

AWS Cloud Security Foundations– AWS-CSF

Dr. Ibrahim Gomaa

Lecture 4 – AWS foundational services – Storage and Database



Course Outline

Lecture 4: AWS Foundational Services – Storage and Database

Topics

- Amazon Elastic Block Store (Amazon EBS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Elastic File System (Amazon EFS)
- Amazon Simple Storage Service Glacier
- Amazon Relational Database Service (Amazon RDS)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Aurora

Activities

- Storage solution case study
- Database case studies

Demo

- Amazon EBS console
- Amazon RDS console

Lab

- Working with Amazon EBS
- Build Your DB Server and Interact with Your DB Using an App



Knowledge check



مجموعة جامعة
الجميع الذكية
للتعليم القابضة

AWS Cloud Security Foundations– AWS-CSF

Lecture 4 – AWS Foundational Services – Storage and Database

Dr. Ibrahim Gomaa

Amazon Elastic Block Store (Amazon EBS)



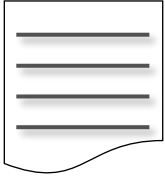
Amazon Elastic Block Store (Amazon EBS)

Amazon EBS (non-volatile storage) is:

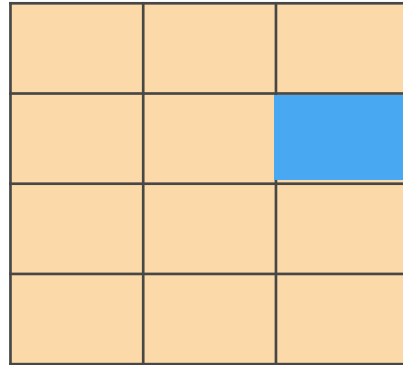


- **persistent** (retains data after power off), **mountable storage** that can be mounted as a device to an Amazon EC2 instance.
- **Amazon EBS can be mounted** to an Amazon EC2 instance only within the same Availability Zone. Only one Amazon EC2 instance at a time can mount an Amazon EBS volume.
- **automatically replicated** within its Availability Zone to protect you from component failure. (designed for high availability and durability, consistent and low-latency performance to run your workloads).
- can **scale your usage up or down within minutes**, while paying a low price for only what you provision.

AWS storage options: Block storage versus object storage

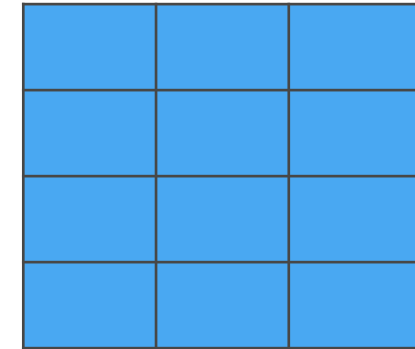


What if you want to change one character in a 1-GB file?



Block storage

Change one block (piece of the file)
that contains the character



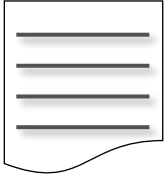
Object storage

Entire file must be updated

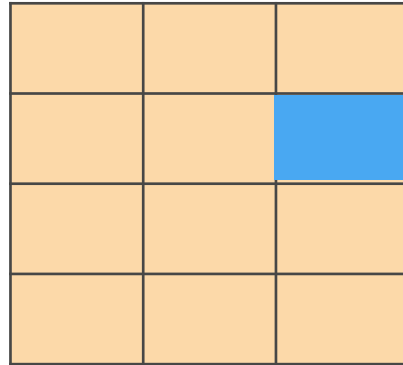
Block storage solutions are typically faster and use less bandwidth, but they can cost more than object-level storage.



AWS storage options: Block storage versus object storage

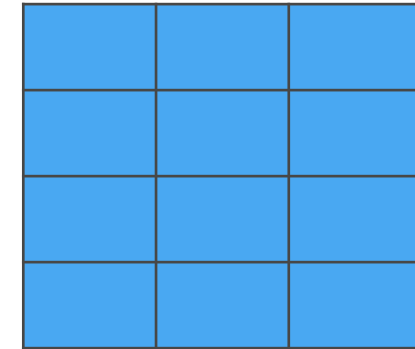


What if you want to change one character in a 1-GB file?



Block storage

Change one block (piece of the file)
that contains the character



Object storage

Entire file must be updated

Block storage solutions are typically faster and use less bandwidth, but they can cost more than object-level storage.



Amazon EBS enables you to **create individual storage volumes** and attach them to an Amazon EC2 instance:

- Amazon EBS offers **block-level storage**.
- Volumes are automatically replicated within its Availability Zone.
- It can be backed up automatically to Amazon S3 through snapshots (The first snapshot is called the baseline snapshot).
- Uses include –
 - Boot volumes and storage for Amazon EC2 instances
 - Data storage with a file system
 - Database hosts
 - Enterprise applications



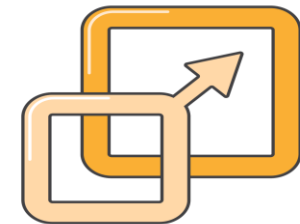
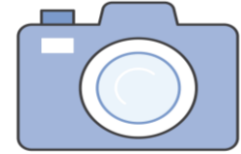
Amazon EBS volume types

- Matching the correct technology to your workload is a best practice for reducing storage costs.
- Only SSDs can be used as boot volumes for EC2 instances.
- The lower-cost options might be a solution for additional storage or use cases other than boot volumes.

	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
	General Purpose	Provisioned IOPS	Throughput-Optimized	Cold
Maximum Volume Size	16 TiB	16 TiB	16 TiB	16 TiB
Maximum IOPS/Volume	16,000	64,000	500	250
Maximum Throughput/Volume	250 MiB/s	1,000 MiB/s	500 MiB/s	250 MiB/s

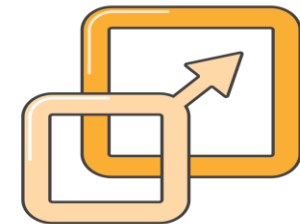
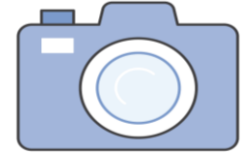
Amazon EBS features

- Snapshots –
 - Point-in-time snapshots
 - Recreate a new volume at any time
- Encryption –
 - Encrypted Amazon EBS volumes
 - No additional cost
- Elasticity –
 - Increase capacity
 - Change to different types



Amazon EBS features

- Snapshots –
 - Point-in-time snapshots
 - Recreate a new volume at any time
- Encryption –
 - Encrypted Amazon EBS volumes
 - No additional cost
- Elasticity –
 - Increase capacity
 - Change to different types



Amazon Simple Storage Service (Amazon S3)



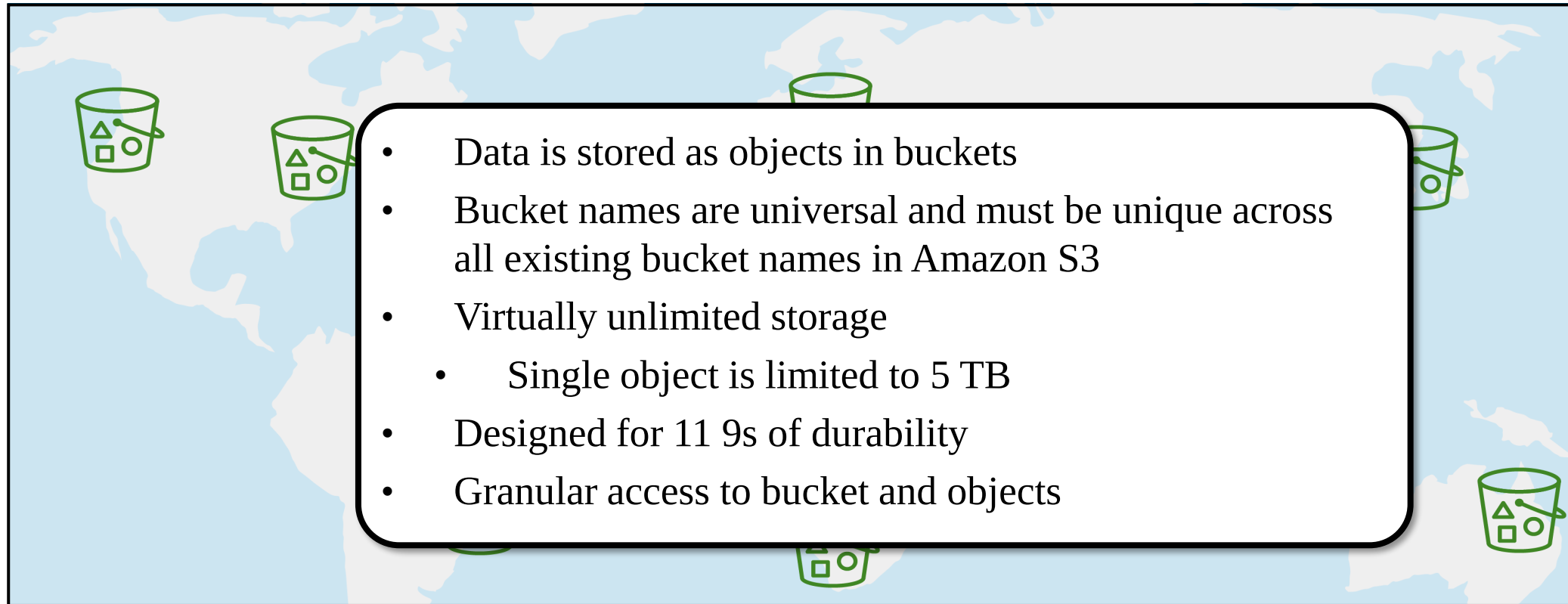
Amazon Simple Storage Service (Amazon S3)



Amazon S3 is:

- **persistent storage** where each file becomes **an object** and is available through a Uniform Resource Locator (URL); it can be accessed from anywhere.
- **object-level storage**, which means that if you want to change a part of a file, you must make the change and then re-upload the entire modified file.
- stores data as objects within resources that are called **buckets**.

Amazon S3 overview



- Data is stored as objects in buckets
- Bucket names are universal and must be unique across all existing bucket names in Amazon S3
- Virtually unlimited storage
 - Single object is limited to 5 TB
- Designed for 11 9s of durability
- Granular access to bucket and objects

Amazon S3 storage classes

Amazon S3 offers a range of object-level storage classes that are designed for different use cases:

- **Amazon S3 Standard** (high durability, availability, performance, low latency and high throughput, use cases: cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics)
- **Amazon S3 Intelligent-Tiering** (optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. Ex: objects that have not been accessed for 30 consecutive days to the infrequent access tier)
- **Amazon S3 Standard-Infrequent Access (Amazon S3 Standard-IA)** (low cost and high performance makes Amazon S3 Standard-IA good for long-term storage and backups, and as a data store for disaster recovery files)
- **Amazon S3 One Zone-Infrequent Access (Amazon S3 One Zone-IA)** (Unlike other Amazon S3 storage classes, which store data in a minimum of three AZ, it stores data in a single AZ and it costs less than Amazon S3 Standard-IA. Use cases storing secondary backup copies of on-premises and cost-effective storage for data that is replicated from another AWS Region by using Amazon S3 Cross-Region Replication.)
- **Amazon S3 Glacier** (secure, durable, and low-cost storage class for data archiving. costs are competitive with—or cheaper than—on-premises solutions. To keep costs low yet suitable for varying needs, Amazon S3 Glacier provides three retrieval options that range from a few minutes to hours)
- **Amazon S3 Glacier Deep Archive** (the lowest-cost storage class for Amazon S3. It supports long-term retention and digital preservation for data that might be accessed once or twice in a year. All objects that are stored in Amazon S3 Glacier Deep Archive are replicated and stored across at least three geographically dispersed Availability Zones, and these objects can be restored within 12 hours)

Amazon S3 bucket URLs (two styles)

To upload your data:

1. Create a **bucket** in an AWS Region.
2. Upload almost any number of **objects** to the bucket.

Amazon S3



[bucket name]



Preview2.mp4

Tokyo Region (ap-northeast-1)

Bucket path-style URL endpoint:

<https://s3.ap-northeast-1.amazonaws.com/bucket-name>

Region code

Bucket name

Bucket virtual hosted-style URL endpoint:

<https://bucket-name.s3-ap-northeast-1.amazonaws.com>

Bucket name

Region code

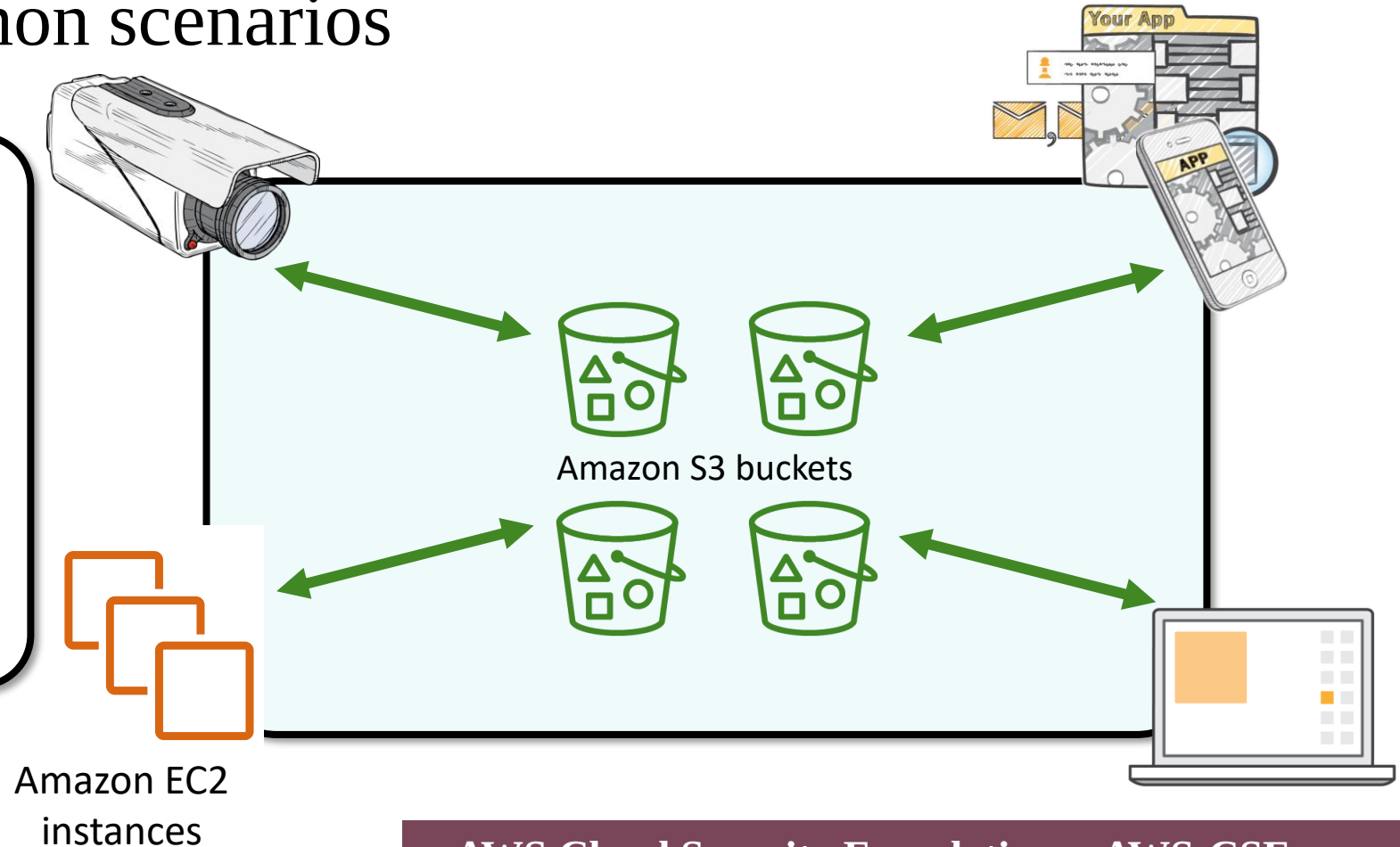
Common use cases

- Storing application assets
- Static web hosting
- Backup and disaster recovery (DR)
- Staging area for big data
- Many more....



Amazon S3 common scenarios

- Backup and storage
- Application hosting
- Media hosting
- Software delivery



Amazon S3 pricing

- Pay only for what you use, including –
 - GBs per month
 - Transfer OUT to other Regions
 - PUT, COPY, POST, LIST, and GET requests
- You do not pay for –
 - Transfers IN to Amazon S3
 - Transfers OUT from Amazon S3 to Amazon CloudFront or Amazon EC2 in the same Region

Amazon S3: Storage pricing

To estimate Amazon S3 costs, consider the following:

1. Storage class type –

- Standard storage is designed for:
 - 11 9s of durability
 - Four 9s of availability
- S3 Standard-Infrequent Access (S-IA) is designed for:
 - 11 9s of durability
 - Three 9s of availability

2. Amount of storage –

- The number and size of objects

Amazon S3: Storage pricing

3. Requests –

- The number and type of requests (**GET, PUT, COPY**)
- Type of requests:
 - Different rates for GET requests than other requests.

4. Data transfer –

- Pricing is based on the amount of data that is transferred out of the Amazon S3 Region
 - Data transfer in is free, but you incur charges for data that is transferred out.



Amazon Elastic File System (Amazon EFS)



Amazon Elastic File System (Amazon EFS)



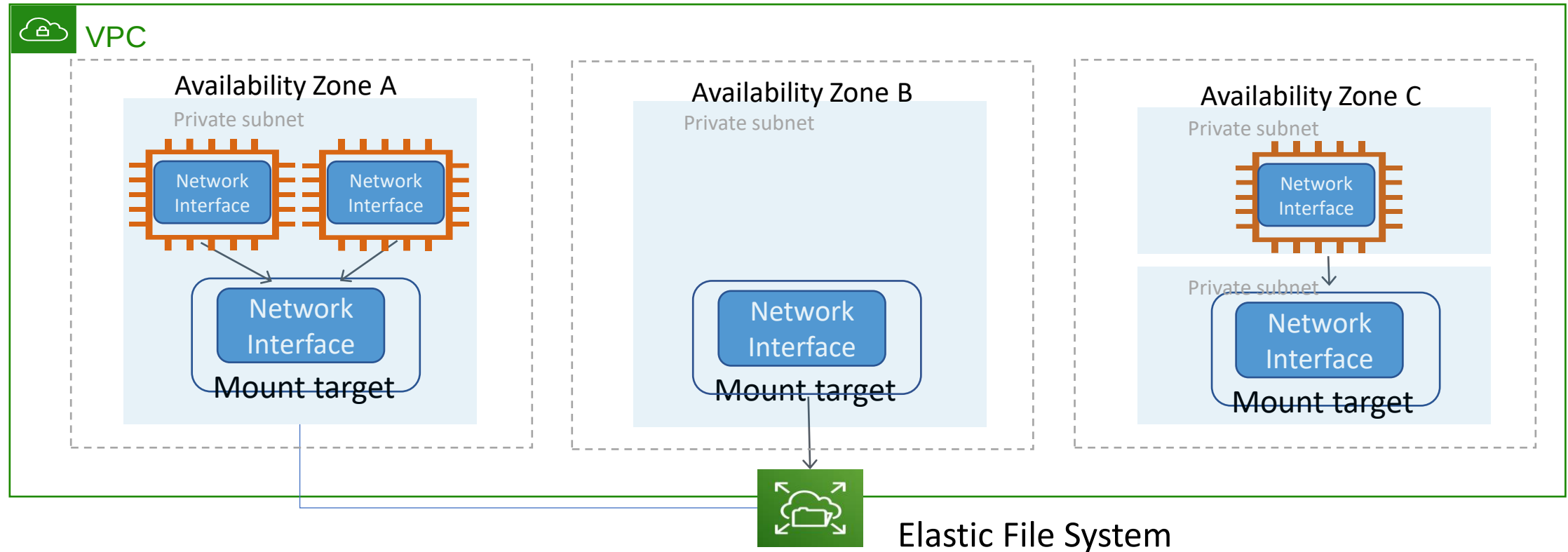
Amazon EFS is

- a **shared file system** that multiple Amazon EC2 instances can mount at the same time.
- simple, scalable, elastic file storage for use with AWS services and on-premises resources.
- **dynamically scale** on demand without disrupting applications—it will grow and shrink automatically as you add and remove files.
- It is designed so that your applications have the storage they need, when they need it.

Amazon EFS features

- File storage in the AWS Cloud
- Works well for big data and analytics, media processing workflows, content management, web serving, and home directories
- Petabyte-scale, low-latency file system
- Shared storage
- Elastic capacity
- Supports Network File System (NFS) versions 4.0 and 4.1 (NFSv4)
- Compatible with all Linux-based AMIs for Amazon EC2

Amazon EFS architecture



Amazon S3 Glacier



Amazon S3 Glacier



Amazon S3 Glacier is:

for cold storage for data that is not accessed frequently
for example,
(when you need long-term data storage for archival or compliance reasons).

Amazon S3 Glacier review

Amazon S3 Glacier is a **data archiving service** that is designed for **security**, **durability**, and an **extremely low cost**.

Amazon S3 Glacier is designed to provide 11 9s of durability for objects.

Three key Amazon S3 Glacier terms:

Archive: Any object such as a photo, video, file, or document that you store in Amazon S3 Glacier. It is the base unit of storage in Amazon S3 Glacier. Each archive has its own **unique ID** and can also have a **description**.

Vault: A container for storing archives. When you create a vault, you specify the **vault name** and **the region** in which you would like to locate the vault.

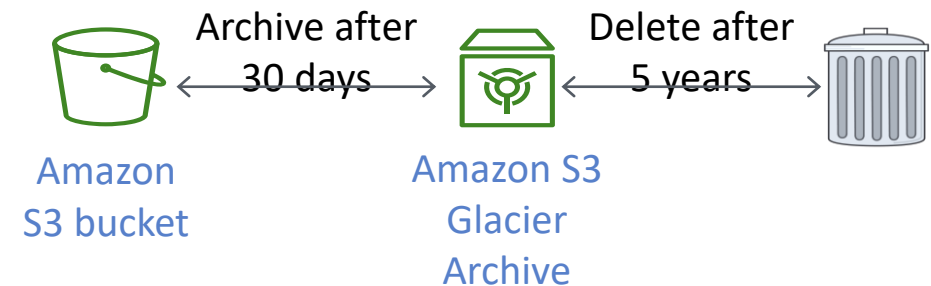
Vault Access Policy: Determine who can and cannot access the data stored in the vault and what operations users can and cannot perform. **One vault access policy** can be created for each vault to manage access permissions for that vault. You can also use a **vault lock policy** to make sure a vault cannot be altered. Each vault can have **one vault access policy and one vault lock policy** that is attached to it.

The Vault Lock feature enforces compliance through a policy.

Extremely low-cost design works well for long-term archiving.

Provides three options for access to archives— **expedited**, **standard**, and **bulk** —retrieval times range from a few minutes to several hours.

- Storage service for low-cost data archiving and long-term backup
- You can configure lifecycle archiving of Amazon S3 content to Amazon S3 Glacier
- Retrieval options –
 - **Standard**: 3–5 hours
 - **Bulk**: 5–12 hours
 - **Expedited**: 1–5 minutes



Amazon S3 Glacier use cases



Media asset archiving



Healthcare information archiving



Regulatory and compliance archiving



Scientific data archiving



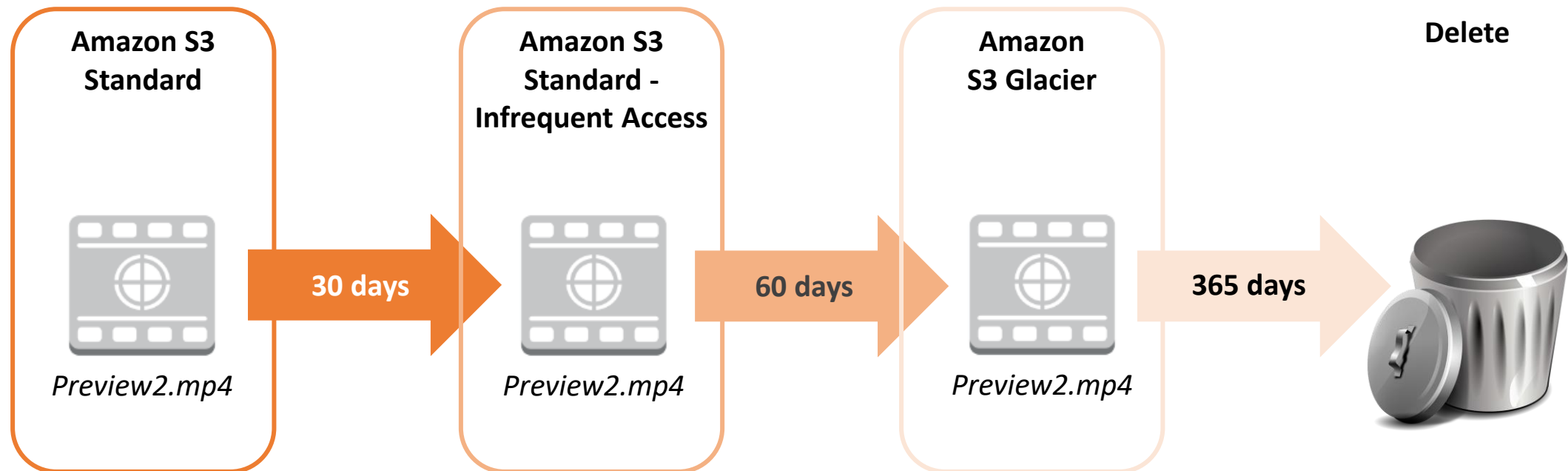
Digital preservation



Magnetic tape replacement

Lifecycle policies

Amazon S3 lifecycle policies enable you to delete or move objects based on age.



Storage comparison

Data Volume

Average Latency

Item Size

Cost/GB per Month

Billed Requests

Retrieval Pricing

Amazon S3	Amazon S3 Glacier
No limit	No limit
ms	minutes/hours
5 TB maximum	40 TB maximum
Higher cost	Lower cost
PUT, COPY, POST, LIST, and GET	UPLOAD and retrieval
¢ Per request	¢¢ Per request and per GB

Security with Amazon S3 Glacier

Amazon S3
Glacier

Control access with **IAM**



Amazon S3 Glacier encrypts
your data with **AES-256**

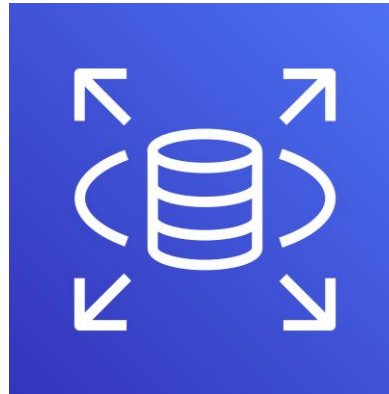


Amazon S3 Glacier manages your
keys for you

Amazon Relational Database Service



Amazon RDS is a fully managed relational database service that creates and operates a relational database in the cloud



Amazon Relational Database Service
(Amazon RDS)

Unmanaged versus managed services

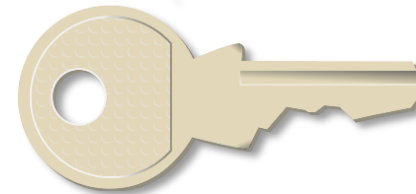
Unmanaged:

Scaling, fault tolerance, and availability are managed by you.



Managed:

Scaling, fault tolerance, and availability are typically built in to the service.



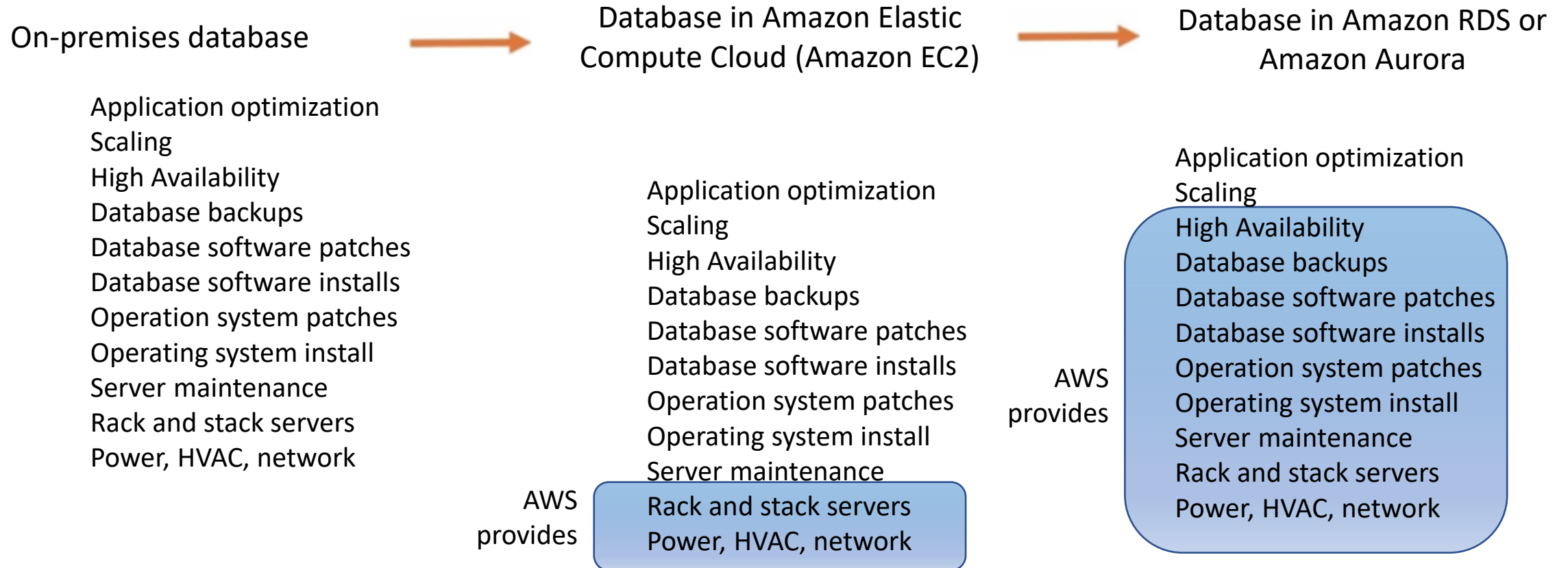
Challenges of relational databases

When you run your own relational database, you are responsible for:

- Server maintenance and energy footprint
- Software installation and patches
- Database backups and high availability
- Limits on scalability
- Data security
- Operating system (OS) installation and patches



From on-premises databases to Amazon RDS



Amazon Relational Database Service

Managed services responsibilities

You manage:

- Application optimization

AWS manages:

- OS installation and patches
- Database software installation and patches
- Database backups
- High availability
- Scaling
- Power and racking and stacking servers
- Server maintenance



Amazon RDS

Amazon Relational Database Service

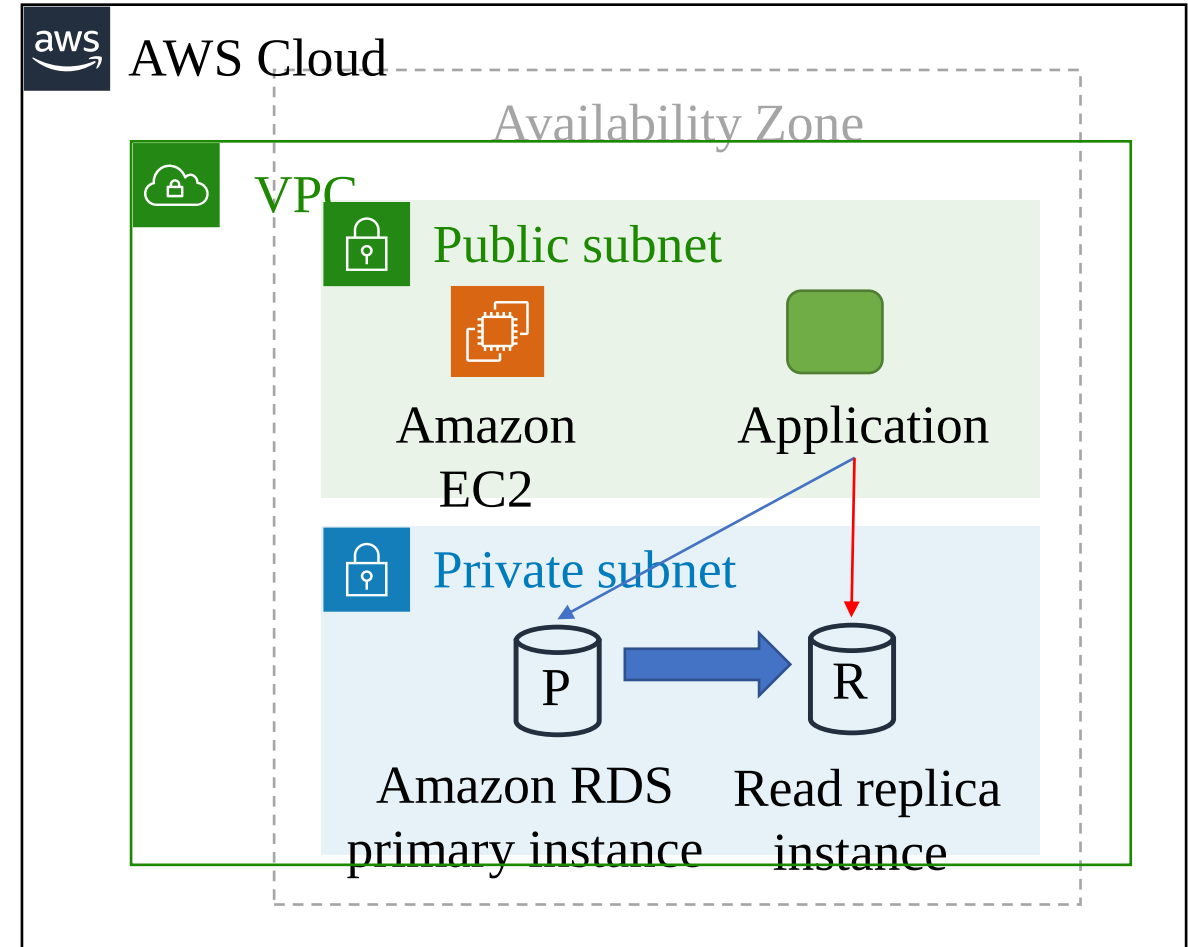
Amazon RDS read replicas

Features

- Offers asynchronous replication
- Can be promoted to primary if needed
- can be created in a different Region than the primary database. (*DR or reduce latency*)

Functionality

- Use for read-heavy database workloads
- Offload read queries



AWS Cloud Security Foundations– AWS-CSF

When to Use Amazon RDS

Use Amazon RDS when your application requires:

- Complex transactions or complex queries
- A medium to high query or write rate – Up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

Do not use Amazon RDS when your application requires:

- Massive read/write rates (for example, 150,000 write/second)
- Sharding due to high data size or throughput demands
- Simple GET or PUT requests and queries that a NoSQL database can handle
- Relational database management system (RDBMS) customization

When to Use Amazon RDS

Use Amazon RDS when your application requires:

- Complex transactions or complex queries
- A medium to high query or write rate – Up to 30,000 IOPS (15,000 reads + 15,000 writes)
- No more than a single worker node or shard
- High durability

Do not use Amazon RDS when your application requires:

- Massive read/write rates (for example, 150,000 write/second)
- Sharding due to high data size or throughput demands
- Simple GET or PUT requests and queries that a NoSQL database can handle
- Relational database management system (RDBMS) customization

Amazon DynamoDB



What is Amazon DynamoDB?

Fast and flexible NoSQL database service for any scale



Amazon DynamoDB

- NoSQL database tables
- Virtually unlimited storage
- Items can have differing attributes
- Low-latency queries
- Scalable read/write throughput

Amazon Relational Database Service

Relational versus non-relational databases

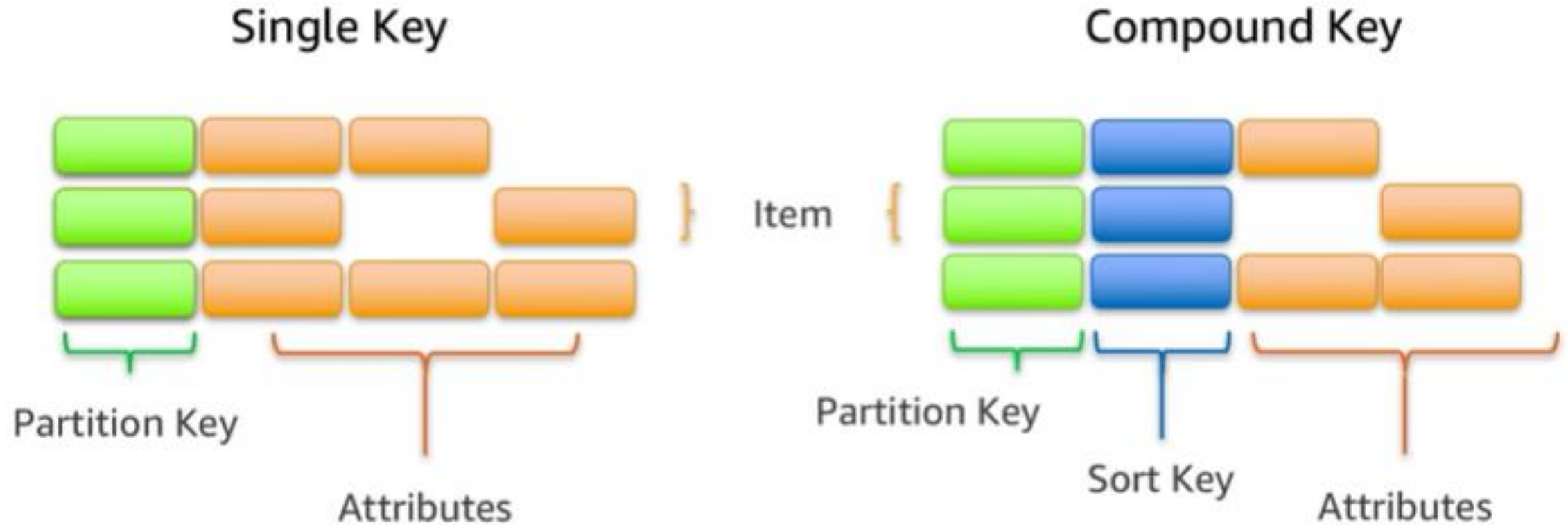
	Relational (SQL)	Non-Relational												
Data Storage	Rows and columns	Key-value, document, graph												
Schemas	Fixed	Dynamic												
Querying	Uses SQL	Focuses on collection of documents												
Scalability	Vertical	Horizontal												
Example	<table><tr><th>ISBN</th><th>Title</th><th>Author</th><th>Format</th></tr><tr><td>3111111223439</td><td>Withering Depths</td><td>Jackson, Mateo</td><td>Paperback</td></tr><tr><td>3122222223439</td><td>Wily Willy</td><td>Wang, Xiulan</td><td>Ebook</td></tr></table>	ISBN	Title	Author	Format	3111111223439	Withering Depths	Jackson, Mateo	Paperback	3122222223439	Wily Willy	Wang, Xiulan	Ebook	<pre>{ ISBN: 3111111223439, Title: "Withering Depths", Author: "Jackson, Mateo", Format: "Paperback" }</pre>
ISBN	Title	Author	Format											
3111111223439	Withering Depths	Jackson, Mateo	Paperback											
3122222223439	Wily Willy	Wang, Xiulan	Ebook											

Amazon DynamoDB core components

- **Tables, items, and attributes** are the core DynamoDB components
 - **A table** is a collection of data.
 - **Items** are a group of attributes that is uniquely identifiable among all the other items.
 - **Attributes** are a fundamental data element, something that does not need to be broken down any further.
- DynamoDB supports two different kinds of primary keys:
 - **The partition key** is a simple primary key, which is composed of one attribute called the **sort** key.
 - **The partition key and sort key** are also known as the **composite primary key**, which is composed of two attributes.

Amazon Relational Database Service

Items in a table must have a key



Amazon Redshift



Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data by using standard SQL and your existing business intelligence (BI) tools

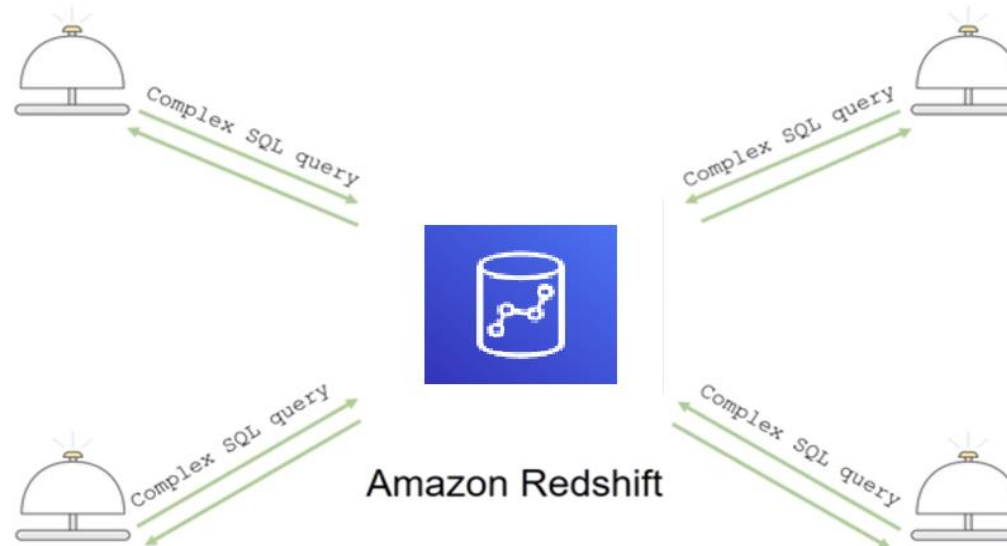


Amazon Redshift

Introduction to Amazon Redshift

Amazon Redshift is a fast and powerful, **fully managed data warehouse** that is simple and cost-effective to set up, use, and scale.

- It enables you to run **complex analytic queries** against **petabytes of structured data**
- Most results come back in **seconds**

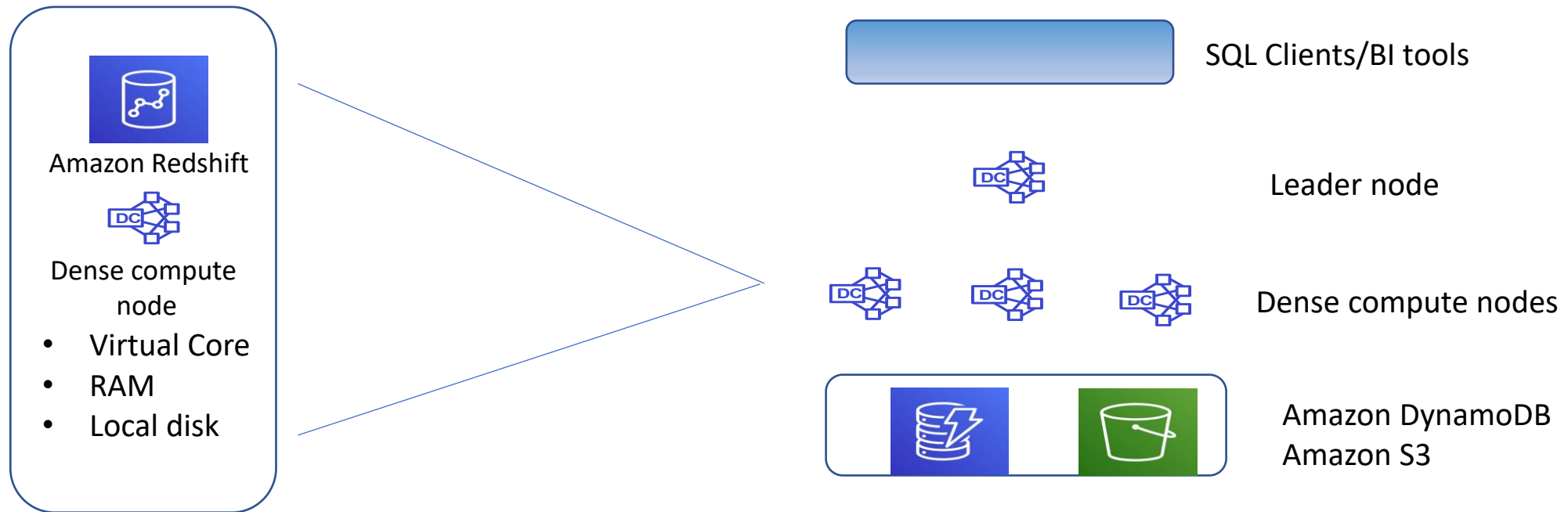


Parallel processing architecture

The **leader node** manages communications with client programs and all communication with **compute nodes**.

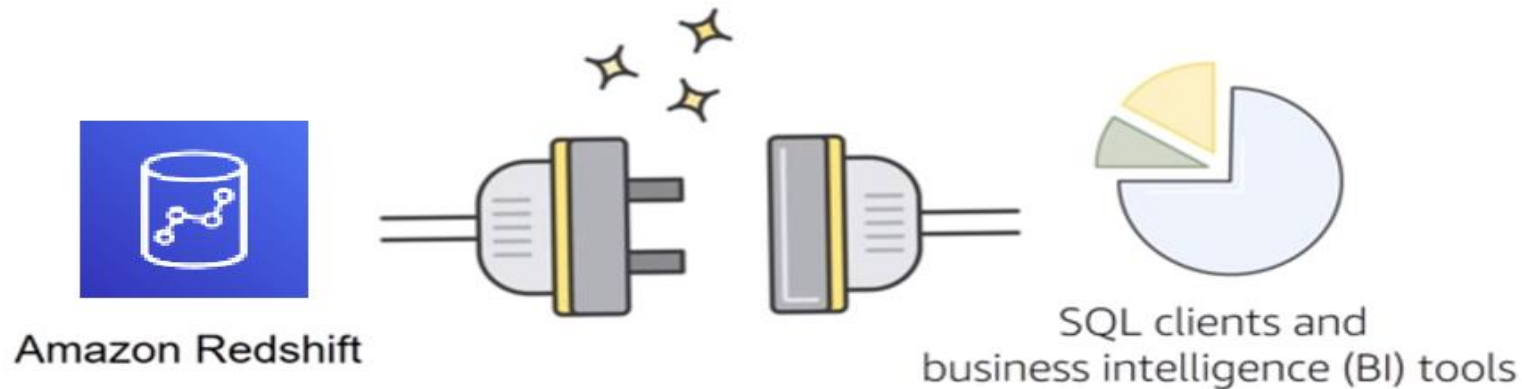
The leader node compiles code for individual elements of the plan and assigns the code to individual compute nodes.

The compute nodes run the compiled code and send intermediate results back to the leader node for final aggregation.



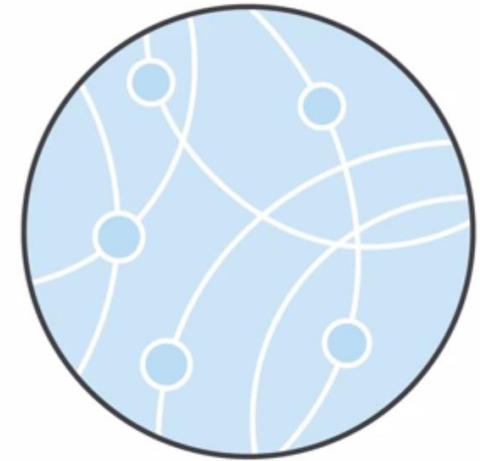
Compatibility

- **Amazon Redshift** supports **standard SQL**.
- It also provides high-performance **Java Database Connectivity (JDBC)** and **Open Database Connectivity (ODBC) connectors**, which enable you to use the SQL clients and BI tools of your choice



Amazon Redshift use cases

- Enterprise **data warehouse** (EDW)
 - Migrate at a pace that customers are comfortable with
 - Experiment without large upfront cost or commitment
 - Respond faster to business needs
- Big data
 - Low price point for small customers
 - Managed service for ease of deployment and maintenance
 - Focus more on data and less on database management



Amazon Aurora





Amazon Aurora

- Enterprise-class relational database
- Compatible with MySQL or PostgreSQL
- Automate time-consuming tasks (such as provisioning, patching, backup, recovery, failure detection, and repair).

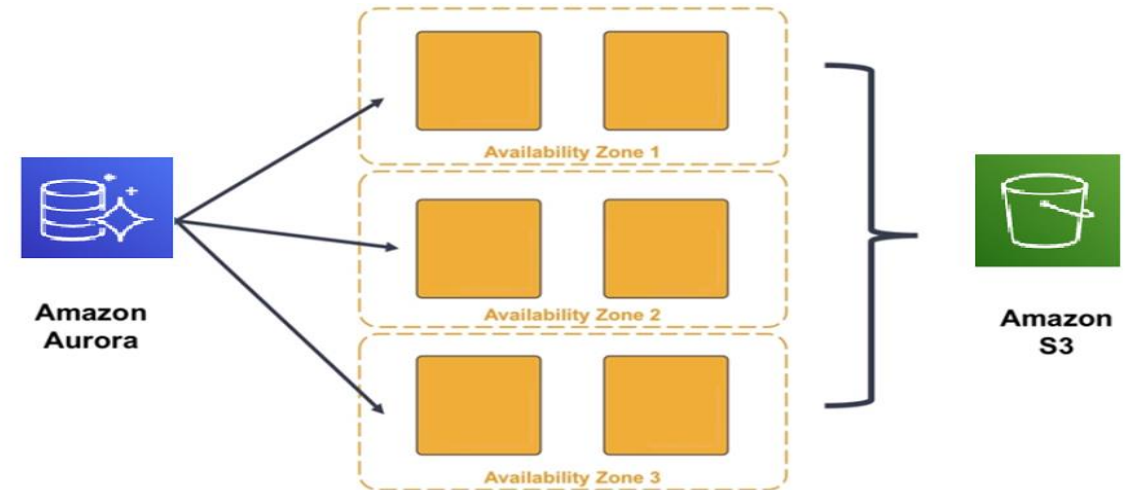
Amazon Aurora service benefits



High availability

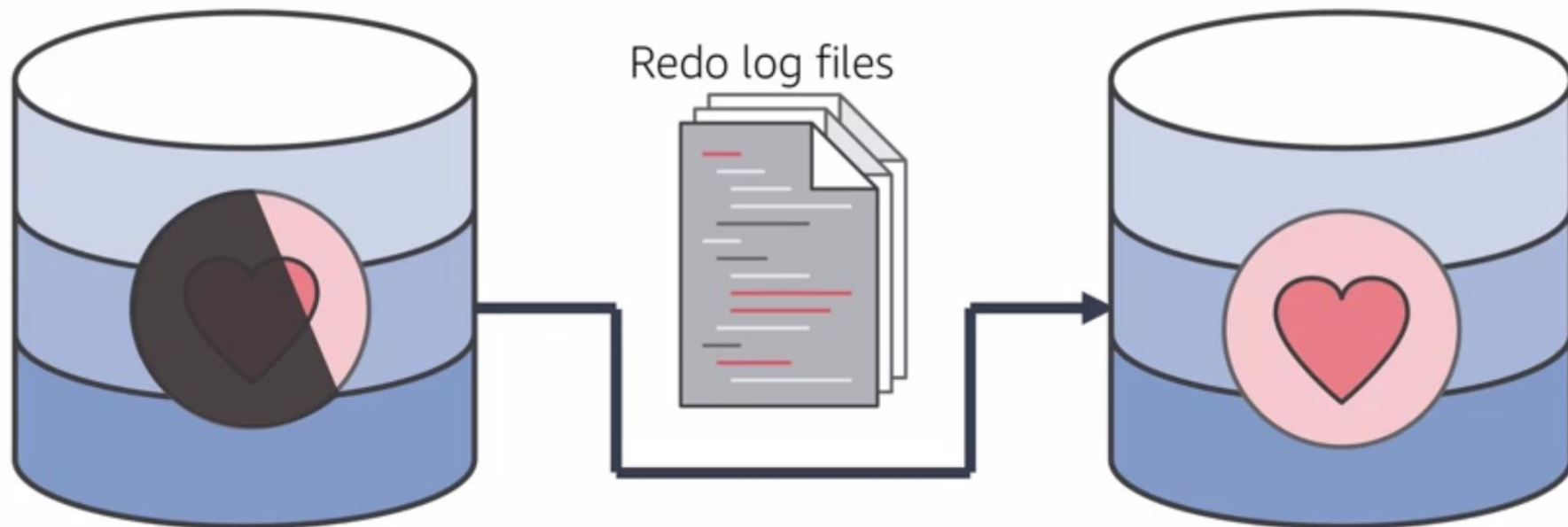
Amazon Aurora is designed to be **highly available**:

- It stores multiple copies of your data across multiple Availability Zones with continuous backups to Amazon S3.
- It can use up to 15 read replicas can be used to reduce the possibility of losing your data.
- It is designed for instant crash recovery if your primary database becomes unhealthy.



Resilient design

After a database crash, Amazon Aurora does not need to replay the redo log from the last database checkpoint. Instead, it performs this on **every read operation**. This reduces the restart time after a database crash to less **than 60 seconds** in most cases.



The right tool for the right job

What are my requirements?

Enterprise-class relational database

Amazon RDS

Fast and flexible NoSQL database service for any scale

Amazon DynamoDB

Operating system access or application features that are not supported by AWS database services

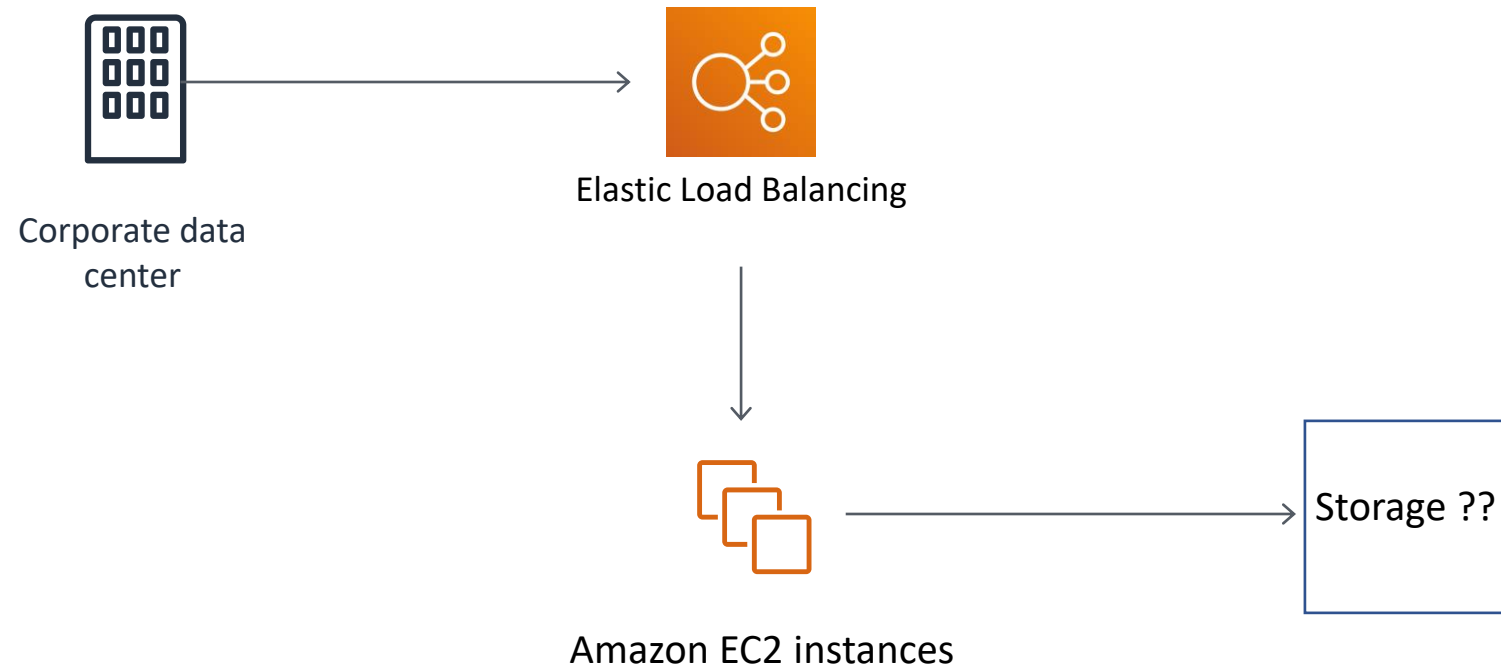
Databases on Amazon EC2

Specific case-driven requirements (machine learning, data warehouse, graphs)

AWS purpose-built database services

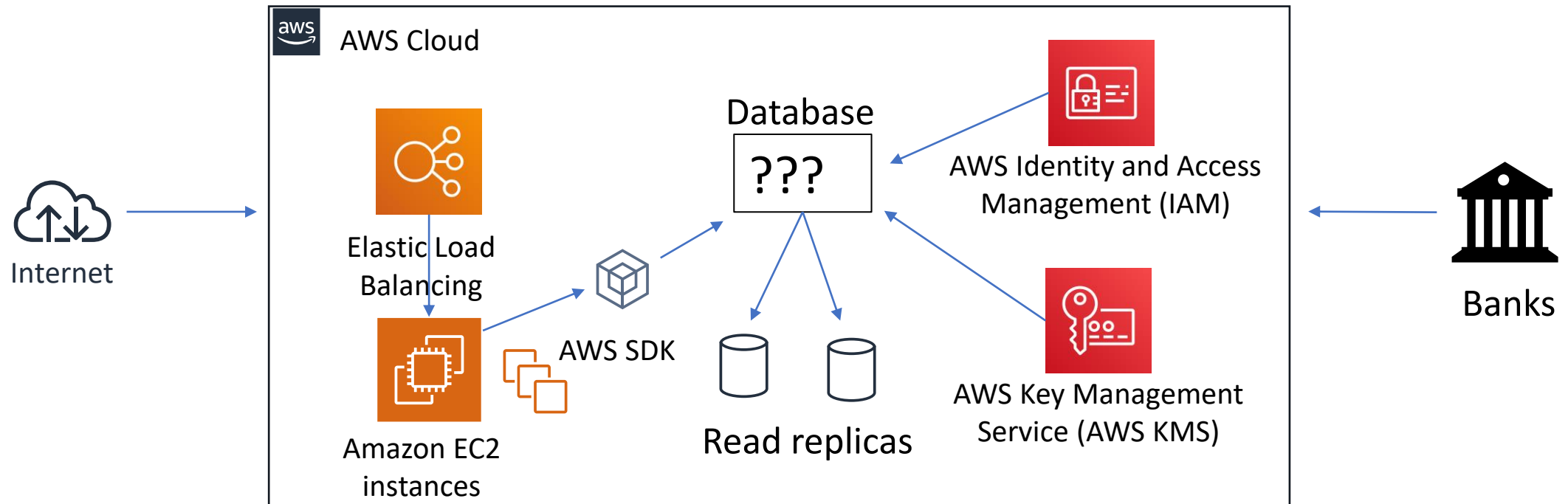
Storage case study activity

A collaboration software company processes email for enterprise customers. They have more than 250 enterprise customers and more than half a million users. They must store petabytes of data for their customers. The following diagram illustrates their architecture.



Database case study activity

An online payment processing company that processes over 1 million transactions per day. They must provide services to ecommerce customers who offer flash sales (sales that offer greatly reduced prices for a limited time), where demand can increase by 30 times in a short time period. They use IAM and AWS KMS to authenticate transactions with financial institutions. They need high throughput for these peak loads. The following diagram illustrates their architecture.



Thanks

