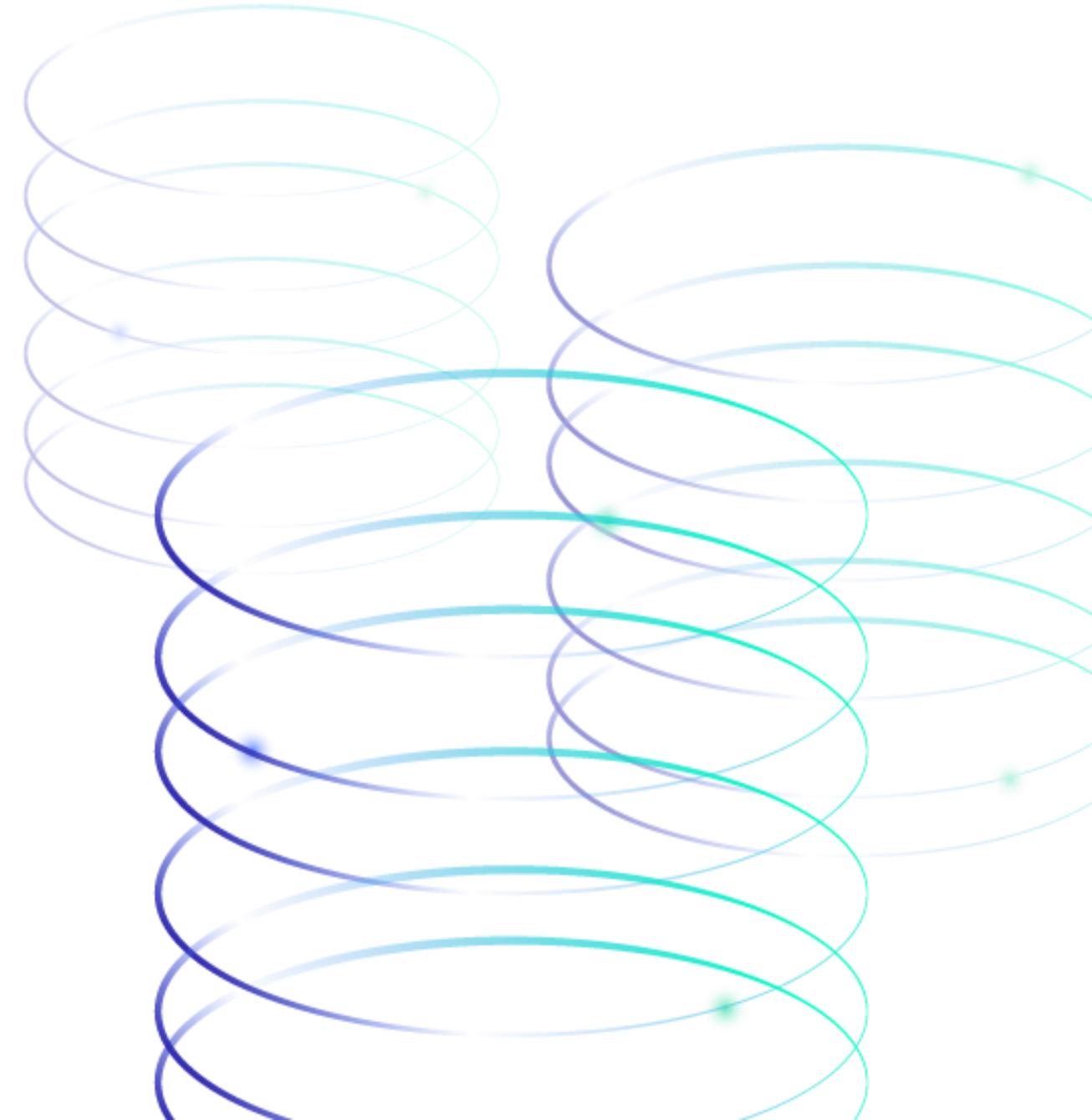


## Aurora MySQL Backtrack을 이용한 빠른 복구

진교선

테크니컬 어카운트 매니저



# 강연 중 질문하는 방법

▼ Questions

☒ Show Answered Questions

Question	Asker

Type answer here

Go to Webinar “Chat/채팅” 창에 자신이 질문한 내역이 표시됩니다. 기본적으로 모든 질문은 공개로 답변 됩니다만 본인만 답변을 받고 싶으면 (비공개)라고 하고 질문해 주시면 됩니다.

## 고지 사항(Disclaimer)

본 콘텐츠는 고객의 편의를 위해 AWS 서비스 설명을 위해 온라인 세미나용으로 별도로 제작, 제공된 것입니다. 만약 AWS 사이트와 콘텐츠 상에서 차이나 불일치가 있을 경우, AWS 사이트(aws.amazon.com)가 우선합니다. 또한 AWS 사이트 상에서 한글 번역문과 영어 원문에 차이나 불일치가 있을 경우(번역의 지체로 인한 경우 등 포함), 영어 원문이 우선합니다.

AWS는 본 콘텐츠에 포함되거나 콘텐츠를 통하여 고객에게 제공된 일체의 정보, 콘텐츠, 자료, 제품(소프트웨어 포함) 또는 서비스를 이용함으로 인하여 발생하는 여하한 종류의 손해에 대하여 어떠한 책임도 지지 아니하며, 이는 직접 손해, 간접 손해, 부수적 손해, 징벌적 손해 및 결과적 손해를 포함하되 이에 한정되지 아니합니다.

# Database on AWS

# 수 년간 발전해 온 어플리케이션의 구조...

# 현 시대 어플리케이션들의 요구사항!

## 더 나은 성능, 확장성, 그리고 가용성을 요구



Users	1M+
Data volume	Terabytes—petabytes
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Virtually unlimited
Economics	Pay-as-you-go
Developer access	Instance API access
Development	Apps & storage are decoupled



E-Commerce



Media streaming



Social media



Online gaming



Shared economy

# 여러 요구 사항에 대응하는 다양한 관리형 데이터베이스



## Relational

Referential integrity, ACID transactions, schema-on-write



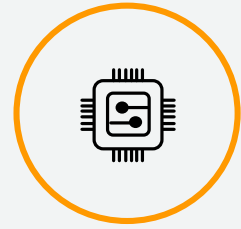
## Key-value

High throughput, Low latency reads and writes, endless scale



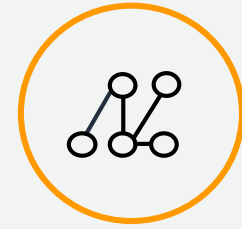
## Document

Store documents and quickly access querying on any attribute



## In-memory

Query by key with microsecond latency



## Graph

Quickly and easily create and navigate relationships between data



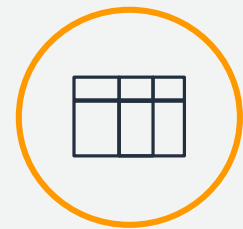
## Time-series

Collect, store, and process data sequenced by time



## Ledger

Complete, immutable, and verifiable history of all changes to application data



## Wide Column

Scalable, highly available, and managed Apache Cassandra-compatible service

AWS Service(s)



Aurora RDS



DynamoDB



DocumentDB



ElastiCache



Neptune



Timestream



QLDB



Keyspaces

Common Use Cases

Lift and shift, ERP, CRM, finance

Real-time bidding, shopping cart, social, product catalog, customer preferences

Content management, personalization, mobile

Leaderboards, real-time analytics, caching

Fraud detection, social networking, recommendation engine

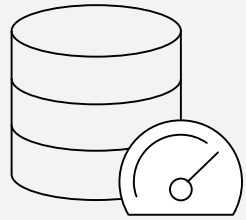
IoT applications, event tracking

Systems of record, supply chain, health care, registrations, financial

Build low-latency applications, leverage open source, migrate Cassandra to the cloud

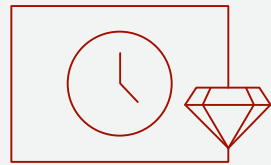
# Amazon Aurora: 오픈소스와 호환되는, 상용 수준 데이터베이스

MySQL과 PostgreSQL에 호환되는 클라우드 기반의 RDBMS로, 상용 데이터베이스 대비 1/10 비용으로 비용 절감과 성능에 최적화



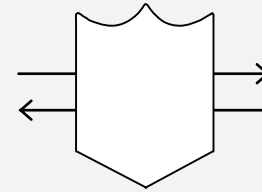
## Performance & scalability

5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas



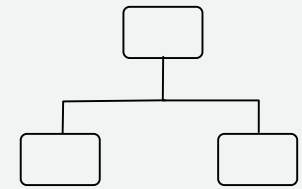
## Availability & durability

Fault-tolerant, self-healing storage; six copies of data across three AZs; continuous backup to S3



## Highly secure

Network isolation, encryption at rest/transit



## Fully managed

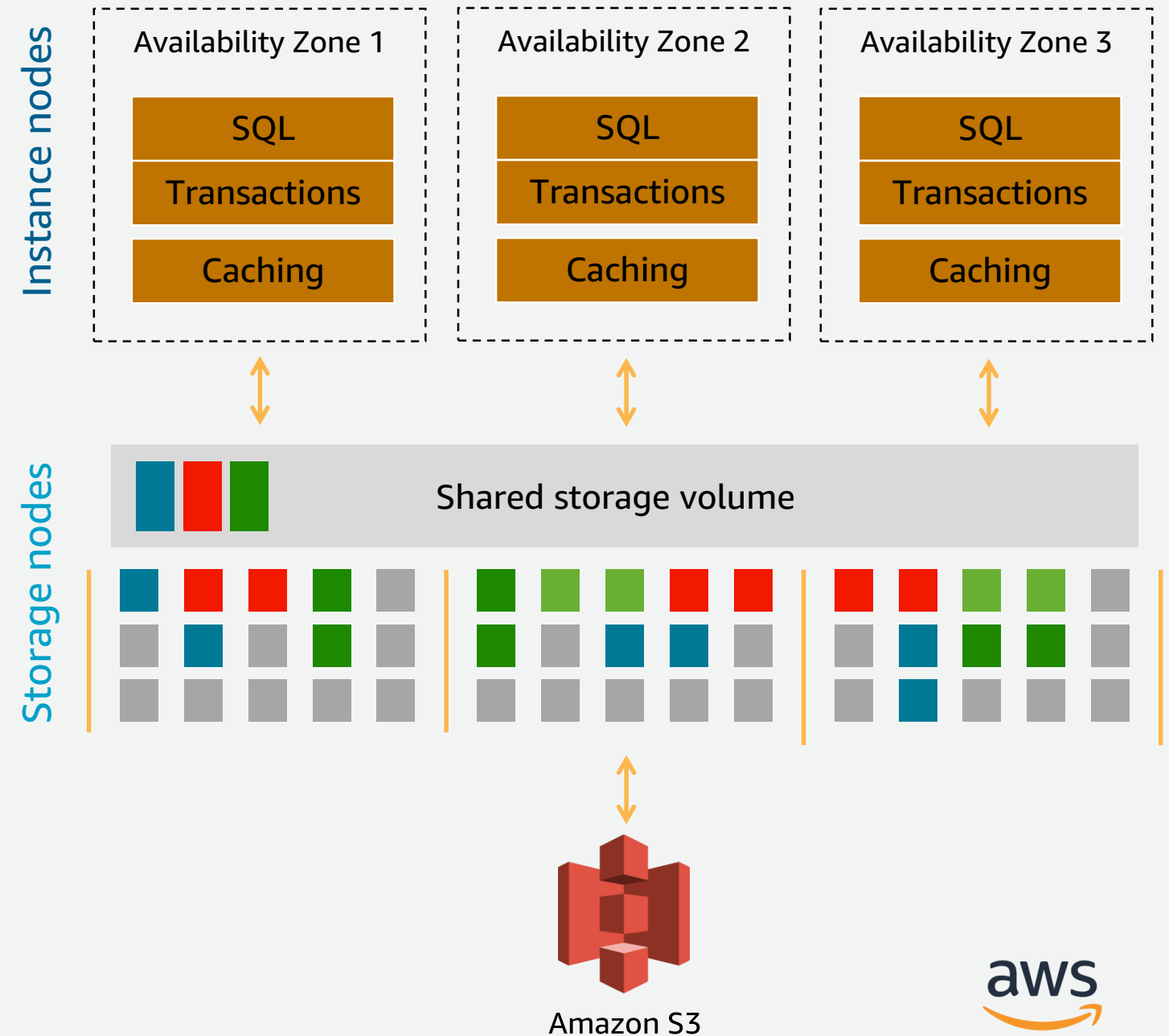
Managed by RDS: no server provisioning, software patching, setup, configuration, or backups

# 확장 가능한 분산 스토리지

데이터 베이스의 목적별 저장을 위한  
스토리지 구조

3개 AZ에 걸친 6개의 데이터 카피로 AZ  
Failure에 대비

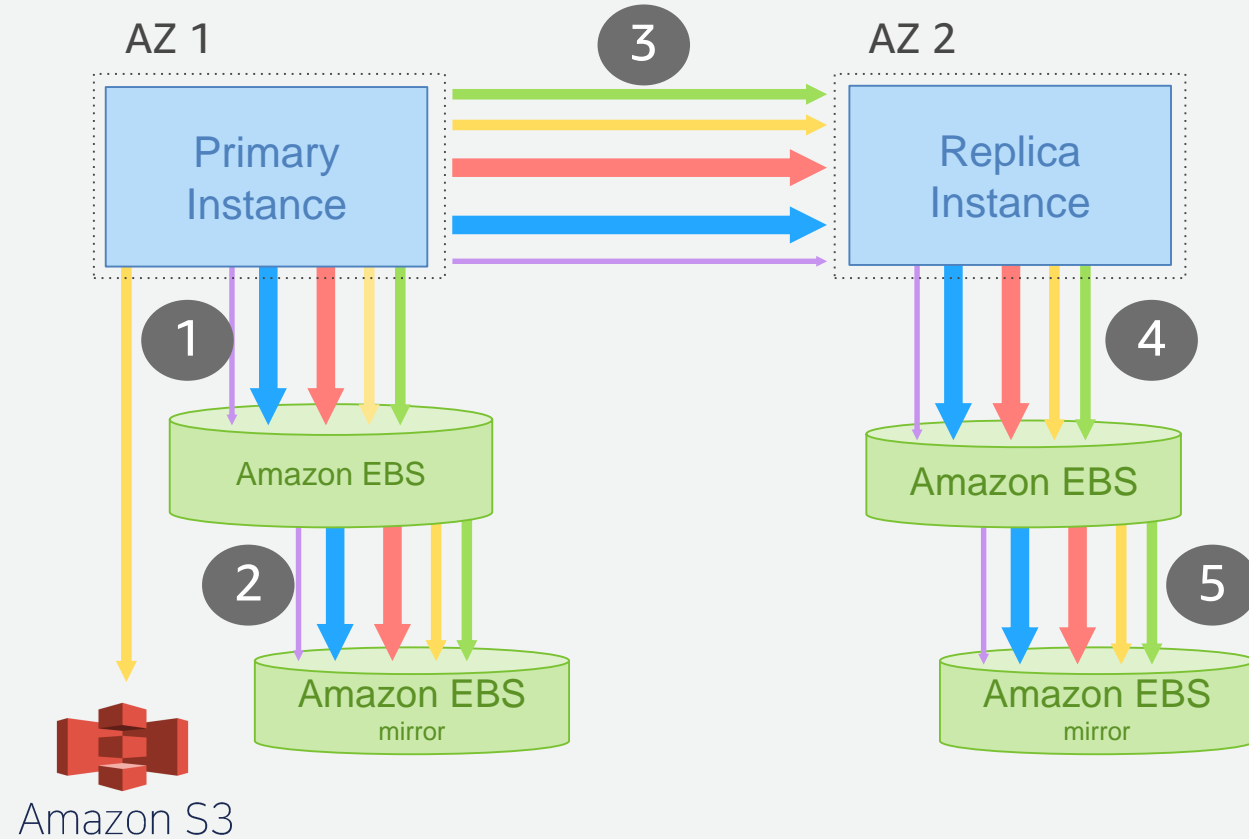
10GB 단위의 스토리지 노드 그룹이  
사용자의 사용량에 따라 자동으로 확장





# Aurora의 I/O 구조

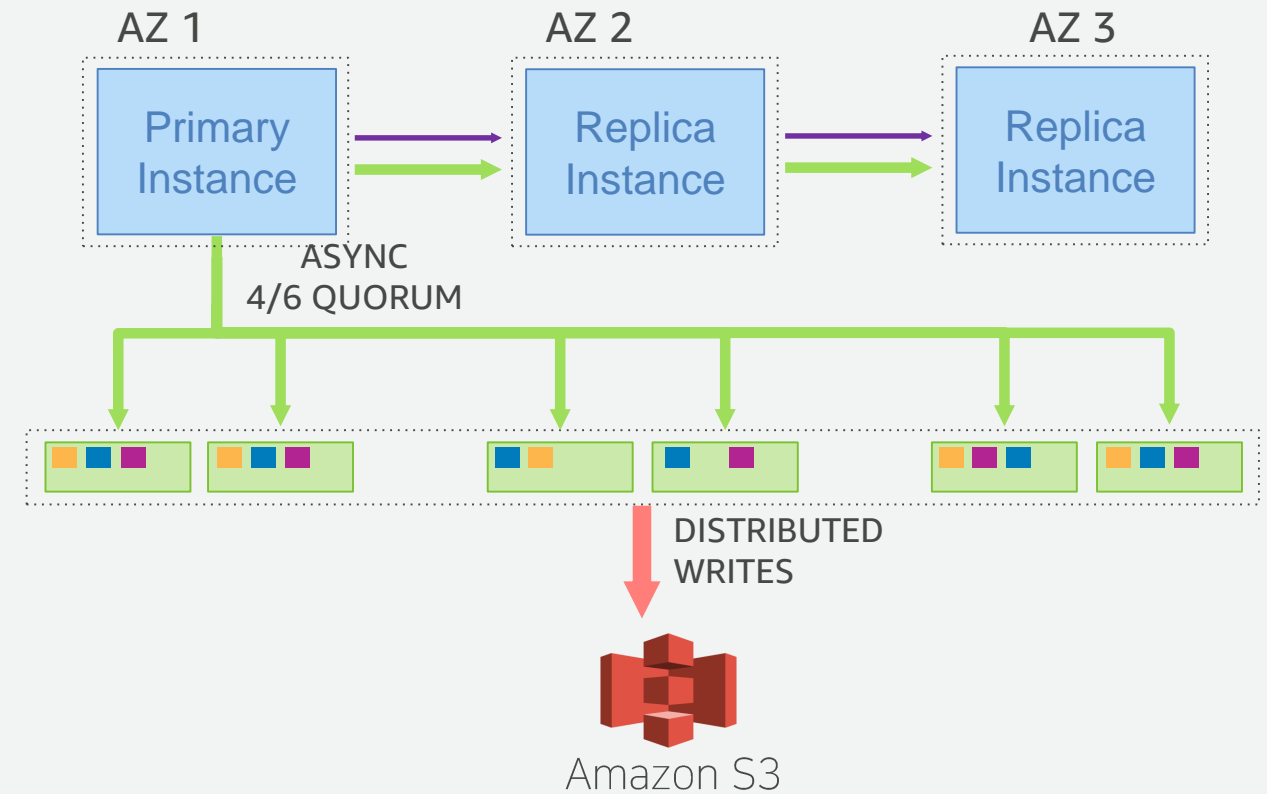
## MYSQL WITH REPLICA



### MySQL I/O profile for 30 min Sysbench run

78만 트랜잭션  
평균 1 트랜잭션 당 7.4 I/O

## AMAZON AURORA



### Aurora I/O profile for 30 min Sysbench run

2,737만 트랜잭션  
평균 1 트랜잭션 당 0.95 I/O

**35X MORE**  
**7.7X LESS**



# HA/DR with Aurora

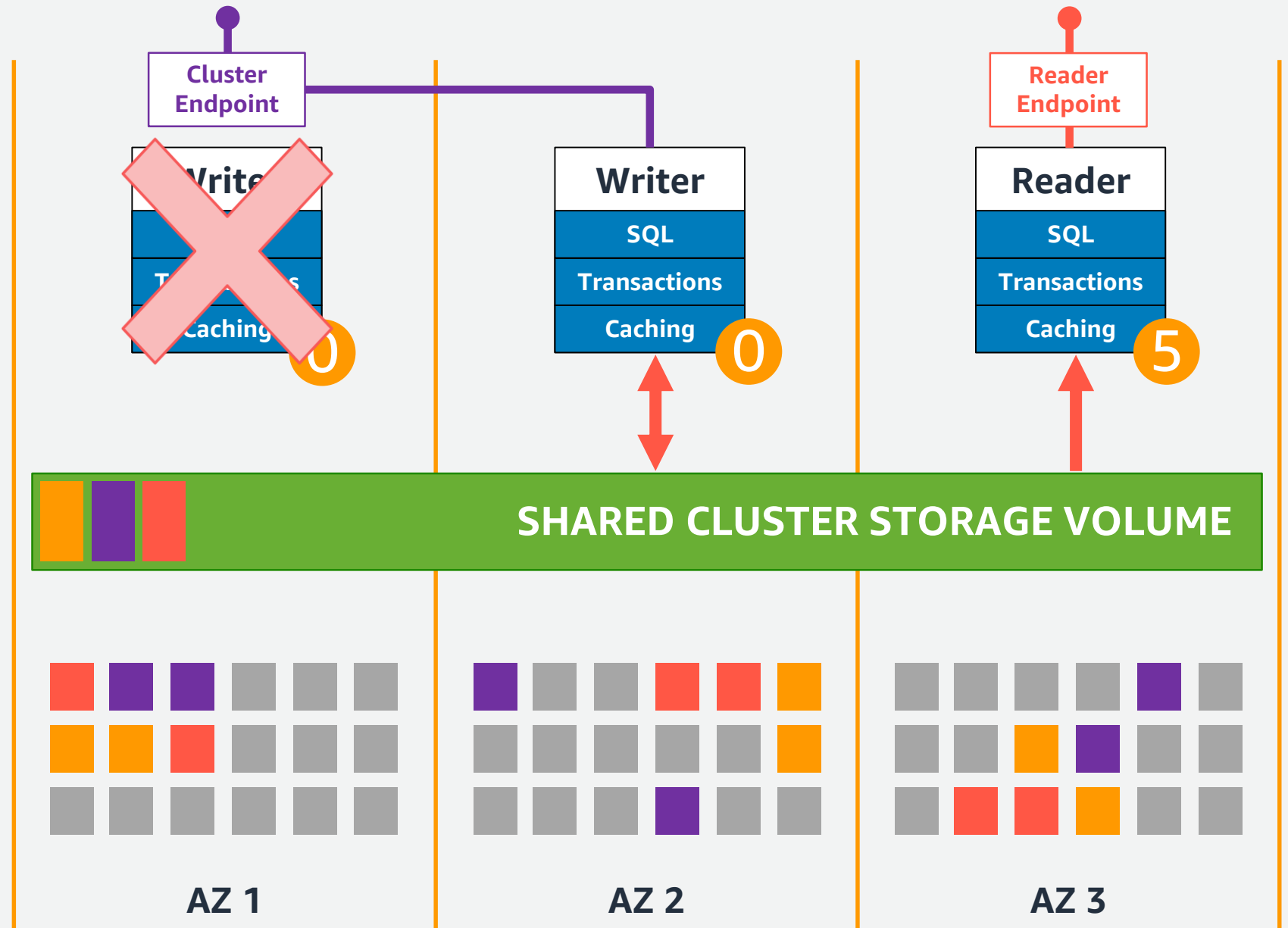
# Failure에 강력한 아키텍처

모든 Reader node는 Writer node로 전환 될 수 있습니다.

**Failover tier** 로 우선 적으로 Failover될 node의 순위를 정할 수 있습니다.

Failure를 겪은 node는 복구 후 Reader로 자동적으로 추가 됩니다.

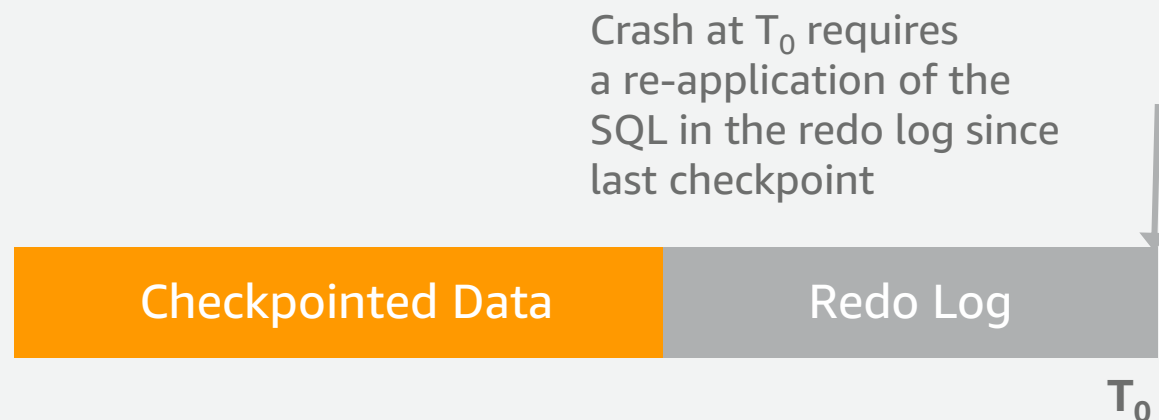
Writer Failover시에 모든 Reader가 재시작 됩니다..



# Near-instant crash recovery

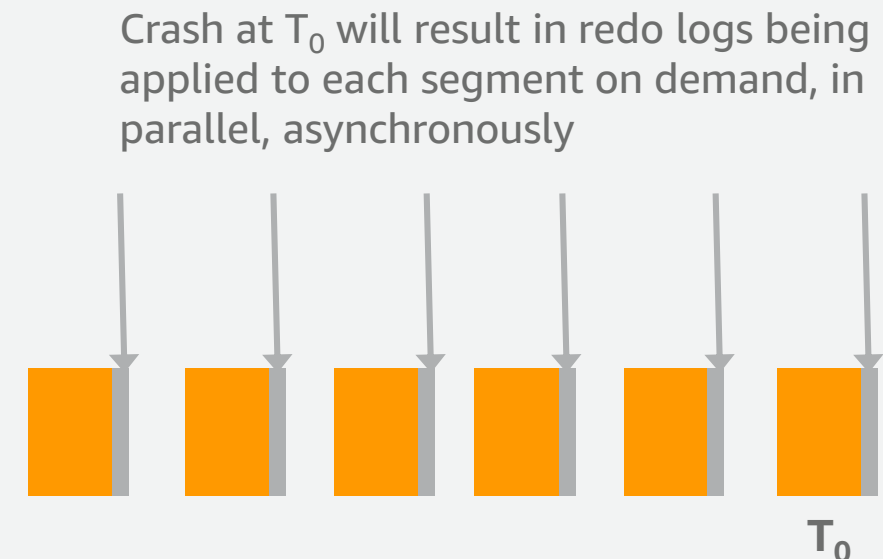
## Traditional database

- 마지막 Checkpoint 기준의 Redo log 보존
- 통상 적으로 5 minutes 간격의 Checkpoint
- MySQL의 Single Thread 복구 특성 상 많은 Disk I/O를 필요로 함



## Amazon Aurora

- 스토리지 레벨에서 redo record를 읽어 복구 진행
- 병렬, 분산, 비동기식 처리
- 인스턴스 재시작 시 replay 불필요

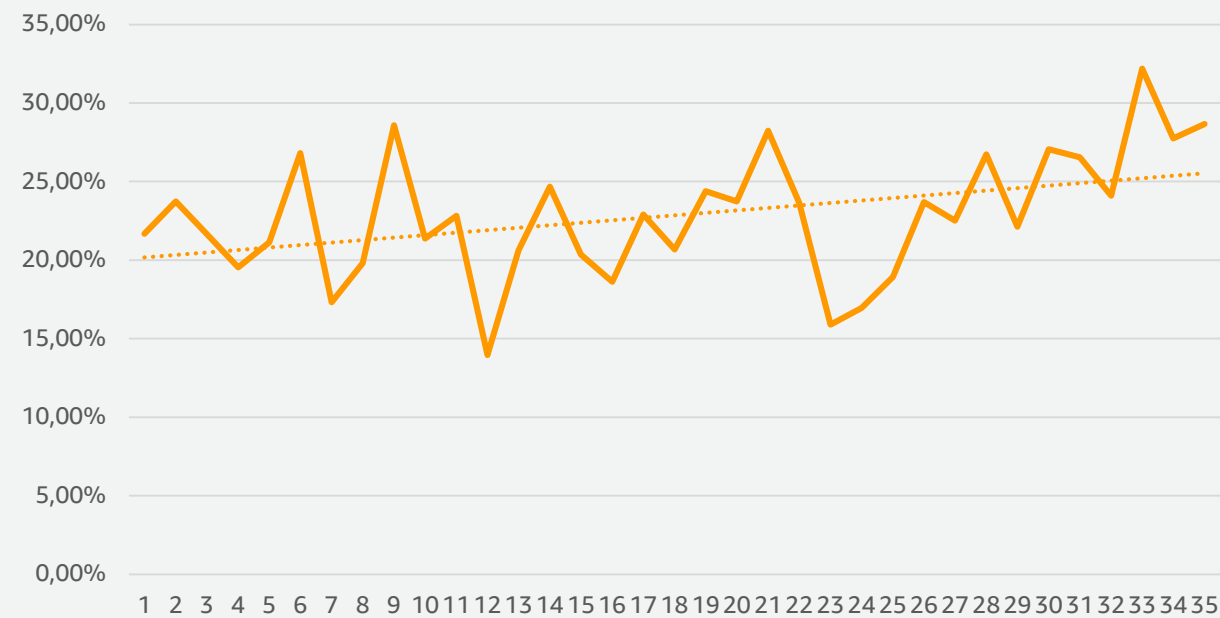


**Note:** Binlogs (master), undo logs와 같은 MySQL Engine 레벨의 구성 요소는 Engine 레벨의 리커버리가 필요합니다.

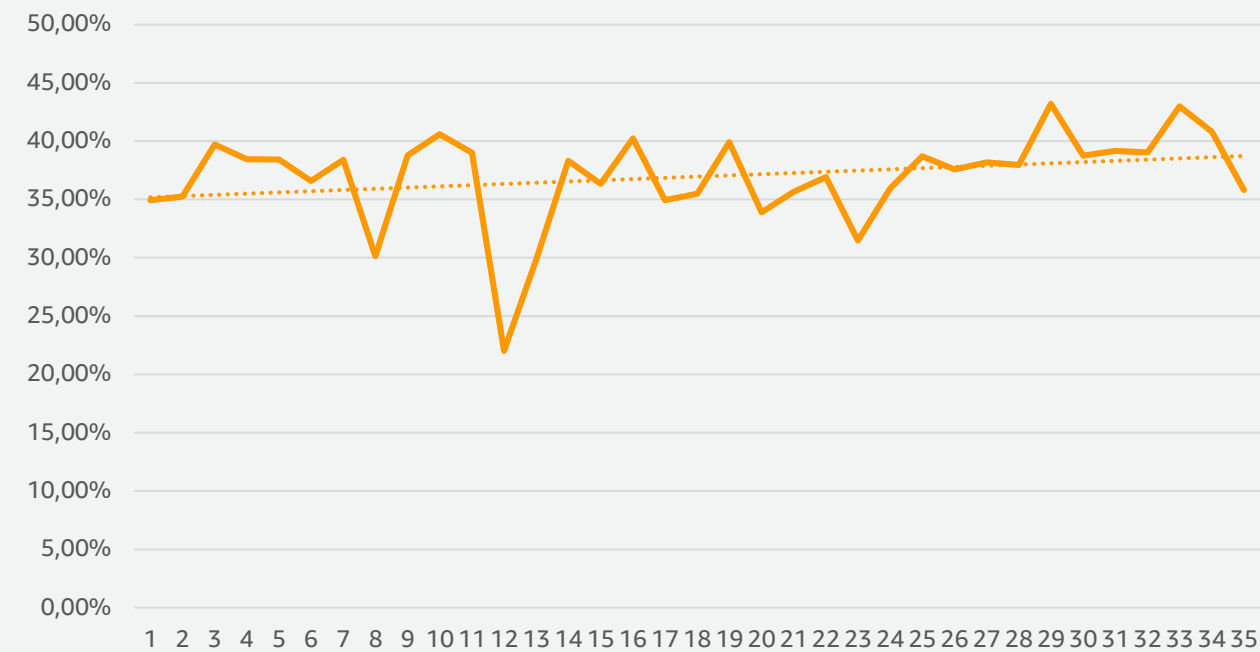


# When the database fails – recovery is fast <30 seconds

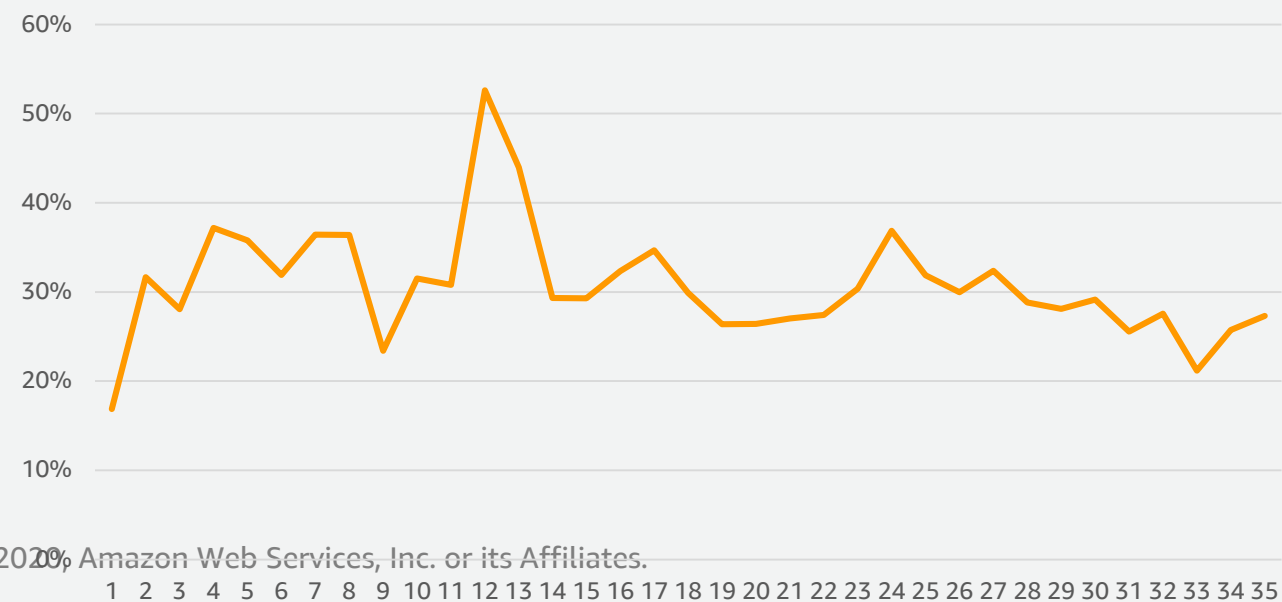
0 - 5s – 30% of fail-overs



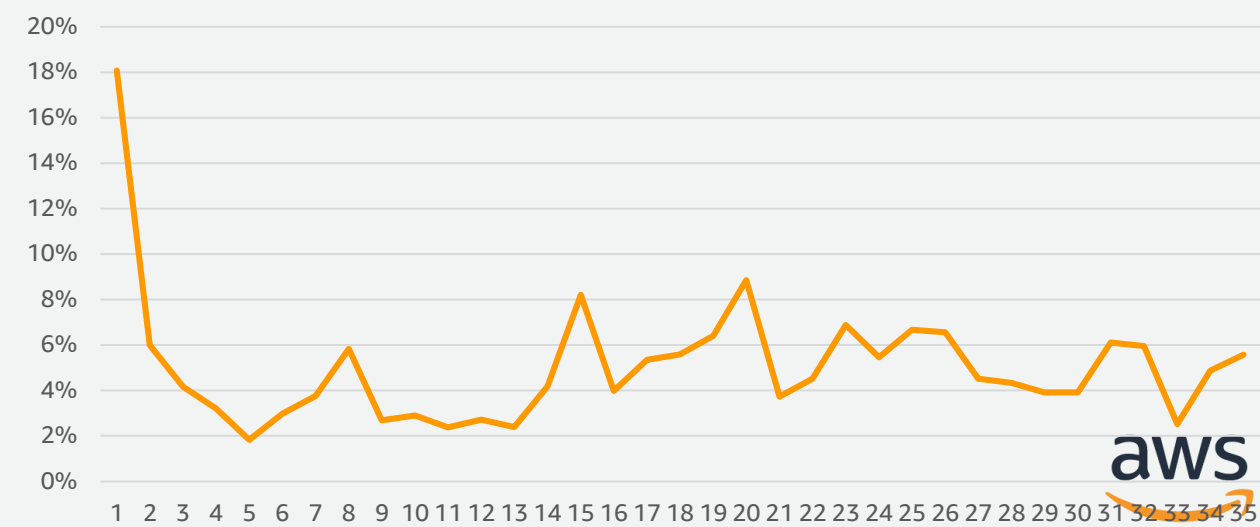
5 - 10s – 40% of fail-overs



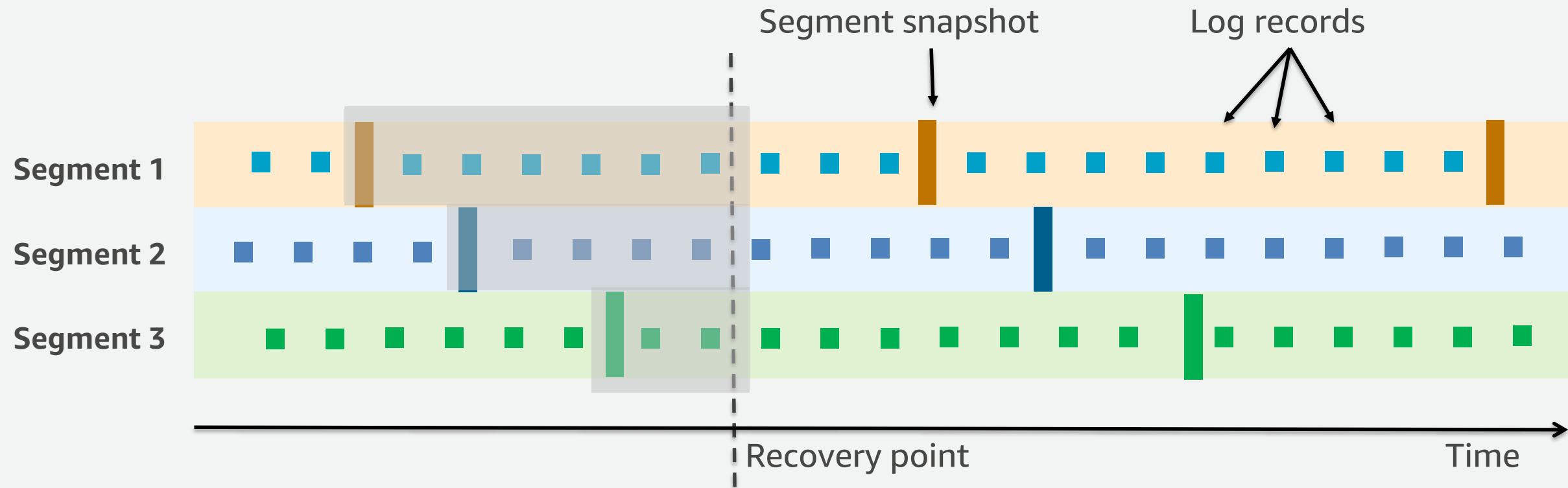
10 - 20s – 25% of fail-overs



20 - 30s – 5% of fail-overs



# Continuous backup with point-in-time restore



- 스토리지 노드에서 실행되는 백업은 성능과 가용성에 영향을 받지 않고 수행됩니다.
- 주기적인 스냅샷은 여러 세그먼트에서 병렬적으로 수행되어 **S3**에 저장됩니다.
- 로그는 **S3**로 지속적으로 전송되어 복구에 활용됩니다.

# Aurora backup and restore

## Automated backups:

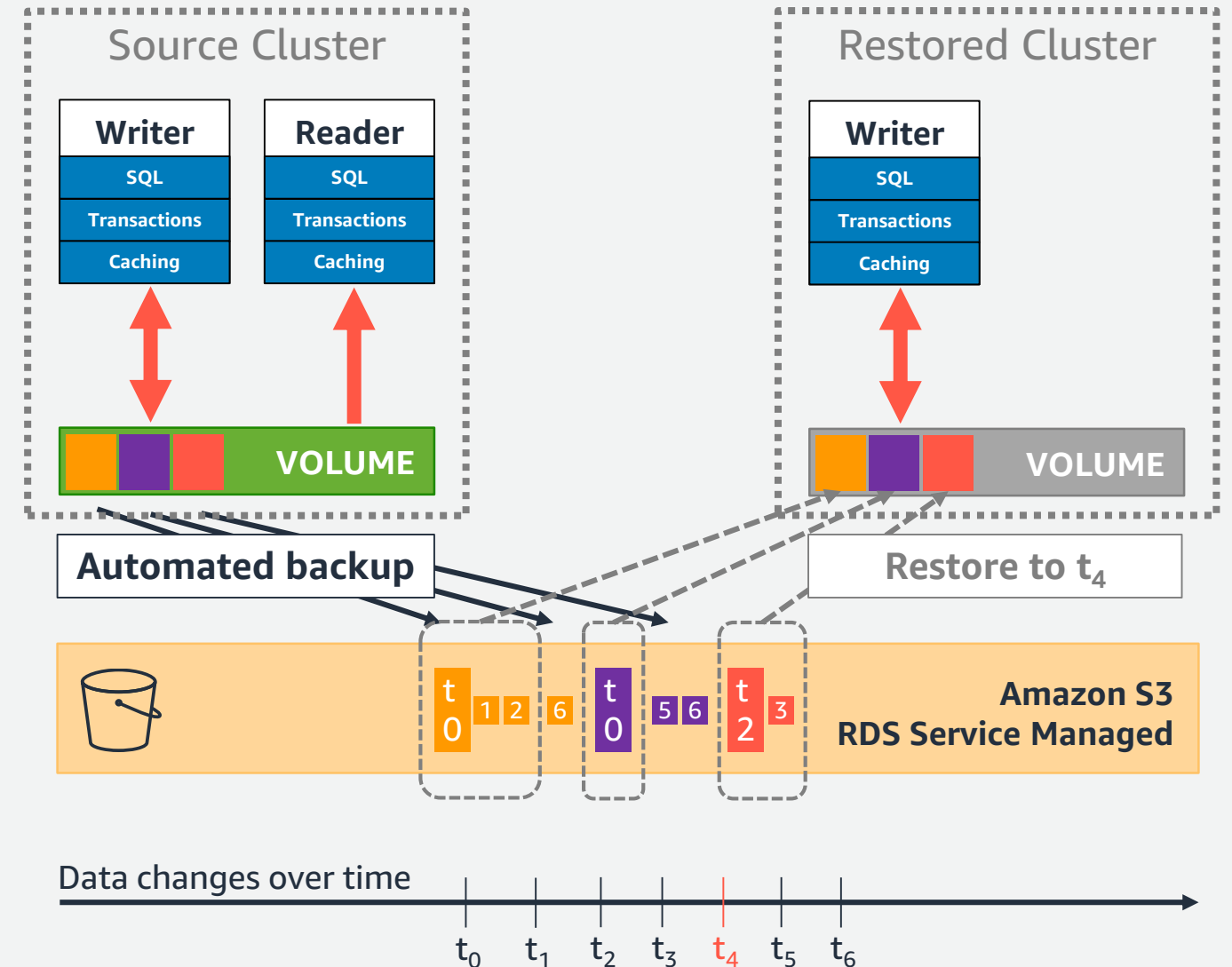
- 1일과 35일 사이의 유지 기간을 설정
- 최소 5분 전 시점의 데이터로 복구 가능

## Snapshots:

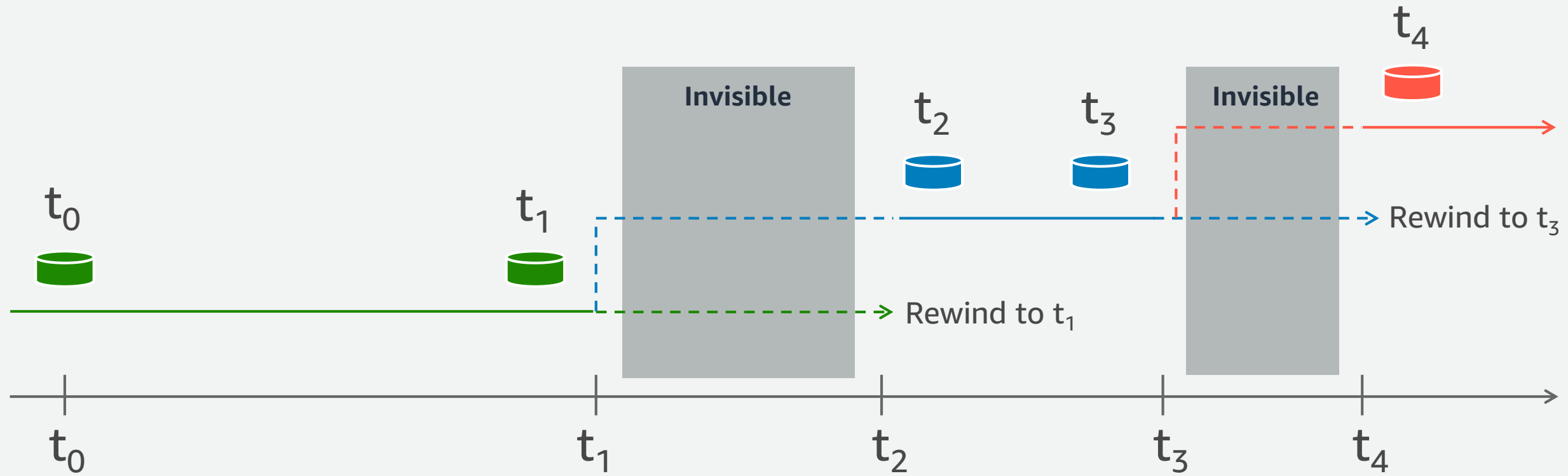
- 유저로부터 생성된 스냅샷
- 성능에 영향 없음
- 다른 지역이나 다른 계정으로 스냅샷 복사 공유 가능

## Restore:

- 클러스터의 스토리지 사용량에 따라 복구 시간 영향
- 새로운 클러스터를 생성하여 데이터 복구



# Aurora Backtrack



**Aurora Backtrack 은 Database의 시점복구 기능을 backup과 restore 를 거치지 않고 복구하는 기능입니다.**

예상치 못한 DML과 DDL로 인한 데이터 유실이나 변조 시 복구에 사용

Backtrack을 여러 번 수행하여 적절한 시점을 찾는 사용 사례

QA에서 테스트 수행 후 테스트 전으로 데이터를 돌려 다시 테스트를 수행하는데 사용하는 사례



# HA&DR: from one region to multiple regions...

## Cross-region read replicas

논리적으로는 Binlog Replication

최대 5개의 Readable Replica 셋 생성

RDS 서비스에 의해 관리됨

## Aurora Global Database

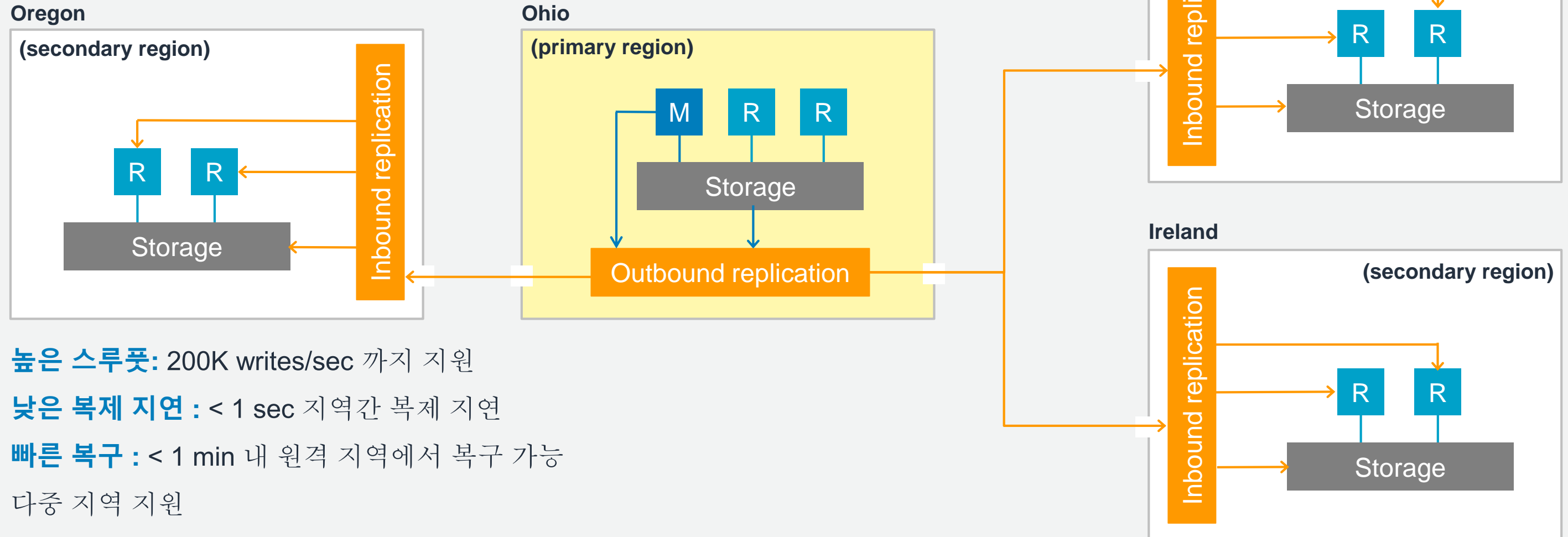
물리적 redo log 복제

최대 16개의 Reader Instance

스토리지 계층의 복제로 성능 영향 없음

# Aurora Global Database

더 빠른 데이터 복구를 위한 솔루션



높은 스루풋: 200K writes/sec 까지 지원

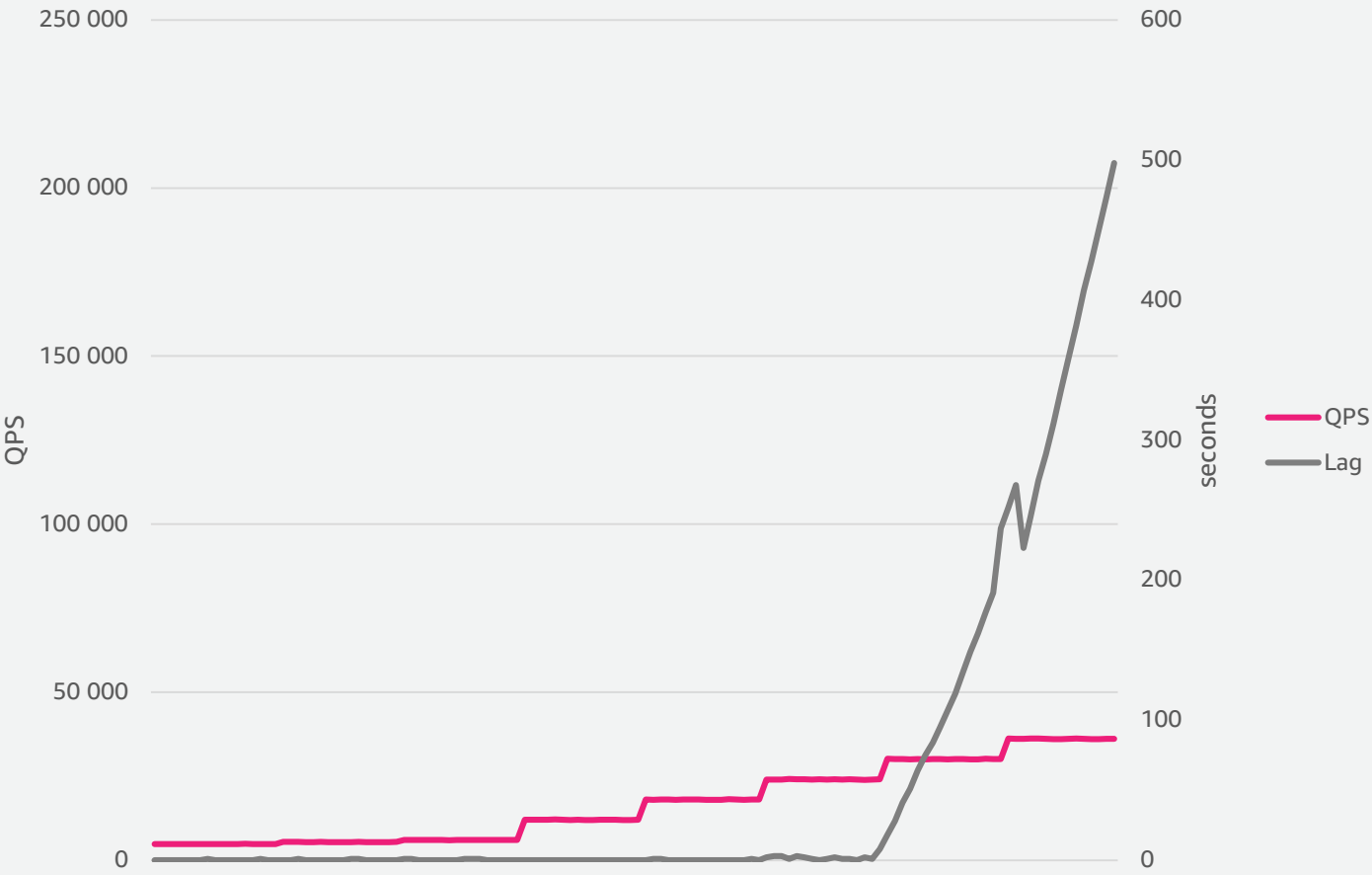
낮은 복제 지연: < 1 sec 지역간 복제 지연

빠른 복구: < 1 min 내 원격 지역에서 복구 가능

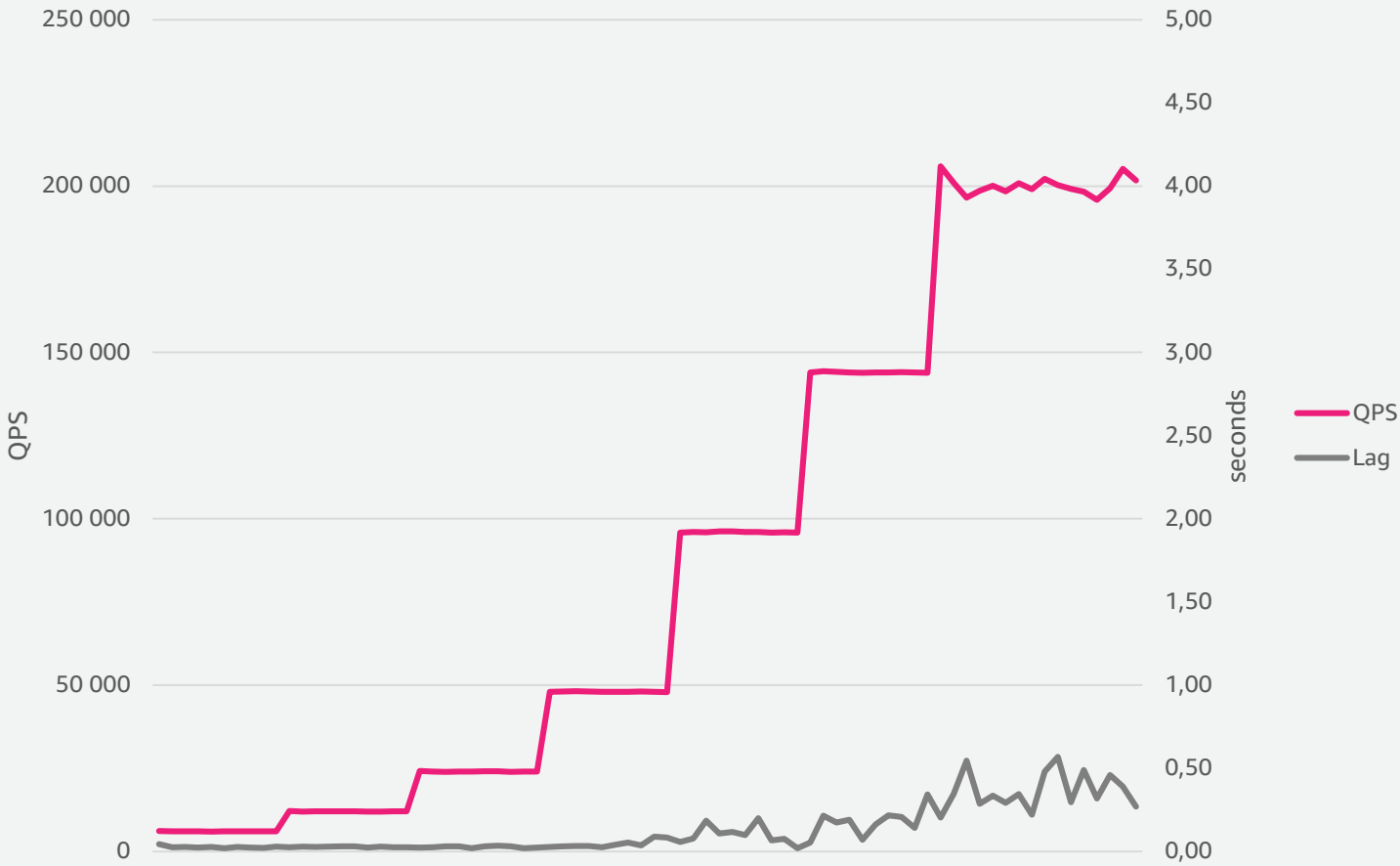
다중 지역 지원



# Performance: logical vs. physical replication



**Logical replication**  
**MySQL Binary logs (MTS)**



**Physical replication**  
**Global Database**

SysBench OLTP (write-only) stepped every 600 seconds on R4.16xlarge



# HA: 다운타임 최소화

## Aurora Multi-Master

일관성이 유지되는 다중 쓰기 인스턴스  
단일 **Writer** 실패 시 쓰기 워크로드 담당

## Efficient Connection Management

RDS Proxy

Failover optimization and testing

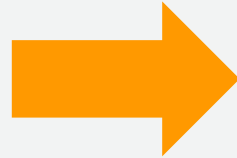
# Continuous availability with Aurora Multi-Master

가용성 향상



스토리지를 공유하는 여러 쓰기 인스턴스

고 내구성



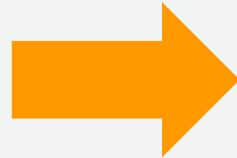
3개 AZ에 걸친 6개의 복제본

SQL 호환성



indexes, constraints, triggers, procedures, functions 등 MySQL 호환성

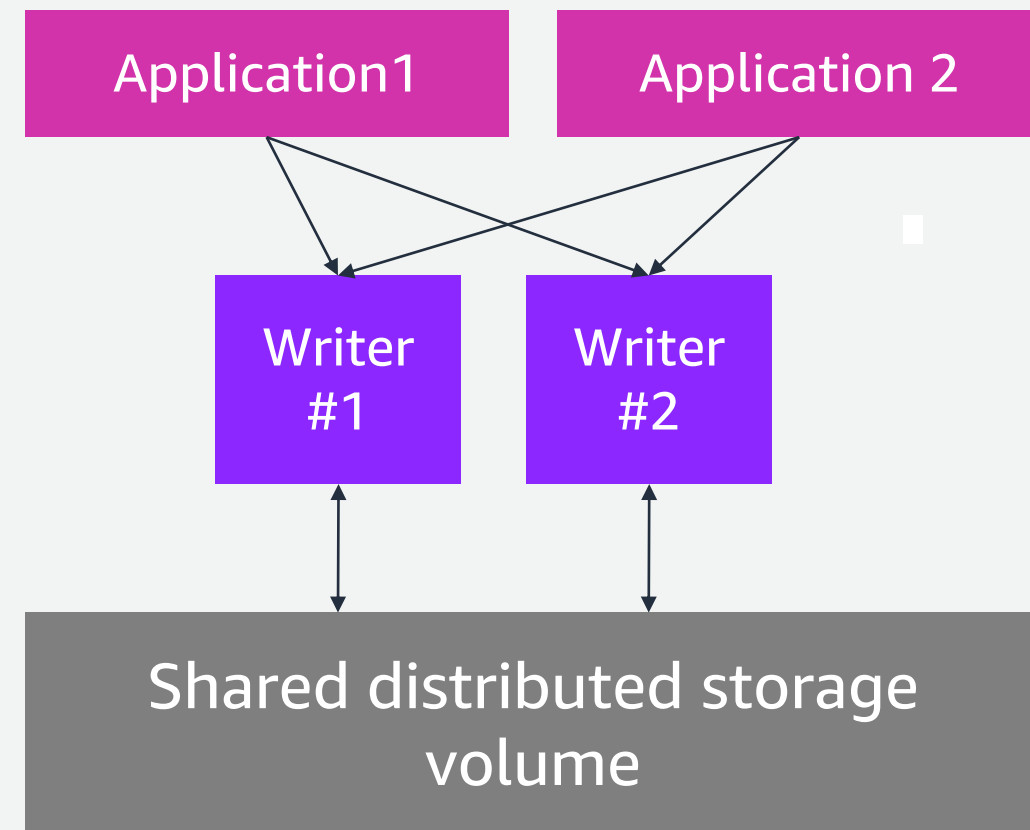
쓰기 분산



Storage Node 주도의 일관성 관리

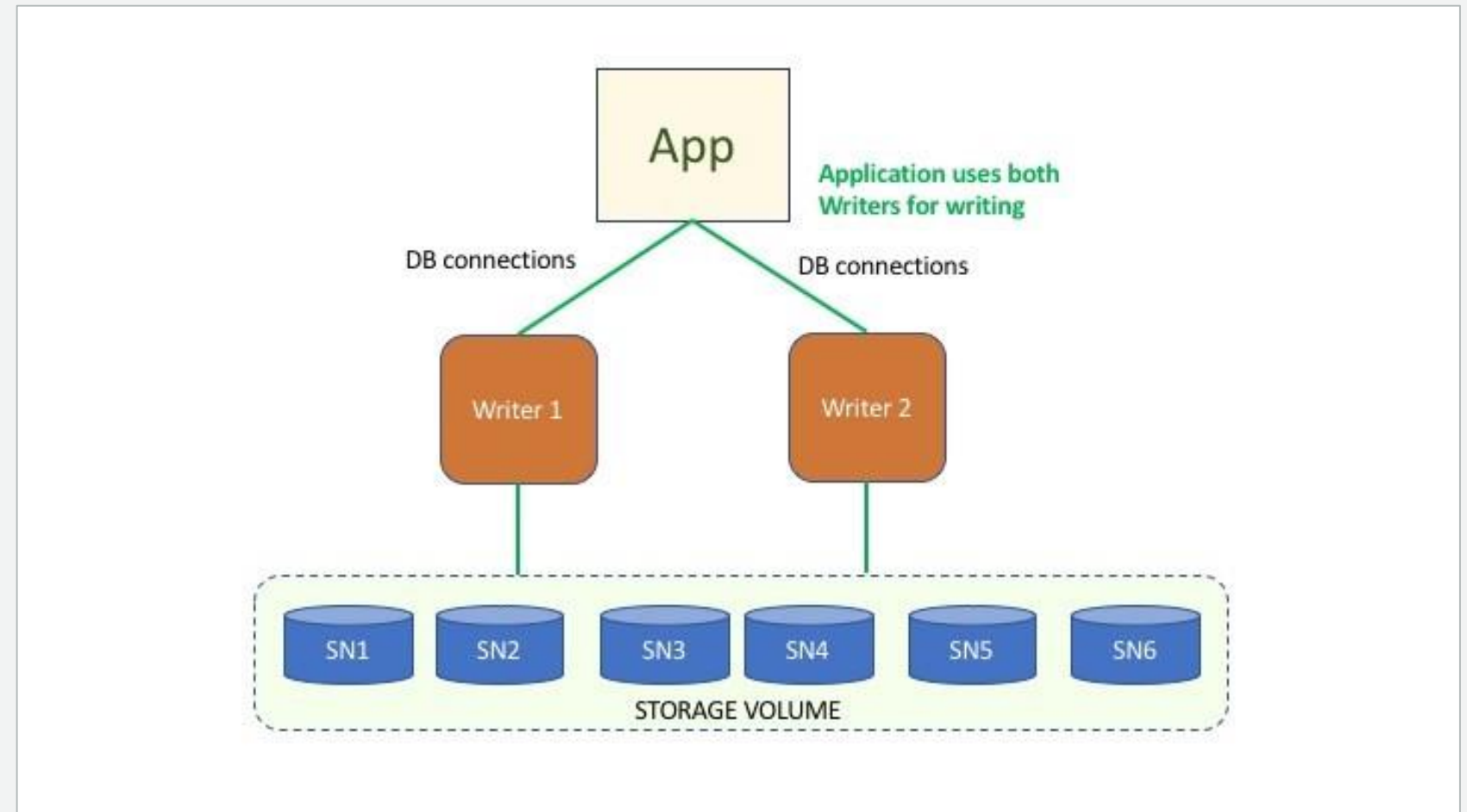
# Introduction to Aurora Multi-Master

- 여러 Node가 쓰기 워크로드 수행
- 역시 분산 스토리지 볼륨을 공유
- 각 Node의 복구는 각자 진행 됨
- 추가비용 없음



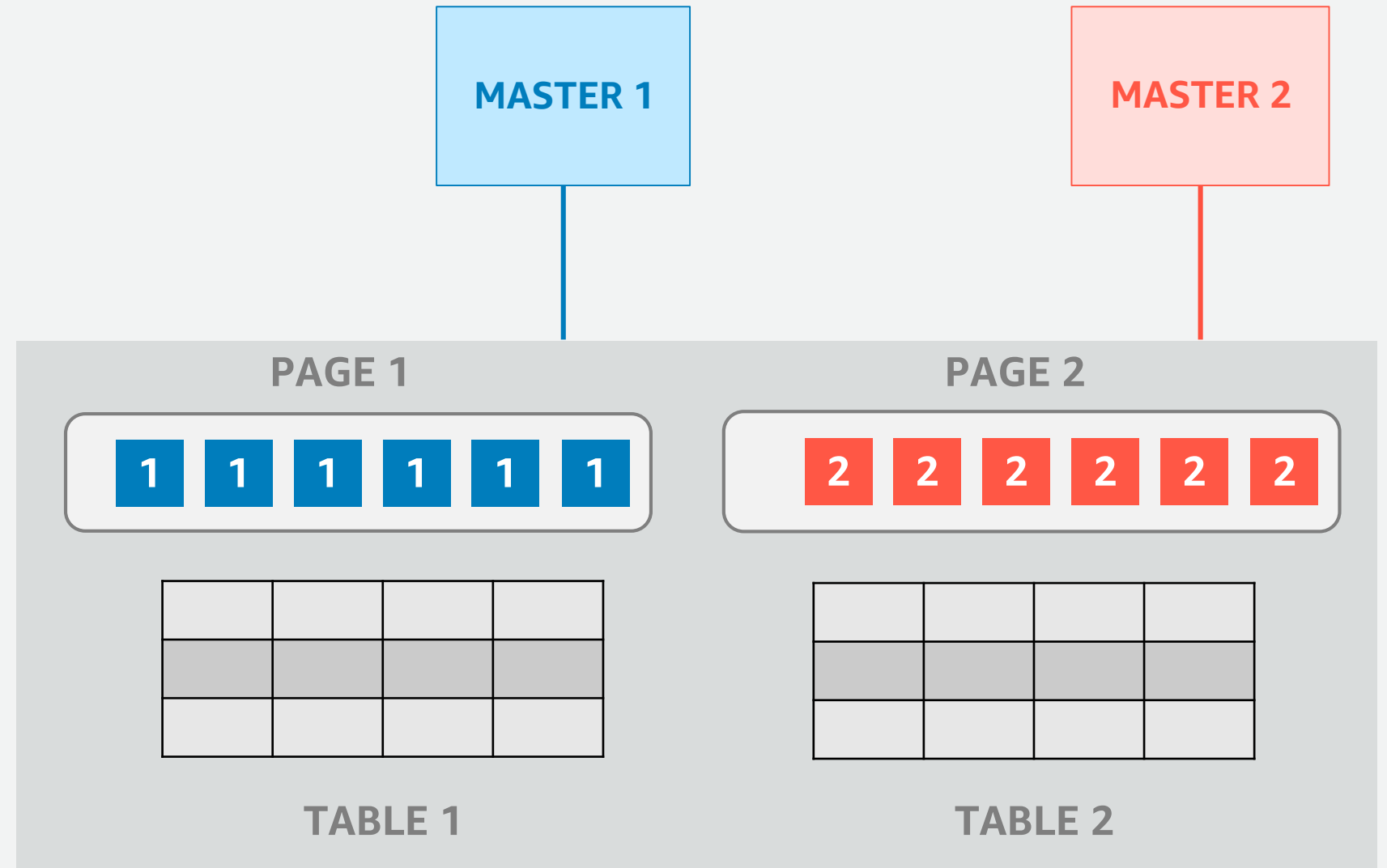
# Using Aurora Multi-Master with your applications

- 최초 Application에 쓰기 인스턴스 할당
- 어플리케이션에서 쓰기 Node의 Health 상태 점검
- 인스턴스 실패 시 다른 쓰기 Node로 커넥션 재 생성



# Multi-Master: example of transactions with no conflict

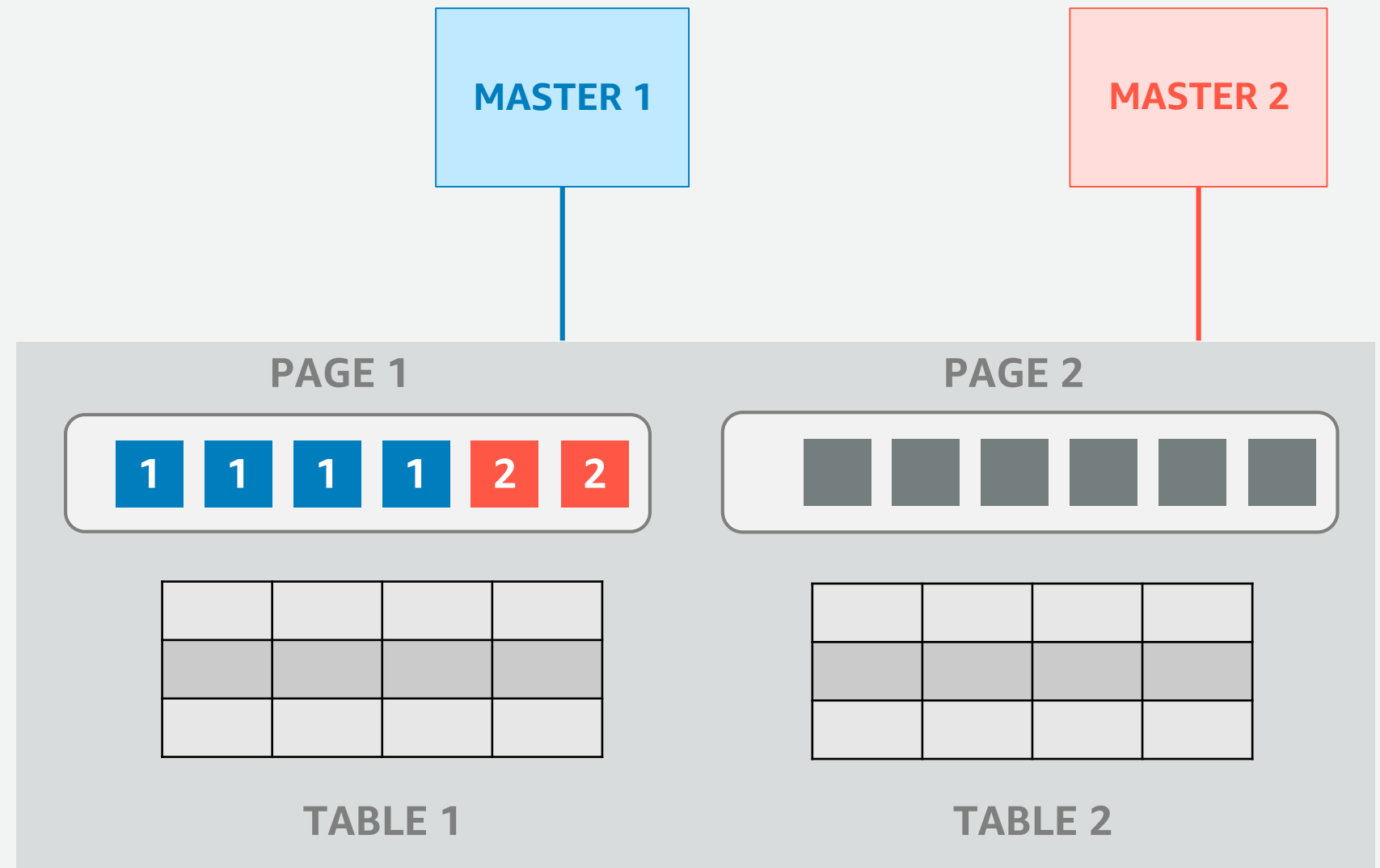
- 트랜잭션 **T1** 과 **T2** 가 각기 다른 테이블을 Table 1 과 Table 2 를 업데이트
- 논리적 물리적 충돌이 없기 때문에 문제 없이 수행





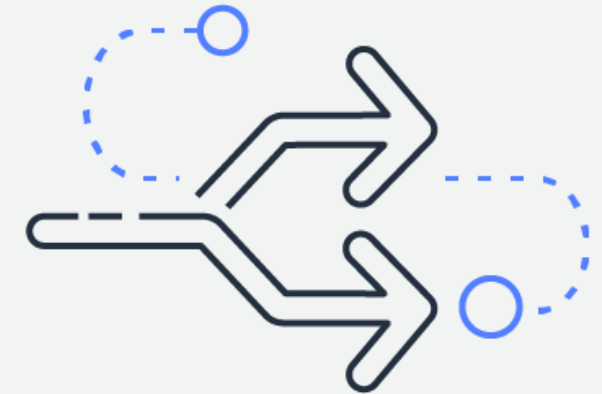
# Multi-Master: example of physical conflict (storage arbitration)

- 트랜잭션 **T1** 과 **T2** 가 같은 테이블 Table 1 을 수정
- 같은 Page의 데이터 끼리 충돌 발생 가능
- 트랜잭션 **T1** 은 4/6 쿼럼을 충족
- 트랜잭션 **T2** 는 쿼럼을 충족 하지 못하여 쓰기 실패하고 롤백 및 어플리케이션에 에러 전달
- 스토리지 노드가 충돌 감시자 역할을 수행



# Connection management to minimize disruptions

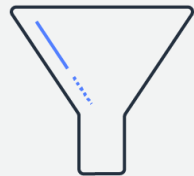
- 1개 Node당 16,000 개 커넥션 생성 가능(readers or writers)
- 읽기 쓰기 분산 필요
- 커넥션 풀링을 사용하여 안정적인 커넥션 관리
- DNS TTL을 적절히 설정
- 장시간 수행 및 대기 쿼리를 제거



# Improving HA with RDS Proxy

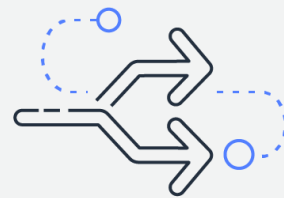
Amazon RDS 를 위한 완전 관리형,고가용성 Proxy 기능

## Connection Pooling



Pool and share DB connections for improved app scaling

## Minimize Failover Disruptions



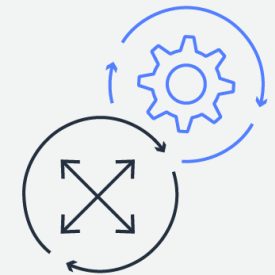
Consistent endpoint, holds connections open and queue up requests, minimizes failover time

## Improved Security



Encryption in transit and improved DB access controls

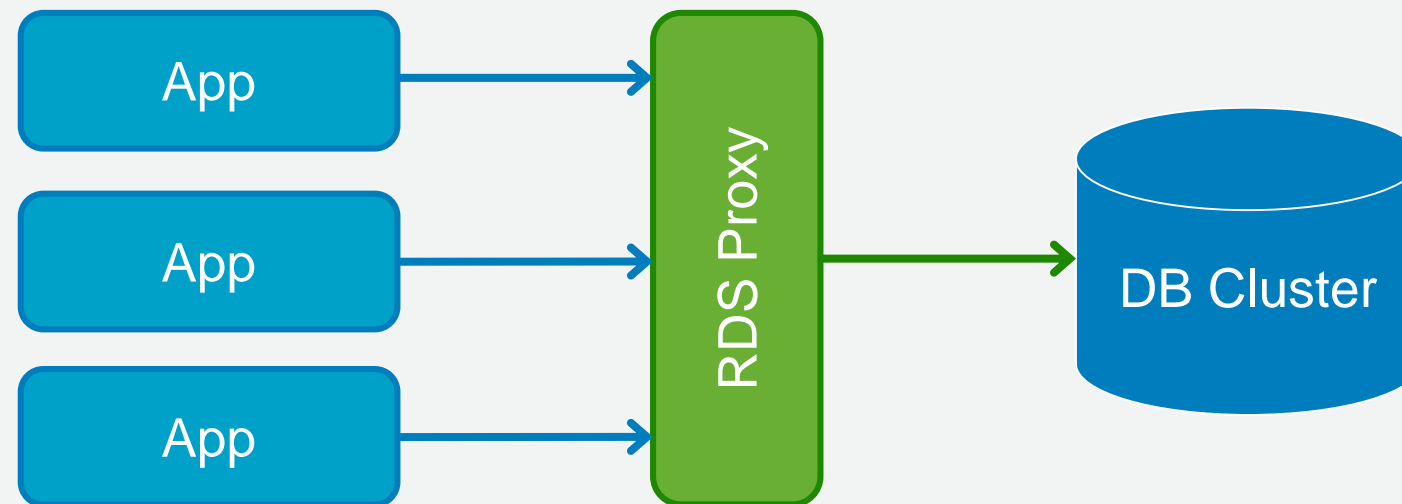
## Fully Managed and Scalable



Delivered as a managed service compatible with RDS databases

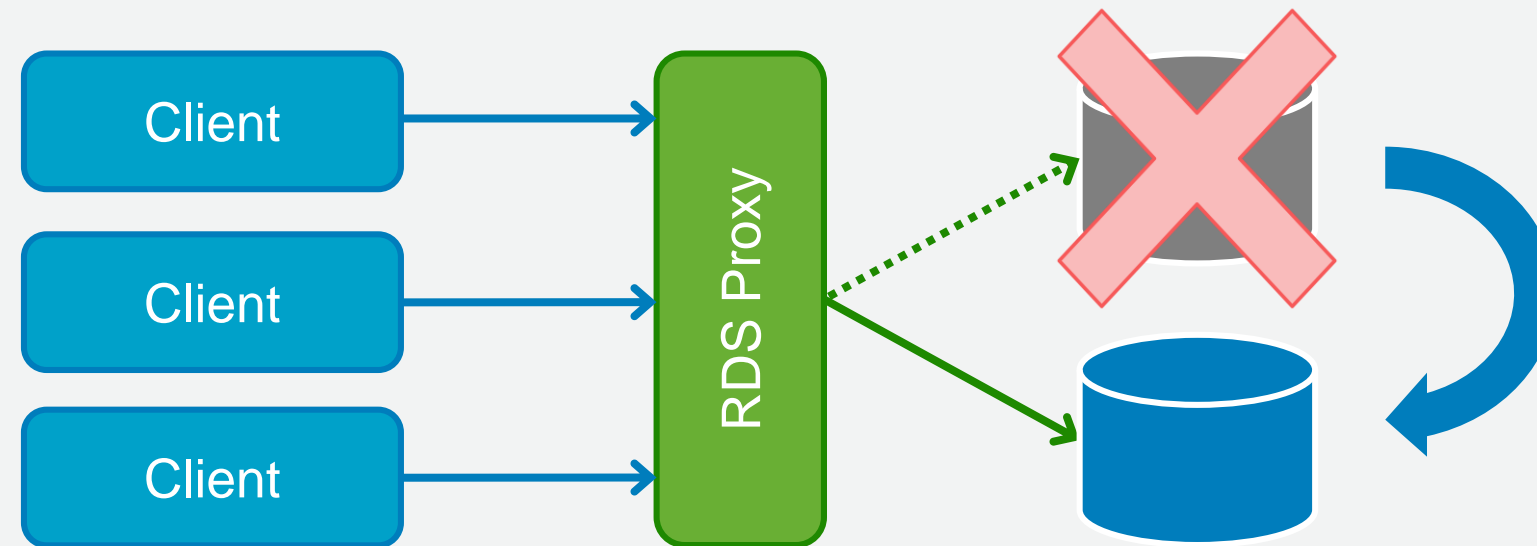
# Connection Pooling with RDS Proxy

- 어플리케이션과 데이터베이스 레벨의 커넥션을 분리하여, 어플리케이션 스케일시에도 데이터베이스에 과도한 커넥션이 발생하지 않도록 방지
- 커넥션 당 트랜잭션을 다중화 하여, 커넥션 재사용으로 인한 성능 향상
- 커넥션들에 공통된 설정값들을 유지하여, 일관된 성능 유지



# Minimize Failover Disruption with RDS Proxy

- 어플리케이션 레벨의 커넥션 풀은 유지하여, 더 빠르게 커넥션 복구
- Active Connection을 먼저 끊어 클라이언트 레벨에서 불완전 처리에 대한 에러 리턴
- 새로운 요청은 Proxy레벨에서 보관되어 추후 수행









# Aurora Backtrack Demo

# Summary

- 다양한 요구사항 다양한 데이터베이스!
- 강력한 HA/DR을 제공하는 오로라
- 스토리지 레벨의 빠른 병렬 복구
- Backtrack 기능을 활용한 데이터베이스 간편 복구
- 스토리지 레벨의 복제로 더욱 빠른 다지역 복제 기능 제공
- RDS Proxy를 활용한 더욱 편한 연결 관리

# AWS Database Modernization Day에 참석해주셔서 대단히 감사합니다.

저희가 준비한 내용, 어떻게 보셨나요?  
더 나은 세미나를 위하여 **설문을 꼭 작성해 주시기 바랍니다.**

-  [aws-korea-marketing@amazon.com](mailto:aws-korea-marketing@amazon.com)
-  [twitter.com/AWSKorea](https://twitter.com/AWSKorea)
-  [facebook.com/amazonwebservices.ko](https://facebook.com/amazonwebservices.ko)
-  [youtube.com/user/AWSKorea](https://youtube.com/user/AWSKorea)
-  [slideshare.net/awskorea](https://slideshare.net/awskorea)
-  [twitch.tv/aws](https://twitch.tv/aws)



# Thank you!