Laboratorio #8

Proyecto Integrador - Arquitectura Completa 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: 180 minutos
Costo: \$0.00 (100 % Gratuito - Free Tier)

Diciembre 2025

Índice

Resumen

Este laboratorio constituye el **proyecto integrador final** de la serie de laboratorios de redes en AWS. El objetivo es diseñar e implementar una **arquitectura corporativa completa** en la nube, integrando de forma coherente todos los conceptos trabajados en los laboratorios anteriores: creación de cuenta e IAM (Lab 1), diseño de VPC y subredes (Lab 2), conectividad a internet con Internet Gateway (Lab 3), instancias EC2 y seguridad de red (Lab 4), seguridad avanzada y defensa en profundidad (Lab 5), VPC Peering (Lab 6) y monitoreo con CloudWatch y VPC Flow Logs (Lab 7).

El estudiante diseñará una red corporativa simplificada para una pequeña-mediana empresa, que incluya una VPC principal con subredes públicas y privadas distribuidas en múltiples zonas de disponibilidad, instancias EC2 para capa web y de administración, grupos de seguridad multicapa, NACLs, conectividad a internet mediante IGW, monitoreo con CloudWatch y VPC Flow Logs, y un esquema teórico de alta disponibilidad y escalabilidad. Se documentarán explícitamente las **decisiones arquitectónicas**, se realizará un **análisis de costos y optimización dentro del Free Tier** y se presentará la solución de forma profesional, como si se tratara de una propuesta para un cliente real.

La implementación práctica se limitará a un subconjunto mínimo de recursos para mantener el costo en $0.00 (Free Tier), mientras que la arquitectura completa se describirá a nivel de diseño lógico y buenas prácticas.

Palabras clave: AWS, Arquitectura de Red, VPC, EC2, Security Groups, VPC Peering, CloudWatch, VPC Flow Logs, Alta Disponibilidad, Free Tier.

1 Objetivos

1.1 Objetivo General

Diseñar y documentar una **arquitectura corporativa completa en AWS**, integrando todos los componentes de red trabajados en los laboratorios 1–7, junto con una implementación básica que se mantenga dentro del Free Tier y refleje buenas prácticas de seguridad, escalabilidad y monitoreo.

1.2 Objetivos Específicos

- Integrar VPC, subredes públicas y privadas, Internet Gateway, EC2, Security Groups, VPC Peering y CloudWatch en una arquitectura coherente.
- Documentar de forma explícita las **decisiones arquitectónicas** tomadas (rangos de direcciones, distribución por AZ, separación por capas, patrones de seguridad, etc.).
- Diseñar un **escenario de alta disponibilidad y escalabilidad** (a nivel teórico) basado en el uso de múltiples zonas de disponibilidad y balanceo de carga.
- Realizar un **análisis de costos** de la arquitectura propuesta, identificando qué se mantiene en Free Tier y qué componentes tendrían costo en producción.
- Implementar de forma práctica una versión mínima de la arquitectura (al menos una VPC, subred pública y privada, 1–2 instancias EC2 y seguridad básica) y verificar su funcionamiento.
- Aplicar mecanismos de **monitoreo y registro** (VPC Flow Logs, métricas y alarmas de CloudWatch) sobre la arquitectura básica implementada.
- Presentar la solución final con un nivel de detalle y formalidad similar al de un **diseño profesional** de red corporativa en la nube.

1.3 Competencias a Desarrollar

- **Diseño arquitectónico en la nube:** Capacidad para diseñar topologías de red completas en AWS considerando seguridad, disponibilidad y monitoreo.
- **Toma de decisiones técnicas:** Selección y justificación de componentes de AWS adecuados a los requisitos funcionales y no funcionales.
- **Análisis de costo-beneficio:** Evaluación de alternativas técnicas a la luz de su impacto económico (Free Tier vs producción).
- **Documentación profesional:** Elaboración de documentos de diseño claros, estructurados y orientados a clientes o equipos técnicos.
- **Operación y monitoreo:** Configuración de registros, métricas y alarmas para tener visibilidad del comportamiento de la red y los servicios.

2 Marco Teórico

2.1 Arquitectura de Red Corporativa en AWS

Una **red corporativa en AWS** suele estructurarse siguiendo patrones de arquitectura bien conocidos:

- **Separación por capas (n-tier):**
  - Capa de presentación (web).
  - Capa de lógica de negocio (aplicación).
  - Capa de datos (bases de datos).
- **Separación por dominios de seguridad:**
  - Subredes públicas: recursos expuestos a internet (por ejemplo, balanceadores y bastions).
  - Subredes privadas: recursos internos (aplicaciones, bases de datos).
- **Distribución en múltiples zonas de disponibilidad (AZs):**
  - Subredes replicadas en al menos 2 AZs para alta disponibilidad.

La arquitectura integradora de este laboratorio se basará en una **VPC principal corporativa** con:

- CIDR base `10.0.0.0/16`.
- Subredes públicas y privadas en al menos dos AZs.
- Instancias EC2 en capa de presentación y administración.
- Reglas de seguridad multicapa (Security Groups + NACLs).
- Integración con CloudWatch para monitoreo.

2.2 Resumen Integrado de Conceptos Anteriores

2.2.1. Cuenta, IAM y MFA (Lab 1)

- **Cuenta AWS:** es el contenedor administrativo de todos los recursos.
- **IAM (Identity and Access Management):** controla quién puede hacer qué.
- **MFA:** protege la cuenta root y usuarios privilegiados.
- **Alertas de facturación:** sirven para evitar costos inesperados.

En el proyecto integrador, se asume:

- Usuario IAM administrativo con MFA habilitado.
- Alertas de facturación activas para mantener el costo en Free Tier.

2.2.2. VPC, Subredes y Rutas (Lab 2 y Lab 3)

- **VPC:** red virtual aislada con un rango CIDR.
- **Subredes:** divisiones lógicas del rango CIDR principal.
- **Internet Gateway (IGW):** proporciona acceso a internet desde y hacia la VPC.
- **Tablas de ruteo:** definen por dónde debe ir el tráfico.

La arquitectura integradora usará:

- Una VPC principal con cidr `10.0.0.0/16`.
- Subredes públicas (por ejemplo, `10.0.1.0/24` y `10.0.2.0/24`).
- Subredes privadas (por ejemplo, `10.0.11.0/24` y `10.0.12.0/24`).
- IGW para salida a internet desde subredes públicas.

2.2.3. Instancias EC2 y Seguridad de Red (Lab 4 y Lab 5)

- **EC2:** máquinas virtuales en la nube.
- **Security Groups (SG):** firewalls *stateful* a nivel de instancia.
- **Network ACLs (NACLs):** listas de control de acceso *stateless* a nivel de subred.
- **Defense in depth:** múltiples capas de seguridad (SG + NACL + IAM + monitoreo).
- **Principio de mínimo privilegio:** solo permitir el tráfico estricto necesario.

En el proyecto integrador se diseñarán:

- Un SG para servidores web.
- Un SG para instancias privadas (aplicación / base de datos).
- Un SG para bastion host (administración).
- NACLs que refuercen las políticas de tráfico por subred.

2.2.4. VPC Peering (Lab 6)

- Permite conectar dos VPCs de forma privada usando la red de AWS.
- **No es transitivo:** si VPC A está conectada con B, y B con C, A no ve a C por defecto.
- Se usa para escenarios multi-VPC (por ejemplo, entorno *producción* y *gestión*).

En este laboratorio, el **peering se considerará a nivel de diseño** para un escenario donde la empresa tenga:

- Una VPC principal de producción.
- Una VPC de herramientas o administración conectada por peering (teórico).

2.2.5. Monitoreo y VPC Flow Logs (Lab 7)

- **CloudWatch:** métrica, logs, alarmas y dashboards.
- **VPC Flow Logs:** registro de tráfico ACCEPT/REJECT.
- **CloudWatch Logs Insights:** consultas sobre logs.
- **Alarmas:** condiciones que generan notificaciones.

En la arquitectura integradora:

- Se habilitarán Flow Logs sobre la VPC o subredes principales.
- Se configurarán métricas derivadas (por ejemplo, conexiones REJECT).
- Se integrarán alarmas clave en un dashboard de monitoreo de red.

2.3 Diseño de Alta Disponibilidad y Escalabilidad (Teórico)

Aunque la implementación práctica se limitará a pocos recursos por Free Tier, el diseño considerará:

- **Alta disponibilidad (HA):**
  - Subredes replicadas en al menos dos AZs.
  - Posible uso de balanceadores de carga (teórico) para distribuir tráfico.
- **Escalabilidad:**
  - Escalado horizontal (más instancias EC2 detrás de un Load Balancer).
  - Auto Scaling Groups (ASG) a nivel teórico.
- **Resiliencia:**
  - Diseñar para que la falla de una AZ no implique caída total.
  - Uso de servicios gestionados (por ejemplo, RDS Multi-AZ en un escenario real).

2.4 Análisis de Costos y Optimización en Free Tier

Un elemento clave en la arquitectura es diferenciar entre:

- **Recursos efectivamente usados en el laboratorio:**
  - 1–2 instancias EC2 t2.micro/t3.micro (bajo límite de horas).
  - VPC, subredes, IGW, NACLs, SGs (sin costo).
  - VPC Flow Logs y CloudWatch con uso moderado.
- **Recursos que quedarían solo a nivel de diseño teórico:**
  - Balanceador de carga (ALB).

- Auto Scaling Groups.
- NAT Gateway (tiene costo, se deja solo como teoría).
- Transit Gateway para multi-VPC complejas.

El diseño debe **separar claramente** lo que se implementa en la práctica sin costo y lo que sería parte de una versión "full" en producción con presupuesto asignado.

3 Requisitos Previos

3.1 Conocimientos Necesarios

- Haber completado (o entendido) los laboratorios 1–7.
- Comprender:
  - Conceptos de VPC, subredes, tablas de ruteo, IGW.
  - Configuración básica de EC2 y acceso SSH.
  - Conceptos de Security Groups y NACLs.
  - Fundamentos de VPC Peering.
  - Uso básico de CloudWatch, VPC Flow Logs y Logs Insights.

3.2 Recursos Técnicos

- Cuenta AWS activa con Free Tier.
- Usuario IAM con permisos administrativos (o al menos sobre VPC, EC2, CloudWatch, logs, IAM de lectura).
- Navegador web moderno.
- Cliente SSH para conectarse a las instancias (PuTTY, OpenSSH, etc.).

3.3 Costos Estimados

Concepto	**Costo Estimado**
1–2 instancias EC2 t2.micro/t3.micro	\$0.00 (dentro de 750 horas/mes Free Tier)
VPC, subredes, IGW, tablas de ruteo	\$0.00
Security Groups y NACLs	\$0.00
VPC Peering (sin tráfico de datos)	\$0.00
CloudWatch métricas básicas y hasta 10 alarmas	\$0.00 (Free Tier)
VPC Flow Logs (bajo volumen, uso limitado)	\$0.00 – costo muy bajo controlado
TOTAL (Laboratorio)	**\$0.00**

Cuadro 1: Costos del Laboratorio 8 en modo práctico

3.4 Tiempo Estimado

- Diseño conceptual de la arquitectura: 45 minutos.
- Implementación básica (VPC, subredes, IGW, EC2, SGs): 60–70 minutos.
- Configuración de Flow Logs, métricas y dashboard: 40–45 minutos.
- Verificación, limpieza y documentación final: 25–30 minutos.
- **TOTAL ESTIMADO:** 180 minutos.

4 Procedimiento Paso a Paso

4.1 Visión General del Proyecto

Antes de iniciar, se define el **escenario corporativo**:

- Empresa ficticia: **AcmeCorp S.A.S.**
- Necesidades:
  - Exponer un sitio web corporativo.
  - Tener una capa de aplicación y datos protegida en subredes privadas.
  - Administrar los servidores de forma segura mediante un bastion host.
  - Monitorear el tráfico de red y detectar anomalías.

Arquitectura Lógica (Descripción)

- VPC principal `AcmeCorp-VPC` (`10.0.0.0/16`).
- Subredes públicas en dos AZs:
  - `10.0.1.0/24` (Public-AZ1).
  - `10.0.2.0/24` (Public-AZ2).
- Subredes privadas en dos AZs:
  - `10.0.11.0/24` (Private-AZ1).
  - `10.0.12.0/24` (Private-AZ2).
- IGW para salida/entrada a internet desde subredes públicas.
- 1 servidor web en subred pública (implementación práctica).
- 1 bastion host en subred pública para administración SSH.
- 1 servidor de aplicación (o datos) en subred privada.
- Security Groups específicos para cada rol.
- VPC Flow Logs habilitados para la VPC.
- Métricas y alarmas en CloudWatch.

4.2 Paso 1: Diseño de Direccionamiento y Subredes

Objetivo: Definir el plan de direccionamiento IP interno.

1. Elegir región (por ejemplo, `us-east-1` o `sa-east-1`).
2. Definir CIDR de la VPC: `10.0.0.0/16`.
3. Plan de subredes:
   - Subred pública AZ1: `10.0.1.0/24`.
   - Subred pública AZ2: `10.0.2.0/24`.
   - Subred privada AZ1: `10.0.11.0/24`.
   - Subred privada AZ2: `10.0.12.0/24`.
4. Reservar espacio extra para posibles subredes futuras (por ejemplo, `10.0.21.0/24` para administración).

4.3 Paso 2: Creación de la VPC y Subredes

Objetivo: Implementar el diseño de VPC y subredes.

1. Ir a **VPC** → **Your VPCs** → **Create VPC**.
2. Parámetros sugeridos:
   - **Name**: `AcmeCorp-VPC`.
   - **IPv4 CIDR**: `10.0.0.0/16`.
   - Opciones adicionales: por defecto.
3. Crear subredes:
   - **Subnet 1 (publica-AZ1)**:
     - Name: `AcmeCorp-Public-AZ1`.
     - VPC: `AcmeCorp-VPC`.
     - AZ: seleccionar una (ej. `us-east-1a`).
     - CIDR: `10.0.1.0/24`.
   - **Subnet 2 (publica-AZ2)**:
     - Name: `AcmeCorp-Public-AZ2`.
     - AZ diferente (ej. `us-east-1b`).
     - CIDR: `10.0.2.0/24`.
   - **Subnet 3 (privada-AZ1)**:
     - Name: `AcmeCorp-Private-AZ1`.
     - CIDR: `10.0.11.0/24`.
   - **Subnet 4 (privada-AZ2)**:
     - Name: `AcmeCorp-Private-AZ2`.
     - CIDR: `10.0.12.0/24`.
4. Marcar las subredes públicas para asignación automática de IPs públicas (opcional).

4.4 Paso 3: Internet Gateway y Tablas de Ruteo

Objetivo: Proveer conectividad a internet a las subredes públicas.

1. Crear un Internet Gateway:
   - En **VPC** → **Internet Gateways** → **Create internet gateway**.
   - Name: `AcmeCorp-IGW`.
   - Crear y luego **Attach to VPC** → seleccionar `AcmeCorp-VPC`.
2. Crear tabla de ruteo pública:
   - **Route Tables** → **Create route table**.
   - Name: `AcmeCorp-Public-RT`.
   - VPC: `AcmeCorp-VPC`.
3. Agregar ruta por defecto a internet:
   - Editar rutas.
   - Destino: `0.0.0.0/0`.
   - Target: `AcmeCorp-IGW`.
4. Asociar subredes públicas a esta tabla:
   - Subredes: `AcmeCorp-Public-AZ1` y `AcmeCorp-Public-AZ2`.
5. Dejar la tabla de ruteo por defecto para subredes privadas (sin ruta a internet).

4.5 Paso 4: Security Groups y NACLs (Defensa en Profundidad)

Objetivo: Definir políticas de seguridad multicapa.

Security Groups

1. SG para servidor web:
   - Name: `SG-Web`.
   - Inbound:
     - HTTP (80) desde `0.0.0.0/0` (para pruebas).
     - SSH (22) **solo** desde IP pública del bastion (luego de creado) o desde la IP pública del administrador.
   - Outbound: `0.0.0.0/0` (por simplicidad en el lab).
2. SG para bastion host:
   - Name: `SG-Bastion`.
   - Inbound:

- SSH (22) solo desde la IP pública del estudiante/administrador.
- Outbound:
  - SSH (22) hacia subredes privadas (instancias internas).

3. SG para servidor interno (aplicación/datos):
   - Name: `SG-App`.
   - Inbound:
     - SSH (22) solo desde `SG-Bastion`.
     - Puerto de aplicación (ej. 8080) solo desde `SG-Web` (teórico).
   - Outbound: `0.0.0.0/0` o restringido según necesidad.

NACLs (Opcional para refuerzo)

1. NACL pública:
   - Allow inbound HTTP(80)/SSH(22) desde internet (según sea necesario).
   - Allow outbound respuestas establecidas.
2. NACL privada:
   - Restringir el tráfico entrante a puertos específicos desde subredes públicas.

4.6 Paso 5: Lanzar Instancias EC2 (Implementación Básica)

Objetivo: Desplegar recursos mínimos de cómputo para probar la arquitectura.

1. **Servidor web:**
   - Tipo: `t2.micro` o `t3.micro` (Free Tier).
   - AMI: Amazon Linux 2 u otra Free Tier.
   - Subred: `AcmeCorp-Public-AZ1`.
   - Auto-assign Public IP: habilitado.
   - SG: `SG-Web`.
   - User data opcional: instalar un servidor web simple (ejemplo con `httpd`).
2. **Bastion host:**
   - Tipo: `t2.micro/t3.micro`.
   - Subred: `AcmeCorp-Public-AZ1` (o AZ2).
   - IP pública asignada.
   - SG: `SG-Bastion`.
3. **Servidor interno:**

- Tipo: `t2.micro/t3.micro`.
- Subred: `AcmeCorp-Private-AZ1`.
- Sin IP pública.
- SG: `SG-App`.

4.7 Paso 6: Pruebas de Conectividad y Seguridad

1. Desde el navegador del estudiante, acceder a la IP pública del servidor web vía HTTP.
2. Conectarse por SSH al bastion host usando la clave privada.
3. Desde el bastion host, hacer SSH al servidor interno (private IP).
4. Verificar que no se pueda acceder directamente al servidor interno desde internet.

4.8 Paso 7: Habilitar VPC Flow Logs y Monitoreo

Objetivo: Integrar monitoreo en la arquitectura.

1. Crear log group en CloudWatch: `/aws/vpc/flow-logs/acmecorp`.
2. En VPC → **Your VPCs** → `AcmeCorp-VPC` → pestaña **Flow Logs**.
3. Crear flow log con:
   - Filter: `ALL`.
   - Destination: CloudWatch Logs.
   - Log group: `/aws/vpc/flow-logs/acmecorp`.
   - Interval: 1 minuto.
4. Generar tráfico de prueba (HTTP permitido, puertos bloqueados).
5. Verificar logs en CloudWatch.

4.9 Paso 8: Métricas Derivadas y Alarmas

1. Crear un metric filter para `REJECT` similar al Lab 7:
   - Log group: `/aws/vpc/flow-logs/acmecorp`.
   - Filter pattern: `REJECT`.
   - Namespace: `AcmeCorp/Network`.
   - Metric name: `RejectedConnections`.
2. Crear una alarma `AcmeCorp-HighRejectedConnections` sobre esta métrica.
3. Opcional: crear alarma sobre `NetworkIn` del servidor web para detectar picos de tráfico.

4.10 Paso 9: Dashboard de Arquitectura y Monitoreo

1. Crear dashboard `AcmeCorp-Network-Dashboard`.
2. Agregar widgets:
   - Gráfico de `NetworkIn/Out` del servidor web.
   - Gráfico de `RejectedConnections`.
   - Widget de estado de alarmas.
   - Opcional: un widget de texto con un diagrama ASCII simplificado de la arquitectura.

4.11 Paso 10: Diseño Teórico de Escalabilidad y Alta Disponibilidad

Este paso es **solo teórico** (no implementar por costo):

- Proponer el uso de:
  - Application Load Balancer (ALB) frente a una flota de servidores web.
  - Auto Scaling Group con mínimo 2 instancias en 2 AZs.
  - NAT Gateway (o NAT instance) para permitir salida a internet desde subredes privadas (teniendo en cuenta el costo).
  - Posible segunda VPC (administración) conectada por VPC Peering.
- Documentar cómo cambiaría la arquitectura básica para llegar a una solución de alta disponibilidad en producción.

5 Tablas de Configuración

5.1 Plan de Direccionamiento

Recurso	Nombre	CIDR
VPC	AcmeCorp-VPC	10.0.0.0/16
Subred pública AZ1	AcmeCorp-Public-AZ1	10.0.1.0/24
Subred pública AZ2	AcmeCorp-Public-AZ2	10.0.2.0/24
Subred privada AZ1	AcmeCorp-Private-AZ1	10.0.11.0/24
Subred privada AZ2	AcmeCorp-Private-AZ2	10.0.12.0/24

Cuadro 2: Plan de direccionamiento de la arquitectura integradora

5.2 Resumen de Componentes Clave

Componente	Nombre	Descripción
VPC	AcmeCorp-VPC	Red corporativa principal
IGW	AcmeCorp-IGW	Conectividad a internet
RT Pública	AcmeCorp-Public-RT	Tabla de rutas para subredes públicas
SG	SG-Web	Seguridad para servidor web
SG	SG-Bastion	Seguridad para bastion host
SG	SG-App	Seguridad para servidor interno
Flow Logs	AcmeCorp-VPC Flow Log	Registros de tráfico de la VPC
Log Group	/aws/vpc/flow-logs/acmecorp	Almacenamiento de VPC Flow Logs
Namespace	AcmeCorp/Network	Métricas personalizadas de red
Métrica	RejectedConnections	Conteo de conexiones REJECT
Alarma	AcmeCorp-HighRejectedConnections	Alerta por tráfico rechazado elevado
Dashboard	AcmeCorp-Network-Dashboard	Panel de monitoreo integrado

Cuadro 3: Componentes principales del proyecto integrador

6 Verificación

6.1 Pruebas de Conectividad

- **Prueba 1: Acceso web**
  - Desde el navegador local, acceder a `http://IP-publica-servidor-web`.
  - Verificar que responde la página por defecto o el contenido configurado.
- **Prueba 2: Acceso SSH seguro**
  - Conectarse por SSH al bastion host usando clave privada.
  - Desde el bastion, conectarse por SSH a la IP privada del servidor interno.
- **Prueba 3: Aislamiento del servidor interno**
  - Intentar conectarse directamente desde internet al servidor interno (debe fallar).
  - Confirmar que solo es accesible a través del bastion.

6.2 Verificación de Flow Logs y Métricas

1. Verificar que el Flow Log de la VPC está en estado **Active**.
2. Revisar el log group `/aws/vpc/flow-logs/acmecorp` para ver registros recientes.
3. Confirmar que la métrica `RejectedConnections` se está actualizando en `AcmeCorp/Network`.

6.3 Verificación de Alarmas y Dashboard

1. Verificar en **CloudWatch Alarms** el estado de `AcmeCorp-HighRejectedConnections`.
2. Forzar algunos intentos de conexión bloqueados (por ejemplo, puertos cerrados) y observar si la métrica aumenta.
3. Revisar el dashboard `AcmeCorp-Network-Dashboard` y comprobar que:
   - Se muestra el tráfico de red del servidor web.
   - Se visualiza la curva de `RejectedConnections`.
   - Se ve el widget con el estado de alarmas.

7 Limpieza de Recursos

Objetivo: Dejar la cuenta libre de recursos que puedan generar costos más allá del laboratorio.

1. **Instancias EC2:**
   - Detener y terminar el servidor web, bastion host y servidor interno si no se van a seguir usando.
2. **Flow Logs:**
   - En la VPC, eliminar el Flow Log configurado para detener la generación de logs.
3. **CloudWatch Logs:**
   - Eliminar el log group `/aws/vpc/flow-logs/acmecorp` si ya no es necesario.
4. **Alarmas y métricas:**
   - Eliminar la alarma `AcmeCorp-HighRejectedConnections`.
   - (Opcional) Eliminar el metric filter asociado.
5. **Dashboard:**
   - Eliminar el dashboard `AcmeCorp-Network-Dashboard` si no se usará más.
6. **VPC y redes:**
   - Eliminar subredes, tablas de ruteo personalizadas, IGW y finalmente la VPC `AcmeCorp-VPC` si fue creada exclusivamente para el laboratorio.

8 Cuestionario de Evaluación

8.1 Preguntas de Selección Múltiple

1. **¿Cuál es el principal objetivo del proyecto integrador del Lab 8?**

   a) Probar únicamente el rendimiento de instancias EC2.

   b) Diseñar una arquitectura corporativa completa integrando los conceptos de los labs 1–7.

   c) Configurar NAT Gateway en producción.

   d) Migrar una base de datos on-premise a AWS RDS.

2. **En la arquitectura propuesta, las subredes privadas se utilizan principalmente para:**

   a) Alojar recursos expuestos directamente a internet.

   b) Almacenar objetos S3 con acceso público.

   c) Instancias internas de aplicación y bases de datos sin IP pública.

   d) Consolas de administración web públicas.

3. **¿Cuál de los siguientes componentes NO genera costo directo en el Free Tier (uso razonable)?**

   a) 1–2 instancias EC2 t2.micro/t3.micro dentro de las 750 horas/mes.

   b) VPC, subredes, tablas de ruteo e IGW.

   c) NAT Gateway con tráfico de salida a internet.

   d) Security Groups y NACLs.

4. **En el diseño de defensa en profundidad, los Security Groups:**

   a) Son *stateless* y se aplican a nivel de subred.

   b) Son *stateful* y se aplican a nivel de instancia.

   c) Solo controlan tráfico saliente, no entrante.

   d) Reemplazan completamente la necesidad de NACLs.

5. **Un bastion host se utiliza principalmente para:**

   a) Servir contenido web público a los clientes.

   b) Proporcionar un punto de acceso SSH seguro a instancias en subredes privadas.

   c) Actuar como NAT Gateway gestionado.

   d) Funcionar como base de datos central.

6. **¿Cuál es la ventaja de distribuir subredes en múltiples zonas de disponibilidad (AZs)?**

a) Reducir el costo de instancias EC2.

b) Incrementar la capacidad de almacenamiento de S3.

c) Mejorar la alta disponibilidad y tolerancia a fallos.

d) Eliminar la necesidad de balanceadores de carga.

7. **VPC Peering en el contexto del proyecto integrador se utiliza (a nivel teórico) para:**

a) Conectar dos VPCs de forma privada usando la red de AWS.

b) Conectar directamente una VPC con internet.

c) Reemplazar un NAT Gateway.

d) Crear una base de datos replicada entre regiones.

8. **¿Cuál de las siguientes afirmaciones sobre VPC Flow Logs es correcta?**

a) Solo registran tráfico aceptado (ACCEPT).

b) Pueden enviar logs a CloudWatch Logs o a S3.

c) Solo funcionan en subredes públicas.

d) Solo registran tráfico de salida a internet.

9. **En el análisis de costos, es importante:**

a) Ignorar el Free Tier y asumir siempre el peor escenario.

b) Conocer qué servicios tienen ofertas gratuitas y cuáles generan cargos desde el inicio.

c) Usar todos los servicios posibles sin importar el costo.

d) Evitar el uso de CloudWatch para reducir costos.

10. **Un CloudWatch Dashboard bien diseñado para esta arquitectura debe incluir, como mínimo:**

a) Solo gráficos de CPU de instancias internas.

b) Gráficos de tráfico de red, métricas de conexiones rechazadas y estado de alarmas relevantes.

c) Únicamente un gráfico con el costo mensual estimado.

d) Un listado de usuarios IAM sin métricas.

8.2 Preguntas Verdadero/Falso

VF1. En el proyecto integrador, se puede lograr una implementación básica completamente dentro del Free Tier si se controla el número de instancias y la duración de uso.

VF2. El diseño teórico de alta disponibilidad con balanceadores, Auto Scaling y NAT Gateway no debe implementarse en este laboratorio para evitar costos, pero sí debe documentarse.

VF3. Una buena práctica es permitir acceso SSH a todas las instancias directamente desde internet para simplificar la administración.

8.3 Escenarios Prácticos

E1. Escenario 1: Cambio de requisitos de seguridad

AcmeCorp decide que el servidor web no debe permitir acceso SSH desde ninguna IP pública, y que toda administración debe hacerse únicamente desde el bastion host.

Pregunta: ¿Qué cambios realizarías en los Security Groups y flujos de acceso para cumplir este requisito sin romper la arquitectura?

E2. Escenario 2: Crecimiento de la empresa

La empresa crece y ahora requiere que la aplicación web soporte el doble de usuarios, manteniendo alta disponibilidad. El presupuesto de producción permite agregar algunos servicios de pago.

Pregunta: ¿Qué modificaciones propondrías a la arquitectura actual (componentes adicionales y cambios) para soportar esta nueva carga, manteniendo buenas prácticas de seguridad y monitoreo?

8.4 Respuestas del Cuestionario

Selección Múltiple

1. **b)** Diseñar una arquitectura corporativa completa integrando los conceptos de los labs 1–7.
2. **c)** Instancias internas de aplicación y bases de datos sin IP pública.
3. **c)** NAT Gateway con tráfico de salida a internet (tiene costo).
4. **b)** Son *stateful* y se aplican a nivel de instancia.
5. **b)** Proporcionar un punto de acceso SSH seguro a instancias en subredes privadas.
6. **c)** Mejorar la alta disponibilidad y tolerancia a fallos.
7. **a)** Conectar dos VPCs de forma privada usando la red de AWS.
8. **b)** Pueden enviar logs a CloudWatch Logs o a S3.
9. **b)** Conocer qué servicios tienen ofertas gratuitas y cuáles generan cargos desde el inicio.
10. **b)** Gráficos de tráfico de red, métricas de conexiones rechazadas y estado de alarmas relevantes.

Verdadero/Falso

1. **Verdadero.** Controlando el número de instancias y el tiempo de ejecución, es posible mantenerse en el Free Tier.
2. **Verdadero.** Se recomienda documentar el diseño completo pero no implementarlo para evitar cargos.
3. **Falso.** La buena práctica es limitar el acceso SSH mediante bastion host y rangos de IP confiables.

Guía para Escenarios

Escenario 1:

- Modificar `SG-Web` para:
  - Eliminar cualquier regla de inbound SSH (22) desde internet.
  - Permitir únicamente HTTP/HTTPS desde `0.0.0.0/0`.
- Asegurar que `SG-Bastion` pueda acceder por SSH al servidor web:
  - Agregar en `SG-Web` una regla SSH con origen = `SG-Bastion`.
- Flujo de acceso:
  - Administrador → bastion (SSH).

- Bastion → servidor web (SSH) y servidor interno.

Escenario 2:

- Propuestas:
  - Agregar un Application Load Balancer frente a la capa web.
  - Crear un Auto Scaling Group con mínimo 2 instancias web en subredes públicas (AZ1 y AZ2).
  - Usar NAT Gateway para permitir salida segura a internet desde subredes privadas (actualizaciones, dependencias).
  - Considerar RDS Multi-AZ para la capa de datos.
  - Extender el monitoreo:
    - Métricas de ALB (RequestCount, errores 4xx/5xx).
    - Alarmas sobre tiempos de respuesta y tasa de errores.
    - Dashboards adicionales para la capa de aplicación y base de datos.
- Mantener buenas prácticas:
  - SGs y NACLs bien definidos.
  - IAM con mínimo privilegio.
  - MFA en cuentas administrativas.

9 Conclusiones

El Laboratorio 8 representa la culminación de la serie de prácticas de redes en AWS, integrando todos los conceptos abordados en los laboratorios previos en una **arquitectura corporativa coherente, segura y monitoreada**. A través del diseño y la implementación básica de la VPC `AcmeCorp-VPC`, subredes públicas y privadas, instancias EC2, Security Groups multicapa, VPC Flow Logs y CloudWatch, el estudiante ha experimentado el ciclo completo de construcción de una red en la nube, desde la planificación hasta la observabilidad.

Los principales aprendizajes incluyen:

- La importancia de un **buen diseño de direccionamiento IP** y separación lógica en subredes públicas y privadas.
- La aplicación práctica del **principio de mínimo privilegio** y la **defensa en profundidad**, combinando Security Groups, NACLs e IAM.
- La relevancia de contar con **mecanismos de administración seguros**, como el uso de un bastion host para acceder a recursos privados.
- La necesidad de **monitorear el tráfico de red** mediante VPC Flow Logs, métricas derivadas y alarmas para detectar comportamientos anómalos.
- La capacidad de diferenciar entre **implementaciones sin costo (Free Tier)** y **diseños de producción** que incluyen componentes adicionales como ALB, NAT Gateway, Auto Scaling y RDS.

Este proyecto integrador ha demostrado que es posible construir, con recursos mínimos, una arquitectura que ya refleja las buenas prácticas de la industria en términos de seguridad, disponibilidad, escalabilidad y observabilidad. En un entorno real de producción, esta base podría evolucionar hacia una solución aún más robusta incorporando servicios gestionados, automatización y despliegues continuos, sin perder los principios fundamentales aprendidos.

En síntesis, el estudiante no solo ha aprendido a **usar servicios individuales de AWS**, sino a **pensar en arquitecturas completas**, evaluando decisiones técnicas, costos, riesgos y mecanismos de operación, capacidades esenciales para cualquier profesional que diseñe redes y sistemas en la nube.

10 Referencias

10.1 Documentación Oficial de AWS

1. AWS Documentation - Página principal
https://docs.aws.amazon.com/

2. Amazon VPC User Guide
https://docs.aws.amazon.com/vpc/latest/userguide/

3. Amazon EC2 User Guide
https://docs.aws.amazon.com/ec2/

4. Amazon CloudWatch Documentation
https://docs.aws.amazon.com/cloudwatch/

5. VPC Flow Logs
https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html

6. AWS Identity and Access Management (IAM)
https://docs.aws.amazon.com/IAM/latest/UserGuide/

7. AWS Free Tier
https://aws.amazon.com/free/

10.2 AWS Well-Architected Framework

1. AWS Well-Architected Framework
https://aws.amazon.com/architecture/well-architected/

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

3. Reliability Pillar - AWS Well-Architected Framework
https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html

10.3 Bibliografía Recomendada

1. Wittig, A., & Wittig, M. (2018). *Amazon Web Services in Action* (2nd ed.). Manning Publications.

2. Erl, T., Mahmood, Z., & Puttini, R. (2013). *Cloud Computing: Concepts, Technology & Architecture*. Prentice Hall.

3. NIST Special Publication 800-145. (2011). *The NIST Definition of Cloud Computing*.
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

Nota: Las URLs de la documentación oficial de AWS fueron verificadas al momento de la elaboración de este laboratorio. Dado que AWS actualiza constantemente sus servicios y guías, se recomienda consultar siempre la versión más reciente en el sitio oficial.