

1. Imagina que tienes una aplicación web de tres capas (frontend, backend y base de datos). Los requisitos indican que: (4 ptos)

- El frontend debe ser accesible desde internet.
- El backend y la base de datos no deben ser accesibles directamente desde internet.
- Los servidores en la subred privada deben poder conectarse a internet para descargar actualizaciones del sistema. Comprueba que realizando un apt-get update puedes actualizar

La VPC funcionará en la subred 15.0.0.0/20 y las subredes 15.0.0.1-2-3/24.

Monta un script usando Python que automatice este proceso.

Deberás enseñar capturas donde muestres:

- **La actualización de la zona privada**
- **Las reglas de seguridad**
- **Las tablas de enrutamiento.**

2. Tienes una VPC con dos subredes: una pública y una privada. Quieres asegurarte de que: (1 pto)

- La subred pública solo permita acceso HTTP/HTTPS desde internet.
- La subred privada solo permita comunicación con la subred pública.

¿Cuál de las siguientes configuraciones describe mejor cómo implementar esto?

1. Configurar un **Network ACL (NACL)** para la subred pública permitiendo solo puertos 80 y 443 desde cualquier IP y denegando todo lo demás. Para la subred privada, permitir comunicación en todos los puertos hacia la subred pública.
2. Configurar un **ACL** en las instancias de la subred pública para restringir tráfico HTTP/HTTPS desde internet. No es necesario configurar NACLs para la subred privada.
3. Configurar **Security Groups** para cada instancia en la subred pública y privada. No es necesario usar NACLs.

4. Configurar un **NACL** que deniegue todo tráfico entrante a la subred pública excepto HTTP/HTTPS y otro que permita todo tráfico interno entre la subred pública y privada.

Explica las ventajas de usar NACLs frente a Security Groups en este escenario y monta el escenario, adjunta las capturas que consideres necesario.

-
3. Tu empresa tiene varias VPC en distintas regiones de AWS y necesita comunicación entre ellas para transferir datos de manera eficiente. Las VPC están configuradas de la siguiente manera: (4 ptos)

- Región 1: Dos VPC en la misma región.
- Región 2: Una VPC que necesita comunicarse con ambas VPC en la Región 1.

¿Cuál sería la mejor solución para conectar estas VPC considerando escalabilidad y eficiencia?

- Crear una **VPC Peering Connection** entre cada par de VPC.
- Implementar un **Transit Gateway** en la Región 1 para conectar las dos VPC locales y usar una conexión directa a la VPC en la Región 2.

Implementa con Python la solución.

- Adjunta capturas:
- Del servicio configurado con sus tablas de enrutamiento.

Provoca un blackhole en alguna de las rutas. (1 pto).