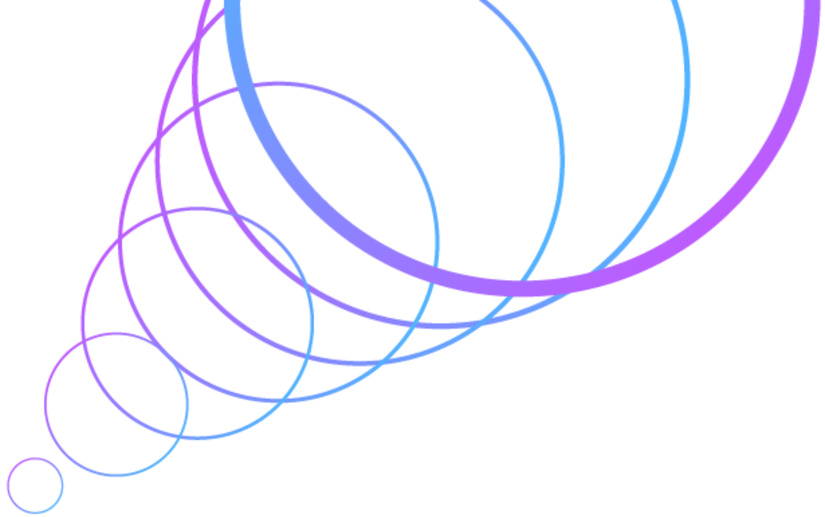
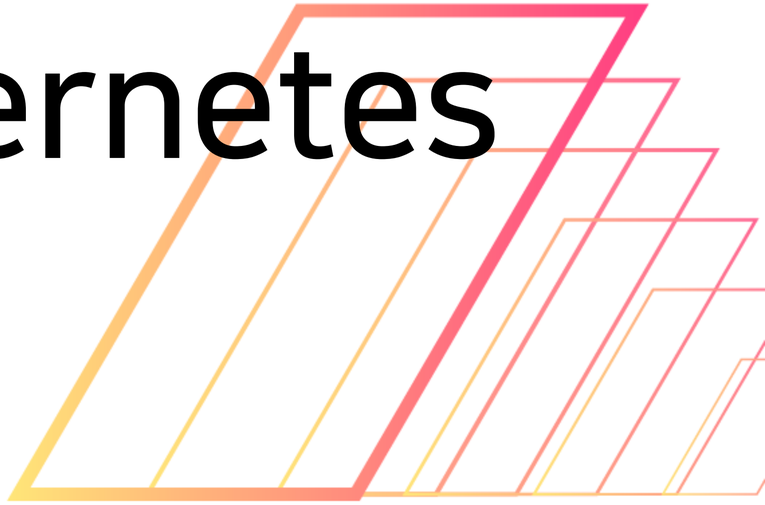




Migration Neo4j Platform from Physical to Kubernetes

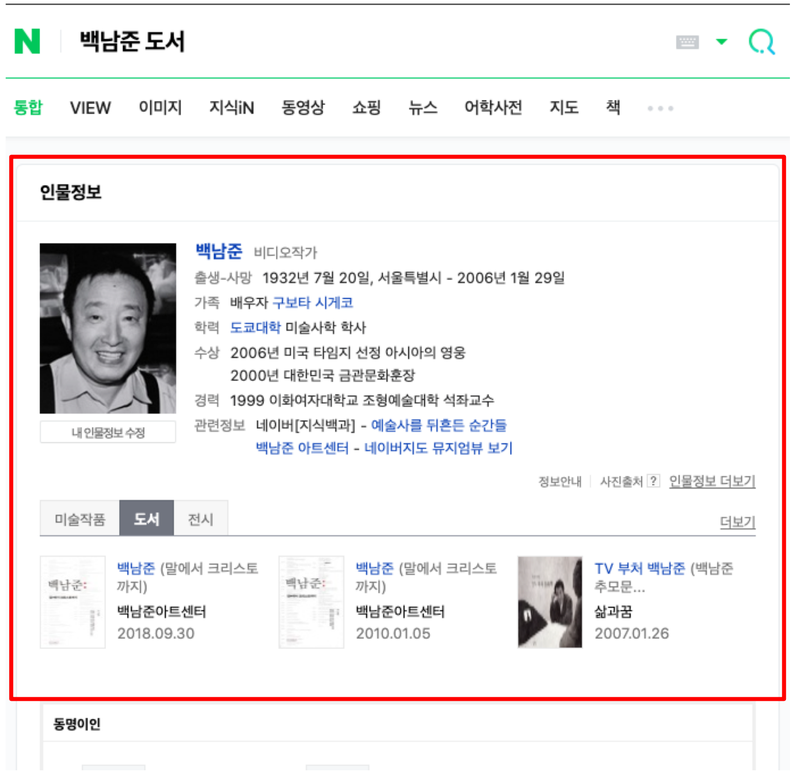
김동국 NAVER

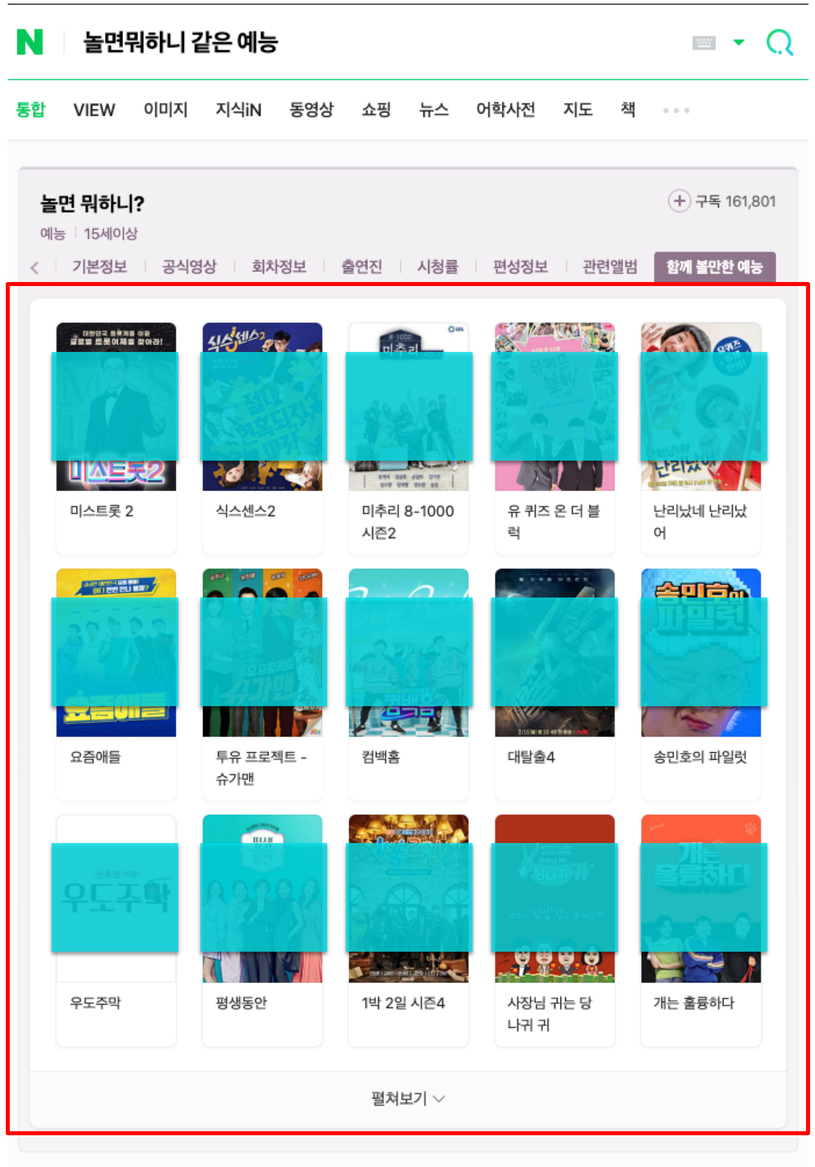
CONTENTS

1. Knowledge Base

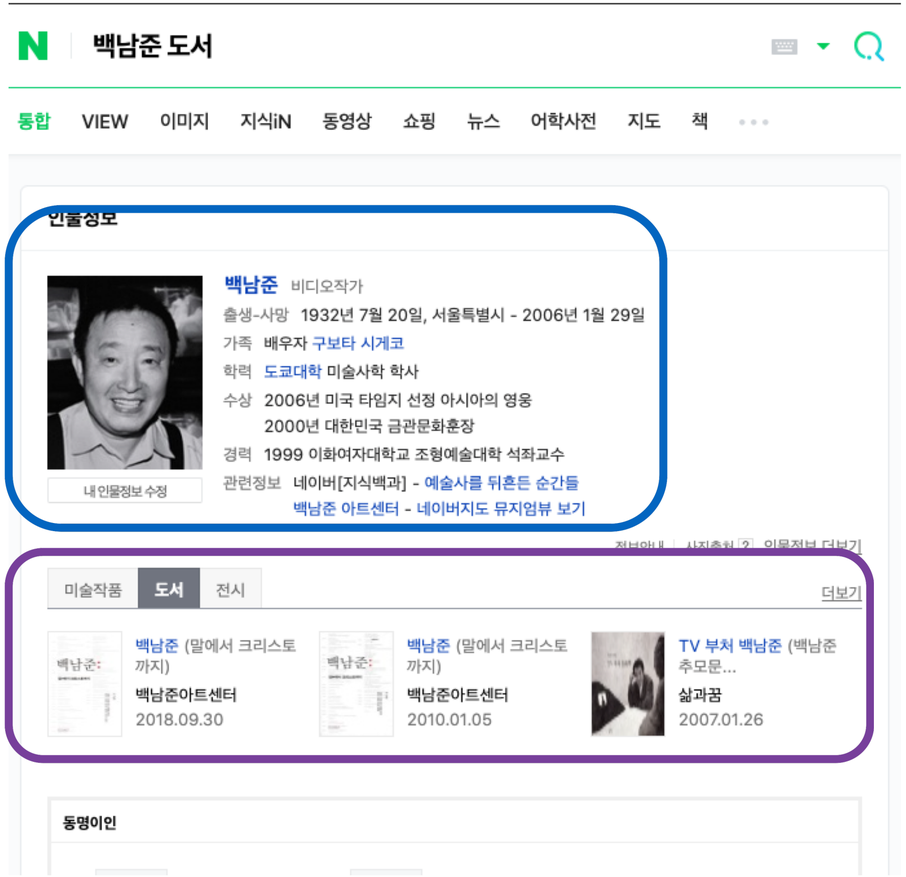
1.1 지식베이스

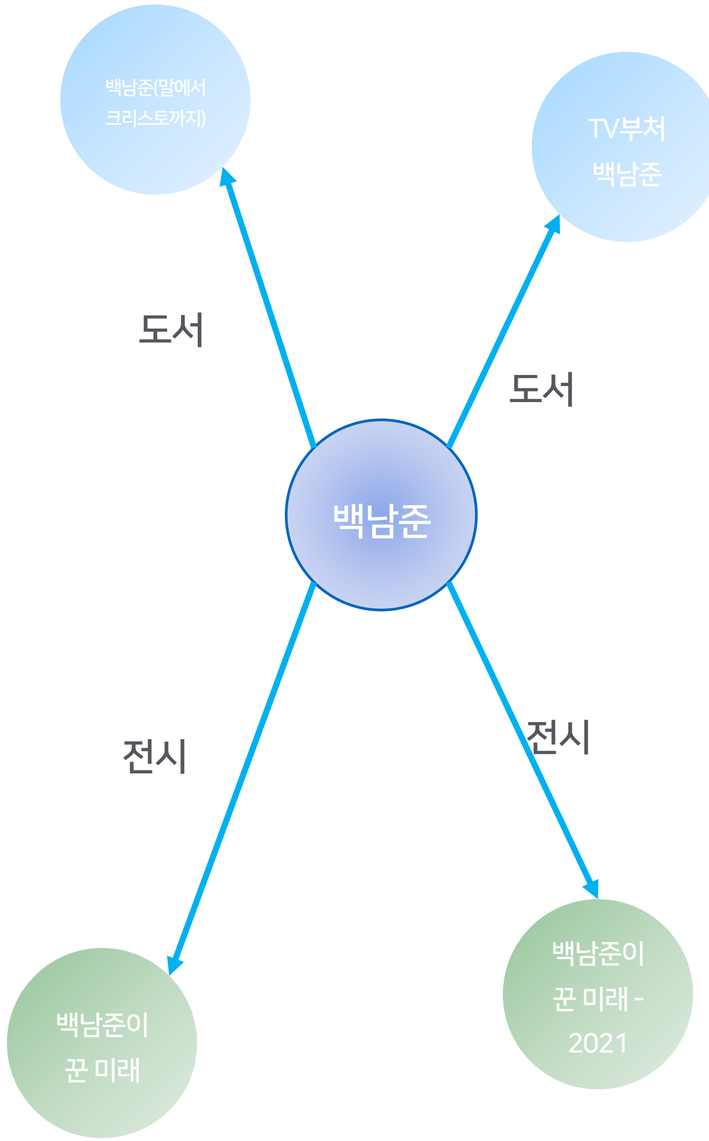
컨텐츠 검색





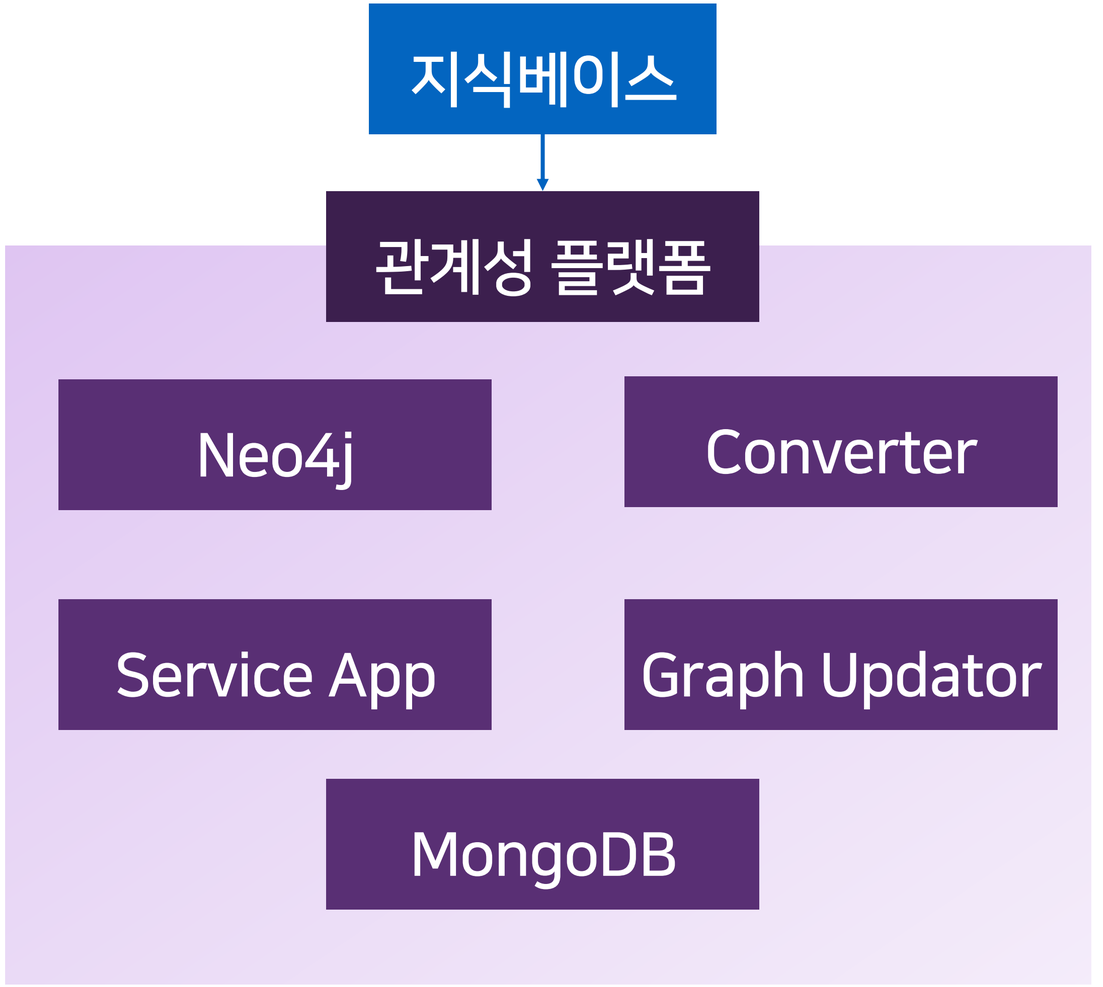
1.2 관계성 데이터

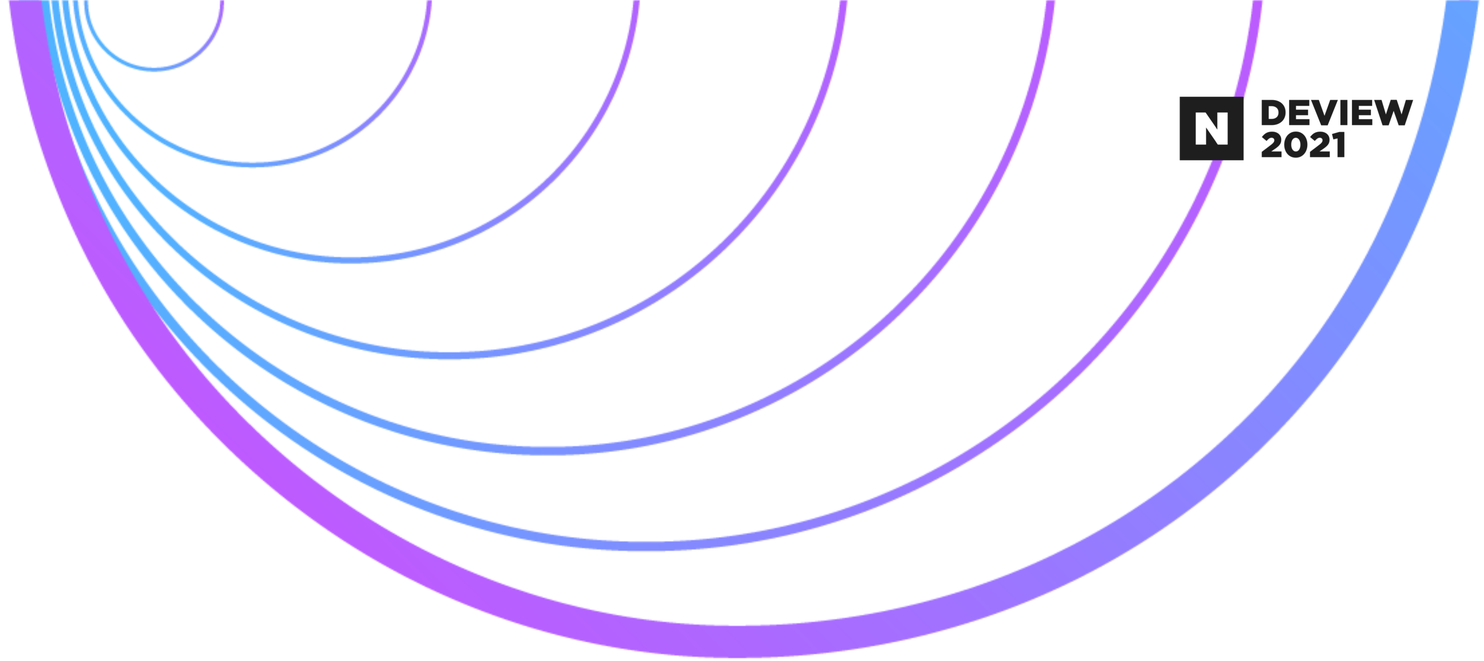

백남준 도서
통합 VIEW 이미지 지식iN 동영상 쇼핑 뉴스 어학사전 지도 책
인물정보
백남준 비디오작가
출생-사망 1932년 7월 20일, 서울특별시 - 2006년 1월 29일
가족 배우자 구보타 시게코
학력 도쿄대학 미술사학 학사
수상 2006년 미국 타임지 선정 아시아의 영웅
2000년 대한민국 금관문화훈장
경력 1999 이화여자대학교 조형예술대학 석좌교수
관련정보 네이버[지식백과] - 예술사를 뒤흔든 순간들
백남준 아트센터 - 네이버지도 뮤지엄뷰 보기
내 인물정보 수정
미술작품 도서 전시 더보기
백남준 (말에서 크리스토까지)
백남준아트센터
2018.09.30
백남준 (말에서 크리스토까지)
백남준아트센터
2010.01.05
TV 부처 백남준 (백남준 추모문...
삶과꿈
2007.01.26
동명이인


백남준(말에서 크리스토까지)
TV부처 백남준
도서
도서
백남준
전시
전시
백남준이 꾼 미래
백남준이 꾼 미래 - 2021

1.3 관계성 플랫폼

- 관계성 정보를 제공하기 위한 플랫폼
- 여러 앱들이 합쳐져 하나의 플랫폼을 구성함





DEVIEW 2021

2.Kubernetes

2.1 Kubernetes(K8S)

DEVIEW 2021

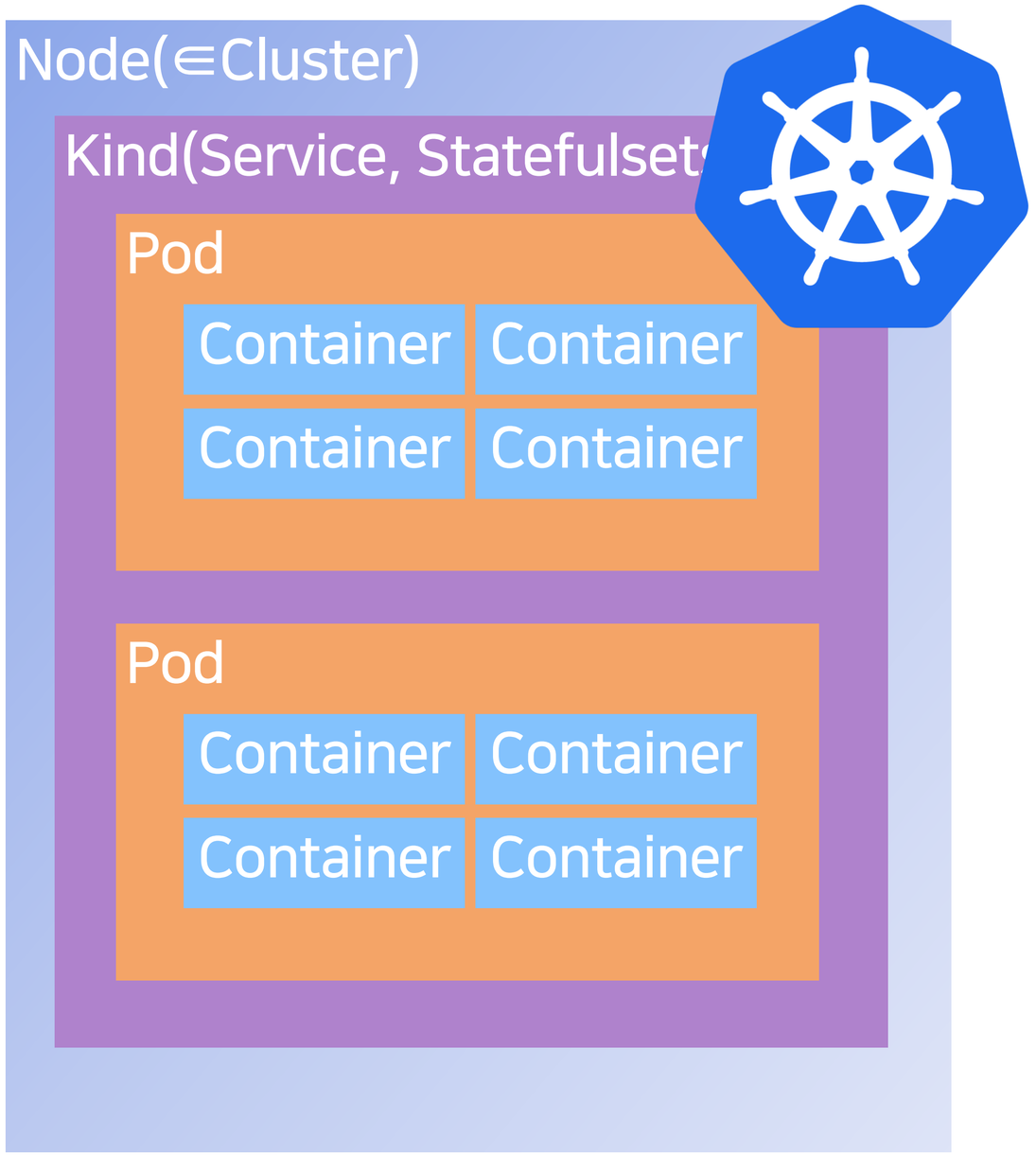
Container Orchestration

- 많은 컨테이너들을 제어

서비스 스펙을 명세서형태로 관리

Node, Pod들로 구성

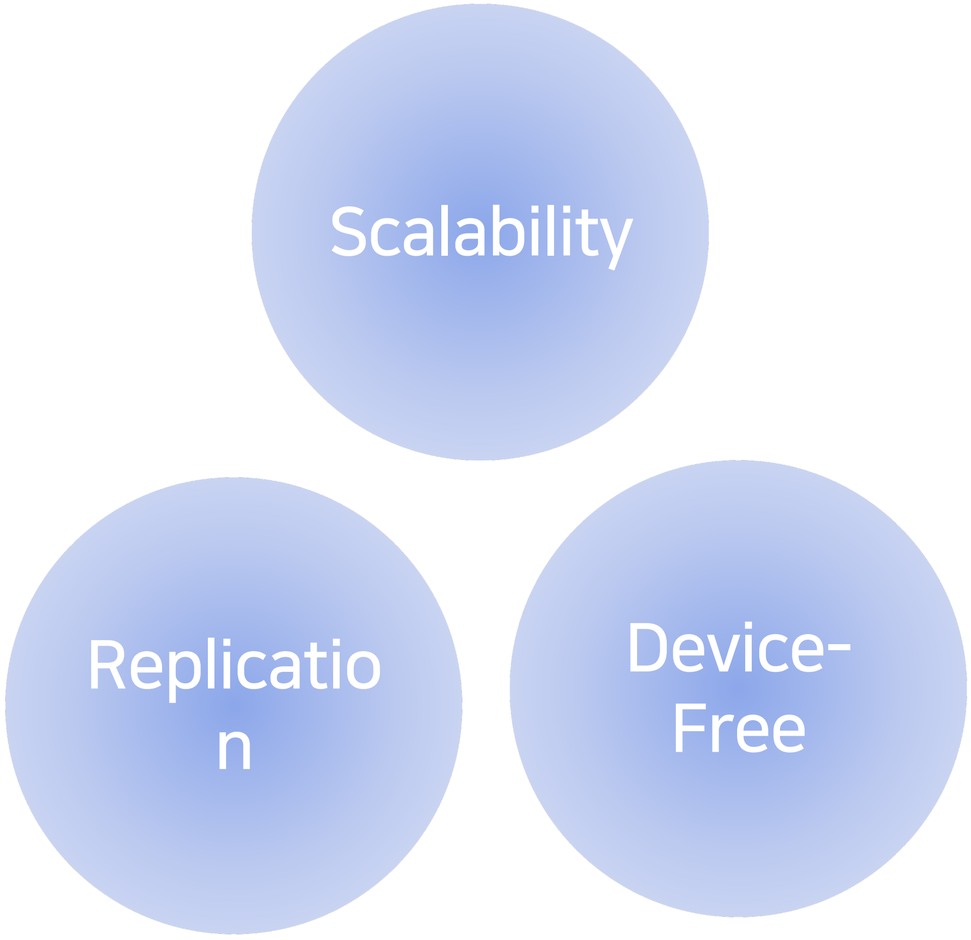
- Node : Cluster
- Kind : 기능 단위
- Pod : K8S 구성체의 최소단위



2.2 물리서버와 비교

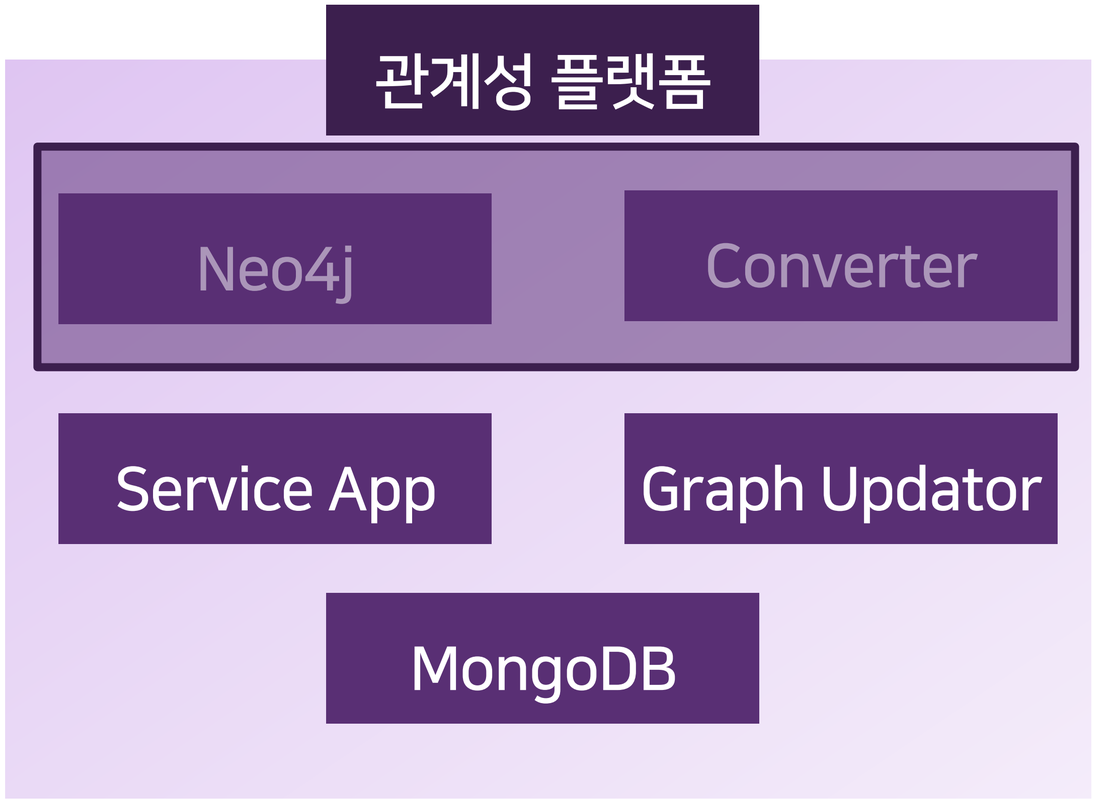
장점
- 서비스 복제 및 리소스 증설이 쉬움
- 다수의 호스트를 관리하기 용이
- 물리서버관리 부담이 없음

단점
- 클러스터 전체 장애 시 대응이 어려움
- 초기 구성 난이도가 높음



2.3 Kubernetes로의 이전

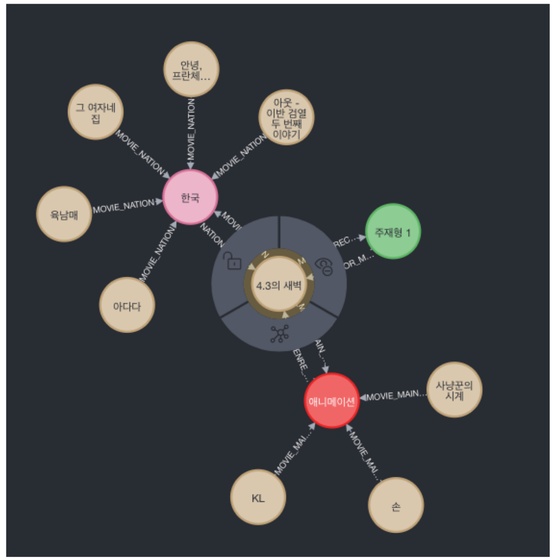
- 관계성 플랫폼에 리소스를 늘려야함
- 물리 서버 기반으로는 확장 과정이 복잡
- Neo4j와 Converter을 먼저 이전



3. Neo4j

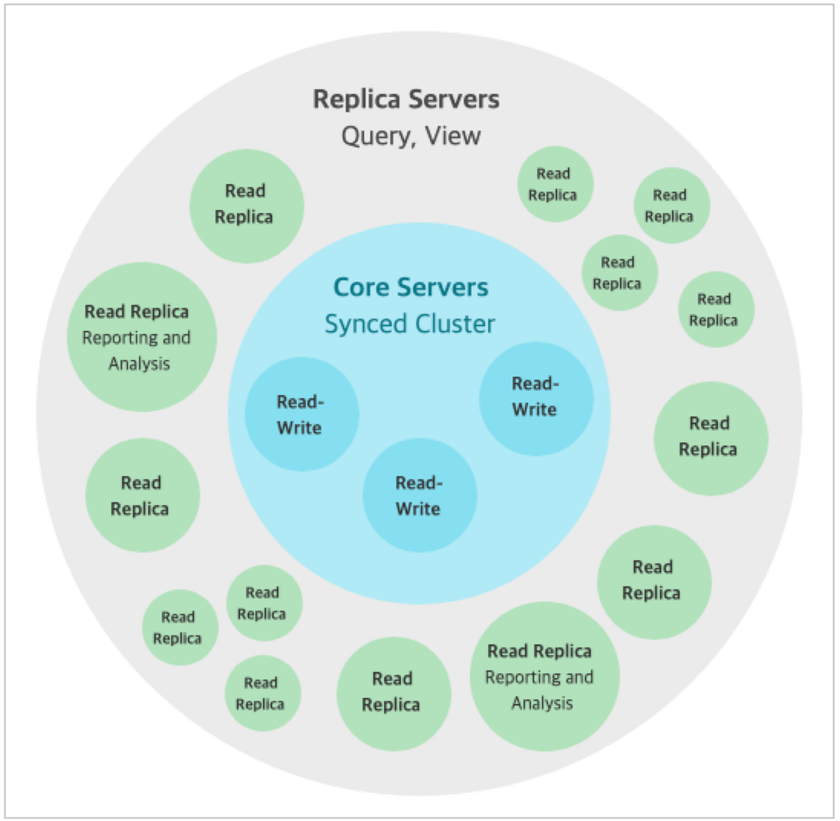
3.1 Neo4j

- Graph DB
- 관계성 정보 저장



3.2 Neo4j on Cluster Environment

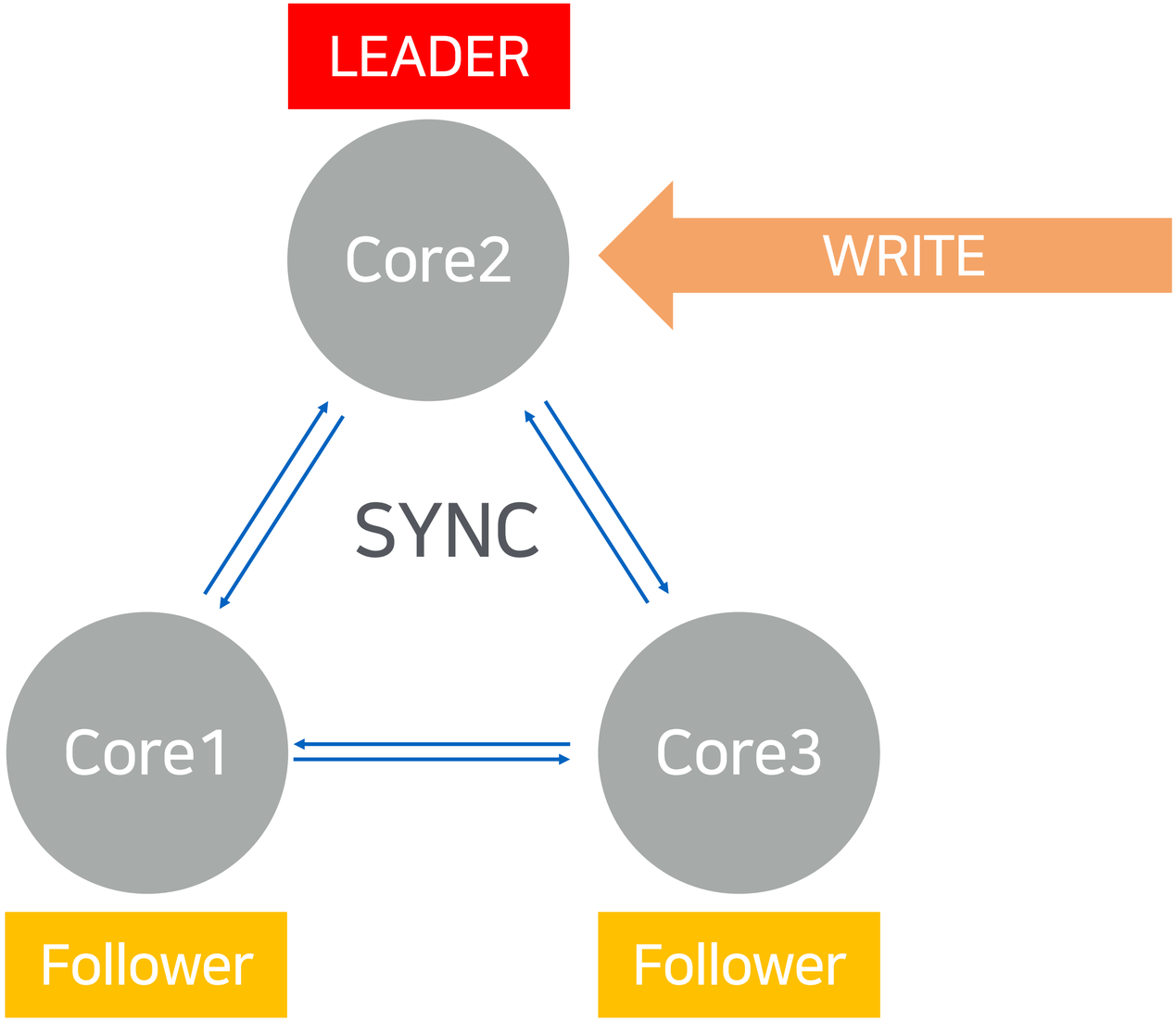
- Core
- Read-Replica



3.3 Neo4j on Cluster Environment

Core

- Leader
- Follower

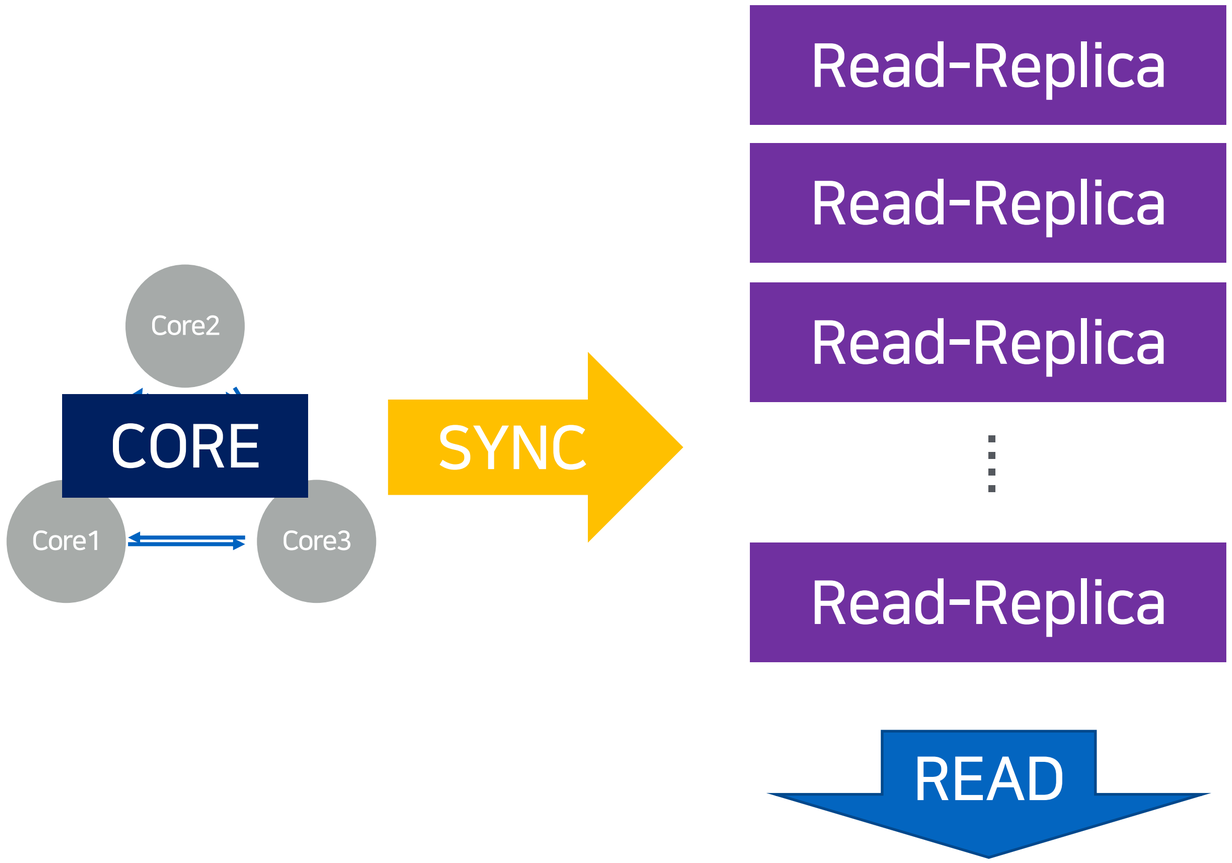


3.4 Neo4j on Cluster Environment

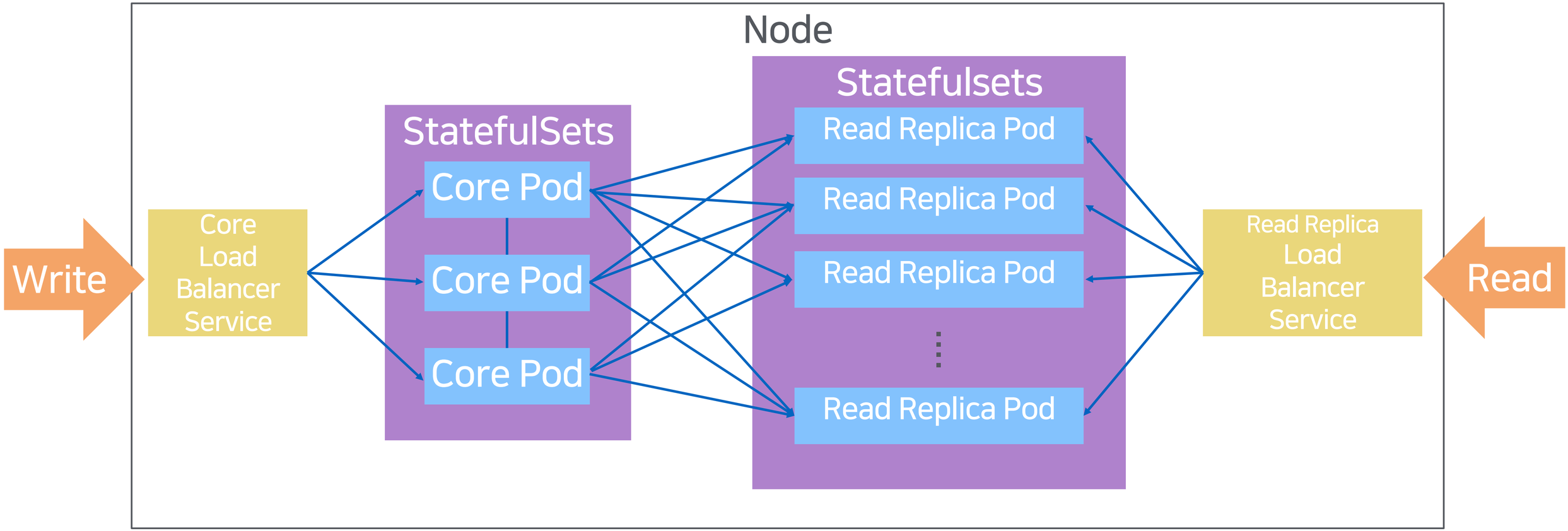
DEVIEW 2021

Read-Replica

- Read Only
- Sync from Core

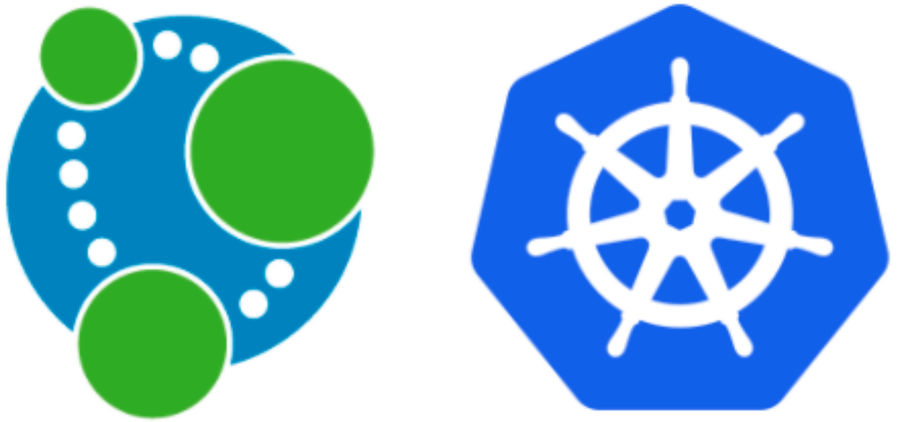


3.5 Neo4j on Cluster Structure



3.6 Helm Chart

- Kubernetes 명세서
- 대부분 시스템이 helm chart는 제공
- 사용하고자하는 환경, 기능에 맞게 수정

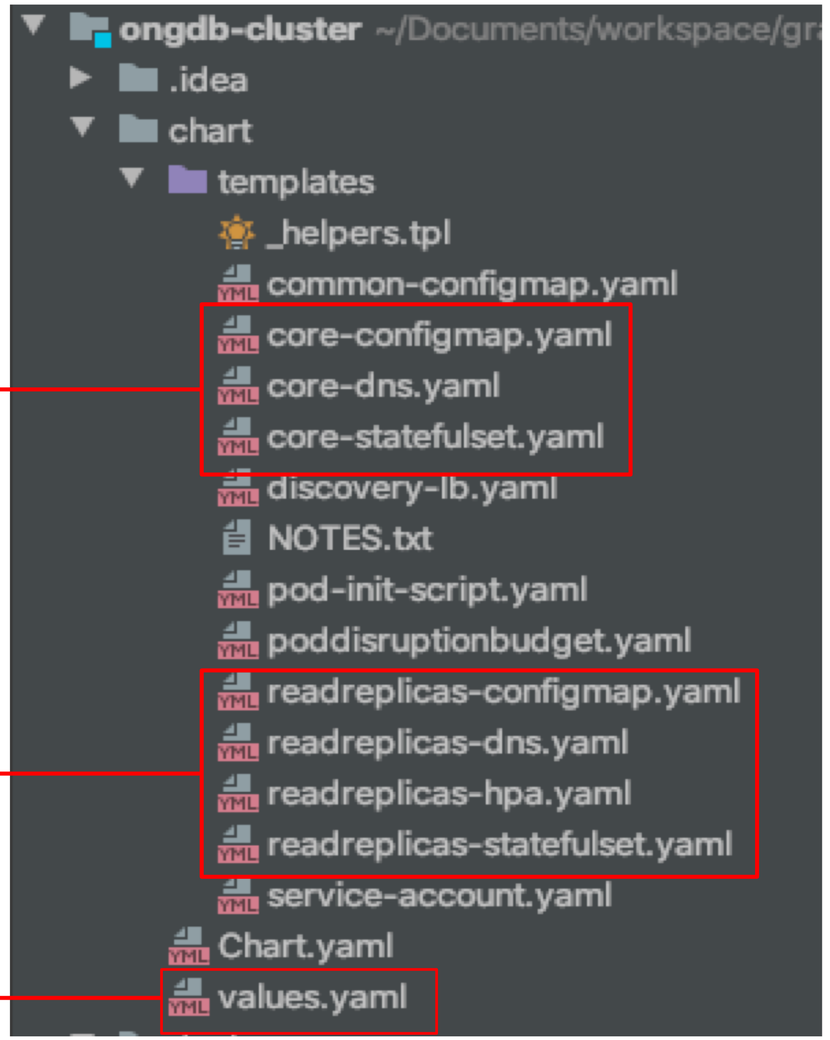


3.7 Helm Chart

DEVIEW 2021

Helm Chart Setting

- Core Pod 설정 값
- Replica Pod 설정 값
- Pod 설정 Parameter들



4.Apply to service

4.1 목표

Kubernetes로 이전 후,

Neo4j가

- 동일 qps에서 유사한 응답 시간을

관계성 플랫폼이

- 동일 qps에서 유사한 응답 시간을

읽기	물리서버	Kubernetes
Neo4j	12ms/450qps	12ms/450qps
관계성 플랫폼	20ms/60qps	20ms/60qps

4.1 목표

- 물리서버가 4대
- Kubernetes는 4배를 감당해야함

읽기	물리서버	Kubernetes
Neo4j	12ms/450qps	12ms/1800qps
관계성 플랫폼	20ms/60qps	20ms/60qps

4.1 목표

- 급격한 qps 증가에 대응
- Neo4j에 3배 트래픽 테스트

읽기	물리서버	Kubernetes
Neo4j	12ms/450qps	12ms/1800qps 20ms/6000qps
관계성 플랫폼	20ms/60qps	20ms/60qps

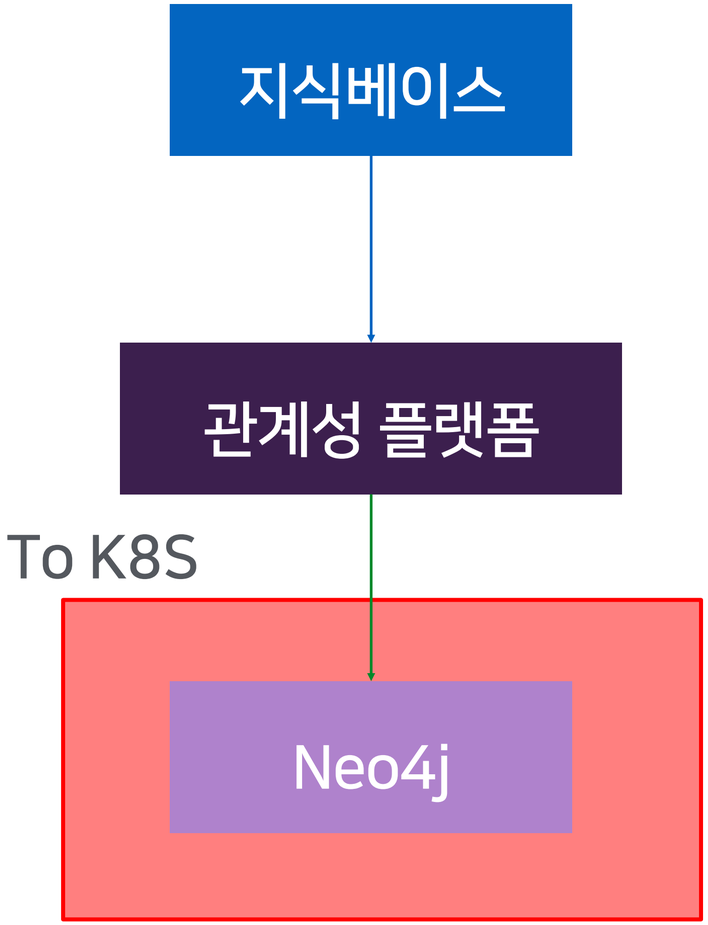
4.2 Stress Test

- Kubernetes가 서비스의 qps를 감당할 수 있는가?
- 벤치마크 : 1일 관계성 플랫폼 접속 로그

4-1. Neo4j 단순 적용

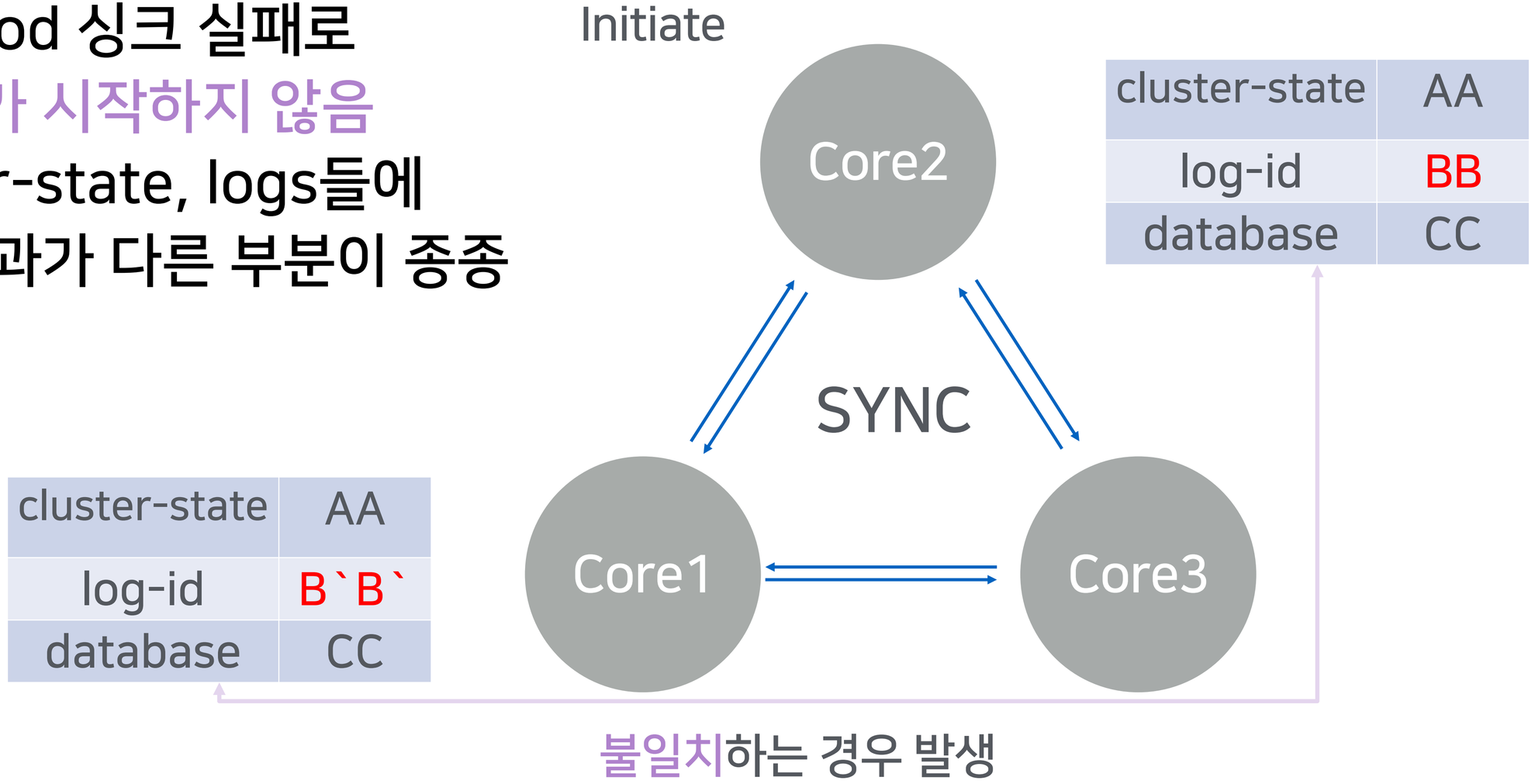
4-1.1 단순 적용안

- 다른 DB 쓰듯이, 붙이기만
- Neo4j를 옮기고 다른 플랫폼을 천천히 옮기기



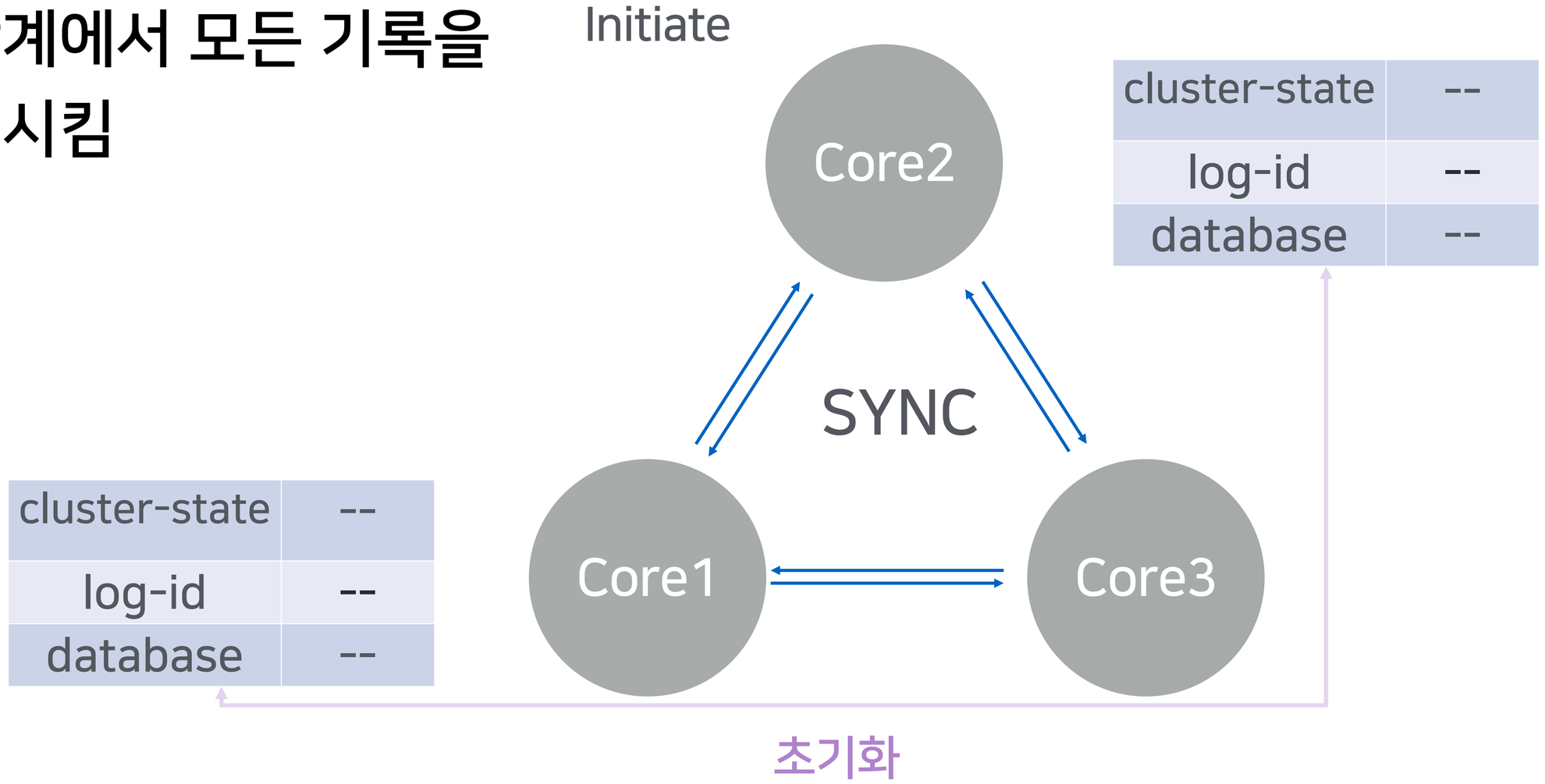
Issue) Core Sync Fail

- Core pod 싱크 실패로 Neo4j가 시작하지 않음
- Cluster-state, logs들에 남은 결과가 다른 부분이 종종 발생

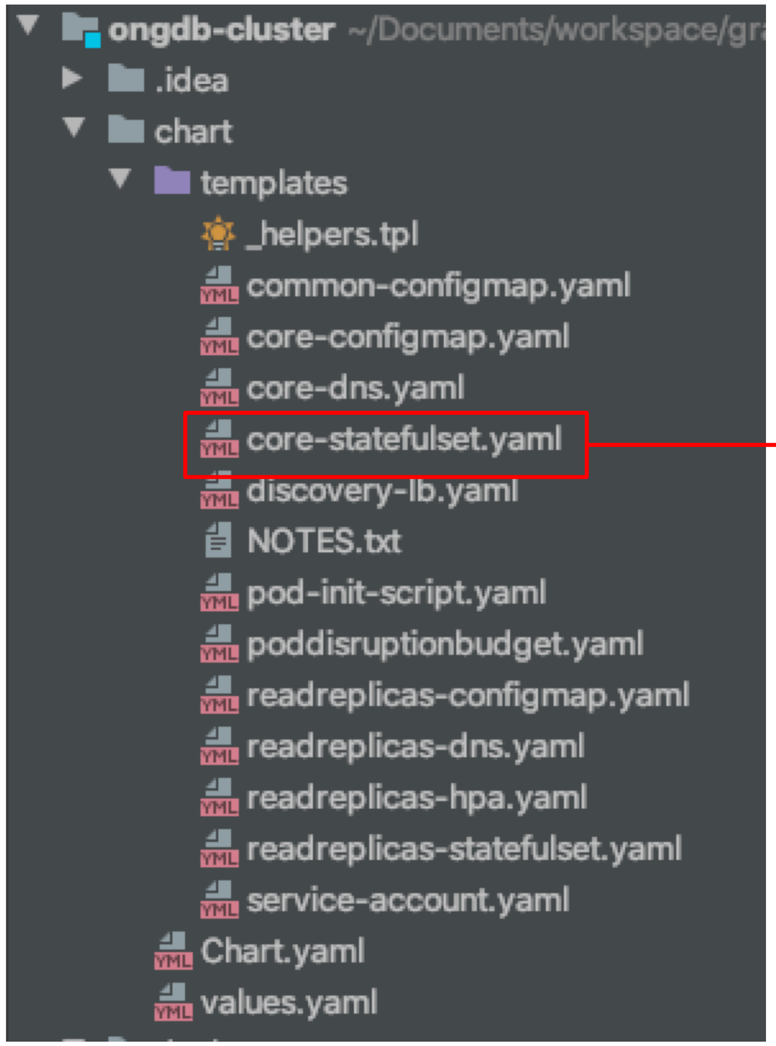


Solution) Sync Data Initialize

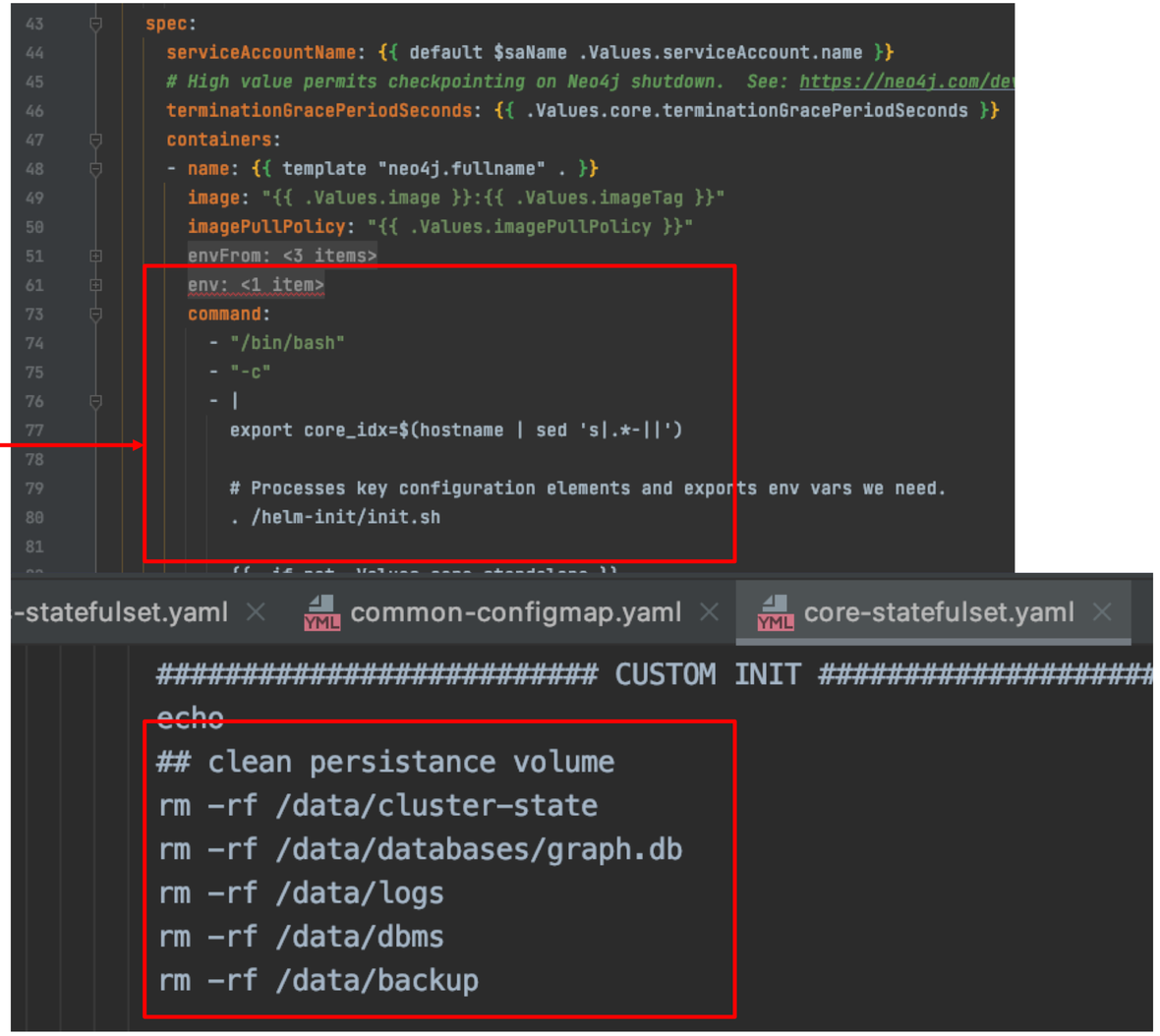
- 시작 단계에서 모든 기록을 초기화 시킴



Solution) Helm Chart



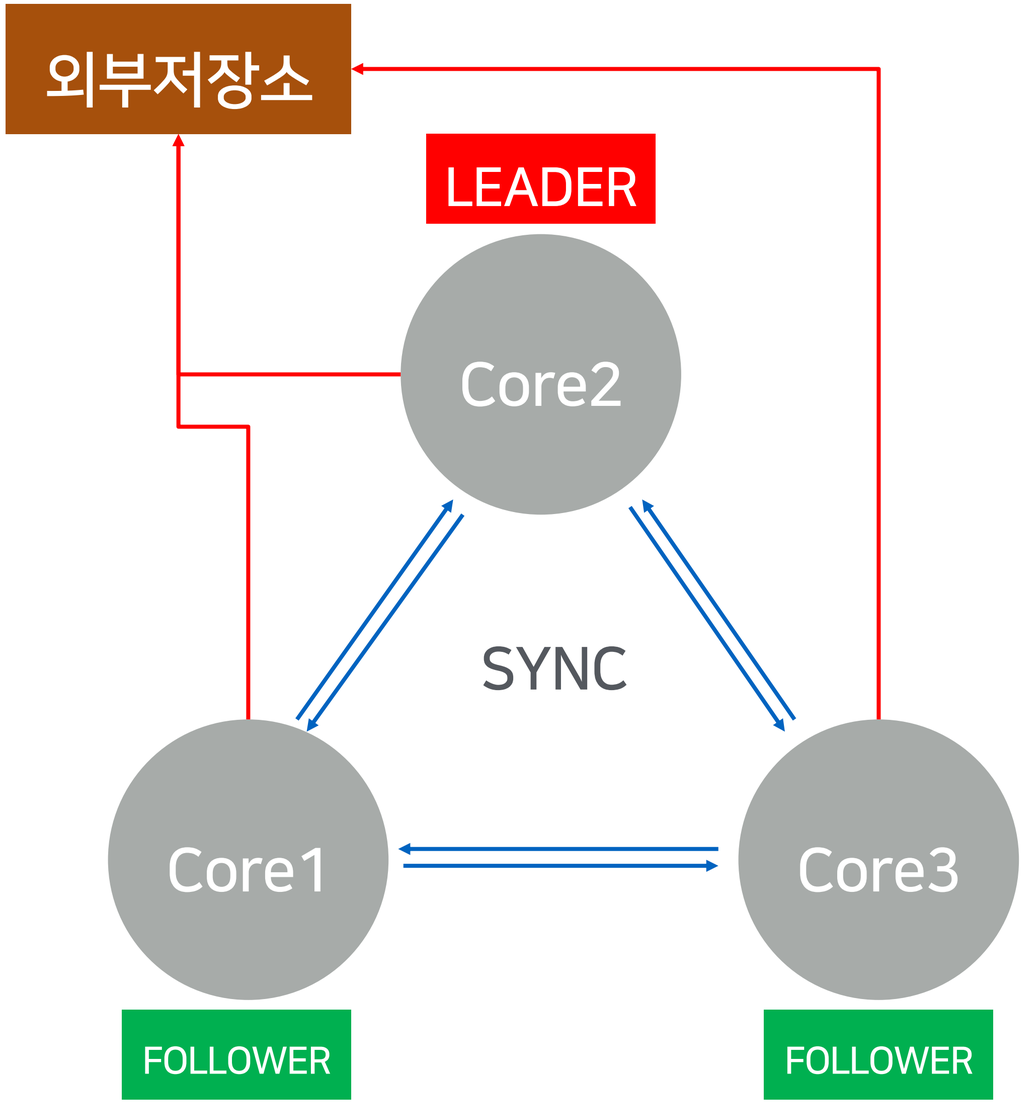
helm chart 수정파일



초기화 코드 삽입

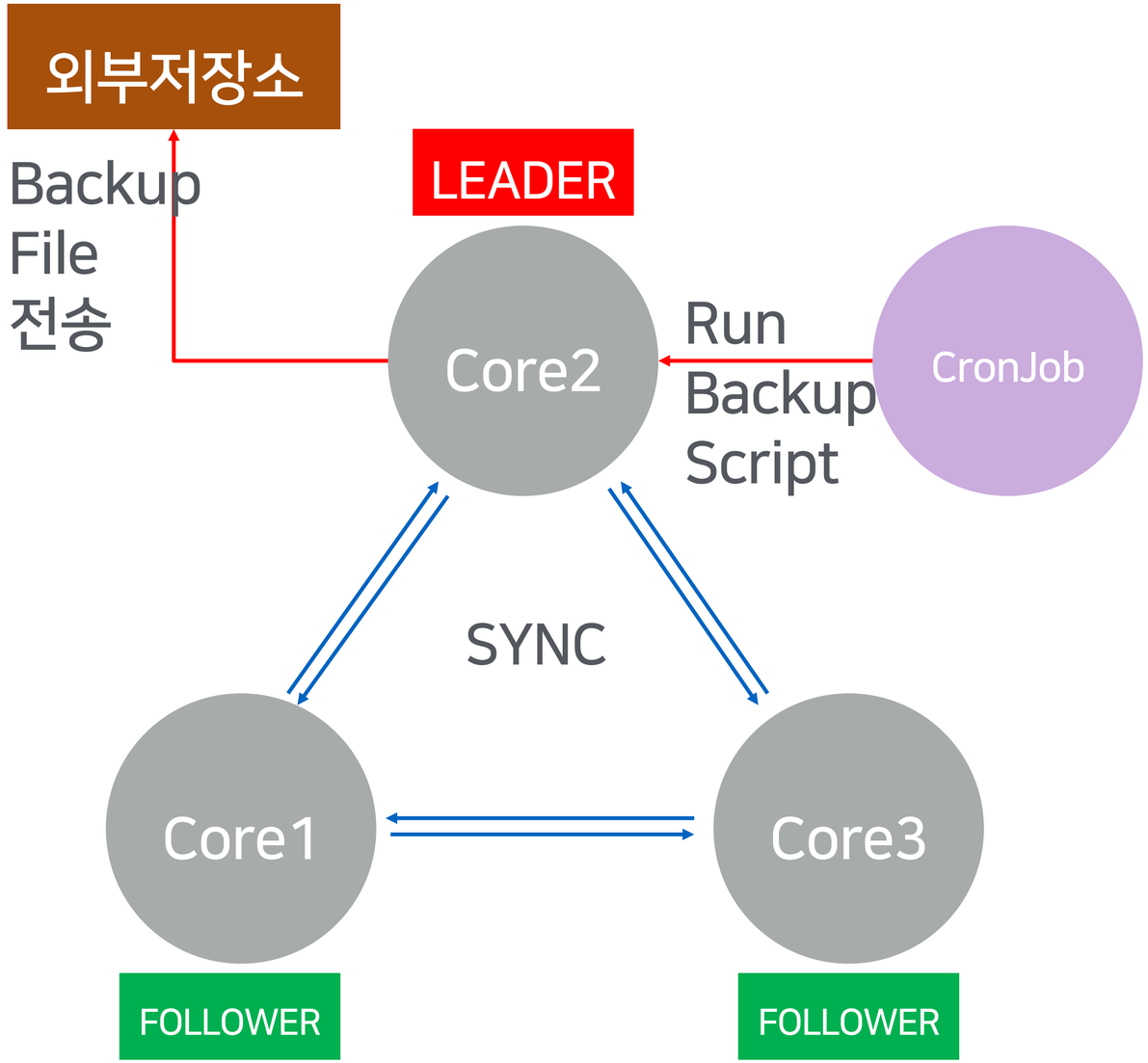
Issue) 백업/복구 시의 싱크 오류 위험

- 복구 시 싱크 오류가 생기지 않는 백업 방식 필요
- 정기적인 backup



Solution) 싱크 오류 없는 백업/복구

- CronJob Pod 사용
- 특정 core에 신호를 보내서, backup 스크립트를 실행하도록 작성

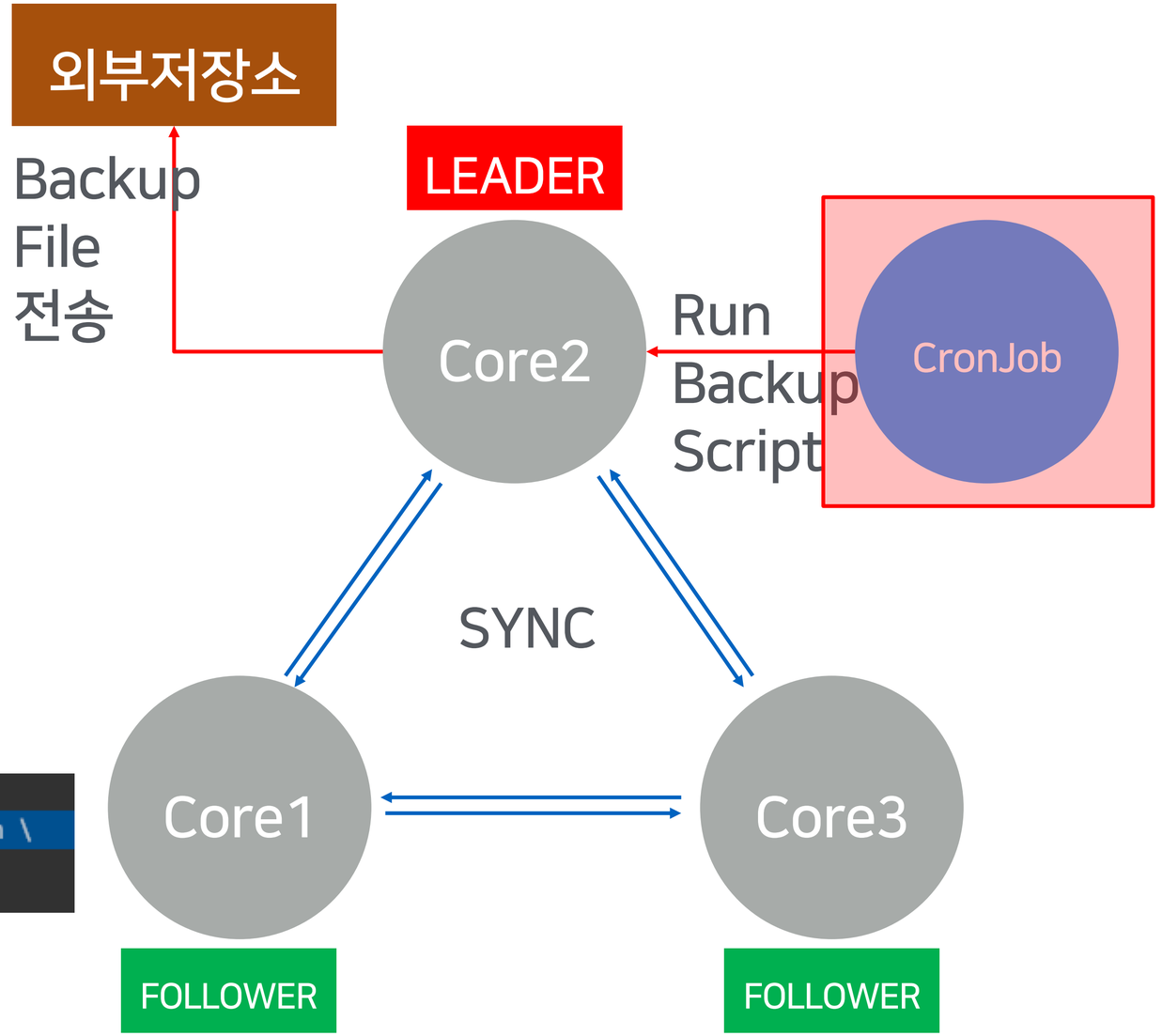


Solution) 싱크 오류 없는 백업/복구

backupJob.yaml | readreplicas-statefulset.yaml | common-configma

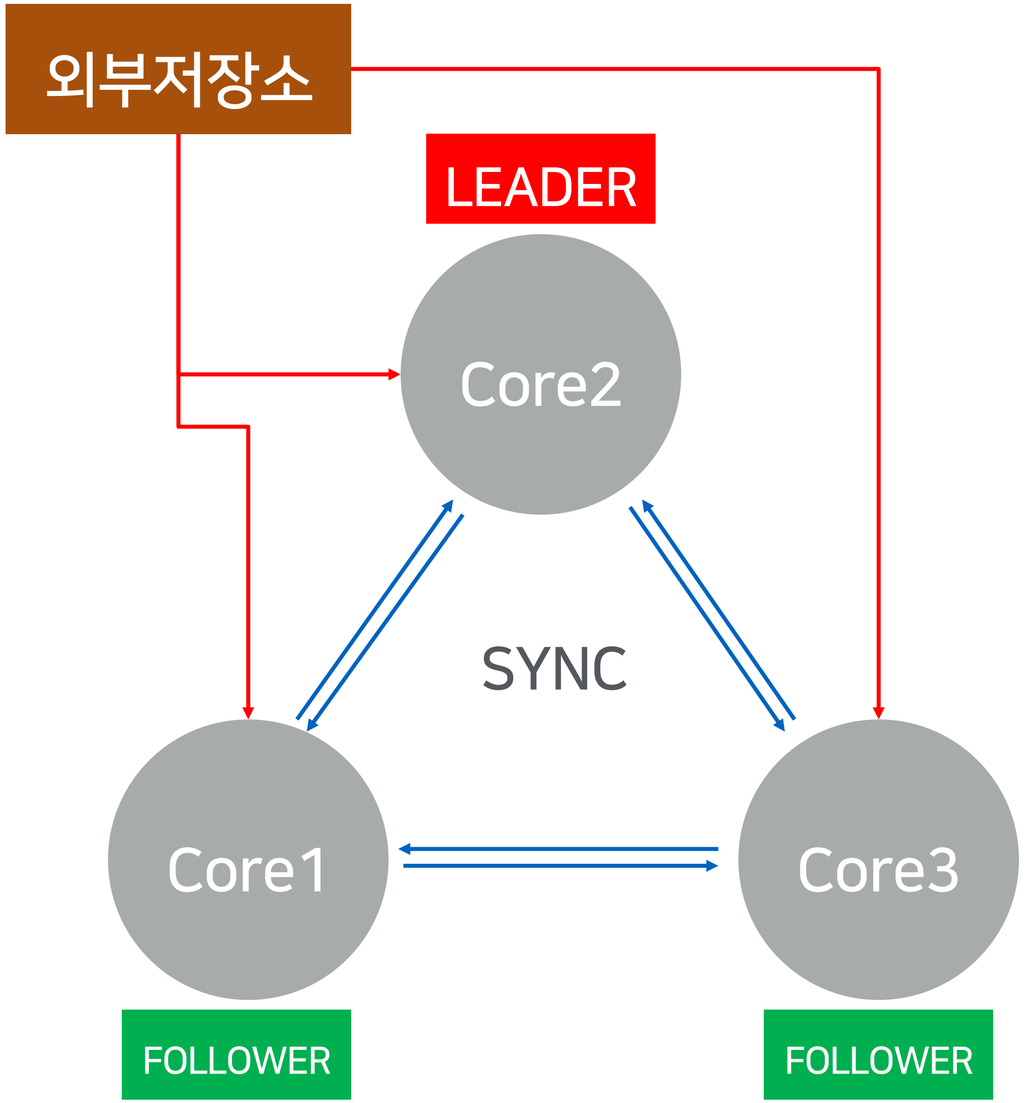
```yaml
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: backup-ongdb
spec:
  schedule: "*/30 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: backup-ongdb
            image: reg.navercorp.com/ongdb/cronjob:latest
            imagePullPolicy: Always
            command:
              - /bin/bash
              - -c
              - |
```

```
echo "BACKUP START"
kubectl exec ongdb-default-ongdb-core-0 -it -- bash -c "sh /scripts/backup.sh \
  a.lookup.nubes.navercorp.com:8080 stella-devops-ar1-airflow-1"
echo "BACKUP END"
```



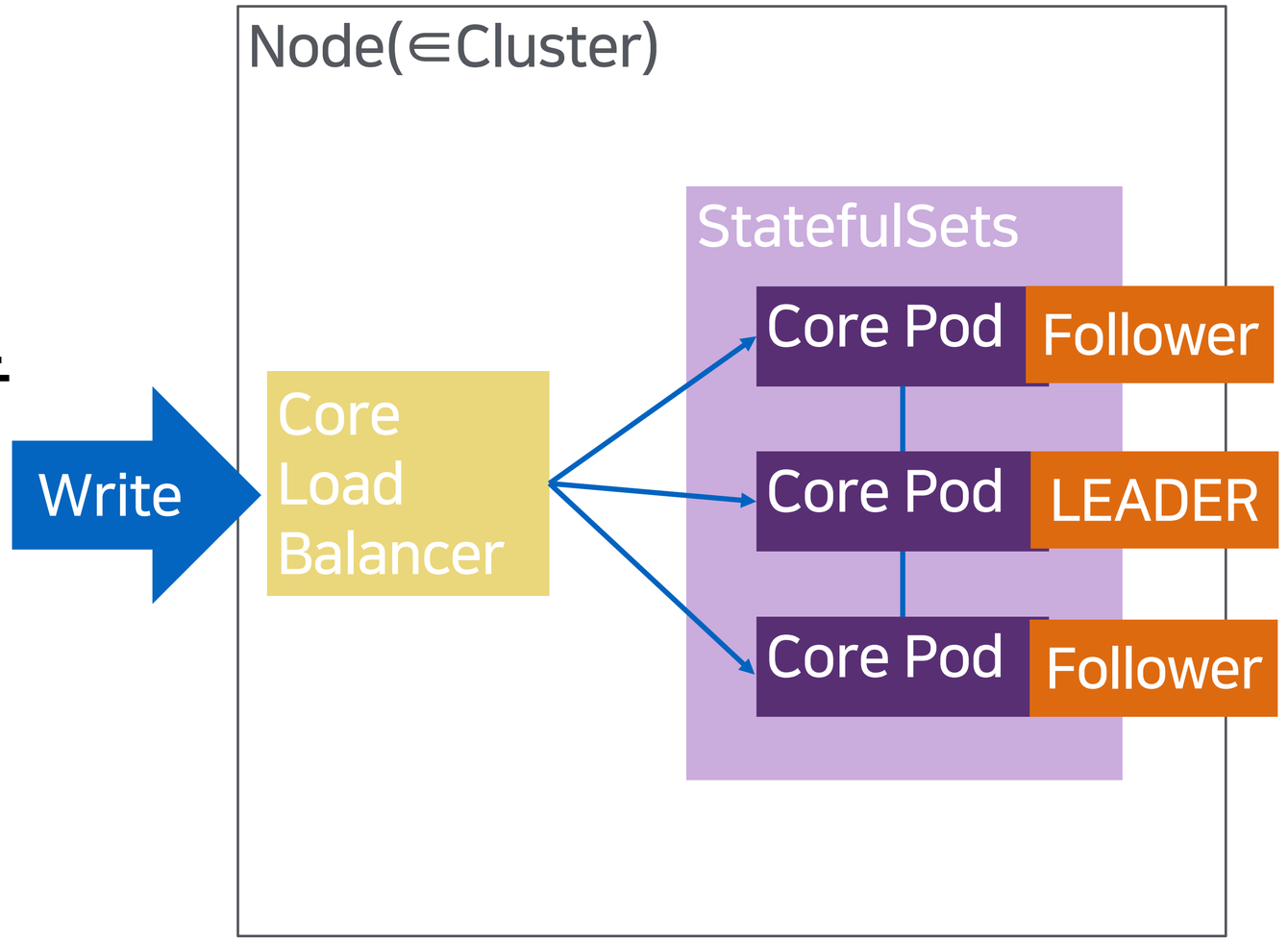
Solution) 싱크 오류 없는 백업/복구

- Back Up 된 파일 기준
- core init에서 복구
- 외부저장소 정보는 Helm chart에 기입



Issue) Critical Write Error

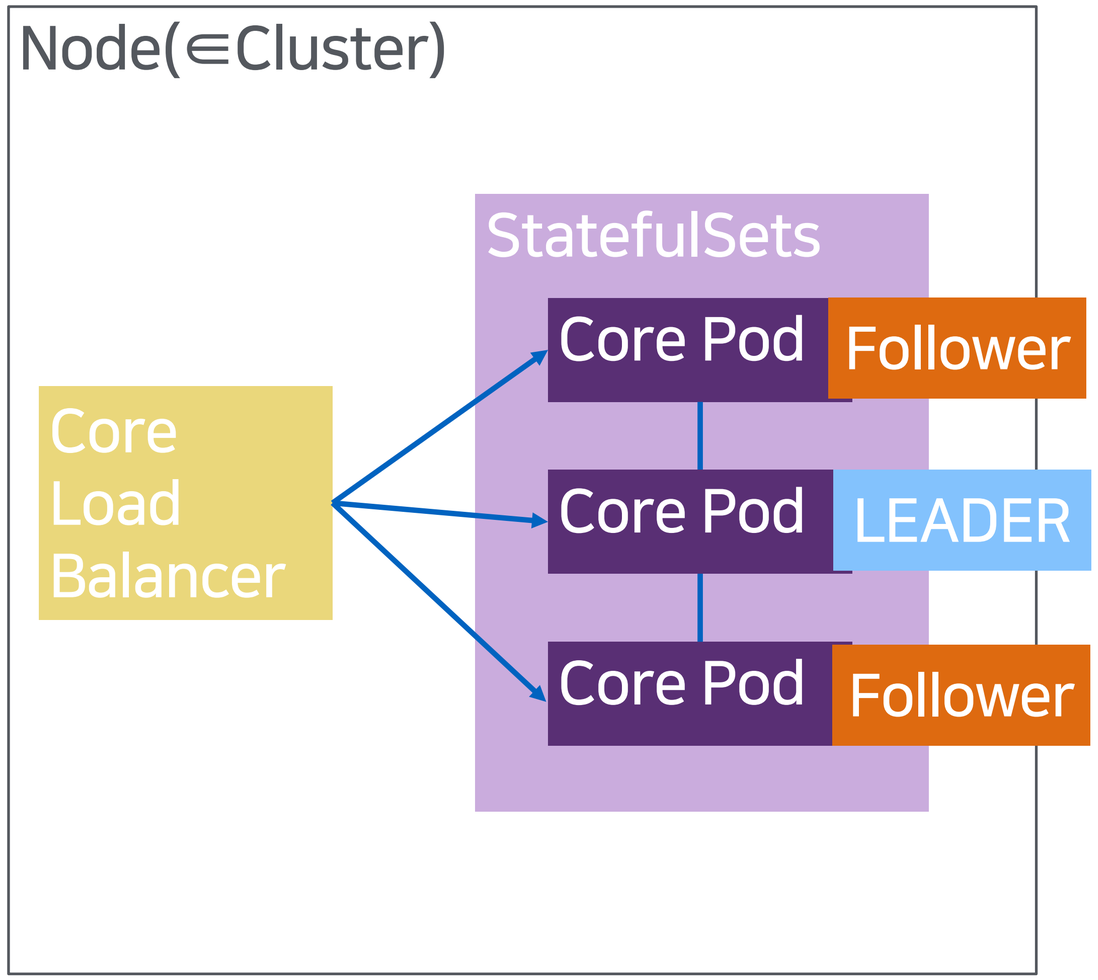
- Write Load Balancer에서 오류
- Leader가 아닌 Follower로 보낼 때가 존재
- 기존 neo4j가 가진 버그



Issue) Critical Write Error

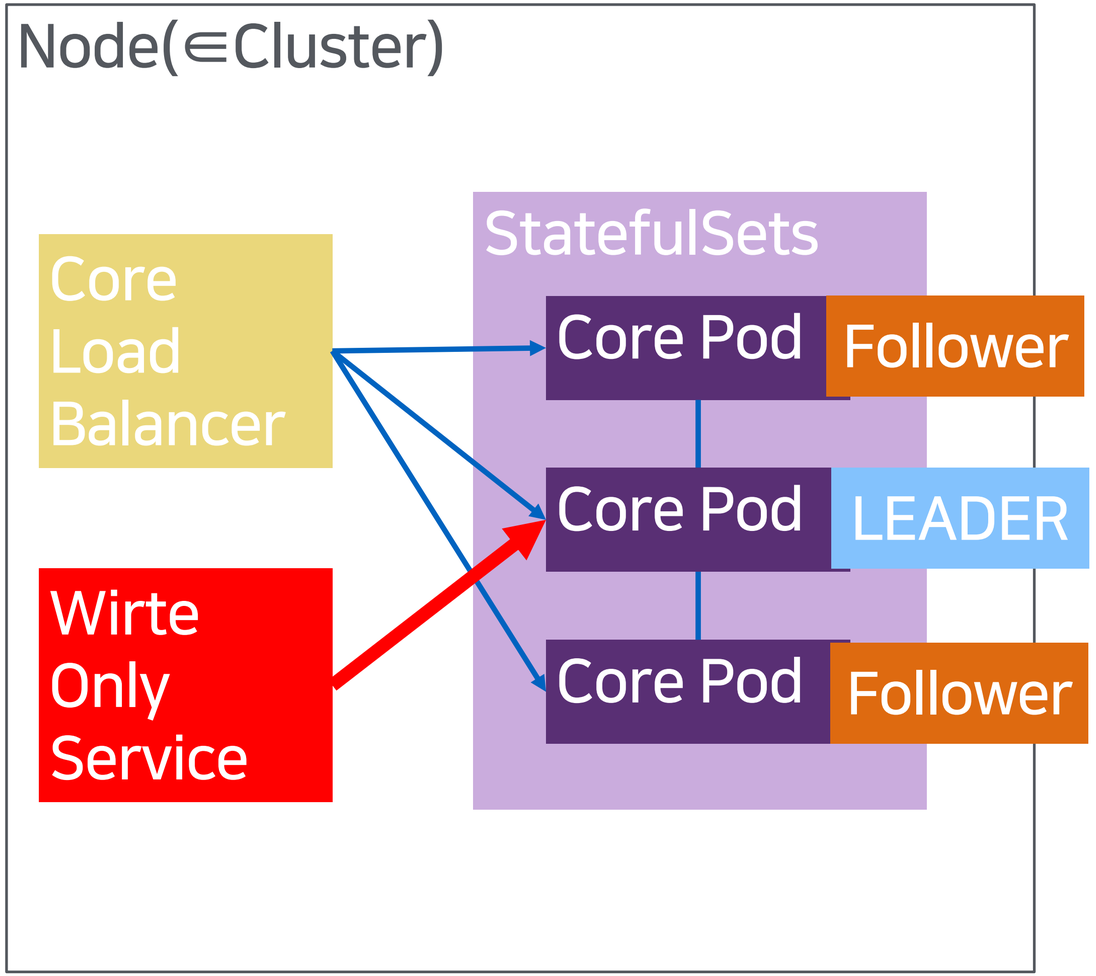
- 1분 간격으로 Leader Pod을 서치
- 해당 pod에 label로 마킹

```
if [ ${writable} == "true" ]; then
    leaderName=`kubectl get pods -o wide | grep ${element} | awk '{print $1}'`
    kubectl label pods ${leaderName} writable=true --overwrite
    echo leader ${element}
```



Issue) Critical Write Error

- 쓰기 전용 서비스 생성



Issue) Helm Code

- 기존 코드 외에 추가로 설정
- Service/CronJob을 추가함

core-leader-checker.yaml

```yaml
kind: Service
apiVersion: v1
metadata:
  name: ongdb-core-write
  labels:
    ncc.navercorp.com/instance-service: ongdb-core-writer
spec:
  selector:
    writable: "true"
  ports:
    - name: http
      protocol: TCP
      port: 7474
    - name: bolt
      protocol: TCP
      port: 7687
  type: LoadBalancer

---
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: leader-checker
spec:
  schedule: "*/1 * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: leader-checker
              image: reg.navercorp.com/ongdb/cronjob:latest
              imagePullPolicy: Always
              command:
                - /bin/bash
                - -c
                - |
```

Test) Write & Read Test

읽기

Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	21616304	0	0.00%	0.50	0	377	0.00	1.00	1.00	1.00	6006.34	1 20.32	1970.83
neo4j request	21616304	0	0.00%	0.50	0	377	0.00	1.00	1.00	1.00	6006.34	1 20.32	1970.83

쓰기

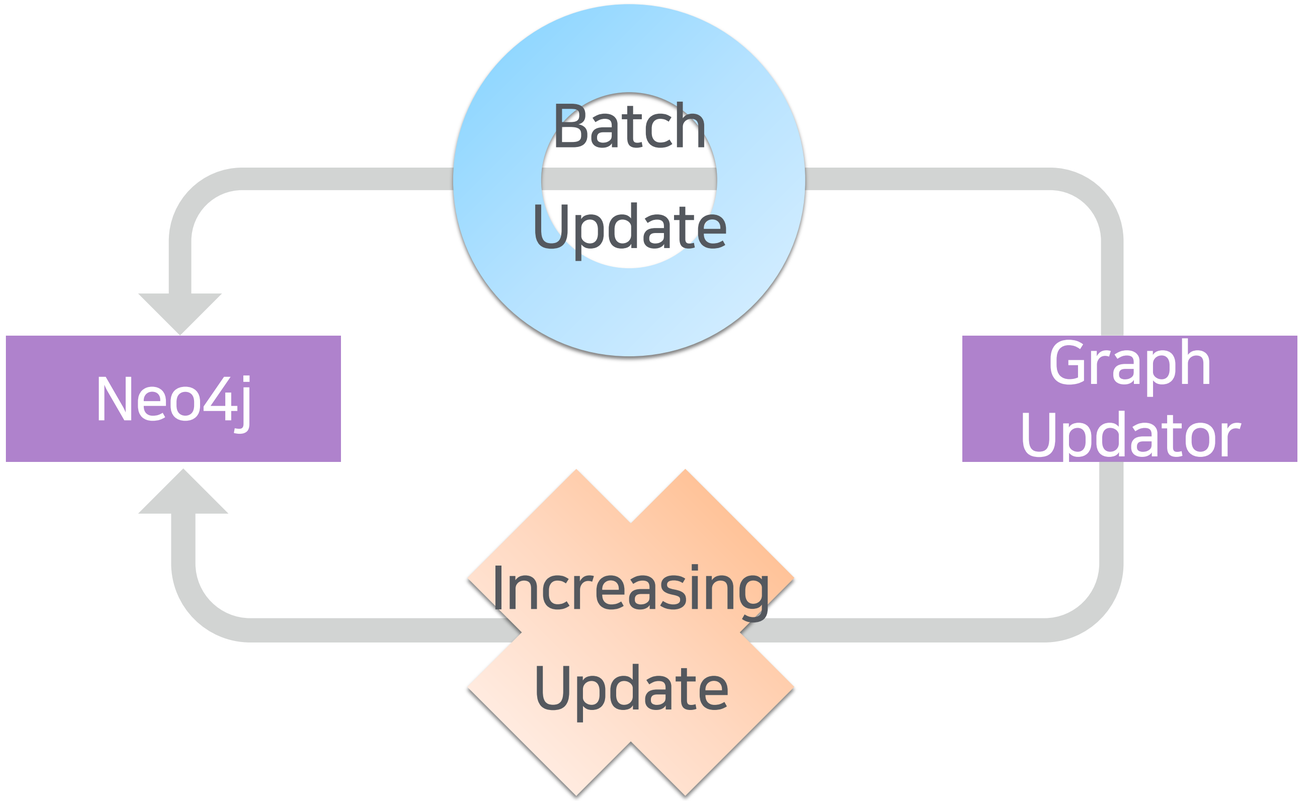
Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	1508419	12905	0.86%	466.56	1	2880	578.00	819.90	1065.00	1789.97	418.97	162.38	243.89
HTTP Request	1508419	12905	0.86%	466.56	1	2880	578.00	819.90	1065.00	1789.97	418.97	162.38	243.89

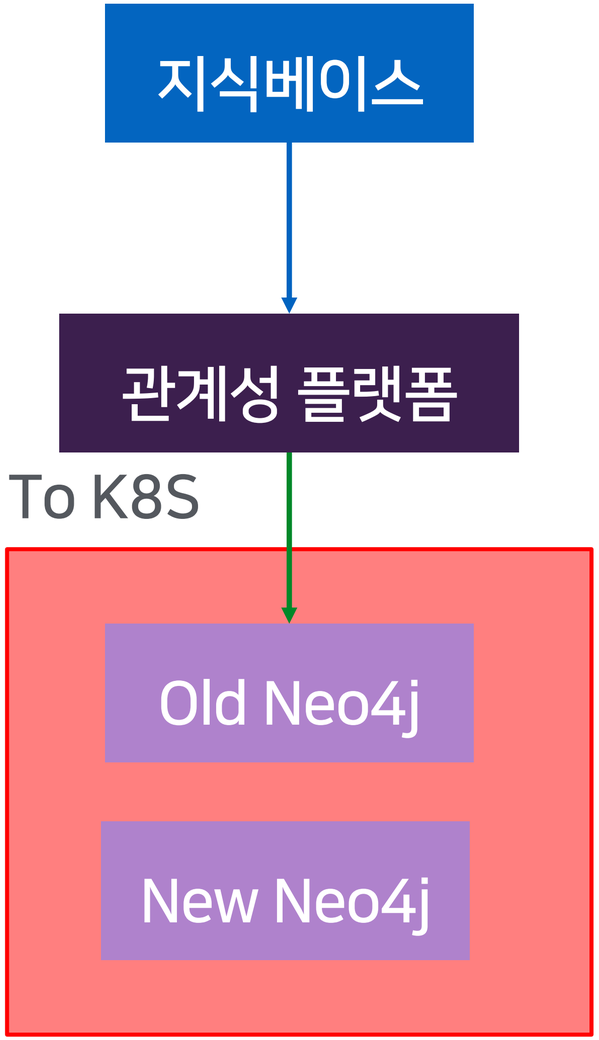
Issue) 관계성 플랫폼은 Batch Update!



4-2. Neo4j 교체 방식

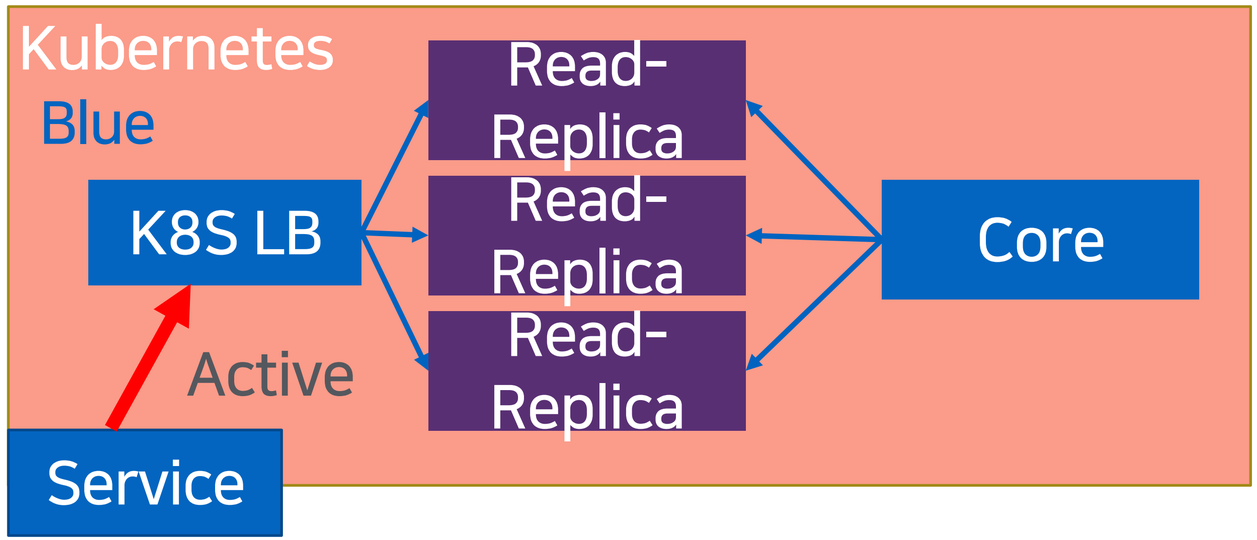
4-2.1 DB-Change

- 일정 주기로, DB를 교체하며 서비스
- 기존 서빙 중인 DB를 제거하고 새로운 데이터가 들어간 DB를 배포하는 방법



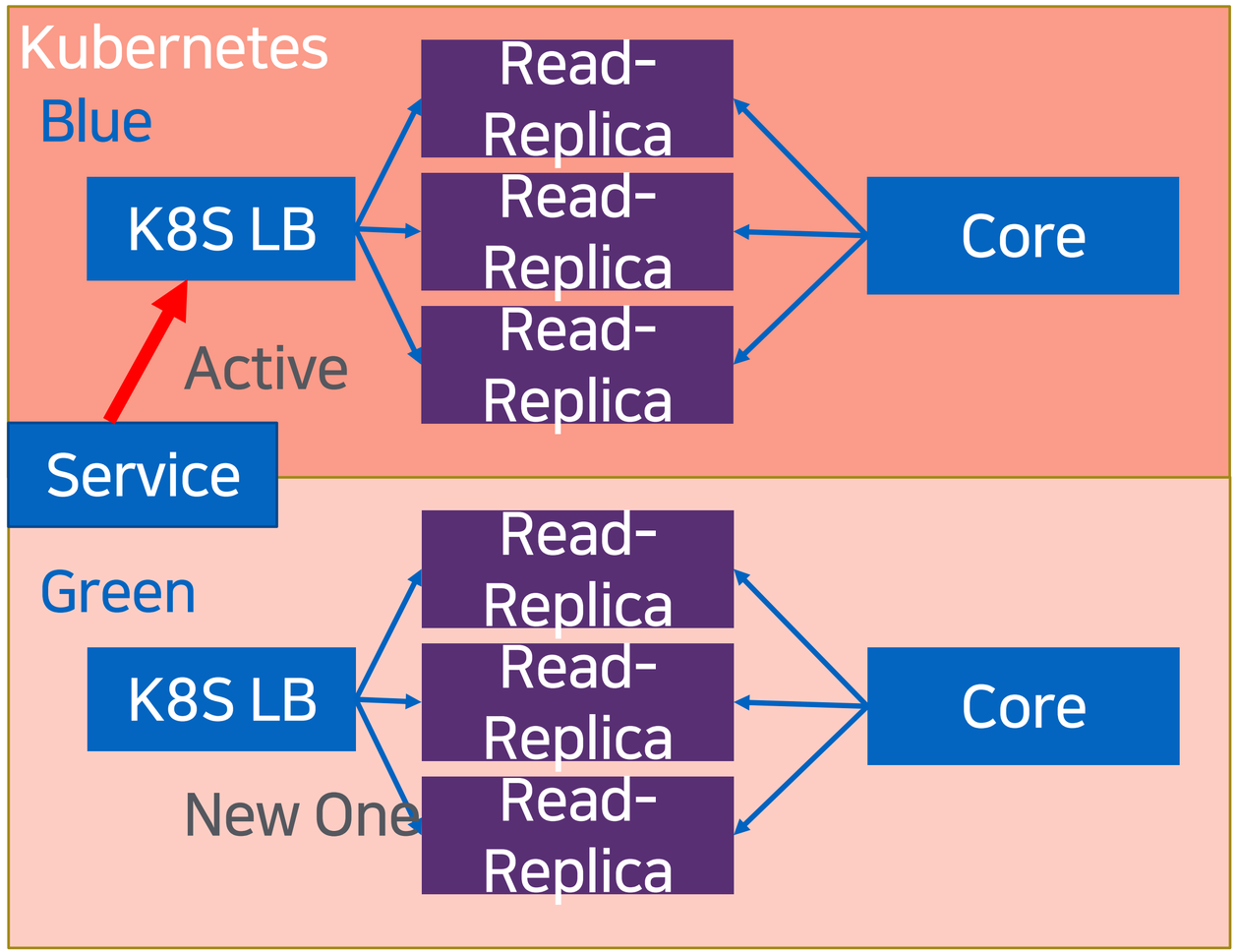
4-2.1 DB-Change

- Blue/Green 배포 방식을 활용해 구현



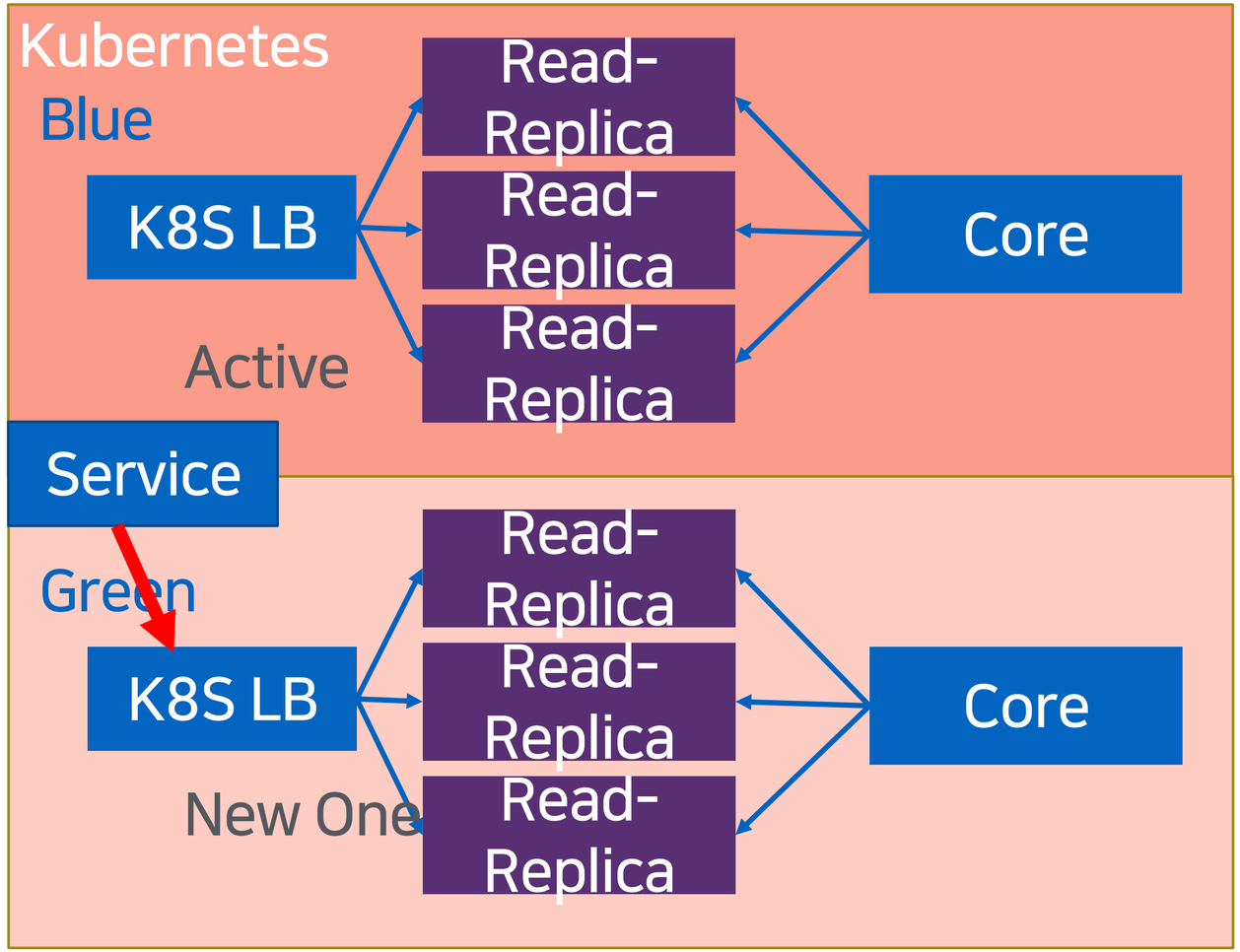
4-2.1 DB-Change

- Blue/Green 배포 방식을 활용해 구현



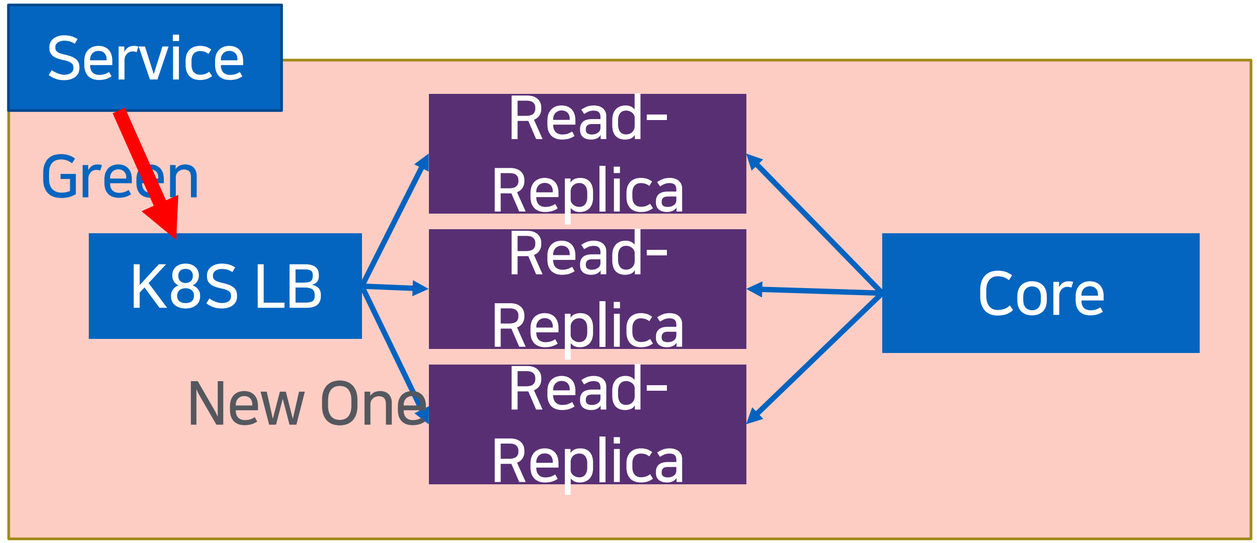
4-2.1 DB-Change

- Blue/Green 배포 방식을 활용해 구현

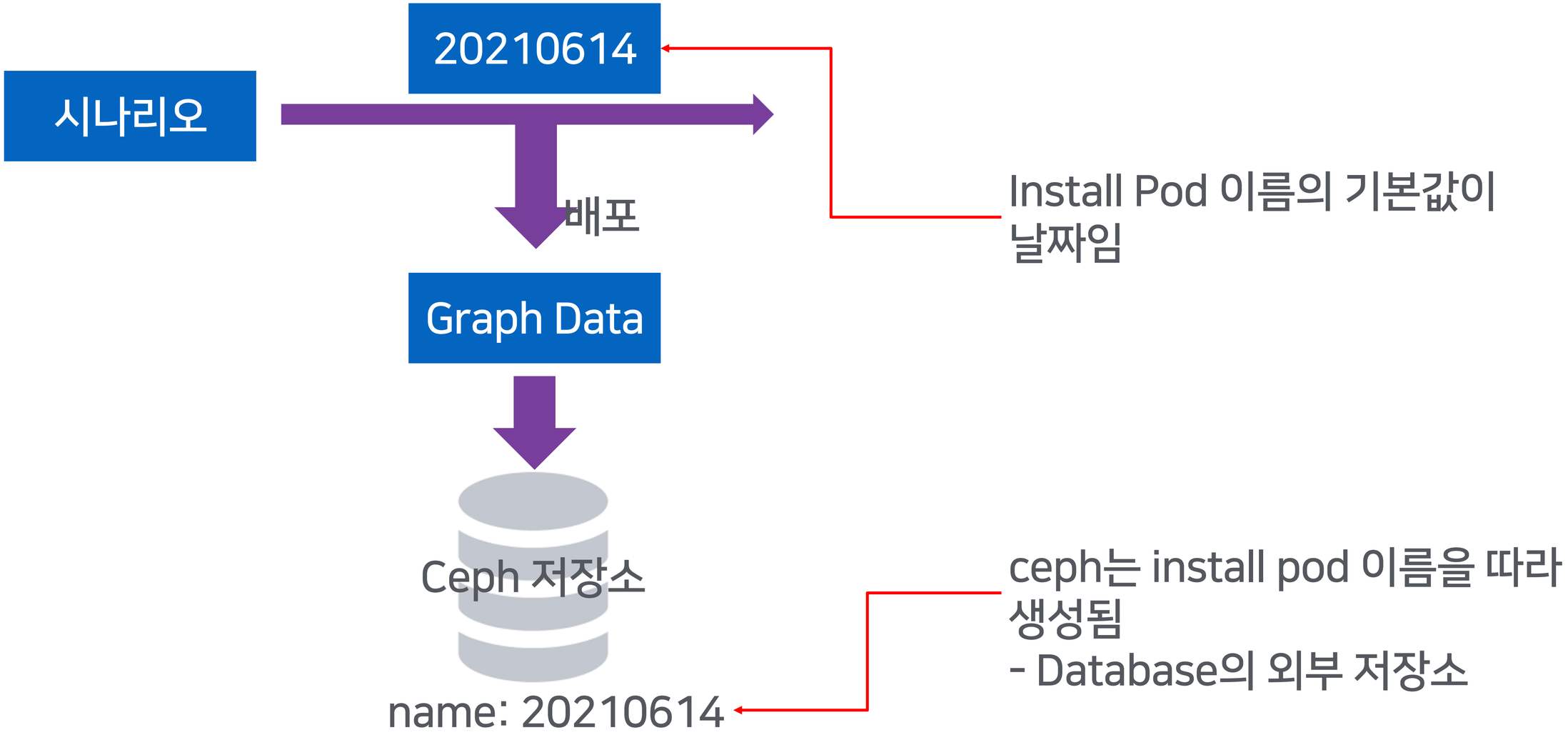


4-2.1 DB-Change

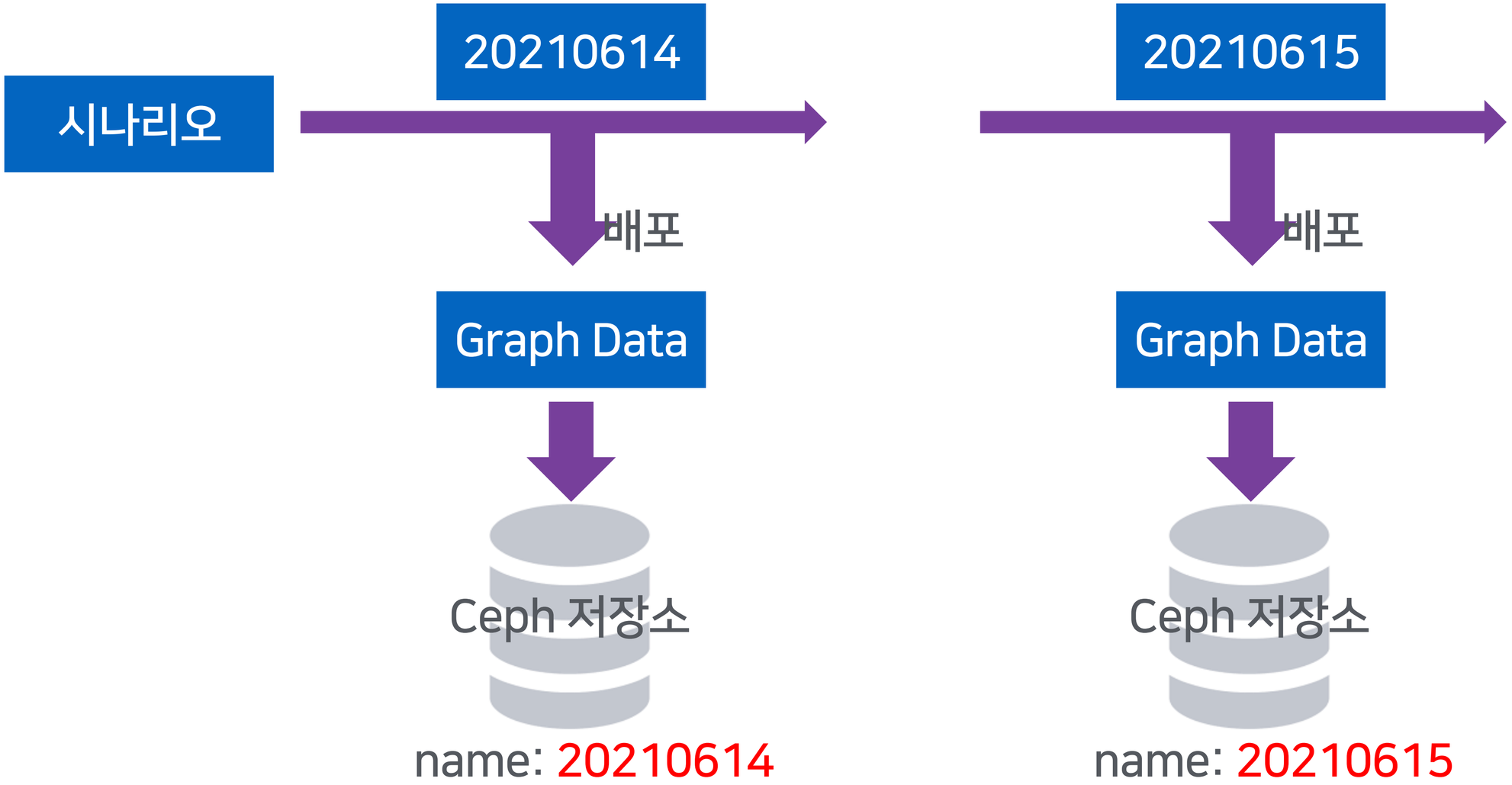
- Blue/Green 배포 방식을 활용해 구현



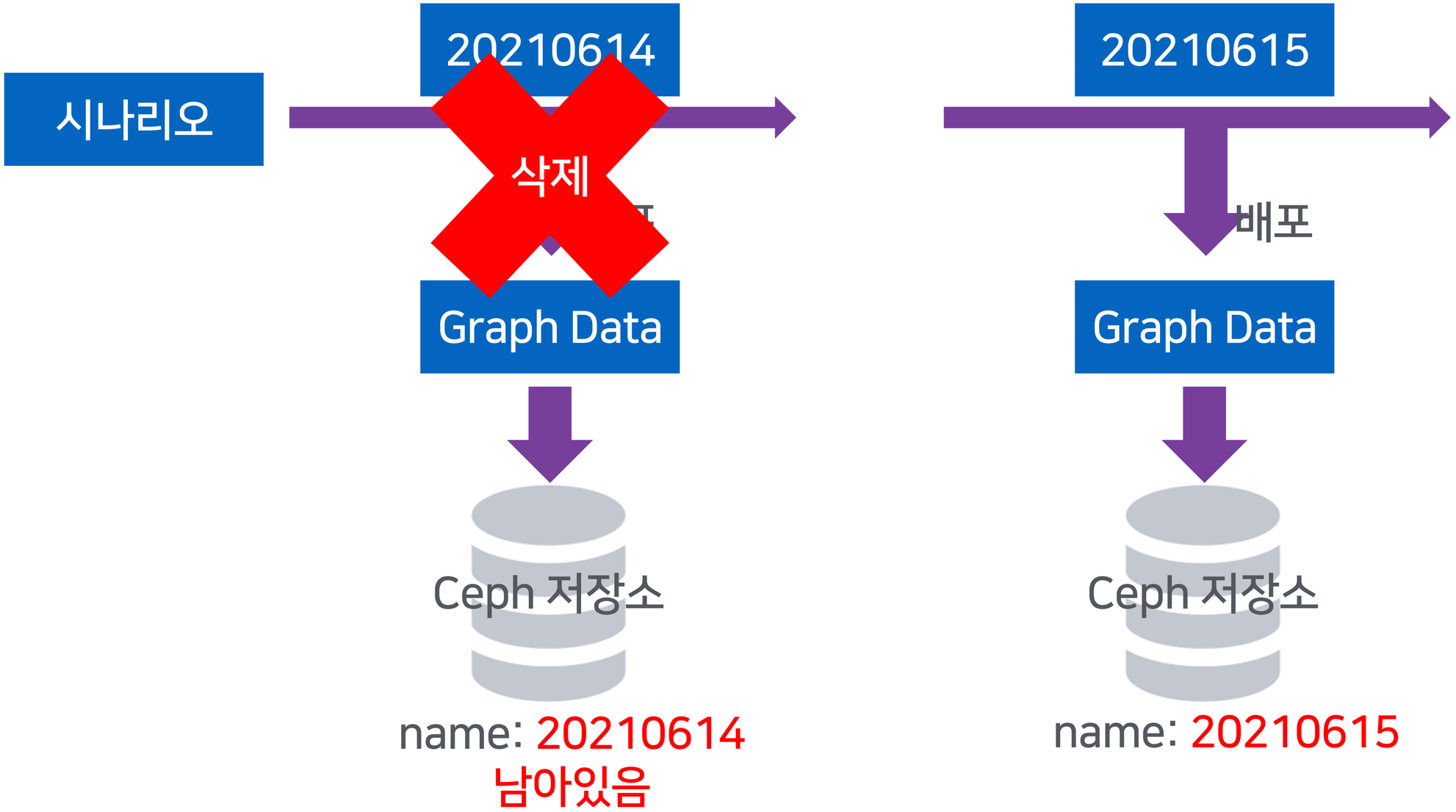
Issue) pvc 저장소 증식



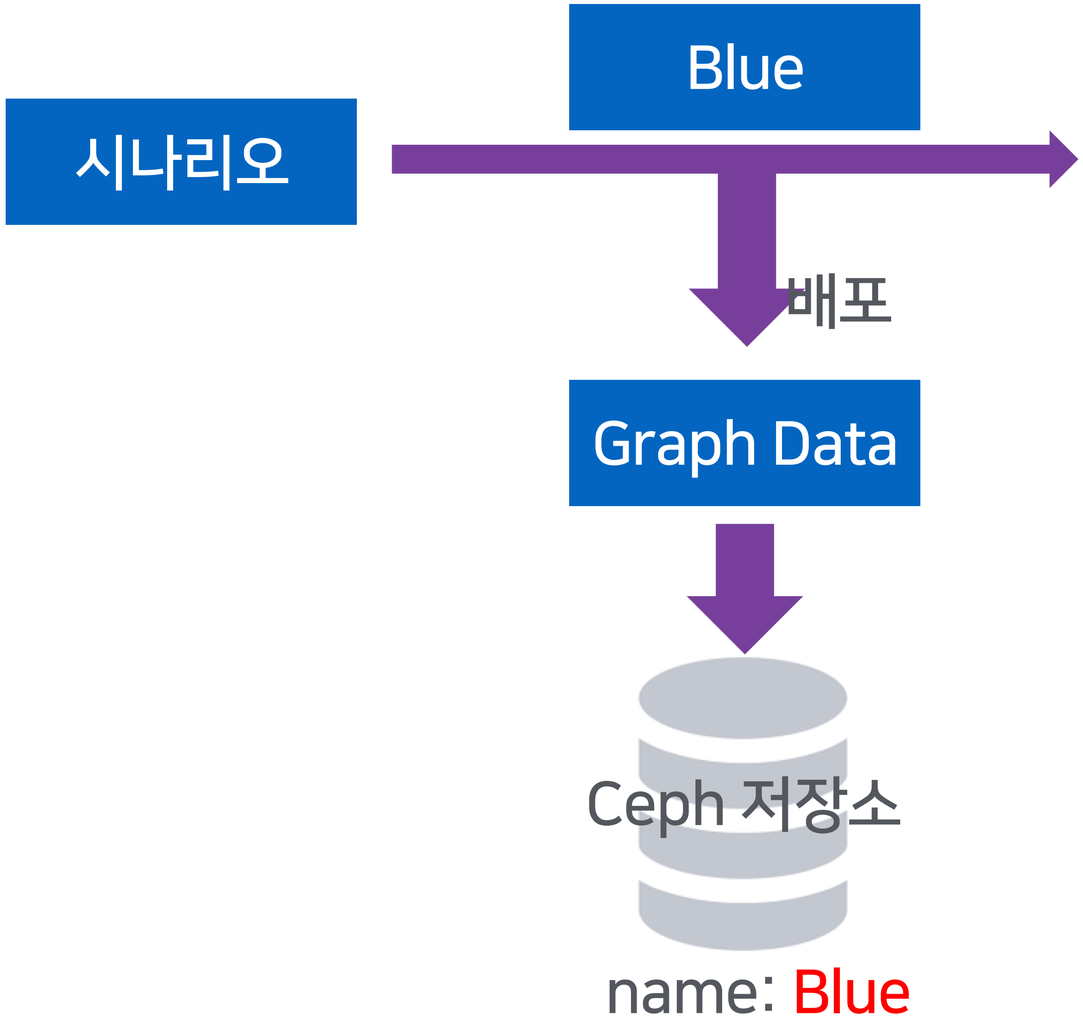
Issue) pvc 저장소 증식

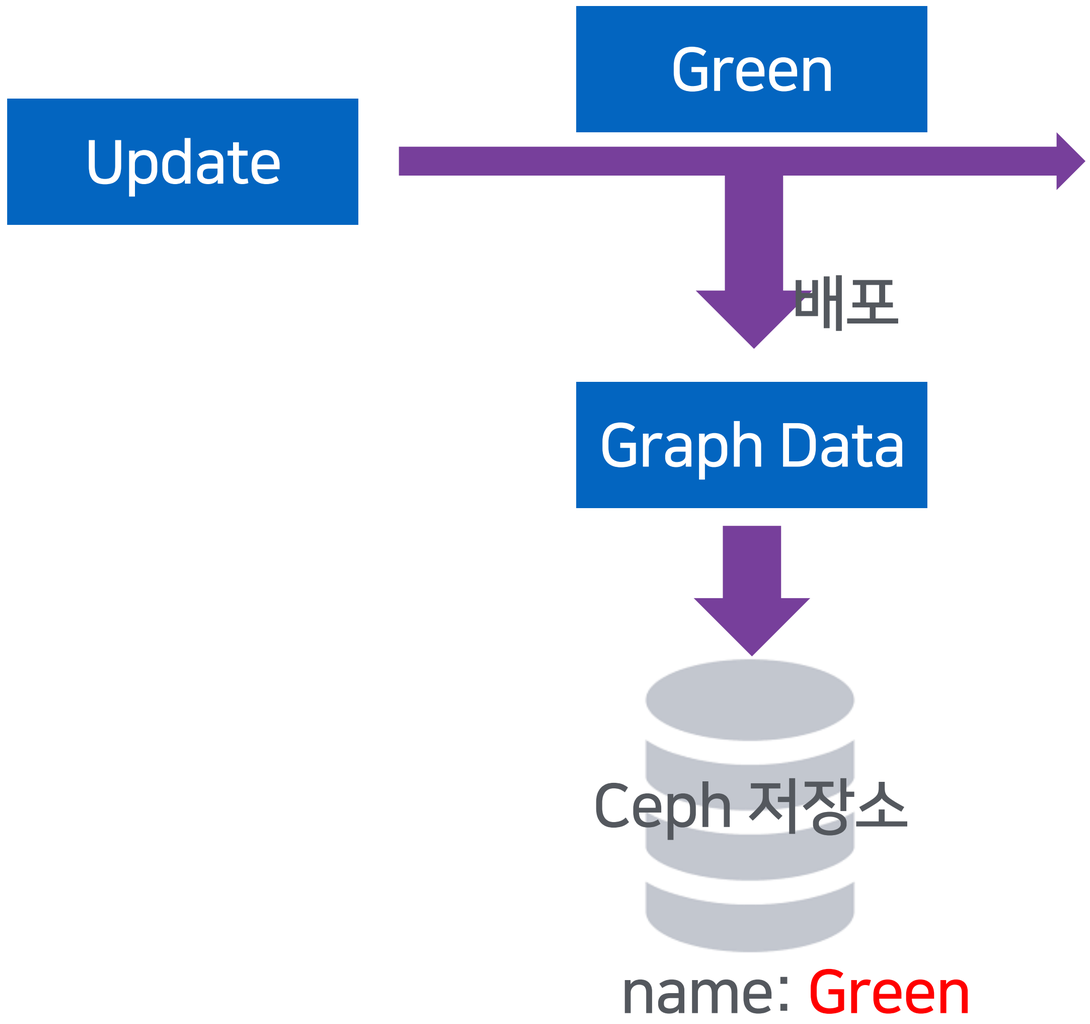


Issue) pvc 저장소 증식

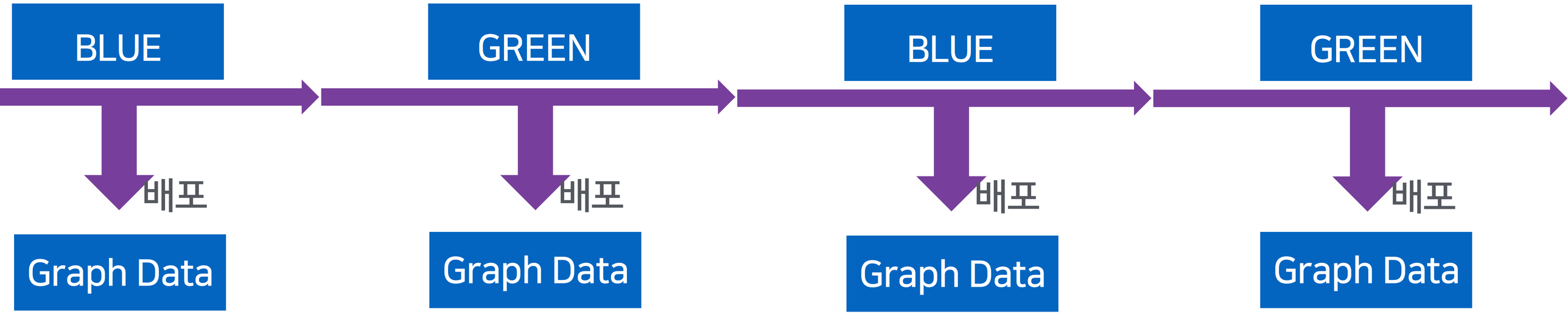


Solution) 2개의 시나리오

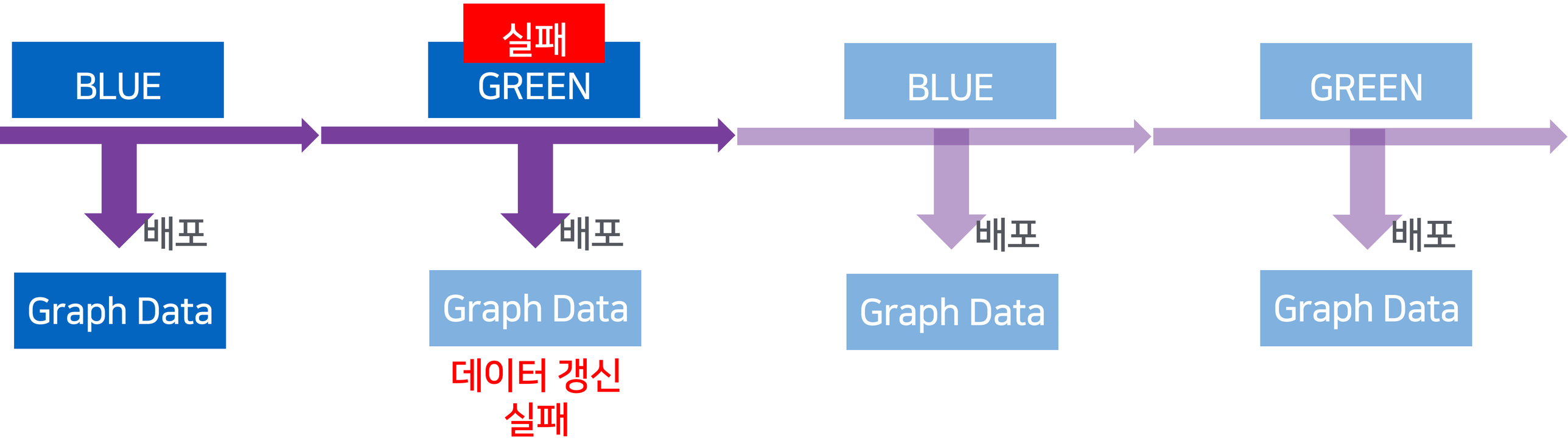




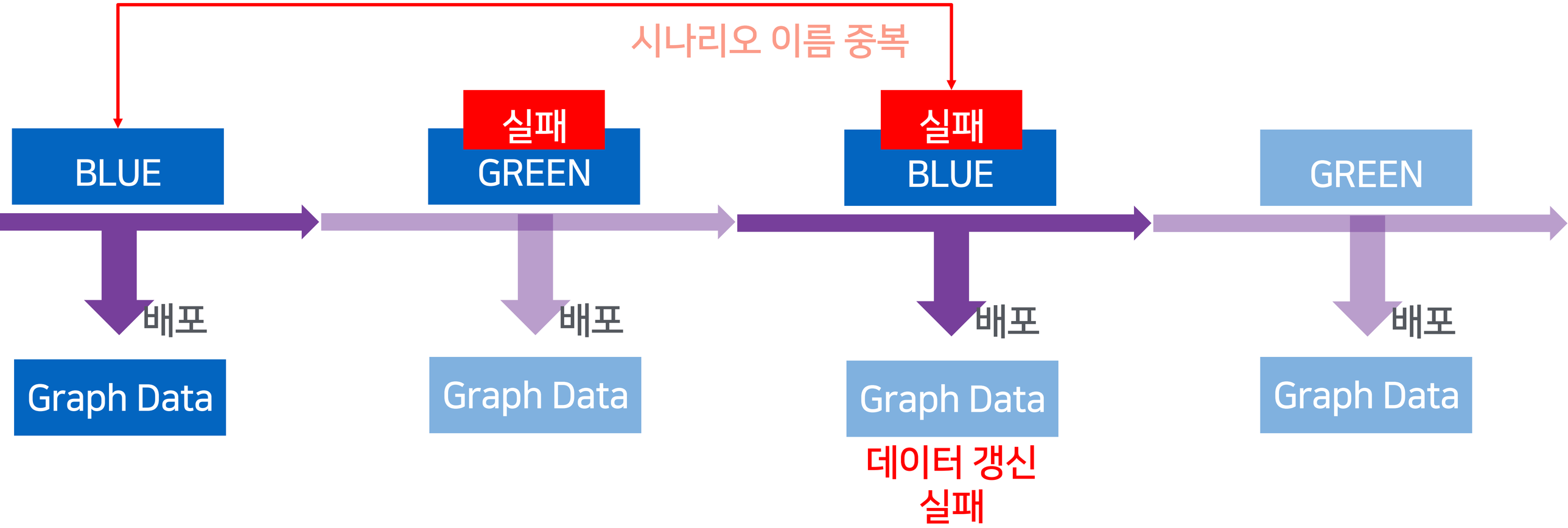
Issue) 시나리오 중복



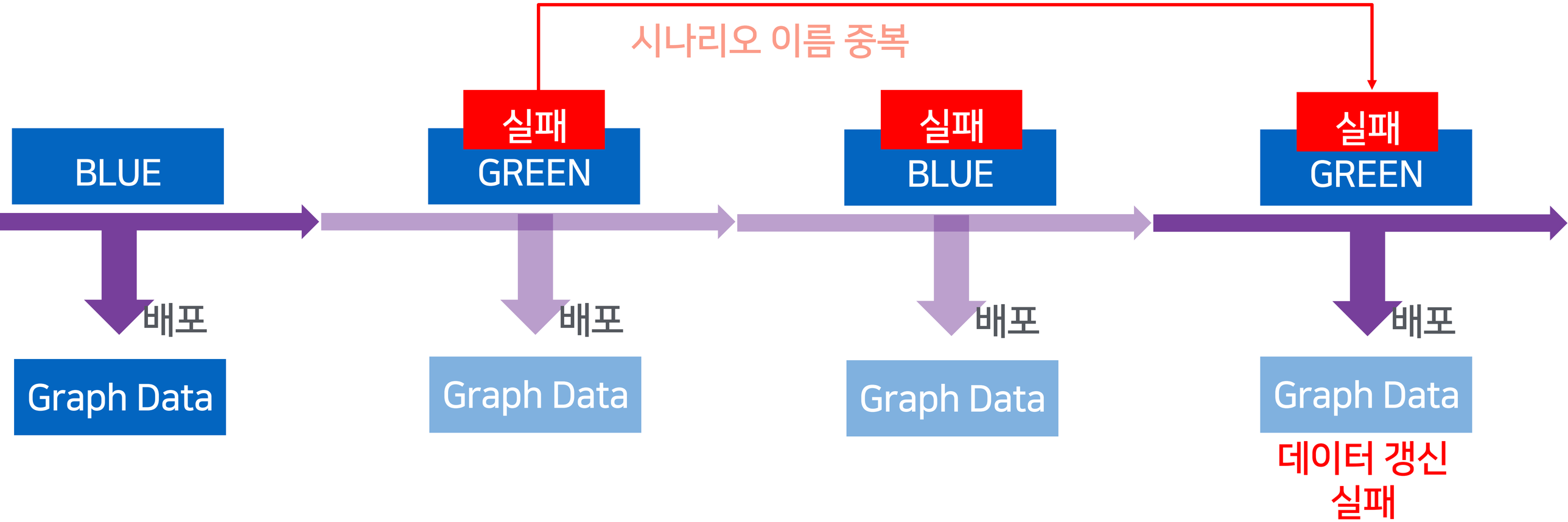
Issue) 시나리오 중복



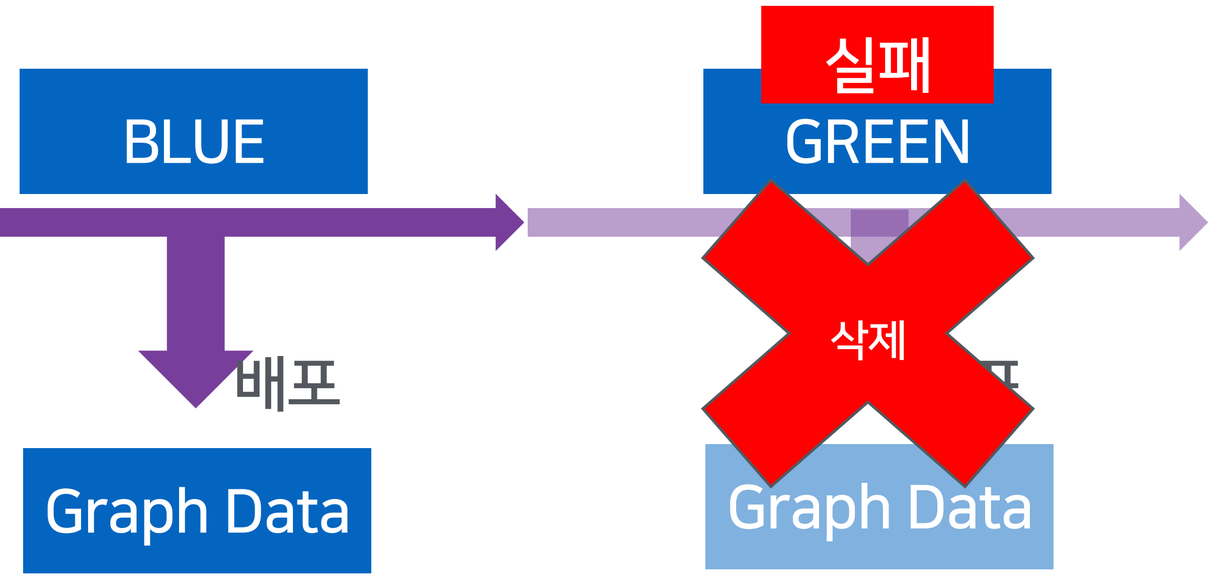
Issue) 시나리오 중복



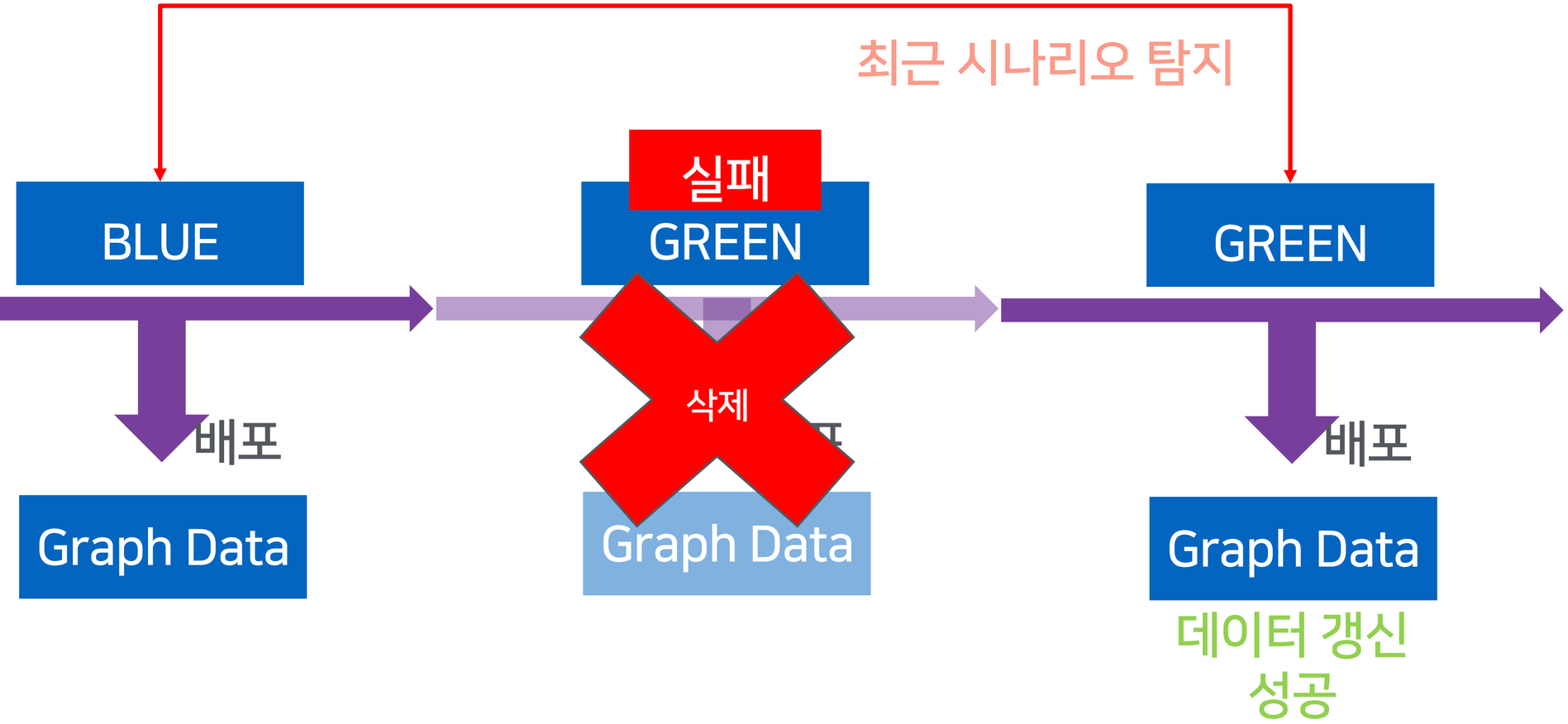
Issue) 시나리오 중복



Solution) Failed Installation 삭제

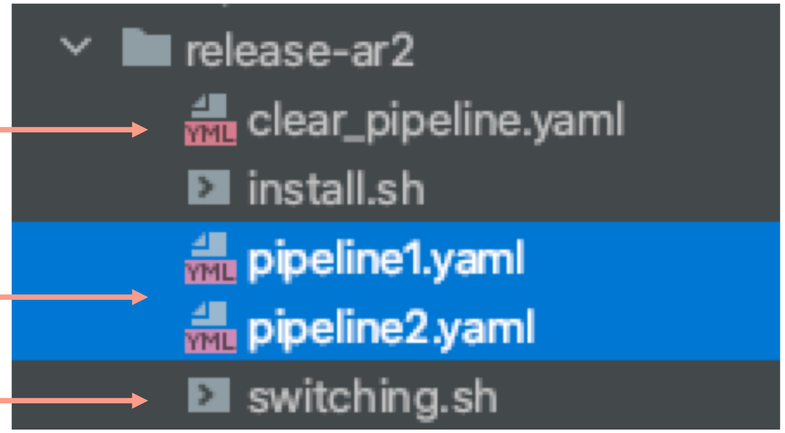


Solution) Failed Installation 삭제



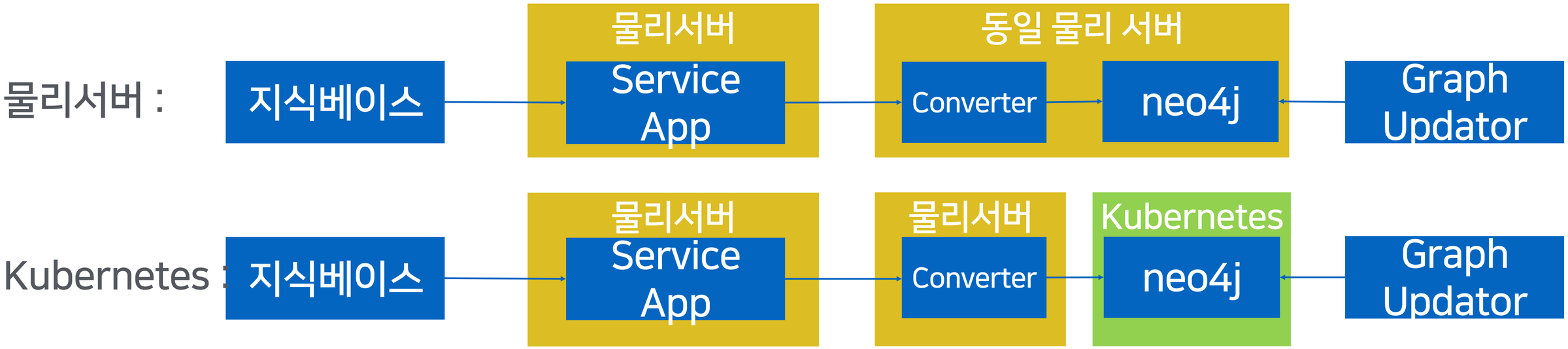
Solution) 시나리오 구성

- 하나의 삭제 시나리오
- 두개의 배포 시나리오
- 운영용 스크립트
- 실제 운영시에는 스크립트로 배포를 제어



5. 관계성 플랫폼과의 연결

5.1 관계성 플랫폼 구조



Service App : 관계성 데이터와 속성 데이터를 합쳐서 돌려줌
Converter : Neo4j와 직접 통신하며, 데이터의 Parsing, neo4j용 cypher 생성
Graph Updator : Neo4j가 사용할 DB를 데이터를 batch update

Issue) JDBC Template Connection

DEVIEW 2021

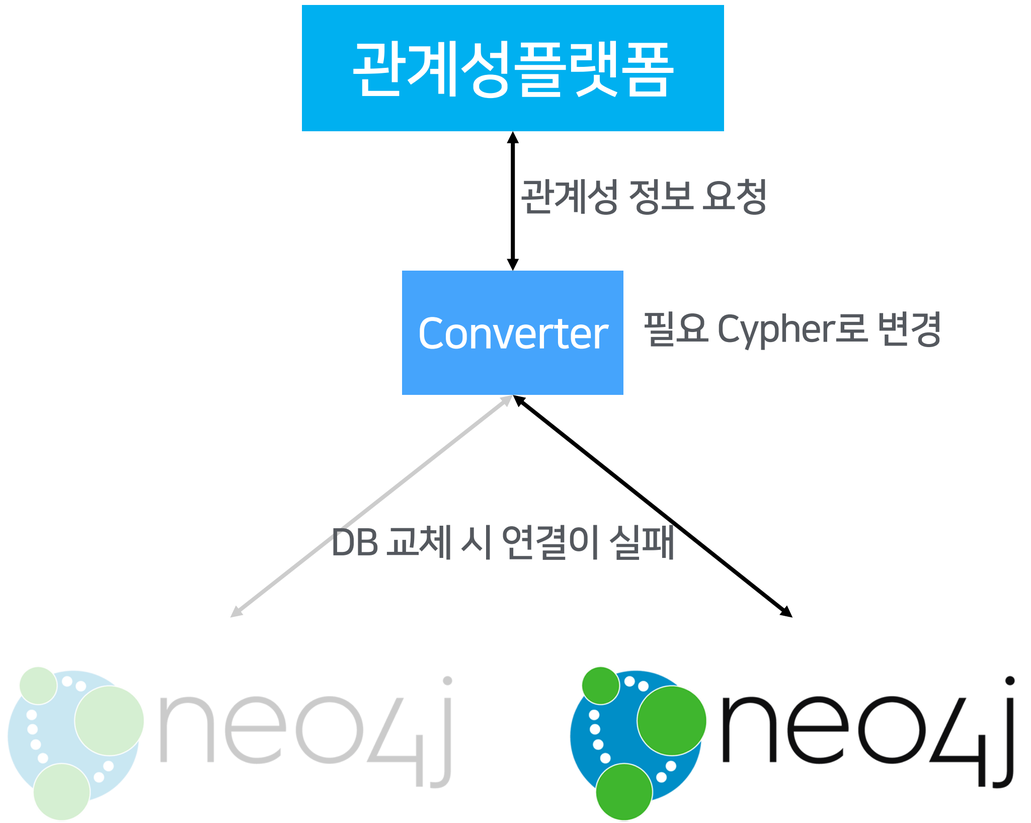
Switching 시점의 Stress Test 결과

```
1620098796 12232001  0.0110  4010  0.0001  4010  0.0106  4010  0.0002  4010     0     0     0     0     0    39  133/16000  0.0526  40   6702  25.63 E0(0)
1620098797 12236001  0.0112  3996  0.0001  3996  0.0108  3996  0.0002  3996     0     0     0     0     0    43  190/16000  0.0427  39   6511  24.81 E0(0)
1620098798 12240001  0.0110  4002  0.0001  4002  0.0107  4002  0.0002  4002     0     0     0     0     1    41   94/16000  0.0451  40   6834  26.09 E104(1)
1620098799 12244001  0.0164  3347  0.0002  3347  0.0160  3347  0.0003  3347     0     0     0     0     0   693 2041/16000  0.3446  33   6937  22.14 E0(0)
1620098800 12248001  0.1649  2428  0.0001  2428  0.1639  2428  0.0009  2428     0     0     0     0     0  2265 5311/16000  1.2156  24   6275  14.53 E0(0)
1620098801 12252001  0.5542  3595  0.0002  3595  0.5529  3595  0.0012  3544     0     0     0    51     0  2670 7989/16000  2.0000  35   7028  23.75 E110(51)
1620098802 12256001  0.6783  4430  0.0002  4430  0.6774  4430  0.0008  4399     0     0     0    31     0  2241 7851/16000  1.9817  44   7236  30.36 E110(31)
1620098803 12260001  0.5067  5084  0.0001  5084  0.5060  5084  0.0005  5070     0     0     0    14     0  1157 7668/16000  1.8040  50   6909  33.41 E110(14)
1620098804 12264001  0.2222  4806  0.0001  4806  0.2216  4806  0.0005  4802     0     0     0     4     0   351 7826/16000  1.5790  48   6737  30.86 E110(4)
1620098805 12268001  0.0826  4129  0.0001  4129  0.0820  4129  0.0004  4127     0     0     0     2     1   222 5884/16000  1.2327  41   6865  27.02 E110(2)
1620098806 12272001  0.0540  4145  0.0002  4145  0.0535  4145  0.0004  4145     0     0     0     0     0    76  416/16000  0.8642  41   6990  27.63 E0(0)
1620098807 12276001  0.0147  4029  0.0001  4029  0.0142  4029  0.0003  4029     0     0     0     0     1    47   93/16000  0.0825  40   7229  27.78 E104(1)
1620098808 12280001  0.0135  3995  0.0002  3995  0.0130  3995  0.0003  3995     0     0     0     0     0    52  436/16000  0.0636  39   6826  26.01 E0(0)
1620098809 12284001  0.0170  3998  0.0001  3998  0.0166  3998  0.0003  3998     0     0     0     0     1    54  203/16000  0.2777  39   7104  27.09 E104(1)
1620098810 12288001  0.0128  4010  0.0001  4010  0.0123  4010  0.0003  4010     0     0     0     0     0    44   94/16000  0.0544  40   6588  25.20 E0(0)
```

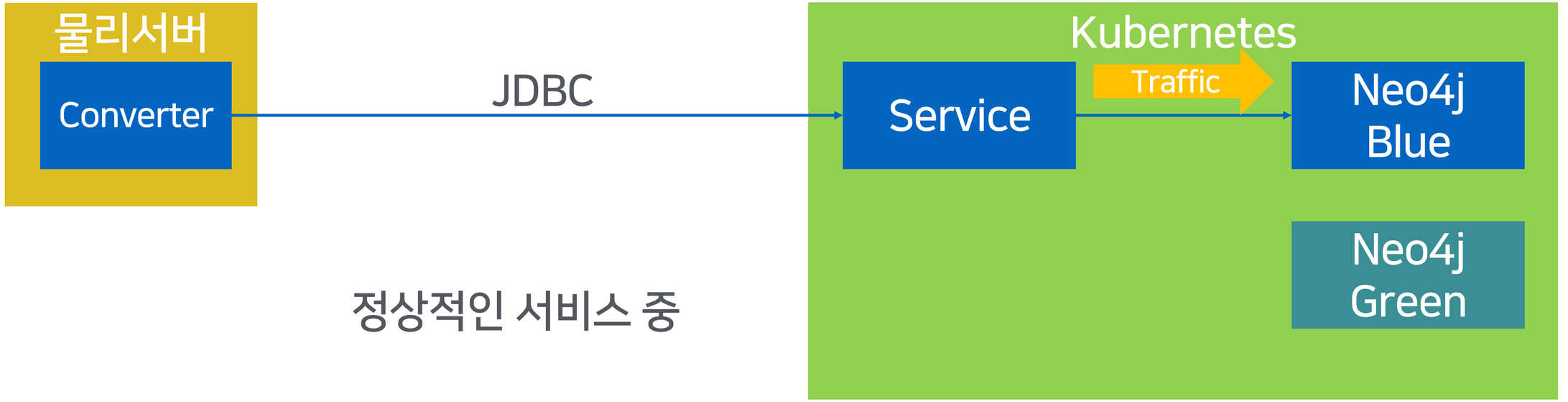
Blue -> Green 전환 시 **대량의 오류 발생**

Issue) Converter와 neo4j 연결 실패

- Neo4j와 관계성 플랫폼 사이의 번역가
- DB 스위칭 시 연결 실패



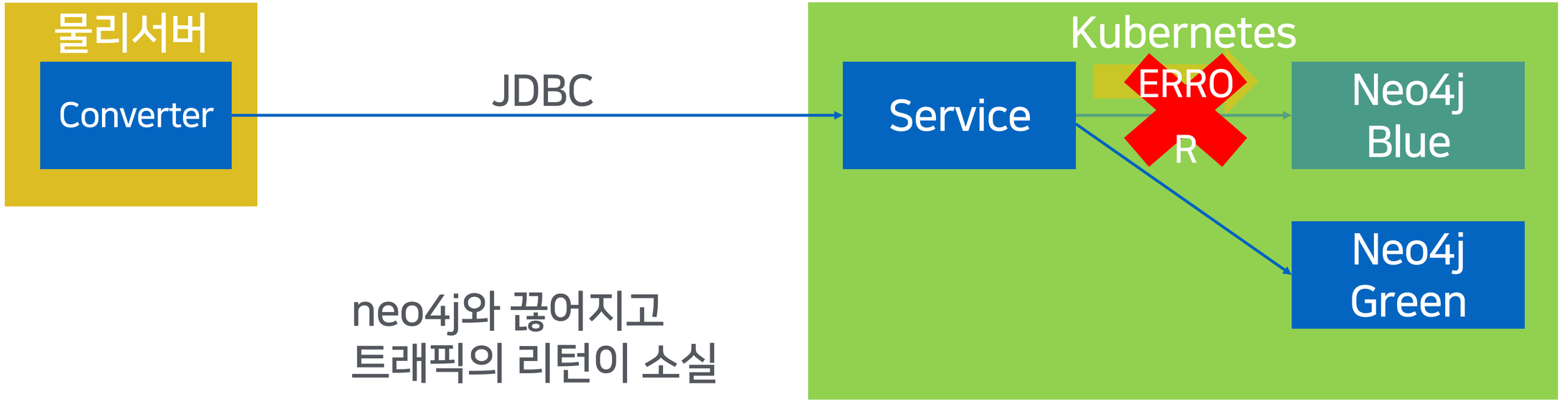
Issue) JDBC Template Connection



정상적인 서비스 중

새 Data의 Neoj4 배포 중

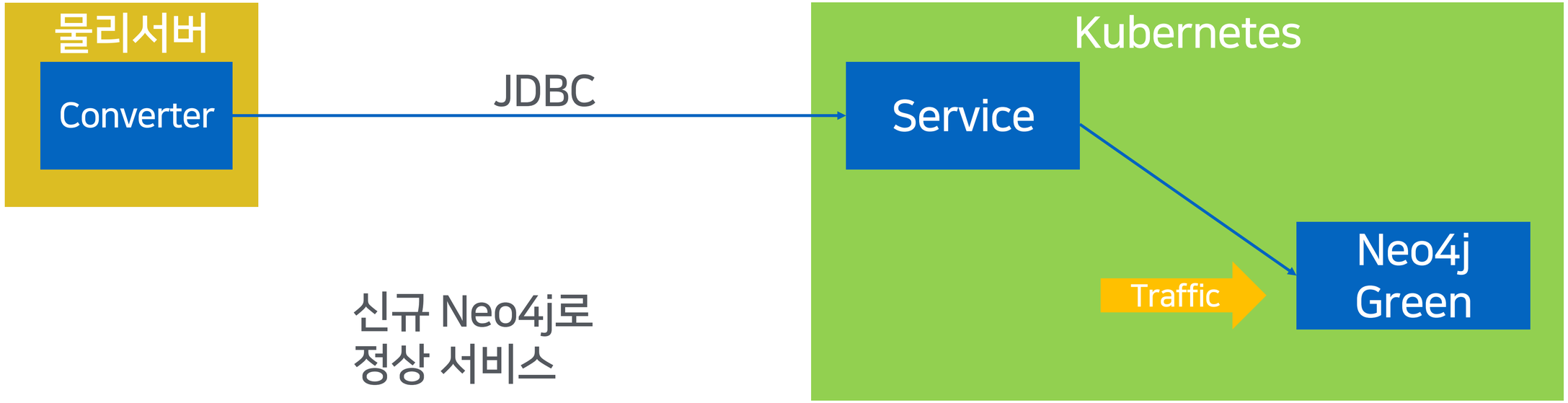
Issue) JDBC Template Connection



neo4j와 끊어지고
트래픽의 리턴이 소실

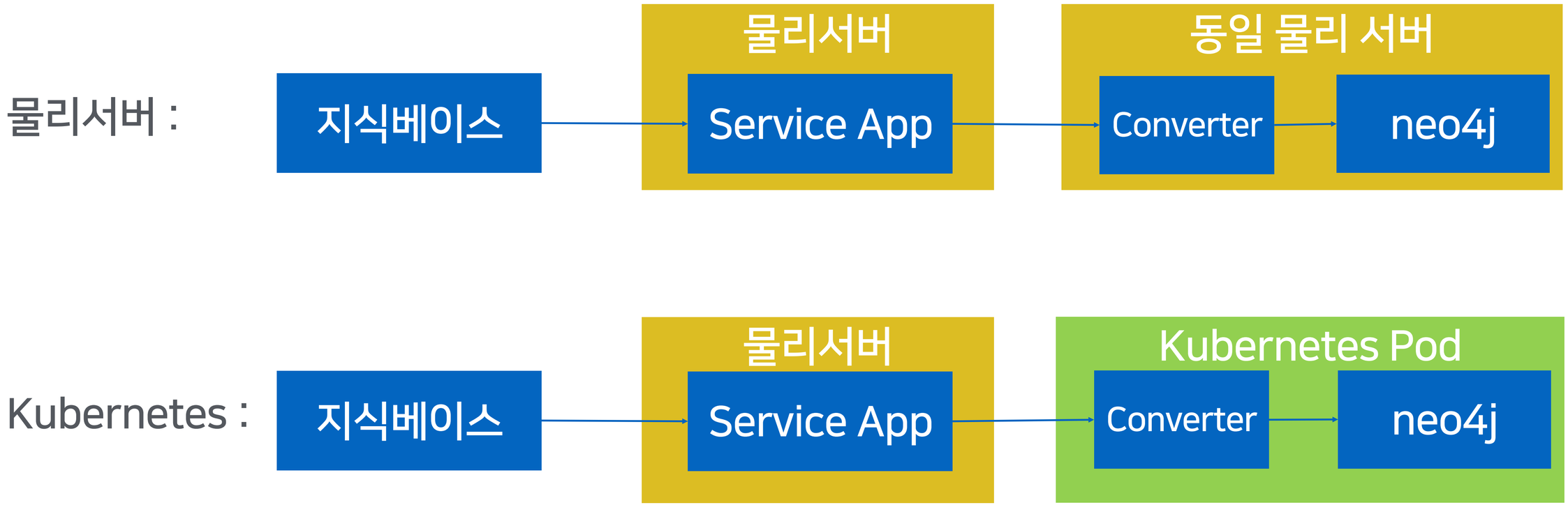
새 Data의 Neoj4 배포

Issue) JDBC Template Connection



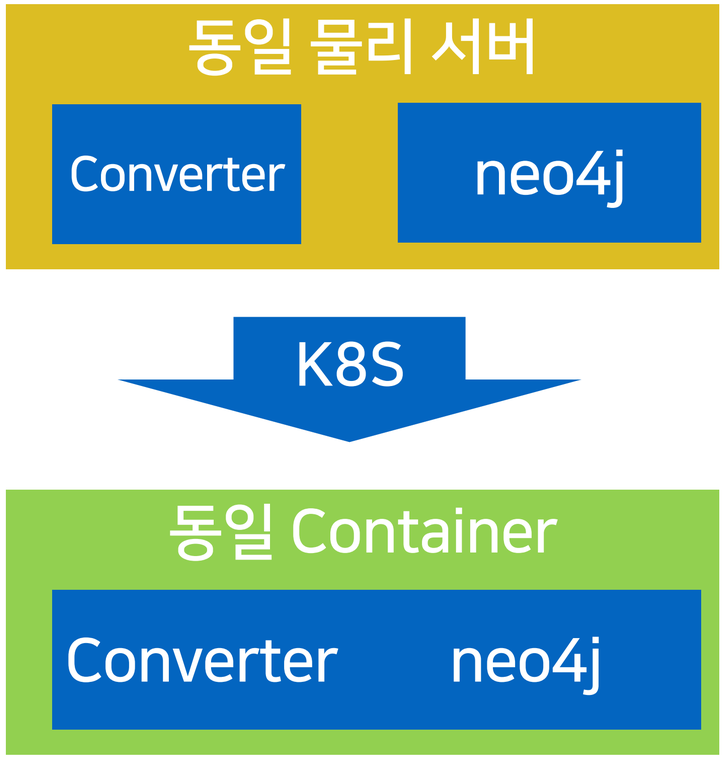
신규 Neo4j로
정상 서비스

Issue) JDBC Template Connection



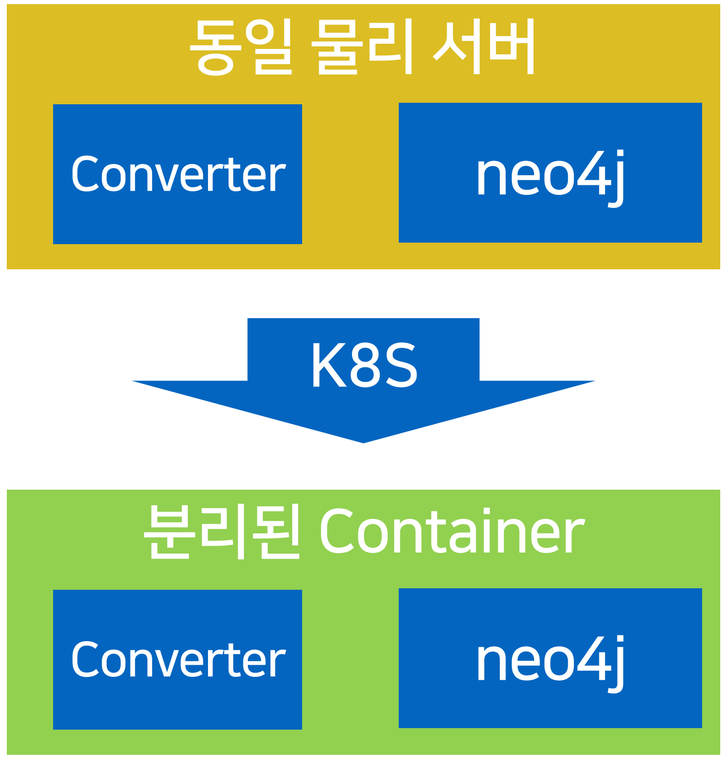
Solution) Converter도 Kubernetes로

1



Entrypoint 스크립트를 수정

2



Read-replica helm chart를 수정

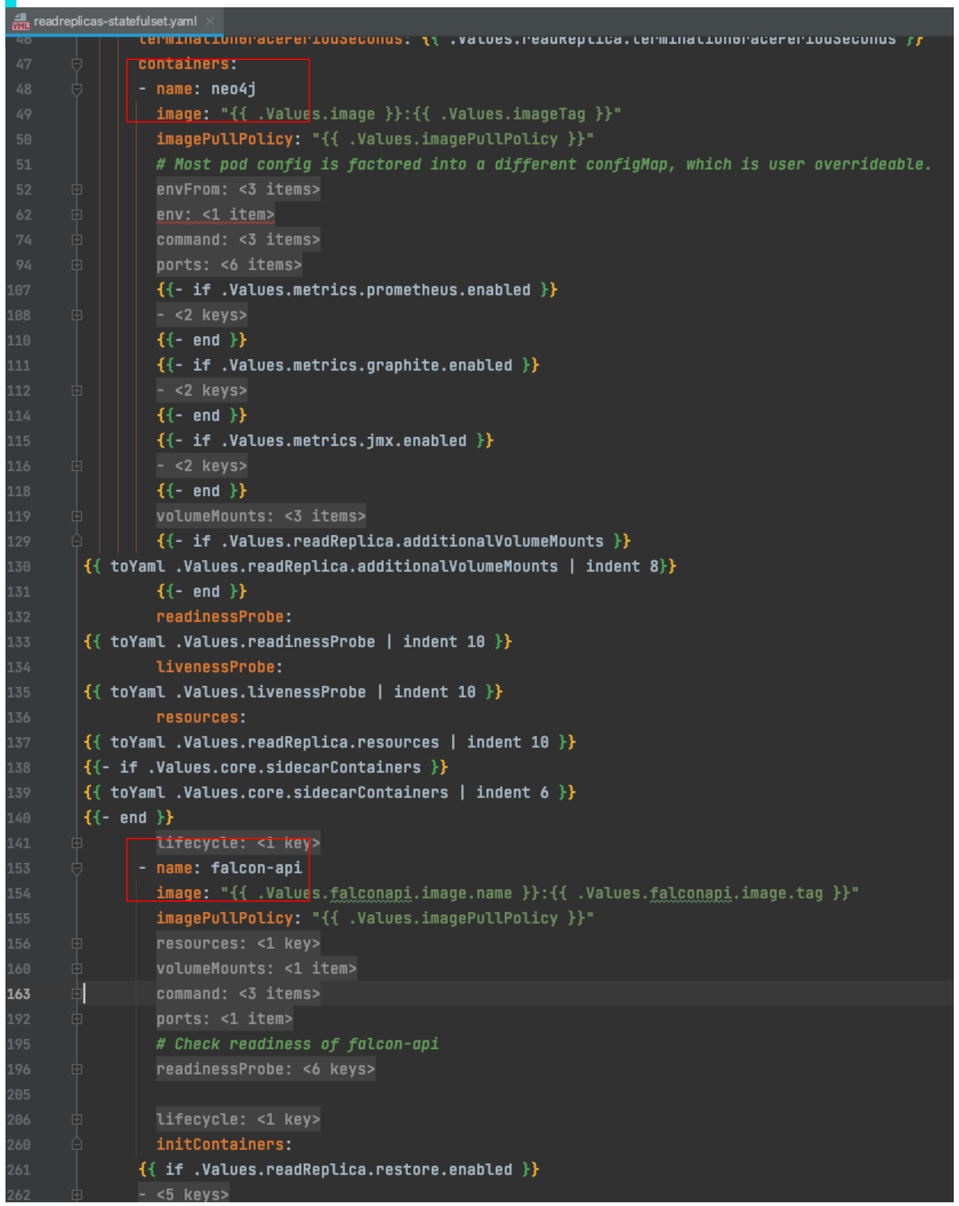
Test1) Container를 하나만

- neo4j의 docker의 entrypoint를 수정
- read-replica에서 neo4j외에 관계성 플랫폼도 실행하도록 수정된 entrypoint를 보도록 변경

```
#!/bin/bash

# Start the first process
/docker-entrypoint.sh "neo4j" &

while :
do
  STATUS_CODE=$(curl -o /dev/null -w "%{http_code}" http://localhost:7474/db/data/)
  SUCCESS_CNT=$[SUCCESS_CNT+1]
  if [ $STATUS_CODE -eq 200 ]; then
    echo "Got $STATUS_CODE : Read-replica Complete"
    break
  else
    echo "Got $STATUS_CODE : Wait for read-replica"
  fi
  sleep 5
done

exec /falcon-entrypoint.sh "start" "alpha5" &> /home1/irteam/log
```

Test2) Container를 나눠서

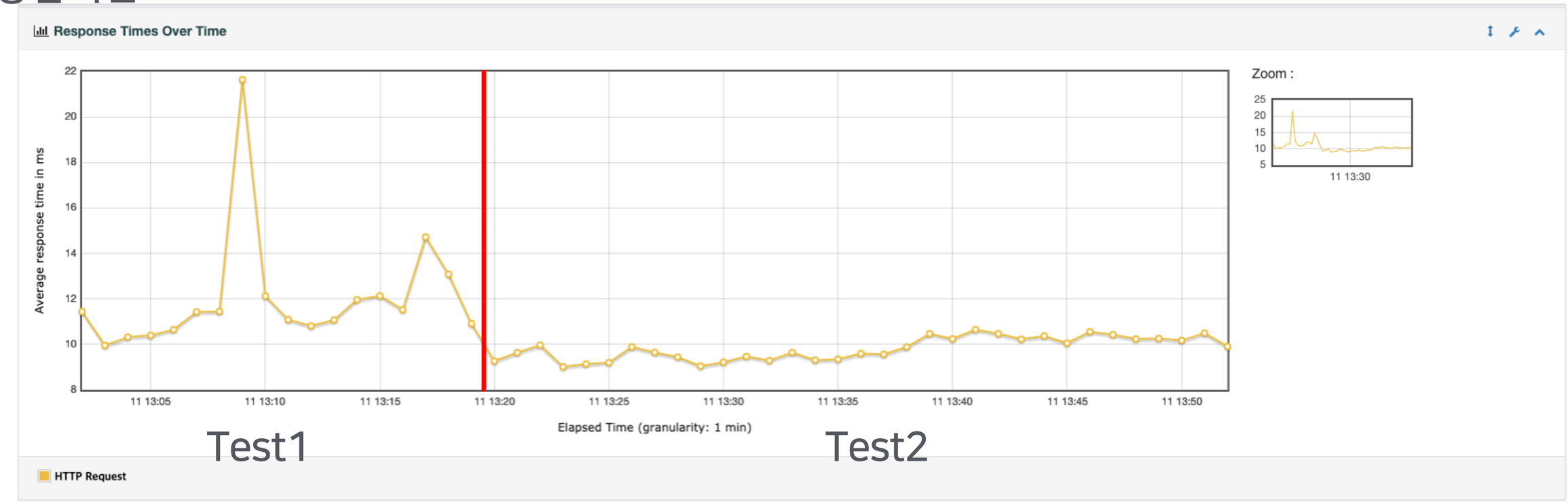
- read-replica 내에 두 개의 container를 형성



Test) 결과

DEVIEW 2021

응답시간



Test1

Test2

Test) Container 안에서는...

DEVIEW 2021

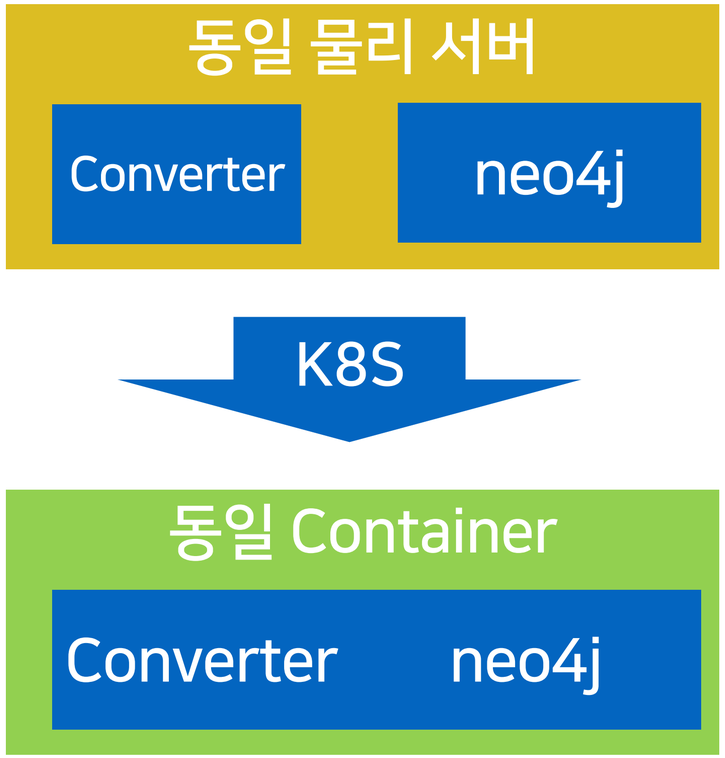
```
top - 12:07:59 up 581 days, 22:19,  0 users,  load average: 23.32, 24.35, 21.67
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu(s):  33.0/5.3    38[
GiB Mem :  251.464 total,   19.680 free,  154.889 used,   76.895 buff/cache
GiB Swap:    0.000 total,    0.000 free,    0.000 used.   82.568 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   20 neo4j     20   0 13.837g 3.378g  29204 S 143.0  1.3  58:49.99 java
    6 root      20   0 30.383g 2.626g  19920 S  53.0  1.0  22:55.53 java
  983 root      20   0   36640   3248   2728 R   0.0  0.0   0:01.51 top
    1 root      20   0    1132      8      0 S   0.0  0.0   0:00.08 init
  970 root      20   0   18508   3432   2956 S   0.0  0.0   0:00.02 bash
```

```
top - 12:40:53 up 581 days, 22:52,  0 users,  load average: 29.59, 26.83, 23.19
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu(s):  42.1/5.7    48[
GiB Mem :  251.464 total,   19.639 free,  157.115 used,   74.711 buff/cache
GiB Swap:    0.000 total,    0.000 free,    0.000 used.   80.361 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   20 neo4j     20   0 13.862g 3.855g  29204 S 200.0  1.5 108:31.20 java
    6 root      20   0 30.414g 2.646g  19920 S  64.0  1.1  41:22.87 java
  983 root      20   0   36640   3200   2728 R   0.0  0.0   0:03.72 top
    1 root      20   0    1132      8      0 S   0.0  0.0   0:00.12 init
  970 root      20   0   18508   3432   2956 S   0.0  0.0   0:00.02 bash
```

```
top - 12:08:21 up 581 days, 22:19,  0 users,  load average: 24.71, 24.52, 21.79
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu(s):  34.5/5.3    40[
GiB Mem :  251.464 total,   19.665 free,  154.875 used,   76.924 buff/cache
GiB Swap:    0.000 total,    0.000 free,    0.000 used.   82.582 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
   20 neo4j     20   0 13.845g 3.380g  29204 S 179.0  1.3  59:25.48 java
    6 root      20   0 30.383g 2.627g  19920 S  61.0  1.0  23:08.93 java
  983 root      20   0   36640   3248   2728 R   0.0  0.0   0:01.54 top
    1 root      20   0    1132      8      0 S   0.0  0.0   0:00.08 init
  970 root      20   0   18508   3432   2956 S   0.0  0.0   0:00.02 bash
```

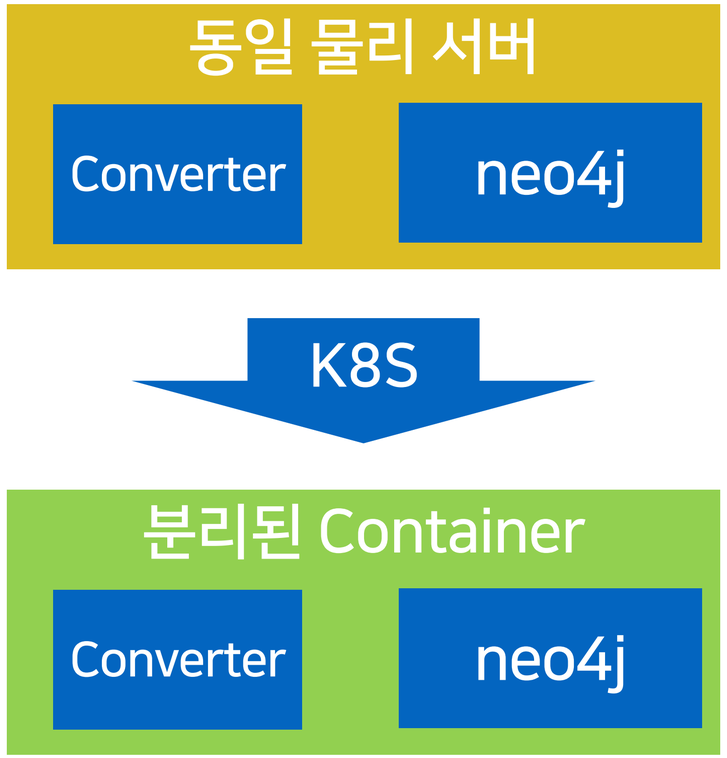
Solution) Converter도 Kubernetes로

1



Entrypoint 스크립트를 수정

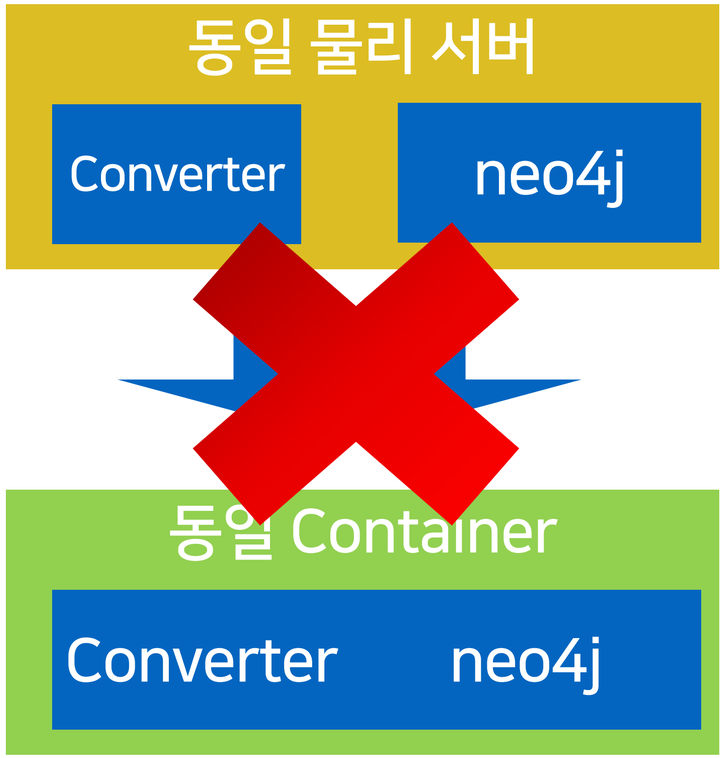
2



Read-replica helm chart를 수정

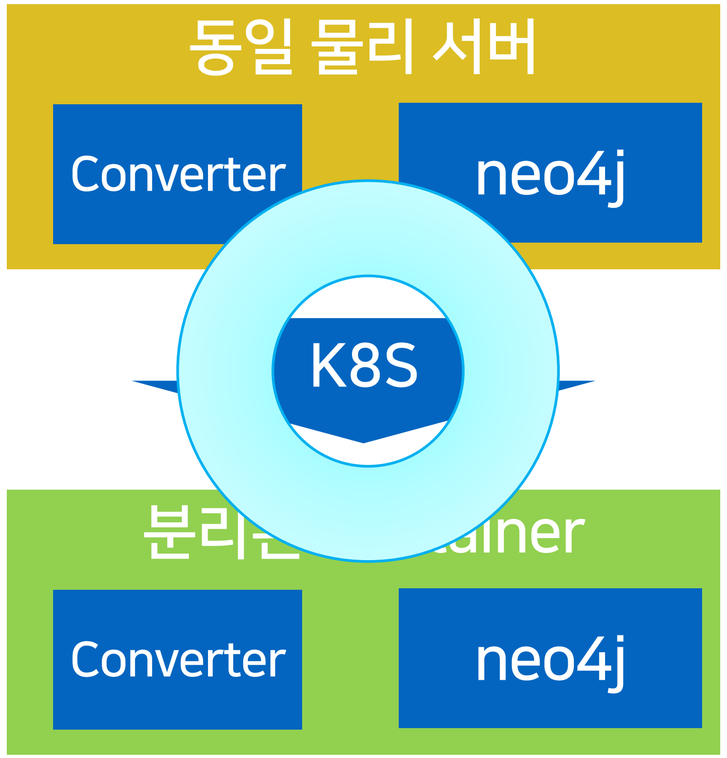
Solution) Converter도 Kubernetes로

1



Entrypoint 스크립트를 수정

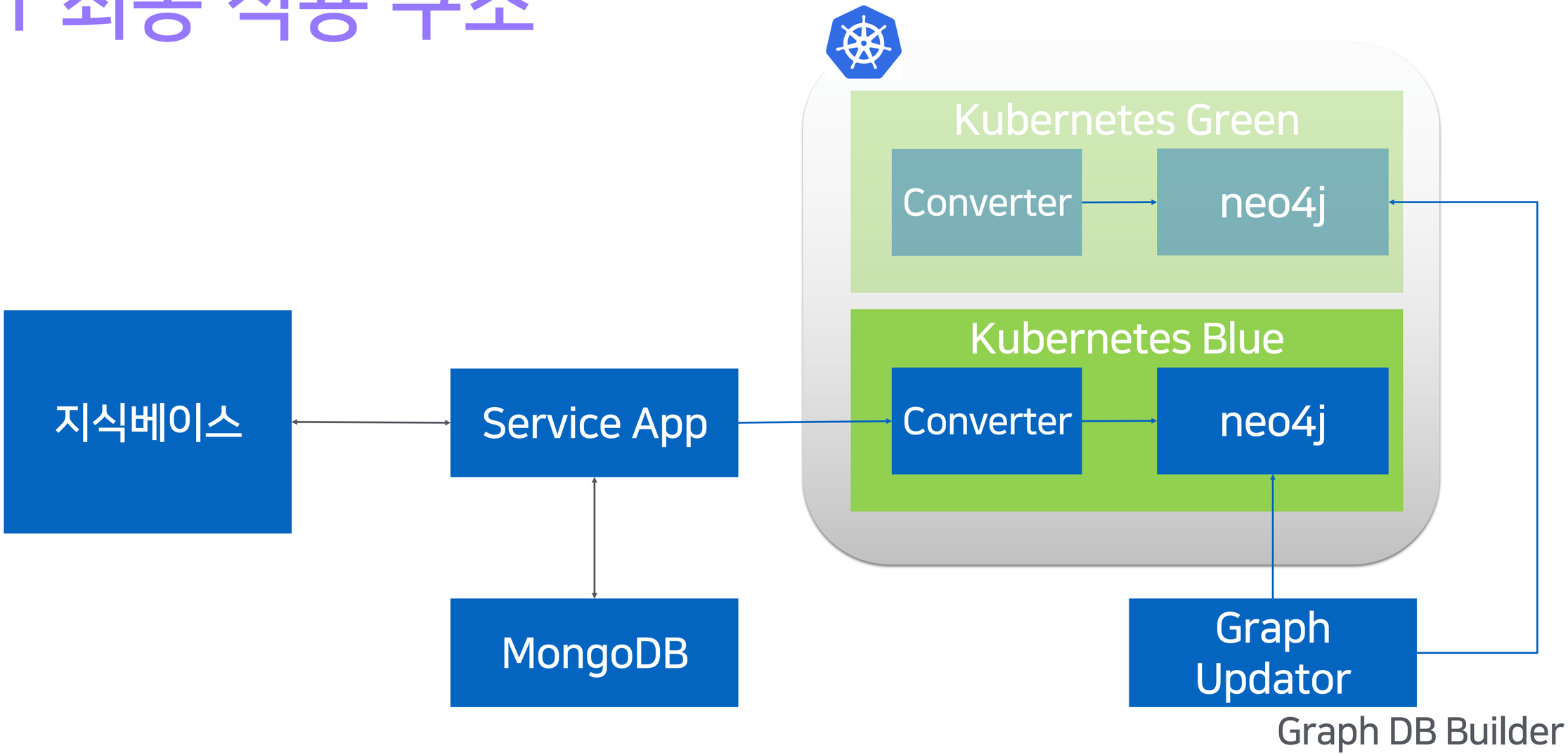
2



Read-replica helm chart를 수정

7. 최종결과

7.1 최종 적용 구조


Kubernetes Green
Converter
neo4j
Kubernetes Blue
Converter
neo4j
지식베이스
Service App
MongoDB
Graph
Updator
Graph DB Builder

7.2 최종 결과(응답시간)

DEVIEW 2021

Kubernetes

Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	5400074	1139	0.02%	11.16	0	4292	9.00	15.00	21.00	41.00	1800.61	9652.58	1819.61
HTTP Request	5400074	1139	0.02%	11.16	0	4292	9.00	15.00	21.00	41.00	1800.61	9652.58	1819.61

물리서버 1대

Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	1346505	42908	3.19%	12.31	0	5041	9.00	18.00	20.00	46.00	448.87	3255.84	460.10
HTTP Request	1346505	42908	3.19%	12.31	0	5041	9.00	18.00	20.00	46.00	448.87	3255.84	460.10

7.2 최종 결과(응답시간)

DEVIEW 2021

Kubernetes 환경 – 보통 qps

Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	5400074	1139	0.02%	11.16	0	4292	9.00	15.00	21.00	41.00	1800.61	9652.58	1819.61
HTTP Request	5400074	1139	0.02%	11.16	0	4292	9.00	15.00	21.00	41.00	1800.61	9652.58	1819.61

Kubernetes 환경 – 과부하 qps

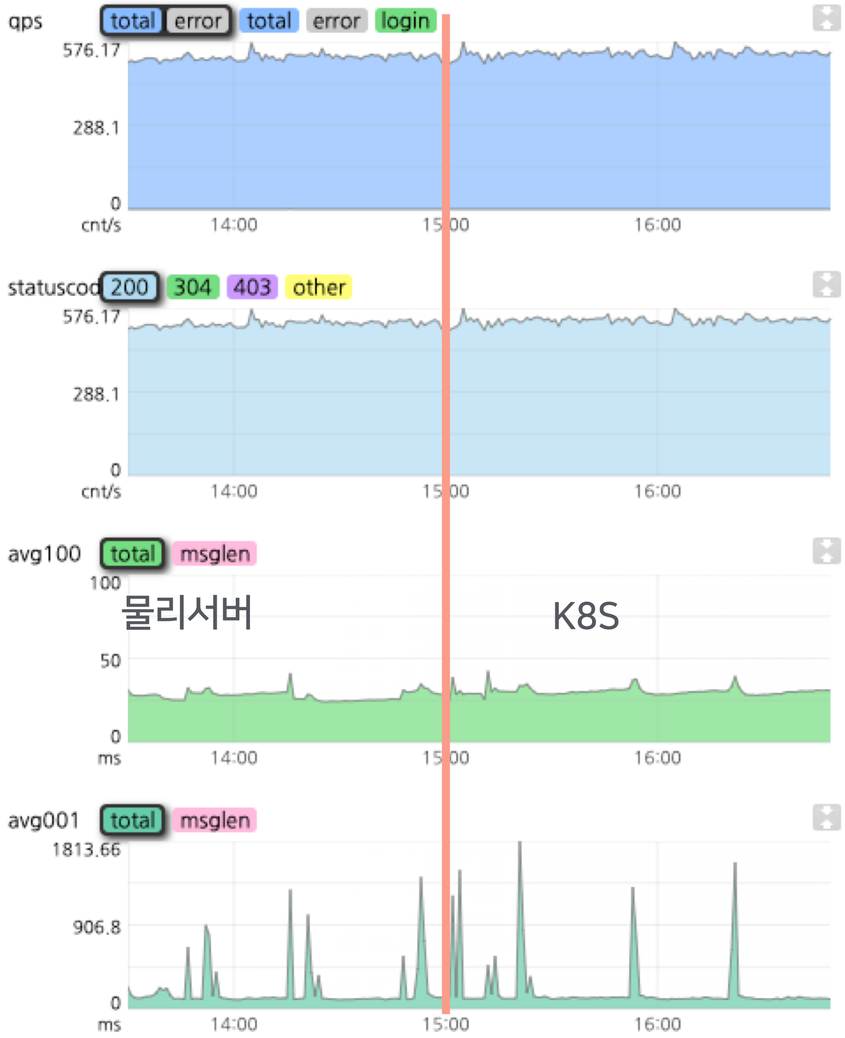
Statistics

평균응답시간

QPS

Requests	Executions			Response Times (ms)							Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	Median	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	17951102	2190	0.01%	14.75	0	7777	10.00	21.00	30.00	53.00	5985.73	37339.62	6268.26
HTTP Request	17951102	2190	0.01%	14.75	0	7777	10.00	21.00	30.00	53.00	5985.73	37339.62	6268.26

7.3 관계성 플랫폼 서비스 결과



	물리서버	Kubernetes
응답시간	24ms	30ms
qps	550	550

Thank You