



# MySQL InnoDB Cluster

---

류수미

MySQL Principal Solution Engineer

MySQL GBU, JAPAC

Sep, 2021

## Safe Harbor

---

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Statements in this presentation relating to Oracle's future plans, expectations, beliefs, intentions and prospects are "forward-looking statements" and are subject to material risks and uncertainties. A detailed discussion of these factors and other risks that affect our business is contained in Oracle's Securities and Exchange Commission (SEC) filings, including our most recent reports on Form 10-K and Form 10-Q under the heading "Risk Factors." These filings are available on the SEC's website or on Oracle's website at <http://www.oracle.com/investor>. All information in this presentation is current as of September 2019 and Oracle undertakes no duty to update any statement in light of new information or future events.

# Agenda

- MySQL HA 솔루션
- InnoDB Cluster 소개
- InnoDB Cluster의 구성
- InnoDB Cluster의 consistency Level
- InnoDB Cluster의 일반적인 설정
- InnoDB Cluster ReplicaSet

# MySQL High Availability 솔루션

| Solution                             | Native | Notes  |
|--------------------------------------|--------|--|
| MySQL Replication (단방향)              | ✓      | 수동으로 애플리케이션 스위치<br>비동기방식                                     |
| MySQL Replication (양방향)              | ✓      | 수동으로 애플리케이션 스위치<br>비동기방식<br>데이터 일관성 문제                       |
| MySQL Replicaset                     | ✓      | 장애 시 수동으로 노드 스위치   |
| Operating System Clustering Solution | ✗      | 서드파티 툴 (해당 툴부분은 기술지원 불가,<br>추가적인 라이선스 비용)<br>Active/Stand-by |
| Virtualization/Container Restart     | ✗      | 서드파티 툴 (해당 툴부분은 기술지원 불가,<br>추가적인 라이선스 비용)<br>Active/Stand-by |
| Group Replication + MySQL Router*    | ✓      | 수동 디플로이 및 관리<br>*Router는 서드파티 툴로 변경가능                        |
| MySQL InnoDB Cluster                 | ✓      | 전체 자동화된 솔루션  |
| MySQL Cluster Carrier Grade Edition  | ✓      | 전체 자동화된 솔루션<br>InnoDB 대신 NDB 기반                              |

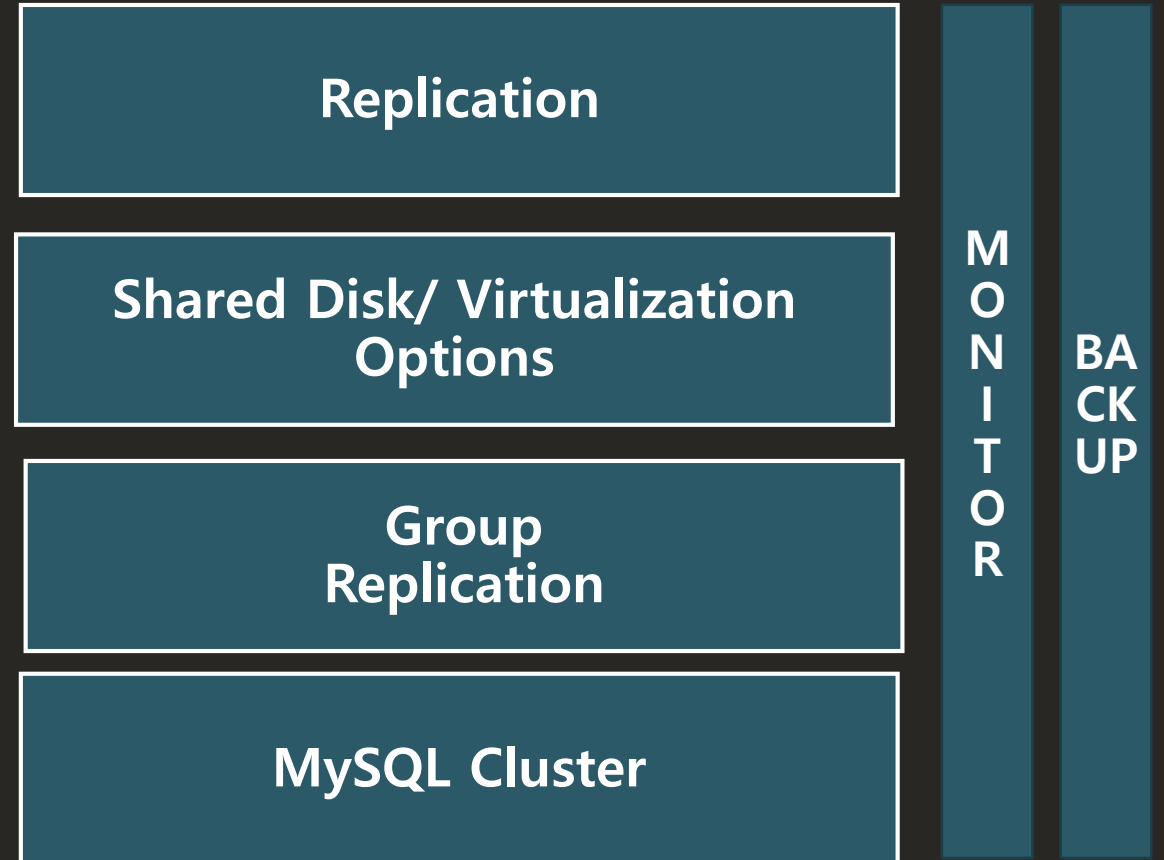
# High Availability 솔루션

전체 솔루션 기술 지원  
(서드파티 제외)

- MySQL Enterprise Monitor
- 24x7 Premier Support

전체 컴포넌트를 오라클  
MySQL 엔지니어링팀에서  
개발

- MySQL Router
- Group Replication
- MySQL Replication
- MySQL Cluster



Replication

Group  
Replication

Shared Disk/  
Virtualization  
Options

MySQL  
Cluster

## Replication

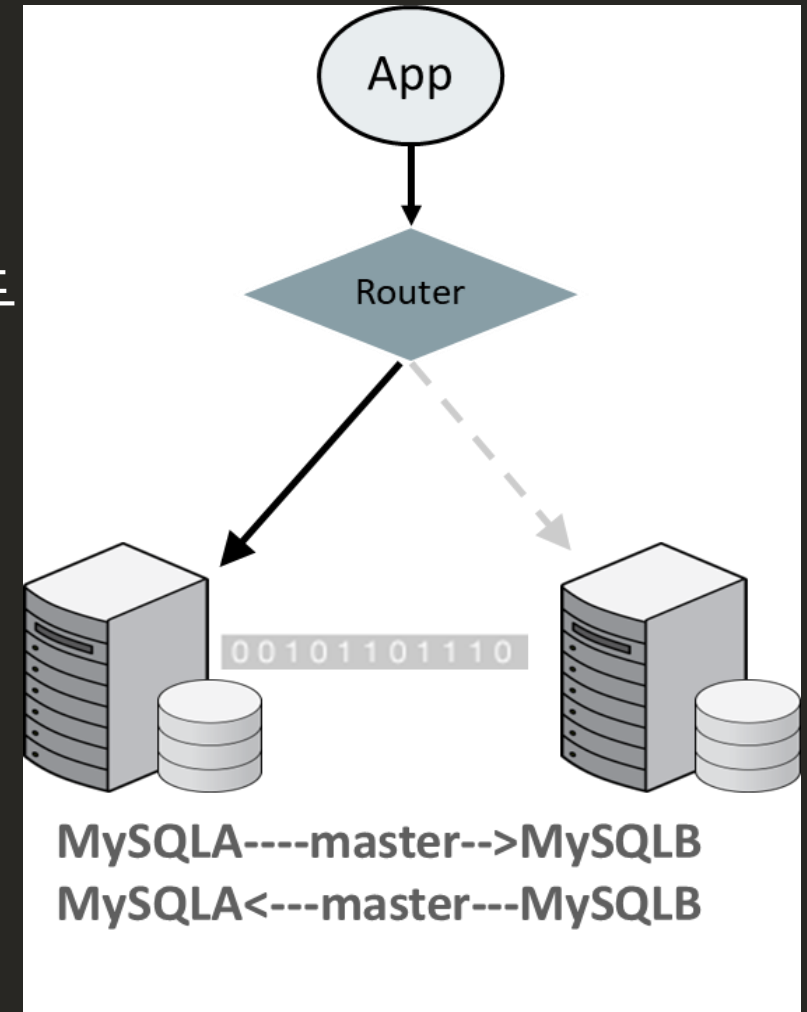
- 비동기, 반동기 방식
- 쓰기는 마스터로, 읽기는 마스트/슬레이브 모두
- Global Transaction Identifier

## 자동 페일오버

- MySQL Utilities ( mysqlfailover )
- 서드파티 솔루션 이용

참고 : MySQL 8.0.22 부터 `START SLAVE` 는 디프리케이트되고, 대신 `START REPLICA` 로 사용하셔야 합니다.

<https://mysqlhighavailability.com/mysql-terminology-updates/>



Replication

Shared Disk/  
Virtualization  
Options

Group  
Replication

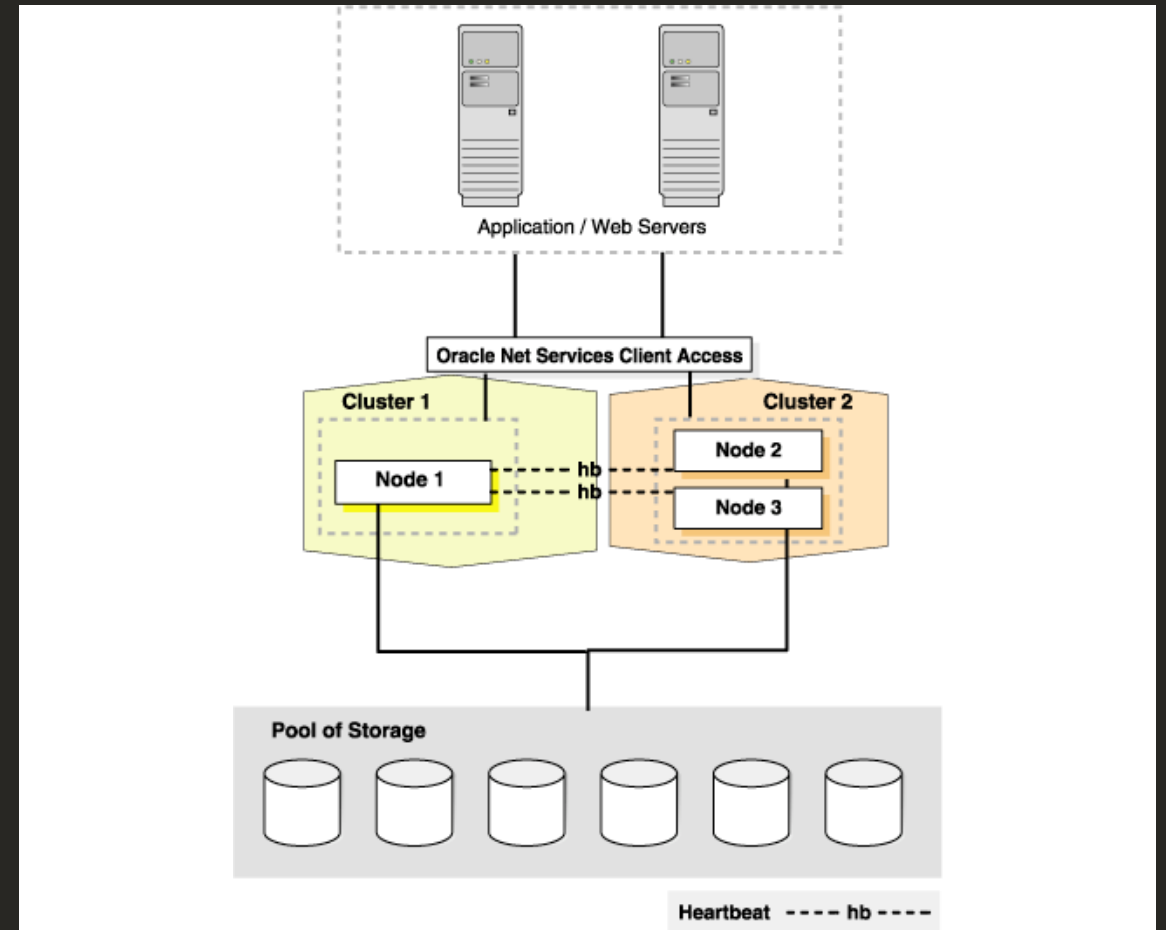
MySQL  
Cluster

## Active/Passive Shared Storage 솔루션

- Oracle Clusterware
- 서드파티 OS 클러스터링 SW  
(Redhat CS, Windows Clustering, 등)

기존 인프라 활용 가능

자동 페일오버





Replication

Shared Disk/  
Virtualization  
Options

Group  
Replication

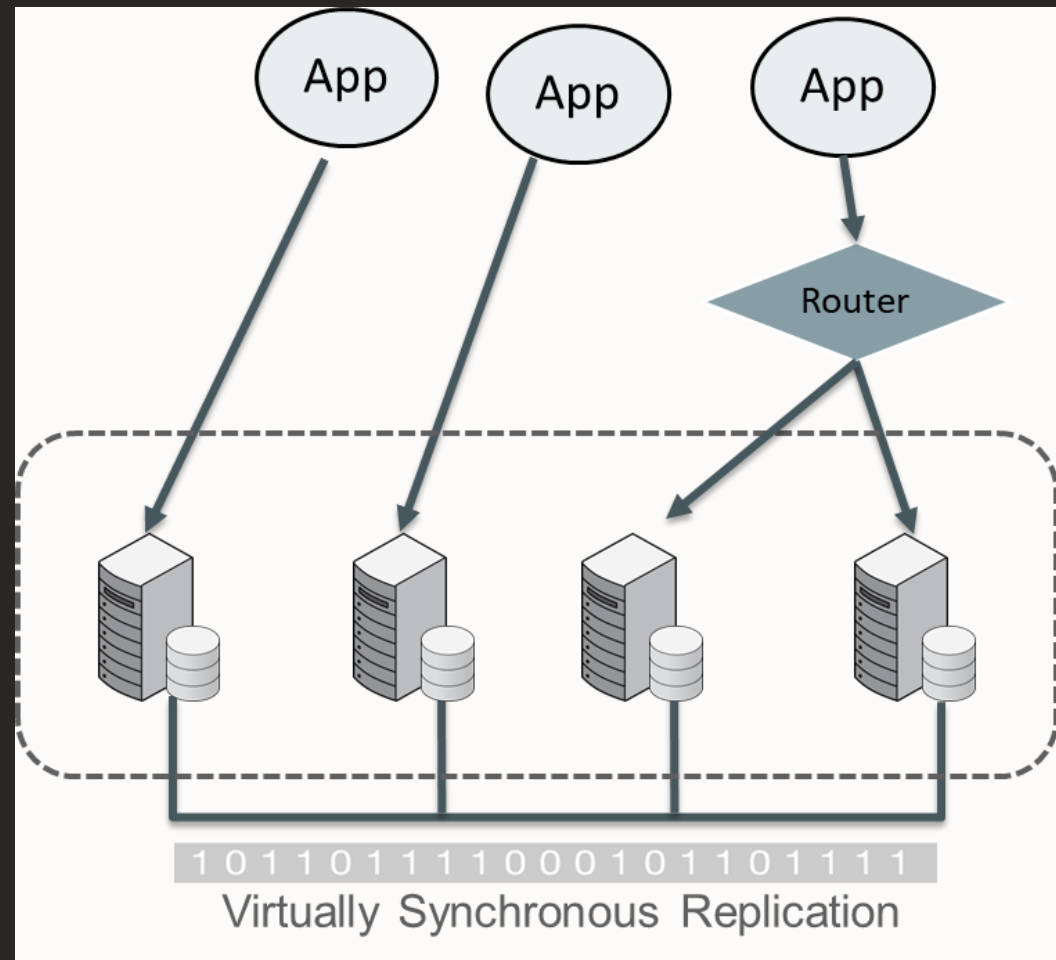
MySQL  
Cluster

## Group Replication

- 동기 방식
- Active/Active
- 탄력적

## 자동 페일오버

- Group Replication
- MySQL Router





Replication

Group  
Replication

Shared Disk/  
Virtualization  
Options

MySQL  
Cluster

Active/Active

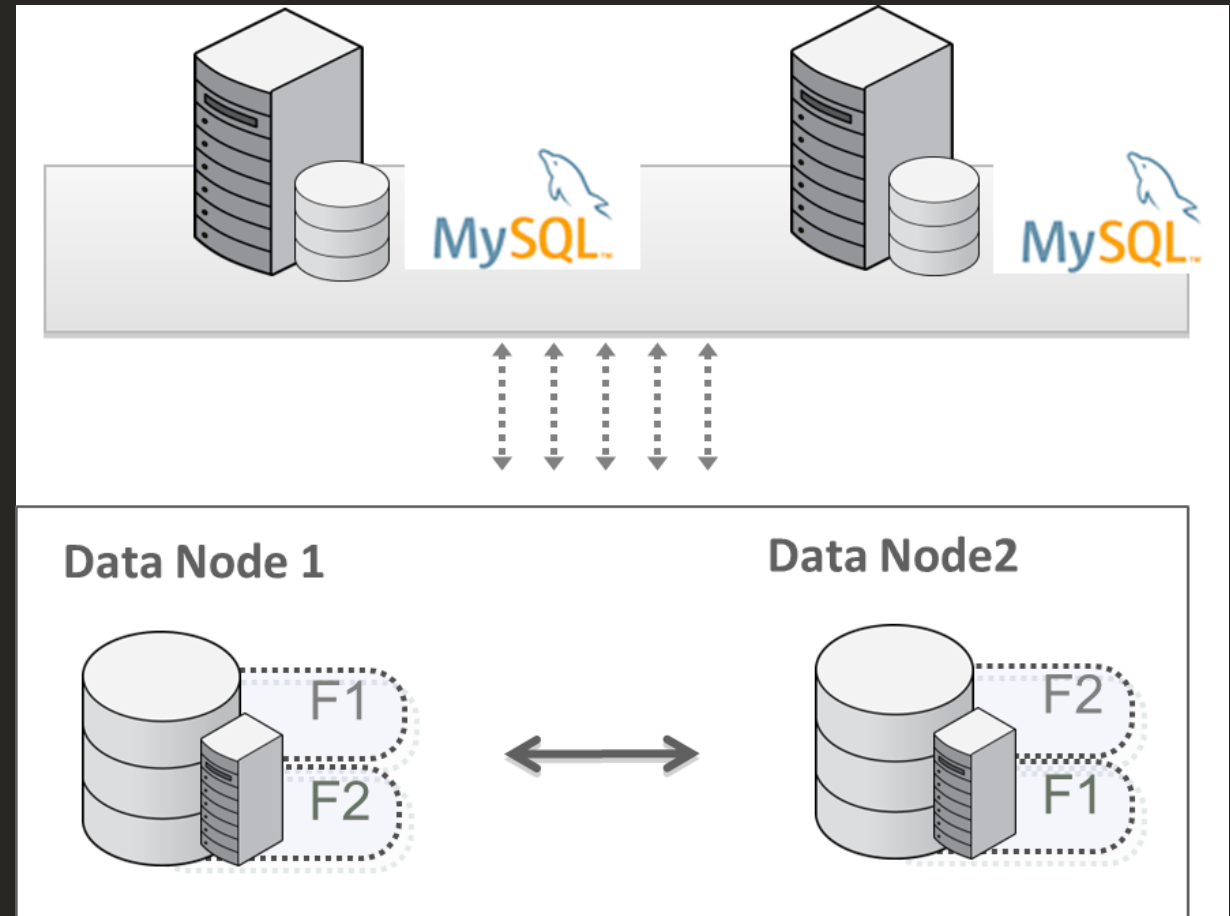
데이터 센터간 배포 포함

99.999% 가용성

단일 장애지점 없음

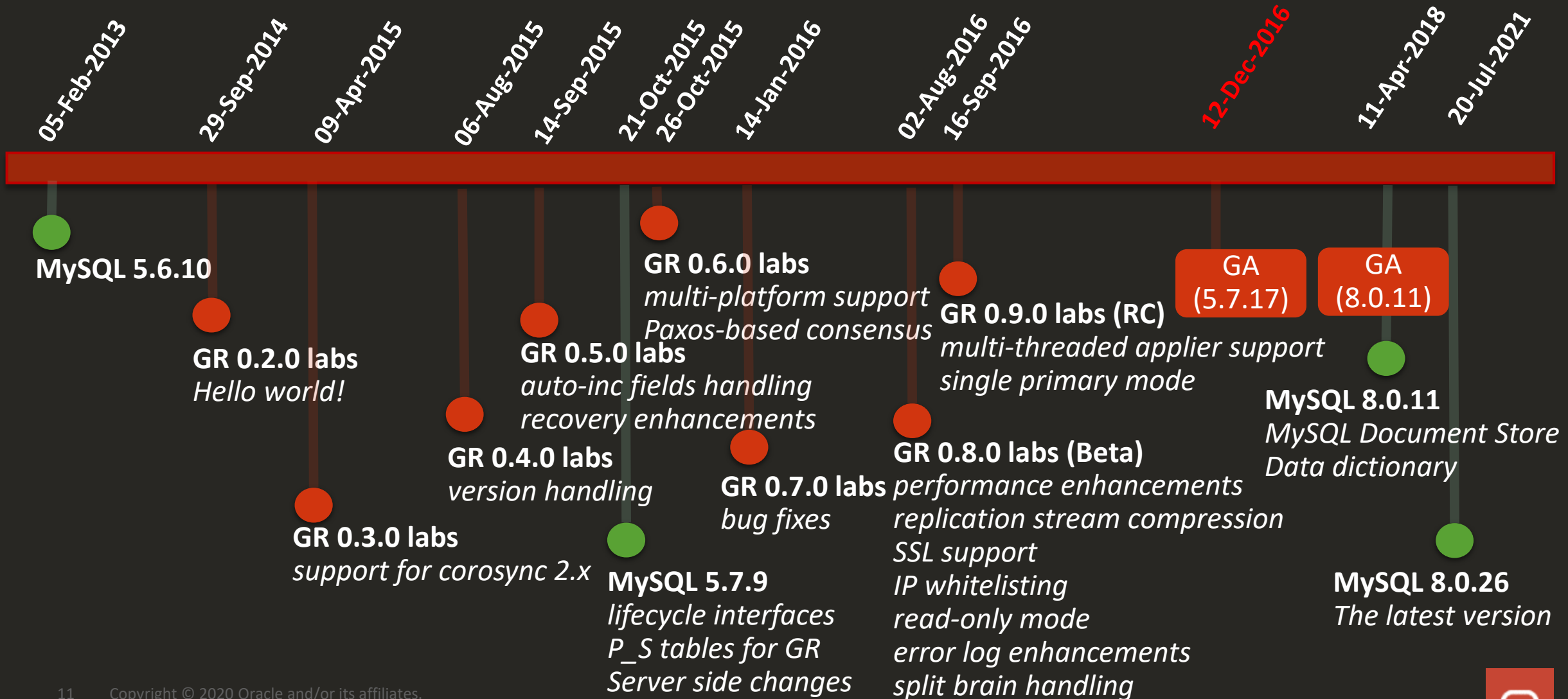
자동 페일오버

NDB Storage Engine



# InnoDB Cluster

# Timeline of Group Replication Releases

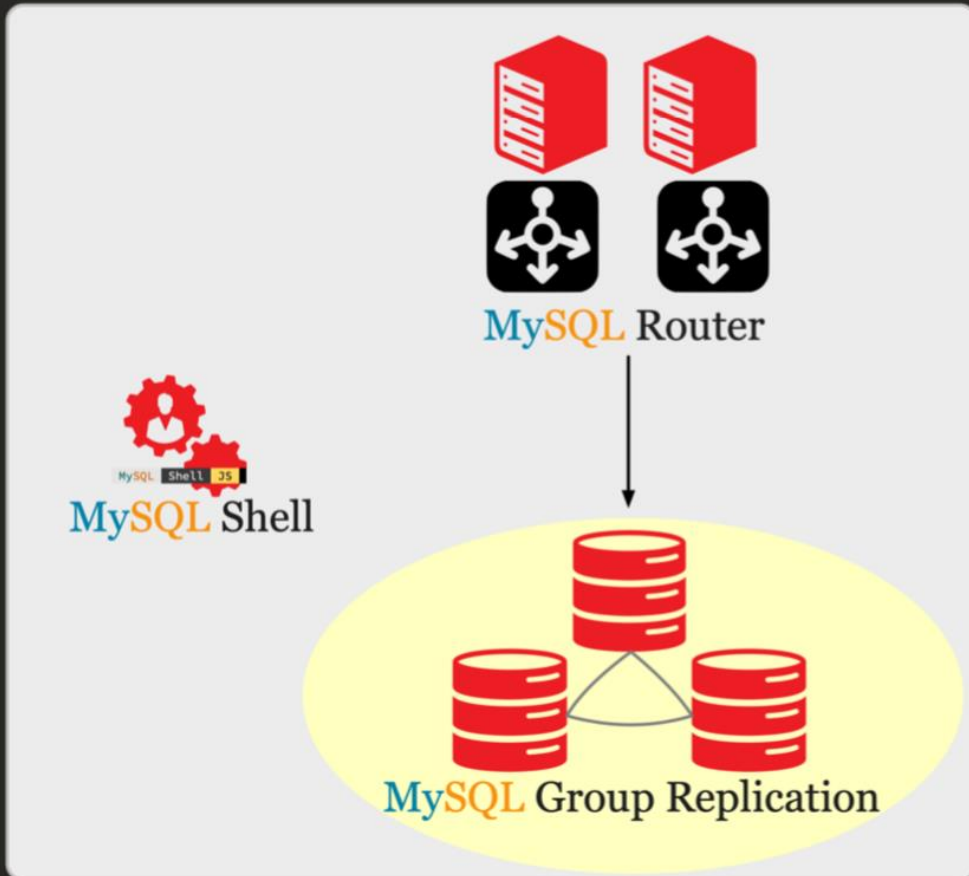


# Terminologies

- 클러스터 노드 및 클러스터 멤버 – 데이터베이스 서버 및 장비의 싱글 엔티티
- Primary 혹은 Master 노드, Source - Read/Write 노드
- Secondary 혹은 Slave 노드, Replica - Read Only 노드

# InnoDB Cluster - A Shared Nothing Cluster Solution

“A Single product – MySQL – with high availability and scaling features backed in; Providing an integrated end-to-end solution that is easy to use.”



MySQL 데이터베이스를 위한 분산방식의 고 가용성

- Fault tolerance
- 자동 페일 오버
- Active/Active 업데이트(제한적으로 적용)
- 자동 멤버십 관리
  - 멤버 추가/삭제
  - 네트워크 파티션 및 장애
- 충돌 감지 및 해결
- 데이터 유실 방지

# InnoDB Cluster 요구사항

- InnoDB 스토리지 엔진
- Primary Key
- GTID
- 안정성과 성능이 좋은 네트워크

Binary log active

- `--log-bin[=log_file_name]`

Slave updates logged

- `--log-slave-updates`

Binary log row format

- `--binlog-format=row`

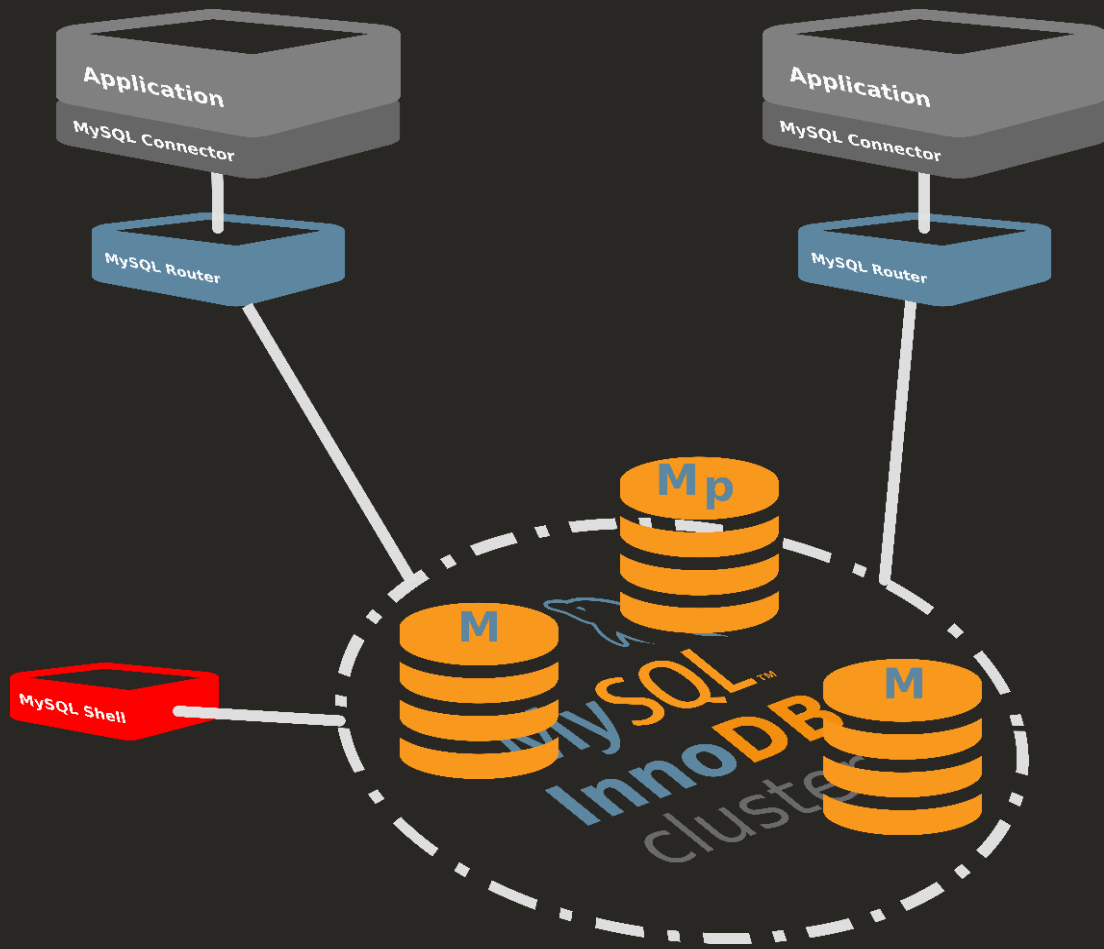
Global transaction identifiers enabled (on)

- `--gtid-mode=ON`

Replication information (metadata) repositories

- `--master-info-repository=TABLE`
- `--relay-log-info-repository=TABLE`

# MySQL InnoDB cluster

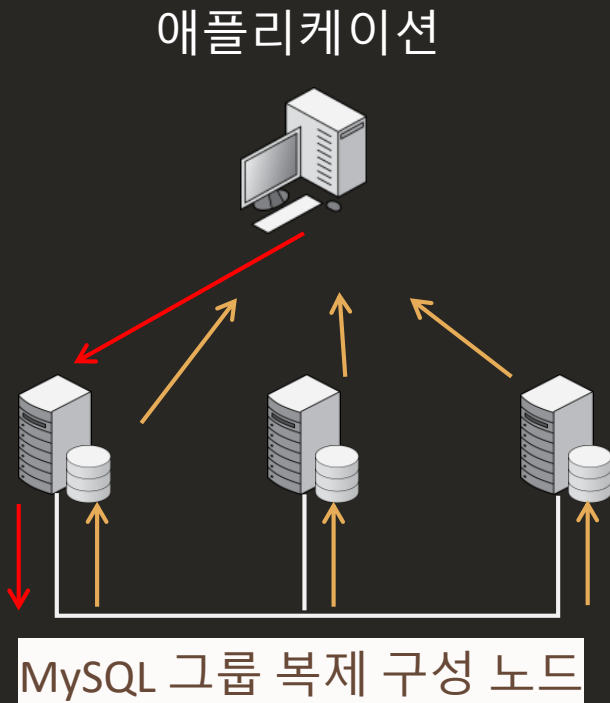


“단일 제품인 MySQL로 구성된고가용성 및 확장성을 제공하는 사용하기 쉬운 통합된 end-to-end 솔루션입니다.”

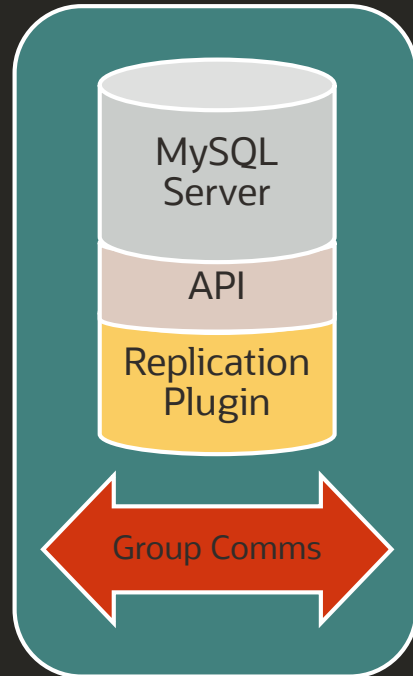
- MySQL Group Replication
- MySQL Shell
- MySQL Router



# Group Replication



→ 쓰기  
→ 읽기



## 자동 그룹 멤버십 관리 및 장애 감지

- 자동으로 서버 페일오버
- 탄력성있는 확장성
- 자동 복구

## 멀티 마스터 업데이트 지원

- 자동으로 데이터 충돌 발견 및 해결
- 데이터 일관성 보장

## 훌륭한 통합성 제공

- InnoDB 엔진
- GTID 기반 복제
- PERFORMANCE\_SCHEMA

# MySQL Shell

다양한 개발 언어를 지원하는 콘솔 인터페이스 기반의 개발 및 관리 기능을 제공하는 새로운 MySQL 클라이언트 툴

- JavaScript, Python 및 SQL 지원
- 도큐먼트 스토아와 함께 관계형 모델 지원

(X 프로토콜 및 MySQL 프로토콜 지원)

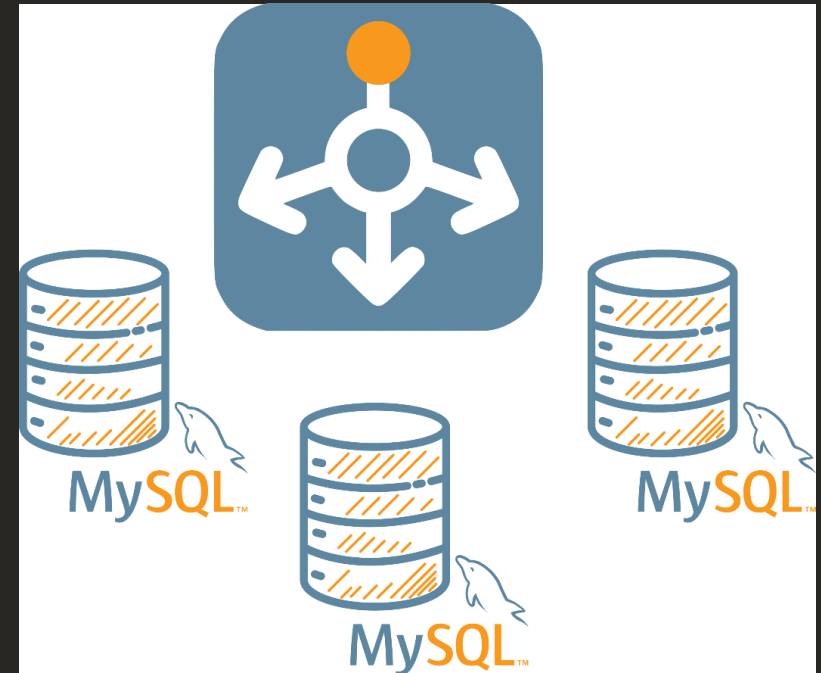
- InnoDB Cluster 생성 및 관리 API 제공
- 운영에 필요한 여러가지 유틸리티 제공

<https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-utilities.html>

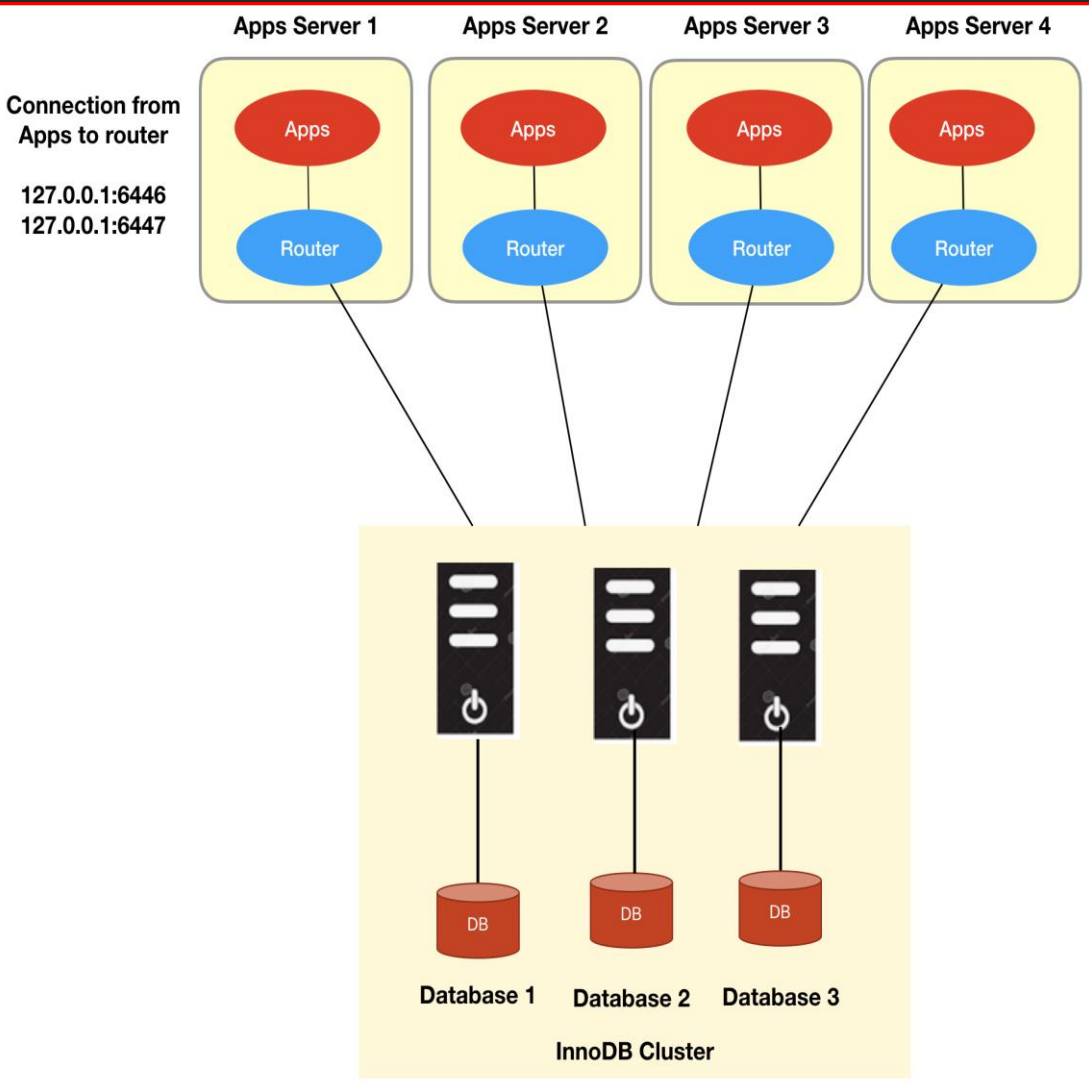
- 명령어, 배치 스크립트 지원

# MySQL Router

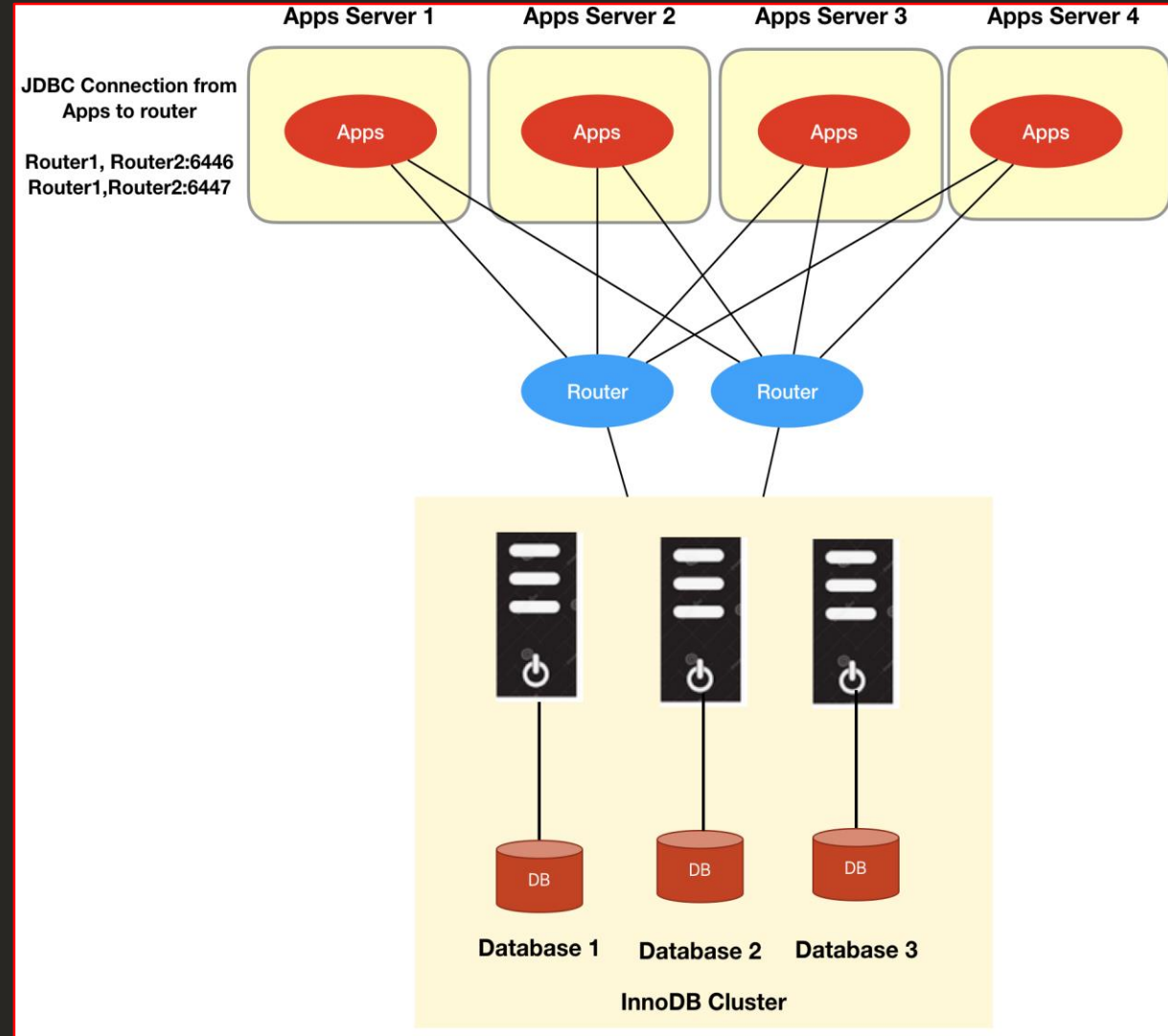
- 경량의 미들웨어
- 애플리케이션 연결 투명성 제공
- 로드 밸런싱 정책 기반으로 커넥션에 대한 자동 페일 오버를 관리
- R/W 포트 (6446, 64460) 로 Primary 멤버에 대한 자동 라우팅 (R/W, DML, Query)
- R/O 포트(6447, 64470) 로 Secondary 멤버에 대한 자동 라우팅 (R/O, Query only)



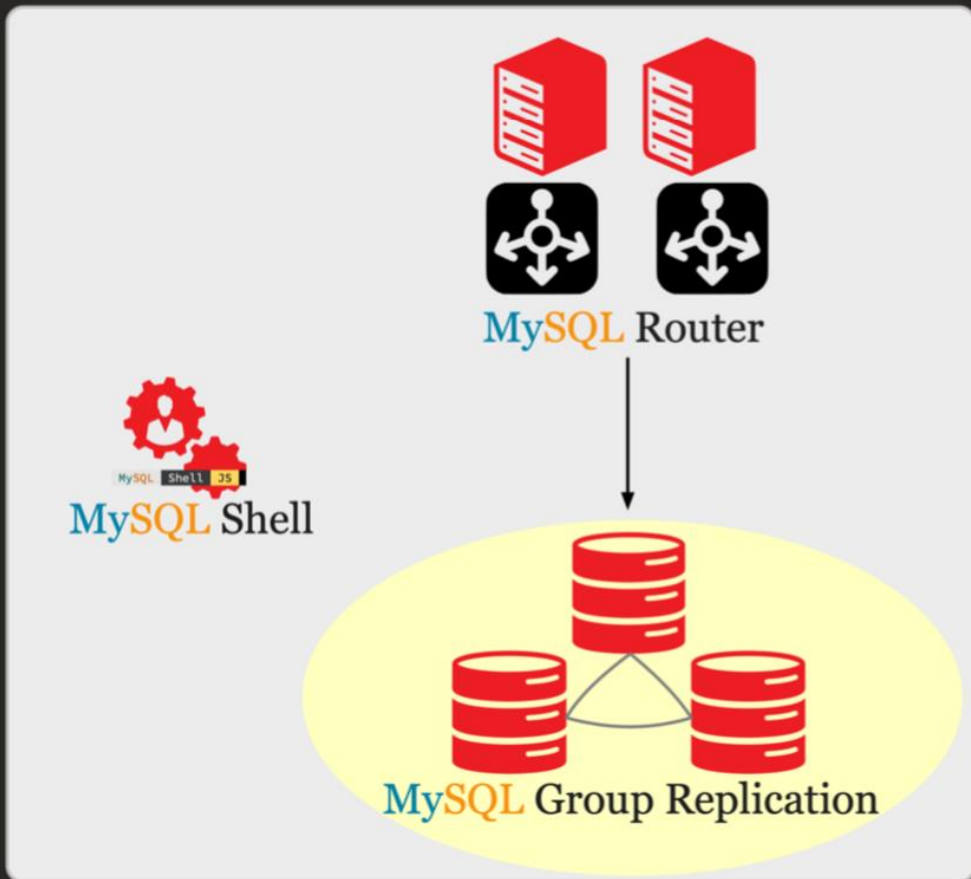
# MySQL Router 배포 방법



Vs.



# InnoDB Cluster 설정 방법



## MySQL InnoDB Cluster

```
mysql-js> \c admin@mysql1  
mysql-js> cluster = dba.createCluster('cluster')
```

Configure server to add later:

```
mysql-js> dba.configureInstance('admin@mysql2')
```

Add server to the Cluster:

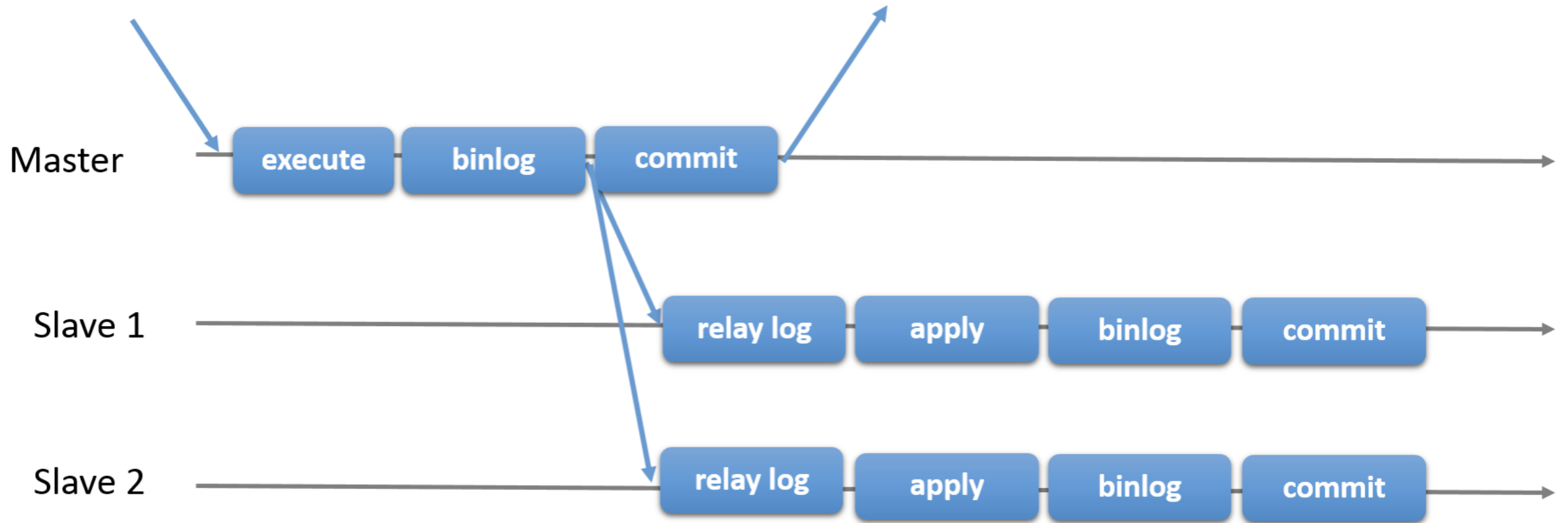
```
mysql-js> cluster.addInstance('admin@mysql2')
```

Bootstrap MySQL Router

```
$ sudo mysqlrouter --user=mysqlrouter --bootstrap  
$ sudo systemctl start mysqlrouter
```

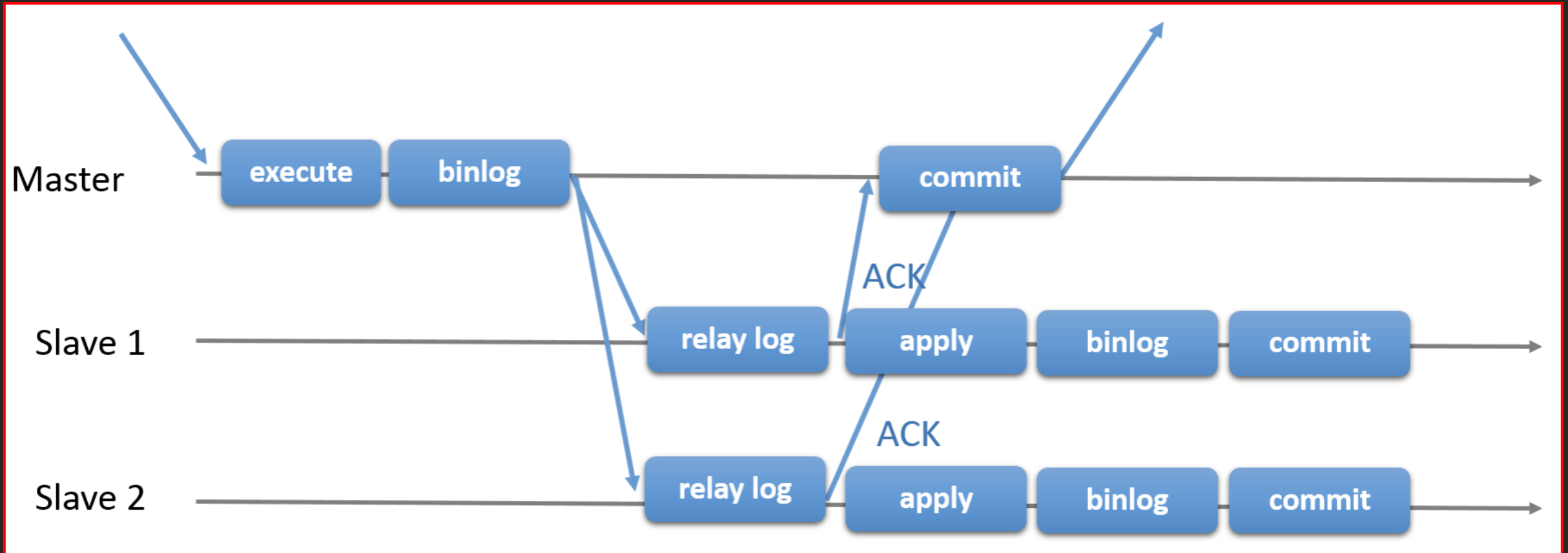
# Group Replication의 동작 방식

# 비동기식 MySQL Master/Slave 복제

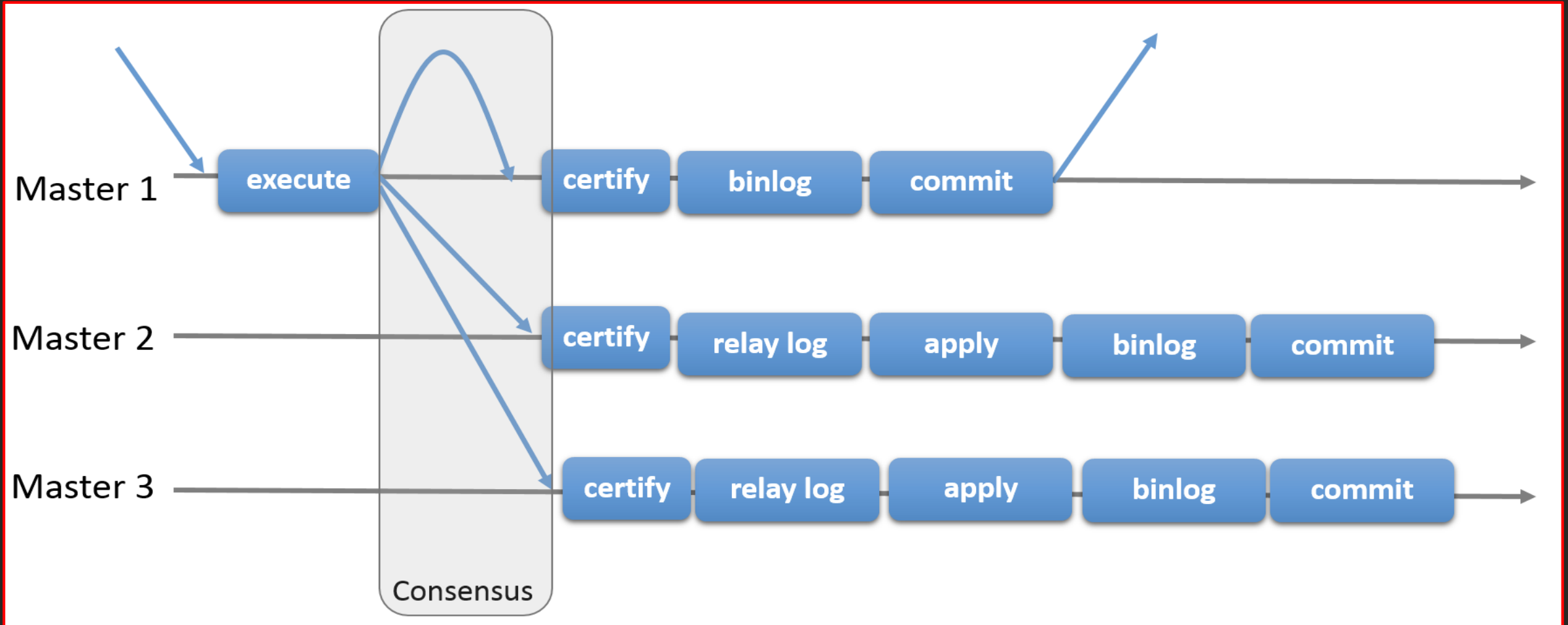




# 반동기식 MySQL Master/Slave 복제



# Group Replication 동작 방식



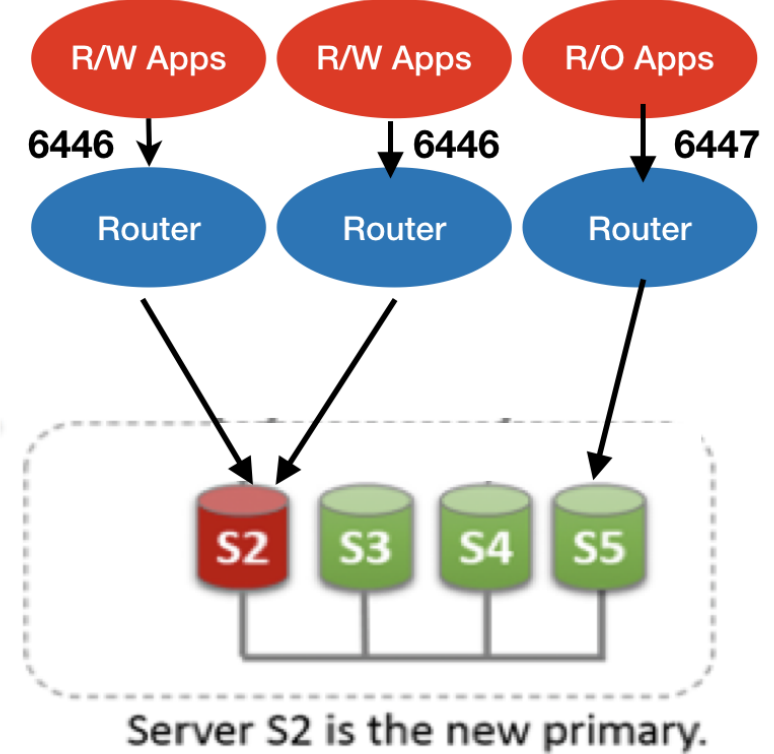
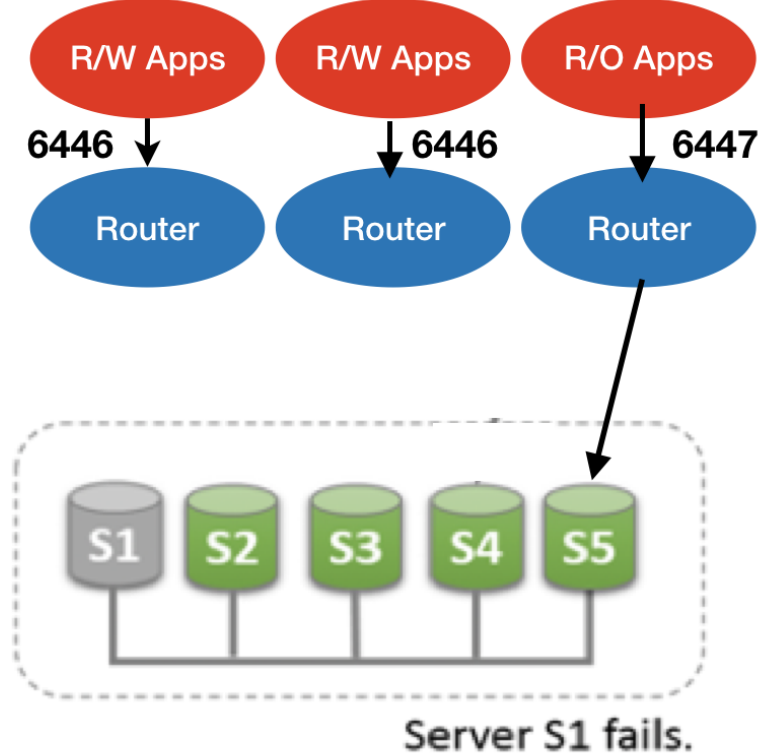
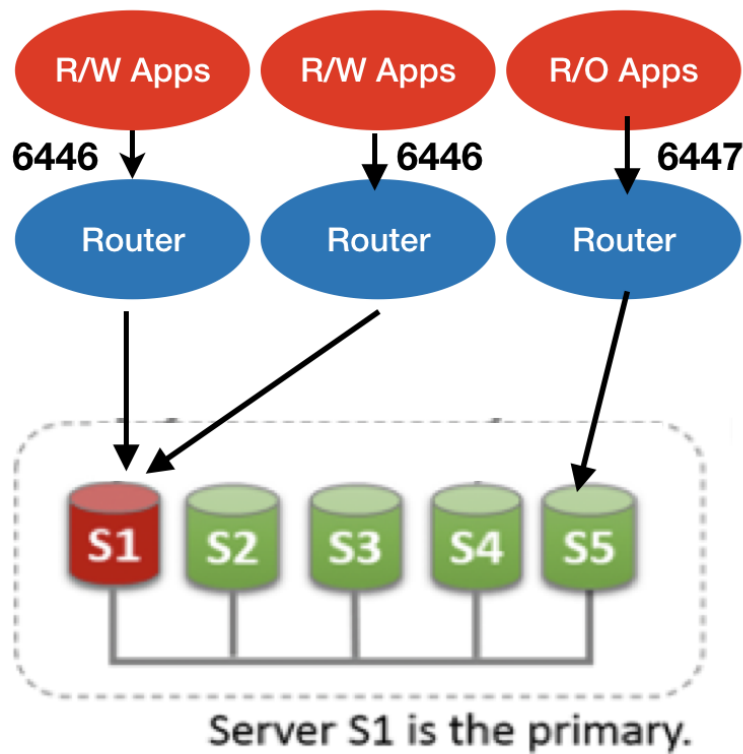
Group Replication Protocol - first transaction that commit "wins"

# Group Replication - Single Primary Mode

서버 장애를 자동 감지

그룹 내에서 장애 여부를 판단, 새로운 Primary 승격

격리된 서버는 OFFLINE 상태로 변경

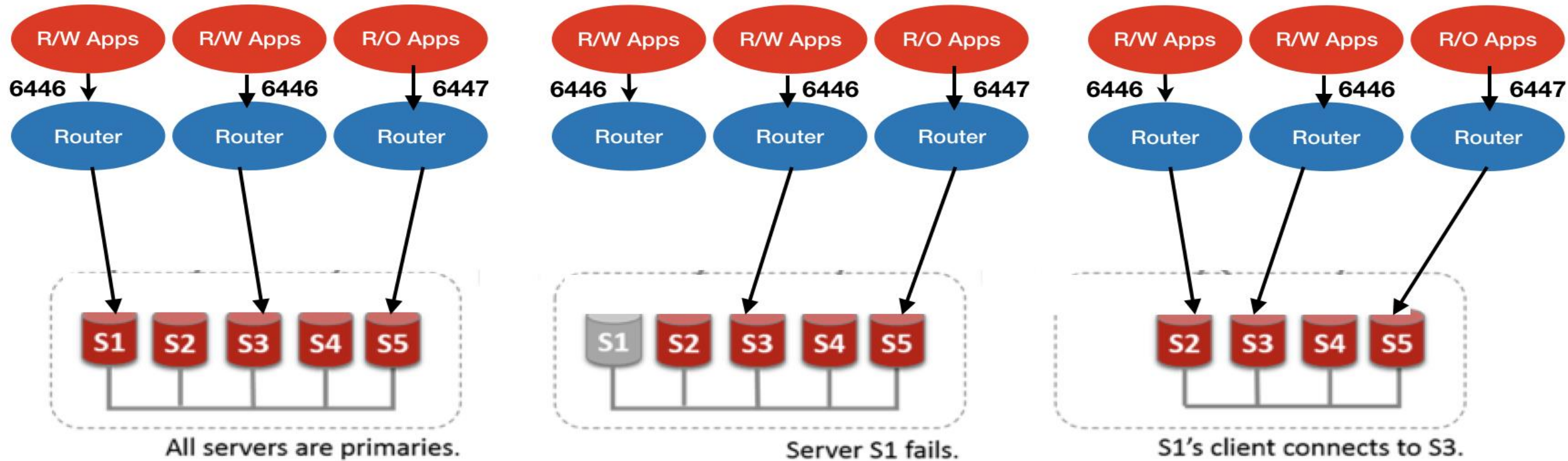


# Group Replication - Multi Primary Mode

모든 서버 R/W 가능

서버 장애 발생시, 클라이언트 요청을 다른 노드로 연결

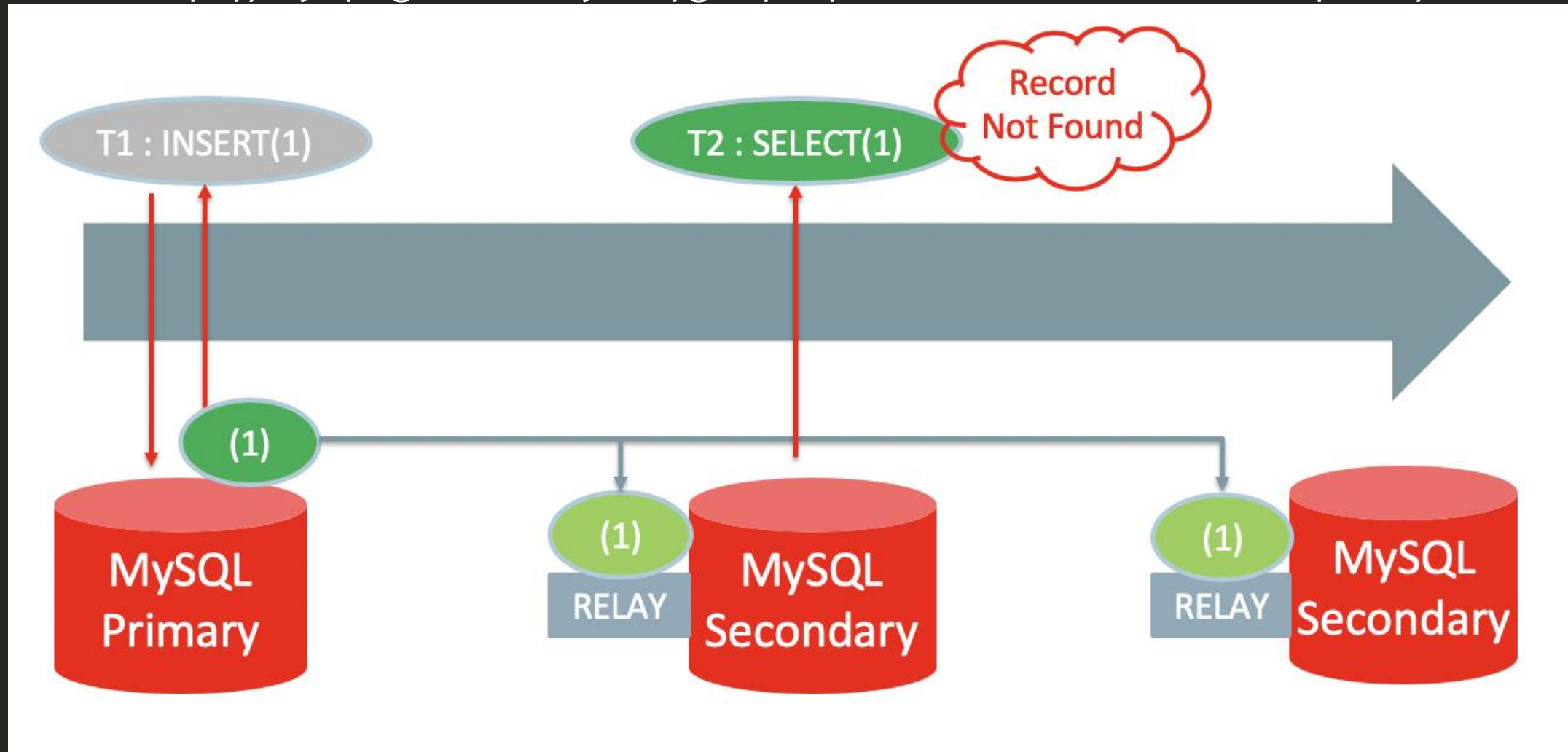
트랜잭션 충돌 해결



# Group Replication Consistency

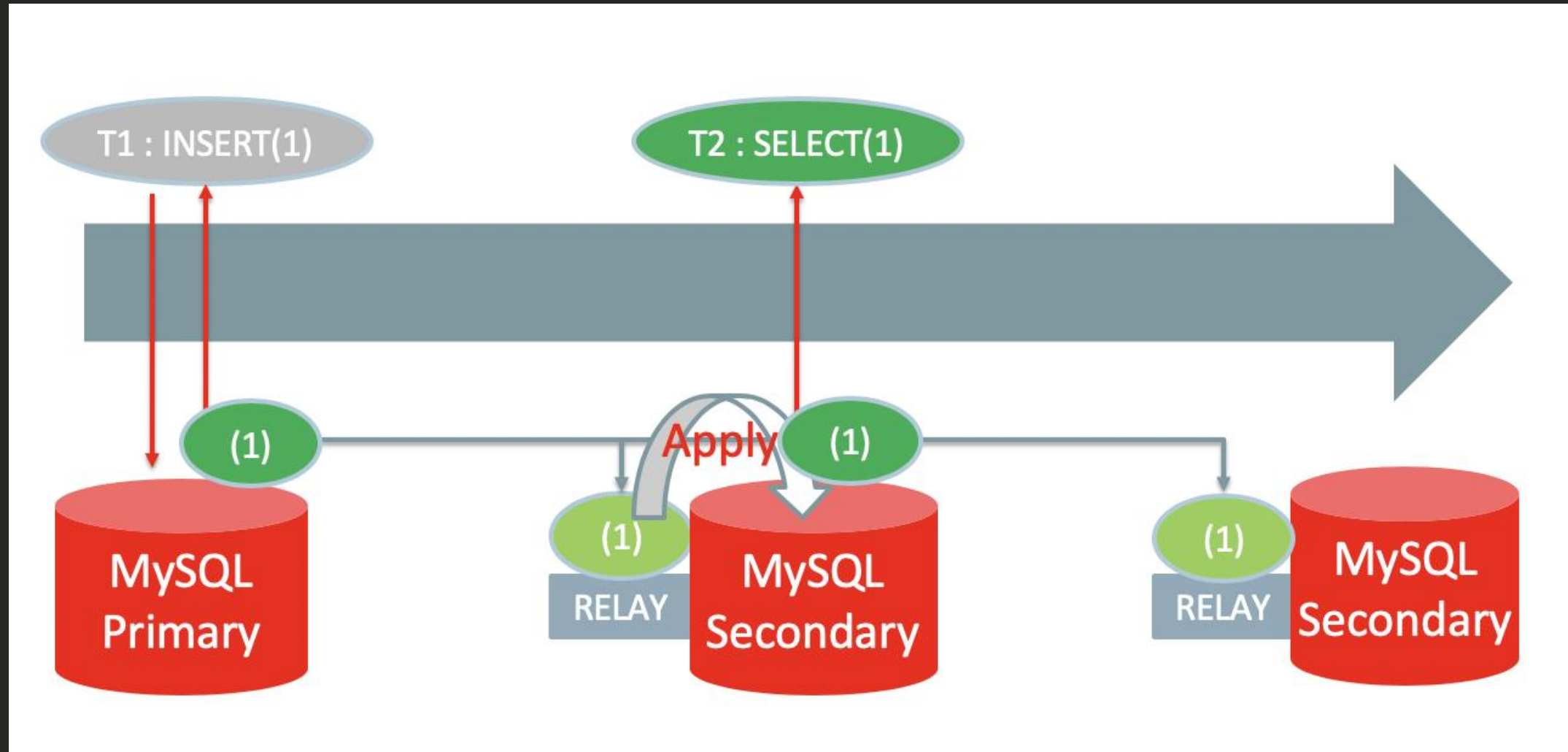
# Group Replication Consistency - Eventual

\* 참고 : <https://mysqlhighavailability.com/group-replication-consistent-reads-deep-dive/>



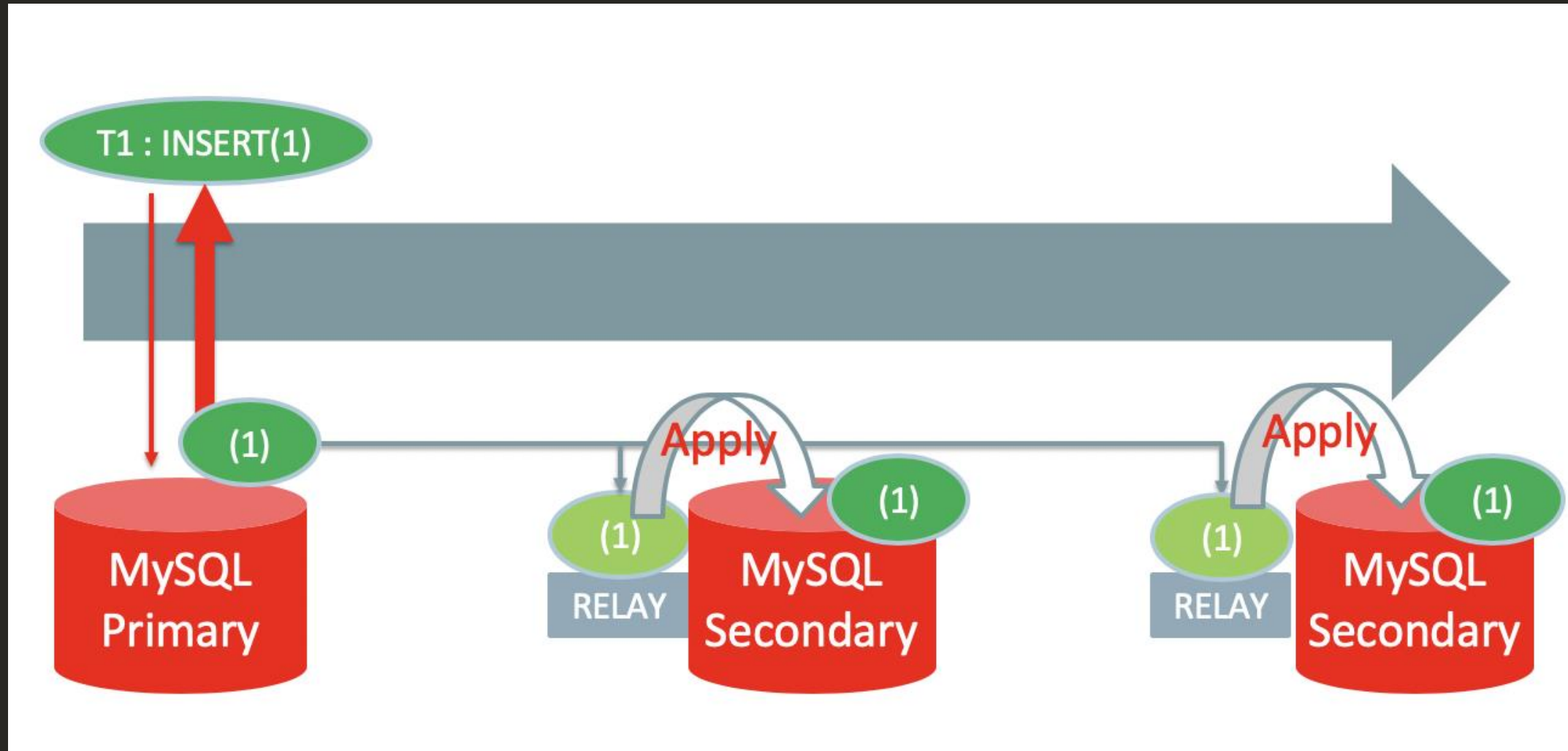


# Group Replication Consistency - Before

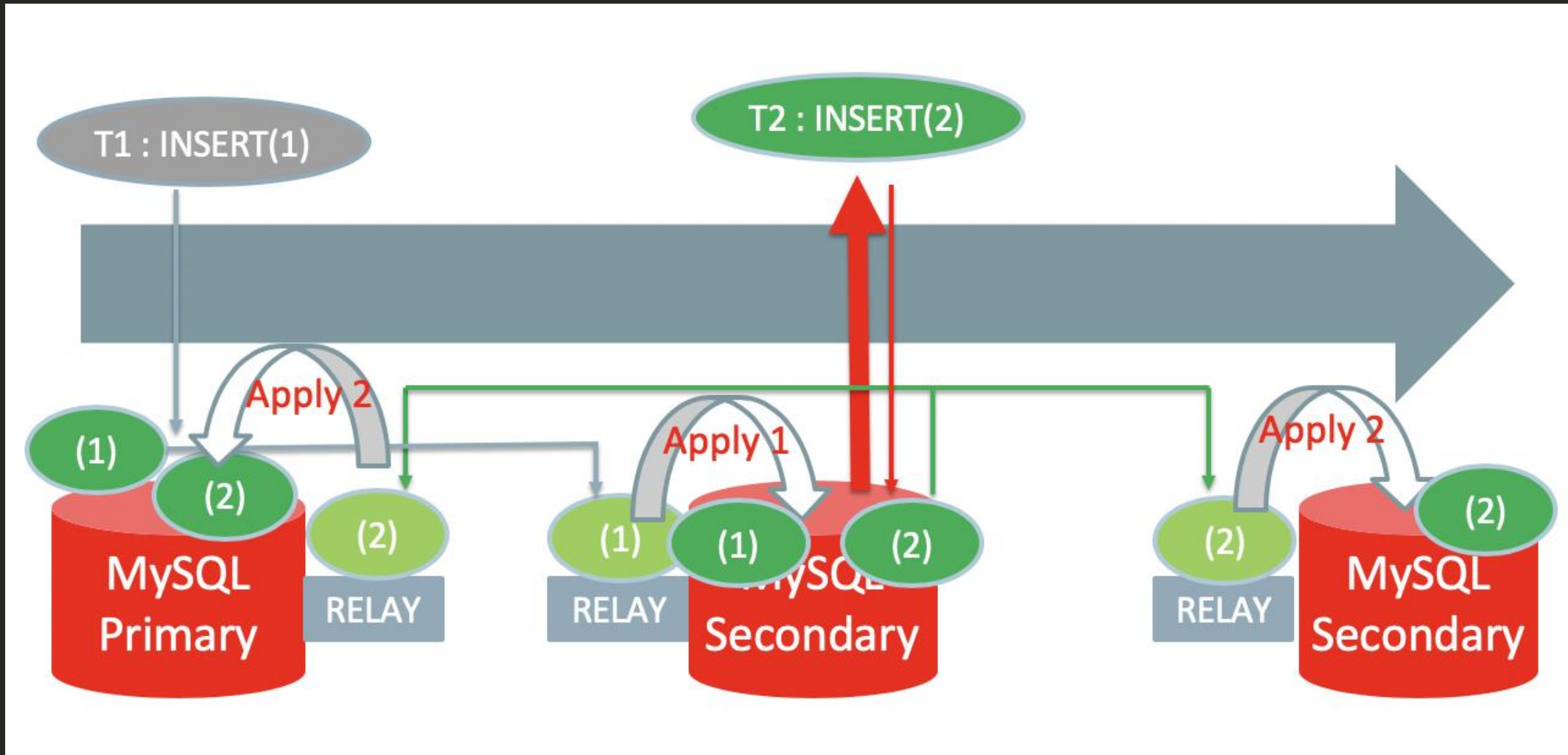




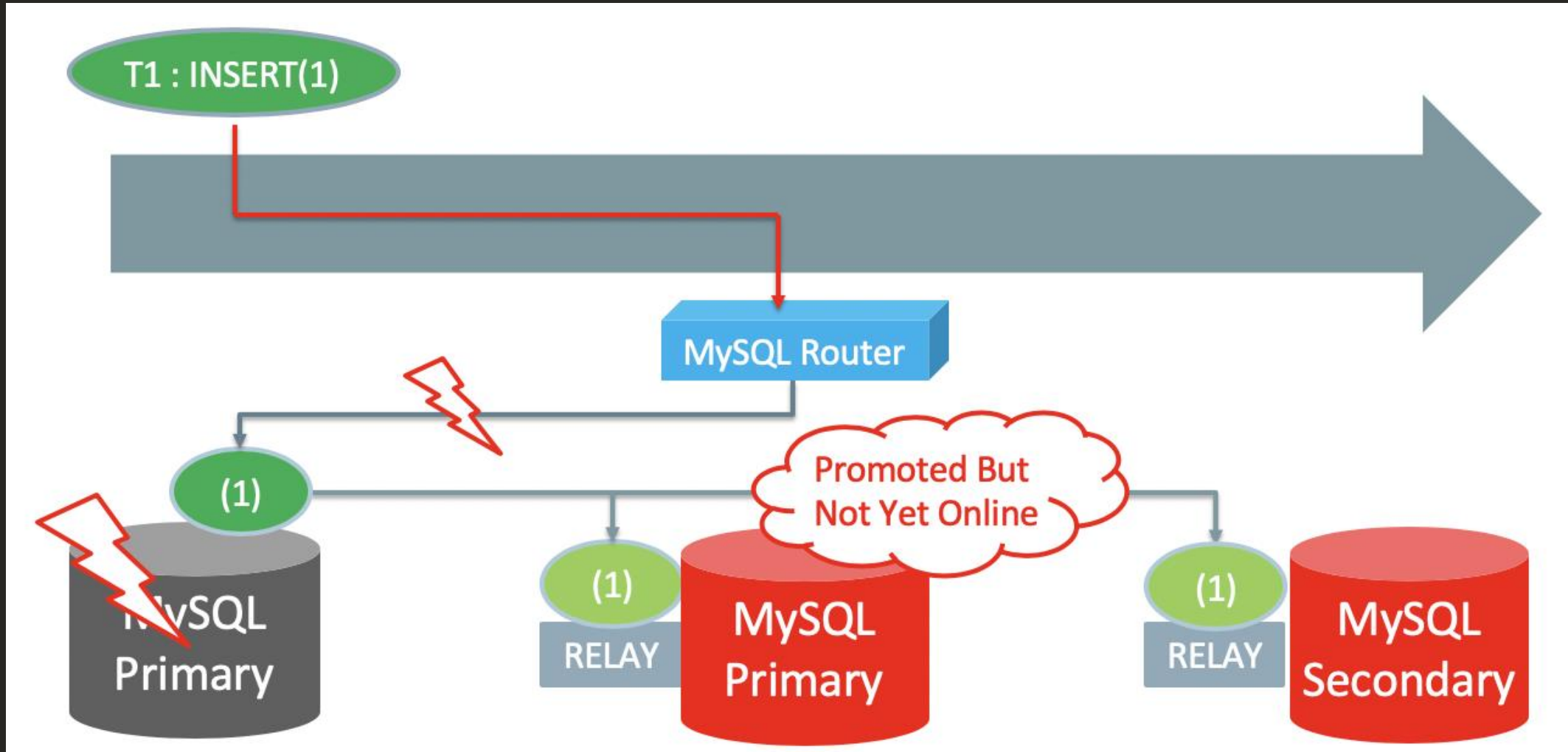
# Group Replication Consistency - After



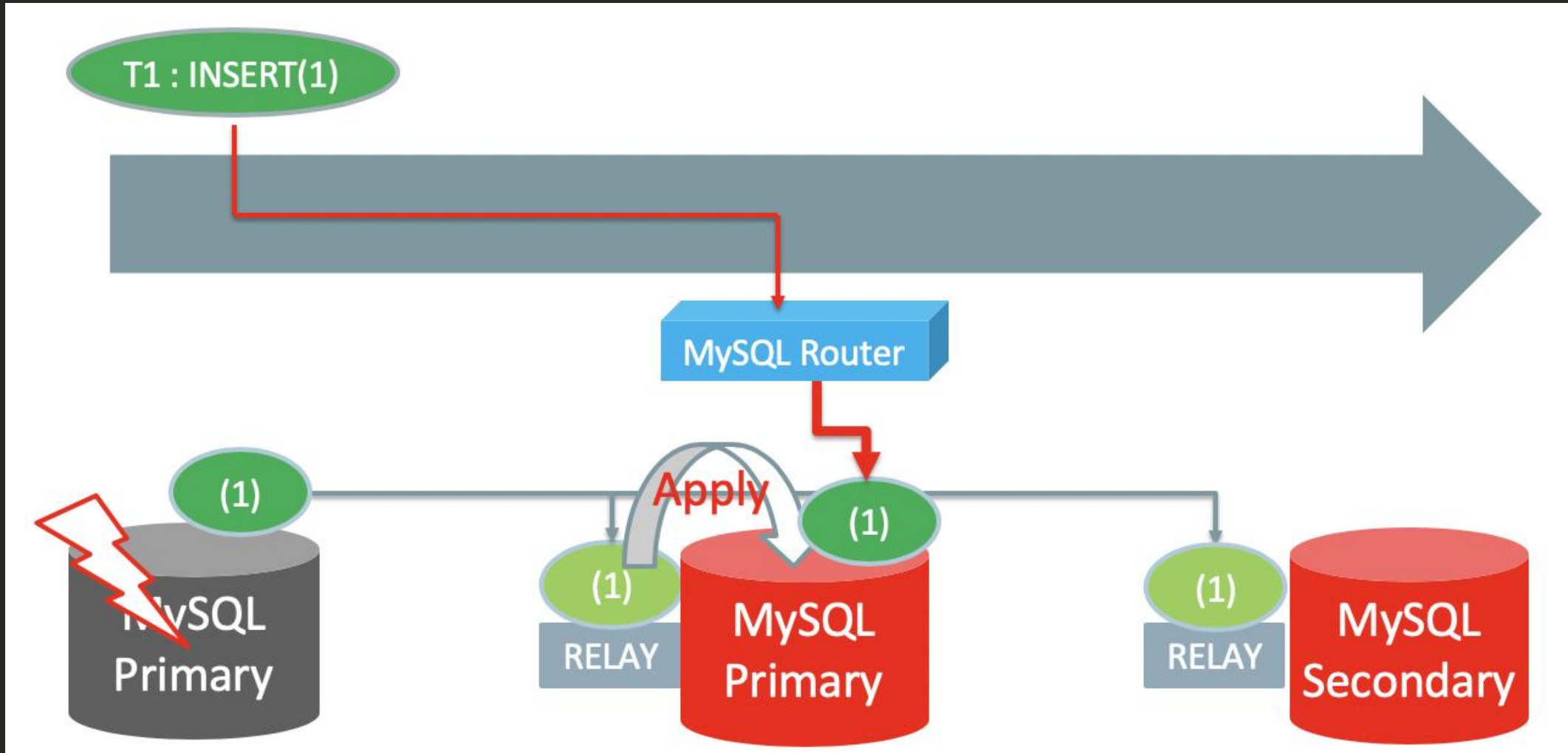
# Group Replication Consistency – Before and After



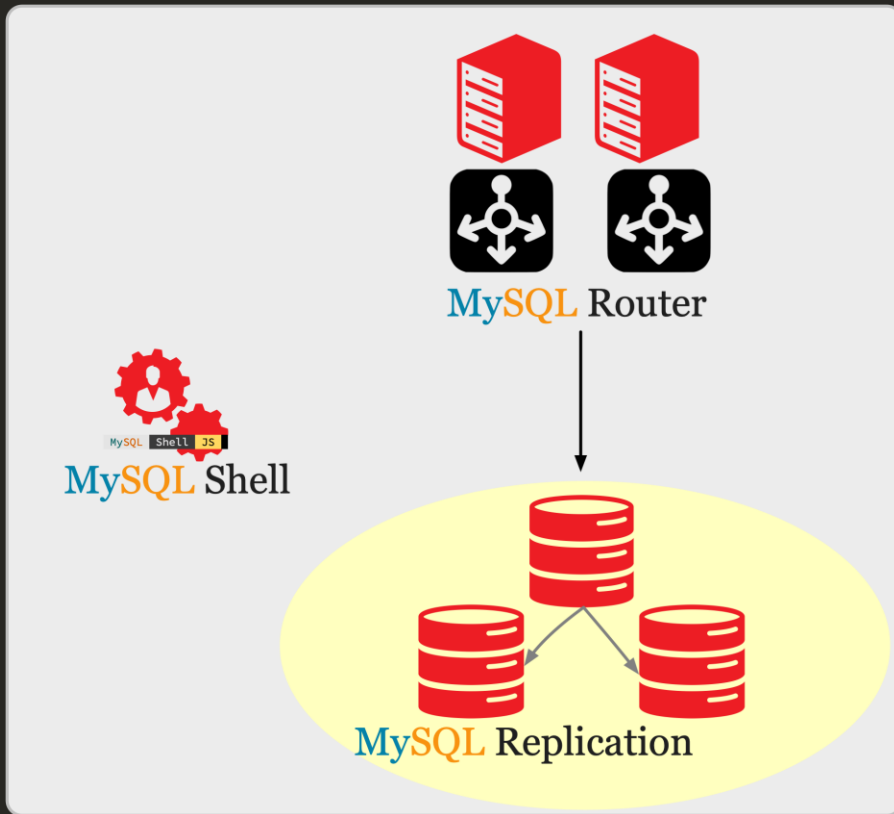
# BEFORE\_ON\_PRIMARY\_FAILOVER



# BEFORE\_ON\_PRIMARY\_FAILOVER



# MySQL InnoDB ReplicaSet



## 비동기 복제 아키텍처

- (수동) Switchover 및 Failover
- (비동기) 읽기 확장
- 단순한 복제 아키텍처

**MySQL Shell** 멤버의 설정, 추가, 삭제

**MySQL Router** 애플리케이션 부하를 분산

**InnoDB CLONE** 자동으로 멤버를 프로비저닝,  
InnoDB와 완전히 통합



# InnoDB ReplicaSet 기능

## Before

- 백업을 복구해서 멤버를 프로비저닝
- 복제 사용자 생성 및 복제 설정
- MySQL 라우터 또는 다른 라우팅 솔루션에서 수동으로 서버 설정, 추가, 삭제
- 수동으로 토폴로지 변경
- 추가 모니터링 도구를 사용해서 토폴로지 확인
- 복잡성: 사용자가 모든 구성 요소 및 해당 설정의 전체 구성에 대한 책임이 있음
- 모든 구성은 커스트마이징된 구성임
- 많은 수동 단계와 추가 소프트웨어가 필요하며 항상 커스트마이징해야 하고 DBA가 관려해야 함

## Now

- 새 멤버 자동 프로비저닝: InnoDB Clone
- MySQL Shell을 이용한 사용자 및 복제 자동 구성
- MySQL 라우터로 로드 밸런싱
- 자동 라우터 부트스트랩 - 구성 필요없음
- 라우터는 stateless이며 토폴로지 변경을 자동으로 적용
- Shell은 표준화된 모범 사례 설정을 통해 서버, 라우터, 복제를 구성하고, 실수를 방지
- 표준 솔루션 - Oracle에서 지원 및 QA 제공
- MySQL 초보자도 사용하기 쉬움

# InnoDB ReplicaSet 요구사항 및 제약사항

MySQL 8 (set persist)

GTID

수동 장애 조치만 가능

- 데이터 일관성!

이와 같은 토폴로지는 데이터 일관성을 보장할 수 없으므로 multi-primary 없음

- 데이터 조정 없음
- 충돌 처리 없음

모든 secondary 멤버는 primary 로 부터 복제

- 단일 계층 복제 지원



# Basic Cluster Administration

- `dba.getCluster().status()` - checking cluster status
- `dba.getCluster().describe()` - viewing the structure of a cluster
- `dba.getCluster().rescan()` - Display any discovered instance and ask if you want to rejoin them to cluster
- `dba.getCluster().addInstance('user@host:port')` - add instance to cluster
- `dba.getCluster().removeInstance('user@host:port')` - remove instance from cluster
- `dba.getCluster().rejoinInstance('user@host:port')` - for instance that has left cluster and not able to rejoin automatically.
- *`dba.getCluster().setPrimaryInstance('user@host:port')` - choose primary node, **8.0 only***
- `dba.getCluster().forceQuorumUsingPartitionOf('localhost:port')` - restoring quorum loss
- `dba.rebootClusterFromCompleteOutage()` - start cluster after instances are restrated

# Common System Variables

- `group_replication_exist_state_action` (Admin API: `exitStateAction`), values: `ABORT_SERVER`, `OFFLINE_MODE`, `READ_ONLY`
- `group_replication_member_weight` (Admin API: `memberWeight`). values range from 0 to 100
- `group_replication_member_expel_timeout` (Admin API: `expelTimeout`), values: 0 or greater than 0
- `group_replication_autorejoin_tries` (Admin API: `autoRejoinTries`), values: 0 to 2016

Thank you !