Servicios de Red Avanzados y y Conectividad para IA

De la VPC al Edge: Construyendo redes redes seguras y de alto rendimiento rendimiento para cargas de trabajo de trabajo de IA

Dr. [Nombre del Profesor]

Universidad [Nombre]

Cloud Computing para Inteligencia Artificial - Sesión 7

Agenda de la Sesión

01

Fundamentos de la Nube Privada Virtual

Profundización en VPC/VNet

02

Aislamiento y Seguridad en la Red

Security Groups vs. Network ACLs

03

Conectividad Privada y Escalable

Endpoints, Peering y Transit Gateway

04

El Mundo Híbrido

Conectando la Nube con On-Premise (VPN y Conexión Dedicada)

Dedicada)

05

Aceleración y Alcance Global

CDNs y DNS Inteligente

06

Defensa Activa

Protección contra ataques de red (DDoS y WAF)

Revisión y Profundización en VPC

Concepto Central

Una **Virtual Private Cloud (VPC)** es tu porción de red privada, lógicamente aislada, dentro de la nube de de la nube de un proveedor. Es la base sobre la que construirás toda tu arquitectura.

Componentes Clave

- **Región:** La VPC vive en una única región
- **Bloque CIDR:** Define el rango de direcciones IP privadas (ej. 10.0.0.0/16)
- La elección del CIDR es **crítica y casi inmutable**

Agnosticismo

- **AWS:** Virtual Private Cloud (VPC)
- **Azure:** Virtual Network (VNet)
- **GCP:** VPC Network

Analogía: Piensa en la VPC como el plano de tu propia oficina en un oficina en un rascacielos gigante (la nube pública). Tú decides decides dónde van los muros, las puertas y quién tiene acceso. acceso.

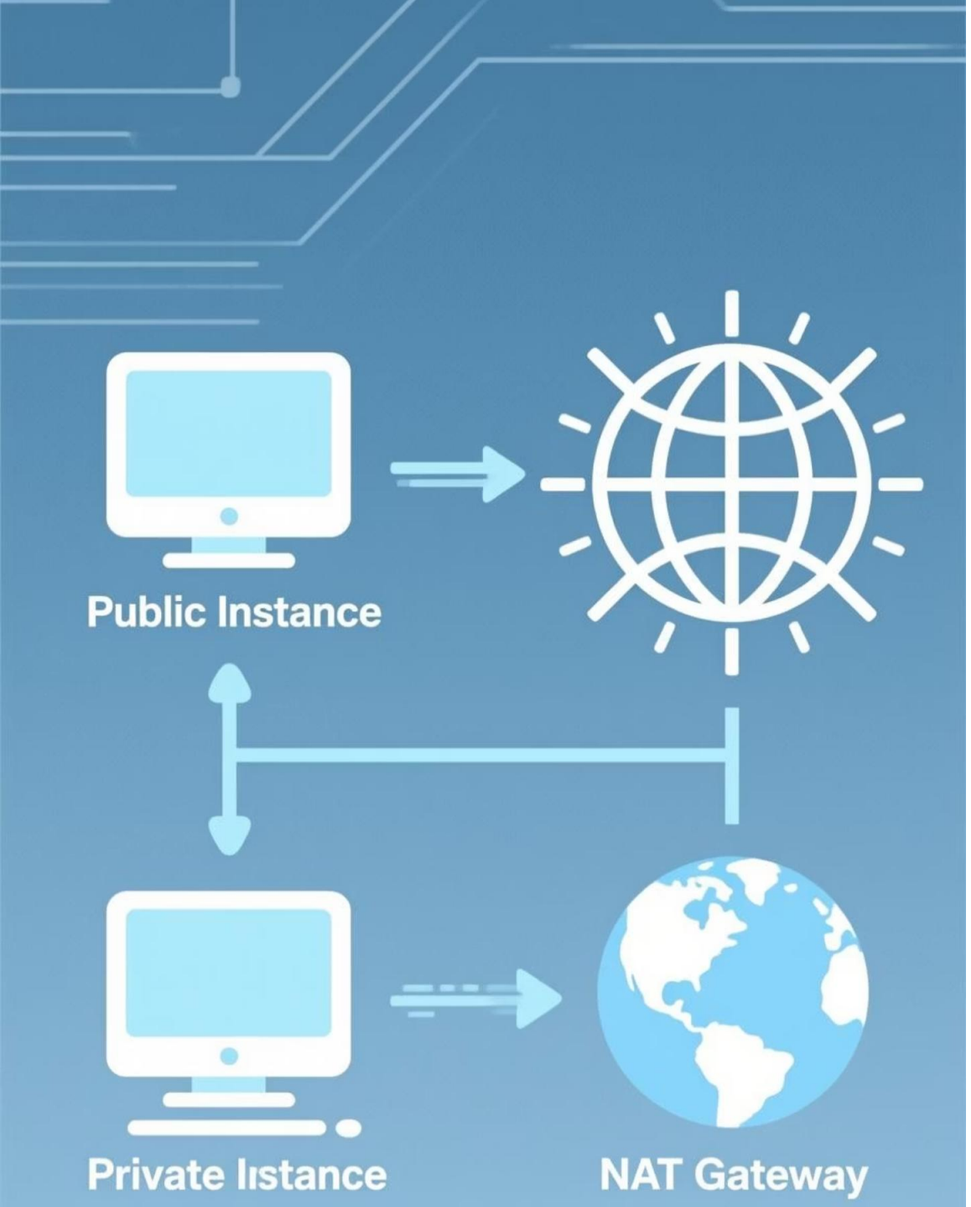
Diseño de VPC: Subredes Públicas vs. Privadas

Subred Pública

- **Definición:** Una subred cuya tabla de rutas tiene una ruta directa a un **Internet Gateway (IGW)**
- **Componentes típicos:** Balanceadores, servidores web, bastiones bastiones
- **Caso de uso IA:** El endpoint de un API Gateway que expone un modelo de inferencia

Subred Privada

- **Definición:** No tiene ruta directa a internet. El acceso a internet se realiza a través de un **NAT Gateway**
- **Componentes típicos:** Servidores de aplicación, bases de datos, clústeres de cómputo
- **Caso de uso IA:** Clúster de instancias EC2 con GPUs procesando datasets procesando datasets sensibles



Gateways: El Acceso al Exterior

Internet Gateway (IGW)

Función: Componente gestionado, redundante y altamente disponible que permite la comunicación entre instancias en tu VPC e internet.

Operación: Se asocia a la VPC y se añade una ruta en la tabla de rutas de la subred pública (0.0.0.0/0 -> igw-id).

NAT Gateway

Función: Permite que las instancias en una subred privada inicien tráfico hacia internet, pero impide que internet inicie conexiones con esas instancias.

Servicio Gestionado: Altamente disponible, escalable bajo demanda. **Recomendado para producción.**

NAT Instance (Legacy): No recomendada: punto único de fallo, requiere gestión manual.

Primera Capa de Defensa: Security Groups (SG)

Concepto

Un firewall virtual **stateful** que actúa a nivel de la interfaz de red elástica (ENI) de una instancia.

Stateful (Con estado)

Si permites el tráfico saliente por un puerto, el tráfico de respuesta es **automáticamente automáticamente permitido** en el camino de vuelta.

Reglas

- Solo se pueden crear reglas de **permiso (allow)**
- Todo el tráfico que no coincide con una regla está **implícitamente denegado**
- Las reglas se evalúan en conjunto, el orden no importa

Agnosticismo

AWS: Security Group, **Azure:** Network Security Group (NSG), **GCP:** VPC Firewall Rules

Caso de uso IA: Un SG para tus servidores web permite entrada por entrada por TCP 80/443 desde 0.0.0.0/0. Un SG para tu base de datos de datos RDS solo permite entrada por TCP 3306 desde el SG de los de los servidores web.

Segunda Capa de Defensa: Network ACLs (NACL)

Concepto

Un firewall virtual **stateless** que actúa como una capa de defensa opcional a nivel de **subred**.

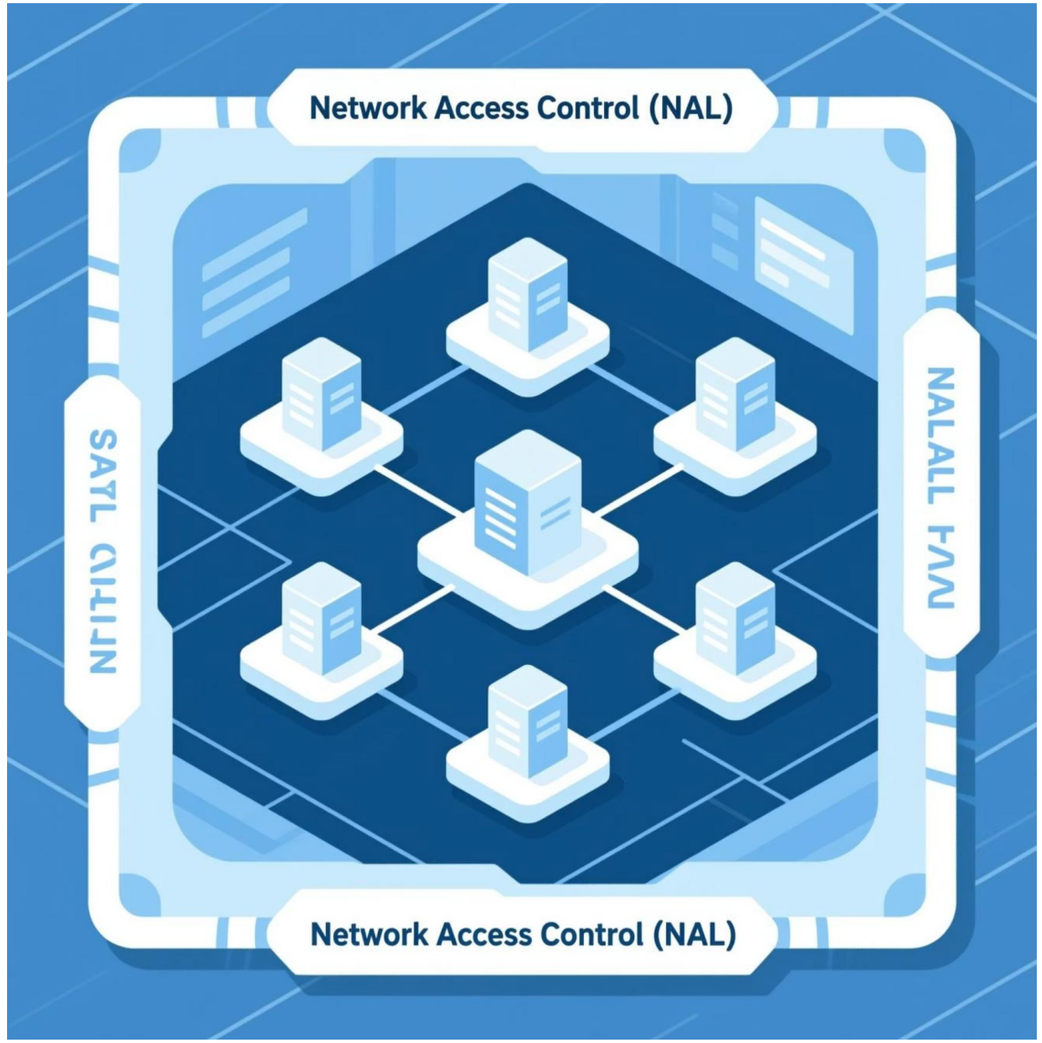
Stateless (Sin estado)

Si permites tráfico entrante por el puerto TCP 443, debes **explícitamente permitir** el tráfico de respuesta saliente en los puertos efímeros (TCP 1024-65535).

Reglas

- Se pueden crear reglas de **permiso (allow)** y de **denegación (deny)**
- Las reglas se evalúan por **orden numérica**, desde el número más bajo al más alto
- Existe una regla final implícita (*) que deniega todo el tráfico

Caso de uso IA: Usar una NACL para bloquear explícitamente una dirección dirección IP maliciosa conocida (1.2.3.4/32) a nivel de toda la subred. subred.
Agnosticismo: Este concepto es más específico de **AWS**. En Azure y GCP se logra con funcionalidades similares pero con otros nombres.



Comparativa: SG vs. NACL

Característica	Security Group (SG)	Network ACL (NACL)
Ámbito	A nivel de Interfaz de Red (Instancia)	A nivel de Subred
Estado	**Stateful** (Con estado)	**Stateless** (Sin estado)
Reglas Permitidas	Solo Allow	Allow y Deny
Evaluación	Todas las reglas se evalúan	Por orden numérico
Uso Principal	"Guardia de seguridad personal" para cada instancia	"Valla perimetral" de la subred

Conclusión Clave: No es uno u otro. La mejor práctica es usar **ambos**. Las NACLs para defensa perimetral amplia y los SGs para un control de acceso granular y específico entre recursos.

¡Elige la Arquitectura Correcta!

Escenario

Tienes una aplicación de IA de 3 capas: un front-end web (React), un API de inferencia (Flask/FastAPI) en contenedores ECS, y una base de datos PostgreSQL en RDS.

Opción A

Un único SG para todos los recursos (Web, API, BBDD) que permite:

- TCP 443 desde Internet
- TCP 5000 (API) desde Internet
- TCP 5432 (RDS) desde Internet

Opción B

- **SG-WEB:** Permite TCP 443 desde 0.0.0.0/0
- **SG-API:** Permite TCP 5000 solo desde la referencia del SG-WEB
- **SG-RDS:** Permite TCP 5432 solo desde la referencia del SG-API

Pregunta: ¿Qué opción sigue el **Principio de Mínimo Privilegio** y es la más segura?

Evitando Internet: VPC Endpoints

Problema

Por defecto, para que una instancia en una subred privada acceda a un servicio de AWS como S3, el tráfico debe ir a través del NAT Gateway, salir a internet, y volver a entrar a la red de AWS. Esto es ineficiente y puede generar costes.

Solución: VPC Endpoints

Permiten conectar tu VPC a servicios de AWS **sin necesidad de un IGW o un NAT Gateway**, manteniendo todo el tráfico dentro de la red privada y segura de AWS.

Ventajas

- **Seguridad Mejorada:** El tráfico no atraviesa la internet pública
- **Fiabilidad:** Menos dependencia de componentes externos
- **Coste:** Ahorro en costes de procesamiento de datos del NAT Gateway

Tipos de VPC Endpoints (AWS)

Gateway Endpoints

Funcionamiento: Son una "puerta de enlace" que se especifica como un destino en tu destino en tu tabla de rutas.

Servicios Soportados: Solo para **Amazon S3** y **DynamoDB**.

Implementación: Se añade una entrada en la tabla de rutas de la subred privada. privada.

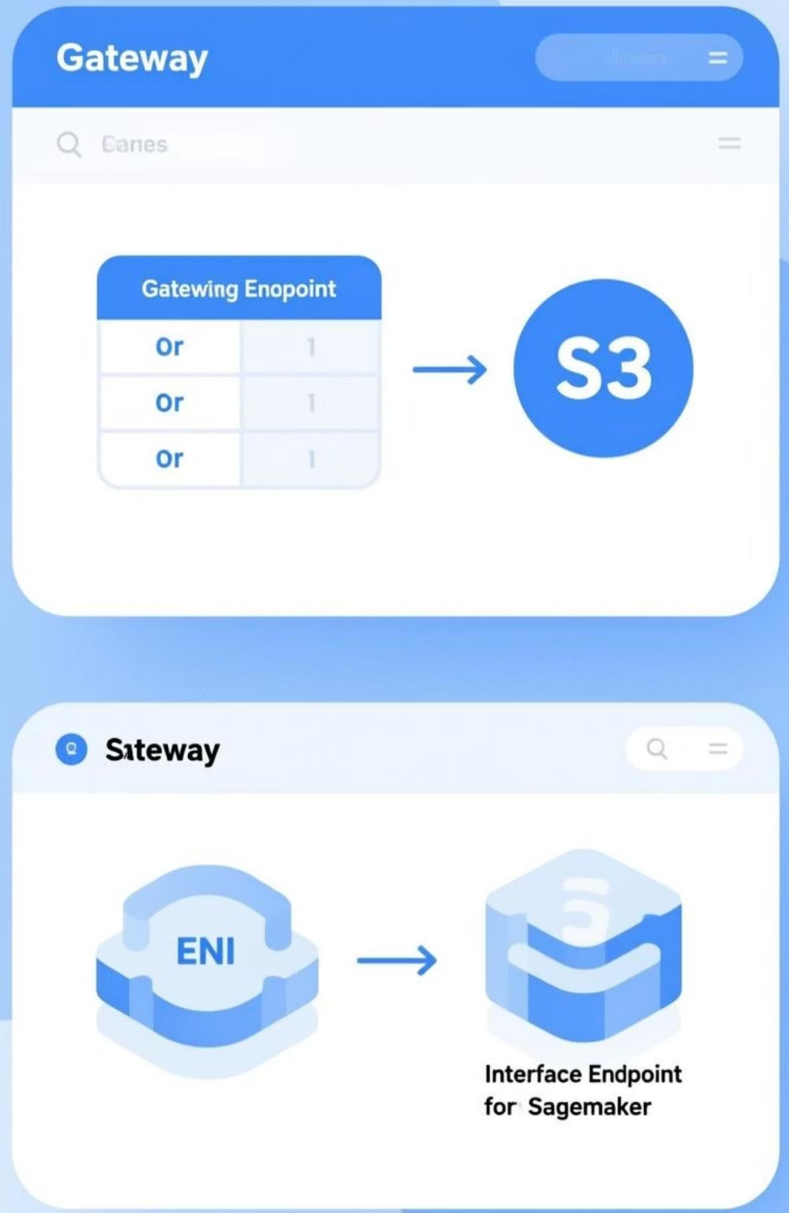
Coste: No tienen coste adicional.

Interface Endpoints (AWS PrivateLink)

Funcionamiento: Crean una Interfaz de Red Elástica (ENI) con una IP privada dentro de tu subred.

Servicios Soportados: La mayoría de los servicios de AWS (SQS, Kinesis, SageMaker, API SageMaker, API Gateway, etc.)

Coste: Tienen un coste por hora y por GB de datos procesados.



Conectando VPCs: VPC Peering

Concepto

Una conexión de red entre dos VPCs que permite que los recursos en ambas se comuniquen entre sí utilizando direcciones IP privadas, como si estuvieran en la misma red.

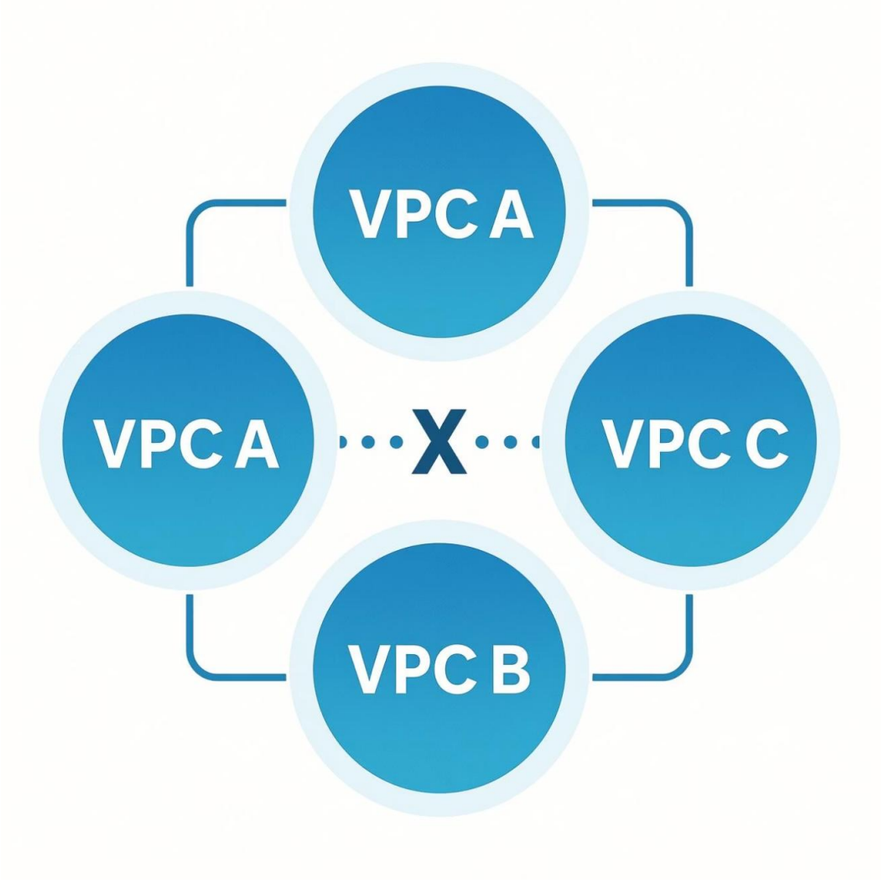
Características Clave

- **No Transitivo:** Si la VPC A está conectada a la B, y la B a la C, la VPC A **no puede** comunicarse comunicarse con la C a través de la B
- **Limitación Crítica:** Los bloques CIDR de las VPCs conectadas **no pueden solaparse**

- Funciona entre regiones y entre cuentas de AWS

Caso de Uso IA

Un equipo de Data Science tiene su VPC con herramientas de análisis y necesita acceder a los acceder a los datos de producción que están en la VPC de la aplicación principal.



Agnosticismo: AWS/GCP: VPC Peering, **Azure:** VNet Peering. Peering.

La Evolución del Peering: Transit Gateway

Problema

A medida que creces, gestionar docenas de conexiones de peering ("full mesh") se vuelve una pesadilla.

Solución: AWS Transit Gateway (TGW)

Actúa como un **hub centralizado (hub-and-spoke)** que simplifica la interconexión de VPCs y conexiones on-premise.

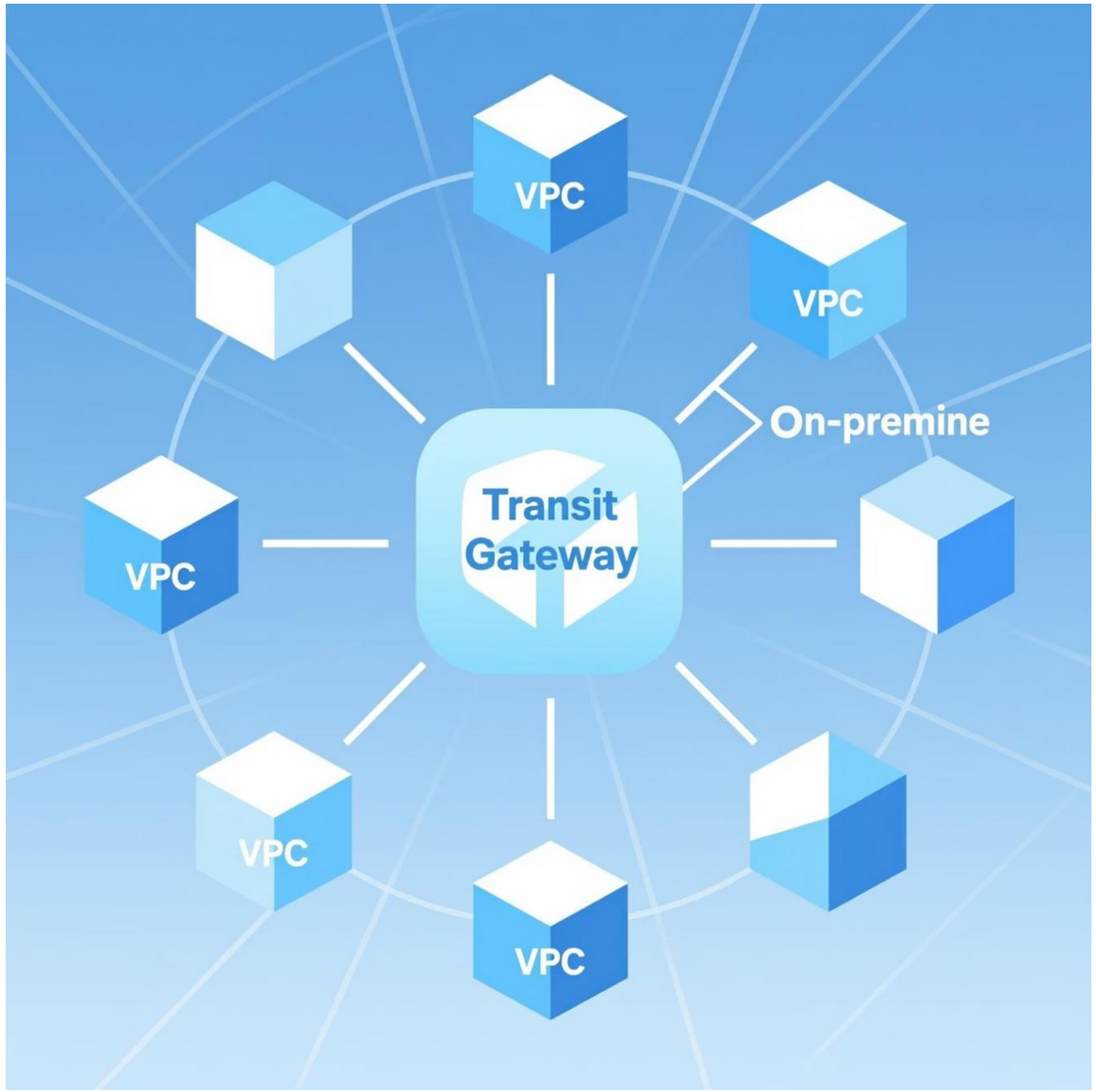
Funcionamiento

Conectas cada VPC ("spoke") al TGW ("hub"). El TGW gestiona el enrutamiento entre todos los spokes.

Ventajas

- **Enrutamiento Transitivo:** ¡Resuelve el principal problema del peering!
- **Gestión Simplificada:** Un único punto para gestionar la conectividad
- **Escalabilidad:** Conecta miles de VPCs

Agnosticismo: AWS: Transit Gateway, **Azure:** Azure Virtual WAN, **GCP:** Network **GCP:** Network Connectivity Center.



Opción 1 - Conectividad Híbrida: Site-to-Site VPN

Concepto

Establece un túnel seguro y encriptado (IPsec) entre tu centro de datos on-premise y tu VPC/VNet a través de la **internet pública**.

Componentes (AWS)

- **Virtual Private Gateway (VPG):** El "concentrador" de VPN en el lado de la VPC
- **Customer Gateway (CGW):** Un recurso en AWS que representa tu dispositivo físico de VPN on-premise

Pros

- Rápido de configurar
- Coste relativamente bajo

Contras

- El rendimiento y la latencia dependen de la calidad de tu conexión a internet
- Ancho de banda limitado

Opción 2 - Conexión Dedicada (Direct Connect)

Concepto

Una conexión de red **física, privada y dedicada** desde tu centro de datos a una ubicación de ubicación de AWS. El tráfico **nunca** toca la internet pública.

Características

- **Ancho de Banda:** Consistente y predecible. Velocidades desde 50 Mbps hasta 100 Gbps
- **Latencia:** Baja y estable
- **Seguridad:** Máxima, al ser una conexión privada

Caso de Uso IA CRÍTICO

- Transferencia inicial o continua de **grandes datasets** (Terabytes o Petabytes) desde on-desde on-premise a S3 para entrenamiento
- Aplicaciones híbridas donde modelos en la nube necesitan acceder a bases de datos de muy datos de muy baja latencia que residen on-premise

Agnosticismo: AWS: Direct Connect, **Azure:** ExpressRoute, **GCP:** Cloud Interconnect.

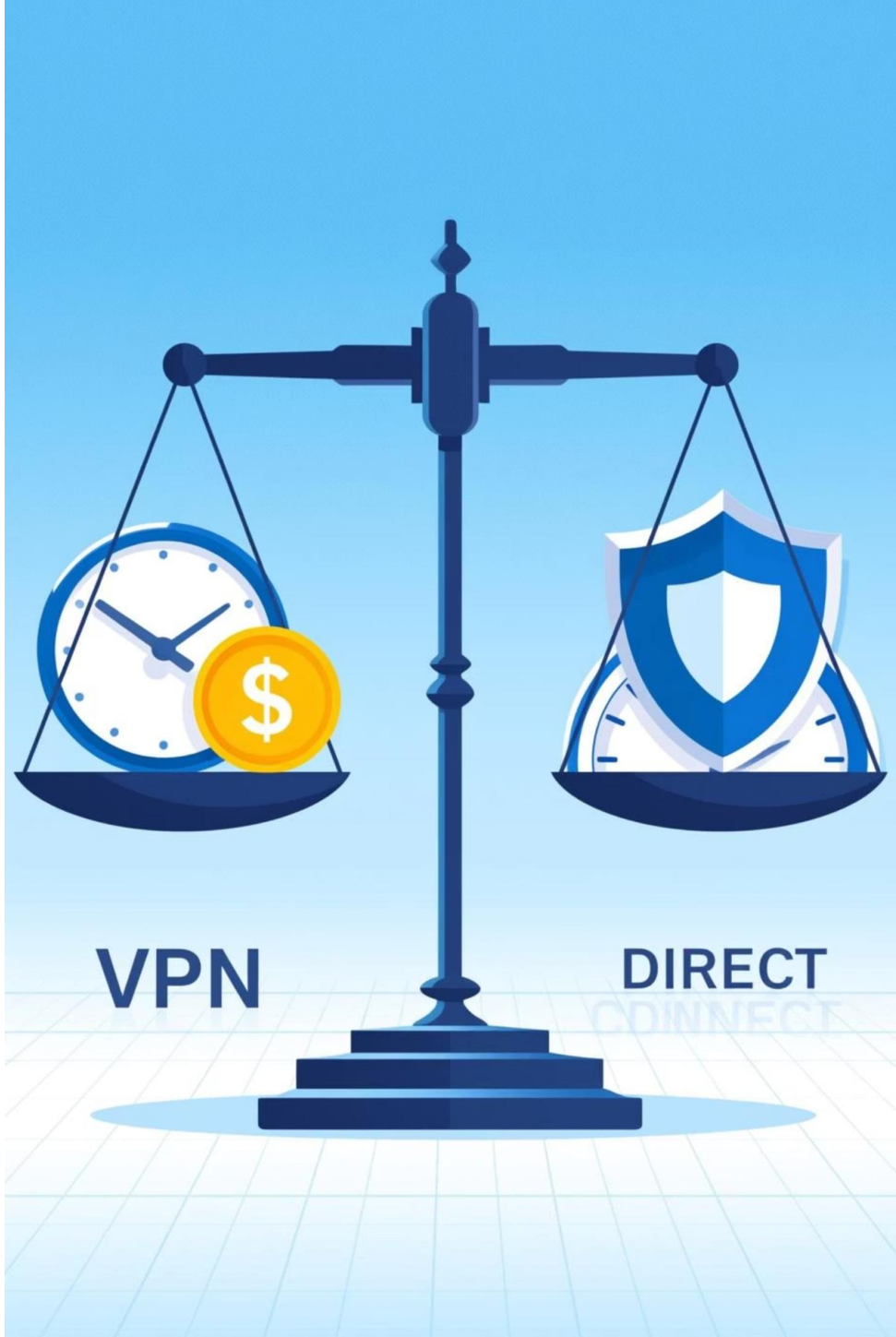
VPN vs. Conexión Dedicada: ¿Cuándo usar cada una?

Usa Site-to-Site VPN cuando:

- Necesitas una solución rápida y de bajo coste
- Las cargas de trabajo no son sensibles a la latencia o a las fluctuaciones del ancho de banda
- El volumen de transferencia de datos es bajo-moderado
- Como solución de backup para una conexión dedicada

Usa Direct Connect / ExpressRoute cuando:

- La consistencia del rendimiento de red es crítica
- Necesitas transferir grandes volúmenes de datos de forma regular
- Requieres una latencia ultra-baja y estable
- Las políticas de seguridad o compliance prohíben el paso de datos por internet



Debate Rápido

Escenario

Una startup de IA médica ha desarrollado un algoritmo que analiza imágenes de resonancia magnética. Su dataset de entrenamiento inicial es de 50 TB y reside en los servidores del hospital cliente. Planean recibir 1 TB de nuevas imágenes cada mes. El hospital exige la máxima seguridad.

Preguntas para debate

1. Para la carga inicial de 50 TB, ¿qué recomendarías? ¿VPN, Direct Connect, o quizás un servicio de transferencia física como AWS Snowball?

2. Para la transferencia mensual de 1 TB, ¿cuál sería la solución de conectividad híbrida más apropiada a largo plazo?

Puntos a debatir: Coste inicial vs. coste operativo, tiempo de transferencia, seguridad, escalabilidad.

Acelerando la Entrega: Content Delivery Networks (CDN)

Concepto

Una red global de servidores proxy (llamados **Edge Locations** o Puntos de Presencia - PoP) que Presencia - PoP) que almacenan en caché tu contenido cerca de tus usuarios finales. finales.

Funcionamiento

1. Un usuario en Japón solicita un archivo
2. La primera petición va a tu servidor de origen (ej. un bucket S3 en EE.UU.)
3. La CDN lo entrega al usuario, pero también guarda una copia en una Edge Location en Tokio
4. El siguiente usuario en Japón que pida el mismo archivo lo recibirá directamente desde Tokio, con una latencia muchísimo menor

Agnosticismo: AWS: CloudFront, **Azure:** Azure CDN, **GCP:** Cloud Cloud CDN.

Casos de Uso de CDN para IA

Distribución de Modelos de Inferencia

Para modelos que corren en el cliente (navegador con TensorFlow.js, TensorFlow.js, aplicaciones móviles), la CDN puede cachear los archivos archivos del modelo (model.json, weights.bin) para una carga y arranque arranque ultra-rápidos.

Entrega de Interfaces de Usuario

Las UIs de aplicaciones como ChatGPT o Midjourney se benefician enormemente de la CDN para cargar rápidamente la interfaz y los assets estáticos.

Resultados de Inferencia Cacheables

Si una inferencia para una entrada común es solicitada frecuentemente (ej. "¿cuál es la capital de España?"), el resultado del API se puede cachear en la CDN por un tiempo corto.

Streaming de Media Procesada

Entregar vídeos o audios que han sido procesados por IA (ej. subtitulado subtitulado automático, reconocimiento de objetos) a una audiencia audiencia global.

DNS y Enrutamiento Global: AWS Route 53

Concepto

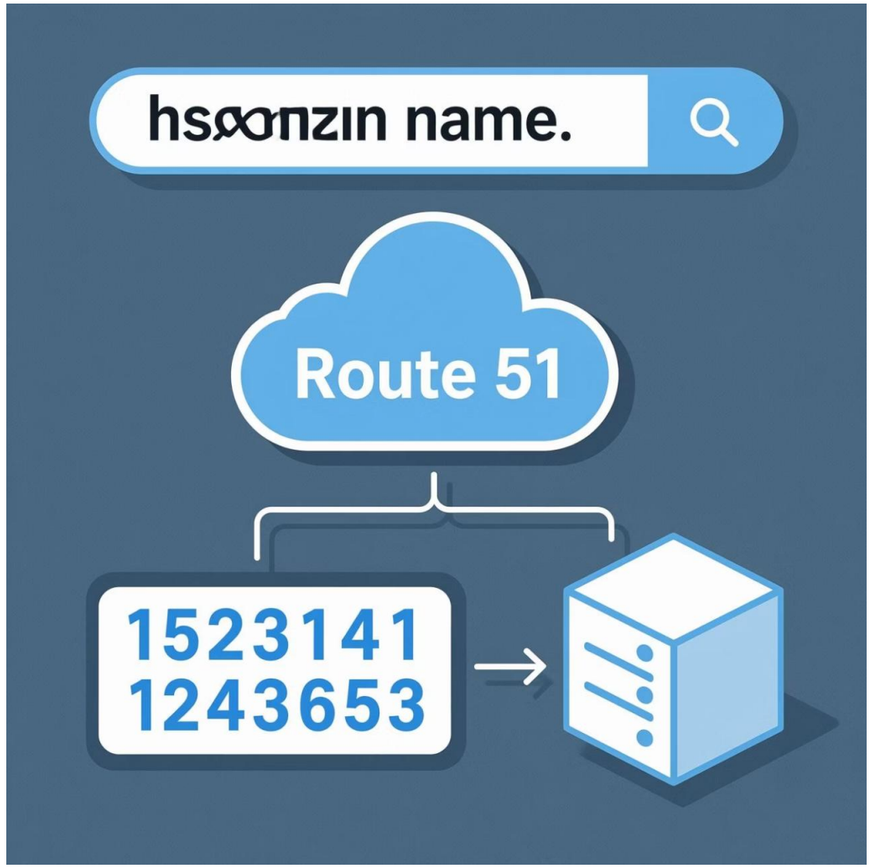
Un servicio de **Sistema de Nombres de Dominio (DNS)** gestionado, altamente altamente disponible y escalable. Traduce nombres de dominio legibles por por humanos (ej. miapi.com) a direcciones IP (54.23.1.182).

Más que un simple DNS

Route 53 (y sus equivalentes) ofrece **políticas de enrutamiento avanzadas** para **avanzadas** para dirigir el tráfico de manera inteligente.

Agnosticismo

AWS: Route 53, **Azure:** Azure DNS, **GCP:** Cloud DNS. (Route 53 es generalmente generalmente considerado el más rico en funcionalidades de enrutamiento). enrutamiento).



Políticas de Enrutamiento Inteligente (Ej: Route 53)

Simple

Enrutamiento básico a una única IP.

Failover

Enrutamiento Activo-Pasivo. Dirige el tráfico a un tráfico a un recurso primario. Si las comprobaciones de estado (health checks) fallan, fallan, automáticamente redirige al recurso secundario en otra región.

Basado en Geolocalización

Dirige el tráfico basado en la ubicación geográfica geográfica del usuario (ej. usuarios de Europa Europa van a servidores en Frankfurt, usuarios de usuarios de Asia a Singapur).

Basado en Latencia

Dirige al usuario a la región de AWS que le proporciona la menor latencia posible. ¡No siempre es la más cercana geográficamente! geográficamente!

Ponderado (Weighted)

Distribuye el tráfico entre múltiples recursos según pesos que tú defines (ej. 90% a la nueva versión de un modelo, 10% a la antigua para un despliegue Canary).

Caso de Uso IA: Despliegue Global de Modelos

Escenario

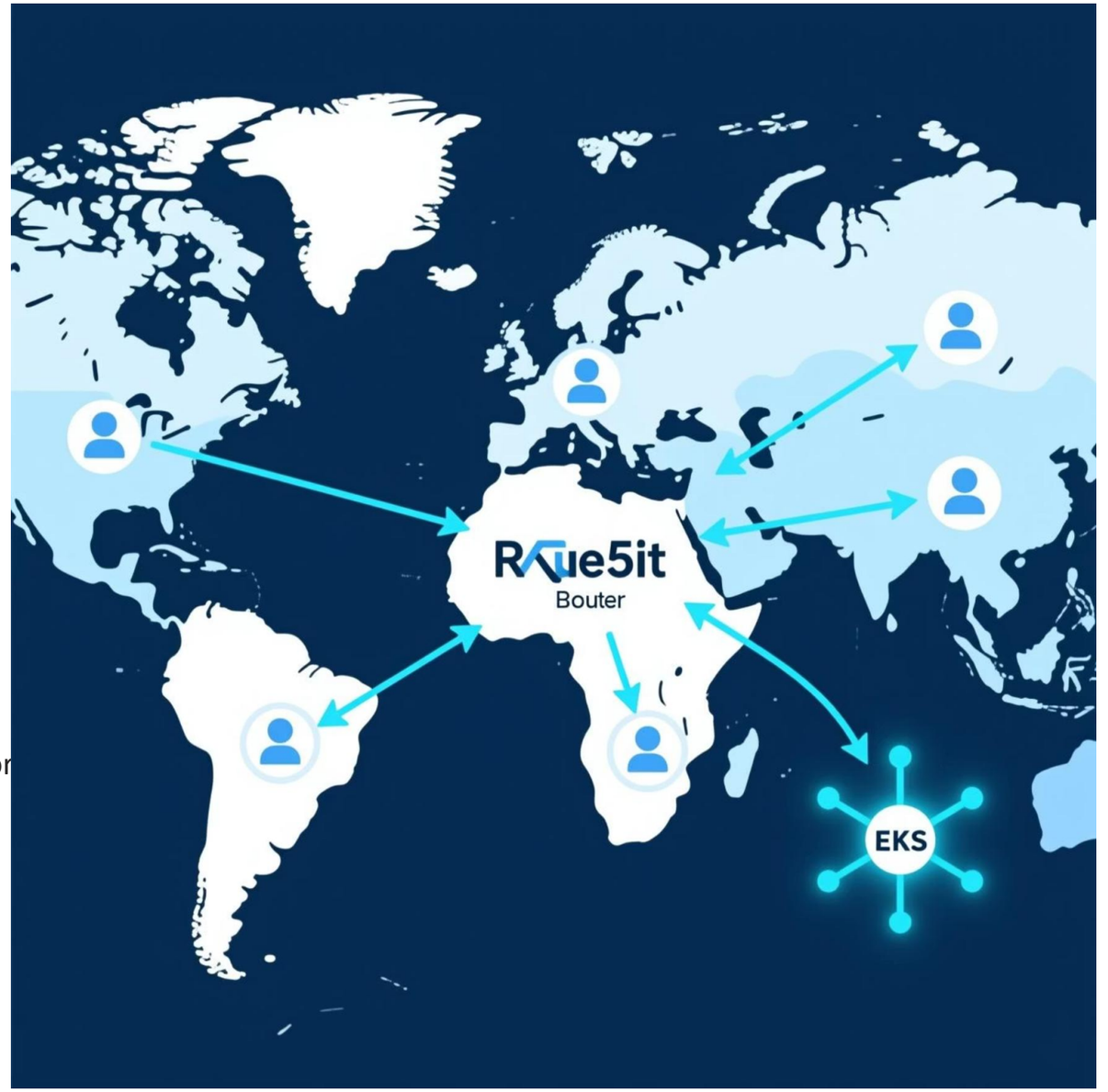
Tienes un modelo de traducción de texto desplegado en clústeres de clústeres de Kubernetes (EKS) en tres regiones: Irlanda (Europa), (Europa), Virginia (EE.UU. Este) y Tokio (Asia).

Arquitectura de Red Global

1. **DNS:** Se configura un registro en **Route 53** para translate.api.com
2. **Política de Enrutamiento:** Se elige la política **Basada en Latencia**
3. **Health Checks:** Route 53 monitoriza constantemente la salud de los los balanceadores de carga en cada región

Flujo

- Un usuario en España resuelve translate.api.com. Route 53 determina que la determina que la latencia más baja es a Irlanda y devuelve la IP del balanceador de esa región
- Un usuario en México hace lo mismo. Route 53 le envía a Virginia
- Si el clúster de Tokio falla un health check, Route 53 deja de enviar tráfico allí tempor



Protección contra Ataques DDoS: AWS Shield

¿Qué es un ataque DDoS?

Un intento malicioso de hacer que un servicio no esté disponible inundándolo con una avalancha de tráfico de internet desde múltiples fuentes.

AWS Shield Standard

- **Activación:** Automática y gratuita para todos los clientes de AWS
- **Protección:** Defiende contra los ataques DDoS más comunes de capa 3 (red) y capa 4 (transporte) que buscan saturar la red

AWS Shield Advanced

- **Servicio de pago**
- **Protección Avanzada:** Detección y mitigación mejorada para ataques de ataques de gran escala y sofisticados, incluyendo ataques a nivel de de aplicación (capa 7).
- **Beneficios:** Acceso 24/7 al Equipo de Respuesta de AWS (DRT), visibilidad visibilidad del ataque casi en tiempo real y protección de costes

Firewall de Aplicaciones Web: AWS WAF

Concepto

Un firewall que ayuda a proteger tus aplicaciones web o APIs contra exploits web comunes que podrían afectar la disponibilidad, comprometer la seguridad o consumir recursos excesivos. Opera en la **Capa 7 (Aplicación)** del modelo OSI.

Protege contra

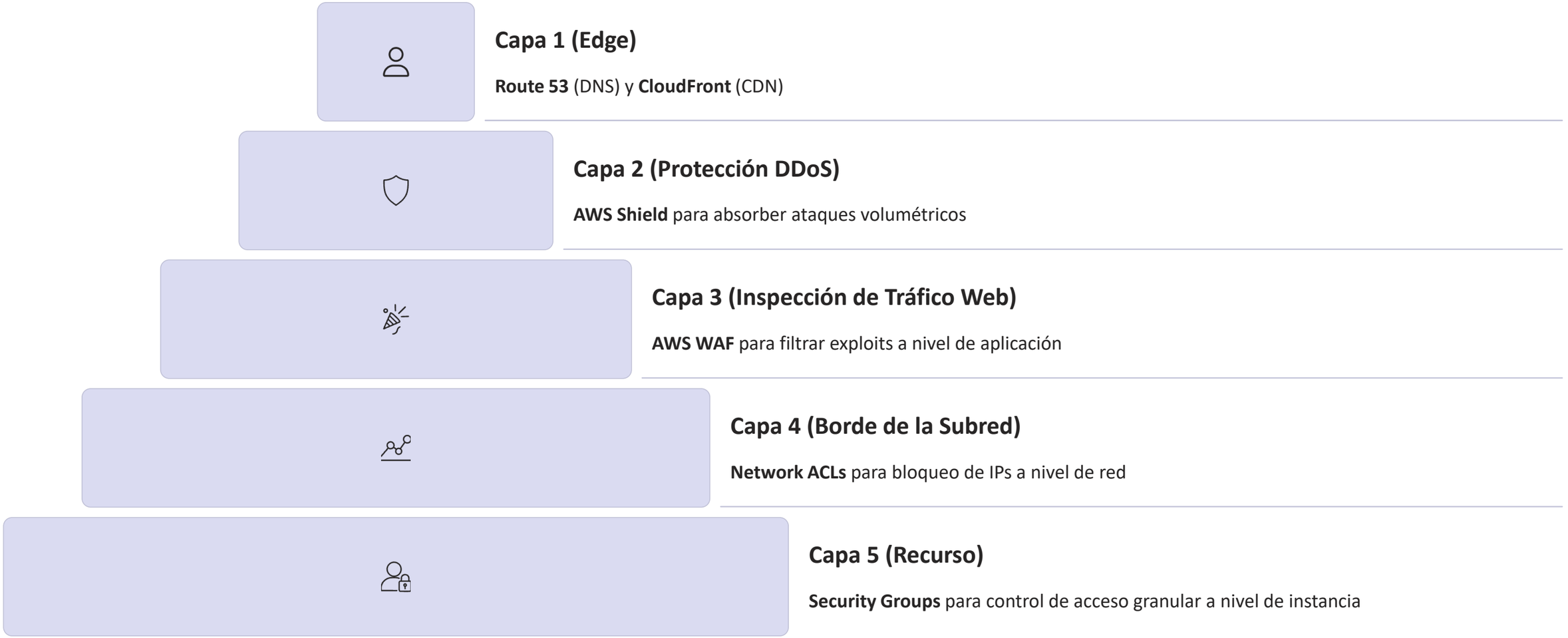
- **Inyección de SQL (SQLi):** Previene que atacantes inserten código SQL malicioso en las peticiones
- **Cross-Site Scripting (XSS):** Evita que se inyecten scripts en tu web que se ejecuten en los navegadores de tus usuarios
- Y otros ataques del OWASP Top 10

¿Cómo funciona?

- Se integra con CloudFront, Application Load Balancer, API Gateway
- Creas reglas (ACLs web) para bloquear peticiones basadas en la IP en la IP de origen, cabeceras HTTP, cuerpo de la petición, etc.
- Puedes usar **reglas gestionadas** por AWS o por partners de seguridad para una protección inmediata

Arquitectura de Seguridad en Profundidad

Recapitulemos las capas de seguridad de red que hemos visto. La estrategia no es usar una, sino todas, en un enfoque de **defensa en profundidad**.



Capa 1 (Edge)

Route 53 (DNS) y **CloudFront** (CDN)

Capa 2 (Protección DDoS)

AWS Shield para absorber ataques volumétricos

Capa 3 (Inspección de Tráfico Web)

AWS WAF para filtrar exploits a nivel de aplicación

Capa 4 (Borde de la Subred)

Network ACLs para bloqueo de IPs a nivel de red

Capa 5 (Recurso)

Security Groups para control de acceso granular a nivel de instancia

Cada capa protege a la siguiente. Si un atacante supera una, se encuentra con la siguiente.

Poniéndolo todo junto: Arquitectura de Red para IA a Escala

Entrenamiento

- **Datos:** On-premise, transferidos vía **Direct Connect** a un bucket S3
- **Acceso a S3:** Desde la VPC de entrenamiento (subred privada) a través de través de un **VPC Gateway Endpoint** para S3
- **Clúster de Cómputo:** Instancias EC2 con GPU en una subred privada, protegidas por **Security Groups** que solo permiten acceso SSH desde un bastión

Inferencia

- **Despliegue:** Modelo servido por un **API Gateway** o un **Application Load Balancer** con contenedores Fargate/EKS
- **Seguridad:** El endpoint está protegido por **AWS WAF** y **Shield**
- **Distribución Global:** El API se sirve a través de **CloudFront (CDN)**

Conectividad: Múltiples VPCs (dev, test, prod) interconectadas vía Transit Gateway

Consideraciones de Coste

Transferencia de Datos (El más importante)

- La entrada de datos (ingress) a la nube suele ser **gratuita**
- La salida de datos (egress) desde la nube hacia internet **tiene coste** por GB
- La transferencia de datos entre Zonas de Disponibilidad (AZs) **tiene coste**. La transferencia dentro de la misma AZ es gratuita

Servicios Gestionados

- **NAT Gateway:** Coste por hora + coste por GB procesado
- **Direct Connect:** Coste por hora del puerto + coste de transferencia de datos
- **Transit Gateway:** Coste por conexión + coste por GB procesado

Optimización

- Usar VPC Endpoints para evitar el NAT Gateway
- Usar CDNs para reducir el egress desde el origen
- Diseñar para minimizar el tráfico inter-AZ si no es estrictamente necesario para la HA

Resumen y Conclusiones Clave

La VPC es la base

Un diseño de CIDR y subredes bien planificado es crucial para el futuro.

La seguridad es por capas

Combina WAF, Shield, NACLs y Security Security Groups para una defensa en profundidad.

La conectividad privada es clave clave

Usa Endpoints, Peering y Transit Gateway para evitar la internet pública dentro de la nube.

El mundo híbrido es una realidad

Elige entre VPN (rapidez, bajo coste) y Direct Connect (rendimiento, (rendimiento, seguridad) según las necesidades de tu carga de carga de trabajo de IA.

Piensa globalmente

Usa CDNs y DNS inteligente (Route 53) para ofrecer baja latencia y alta disponibilidad a tus aplicaciones de IA en todo el mundo.

Próximos Pasos y Lecturas

Próxima Sesión

"Sesión 8: Contenedores y Orquestación para IA: Docker, Kubernetes Kubernetes (EKS/GKE/AKS) y Flujos de MLOps".

Lectura Recomendada

- Whitepaper de AWS: "Building a Scalable and Secure Multi-VPC Network VPC Network Infrastructure"
- Documentación de Azure: "Azure Virtual Network concepts and best practices"

Práctica sugerida

Desplegar una aplicación de 3 capas en una VPC (web en subred pública, app/BBDD en privada) usando Terraform o CloudFormation. Configurar los Security Groups para un aislamiento estricto.

Preguntas y Debate Abierto

¿Preguntas?

¡Gracias por vuestra atención!

Email: [email@universidad.es]

Slack: [canal-curso]