



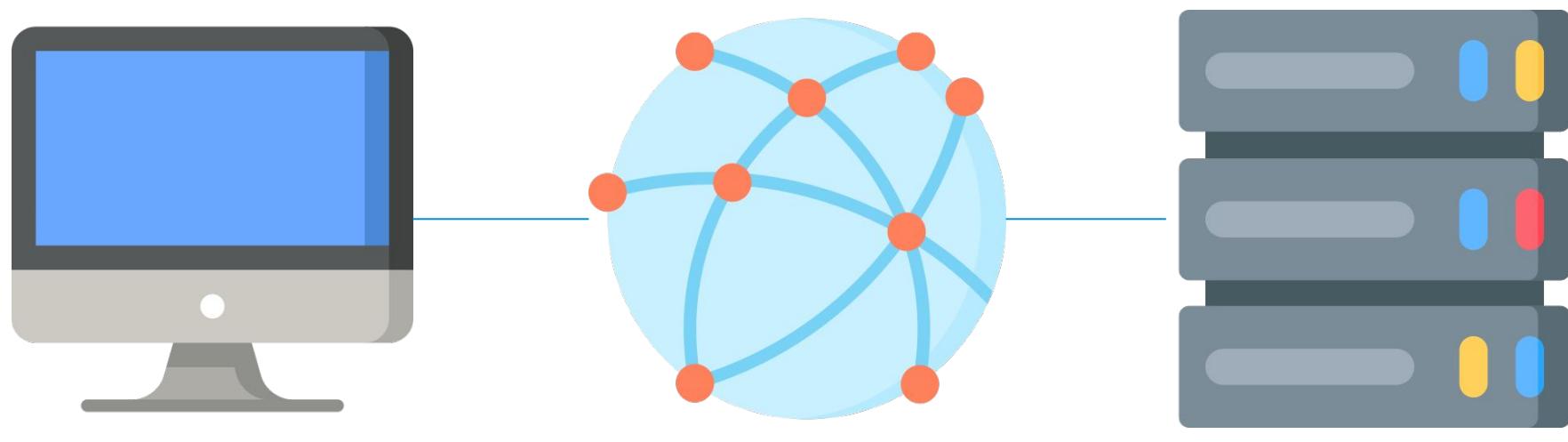
# Curso de Introducción a AWS: Fundamentos del Cloud Computing

Alexis Araujo

---

# Visión general de las IT tradicionales

# Cómo funciona un sitio web



**Clientes:** tienen  
direcciones IP

**Network**

**Servidores:** tienen  
direcciones IP

# Así mismo cuando enviamos un correo postal

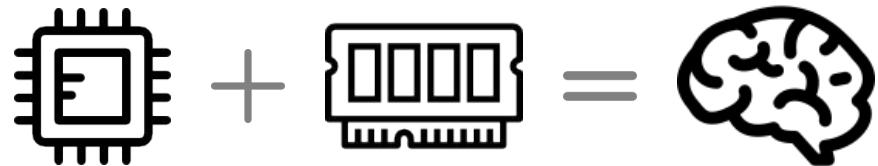


¿De qué está  
compuesto un  
servidor?

---

**Cómputo:** CPU

**Memoria:** RAM



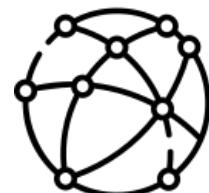
**Almacenamiento:**  
Datos.



**Bases de datos:**  
Información  
almacenada de forma  
estructurada.

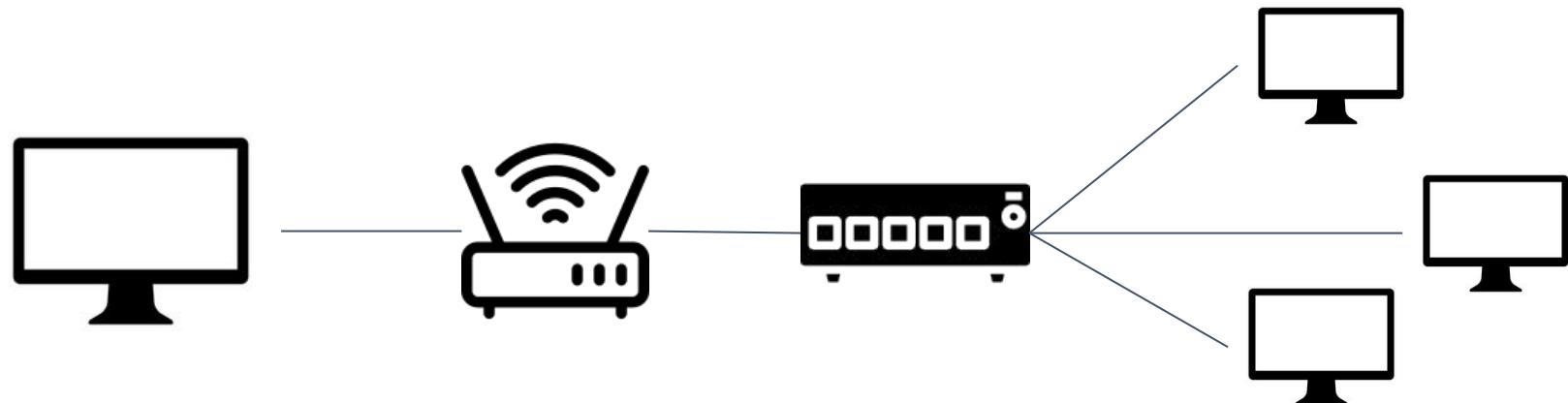


**Redes:**  
Routers, switch,  
servidor DNS.



# Terminología de IT

- **Redes**: Cables, routers y servidores conectados unos a otros.
- **Router**: Un dispositivo de red que reenvía paquetes de datos entre redes informáticas.
- **Switch**: Toma un paquete y lo envía al servidor/cliente correcto en la red.



# Diseño de infraestructura "tradicional"



# Problemas del enfoque de IT Tradicional



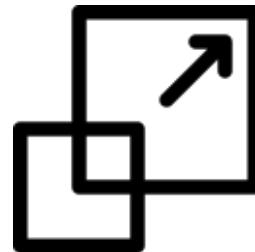
Renta



Mantenimiento



Reemplazar y agregar hardware



Escalamiento limitado



Contratar 24/7 para monitorear



Desastres Naturales

---

¿Qué es la  
computación  
en la nube?

“

*La computación en la nube es la **entrega bajo demanda** de computación, almacenamiento de bases de datos, aplicaciones y otros recursos de TI a través de una plataforma de servicios en la nube por medio de Internet con precios de pago por uso.*

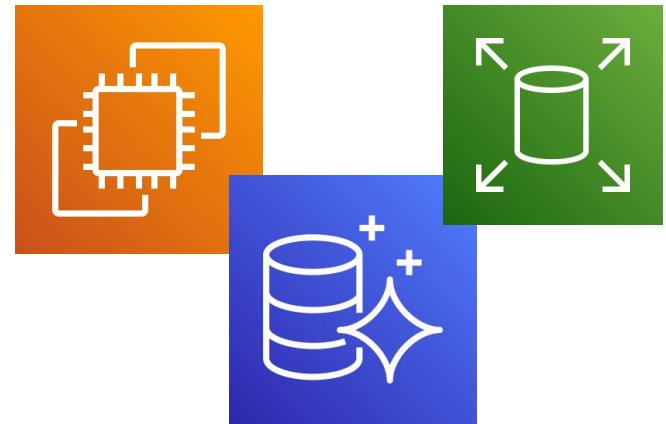
”

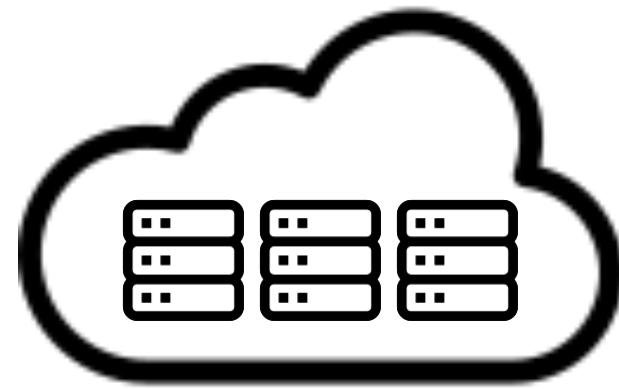
*¡Piensa en ello como  
simplemente alquilar la  
computadora de otra  
persona!*

---

# ¿Qué es la computación en la nube?

- Suministras el **tipo y tamaño exactamente correctos** de los recursos informáticos que necesitas.
- Puedes acceder **al instante** a todos los recursos que necesitas.
- Una **forma sencilla de acceder** a servidores, almacenamiento, bases de datos y un conjunto de servicios de aplicaciones.





# Has estado usando alguno de estos servicios



**Gmail**



**Dropbox**



**Netflix**

- Servicio de email en la nube.
- Pagas solo por tus emails almacenados (No infraestructura).

- Servicio de almacenamiento en la nube.
- Originalmente se construyó en AWS.

- Servicio de video en demanda.
- Construido en AWS.

# Tipos de modelos de computación en la nube

---

# Nube Privada

- Servicios de nube usados por una organización. (No está expuesta al público)
- Control total.
- Seguridad para aplicaciones sensibles.
- Satisface necesidades comerciales específicas.



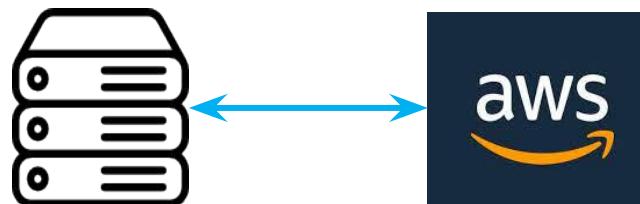
# Nube Pública

- Recursos propios en la nube y operados por proveedores de nube de terceros a través de internet.
- Seis ventajas del cómputo en la nube.



# Nube Hibrida

- Mantener algunos servidores en las instalaciones y extender otras capacidades en la nube.
- Control sobre activos sensibles en tu infraestructura privada
- Flexibilidad y rentabilidad de la nube pública.



# 5 características de la computación en la nube



Autoservicio  
en demanda



Amplio  
acceso a la  
red



Múltiples  
Inquilinos



Elasticidad y  
escalabilidad



Servicio  
medido

# 6 ventajas de la computación en la nube



Gastos de capital comercial sobre gastos operativos



Economías de escala



Dejar de adivinar la capacidad



Incrementar la velocidad y la agilidad

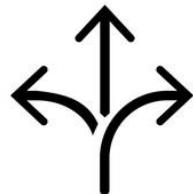


Dejar de gastar dinero en la ejecución



Globalizar en minutos

# Problemas resueltos por la nube



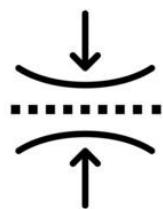
Flexibilidad



Rentabilidad



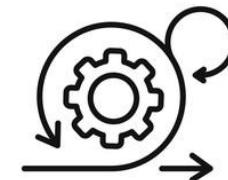
Escalabilidad



Elasticidad



Alta disponibilidad



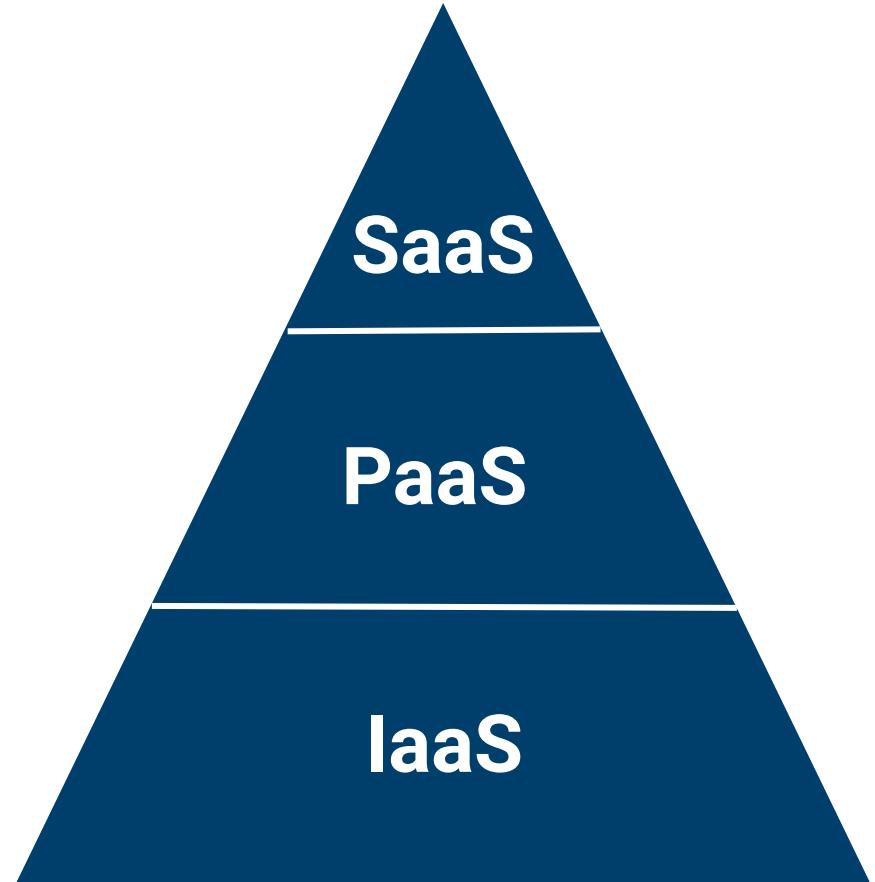
Agilidad

---

# Los diferentes tipos de cómputo en la nube

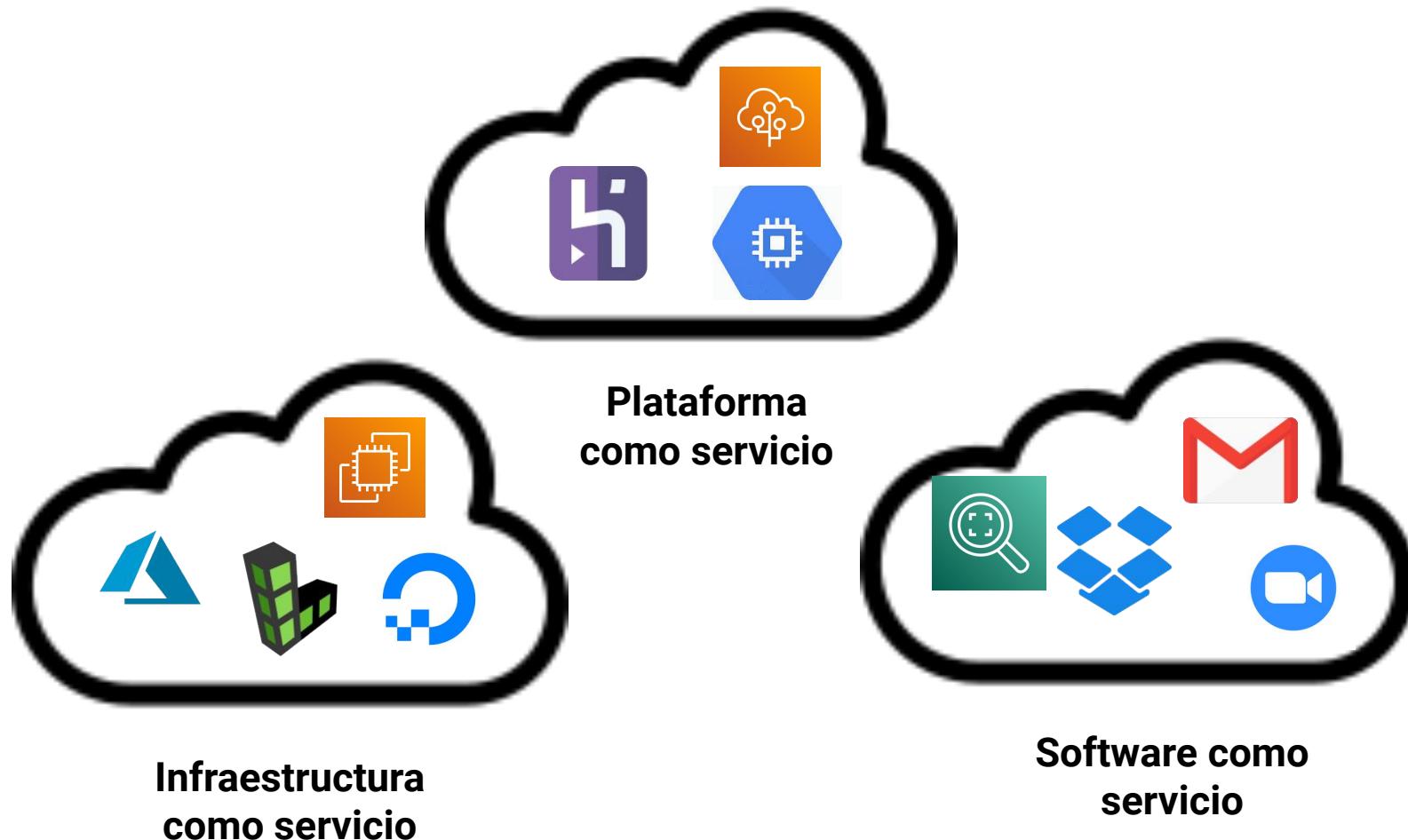
# Tipos de cómputo en la nube

- **IaaS:** Infraestructura como Servicio.
- **PaaS:** Plataforma como Servicio.
- **SaaS:** Software como Servicio.



On-premises	Infraestructura como servicio	Plataforma como servicio	Software como servicio
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking
		Administrado por nosotros	Administrado por otros

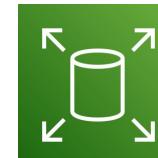
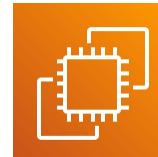
# Ejemplos de los tipos de computación de la nube



# Precio de la nube

AWS tiene 3 precios fundamentales:

- Computo.
- Almacenamiento.
- Información transferida fuera de la nube.



**Resuelve muchos de los problemas de IT tradicional relacionados con el costo.**

---

# Una pequeña historia de AWS

# Facts



**Benjamin Black**



**Chris Pinkham**

# Línea del tiempo de AWS

2002:  
Internamente  
lanzado

2004:  
Se lanza al  
público SQS

2007:  
Se lanza  
en Europa

2010:  
Se lanza  
route 53

The future

2003:  
La infraestructura de Amazon es  
una de sus principales fortalezas  
Idea para comercializar.

2006:  
Se relanza al  
público SQS,  
S3, EC2.

2009:  
Se lanza RDS.

2012:  
Se lanza  
DynamoDB.

# AWS en números

- AWS tuvo **\$35.02 mil millones en ingresos anuales** en 2019.
- AWS representó el **47% del mercado** en 2019 (Microsoft es 2º con 22%).
- Más de **1.000.000 de usuarios activos**.

Figure 1: Magic Quadrant for Cloud Infrastructure and Platform Services



Source: Gartner (July 2021)

---

# Una visión global



- Regiones
- Próximamente

Ohio	GovCloud Este	París	Ciudad del cabo	Seúl
Oregón	Canadá Central	Baréin	Singapur	Ningxia
Norte de California	São Paulo	Irlanda	Pekín	Osaka
Norte de Virginia	Frankfurt	Milán	Sídney	Bombay
GovCloud Oeste	Londres	Estocolmo	Tokio	Hong Kong

# AWS Regions

- AWS tiene regiones alrededor de todo el mundo.
- La mayoría de servicios tienen un ámbito regional.
- Los nombres pueden ser *us-east-1, ca-central-1...*

US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
<b>US West (Oregon)</b>	<b>us-west-2</b>
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

# Cómo escoger una región de AWS

**Si necesito lanzar una nueva aplicación, ¿dónde debo hacerlo?**



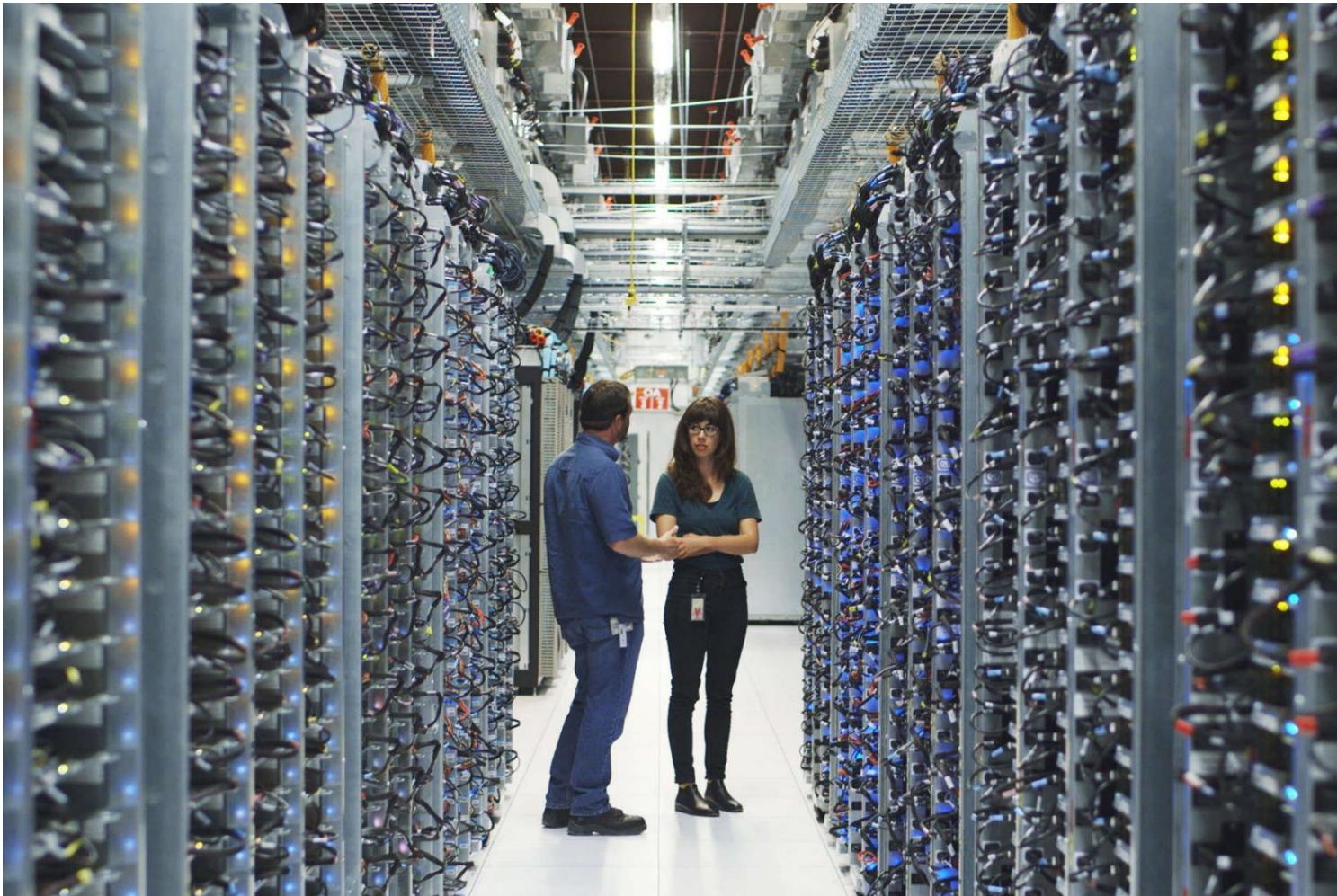
- **Cumplimiento** de los requisitos legales y de gobernanza de datos: los datos nunca abandonan una región sin su permiso explícito.
- **Proximidad** a los clientes: latencia reducida.
- **Servicios disponibles** dentro de una región: los nuevos servicios y las nuevas funciones no están disponibles en todas las regiones.
- Los **precios** varían de región a región y son transparentes en la página de precios del servicio.

Piensa que una zona de disponibilidad es un data center

---



# Un data center solo es un edificio lleno de servidores

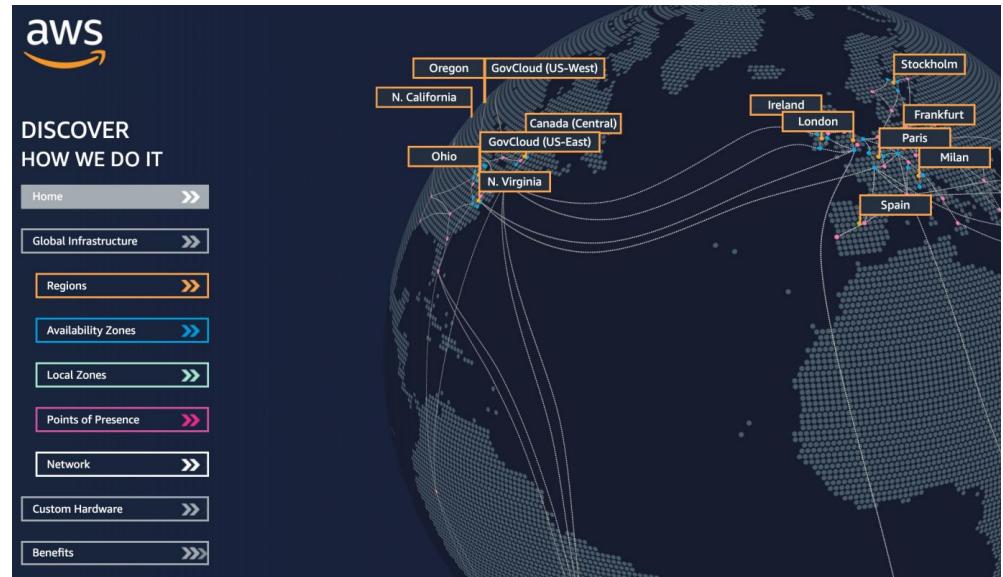


# Zonas de disponibilidad

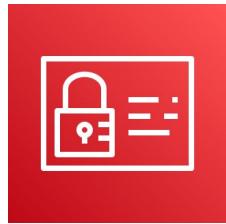


# Infraestructura global

- AWS Regiones.
- AWS Zonas de disponibilidad.
- AWS Centros de Datos.
- Ubicaciones de AWS Edge / puntos de presencia.



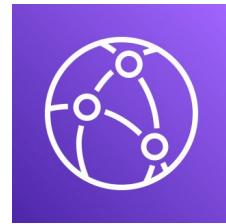
# Servicios Globales



IAM



Route 53

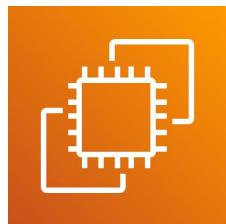


Cloudfront



WAF

# Servicios Regionales



EC2



Beanstalk

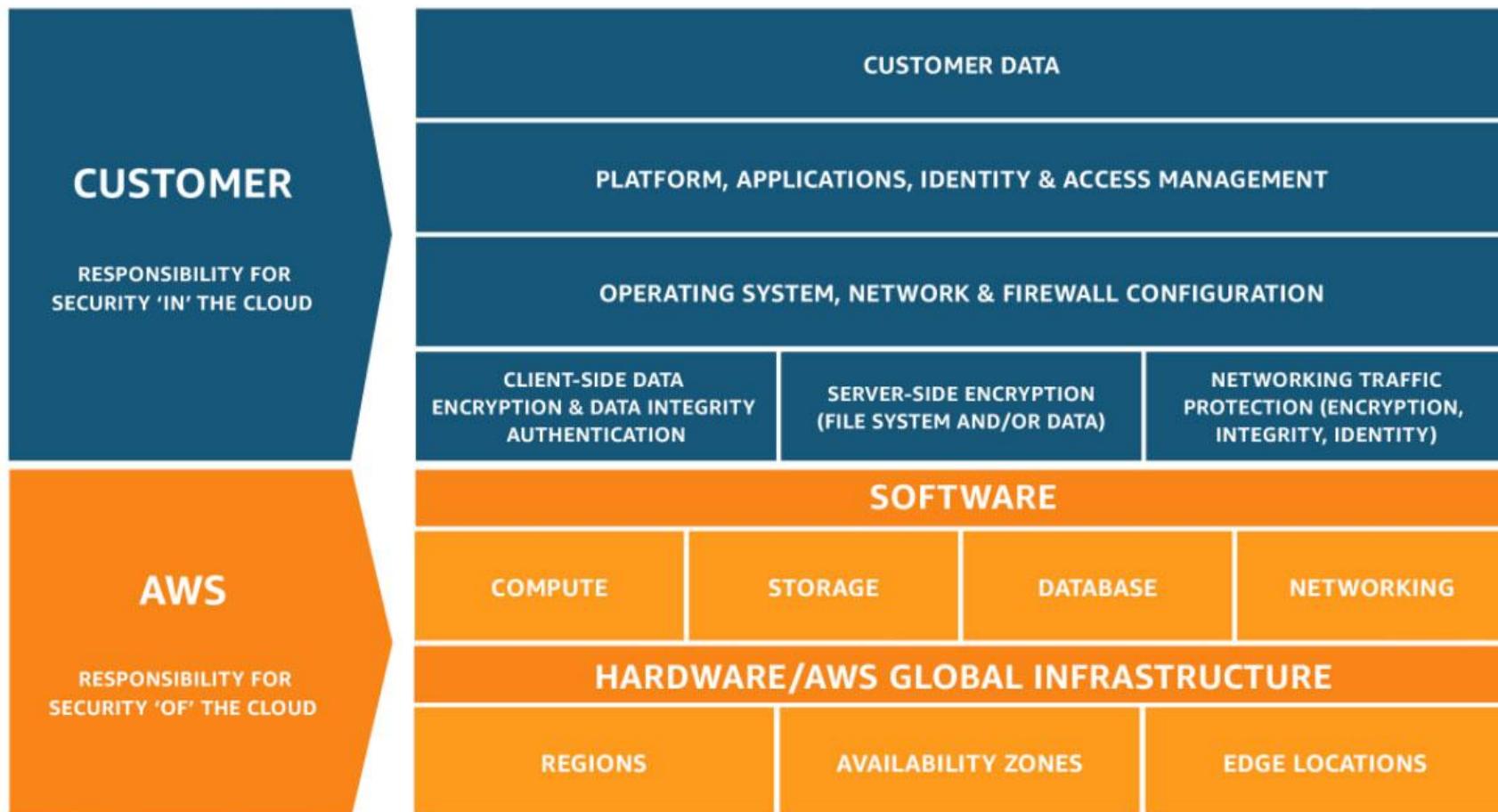


Lambda



Rekognition

# Diagrama del modelo de responsabilidad compartida



---

# Seguridad e Identidad

# Protección a Datos



Amazon Macie



AWS Key  
Management  
Service



AWS CloudHSM



AWS Certificate  
Manager



AWS Secrets  
Manager

# Protección de la infraestructura



AWS Shield



AWS Web Application Firewall



AWS Firewall Manager

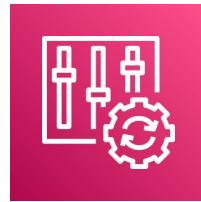
# Detección de amenazas



Amazon  
GuardDuty



Amazon  
Inspector



Amazon Config



Amazon  
CloudTrail

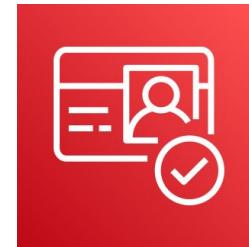
# Gestión de identidades



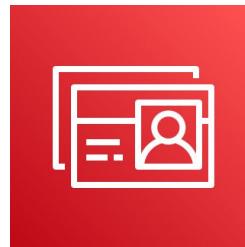
AWS Identity  
and Access  
Management



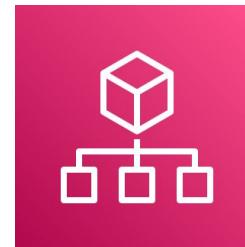
AWS Inicio de  
sesión único



Amazon  
Cognito



AWS Servicio  
de Directorio



AWS  
Organizaciones

---

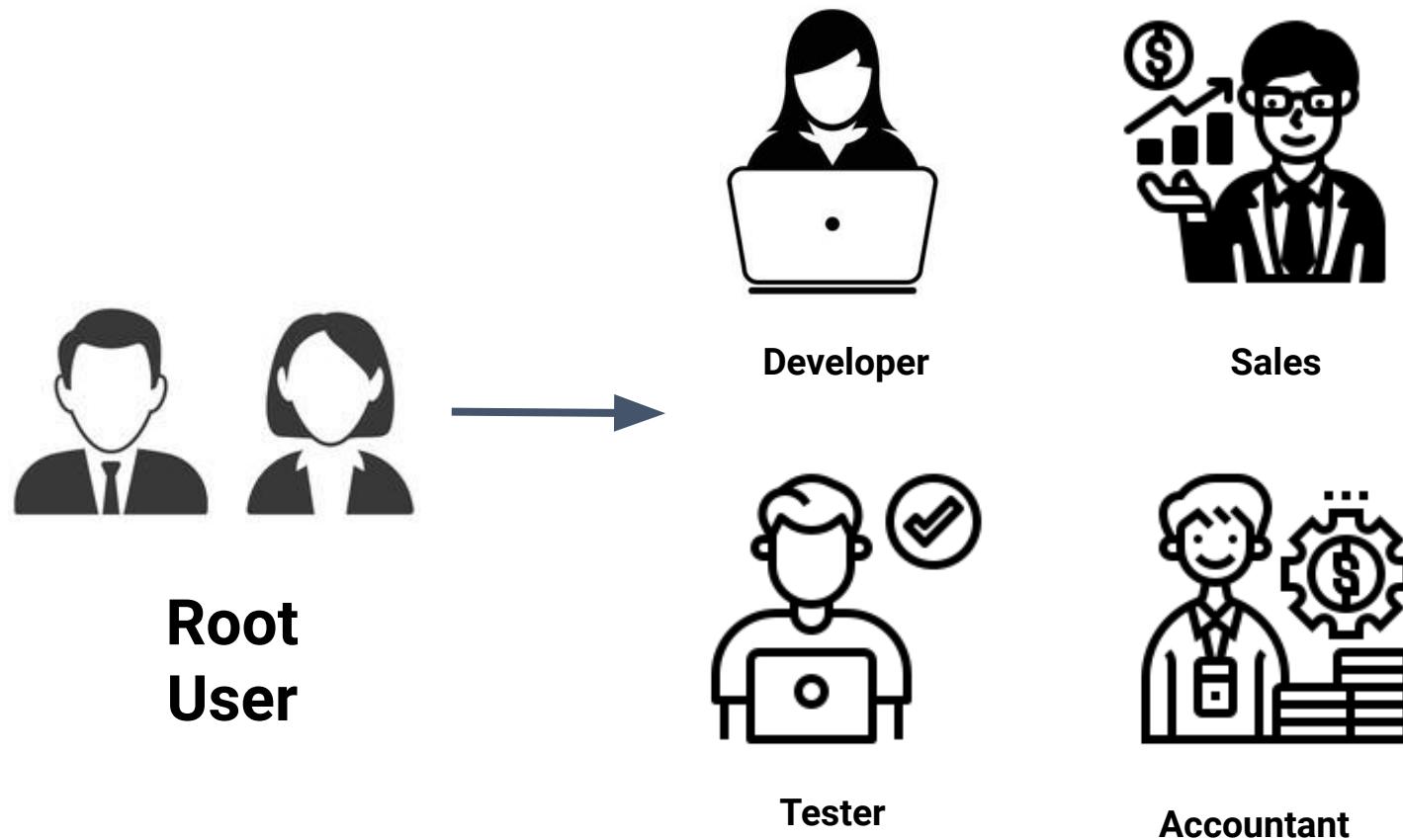
# IAM Ilustrado

# IAM

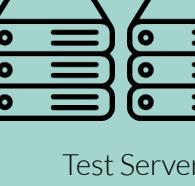


- Nos ayuda a administrar **quién puede acceder a qué** en los servicios y recursos de tu cuenta en AWS.
- Puedes crear usuarios y grupos.
- Establecer permisos para **permitir o denegar el acceso** a los recursos de AWS mediante el uso de políticas.
- IAM es gratuito y está incluido en todas las cuentas de AWS.

# Usuarios IAM



# Usuarios IAM

						
Developer	Developer	Developer	Developer			Release Engineer
Developer	Developer	Developer	Developer			Development Servers
Sales	Sales	Sales	Sales			Cloud Storage
Tester	Tester					Test Server

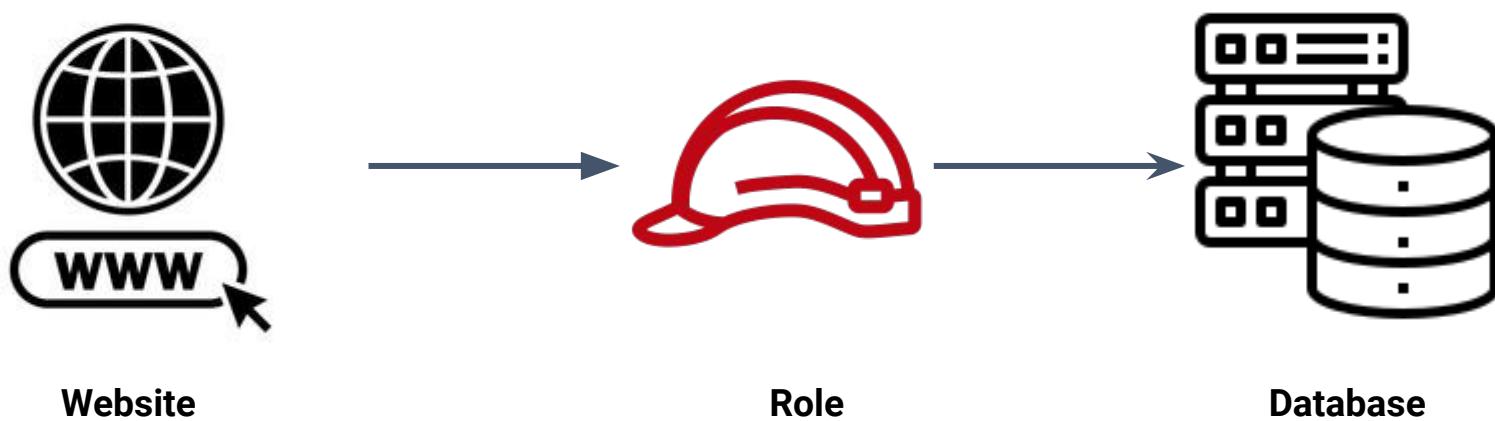
# Ejemplo de política de IAM: Usuario Administrador

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "*",  
      "Resource": "*"  
    }  
  ]  
}
```

# Ejemplo de política de IAM: Acceso a un bucket de S3

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3>ListBucket",  
      ],  
      "Resource": "arn:aws:s3:::bucket-name"  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3GetObject",  
        "s3PutObject",  
      ],  
      "Resource": "arn:aws:s3:::bucket-name/*"  
    }  
  ]  
}
```

# IAM Roles



---

# Secrets Manager

# Ejemplo de una conexión de MySQL hardcodeada

```
import mysql.connector

connection = mysql.connector.connect(
    host="localhost",
    database="mydb",
    user="root",
    password="root"
)
```

# Acerca del servicio



- Protege los secretos que son necesarios para acceder a sus aplicaciones, servicios y recursos.
- Rotarlos automáticamente.
- Los secretos pueden ser contraseñas, claves y tokens.

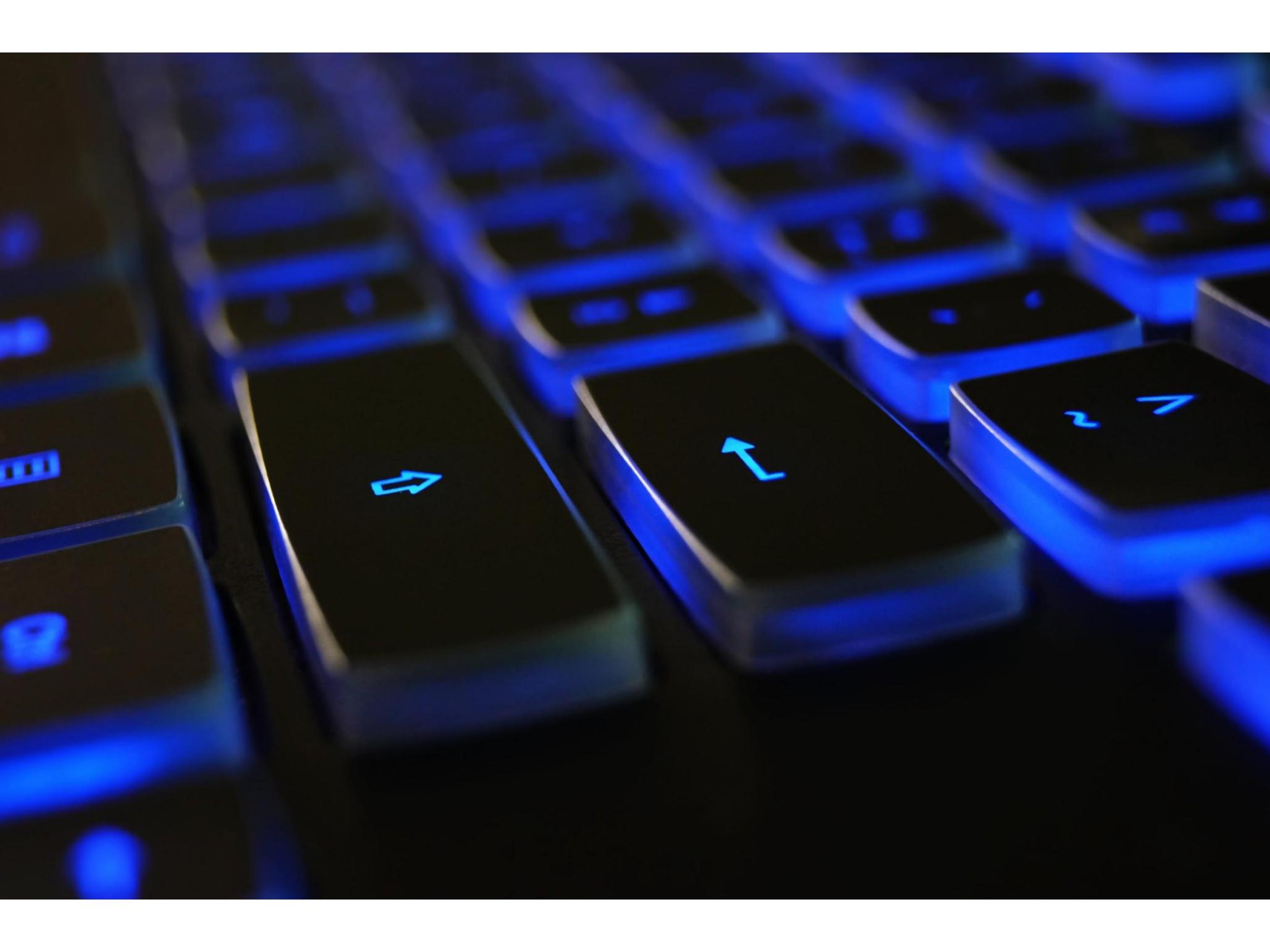
# Ejemplo secrets manager MySQL conexion

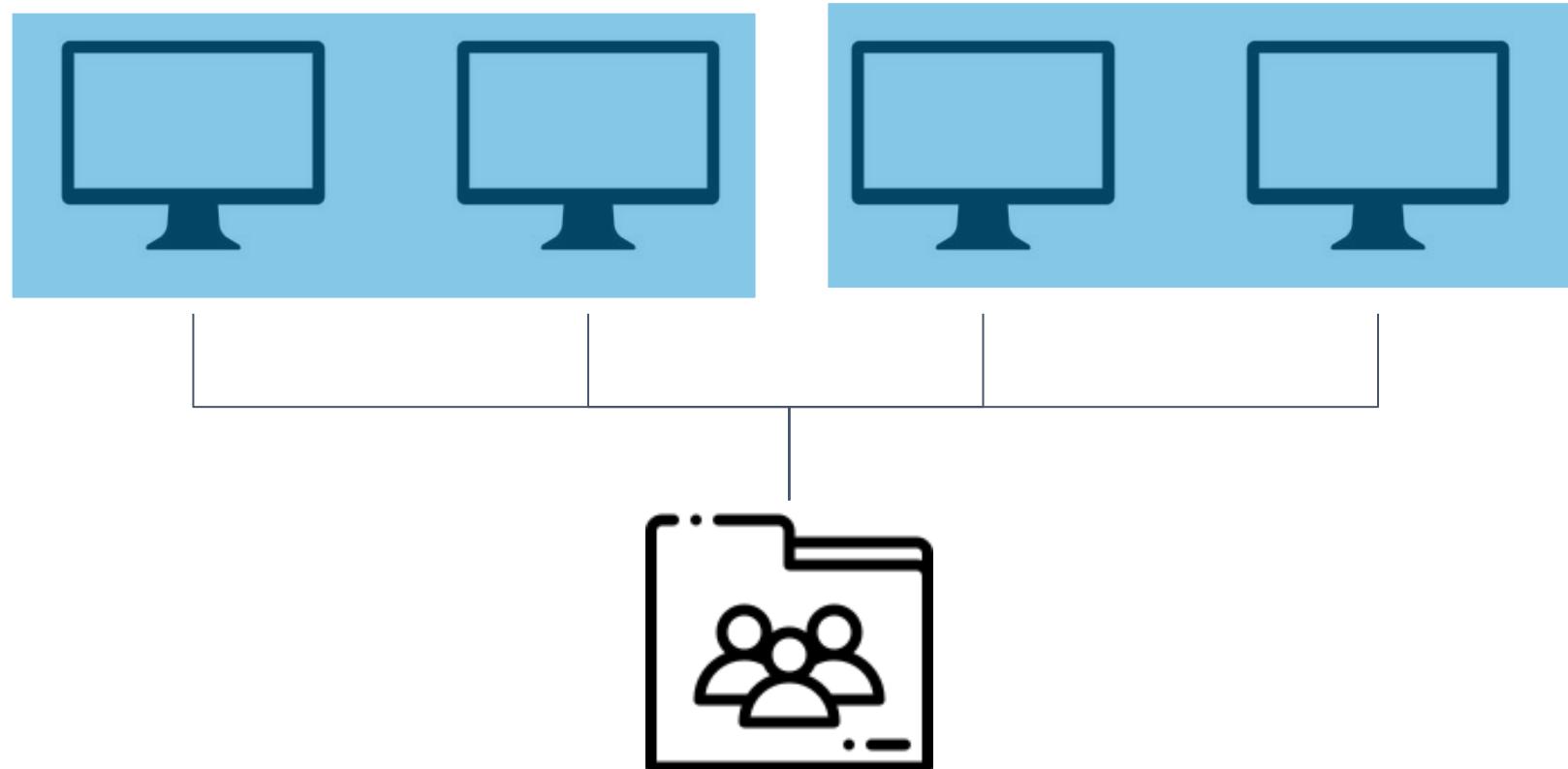
```
import mysql.connector

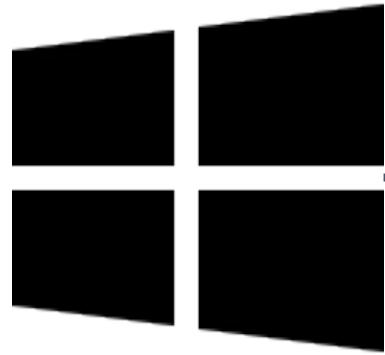
connection = mysql.connector.connect(
    host="localhost",
    database="mydb",
    user="root",
    password=get_secret_value['SecretString']
)
```

---

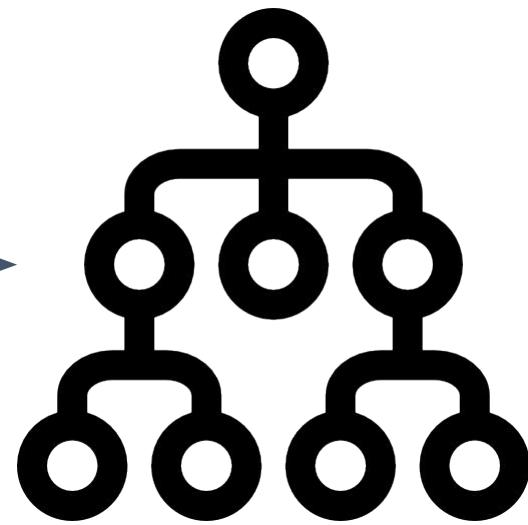
# Servicio de directorio dentro de AWS





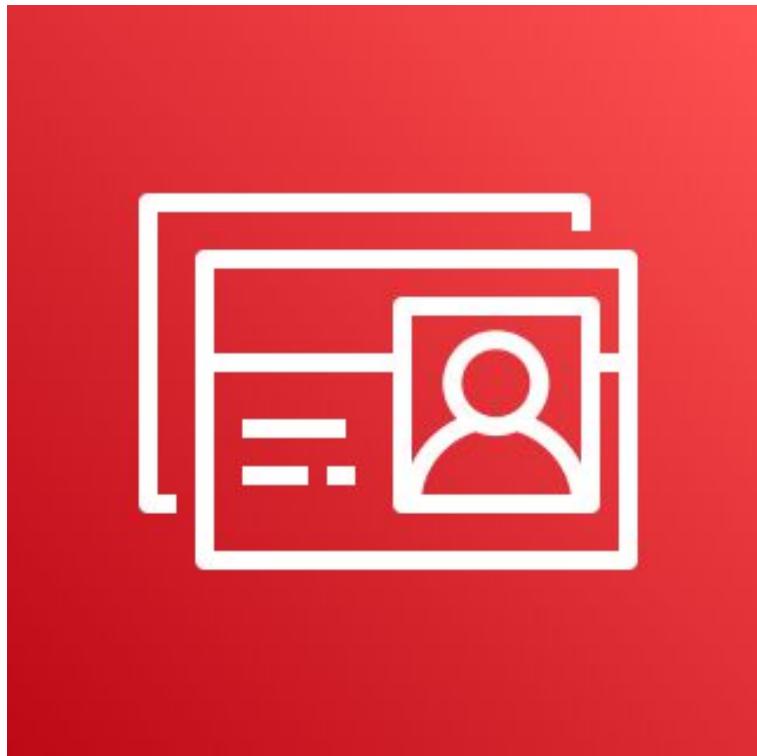


**Windows Sistema Operativo**



**Servicio Directorio Activo**

# AWS Directory Service



- Directorio Activo administrado.
- Opción de directorio activo simple.
- Conector AD.
- Servicio distribuido con error automático.
- Compatible con otros servicios de AWS.

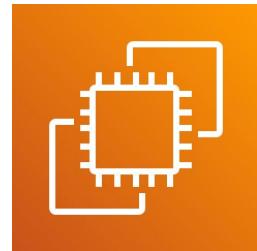
# Curso de Introducción a AWS: Almacenamiento, Cómputo y Bases de Datos

Alexis Araujo

---

# Cómputo dentro de AWS

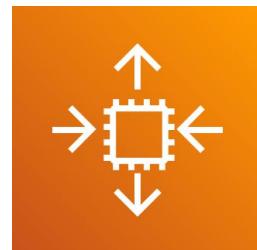
# Instancias



**Amazon EC2**  
**Máquinas virtuales**  
**seguras y**  
**redimensionables**



**Amazon EC2**  
**Spot**



**Amazon EC2**  
**AutoScaling**



**Amazon EC2**  
**LightSail**

# Contenedores



Amazon ECS



Amazon ECR



Amazon EKS

# Serveless

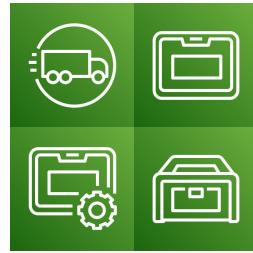


Amazon  
Lambda

# Edge



**Amazon  
Outposts**



**Amazon Snow  
Family**



**AWS Wavelength**



**VMWare AWS**

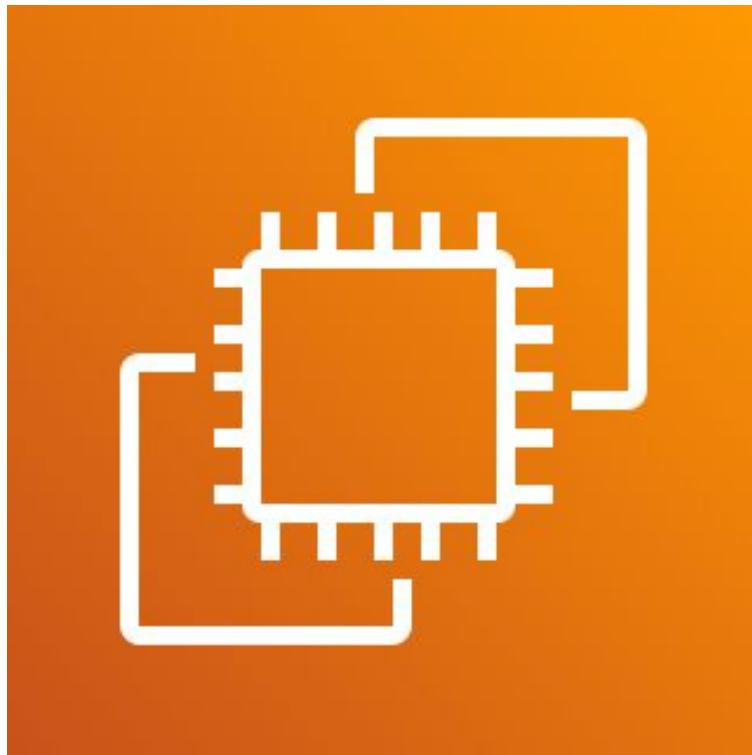


**AWS Local Zones**

---

# Explorando EC2

# AWS EC2



- Permite alquilar computadoras virtuales.
- Diferentes tipos de EC2 con diferente CPU, RAM y almacenamiento.
- Instancias optimizadas en computación.
- Paga por hora o segundo.

# Ejemplo de precios.

$$24 \times 0.10 = 2.40$$



$$24 \times 0.10 = 2.40$$

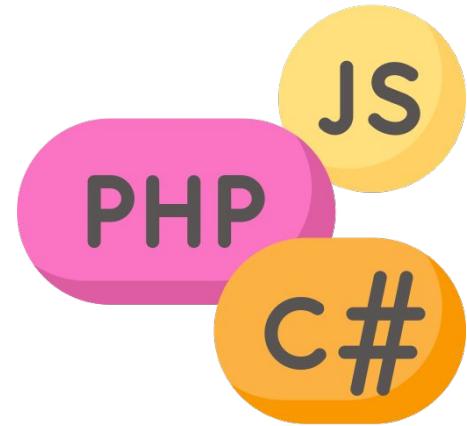
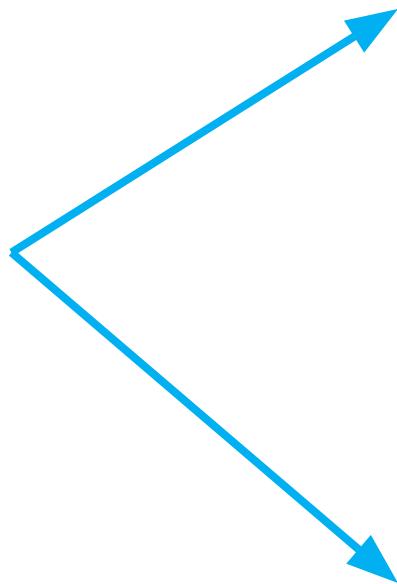
# Opciones y precios bajo demanda

Nombre de la instancia	Especificaciones	Precio
<b>t3.nano</b>	2 vCPU's, 0.5 GiB RAM	\$0.0052/hour
<b>t3.xlarge</b>	4 vCPU's, 16 GiB RAM	\$0.1664/hour
<b>c6g.8xlarge</b>	32 vCPU's, 64 GiB RAM	\$1.088/hour
<b>X1e.xlarge</b>	128 vCPU's 3904 GiB RAM, 2x 1920 GB SSD	\$26.688/hour

---

# Contenedores

# Problema Común



# Código que usa una librería.

```
// create a web server
const http = require('http');
const requestListener = function (request, response) {
  response.writeHead(200, { 'Content-Type': 'text/plain' });
  response.end('Hello World\n');
};
const server = http.createServer(requestListener);
server.listen(8080);
```

# Versiones de librerías

```
{  
  "name": "my-awesome-package",  
  "dependencies": {  
    "axios": "0.21.0",  
    "express": "4.17.1",  
  }  
}
```

```
→ express --version  
4.16.1
```

```
→ nvm ls  
->      v16.13.0  
          system  
default -> node  (-> v16.13.0)  
system -> default (-> v16.13.0)  
node -> stable  (-> v16.13.0) (default)  
stable -> 16.13 (-> v16.13.0) (default)  
iojs -> N/A (default)  
unstable -> N/A (default)  
lts/* -> lts/gallium (-> v16.13.0)  
lts/argon -> v4.9.1 (-> N/A)  
lts/boron -> v6.17.1 (-> N/A)  
lts/carbon -> v8.17.0 (-> N/A)  
lts/dubnium -> v10.24.1 (-> N/A)  
lts/erbium -> v12.22.7 (-> N/A)  
lts/fermium -> v14.18.1 (-> N/A)  
lts/gallium -> v16.13.0
```

“

*Y ahora... ¿Quién podrá  
ayudarnos?*

”

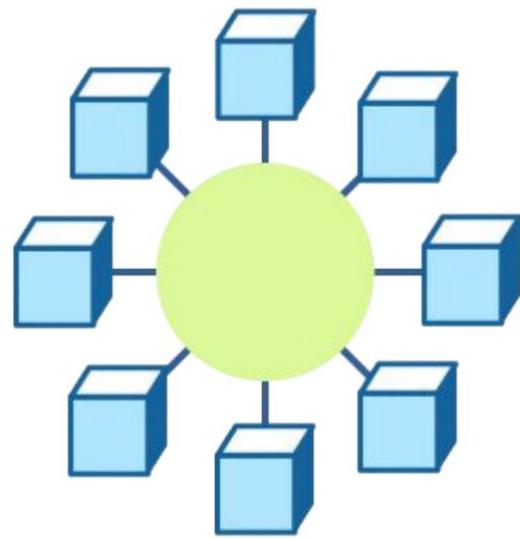
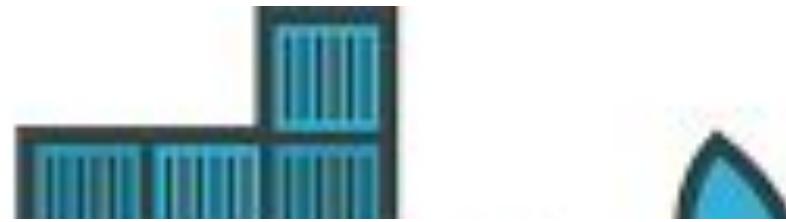
– *Desarrolladores desesperados por los  
problemas del software profesional.*



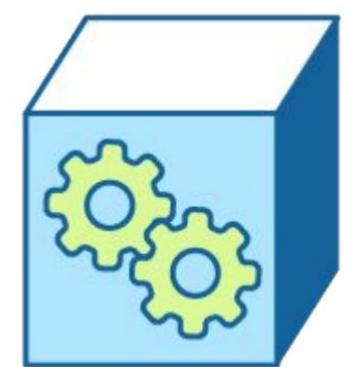




Construir

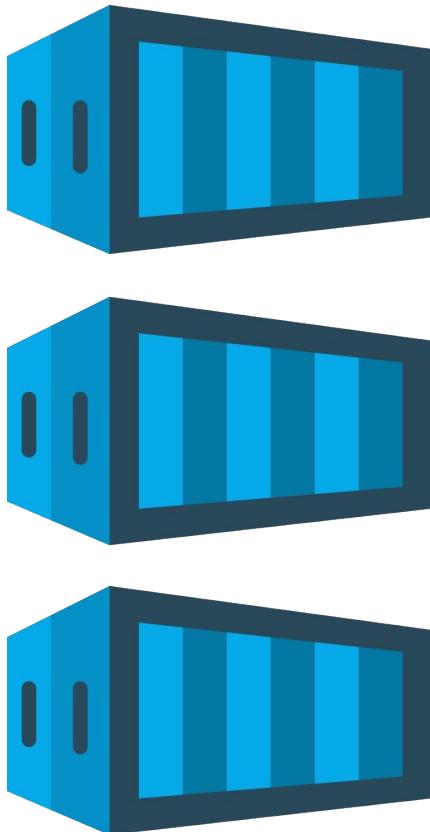


Distribuir



Ejecutar

## Contenedores

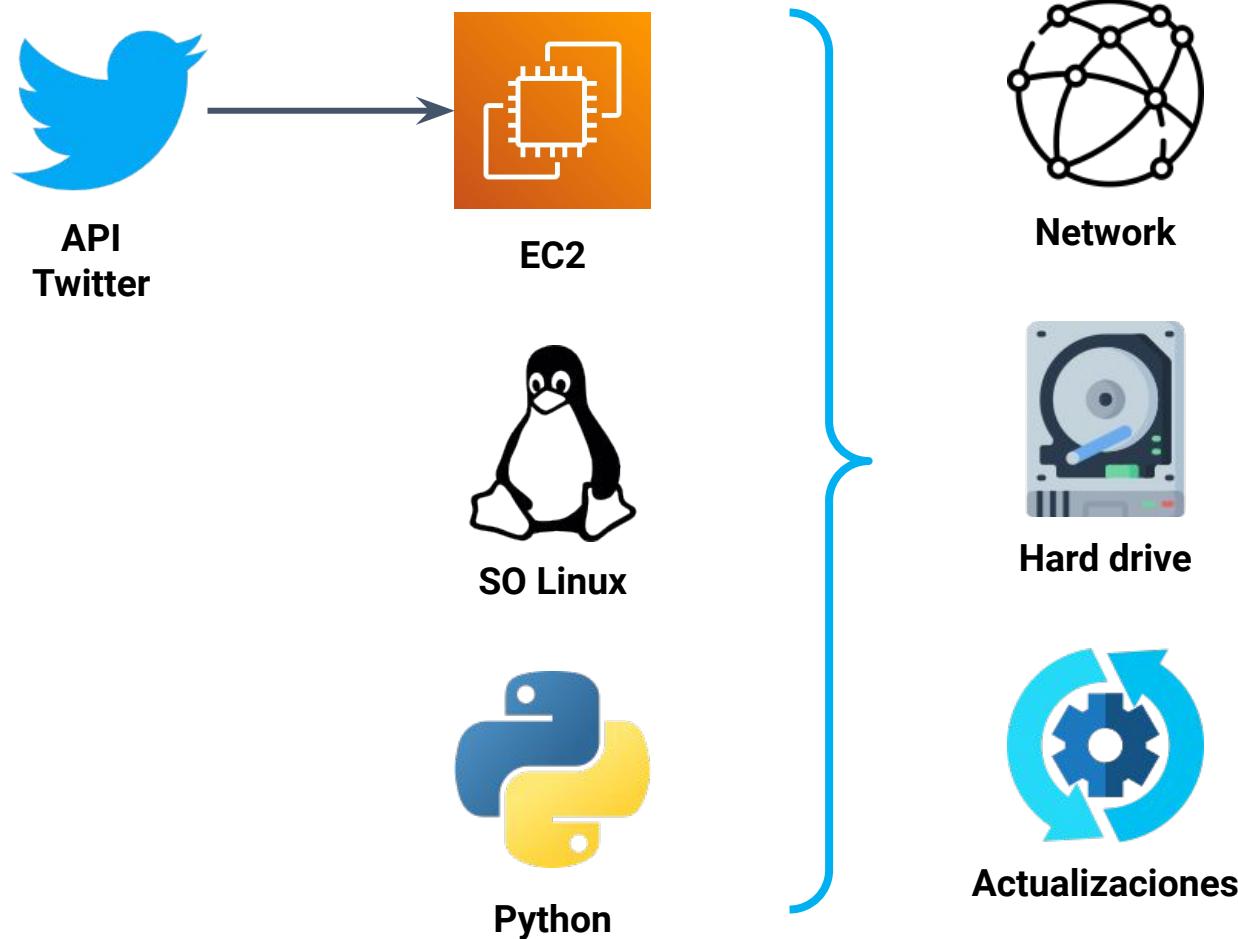


Amazon Elastic  
Container Service

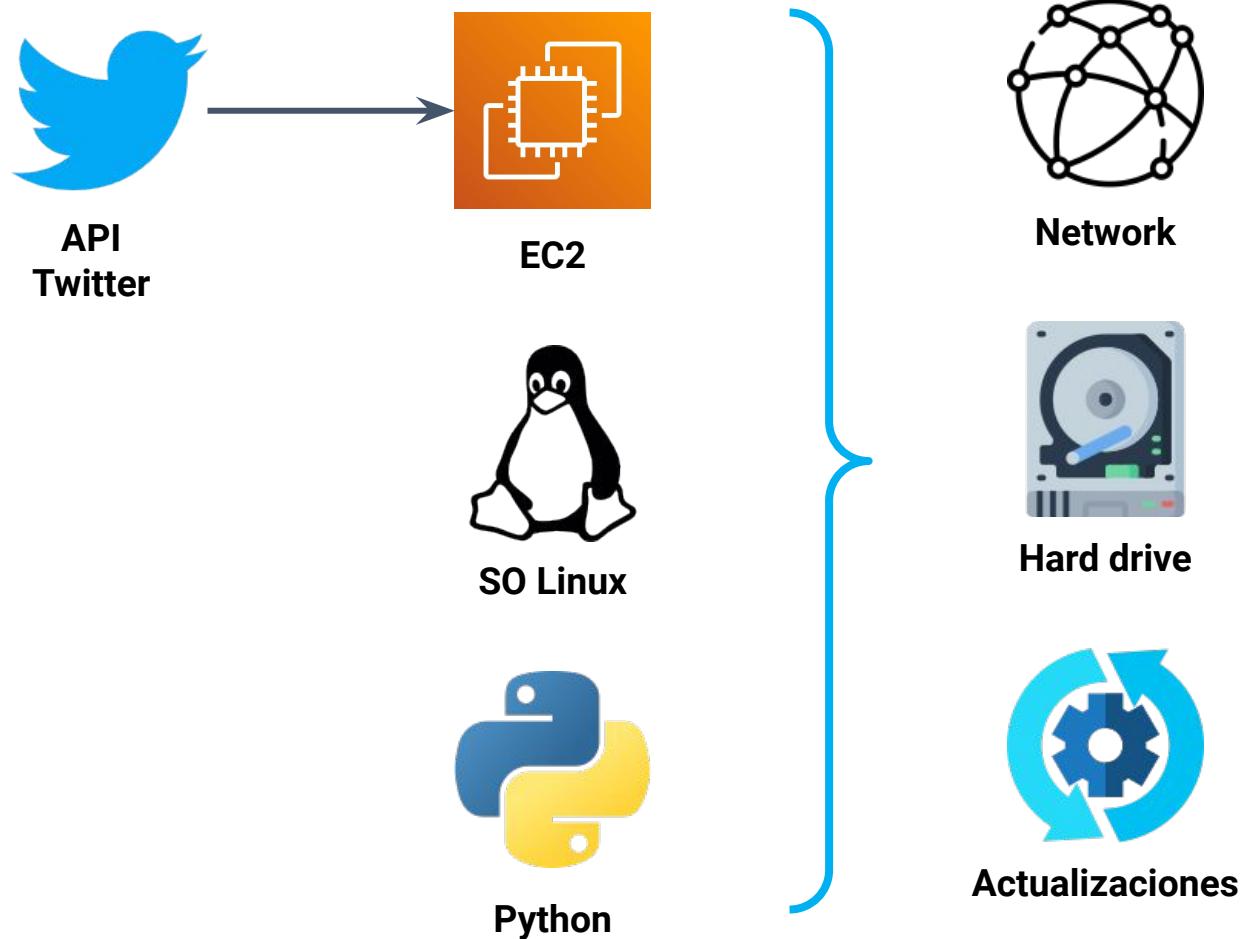
---

# Aprendiendo sobre lambda

# Ejemplo de una lambda



# Ejemplo de una lambda



# AWS Lambda



- Servicio informático sin servidor.
- Ejecuta su código en respuesta a eventos.
- AWS Lambda ejecuta el código en alguna parte.

# Caso de uso



# Facturación por milisegundos

- El precio depende del uso de RAM.
- 128MB RAM x 30M eventos por mes.
- \$11.63 USD.

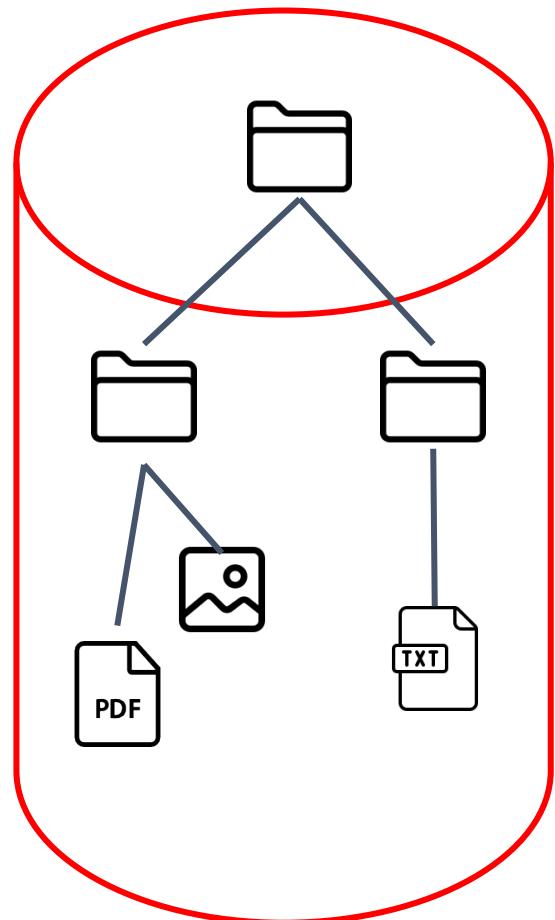


---

# Almacenamiento

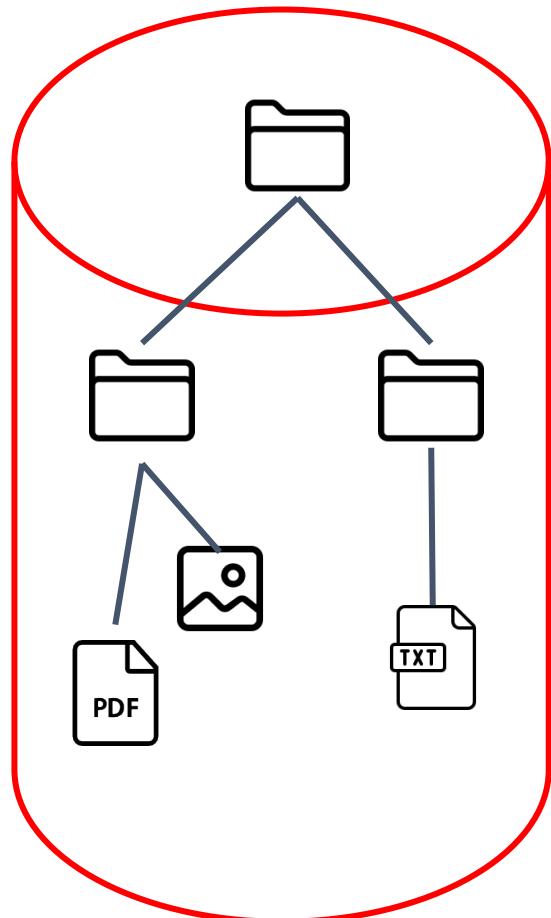
# Almacenamiento

## Archivo



# Almacenamiento

Archivo



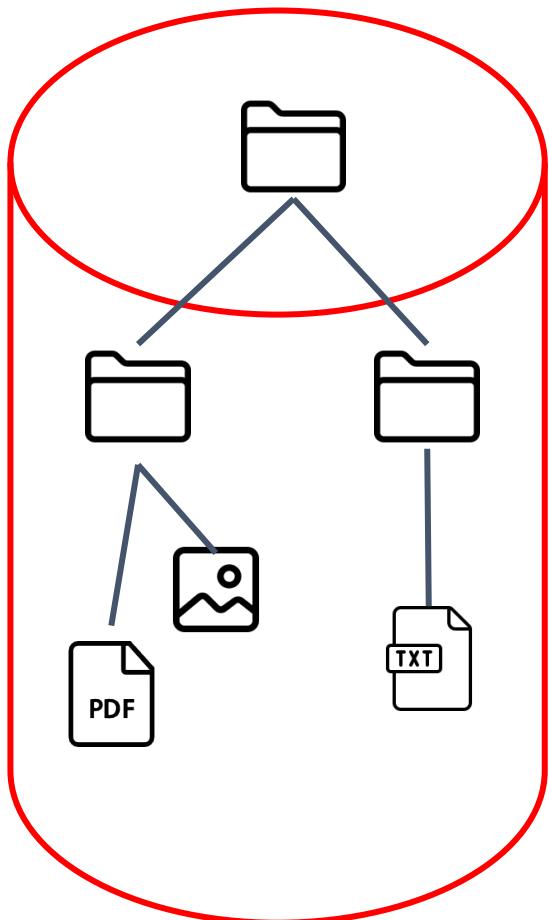
Bloque

The diagram illustrates a block-based storage structure. It features a 9x6 grid of numbers, each representing a data block. The grid is outlined with a blue border.

0	1	2	3	4	5
8	3	2	2	1	1
1	4	3	3	2	2
6	5	4	4	3	2
2	6	5	5	4	3
4	9	6	9	5	2
5	8	7	8	7	3
6	7	7	8	6	4

# Almacenamiento

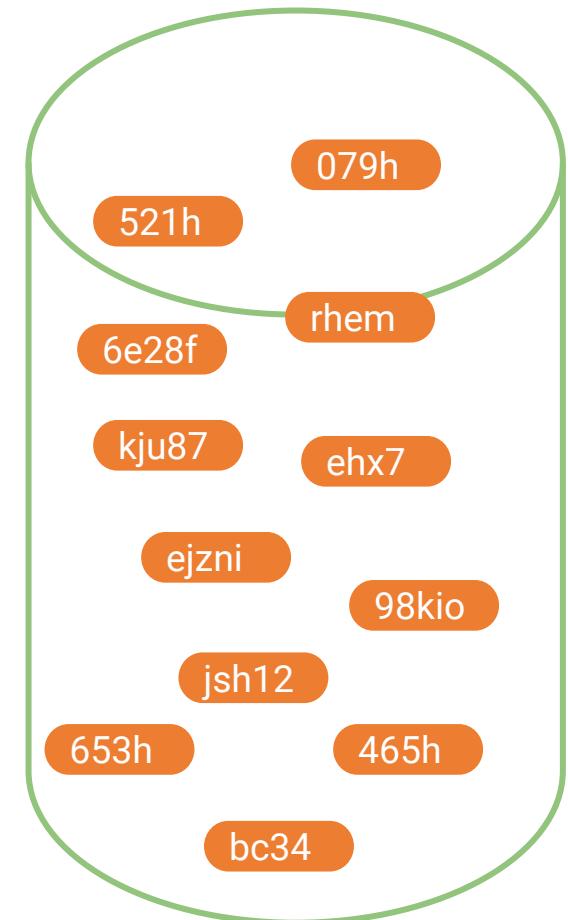
Archivo



Bloque

0	1	2	3	4	5
8	3	2	2	1	1
1	4	3	3	2	2
6	5	4	4	3	2
2	6	5	5	4	3
4	9	6	9	5	2
5	8	7	8	7	3
6	7	7	8	6	4

Objeto



# Almacenamiento de Archivos

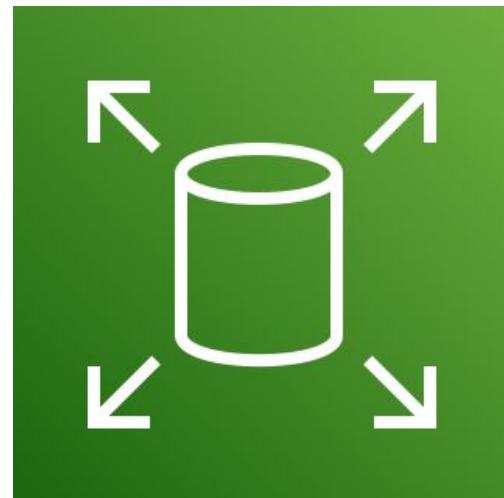


**Amazon  
EFS**



**Amazon FSx  
For Windows  
File Server**

# Almacenamiento en bloque



Amazon  
EBS

# Almacenamiento en objeto



**Amazon  
S3**

# Respaldo



**Amazon  
Backup**

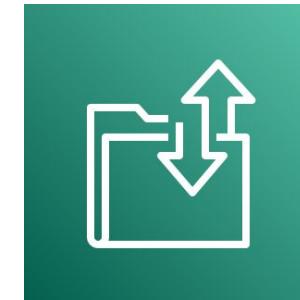
# Transferencia de datos



**AWS  
Storage  
Gateway**



**AWS  
DataSync**



**AWS  
Transfer  
Family**

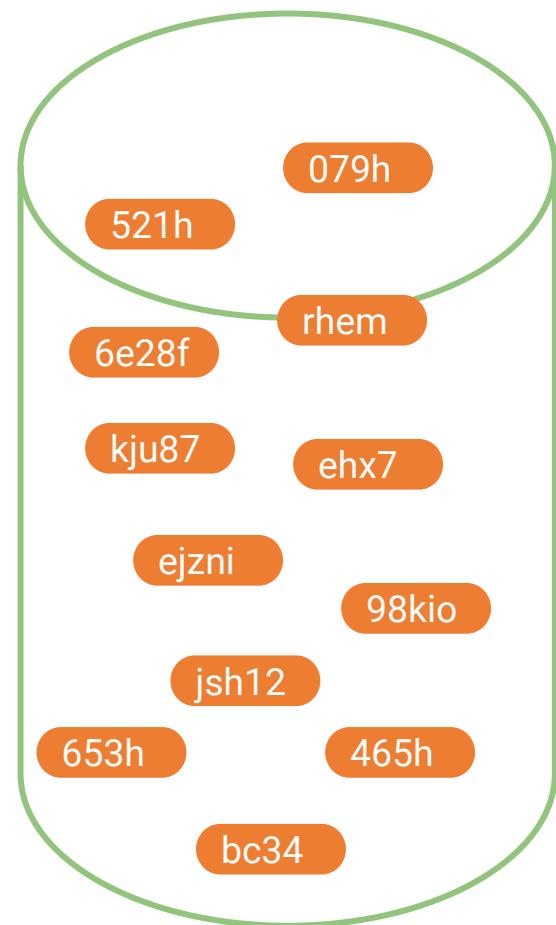
---

# S3 y S3 Glacier

# Amazon S3



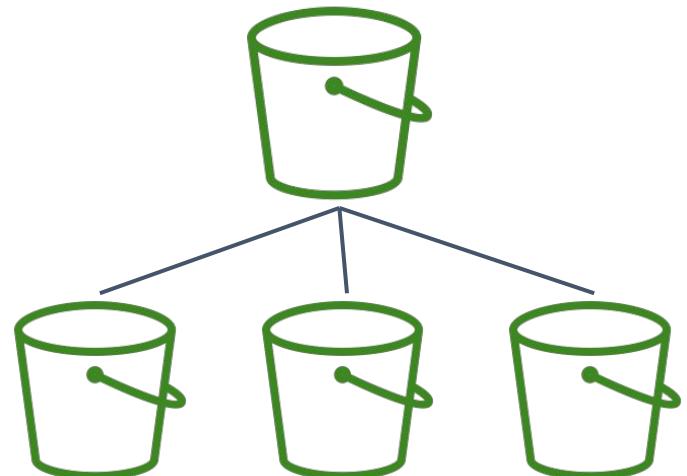
**Objeto**



# Durabilidad líder en la industria

11 9's

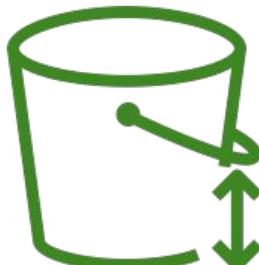
99.999999999%



# Clases de almacenamiento



S3  
Standard



S3  
Standard-IA



S3 One  
Zone-IA



S3 Glacier

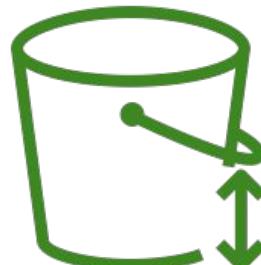


S3 Glacier  
Deep  
Archive

# Clases de almacenamiento



S3  
Standard



S3  
Standard-IA



S3 One  
Zone-IA



S3 Glacier



S3 Glacier  
Deep  
Archive



S3  
Intelligent-  
Tiering

# Amazon S3 Glacier



- Archivo de datos y copia de seguridad a largo plazo.
- \$1/TB/month
- Funcionalidad de consulta

# Amazon S3 Glacier

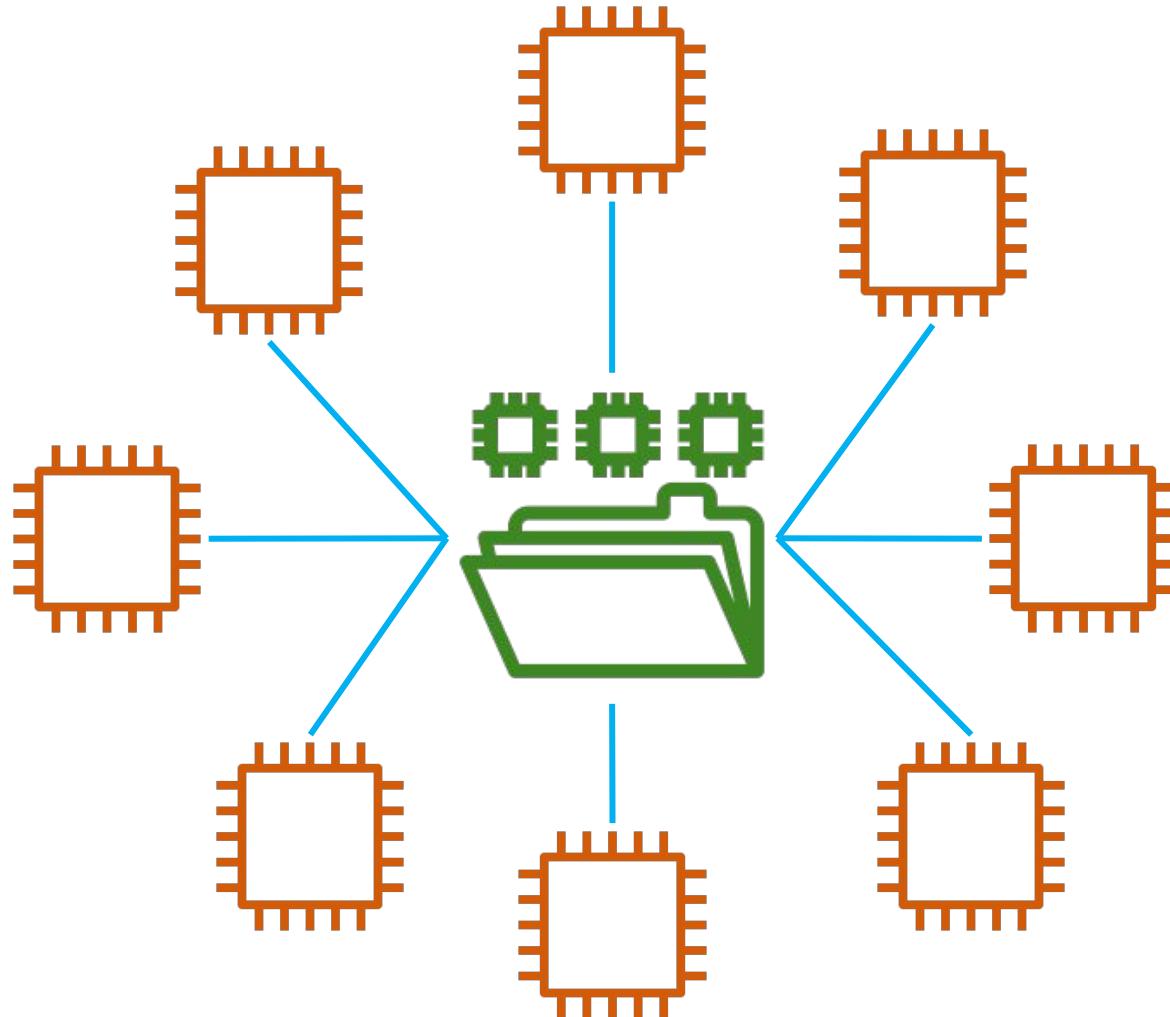


- Archivo de datos y copia de seguridad a largo plazo.
- \$1/TB/month
- Funcionalidad de consulta

---

# Explicando EFS

# EFS

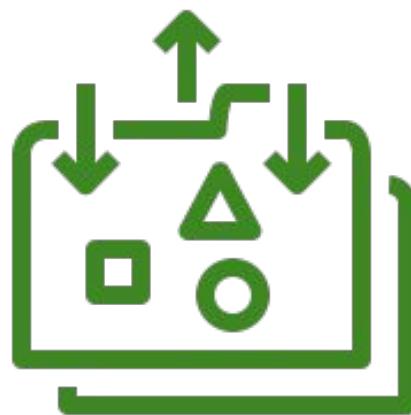


# Amazon EFS



- Altamente disponible y duradero.
- EFS ha incorporado protección contra una interrupción de la zona de disponibilidad

# Clases de almacenamiento

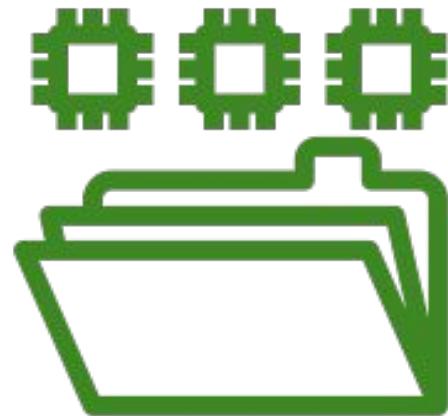


**Standard**



**Standard  
Infrequent  
Access**

# Características



# Encriptacion



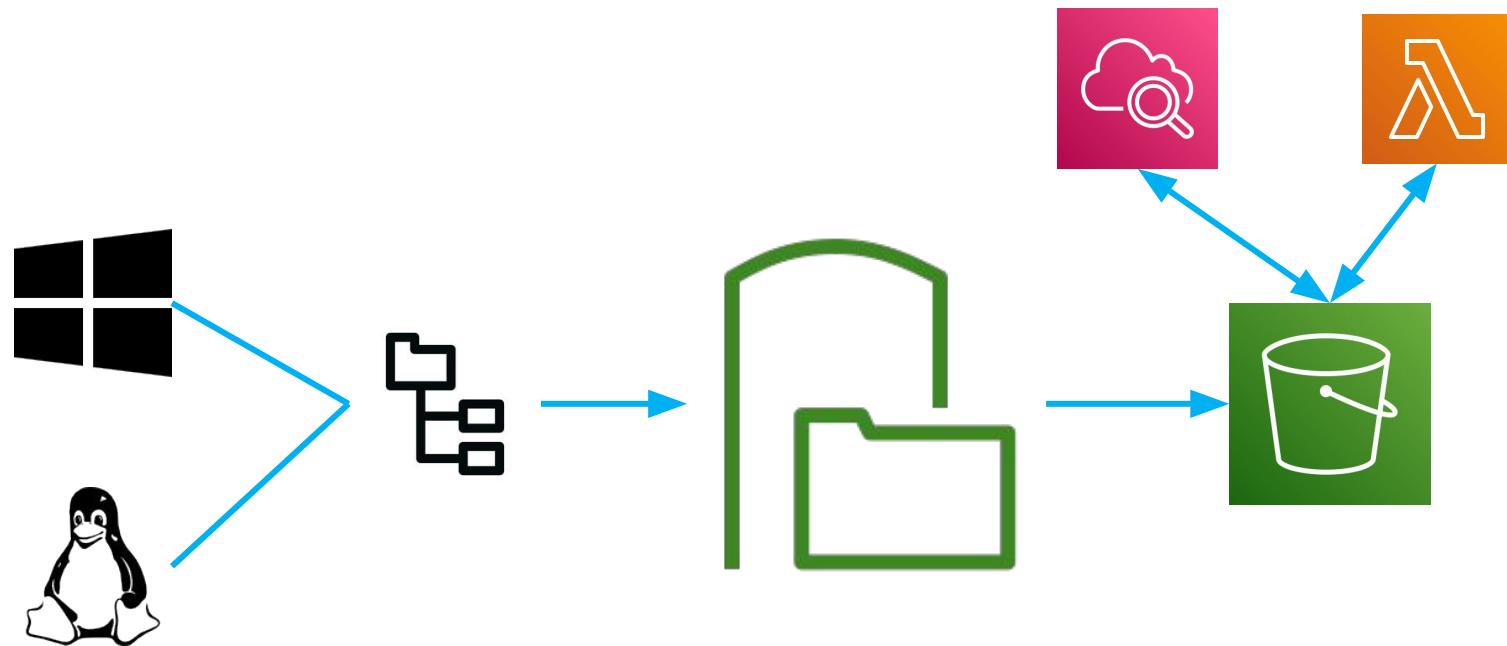
---

# AWS Storage Gateway

# AWS Storage Gateway



# File Gateway



# Tape Gateway



COMMVAULT®

veeAM

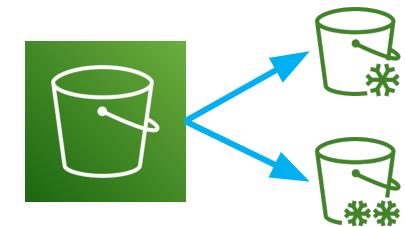
VERITAS



Virtual  
tape  
library



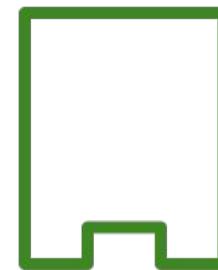
Tape  
gateway



# Volume Gateway



**Volume  
gateway**



**Non-cached  
volume**



**Cached  
volume**

# Storage Gateway



File  
gateway



Tape  
gateway



Volume  
gateway

---

# Definición de base de datos

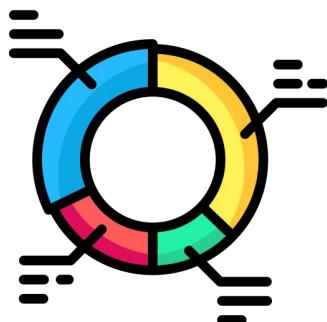
# Bases de datos

SQL

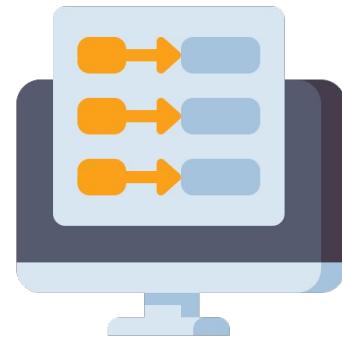
Relational



Analytical (OLAP)

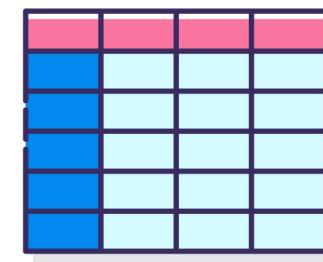


Key - Value

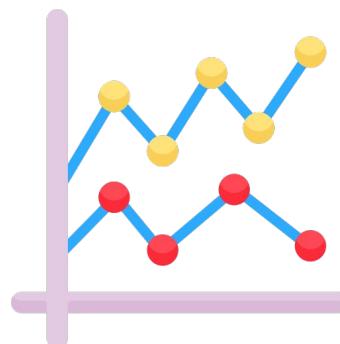


NoSQL

Column Family



Graph



Document



# Bases de datos relacionales



**Amazon Aurora**



**Amazon Relational  
Database Service  
(Amazon RDS)**



**Amazon Redshift**

# Bases de datos clave-valor

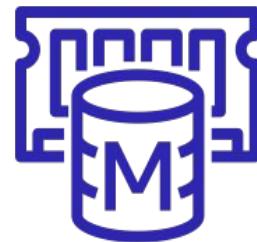


Amazon  
DynamoDB

# Bases de datos en memoria



Amazon  
**ElastiCache**



ElastiCache  
para  
Memcached



ElastiCache  
para Redis

# Bases de datos de documentos



**Amazon DocumentDB  
(compatibilidad con  
MongoDB)**

---

# Revisando RDS

# Ejemplo de una base de datos relacional

Student ID	First Name	Last Name	DOB	Phone	Address
1	Rozele	Frankum	1/30/2021	564-621-8383	9 Myrtle Center
2	Monika	Petschelt	4/2/2021	745-829-9348	057 Northview Avenue
3	Bruis	Lomasna	4/7/2021	537-190-6357	266 Clove Trail
4	Almira	Cloutt	12/18/2021	140-756-0025	935 Grasskamp Plaza
5	Nicol	Hampson	10/13/2021	492-851-5815	7 Lawn Alley

# Motores de base de datos



**ORACLE**



# Amazon RDS



- Facilita la configuración.
- Servicio completamente administrado.
- Altamente escalable.
- Copias de seguridad automáticas.
- Realmente rentable.

# Amazon RDS



- Facilita la configuración.
- Servicio completamente administrado.
- Altamente escalable.
- Copias de seguridad automáticas.
- Realmente rentable.

---

# Discutiendo sobre DynamoDB

# DynamoDB

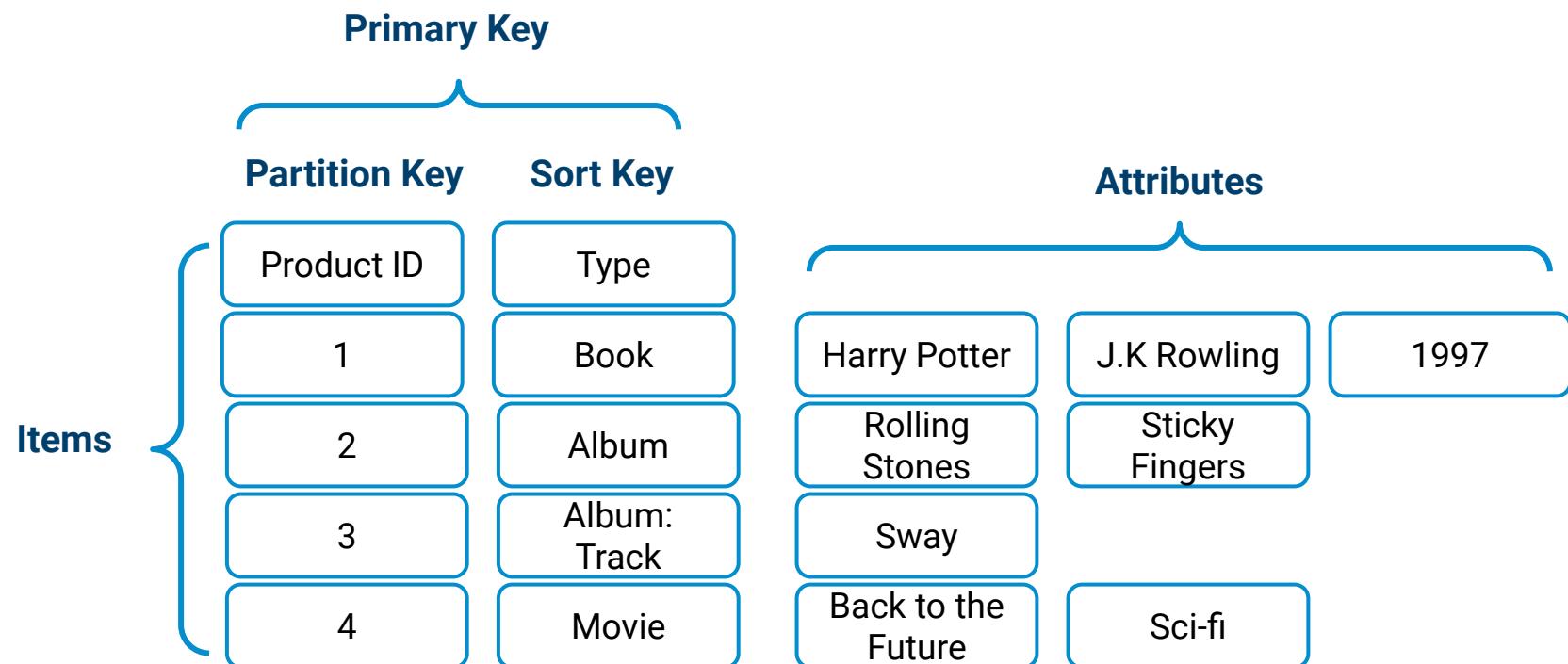


# DynamoDB



- Base de datos de documentos de clave -valor
- Rendimiento de milisegundos de un solo dígito

# Ejemplo de un documento.

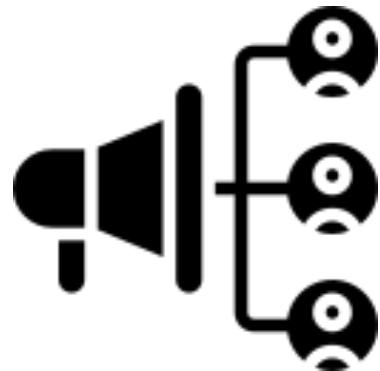


# DynamoDB



- Completamente Administrado.
- Funciona en múltiples regiones.
- Seguridad, respaldo y restauración integrados.
- Admite picos de 20,000,000 de solicitudes por segundo.
- Realmente rentable.

# Casos de uso



---

# Evaluando ElastiCache

# Qué es una base de datos en memoria



# Qué es una base de datos en memoria



# Caso de uso



# Motores



**ElastiCache**  
para Redis



**ElastiCache**  
para  
Memcached

