Punto de Acceso

Primero, el estudiante debe obtener el **Nombre DNS (URL)** de su Application Load Balancer (ALB) desde la consola de EC2.

- **URL Base:** http://<DNS_DEL_ALB>

2. Comando de Prueba del Despliegue Canario

La API (tanto V1 como V2) escucha en el *endpoint* /api/v1/recommendation. Se utiliza el comando curl para enviar una petición POST con datos en formato JSON.

Comando	Acción
curl -X POST	Envía una petición HTTP de tipo **POST** (necesaria para el *endpoint* de la API).
-H "Content-Type: application/json"	Indica al servidor que el cuerpo de la petición es JSON.
-d '{"user_id": 101}'	Contiene los datos de prueba (payload) que el modelo espera recibir.

Comando a Ejecutar

Bash

```
# Ejecuta este comando al menos 20 veces para verificar la distribución
curl -X POST \
  http://<DNS_DEL_ALB>/api/v1/recommendation \
  -H "Content-Type: application/json" \
  -d '{"user_id": 101}'
```

Verificación del Éxito

El estudiante debe **ejecutar el comando repetidamente** y verificar las respuestas JSON devueltas:

- **90% de las respuestas:** Devolverán "model_id": "V1 - Stable".
- **10% de las respuestas:** Devolverán "model_id": "V2 - DeepLearning - Canary".

Esto confirma que el ruteo basado en peso (90/10) está operativo y el despliegue canario está activo.

Prueba con Postman

Secuencia de comandos del Data User en instancias EC2

Otra de las cuestiones planteadas tiene que ver con la ejecución del script que ubicamos en el Data User de cada instancia. En este sentido debemos realizar dos cuestiones. Previamente a introducir las instrucciones en el Data User de las instancias EC2, debemos probarlo en alguna instancia para ver que todo funciona correctamente.

Acceso vía SSH desde la red pública

Para ello necesitaremos conectarnos a una instancia EC2 y únicamente las hemos creado en las redes privadas. Debemos usar un servidor de salto o Bastion Host. No es más que una instancia EC2 temporal que crearemos en la red pública solo para permitirnos entrar a través de la red privada vía SSH, como en las otras prácticas.

En la práctica actual, en la red pública sólo tendríamos el Load Balancer, por lo que no podríamos conectarnos desde nuestro ordenador vía SSH si no creamos el servidor de salto.

1. El Rol del Bastion Host

El Bastion Host es una instancia EC2 que actúa como un **puente seguro**. Es la única máquina en la subred pública que se permite acceder a la subred privada.

- **Ubicación:** Subred Pública.
- **Seguridad:** Su Grupo de Seguridad solo permite tráfico **SSH (Puerto 22)** desde **tu dirección IP** (o un rango de IP muy restringido). Esto minimiza el riesgo de exposición.

2. Flujo de Conexión

El proceso de conexión es de dos pasos (el "salto"):

1. **Conexión Directa:** Tú te conectas por SSH desde tu máquina local a la **IP Pública** del Bastion Host.
2. **Conexión de Salto:** Una vez dentro del Bastion Host, utilizas un segundo comando SSH para saltar a la **IP Privada** de tu instancia de *back-end*.

3. Pasos a seguir

1. **Lanzar el Bastion Host:** Lanza una nueva instancia EC2 en la Subred Pública. Asóciale un Grupo de Seguridad que solo permita SSH desde tu IP.
2. **Transferir la Clave:** Sube el archivo de clave privada (mi-clave.pem) a esta nueva instancia Bastion utilizando scp.
3. **Conectar al Bastion:** ssh -i "mi-clave.pem" <usuario>@<IP_PÚBLICA_BASTION>
4. **Conectar al Back-end:** Desde la terminal del Bastion, ejecuta el comando de salto a tu instancia privada: ssh -i "mi-clave.pem" <usuario>@<IP_PRIVADA_BACKEND>

Este método te da acceso completo a tus instancias privadas sin comprometer la seguridad de la red.

Corrección de la secuencia de comandos del Data User

En la secuencia de comandos del script de inicialización de cada instancia (Data User) debemos corregir la parte de configuración de credenciales porque falta el token.

Sustituir el antiguo por el fragmento:

```
# 2. Configurar Credenciales AWS CLI (Obligatorio para autenticar la descarga)
# Reemplaza TUS_CLAVES y tu-region
sudo aws configure set aws_access_key_id TU_CLAVE_DE_ACCESO_ID
sudo aws configure set aws_secret_access_key TU_CLAVE_SECRETA_COMPLETA
sudo aws configure set aws_session_token 'TU_TOKEN_DE_SESION'
sudo aws configure set default.region tu-region
sudo aws configure set default.output json
```

Los datos los obtengo desde la pantalla principal de la consola cuando lanzo el Lab, en la opción AWS Details.

Podéis comprobar su contenido usando la siguiente sentencia:

```
sudo cat /root/.aws/credentials
```