

DB는 있지만 DBA는 없는 세상의 노하우

이준영 / NHN Cloud

김가림 / NHN Cloud

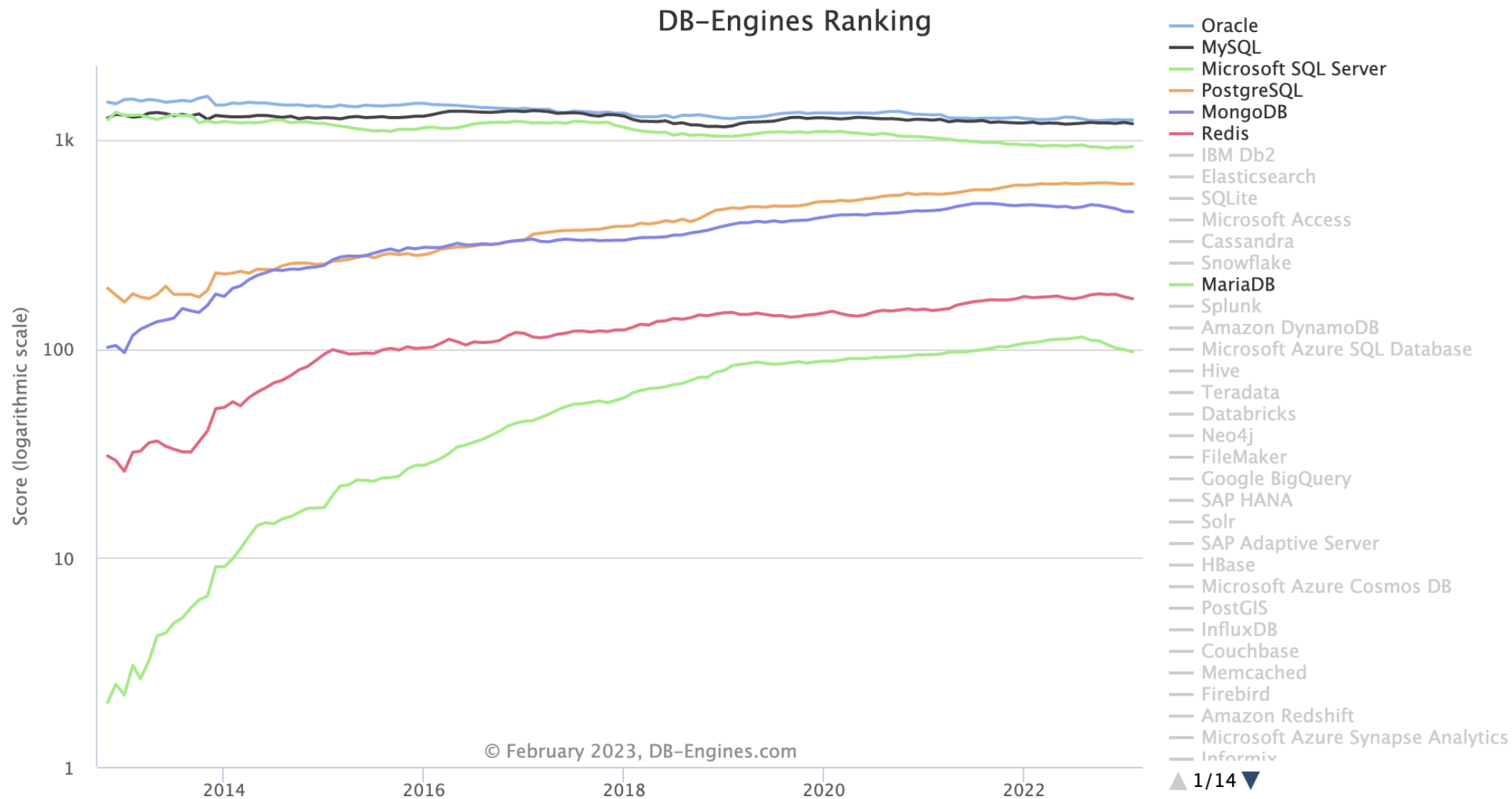
DBA?

DBA가 하는 일

- 데이터베이스 서버 소프트웨어 설치와 구성 및 업그레이드
- 소프트웨어 기능 및 관련 제품을 평가
- 백업 및 복구 정책과 절차를 수립하고 유지
- 데이터베이스 보안(사용자 및 그룹을 관리)
- 데이터베이스와 튜닝 및 성능 모니터링
- 표준 관리 문서를 작성
- 용량 증가 및 변화를 관리
- 트러블 슈팅 및 데이터베이스 복구
- 365일 24시간 연중무휴



DBA 업무의 확장



설치의 자동화

자동화 도구 도입

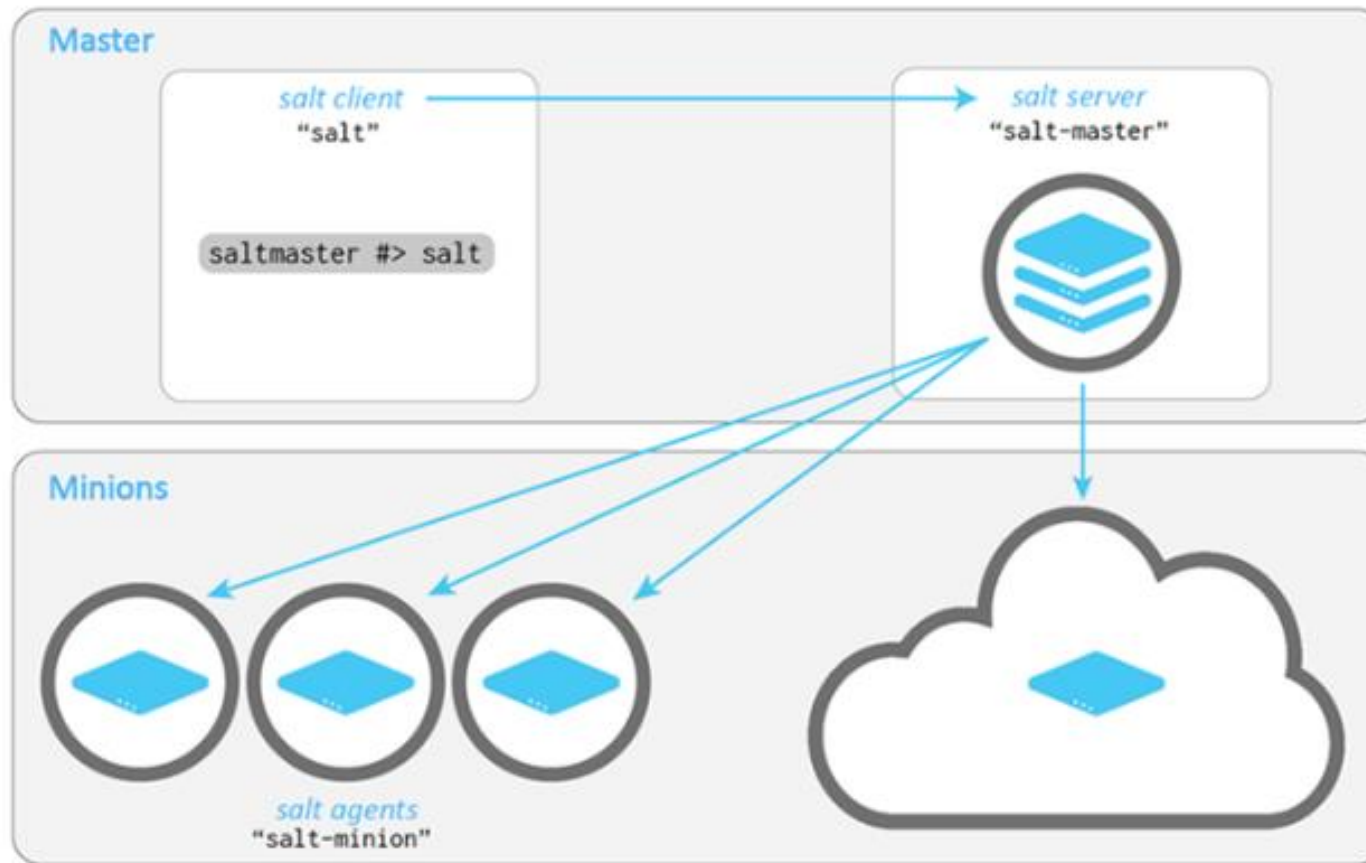
- 중앙 관리 도구의 필요성
 - 자동 설치
 - 모니터링 지표 수집
 - 버전 업그레이드
 - 일괄 작업 수행

SALTSTACK

- 오픈소스 프로젝트
- 대규모 인프라 관리 자동화 소프트웨어
- 분산 메시징 시스템 사용
 - SSH 소켓을 사용하지 않고 통신
 - 수천 개의 노드를 지원

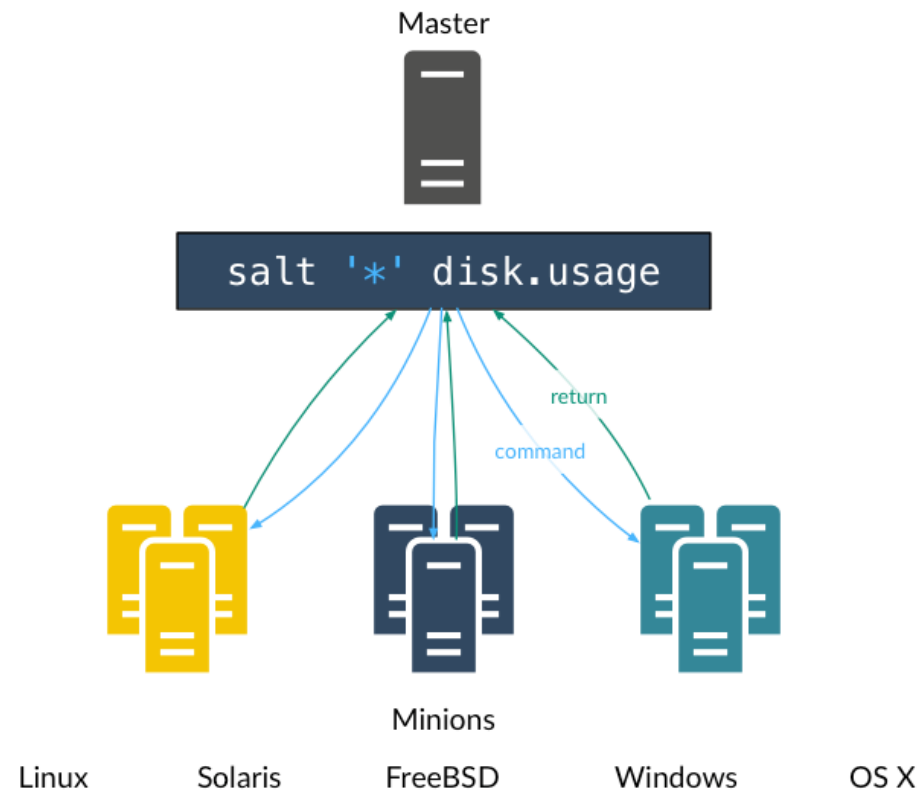
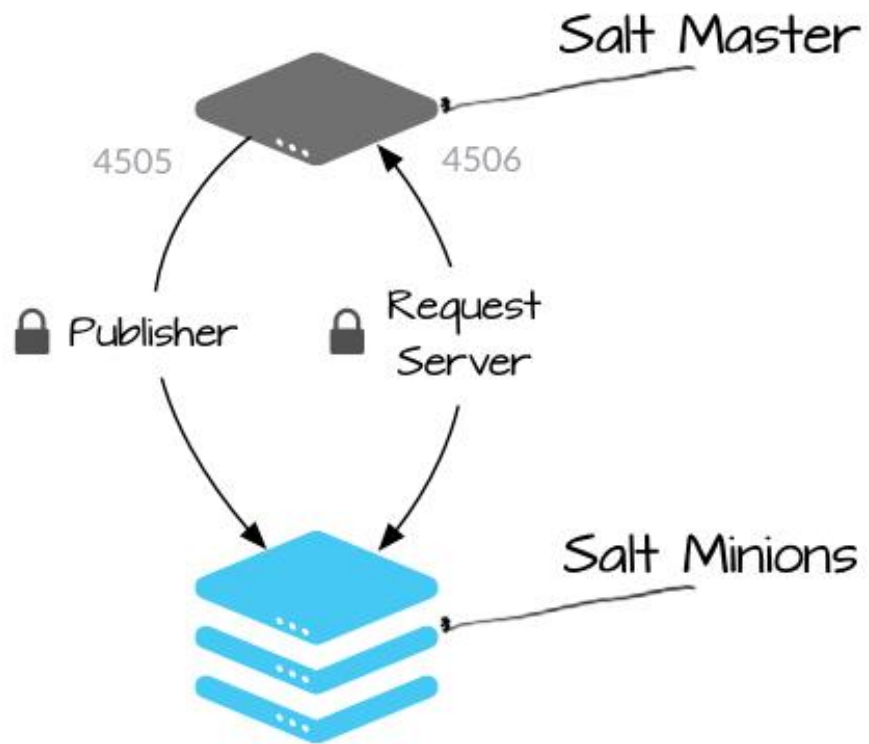


SALTSTACK



- 서버-에이전트 모델

SALTSTACK



- publish-subscribe 패턴
- ZeroMQ를 이용한 비동기 통신

SALTSTACK

```

ID:
  module.function:
    - name: name
    - argument: value
    - argument:
      - value1
      - value2
  
```



ID

String that describes this state.
Must be unique.

module.function

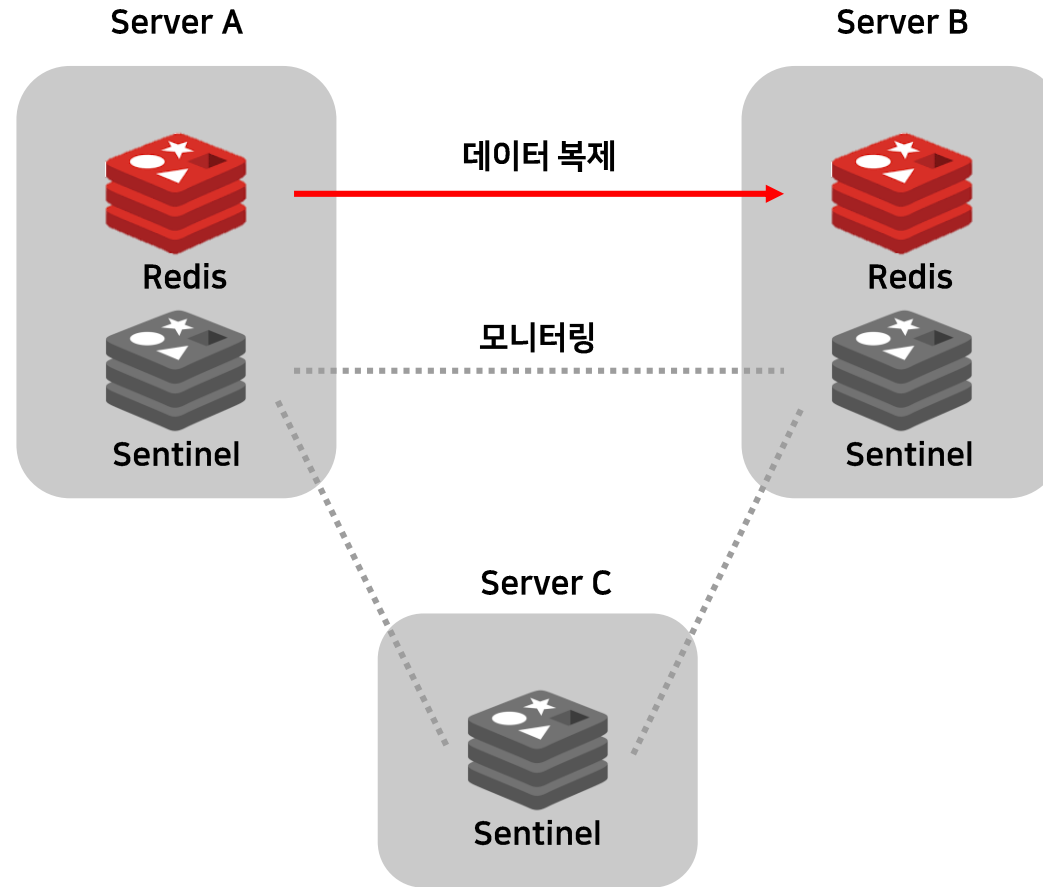
The state module and function that you want to call (salt.state.*).

Arguments

Every function takes 'name' as the first argument.
Other arguments are listed under the function.

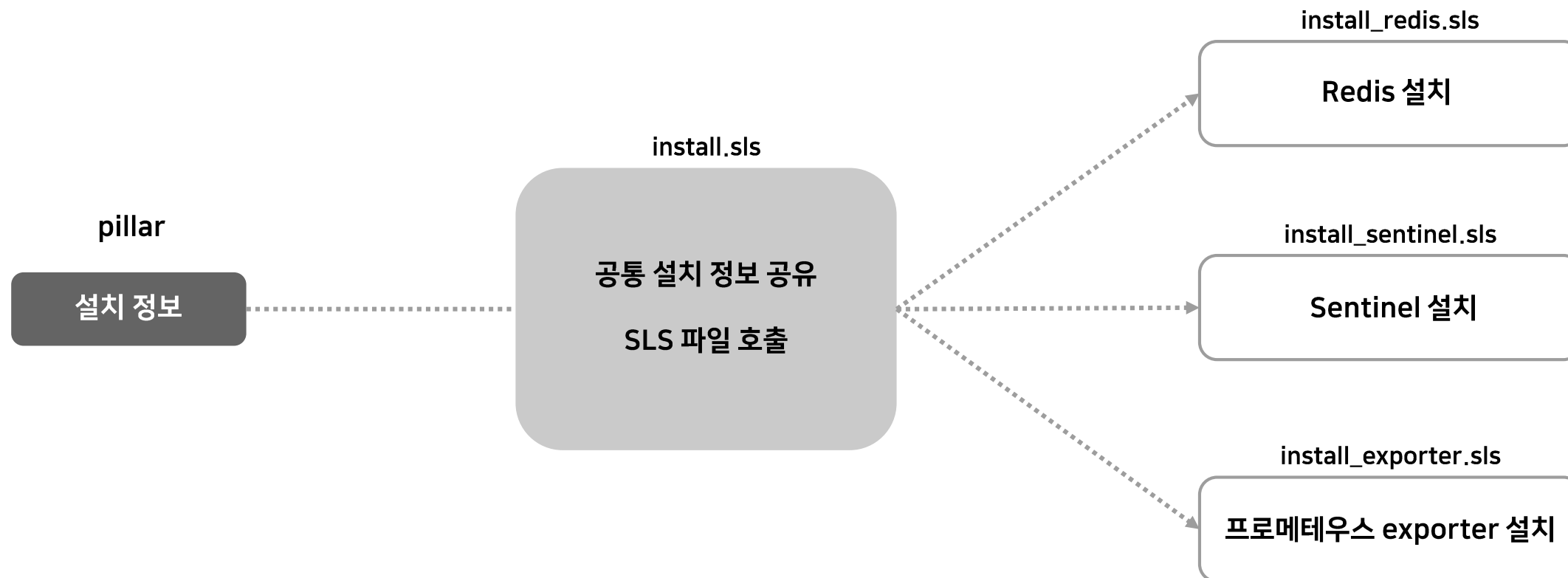
Salt State(Formula)

SALTSTACK을 이용한 레디스 HA 자동 설치



SALTSTACK을 이용한 레디스 HA 자동 설치

SaltStack의 Orchestration 사용



SALTSTACK을 이용한 레디스 HA 자동 설치

- pillar 파일은 사용자 정의 변수
- 설치할 대상 인스턴스의 정보 입력
- 설치 시마다 pillar 파일 재작성

pillar

```
install_method:
sentinel
sentinel_port: 16379
service_name:
garim_test
nodes:
  - role: master
    version: 7.0.8
    host: serverA
    ip: 10.10.10.1
    port: 6379
  - role: replica
    version: 7.0.8
    host: serverB
    ip: 10.10.10.2
    port: 10379
  - role: sentinel
    version: 7.0.8
    host: server
    ip: 10.10.10.3
    port: 6379
```

SALTSTACK을 이용한 레디스 HA 자동 설치

- pillar의 install_method 확인해서 설치 방법 결정
- 동일한 복제 그룹끼리 같은 내용을 공유하기 위해 오케스트레이션 사용

install.sls

```
{% set pw = salt['random.get_str'](length=30, punctuation=False) %}

{% if pillar['install_method'] == 'single' %}
install_single_redis:
  salt.state:
    - tgt: {{ pillar['host'] }}
    - sls:
      - install_redis
      - install_exporter
    - pillar:
      rversion: {{ pillar['version'] }}
      rip: {{ pillar['ip'] }}
      rport: {{ pillar['port'] }}
      rpw: {{ pw }}
```

SALTSTACK을 이용한 레디스 HA 자동 설치

```
{% elif pillar['install_method'] == 'sentinel' %}

{% for node in pillar['nodes'] %}
install_redis_{{ node['role'] }}:
  salt.state:
    - tgt: {{ node['host'] }}
    - sls:
      - install_redis
    - pillar:
        rversion: {{ node['version'] }}
        rip: {{ node['ip'] }}
        rport: {{ node['port'] }}
        rpw: {{ pw }}

{% endfor %}

make_replication:
  salt.function:
    - name: cmd.run
    - tgt: {{ node['host'] }}
    - arg:
      - /home/centos/redis_{{ node['port'] }}/bin/redis-cli -p {{ node['port'] }} -a {{ pw }} replicaof
        {{ ns.mip }} {{ ns.mport }}
```

SALTSTACK을 이용한 레디스 HA 자동 설치

```
{% for node in pillar['nodes'] %}

install_sentinel_{{ node['role'] }}:
  salt.state:
    - tgt: {{ node['host'] }}
    - sls:
      - install_sentinel
    - pillar:
      rport: {{ node['port'] }}
      rpw: {{ pw }}
      sport: {{ pillar['sentinel_port'] }}
      mip: {{ ns.mip }}
      mport: {{ ns.mport }}

{% endfor %}
```

SALTSTACK을 이용한 레디스 HA 자동 설치

- redis 설치 시나리오
 - 공통 정보를 pillar에서 사용 가능

install_redis.sls

```
redis-dependencies:
  pkg.installed:
    - names:
      - make
      - libxml2-devel
      - gcc
      - jemalloc
      - iptables-services

get-redis:
  archive.extracted:
    - name: {{ base_dir }}/redis_{{ pillar['rport'] }}
    - source: salt://packages/nhn-redis-{{ pillar['rversion'] }}.tar.gz
    - trim_output: True
    - enforce_toplevel: False
    - options: "--strip-components=1"
```

SALTSTACK을 이용한 레디스 HA 자동 설치

change-config:

cmd.run:

- cwd: {{ base_dir }}/redis_{{ pillar['rport'] }}
- names:
 - sed -i "s/port 6379/port {{ pillar['rport'] }}/" redis.conf
 - sed -i "s/bind 127.0.0.1 -:::1/bind {{ grains['fqdn_ip4']|join(' ') }} 127.0.0.1 -:::1/" redis.conf
 - sed -i "s/# maxmemory <bytes>/maxmemory {{ grains['mem_total'] // 2}}mb /" redis.conf
 - sed -i "s/# requirepass foobared/requirepass

{{ pillar['rpw']|default(salt['random.get_str'](length=30, punctuation=False), true) }}/" redis.conf

- sed -i "s/# masterauth <master-password>/masterauth

{{ pillar['rpw']|default(salt['random.get_str'](length=30, punctuation=False), true) }}/" redis.conf

make-install:

cmd.run:

- cwd: {{ base_dir }}/redis_{{ pillar['rport'] }}
- runas: irteam
- names:
 - make -j 4
- require:
 - pkg: redis-dependencies

SALTSTACK을 이용한 레디스 HA 자동 설치

- sentinel 설치 시나리오
 - 공통 정보를 pillar에서 사용 가능

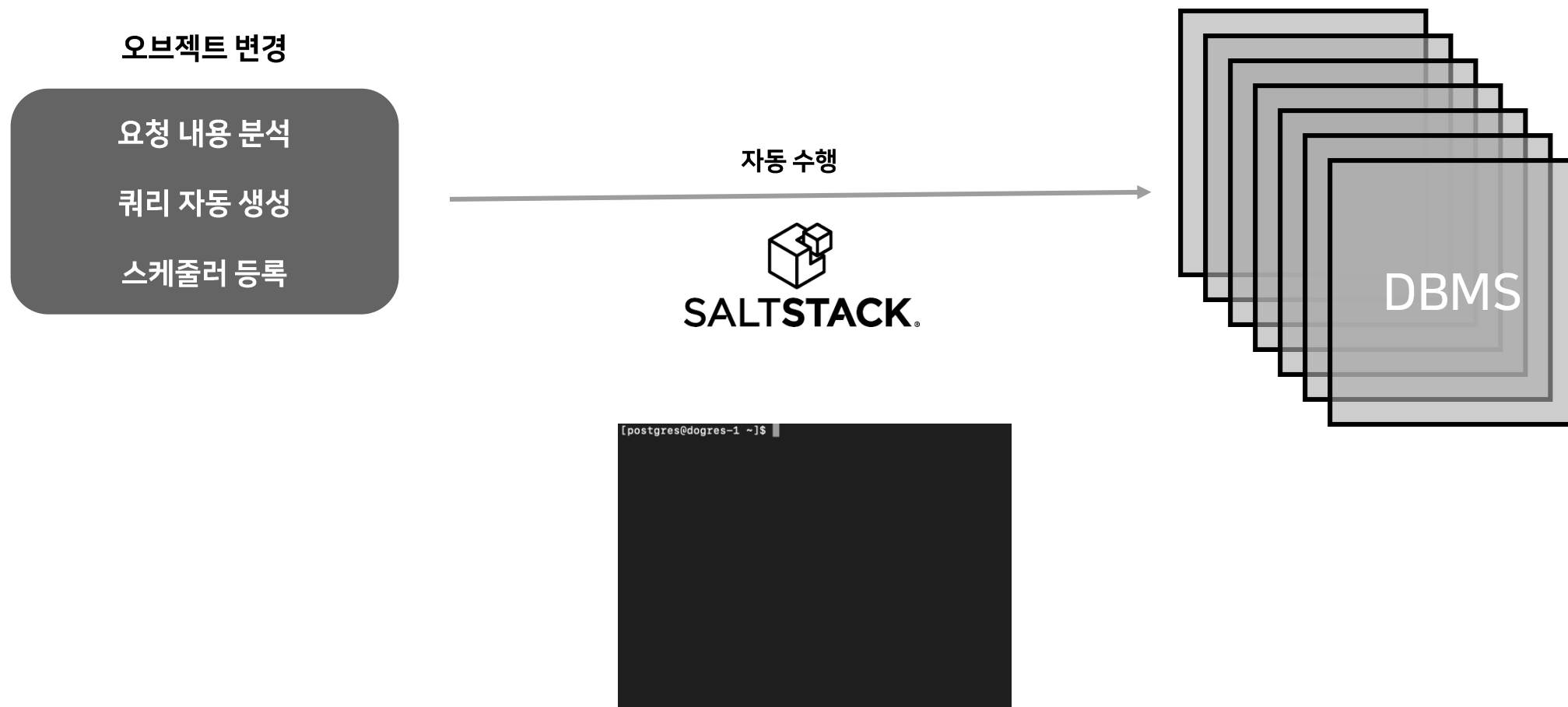
install_sentinel.sls

```
{% set base_dir = '/home1/irteam' -%}

change-config:
  cmd.run:
    - cwd: {{ base_dir }}/redis_{{ pillar['rport'] }}
    - runas: irteam
    - names:
      - sed -i "s/port 26379/port {{ pillar['sport'] }}/" sentinel.conf
      - sed -i 's/daemonize no/daemonize yes/' sentinel.conf
      - sed -i "s/mymaster 127.0.0.1 6379/{{ pillar['service_name'] }} {{ pillar['mip'] }}
{{ pillar['mport'] }}/" sentinel.conf
      - sed -i "s/mymaster/{{ pillar['service_name'] }}/" sentinel.conf
      - echo "sentinel auth-pass {{ pillar['service_name'] }} {{ pillar['rpw'] }}" >> sentinel.conf
```

관리의 자동화

오브젝트 변경의 자동화



데이터베이스 관리의 자동화

백업 결과 / 백업 실패 관리 자동화

백업 실패 관리

기간

시작일

2023-01-29

종료일

2023-02-05

검색

백업일	백업실패(로컬)	백업실패(원격)	미확인
2023-01-29	0	0	0
2023-01-30	0	0	0
2023-01-31	3	3	2
2023-02-01	1	1	0
2023-02-02	0	1	0
2023-02-03	0	0	0

광기

02.신규

Pa

Pa

NE

캐시

장비

To

To

백업

백업

백업

백업

추천

백업일 : 2023-02-01

서비스	DBMS	호스트	PORT/Linux_id	백업방법	결과(원격)	DBA	백업실패처리	처리일시
Dc	MySQL	dr	13306	DUMP			서비스 오픈전 (신규서버)	2023-02-03 09:44:43

닫기

장비	백업일	백업시간	백업종류	백업방법	결과(원격)	DBA	백업용량	백업상태	백업처리	백업일시	백업용량	백업상태
장비	2023-02-03	ba	23306	XTRA_y	03:00:02	03:15:25	192TB	SUCCESS	08:00:23	09:42:20	586TB	성공
To	2023-02-03	ba	13306	TRAN_y	01:00:02	01:01:13	1.2MB	SUCCESS	02:30:00	02:30:03	1.3MB	성공
To	2023-02-03	ba	13306	TRAN_y	01:00:02	01:01:12	2.4MB	SUCCESS	03:00:14	03:00:18	2.5MB	성공
백업	2023-02-03	bg	13306	TRAN_y	02:29:16	02:29:16	37TB	SUCCESS	09:00:13	09:12:38	37TB	성공
백업	2023-02-03	bg	13306	TRAN_y	01:00:02	01:13:14	39TB	SUCCESS	04:30:13	04:41:57	39TB	성공
백업	2023-02-03	bg	13306	TRAN_y	02:00:02	02:01:12	6.1MB	SUCCESS	07:00:33	07:00:38	6.2MB	성공
백업	2023-02-03	bg	13306	TRAN_y	01:10:02	01:11:13	85MB	SUCCESS	04:00:26	04:00:33	85MB	성공
추천	2023-02-03	bg	13306	TRAN_y	01:00:02	07:36:26	74TB	SUCCESS	08:00:24	08:31:14	74TB	성공

복구에는 실패해도 백업에 실패한 DBA는 용서받지 못한다.
(리얼 월드에서는 둘 다 용서받지 못합니다.)

<

1

2

3

4

5

...

32

33

34

35

>

데이터베이스 관리의 자동화

프로세스 리스트 수집 자동화

모니터링 - Processlist

호스트

aspm-mbs-t1001:13306

YYYYMMDD

20230204

HHMMSS

10

REPL, ADMIN

☐ 제외

LIMIT

100

검색

No	LogTime	HOST	PORT	PID	USER	IP	DB	Command	Exec Time	State	Query
20	20230204100011	as,	13306	26515196	admin	10.1	mysql	Query	0	starting	SHOW FULL PROCESSLIST SELECT r.* FROM (SELECT c.coupon_creation_cd AS couponNo ,c.coupon_nm AS couponName ,c.coupon_id AS couponId ,c.issue_type_cd AS issueTypCode
21	20230204100011	as	13306	26489043	mer	10.1	me	p Execute	0	Sending data	
22	20230204100012	as	13306	6126123	system user		NULL	Connect	4	Slave has read all relay log; waiting for more updates	NULL

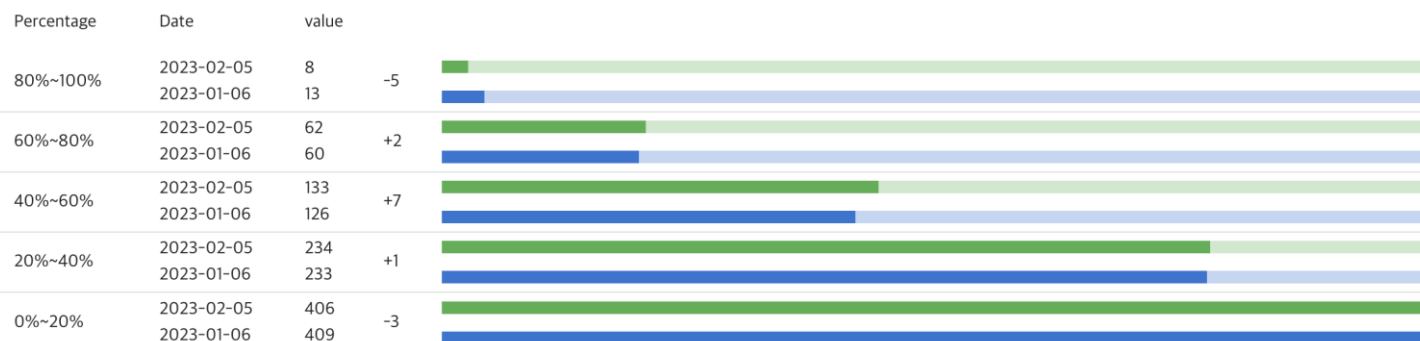
실시간 lock 정보 확인

[illegible]

데이터베이스 관리의 자동화

Disk 사용량 조회

통계 - Disk 사용량



DBA 디스크 사용률 서비스 분류 서비스명 COMPARE : 일 전

DATE : 2023-02-05, COUNT : 8

서비스명	DBA	HOST	PORT	TOTAL	USED	FREE	사용률	전체증가량	하루증가량	Due Date	ISSUE
추	김	bg	3306	802G	715G	87G	90%	7 G	255M	349	OK
추	김	bg	3306	803G	672G	131G	84%	5 G	200M	672	OK
pa	김	hsj	3306	569G	488G	81G	86%	3 G	111M	751	OK
pa	김	hsj	3306	569G	489G	80G	86%	3 G	104M	785	OK
Dc	백	dri	3306	976G	771G	205G	80%	6 G	208M	1010	OK
남	남	tcr	3306	278G	227G	52G	82%	90M	3M	17670	OK
아	김	ian	3306	776G	615G	161G	80%	-2742M	-91M	-	OK
신	권	gsi	3306	2656G	2142G	514G	81%	-225476M	-7516M	-	OK

데이터베이스 관리의 자동화

DB / HA 자동 설치

MMM 자동 설치 등록

MMM 모니터

infm-m

MMM 포트

MMM 버전

20131115

PREFIX

/hom

MySQL HOST:PORT

+ 추가

+ SLAVE 인과 추가

VIP

WRITER VIP

설치 진행 현황

☐

MMM 모니터

☐

infm-t1901@12101

MySQL 자동 설치 등록

Host:Port / IP Addr

MySQL Version

8.0.32_오브젝트_대소문자구분

Prefix

/home1/.../mysql

복제구성 (마스터 Host:Port...)

☒

Start Slave

+ 추가

☐

상세 설정

- 삭제

설치 등록

설치 진행 현황

☐	인스턴스	DB구성정보	DiskType	MEM Server	진행상태	최종변경일시
☐	mymm-t1802:13306	5740, /utf8mb4 change to master : i	ALL	prometheus	설치완료	2023-01-31 17:25:56

데이터베이스 관리의 자동화

구성(configuration) 자동 비교

DB Variables 비교

MYSQL HOST:PORT

+ 추가

+ SLAVE 일괄 추가

- 일괄 삭제

aspm-13306

- 삭제

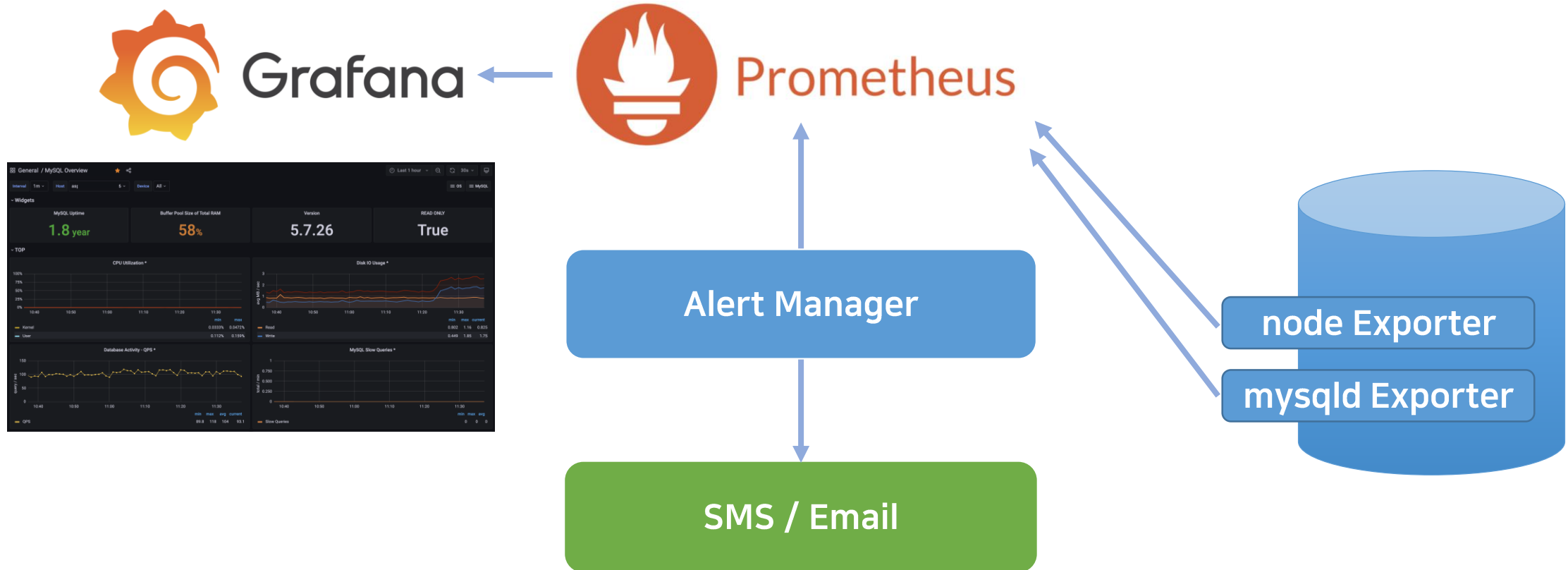
aspm-:13306

- 삭제

Variable	aspm-mbs-t1001	aspm-mbs-t1002
general_log_file	/home1/mysql/data/aspm-t1001.log	/home1/mysql/data/aspm-t1002.log
hostname	aspm-mbs-t1001	aspm-mbs-t1002
log_error	/home1/mysql/data/aspm-t1001.err	/home1/mysql/data/aspm-t1002.err
pid_file	/home1/mysql/data/aspm-t1001.pid	/home1/mysql/data/aspm-t1002.pid
read_only	OFF	ON
report_host	aspm-t1001	aspm-t1002
server_id	9441	9445
server_uuid	0a152b25-8c43-11eb-8858-48df37bcfe6c	0bc602b7-8c43-11eb-88f7-48df37bcf50c

모니터링의 자동화

모니터링 시스템 구성



모니터링의 자동화

알림 설정 대상과 임계치 설정

SMS - Event

선택
Ever

준영 테스트목록

NO		Host Name
5	☐	Host Name
6	☐	ifsr-dev
7	☐	ifsr-dev
8	☐	ifsr-dev
	☐	ifsr-dev
	☐	ifsr-dev

Skip 대상

☐ Host ☒ Hostgroup ☐ Service

Host Group

준영 테스트목록

검색

Event

ALL TROUBLES

Skip 기간

☒ 요일 ☐ 기간 ☐ 반복

ex) 1(Mon),2,..7 or 0(Everyday)

요일

0

시작/종료 시간

HH:MM

15:00

~

HH:MM

16:00

메모

준영/테스트중

사용여부

☐ 사용 ☒ 미사용

HOST 추가

HOST 삭제

EVENT 등록

NHN Cloud에서 데이터베이스 운영하기

NHN Cloud의 데이터베이스

- 관리형 데이터베이스
 - 자동 설치, 백업, 복원
 - 모니터링, 알람 설정 등
- 상품 종류
 - RDS for MySQL
 - RDS for MariaDB
 - RDS for MS-SQL
 - EasyCache(Redis)



Database



RDS for MySQL UPDATE

복잡한 설치 및 설정 과정 없이, 클라우드 환경에서 관계형 데이터베이스를 사용할 수 있습니다.



RDS for MariaDB UPDATE

복잡한 설치 및 설정 과정 없이, 클라우드 환경에서 관계형 데이터베이스를 사용할 수 있습니다.



RDS for MS-SQL UPDATE

복잡한 설치 및 설정 과정 없이, 클라우드 환경에서 관계형 데이터베이스를 사용할 수 있습니다.



EasyCache UPDATE

간단하게 설정해 클라우드 환경에서고가용성의 인메모리 캐시를 사용할 수 있습니다.

NHN Cloud의 데이터베이스

- 설치형 데이터베이스

- 원하는 데이터베이스가 설치된 인스턴스

- 추가 금액 없음

- 상품 종류

- MS-SQL Instance

- MySQL Instance

- PostgreSQL Instance

- Cubrid Instance

- MariaDB Instance

- Tiberio Instance

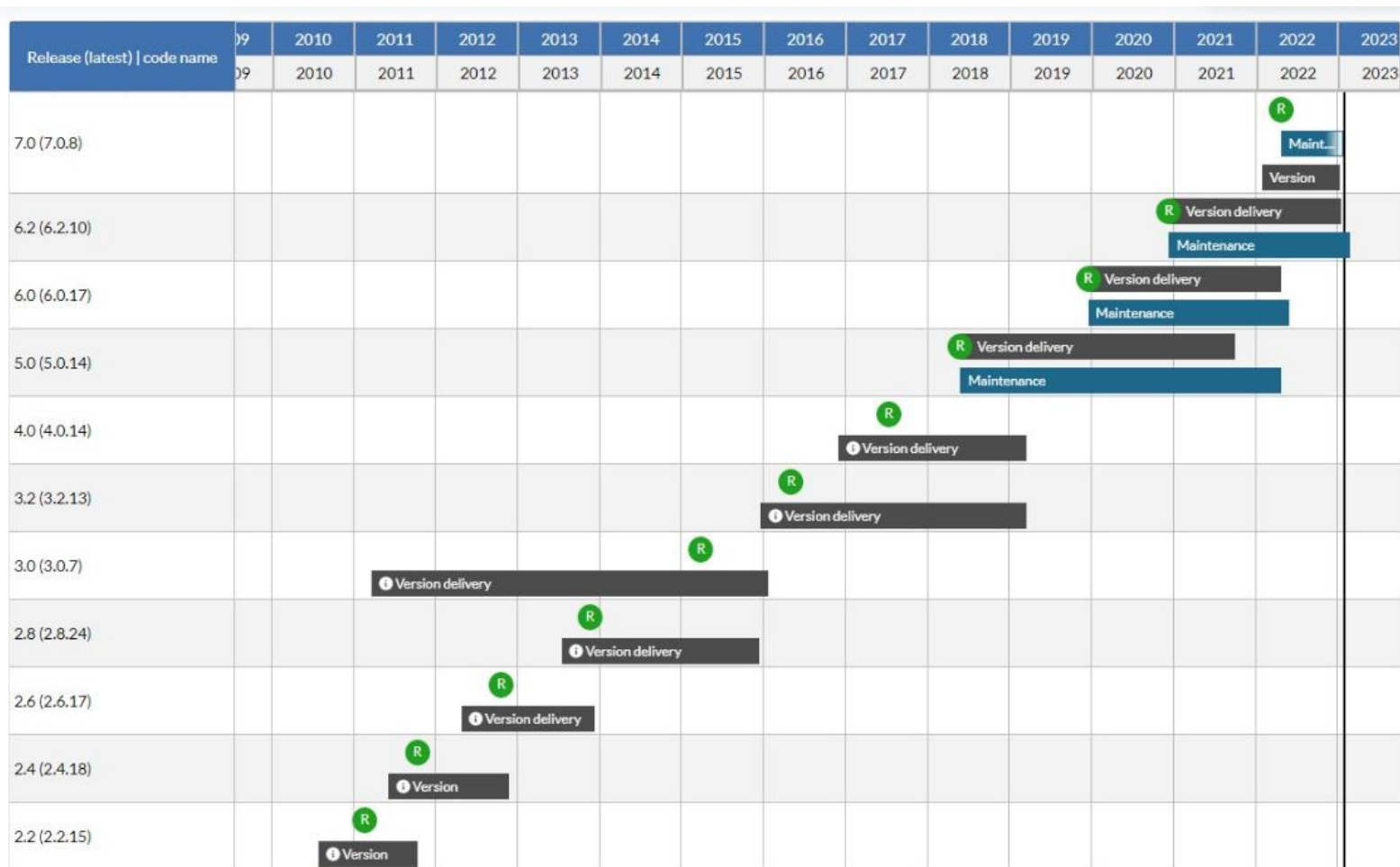
- Redis Instance

- Kafka Instance

RDS for MySQL

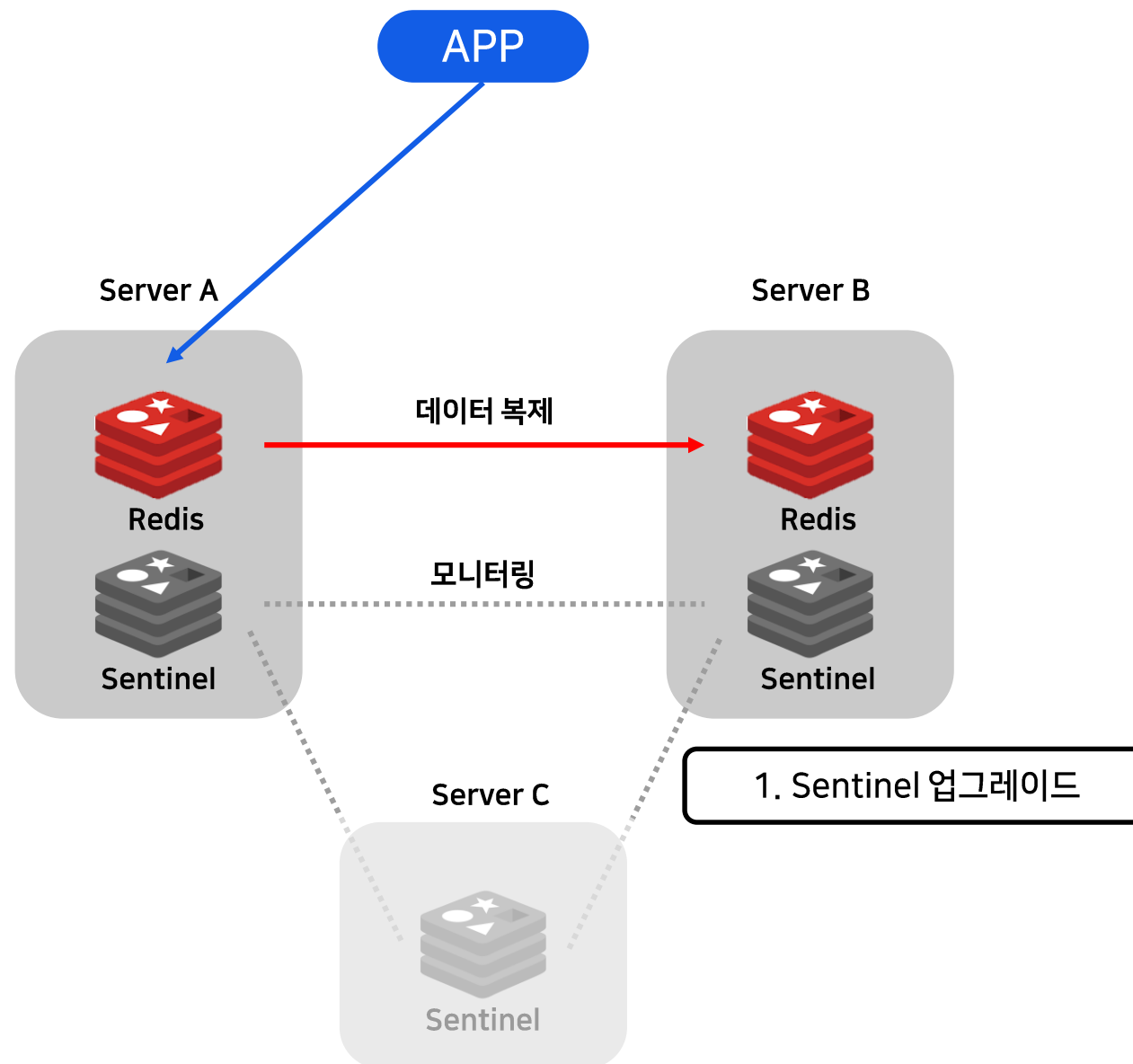
- RDS 설치/복원 시연
 - 기존 설치와 복원이라면?
 - 서버 발급 신청, 사용할 버전의 설치파일 다운로드 등
 - 백업파일을 복구할 서버 조회 등

레디스의 EOL



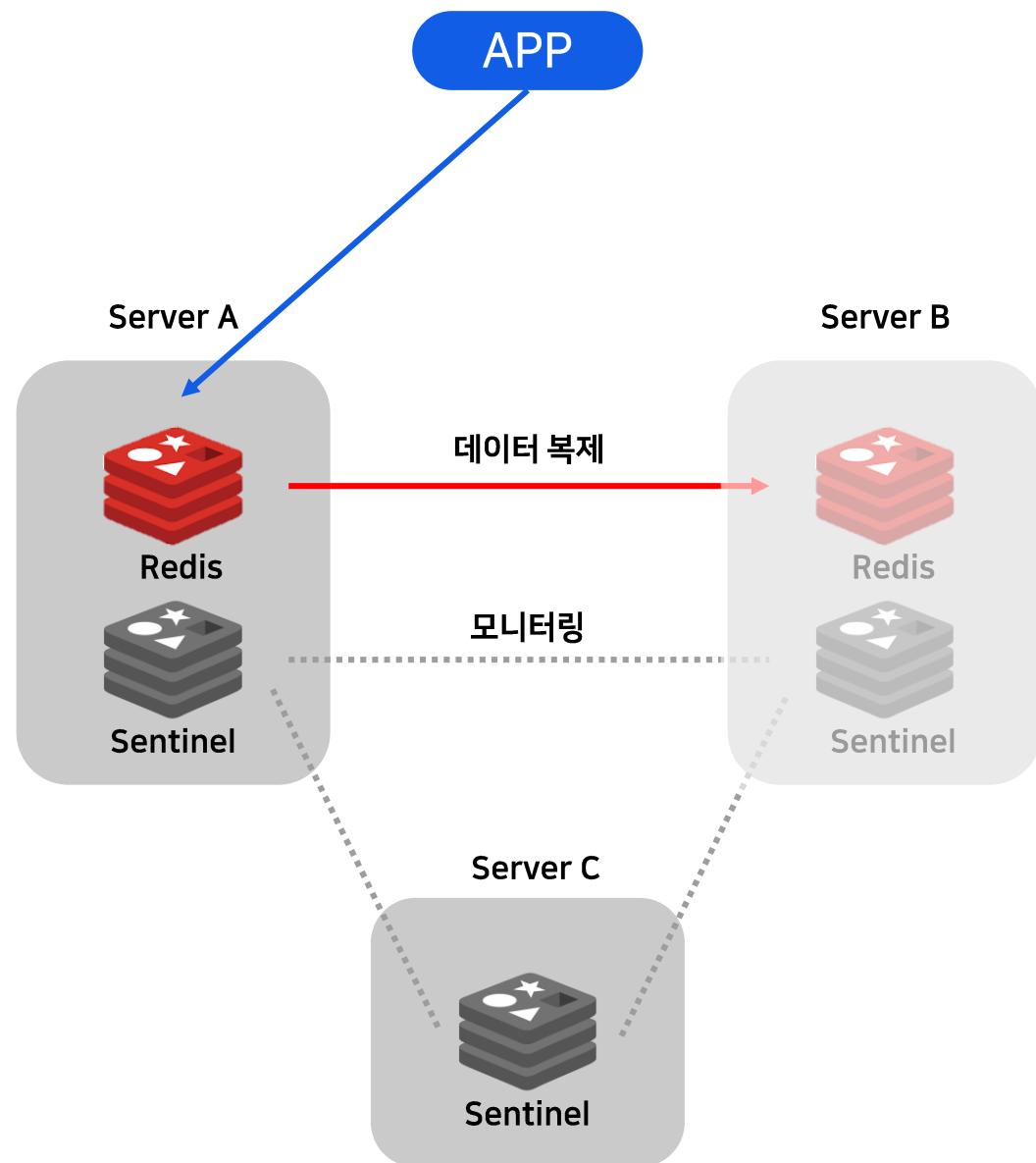
EasyCache

버전 업그레이드



EasyCache

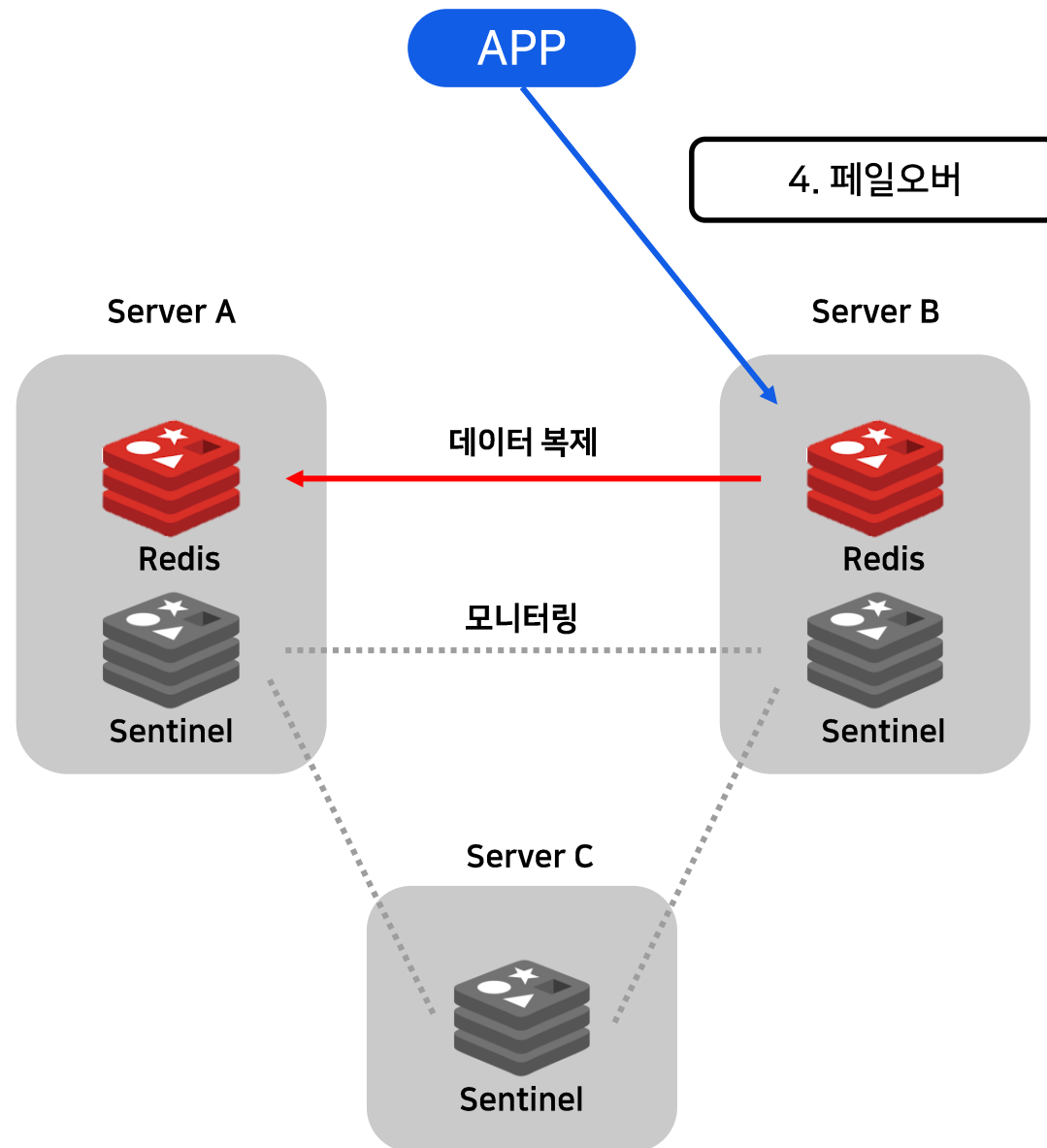
버전 업그레이드



- 2. Replica Sentinel 업그레이드
- 3. Replica Redis 업그레이드

EasyCache

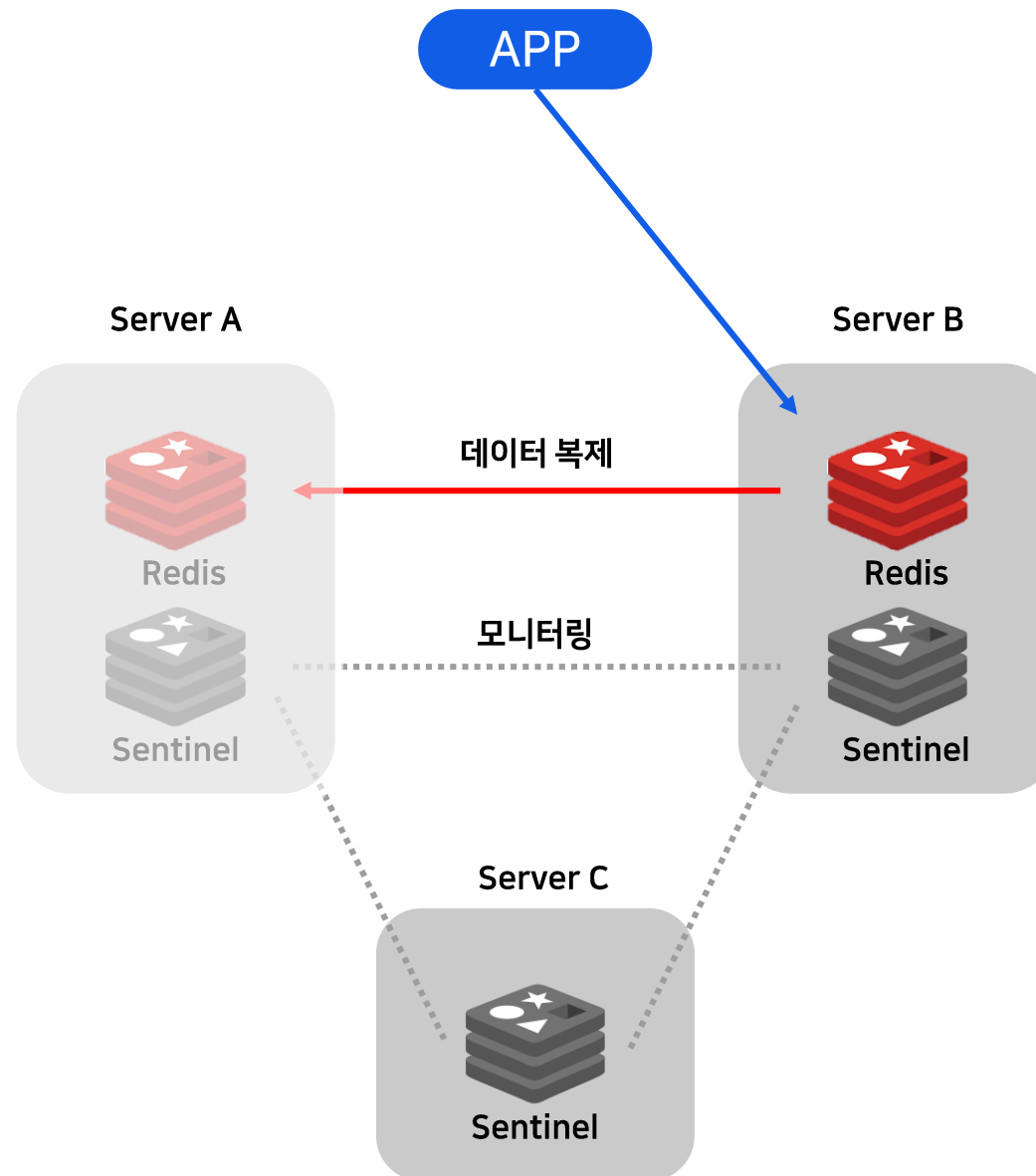
버전 업그레이드



EasyCache

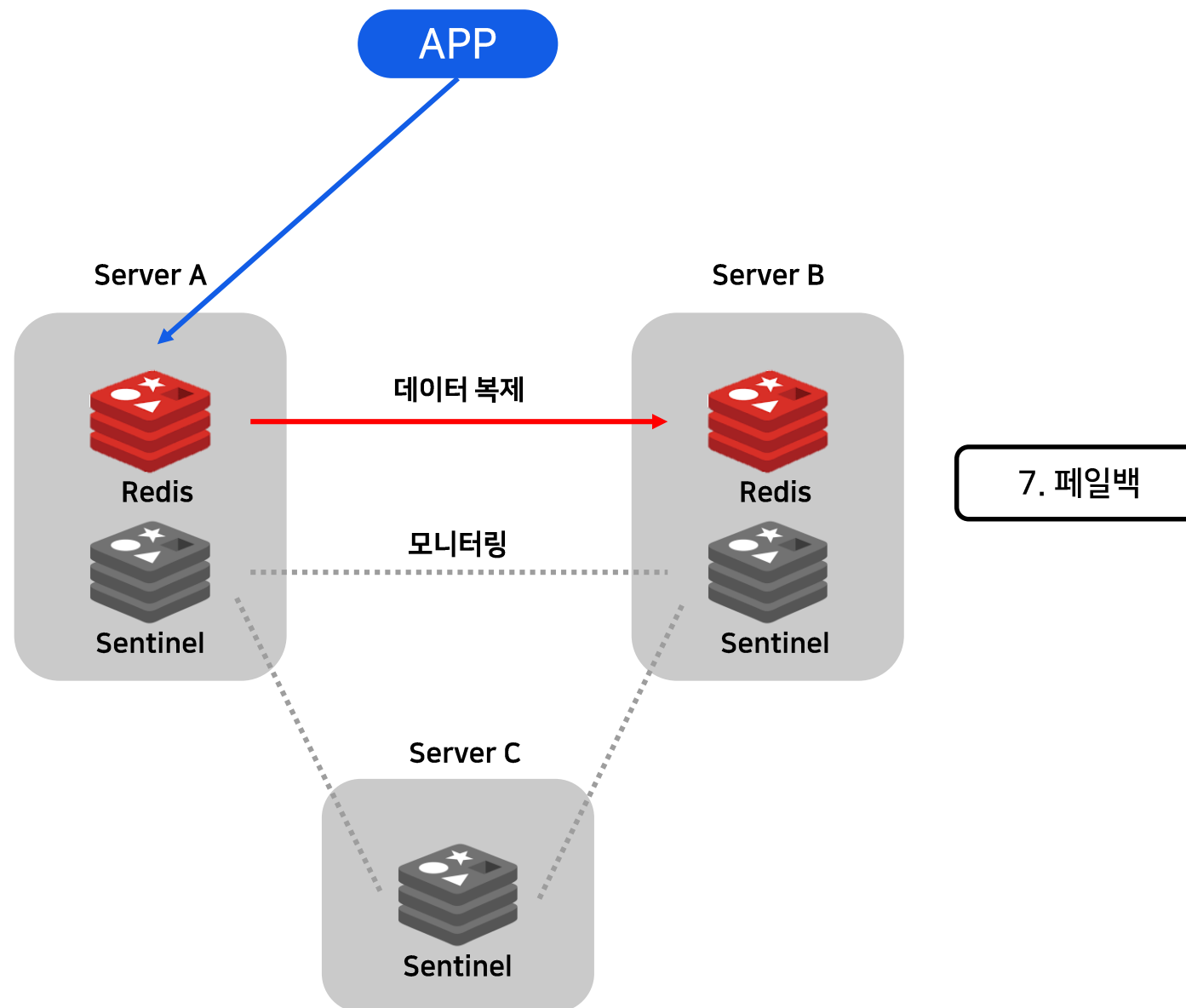
버전 업그레이드

- 5. Master Sentinel 업그레이드
- 6. Master Redis 업그레이드



EasyCache

버전 업그레이드



EasyCache

모니터링과 알람 구축

EasyCache > 모니터링

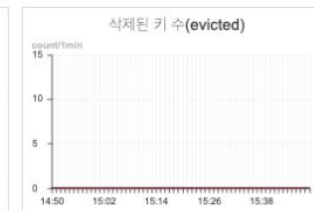
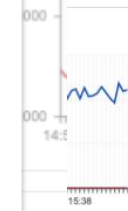
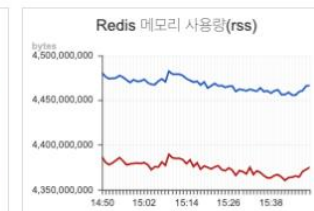
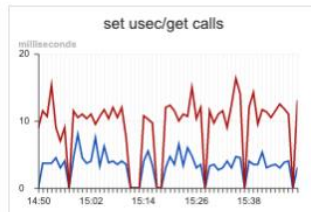
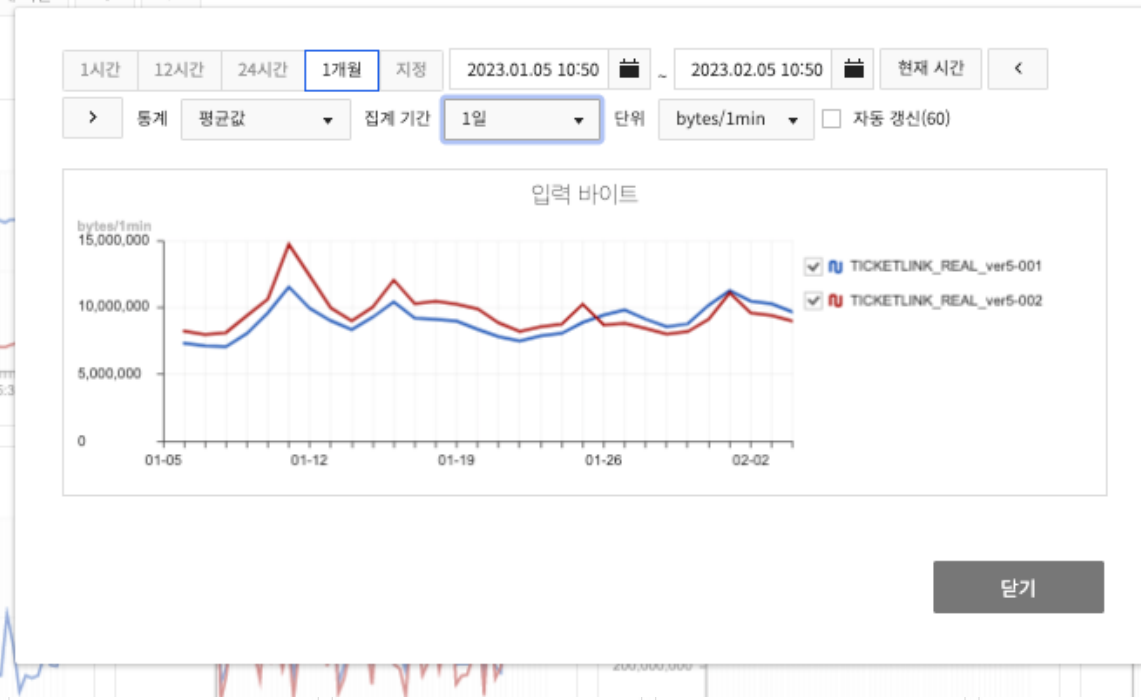
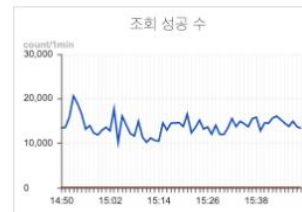
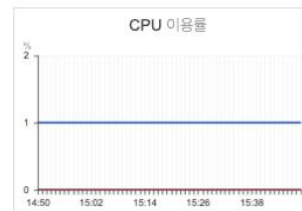
복제 그룹 모니터링 백업 프로파일 설정 알람 이벤트

URL & Appkey 사용자 가이드 메뉴 가이드

1시간 12시간 24시간 1개월 지정 2023.02.05 14:50 ~ 2023.02.05 15:50 현재 시간 < >

TICKETLINK_PRODUCT < > 자동 갱신(60) < >

필터 조건



Redis Instance

- 클러스터 자동 생성 시연

클라우드 환경에서 우리는 뭘 먹고 살죠?

<https://www.roberthalf.com/salary-guide>

- 로버트 하프 테크놀로지의 2023년 IT 유망 직종에 따르면..
 - 1위 클라우드 엔지니어
 - 2위 데이터베이스 개발자
 - 3위 데브옵스 엔지니어
- 클라우드 환경이 가속되면서 안정된 공급이 필요해짐.

데이터베이스 운영을 꿀같이...

유용한 도구 소개

<https://dbeaver.io/>



DBeaver Community

Free Universal Database Tool

Star 30,247 Follow @dbeaver_news

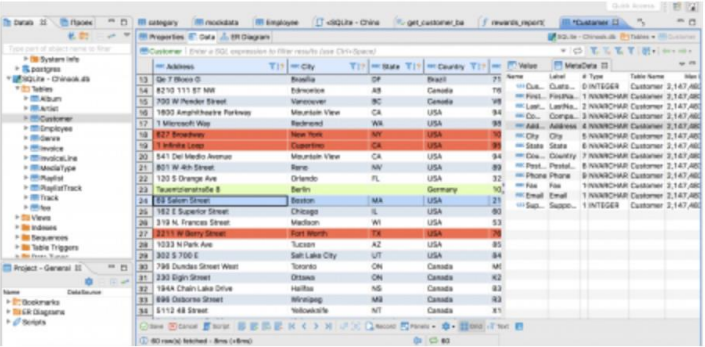
search here ... Go

- Home
- About
- Download
- Sources
- Documentation
- News
- Support
- DBeaver PRO
- CloudBeaver

Universal Database Tool

Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.

[Download](#)



1. DB 관리 툴
2. 다양한 커넥션 제공
3. 무료
4. 윈도우, 맥 사용 가능

유용한 도구 소개

<https://dbeaver.io/>

The screenshot displays the DBeaver 22.3.3 interface. The top panel shows a SQL script: `select * from etladm.r_step rs`. The bottom panel shows the results of the query, which is a table with 14 rows and 5 columns: `id_step`, `id_transformation`, `NAME`, `description`, and `id_step_type`. The results are as follows:

	id_step	id_transformation	NAME	description	id_step_type
1	15,781	1,171	Blocking Step	[NULL]	15
2	15,782	1,171	Calculator 3	[NULL]	16
3	15,783	1,171	Detect empty stream	[NULL]	40
4	15,784	1,171	Get System Info 3	[NULL]	71
5	15,785	1,171	Hadoop File Output 2	[NULL]	80
6	15,786	1,171	Metadata structure of stream	[NULL]	117
7	15,787	1,171	Metadata structure of stream 2	[NULL]	117
8	15,788	1,171	RUNSSH_ETL	[NULL]	215
9	15,789	1,171	RUNSSH_SPOON	[NULL]	215
10	15,790	1,171	Switch / Case	[NULL]	201
11	15,791	1,171	Table input1	[NULL]	207
12	15,792	1,171	Value Mapper	[NULL]	222
13	15,793	1,171	Write_Log_End	[NULL]	52
14	15,794	1,171	Write_Log_Start	[NULL]	52

The status bar at the bottom indicates that 200 row(s) were fetched in 19ms on 2023-02-05 at 15:53:37. The interface also shows a left sidebar with a tree view of database objects and a bottom panel with project settings.

유용한 도구 소개

<https://dbeaver.io/>



MySQL



PostgreSQL



Db2 for LUW



DuckDB



MariaDB



Oracle



SQL Server



SQLite



Apache Calcite
Avatica



Apache Drill



Apache Hive



Apache Ignite



Apache Kylin



Apache Kyuubi



Apache Phoenix



Apache Spark



Athena



Azure
Databricks



Azure SQL
Server



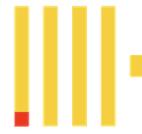
Babelfish via
TDS (beta)



Cache



ClickHouse



ClickHouse
(Legacy)



Cloudera Impala



CockroachDB



CrateDB



CSV



CUBRID



Data Virtuality



Db2 for IBM i



Db2 for LUW
(Old 8.x)



Db2 for z/OS



DBF



Denodo 8



Derby
Embedded



Derby Server

유용한 도구 소개

<https://dbeaver.io/>



Dremio



EDB



Elasticsearch

Exasol

Exasol



Firebird



Gemfire XD



Google BigQuery



Google Cloud Spanner



Google Cloud SQL - PostgreSQL



Greenplum



H2 Embedded



H2 Embedded V.2



H2 Server



H2GIS Embedded



H2GIS Server



HANA



HSQL Embedded



HSQL Server



Informix

INGRES

Ingres



InterSystems IRIS



Jennifer



Kognitio

MACH BASE

Machbase



MaxDB



Mimer SQL



MonetDB



MS Access (UCanAccess)



MySQL (Old)



MySQL 5



NDB Cluster



Neo4j



Netezza



NuoDB



OceanBase



Ocient

유용한 도구 소개

<https://dbeaver.io/>



OmniSci
(formerly
MapD)



Open Distro
Elasticsearch



OpenEdge



OpenSearch



OrientDB



Pervasive SQL



PostgreSQL
(Old)



PrestoDB



PrestoSQL



Redshift



Salesforce CDP



SAP ASE
jConnect



SnappyData



Snowflake



Solr



Microsoft
SQL Server
SQL Server (Old
driver, jTDS)



Microsoft
SQL Server
SQL Server (Old
driver, MS)



SQream DB



Sybase
jConnect



Sybase jTDS



Teradata



TimescaleDB



Trino



Vertica



Virtuoso



Yellowbrick



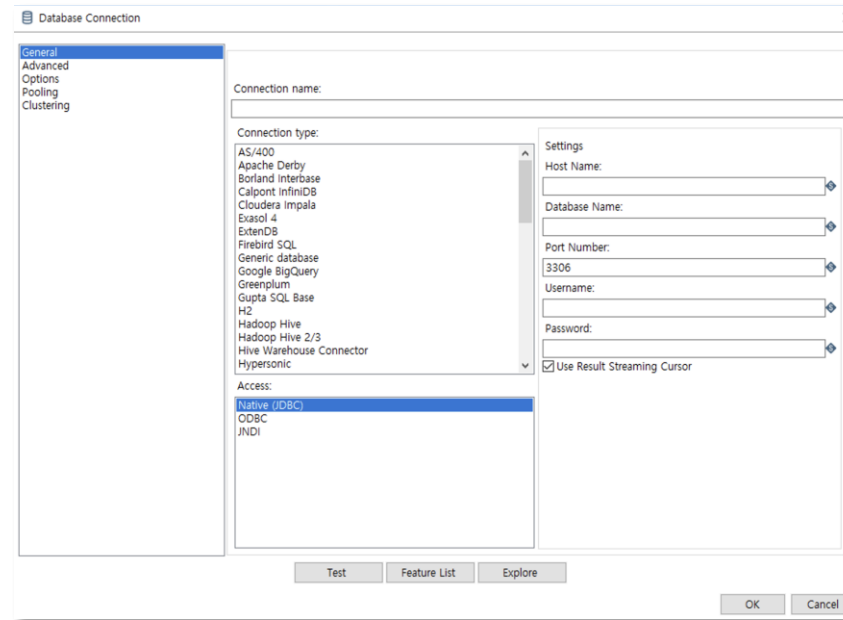
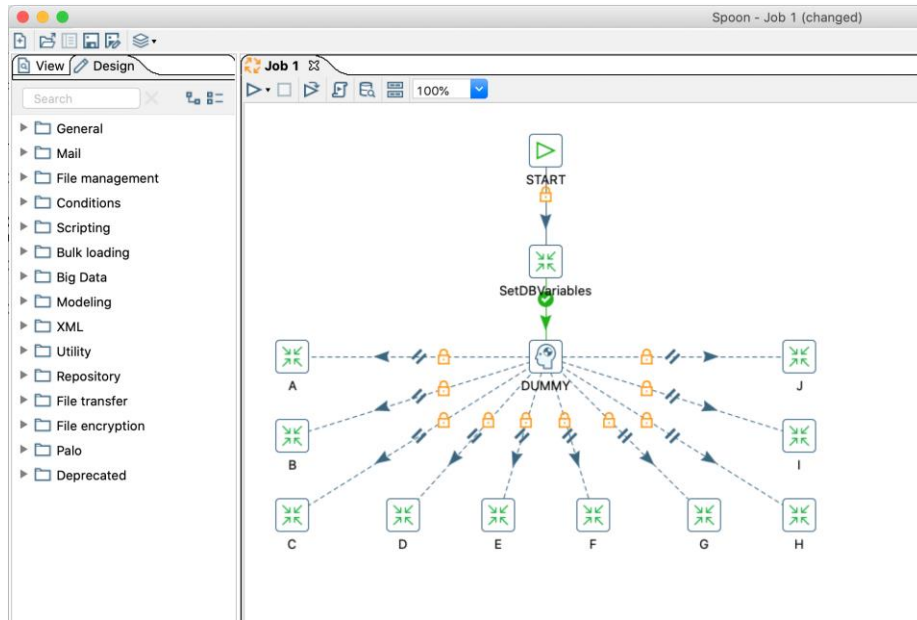
yugabyteDB
YugabyteDB

유용한 도구 소개

<https://sourceforge.net/projects/pentaho/>



1. 다양한 커넥션 제공
2. ETL 기능만 사용해도 강력
3. 무료(GNU2)
4. 윈도우, 리눅스 사용 가능



유용한 도구 소개

PDI 예제

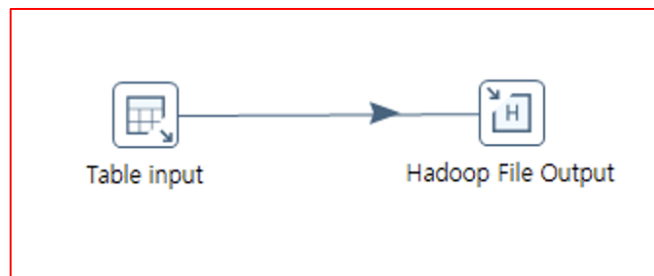


Table input

Step name: Table input

Connection: gsmm-mlogdb-t1002_NG_Log_1330

SQL

```
SELECT
match_no
, card_rule
, bet_rule
, seed
, rematch_cnt
, after_total_game_money
, total_betting_money
, total_rematch_joinextra_money
, dealer_money
, extra_log
, start_date
, last_date
, insert_date
FROM ng_sudda_game_result_log
where insert_date >= DATE_FORMAT(date_add('${VAR_PARTITIONKEY}', interval 0 hour), '%Y%m%d')
and insert_date < DATE_FORMAT(date_add('${VAR_PARTITIONKEY}', interval 24 hour), '%Y%m%d')
```

Hadoop file output

Step name: Hadoop File Output

Hadoop Cluster: DOG

Folder/File: /bidw/bistaging/\${Internal.Transformation.Name}/partitionkey=\${VAR_PARTITIONKEY}

Create Parent folder: ☒

Do not create file at start: ☐

Accept file name from field: ☐

File name field:

Extension: txt

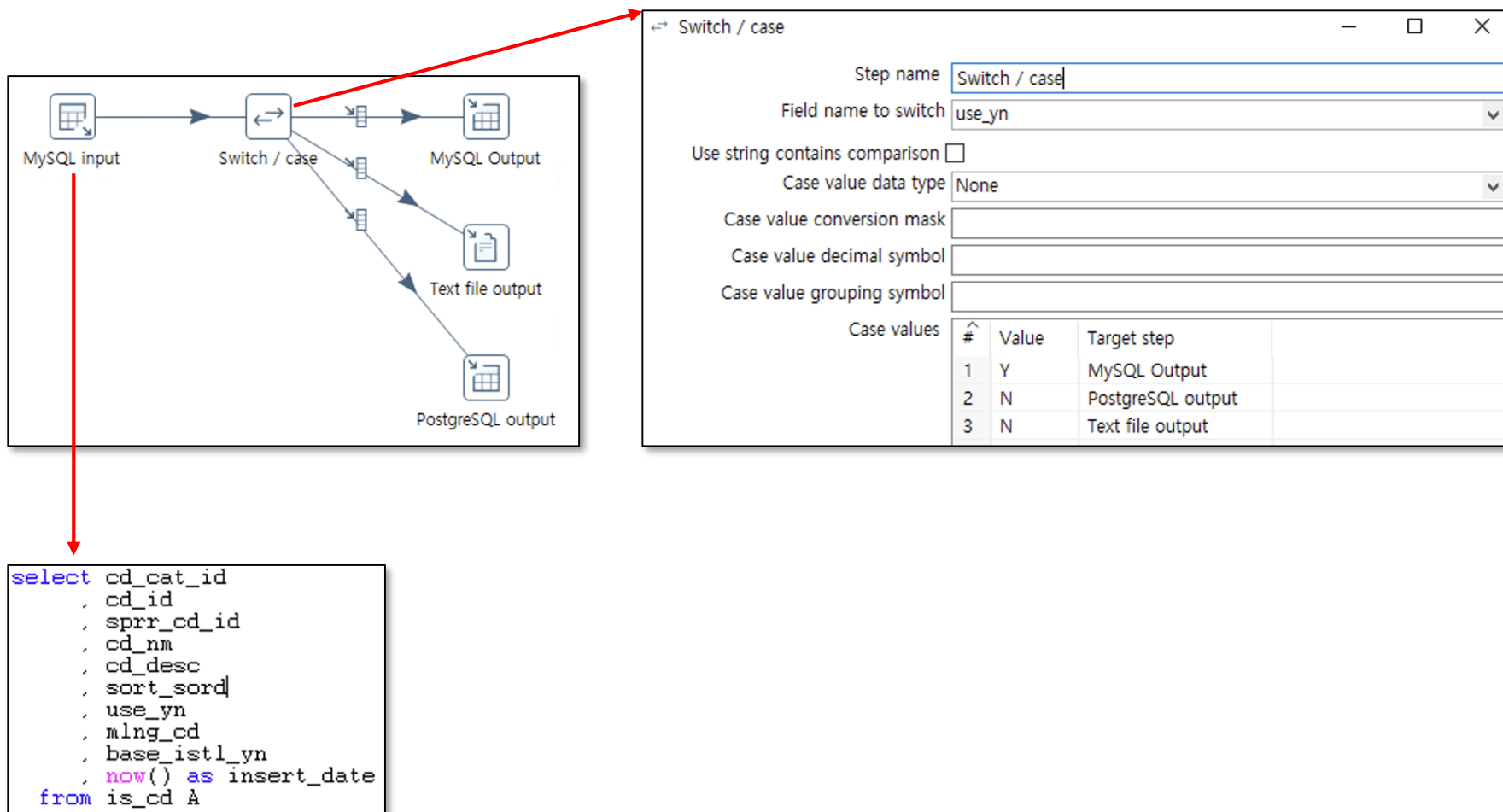
Hadoop file output

Step name: Hadoop File Output

#	Name	Type	Format	Length	Precision	Currency	D
1	match_no	Integer	0				
2	card_rule	Integer	0				
3	bet_rule	Integer	0				
4	seed	Integer	0				
5	rematch_cnt	Integer	0				
6	last_rematch_type	Integer	0				
7	rematch_win_to_win_cnt	Integer	0				
8	rematch_94_cnt	Integer	0				
9	rematch_10JA94_cnt	Integer	0				

유용한 도구 소개

PDI 예제



개발/운영 터미널을 구분

```
DOG - cubrid1@ifsc-dev2-wa801:/home1/cubrid1 - ssh - db - 127x59

-rw-r--r-- 1 cubrid1 cubrid 95860318 2월 5 21:07 service_rw_154.sql.log
-rw-r--r-- 1 cubrid1 cubrid 63956805 2월 5 21:07 service_rw_166.sql.log
-rw-r--r-- 1 cubrid1 cubrid 22938941 2월 5 21:07 service_rw_167.sql.log
-rw-r--r-- 1 cubrid1 cubrid 78331146 2월 5 21:07 service_rw_187.sql.log
-rw-r--r-- 1 cubrid1 cubrid 39191817 2월 5 21:07 service_rw_122.sql.log
-rw-r--r-- 1 cubrid1 cubrid 40390936 2월 5 21:07 service_rw_158.sql.log
-rw-r--r-- 1 cubrid1 cubrid 31003039 2월 5 21:07 service_rw_160.sql.log
-rw-r--r-- 1 cubrid1 cubrid 82787476 2월 5 21:07 service_rw_187.sql.log
-rw-r--r-- 1 cubrid1 cubrid 78097323 2월 5 21:07 service_rw_40.sql.log
-rw-r--r-- 1 cubrid1 cubrid 70270238 2월 5 21:07 service_rw_42.sql.log
-rw-r--r-- 1 cubrid1 cubrid 72032063 2월 5 21:07 service_rw_47.sql.log
-rw-r--r-- 1 cubrid1 cubrid 83805617 2월 5 21:07 service_rw_50.sql.log
-rw-r--r-- 1 cubrid1 cubrid 13195727 2월 5 21:07 service_rw_126.sql.log
-rw-r--r-- 1 cubrid1 cubrid 36913187 2월 5 21:07 service_rw_25.sql.log
-rw-r--r-- 1 cubrid1 cubrid 84866768 2월 5 21:07 service_rw_26.sql.log
-rw-r--r-- 1 cubrid1 cubrid 26973452 2월 5 21:07 service_rw_27.sql.log
-rw-r--r-- 1 cubrid1 cubrid 54426883 2월 5 21:07 service_rw_28.sql.log
-rw-r--r-- 1 cubrid1 cubrid 38410388 2월 5 21:07 service_rw_29.sql.log
-rw-r--r-- 1 cubrid1 cubrid 16222670 2월 5 21:07 service_rw_189.sql.log
-rw-r--r-- 1 cubrid1 cubrid 57447980 2월 5 21:07 service_rw_2.sql.log
-rw-r--r-- 1 cubrid1 cubrid 55847453 2월 5 21:07 service_rw_7.sql.log
-rw-r--r-- 1 cubrid1 cubrid 84189229 2월 5 21:07 service_rw_9.sql.log
-rw-r--r-- 1 cubrid1 cubrid 97286994 2월 5 21:07 service_rw_10.sql.log
-rw-r--r-- 1 cubrid1 cubrid 40231273 2월 5 21:07 service_rw_11.sql.log
-rw-r--r-- 1 cubrid1 cubrid 90405802 2월 5 21:07 service_rw_19.sql.log
-rw-r--r-- 1 cubrid1 cubrid 61341248 2월 5 21:07 service_rw_48.sql.log
-rw-r--r-- 1 cubrid1 cubrid 65944601 2월 5 21:07 service_rw_53.sql.log
-rw-r--r-- 1 cubrid1 cubrid 16395860 2월 5 21:07 service_rw_15.sql.log
-rw-r--r-- 1 cubrid1 cubrid 25352777 2월 5 21:07 service_rw_23.sql.log
-rw-r--r-- 1 cubrid1 cubrid 84783591 2월 5 21:07 service_rw_3.sql.log
-rw-r--r-- 1 cubrid1 cubrid 92173241 2월 5 21:07 service_rw_44.sql.log
-rw-r--r-- 1 cubrid1 cubrid 4711939 2월 5 21:07 service_rw_57.sql.log
-rw-r--r-- 1 cubrid1 cubrid 71096577 2월 5 21:07 service_rw_145.sql.log
[cubrid1@ifsc-dev2-wa801] 2월 05 21:07:20 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:23 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:24 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:24 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:24 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:24 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:24 ~/CUBRID/log/broker/sql_log/service_rw.js
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:25 ~/CUBRID/log/broker/sql_log/service_rw.js cd ..
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:27 ~/CUBRID/log/broker/sql_log$ cd
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:27 ~$ ls
CUBRID CUBRID_R9.2.24.0002 DB DB_log bin csq1.err cunitor script scripts src work
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:27 ~$ ls -ltr
합계 2052
drwx----- 2 cubrid1 cubrid 6 2월 23 2022 cunitor
lrwxrwxrwx 1 cubrid1 cubrid 19 2월 23 2022 CUBRID -> CUBRID_R9.2.24.0002
drwx----- 2 cubrid1 cubrid 48 2월 23 2022 src
lrwxrwxrwx 1 cubrid1 cubrid 16 3월 30 2022 DB -> /data/cubrid1/DB
lrwxrwxrwx 1 cubrid1 cubrid 20 3월 30 2022 DB_log -> /data/cubrid1/DB_log
drwxr-xr-x 3 cubrid1 cubrid 26 3월 30 2022 script
drwx----- 18 cubrid1 cubrid 253 3월 30 2022 CUBRID_R9.2.24.0002
drwx----- 7 cubrid1 cubrid 121 4월 28 2022 work
drwx----- 3 cubrid1 cubrid 88 8월 24 12:17 bin
drwx----- 2 cubrid1 cubrid 85 1월 10 16:21 scripts
-rw-r--r-- 1 cubrid1 cubrid 2096767 2월 5 10:00 csq1.err
[cubrid1@ifsc-dev2-wa801] 2월 05 21:08:28 ~$ cd
```

```
shell - cubrid1@ngdcdb-dbs901:/home1/cubrid1/CUBRID/log - zsh - 138x60

~/shell - centos@dogres-1:~ - zsh ... ~/shell - centos@dogres-1:~ - zsh ... .dbs901:/home1/cubrid1/CUBRID/log - zsh +

HA-Process Info (master 2672, state master)
Applylogdb ngrinder@localhost:/home1/cubrid1/DB/nginder_ngdcdb-dbs902 (pid 4896, state registered)
Copylogdb ngrinder@ngdcdb-dbs902:/home1/cubrid1/DB/nginder_ngdcdb-dbs902 (pid 19539, state registered)
Server ngrinder (pid 9421, state registered and active)

[cubrid1@ngdcdb-dbs901] 1월 06 10:10:34 ~/CUBRID/conf$ c
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:22 ~/CUBRID/conf$ cd ..
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:26 ~/CUBRID$ cd log
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:27 ~/CUBRID/log$ ls
broker ngrinder@localhost_acldb.err ngrinder@localhost_spacedb.err ngrinde
r_addvoldb.err ngrinder@localhost_applylogdb_ngrinder_ngdc-cdb-t1901.err ngrinder@localhost_spacedb.err.bak ngrinde
r_applyinfo.err ngrinder@localhost_applylogdb_ngrinder_ngdc-cdb-t1902.err ngrinder@localhost_statdump.err ngrinde
demodb_createdb.err ngrinder@localhost_applylogdb_ngrinder_ngdc-cdb-t1902.err ngrinder@localhost_statdump.err
r_createdb.err ngrinder@localhost_applylogdb_ngrinder_ngdcdb-dbs902.err ngrinder@localhost_statdump.err.bak server
demodb_loaddb.err ngrinder@localhost_applylogdb_ngrinder_ngdcdb-dbs902.err.bak ngrinder@ngdc-cdb-t1901_copylogdb.err
manager ngrinder@localhost_acldb.err ngrinder@localhost_applylogdb_ngrinder_ngdcdb-dbs902.err.bak ngrinder@ngdc-cdb-t1902_copylogdb.err
ngdcdb-dbs901_master.err ngrinder@localhost_changemode.err.bak ngrinder@ngdcdb-dbs902_copylogdb.err
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:28 ~/CUBRID/log$ ls -ltr
합계 646548
drwx----- 2 cubrid1 cubrid 4096 2015-03-12 10:22 manager
-rwx----- 1 cubrid1 cubrid 0 2015-10-13 14:31 demodb_createdb.err
-rwx----- 1 cubrid1 cubrid 0 2015-10-13 14:31 demodb_loaddb.err
-rw-r--r-- 1 cubrid1 cubrid 23199 2015-10-13 14:35 ngrinder_createdb.err
-rw-r--r-- 1 cubrid1 cubrid 1731 2015-10-13 14:36 ngrinder_addvoldb.err
-rw-r--r-- 1 cubrid1 cubrid 30484 2015-10-27 23:11 ngrinder_applyinfo.err
drwxr-xr-x 5 cubrid1 cubrid 4096 2016-04-25 17:58 broker
-rw-r--r-- 1 cubrid1 cubrid 17042 2017-05-21 11:13 cub_client.err
-rw-r--r-- 1 cubrid1 cubrid 100000291 2022-07-26 22:00 ngrinder@localhost_statdump.err.bak
-rw-r--r-- 1 cubrid1 cubrid 100000058 2022-10-13 10:07 ngrinder@localhost_spacedb.err.bak
-rw-r--r-- 1 cubrid1 cubrid 100000272 2022-12-04 15:40 ngrinder@localhost_changemode.err.bak
-rw-r--r-- 1 cubrid1 cubrid 449 2022-12-16 16:30 ngrinder@localhost_acldb.err
-rw-r--r-- 1 cubrid1 cubrid 100000276 2022-12-29 15:59 ngrinder@localhost_applylogdb_ngrinder_ngdcdb-dbs902.err.bak
drwxr-xr-x 2 cubrid1 cubrid 4096 2023-01-05 22:14 server
-rw-r--r-- 1 cubrid1 cubrid 794473 2023-01-06 01:00 ngrinder@localhost_acldb.err
-rw-r--r-- 1 cubrid1 cubrid 1569286 2023-01-06 03:01 ngrinder@ngdcdb-dbs902_copylogdb.err
-rw-r--r-- 1 cubrid1 cubrid 73167893 2023-01-06 10:10 ngrinder@localhost_spacedb.err
-rw-r--r-- 1 cubrid1 cubrid 70540590 2023-01-06 10:10 ngrinder@localhost_statdump.err
-rw-r--r-- 1 cubrid1 cubrid 20758728 2023-01-06 10:10 ngdcdb-dbs901_master.err
-rw-r--r-- 1 cubrid1 cubrid 703 2023-01-06 10:10 ngrinder@ngdc-cdb-t1901_copylogdb.err
-rw-r--r-- 1 cubrid1 cubrid 3106 2023-01-06 10:11 ngrinder@ngdc-cdb-t1902_copylogdb.err
-rw-r--r-- 1 cubrid1 cubrid 1520 2023-01-06 10:11 ngrinder@localhost_applylogdb_ngrinder_ngdc-cdb-t1901.err
-rw-r--r-- 1 cubrid1 cubrid 5817 2023-01-06 10:11 ngrinder@localhost_applylogdb_ngrinder_ngdc-cdb-t1902.err
-rw-r--r-- 1 cubrid1 cubrid 9431860 2023-01-06 10:11 ngrinder@localhost_applylogdb_ngrinder_ngdcdb-dbs902.err
-rw-r--r-- 1 cubrid1 cubrid 85419663 2023-01-06 10:11 ngrinder@localhost_changemode.err
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:28 ~/CUBRID/log$ vi ngdcdb-dbs901_master.err
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:42 ~/CUBRID/log$ free -m
total used free shared buffers cached
Mem: 7872 7708 164 3 270 4666
-/+ buffers/cache: 2771 5100
Swap: 2047 0 2047
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:46 ~/CUBRID/log$ vi ngdcdb-dbs901_master.err C
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:49 ~/CUBRID/log$ ^C
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:49 ~/CUBRID/log$ ^C
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:49 ~/CUBRID/log$ ^C
[cubrid1@ngdcdb-dbs901] 1월 06 10:11:49 ~/CUBRID/log$ client_loop: send disconnect: Broken pipe
DOG@AL01600590 shell %
```

서비스 계정을 세분화

- SERVICE_RW : DML을 하는 메인 서비스 계정
- SERVICE_READ : 조회 전용 서비스 계정
- ETLUSER : 데이터 추출 전용 계정(분석용)
- API_MEMBER : 특정 서비스의 API 계정
- ADMIN : 관리자 계정

내 서비스를 사용해 보자

- 내 DB는 어떤 역할을 하고 있나
- 데이터의 흐름에 호기심을 가져보자
- 서비스를 이해하면 커뮤니케이션에 도움

가장 민감한 구간의 관리자로서의 책임감

- DBA의 가장 큰 덕목은 도덕
- 화려하지 않은 자리
- DB는 서비스의 시작
- 체력

Cloud

**유연하게, 안전하게
비즈니스에 힘이 되다.**

