



# AWS Cloud Practitioner

## Datacamp

≡ Author	Datacamp
⌘ Status	In progress



## ▼ Capítulo 1: Beneficios, Well-Architected Framework and Migration

### 1.1. Beneficios AWS

## 6 key advantages of using the cloud



<b>1. Pay for what you use:</b> <ul style="list-style-type: none"><li>- Consumption based model: start small and pay for what you use</li><li>- Can lead to variable month-to-month costs</li><li>- Allows for opportunities to save</li></ul>	<b>2. Economies of scale:</b> <ul style="list-style-type: none"><li>- AWS has the ability to offer pay-as-you-go costs due to its vast customer base.</li></ul>	<b>3. Capacity:</b> <ul style="list-style-type: none"><li>- Provision and scale resources to meet your exact needs.</li><li>- The ability to resize resources makes AWS elastic.</li></ul>
<b>4. Speed and agility:</b> <ul style="list-style-type: none"><li>- Rapid environment setup</li><li>- Lower cost of failure</li></ul>	<b>5. Save money on datacenters:</b> <ul style="list-style-type: none"><li>- AWS manages infrastructures</li></ul>	<b>6. Go global in minutes:</b> <ul style="list-style-type: none"><li>- Deploy applications to multiple regions quickly</li></ul>

## 1.2. Well-Architected Framework

The Well-Architected Framework provides cloud architects with strategies and guidelines for building resilient, scalable, and secure architectures on AWS

# The six pillars



<b>1. Security</b> <ul style="list-style-type: none"><li>- Protecting data, systems and assets</li><li>- Confidential data through robust identity and access management</li><li>- Controls to detect and mitigate security threats</li></ul>	<b>2. Cost optimization</b> <ul style="list-style-type: none"><li>- Avoid unnecessary expenses</li><li>- Cost-effective decisions to ensure value on your cloud investment</li></ul>	<b>3. Performance Efficiency</b> <ul style="list-style-type: none"><li>- Compute resources used effectively</li><li>- Optimal performance to meet demands</li><li>- Efficiency as workloads evolve</li></ul>
<b>4. Sustainability</b> <ul style="list-style-type: none"><li>- Minimize environmental impact of running cloud</li><li>- Aligning your cloud strategy to your corporate environmental goals</li></ul>	<b>5. Operational Excellence</b> <ul style="list-style-type: none"><li>- Organize teams around business outcomes and implement observability for actionable insights</li><li>- Automate where possible</li><li>- Make frequent, small, reversible changes and refine operations procedures frequently</li><li>- Anticipate failure and learn from all operational events and metrics</li><li>- Use managed services</li></ul>	<b>6. Reliability</b> <ul style="list-style-type: none"><li>- Automatically recover from failure</li><li>- Test recovery procedures</li><li>- Horizontal scaling</li><li>- Stop guessing capacity</li></ul>

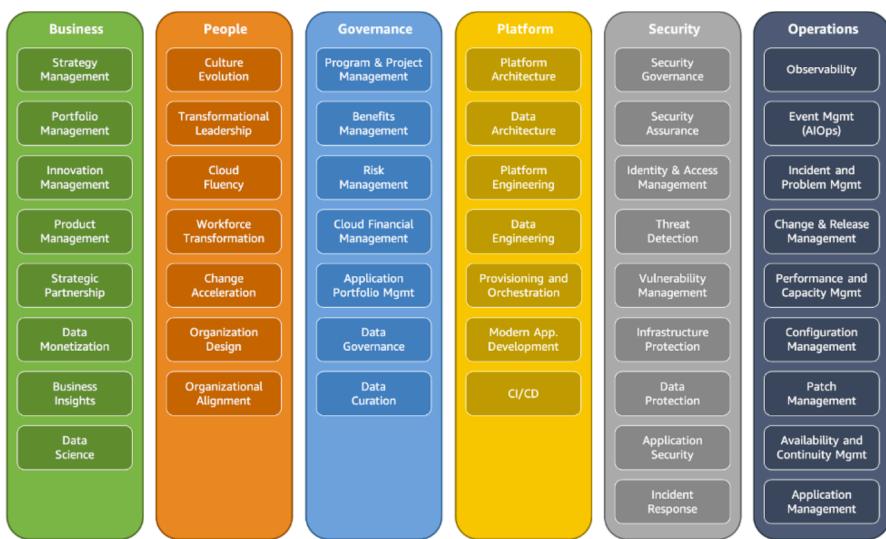
## 1.3. Migrating to the cloud

Moving your organization's data, applications, and workloads to the cloud

### Benefits:

- Significant cost savings
- Increased flexibility
- Improved scalability

## The six perspectives



## 1.4. Benefits of the Cloud Adoption Framework (CAF)

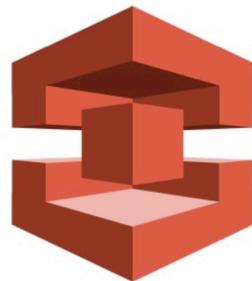
AWS Cloud Adoption Framework (AWS CAF) Structured approach to help organizations design and implement a migration strategy.

1. **Reducing Risk:** ex. backing up data and defining transfer protocols
2. **Improving ESG Performance:** Goal is to implement sustainable cloud solutions.
3. Optimize usage, reduce waste, lower energy consumption
4. **Increasing Revenue and Operational Efficiency:** Automate migration operations leading to greater operational efficiency

## Common migration strategies



AWS Database Migration Service (DMS)



AWS Snowball

- Replicating existing databases with minimal disruption
- Fully operational during transition
- Physical data transport for large-scale migrations
- For moving data quickly and securely

## ▼ Capítulo 2: Cloud economics, Deployments and Global Architecture

### 2.1. Cloud Economics

Refers to the financial considerations of moving to the cloud

Opportunities to significantly reduce IT expenses compared to traditional IT setups

- Flexible pricing models
- Efficient resource usage
- Elimination of fixed costs

#### Fixed vs variable costs

Fixed



Variable



- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>- Hardware</li><li>- Data centers</li><li>- Maintenance</li></ul> | <ul style="list-style-type: none"><li>- Pay for what you use</li><li>- Scale costs with demand</li></ul> |
|---|--|

- Paying for full capacity whether it's used or not
- Also incur ongoing costs

### Bring Your Own Licenses (BYOL)

- Avoid sunk costs
- Port existing licenses to AWS infrastructure
- Cost-effective by leveraging current investment
- Alternative: AWS includes licenses bundled into service



### Right-sizing



- Adjust resources to meet actual demand
- Flexibility: significant cost savings and dynamic environment

### Automation



- Automate the setup/management of resources
- Reduces time, minimizes human error and ensures consistent configuration

### Managed services



- Amazon RDS for databases
- Amazon Lambda for serverless computing
- Amazon S3 for storage

## 2.1. Deploying and operating in AWS

### Provisioning resources in AWS

- Manual vs. automatic resource creation
- CloudFormation
  - Infrastructure as Code (IaC)
  - Define your environment in templates
  - Consistent; reduces manual errors

### Accessing AWS services

- AWS Management Console
  - Web based user interface
- Programmatic Access
  - AWS Command Line Interface (CLI)
  - Software Development Kits (SDKs)
  - APIs
- Infrastructure as Code
  - AWS CloudFormation
  - Third-party solutions like Terraform
  - Automatic deployment

## Cloud deployment models

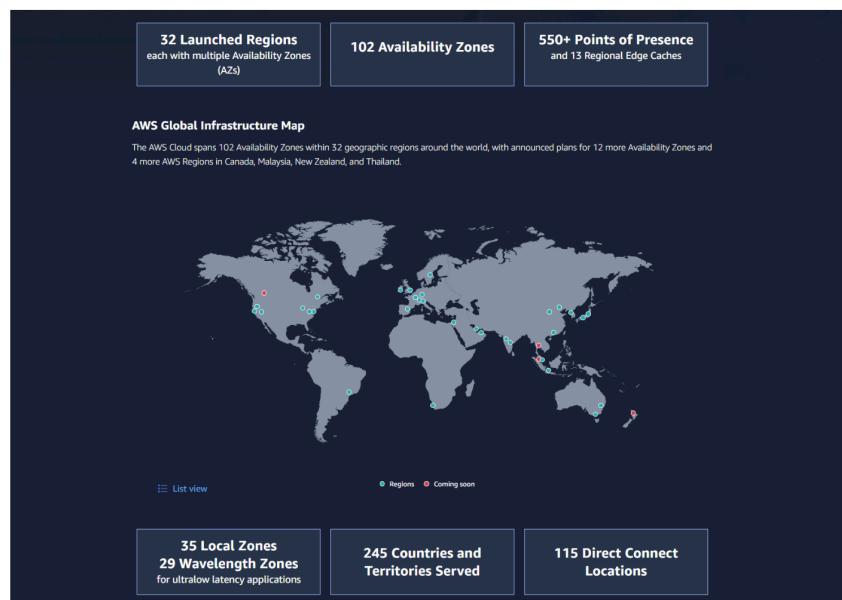
- Public Cloud
  - All resources are hosted on AWS and shared with others
  - Cost efficient and scalable
- Private Cloud
  - Dedicated resources to a single organization
  - Strict compliance or security
- Hybrid Cloud
  - Mix of on-premises and cloud services

## Connectivity options

- AWS VPN
  - Encrypted connection between your network and AWS
  - Flexible and cost-effective
  - Temporary connections
- AWS Direct Connect
  - Private network connection
  - Higher bandwidth and better performance
- Public Internet
  - Resources that are public-facing

## 2.1. AWS Global Architecture

### AWS's geographical diversity



## AWS data centers

- Data centers in each region
- Highly secure, reliable, and high-speed

## Edge locations: speed of light

- Over 400 Edge Locations worldwide
- Stores copies of data
- Faster content delivery with AWS CloudFront



## Availability zones

- Multiple Availability Zones (AZs) per region
- Each AZ is one or more discrete data centers
- Built for redundancy and availability



### Exploring edge locations in Amazon CloudFront

Understanding how to leverage AWS Regions and edge locations is essential for building resilient and high-performing applications. AWS provides various services that allow you to distribute your applications and data across different geographic areas, ensuring high availability and low latency.

In this exercise, you will explore AWS Global Accelerator and Amazon CloudFront to understand how they use multiple Regions and edge locations to achieve these goals. This knowledge is crucial for designing systems that can withstand failures and deliver content quickly to users worldwide.

Instructions 100XP

1 2 3 4

- Locate the *Price Class* header, under *Settings* to explore the different levels of edge location coverage that you can select (no distribution will be setup in this exercise).

Think about scenarios where global coverage might be crucial despite higher costs, versus situations where regional coverage could be sufficient and more cost-effective. Reflect on the implications of choosing "Use All Edge Locations" versus a more limited selection.

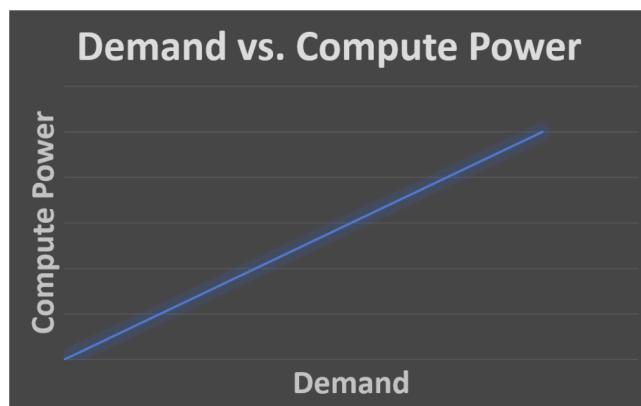
The screenshot shows the AWS CloudFront 'Create Distribution' page. In the 'Settings' section, there is a heading 'Anycast static IP list - optional' with a link to 'Info'. Below it says 'Deliver traffic from a small set of IP addresses' and 'There are no Anycast static IP lists available'. A 'Create an Anycast static IP list' button is present. Under 'Price class', there is a heading 'Price class' with a link to 'Info', followed by the instruction 'Choose the price class associated with the maximum price that you want to pay.' Three radio buttons are shown: 'Use all edge locations (best performance)' (selected), 'Use only North America and Europe', and 'Use North America, Europe, Asia, Middle East, and Africa'. The 'Alternate domain name (CNAME) - optional' section is also visible.

## ▼ Capítulo 3: Compute, Database services

### 3.1. Compute services

#### Compute: the backbone of digital solutions

- Definition: providing computing power on-demand
- Importance: scalability, flexibility, and cost-efficiency



# Meeting the challenge with AWS

## Server Based

- Continuous availability
- Dedicated resources
- More control
- Customization (like owning a car)



## Serverless

- On-demand execution
- No server management
- Event-driven
- Cost-effective
- It's like using a taxi service; it's there when you need it and gone when you don't



## In real life

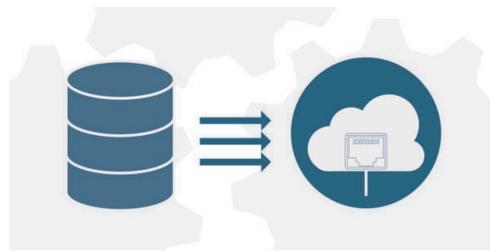
### EC2

- Hosting websites
- Scalability and customization



### Lambda

- Real-time image processing
- Event-driven tasks



## EC2 unpacked



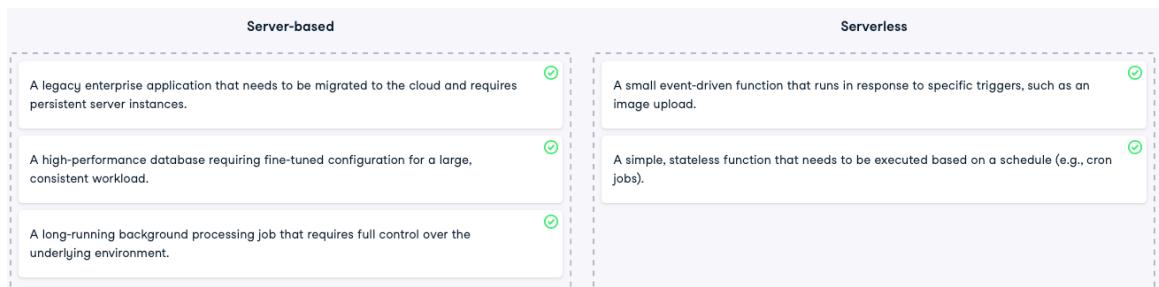
Amazon EC2

- Virtual servers in the cloud
- Customizable configurations (OS, storage, location)
- Focus on customization

## Lambda unpacked



- Serverless computing platform
- Name comes from Lambda calculus
- Event-driven architecture (file uploads, database changes)
- Focus on convenience



#### Exploring serverless compute options

In this exercise, you will explore AWS Lambda, a key serverless compute service, by creating and testing a simple function. You'll understand how Lambda handles scaling, automation, and server management without needing direct intervention.

Instructions 100XP

1 2 3 4 5

There are three ways to author functions: from scratch, with a blueprint and using a container image.

- Select *Use a blueprint* for the method we are going to use for authoring.
- Update the *Blueprint name* to the Hello world function - python 3.10 blueprint
- Define the *Function name*, for example "hello-world-python"
- The execution role should be set to *Create a new role with basic Lambda permissions*
- Click *Create function* to create your Lambda function!

#### Exploring serverless compute options

In this exercise, you will explore AWS Lambda, a key serverless compute service, by creating and testing a simple function. You'll understand how Lambda handles scaling, automation, and server management without needing direct intervention.

Instructions 100XP

1 2 3 4 5

There are three ways to author functions: from scratch, with a blueprint and using a container image.

- Select *Use a blueprint* for the method we are going to use for authoring.
- Update the *Blueprint name* to the Hello world function - python 3.10 blueprint
- Define the *Function name*, for example "hello-world-python"
- The execution role should be set to *Create a new role with basic Lambda permissions*
- Click *Create function* to create your Lambda function!

This function contains external libraries.

```

1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event['key1'])
9     print("value2 = " + event['key2'])
10    print("value3 = " + event['key3'])
11    print("value4 = " + event['key4'])
12    raise Exception('Something went wrong')
13

```

#### Exploring serverless compute options

In this exercise, you will explore AWS Lambda, a key serverless compute service, by creating and testing a simple function. You'll understand how Lambda handles scaling, automation, and server management without needing direct intervention.

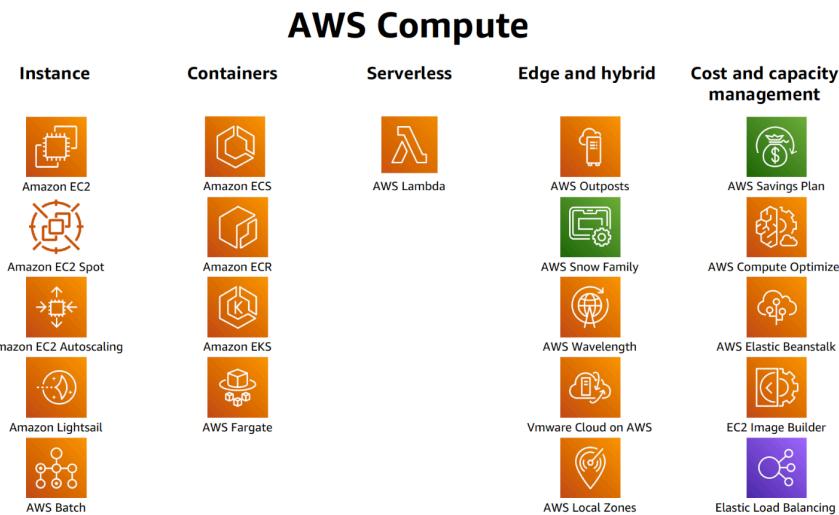
Instructions 100XP

1 2 3 4 5

Time to deploy your function to the Lambda environment using the *Deploy* button.

Typically the next step would be to test your function works as expected, but the purpose of this exercise was to explore the setup. We'll dive deeper into Lambda in future courses.

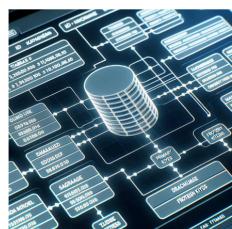
# Beyond EC2 and Lambda



## 3.2. Database services

### Understanding database types

- Relational Databases (RDS)
  - Like a well-organized bookshelf
  - Use MySQL, PostgreSQL, Oracle, etc.
  - Ideal for traditional applications
  - **AWS RDS: the sturdy bookshelf of the digital world**
- NoSQL Databases (DynamoDB)
  - Like a dynamic magazine rack
  - Flexible schema for unstructured data
  - Ideal for mobile apps, IoT, gaming
  - **AWS DynamoDB: adaptable and ready for ever-changing content**



### Diving into RDS

- Scalable and cost-effective
- Supports multiple database engines like MySQL, PostgreSQL, etc

### DynamoDB unpacked

- Designed for web-scale applications
- Provides single-digit millisecond latency
- DynamoDB uses a key-value model
- A key maps to a value

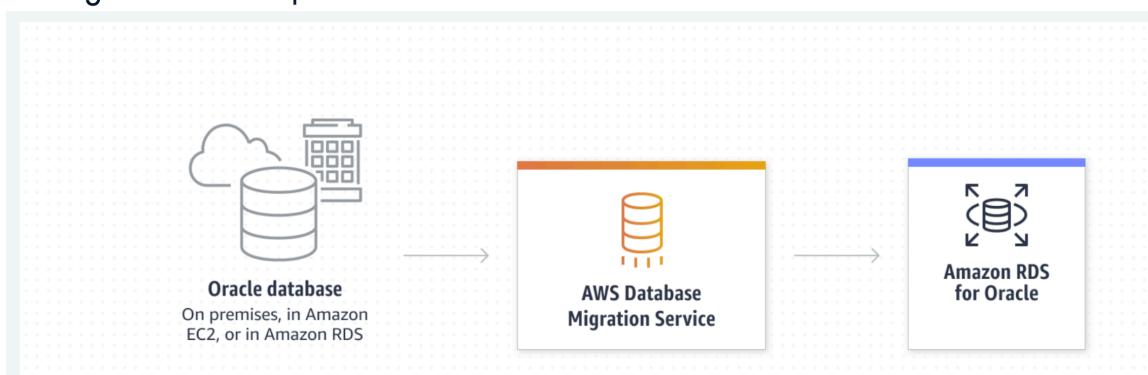
Relational Databases	NoSQL Databases	Memory-Based Databases
Customer orders: Structured data requiring high availability	Product catalog: Unstructured or semi-structured data that can scale easily	Leaderboard data for a gaming app: Data that needs to be frequently updated and accessed in real-time
E-commerce transactions: High volume of structured data requiring complex queries	Chat application messages: Real-time data requiring quick access and updates	User session data: Data that needs to be accessed and updated frequently with low latency

## Beyond RDS and DynamoDB



## AWS database migration services

Moving data from on-premises databases to AWS



### 3.3. Storage services

# Storage vs. Databases

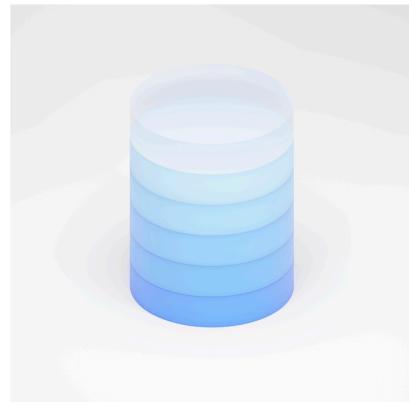
## Storage

- Keep data safe and accessible
- Backups, large files, documents
- Disaster recovery and archiving



## Databases

- Organizing and querying structured data



## Storage services

### AWS storage services

Object, file, and block storage



**Amazon Simple Storage Service (S3)**

Object storage with industry-leading scalability, availability, and security for you to store and retrieve any amount of data from anywhere.



**Amazon Elastic File System (EFS)**

A simple, serverless, elastic, set-and-forget file system for you to share file data without managing storage.



**FSX      Amazon FSx**

Fully managed, cost-effective file storage offering the capabilities and performance of popular commercial and open-source file systems.



**Amazon Elastic Block Store (EBS)**

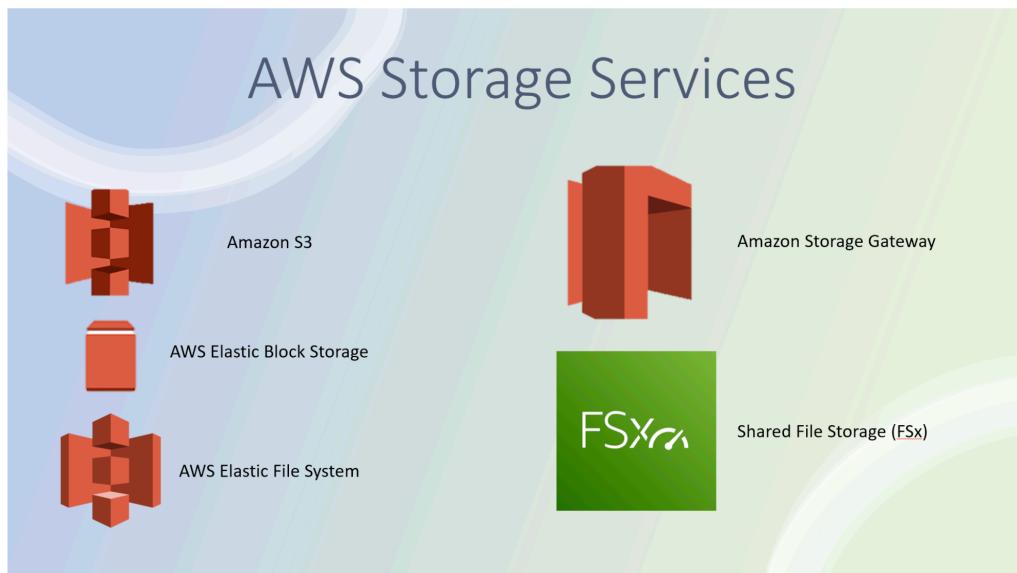
Easy to use, high-performance block storage service for both throughput and transaction-intensive workloads at any scale.



**Amazon File Cache**

High-speed cache for datasets stored anywhere, accelerate cloud bursting workloads.

# Other storage services



## Understanding storage types

### Active Storage (Direct Storage)

- Like your **recent emails**, readily accessible
- Ideal for **day-to-day operations**
- **AWS S3:** designed for ease of access and management
  - Object storage service
  - Used for storing and retrieving any amount of data, anytime, from anywhere
  - Can get pricey



### Archival Storage

- Like **old emails**, accessed infrequently
- Ideal for **long-term data retention**
- **AWS Glacier:** cost-effective for long-term storage
  - Used for data archiving and long-term backup
  - Long-term, low-cost, and secure cloud storage service



## Diving into S3

- S3 stands for Simple Storage Service
- Highly scalable, durable, and secure
- Wide variety of use cases like website hosting, data backup, and content distribution



# Storage classes

## S3 Standard:

- Frequently accessed data
- Low latency, high throughput
- Content distribution and dynamic websites

## S3 Intelligent-Tiering:

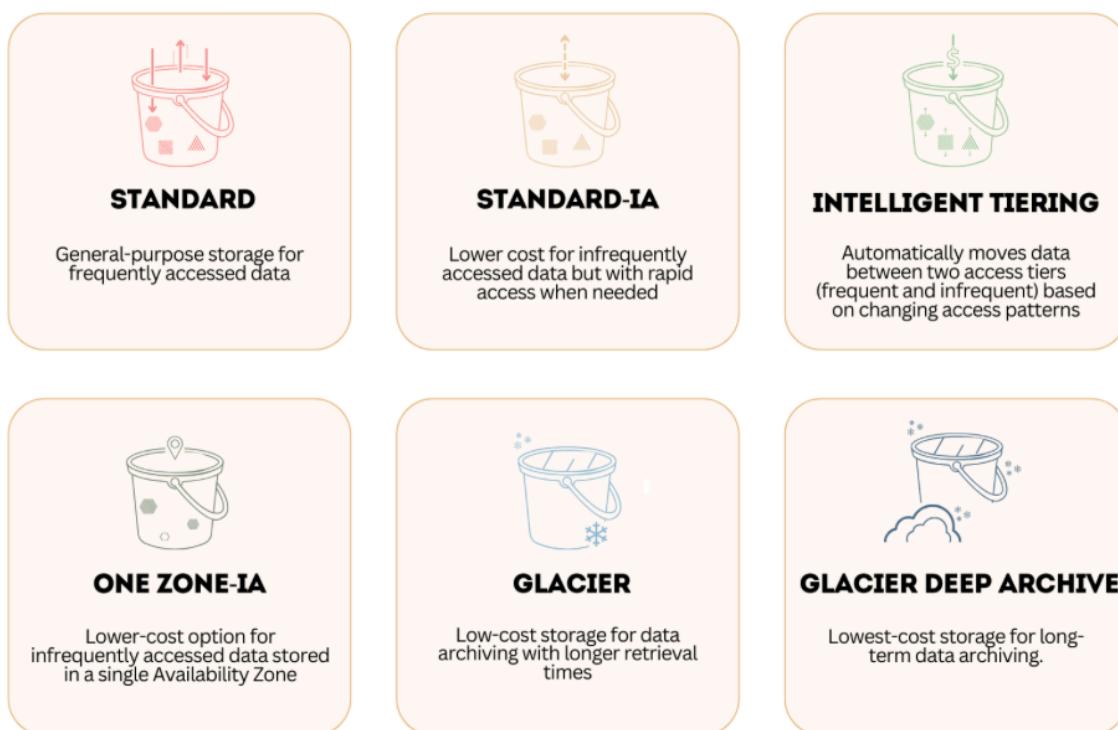
- Moves data between frequent and infrequent access tiers
- Optimizes storage costs

## S3 Standard-IA (Infrequent Access):

- Less frequently accessed data, but still needs rapid access
- Backups and disaster recovery

## S3 Glacier and S3 Glacier Deep Archive:

- Long-term archival of rarely accessed data



### Creating an S3 bucket

In this exercise, you'll explore Amazon's object storage service: S3. You'll learn how to create, manage, and upload files to an S3 bucket and how to configure important settings like versioning and access permissions, as well exploring S3 storage classes.

Instructions 100XP

1 2 3 4 5 6

Which storage class has been applied to the sample image we uploaded to the S3 bucket?

Standard-IA  
 Intelligent-Tiering  
 Standard  
 Glacier

Hint Submit

Creating an S3 bucket us-east-1.console.aws.amazon.com/s3/object/s3bucket-minebucketdg?region=us-east-1&bucketType=general&prefix=datacamp-logo.png&tab=details

Expiration date The object will be made noncurrent and generate a delete mark

Storage class Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

Storage class Standard

Server-side encryption settings Info

Server-side encryption protects data at rest.

Encryption type Info

Server-side encryption with Amazon S3 managed keys (SSE-S3)

## ▼ Capítulo 4: AWS Cloud Technology and Services

### 4.1. EC2: Elastic Compute Cloud

- Provides resizable compute capacity in the cloud
- Each individual EC2 machine is referred to as an instance

#### Key characteristics:

- Ability to scale up or down based on demand
- Availability of a varied range of instance types for specialized use cases

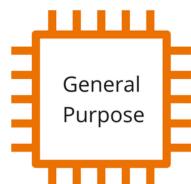
### 4.2. EC2 instance types

AWS offers six categories of EC2 instances for specialized workloads

- General purpose
- Compute optimized
- Memory optimized
- Storage optimized
- Accelerated computing
- High Performance Computing (HPC) optimized

#### General purpose instances

- Balance of compute, memory, and networking resources
- Use cases:
  - Hosting dynamic websites
  - Maintaining code repositories



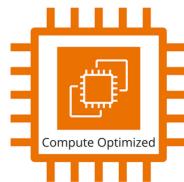
#### Storage optimized instances

- High, sequential read and write access to large datasets
- Use cases:
  - Data warehousing
  - Refactoring large relational databases



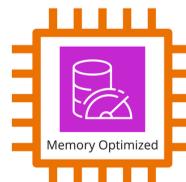
#### Compute optimized instances

- Compute-intensive and high-performance workloads
- Use cases:
  - Scientific simulations
  - Financial modeling



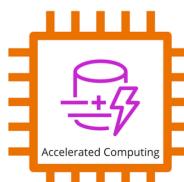
#### Memory optimized instances

- Memory-intensive workloads not requiring high storage
- Use cases:
  - Real-time stream data analytics
  - Generating close captions



#### Accelerated computing instances

- Contain specialized hardware accelerators, like GPUs or FPGAs
- Use cases:
  - Deep learning
  - Rendering gaming graphics

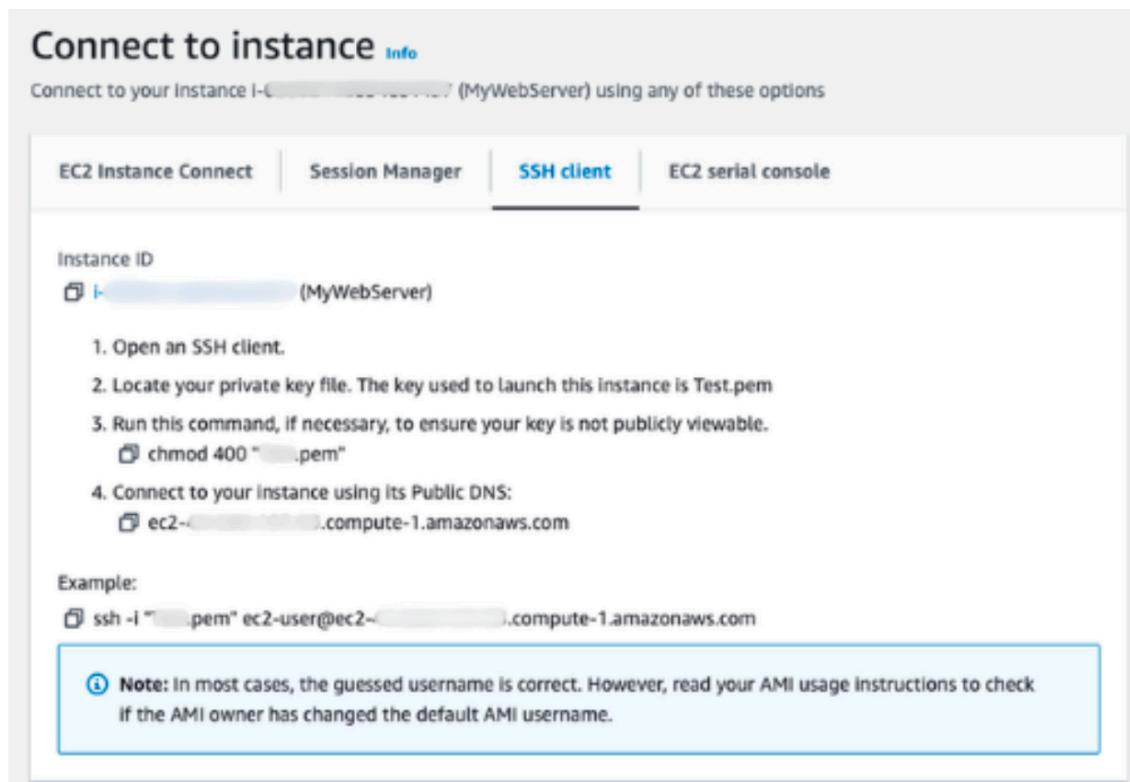


#### HPC optimized instances

- Best price performance for running high performance workloads at scale
- Use cases:
  - Weather forecasting
  - Crash simulations



### 4.3. Connecting to EC2 instance:



a. **SSH client:**

SSH connects with a private key

Those keys must be managed

b. **AWS session manager:**

Keyless access via the Management Console

Integrates with Identity and Access Management (IAM)

c. **EC2 instance connect:**

Browser-based connection

Quick and temporary access

## 4.4. Load balancing in AWS:

Load balancing ensures even distribution of incoming traffic among multiple EC2 instances,

preventing overload on a single server

- Ensures high availability
- Provides horizontal scaling



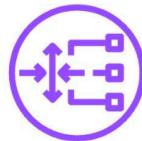
Classic Load Balancer



Network Load Balancer



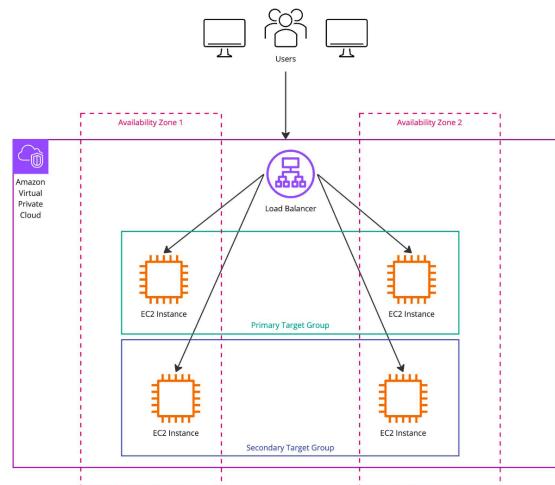
Application Load Balancer



Gateway Load Balancer

## How does load balancing work?

1. Users send requests
2. Requests hit the load balancer
3. Primary target group is instantiated by the application load balancer first
4. If demand increases, the load balancer activates the secondary target group and distributes the load across all instances



## 4.5. Compute elasticity:

Elasticity ensures your system can scale up or down based on demand, providing flexibility in resource allocation

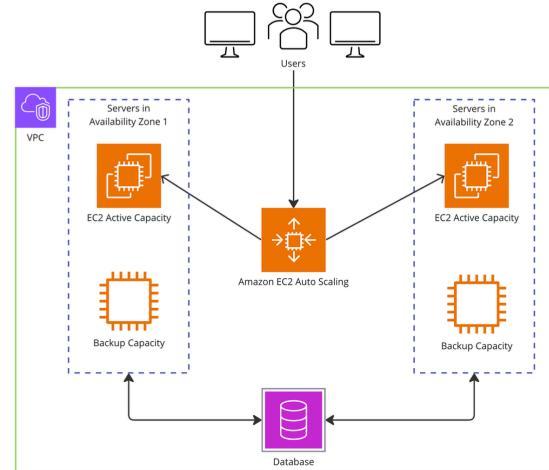
- EC2 instances achieve elasticity through EC2 Auto Scaling

### What is EC2 Auto Scaling?

- Automatically adjust the number of active instances based on usage and requirement
- Optimize costs
- Prevent over-provisioning

## How does auto-scaling work?

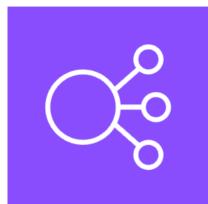
1. Users send requests
2. The requests are routed to EC2 Auto Scaling service
3. The service then routes requests to the active EC2 instances
4. If demand increases, it starts adding new EC2 instances to manage the additional load
5. As demand goes down, the newly added EC2 instances are shut down



## Load balancing vs. auto-scaling

### Load balancing

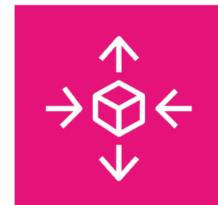
- Route traffic evenly
- Utilize existing EC2 instances



AWS Load Balancer

### Auto-scaling

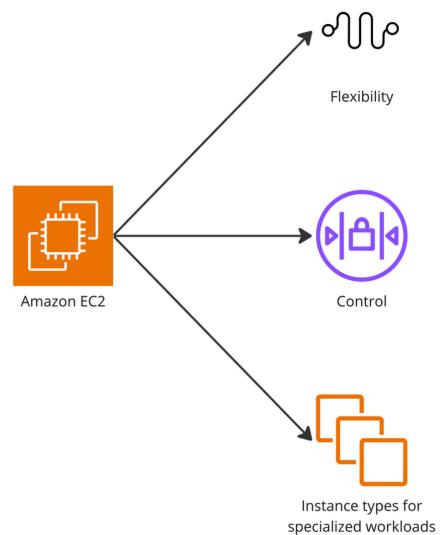
- Ensure demand is always met
- Ability to add/remove EC2 instances



EC2 Auto Scaling

### EC2 Recap:

- Amazon EC2 is a service that provides compute capacity in the AWS cloud
- Using EC2 gives higher flexibility and control
- Variety of EC2 instance types optimized for different workloads



## 4.6. Containers

Containers encapsulate applications and their dependencies, in lightweight singular units.

- Persistent environments
- Predictable workloads
- Resource-intensive applications

### Why containers?

- Isolate applications from underlying system dependencies
- Share host OS for efficient resource utilization
- Easily movable and portable across environments



Amazon Elastic Container  
Service



Amazon Elastic Kubernetes  
Service

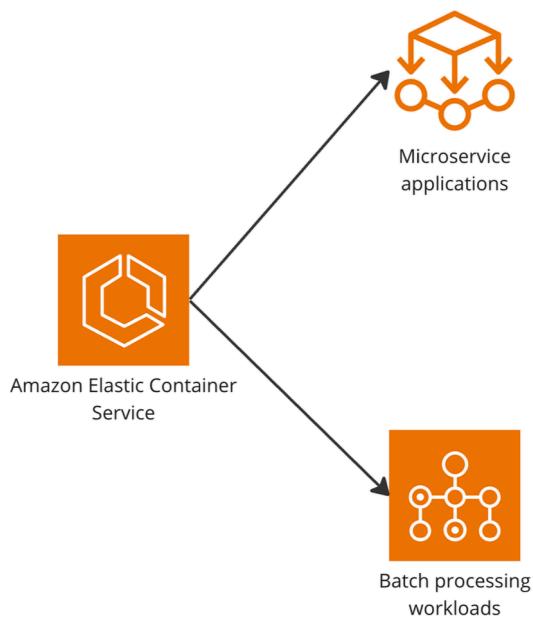
- Easily scale containerized applications up or down
- Integrate with other AWS services

#### a. **Amazon ECS (Elastic Container Service):**

Fully managed service for efficient deployment, management, and scaling of containerized applications

### Use cases

- Deploying and managing microservices-based applications
- Plan, schedule, and run batch processing workloads across AWS services

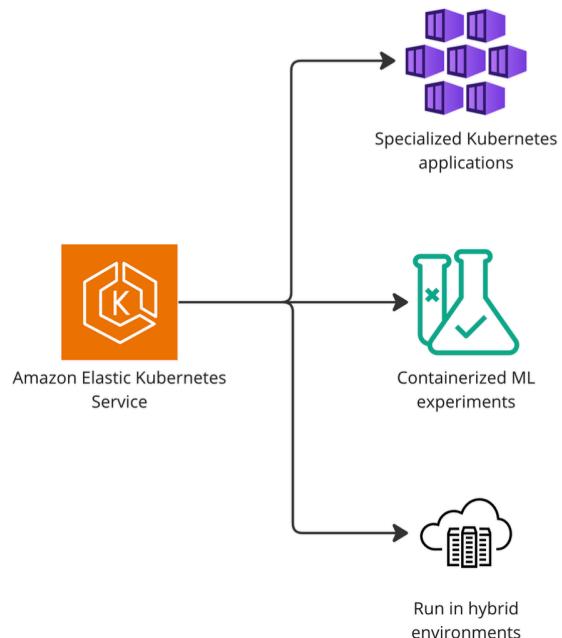


### b. Amazon EKS (Elastic Kubernetes Service):

Container orchestration service specializing in running Kubernetes-powered applications

### Use cases

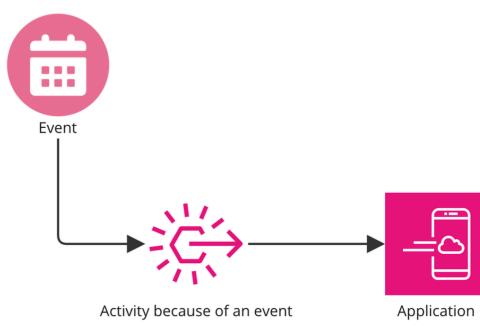
- Pair with EC2 accelerated computing instances to run ML containers
- Manage clusters and applications in hybrid cloud environments



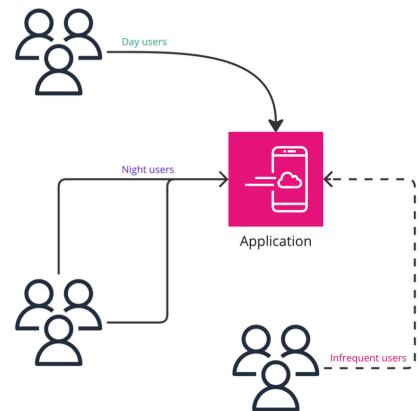
## 4.7. Serverless

## More forms of compute?

Event-based compute changes



Balancing compute for sporadic traffic loads



- No server management: forget about provisioning, scaling, or maintenance
- Event-driven: functions triggered by events in real-time
- Cost-efficient: pay only for actual usage, not pre-allocated resources



AWS Lambda

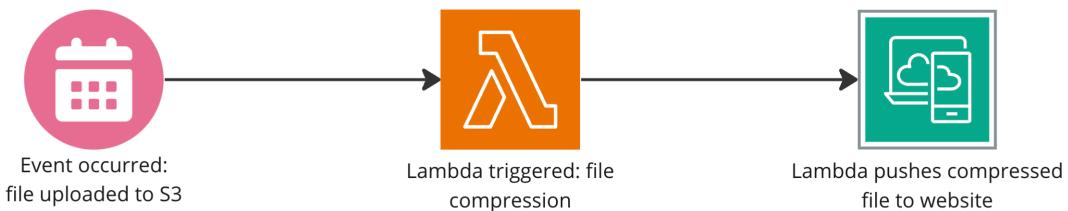


AWS Fargate

### a. AWS Lambda:

## AWS Lambda

- Run code in response to events without provisioning or managing servers
- Automated compute scaling capabilities

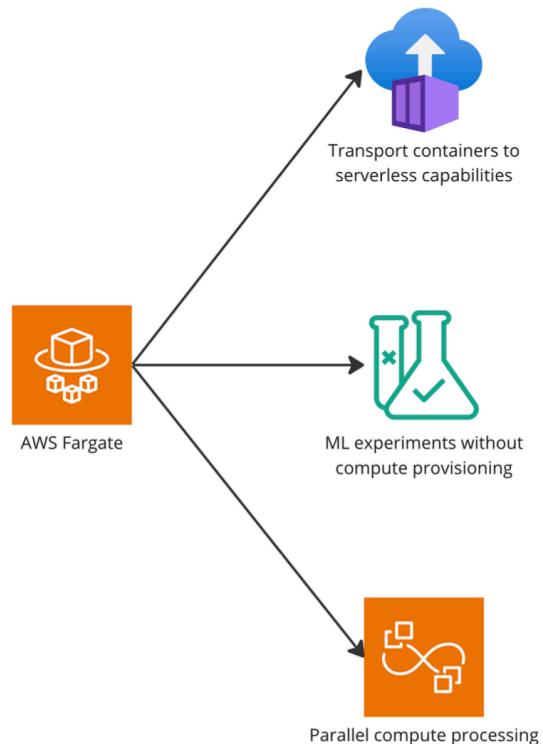


### a. AWS Fargate:

Streamlines application development by providing serverless compute for containers

#### Use cases

- Enable AI and ML applications without the need for excessive server provisioning
- Batch processing of large datasets with parallel compute capabilities



Amazon ECS	Amazon EKS
Works best with microservices-based applications	Ideal for compute-intensive tasks such as machine learning
Integrates seamlessly with other AWS services such as Fargate	Provides native Kubernetes support

## ▼ Capítulo 5: Exploring AWS Database Resources

## 5.1. DBaaS: Database as a service:

Access databases without configuring physical infrastructure or installing software



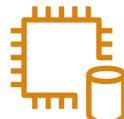
Relational databases



NoSQL databases



Memory-based databases



Compute-hosted databases

[https://miro.com/app/board/uXjVIFZscHM=/?share\\_link\\_id=782242498622](https://miro.com/app/board/uXjVIFZscHM=/?share_link_id=782242498622)

### a. Relational Databases:

Structured storage systems  
organizing data into tables  
with predefined relationships.

#### Why use relations databases?

- Data integrity
- Consistency
- Scalability



Amazon RDS



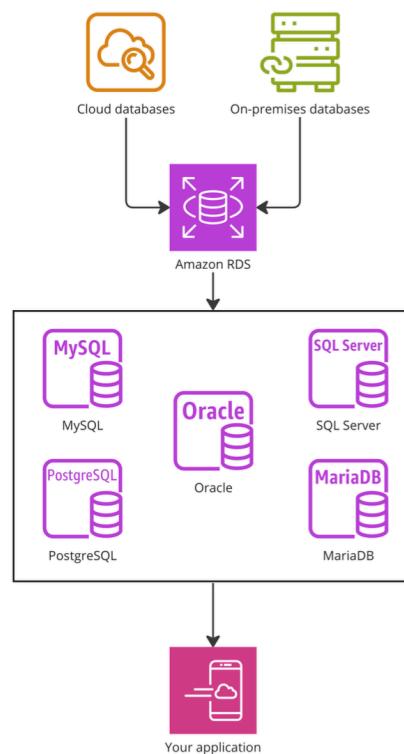
Amazon Aurora

#### 1. Amazon RDS:

Fully managed relational database service.

### Key features

- Wide range of supported database engines
- Efficient scaling for cloud and on-premises databases

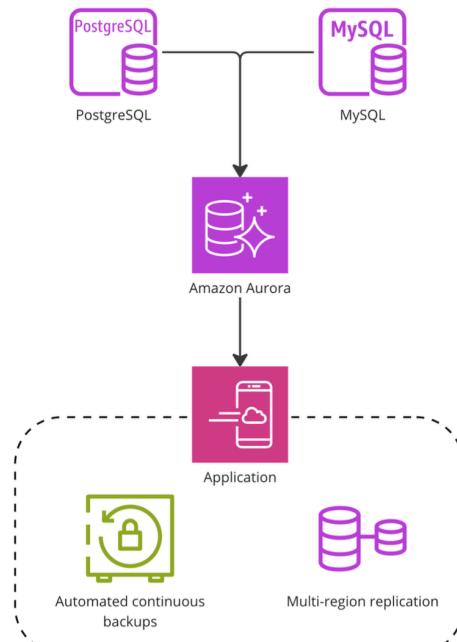


## 2. AWS Aurora:

Relational database service optimized for MySQL and PostgreSQL engines

### Key features

- High performance at approximately one-tenth of the cost
- In-built continuous backup
- Automated multi-region replication



## c. No Relational Databases:

Accommodate diverse data models beyond traditional relational databases, such as JSON and raw documents



Amazon DynamoDB

### Key features

- Schema flexibility
- Horizontal scalability



Amazon DocumentDB

#### 1. **DynamoDB:**

Serverless, NoSQL, fully managed database with single-digit millisecond performance at any scale

##### Key features

- High performance with nearly unlimited throughput and storage
- 99.999% global availability
- Serverless capabilities for seamless scaling

##### Use cases:

- Real-time video streaming and interactive content in media/entertainment
- Tracking inventory or shopping carts based on customer profiles
- Game platform with player data, session history, and leaderboards

#### 2. **DocumentDB:**

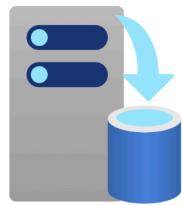
- Fully managed native JSON document database with MongoDB compatibility
- Operate critical document workloads of any scale



MongoDB compatibility



Heavy document workloads



No infrastructure management

### Use cases:

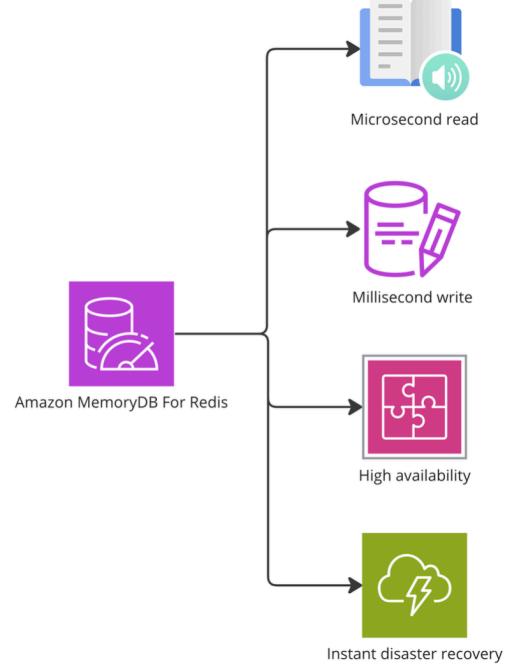
- Fast, reliable access to content in your CMS like reviews and images
- Generate customer recommendations and manage millions of user profiles
- Unlock GenAI use cases such as semantic search, product recommendations, and chatbots

### c. Memory-based Databases:

Designed for high-performance data storage and retrieval, utilizing RAM for faster access

#### Optimal use cases

- Caching frequently accessed data
- Real-time analytics and data processing
- High-speed transactional applications



#### 1. MemoryDB for Redis

- Super-fast read/write capabilities via RAM
- 99.99% availability

- Instant recovery without any data loss

#### **d. EC2-hosted Databases:**

Deployed on EC2 instances following a traditional, manual approach to database hosting

- Provides granular control over configuration and management
- Responsibility for backups and patching lies with the user

## **Compute databases vs. static databases?**

EC2-hosted databases



Manual setup

AWS-managed databases



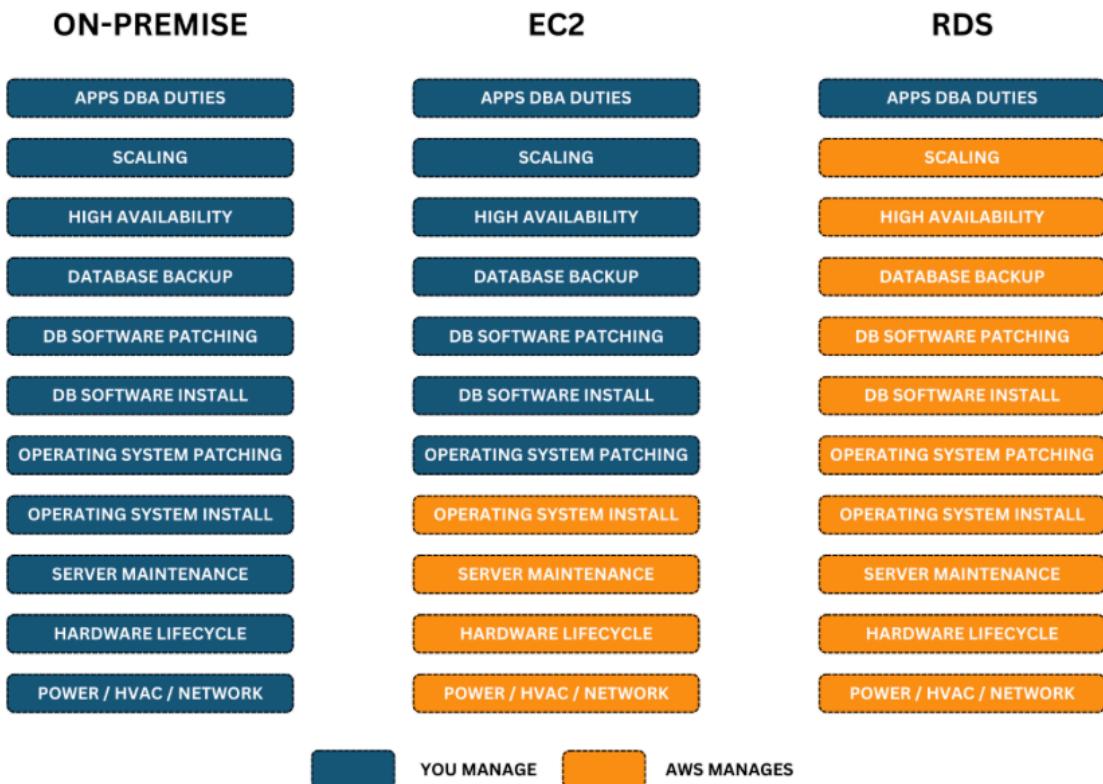
Automated deployment



Full control over configurations



Managed maintenance and administration



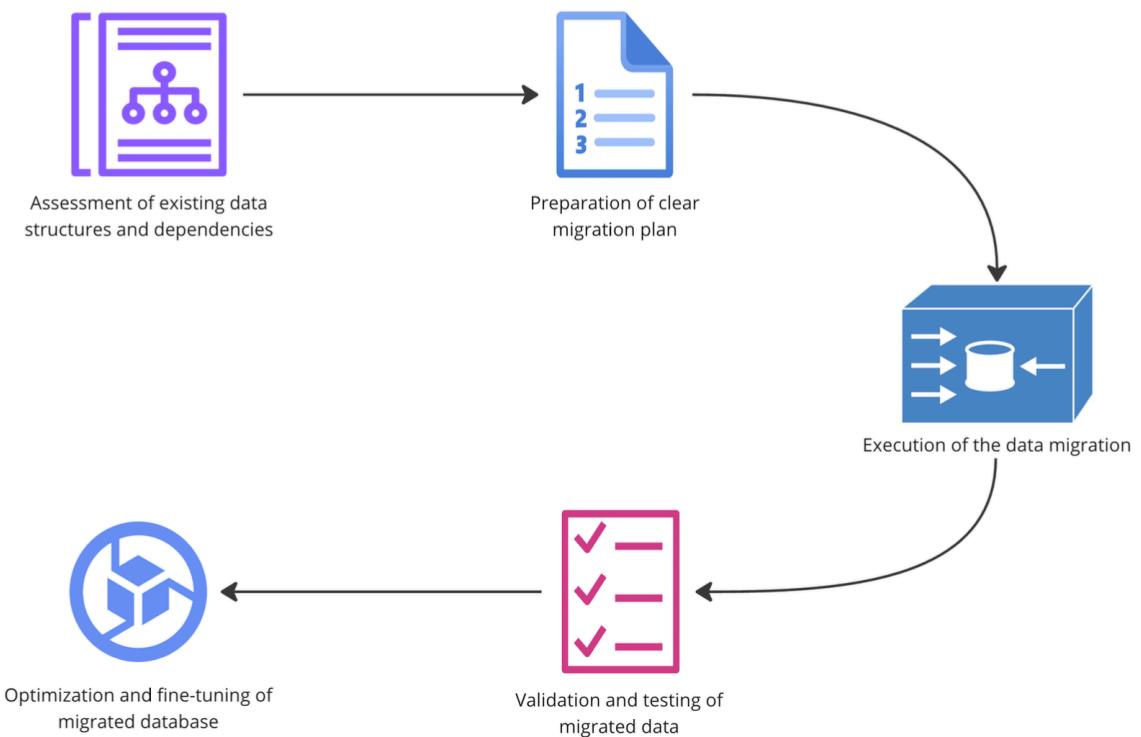
## ▼ Capítulo 6: Database Migration

On-premise to cloud.

[https://miro.com/app/board/uXjVIFYvMhY=?share\\_link\\_id=650738049085](https://miro.com/app/board/uXjVIFYvMhY=?share_link_id=650738049085)

### Why?

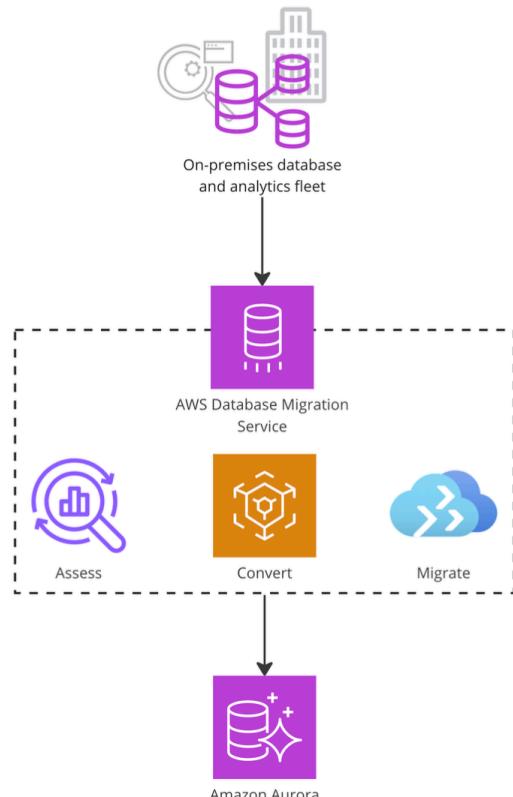
- Legacy systems unable to meet rapid scalability and efficiency demands
- Generative AI has advanced systemic needs for compute and storage
- Adapt to evolving business needs for scalability and efficiency
- Modernize from legacy systems to meet the demands of contemporary applications
- Facilitate unified data management for a diverse sources



## 1. Database Migration Service:

Managed service facilitating the movement of 20+ database and analytics engines to AWS

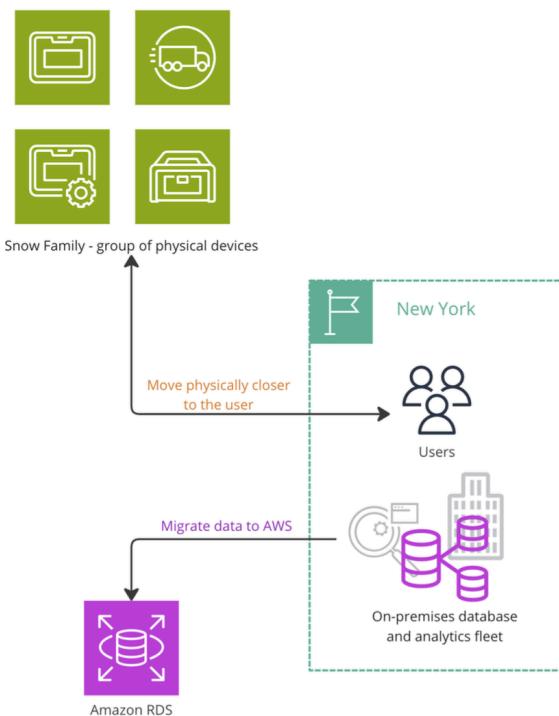
- Minimizes downtime with automated real-time data replication
- Supports diverse source and target databases
- Provides validation checks and task monitoring



## 2. Snow Family:

Offers physical devices to facilitate offline data transfer of petabyte-scale to and from AWS

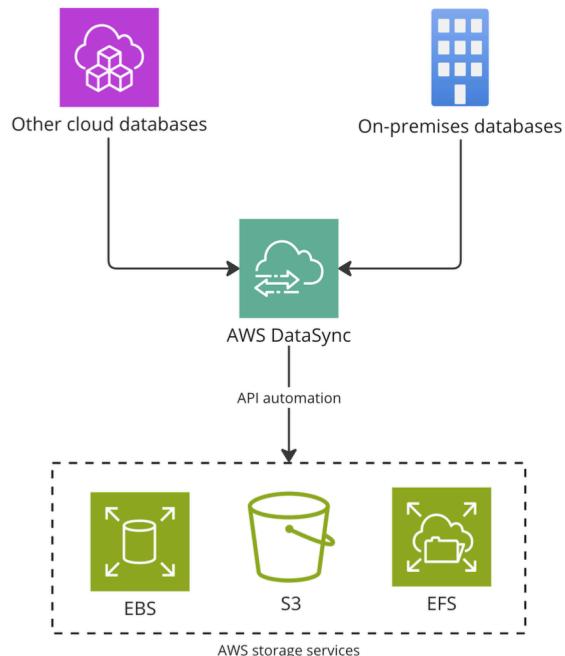
- Enables processing data at the edge for fast large-scale data movement
- Robust security measures to ensure data integrity



### 3. DataSync:

Simplifies, automates, and accelerates large data transfers between on-premises storage and AWS Cloud

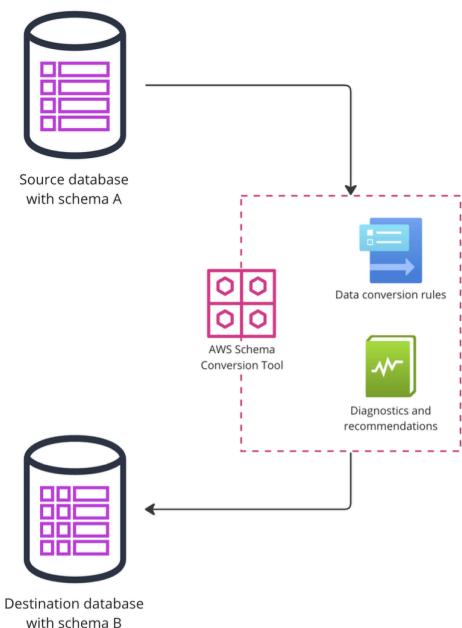
- Fast and efficient data transfer through parallel processing
- Seamless integration with other AWS storage services
- Automation through AWS Management Console or APIs



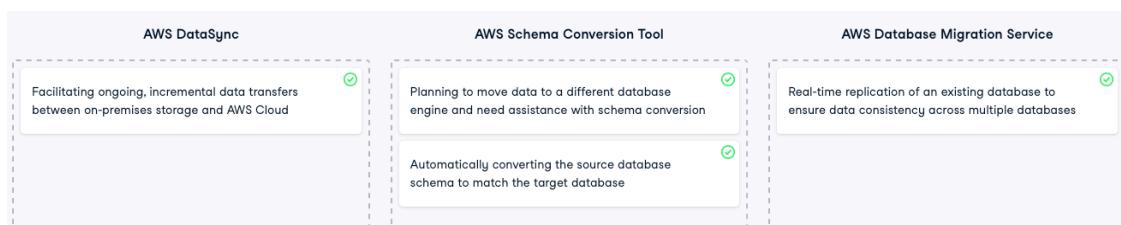
### 4. Schema Conversion Tool:

Facilitates database migrations by automatically converting source database schema to match the target

- Automates database schema conversion
- Customization of conversion rules
- Diagnostics and recommendations

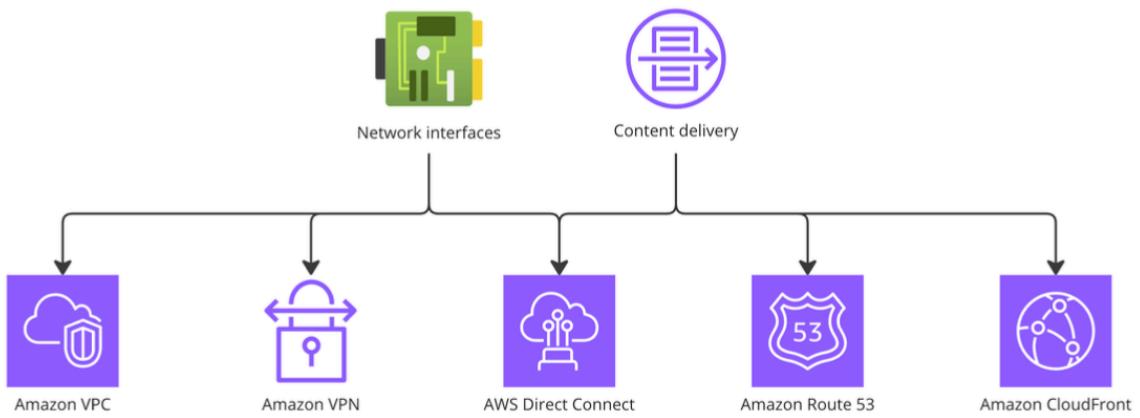


## When to use which service?



## ▼ Capítulo 7: Network services

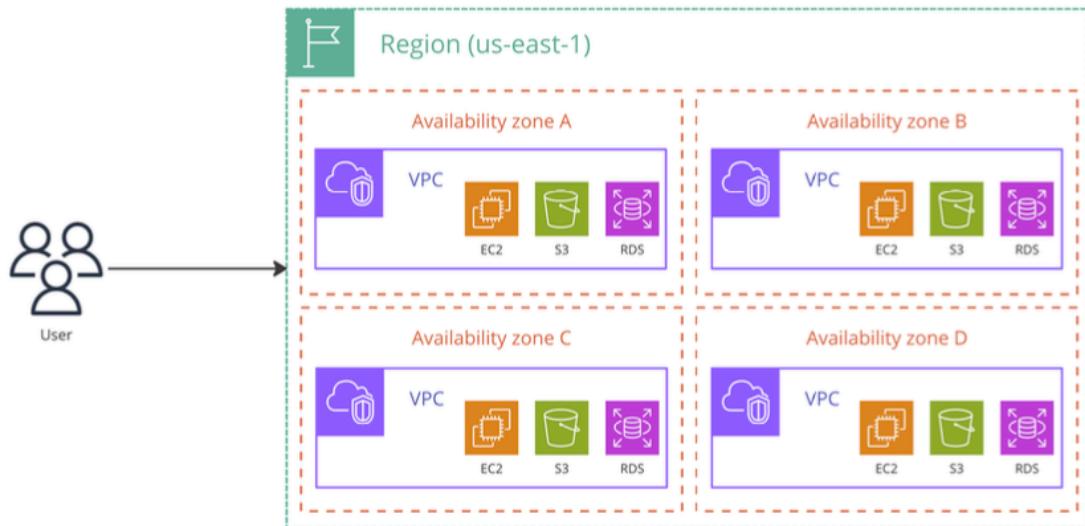
- Building secure and efficient network environments in the cloud
- Designing robust content delivery mechanisms



[https://miro.com/app/board/uXjVIDnQNCE=/?share\\_link\\_id=593697432928](https://miro.com/app/board/uXjVIDnQNCE=/?share_link_id=593697432928)

### 1. Virtual private cloud (VPC):

Logically isolated section of the AWS cloud to launch and manage your own resources



- IPv4 and IPv6 support with **custom configurable IP address range**: Your virtual address space in the cloud



User

10.1.8.190  
·  
·  
·  
10.1.8.200

Custom IP range



Internet

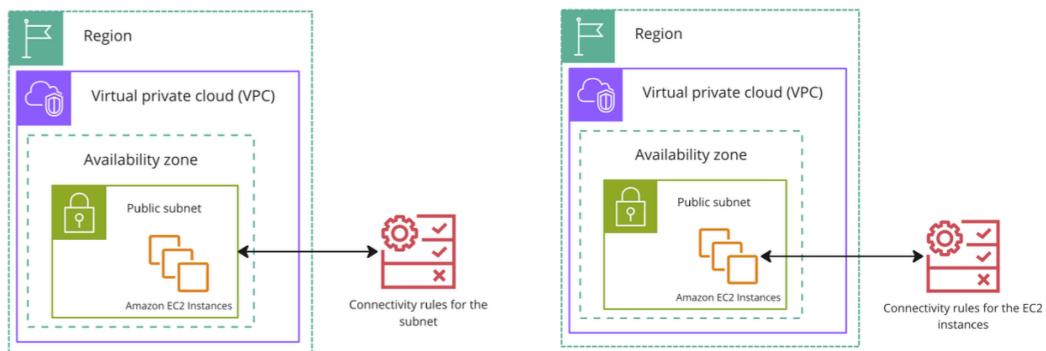
- **Security Layers:** security groups and network access control lists (ACLs):

### Network Access Control Lists (ACL)

Firewall for your subnet

### Network Security Groups (NSGs)

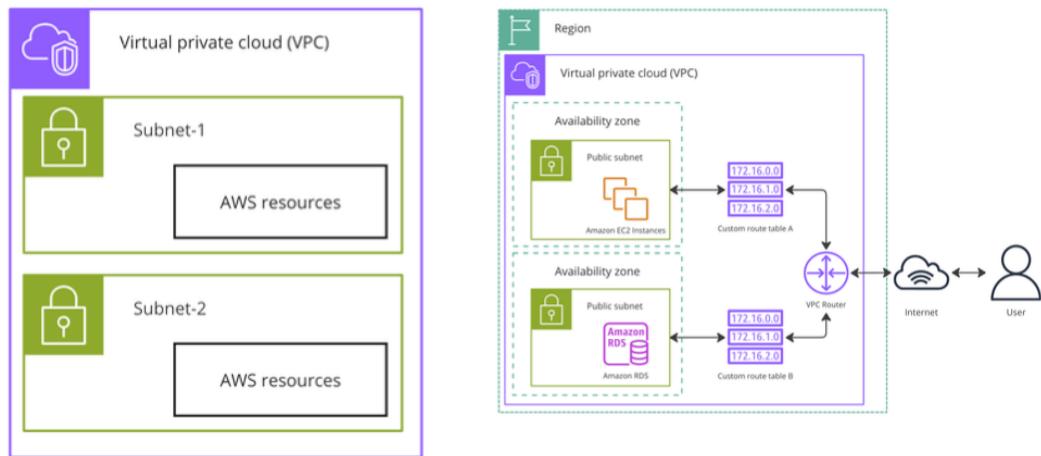
Firewall for your instances



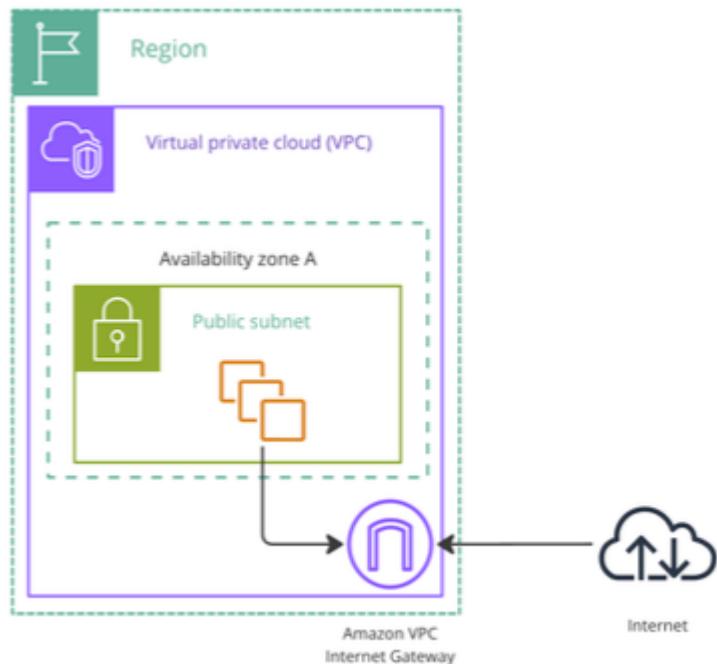
- **Complete control:** subnets, route tables and network gateways:

**Subnets:** Dividing your VPC IP address range into smaller, manageable segments

**Route tables:** Determines where network traffic from your subnet or gateway is directed



- **Network Gateway:** Connects your VPC to the internet or other VPCs and controls inbound and outbound traffic



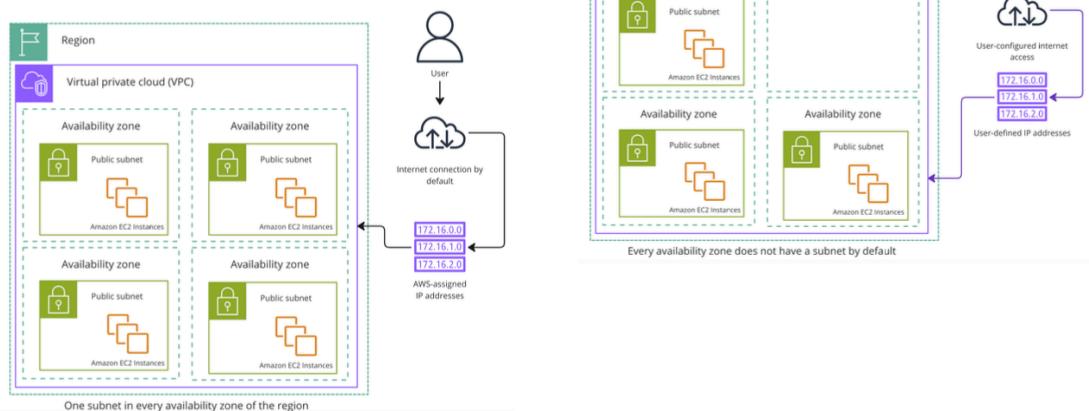
#### Default Amazon VPC:

- Automatic creation with AWS-assigned IP addresses
- Pre-configured settings with a subnet in every availability zone

#### Custom Amazon VPC

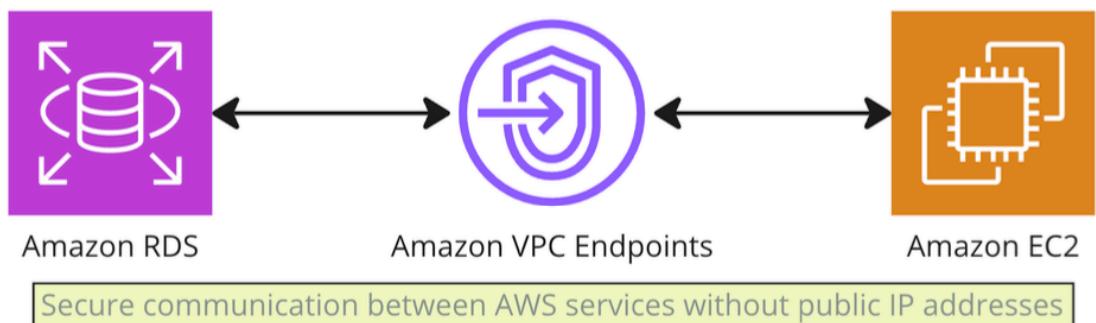
- User-defined
- Customize IP address range, subnets, and -route tables
- Require explicit configuration for internet access

- Communicate with the internet by default



### VPC Endpoints:

- Enable private connections between AWS services
- Enhances security by allowing communication between services without public IP addresses

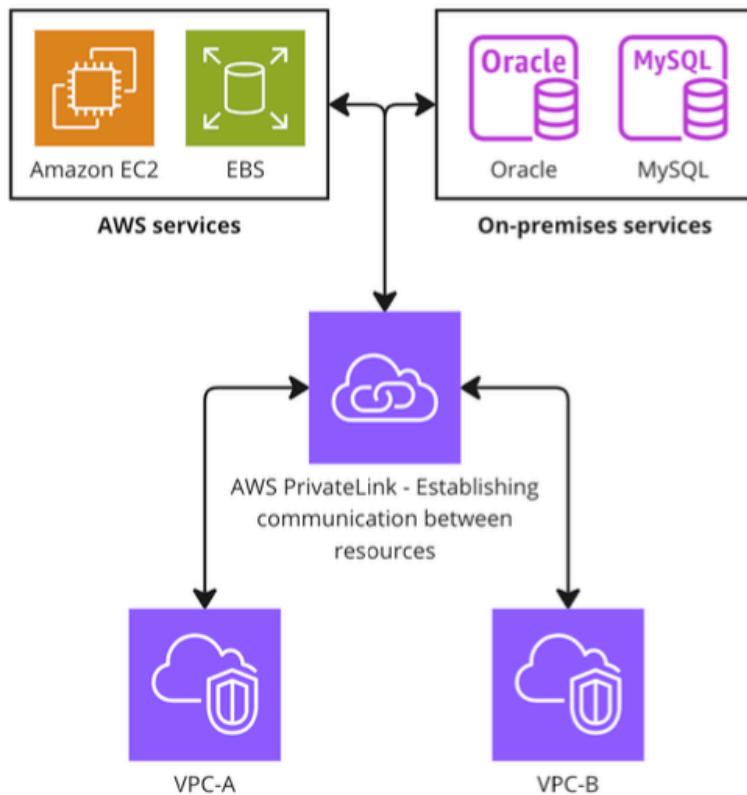


### AWS PrivateLink:

Private connectivity between VPCs, supported AWS services, and on-premises networks

### Benefits

- Secure data exchange with SaaS applications
- Simplified network management



## VPC (Virtual Private Cloud)

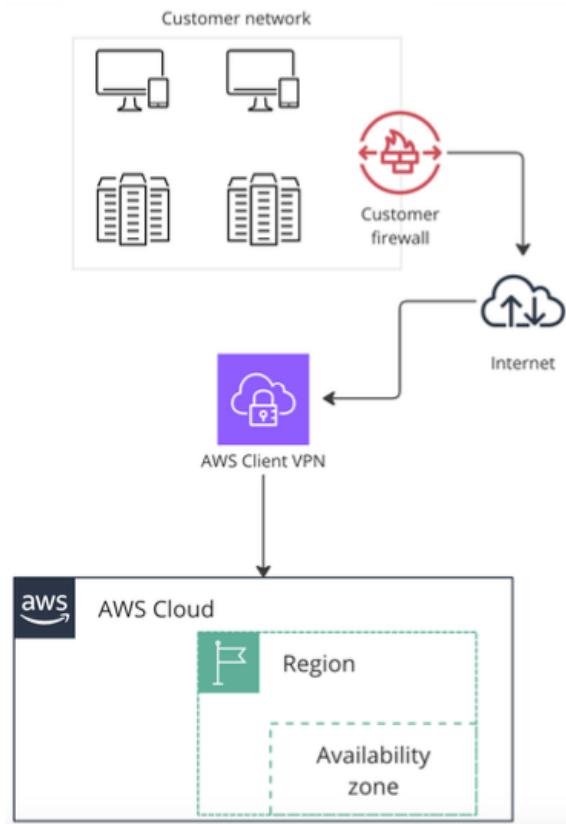
- **Qué es:** Una nube privada virtual dentro de un proveedor de servicios como AWS, Azure o GCP.
- **Para qué sirve:** Para tener una red virtual aislada dentro de la nube, donde puedes desplegar servidores, bases de datos, etc., como si fuera una red interna.
- **Uso típico:**
  - Crear infraestructuras seguras y segmentadas en la nube.
  - Control total sobre las IP, subredes, firewalls y enrutamientos.
- **Cómo funciona:** Es una sección privada de la nube pública donde tú defines cómo se comporta la red, qué entra, qué sale, y cómo se conectan los recursos entre sí.

## VPN (Virtual Private Network)

- **Qué es:** Una red privada virtual.
- **Para qué sirve:** Para conectarte de forma segura a otra red a través de internet.
- **Uso típico:**
  - Acceder de manera segura a recursos de una red corporativa desde casa.
  - Proteger tu conexión en redes públicas.
  - Evadir restricciones geográficas.
- **Cómo funciona:** Crea un túnel cifrado entre tu dispositivo y el servidor al que te conectas, ocultando tu IP y cifrando tus datos.

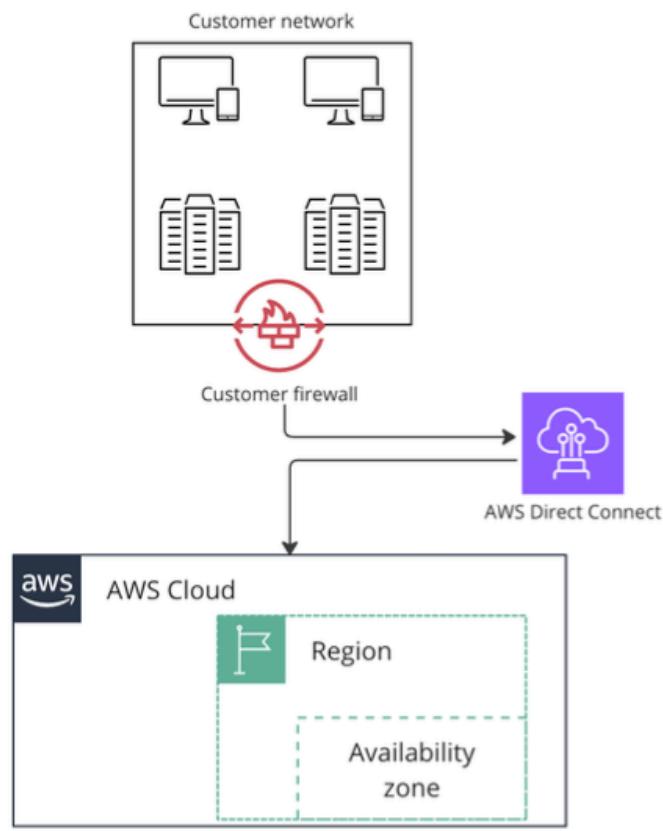
## 2. Connections - AWS VPN

- Securely connects your on-premises network to AWS over the internet
- Flexible and accessible, suitable for smaller workloads or temporary connections



## 3. Connections - AWS Direct Connect

- Dedicated network connection between your on-premises data center and AWS.
- High bandwidth, low latency, ideal for consistent and mission-critical workloads



#### 4. Route 53

Manages domain names and translates them to IP addresses.

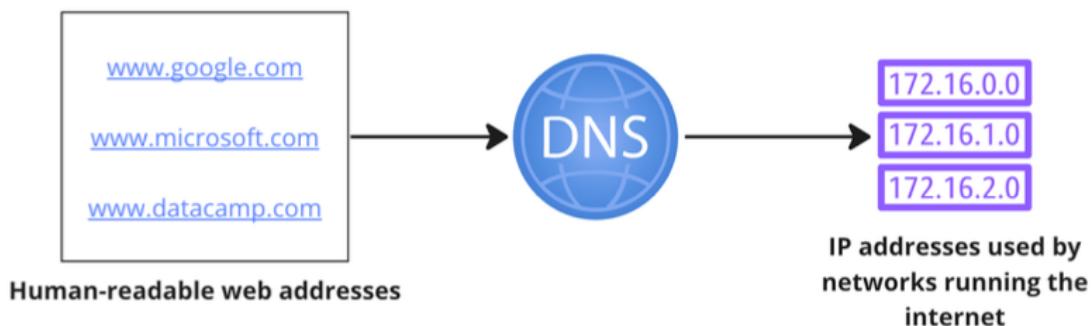
Integration with AWS ecosystem and external services.

- Scalability
- High availability

#### Domain Name System (DNS):

Resolves human-readable domain names to IP addresses used by computers:

- Communication between devices on the internet
- Access websites using user-friendly domain names



## 5. Cloud Front

Enhance the speed and security of content delivery to end-users in AWS

- Integrates seamlessly with AWS services
- Accelerates web content, APIs, and streaming
- Enhances security with DDoS protection and HTTPS support

Uses:



Accelerate website content movement



Optimize streaming & reduced buffer times



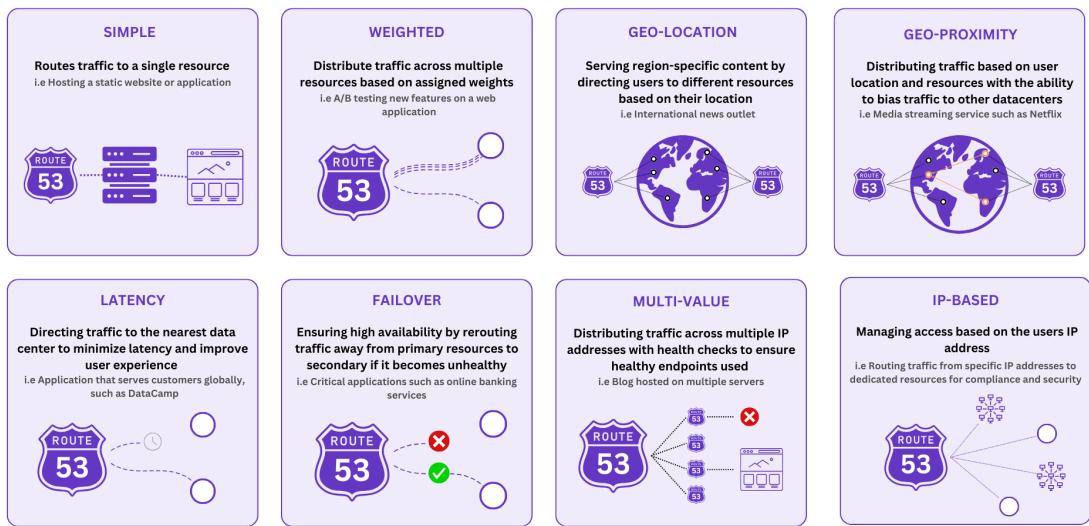
Transmit over-the-air updates to devices

## Content Delivery Networks (CDNs):

- Es una red de **servidores distribuidos por todo el mundo** que se encargan de **entregar contenido web (imágenes, videos, archivos, etc.) de forma más rápida y eficiente** a los usuarios.

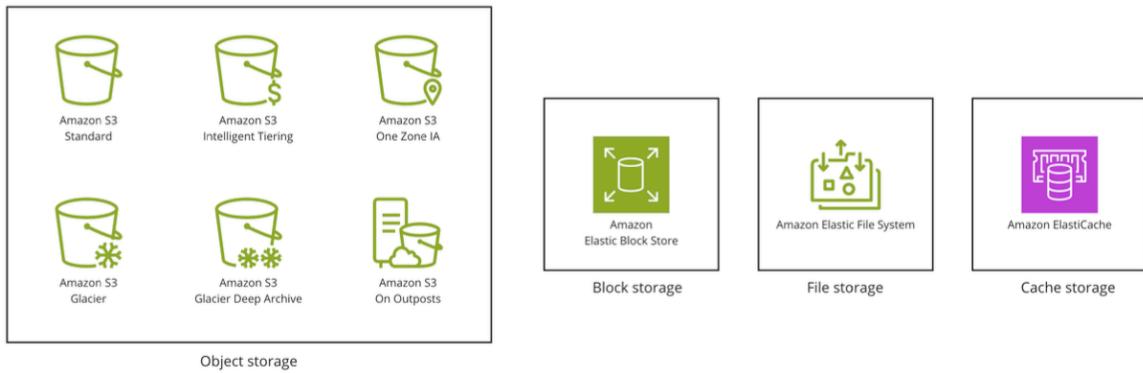
### Key characteristics:

- Caching for faster content loading
- Delivering digital content to end-users over the internet
- Efficiently handle increased user traffic and demand



## ▼ Capítulo 8: Storage services

[https://miro.com/app/board/uXjVIDyBA-U=/?share\\_link\\_id=519430447509](https://miro.com/app/board/uXjVIDyBA-U=/?share_link_id=519430447509)



### 1. Object storage:

Storage architecture that manages and organizes data as discrete units called "objects"

#### Key characteristics:

- Horizontal scaling
- Metadata management

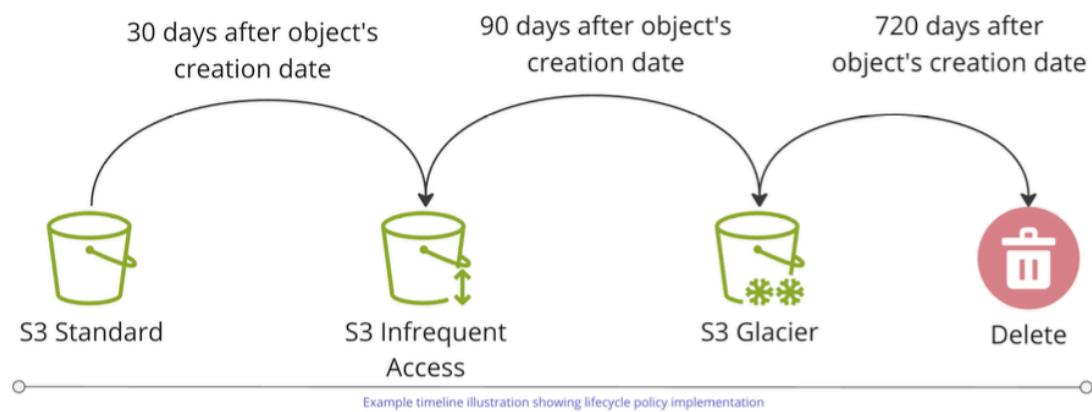
- Storing unstructured data

### 1.1. Amazon S3:

#### Storage lifecycle policies

Defines the transition of objects between storage classes in S3, based on predefined rules

- Cost and performance optimization
- Improves data management and compliance



- A highly scalable and durable object storage service offered by AWS
- Designed for 99.9% durability
- Available in all AWS regions



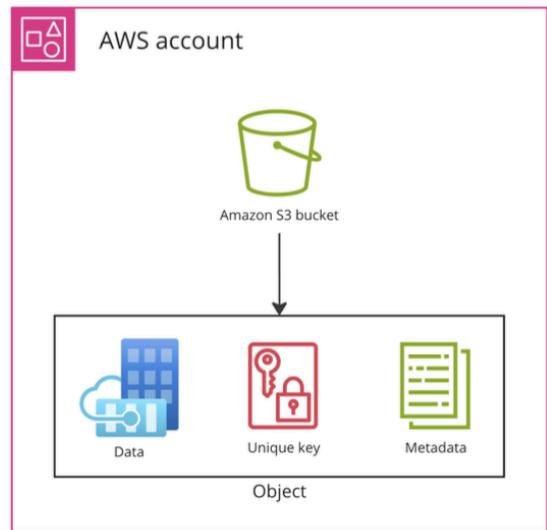
Amazon S3 Standard



Amazon S3 Intelligent Tiering



Amazon S3 One Zone IA



Amazon S3 Glacier



Amazon S3 Glacier Deep Archive



Amazon S3 On Outposts

#### a. Amazon S3 Standard:

#### b. Intelligent Tiering

- Durable, scalable, and available in all AWS regions
- Suitable for frequently accessed data
- Automatic cost optimization
- Moves objects between tiers based on changing access patterns

### c. One Zone-Infrequent Access (IA)

- Cost-effective, single availability zone
- Ideal for infrequently accessed data that can be easily reproduced

### e. Glacier Deep Archive

- Lowest cost, longest retrieval time
- Data with minimal access requirements

### d. Glacier

- Low cost, archival storage
- Long-term archival with retrieval times ranging from minutes to hours

### f. S3 Outposts

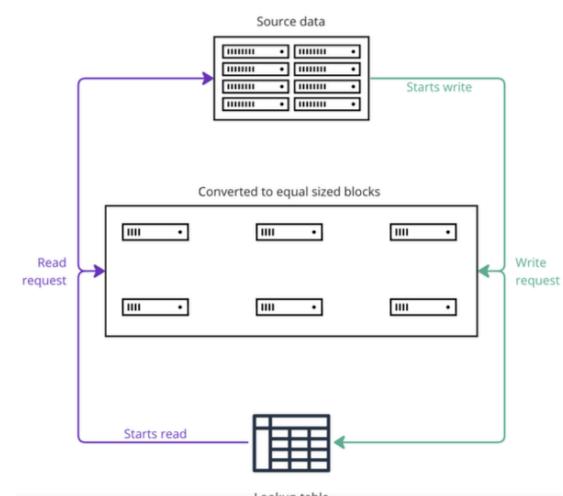
- On-premises storage extension
- Combine private and public cloud data

## 2. Block Storage:

Divides data into fixed-sized blocks, each with its unique address

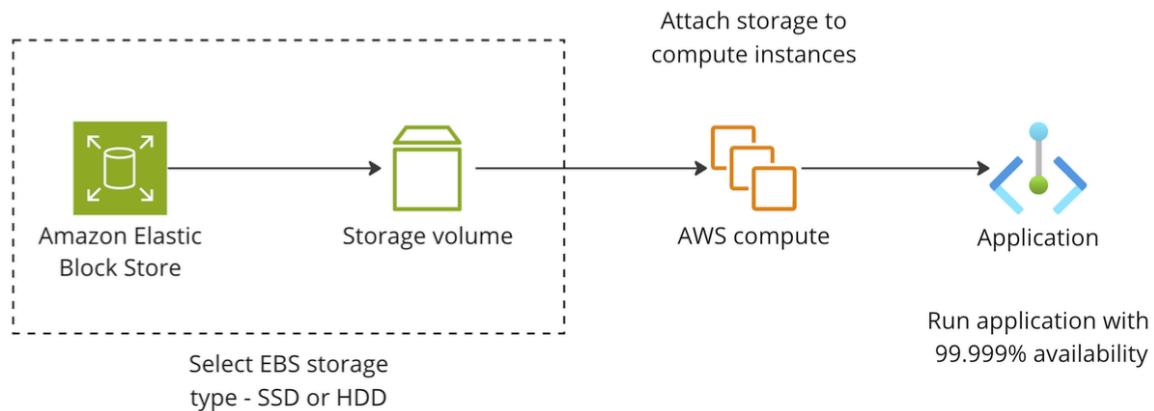
### Use cases

- Running I/O intensive transactional web applications
- Right-size big data analytics engines



### 2.1. Amazon EBS:

A scalable, high-performance block storage service designed for use with Amazon compute services

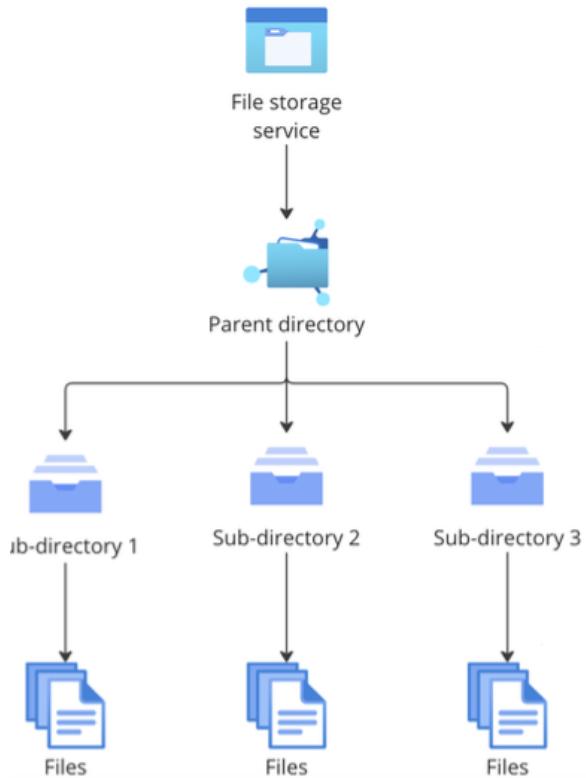


### 3. File Storage:

Organizes and stores data in a hierarchical structure

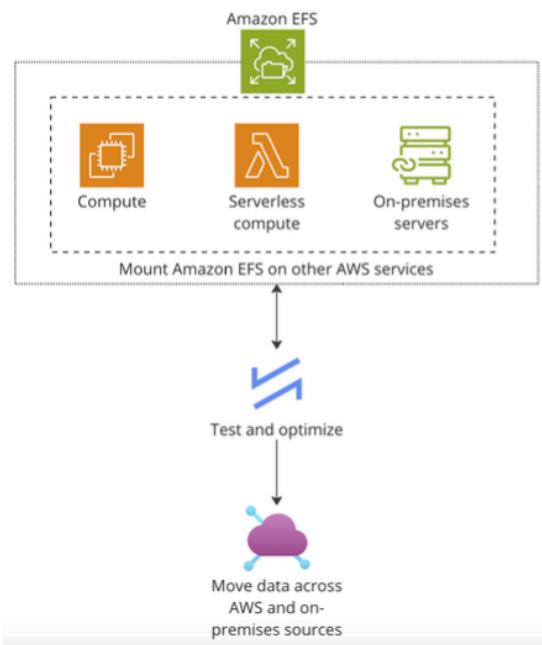
#### Key characteristics

- Allows multiple concurrent reads and writes across users and services
- Stores metadata about files



### 3.1. Amazon EFS

File storage service designed for use with AWS cloud services and on-premises resources



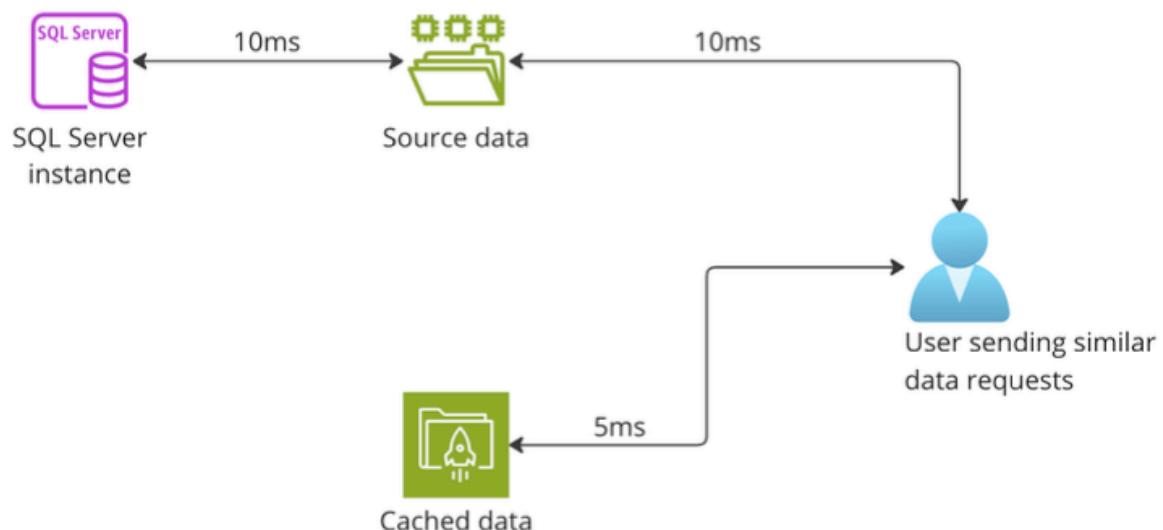
#### Use cases

- Simplify DevOps
- Enhance content management systems
- Accelerate data science

### 4. Cache Storage:

Storing frequently accessed data in a quickly retrievable location

- Accelerates application response times by reducing data retrieval latency
- Minimizes the load on backend servers

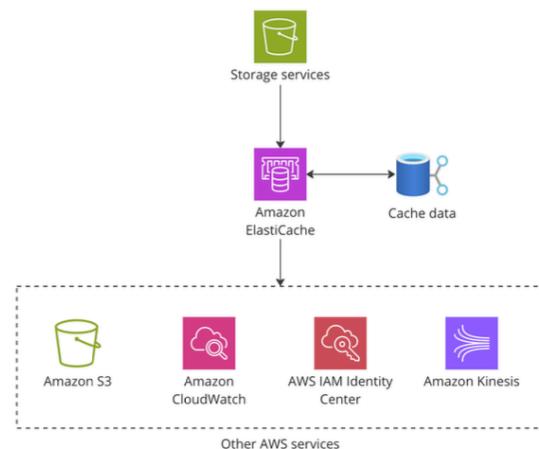


## 4.1. Amazon ElastiCache

Caching service that enables seamless, high-speed access to frequently used data

### Use cases

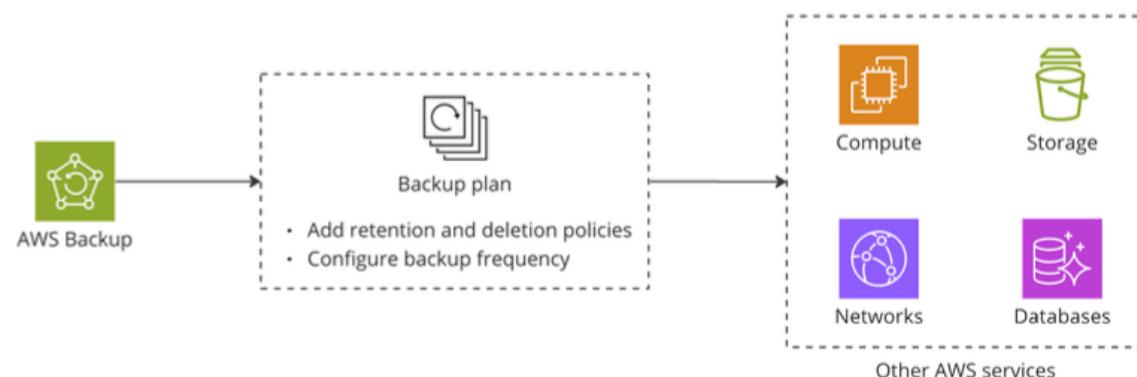
- Store web application session data in-memory
- Accelerates access to real-time analytics data



## 5. AWS Backup:

Cost-effective, fully managed service that centralizes and automates backup across AWS services

- Cross-region backups
- Set retention and deletion policies



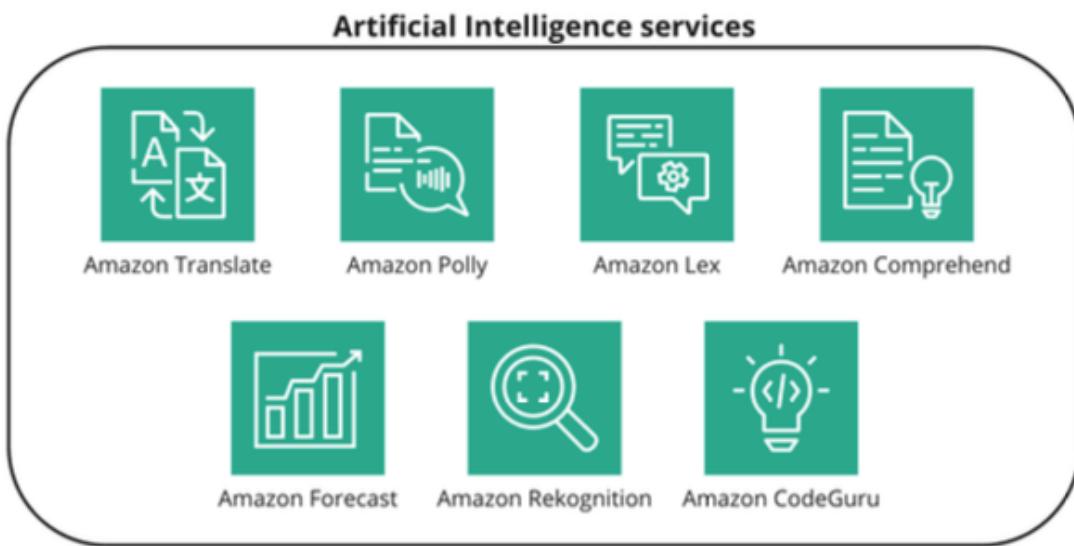
Amazon EBS	Amazon EFS	Amazon ElastiCache
A storage solution that provides high-performance disk storage for EC2 instances, often used for database workloads and boot volumes.	A scalable system that allows multiple EC2 instances to share the same file system, ideal for shared storage needs across instances.	A service that speeds up data retrieval by providing in-memory caching for frequently accessed data, improving performance for read-heavy applications.

# ▼ Capítulo 9: AI & ML services

### Artificial intelligence

- Simulates human intelligence in machines
- Involves tasks like problem-solving, speech recognition, and learning
- Pre-trained, auto-trained models

- No machine learning background needed



### **Amazon Translate**

- Translate text between languages

### **Amazon Polly**

- Convert text into lifelike, natural-sounding speech

### **Amazon Lex**

- Create conversational interfaces
- Design and deploy chatbots

### **Amazon Comprehend**

- Extract insights from text
- Identify sentiments, entities, and language

### **Machine learning**

- Subset of AI
- Focuses on systems that learn from data
- Enable developers to build custom ML models
- Tailored for those with ML expertise and specific use cases

### **Amazon Forecast**

- Build accurate time-series predictions

### **Amazon Rekognition**

- Analyze and recognize objects in images
- Perform facial analysis and sentiment detection in videos

### **Amazon CodeGuru**

- Automate code reviews
- Improve code quality with intelligent recommendations

## Machine Learning services



### Amazon SageMaker

- Fully managed service for end-to-end ML lifecycle
- Integrated Jupyter notebooks for model development
- One-click training and deployment



Predictive analytics



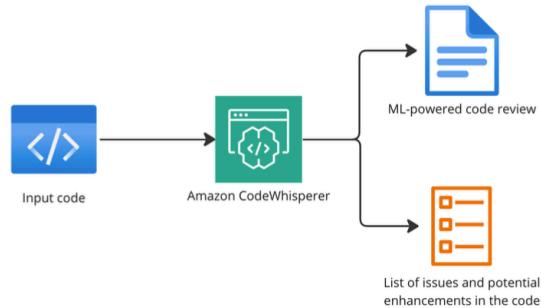
Computer Vision



Natural language processing

### Amazon CodeWhisperer

- Automated ML-powered and streamlined code review
- Enhances code quality and identifies issues



## Framworks:

- Open-source frameworks for diverse ML workflows
- Enables robust, scalable, and seamless deployment
- Caters to a wide range of uses



TensorFlow

Development and deployment of scalable ML models on AWS



PyTorch

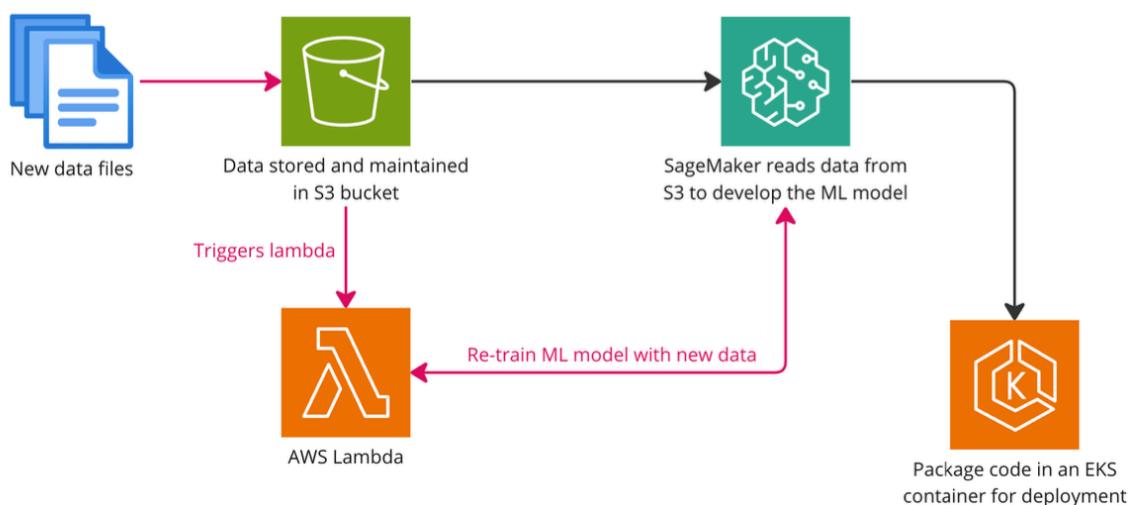
Execute machine learning and computational graph-based systems on-the-fly



MxNet

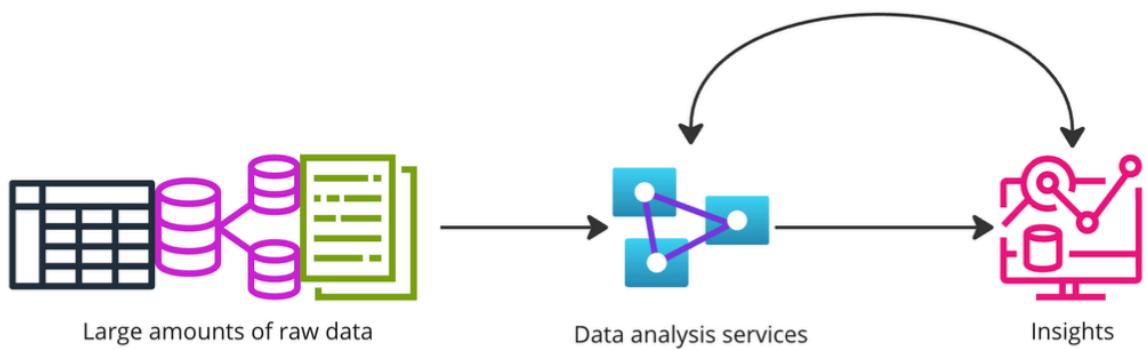
Large-scale, distributed training of deep neural networks

## Sample ML pipeline



## ▼ Capítulo 10: Analytics and BI services

- Turning raw data into actionable insights
- Iterative process



### Tools:



Amazon Athena



Amazon QuickSight



Amazon Kinesis



Amazon Redshift



Amazon Macie



AWS Glue

#### **Amazon Athena**

Serverless, interactive analytics service for analyzing petabyte-scale data at source

#### **Key features**

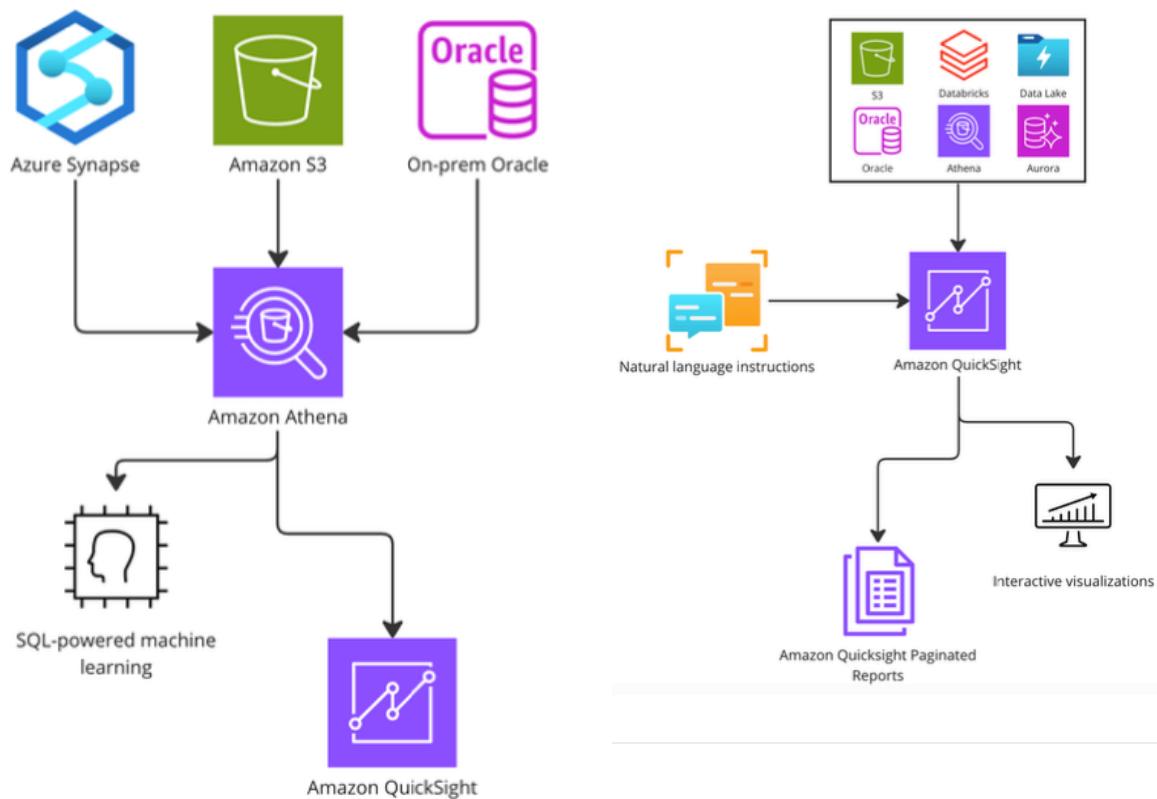
- Versatile data integration
- Simplified pricing of paying per query

#### **Amazon QuickSight**

Unified business intelligence service supporting interactive dashboards

#### **Key features**

- Automatic scaling without server setup
- Generative BI capabilities
- Paginated reports



### Amazon Kinesis

Enables the collection, processing, and analysis of data streams at scale

#### Key features

- Real-time processing
- Serverless infrastructure
- Scalability

### Amazon Redshift

AI-powered, Massively Parallel Processing (MPP) cloud data warehouse for analytics

#### Key features

- Achieve up to 6x better price performance
- Zero-ETL approach
- Secure collaboration and governance



Real-time live leaderboards



IoT sensor data processing



Improve data transfer speeds



### Securing data with Amazon Macie

ML-driven data security service to enable sensitive data discovery and protection

### AWS Glue

Serverless data integration service allowing the discovery, preparation and integration of data

#### Key features

- Automated discovery
- Cost-efficient visibility
- Reduce triage time with actionable reporting on sensitive data

#### Key features

- Support both batch and streaming data
- Prepare data for machine learning and deep learning
- Serverless petabyte-scale capacity



Store PII data confidently



Batch and streaming data integration



Stay compliant



Prepare data for ML and DL tasks

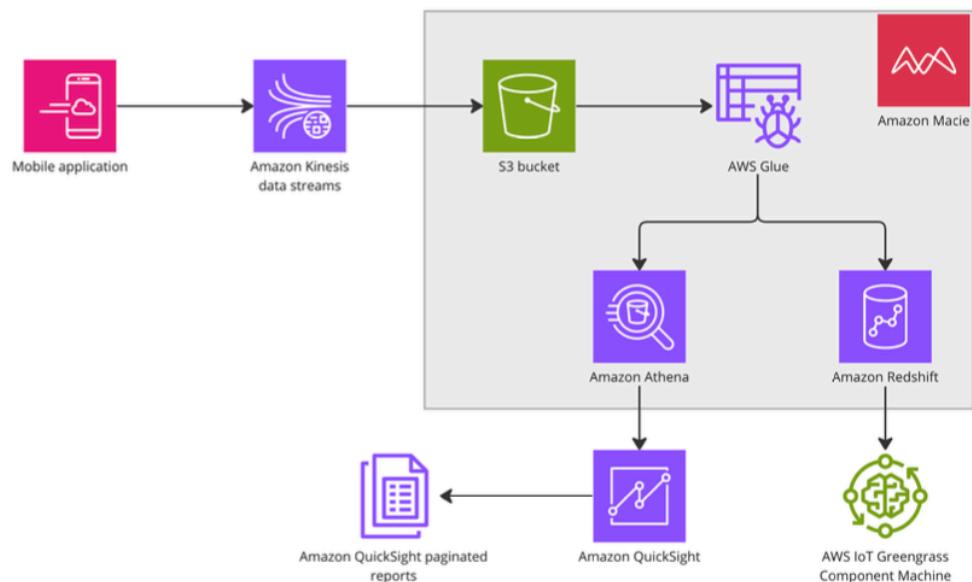


Identify risks before migrating data to the cloud



Simplify ETL workflows

## Creating an end-to-end data workflow



## ▼ Capítulo 11: Secondary services

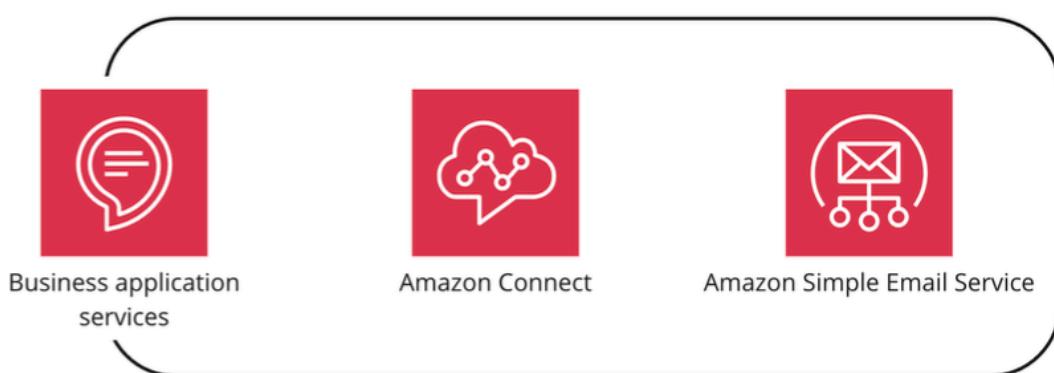
### 1. Application integration services

- Enhance communication between diverse AWS applications
- Coordinate data and workflows for efficiency
- Crucial for scalable, flexible, interconnected architectures



### 2. Business application services

- Streamline operations and enhance efficiency of business applications
- Seamless integration, automation, and improved productivity



### 3. Developer services

- Cohesive environment for simplifying and accelerating the development lifecycle
- Enable collaborative development within teams and tools
- Automated delivery and deployment using CI/CD



#### 4. Advanced intelligence services

- Extend artificial intelligence reach by linking it to physical devices
- Expand computational capabilities with revolutionary quantum processing technology

