Cloud Computing para Inteligencia Artificial

Sesión 3: Servicios de Computación I - Computación I - Máquinas Virtuales Virtuales

Departamento de Informática

Universidad de Ciencias Aplicadas

Agenda de la Sesión

01

Fundamentos

¿Qué es una Máquina Virtual (VM) en la Nube?

02

El "Zoológico" de Instancias

Tipos y familias en AWS EC2

03

Configuración Inicial

Imágenes (AMIs) y Almacenamiento

04

Protegiendo el Perímetro

Redes y Seguridad para VMs

05

Optimización de Costes

Modelos de Precios

06

Elasticidad

Escalado y Balanceo de Carga

07

Aplicaciones en IA

Casos de uso prácticos

08

Comparativa

AWS vs. Azure vs. GCP

Repaso y Conexión con la Sesión Anterior

Recordatorio Sesión 2: "Fundamentos del Cloud Computing"

- Modelo de Responsabilidad Compartida
- Regiones, Zonas de Disponibilidad y Edge Locations
- Modelos de servicio: IaaS, PaaS, SaaS

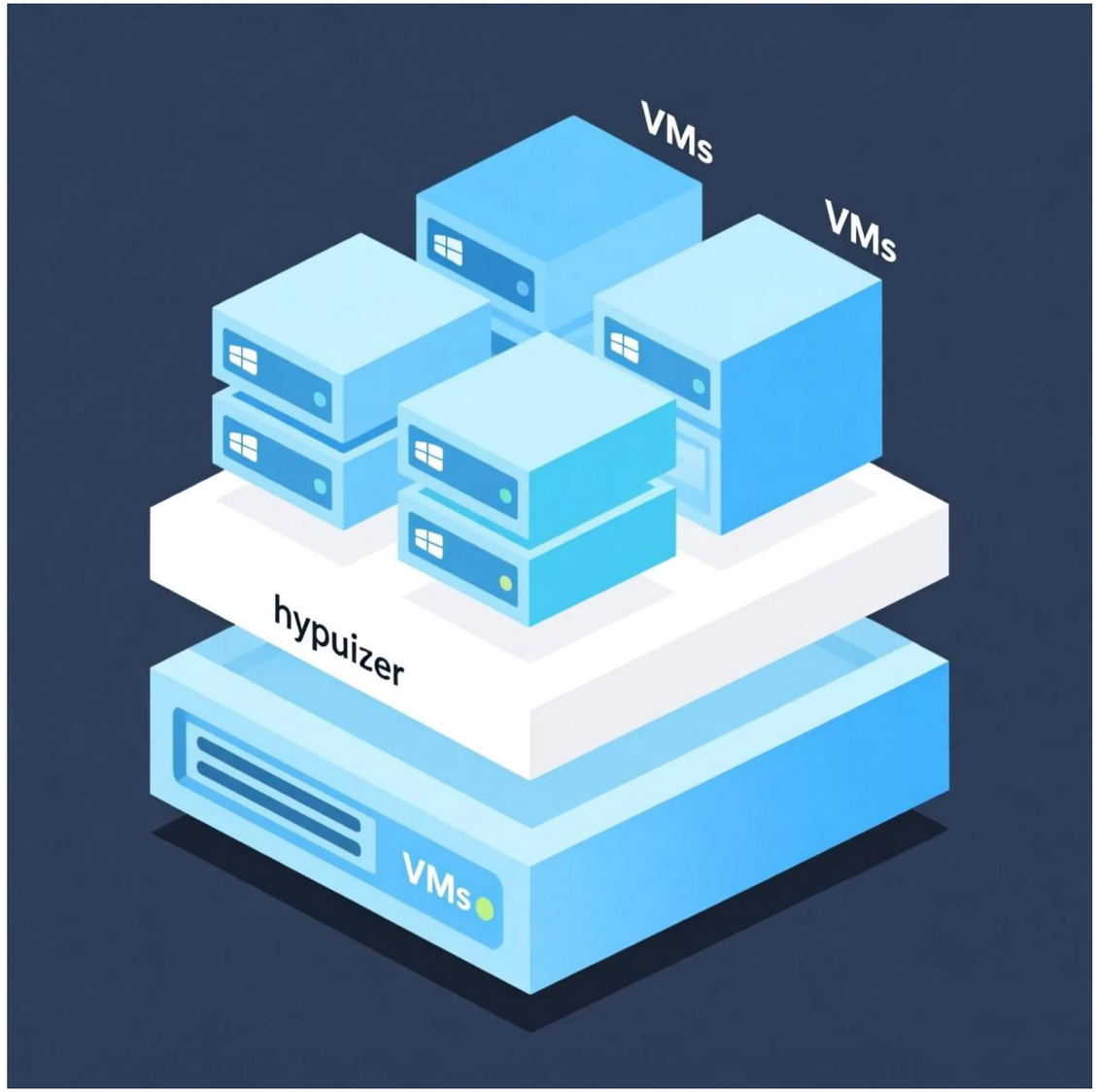
¿Qué es una Máquina Virtual en la Nube?

Una **Máquina Virtual (VM)** es una emulación por software de un sistema informático completo. En la nube, es tu propio servidor privado, pero virtualizado.

Abstracción del Hardware: La VM te proporciona recursos virtualizados virtualizados (vCPU, RAM, Disco, Red) que son una porción del hardware hardware físico subyacente de un centro de datos del proveedor cloud. cloud.

El Rol del Hipervisor: Es el software clave que crea y gestiona las VMs. Se VMs. Se ejecuta en el servidor físico (host) y asigna los recursos a cada VM cada VM (guest).

- **Tipo 1 (Bare-metal):** Xen, KVM, Hyper-V. Se ejecutan directamente sobre el hardware.



Ventajas Clave de las VMs en la Nube

Aislamiento (Isolation)

Cada VM está lógicamente aislada de las demás, aunque compartan hardware. Un fallo o una brecha de seguridad en una VM no afecta directamente a las otras.

Flexibilidad (Flexibility)

Control total sobre el sistema operativo, las aplicaciones, las librerías y la configuración. Puedes instalar casi cualquier software que correrías en un servidor físico. ¡Es tu "lienzo en blanco"!

Control a Nivel de SO

Tienes acceso de administrador (root o Administrator), lo que te permite ajustar el kernel, gestionar parches y configurar el sistema a bajo nivel.

Portabilidad

Las imágenes de las VMs pueden ser migradas entre diferentes hosts diferentes hosts físicos o incluso entre nubes (con herramientas de herramientas de conversión).

Introducción a las Instancias de AWS EC2

EC2 (Elastic Compute Cloud) es el servicio de AWS para proporcionar capacidad de cómputo segura y redimensionable en la nube. Es uno de los servicios más antiguos y fundamentales de AWS.

El término **"Instancia"** es la forma en que AWS se refiere a una Máquina Máquina Virtual.

La clave de EC2 es la **variedad**. No existe "una" VM, sino una enorme selección de "familias" y "tamaños" de instancias, cada una optimizada para un tipo de carga de trabajo diferente.

Sintaxis del Nombre: familia + generación + atributos_adicionales . tamaño . tamaño

Ejemplos: t3.large, m5.xlarge, p4d.24xlarge

Familias de Instancias: Propósito General

Descripción

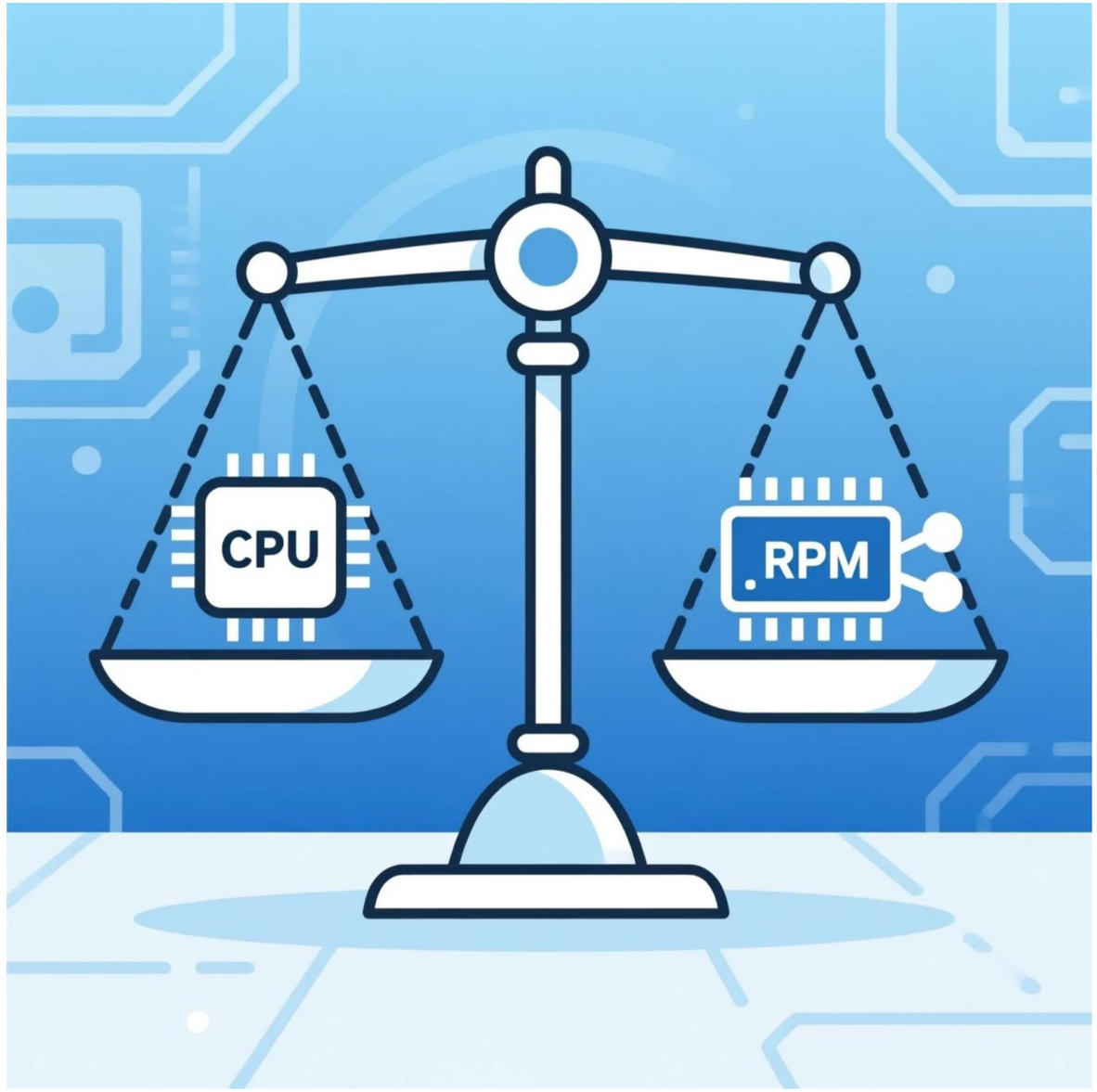
Ofrecen un equilibrio entre recursos de cómputo (CPU), memoria (RAM) y red. (RAM) y red. Son la navaja suiza de las instancias.

Series Principales en AWS

- **Serie M (ej. m5, m6g):** El pilar para una amplia variedad de aplicaciones. Relación balanceada de vCPU:RAM (típicamente 1:4).
- **Serie T (ej. t2, t3, t4g):** Instancias de rendimiento ampliable (burstable). Tienen una línea base de rendimiento de CPU y acumulan "créditos de CPU" cuando están inactivas.

Casos de Uso Típicos

- Servidores web y de aplicaciones
- Entornos de desarrollo y pruebas
- Bases de datos pequeñas y medianas
- Microservicios



Familias de Instancias: Optimizadas para Cómputo

Descripción

Priorizan la potencia de procesamiento. Tienen una alta relación de vCPU por cada GB de RAM.

Serie Principal en AWS

- **Serie C (ej. c5, c6g):** Utilizan los procesadores más rápidos y están diseñadas para aplicaciones que requieren un alto rendimiento de CPU.

Casos de Uso en IA y Computación

- **Procesamiento por lotes (Batch Processing):** Procesar grandes volúmenes de datos donde la CPU es el cuello de botella.
- **Codificación de Vídeo (Transcoding):** Tareas de alta intensidad de CPU.
- **Modelado Científico y HPC:** Simulaciones, modelado molecular, etc.
- **Inferencia de Machine Learning:** Cuando el modelo está muy optimizado para CPU.

Familias de Instancias: Optimizadas para Memoria

Descripción

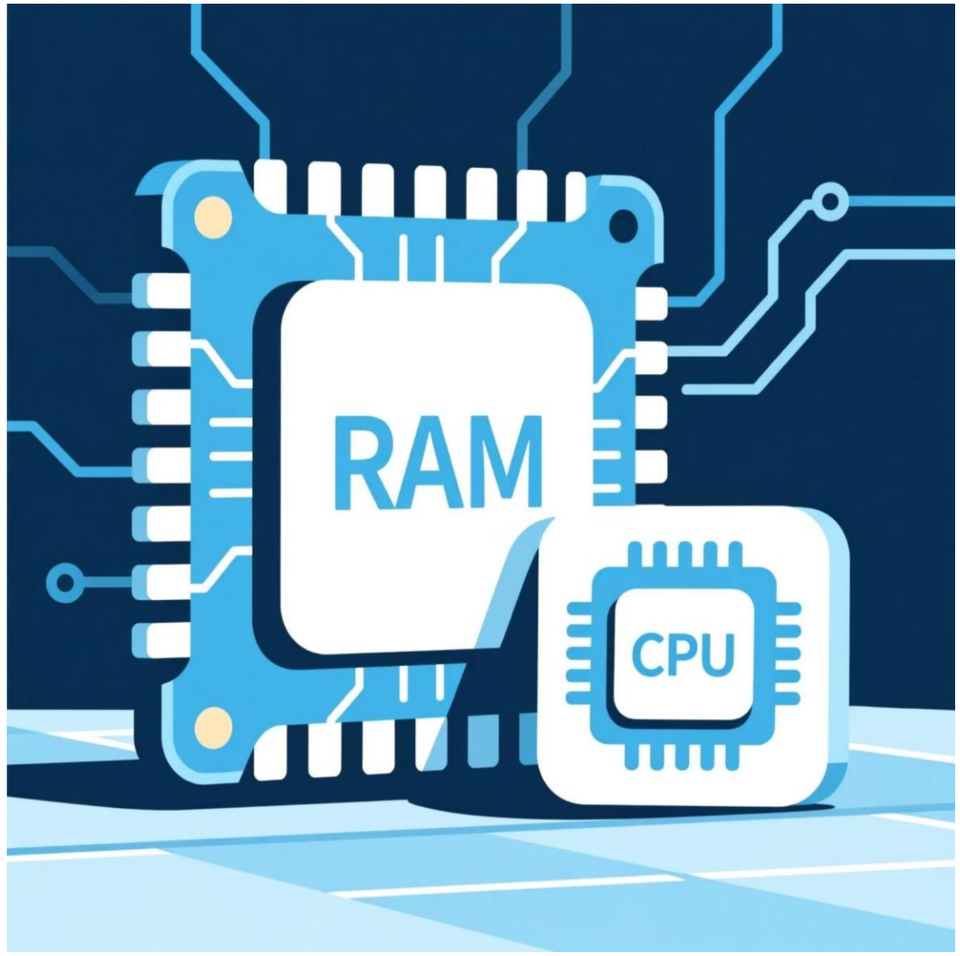
Diseñadas para cargas de trabajo que procesan grandes conjuntos de datos en memoria. Ofrecen una gran cantidad de RAM por cada vCPU.

Series Principales en AWS

- **Serie R (ej. r5, r6g):** Punto de partida para aplicaciones con uso intensivo de memoria.
- **Serie X (ej. x1, x2g):** Tienen una de las mayores proporciones de RAM por vCPU. Pueden llegar a tener terabytes de RAM.
- **Serie Z (ej. z1d):** Combinan una frecuencia de CPU muy alta con una gran cantidad de memoria.

Casos de Uso en IA y Datos

- **Bases de datos en memoria** como SAP HANA, Redis o Memcached.
- **Análisis y minería de datos** a gran escala.
- **Pre-procesamiento de grandes datasets** para entrenamiento de modelos de IA.



Familias de Instancias: Optimizadas para Almacenamiento

Descripción

Ofrecen un almacenamiento local muy rápido y de baja latencia, ideal para aplicaciones que necesitan un acceso de alta velocidad a los datos.

Series Principales en AWS

- **Serie I (ej. i3, i4i):** Optimizadas para un alto número de operaciones de entrada/salida entrada/salida por segundo (IOPS) con almacenamiento local NVMe SSD.
- **Serie D (ej. d2, d3):** Optimizadas para almacenamiento denso, proporcionando terabytes de almacenamiento local en HDD.

Casos de Uso en IA y Big Data

- **Bases de datos NoSQL** como Cassandra o ScyllaDB.
- **Data Warehousing** a gran escala.
- **Sistemas de ficheros distribuidos** como HDFS (Hadoop).
- Cargas de trabajo que generan muchos datos temporales que necesitan ser leídos y escritos rápidamente.

Familias de Instancias: Cómputo Acelerado (¡Clave para IA!)

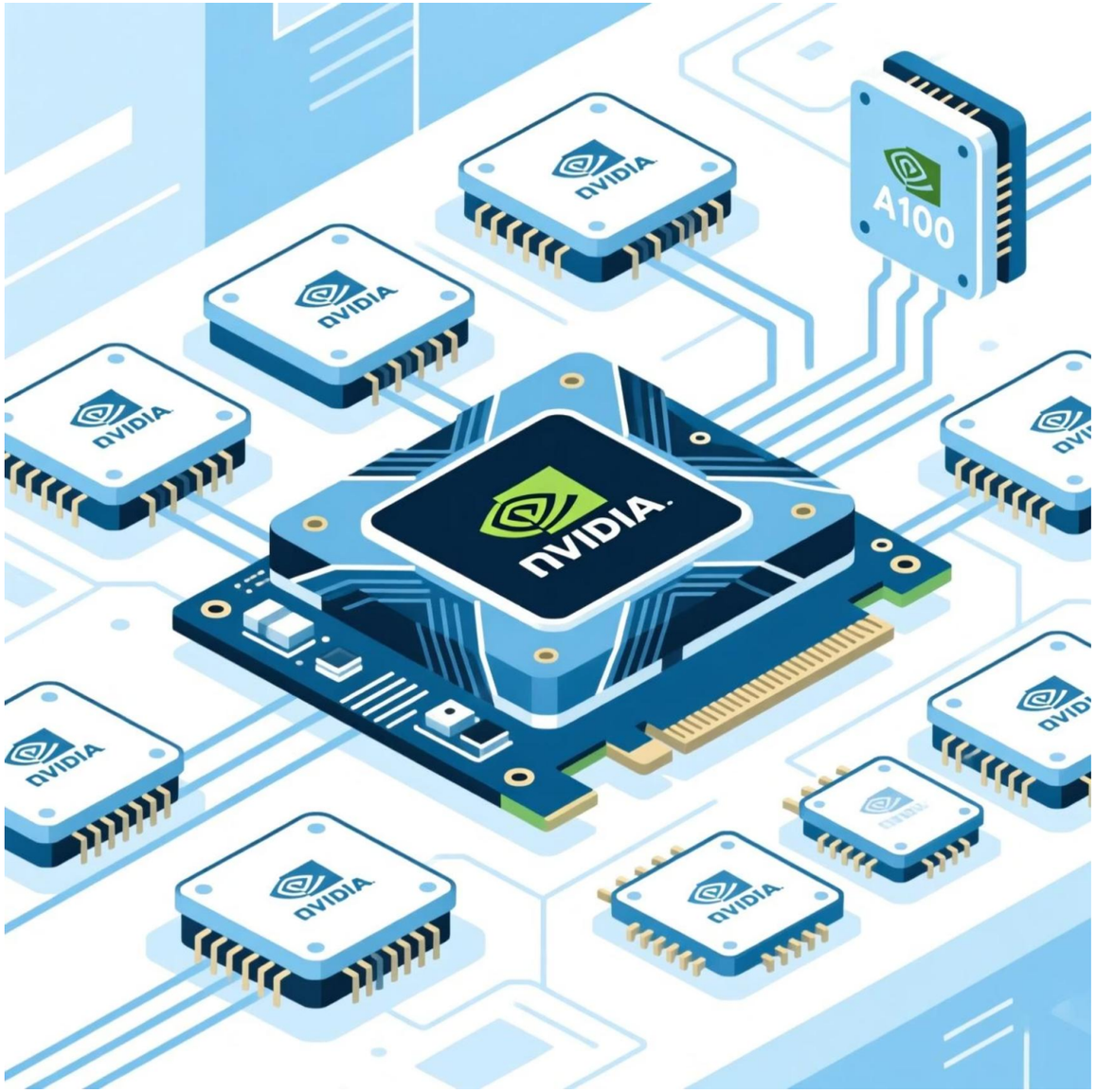
Descripción

Instancias equipadas con aceleradores de hardware (GPUs, FPGAs o chips personalizados) para potenciar tareas de computación paralela masiva.

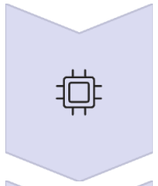
Series Principales en AWS

- **Serie P (ej. p3, p4d):** Equipadas con GPUs NVIDIA de alta gama (como A100). El estándar para el entrenamiento de modelos de Deep Learning.
- **Serie G (ej. g4, g5):** GPUs más versátiles, buenas para inferencia de ML, renderizado de gráficos y transcodificación.
- **Serie Inf (ej. Inf1, Inf2):** Usan el chip **AWS Inferentia**, diseñado a medida para una inferencia de ML de alto rendimiento y bajo coste.

- **Serie Trn (ej. Trn1):** Usan el chip **AWS Trainium**, diseñado a medida para el entrenamiento de modelos de Deep Learning a gran escala.



¿Cómo Elegir la Instancia Correcta?

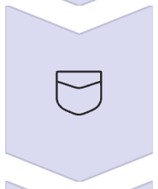


vCPU y Arquitectura

¿Necesitas un procesador Intel, AMD o ARM (Graviton de AWS)? ¿Cuántos cores?

Memoria (RAM)

¿Tu aplicación necesita cargar grandes datasets en memoria?



Almacenamiento

¿Necesitas almacenamiento local ultra-rápido (NVMe) o almacenamiento persistente en red (EBS)?

Rendimiento de Red

¿Tu aplicación es sensible a la latencia o necesita un gran ancho de banda?

Aceleradores de Hardware

¿La carga de trabajo se beneficia de GPUs o chips específicos de ML?

Coste

Siempre es un factor determinante.

Elegir la instancia adecuada es un balance crucial entre rendimiento y coste.

Comparativa de Instancias Cloud

El concepto de familias de instancias es común en todos los proveedores cloud.

AWS EC2

- **Familias:** M (General), C (Cómputo), R/X (Memoria), P/G/Inf/Trn (Aceleradas). (Aceleradas).

Azure Virtual Machines

- **Series:** D-series (General), F-series (Cómputo), E-series (Memoria), L-series L-series (Almacenamiento), N-series (GPU).

Google Compute Engine (GCE)

- **Tipos de Máquina:** E2 (General), C2 (Cómputo), M1/M2 (Memoria), A2 (GPU).

- **Diferenciador Clave:** Permite crear "Custom Machine Types", donde puedes puedes especificar el número exacto de vCPUs y la cantidad de RAM que RAM que necesitas.

Imágenes de Máquina (AMIs)

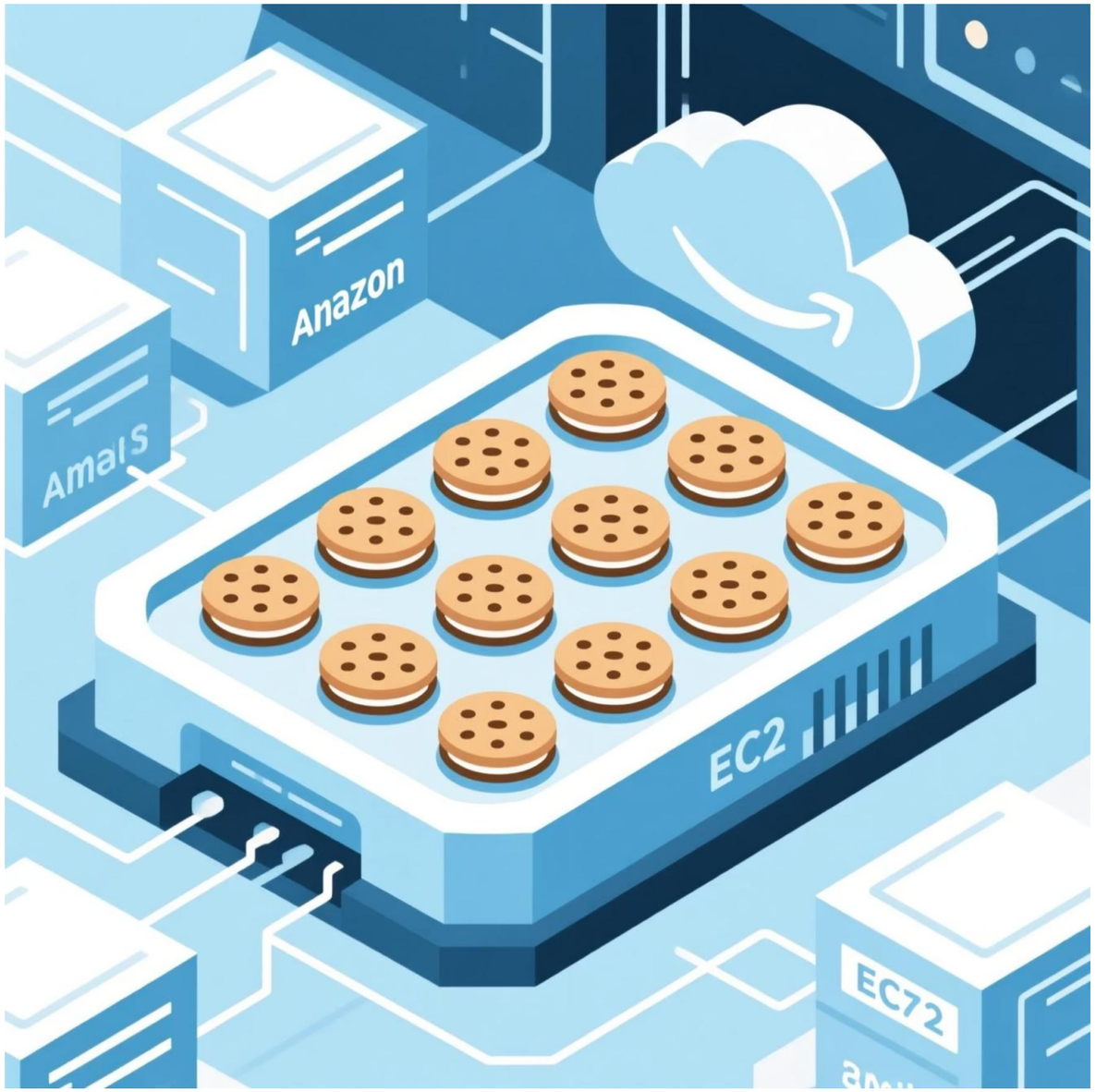
Una **Amazon Machine Image (AMI)** es una plantilla preconfigurada que preconfigurada que contiene todo lo necesario para lanzar una una instancia. Es el "plano" de tu servidor.

Contenido de una AMI

1. Un Sistema Operativo (Linux, Windows Server).
2. Software de aplicación y librerías (ej: un servidor web, un stack de Python con TensorFlow, CUDA drivers).
3. Configuraciones de permisos y de almacenamiento.

Importancia Estratégica

- **Estandarización:** Asegura que cada instancia lanzada desde la misma AMI sea idéntica.
- **Despliegue Rápido (Time-to-Market):** Reduce drásticamente el tiempo de configuración. En lugar de instalar y configurar todo desde cero, lanzas una instancia instancia desde una AMI y está lista en minutos.



Tipos de AMIs

1. AMIs Públicas (Quick Start)

Proporcionadas y mantenidas por AWS, con SOs comunes (Amazon Linux, Ubuntu, Windows). Son la base más habitual.

2. AMIs del Marketplace

Ofrecidas por terceros (vendedores de software). Son imágenes "listas para usar" con software comercial o de código abierto preinstalado y con licencia (ej: una AMI con una base de datos Oracle, una firewall de Fortinet, o una AMI de Deep Learning de NVIDIA).

3. AMIs Personalizadas (My AMIs)

Son las que tú creas. Lanzas una instancia, la configuras exactamente como la necesitas (instalas software, aplicas parches, ajustas configuraciones) y luego creas una AMI a partir de ella. Es la práctica recomendada para tus aplicaciones.

Almacenamiento para Instancias I: Efímero

Almacenamiento de Instancia (Instance Store / Ephemeral Storage)

¿Qué es? Discos locales conectados físicamente al servidor host que aloja tu instancia.

Ventajas

- **Rendimiento Extremo:** Ofrece una latencia muy baja y un altísimo número número de IOPS, ya que no hay latencia de red. Ideal para datos temporales. temporales.

¡ADVERTENCIA! - Datos Volátiles

Los datos en el Instance Store **se pierden permanentemente** si la instancia se instancia se **detiene (stop)**, **hiberna** o **termina (terminate)**. También se pueden pueden perder en caso de fallo del hardware subyacente.

Caso de Uso

Espacio para swap, cachés, datos temporales que se generan y eliminan durante un proceso.

Almacenamiento para Instancias II: Persistente

Amazon Elastic Block Store (EBS)

¿Qué es? Son volúmenes de almacenamiento en bloque (como un disco duro externo) externo) basados en red, diseñados para una alta disponibilidad y durabilidad. durabilidad.

Características Clave

- **Persistente:** Los datos persisten independientemente del ciclo de vida de la la instancia. Puedes desacoplar un volumen de una instancia y acoplarlo a otra. otra.
- **Flexible:** Puedes elegir el tipo de volumen según tus necesidades de rendimiento y coste (HDD, SSD de propósito general, SSD de IOPS provisionadas).

- **Backups (Snapshots):** Puedes tomar "instantáneas" (snapshots) de tus volúmenes volúmenes EBS y guardarlas en Amazon S3.

Redes para VMs: Direcciones IP

Cada instancia en la nube se lanza dentro de una red virtual (VPC en AWS).

Dirección IP Privada

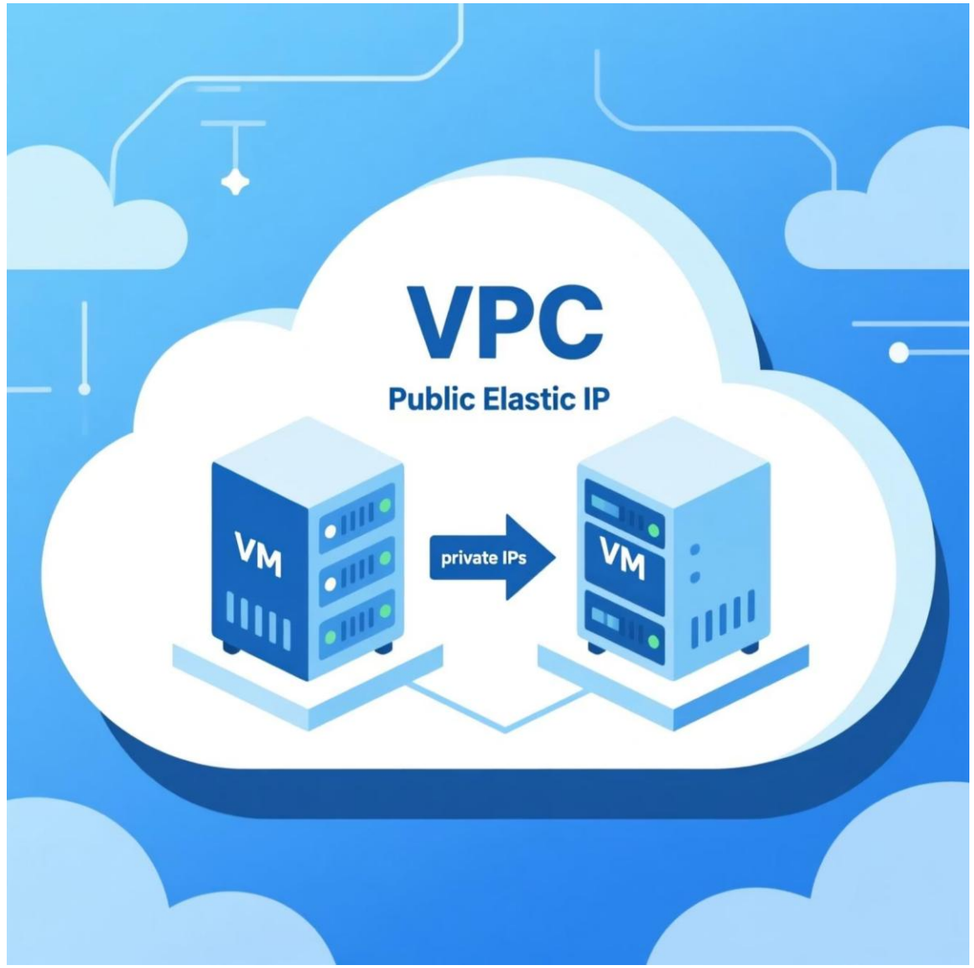
- Asignada automáticamente a la instancia desde el rango de la red.
- Se usa para la comunicación entre instancias dentro de la misma red virtual. No es accesible desde Internet.
- Es fija mientras la instancia está en ejecución.

Dirección IP Pública

- Asignada para que la instancia pueda comunicarse con Internet.
- Por defecto, es dinámica: si detienes e inicias la instancia, la IP pública cambiará.

IP Elástica (Elastic IP en AWS)

Es una dirección IP pública **estática** que puedes asignar a tu cuenta y asociarla a asociarla a una instancia. Permanece fija hasta que la liberes explícitamente. explícitamente.



Seguridad para VMs: Security Groups

¿Qué son?

Un **Security Group (SG)** actúa como un firewall virtual **a nivel de instancia (stateful)**. Controla el tráfico de red entrante (inbound) y saliente (outbound).

¿Cómo funcionan?

- Se definen **reglas**. Cada regla especifica un protocolo (TCP, UDP, ICMP), un rango de puertos y un origen (para inbound) o destino (para outbound).
- El origen/destino puede ser una dirección IP, un rango de IPs (formato CIDR), u otro Security Group.
- **Stateful:** Si permites el tráfico entrante en un puerto, la respuesta saliente correspondiente se permite automáticamente.

Ejemplo para un Servidor Web

- **Regla Inbound:** Permitir tráfico TCP en el puerto 80 desde cualquier origen (0.0.0.0/0).
- **Regla Inbound:** Permitir tráfico TCP en el puerto 22 (SSH) solo desde la IP de tu oficina.

Acceso Seguro a Instancias: Key Pairs

Para conectarse a una instancia, se utilizan mecanismos de autenticación autenticación seguros en lugar de contraseñas.

Para Instancias Linux

- Se usa un **Par de Claves (Key Pair)** basado en criptografía de clave pública-privada.
- Al crear el par de claves, AWS genera una **clave pública** (que se guarda en la instancia) y una **clave privada** (que tú descargas y guardas de forma segura, archivo .pem).
- Para conectarte vía **SSH (Secure Shell)**, tu cliente SSH utiliza la clave privada clave privada para autenticarse con la clave pública en la instancia.
- `ssh -i mi-clave-privada.pem ec2-user@<ip-publica-instancia>`

Para Instancias Windows

Se usa el mismo par de claves para obtener la contraseña de administrador administrador inicial y conectarse vía **RDP (Remote Desktop Protocol)**. **Protocol)**.

GAMIFICACIÓN - Elige la Arquitectura Correcta

Escenario (Pregunta para Mentimeter / Kahoot):

> "Estás liderando un proyecto para **entrenar un modelo de lenguaje grande (LLM)**. El proceso de entrenamiento dura aproximadamente 40 horas y es tolerante a interrupciones (puedes reanudarlo desde checkpoints). Tu presupuesto es muy limitado."

¿Qué combinación de instancia y modelo de precios elegirías en AWS?

1

Instancia m5.large (Propósito General) con precio On-Demand Demand

Seguro pero lento y caro.

2

Instancia p4d.24xlarge (GPU A100) con una Instancia Reservada a 3 años

Potente pero un compromiso excesivo y carísimo para un solo entrenamiento.

3

Instancia p3.2xlarge (GPU V100) como Spot Instance, guardando guardando checkpoints periódicamente en S3

Potente, coste muy bajo y el riesgo de interrupción se mitiga con checkpoints.

4

Instancia c5.xlarge (Optimizada para Cómputo) con precio On-Demand On-Demand

Tipo de instancia incorrecto, el entrenamiento de LLMs necesita GPUs.

Modelos de Precios I: On-Demand

Concepto

Paga por la capacidad de cómputo por segundo (o por hora, dependiendo del dependiendo del SO), sin compromisos a largo plazo.

Pros

- **Máxima Flexibilidad:** Lanza y termina instancias cuando quieras.
- **Sin Inversión Inicial:** No hay pagos por adelantado.

Contras

- **El más caro:** Es el precio de lista, el más alto por hora.

Ideal para

- Aplicaciones con cargas de trabajo irregulares y a corto plazo que no pueden ser interrumpidas.
- Desarrollo y pruebas.
- Cuando lanzas una aplicación por primera vez y no conoces la demanda.

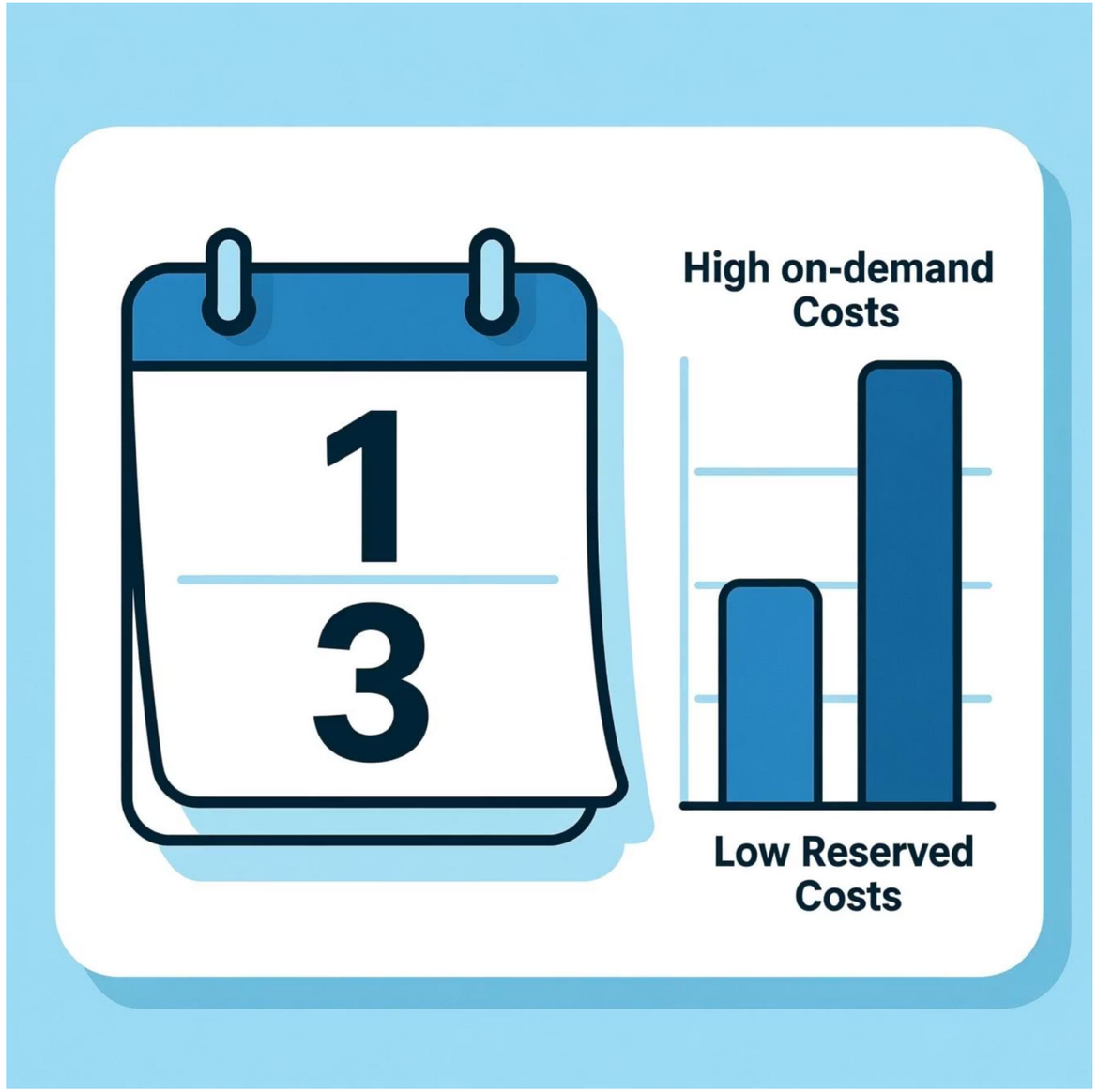
Modelos de Precios II: Reservas y Ahorros

Instancias Reservadas (Reserved Instances - RIs)

- Te comprometes a usar una configuración específica de instancia (familia, (familia, tamaño, SO) en una región, por un periodo de **1 o 3 años**.
- A cambio, obtienes un **descuento significativo** (hasta 72%) sobre el precio On-Demand.

Savings Plans

- Un modelo más flexible. Te comprometes a un gasto de cómputo por hora (ej: $10/hora) por un periodo de 1 o 3 años.
- Este descuento se aplica automáticamente a cualquier uso de EC2 (independientemente de la familia, tamaño o región), hasta alcanzar tu compromiso.



Ideal para

Modelos de Precios III: Spot Instances (¡El Secreto del Ahorro en IA!)

Concepto

AWS (y otros proveedores) tienen una enorme cantidad de capacidad de cómputo ociosa. Las **Spot Instances** te permiten "pujar" por esa capacidad no utilizada.

Resultado

Puedes obtener descuentos de **hasta el 90%** sobre el precio On-Demand.

La Contrapartida (¡Muy importante!)

AWS puede **interrumpir y reclamar tu instancia Spot** con solo 2 minutos de aviso si necesita aviso si necesita esa capacidad para usuarios On-Demand.

Ideal para

Tabla Comparativa de Modelos de Precios

Característica	On-Demand	Savings Plans / RIs	Spot Instances
Ideal para	Cargas impredecibles	Cargas estables	Cargas tolerantes a fallos
Compromiso	Ninguno	1 o 3 años	Ninguno
Ahorro vs On-Demand	0%	Hasta 72%	Hasta 90%
Riesgo de Interrupción	No	No	Sí (con 2 min de aviso)
Flexibilidad	Máxima	Media/Alta	Alta (pero con riesgo)

Cada modelo de precio tiene sus ventajas y desventajas. La elección óptima dependerá de tu caso de uso específico, predictibilidad de la carga y tolerancia a interrupciones.

Escalado y Balanceo: Escalado Vertical vs. Horizontal

Escalado Vertical (Scaling Up)

- Aumentar los recursos de una única instancia (más vCPU, más RAM, un tamaño de instancia mayor).
- **Pros:** Sencillo de implementar.
- **Contras:** Requiere un **reinicio** de la instancia (tiempo de inactividad). Tiene un límite físico. Se convierte en un punto único de fallo.

Escalado Horizontal (Scaling Out)

- Añadir más instancias para distribuir la carga entre ellas.
- **Pros:** Aumenta la disponibilidad y la tolerancia a fallos. No hay límite teórico de escalado.
- **Contras:** Requiere una arquitectura más compleja (balanceo de carga).

¿Qué hacer cuando una sola VM no es suficiente para manejar la carga?

Escalado Horizontal Automatizado: Auto Scaling Groups

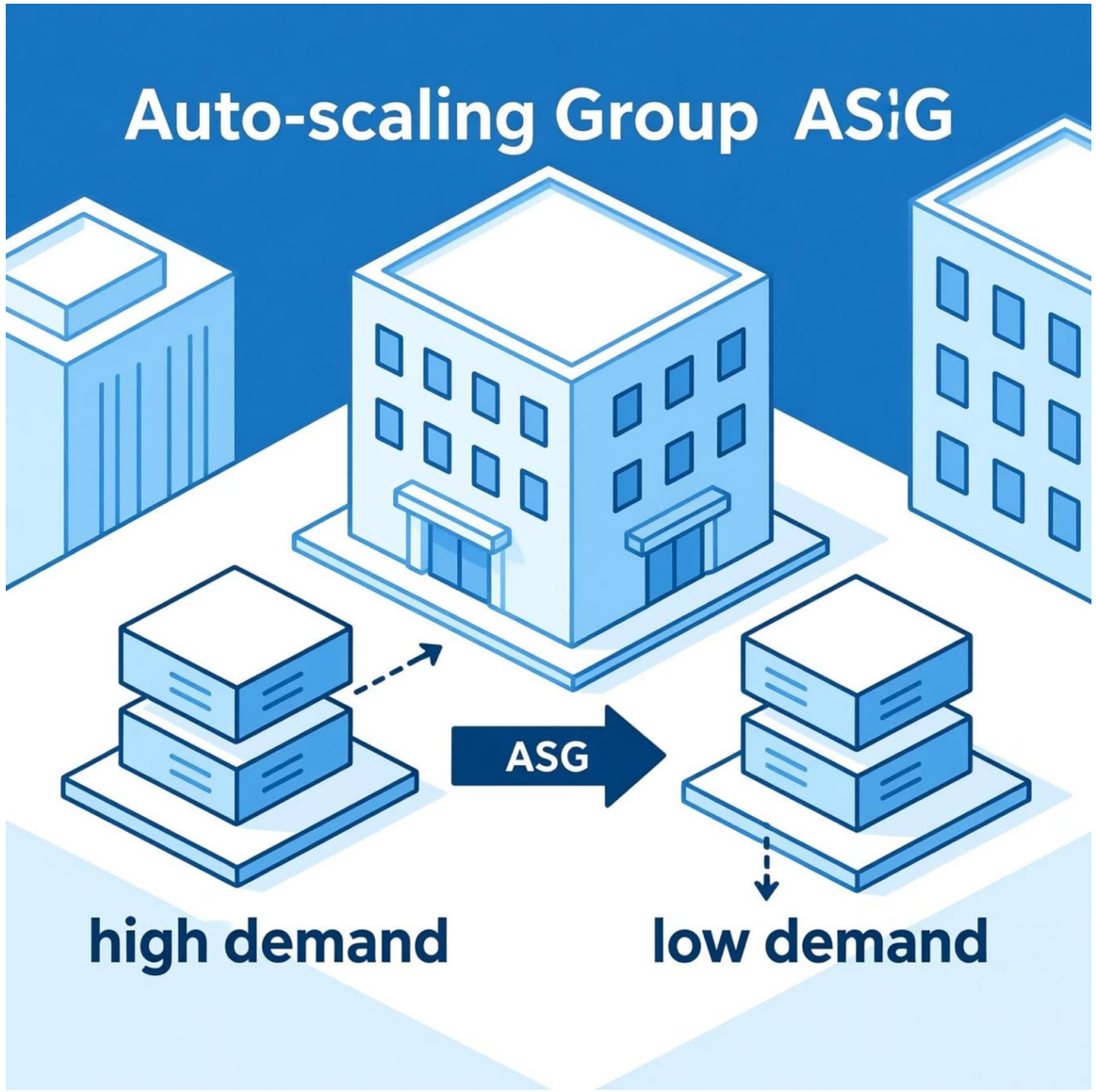
Concepto

Un **Auto Scaling Group (ASG)** en AWS gestiona una colección de instancias EC2. Su objetivo es asegurar que siempre tengas el número correcto de instancias para manejar la carga de tu aplicación.

Funciones Clave

1. **Mantenimiento de la Capacidad:** Si una instancia falla, el ASG lanza una nueva para reemplazarla, manteniendo la capacidad deseada.
2. **Escalado Dinámico:** Puedes definir políticas para escalar horizontalmente de forma automática.
    - **Scale Out (Añadir instancias):** si el uso promedio de CPU supera el 70%.
    - **Scale In (Quitar instancias):** si el uso promedio de CPU cae por debajo del 30%.
3. **Escalado Programado:** Escalar en momentos predecibles (ej: añadir servidores a las 9 servidores a las 9 am antes de que los usuarios lleguen).

Equivalentes: Azure Virtual Machine Scale Sets, GCP Managed Instance Groups.



Distribución del Tráfico: Elastic Load Balancing

¿Qué es?

Un **Elastic Load Balancer (ELB)** distribuye automáticamente el tráfico de red entrante a través de múltiples instancias EC2 en diferentes zonas de disponibilidad.

Beneficios

- **Alta Disponibilidad:** Si una instancia falla, el ELB deja de enviarle tráfico y lo redirige a las instancias sanas.
- **Tolerancia a Fallos:** Puede operar a través de múltiples AZs. Si una zona de disponibilidad completa cae, el tráfico se enruta a las AZs restantes.
- **Escalabilidad:** Se integra perfectamente con Auto Scaling Groups.

Tipos Principales en AWS

- **Application Load Balancer (ALB):** Capa 7 (HTTP/HTTPS). Inteligente, puede Inteligente, puede enrutar el tráfico basado en el contenido de la petición.
- **Network Load Balancer (NLB):** Capa 4 (TCP/UDP). Ultra-alto rendimiento, latencia muy baja.

petición.

Casos de Uso de VMs para Inteligencia Artificial

Entornos de Desarrollo

Lanzar una instancia con GPU (ej: serie G de AWS) y una AMI de Deep Learning preconfigurada para prototipar modelos rápidamente en un entorno Jupyter Notebook.

Entrenamiento de Modelos

Utilizar instancias P o Trn con múltiples GPUs. Aprovechar Spot Instances para reducir el coste del entrenamiento hasta en un 90%.

Inferencia y Despliegue

Desplegar modelos entrenados en endpoints API usando instancias optimizadas para cómputo (Serie C) o aceleradas para inferencia (Serie Inf) dentro de un Auto Scaling Group.

Procesamiento de Datos

Crear clusters de VMs para ejecutar frameworks como Apache Spark o Spark o Hadoop para el pre-procesamiento de terabytes de datos.

Resumen, Comparativa Final y Preguntas

Resumen Clave de Hoy

- Las VMs (EC2) son el pilar IaaS del cloud, ofreciendo control y flexibilidad.
- Elegir la **familia**, el **almacenamiento** y el **modelo de precios** correctos es crucial para optimizar rendimiento y coste.
- Para cargas de trabajo de IA, las **instancias aceleradas (GPU, Inferentia)** y las **Spot Spot Instances** son herramientas increíblemente poderosas.
- La combinación de **Auto Scaling** y **Load Balancing** permite crear arquitecturas elásticas y resil

Comparativa Final