Laboratorio #6

VPC Peering Teórico-Práctico en AWS

Proyecto:
Laboratorios Virtuales de Redes en AWS para el
Fortalecimiento de Competencias en Redes de Nueva Generación

Estudiantes:
Nicolás Carreño Tascón
Juan Manuel Canchala Jiménez

Director:
Carlos Olarte

Asignatura:
Redes de Nueva Generación

Duración Estimada: 90 minutos
Costo: \$0.00 (100 % Gratuito - Free Tier)

Diciembre 2025

Índice

Resumen **3**

1. Objetivos **4**
1.1. Objetivo General . . . 4
1.2. Objetivos Específicos . . . 4
1.3. Competencias a Desarrollar . . . 4

2. Marco Teórico **5**
2.1. Conceptos Fundamentales de VPC Peering . . . 5
2.1.1. Definición de VPC Peering . . . 5
2.1.2. Requisitos Clave . . . 5
2.2. Características Importantes de VPC Peering . . . 5
2.3. Casos de Uso Reales . . . 6
2.4. Limitaciones de VPC Peering . . . 6
2.4.1. No Transitividad . . . 6
2.4.2. Rangos IP no superpuestos . . . 6
2.4.3. Restricciones de Edge-to-Edge . . . 6
2.5. Diseño de Arquitectura con VPC Peering . . . 7
2.5.1. Ejemplo de Arquitectura Lógica . . . 7
2.5.2. Diagrama Conceptual con TikZ . . . 7
2.6. Alternativa: AWS Transit Gateway (Teoría) . . . 7

3. Requisitos Previos **9**
3.1. Conocimientos Necesarios . . . 9
3.2. Recursos Requeridos . . . 9
3.3. Costos Estimados . . . 9
3.4. Tiempo Estimado . . . 9

4. Procedimiento Paso a Paso **11**
4.1. Paso 1: Definir el Escenario y Rangos de Direcciones . . . 11
4.2. Paso 2: Crear la VPC-A (si no existe) . . . 11
4.2.1. Instrucciones . . . 11
4.3. Paso 3: Crear la VPC-B . . . 12
4.3.1. Instrucciones . . . 12
4.4. Paso 4: Lanzar Instancias EC2 de Prueba (Opcional pero Recomendado) . . . 12
4.5. Paso 5: Crear la Conexión de VPC Peering . . . 13
4.5.1. Instrucciones . . . 13
4.6. Paso 6: Aceptar la Conexión de Peering . . . 13
4.7. Paso 7: Actualizar las Tablas de Ruteo . . . 14
4.7.1. Ruteo en VPC-A . . . 14
4.7.2. Ruteo en VPC-B . . . 14
4.8. Paso 8: Ajustar Security Groups para Permitir Tráfico desde la VPC Remota 14
4.8.1. Security Group de Instance-A-Lab6 . . . 14
4.8.2. Security Group de Instance-B-Lab6 . . . 15

4.9. Paso 9: Pruebas de Conectividad (Controladas) 15
4.10. Paso 10: Explorar la No Transitividad (Ejercicio Conceptual) 15

5. Tablas de Configuración **16**
5.1. Resumen de VPCs y Subredes 16
5.2. Tablas de Ruteo 16
5.3. Security Groups 16

6. Verificación del Funcionamiento **17**
6.1. Verificación del Estado de la Conexión de Peering 17
6.2. Verificación de Rutas 17
6.3. Pruebas de Conectividad entre Instancias 17
6.4. Verificación de No Transitividad (Si se creó VPC-C) 17

7. Limpieza de Recursos **18**
7.1. Pasos de Limpieza 18
7.2. Comando CLI Opcional para Ver Instancias 18

8. Cuestionario de Evaluación **19**
8.1. Preguntas de Selección Múltiple 19
8.2. Preguntas Verdadero/Falso 20
8.3. Escenarios Prácticos 21
8.4. Respuestas del Cuestionario 22

9. Conclusiones **23**

10. Referencias **24**
10.1. Documentación Oficial de AWS 24
10.2. Recursos de Arquitectura y Mejores Prácticas 24
10.3. Bibliografía General de Redes y Arquitectura 24

Resumen

Este laboratorio introduce y profundiza en el concepto de **VPC Peering** en Amazon Web Services (AWS), combinando una parte teórica sólida con una práctica guiada detallada. El objetivo es que el estudiante comprenda cómo interconectar redes virtuales privadas (VPCs) de forma segura utilizando peering, conozca sus casos de uso reales, sus **limitaciones (especialmente la no transitividad)** y sea capaz de diseñar y configurar una arquitectura básica con peering entre dos VPCs sin necesidad de transferir grandes volúmenes de datos.

Durante la práctica, se diseña una arquitectura con dos VPCs que poseen rangos de direcciones no superpuestos; se crea una conexión de peering entre ellas, se actualizan las tablas de ruteo y se ajustan reglas de seguridad para permitir comunicación controlada. Además, se analizan escenarios típicos donde el tráfico no funciona por limitaciones del modelo (como la falta de transitividad) y se discuten alternativas de diseño más escalables como **AWS Transit Gateway** (a nivel teórico).

El laboratorio está diseñado para operar dentro del **Free Tier** de AWS: la creación del peering no tiene costo, y las pruebas sugeridas generan volúmenes mínimos de tráfico, por lo que el costo estimado del ejercicio es de $0.00 siempre que se sigan las buenas prácticas de limpieza de recursos.

Palabras clave: VPC Peering, VPC, Routing, AWS Transit Gateway, Redes Privadas, No Transitividad, Arquitectura en la Nube.

1 Objetivos

1.1 Objetivo General

Comprender y aplicar el concepto de **VPC Peering** en AWS mediante el diseño, configuración y verificación de una arquitectura que conecta dos VPCs de forma segura, identificando sus casos de uso, limitaciones y alternativas de diseño.

1.2 Objetivos Específicos

- Definir claramente qué es una conexión de VPC Peering y cómo se integra en la arquitectura de red de AWS.
- Identificar **casos de uso comunes** de VPC Peering en entornos reales (multi-cuenta, multi-ambiente, compartición de servicios).
- Reconocer las **limitaciones** de VPC Peering, especialmente la **no transitividad del ruteo** y las restricciones con direcciones IP superpuestas.
- Diseñar una arquitectura básica con dos VPCs conectadas mediante VPC Peering, con rangos de direcciones no solapados.
- Configurar paso a paso una conexión de VPC Peering, incluyendo la creación/aceptación de la conexión y la actualización de tablas de ruteo y reglas de seguridad.
- Realizar pruebas de conectividad controladas que evidencien el funcionamiento correcto del peering sin generar grandes volúmenes de tráfico.
- Discutir y comparar la alternativa de **AWS Transit Gateway** para escenarios de conectividad más complejos.

1.3 Competencias a Desarrollar

- **Diseño de Redes en la Nube:** Capacidad para diseñar topologías que interconectan múltiples VPCs de forma segura.
- **Gestión de Ruteo:** Habilidad para configurar tablas de ruteo y entender el flujo de tráfico entre redes.
- **Análisis de Limitaciones Arquitectónicas:** Comprender los límites del modelo de peering (no transitivo, no edge-to-edge) y su impacto en el diseño.
- **Evaluación de Alternativas:** Comparar VPC Peering con Transit Gateway en términos de escalabilidad, simplicidad y costo.
- **Buenas Prácticas de Seguridad:** Aplicar el principio de mínimo privilegio y la segmentación adecuada de redes.

2 Marco Teórico

2.1 Conceptos Fundamentales de VPC Peering

2.1.1. Definición de VPC Peering

Un **VPC Peering** es una conexión de red entre dos Virtual Private Clouds (VPCs) que permite enrutar tráfico de forma privada utilizando direcciones IP internas, como si ambas VPCs formaran parte de una misma red lógica. Esta conexión:

- Es **uno a uno** entre dos VPCs.
- Puede ser **intra-región** (dentro de la misma región) o **inter-región** (entre regiones diferentes, en las que el servicio esté soportado).
- Puede ser **intra-cuenta** (misma cuenta AWS) o **cross-account** (distintas cuentas).
- No requiere un gateway central ni dispositivos físicos; se basa en la infraestructura de red de AWS.

Una vez establecido el peering y actualizadas las tablas de ruteo, el tráfico entre las VPCs viaja por la red interna de AWS, sin salir a internet público.

2.1.2. Requisitos Clave

Para crear una conexión de VPC Peering válida se requiere:

- Que los rangos de direcciones IPv4 (y/o IPv6) **no se solapen** entre las VPCs.
- Que las tablas de ruteo de cada VPC se actualicen para enviar tráfico hacia el rango de la otra a través del peering.
- Que las reglas de seguridad (*Security Groups* y, en su caso, NACLs) permitan el tráfico entre las VPCs.

2.2 Características Importantes de VPC Peering

- **No es transitivo:** Si la VPC A está conectada con la VPC B, y B con C, el tráfico de A hacia C **no se enruta automáticamente** a través de B.
- **Sin NAT ni VPN adicionales:** La comunicación se realiza con direcciones IP privadas, sin necesidad de IP públicas ni túneles.
- **Alta disponibilidad:** La conexión de peering se construye sobre la infraestructura redundante de AWS.
- **Facturación:** La **creación y mantenimiento** del peering no tiene costo. Sin embargo, la **transferencia de datos** entre VPCs puede facturarse según la región y el tipo de tráfico (especialmente en peering inter-región).

2.3 Casos de Uso Reales

Algunos escenarios típicos donde se utiliza VPC Peering son:

- **Separación por ambiente:** Una VPC para desarrollo (`Dev-VPC`) y otra para producción (`Prod-VPC`), donde ciertos servicios (por ejemplo, herramientas compartidas) necesitan comunicarse.
- **Arquitectura multi-cuenta:** Cada equipo o unidad de negocio tiene su propia cuenta AWS y VPC aislada; los servicios centrales (logging, monitoreo, directorio) viven en una VPC compartida a la que se hace peering.
- **Compartición de servicios comunes:** Una VPC contiene servicios compartidos como *bastion hosts*, servidores de actualización, repositorios de artefactos, y se hace peering con varias VPCs de aplicación.
- **Migraciones progresivas:** Una VPC antigua y una nueva conviven mientras se migra la carga de trabajo; el peering actúa como puente temporal.

Estos casos ilustran cómo el peering permite **aislar** aplicaciones pero seguir facilitando la conectividad controlada.

2.4 Limitaciones de VPC Peering

2.4.1. No Transitividad

La limitación más relevante es que **VPC Peering no es un mecanismo de ruteo transitivo**. Esto implica:

- Si tienes la VPC A en peering con B, y B en peering con C:
  - A puede hablar con B.
  - B puede hablar con C.
  - A **no puede** hablar con C a través de B utilizando solo peering.
- No es posible usar una VPC como "hub" de ruteo para otras VPCs únicamente con VPC Peering.

2.4.2. Rangos IP no superpuestos

Las VPCs involucradas en un peering deben tener rangos CIDR que **no se solapen**. Si los rangos tienen intersección, la conexión de peering no se puede establecer.

2.4.3. Restricciones de Edge-to-Edge

AWS establece restricciones para evitar que VPC Peering se utilice como un túnel arbitrario entre endpoints:

- No se permite usar VPC Peering para enrutar tráfico desde o hacia:

- Internet Gateway (IGW) de forma transitiva.
- VPNs o Direct Connect de otra VPC.
- Endpoints de VPC en otra VPC (en general, no se permite saltar)

En otras palabras, un VPC Peering es para comunicación **directa** entre las dos VPCs, no para crear un backbone de ruteo global.

2.5 Diseño de Arquitectura con VPC Peering

2.5.1. Ejemplo de Arquitectura Lógica

Consideremos el siguiente escenario:

- **VPC-A (Aplicaciones)**: `10.10.0.0/16`, con subredes públicas y privadas, donde viven los servidores web.
- **VPC-B (Servicios Compartidos)**: `10.20.0.0/16`, donde viven servicios de monitorización, logging o bases de datos compartidas.

Queremos que las instancias en la subred privada de VPC-A puedan comunicarse con un servidor de logging en VPC-B sin exponer nada a internet.

2.5.2. Diagrama Conceptual con TikZ

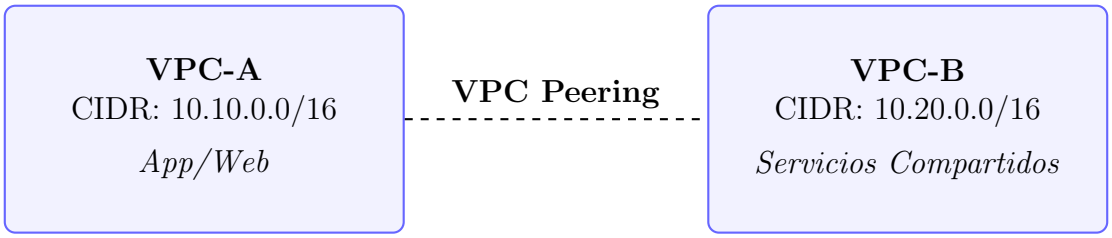

Figura 1: Arquitectura lógica básica con VPC Peering entre dos VPCs

2.6 Alternativa: AWS Transit Gateway (Teoría)

Cuando el número de VPCs empieza a crecer (por ejemplo, decenas o cientos de VPCs en múltiples cuentas y regiones), gestionar conexiones de VPC Peering **en malla completa** se vuelve complejo:

- Para N VPCs, una malla completa requeriría $\frac{N(N-1)}{2}$ conexiones de peering.

AWS Transit Gateway (TGW) es un servicio que actúa como un **hub central** (router gestionado por AWS) al que se adjuntan VPCs, redes on-premise (VPN, Direct Connect) y, opcionalmente, otras Transit Gateways.

Características principales:

- Proporciona **ruteo transitivo**: las VPCs conectadas al TGW pueden comunicarse entre sí siguiendo las tablas de ruteo del TGW.

- Facilita arquitecturas **hub-and-spoke**: un núcleo central y múltiples VPCs conectadas como radios.
- Es un servicio de pago: se cobra por hora de uso y por volumen de datos procesados.

En este laboratorio, Transit Gateway se aborda **solo a nivel conceptual** como alternativa escalable a VPC Peering en topologías grandes.

3 Requisitos Previos

3.1 Conocimientos Necesarios

- Haber completado los laboratorios previos de VPC, subredes, routing e instancias EC2.
- Conocer los conceptos básicos de:
  - Direccionamiento IP y CIDR.
  - Tablas de ruteo (*Route Tables*).
  - Security Groups.
- Manejo básico de la consola de AWS.

3.2 Recursos Requeridos

- **Cuenta AWS** activa y dentro de los límites del Free Tier.
- **Usuario IAM** con permisos administrativos sobre VPC, EC2 y CloudWatch.
- **Región AWS** seleccionada (por ejemplo, `us-east-1` o `sa-east-1`).
- **Infraestructura base mínima:**
  - VPC existente de laboratorios anteriores (`VPC-Principal`) o una nueva creada en este lab.

3.3 Costos Estimados

Concepto	**Costo Estimado**
Creación de VPC Peering	\$0.00
Tráfico entre VPCs (bajo volumen de pruebas)	\$0.00 (Free Tier / costo despreciable)
Instancias EC2 t2.micro/t3.micro de prueba	Incluidas en Free Tier (si se respetan horas)
TOTAL	**\$0.00**

Cuadro 1: Costos del Laboratorio 6

NOTA: Crear la conexión de peering es gratuito. El tráfico de datos entre VPCs puede generar costo dependiendo de la región y del volumen, pero en este laboratorio se realizan solo pruebas mínimas y breves, por lo que el costo esperado es \$0.00.

3.4 Tiempo Estimado

- Lectura del marco teórico: 20 minutos.
- Creación de VPC secundaria y configuración básica: 20 minutos.
- Creación y aceptación de VPC Peering: 15 minutos.

- Actualización de tablas de ruteo y SGs: 20 minutos.
- Verificación, pruebas y limpieza: 15 minutos.
- **TOTAL ESTIMADO:** 90 minutos.

4 Procedimiento Paso a Paso

En este laboratorio crearemos dos VPCs simples y estableceremos una conexión de VPC Peering entre ellas. Luego, configuraremos ruteo y reglas de seguridad para permitir comunicación controlada.

4.1 Paso 1: Definir el Escenario y Rangos de Direcciones

Objetivo: Definir qué VPCs se van a interconectar y con qué rangos CIDR.

Valores Recomendados

- **VPC-A (Aplicaciones):**
  - Nombre: `VPC-A-Lab6`
  - CIDR: `10.10.0.0/16`
  - Subred pública: `10.10.1.0/24`
- **VPC-B (Servicios):**
  - Nombre: `VPC-B-Lab6`
  - CIDR: `10.20.0.0/16`
  - Subred pública: `10.20.1.0/24`

Requisito crítico: Los rangos `10.10.0.0/16` y `10.20.0.0/16` **no se solapan**.

4.2 Paso 2: Crear la VPC-A (si no existe)

Si ya tienes una VPC principal de laboratorios anteriores con un rango similar, puedes usarla. En caso contrario:

4.2.1. Instrucciones

1. En la consola de AWS, ir a **VPC**.
2. Hacer clic en **Your VPCs**.
3. Hacer clic en **Create VPC**.
4. Seleccionar **VPC only**.
5. Completar:
   - **Name tag:** `VPC-A-Lab6`
   - **IPv4 CIDR:** `10.10.0.0/16`
   - Dejar el resto por defecto.
6. Crear la VPC.

7. Crear una subred pública:
   a) Ir a **Subnets** → **Create subnet**.
   b) Seleccionar `VPC-A-Lab6`.
   c) **Name:** `Public-Subnet-A`.
   d) **Availability Zone:** cualquiera.
   e) **CIDR:** `10.10.1.0/24`.
   f) Crear.

4.3 Paso 3: Crear la VPC-B

Objetivo: Crear una segunda VPC independiente que se conectará mediante peering.

4.3.1. Instrucciones

1. En **Your VPCs**, hacer clic en **Create VPC**.
2. Seleccionar **VPC only**.
3. Completar:
   - **Name tag:** `VPC-B-Lab6`
   - **IPv4 CIDR:** `10.20.0.0/16`
4. Crear la VPC.
5. Crear subred pública:
   a) Ir a **Subnets** → **Create subnet**.
   b) VPC: `VPC-B-Lab6`.
   c) **Name:** `Public-Subnet-B`.
   d) **CIDR:** `10.20.1.0/24`.

4.4 Paso 4: Lanzar Instancias EC2 de Prueba (Opcional pero Recomendado)

Objetivo: Tener una instancia de prueba en cada VPC para validar conectividad.

1. Ir al servicio **EC2**.
2. Lanzar una instancia en `VPC-A-Lab6`:
   - Nombre: `Instance-A-Lab6`.
   - AMI: Amazon Linux 2 (Free Tier).
   - Tipo: `t2.micro` o `t3.micro` (Free Tier).
   - Red: `VPC-A-Lab6`.

- Subred: `Public-Subnet-A`.
- Auto-assign Public IP: habilitado (para conectarte desde tu equipo si quieres).
- Security Group: crear uno llamado `SG-A-Lab6`, con:
  - SSH (22) desde tu IP (`My IP`).
  - ICMP (All) desde `10.0.0.0/8` (para pruebas entre VPCs).

3. Repetir para `VPC-B-Lab6`:

- Nombre: `Instance-B-Lab6`.
- Red: `VPC-B-Lab6`.
- Subred: `Public-Subnet-B`.
- Security Group: `SG-B-Lab6`, similar al anterior.

Nota: Estas instancias se usarán solo para **pings y pruebas ligeras**, sin transferencia intensiva de datos.

4.5 Paso 5: Crear la Conexión de VPC Peering

Objetivo: Establecer el peering entre `VPC-A-Lab6` y `VPC-B-Lab6`.

4.5.1. Instrucciones

1. Ir al servicio **VPC**.
2. En el panel izquierdo, hacer clic en **Peering Connections**.
3. Hacer clic en **Create peering connection**.
4. Completar:

- **Name tag:** `Peering-A-B-Lab6`.
- **VPC (requester):** seleccionar `VPC-A-Lab6`.
- **Account:** `My account` (para este laboratorio).
- **Region:** misma región.
- **VPC (accepter):** seleccionar `VPC-B-Lab6`.

5. Hacer clic en **Create peering connection**.

En este punto, la conexión queda en estado `Pending Acceptance`.

4.6 Paso 6: Aceptar la Conexión de Peering

1. En **Peering Connections**, seleccionar `Peering-A-B-Lab6`.
2. En la parte superior, hacer clic en **Actions** → **Accept request**.
3. Confirmar la aceptación.
4. Verificar que el estado cambie a `Active`.

4.7 Paso 7: Actualizar las Tablas de Ruteo

Objetivo: Indicar explícitamente que el tráfico hacia la otra VPC debe salir por la conexión de peering.

4.7.1. Ruteo en VPC-A

1. Ir a **Route Tables**.
2. Filtrar por `VPC-A-Lab6`.
3. Seleccionar la tabla de ruteo asociada a `Public-Subnet-A`.
4. En la pestaña **Routes**, hacer clic en **Edit routes**.
5. Agregar una ruta:
   - **Destination:** `10.20.0.0/16` (rango de `VPC-B-Lab6`).
   - **Target:** seleccionar la conexión de peering `Peering-A-B-Lab6`.
6. Guardar cambios.

4.7.2. Ruteo en VPC-B

1. Repetir el proceso anterior, pero ahora:
   - Filtrar por `VPC-B-Lab6`.
   - Seleccionar la tabla de ruteo de `Public-Subnet-B`.
   - Agregar ruta:
     - **Destination:** `10.10.0.0/16`.
     - **Target:** `Peering-A-B-Lab6`.

4.8 Paso 8: Ajustar Security Groups para Permitir Tráfico desde la VPC Remota

Objetivo: Permitir tráfico ICMP (ping) y, opcionalmente, SSH entre las instancias de ambas VPCs.

4.8.1. Security Group de Instance-A-Lab6

1. Ir a **EC2** → **Security Groups**.
2. Seleccionar `SG-A-Lab6`.
3. En **Inbound rules** → **Edit inbound rules**.
4. Asegurar que existe una regla:
   - **Type:** All ICMP - IPv4.

- **Source:** `10.20.0.0/16`.
- **Description:** `Ping desde VPC-B-Lab6`.

5. Opcionalmente, permitir SSH desde `10.20.0.0/16` (solo para pruebas internas).

4.8.2. Security Group de Instance-B-Lab6

1. Repetir el proceso para `SG-B-Lab6`, agregando:
   - **Type:** All ICMP - IPv4.
   - **Source:** `10.10.0.0/16`.
   - **Description:** `Ping desde VPC-A-Lab6`.

4.9 Paso 9: Pruebas de Conectividad (Controladas)

Objetivo: Validar que la conexión de peering, las rutas y los SGs están configurados correctamente, sin necesidad de transferir grandes volúmenes de datos.

1. Conectarse por SSH a `Instance-A-Lab6` desde tu máquina (usando su IP pública).
2. Dentro de `Instance-A-Lab6`, hacer ping a la IP privada de `Instance-B-Lab6`, por ejemplo:

```
ping 10.20.1.10
```

Listing 1: Ping desde VPC-A hacia VPC-B

3. Deberías recibir respuestas ICMP exitosas (tiempos de respuesta bajos).
4. Repetir desde `Instance-B-Lab6` hacia la IP privada de `Instance-A-Lab6`.
5. Estas pruebas son de tráfico mínimo (ICMP), suficientes para validar la conectividad.

4.10 Paso 10: Explorar la No Transitividad (Ejercicio Conceptual)

Opcional (Teórico/Práctico): Si el tiempo lo permite, se puede crear una tercera VPC `VPC-C-Lab6` y establecer peering con `VPC-B-Lab6`, comprobando que:

- A puede hablar con B.
- B puede hablar con C.
- A **no** puede hablar con C a través de B usando solo peering.

El estudiante debe verificar que, aunque se intenten agregar rutas, AWS no permite usar peering como enlace transitivo.

5 Tablas de Configuración

5.1 Resumen de VPCs y Subredes

Recurso	Nombre	CIDR	Descripción
VPC	VPC-A-Lab6	10.10.0.0/16	VPC de aplicaciones
Subred	Public-Subnet-A	10.10.1.0/24	Subred pública en VPC-A
VPC	VPC-B-Lab6	10.20.0.0/16	VPC de servicios
Subred	Public-Subnet-B	10.20.1.0/24	Subred pública en VPC-B

Cuadro 2: VPCs y subredes utilizadas en el laboratorio

5.2 Tablas de Ruteo

Tabla de Ruteo	Destino (CIDR)	Target
RTB-A	10.10.0.0/16	Local
RTB-A	10.20.0.0/16	Peering-A-B-Lab6
RTB-B	10.20.0.0/16	Local
RTB-B	10.10.0.0/16	Peering-A-B-Lab6

Cuadro 3: Rutas configuradas en las tablas de ruteo

5.3 Security Groups

SG	Tipo	Reglas Inbound Principales
SG-A-Lab6	Instancia A	SSH (22) desde My IP; ICMP desde 10.20.0.0/16
SG-B-Lab6	Instancia B	SSH (22) desde My IP; ICMP desde 10.10.0.0/16

Cuadro 4: Reglas de Security Groups para pruebas de peering

6 Verificación del Funcionamiento

6.1 Verificación del Estado de la Conexión de Peering

1. En el servicio **VPC**, ir a **Peering Connections**.
2. Verificar que `Peering-A-B-Lab6` aparece con estado `Active`.
3. Confirmar que las VPCs asociadas son `VPC-A-Lab6` y `VPC-B-Lab6`.

6.2 Verificación de Rutas

1. En **Route Tables**, seleccionar la tabla de ruteo de `Public-Subnet-A`.
2. Verificar que existe una ruta hacia `10.20.0.0/16` apuntando a `Peering-A-B-Lab6`.
3. Repetir para `Public-Subnet-B`, confirmando la ruta hacia `10.10.0.0/16`.

6.3 Pruebas de Conectividad entre Instancias

1. Conectarse por SSH a `Instance-A-Lab6`.
2. Ejecutar:

```
ping 10.20.1.X
```

Listing 2: Prueba de ping hacia la instancia en VPC-B

3. Verificar que se reciben respuestas (TTL y tiempo).
4. Hacer la prueba inversa desde `Instance-B-Lab6`.

6.4 Verificación de No Transitividad (Si se creó VPC-C)

Si se creó una tercera VPC `VPC-C-Lab6`, con peering solo hacia `VPC-B-Lab6`, el estudiante puede comprobar que:

- Aún agregando rutas, AWS no permite usar una conexión de peering como tránsito para otra VPC.
- El tráfico de `VPC-A-Lab6` hacia `VPC-C-Lab6` no funcionará mediante peering doble.

7 Limpieza de Recursos

Objetivo: Evitar costos innecesarios y mantener el entorno ordenado.

7.1 Pasos de Limpieza

1. **Instancias EC2**
   - Detener y terminar `Instance-A-Lab6` y `Instance-B-Lab6` si no se reutilizarán.
2. **Peering Connection**
   - Ir a **Peering Connections**.
   - Seleccionar `Peering-A-B-Lab6`.
   - **Actions** → **Delete peering connection**.
3. **VPC-B-Lab6 (si solo se usa en este lab)**
   - Eliminar recursos asociados (subredes, tablas de ruteo personalizadas, gateways).
   - Eliminar la VPC desde **Your VPCs**.
4. **VPC-A-Lab6**
   - Si es una VPC creada solo para este lab y no se reutilizará, eliminar recursos y luego eliminar la VPC.

7.2 Comando CLI Opcional para Ver Instancias

```
aws ec2 describe-instances \
  --query 'Reservations[*].Instances[*].[InstanceId,State.Name,
     PrivateIpAddress,Tags]' \
  --output table
```

Listing 3: Listar instancias EC2 y sus estados

8 Cuestionario de Evaluación

Instrucciones: Contesta las siguientes preguntas. Las respuestas sugeridas se encuentran al final del cuestionario.

8.1 Preguntas de Selección Múltiple

1. **¿Qué es una conexión de VPC Peering en AWS?**

   a) Un túnel VPN entre una VPC y un data center on-premise.

   b) Un enlace privado entre dos VPCs que permite tráfico usando direcciones IP privadas.

   c) Un gateway público para exponer una VPC a internet.

   d) Un servicio administrado para enrutar tráfico entre regiones y on-premise.

2. **¿Cuál de las siguientes afirmaciones sobre VPC Peering es correcta?**

   a) Es transitivo: si A se conecta con B y B con C, A puede contactar C.

   b) Requiere que las VPCs tengan rangos de IP superpuestos.

   c) No es transitivo y requiere rangos de IP no superpuestos.

   d) Solo funciona entre VPCs en diferentes cuentas.

3. **¿Cuándo tiene sentido usar VPC Peering?**

   a) Para conectar dos VPCs que requieren comunicación privada directa.

   b) Para conectar cientos de VPCs y actuar como router central.

   c) Para exponer una VPC entera a internet.

   d) Para reemplazar completamente el uso de Security Groups.

4. **¿Qué se necesita para que dos instancias en VPCs conectadas por peering se comuniquen?**

   a) Solo crear la conexión de peering; no se requiere nada más.

   b) Crear la conexión de peering y actualizar tablas de ruteo y reglas de seguridad.

   c) Deshabilitar todas las reglas de seguridad en ambas VPCs.

   d) Asignar IPs públicas a todas las instancias.

5. **En una conexión de VPC Peering, la creación de la conexión es:**

   a) Siempre de pago, independiente del tráfico.

   b) Gratuita; el costo se asocia principalmente al tráfico de datos.

   c) Gratuita solo si las VPCs están en distintas regiones.

   d) Gratuita solo si las VPCs están en distintas cuentas.

6. **¿Cuál de las siguientes es una limitación de VPC Peering?**

a) No permite comunicación entre VPCs en la misma región.

b) Solo permite tráfico HTTP.

c) No soporta ruteo transitivo.

d) Solo permite tráfico unidireccional.

7. **¿Qué alternativa ofrece AWS para escenarios con muchas VPCs que necesitan conectividad transitiva?**

a) Amazon S3.

b) AWS Transit Gateway.

c) Amazon CloudFront.

d) AWS Lambda.

8. **¿Cuál es un caso de uso típico de VPC Peering?**

a) Conectar una VPC con internet público.

b) Conectar una VPC de producción con una VPC de servicios compartidos.

c) Conectar varias VPCs para formar un gran backbone global transitivo.

d) Reemplazar todas las VPN site-to-site.

9. **En el contexto de seguridad, usar VPC Peering permite:**

a) Evitar el uso de Security Groups.

b) Seguir aplicando el principio de mínimo privilegio a nivel de SG y rutas entre VPCs.

c) Obligar a que todo tráfico pase por internet público.

d) Deshabilitar el uso de NACLs en ambas VPCs.

10. **Si dos VPCs tienen rangos `10.0.0.0/16` y `10.0.1.0/24`, ¿pueden hacer peering?**

a) Sí, porque son rangos completamente distintos.

b) Sí, pero solo si están en diferentes regiones.

c) No, porque los rangos se superponen.

d) Solo si el tráfico es únicamente ICMP.

8.2 Preguntas Verdadero/Falso

VF1. En una conexión de VPC Peering, el tráfico entre VPCs siempre viaja por internet público.

VF2. VPC Peering puede ser utilizado entre VPCs en la misma cuenta o entre cuentas diferentes.

VF3. Las tablas de ruteo de cada VPC deben actualizarse explícitamente para utilizar la conexión de peering.

8.3 Escenarios Prácticos

E1. Escenario 1: Entorno multi-ambiente

Tienes dos VPCs en la misma cuenta:

- `VPC-Dev`: ambiente de desarrollo.
- `VPC-Prod`: ambiente de producción.

El equipo de Dev necesita acceder a un servicio compartido (por ejemplo, un repositorio interno) que vive en `VPC-Prod`, pero no debe acceder libremente a todas las instancias de producción.

Pregunta: ¿Cómo diseñarías el VPC Peering y las reglas de ruteo/seguridad para permitir solo el acceso necesario al servicio compartido, aplicando mínimo privilegio?

E2. Escenario 2: Crecimiento de VPCs

Una organización empezó con una arquitectura pequeña y usó VPC Peering para conectar 3 VPCs. Ahora planea tener 20 VPCs en distintas cuentas y quiere conectividad flexible entre muchas de ellas.

Pregunta: ¿Por qué VPC Peering puede empezar a ser problemático en este escenario? ¿Por qué AWS Transit Gateway podría ser una mejor opción a mediano y largo plazo?

8.4 Respuestas del Cuestionario

Selección Múltiple

1. **b)** Un enlace privado entre dos VPCs que permite tráfico usando direcciones IP privadas.
2. **c)** No es transitivo y requiere rangos de IP no superpuestos.
3. **a)** Para conectar dos VPCs que requieren comunicación privada directa.
4. **b)** Crear la conexión de peering y actualizar tablas de ruteo y reglas de seguridad.
5. **b)** Gratuita; el costo se asocia principalmente al tráfico de datos.
6. **c)** No soporta ruteo transitivo.
7. **b)** AWS Transit Gateway.
8. **b)** Conectar una VPC de producción con una VPC de servicios compartidos.
9. **b)** Seguir aplicando el principio de mínimo privilegio a nivel de SG y rutas entre VPCs.
10. **c)** No, porque los rangos se superponen.

Verdadero/Falso

1. **Falso.** El tráfico viaja por la red interna de AWS, no por internet público.
2. **Verdadero.** VPC Peering soporta escenarios intra-cuenta y cross-account.
3. **Verdadero.** Sin actualización de tablas de ruteo, el peering no se usa para enrutar tráfico.

Guía para Escenarios

Escenario 1:

- Establecer peering entre `VPC-Dev` y `VPC-Prod`.
- En `VPC-Prod`, aislar el servicio compartido en subred y SG específicos.
- Configurar rutas que permitan tráfico solo hacia el rango del servicio compartido.
- Ajustar Security Groups para permitir tráfico desde rangos/SGs de `VPC-Dev` únicamente hacia el puerto/servicio necesario.

Escenario 2:

- Con 20 VPCs, una malla completa de VPC Peering requiere muchas conexiones y mantenimiento complejo.
- No se obtiene ruteo transitivo, lo que complica aún más los diseños.
- Un Transit Gateway actúa como hub, permitiendo ruteo centralizado, simplificando la gestión y haciendo más escalable la topología.

9 Conclusiones

En este laboratorio se ha explorado de forma teórico-práctica el uso de **VPC Peering** como mecanismo fundamental para interconectar redes privadas en AWS. A través de la construcción de un escenario con dos VPCs y la configuración detallada de peering, ruteo y seguridad, el estudiante ha podido observar cómo se habilita comunicación privada entre dominios de red aislados sin exponer recursos a internet.

Se ha destacado la importancia de:

- Utilizar rangos de direcciones **no superpuestos**.
- Actualizar adecuadamente las **tablas de ruteo** en ambos lados.
- Ajustar **Security Groups** para permitir solo el tráfico estrictamente necesario, alineado con el **principio de mínimo privilegio**.

Asimismo, se han analizado las **limitaciones** del modelo, en particular la **no transitividad** y las restricciones de edge-to-edge. Esto es clave para evitar diseños incorrectos donde se pretendan usar VPCs como routers genéricos. Finalmente, se introdujo **AWS Transit Gateway** como solución más adecuada cuando el número de VPCs y la complejidad de la topología crecen, resaltando la necesidad de elegir la herramienta correcta según la escala del problema.

Este laboratorio no solo fortalece conocimientos técnicos sobre VPC Peering, sino que también refuerza la capacidad de razonar sobre arquitectura de redes en la nube, sopesar alternativas y aplicar buenas prácticas de seguridad y diseño.

10 Referencias

10.1 Documentación Oficial de AWS

1. **VPC Peering**
   - VPC Peering Guide
     https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html
   - VPC Peering Limitations
     https://docs.aws.amazon.com/vpc/latest/peering/vpc-peering-basics.html
2. **Amazon VPC**
   - Amazon VPC Documentation
     https://docs.aws.amazon.com/vpc/
   - IP Addressing in Your VPC
     https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html
3. **AWS Transit Gateway**
   - AWS Transit Gateway Documentation
     https://docs.aws.amazon.com/transitgateway/
   - Transit Gateway Peering and Routing
     https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html
4. **Security Groups y Routing**
   - Amazon EC2 Security Groups for Linux Instances
     https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
   - Route Tables
     https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Route_Tables.html

10.2 Recursos de Arquitectura y Mejores Prácticas

1. AWS Architecture Center
   https://aws.amazon.com/architecture/
2. AWS Well-Architected Framework - Security Pillar
   https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html

10.3 Bibliografía General de Redes y Arquitectura

1. Tanenbaum, A. S., & Wetherall, D. (2011). *Redes de Computadoras*. Pearson.
2. White, T. (2015). *Networking for Systems Administrators*. No Starch Press.

Nota: Las URLs fueron verificadas al momento de elaboración de este laboratorio. Se recomienda consultar la documentación oficial de AWS para obtener actualizaciones y detalles adicionales.