

# AWS de la teoría a la Practica Modulo 1

→ Release 12-2022.01.31

# Instructor



## **Diego Facundo Gracilieri**

Cloud Leader/Cloud Architected/Cloud Consultant  
Certificaciones Cloud:

- AWS Certified Solutions Architect - Associate
- AWS Certified SysOps Administrator - Associate
- AWS Certified Developer - Associate
- AWS Certified Devops Engineer - Professional
- AWS Certified Solution Architect - Professional
- Microsoft Certified: Azure Administrator Associate

# Agenda Dia 1

- Introducción Cloud.
- Proveedores nube.
- Cómputo en virtualizando.
  - Infraestructura Global.
  - VPC.
  - EC2.
  - Creando instancias.
  - ASG y ELB.
  - AWS S3.
  - AWS EBS.
  - AWS EFS.
  - laboratorio.

# Agenda Dia 2

- Monitoreo.
  - Cloudwatch y Cloudtrail.
  - Colectores y métricas.
  - Alarmas, logs, eventos y filtros.
  - Alertas y tableros.
  - Creando monitoreo.
  - VPC Flow Logs.
  - Costos.
  - Laboratorio.

# Agenda Dia 3

- Seguridad y Redes.
  - IAM más allá de LDAP.
  - Cómo aplicar el mínimo privilegio Viable.
  - Cifrando mi información.
  - Diseñando mi red en AWS.
  - balanceando y enrutando mi tráfico.
  - Transfiriendo mis datos.
  - Laboratorio.

# Agenda Dia 4

- Base de datos Relacionales.
  - Ventajas y casos de uso.
  - Zero down time.
  - RDS HA.
  - Creación y configuración.
  - Copias de Seguridad y restauración.
  - Monitoreo.
  - Configuración de parámetros.
  - Laboratorio

# Agenda Dia 5

- Introducción a contenedores.
  - Que es Docker.
  - Que es Kubernetes.
  - Que es ECS.
  - Que es EKS.
  - Laboratorio de despliegue de la solución ( ECS ).

# Agenda

[Dia 1](#)

[Dia 2](#)

[Dia 3](#)

[Dia 4](#)

[Dia 5](#)

# Dia 1

→ Por que hace sentido la Nube

# On Premise vs Cloud, ¿cuál es la mejor para tu empresa?

Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet. Esta tecnología evita que las empresas tengan que encargarse de aprovisionar, configurar o gestionar los recursos y permite que paguen únicamente por los que usen.

**IaaS** : brinda todos los beneficios de los recursos informáticos locales . los usuarios se encargan de las aplicaciones, los datos, el sistema operativo.



# On Premise vs Cloud, ¿cuál es la mejor para tu empresa?

**PaaS:** Un proveedor de servicios externo se encarga de proporcionar y gestionar el hardware y una plataforma de software de aplicaciones, pero el usuario es quien maneja la aplicación y los datos.

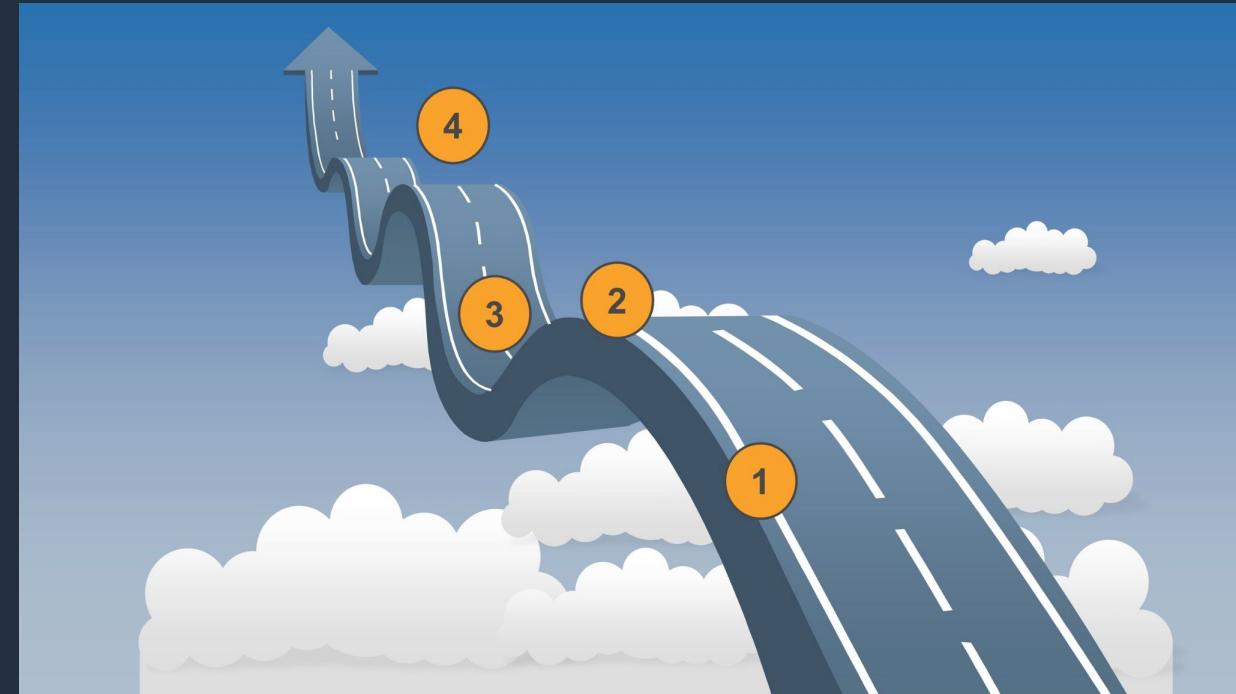
**SaaS:** Es un servicio que ofrece a los usuarios una aplicación web a través de un explorador, de cuya gestión se ocupa el proveedor de servicios.



# ¿ Por qué la Nube ?

Sin duda depende de las necesidades de la organización. No obstante, es innegable las grandes ventajas que ofrecen las soluciones en la nube para las empresas que quieren ahorrar costos y al mismo tiempo tener la posibilidad de escalar su negocio.

Como primer paso a considerar si la nube es la opción existen diversas estrategias que los proveedores de nube brindan para evaluar el modelo de madurez entre los cuales podemos encontrar que son el Journey To Cloud y el Well Architected Framework.



# Principales Proveedores Nube

## AWS

Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global, con una alta presencia global y zonas de infraestructura Global.



## Azure

La plataforma Azure está compuesta por más de 200 productos y servicios en la nube diseñados para ayudarle a enfocarse en los servicios de Microsoft.



## GCP

Google Cloud Platform proveedor de nube que centra sus servicios con un fuerte enfoque en servicios de analítica y Kubernetes.



# Beneficios de la Nube

## Facil

Diseña para facilitar tareas habituales de administración, despliegue de soluciones, monitoreo, respuesta a incidentes, ajustarse fácilmente a las necesidades de las empresas y altamente escalable y resiliente.

## Elastico

A través de diferentes mecanismo de escalamiento automático posibilita que la infraestructura se adapte de un manera sencilla ante los cambios del negocio de una manera automática disminuyendo carga en los equipos de operación.

## Costos

Costo eficiencia se refiere no tan solo ahorrar costos, sino hace referencia al uso inteligente del presupuesto de manera que el dinero de la organización se logre invertir de una mejor manera haciendo uso del pago por uso.

# Beneficios de la Nube Escalable

Con las herramientas de AWS, Auto Scaling Group y Elastic Load Balancing, su aplicación podrá ampliarse o reducirse según la demanda. Gracias al respaldo de la sólida infraestructura de Amazon, tendrá acceso a los recursos informáticos y de almacenamiento siempre que los necesite.

## Seguridad.

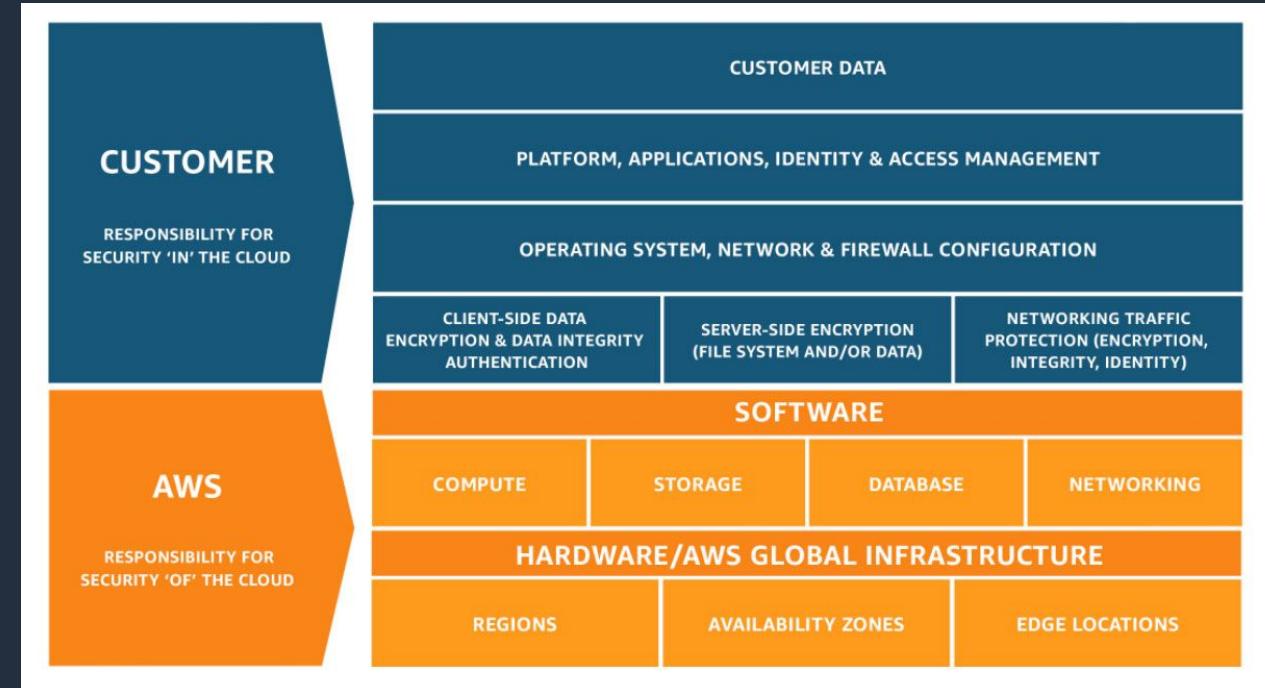
Aplica un enfoque integral para proteger y reforzar nuestra infraestructura, incluidas medidas físicas, operativas y de software. Para obtener más información, consulte el Centro de seguridad de AWS. Estableciendo un modelo de seguridad

llamado responsabilidad compartida, donde se explica cuál es la responsabilidad del proveedor cloud y cual es la responsabilidad del cliente.

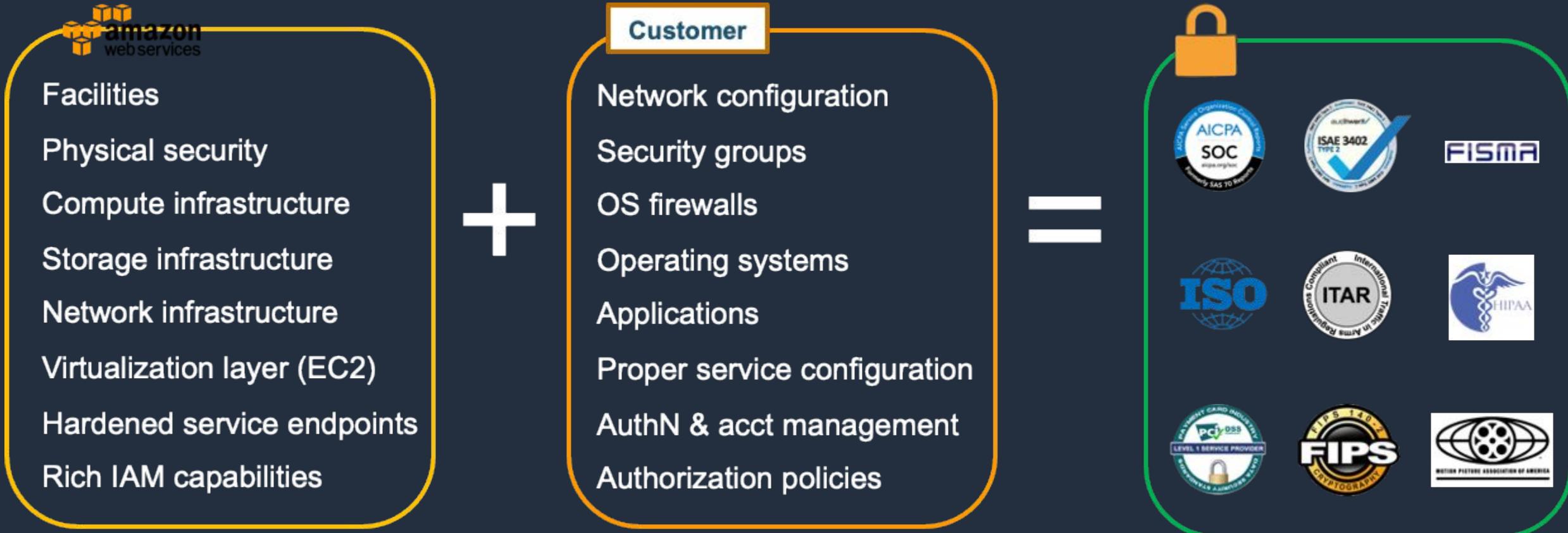
# Modelo de responsabilidad compartida

**AWS es responsable de proteger la infraestructura** que ejecuta todos los servicios provistos en la nube de AWS. Esta infraestructura está conformada por el hardware, el software, las redes y las instalaciones que ejecutan los servicios de la nube de AWS.

La responsabilidad del cliente estará determinada por los servicios de la nube de AWS que el cliente seleccione. Esto determina el alcance del trabajo de configuración a cargo del cliente como parte de sus responsabilidades de seguridad. Es decir, el cliente tiene la responsabilidad de proteger la información en los recursos desplegados.



# Modelo de responsabilidad compartida



# Computo Virtualizado

→ Hablaremos acerca del servicio de EC2, VPC y mucho más.

# Infraestructura Global

## 26 regiones lanzadas

Cada una con varias zonas de disponibilidad (AZ)

## 84 zonas de disponibilidad

## 17 zonas locales

### 25 zonas Wavelength

Para aplicaciones con latencia ultrabaja

## 8 regiones anunciadas

## 32 zonas locales anunciadas

## El doble de regiones

Con múltiples AZ que el siguiente proveedor de nube más grande

## 245 países y territorios atendidos

## 108 ubicaciones de Direct Connect

## Más de 310 puntos de presencia

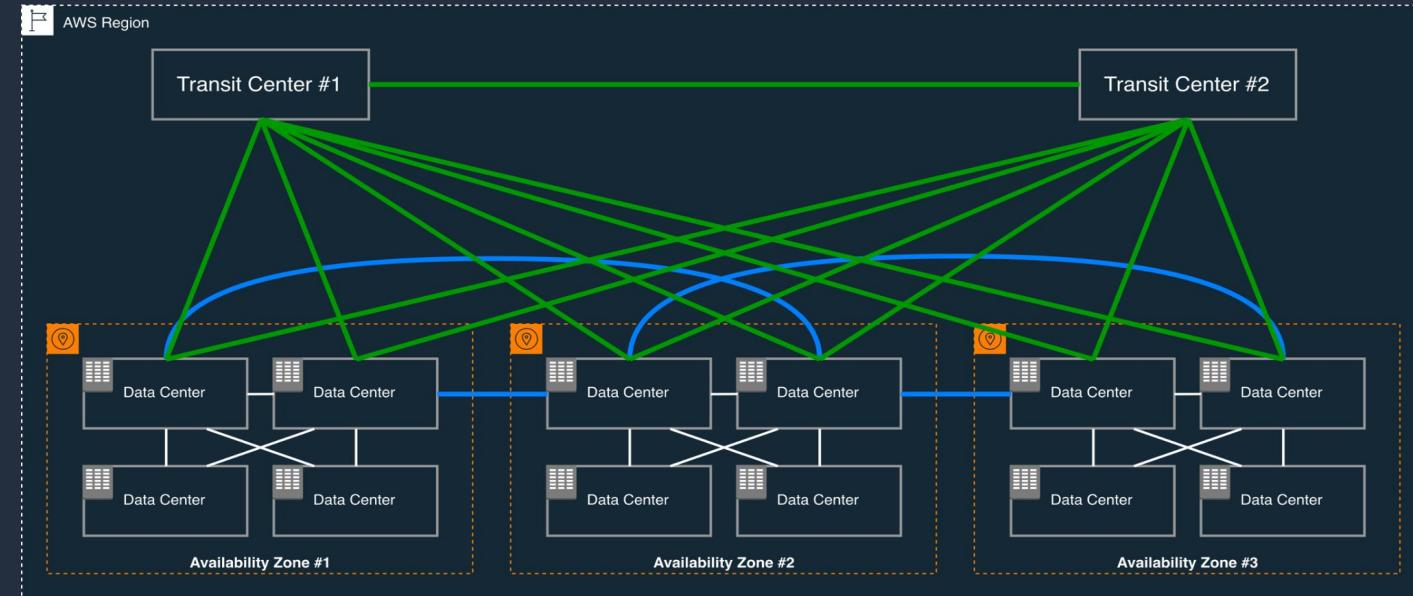
Más de 300 ubicaciones de borde y 13 cachés de borde regionales

# Infraestructura Global: Regiones

- Ubicación física en todo el mundo donde agrupamos los centros de datos.
- Consta de varias zonas de disponibilidad aisladas y separadas físicamente dentro de un área geográfica.
- Las regiones de infraestructura de AWS cumplen con los niveles más altos de seguridad, cumplimiento y protección de datos.
- Dependiendo de la región , será cuántos servicios de AWS estén disponibles, no todos los servicios están en todas las regiones.
- El caso de uso es para recuperación de desastres DRP.

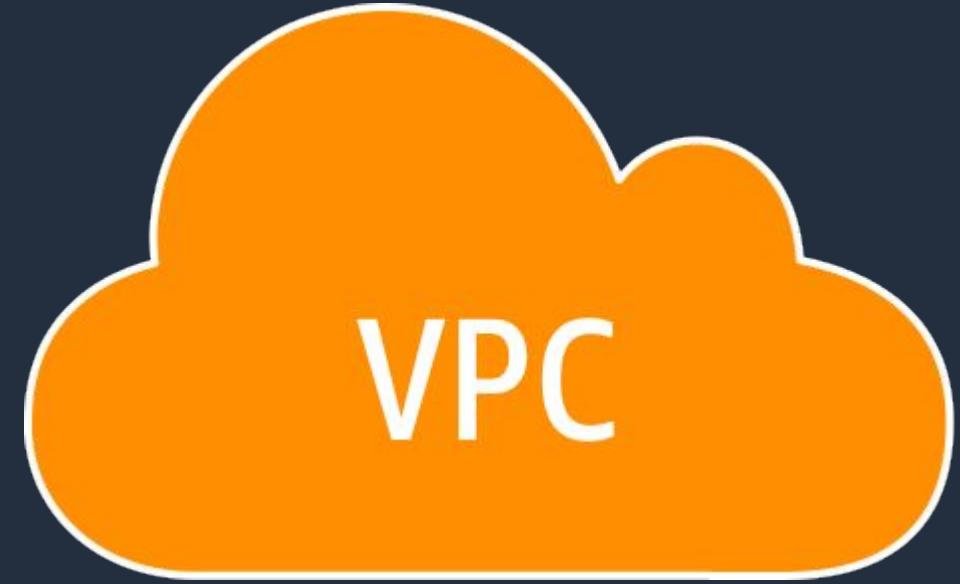
# Infraestructura Global: Zonas de disponibilidad

- Una zona de disponibilidad (AZ) es uno o más centros de datos discretos con alimentación, redes y conectividad redundantes en una región de AWS.
- Todas las zonas de disponibilidad en una región de AWS están interconectadas con redes de alto ancho de banda y baja latencia.
- La distancia entre zonas de disponibilidad es de hasta 100 km.
- Todo el tráfico de red entre las zonas de disponibilidad está encriptado.
- El caso de uso es tolerancia a fallos, disponibilidad.



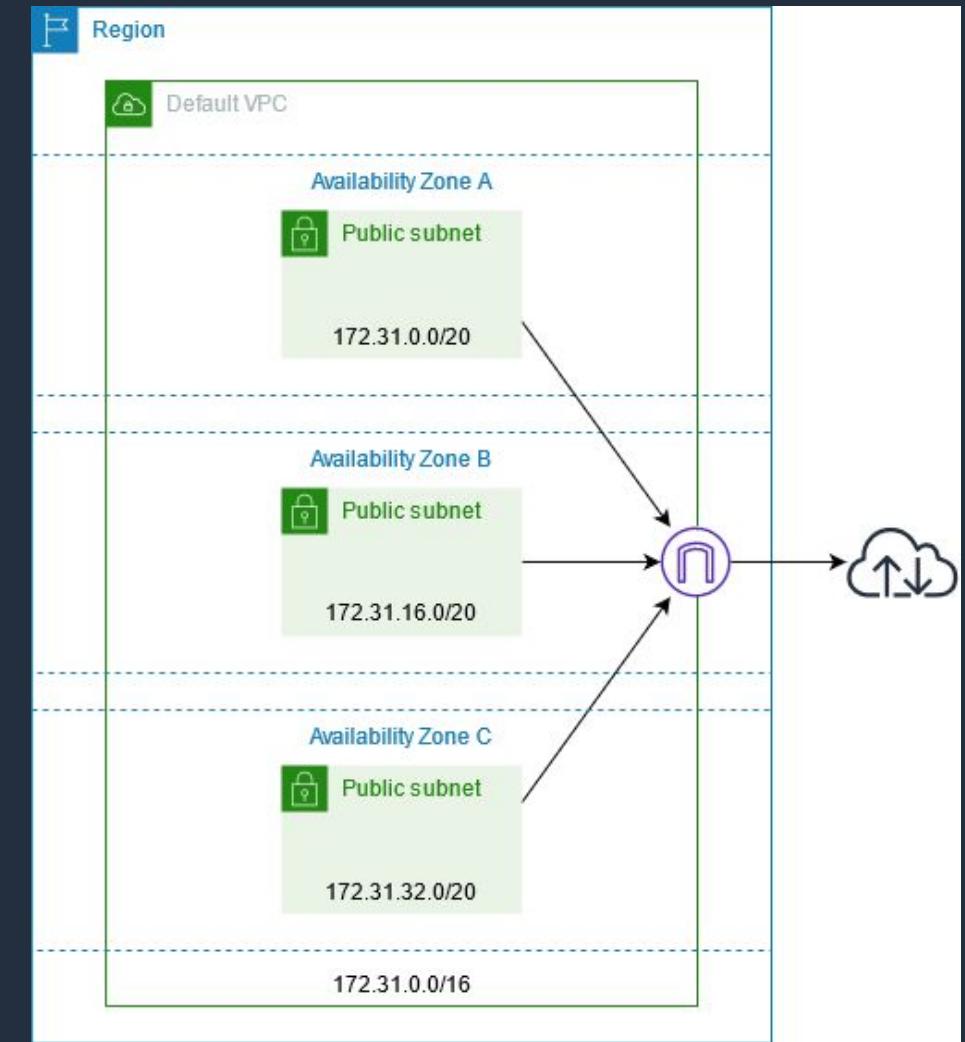
# VPC: Generalidad

- Una sección privada y aislada de la nube de AWS.
- Se asemeja a una red tradicional en su propio centro de datos.
- Todas las cuentas tienen una VPC por default.
- Ofrecen cumplimiento de PCI.
- Ofrecen una capa de seguridad a través de SG y NACL.
- Soportan IPV4 y IPV6.
- Soportan VPN con IPSec.
- Soportan Subneteo.
- CIDR una vez creado no puede modificarse.
- Conexiones dedicadas con Direct Connect.



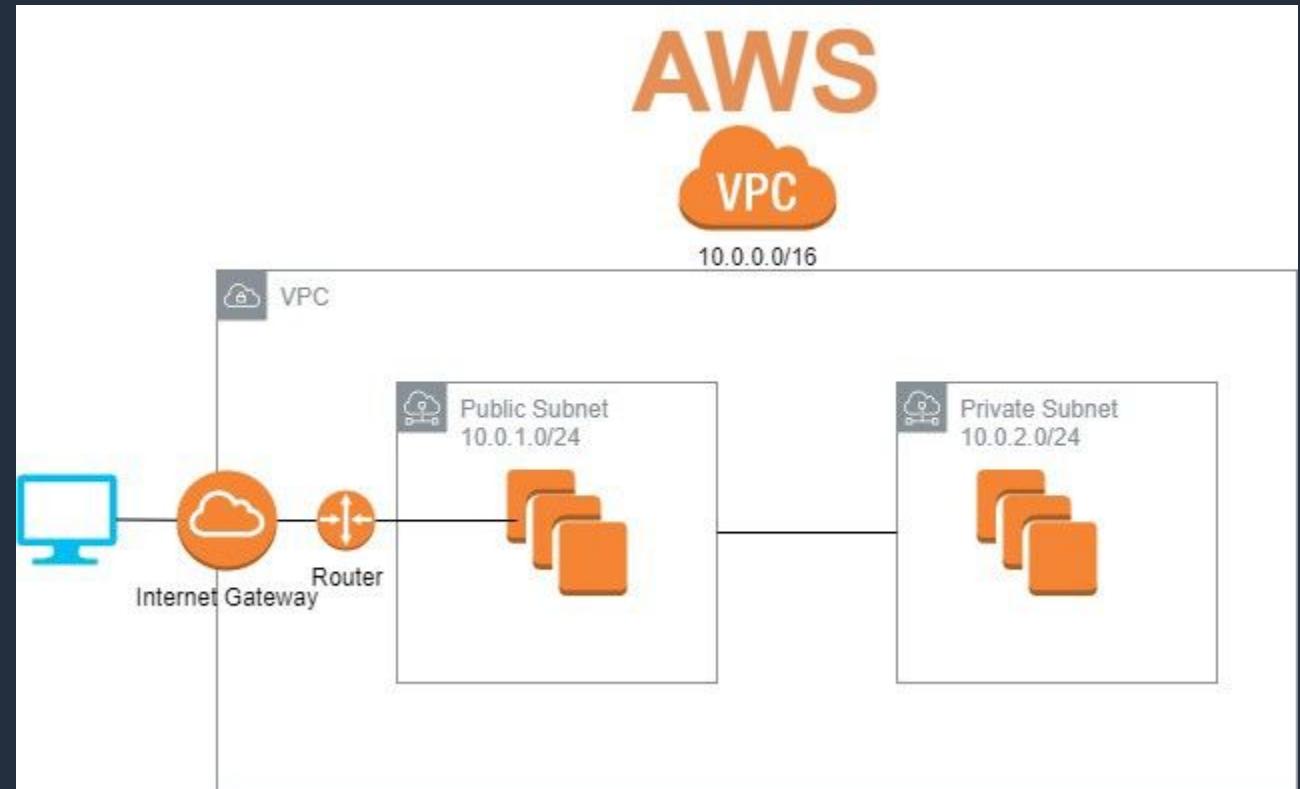
# VPC: Como Crearla y consideraciones

- Una VPC predeterminada viene con una subred pública en cada zona de disponibilidad, una puerta de enlace de Internet y configuraciones para habilitar la resolución de DNS.
- Con la opción DHCP Options permite utilizar un servidor DHCP dentro de la VPC.
- Tenemos la capacidad de monitorear el tráfico con servicios como VPC Flow Logs.
- Podemos utilizar el asistente o bien crear la VPC de manera manual.



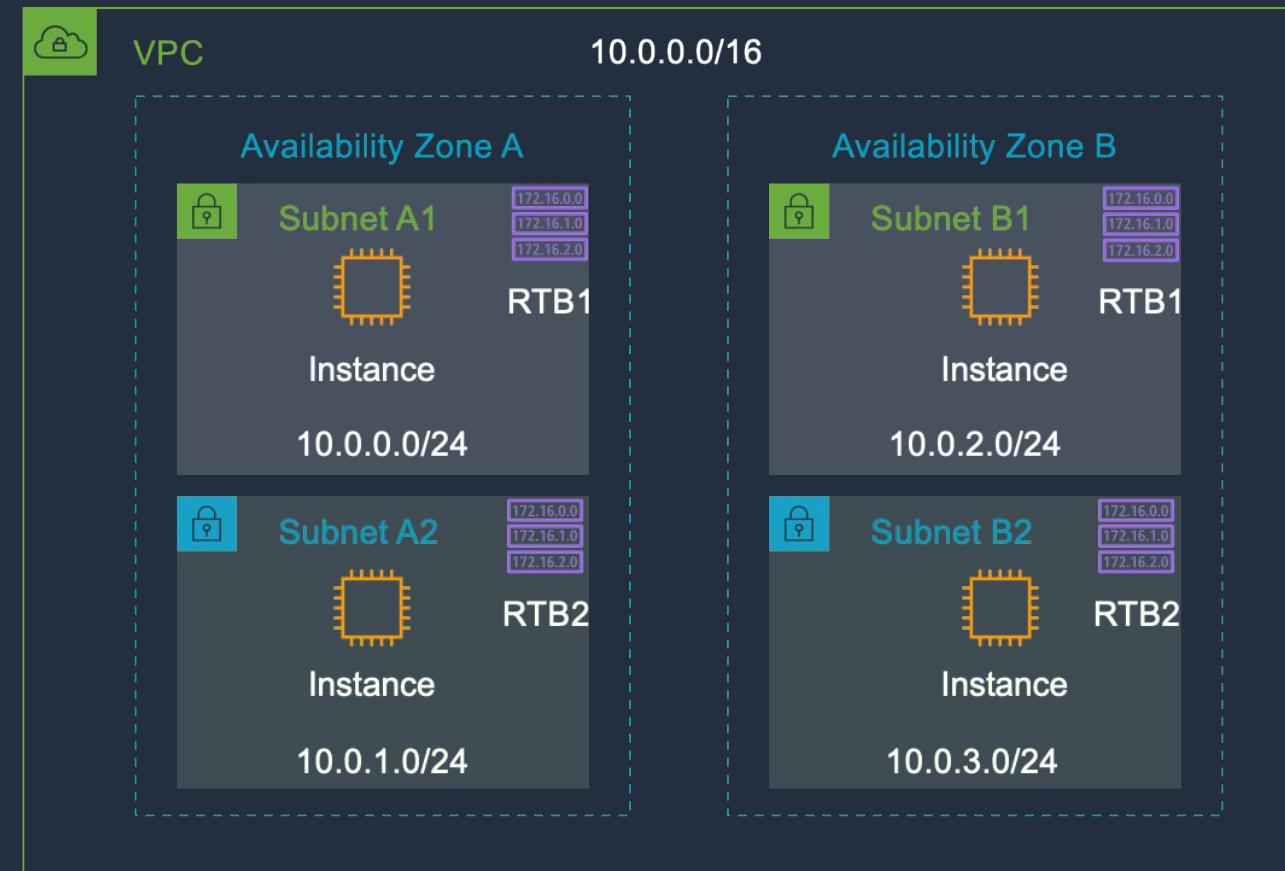
# VPC: Creando una VPC

- Debemos seleccionar el tamaño de la VPC realizando subneteo si aplica.
- Si no utilizamos el asistente deberemos crear todo nosotros mismos.
- Internet Gateway.
- Tablas de ruteo o routing table.



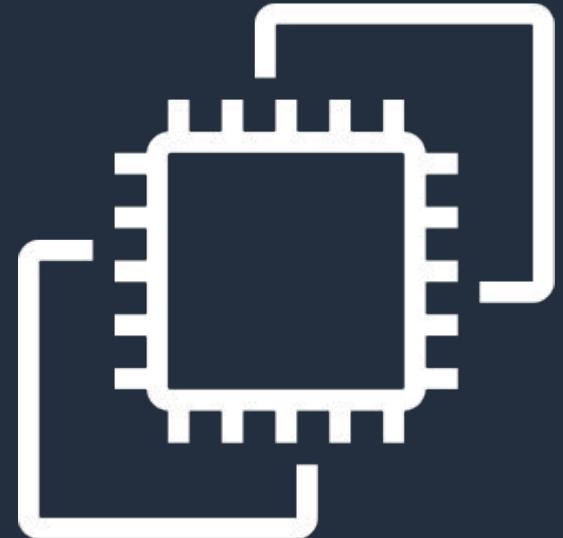
# VPC: Tablas de ruteo

- Cada subnet está asociada a un RT.
- Una RT puede estar asociada a múltiples subnet.
- Las RT puede direccionar tráfico entre:
  - IG / Nat Gateways.
  - VPC Peering / AWS Transit Gateway
  - VPN Gateway / Direct Connect



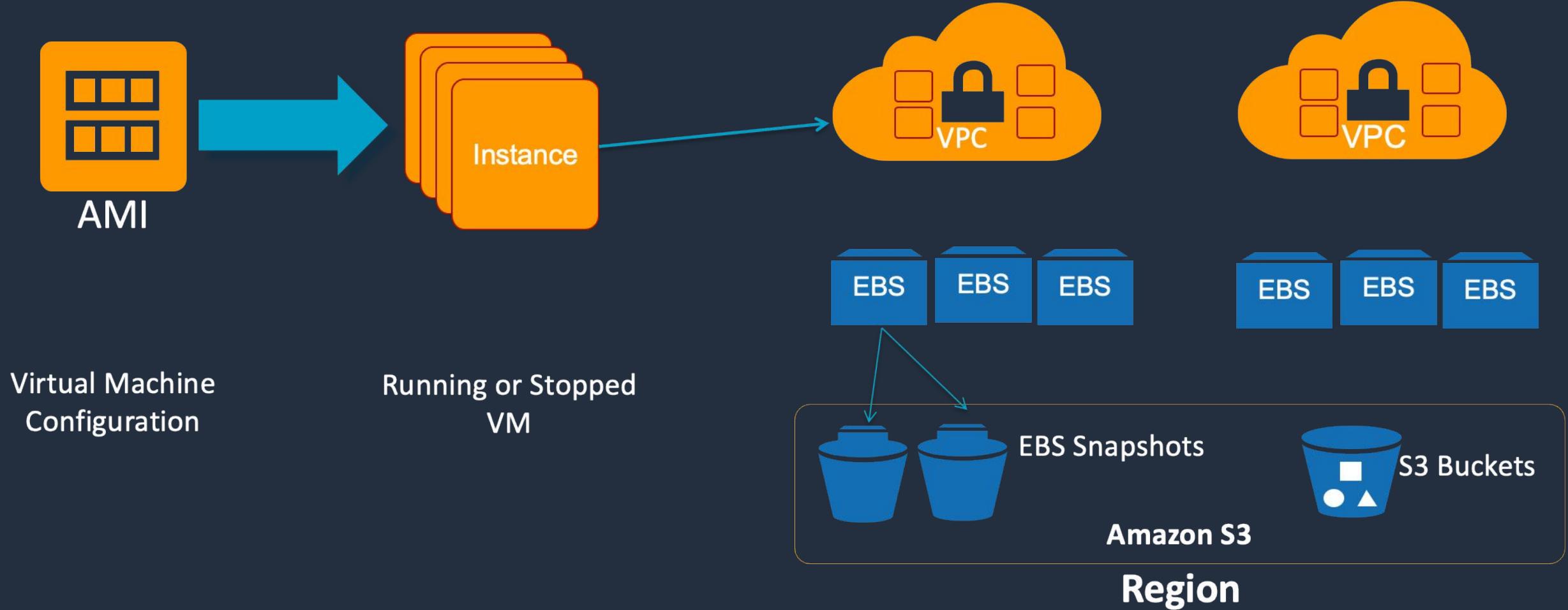
# EC2: Generalidad

- Soportan Linux, Windows y Mac.
- Arquitecturas ARM y x86
- Contamos con instancias de uso general o cómputo especializado de acuerdo al caso de uso.
- Cluster de instancia utilizando [placement Group](#)
  - Partition placement groups
  - Spread placement groups.
- Se ofrecen diferentes modelos de compras, bajo demanda, reservadas y Spot.



Amazon EC2

# EC2: Generalidad



# EC2: Nombre de un tipo de instancia

Instance generation

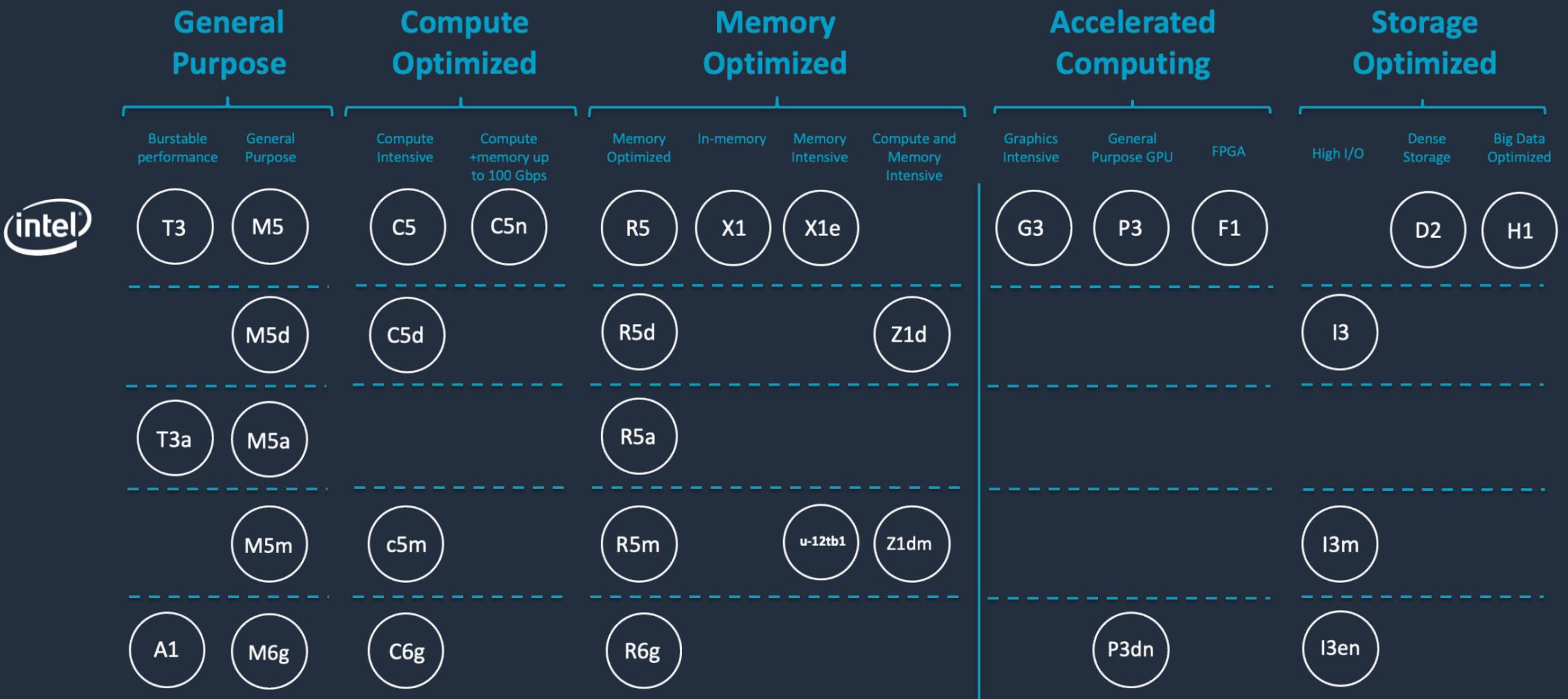
c5n.xlarge

Instance  
family

Attribute

Instance size

# EC2: Tipo de instancia

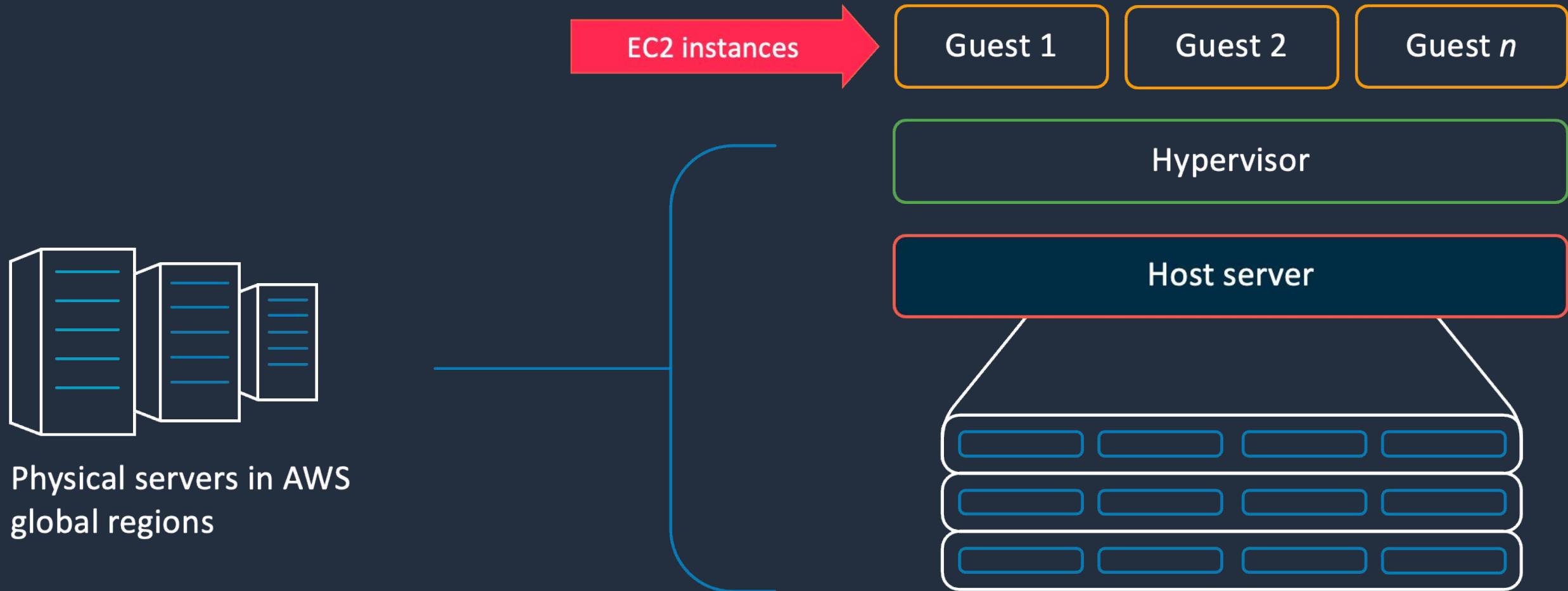


# EC2: Sistemas operativos soportados

- Windows 2003R2\*/2008\*/2008R2\*/2012/2012R2/2016/2019
- Amazon Linux
- Debian
- Suse
- CentOS
- Red Hat Enterprise Linux
- Ubuntu



# EC2: Arquitectura Virtualización EC2



# EC2: Arquitectura Virtualización EC2

- Arquitectura tradicional de virtualización : Xen-based
  - Hypervisor consume recursos del host
  - Optimización limitada.
- AWS Nitro Hypervisor: Basado en hypervisor KVM optimizado.
  - AWS Nitro System (lanzado Nov 2017)
  - Se utilizan menos recursos del servidor, así como recursos disponibles para uso del cliente.
  - Optimizado por AWS.
- Bare metal: Acceso directo al procesador y memoria processor
  - AWS Nitro system
  - Habilita la capacidad hypervisors y micro-VM runtimes.

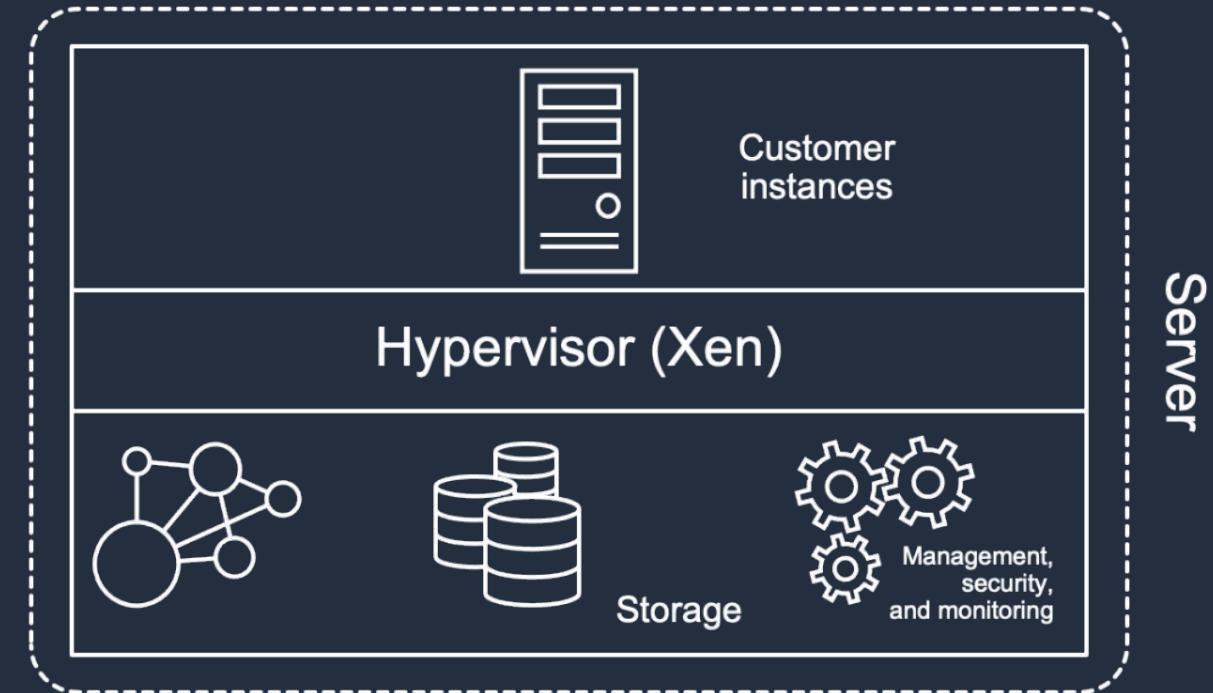
# EC2: Arquitectura Virtualización tradicional

Original EC2 host architecture

All resources were on the server

Instance goals

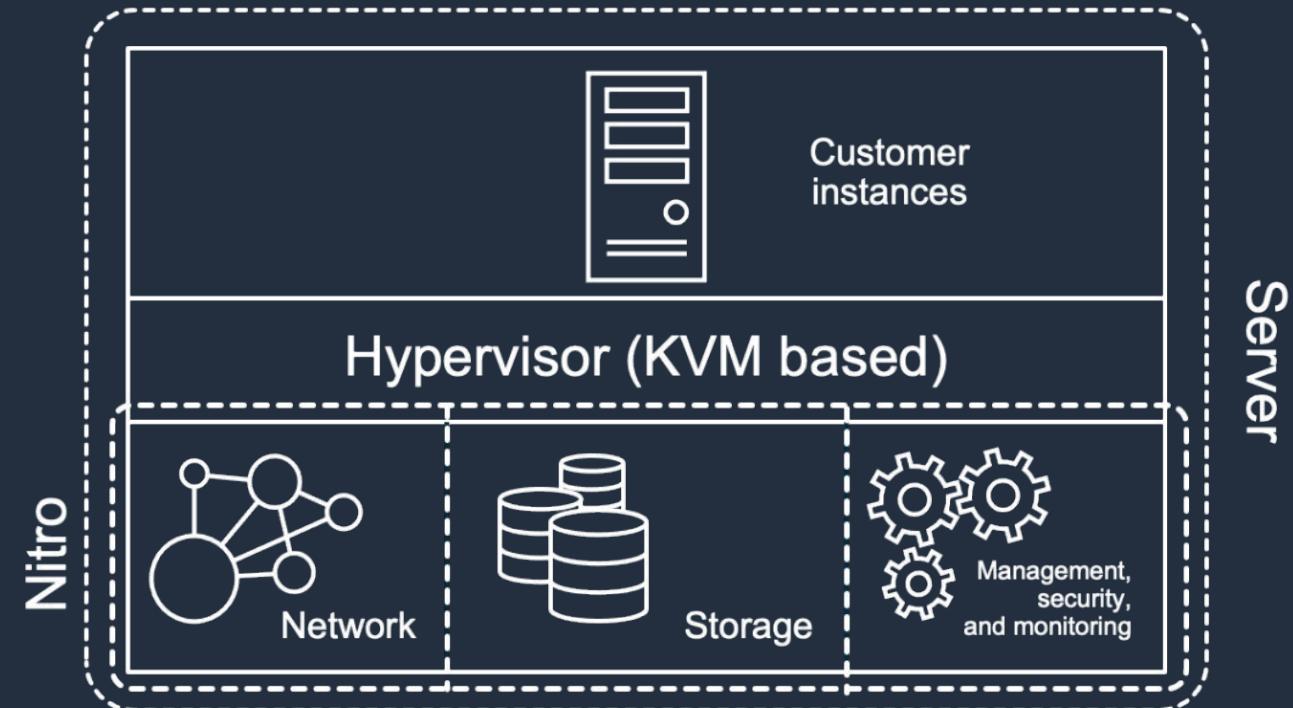
- Security
- Performance
- Familiarity



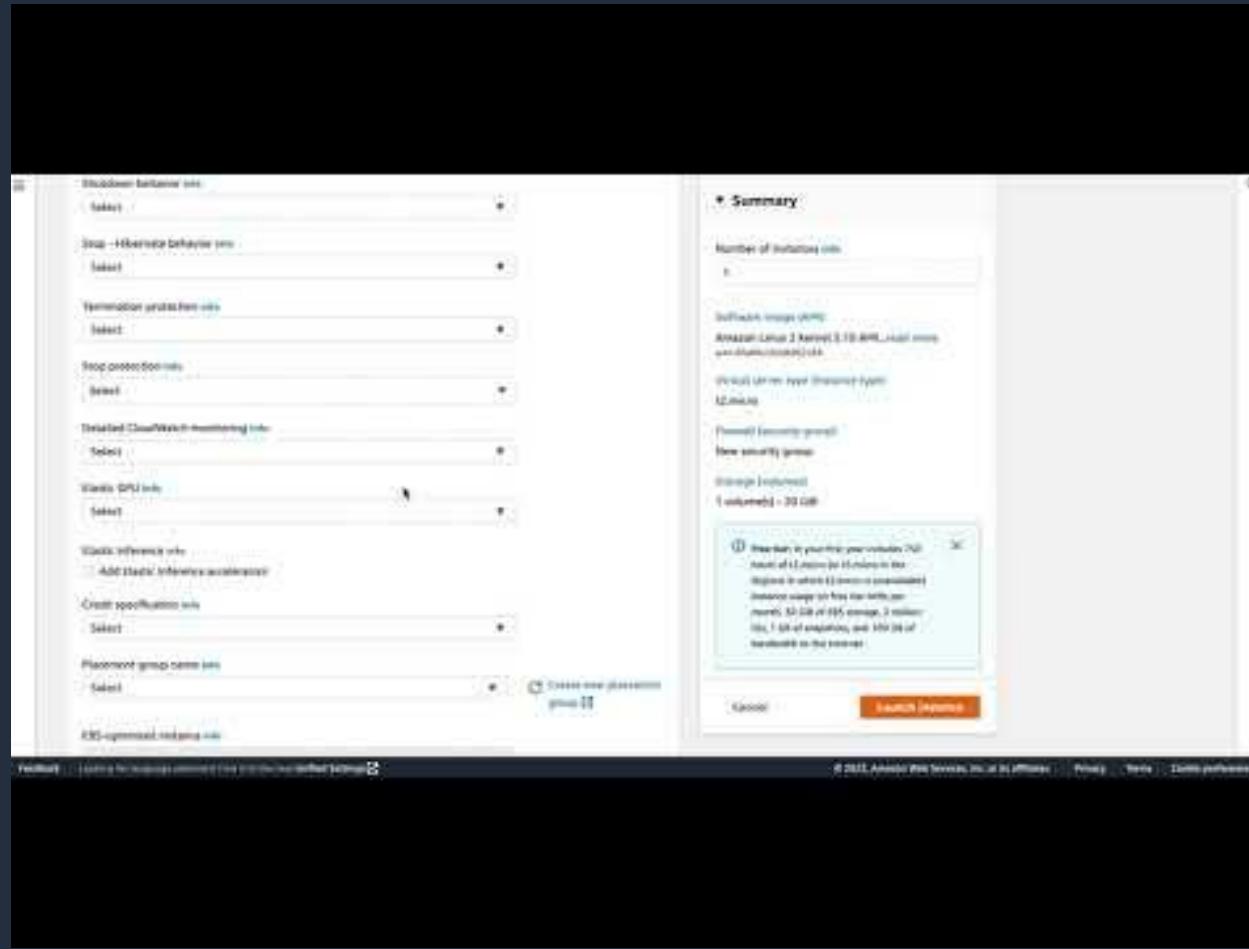
# EC2: Arquitectura Virtualización Nitro

Nearly 100% of available compute resources available to customers' workload

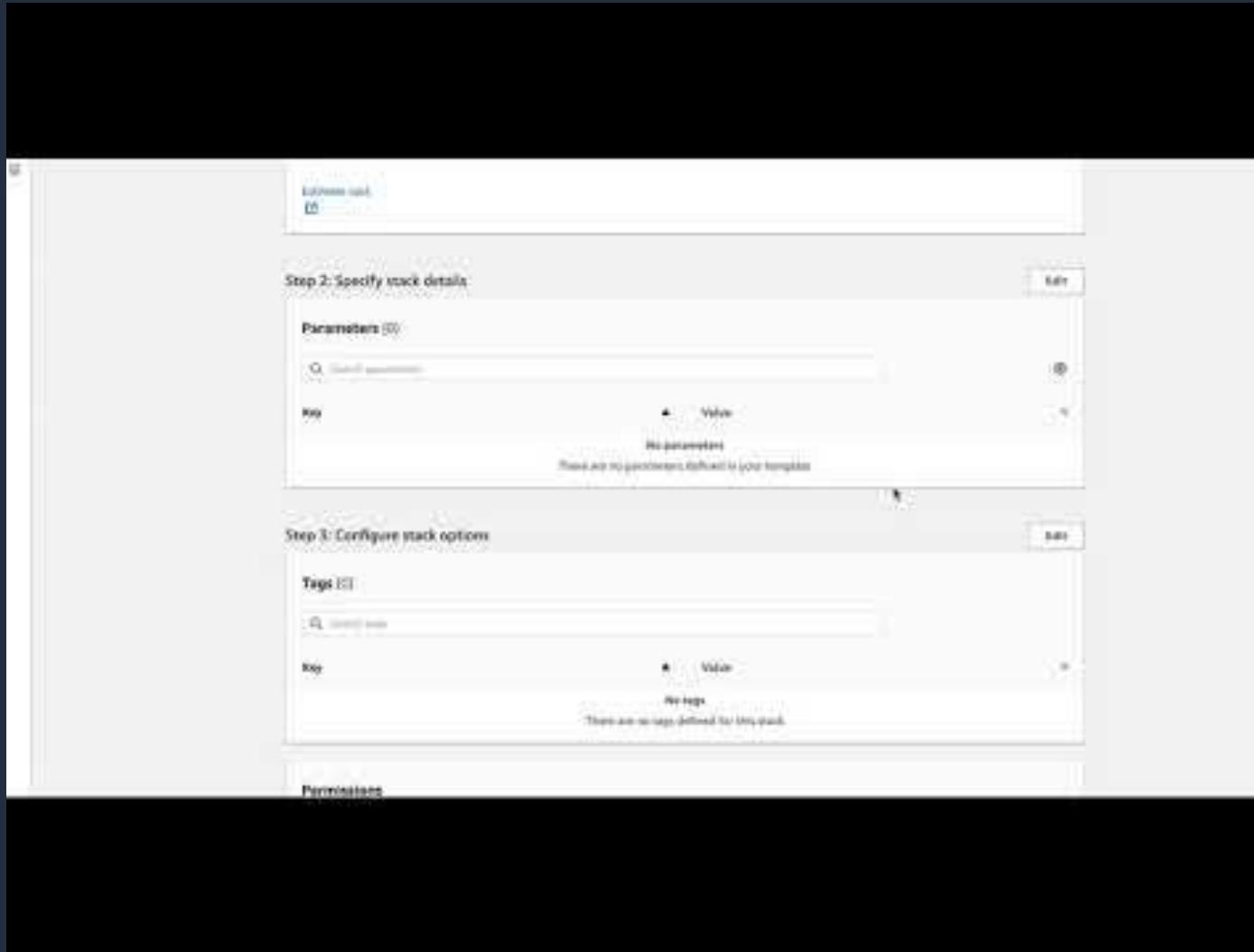
Improved security



## EC2: Como crear una instancia desde consola

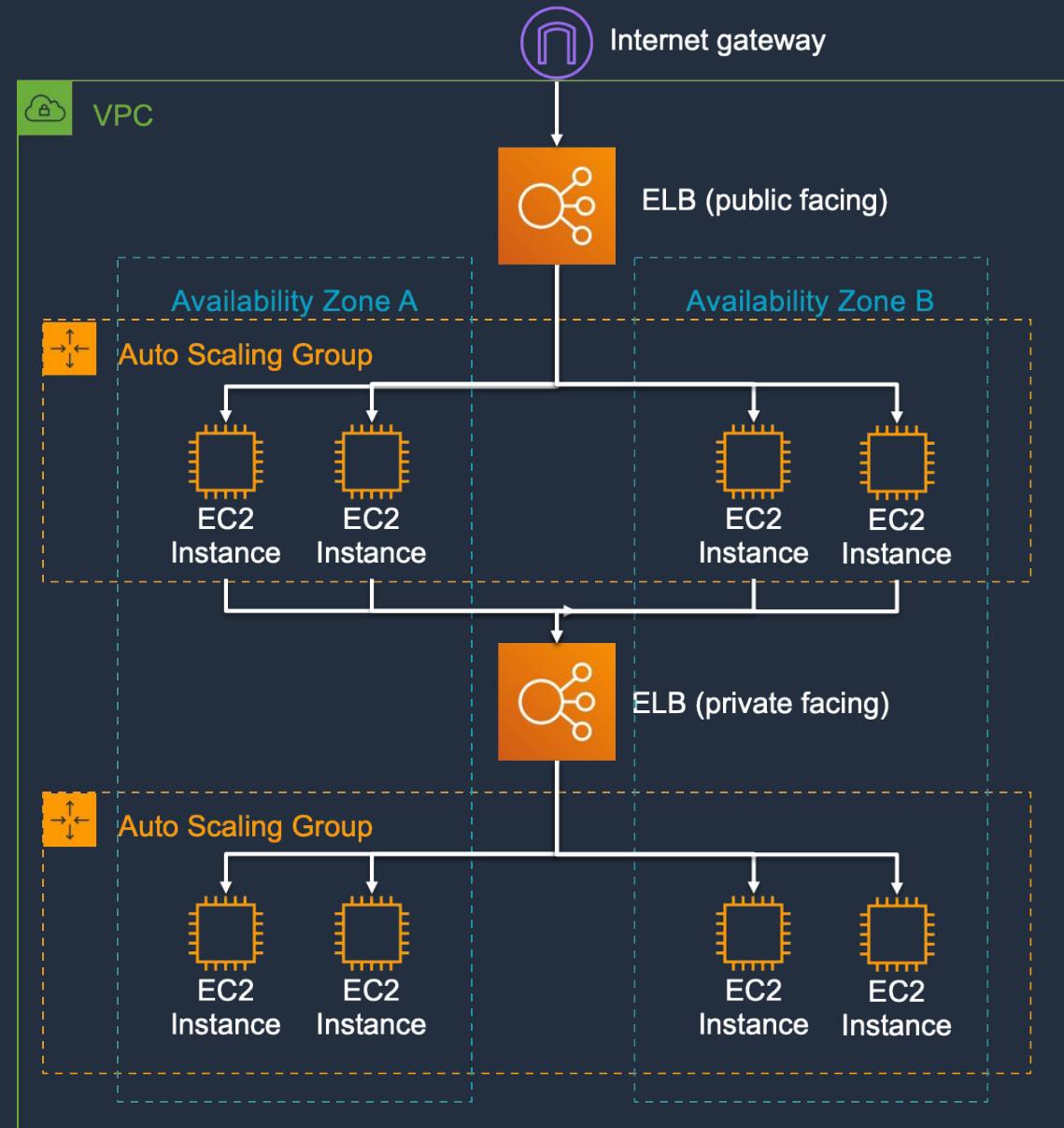


# EC2: Como crear una instancia desde laC



# ELB: Elastic Load Balancing

- Elasticidad y disponibilidad.
- Escalado horizontal.
  - Distribuir el tráfico ( EC2, Contenedores , IP).
  - Brinda mayor elasticidad.
  - Es más costo eficiente.
- Es altamente disponible, distribuyendo tráfico en una AZ.
- ELB escala automáticamente.
- Soporta ASG con EC2.
- Soporta protocolos seguros.



# ELB: Tipos de balanceadores

## NLB

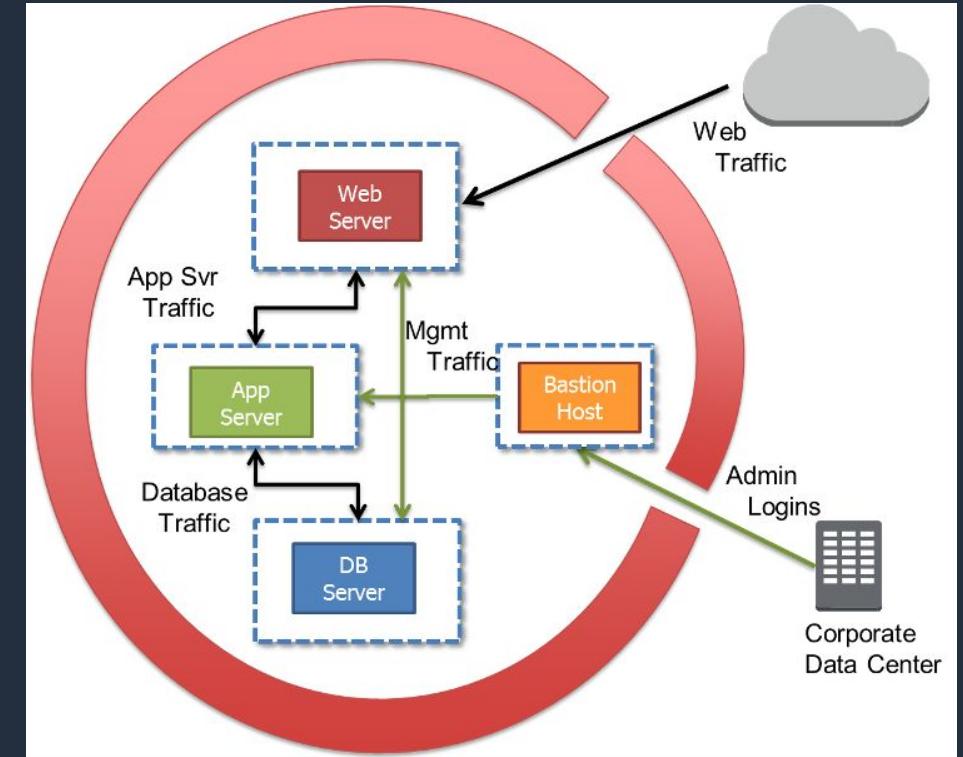
- Balanceador de Capa 4.
- Alto Throughput.
- Baja latencia.
- Mantiene la IP de origen.
- IPs estatica.
- Conexiones TCP larga duracion.
- Direcciones IP como destino.
- Protocolos soportados TCP, UDP, TLS.
- Sesiones persistentes.

## ALB

- Balanceador de Capa 7.
- Enrutamiento basado en contenido (host y ruta de aplicación)
- Soporta aplicaciones en contenedores (ECS, EKS).
- Soporte HTTP/2.
- Request Tracing
- Integracion con Web Application Firewall (WAF) .
- Protocolos soportados HTTP, HTTPS, gRPC.
- Sesiones persistentes.

# EC2: Security Group

- Capa de seguridad.
  - Mantienen estado.
  - Filtran IP, Puerto, Rango, Protocolo, Origen, destino
  - Estructura simple.
  - Filtran tráfico entrante ( reglas de Allow ).
  - Nivel de filtrado.
  - Servicios compatibles.
    - Instancias.
    - ECS.
    - EKS.
    - RDS
    - Computo en general.

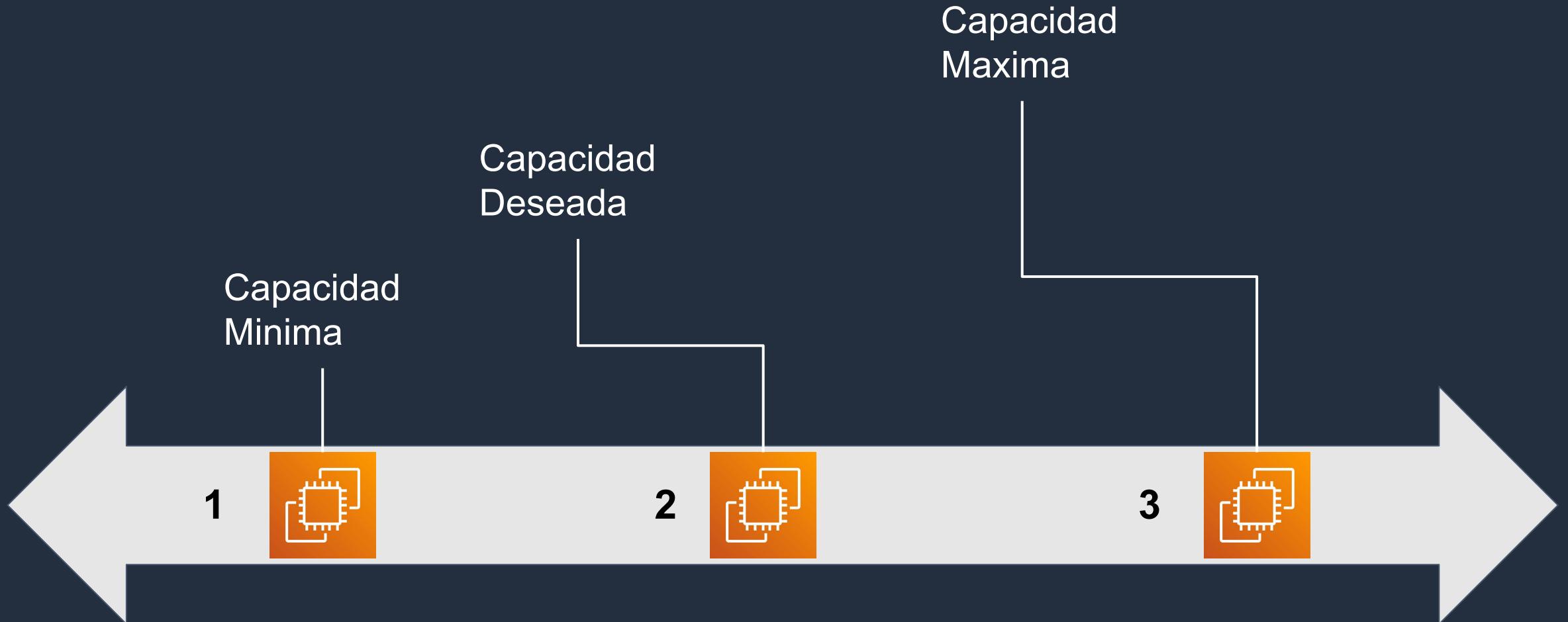


# EC2: ASG ( Grupos de autoescalamiento )

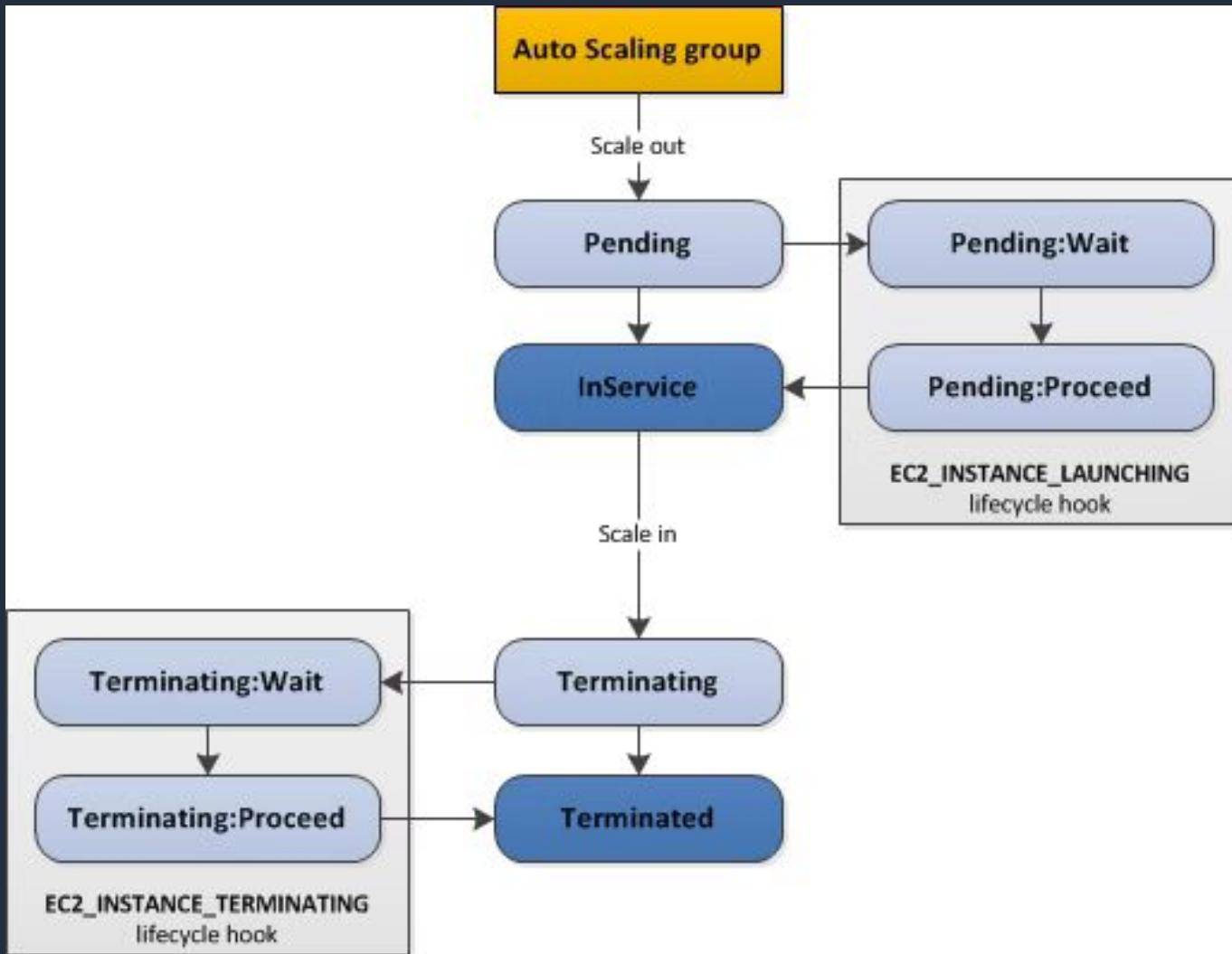
- Herramienta principal que fortalece la elasticidad.
- Tipo de escalamiento.
  - Horizontal.
  - Vertical.
- Como funciona ?
  - Basado en políticas.
  - Estrategias de capacidades.
  - Garantiza rendimiento y ROI.



# EC2: ASG ( Grupos de autoescalamiento )



# EC2: ASG ( Grupos de autoescalamiento )



# Almacenamientos: S3

- Sistema de almacenamiento de archivos.
- Altamente durable 9,99999999999.
- Altamente disponible.
- Altamente escalable ( capacidad infinita de escalamiento ).
- Diversos casos de uso como:
  - Sitios web estáticos.
  - Copias de seguridad y restauración.
  - Almacenamiento de logs.
  - Sistema de almacenamiento en soluciones de Big Data.
  - Sistema de DRP.
  - Sistema de almacenamiento de archivos históricos.
- Bajo costo.
- Servicio Regional.
- Soporta cifrado tránsito y reposo.



AWS S3

# S3 : Características

- Los objetos son almacenados en directorios llamados Buckets.
- El nombre de estos Buckets Global.
- Convención para el nombrado:
  - No es posible utilizar mayúsculas.
  - No es posible utilizar guiones.
  - Longitud entre 3 y 36 caracteres.
  - No se puede utilizar una ip.
  - Debe comenzar con un número o una letra en minúscula.
- Los objetos almacenados pueden ser cualquier tipo de archivo su nombre completo se forma:
  - Prefijo + Nombre del objeto.
  - S3://bucketS3/archivo.txt
- Tamaño max por objeto 5 TB



**AWS S3**

# S3: Policy

- Buckets Policy : Permisos a peticiones que vienen de otras cuentas y acceso público.
- ACL : Puede ser a nivel de objeto o buckets y permite tener un mayor detalle de los permisos.
- IAM Policy : Son políticas que se aplican a un usuario basado en un principal IAM User.
  - Caso donde un usuario necesita acceder a un buckets se da permisos al usuario.
  - Casos donde un recurso necesita acceder a un bucket se crea un rol para el recurso y se asigna permisos al rol en la política de IAM.
- Son documentos JSON y se puede utilizar el policy builder de AWS.



AWS S3

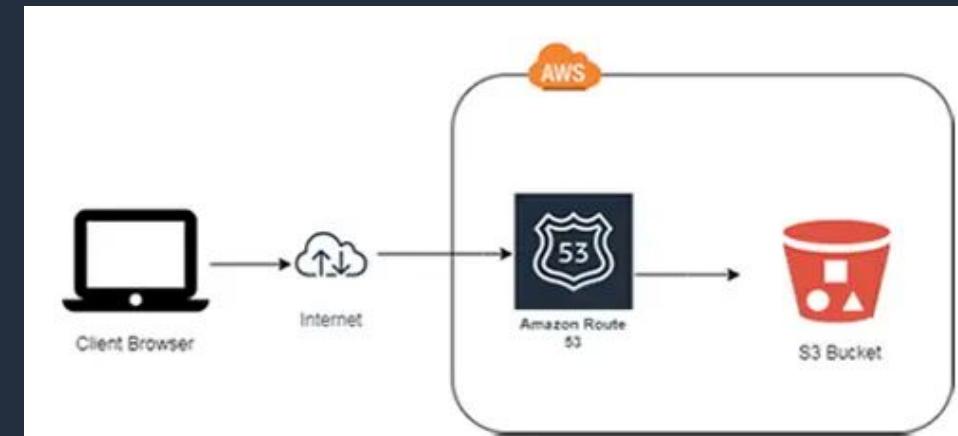
# S3: Policy

- Resources: Buckets y objetos.
- Actions: Acciones específicas.
- Effect: Allow / Deny.
- Principal: Cuenta , usuario o recurso que se aplica la política.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "eks>ListClusters",  
                "eks>CreateCluster"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Sid": "VisualEditor1",  
            "Effect": "Allow",  
            "Action": "eks:*",  
            "Resource": "arn:aws:eks:*:*:cluster/*"  
        }  
    ]  
}
```

# S3: WebSite

- Es una configuración que permite publicar sitios web estáticos.
- La URL por default está compuesta
  - NombreBucket.s3-website<region>.amazonaws.com
- El bucket debe ser público sino arrojará un error 404.
- Luego cargar la información al bucket en la propiedades.
  - Habilitar la opción como sitio web estático.
  - Indicar cuál será el index.html.
  - Indicar cuál será el error.html.
- Para utilizar DNS es posible usar un proveedor propio o utilizar Route53.



# S3: Versionado, Access Logging

- **Versionando**

- Es una buena práctica tener versiones de objetos y evitar indisponibilidad.
- Se habilita a nivel de bucket.
- Genera un costo adicional en almacenamiento.
- No está habilitado por default.
- Si se suspende no se borran versiones anteriores.



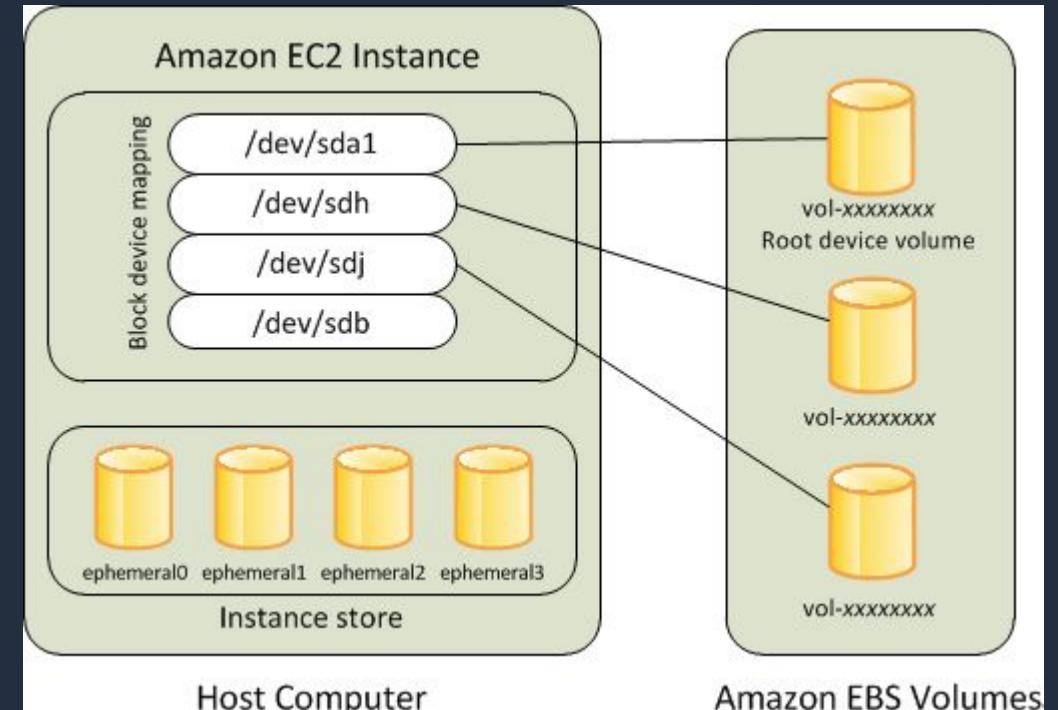
- **Access logging**

AWS S3

- Se utiliza para procesos de auditoría, registra toda la actividad en el bucket.
- Ayuda en la identificación de causa raíz.
- Se debe crear primeramente otro bucket donde se almacenarán los registros.
- Cuando se habilita se debe indicar el bucket donde se almacenarán los registros.

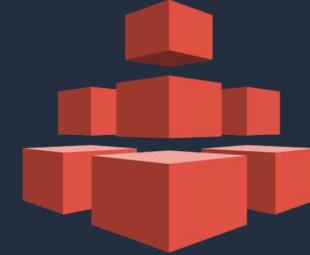
# Almacenamientos: EBS Volume

- Dispositivos para almacenamiento persistente.
- Permite conservar la información aun con la terminación de la instancia.
- Solo se puede montar un EBS por instancia.
- Se asocia a una AZ.
- La capa gratuita incluye un volumen de hasta 30 GB.
- Es un volumen de red no físico.
- Se puede proteger contra borrado cuando son creados desde EC2.
- Root tiene habilitado el borrado del volumen por defecto.
- Soportan snapshots.
- Se pueden copiar entre regiones los snapshots.



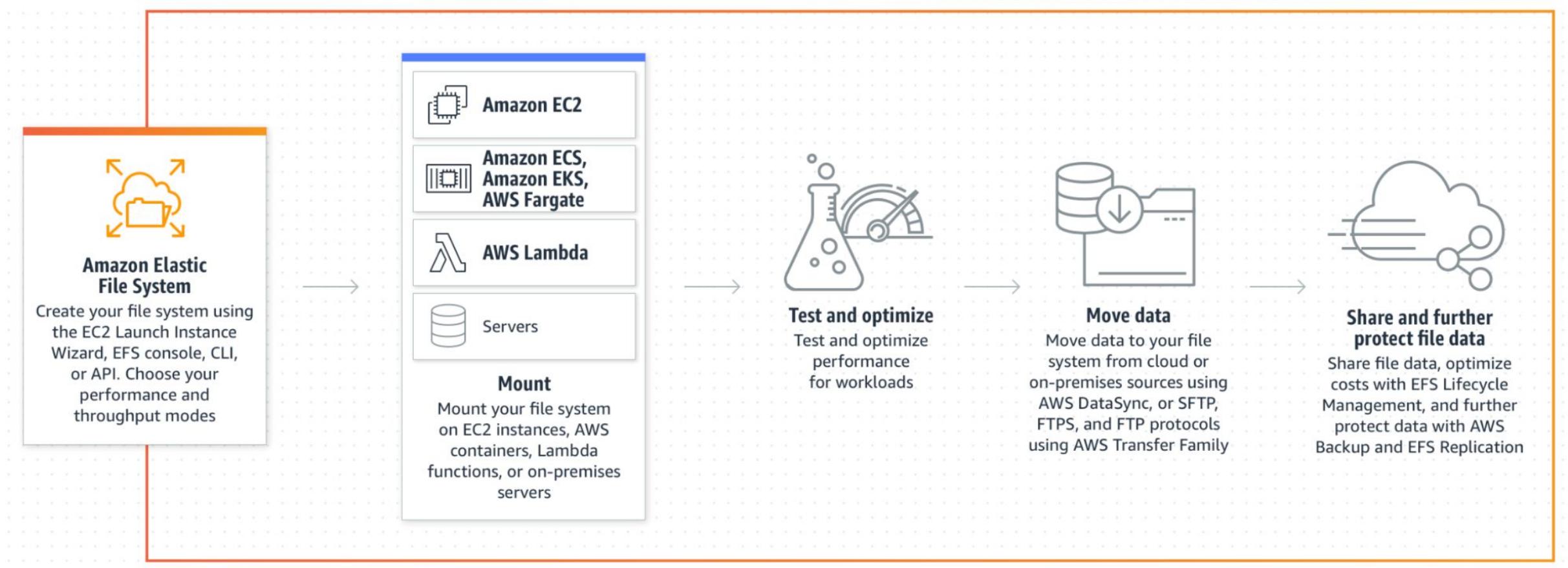
# Almacenamientos: EFS

- Sistema de archivo basado en NFS.
- Compatible EC2 Linux dentro de un multi AZ.
- Pago por uso no es necesario aprovisionar capacidad.
- Admite sincronización entre regiones.
- Tiene una capacidad de Acceso poco frecuente como S3.
  - Basado en políticas de acceso.
  - Mueve automáticamente los archivos de una zona a otra.
  - Permite un ahorro de hasta el 90% sobre la versión estándar.



**AWS EFS**

# Almacenamientos: EFS





# Preguntas

# Dia 2

→ Entendiendo que sucede dentro de la Nube

# Monitoreo: Cloudwatch

- Plataforma nativa AWS monitoreo y observabilidad.
- Los Amazon CloudWatch Logs permiten utilizar los logs personalizados, del sistema y de las aplicaciones de los que ya dispone para monitorizar los sistemas e identificar los problemas que surjan en sistemas y aplicaciones.
- Métricas detalladas.
- Periodo de retención desde 14 días hasta 15 meses.
  - Datos con un período menor que 60 segundos están disponibles durante 3 horas. Son métricas personalizadas de alta resolución.
  - Datos con un período de 60 segundos disponibles 15 días
  - Datos con un período de 300 segundos, disponibles durante 63 días
  - Datos con un período de 3600 segundos (1 hora) están disponibles durante 455 días (15 meses).



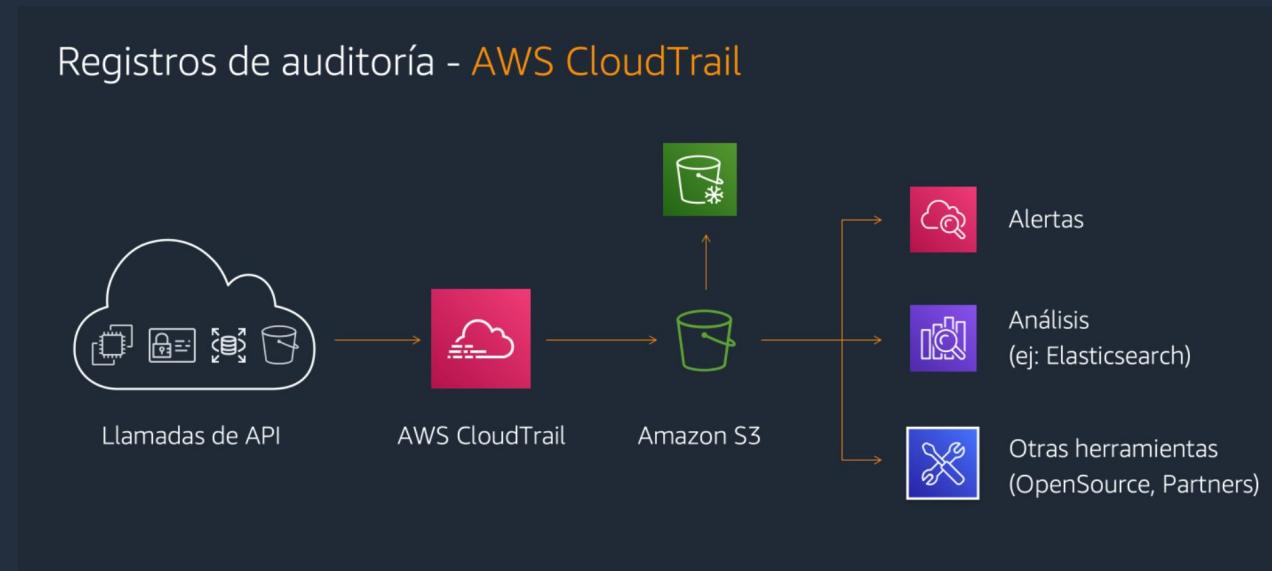
# Cloudwatch : Ventajas

- Monitoreo sus recursos AWS en tiempo real.
- CloudWatch para realizar un seguimiento de las métricas integradas y personalizadas.
- CloudWatch Alarms para implementar condiciones que permitan a su solución reaccionar a los cambios.
- CloudWatch Events para automatizar la infraestructura.
- CloudWatch Logs para colectar logs , agregación y monitoreo.
- Tableros con vistas personalizables para ver los diferentes recursos de la infraestructura.



# Monitoreo: Cloudtrail

- Servicio orientado a hacia soluciones que impacten la seguridad.
- seguimiento de la actividad de los usuarios y el uso de las API.
- Identificar los diferentes eventos relacionados con la seguridad.
- Sirve como apalancador para implementar reglas de cumplimiento.
- Permite encriptar los logs.
- Se integra con Cloudwatch logs.
- Pago por uso.



# Cloudtrail : Funcionamiento

- CloudTrail utiliza estos eventos en tres funciones:
  - **Trails** : Permite la entrega y el almacenamiento de eventos en Amazon S3, con entrega opcional a Amazon CloudWatch Logs y Event Bridge ( Cloudwatch Event ).
  - **Insights** : Analiza los eventos del plano de control en busca de comportamientos anómalos en los volúmenes de llamadas de la API.
  - **Event History** : Proporciona un historial de 90 días de acciones del plano de control de forma gratuita. Como parte de sus funciones principales de auditoría, CloudTrail proporciona claves administradas por el cliente para el cifrado y la validación de archivos de registro para garantizar la inmutabilidad.

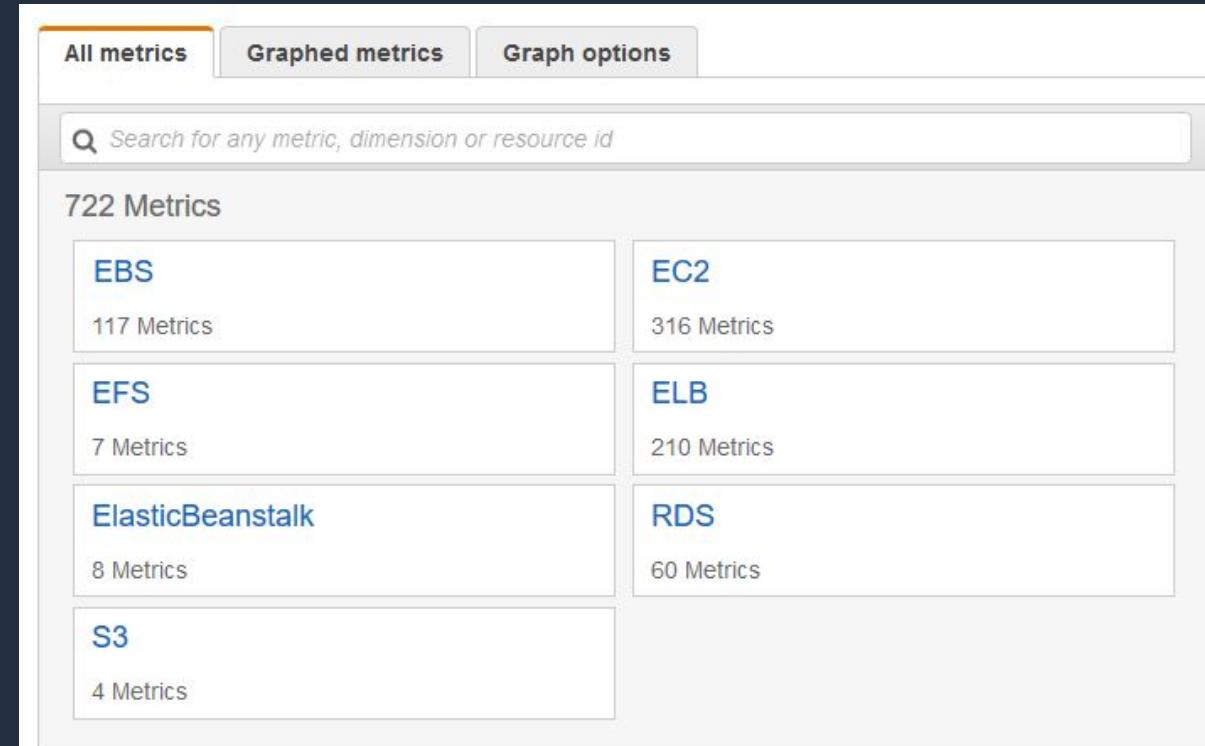
# Cloudwatch: Agente.

- Permite colectar de métricas y registros de instancias de Amazon EC2 y servidores locales con el agente de CloudWatch.
- Instalación descargando el paquete [agent](#)
- Instalacion desde [System Manager](#)
- Configuración de [Agente.](#)
  - Configuración asistida es a través del siguiente comando `amazon-cloudwatch-agent-config-wizard`
  - Se puede modificar la configuración establecida accediendo directamente al archivo. JSON `installation-directory/doc/amazon-cloudwatch-agent-schema.json` en Linux o `installation-directory/amazon-cloudwatch-agent-schema.json` en Windows.



# Monitoreo: Metricas

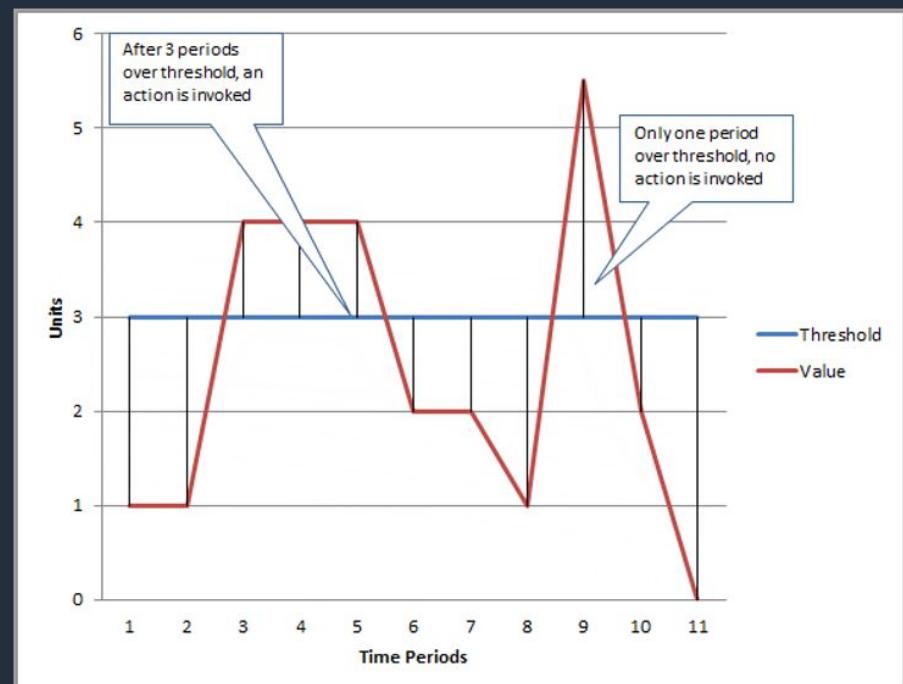
- Existen dos categorías de métricas básicas y detalladas.
  - Básicas : Son métricas no tienen costo, [listado público](#)
  - Detalladas : Solo algunos servicios lo tienen. También genera cargos. Para usarlo para un servicio de AWS, debe activarse.
- CloudWatch Metrics [Insights](#) es un potente motor de consulta SQL de alto rendimiento que puede utilizar para consultar sus métricas a escala. Puede identificar tendencias y patrones dentro de todas sus métricas de CloudWatch en tiempo real.



# Monitoreo: Alarmas, logs, eventos y filtros.

- **Alarma:**

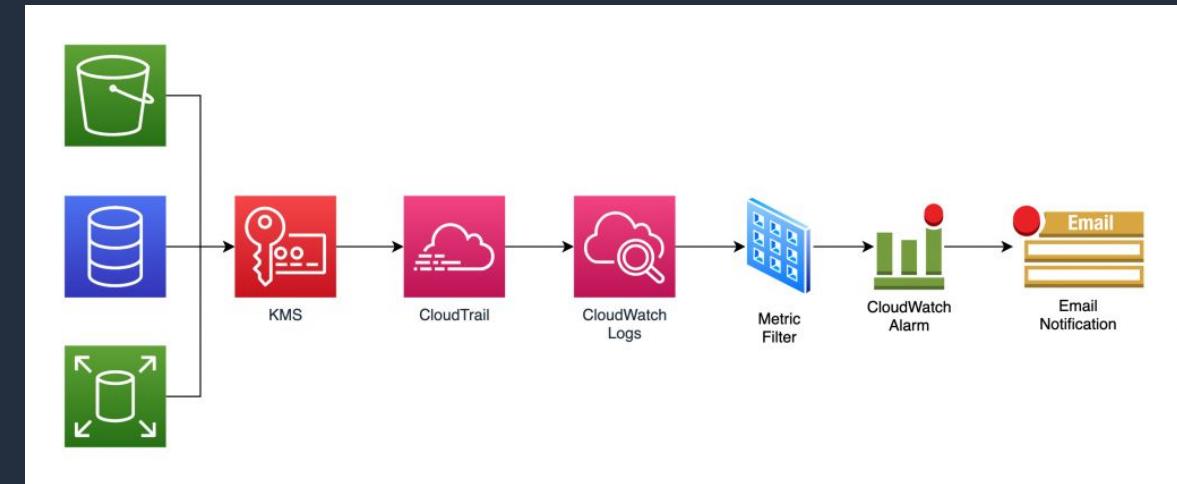
- Envía notificaciones basadas en el estado de los recursos monitoreados.
- Observa la métrica durante un período de tiempo y envía mensajes cuando se alcanza un umbral de métrica definido.
- Se utiliza con Amazon Simple Notification Service (SNS) para enviar mensajes.
- El estado de alarma debe mantenerse durante un número específico de períodos.
- Se integra con Auto Scaling.
- Posibles estados de una Alarma:
  - OK: Está dentro del umbral definido.
  - ALARM: Está fuera del umbral definido
  - INSUFFICIENT\_DATA: La métrica no está disponible o no hay suficientes datos disponibles para que la métrica determine el estado de la alarma.



# Monitoreo: Alarmas, logs, eventos y filtros.

- **Logs:**

- Permite capturar y monitorizar la actividad.
- Servicio administrado para colectar y almacenar los diferentes registros.
- Agregue y centralice registros en múltiples fuentes de información.
- Agente de CloudWatch Logs para instancias de Linux y Windows.
- Integración con métricas y alarmas.
- Exportar los datos a S3 para análisis o archivo.
- Transmita a Amazon ElasticSearch Service o AWS Lambda



# Cloudwatch Logs: Conceptos.

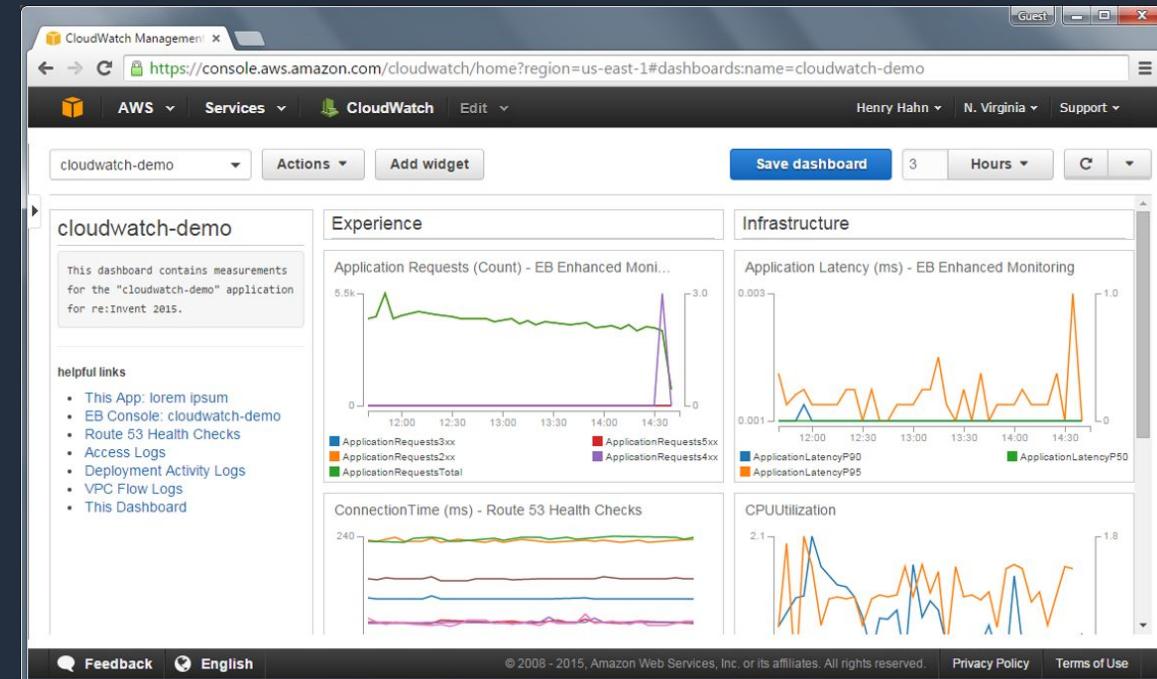
- **Log events:**
  - Un evento es cualquier actividad por el recurso monitoreado, estos tienen dos propiedades importantes, timestamp y mensaje que contiene el evento.
- **Log streams:**
  - Es una secuencia de eventos de registro que comparten la misma fuente, Por ejemplo, un stream puede estar asociado con un registro de acceso de Apache en un host específico.
- **Log groups:**
  - Definen grupos de stream que comparten la misma configuración de retención, supervisión y control de acceso. Por ejemplo, si tiene un stream independiente para los registros de acceso de Apache de cada host, podría agrupar esos stream en un solo log group.
- **Metric filters:**
  - Los filtros se utilizan para extraer información de eventos recopilados y transformarlos en puntos de datos en una métrica de CloudWatch. Se asignan a grupos de registros y todos los filtros asignados a un log group.

# Monitoreo: Alarmas, logs, eventos y filtros.

- **Event:**
  - Actualmente Cloudwatch Event y Event Bridge son el mismo servicio.
  - Ofrece un stream cercano al tiempo real de eventos del sistema que describen cambios en los recursos de Amazon Web Services (AWS).
  - Responde a estos cambios operativos y toma medidas correctivas según sea necesario, enviando mensajes para responder al entorno, activando funciones, realizando cambios y capturando información de estado.
  - El evento indica cuál es el cambio que se realizó en su infraestructura o aplicación.
  - Rule: hace coincidir los eventos entrantes y los enruta a los destinos para su procesamiento. Una regla puede enrutar a varios destinos, todos los cuales se procesan en paralelo.
  - Target: procesa eventos. Los destinos pueden incluir instancias de Amazon EC2, funciones de AWS Lambda, secuencias de Kinesis, tareas de Amazon ECS, máquinas de estado de Step Functions, temas de Amazon SNS, colas de Amazon SQS y destinos integrados. Un objetivo recibe eventos en formato JSON

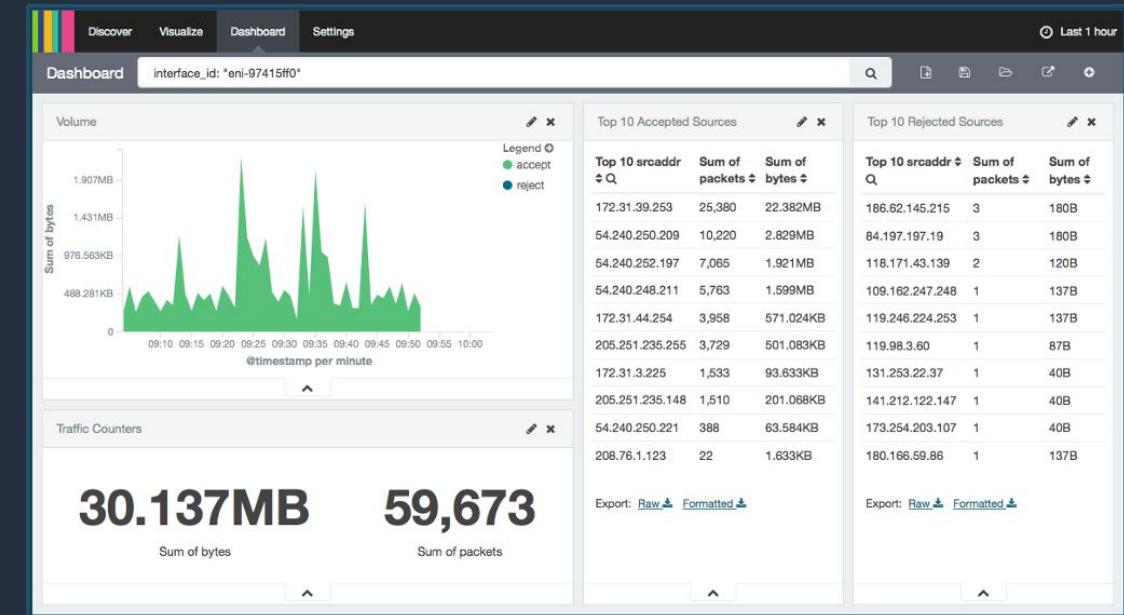
# Monitoreo: Tableros.

- Puede usar para monitorear sus recursos en una sola vista, incluso los recursos que se encuentran distribuidos en diferentes regiones.
- Crear vistas personalizadas de las métricas y alarmas.
- Se crean a partir de widgets que luego se colocan automáticamente.
- Puede crear un dashboard desde la consola, AWS CLI o por llamados a la API PutDashboard.
- Para acceder a los dashboard , necesita uno de los siguientes permisos:
  - AdministratorAccess
  - CloudWatchFullAccess
  - Regla personalizada que incluya los permisos específicos.



# Monitoreo: VPC Flow Logs.

- Permite capturar información sobre el tráfico de IP que va y viene de las interfaces de red en su VPC. Los datos se pueden publicar en Amazon CloudWatch Logs o Amazon S3.
- Se puede habilitar en :
  - VPC, subred o en una interfaz de red.
  - VPC & Subnet habilita el registro para todas las interfaces en la VPC/subred.
  - Cada interfaz de red tiene un flujo de registro único.
- Los Flow logs no capturan en tiempo real para sus interfaces de red.
- Filtrar el resultado deseado según la necesidad.
  - All, Reject, Accept
  - Troubleshooting o seguridad.

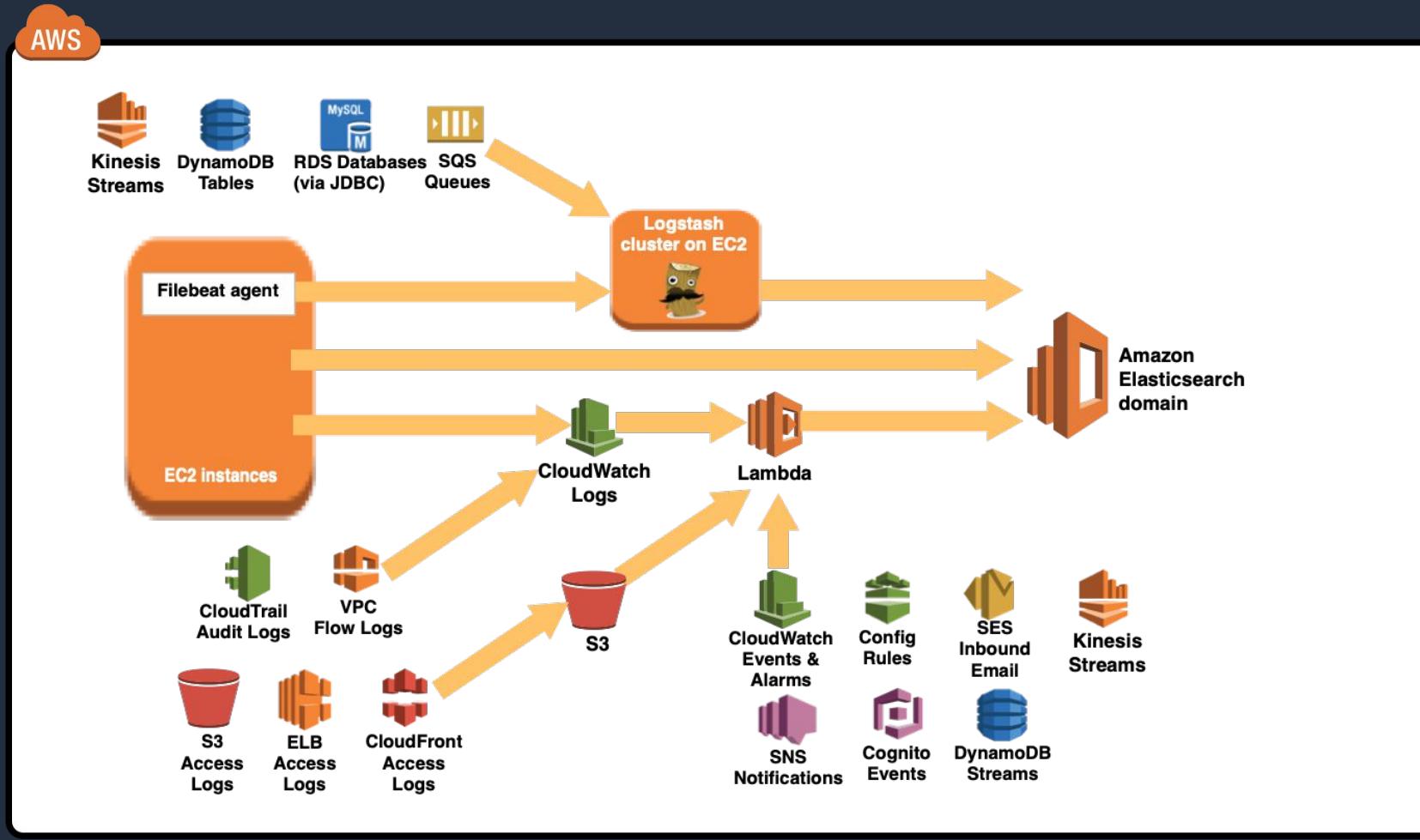


# Monitoreo: VPC Flow Logs.

- Agentless
- Se puede habilitar ENI, subred o VPC
- Registra con AWS CloudWatch Logs
- Crear métricas de CloudWatch a partir de datos de registro.

AWS account	Interface	Source IP	Source port	Protocol	Packets	Bytes	Start/end time
<b>Event Data</b>							
	2 41747	eni-b30b9cd5	119.147.115.32	10.1.1.179	6000 22	6 1 40	1442975475 1442975535 REJECT OK
	2 41747	eni-b30b9cd5	169.54.233.117	10.1.1.179	21188 80	6 1 40	1442975535 1442975595 REJECT OK
	2 41747	eni-b30b9cd5	212.7.209.6	10.1.1.179	3389 3389	6 1 40	1442975596 1442975655 REJECT OK
	2 41747	eni-b30b9cd5	189.134.227.225	10.1.1.179	39664 23	6 2 120	1442975656 1442975716 REJECT OK
	2 41747	eni-b30b9cd5	77.85.113.238	10.1.1.179	0 0 1 1	100	1442975656 1442975716 REJECT OK
	2 41747	eni-b30b9cd5	10.1.1.179	198.60.73.8	512 123	17 1 76	1442975776 1442975836 ACCEPT OK

# Monitoreo: Resumen



# Monitoreo: Costos.

- Es posible monitorear Amazon CloudWatch de forma gratuita. La mayoría de las métricas de ventas de los servicios de AWS (EC2, S3, Kinesis, etc.) están disponibles automáticamente de manera gratuita en CloudWatch.
- Métricas de monitoreo básico (frecuencia de 5 minutos)
- 10 métricas de monitoreo detallado (frecuencia de 1 minuto)
- 1 millón de solicitudes API (no aplicable a GetMetricData ni GetMetricWidgetImage).
- 3 dashboard para hasta 50 métricas al mes.
- 5 GB de datos (incorporación, almacenamiento y archivo, y datos escaneados por las consultas de Logs Insights).
- Se incluyen todos los eventos excepto los personalizados
- Luego los costos dependen del uso de los recursos y la región donde están desplegados.



# Preguntas

# Dia 3

→ Construyendo una solución segura en Nube

# Seguridad : Buenas prácticas

- Mínimo privilegio viable.
- No uso root.
- Separación de ambientes.
  - Separación VPC.
  - Separación por cuentas.
- Gestión de cuentas.
  - AWS Organization.
  - AWS Control Tower.
- Reducción superficie de ataque.
  - AWS SG.
  - AWS NACL.
  - AWS WAF.
  - AWS Shield.
  - AWS Cloudfront.
  - Firewall.
- AAA.
  - Autenticación.
  - Autorización.
  - Auditoria.

# IAM : Usuarios, políticas y grupos.

- IAM es un servicio Global.
- La cuenta Root sólo debe usarse para configurar la cuenta y nunca debería ser compartida.
- Como buena práctica los usuarios con funciones afines deben estar en un grupo.
- Por buena práctica las políticas deberían aplicarse a los grupos.
- Estructura de una política:
  - Lenguaje que se utiliza json.
  - Statement.
  - SID.
  - Effect.
  - Principal.
  - Action.
  - Resource.



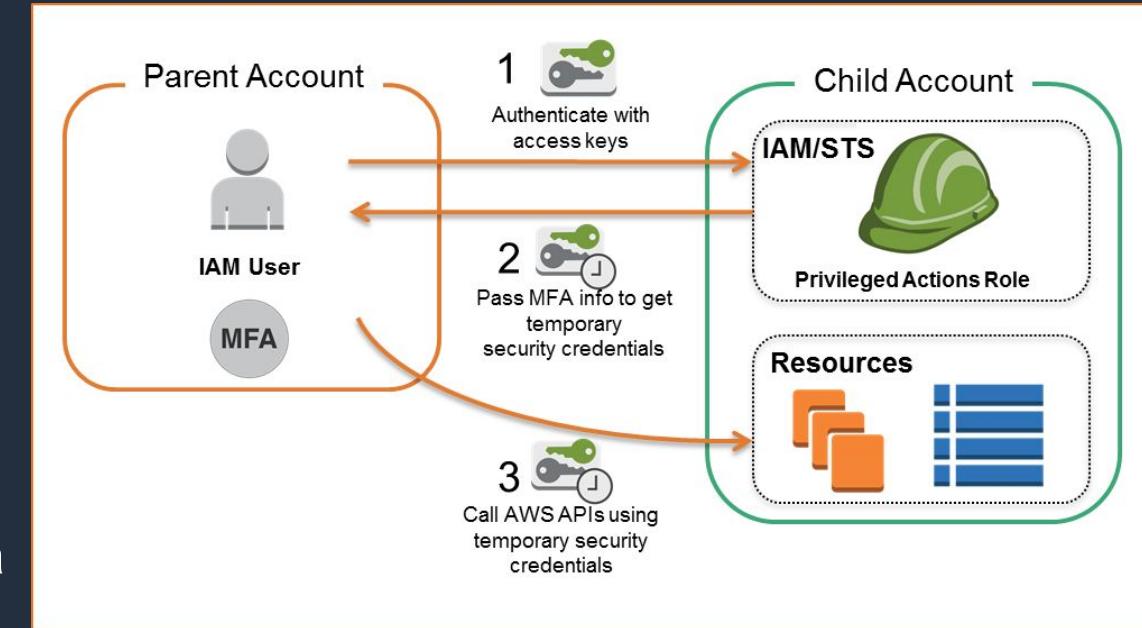
# IAM : Estructura de una política.

```
{  
    "Version": "2012-10-17",  
    "Id": "S3-Account-Permissions",  
    "Statement": [  
        {  
            "Sid": "1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": ["arn:aws:iam::123456789012:root"]  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Resource": ["arn:aws:s3:::mybucket/*"]  
        }  
    ]  
}
```



# IAM : Protección credenciales

- **Password Policy**
  - Cantidad mínima de caracteres.
  - Tipos de caracteres específicos
  - Permitir a los usuarios cambiar sus contraseñas.
  - Configurar el tiempo de vida de las contraseñas.
  - Se puede configurar el bloqueo de re uso de contraseñas.
- **MFA**
  - Contraseña conocida más un dispositivo.
  - Los usuarios deben tenerlo habilitado como buena práctica incluido root.
  - Dispositivos soportados.
    - Virtual MFA ( google auth, Authy ).
    - Universal Second Factor ( Yubikey ).
    - Dispositivos físicos ( Gemalto).



# IAM : Acceso AWS.

- AWS Management Console: Usuario/contraseña + MFA.
- AWS CLI : Claves de acceso ( Access Keys ).
- AWS SDK : Claves de acceso ( Access Keys ).
- Las claves se generan desde la consola de AWS.

## Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

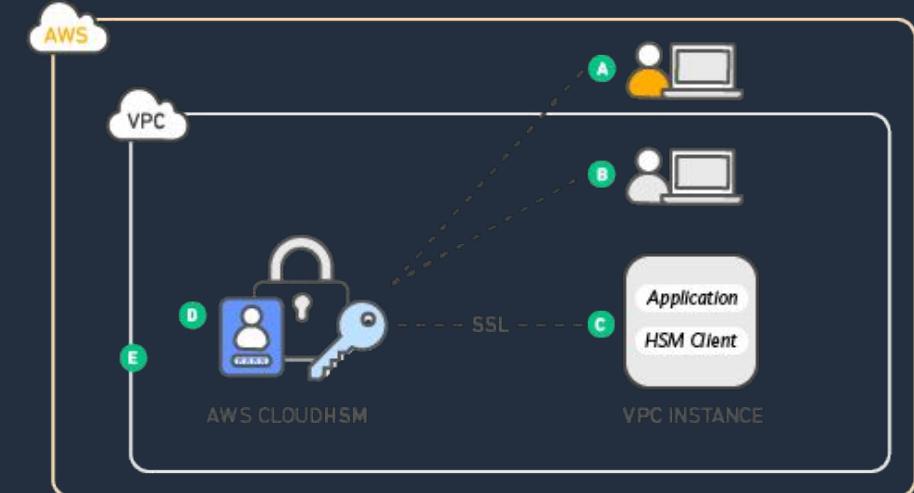
[Create access key](#)

Access key ID	Created	Last used	Status	
AKIASK4E37PV4TU3RD6C	2020-05-25 15:13 UTC+0100	N/A	Active	<a href="#">Make inactive</a> 



# Cifrado: Cloud HSM, KMS

- **Cifrado reposo :**
  - Datos almacenados.
  - S3, RDS, EFS, EBS
- **Cifrado en tránsito :**
  - Datos en movimiento.
  - Transferencia de datos.
  - Consultas sitios web, etc.
- **KMS.**
  - Servicio administrado gestión de claves.
  - Permite encriptar EBS, S3, RDS, RedShift, EFS..
  - Encripta automáticamente, Cloudtrail logs, S3 logs, Storage Gateways, Dynamo.
- **CMK.**
- **Cloud HSM.**
  - Hardware de encriptación dedicado.
  - Soporta cifrado curva elíptica.
  - PaaS.
  - Administración de clave responsabilidad del cliente.



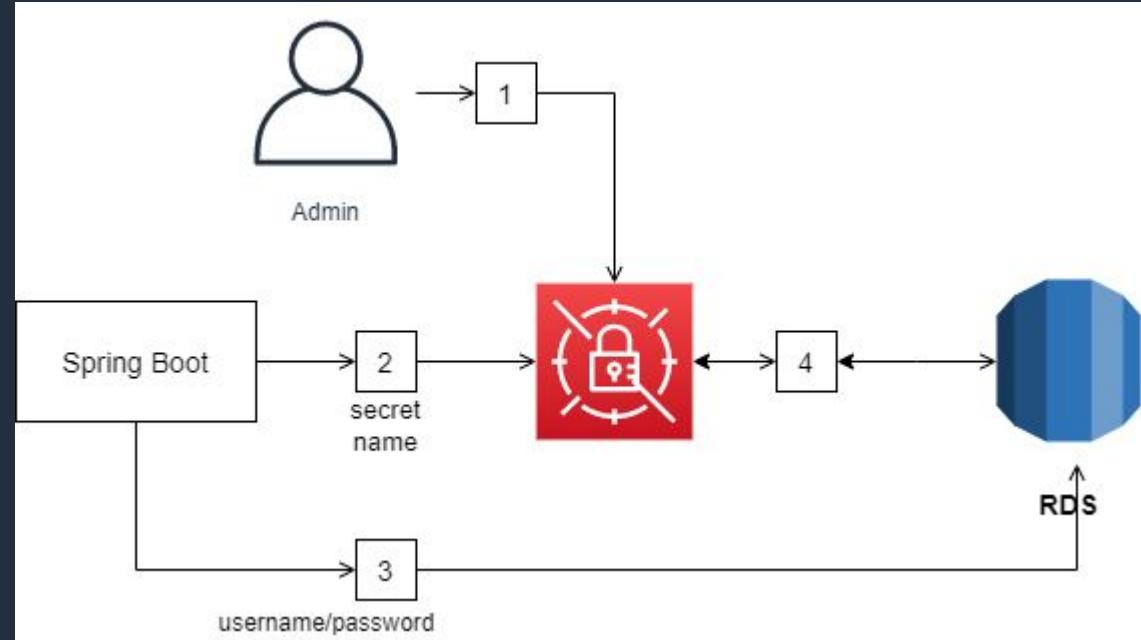
# Cifrado: ACM, Parameter Store

- **ACM.**
  - Aprovisionar, administrar y desplegar Certificados Digitales.
  - Cifrar en tránsito.
  - Soporta TLS públicos y privados.
  - Gratuito para TLS públicos.
  - Renovación automática.
  - Integración con ELB, Cloudfront y API Gateway.
- **Parameters store.**
  - Funcionalidad incluida en System Manager.
  - Se puede utilizar para almacenar variables y parámetros cifrados como secretos.
  - Se integra con otros servicios de AWS como KMS, SNS, Cloudwatch, Cloudtrail.



# Cifrado: Secret Manager

- Secret Manager.
  - Orientado administración Secretos.
  - Rotación programable.
  - Integración con lambda para obtener secretos rotados.
  - Integrado con RDS.
  - Encriptar usando KMS.
  - 30 días gratis.
  - 0,40 x mes x secreto.
  - 0,05 x 10000 llamados a la API.



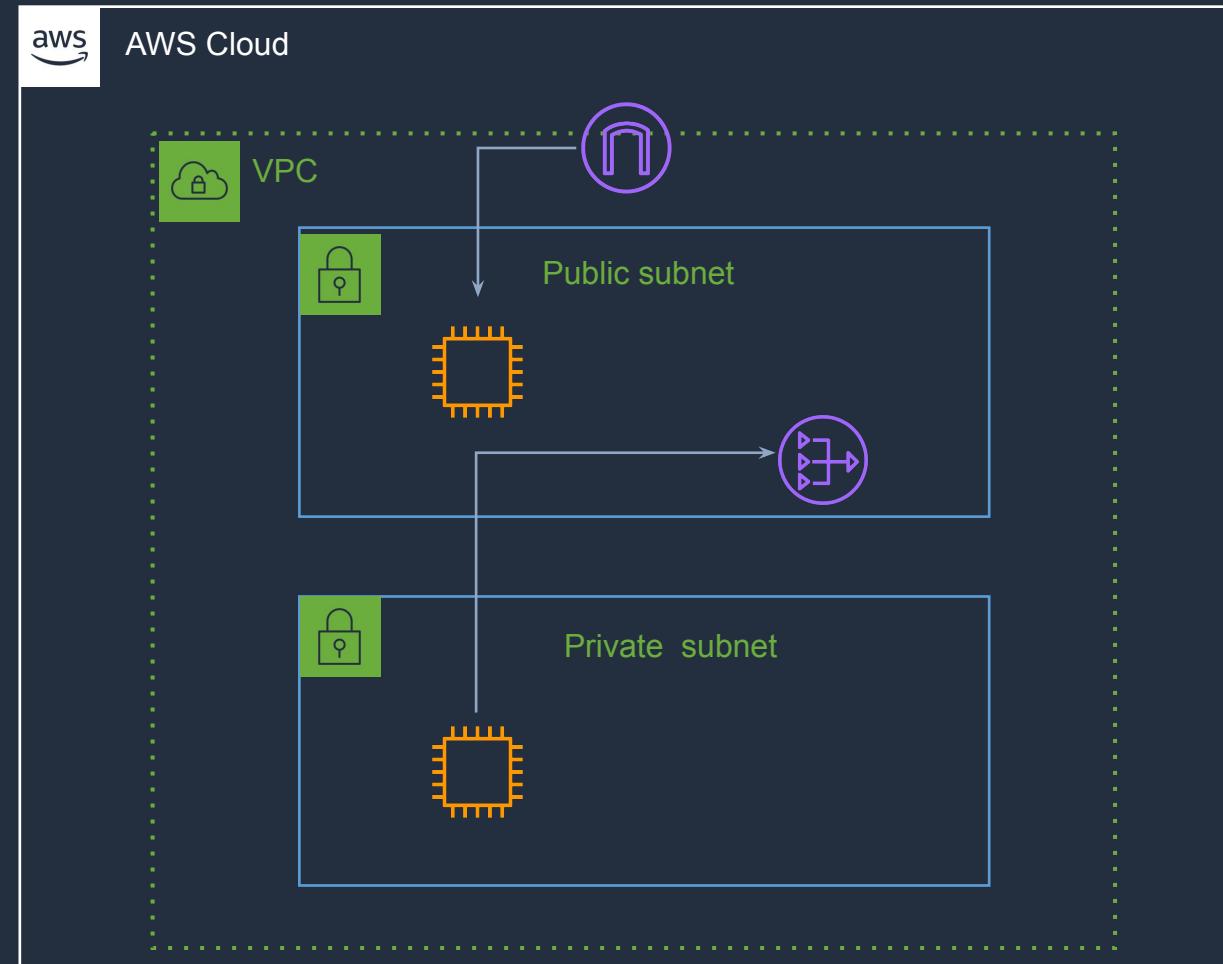
# Redes: Diseñando mi red en AWS.

- **Internet Gateway**

- Dispositivo que permite dar acceso a internet a una VPC.
- Por medio de tablas de ruteo.
- Escala horizontalmente.

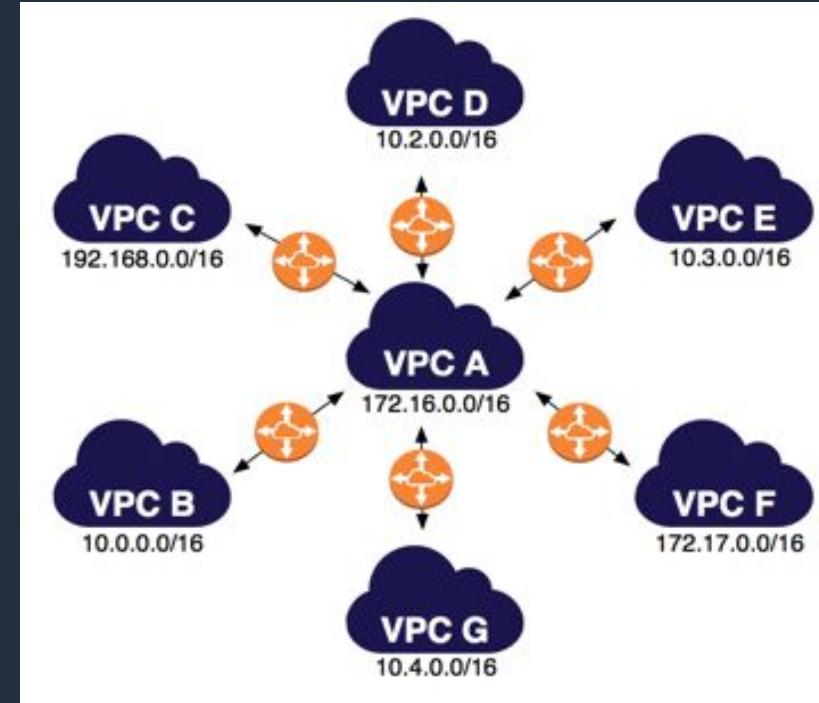
- **Nat Gateway**

- Es un servicio administrado.
- Permite dar acceso a instancias privadas a internet.
- Mantiene la subred privada.
- Por medio de tablas de ruteo.



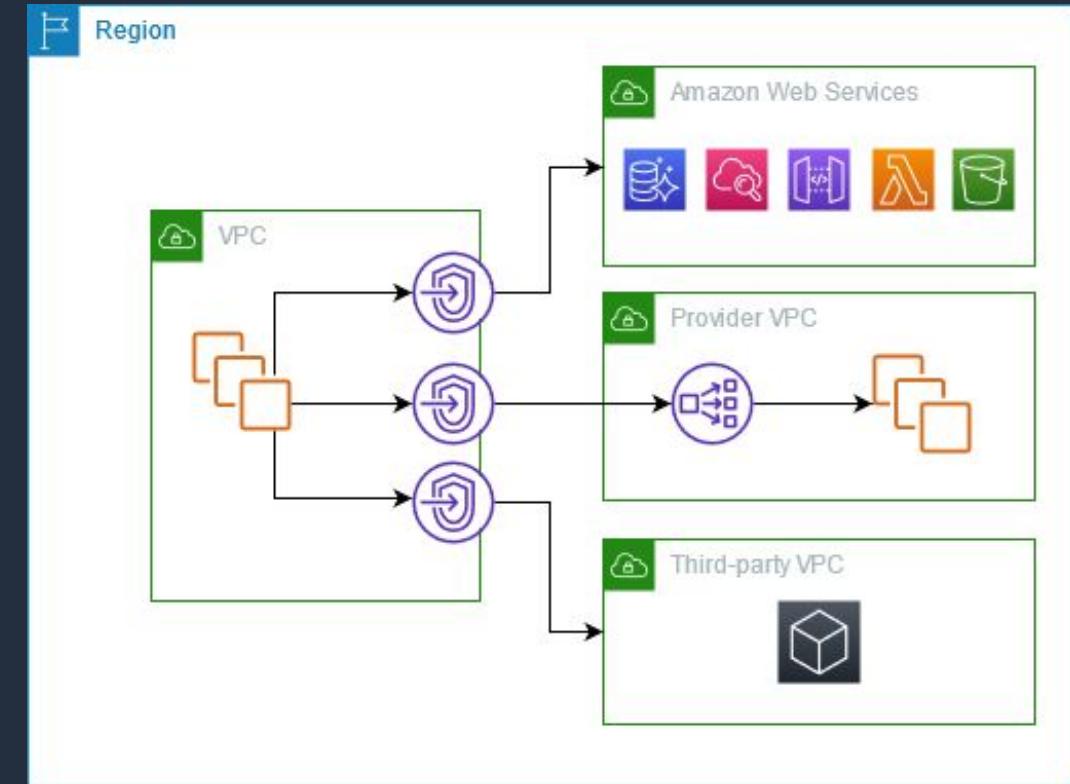
# Redes: Diseñando mi red en AWS.

- **VPC Peering**
  - Conectar dos VPC o más de manera privada usando la red de AWS.
  - No soporte overlapping.
  - No es transitiva.
  - Utiliza tablas de ruteo para hacer uso del peering.



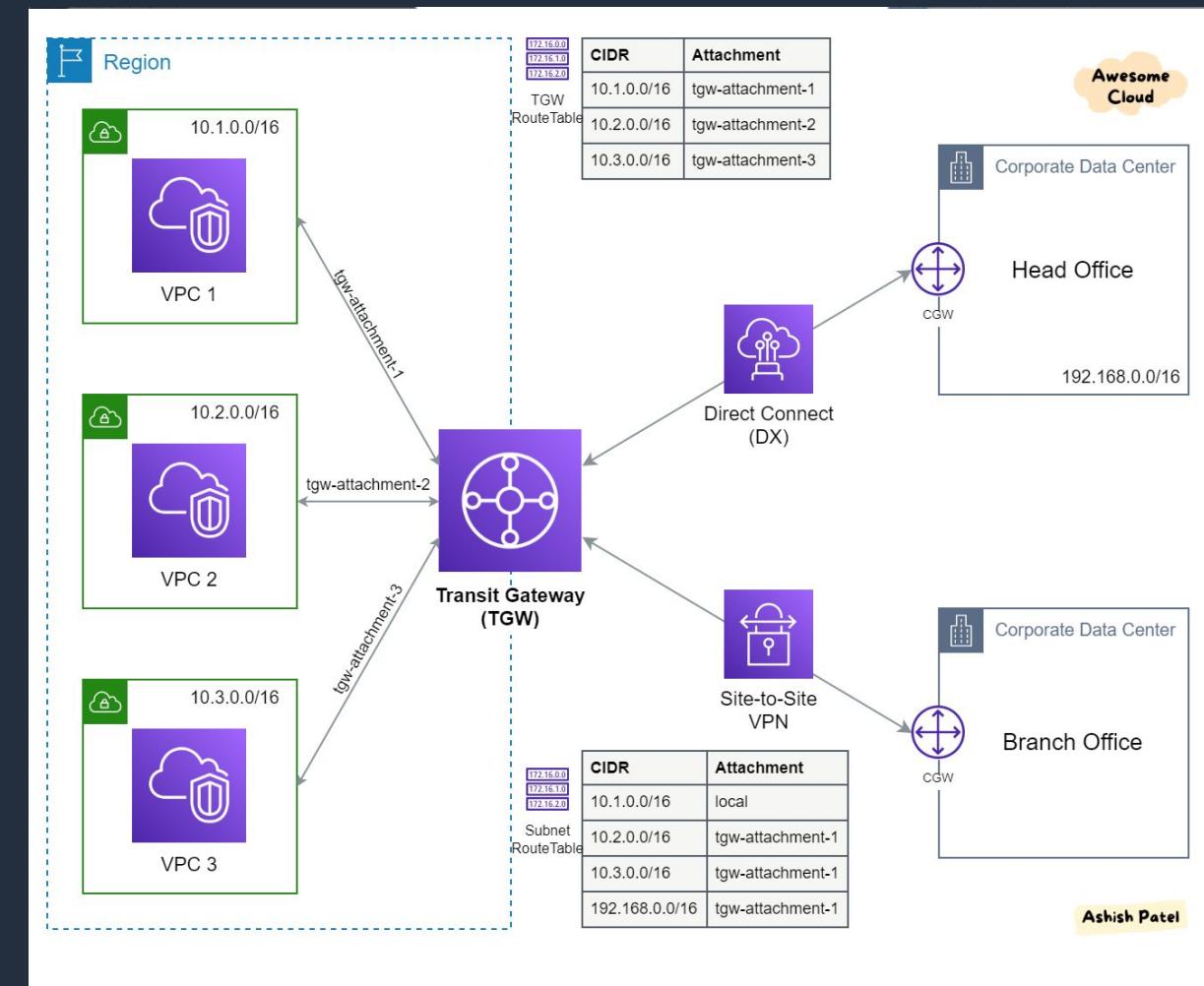
# Redes: Diseñando mi red en AWS.

- **VPC Endpoint ( Private Link )**
  - Posee todas las ventajas de un servicio autoadministrado.
  - Permite conectar servicios de AWS de manera privada usando la red de AWS, sin exponerlos a internet.
  - No requiere dispositivos adicionales como IG, Nat Gateway.
  - **Gateway :** Solo disponible entre S3 y DynamoDB.
    - La conexión se hace a nivel VPC.
    - Utiliza tablas de ruteo.
    - Restringir a través de IAM Policy.
    - HA.
  - [Interface](#):
    - Se debe crear por interfaz de red.
    - Solo soporta TCP.
    - Por defecto soporta 10 Gbps y puede escalar hasta 40 Gbps.
    - Para tener HA necesita Multi AZ.



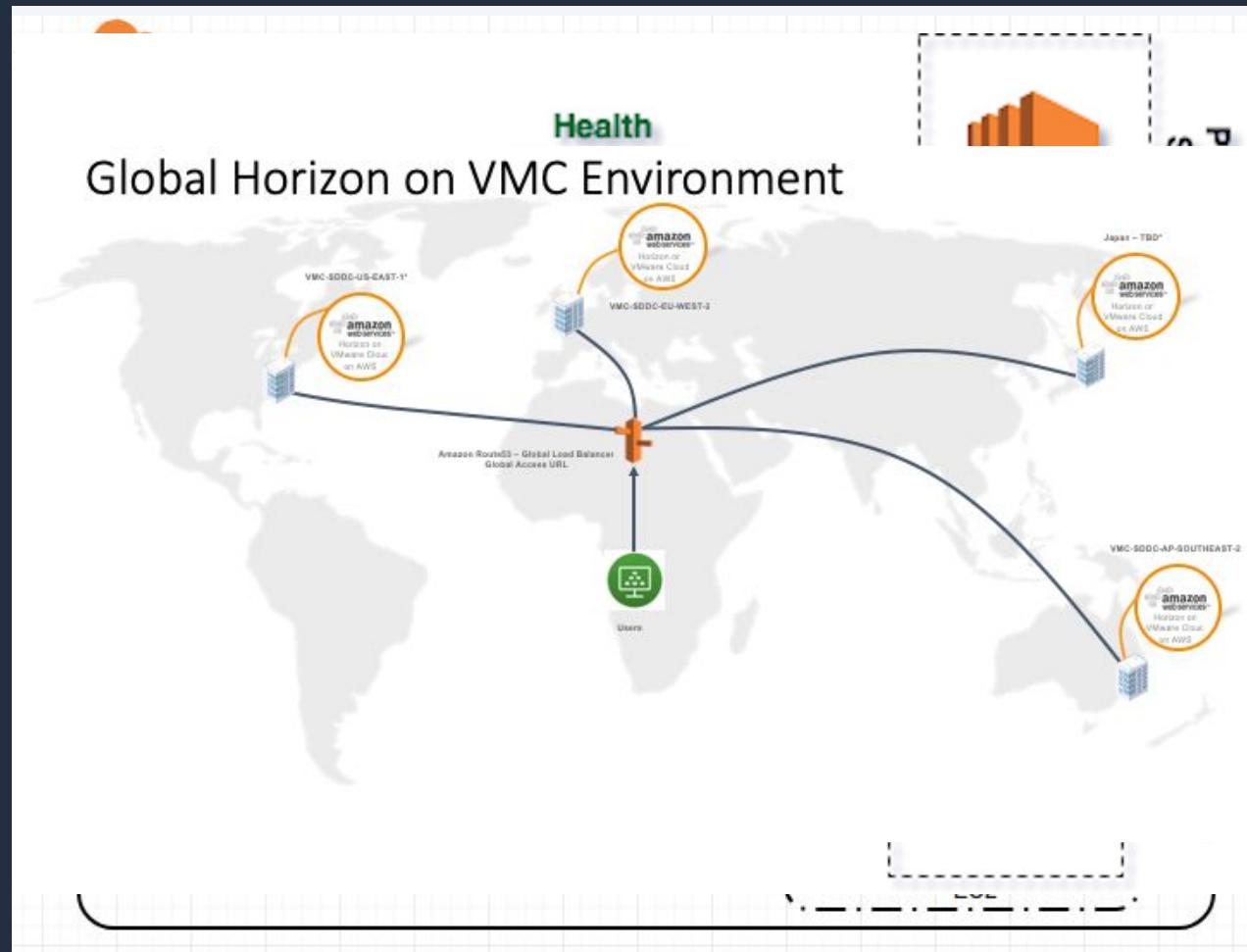
# Redes: Diseñando mi red en AWS.

- **Direct connect**
  - Conexión física entre on-premise y Cloud.
  - Baja latencia.
  - Tarda hasta un mes.
  - Canales dedicados entre 1 Gbps y 10 Gbps.
  - Canal privado sin exposición a internet.
- **Transit Gateway**
  - Gateway transitivo.
  - Conectar entre on-premise y VPC.
  - Soporta Direct Connect Gateway y VPN.
  - Centralizar la integración entre VPC y On-premise.



# Redes: Route53.

- Servidor de DNS auto administrado.
- Registros.
  - A : registro IPV4.
  - AAAA : registro IPV6.
  - CNAME : Hostname.
  - Alias : Utilizado para recurso AWS.
- Políticas de ruteo:
  - Simple : Basado en un registro de tipo.
  - Latency : Basado en la latencia de conexión.
  - Weighted : Ponderado puede enrutar por peso.
  - Failover : Tipo método activo pasivo.
  - Geolocalización : Basado en la ubicación geográfica de los clientes.





# Preguntas

# Dia 4

→ Construyendo soluciones de almacenamiento de datos en Nube

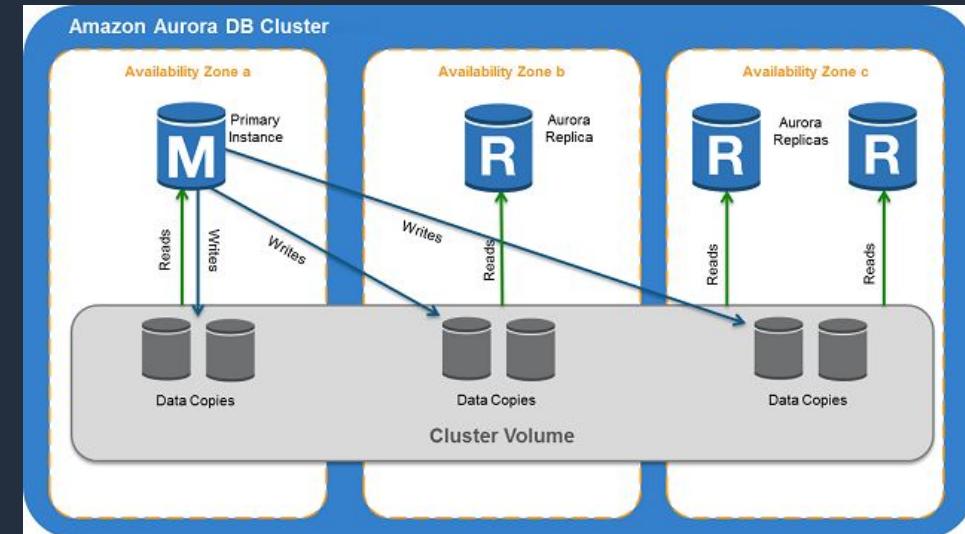
# Base de datos: RDS.

- Servicio de Base de datos Relacional.
- Los motores soportados son
  - Postgres.
  - Mysql.
  - MariaDB.
  - Oracle.
  - SQL Server.
  - Aurora ( propietario AWS ).
- Servicio Administrado AWS se encarga de lo siguiente:
  - Aprovisionamiento, parches S.O.
  - Backup regulares y fácil restauración.
  - Tableros monitoreo.
  - Replicas lectura.
  - Multi AZ.
  - Ventana de mantenimiento y capacidad de escalamiento.



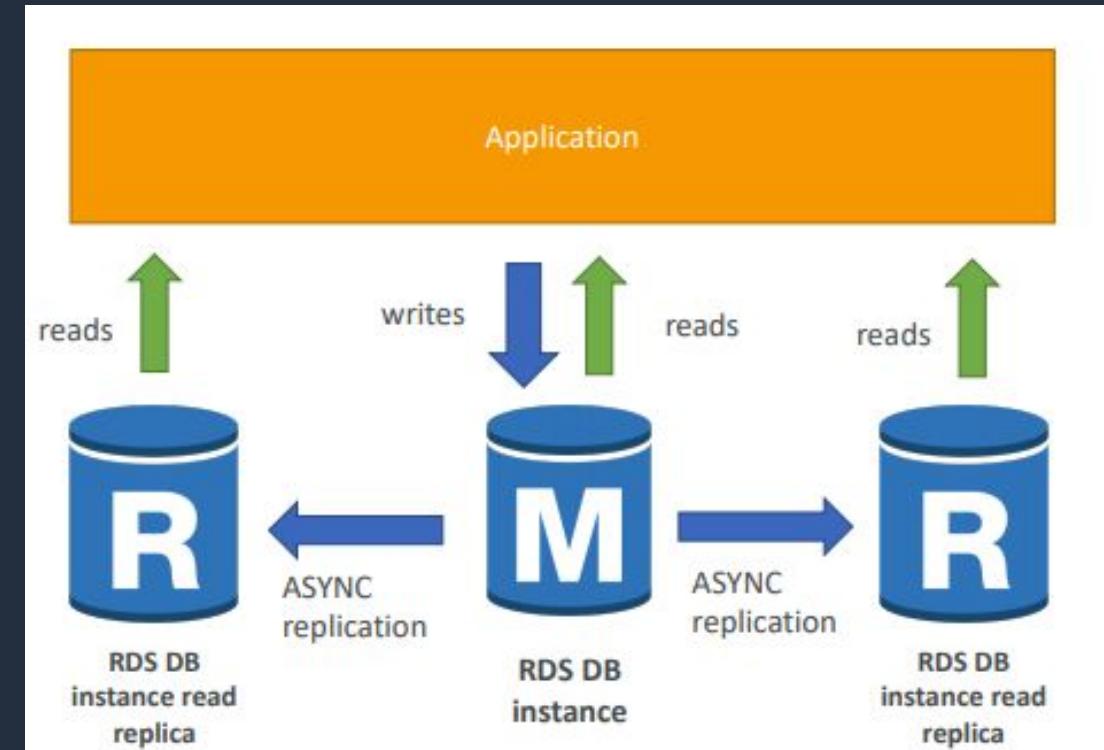
# Base de datos: RDS Aurora.

- Tecnología desarrollada por AWS.
- Compatible Mysql y Postgres.
- Optimizado para la operación en nube.
  - 5x Mysql.
  - 3x Postgres.
- Almacenamiento escala automáticamente de 10 Gb - 64 TB.
- Modelo de BD Global ideal esquemas DRP.
- No posee capa gratuita.
- Posee múltiples réplicas para conmutación por error.
- Hasta 5 réplicas de lectura Mysql.
- Aurora ML integración directa con SageMaker



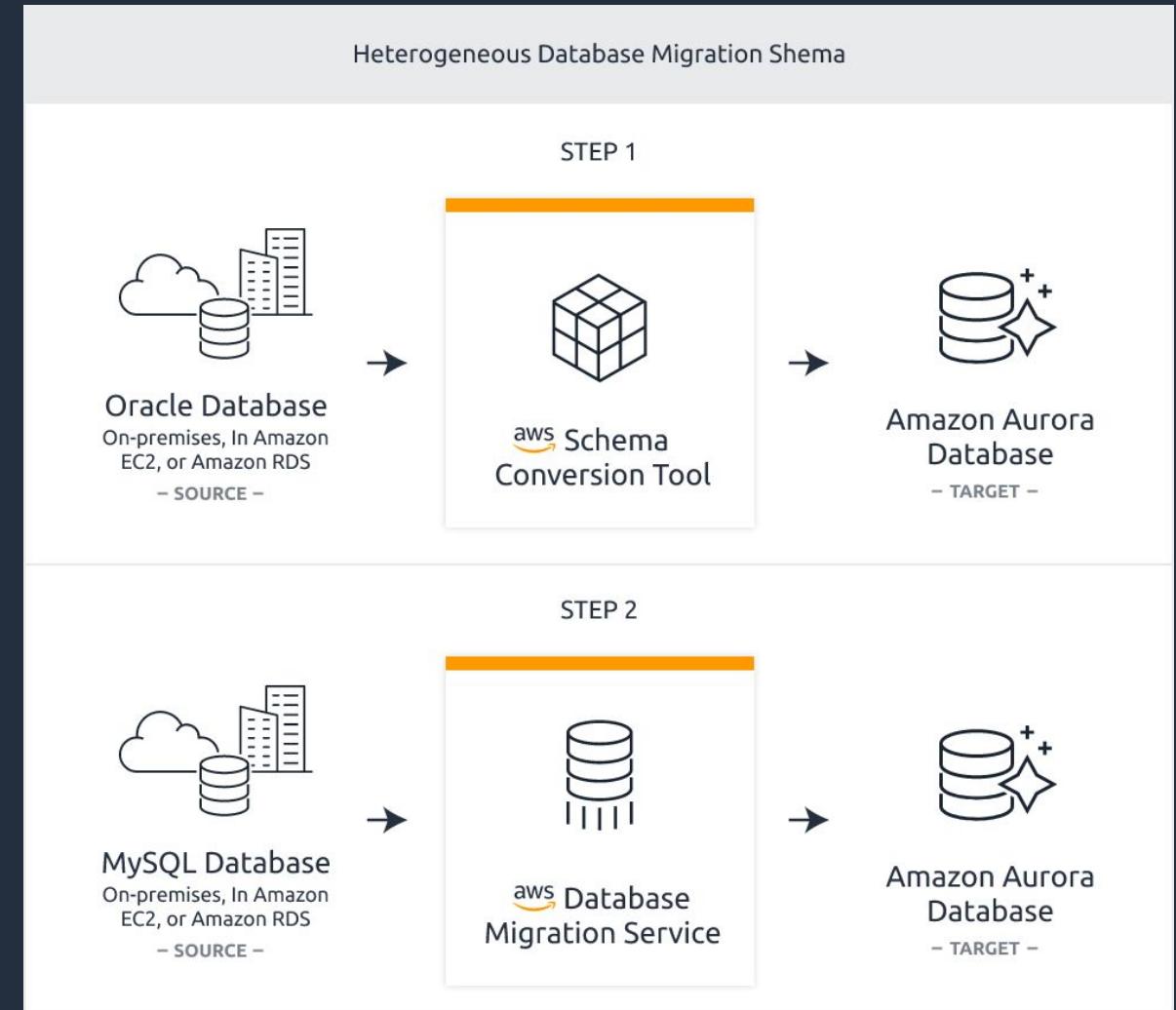
# RDS: Deployments options.

- Read Replica.
  - Es una réplica del nodo Master.
  - Permite que su solución escale lectura casos de soluciones de analítica.
  - Se pueden crear hasta 5 réplicas de lectura.
  - Los datos solo se escriben sobre el principal.
- Multi A-Z.
  - Para casos de un de una falla de AZ (HA)
  - Facilita actividades de Mantenimiento.
  - La conmutación es automática pero solo en 1 AZ.
- Multi Region.
  - Se basa en read réplicas en diferentes regiones.
  - Las escrituras están solo sobre el máster.
  - Mejora rendimiento por baja latencia.



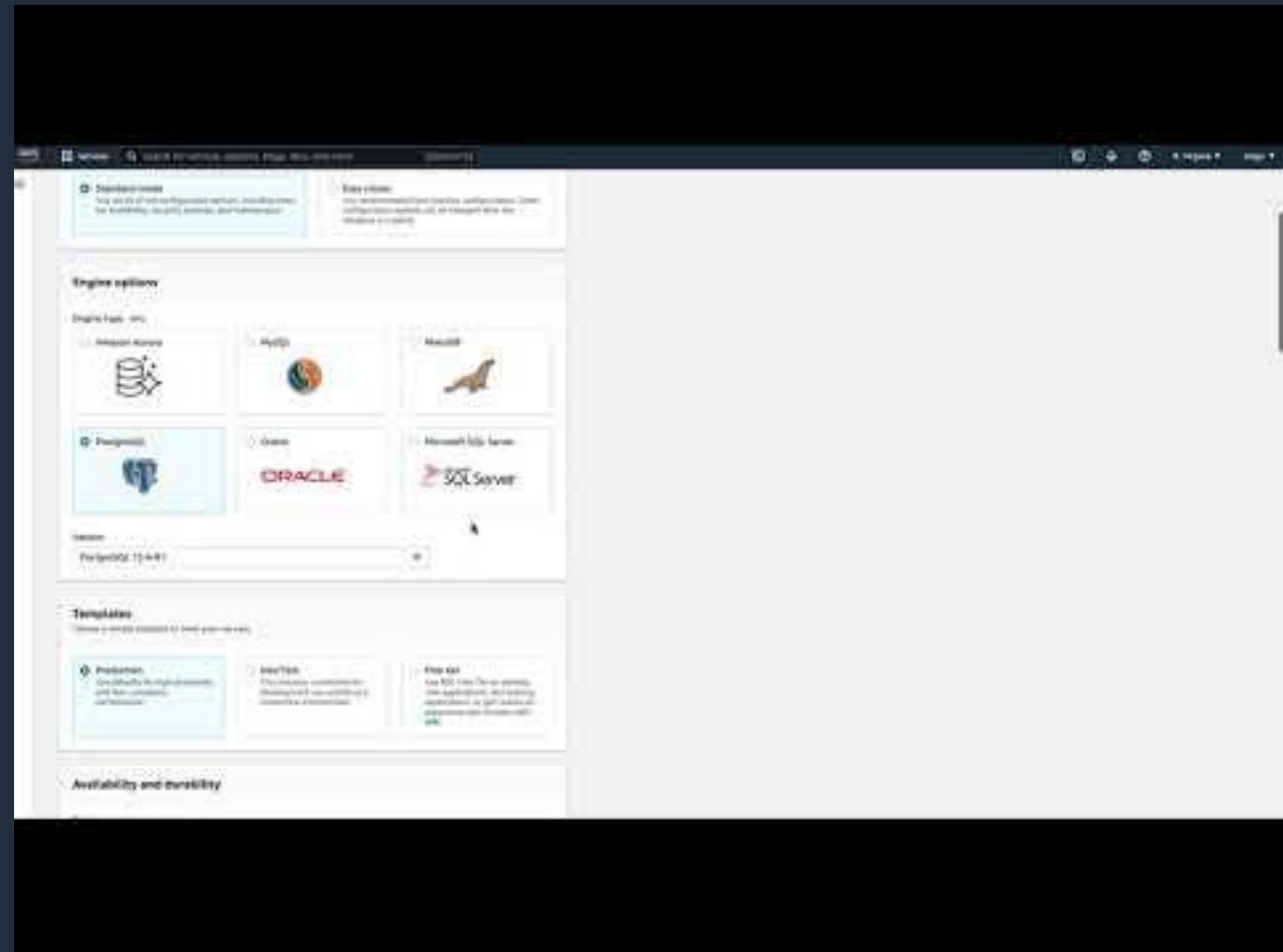
# RDS: Zero down time.

- Zero down time.
  - Estrategias para la migración.
    - DMS.
  - Copias de Seguridad y restauración.
    - RDS Backups.
      - Automaticos.
      - Punto de restauración.
      - Snapshot.
  - Read replica.
  - Cifrado de BD.
    - AES 256.
    - Oracle y SQL Server TDE.
    - KMS.
    - SSL / TLS.
  - RDS Proxy.



# RDS: Creacion.

- Creación y configuración.
  - Seleccionar Standard salvo que sean laboratorios.
  - Seleccionar tipo de motor.
  - Version del motor.
  - Template.
  - Nombre de la instancia.
  - Credenciales.
  - Tipo de instancia.
  - Almacenamiento.
  - Escalamiento.
  - Mutl AZ.

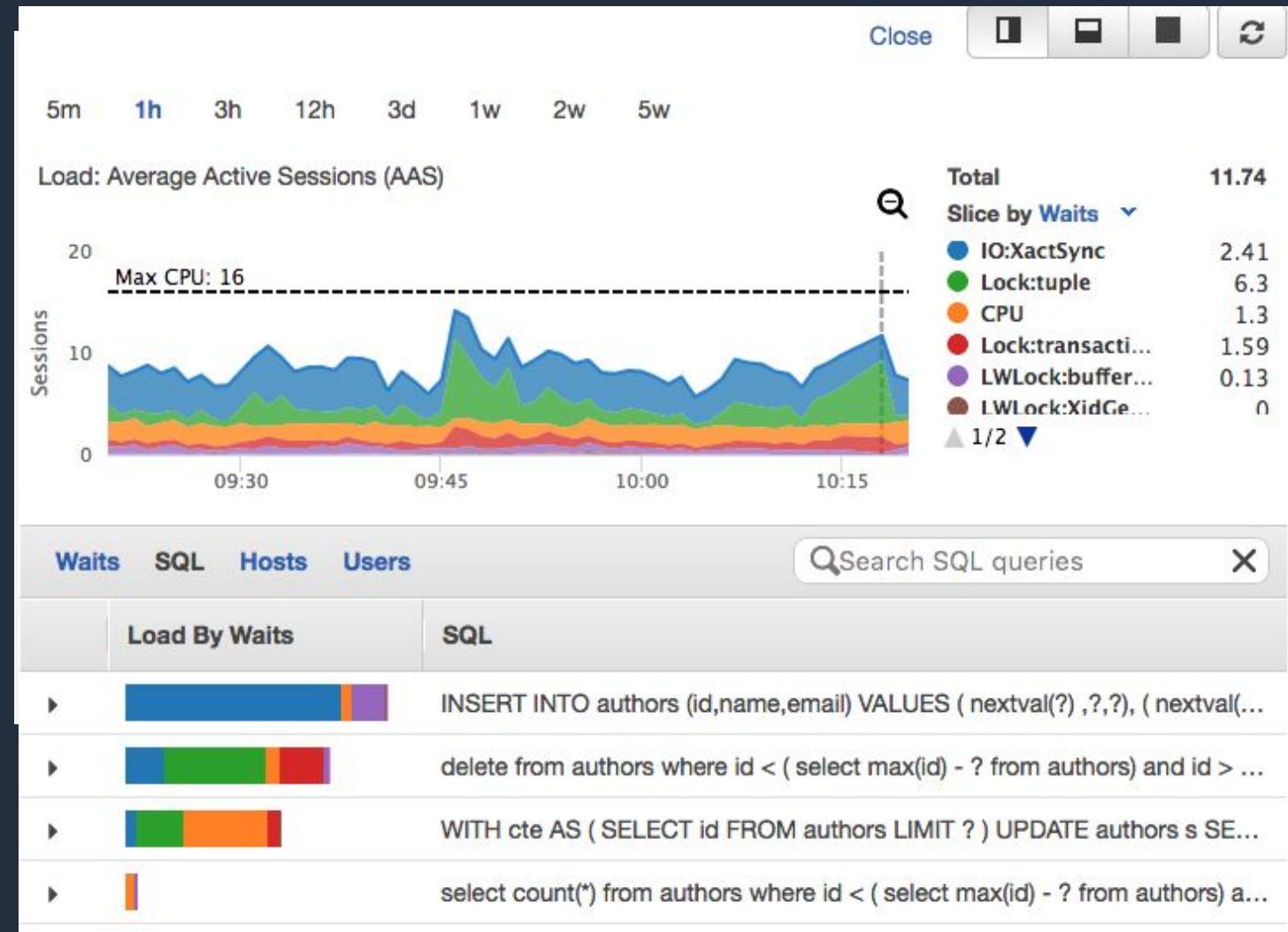


# RDS: Creación y configuración.

- Cuando necesitamos realizar configuraciones posteriores sobre el motor de base de datos.
- Actúa como un contenedor para los valores de configuración del motor que se aplican a una o más instancias de base de datos.
- Si crea una instancia sin especificar un grupo de parámetros , la instancia utiliza un grupo de parámetros predeterminado
- Los parámetros de la instancia son estáticos o dinámicos . Cuando cambia un parámetro estático y guarda el grupo , el cambio de parámetro entra en vigor después de reiniciar manualmente las instancias de base de datos asociadas.
- Cuando cambia un parámetro dinámico, de manera predeterminada, el cambio de parámetro se aplica a su instancia de base de datos de inmediato, sin necesidad de reiniciar.
- [https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER\\_WorkingWithParamGroups.html](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithParamGroups.html).

# RDS: Monitoreo.

- Cloudwatch.
  - Consumo elevado de CPU o RAM.
  - Consumo de espacio en disco.
  - Tráfico de red.
  - Conexiones.
  - IOPS.
- Performance insight.
  - Evalúa la carga en su base de datos y determina cuándo y dónde realizar acciones.
  - Disponible para Aurora, RDS MySQL y RDS PostgreSQL





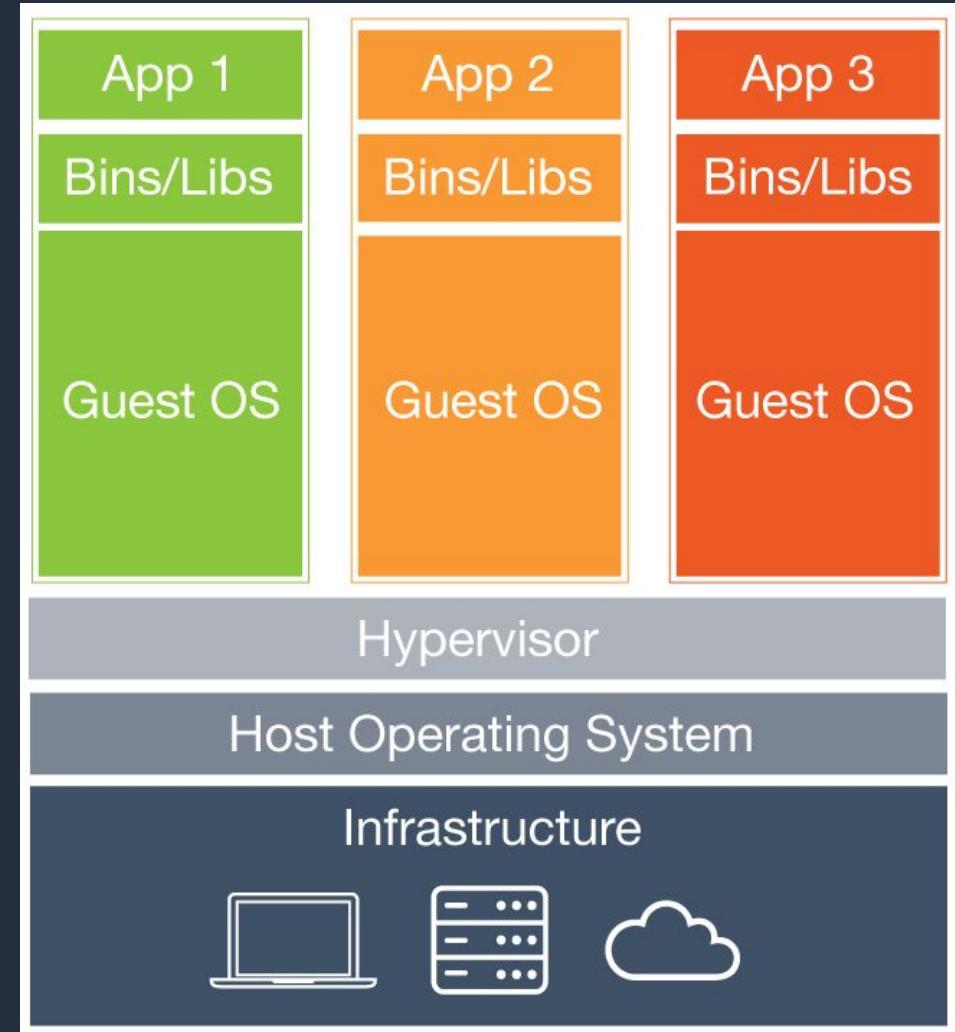
# Preguntas

# Dia 5

→ Construyendo una contenedores en Nube

# Contenedores: Introducción.

- El desaprovechamiento de recursos de hardware sobre los sistemas operativos, ha sido durante mucho tiempo uno de los principales desafíos a tener en cuenta por las plataformas de tecnología que basan su operación en la creación de herramientas para la gestión del desarrollo de software.
- Si bien la virtualización de servidores surgió como una opción revolucionaria para suplir esta necesidad, con el paso del tiempo se presentaron de nuevo ciertos desperdicios en cuanto a capacidad computacional, lo cual permitió el surgimiento de la tecnología que hoy conocemos como contenerización.



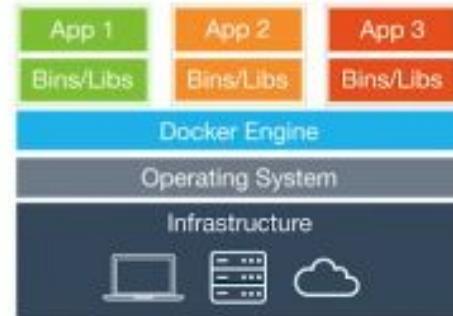
# Contenedores: Introducción.

- Es un método de virtualización sobre el kernel del sistema operativo que permite que existan múltiples instancias aisladas de espacios de usuario, en lugar de solo uno. Tales instancias, las cuales son llamadas contenedores, contenedores de software, jaulas o prisiones, pueden verse y sentirse como un servidor real desde el punto de vista de sus dueños y usuarios. Al software que permite el alojamiento de distintos contenedores se le llama motor de contenedores. Además de mecanismos de aislamiento, el kernel a menudo proporciona mecanismos de administración de recursos para limitar el impacto de las actividades de un contenedor sobre otros contenedores. Algunas ventajas son:

- Menos gastos generales
- Mayor portabilidad
- Funcionamiento más coherente
- Mayor eficiencia
- Mejor desarrollo de aplicaciones



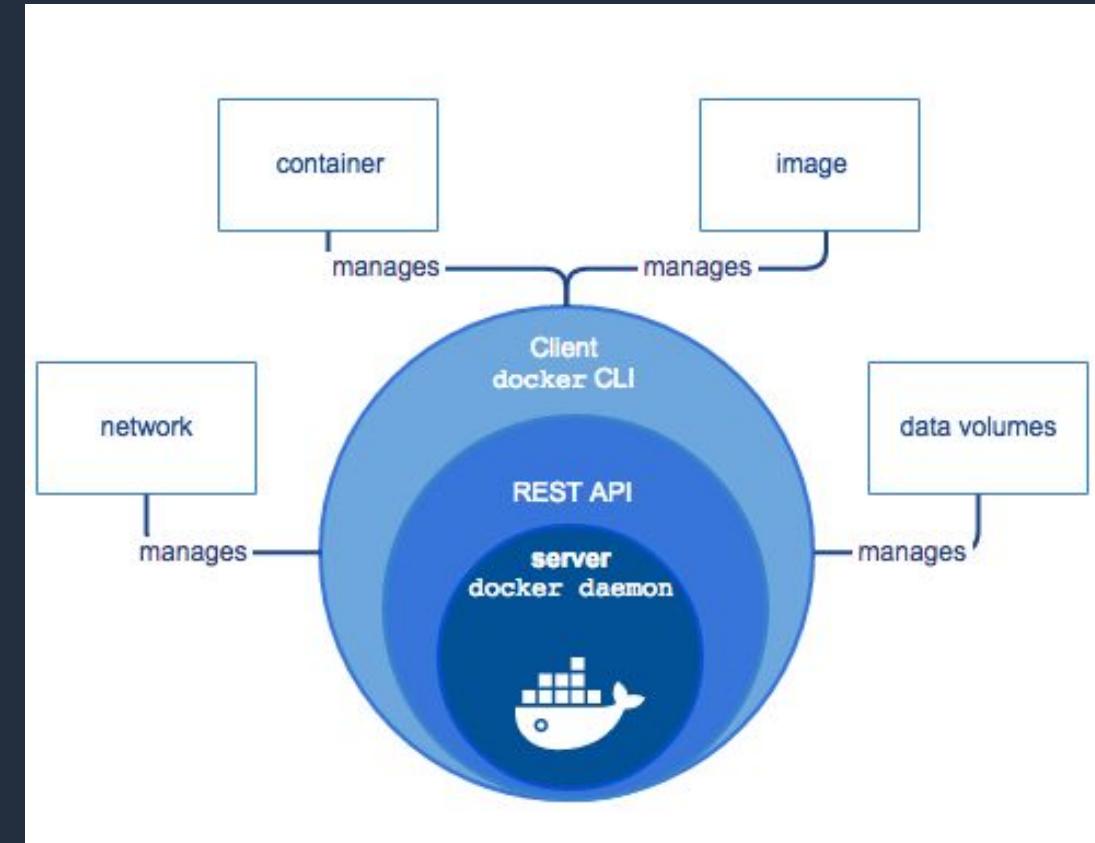
VMs



Contenedores

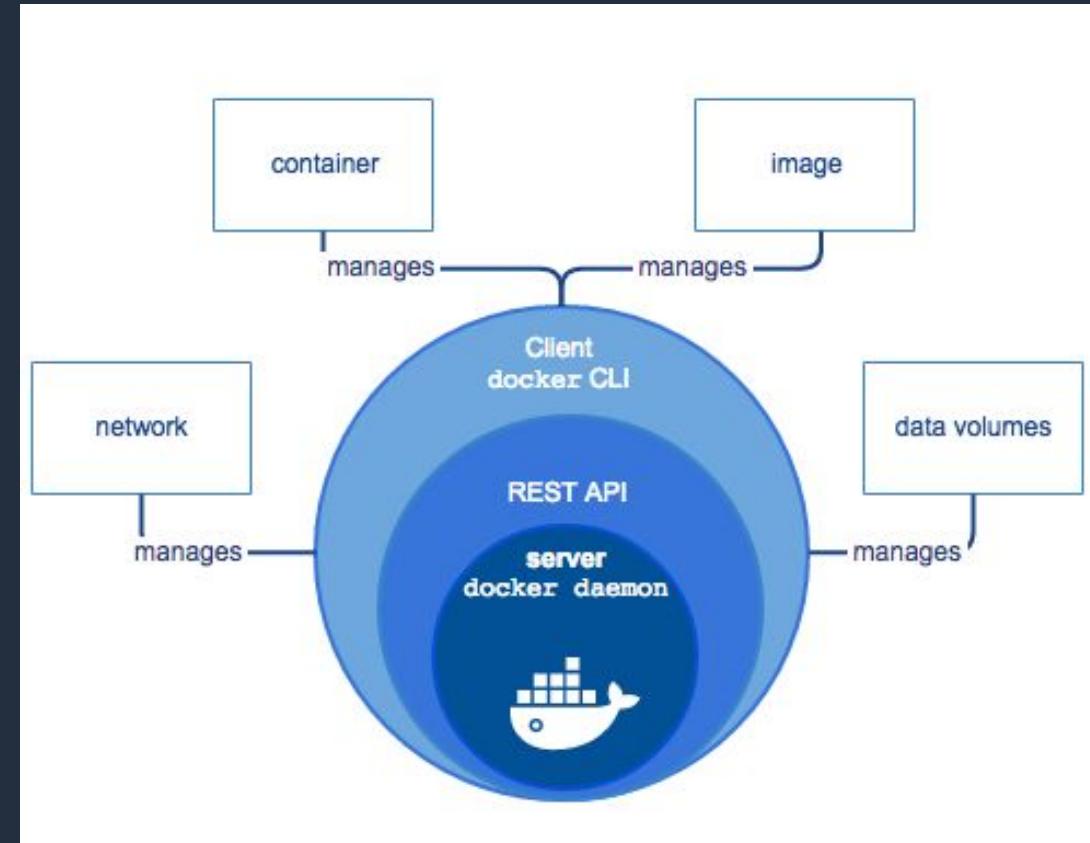
# Conceptos: Docker.

- **Arquitectura** : Docker se basa en una arquitectura, donde el cliente se comunica con el servidor (que es un proceso daemon) mediante un API para poder gestionar el ciclo de vida de los contenedores y así poder construir, ejecutar y distribuir los contenedores.
- Sus componentes principales están divididos en dos:
  - **Docker Engine**: Este es el motor de docker que posee tres
    - **Servidor**: es el proceso principal y que funciona como proceso del sistema. Es el encargado de gestionar los objetos que hay. Se representa mediante el comando dockerd.



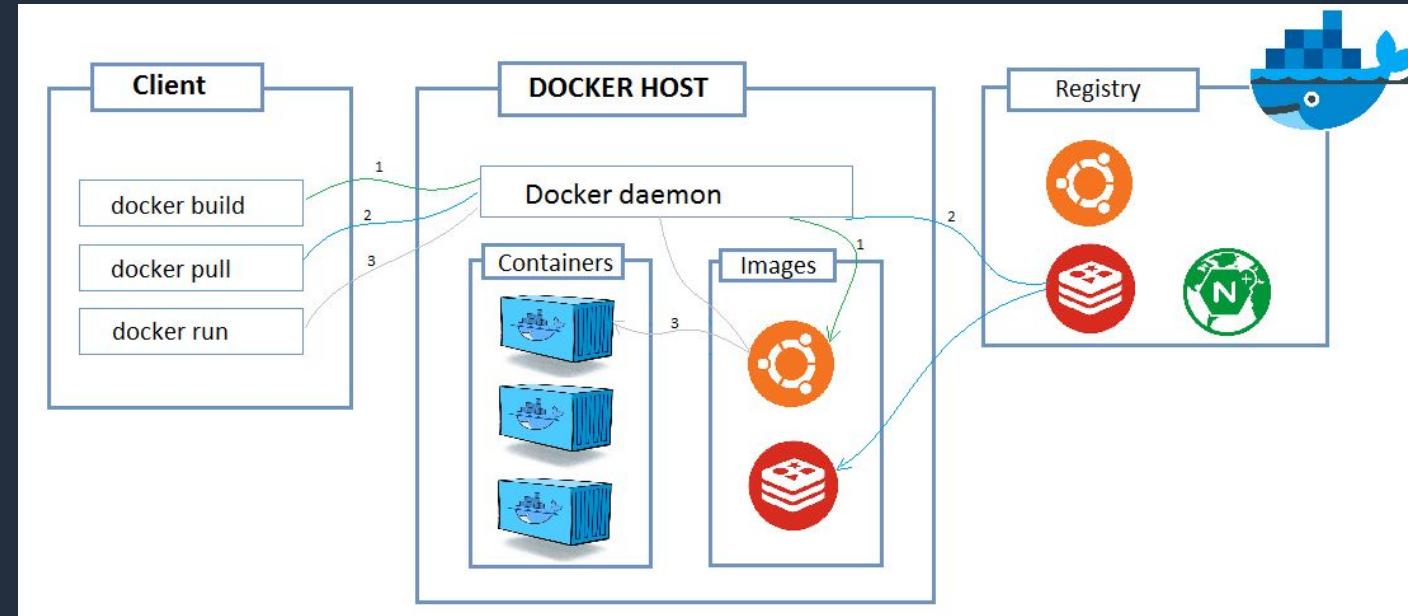
# Conceptos: Docker.

- Sus componentes principales están divididos en dos:
  - **Docker Engine:** Este es el motor de docker que posee tres
    - **API Rest:** nos permite acceder a las capacidades del servidor y ejecutar comandos sobre él. Podemos utilizar un simple curl para acceder a las capacidades del API de Docker
    - **Cliente:** es la línea de comandos representada por el comando docker. El cliente habla vía el API Rest para poder ejecutar los comandos. Es lo que utilizaremos para poder ir gestionando el ciclo de vida de nuestras imágenes y contenedores.



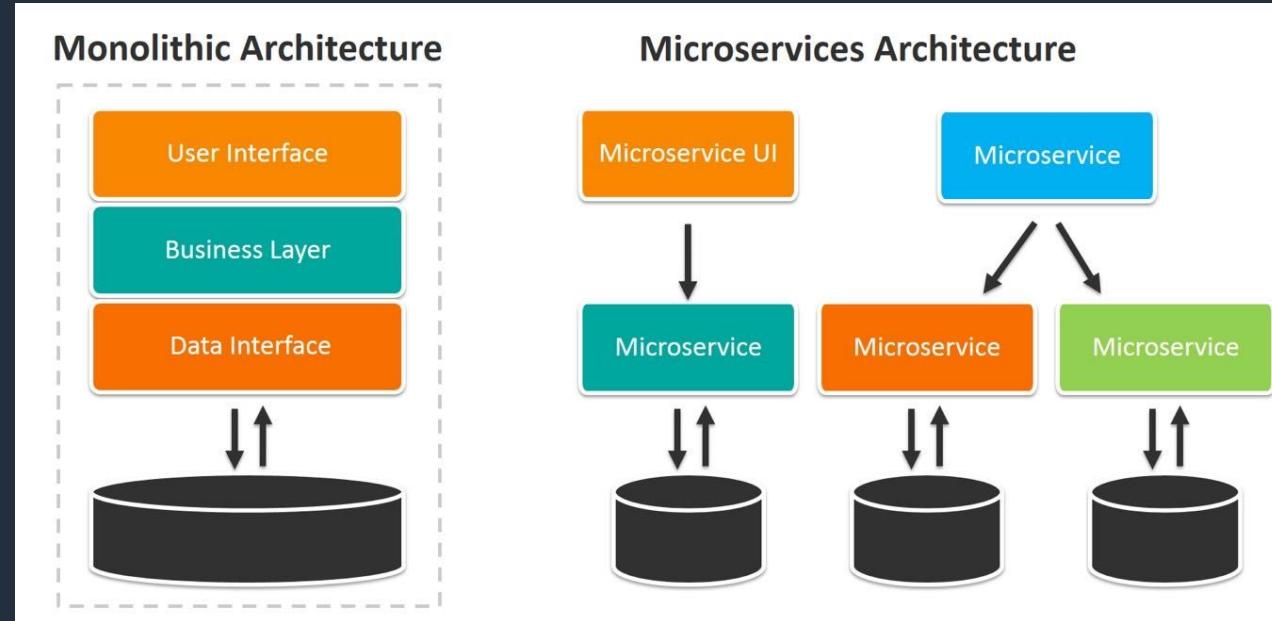
# Conceptos: Docker.

- **Docker Registry**: son los que almacenan imágenes Docker. El Docker Hub es un registro público que almacena múltiples imágenes, algunas de ellas certificadas por Docker.
- Por defecto, cuando ejecutamos un comando para crear un contenedor, se buscan las imágenes en Docker Hub. Si bien se pueden crear registros privados de imágenes mediante Docker Datacenter (DDC) y Docker Trusted Registry (DTR)



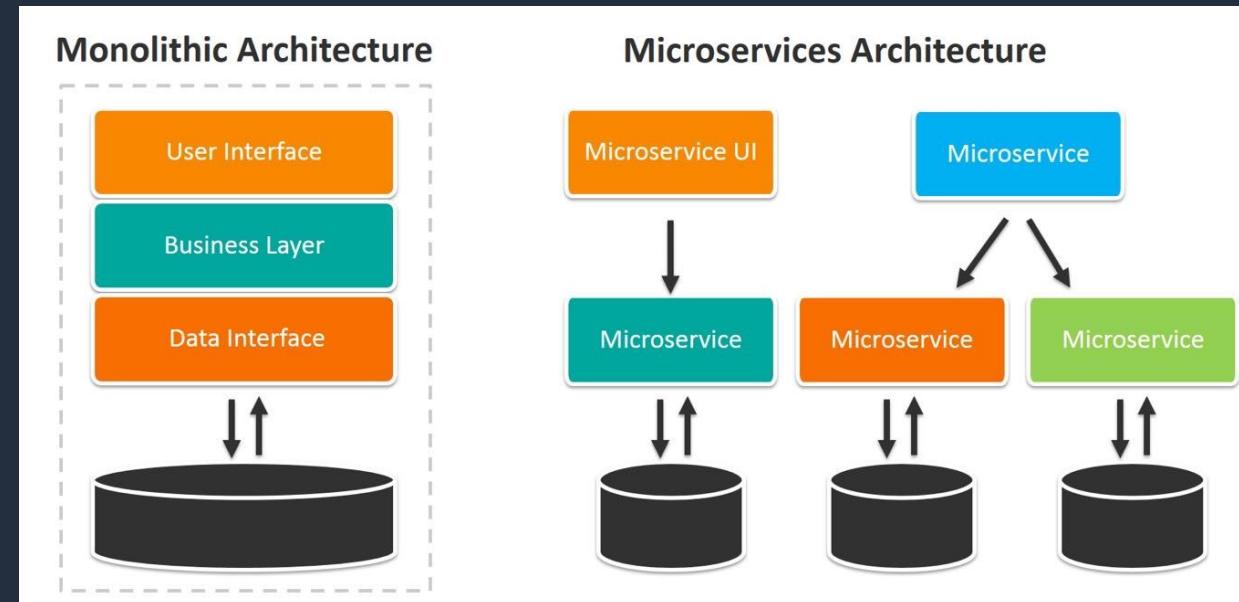
# Conceptos: Microservicios.

- **Definición :** Los microservicios son tanto un estilo de arquitectura como un modo de programar software. Con los microservicios, las aplicaciones se dividen en sus elementos más pequeños e independientes entre sí.



# Conceptos: Microservicios.

- **Arquitectura Monolítica vs Arquitectura de Microservicios :** Con las arquitecturas monolíticas, todos los procesos están estrechamente asociados y se ejecutan como un solo servicio. Esto significa que, si un proceso de una aplicación experimenta un pico de demanda, se debe escalar toda la arquitectura. Con una arquitectura de microservicios, una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Estos servicios se comunican a través de una interfaz bien definida mediante API ligeras.



# Conceptos: Kubernetes (K8s).

## ¿Qué es ?

Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa, código liberado por google en el 2014

Kubernetes ofrece un entorno de administración centrado en contenedores. Kubernetes orquesta la infraestructura de cómputo, redes y almacenamiento para que las cargas de trabajo de los usuarios no tengan que hacerlo. Esto ofrece la simplicidad de las Plataformas como Servicio (PaaS) con la flexibilidad de la Infraestructura como Servicio (IaaS) y permite la portabilidad entre proveedores de infraestructura.

## Ventajas de usar Kubernetes

- Ágil creación y despliegue de aplicaciones: eficiencia al crear imágenes
- Desarrollo, integración y despliegue continuo: Facilita rollback por que las imágenes son inmutables.
- Observabilidad: No solamente se presenta la información y métricas del sistema operativo, sino la salud de la aplicación y otras señales.
- Portabilidad entre nubes y distribuciones: Funciona en Ubuntu, RHEL, CoreOS, tu datacenter físico, Google Kubernetes Engine y todo lo demás
- Microservicios
- Aislamiento de recursos : Ayuda
- Utilización de recursos : Mejora la eficiencia.



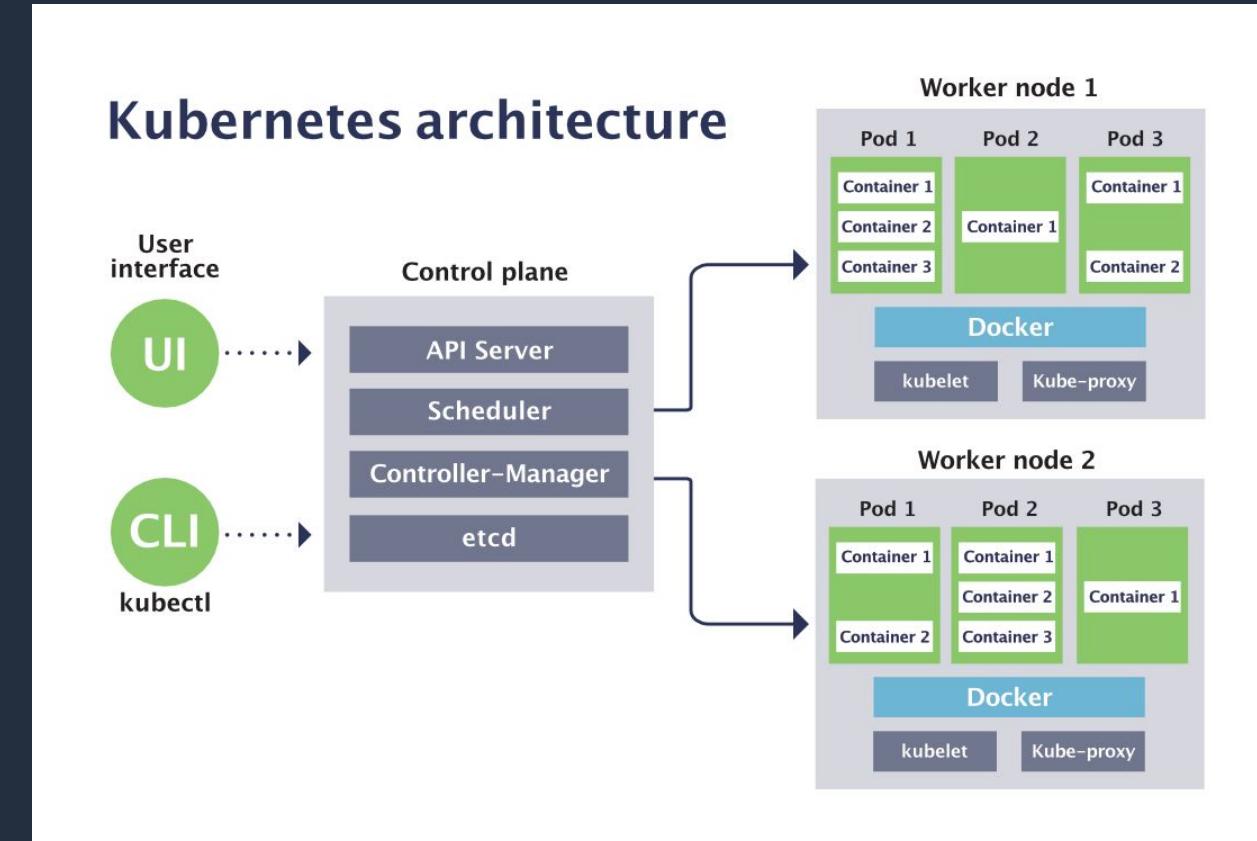
# Conceptos: Kubernetes.

## Arquitectura de kubernetes

**Nodos:** Un nodo puede ser una máquina virtual o física, dependiendo del tipo de clúster. Cada nodo está gestionado por el componente máster y contiene los servicios necesarios para ejecutar pods.

**Pod:** Son los objetos más pequeños y básicos que se pueden implementar. Representa una instancia única de un proceso en ejecución en tu clúster. Los pods contienen uno o más contenedores.

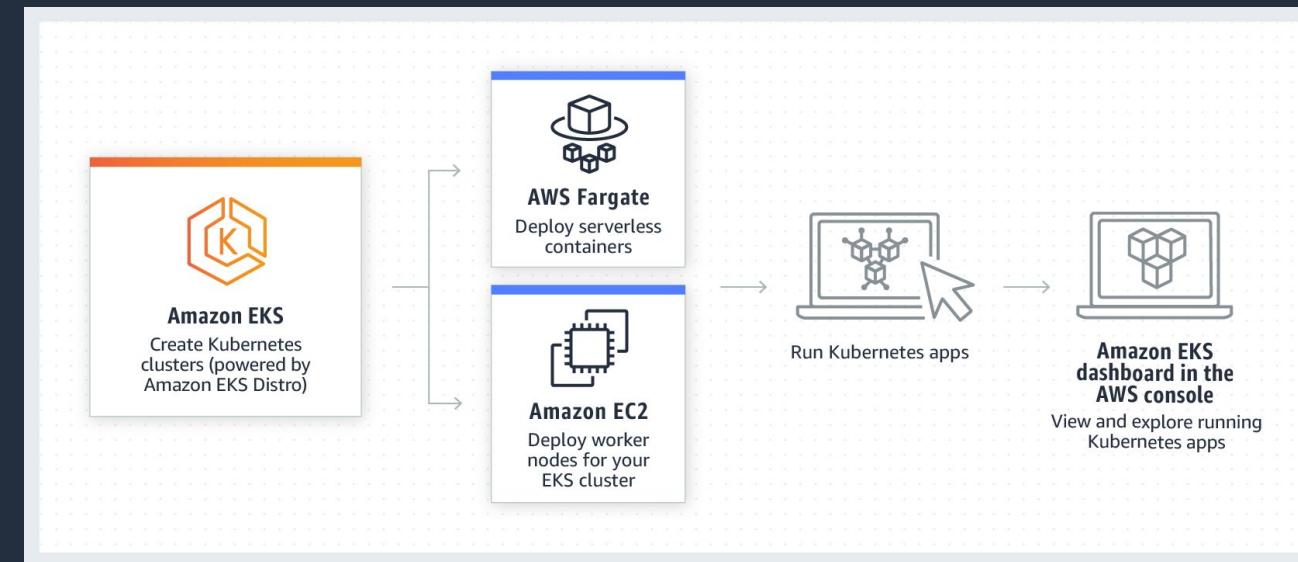
**Control Plane:** Los componentes que forman el plano de control toman decisiones globales sobre el clúster (por ejemplo, la planificación) y detectan y responden a eventos del clúster, como la creación de un nuevo pod cuando la propiedad réplicas de un controlador de replicación no se cumple.



# Kubernetes en AWS

Amazon Elastic Kubernetes Service (EKS) es un servicio de Kubernetes administrado que facilita la ejecución de esta recurso en AWS y a nivel local. EKS cuenta con certificación de conformidad con Kubernetes, por lo que las aplicaciones existentes que se ejecutan en el Kubernetes ascendente son compatibles con Amazon EKS.

EKS le permite administrar las aplicaciones de Kubernetes tanto en Amazon EC2 como en AWS Fargate, lo que le ofrece computación para sus contenedores en ausencia de servidor. Fargate aprovisiona y escala automáticamente la computación de sus contenedores. Con Fargate, solo tendrá que abonar los recursos solicitados por sus aplicaciones en ejecución. Cada pod que se ejecuta en Fargate se encuentra aislado en virtud de su diseño, lo que mejora la seguridad de la aplicación.



# Servicios de red en EKS

- **Amazon VPC y subredes:** todos los recursos de Amazon EKS se implementan en una región en una subred existente en una VPC existente. Cada subred existe en una zona de disponibilidad. La VPC y las subredes deben cumplir requisitos como los siguientes:
  - Las VPC y las subredes deben etiquetarse adecuadamente para que Kubernetes sepa que puede utilizarlas para implementar recursos, como los balanceadores de carga. Si implementa la VPC mediante una plantilla de CloudFormation eksctl, entonces la VPC y las subredes se etiquetan apropiadamente.
  - Una subred puede o no tener acceso a internet. Si una subred no tiene acceso a Internet, los pods desplegados dentro de ella deben poder acceder a otros servicios de AWS para extraer imágenes de los contenedores, como Amazon ECR.
  - Las subredes públicas que utilice deben configurarse para asignar automáticamente direcciones IP públicas a instancias de Amazon EC2 lanzadas dentro de ellas.



## Amazon EKS

# Etiquetado red EKS

- **Amazon VPC y subredes:** todos los recursos de Amazon EKS se implementan en una región en una subred existente en una VPC existente. Cada subred existe en una zona de disponibilidad. La VPC y las subredes deben cumplir requisitos como los siguientes:
  - Las VPC y las subredes deben etiquetarse adecuadamente para que Kubernetes sepa que puede utilizarlas para implementar recursos, como los balanceadores de carga. Si implementa la VPC mediante una plantilla de CloudFormation eksctl, entonces la VPC y las subredes se etiquetan apropiadamente.
  - Una subred puede o no tener acceso a internet. Si una subred no tiene acceso a Internet, los pods desplegados dentro de ella deben poder acceder a otros servicios de AWS para extraer imágenes de los contenedores, como Amazon ECR.
  - Las subredes públicas que utilice deben configurarse para asignar automáticamente direcciones IP públicas a instancias de Amazon EC2 lanzadas dentro de ellas.



## Amazon EKS

# Servicios de red en EKS

- **Amazon VPC y subredes:** todos los recursos de Amazon EKS se implementan en una región en una subred existente en una VPC existente. Cada subred existe en una zona de disponibilidad. La VPC y las subredes deben cumplir requisitos como los siguientes:
  - Las VPC y las subredes deben etiquetarse adecuadamente para que Kubernetes sepa que puede utilizarlas para implementar recursos, como los balanceadores de carga. Si implementa la VPC mediante una plantilla de CloudFormation eksctl, entonces la VPC y las subredes se etiquetan apropiadamente.
  - Una subred puede o no tener acceso a internet. Si una subred no tiene acceso a Internet, los pods desplegados dentro de ella deben poder acceder a otros servicios de AWS para extraer imágenes de los contenedores, como Amazon ECR.
  - Las subredes públicas que utilice deben configurarse para asignar automáticamente direcciones IP públicas a instancias de Amazon EC2 lanzadas dentro de ellas.



## Amazon EKS

# Etiquetado en EKS

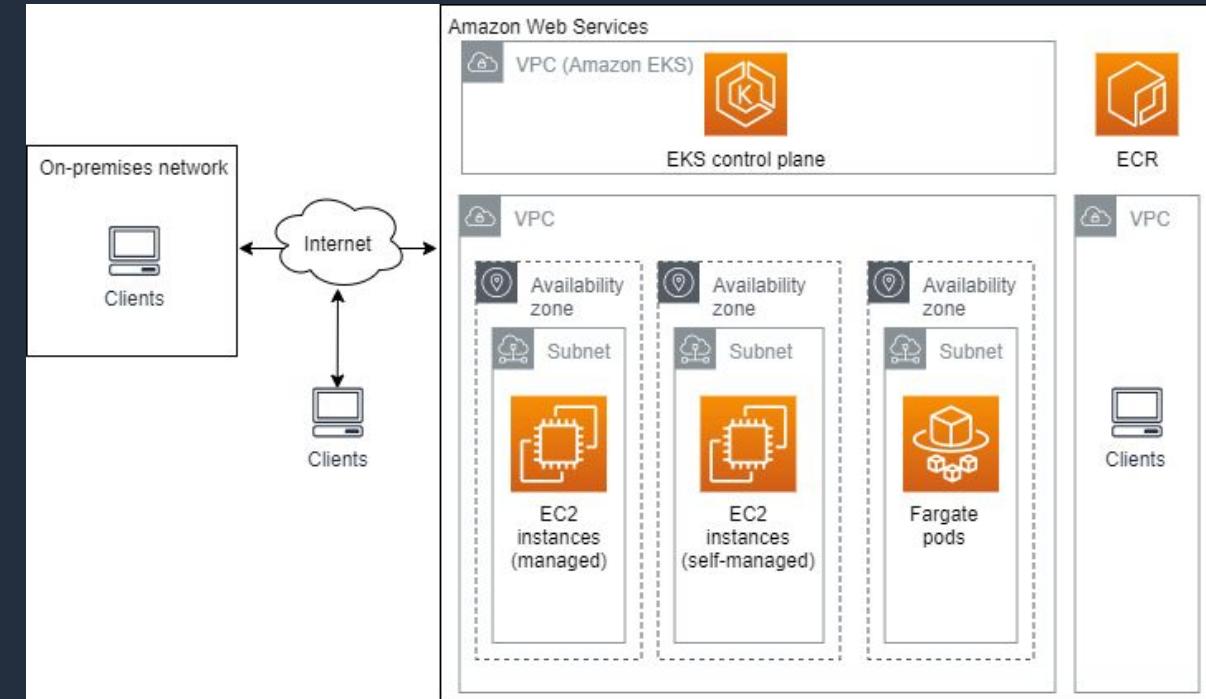
- Para los clústeres 1.18 y anteriores, Amazon EKS agrega la siguiente etiqueta a todas las subredes asociadas durante la creación del clúster. Amazon EKS no agrega la etiqueta a las subredes asociadas al crear clústeres 1.19. Si la etiqueta existe en las subredes utilizadas por un clúster creado en una versión anterior a 1.19 y actualiza el clúster a 1.19, la etiqueta no se elimina de las subredes.
  - Clave = kubernetes.io/cluster/<cluster-name>
  - Valor = shared
- Uno, y sólo uno, de los grupos de seguridad asociados a los nodos debe tener la siguiente etiqueta aplicada:
  - Key = kubernetes.io/cluster/<cluster-name>
  - Valor = owned



## Amazon EKS

# Servicios de red en EKS

- Los nodos y el plano de control deben poder comunicarse a través de todos los puertos a través de la etiqueta apropiada en los grupos de seguridad.
- Se puede implementar una VPC y subredes que cumplan los requisitos de Amazon EKS mediante la configuración manual o mediante la implementación de la VPC y las subredes mediante eksctl o una plantilla para Amazon EKS de CloudFormation.



# Servicios de red en EKS

- VPC
- Subnets
- Internet Gateway
- NAT Gateway
- Route tables
- Security groups
- CNI:
  - Amazon EKS admite redes VPC nativas con el complemento Amazon VPC Container Network Interface (CNI) para Kubernetes. Este plugin asigna una dirección IP de su VPC a cada pod. El complemento es un proyecto de código abierto mantenido en GitHub
- CoreDNS
  - CoreDNS es un servidor DNS flexible y extensible que puede servir como DNS del clúster de Kubernetes. Al iniciar un clúster de Amazon EKS con al menos un nodo, se implementan dos réplicas de la imagen CoreDNS de forma predeterminada, independientemente del número de nodos desplegados en el clúster. Los Pods CoreDNS proporcionan resolución de nombres para todos los Pods del clúster.

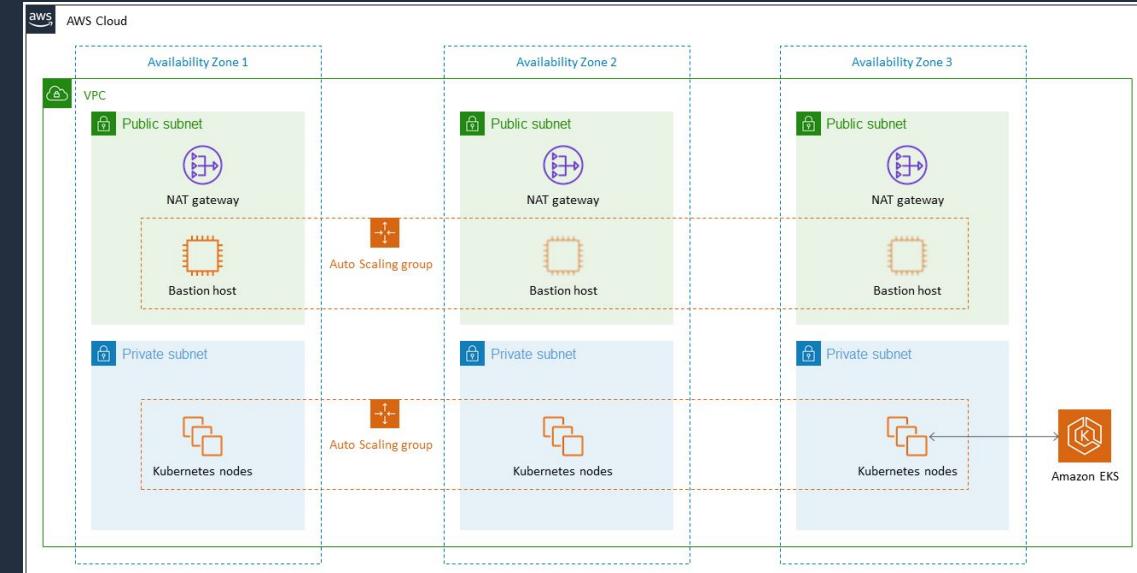


# Arquitectura de EKS

- EKS es un servicio de Kubernetes administrado por AWS, cada carga que corre en eks puede correr de la misma manera en Kubernetes.
- Dentro de esta arquitectura AWS se encarga del plano de control y el cliente es responsable de la administración de cada nodo de trabajo.
- **Plano de control en EKS:** La infraestructura del plano de control no se comparte entre clústeres. Consta de al menos dos instancias de servidor API , las instancias que se ejecutan en tres zonas de disponibilidad de una región.
- **Supervisa** activamente la carga en las instancias del plano de control y las escala automáticamente .
- **Detecta y reemplaza** automáticamente las instancias en mal estado y las reinicia en las zonas de disponibilidad de la región según sea necesario.
- **Aprovecha** la arquitectura de AWS Regiones con el fin de mantener una alta disponibilidad.

# Arquitectura de EKS

- Para la construcción de arquitectura de referencia es importante poder apalancarse en las arquitecturas de referencia de AWS.
- Teniendo en cuenta versiones.
- Ciclo de vida.
- Necesidades de la organización.
- Tipo de instancias.
- Arquitectura de microservicios.
- Automatización de la destrucción de Namespace.
- Definición de Cluster dentro de la organización.



# Generalidades de EKS

- **Nodos gestionados** Amazon EKS le permite crear, actualizar o finalizar nodos de trabajo para su clúster con un solo comando. Estos nodos también pueden aprovechar las instancias de spot de Amazon EC2 para reducir los costes. Los grupos de nodos administrados ejecutan instancias EC2 con las últimas AMI optimizadas para EKS en su cuenta de AWS, mientras que las actualizaciones y terminaciones agotan los nodos para garantizar que sus aplicaciones permanezcan.
- **Compatibilidad ARM** EKS es compatible con la tecnología ARM como ser las instancias Graviton 2 que brindan beneficios desde el rendimiento hasta los costos pero es importante que las imágenes construidas para los microservicios utilicen esta misma tecnología
- **Compatibilidad con Fargate** EKS admite AWS Fargate para ejecutar sus aplicaciones de Kubernetes utilizando computación sin servidor. Fargate elimina la necesidad de aprovisionar y administrar servidores, permite especificar y pagar recursos por aplicación y mejora la seguridad mediante el aislamiento de aplicaciones por diseño.
- **Apagado de un cluster** Como EKS IaaS no existe modelo de apagado del cluster sino es destrucción por lo cual la práctica recomendable es trabajar con Namespace y cuando no se utiliza automatizar la destrucción del namespace

# AWS EKS vs AWS ECS

- Seleccionar una solución de contenedor con los atributos más alineados con sus requisitos de aplicación o preferencias operativas.
- ECS ofrece mayor simplicidad al momento de implementarlo.
- ECS reduce tiempos de despliegue.
- EKS ofrece la flexibilidad de Kubernetes con la seguridad y la resistencia de ser un servicio administrado por AWS que está optimizado para los clientes que crean servicios de alta disponibilidad.
- ECS es totalmente auto administrado por AWS.
- Amazon ECS y Amazon EKS trabajan en conjunto a la perfección con operaciones compartidas, herramientas de seguridad integradas, IAM común y herramientas de administración coherentes para las opciones de red y computación. Aproveche la simplicidad de los servicios cohesivos de AWS en Amazon ECS o desarrolle los suyos propios utilizando la flexibilidad de Kubernetes en Amazon EKS.

