Cloud Infrastructure Week#2

Emrah Mutlu

January 2024

AGENDA – WEEK#2

- Data, Data Types, Databases
- Cloud Storage Services
- Cloud Database Services
- Caching
- Data Redundancy & Replication
- Database Migration to Cloud
- Data Security
- Lab Sessions:
  - Lab: Testing replication & disaster recovery scenarios on Cloud
  - Lab: Creating an SQL DB on Cloud
  - Lab: Creating a File Share on Cloud

Orion Innovation

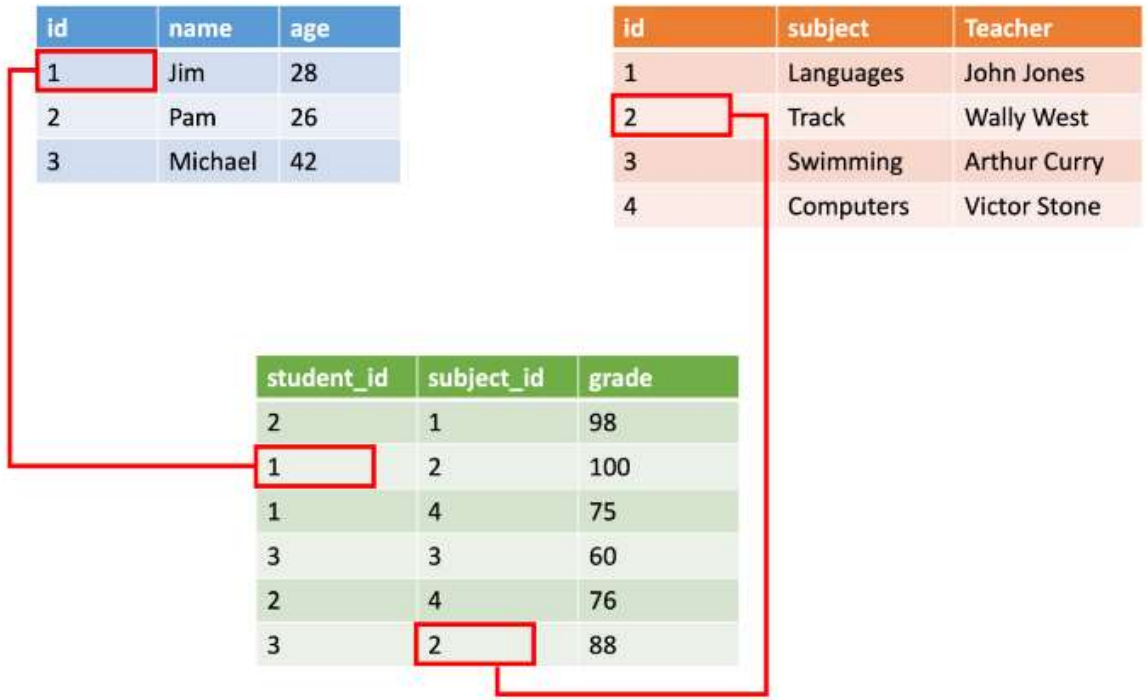
Data, Data Types & Databases

Data: In the pursuit of knowledge, ***data*** is a collection of discrete values that convey information, describing quantity, quality, fact, statistics, other basic units of meaning, or simply sequences of symbols that may be further interpreted.

Examples of data types: Media files (videos, photos, etc.), business data (mail, product catalogs, etc.), financial data, sensor data, etc.

Classification of Data Types:

1. Structured Data: Structured data, sometimes referred to as *relational data*, is data that adheres to a strict schema, so all of the data has the same fields or properties. The shared schema allows this type of data to be easily searched with query languages such as ***SQL (Structured Query Language).***



id	name	age
1	Jim	28
2	Pam	26
3	Michael	42

id	subject	Teacher
1	Languages	John Jones
2	Track	Wally West
3	Swimming	Arthur Curry
4	Computers	Victor Stone

student_id	subject_id	grade
2	1	98
1	2	100
1	4	75
3	3	60
2	4	76
3	2	88

2. Semi-Structured Data: Semi-structured data is less organized than structured data, and is not stored in a relational format, as the fields do not neatly fit into tables, rows, and columns. Semi-structured data contains tags that make the organization and hierarchy of the data apparent - for example, key/value pairs. Semi-structured data is also referred to as non-relational or NoSQL data. The expression and structure of the data in this style is defined by a serialization language (XML, YAML, JSON).

```xml
<Person Age="23">
    <FirstName>John</FirstName>
    <LastName>Smith</LastName>
    <Hobbies>
        <Hobby Type="Sports">Golf</Hobby>
        <Hobby Type="Leisure">Reading</Hobby>
        <Hobby Type="Leisure">Guitar</Hobby>
   </Hobbies>
</Person>
```

XML

```json
{
    "firstName": "John",
    "lastName": "Doe",
    "age": "23",
    "hobbies": [
        { "type": "Sports", "value": "Golf" },
        { "type": "Leisure", "value": "Reading" },
        { "type": "Leisure", "value": "Guitar" }
    ]
}
```

JSON

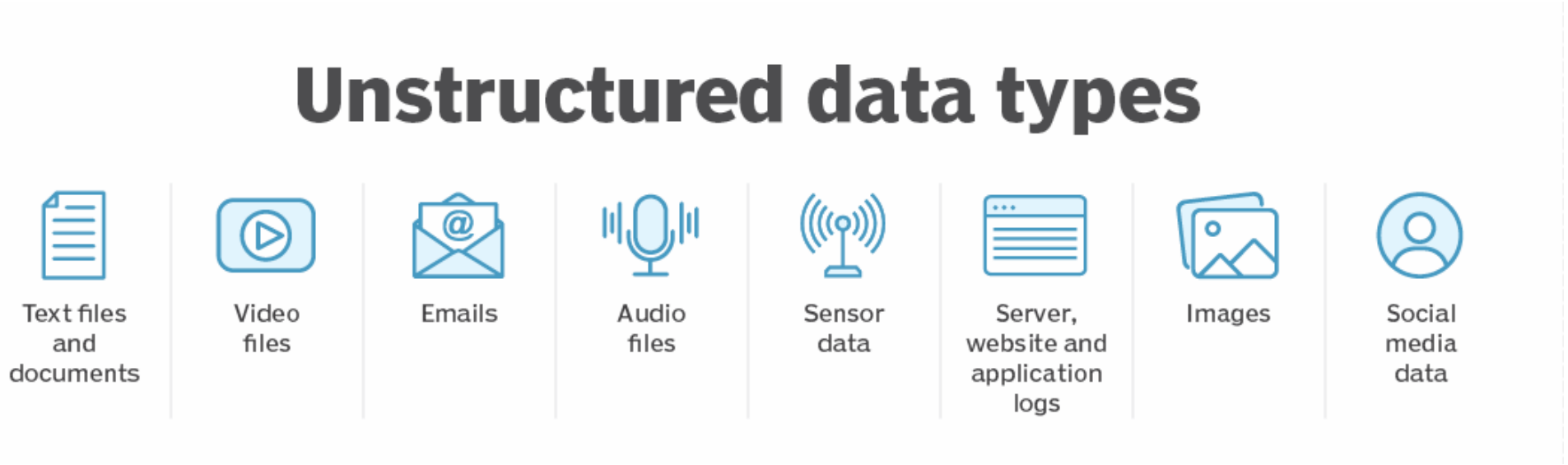
```yaml
firstName: John
lastName: Doe
age: 23
hobbies:
    - type: Sports
      value: Golf
    - type: Leisure
      value: Reading
    - type: Leisure
      value: Guitar
```

YAML

3. Unstructured Data: The organization of unstructured data is ambiguous. Unstructured data is often delivered in files, such as photos or videos.

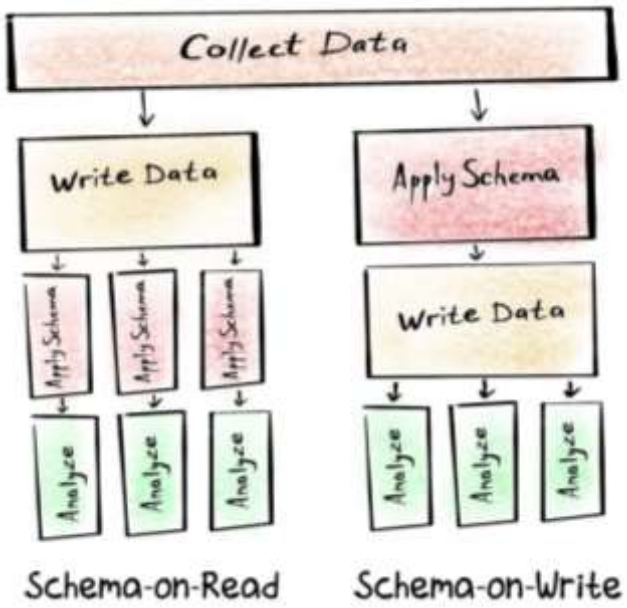
Examples of unstructured data include;

- Media files, such as photos, videos, and audio files
- Office files, such as Word documents
- Text files
- Log files



DIFFERENCES BETWEEN STRUCTURED AND UNSTRUCTURED DATA

	Structured data	Unstructured data
ANALYSIS	Quantitative	Qualitative
SCHEMA CREATION	Schema-on-write	Schema-on-read
SEARCHING	Easy using SQL-based methods	May need special tools
FORMAT	Predefined, using alphanumeric characters	Typically non-character-oriented digital representations
STORAGE	May require more storage to accommodate defined data structures	Some forms require less storage; others have large file formats, requiring more storage
STORAGE FORMAT	Relational database management systems, data warehouses	Applications, NoSQL databases, data lakes



- Define the types of the following data types:
  - **Product catalog data**
  - Product catalog data contains the following;
    - Product SKU
    - Description
    - Quantity
    - Price
    - Size
    - Color
    - Photo / Video, etc.
  - However, when a new feature is defined (like a Bluetooth featured shoes), this new feature will require a new schema. As a result, the data will not be homogenous (since adding this new feature to all existing shoes will be a cost).
  - **Answer: Semi-Structured Data**

Q&A

- Define the types of the following data types:
  - **Business data**
  - Business data may contain the following types:
    - Sales data
    - KPI (Key Performance Indicator) data
    - Finance data
    - Should be comparable with previous months
    - Should be visualized via BI (Business Intelligence) tools like PowerBI
    - Should be evaluated through the pipelines
  - **Answer: Structured Data**

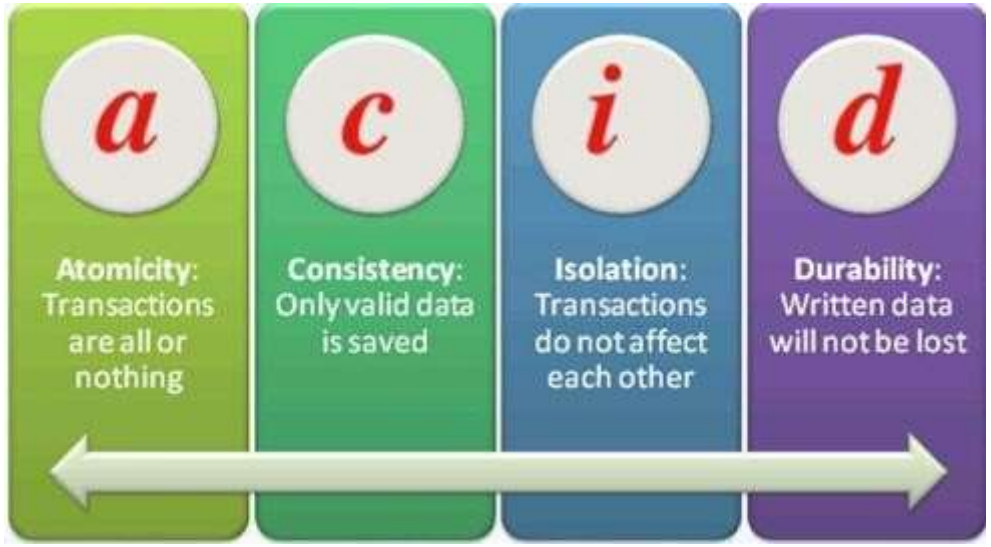
Q&A

- Define the types of the following data types:
  - **Industry 4.0 data**
  - Industry 4.0 data may contain the following types:
    - Data coming from various types of sensors (IoT)
    - Processed data from Edge devices
    - Unprocessed data from local resources
    - ERP (Enterprise Resource Planning) data from ERP systems
    - HMI (Human Machine Interface) data from HMI devices
    - AR/VR data from planning systems
  - As a result, a big data system that contains both structured & unstructured data
  - **Answer: Unstructured Data**

TRANSACTION

Transaction: A transaction is a logical group of database operations that execute together.

- If a change in a piece of data will impact another, then Transaction is required.
- Transactions are often defined by a set of four requirements, referred to as ACID guarantees. ACID stands for **A**tomicity, **C**onsistency, **I**solation, and **D**urability:
  - **Atomicity** means a transaction must execute exactly once and must be atomic; either all of the work is done, or none of it is. Operations within a transaction usually share a common intent and are interdependent.
  - **Consistency** ensures that the data is consistent both before and after the transaction.
  - **Isolation** ensures that one transaction is not impacted by another transaction.
  - **Durability** means that the changes made due to the transaction are permanently saved in the system. Committed data is saved by the system so that even in the event of a failure and system restart, the data is available in its correct state.
- When a database offers ACID guarantees, these principles are applied to any transactions in a consistent manner.



OLTP & OLAP

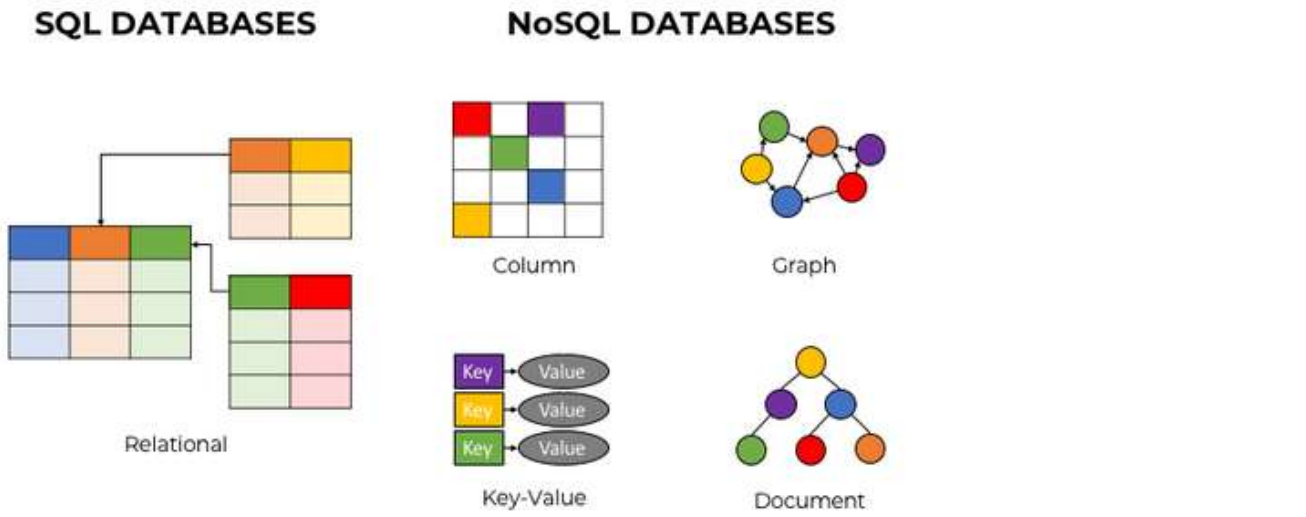
- **Transactional databases** are generally called **OLTP** (Online Transaction Processing) systems.
- **OLTP** systems;
  - Supports lots of users
  - Have quick response times
  - Handle large volumes of data
  - Has HA (Highly available) structure
  - Typically handle small or relatively simple transactions.
- **OLAP** (Online Analytical Processing) systems;
  - Commonly support fewer users
  - Have longer response times
  - Can be less available
  - Typically handle large and complex transactions.

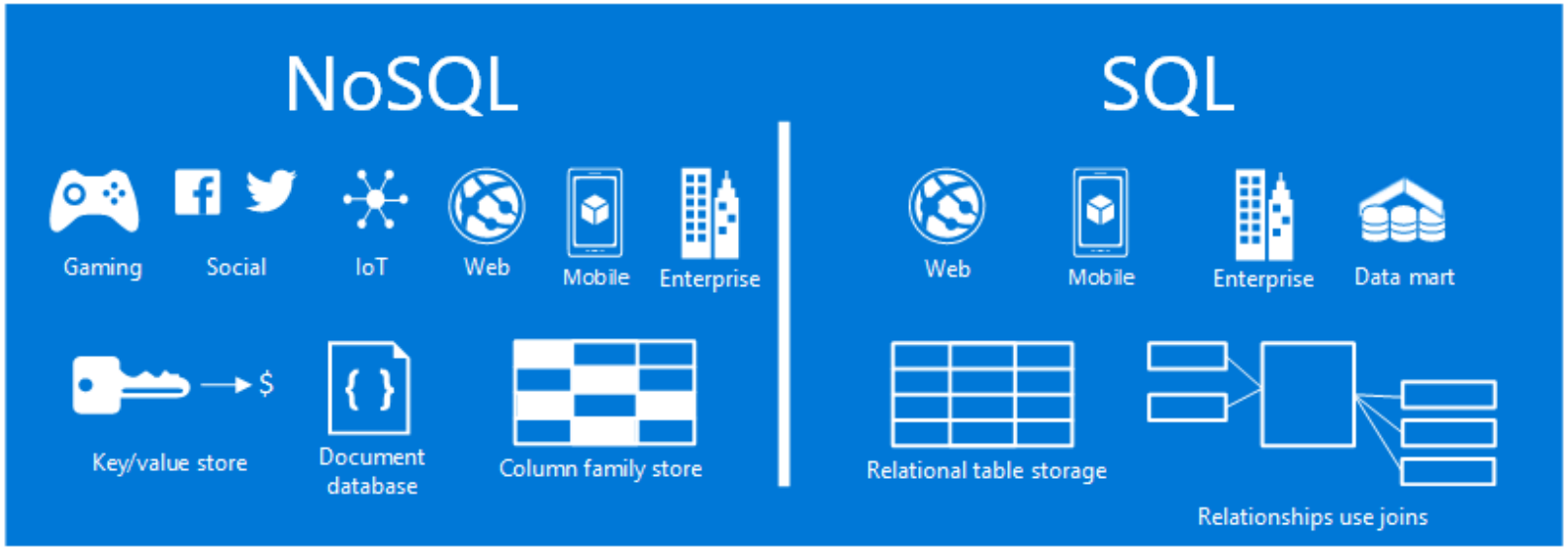
Q&A

- **Define the types of the following needs (OLTP or OLAP):**
  - Product Catalog Data
  - Photos & Videos
  - Business Data

- Product Catalog Data: Transaction is a must especially on payments, inventory updates, etc. → OLTP
- Photos & Videos: These files are changed when a change in the file itself, no transaction requirement.
- Business Data: All the data is historical and not changing, so no transaction requirement. → OLAP

SQL VS NOSQL





SQL VS NOSQL

SQL	NoSQL
RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)	Non-relational or distributed database system.
These databases have fixed or static or predefined schema	They have a dynamic schema
These databases are not suited for hierarchical data storage.	These databases are best suited for hierarchical data storage.
These databases are best suited for complex queries	These databases are not so good for complex queries
Vertically Scalable	Horizontally scalable
Follows ACID property	Follows CAP (consistency, availability, partition tolerance)
Examples: MySQL, PostgreSQL, Oracle, MS-SQL Server, etc	**Examples:** MongoDB, HBase, Neo4j, Cassandra, etc

Orion Innovation

Cloud Database Services

CLOUD DATABASE SERVICES (AZURE)

IF YOU WANT TO	USE THIS
Build modern cloud applications with an always up-to-date relational database service that includes serverless compute, hyperscale storage, and AI-powered and automated features to optimize performance and durability	Azure SQL Database
Migrate your SQL workloads to Azure while maintaining complete SQL Server compatibility, with all the benefits of a fully managed and evergreen PaaS	Azure SQL Managed Instance
Migrate your SQL workloads to Azure while maintaining complete SQL Server compatibility and operating system-level access	SQL Server on Virtual Machines
Build scalable, secure, and fully managed enterprise-ready apps on open-source PostgreSQL, scale out single-node PostgreSQL with high performance, or migrate PostgreSQL and Oracle workloads to the cloud	Azure Database for PostgreSQL
Deliver high availability and elastic scaling to open-source mobile and web apps with a managed community MySQL database service, or migrate MySQL workloads to the cloud. Supports LAMP stack (Linux, Apache, MySQL, PHP).	Azure Database for MySQL
Deliver high availability and elastic scaling to open-source mobile and web apps with a managed community MariaDB database service	Azure Database for MariaDB
Build applications with guaranteed low latency and high availability anywhere, at any scale, or migrate Cassandra, MongoDB, and other NoSQL workloads to the cloud	Azure Cosmos DB
Power fast, scalable applications with an open-source-compatible in-memory data store	Azure Cache for Redis
Accelerate your transition to the cloud using a simple, self-guided migration process	Azure Database Migration Service

COMPARISION OF DATABASE SERVICES

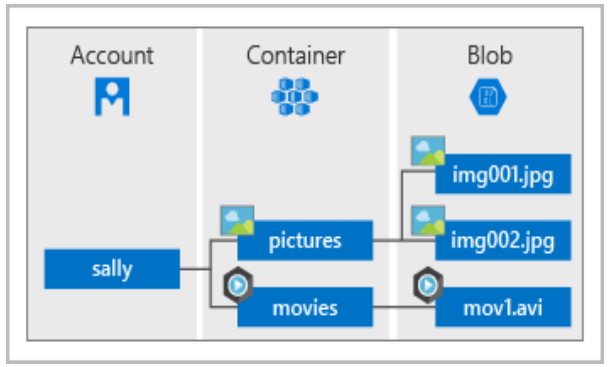
	Azure SQL Database	Azure SQL Managed Instance	SQL Server on Virtual Machines	Azure Database for PostgreSQL	Azure Database for MySQL	Azure Database for MariaDB	Azure Cosmos DB	Azure Cache for Redis
Relational Database	✓	✓	✓	✓	✓	✓		
Non-Relational Database (NoSQL)							✓	
In-Memory Database								✓
Data Models	Relational	Relational	Relational	Relational	Relational	Relational	Multi-Model: Document Wide-column Key-Value Graph	Key-Value
Hybrid	✓	✓	✓	✓ *(Hyperscale)*				
Serverless Compute	✓						✓	
Storage Scale Out	✓ *(Hyperscale)*			✓ *(Hyperscale)*			✓	✓
Compute Scale Out	✓ *(Hyperscale read only)*			✓ *(Hyperscale)*			✓	✓
Distributed Multi-Master Writes (Write data to different regions)							✓	✓ *(Coming Soon)*
OSS Based Service (Community edition and open extension support)				✓	✓	✓		✓
HTAP (Available with Azure Synapse Link)	✓ *(Coming Soon)*			✓ *(Coming Soon)*			✓	

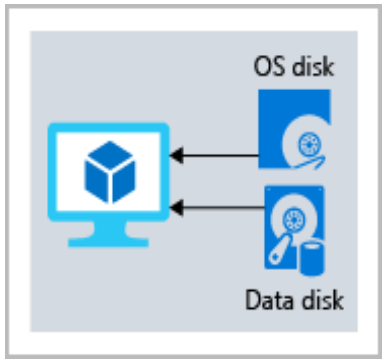
Orion Innovation

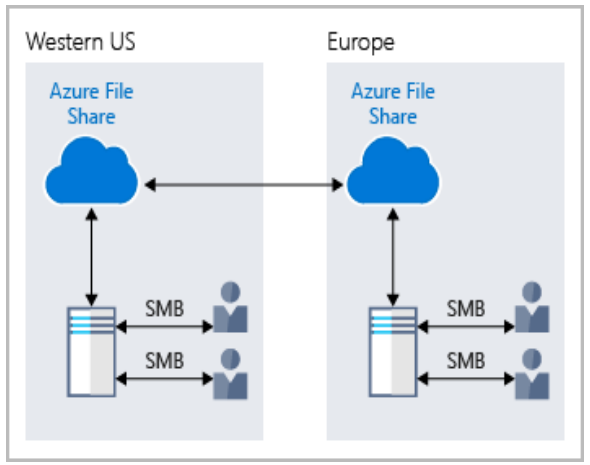
Cloud Storage Services

CLOUD STORAGE SERVICES (AZURE)

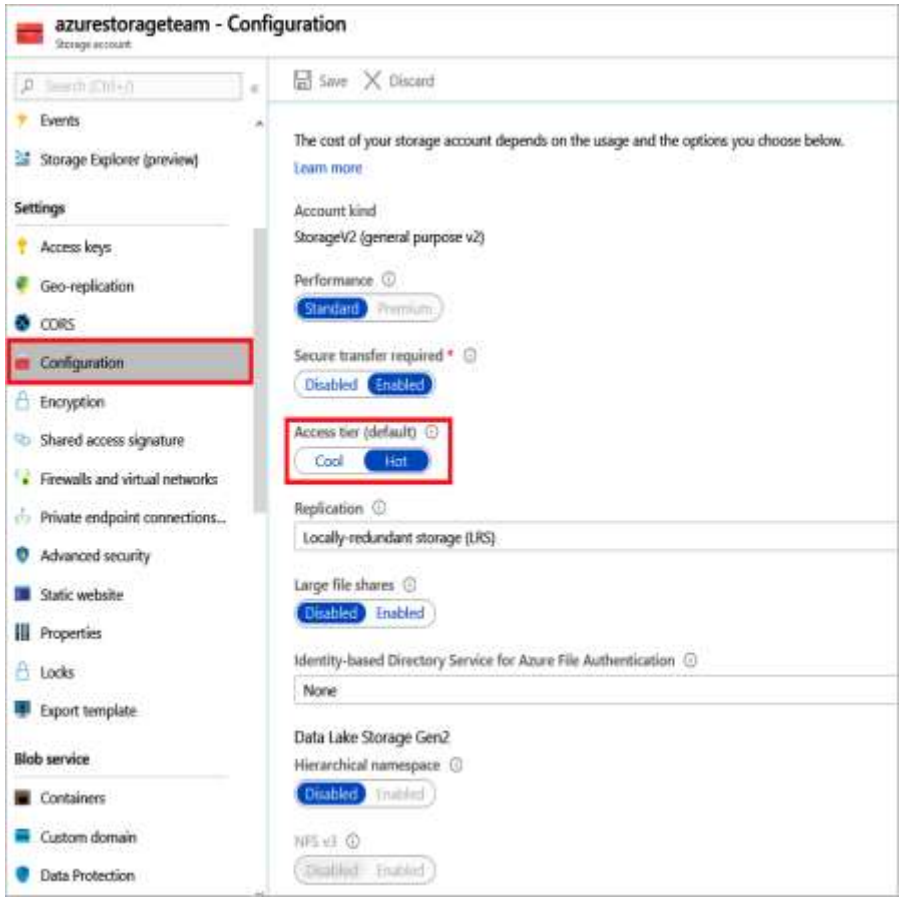
Service name	Service function
Azure Blob storage	Storage service for very large objects, such as video files or bitmaps (Containers).
Azure File storage	File shares that can be accessed and managed like a file server.
Azure Queue storage	A data store for queuing and reliably delivering messages between applications.
Azure Table storage	A NoSQL store that hosts unstructured data independent of any schema.







* Azure VMs use Azure Disk Storage to store virtual disks. However, you can't use Azure Disk Storage to store a disk outside of a virtual machine.



Azure Storage offers different access tiers for your blob storage, helping you store object data in the most cost-effective manner. The available access tiers include:

Hot access tier: Optimized for storing data that is accessed frequently (for example, images for your website).

Cool access tier: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).

Archive access tier: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups).

Q&A

- Based on the DB types & storage services, determine the best DB / Storage to these services:
  - Product Catalog Data
  - Photos & Videos
  - Business Data

Tip: Define the data classification, operation volume, latency, throughput and transaction needs before answering the question.

You have 5 minutes ☺

Q&A

- **Product Catalog Data**
  - Data Classification: Semi-structured (changing the products, etc.)
  - Operations: Customers require high number of read operations, business need high number of write operations
  - Latency & Throughput: High throughput & low latency
  - Transaction Support: Required

- **Recommended Solution:** Azure Cosmos DB
  - Why?
    - NoSQL DB requirement
    - Transaction support / ACID compliant
    - Replication support for operation
    - Supports several consistency levels for the optimized throughput & latency
  - Why not other solutions?

- **Photos & Videos**
  - Data Classification: Unstructured
  - Operations: Need to be retrieved by ID, high number of read by customers with low latency, create/modify operations will be infrequent, could tolerate high latency.
  - Latency & Throughput: Retrievals by ID need to support low latency and high throughput. Creates and updates can have higher latency than read operations.
  - Transaction Support: Not required

- **Recommended Solution:** Azure Blob Storage
  - Why?
    - Best place to home photos & videos
    - Supports Azure CDN (Content Delivery Network) for Edge servers to store frequently reached files
    - Tiered approach (Hot, Cool, Archive) for better pricing / performance
  - Why not other solutions?

Q&A

- **Business Data**
  - Data Classification: Structured (stable data, not changing)
  - Operations: Read-only, complex analytical queries across multiple databases
  - Latency & Throughput: Some latency in the results is expected based on the complex nature of the queries.
  - Transaction Support: Not required

- **Recommended Solution:** Azure SQL Database
  - Why?
    - SQL DB requirement
    - Transaction support / ACID compliant
    - Works better with BI tools (PowerBI, etc.)
    - Cross-database query support for data analysis
    - SQL Query Support for data analysis
  - Why not other solutions? (Azure Synapse, Azure Analysis Services, Azure Stream Analytics)

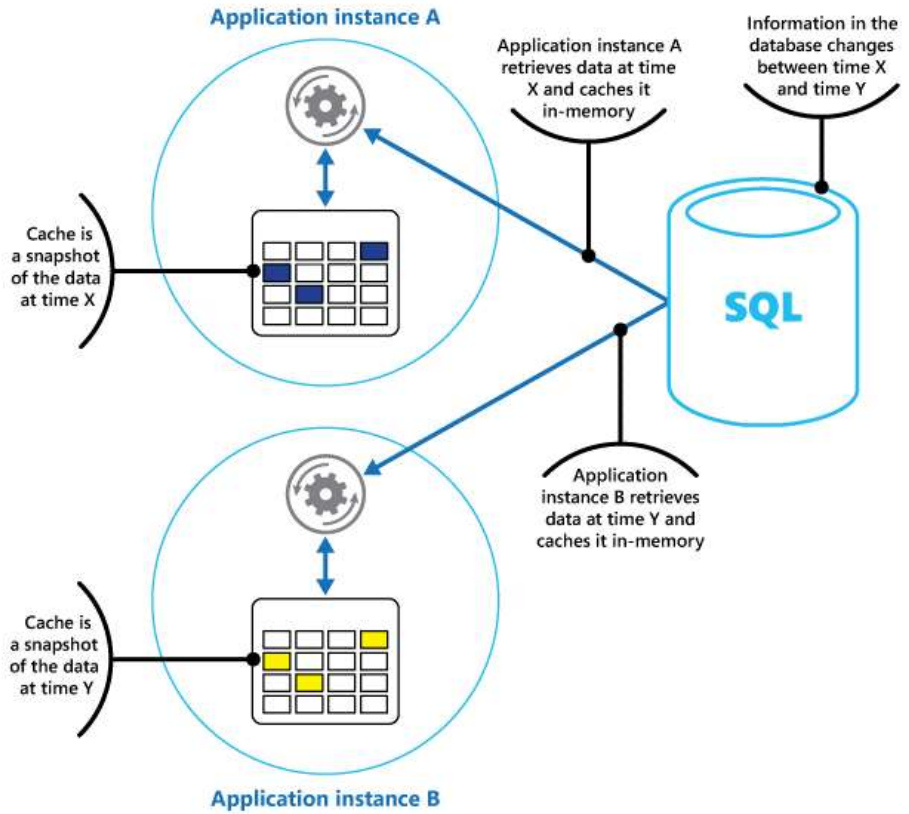
Caching

CACHING

- **Caching:** Caching is a common technique that aims to improve the performance and scalability of a system. It caches data by temporarily copying frequently accessed data to fast storage that's located close to the application. If this fast data storage is located closer to the application than the original source, then caching can significantly improve response times for client applications by serving data more quickly.
- Caching is most effective when a client instance repeatedly reads the same data, especially if all the following conditions apply to the original data store:
  - It remains relatively static.
  - It's slow compared to the speed of the cache.
  - It's subject to a high level of contention.
  - It's far away when network latency can cause access to be slow.
- **Caching in distributed applications**

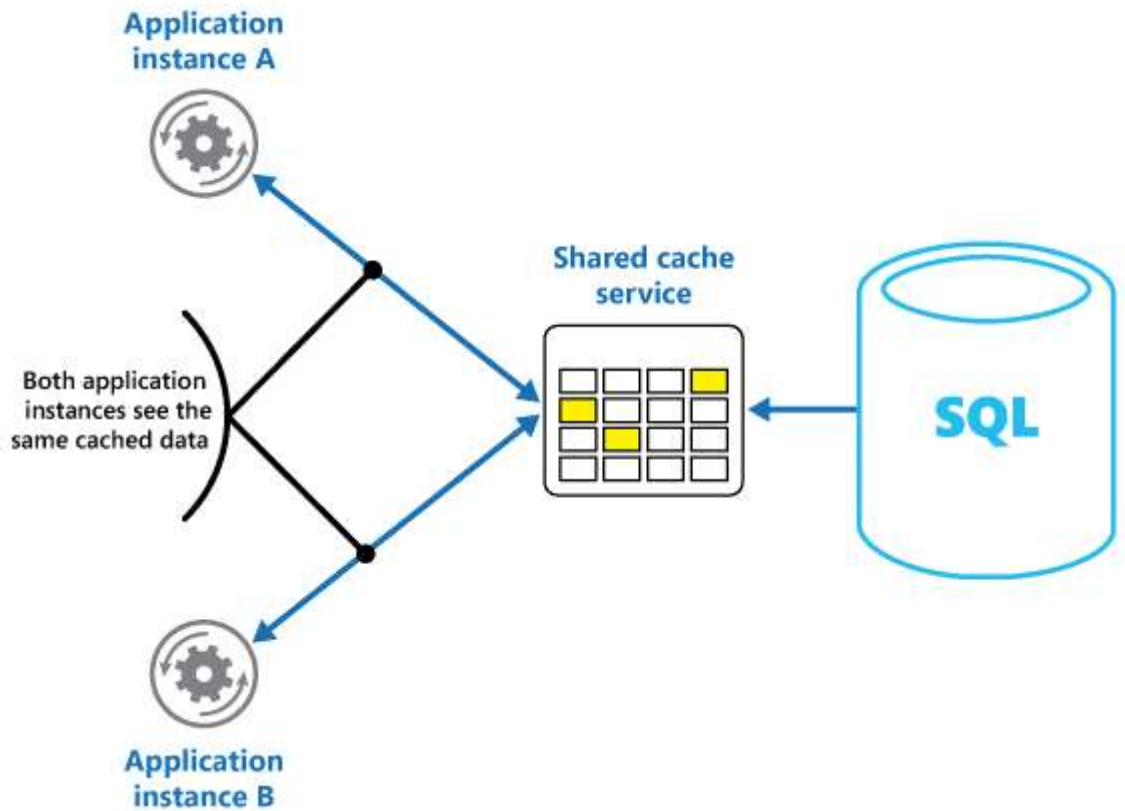
  Distributed applications typically implement either or both of the following strategies when caching data:
  - They use a private cache, where data is held locally on the computer that's running an instance of an application or service.
  - They use a shared cache, serving as a common source that can be accessed by multiple processes and machines.
  - In both cases, caching can be performed client-side and server-side. Client-side caching is done by the process that provides the user interface for a system, such as a web browser or desktop application. Server-side caching is done by the process that provides the business services that are running remotely.

- **Private Caching:** The most basic type of cache is an in-memory store. It's held in the address space of a single process and accessed directly by the code that runs in that process. This type of cache is quick to access. It can also provide an effective means for storing modest amounts of static data. The size of a cache is typically constrained by the amount of memory available on the machine that hosts the process.



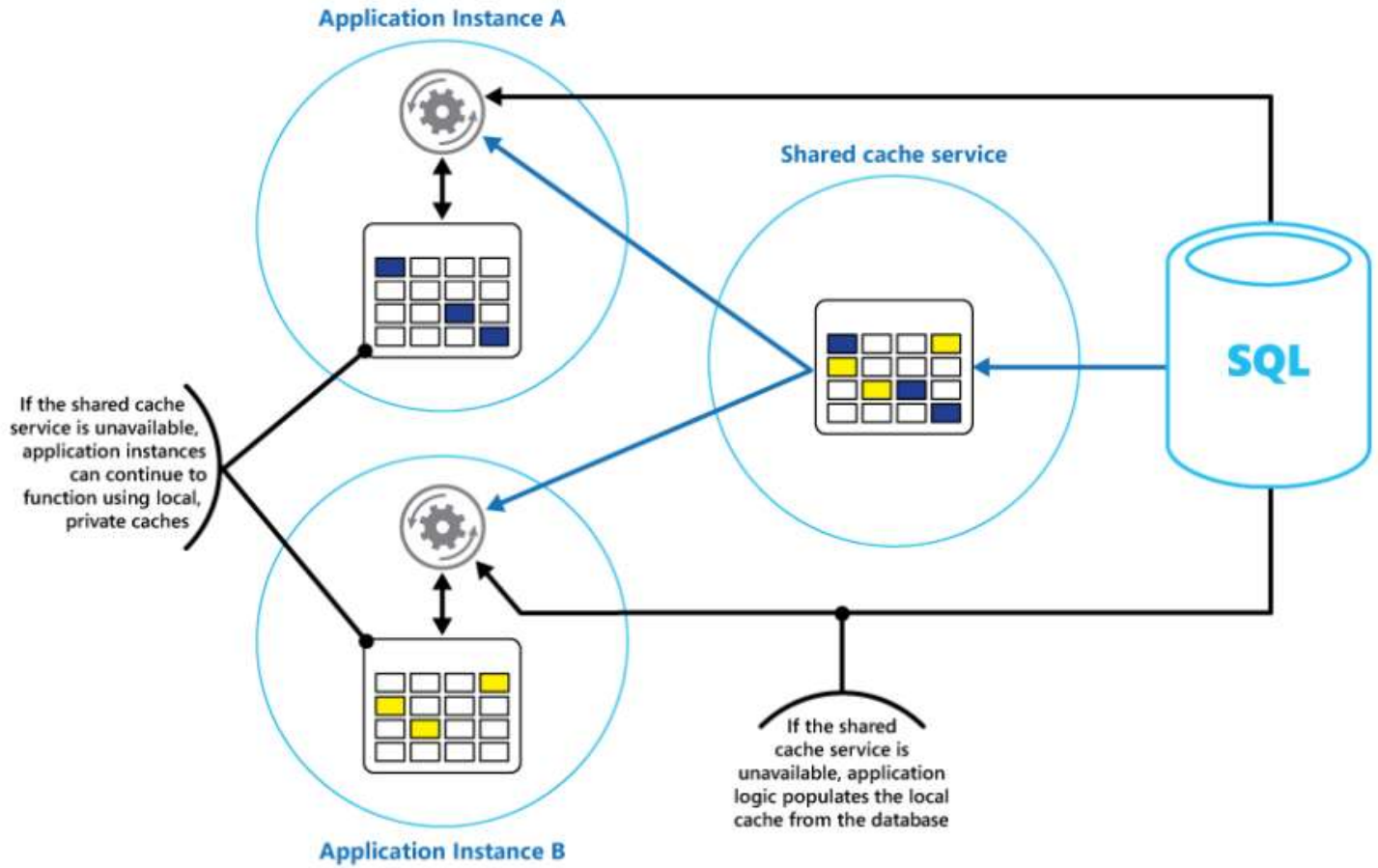
Private Caching

- **Shared Caching:** If you use a shared cache, it can help alleviate concerns that data might differ in each cache, which can occur with in-memory caching. Shared caching ensures that different application instances see the same view of cached data. It locates the cache in a separate location, which is typically hosted as part of a separate service.



Shared Caching

- **Mixed Caching:** A combination of both Private and Shared caching for high available / BC App. scenario.



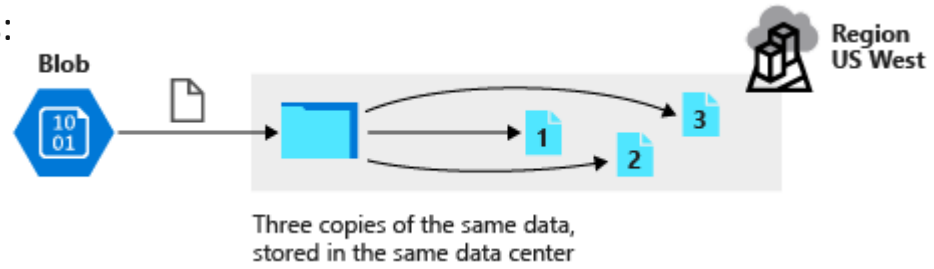
Mixed Caching

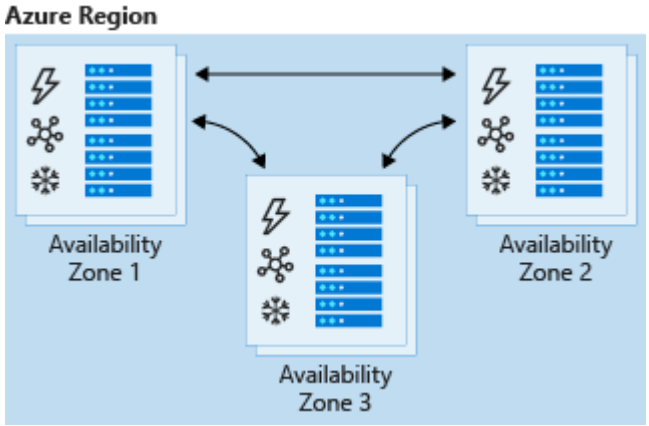
Orion Innovation

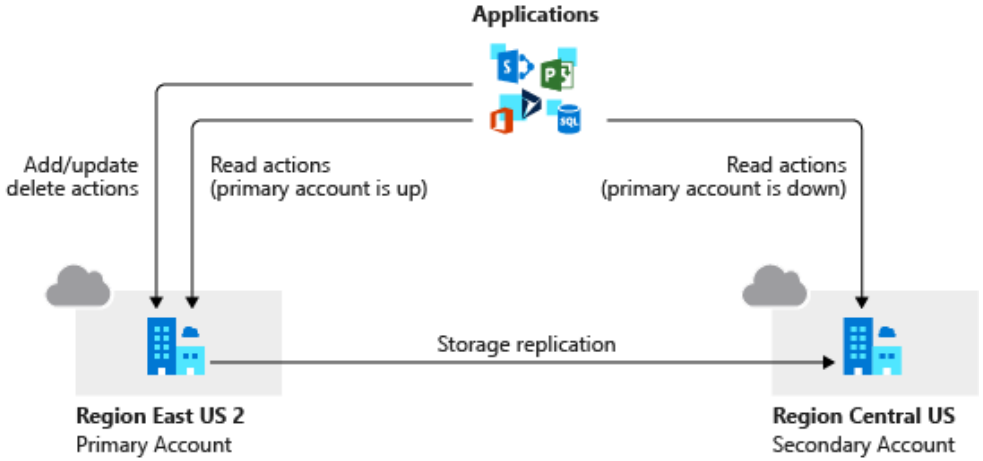
Data Redundancy & Replication

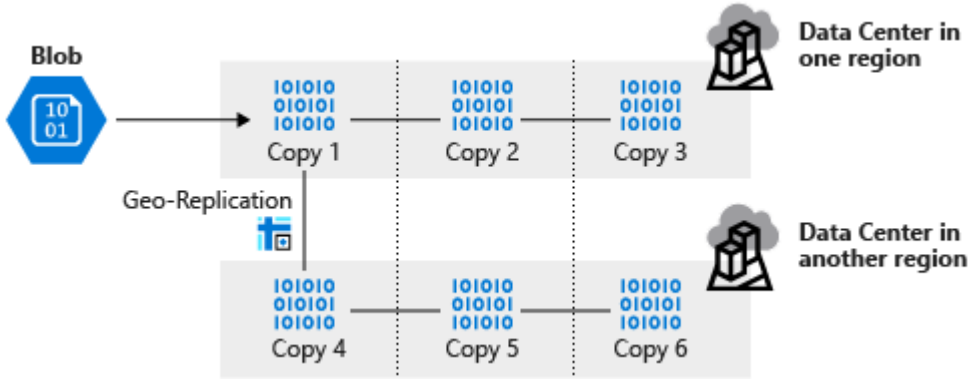
DATA REDUNDANCY

- **Data redundancy:** Data in Azure is replicated to ensure that it's always available, even if a datacenter or region becomes inaccessible or a specific piece of hardware fails. You have the following replication options:
  - Locally redundant storage (LRS)
  - Zone-redundant storage (ZRS)
  - Geographically redundant storage (GRS)
  - Read-access geo-redundant storage (RA-GRS)
  - Geo-zone-redundant storage (GZRS)
  - Read-access geo-zone-redundant storage (RA-GZRS)
- Each replication option provides a different level of redundancy and durability.





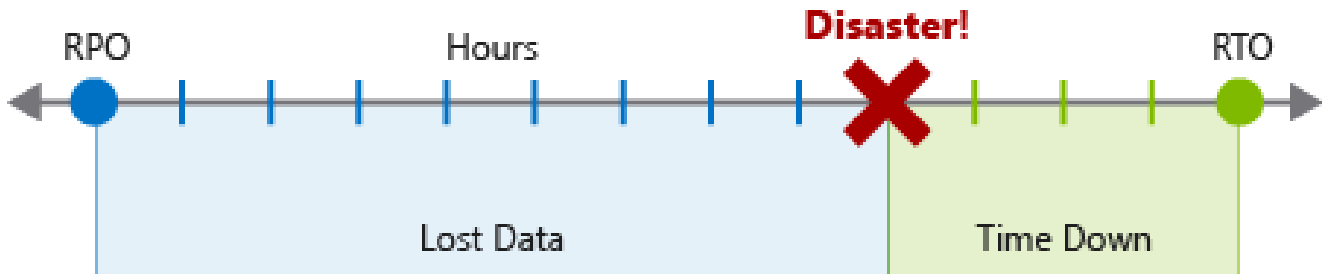




DATA REDUNDANCY COMPARISION

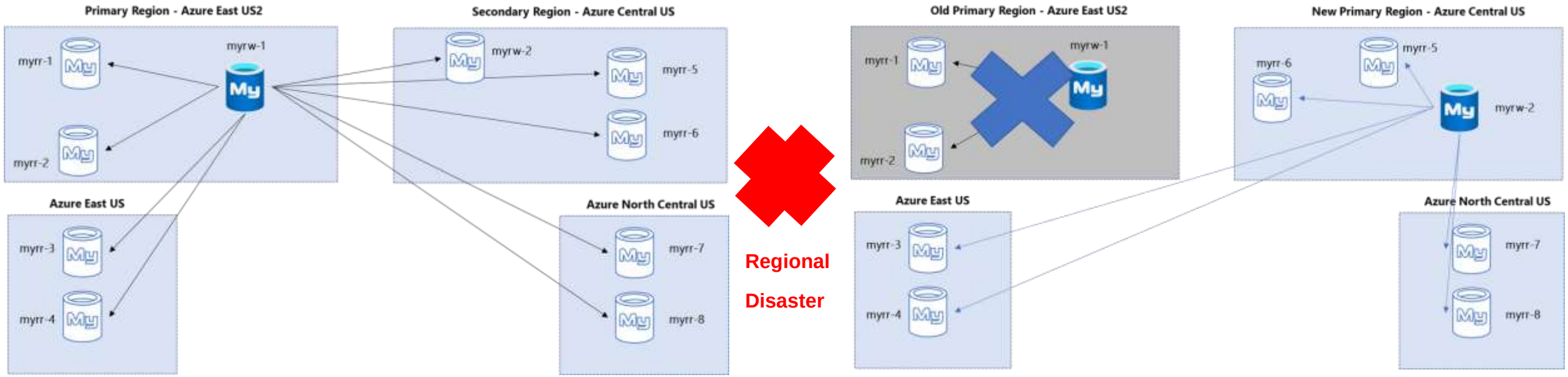
Replication type	Copies	Use case
LRS	3	Data remains highly available, but for compliance reasons, isn't allowed to leave the local datacenter.
ZRS	3	Need redundancy in multiple physical locations, but because of compliance, data isn't allowed to leave a region.
GRS	6	App has access to the data, even if an entire region has an outage.
RA-GRS	6	App reads from multiple geographical locations, so you can serve users from a location that's closer to them.
GZRS	6	App can access data, even if the primary region has failed, and your secondary region has a datacenter that's experiencing an outage. But you don't want to read from the secondary region unless the primary region is down.
RA-GZRS	6	Regularly read data from your secondary region, perhaps to serve users from a location closer to them, even if a datacenter is up in your primary region.

- **Replication:** Replication is used for business continuity and disaster recovery scenarios. Each DB has a different replication technique in Azure.
- **RTO** (Recovery Time Objective) and **RPO** (Recovery Point Objective) are the main criteria while designing replication:
  - **RTO:** An RTO is a measure of the maximum amount of time your business can survive after a disaster before normal service is restored. Let's assume your RTO is 12 hours, which means that operations can continue for 12 hours without the business's core services functioning. If the downtime is 24 hours, your business would be seriously harmed.

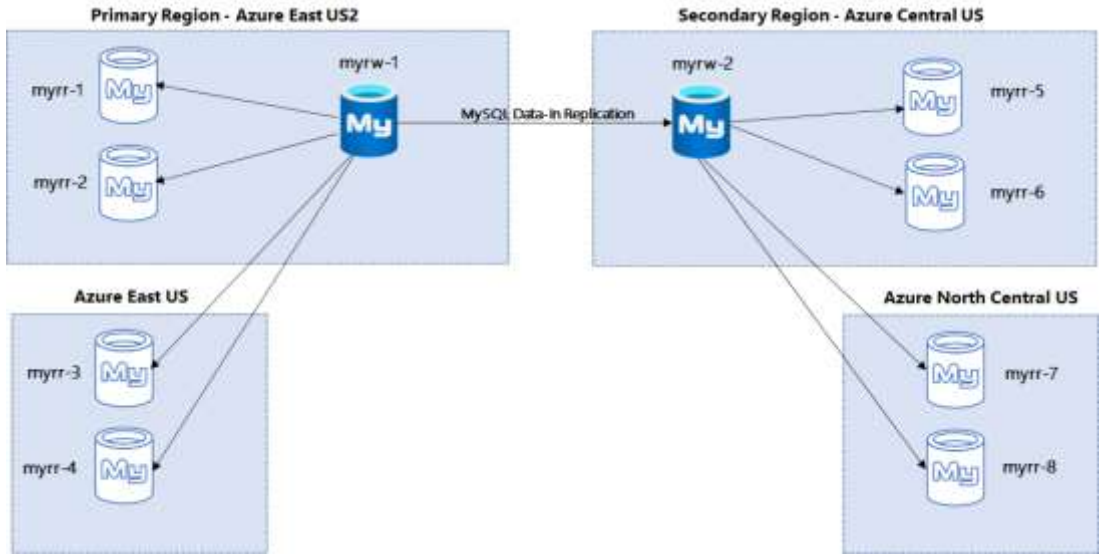


  - **RPO:** An RPO is a measure of the maximum amount of data loss that's acceptable during a disaster. A business can typically decide to do a backup every 24 hours, 12 hours, or even in real time. If a disaster occurs, there's always some data loss.

DB REPLICATION EXAMPLE: AZURE DB FOR MYSQL



Alternate Replication



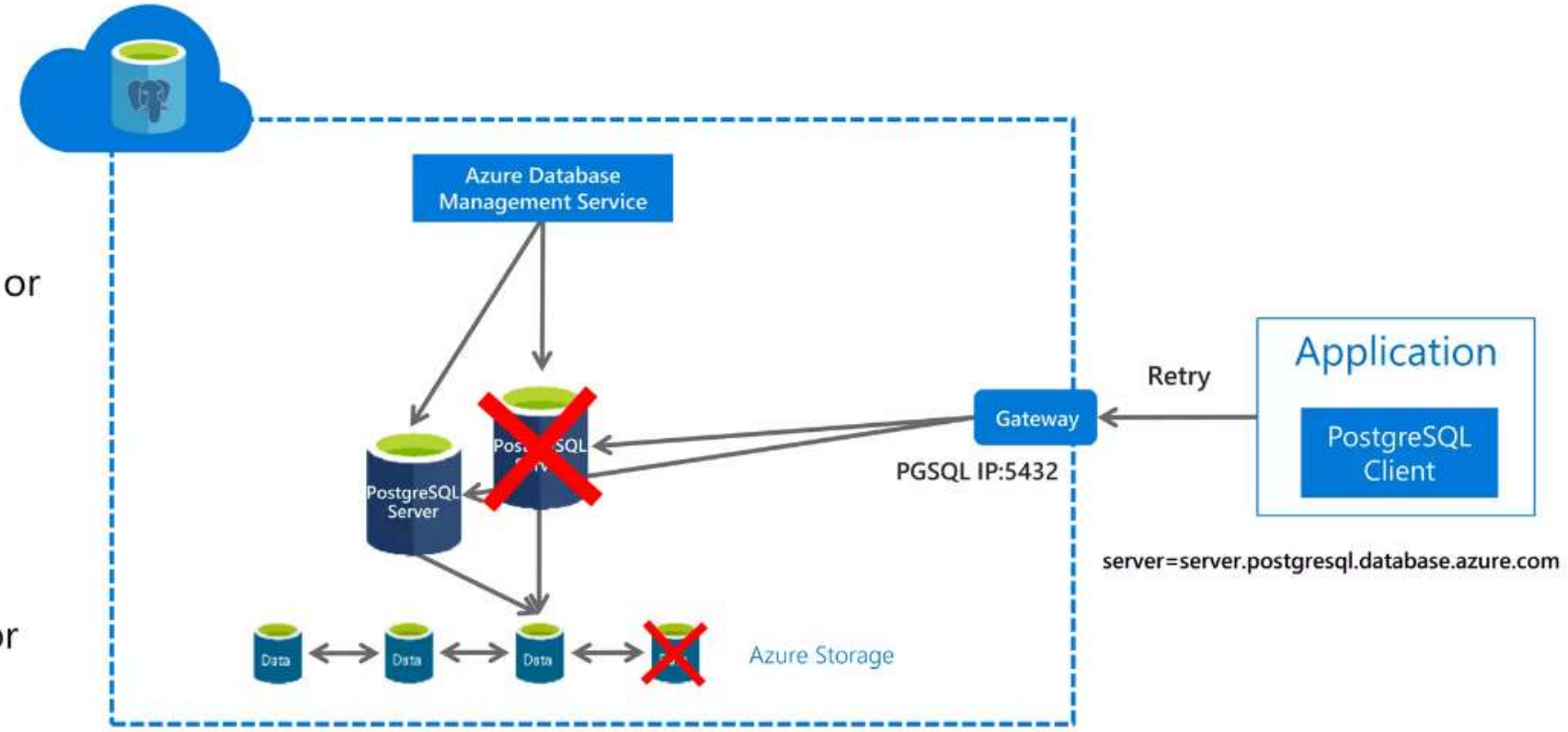
DB REPLICATION EXAMPLE: POSTGRESQL

- PostgreSQL DB uses Built-in HA structure that Azure Cloud could control the replication automatically.

Scale compute up or down in seconds

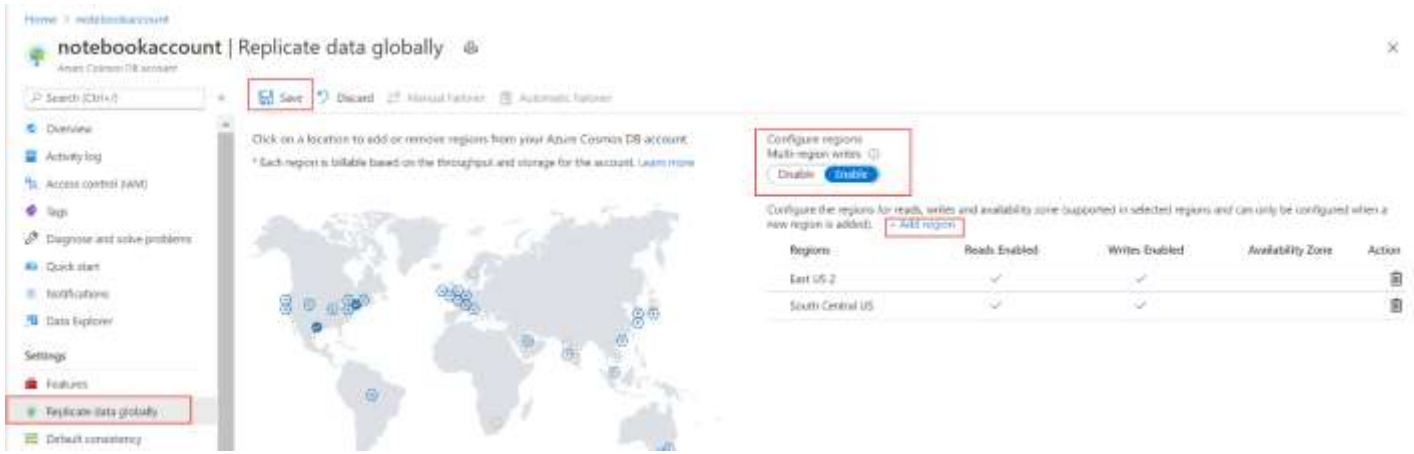
Scale storage instantaneously

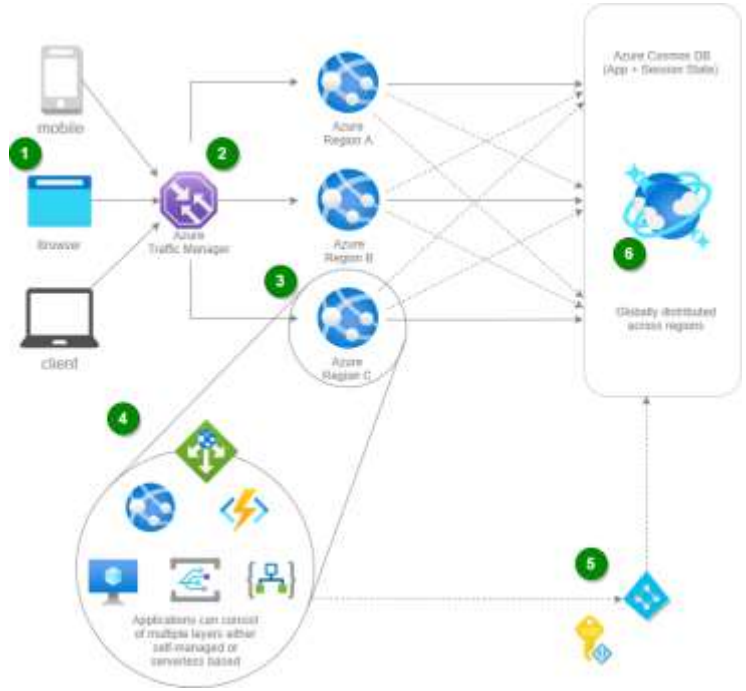
3 copies of data for data reliability



DB REPLICATION EXAMPLE: AZURE COSMOS DB

- Azure Cosmos DB uses Built-in multi-region write copies that Azure Cloud could control the replication automatically.



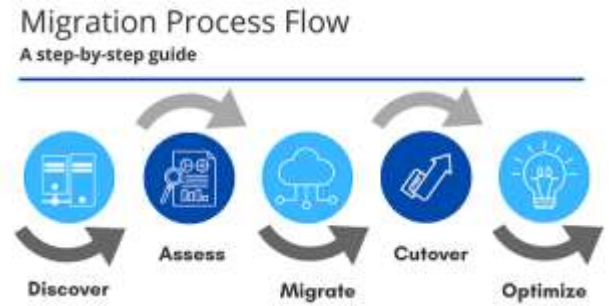


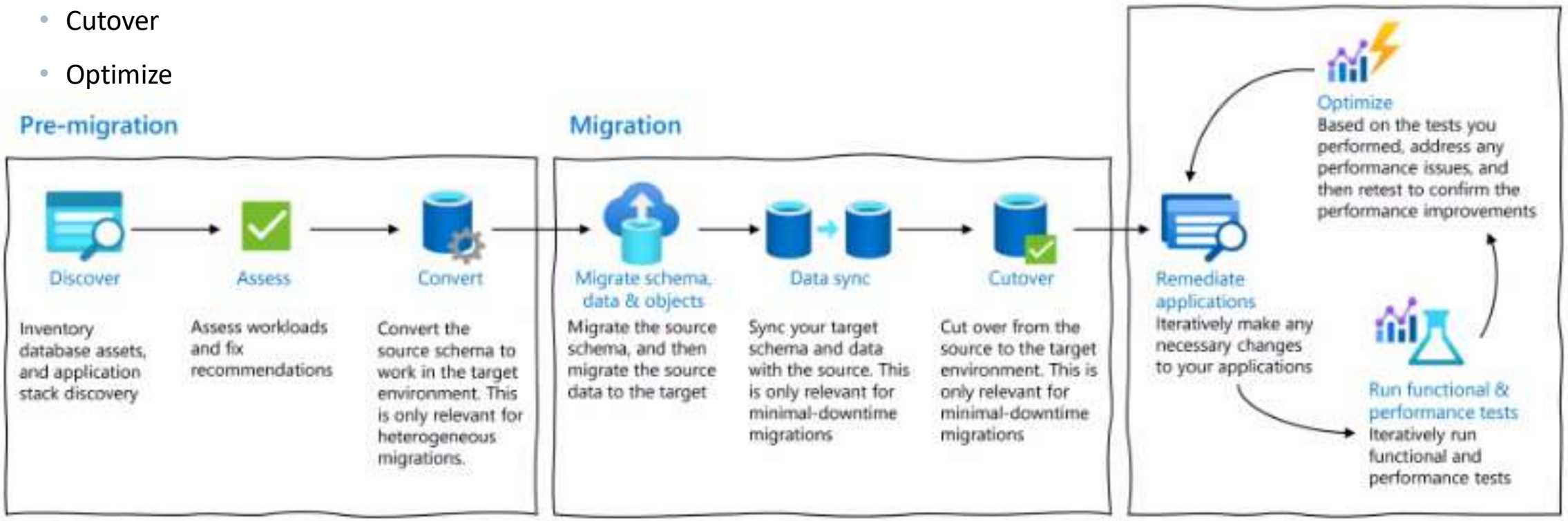
Orion Innovation

DB Migration to Cloud

DB MIGRATIONS

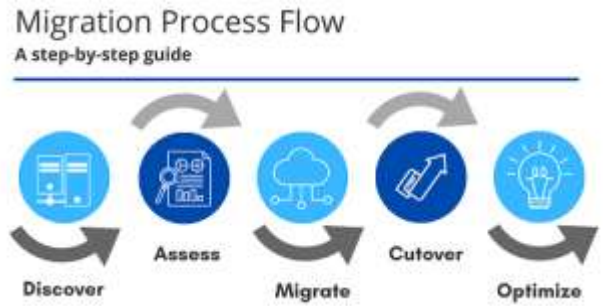
- Migration to Cloud is a step-by-step procedure to be succeeded.
- Steps are;
  - Discover
  - Assess
  - Migrate
  - Cutover
  - Optimize





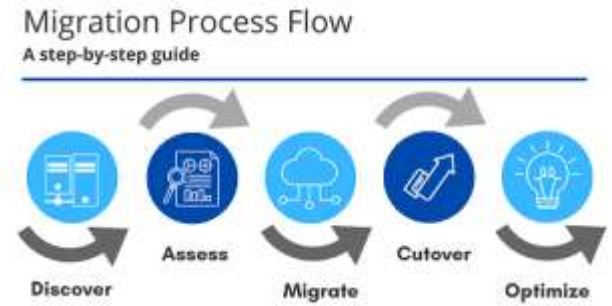
DB MIGRATION EXAMPLE: SQL SERVER TO AZURE SQL MANAGED INSTANCE

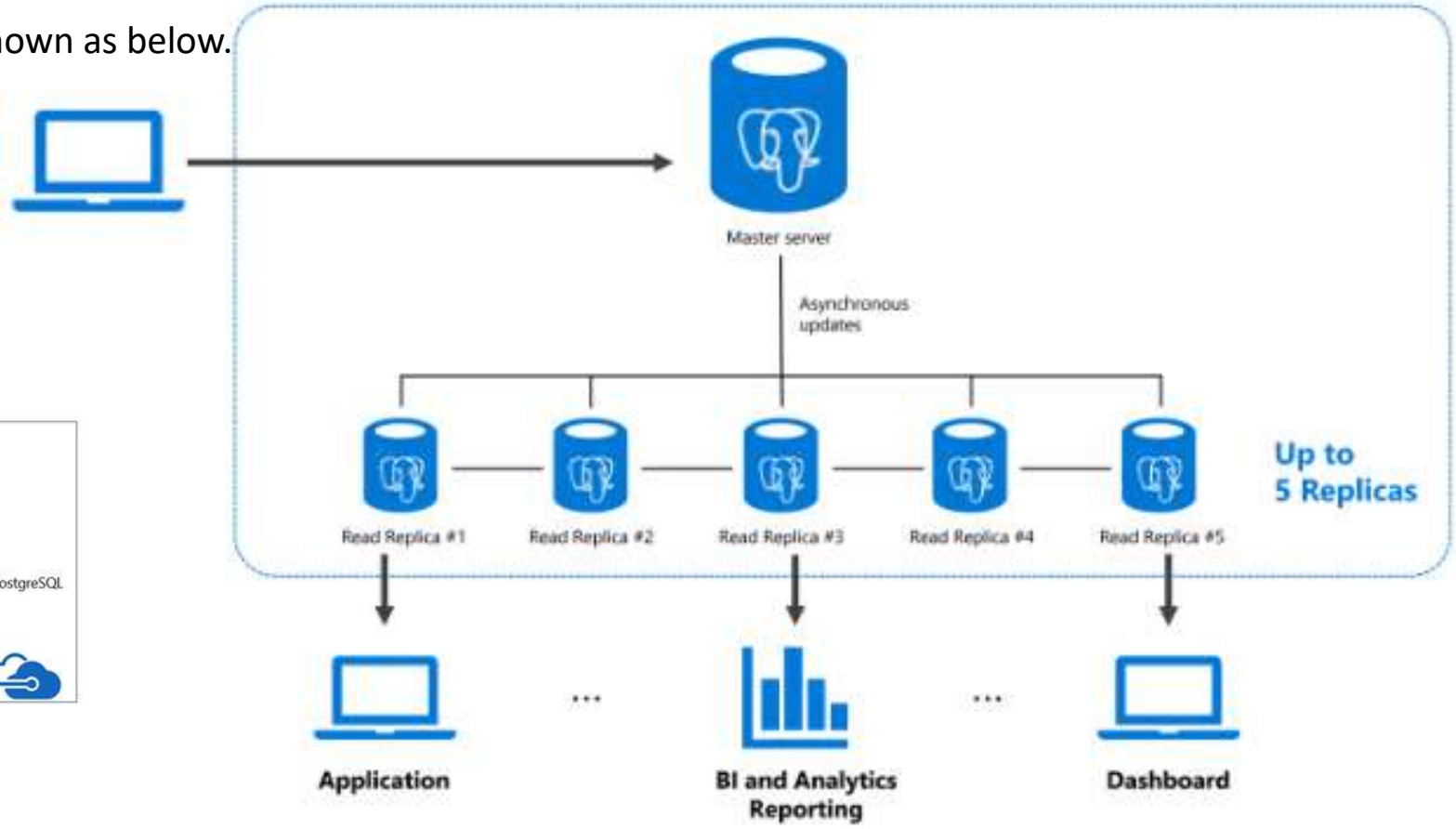
- Customer has several on-prem SQL servers that contains the following data:
  - Business Critical Applications and their database
  - Company critical data (Personal info, financial info, etc.)
  - CRM data (Customer Relationship Management – opportunities, sales, etc.)
- Determine a way forward to move all data to Azure SQL MI.
- Steps:
  - Discover: Use a discovery tool to analyze the needs (Azure Migrate or some 3rd party tools)
  - Assess: Assess all the needs and determine the migration method;
    - BC App & CRM: Need low downtime requirement due to business continuity.
    - Company Data: Do not need low downtime requirements.
    - Use Azure Migrate Assess tool (or any other 3rd party tool) to assess SQL servers to be migrated.
    - Review the assessment results and determine the migration method: BC App & CRM: Azure Data Studio for zero downtime / Company Data: Native backup & restore – need some downtime.
  - Migrate: Use Azure Data Studio to migrate BC App & CRM DBs. Use Backup & Restore to migrate Company data.
  - Cutover: Perform migration cutover and make changes in the applications to use Azure SQL MI as the primary DB.
  - Optimize: Perform tests (validation, performance, etc.) and make necessary changes to optimize the workload.



DB MIGRATION EXAMPLE: ORACLE TO POSTGRESQL ON AZURE

- Oracle to PostgreSQL migration could cause unexpected issues.
- The best practices advise to use a 3rd party tool directly to overcome known issues:
  - Ora2pg tool
- Follow the migration process flow explained in Ora2pg.
- The Master-Replica infrastructure could be shown as below.





Follow the link to see more DB migration best practices.

- https://learn.microsoft.com/en-us/data-migration/

Orion Innovation

Data Security

Security is a completely different chapter that we will focus later but would like to summarize a few options about data security.

Cloud systems provide the following guidelines to protect the data;

1. **Encryption at Rest:** Storage devices support AES-256, FIPS-140-2 and they are enabled by default, i.e. the data is encrypted automatically. Virtual Disks are encrypted with Disk Encryption algorithms (i.e. BitLocker for Windows, dm-crypt for Linux). Key Manager systems (i.e. Azure Key Vault) stores the keys and secrets, even if these images are downloaded, they cannot be decrypted without these keys/secrets. SQL DBs could protect columns / rows and alert if they are reached.
2. **Encryption in Transit:** Transport level security like HTTPS for API calls and SMB 3.0 for File shares.
3. **CORS Support:** Cross-Origin Resource Sharing ensures only authenticated services / apps reach to the desired media / docs.
4. **RBAC:** Role-Based Access Control ensures authentication and authorization levels are set accordingly.
5. **Access Audit:** Storage Analytics tool (in Azure) could log all activities to see which files are reached / denied by which users.
   a) **Account Keys:** Storage Account keys are provided to the applications for CORS support.
   b) **SAS Services:** Shared Access Signatures could be defined in service-level or account-level to secure the storage account.
6. **Network Security:** Some examples are Firewall, NSG, ASG, WAF, NVA and VPN services (will be explained later).

Orion Innovation

Lab Session

LAB#3: TESTING REPLICATION & DISASTER RECOVERY SCENARIOS ON CLOUD

- Follow the instructions: https://learn.microsoft.com/en-us/training/modules/provide-disaster-recovery-replicate-storage-data/5-exercise-failover-secondary-location

LAB#4: CREATING AN SQL DB ON CLOUD

Follow the instructions: https://learn.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?view=azuresql&tabs=azure-portal

LAB#5: CREATING A FILE SHARE ON CLOUD

Follow the instructions: https://learn.microsoft.com/en-us/azure/storage/files/storage-how-to-use-files-portal?tabs=azure-portal

HOMEWORK

- Compare AWS and Azure's DB services. Provide a short summary.
- Research and provide a summary about Azure & AWS Backup solutions.
- Try to learn more about Network:
  - Network Devices: https://www.youtube.com/watch?v=H8W9oMNSuwo (30 min. watch)
  - OSI Reference Model: https://www.youtube.com/watch?v=t-ai8JzhHuY (32 min. watch)

Further Reading:

Comparison of SQL Service vs SQL on Azure VM

Thank You

orioninc.com