AWS RDS

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa

Content

- Amazon RDS
  - DB instance and engine
  - Parameter group
  - Standby Instances and Read Replicas
- Amazon Aurora
  - Aurora DB cluster and storage
  - Aurora features
  - Cloning
  - Aurora serverless

Amazon Relational Database Service (RDS)

RDS is a web service that makes it easier to **set up**, **operate**, and **scale** a **relational database** in the AWS Cloud.

It provides cost-efficient, resizable capacity for an industry-standard relational databases and manages common database administration tasks.

To deliver a managed service experience, Amazon RDS doesn't provide shell access to DB instances. It also **restricts access** to certain system procedures and tables that require advanced privileges.

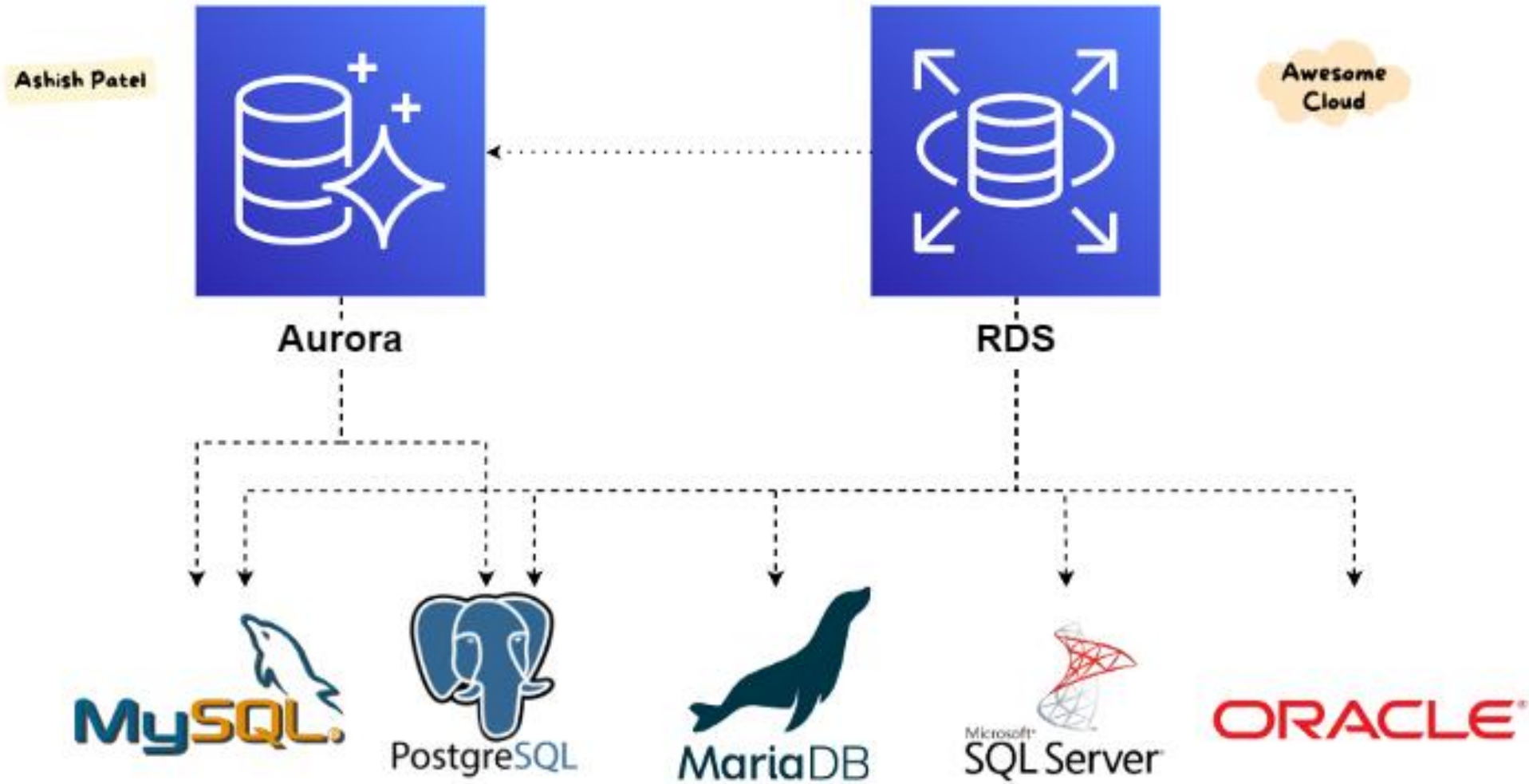
RDS benefits

- **Backups** – RDS automatically takes a backup every day. Transaction logs are stored on AWS which allows you to restore data at any point in time.
- **Availability and durability** – RDS improves HA by creating a standby instance or read replica in a different AZ. In Aurora, your data is automatically replicated in multiple AZs.
- **Automatic Failover** – If the primary database instance goes down, the standby or reads replicas can be promoted to a primary instance in 2 minutes.
- **Increase Read Capacity** – You can offload significant amounts of read operations from the primary instance by reading the data from a read replica.
- **Security** – Data in transit and stored are replicated.
- **Manageability** – You will receive a set of metrics about your database instance. You can enforce SG, parameter group, instance configuration with AWS Config and do some actions automatically based on the events generated by your DB.

When having a DB on EC2, you will lose all the benefits above. The only reason for using a DB on EC2 over RDS is that you need full control over your database.

DB instance and engine

- **DB instance** is an isolated database environment in the AWS Cloud. Your DB instance can contain multiple databases. You run DB instance in the **VPC** in private subnet. A **security group** controls the access to a DB instance.
- Alternative to SG, you can also use IAM tokens to access your databases. You don’t need to store the DB username and password on the application server. Instead, directly access the database from the EC2 instance. All you need is the “rds-db:connect” policy in the EC2 IAM profile.
- RDS supports 5 engines MySQL, MariaDB, PostgreSQL, Oracle, SQL Server. Each DB engine has its own supported features. Additionally, each DB engine has a set of parameters in a DB parameter group that control the behavior of the databases.



Source: [AWS — Difference between Amazon Aurora and Amazon RDS]

RDS storage types

For non-Aurora engines, you can select storage volume type:

- **General Purpose SSD** – Offers cost-effective storage. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS. Baseline performance for these volumes is determined by the volume's size. Use it for development and testing environments.
- **Provisioned IOPS** – Provisioned IOPS storage is designed to meet the needs of I/O-intensive workloads, particularly database workloads, that require low I/O latency and consistent I/O throughput. It is consistent. Use it for production.
- **Magnetic** – Legacy volume. AWS recommends that you use General Purpose SSD or Provisioned IOPS for any new storage needs.

Question

- You have a production server with **general purpose EBS**. The application intermittently gets so slow. When you check the CPU and memory utilization of the server and database, it is relatively low all the time. What would be the cause?

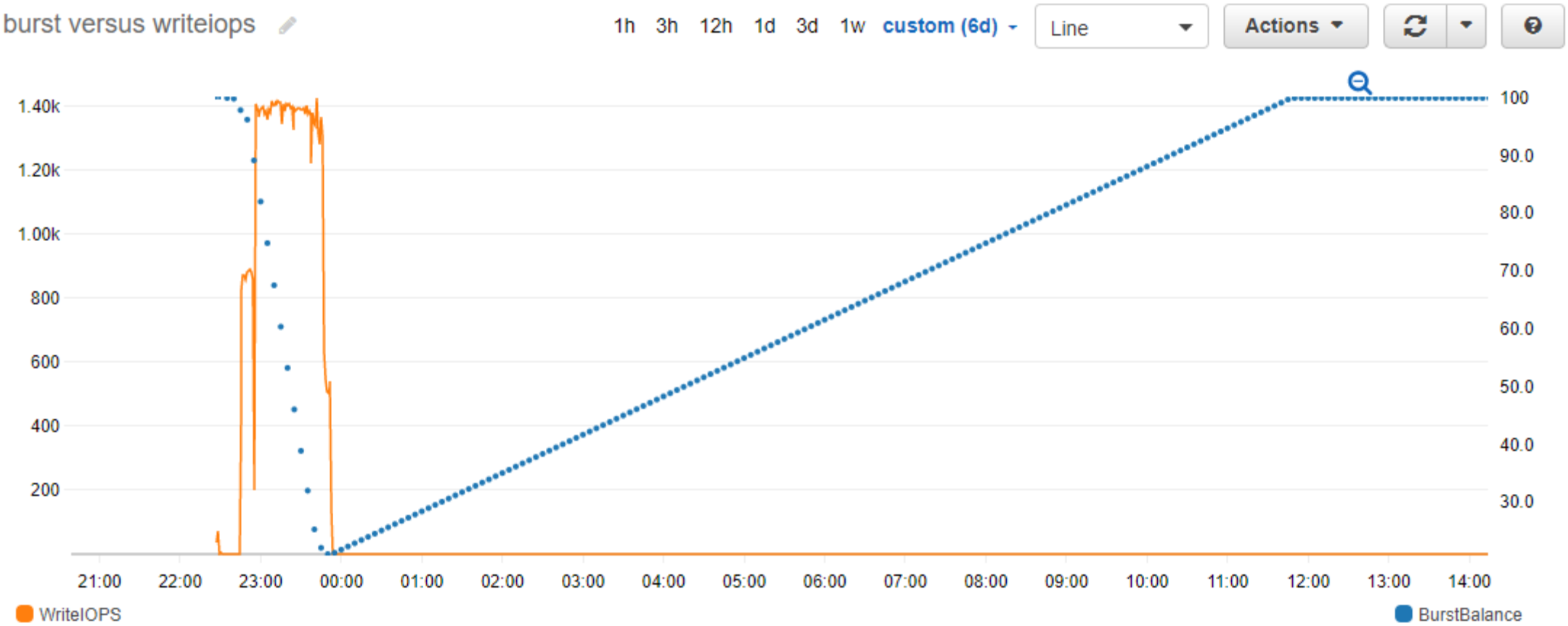
IOPS credits of the general purpose EBS

- The maximum IOPS of general-purpose EBS volume is 3000 IOPS. Use provisioned IOPS volumes if you need more than 3000 IOPS. If the volume size is more than 1000 GiB, then the general-purpose EBS volume can sustain the best (3000) IOPS all the time. If it is less than 1000 GiB, then it can sustain the best IOPS for a while for instance 30 minutes then the IOPS decreases until the base IOPS. To calculate the base IOPS, multiply the volume size by 3. For instance, if the volume size is 100 GiB, the base IOPS is 300.

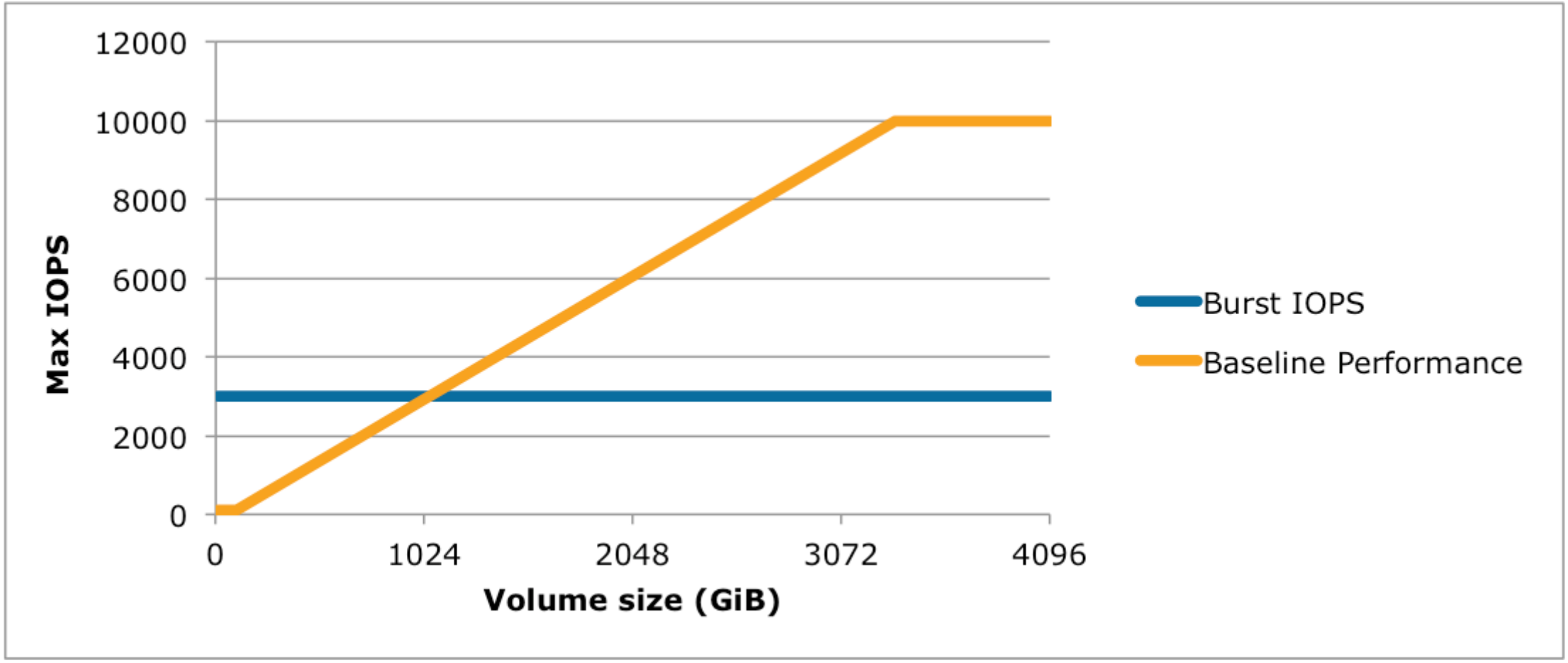
- The server became slower because it run out of IOPS credits.

Burst vs. Baseline Performance

- The gp2 storage type has a **base IOPS** that is set when the volume is created. However, you don't provide a value for the IOPS directly—instead, IOPS is a function of the size of the volume.
- The IOPS for a gp2 volume is the size of the volume in GiB x 3, with a minimum of 100 IOPS and a maximum of 10K IOPS.
- To understand burst mode, you must be aware that every gp2 volume regardless of size starts with 5.4 million I/O credits at 3000 IOPS. **Works out to 3000 IOPS for 30 minutes**. **Then it gets exhausted**! There will be too much latency even if CPU and memory utilization is low.
- This is ideal for "bursty" workloads, such as daily reporting and recurring jobs.


burst versus writeiops
1h 3h 12h 1d 3d 1w custom (6d)
Line
Actions
1.40k
1.20k
1.00k
800
600
400
200
100
90.0
80.0
70.0
60.0
50.0
40.0
30.0
21:00 22:00 23:00 00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00
WriteIOPS
BurstBalance

An important thing to note is that for any gp2 volume larger than 1 TiB, the baseline performance is greater than the burst performance. For such volumes, burst is irrelevant because the baseline performance is better than the 3,000 IOPS burst performance.

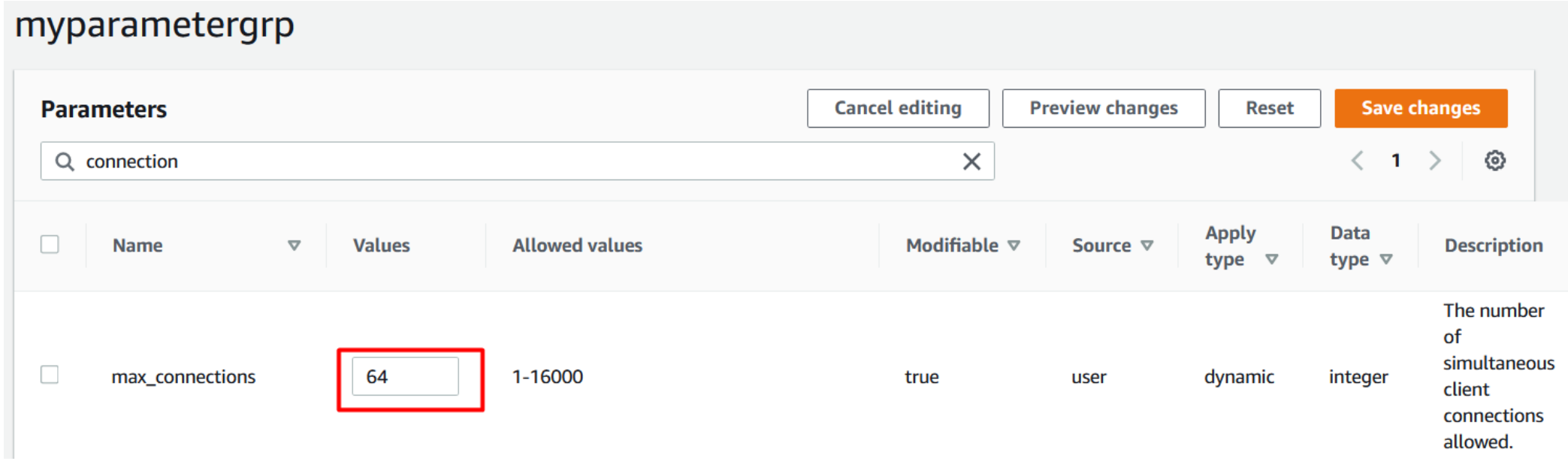


RDS parameter group

You manage your **DB engine configuration** using **parameter groups**.

Examples of settings in parameter group:

- Increase maximum number of connections.
- Enable BinLog to export data to data warehouse.



RDS Backups and Snapshots

Backups are **automatically enabled** in RDS daily with the retention period.

Transaction logs are backed-up by RDS every 5 minutes, which gives us the ability to **restore to any point in time** except the last 5 minutes.

DB Snapshots are backups manually **triggered by the user**. When you restore backups, you need to create a new instance which takes approximately 45 minutes. To reduce outage period, you create a standby instance along with your database. But remember, then that doubles the cost.

Multi-AZ deployment

Increases availability by creating another DB instance in another AZ.

In RDS (Non-Aurora), it creates a **synchronous standby** instance or cluster.

1. Standby **instance** – You cannot read.
2. Standby **cluster** – You can read data. That means now you are not just paying double just in case the primary instance goes down.

In Aurora, it creates an **asynchronous read replica**.

During the outage, the standby instance (in non-Aurora) or read replica (in Aurora) is promoted to a primary instance in 2 minutes. During the automatic failover, the **URL stays the same**.

Standby Replica

In a Multi-AZ deployment, Amazon RDS (Non-Aurora) automatically provisions and maintains a **synchronous standby** replica in a different Availability Zone.

Standby replica provides

- data redundancy
- eliminate I/O freezes
- minimize latency spikes during system backups – The backup is taken from the standby instance.
- enhance availability during planned system maintenance


Region
Availability Zone
Amazon RDS
Amazon RDS standby replica
Availability Zone
Availability Zone
Amazon RDS
Amazon RDS DB instance

RDS Read Replicas

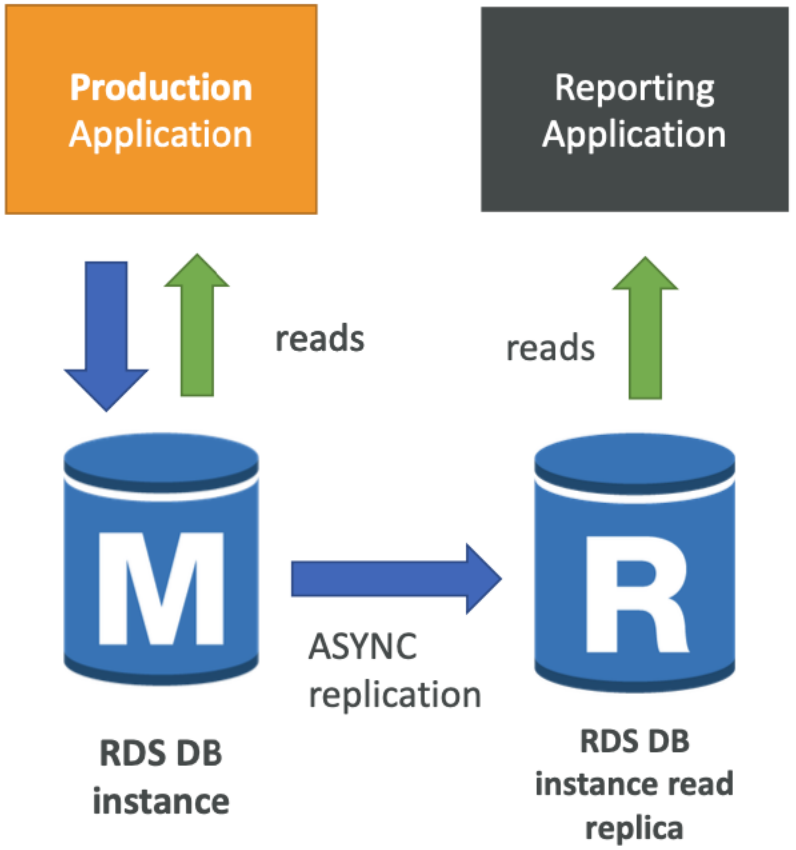
Amazon RDS Read Replicas improve

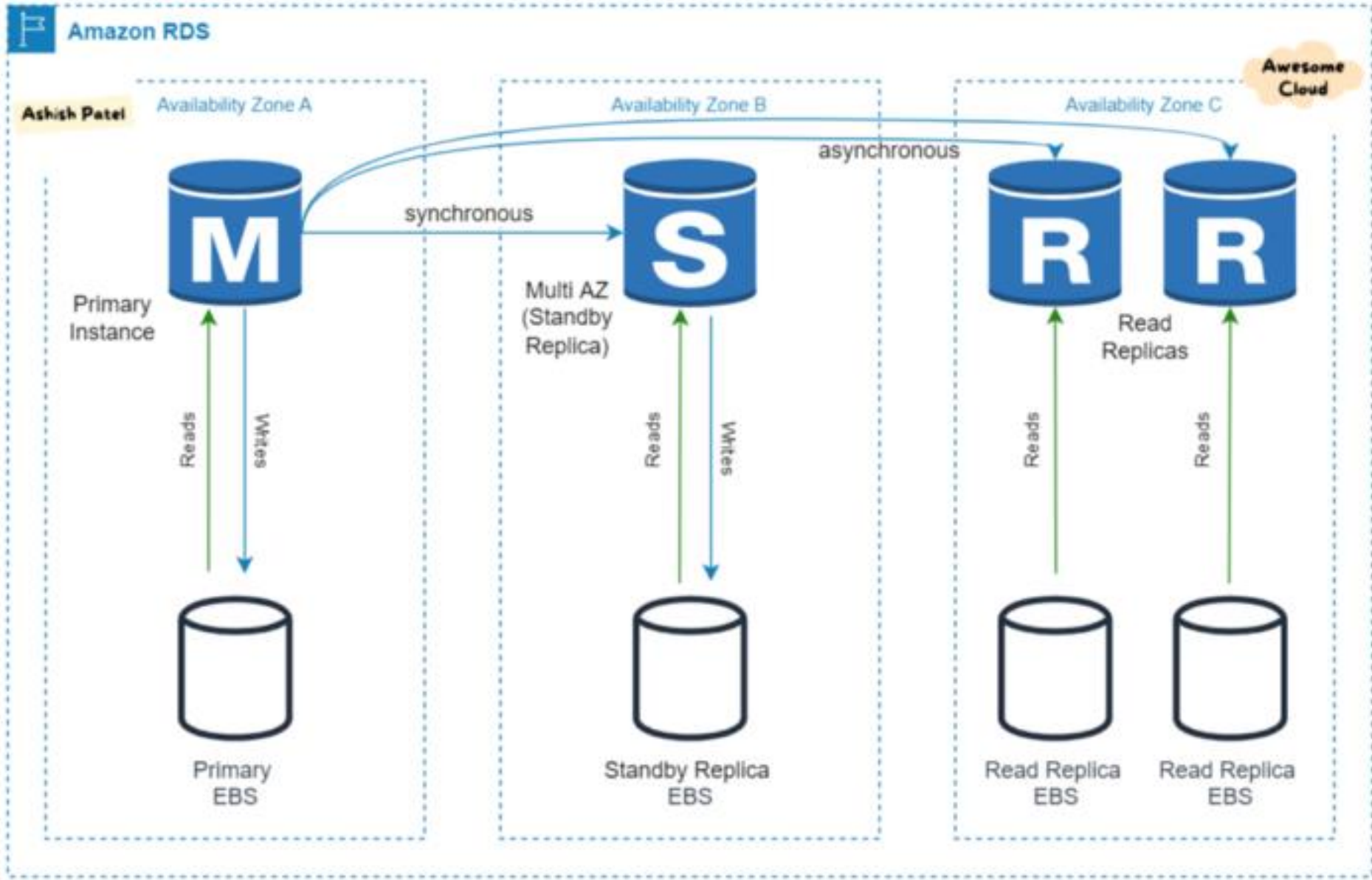
- enhanced performance
- scalability
- increased read throughput
- durability

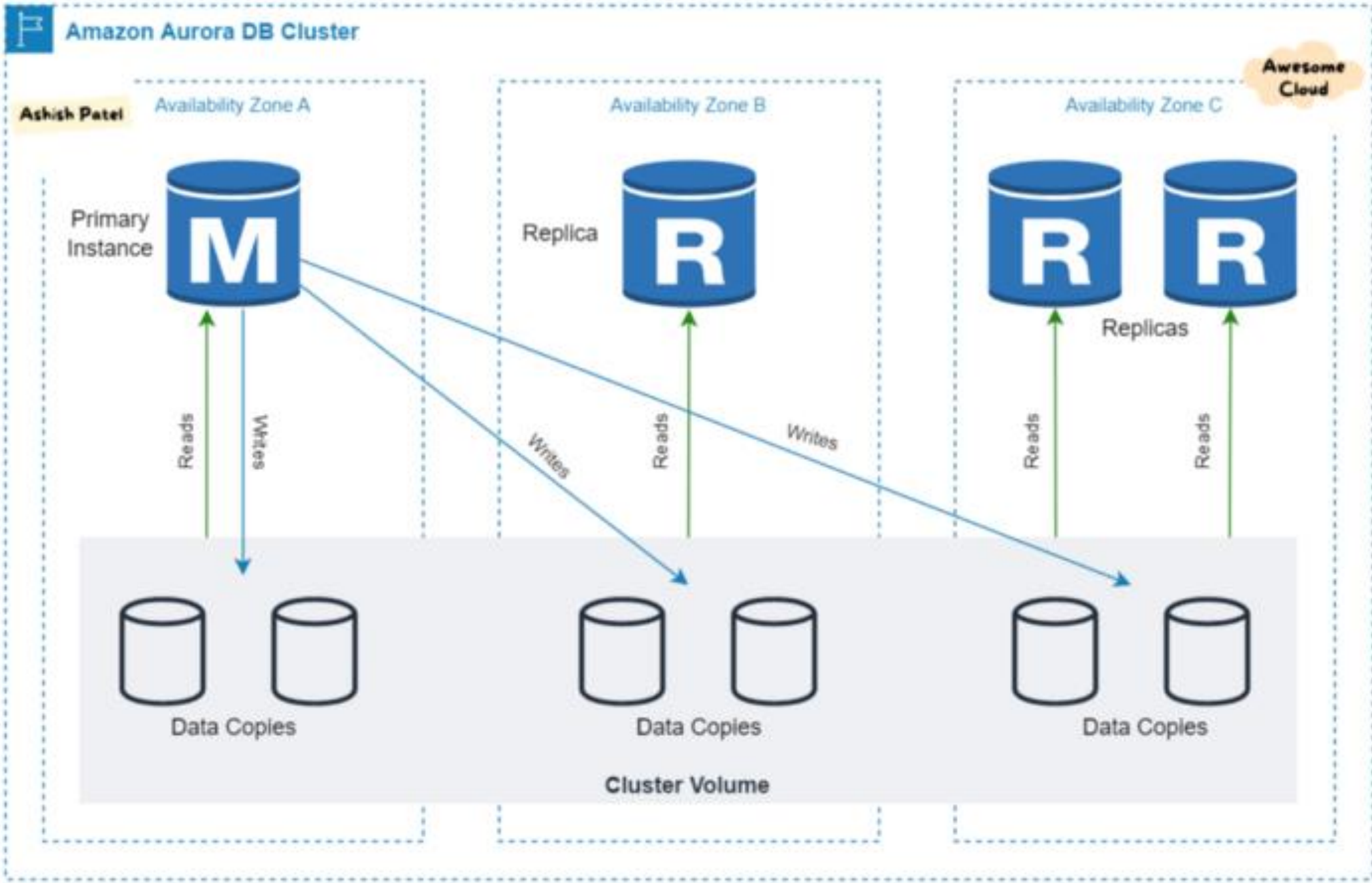
Replication is ASYNC, so reads are eventually consistent. All transactions are secure.

Replicas can be promoted to their own DB.

The Read Replicas be setup as **Multi AZ** for Disaster Recovery.




Amazon RDS
Ashish Patel
Availability Zone A
Availability Zone B
Availability Zone C
Awesome Cloud
asynchronous
synchronous
M
S
R
R
Primary Instance
Multi AZ (Standby Replica)
Read Replicas
Reads
Writes
Reads
Writes
Reads
Reads
Primary EBS
Standby Replica EBS
Read Replica EBS
Read Replica EBS


Amazon Aurora DB Cluster
Awesome Cloud
Ashish Patel
Availability Zone A
Availability Zone B
Availability Zone C
Primary Instance
Replica
Replicas
Reads
Writes
Writes
Reads
Writes
Reads
Reads
Data Copies
Data Copies
Data Copies
Cluster Volume

Amazon Aurora

AWS build Amazon Aurora on top of the open-source **MySQL** and **PostgreSQL**. It provides more features and is fully managed by AWS.

Aurora can deliver up to five times the throughput of MySQL and up to three times the throughput of PostgreSQL. It takes advantage of fast **distributed storage**. The underlying storage grows automatically as needed up.

With Amazon Aurora, compute and storage is decoupled.

Amazon Aurora DB clusters

An Amazon Aurora DB cluster consists of one or more **DB instances** and a cluster **volume.** An Aurora **cluster volume** is a virtual database storage volume that spans **3 AZs**, with each AZ having a copy (**replication**) of the DB cluster data. Two types of DB instances make up an Aurora DB cluster:

1. **Primary DB instance** – Supports read and write operations, and performs all of the data modifications to the cluster volume. Each Aurora DB cluster has one primary DB instance.
2. **Aurora Replica** – Connects to the same storage volume as the primary DB instance and supports **only read** operations. Replication is done **10** ms.

Amazon Aurora Features

Global databases – a single database that spans multiple AWS Regions, enabling low-latency global reads and disaster recovery from any Region-wide outage. It provides built-in fault tolerance.

Parallel queries – provides faster analytical queries over your current data while maintaining high throughput for your core transactional workload.

Machine learning – simple integration with ML services in AWS. You can use standard SQL that call ML models, pass data to them, and return predictions as query results.

Survivable cache warming – Aurora caches common queries that improves DB performance.

Amazon Aurora Read Replica

Amazon Aurora replicas **share the same underlying storage** as the source instance, lowering costs and **avoiding the need to copy data** to the replica nodes.

Each Aurora DB cluster can have up to 15 Aurora Replicas in addition to the primary DB instance. Maintain **high availability** by locating Aurora Replicas in separate Availability Zones.

Aurora automatically **fails over** to an Aurora Replica in case the primary DB instance becomes unavailable in 30 seconds.

Amazon Aurora Read Replica and Fail-Over

Amazon Aurora doesn't copy or transfer any data during the replication. Instead, just share the underlying storage which helps reach the **minimal replica lag**. This lag is usually much less than 10 milliseconds after the primary instance has written an update.

If the primary instance in a DB cluster using single-master replication fails, Aurora automatically fails over to a new primary instance in one of two ways:

1. By promoting an existing Aurora Replica to the new primary instance (in 30 seconds, ~~1 to 2 minutes~~)
2. By recreating a new primary instance if there isn't any replicas (10 minutes)

RDS vs Aurora

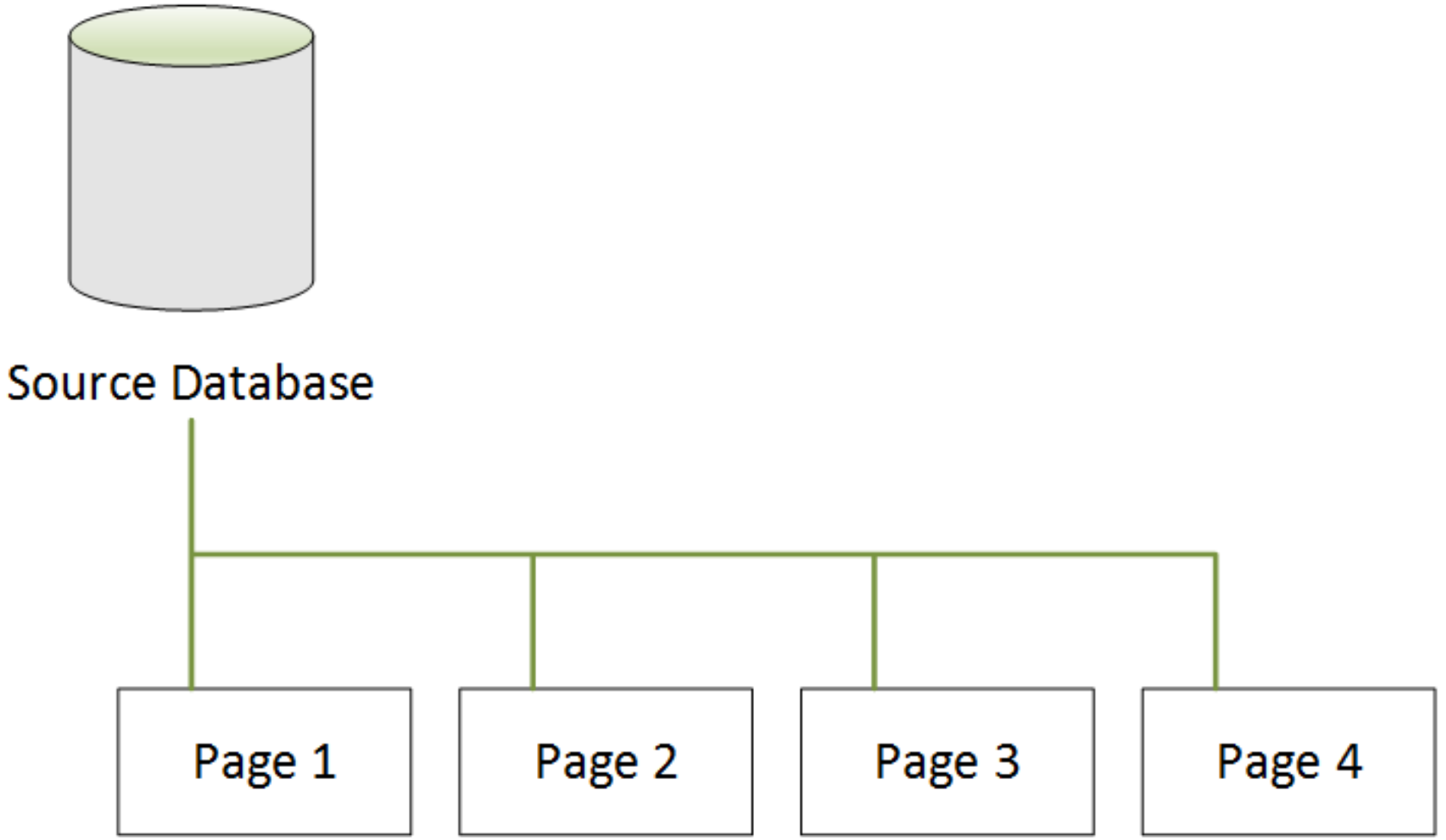
	Multi-AZ deployment (Fail over)	Scalability	Volume
RDS	Standby instance (sync)	Read replica	Inside the instance
Aurora	Read replica (async)	Read replica (async)	Shared (one cluster volume)

Cloning an Aurora DB cluster volume

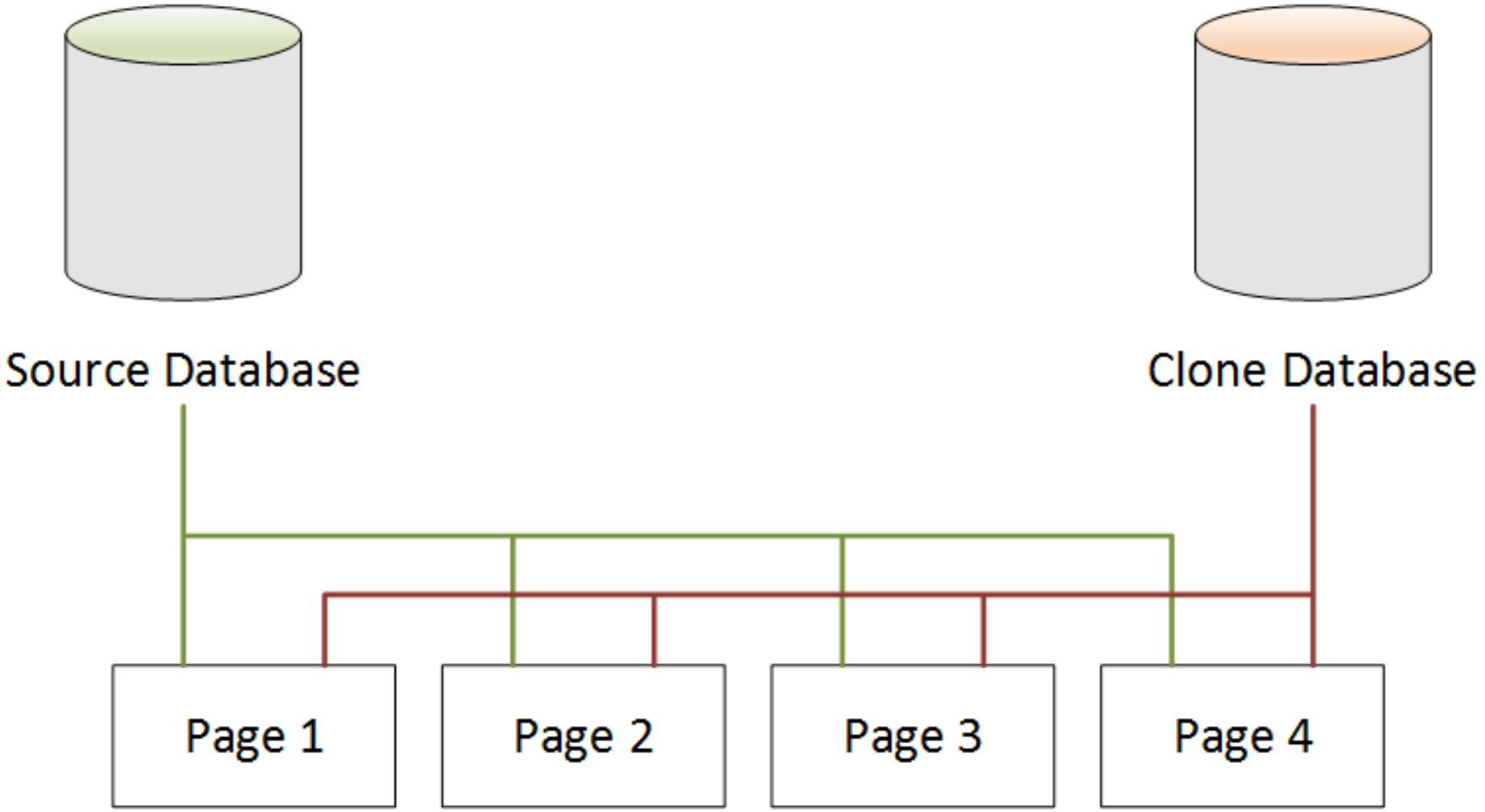
Using the Aurora cloning feature, you can quickly and cost-effectively create a new cluster containing a **duplicate of an Aurora cluster volume** and **all its data**. We refer to the new cluster and its associated cluster volume as a clone.

Creating a clone is faster and more space-efficient than physically copying the data using a different technique such as restoring a snapshot.

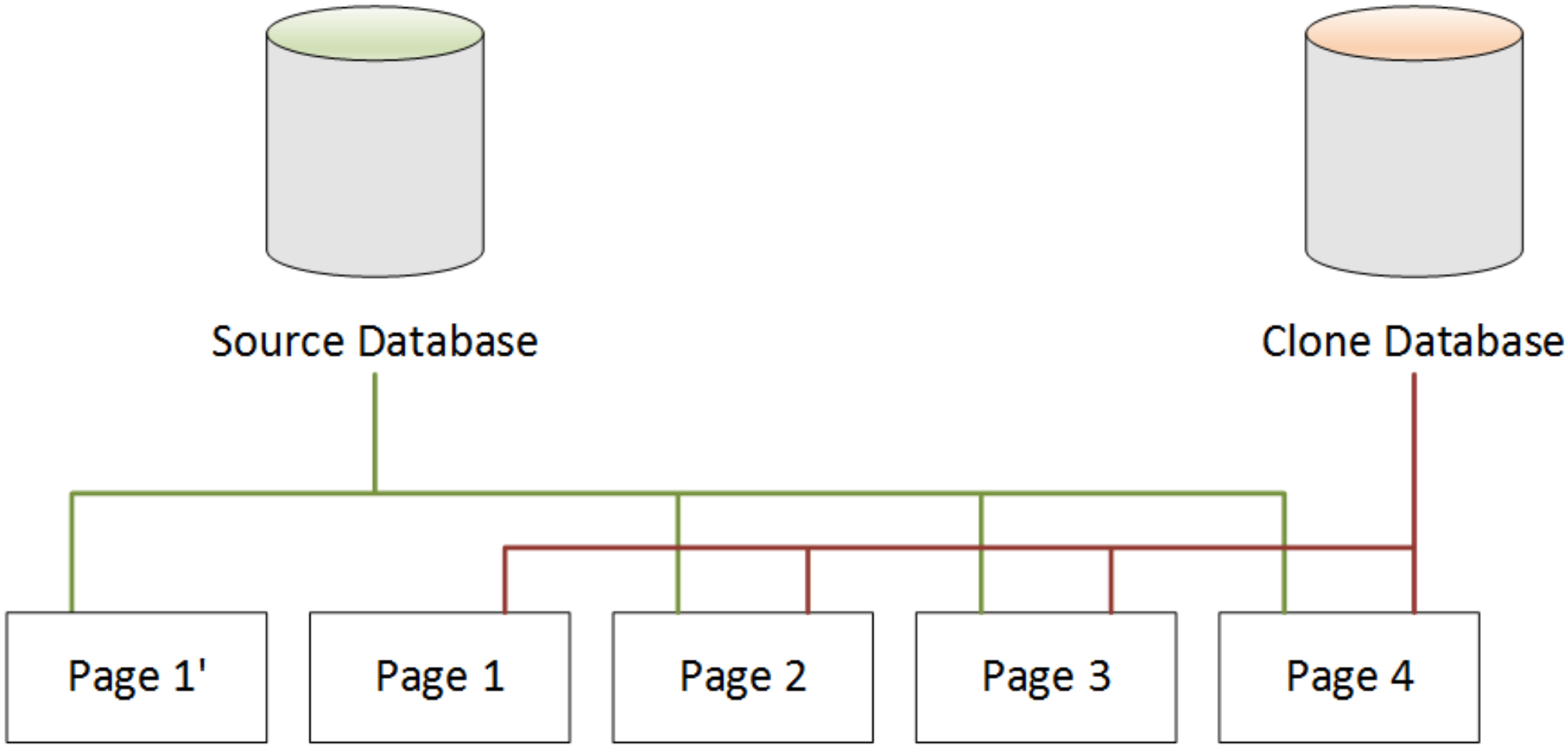
Before cloning



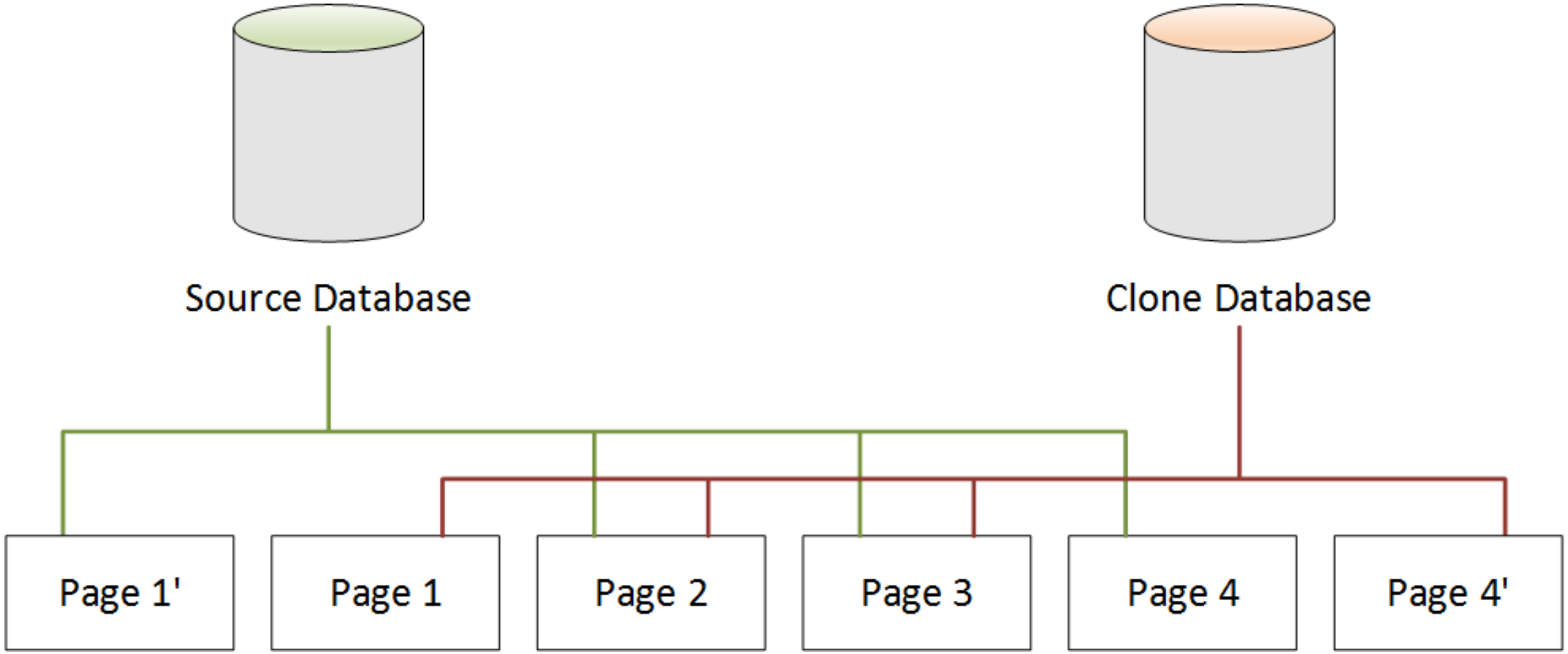
After cloning



When a change occurs on the source cluster volume



When a change occurs on the clone



Aurora serverless

Aurora serverless is an **on-demand autoscaling** configuration for Amazon Aurora. An Aurora Serverless DB cluster is a DB cluster that scales compute capacity up and down based on your application's needs.

This contrasts with Aurora provisioned DB clusters, for which you **manually** manage capacity.

Aurora Serverless is relatively simple, cost-effective option for **infrequent, intermittent,** or **unpredictable** workloads.

Aurora serverless v2

Amazon Aurora Serverless v2 has been architected from the ground up to support serverless DB clusters that are **instantly scalable**. Scales up to hundreds-of-thousands of transactions in a fraction of a second.

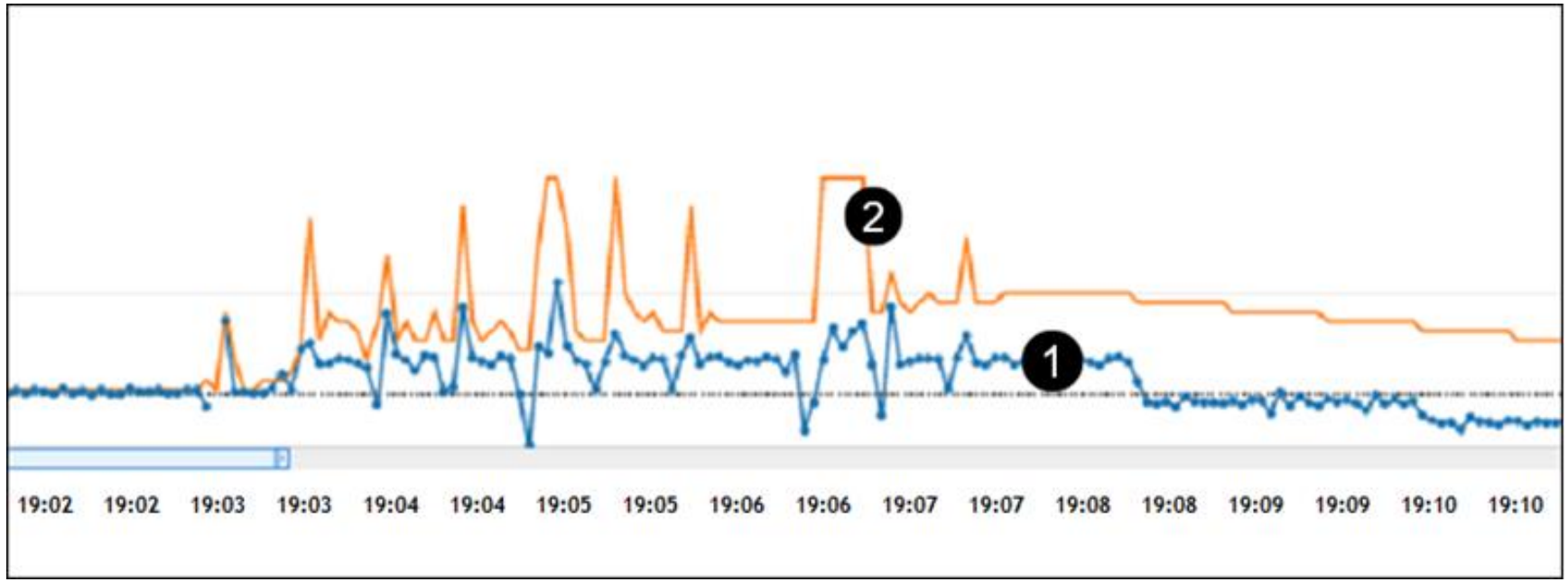
Aurora capacity units (ACUs) - One ACU provides 2 GiB (gibibytes) of memory (RAM) and associated virtual processor (vCPU) with networking.

Minimum ACUs (starts from 0.5 ACU) – The smallest number of ACUs down to which your Aurora Serverless v2 DB cluster can scale.

Maximum ACUs (Up to 256 ACUs) –The largest number of ACUs up to which your Aurora Serverless v2 DB cluster can scale.

Aurora serverless v2 scaling

Unlike Aurora Serverless v1, which scales by doubling ACUs each time the DB cluster reaches a threshold, Aurora Serverless v2 (preview) can increase ACUs incrementally.



1 - Orders processed each second, 2 - Aurora capacity units (ACUs) – Memory and CPU applied over time to increasing and decreasing demand.

Scaled instantly as expected. Full demo is [Aurora Serverless v2 – Instant scaling].

Amazon Aurora v2 updates

- **Multi A-Z support**: Aurora Serverless v2 comes with support for multi-AZ. No need to worry about downtime due to a failure in one particular zone
- **Read Replicas**: With v2, you can now access data from the read replicas of your DB cluster
- More expensive

Read more on a Medium blog

Instance classes and billing

RDS instance classes and billing model are similar to EC2. You have memory optimized, low latency and high IOPS, general purpose instance class types. For billing, you can choose either on-demand (expensive) or reserved (cheaper, 1 to 3-year commitment) instances.

You will be charged by:

- Per hour
- Storage
- I/O requests
- Provisioned IOPS
- Backup storage
- Data transfer

RDS is an expensive service. There will be multiple big instances running if you enabled multi-AZ. When you stop it, after a week it starts automatcally. Delete the instances once completed the lab task.