

Databases



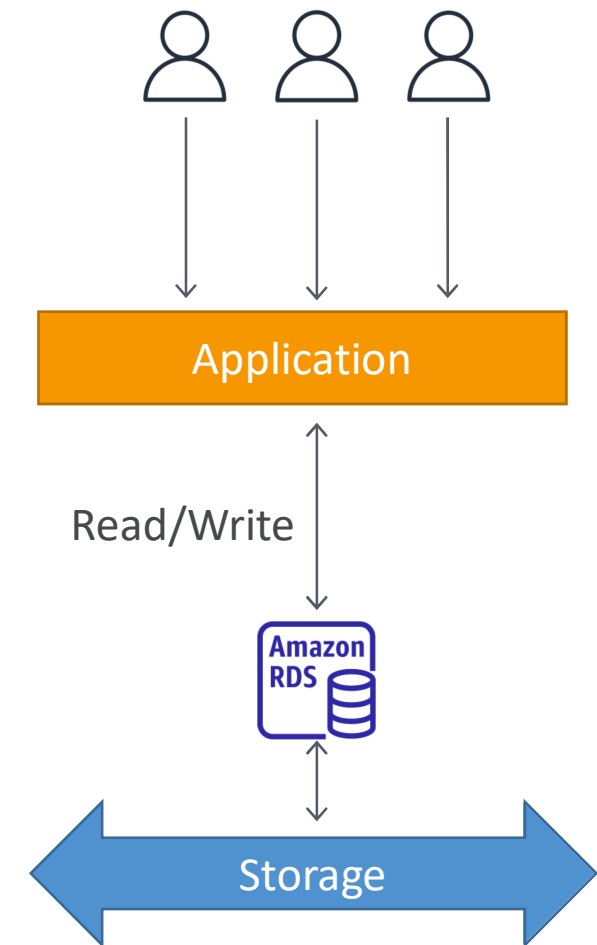


Amazon RDS Overview

- RDS stands for Relational Database Service
- It's a managed DB service for DB use SQL as a query language.
- It allows you to create databases in the cloud that are managed by AWS
 - Postgres
 - MySQL
 - MariaDB
 - Oracle
 - Microsoft SQL Server
 - Aurora (AWS Proprietary database)

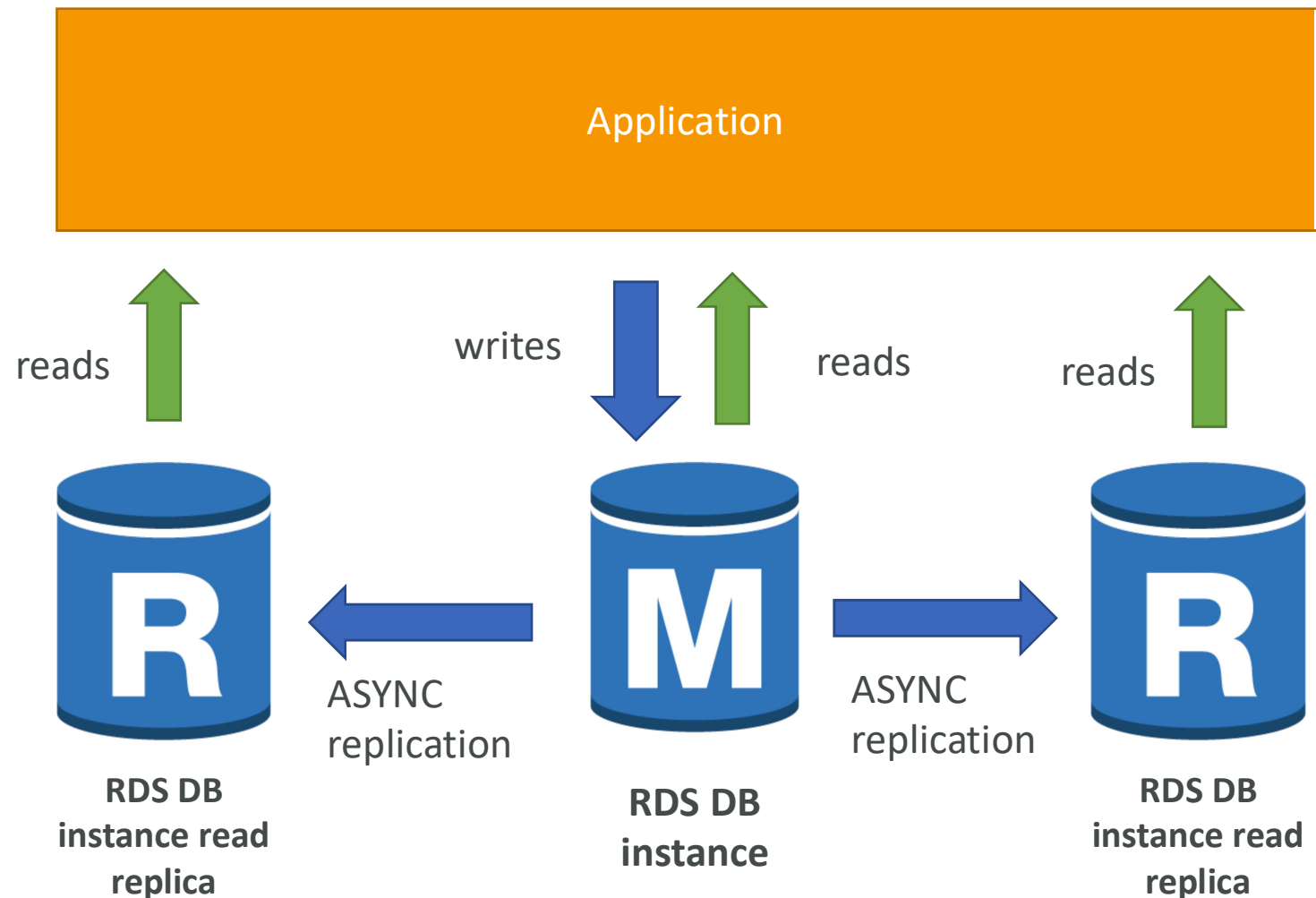
RDS – Storage Auto Scaling

- Helps you increase storage on your RDS DB instance dynamically
- When RDS detects you are running out of free database storage, it scales automatically
- Avoid manually scaling your database storage
- You have to set **Maximum Storage Threshold** (maximum limit for DB storage)
- Automatically modify storage if:
 - Free storage is less than 10% of allocated storage
 - Low-storage lasts at least 5 minutes
 - 6 hours have passed since last modification
- Useful for applications with **unpredictable workloads**
- Supports all RDS database engines (MariaDB, MySQL, PostgreSQL, SQL Server, Oracle)



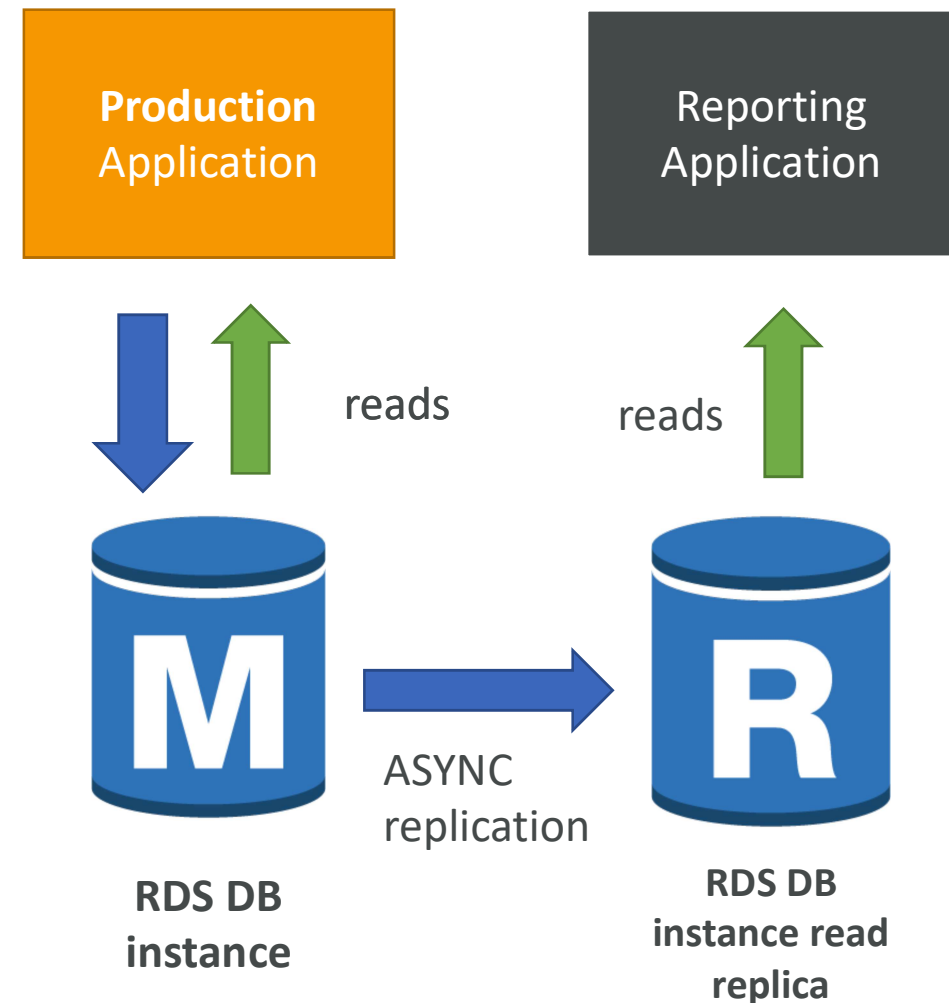
RDS Read Replicas for read scalability

- Up to 5 Read Replicas
- Within AZ, Cross AZ or Cross Region
- Replication is **ASYNC**, so reads are eventually consistent
- Replicas can be promoted to their own DB
- Applications must update the connection string to leverage read replicas



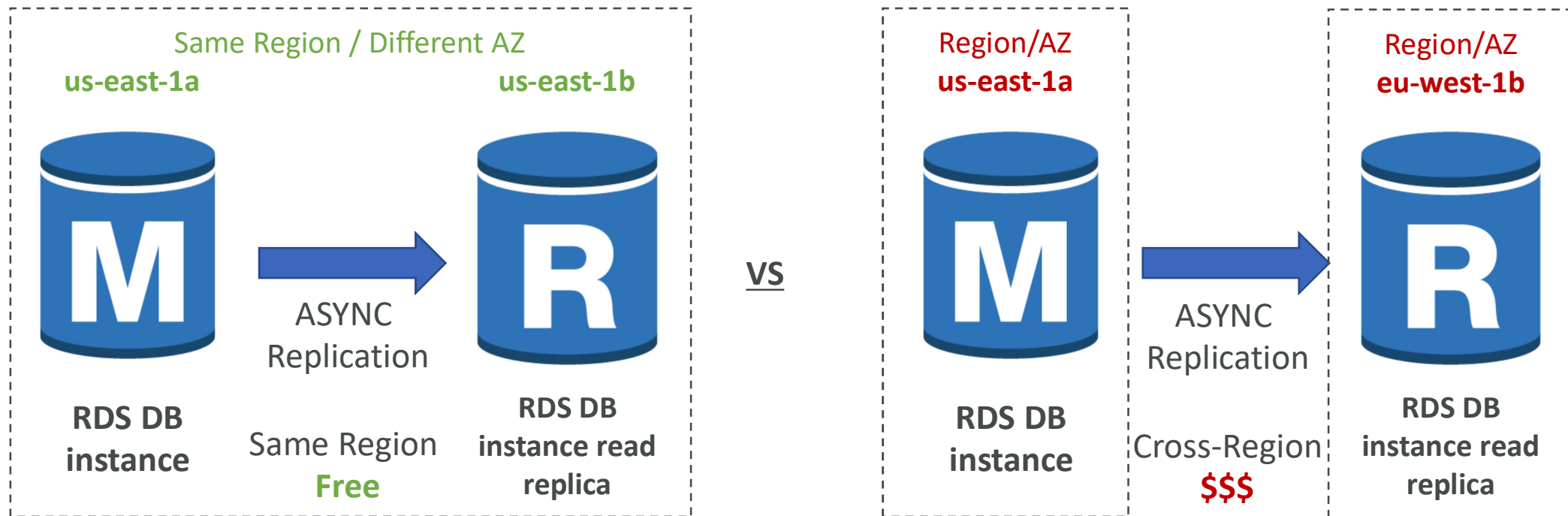
RDS Read Replicas – Use Cases

- You have a production database that is taking on normal load
- You want to run a reporting application to run some analytics
- You create a Read Replica to run the new workload there
- The production application is unaffected
- Read replicas are used for SELECT (=read) only kind of statements (not INSERT, UPDATE, DELETE)



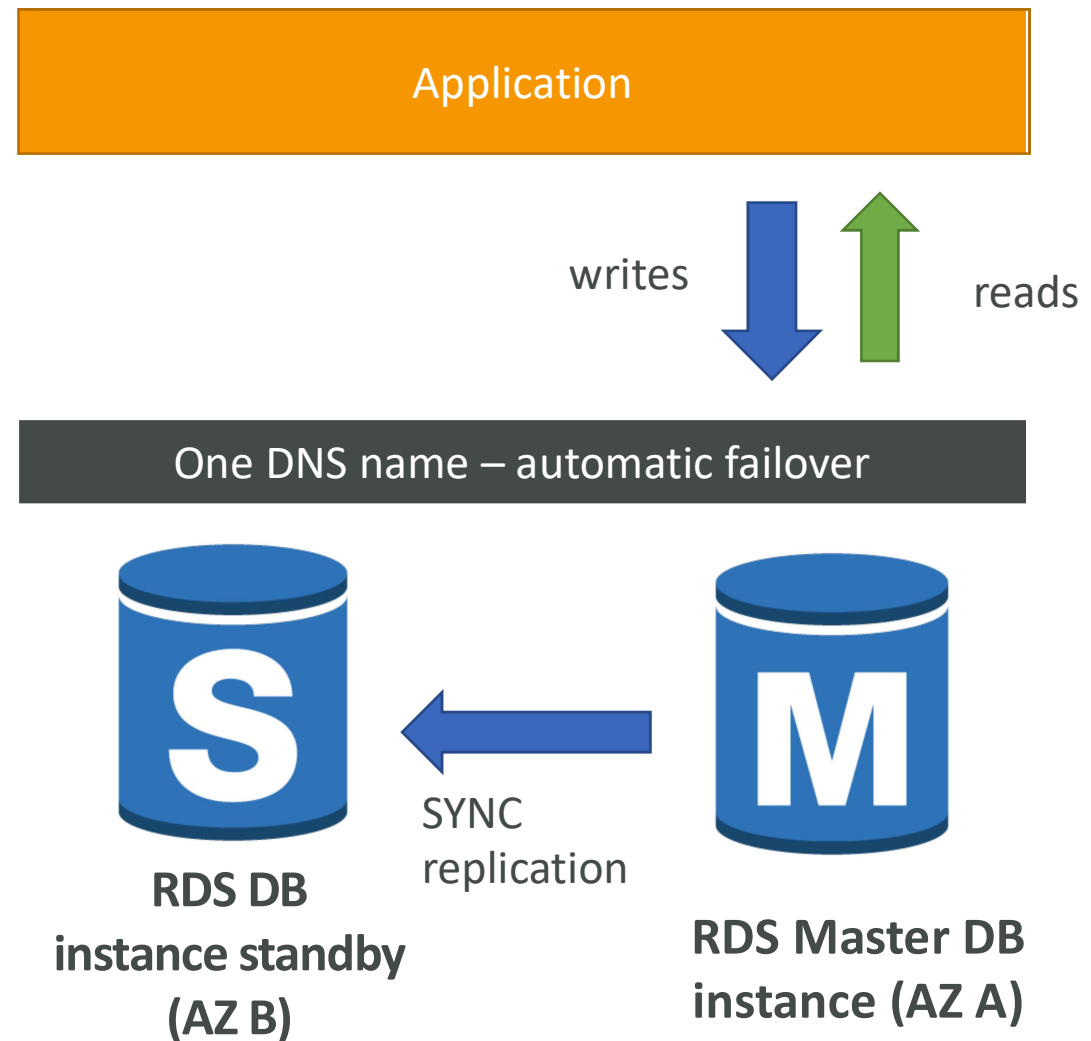
RDS Read Replicas – Network Cost

- In AWS there's a network cost when data goes from one AZ to another
- For RDS Read Replicas within the same region, you don't pay that fee



RDS Multi AZ (Disaster Recovery)

- SYNC replication
- One DNS name – automatic app failover to standby
- Increase **availability**
- Failover in case of loss of AZ, loss of network, instance or storage failure
- No manual intervention in apps
- Not used for scaling
- Multi-AZ replication is free
- Note: The Read Replicas be setup as Multi AZ for Disaster Recovery (DR)



DB Parameter Groups

- You can configure the DB engine using Parameter Groups
- Dynamic parameters are applied immediately
- Static parameters are applied after instance reboot
- You can modify parameter group associated with a DB (must reboot)
- See documentation for list of parameters for a DB technology
- **Must-know parameter:**
 - PostgreSQL / SQL Server: `rds.force_ssl=1` => force SSL connections
 - Reminder: for SSL on MySQL / MariaDB, you must run:
`GRANT SELECT ON mydatabase.* TO 'myuser'@'%' IDENTIFIED BY '....' REQUIRE SSL;`

RDS with CloudWatch

- CloudWatch metrics associated with RDS (gathered from the hypervisor):
 - DatabaseConnections
 - SwapUsage
 - ReadIOPS / WriteIOPS
 - ReadLatency / WriteLatency
 - ReadThroughPut / WriteThroughPut
 - DiskQueueDepth
 - FreeStorageSpace
- Enhanced Monitoring (gathered from an agent on the DB instance)
 - Useful when you need to see how different processes or threads use the CPU
 - Access to over 50 new CPU, memory, file system, and disk I/O metrics



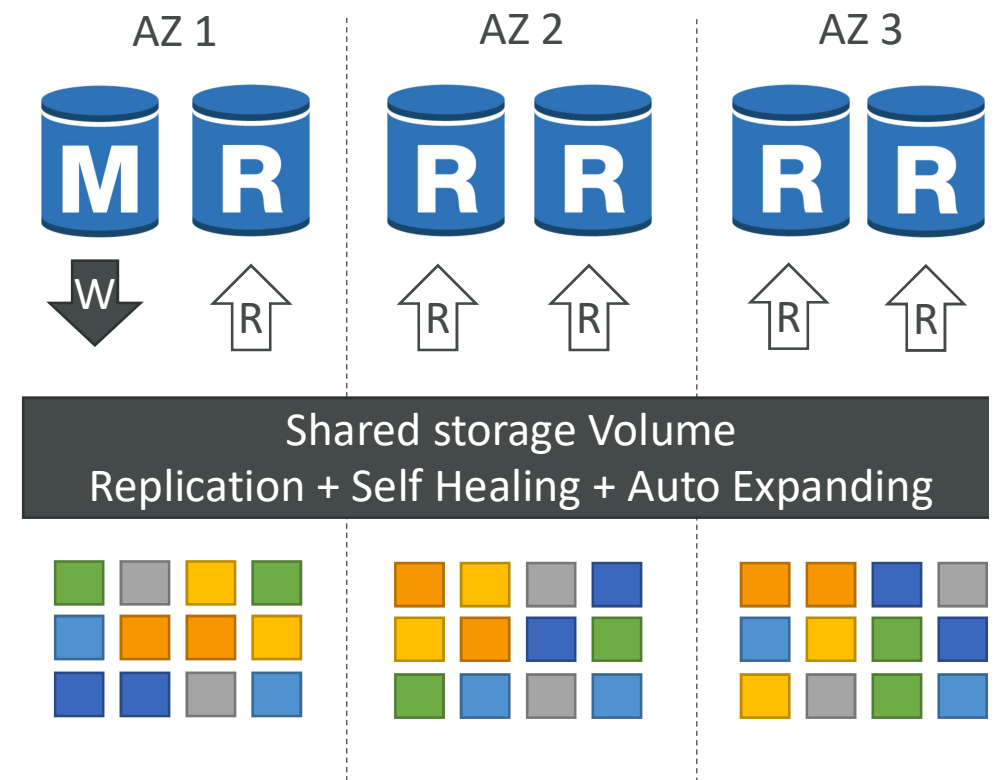
Amazon Aurora



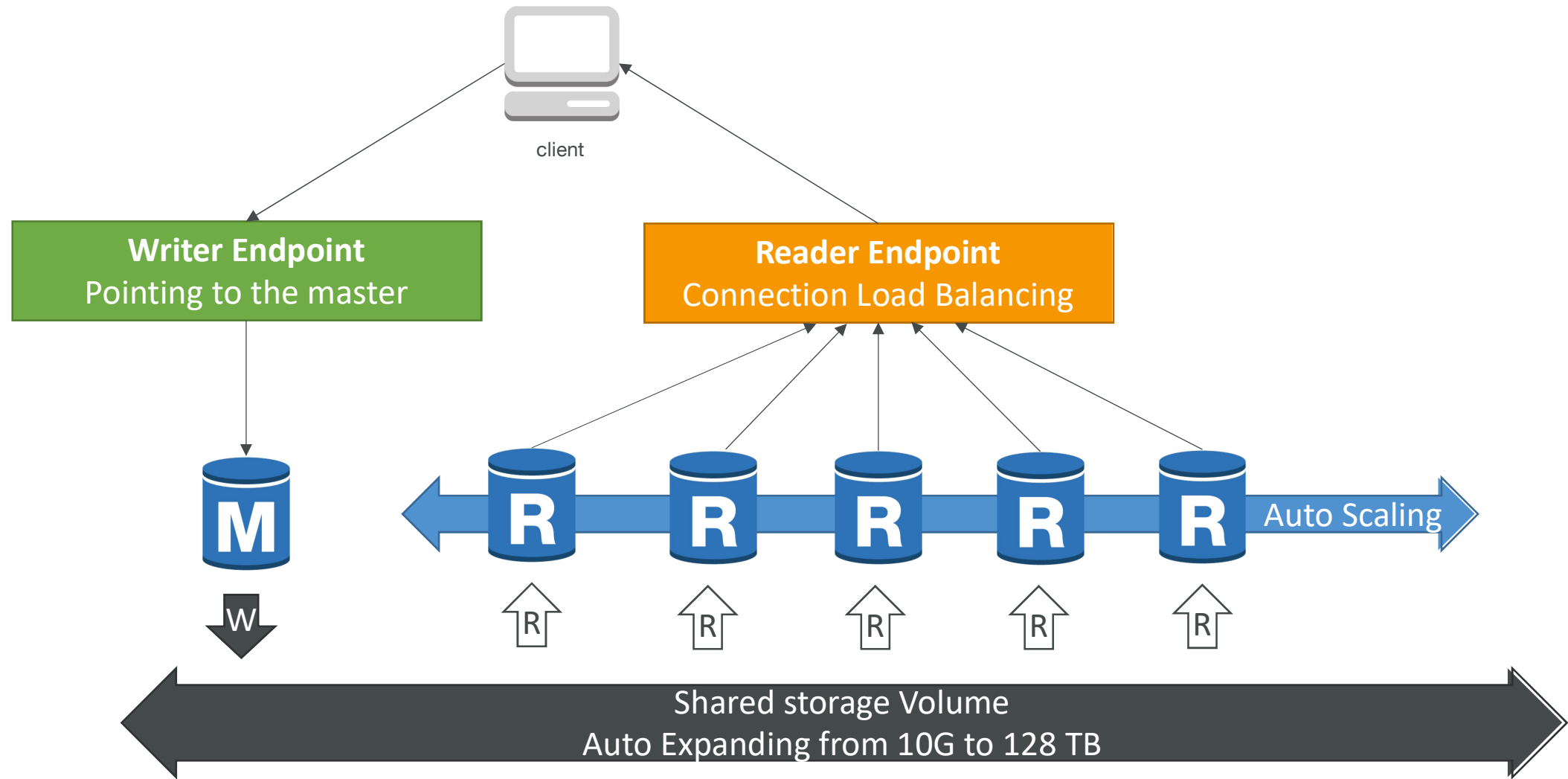
- Aurora is a proprietary technology from AWS (not open sourced)
- Postgres and MySQL are both supported as Aurora DB (that means your drivers will work as if Aurora was a Postgres or MySQL database)
- Aurora is “AWS cloud optimized” and claims 5x performance improvement over MySQL on RDS, over 3x the performance of Postgres on RDS
- Aurora storage automatically grows in increments of 10GB, up to 128 TB.
- Aurora can have 15 replicas while MySQL has 5, and the replication process is faster (sub 10 ms replica lag)
- Failover in Aurora is instantaneous. It's HA (High Availability) native.
- Aurora costs more than RDS (20% more) – but is more efficient

Aurora High Availability and Read Scaling

- 6 copies of your data across 3 AZ:
 - 4 copies out of 6 needed for writes
 - 3 copies out of 6 need for reads
 - Self healing with peer-to-peer replication
 - Storage is striped across 100s of volumes
- One Aurora Instance takes writes (master)
- Automated failover for master in less than 30 seconds
- Master + up to 15 Aurora Read Replicas serve reads
- Support for Cross Region Replication



Aurora DB Cluster



Features of Aurora

- Automatic fail-over
- Backup and Recovery
- Isolation and security
- Industry compliance
- Push-button scaling
- Automated Patching with Zero Downtime
- Advanced Monitoring
- Routine Maintenance
- Backtrack: restore data at any point of time without using backups



Backups, Backtracking & Restores in Aurora

- **Automatic Backups**

- Retention period 1-35 days (can't be disabled)
- PITR, restore your DB cluster within 5 minutes of the current time
- Restore to a new DB cluster

- **Aurora Backtracking**

- Rewind the DB cluster back and forth in time (up to 72 hours)
- Doesn't create a new DB cluster (in-place restore)
- Supports Aurora MySQL only

- **Aurora Database Cloning**

- Creates a new DB cluster that uses the same DB cluster volume as the original cluster
- Uses *copy-on-write* protocol (use the original/single copy of the data and allocate storage only when changes made to the data)
- Example: create a test environment using your production data



Aurora for SysOps

- You can associate a priority tier (0-15) on each Read Replica
 - Controls the failover priority
 - RDS will promote the Read Replica with the highest priority (lowest tier)
 - If replicas have the same priority, RDS promotes the largest in size
 - If replicas have the same priority and size, RDS promotes arbitrary replica
- You can migrate an RDS MySQL snapshot to Aurora MySQL Cluster





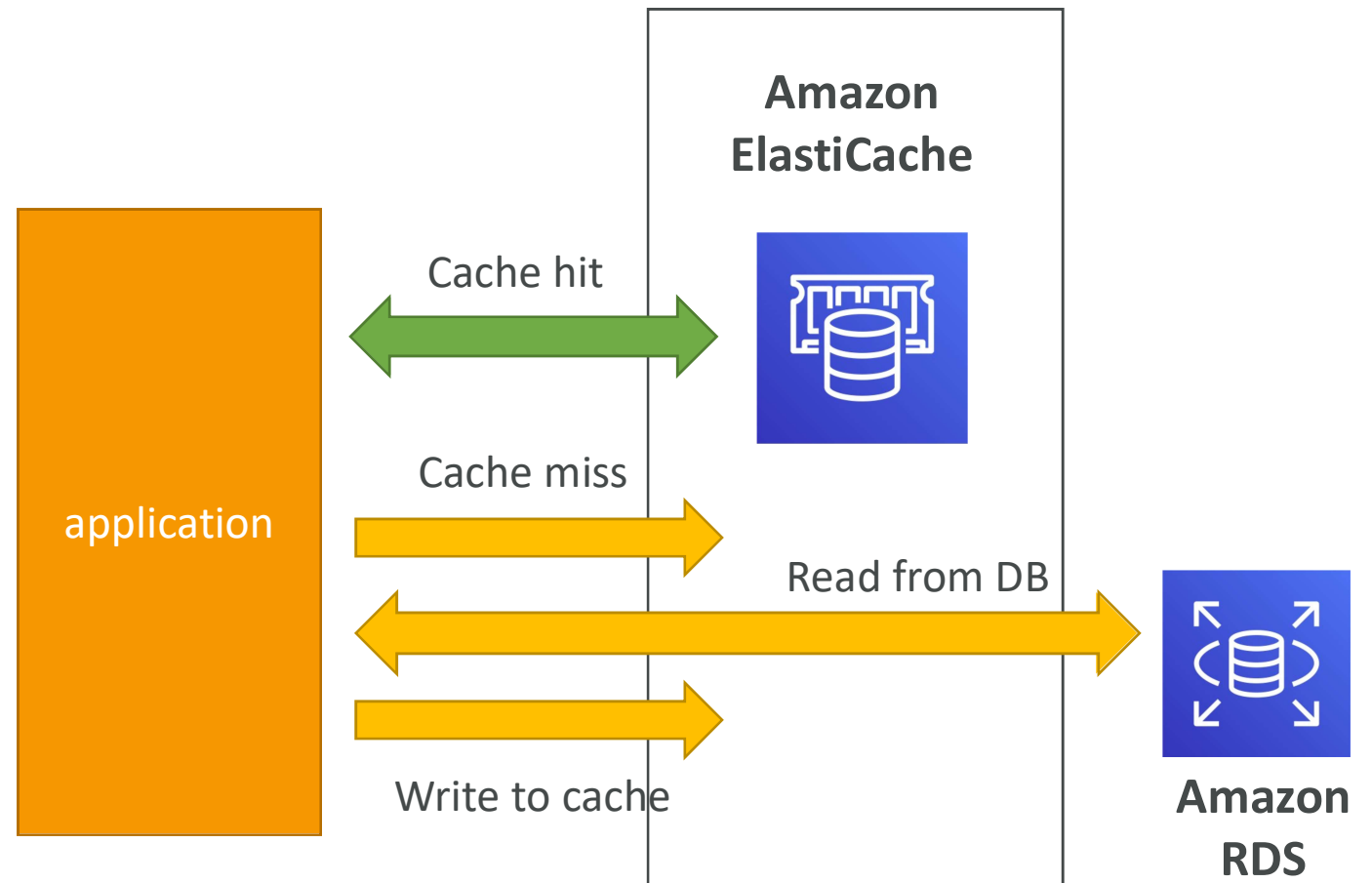
Amazon ElastiCache Overview

- The same way RDS is to get managed Relational Databases...
- ElastiCache is to get managed Redis or Memcached
- Caches are in-memory databases with really high performance, low latency
- Helps reduce load off of databases for read intensive workloads
- Helps make your application stateless
- AWS takes care of OS maintenance / patching, optimizations, setup, configuration, monitoring, failure recovery and backups
- Using ElastiCache involves heavy application code changes

ElastiCache

Solution Architecture - DB Cache

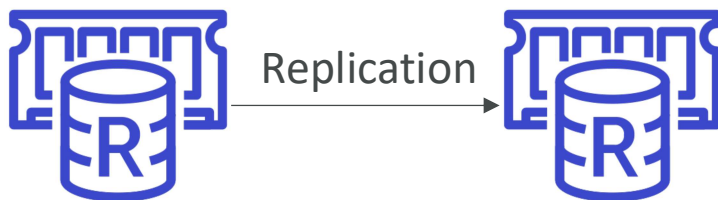
- Applications queries ElastiCache, if not available, get from RDS and store in ElastiCache.
- Helps relieve load in RDS
- Cache must have an invalidation strategy to make sure only the most current data is used in there.



ElastiCache – Redis vs Memcached

REDIS

- Multi AZ with Auto-Failover
- Read Replicas to scale reads and have high availability
- Data Durability using AOF persistence
- Backup and restore features



MEMCACHED

- Multi-node for partitioning of data (sharding)
- No high availability (replication)
- Non persistent
- No backup and restore
- Multi-threaded architecture

