



**이기종 데이터베이스 환경 구축을 위한
능동적인 데이터 이관, 통합, 분산 전략**

Quest®

Where Next Meets Now.

Agenda

1. 데이터베이스 시장 동향과 하이브리드 클라우드
2. 실시간 데이터베이스 동기화를 통한 데이터 마이그레이션
3. SharePlex 기타 활용
4. SharePlex 레퍼런스



데이터베이스 시장 동향과 하이브리드 클라우드

Database Trends Ranking

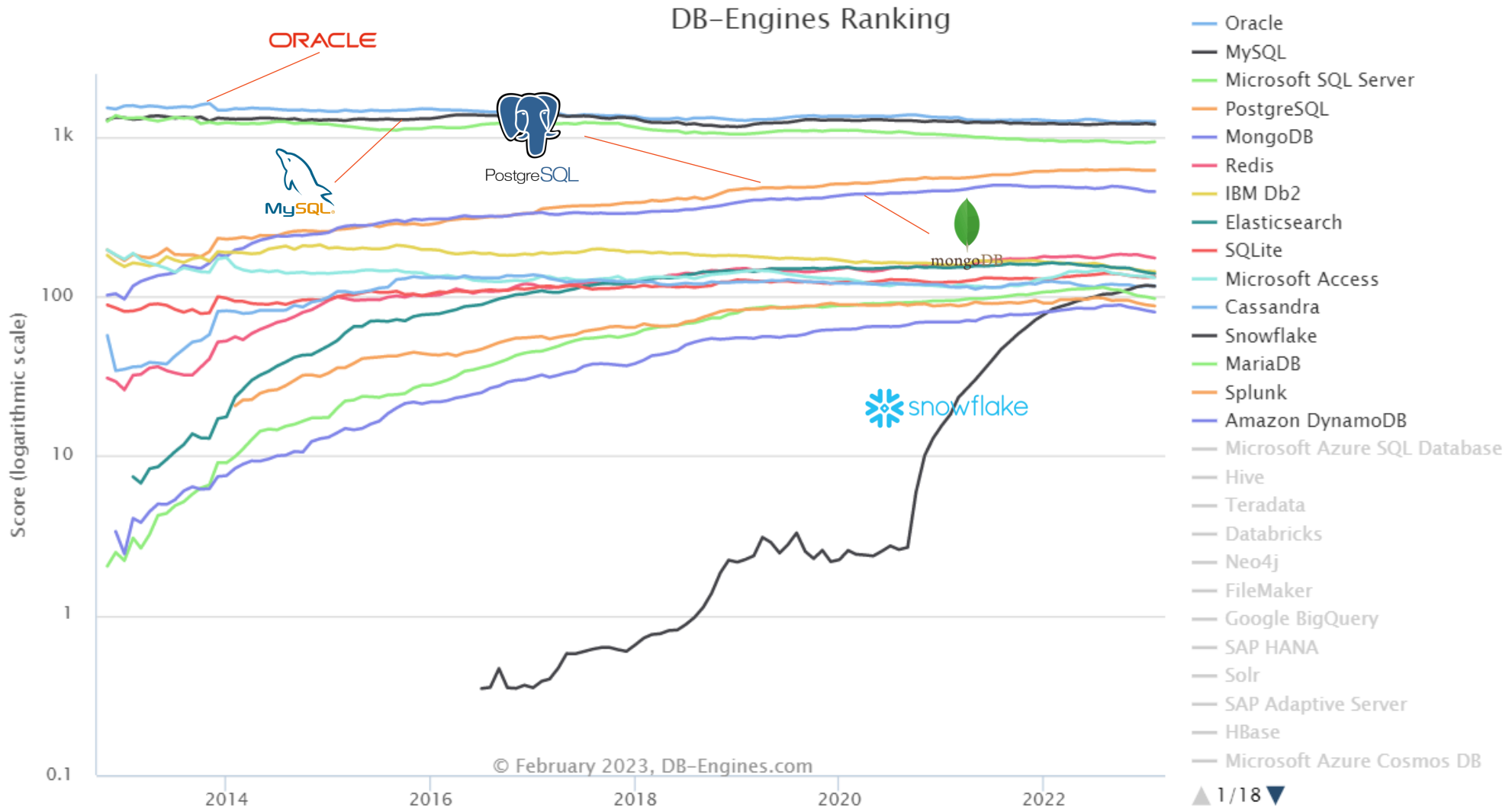
데이터베이스 시장 동향과 하이브리드 클라우드

410 systems in ranking, February 2023

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle +	Relational, Multi-model <i>i</i>	1247.52	+2.35	-9.31
2.	2.	2.	MySQL +	Relational, Multi-model <i>i</i>	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model <i>i</i>	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL +	Relational, Multi-model <i>i</i>	616.50	+1.65	+7.12
5.	5.	5.	MongoDB +	Document, Multi-model <i>i</i>	452.77	-2.42	-35.88
6.	6.	6.	Redis +	Key-value, Multi-model <i>i</i>	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model <i>i</i>	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model <i>i</i>	138.60	-2.56	-23.70
9.	↑ 10.	↑ 10.	SQLite +	Relational	132.67	+1.17	+4.30
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	131.03	-2.33	-0.23

<출처 : db-engines.com>

Database Trends Ranking



Public Cloud

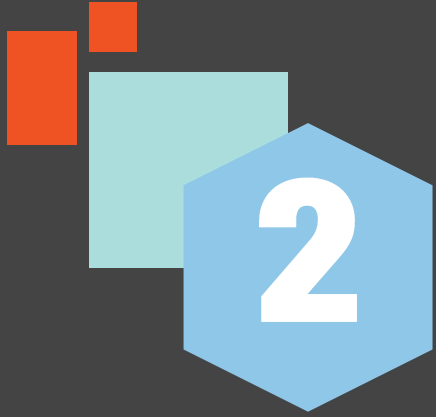
- 서비스를 벤더에서 제공
- Public Internet을 통해 누구나 사용
- 빠르고 편리하게 확장 가능

Hybrid Cloud

- Public Cloud와 Private Cloud를 결합한 형태
- Public Cloud와 Private Cloud의 장점을 모두 적용
- 중요한 데이터 및 프로세스를 보다 엄격하게 제어할 수 있도록 지원

Private Cloud

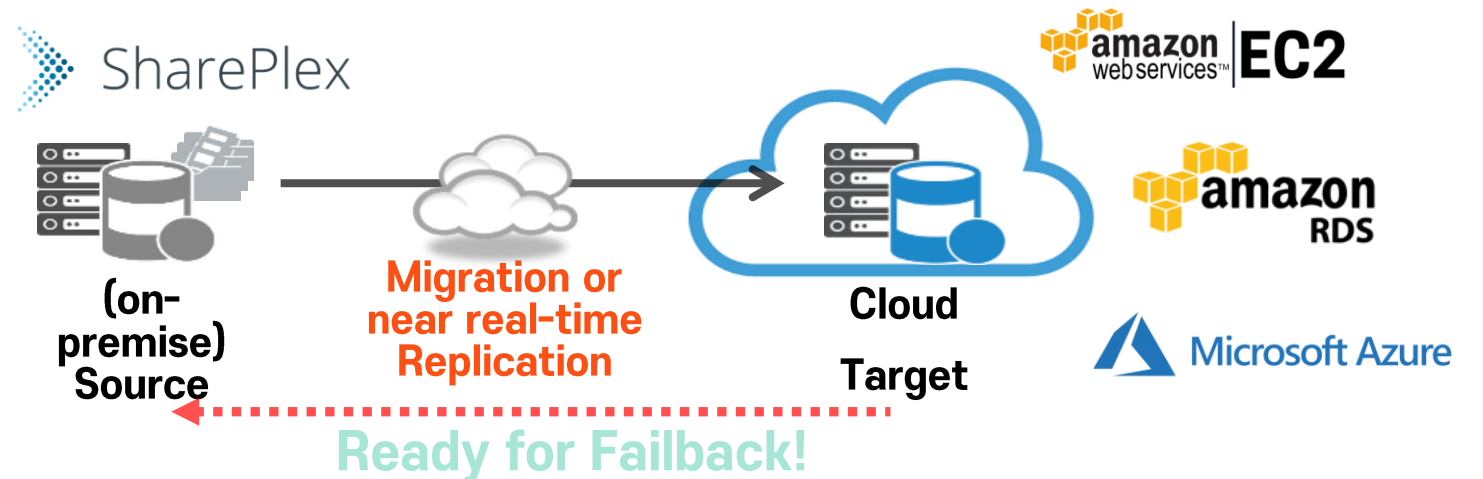
- 내부망을 통해 사용자가 직접 서비스를 운영
- 강력한 보안 제어
- 자체적인 데이터 센터 및 유지 관리 필요



실시간 데이터베이스 동기화를 통한 마이그레이션

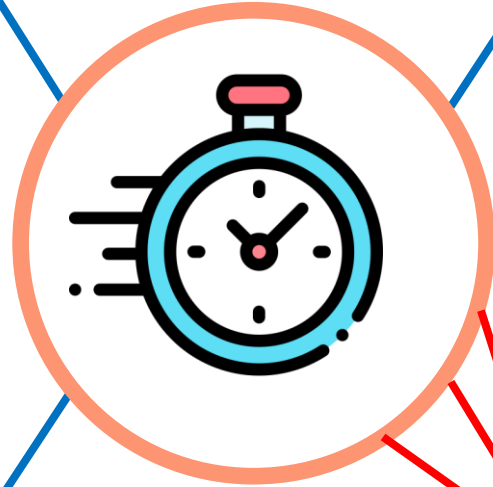


- ✓ 마이그레이션 이후 예상치 못한 심각한 장애 발생 대비
- ✓ 전환 된 이후의 변경 된 데이터 고려
- ✓ SharePlex 등 CDC 솔루션을 이용한 계획 수립



예상하지 못한 다운타임의 지속
→ 서비스 장애로 이어짐

어플리케이션의 허용가능
다운 타임 확인



사전에 충분한 테스트를 통해
다운타임 예측

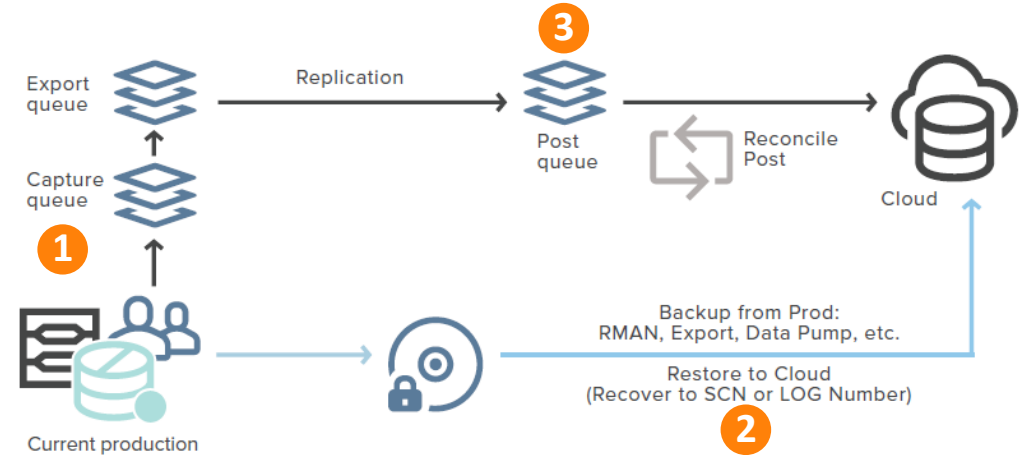
서비스 다운 타임을 최소화 하는 방법?



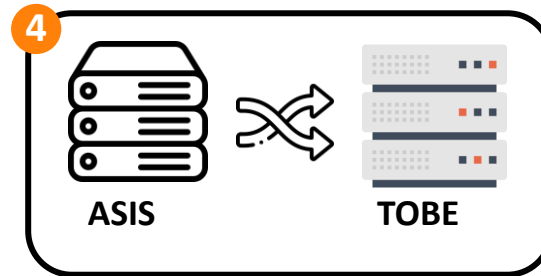
SharePlex를 이용한 마이그레이션

마이그레이션 절차

- 1) 기존 운영 DB에서 변경 분 Capture 시작
- 2) 신규 운영 DB에 데이터를 초기 적재 (RMAN, Exp/Imp, Etc)
- 3) 초기 적재 이후 변경 데이터를 동기화 (SharePlex)
- 4) 신규 운영 DB로 서비스 전환 (Network, Application)
- 5) 신규 운영 DB 에서 서비스 모니터링



On-Premise to Cloud Migration



시스템 자원 사용을 최소화

- DBMS에 부하를 주는 Log Miner를 유틸리티를 사용하지 않고 Redo log에 Direct Access
- Queue mechanism을 사용하여 Disk IO 최소화
- Commit 전 데이터를 복제, 변경 즉시 Target으로 전송하여 Target System에서의 자원 사용률에 대한 부담을 최소화

구분	SharePlex	타사
복제 데이터 보관/전송	<ul style="list-style-type: none">▪ Queue - 메모리(성능) + 디스크(안정성)	<ul style="list-style-type: none">▪ 디스크 보관/전송
데이터 복제 처리 방식	<ul style="list-style-type: none">▪ Commit 전 데이터 복제<ul style="list-style-type: none">✓ 변경 즉시 전송/반영하여 소스/타겟 시스템 자원 부담 최소화	<ul style="list-style-type: none">▪ Commit 후 데이터 복제<ul style="list-style-type: none">✓ Commit 완료까지 소스시스템에 적재하여 Disk I/O, CPU 자원 사용 증가✓ 타겟 시스템에도 동일 데이터 복제량을 보유✓ 한꺼번에 전송함으로써 Network 부하 유발

데이터 무결성 보장

- 실시간으로 데이터 무결성을 검증하기 위해 Target에 반영하기 전, 복제 데이터의 정합성을 판단
- Before/After 이미지 관리를 통한 In-Flight(실시간) 데이터 무결성 검증 (parameter 통한 on/off 가능)

Primary	Update OK	Secondary	Update OK												
<table border="1"><thead><tr><th>Id</th><th>Name</th><th>Address</th></tr></thead><tbody><tr><td>1</td><td>John</td><td>22 Second St</td></tr></tbody></table>	Id	Name	Address	1	John	22 Second St		<table border="1"><thead><tr><th>Id</th><th>Name</th><th>Address</th></tr></thead><tbody><tr><td>1</td><td>John</td><td>11 First St</td></tr></tbody></table>	Id	Name	Address	1	John	11 First St	
Id	Name	Address													
1	John	22 Second St													
Id	Name	Address													
1	John	11 First St													


```
UPDATE customer
  SET address = '22 Second St'
  WHERE id = 1;
COMMIT;
```

↓ Write to Redo

```
UPDATE customer
  SET address = '22 Second St'
  WHERE id = 1
  AND address = '11 First St'
  AND rownum = 1;
COMMIT;
```

↑ SQL to Post

<ROWID>:<OBJ_ID>:<COL_ID>:22 Second St:11 First St:1

UPDATE OOS Example

Primary	Update OK	Secondary	Out-Of-Sync												
<table border="1"><thead><tr><th>Id</th><th>Name</th><th>Address</th></tr></thead><tbody><tr><td>1</td><td>John</td><td>22 Second St</td></tr></tbody></table>	Id	Name	Address	1	John	22 Second St		<table border="1"><thead><tr><th>Id</th><th>Name</th><th>Address</th></tr></thead><tbody><tr><td>1</td><td>John</td><td>33 Third St</td></tr></tbody></table>	Id	Name	Address	1	John	33 Third St	
Id	Name	Address													
1	John	22 Second St													
Id	Name	Address													
1	John	33 Third St													


```
UPDATE customer
  SET address = '22 Second St'
  WHERE id = 1;
COMMIT;
```

↓ Write to Redo

```
UPDATE customer
  SET address = '22 Second St'
  WHERE id = 1
  AND address = '11 First St'
  AND rownum = 1;
COMMIT;
```

↑ SQL to Post

Record Not Found

<ROWID>:<OBJ_ID>:<COL_ID>:22 Second St:11 First St:1

다양한 복제 단위

- **TABLE, COLUMN, ROW, SCHEMA** 단위의 복제 대상 지정
- SCHEMA 단위 복제 설정 시, 새로운 TABLE을 생성/변경/삭제 할 경우 자동으로 대상 업데이트
- DDL 복제를 Parameter를 통해 명시적으로 제어 가능

Object	DDL Operation	SharePlex Parameter
Table and IOT	Create, Alter, Drop, Truncate, Comment, CTAS	SP_OCT_REPLICATE_DDL=3 SP_OCT_AUTOADD_ENABLE=1
Sequence	Create, Alter, Drop	SP_OCT_AUTOADD_SEQ=1 SP_OCT_AUTOADD_ENABLE=1 SP_SYS_TARGET_COMPATIBILITY=8.6.2 or higher
Partition/Sub-partition	Add, Split, Merge, Drop, Modify, Coalesce, Exchange, Move, Truncate, Rename, Set	SP_OCT_REPLICATE_DDL=3 SP_OCT_AUTOADD_ENABLE=1
Index	Create, Alter, Drop	SP_OCT_REPLICATE_DDL=3 SP_OCT_AUTOADD_ENABLE=1
View	Create, Alter, Drop, Comment	SP_OCT_REPLICATE_ALL_DDL=1
Stored procedure	Create, Alter, Drop	SP_OCT_REPLICATE_ALL_DDL=1

Compare(검증)/Repair(보정)

- 다양한 이유로 정합성이 맞지 않게 될 경우 'event log' 파일에 해당 내용이 logging
- Online으로 SharePlex에 내장 된 Compare/Repair Utility를 통하여 정합성 검증 및 보정 기능 제공

Insert OOS Example in sprod_errlog.sql

```
-- [3] Tue Oct 9 14:26:38 2011
-- redolog seq#/offset 81319/159081124
-- redolog timestamp 635305391 (10/07/11)
-- original rowid AAakIvAGQAAAPfJABc
-- ORA-00001: unique constraint
(OSI_SIEBEL2.S_ORDER_ITEM_XA_P1) violated
-- CONSTRAINT VIOLATION
insert into "OSI_SIEBEL2"."S_ORDER_ITEM_XA_P1"
("ROW_ID", "CREATED", "CREATED_BY", "LAST_UPDATE_DATE",
"MODIFICATION_NUM", "CONFLICT_ID", "ATTR_NAME", "ORDER_ITEM_ID",
"READ_ONLY_FLG", "REQUIRED_FLG", "ATTR_ID", "CHAR_VAL", "DATA_TYPE_CD", "DIS
"VLDTN_LOV_TYPE_CD")
values ('1-43RHNT8', '10072011014311', '1-10072007014311', '1-T8E-750', '0', '0', 'La
'N', '1-43RHNT7', 'N', 'Y', 'Update', '1-08DV
'Thai', 'Text', 'Language', 'AIS_VMS_LANGUA
```

√ Compare 결과 확인

```
Host: SPSRC
ID      Tablename      Since      Repair
-----
6208 "SPXUSER"."TB_CUST" 03-Mar-11 11:28:30 Off

Source table : "SPXUSER"."TB_CUST"
Source node  : SPSRC
Target table  : "SPXUSER"."TB_CUST"
Target route : SPTGT@ORCL9
Where clause  :
Select hint  :
Status       : Out Sync   데이터 비교 결과 18,000 Row 가 다름
                ( Target 에 INSERT 18,000 row 가 필요함 )

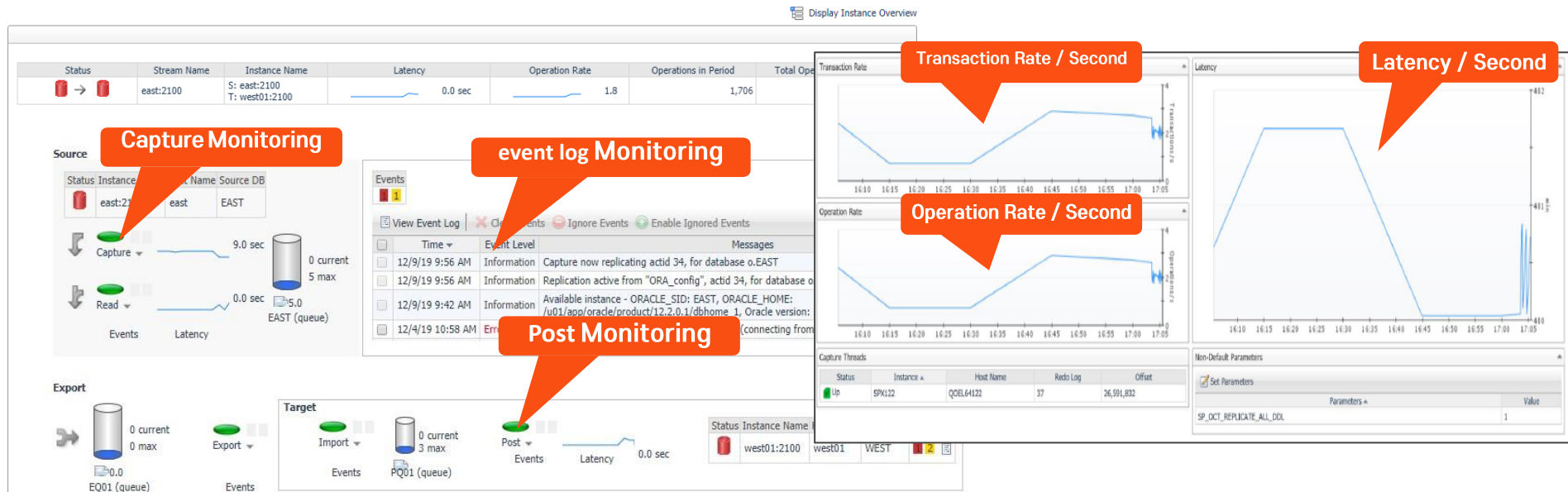
* Compare Results
* 18000 source and 0 target rows compared successfully (L387).
* 18000 out-of-sync row(s) found in this table.
* Inserts : 18000
* Updates : 0
* Deletes : 0
*/
```

√ Compare 결과 Repair 를 위한 SQL 문 확인

```
/*
* Compare Report
* Session ID : 6208
* Schema    : SPXUSER
* Table     : TB_CUST
* Repair    : Off
* Target Route: SPTGT@ORCL9
* Key Compare : Off
* Select Hint :
* Log File  : /splex75/var/log/ORCL9_SPXUSER-TB_CUST-6208-20353.sql
* Date     : Thu Mar 3 11:28:35 2011
*/
데이터 비교 결과 INSERT 18,000 Row SQL문
/* source rowid='AAAH0VAALAAAAPnAAA' */
insert into "SPXUSER"."TB_CUST" ("CUST_NO", "NAME", "ADDRESS", "IN_DATE", "IN_TIME")
values ('3721', 'Cust Name_1', 'Cust Address_1', '20110215', '135616 ');
/* source rowid='AAAH0VAALAAAAPnAAB' */
insert into "SPXUSER"."TB_CUST" ("CUST_NO", "NAME", "ADDRESS", "IN_DATE", "IN_TIME")
values ('3723', 'Cust Name_1', 'Cust Address_1', '20110215', '135616 ');
```

Monitoring

- 대시보드 화면을 통해 프로세스 상태, 처리 건수, 지연 시간 등을 쉽게 확인
- 장애 발생 시 이벤트 창을 통해 즉시 확인 가능



장애 상황 복구

- SharePlex Process failed

- 사용자 실수에 의한 process kill이나 강제 종료의 이유로 데이터 동기화가 중단되는 상황 발생 시, **자동으로 process가 기동 시도** 되어 정상적인 복제 시작

- DB failed

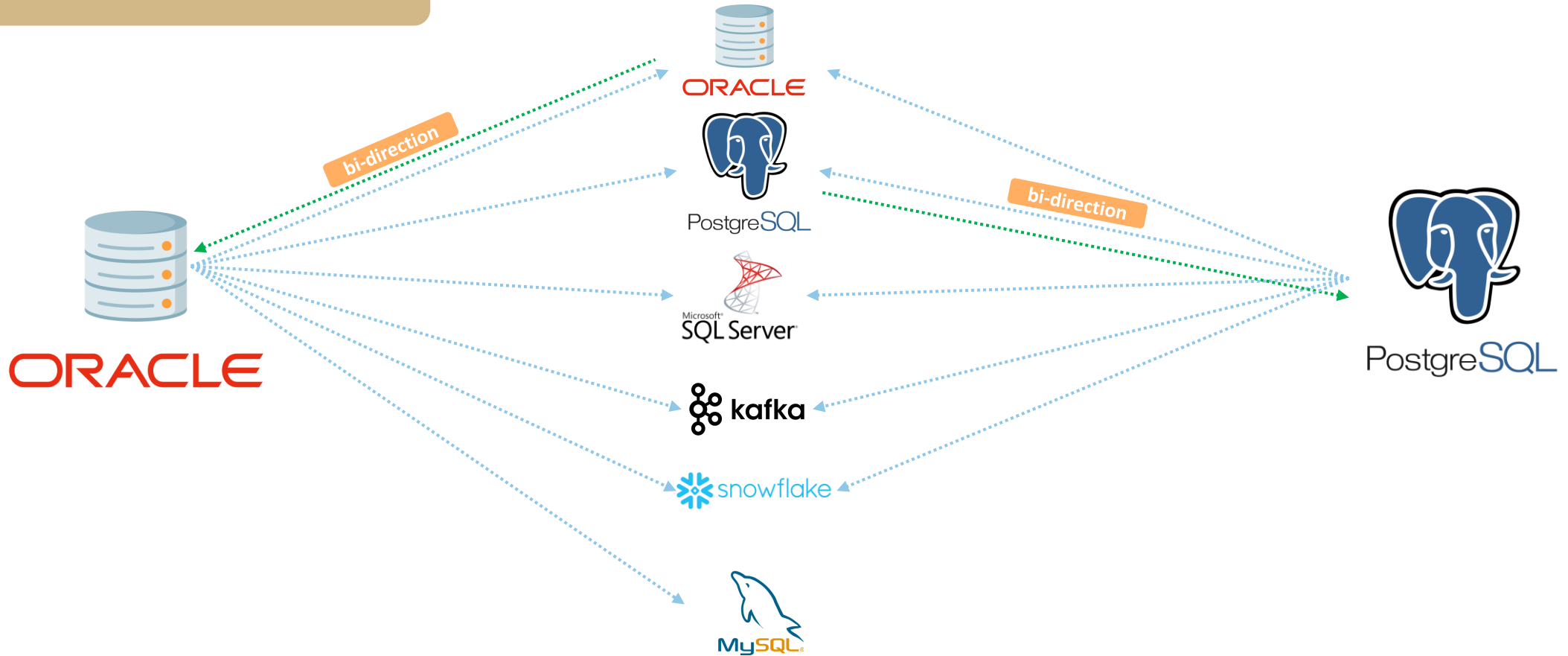
- Source DB fail : Redo log로 부터 추출한 데이터까지의 Offset 정보 관리를 통한 **데이터 연속성 자동 유지**
- Target DB fail : DB에 적용 된 데이터에 대한 메타 정보를 확인하여 **장애시점부터의 연속적 복제 Data 반영**

- Network failed

- Network **정상 여부 자동 체크** 및 복구를 통한 **복제 연속성 유지**
- Network 장애에 대비하여 Queue가 보관되는 **Disk 공간에 대한 적절한 산정 필요**

SharePlex

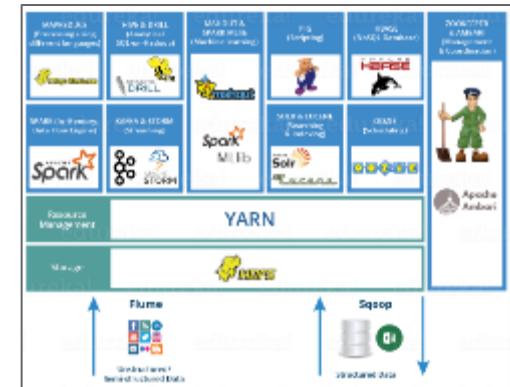
이기종 데이터베이스 복제 지원



SharePlex for Kafka 는 빅데이터 연계 Hub로서 Kafka에 대한 실시간 데이터 복제를 지원합니다. 분산 스트리밍 플랫폼인 Kafka를 통해서 다양한 빅데이터 시스템에 변경 데이터를 전달할 수 있습니다.



빅데이터 Ecosystem

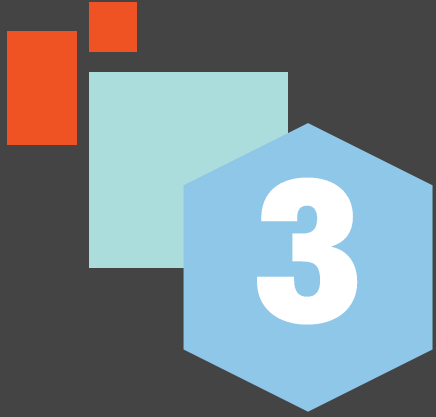


- **SharePlex for Kafka**

- 빅데이터 연계 Hub로서 Kafka에 대한 복제 지원
- SharePlex 8.6.6 이후 버전부터 Kafka 를 지원 (JSON format 은 9.0 이후 버전)
- 메시지 포맷: XML / JSON
- Kafka broker clustering 과 multiple partitions / topics 을 모두 지원
- Only add column, drop column and truncate are supported`

지원하는 데이터베이스 및 플랫폼

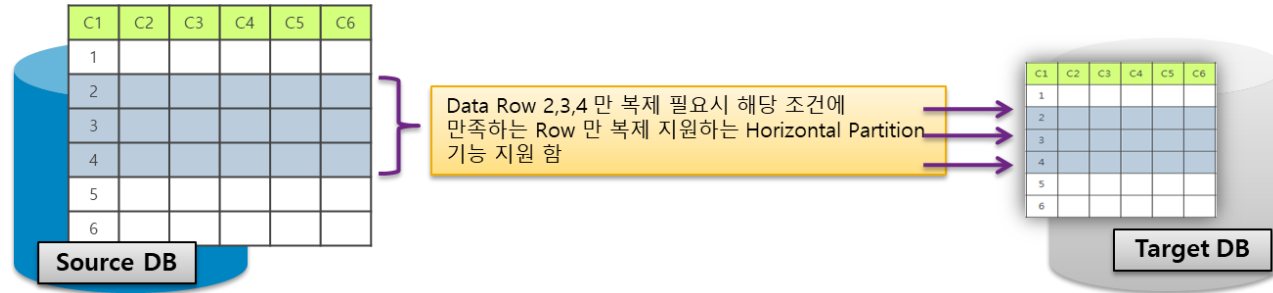
DBMS	Version	Platform (only 64-bit)
Oracle (Source, Target)	11gR2, 12cR1, 12cR2, 18c, 19c(for Linux 7 later) EE & SE2	AIX 7.1, 7.2 CentOS Linux 5.x, 6.x, 7.x, 8.x Oracle Linux 5.x, 6.x, 7.x, 8.x RHEL 5.x, 6.x, 7.x, 8.x SuSE SLES 11.x, 12.x, 15.x Oracle Solaris(SPARC and x86) 10, 11, 11.4 Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019 HP-UX 11.31 Itanium IaaS Azure and AWS
MS SQL Server (Target)	SQL Server 2008, 2008 R2, 2012, 2014, 2016, 2017, and 2019	Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, and 2019
Kafka (Target)	0.8.x, 0.9.x, 0.10.x, 0.11.x, 1.0.x, 1.1.x, 2.0.x, 2.1.0, 2.3.1, 2.5.0, 3.0 (Supported File Types XML file, JSON file)	RHEL 7.x, 8.x
MySQL (Target)	5.5, 5.6, 5.7, and 8	RHEL 7.x, 8.x IaaS Azure and AWS
PostgreSQL (Source, Target)	PostgreSQL 9.4, 9.5, 9.6, 10, 13.4, 13.6, 14.1	# Source AIX 7.1, 7.2 RHEL, CentOS, Oracle Linux 7.x, 8.x, SuSe SLES 12.x and 15.x HP-UX 11.31 Itanium # Target RHEL, CentOS, Oracle Linux 7.x, 8.x Oracle Solaris 11 and 11.4 SuSe SLES 12.x and 15.x Windows Server 2016, 2019



SharePlex 기타 활용

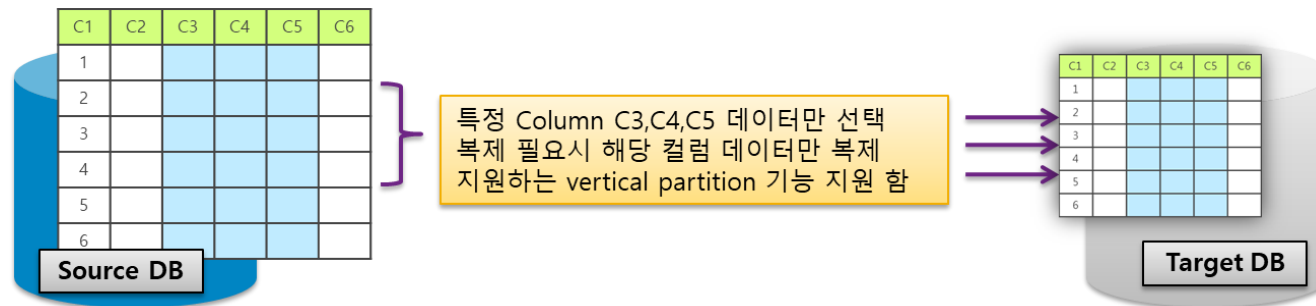
✓ Row 단위의 복제 (Horizontally partitioned replication)

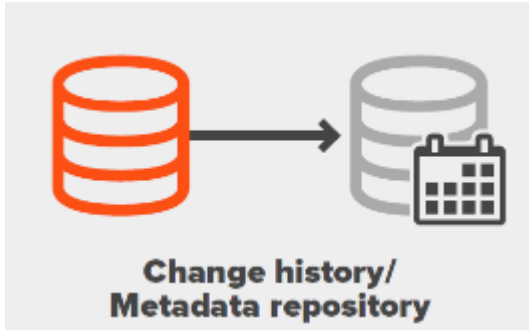
- 테이블의 복제 대상이 되는 조건(where)을 추가하여 조건에 해당하는 Row (C1=2,3,4)만 복제



✓ Column 단위의 복제 (Vertically Partitioned replication)

- 특정 Column (C3,C4,C5) 의 데이터만 복제하는 vertical partition 기능 지원



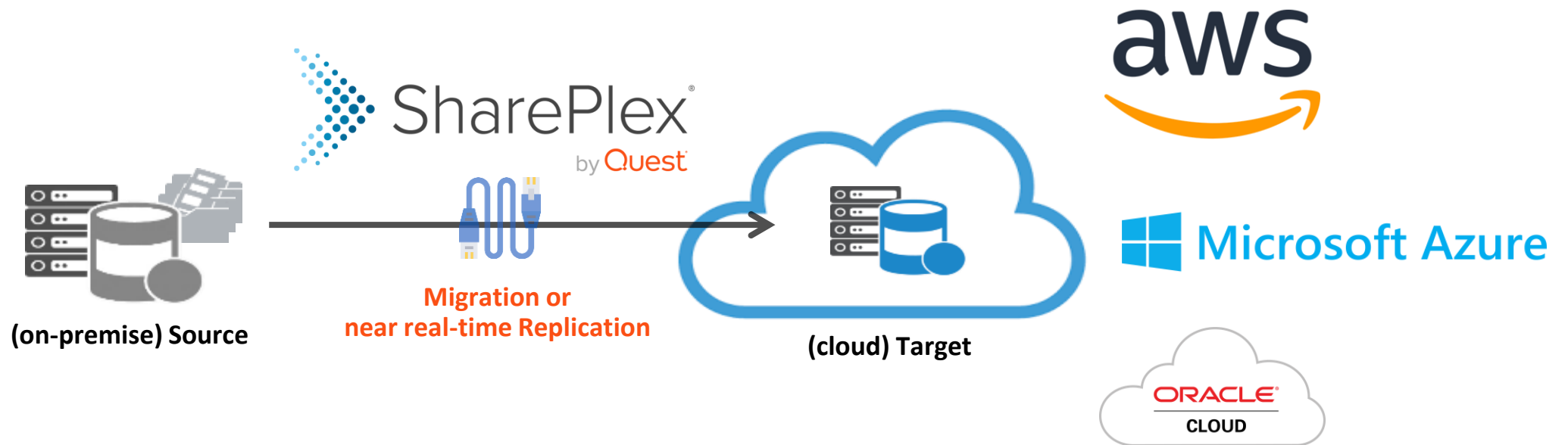


```
SHAREPLEX_SOURCE_TIME  
SHAREPLEX_SOURCE_USERID  
SHAREPLEX_SOURCE_OPERATION  
SHAREPLEX_SOURCE_SCN  
SHAREPLEX_SOURCE_ROWID  
SHAREPLEX_SOURCE_TRANS  
SHAREPLEX_OPERATION_SEQ  
SHAREPLEX_SOURCE_HOST  
SHAREPLEX_QUEUE_NAME  
SHAREPLEX_SOURCE_ID
```

- **Change History**
 - 변경이력 추적 및 감사 목적의 데이터 수집 (For auditing and discovery purposes)
 - 공.금융기관에서 활용 중 (국내/해외 포함)
- **Source DBMS 의 Metadata**
 - user id, time, operation type, scn, row id, transaction id 등
- **Operations supported**
 - INSERT/UPDATE/DELETE/TRUNCATE/DROP COLUMN

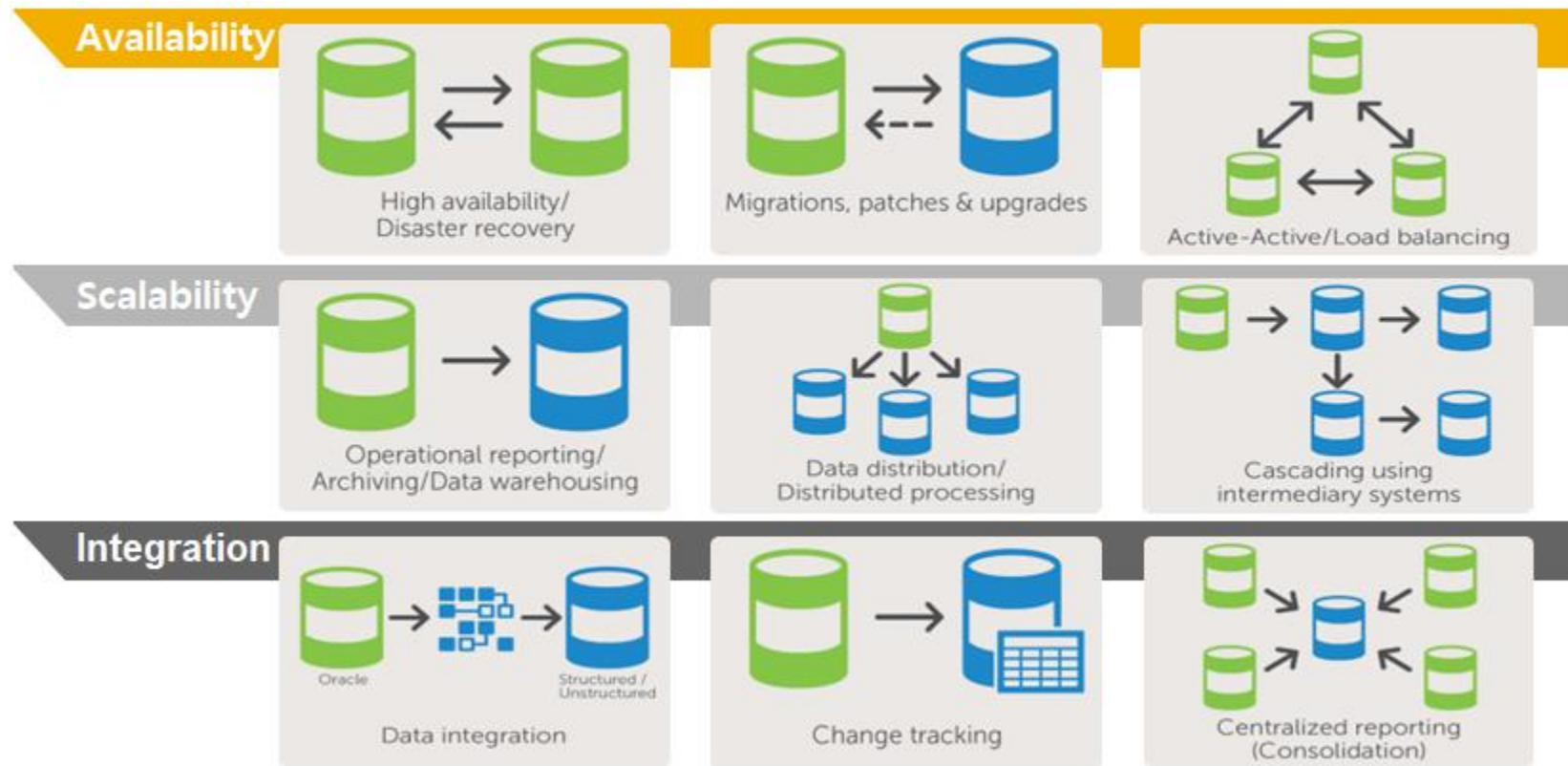
Oracle on Cloud

- ✓ On-premise 에서 **AWS 및 Azure, Oracle 클라우드 플랫폼**으로의 복제를 지원
- ✓ Cloud 로 마이그레이션 수행 시 Data Migration 과 실시간 동기화로 **다운타임 최소화**
- ✓ On-premise to Cloud, Cloud to Cloud, Hybrid Cloud (Public + Private) 등 다양한 복제 환경의 실시간 데이터 동기화
- ✓ 민감한 데이터를 보호, Selective Replication (선택적 복제)



다양한 환경에서의 실시간 데이터 복제

SharePlex는 1:N N:1 N:M 등의 유연한 복제 구조를 가지고 있어 ODS, 마이그레이션, 리포팅, HA/DR 등 고객의 모든 요구사항을 만족할 수 있는 다양한 복제 구성을 제공합니다.





SharePlex 레퍼런스

SharePlex - 국내 (금융) 고객사들

SharePlex 사례

- ✓ SharePlex 는 CDC 분야 No.1 솔루션으로 국내에 가장 많은 금융 고객사에서 사용 중이며, ODS 구축과 마이그레이션, HA/DR 등 다양한 복제 및 구축 노하우를 보유하고 있습니다.



SharePlex - 국내 (공공기관, 제조) 고객사들

- ✓ SharePlex 는 공공기관과 제조 등 다양한 업종에서 중요한 업무의 실시간 데이터 연계와 중앙 집중화된 복제 구성, 그리고 DR 등 다양한 용도로 사용되고 있습니다.



SharePlex - 국내 (공공기관, 제조) 고객사들

SharePlex 사례

- ✓ SharePlex 는 공공기관과 제조 등 다양한 업종에서 중요한 업무의 실시간 데이터 연계와 중앙 집중화된 복제 구성, 그리고 DR 등 다양한 용도로 사용되고 있습니다.



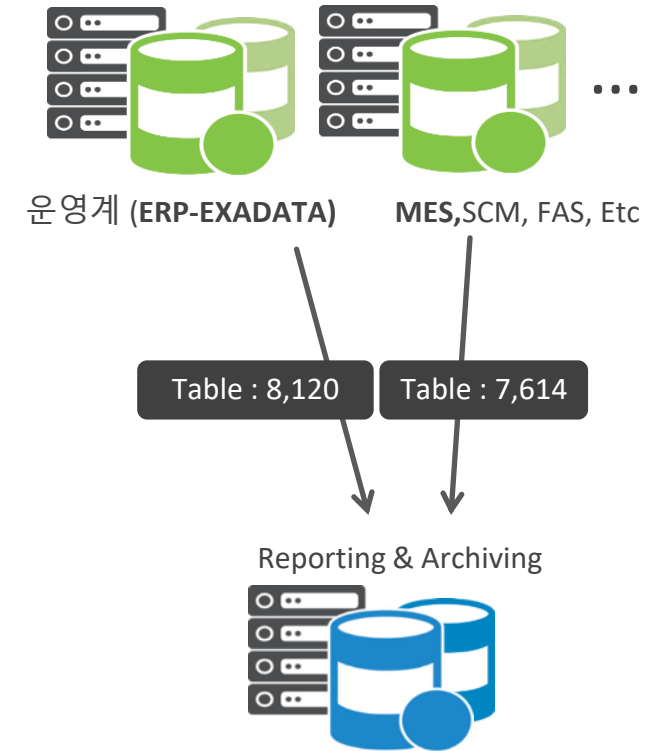
제조사 데이터 통합 사례(P사)

✓ 도입 배경/효과

- **운영서버** : 조회성 업무를 **리포팅 서버로 분산**함으로써 **ERP 는 안정적인 업무 수행**
- **리포팅 서버**: 운영 서버 대신 **Report용으로 활용** 및 **Archiving** 형태로 운영 데이터 보다 **장기간의 정보를 보유**
- ERP에서 제공하는 대량의 월단위 매출/결산 정보의 실시간 처리 및 분석 보고서를 **리포팅 서버에서 제공**

✓ 복제구성의 특이사항

- **Reporting (11:1) & Archiving**



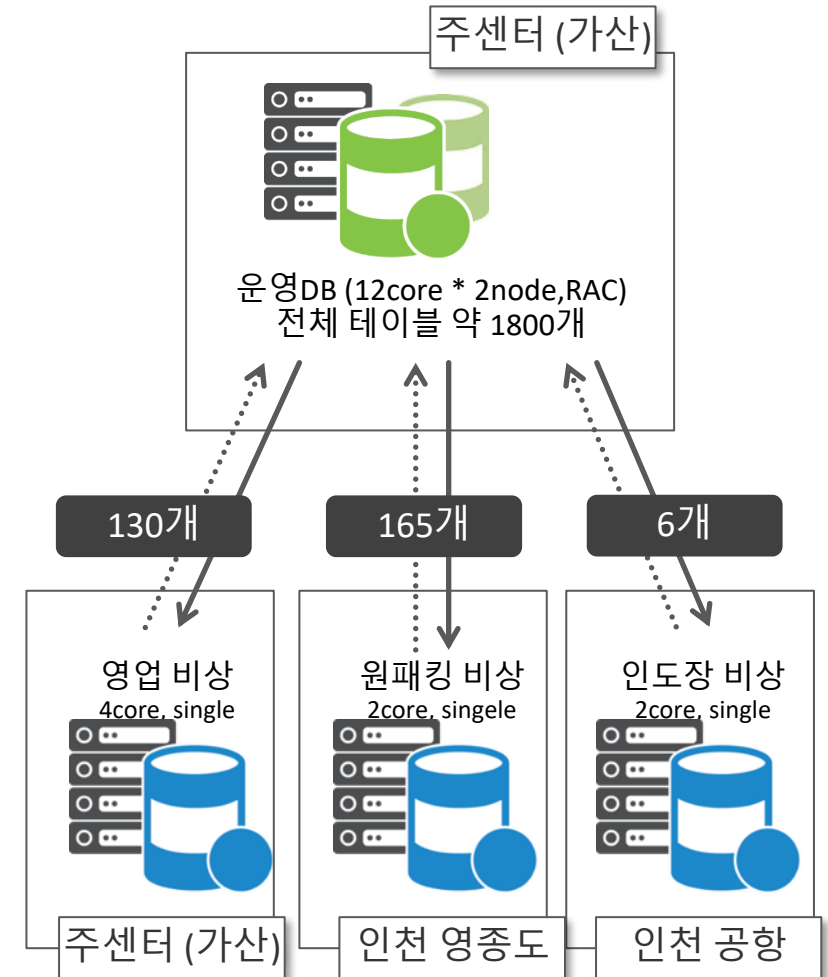
DR 복제 사례(L면세점)

✓ 도입 배경/효과

- 장애발생 시 업무별 서비스 연속성 보장을 위한 **비상시스템(DR)**을 **SharePlex** 로 구축

✓ 복제구성의 특이사항

- DR & 조회를 위한 구성 (**1:N 복제**)
- Oracle DB 버전 동일하나, 소스-Ent.와 타겟-Std.로 운영
- 주 센터 내 in-house DR, 타 지역으로 DR**
- 원복(failback)시를 위해 SharePlex 역방향 구성
- RPO = 0, RTO < 3 hour**



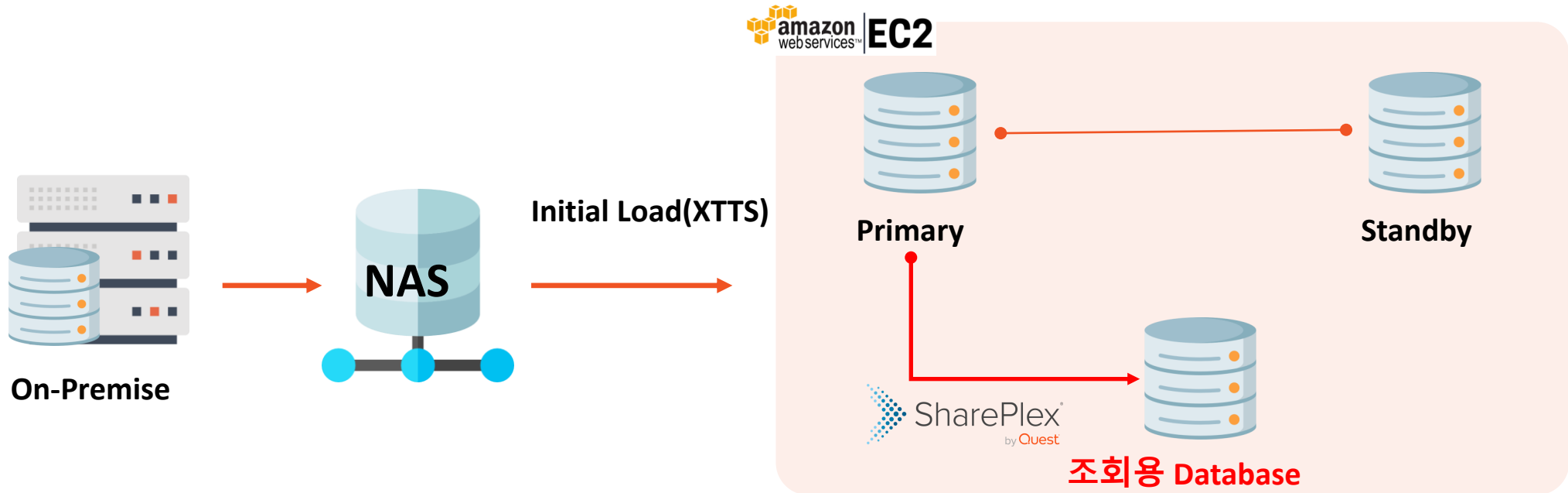
Cloud 항공사 환경 사례

✓ 도입 배경

- 운영 서버의 업무 부담을 분산하기 위하여 SharePlex 를 도입

✓ 도입 효과

- 조회성 업무를 조회 DB로 분산함으로써, 운영 서버의 안정성과 성능 확보

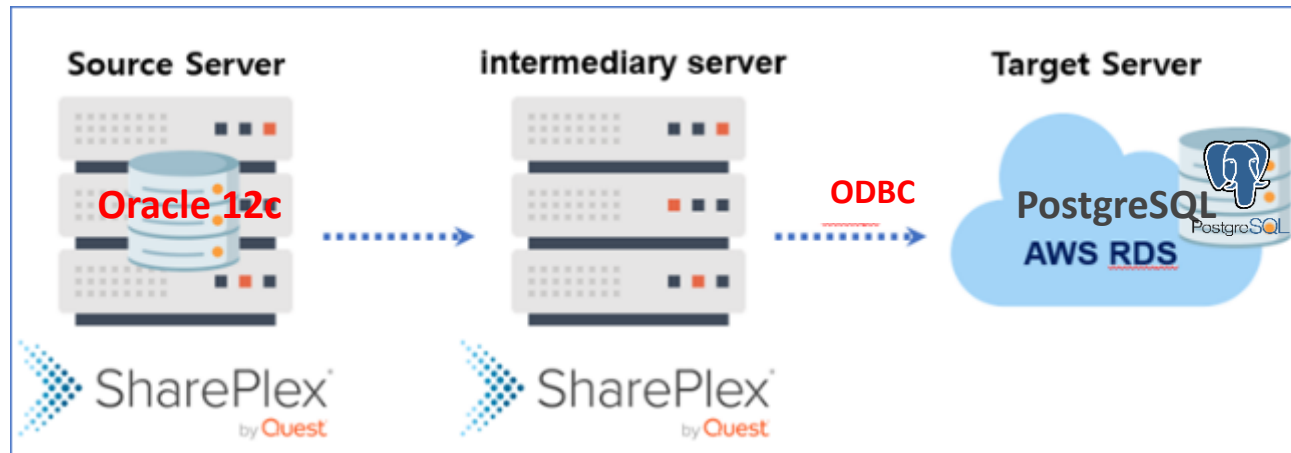


✓ 도입 배경

- 소스 데이터베이스의 **workload 분산**을 위해 SharePlex를 도입하여 AWS RDS로 주요 데이터를 실시간 동기화를 구축
- RDS(SaaS)의 환경적인 특성으로 중계서버를 이용하여 SharePlex를 구성

✓ 도입 효과

- 운영계 workload 분산으로 운영 서버의 안정성과 성능 확보



금융권 SharePlex for kafka 사례 (H생명)

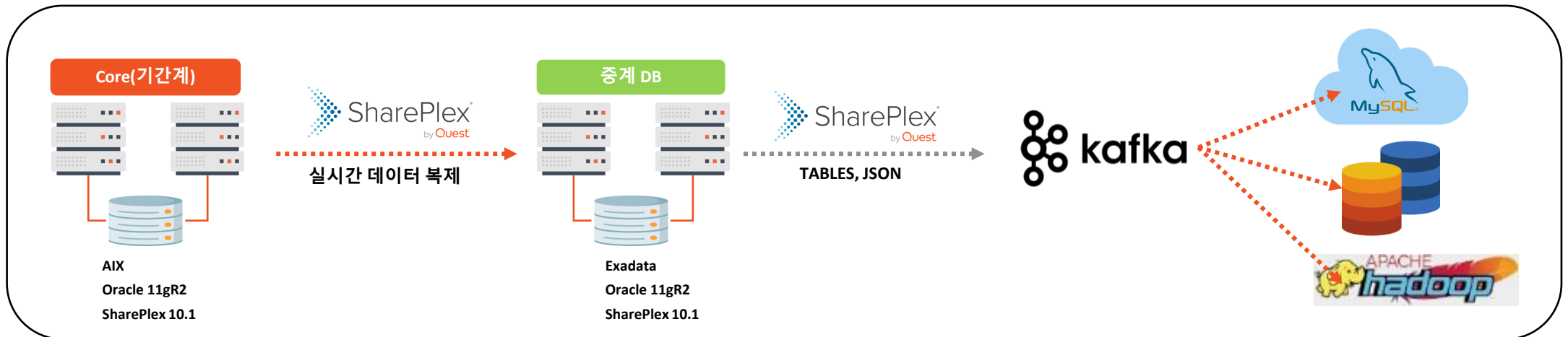
✓ 도입 배경

- 마이데이터 구축을 위해 kafka를 도입하면서, kafka에 기간계 data를 실시간으로 제공하기 위해 SharePlex를 도입한 사례

✓ 도입 요건

- 기간계 서버 영향도를 최소화 하기 위해, 중계 서버를 거쳐 kafka로 데이터를 전송하는 형태의 구성
- 전송 타입은 JSON과 XML 둘 다 지원

✓ 구성 환경

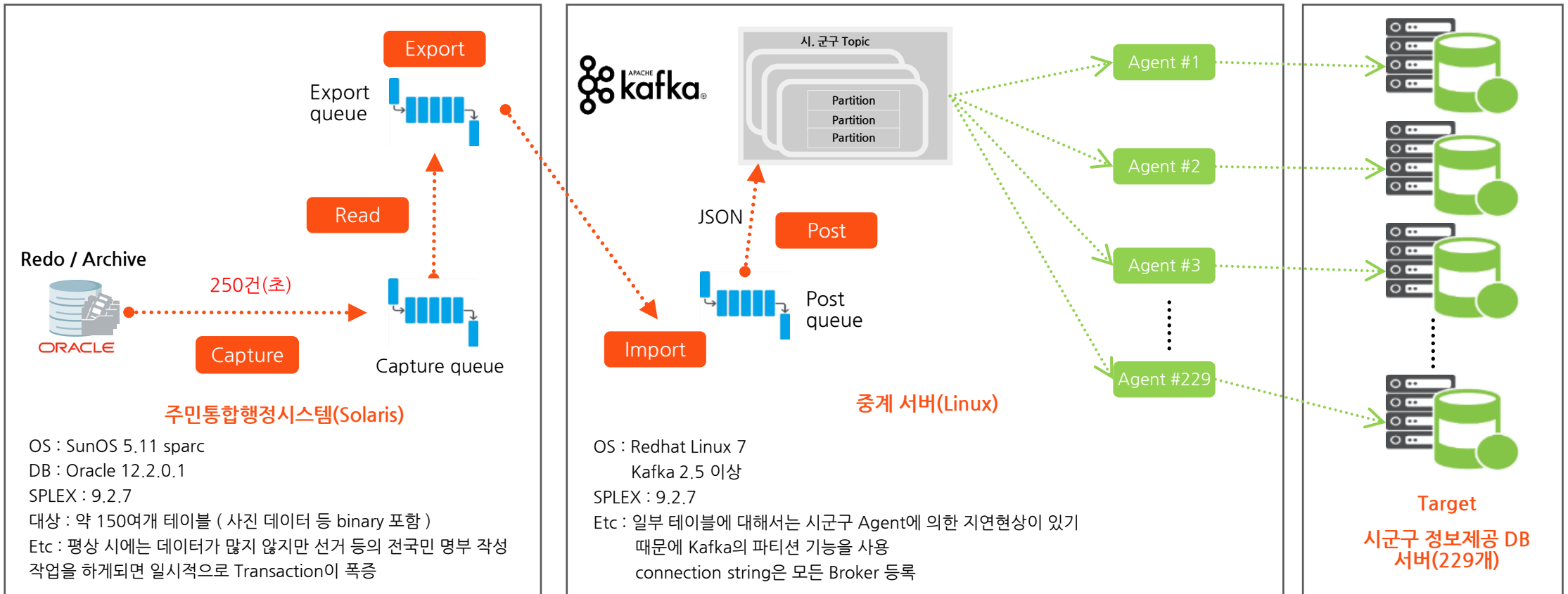


공공기관 SharePlex for kafka 사례

✓ 도입 배경

- 행정업무에 사용되는 기준정보를 229개 시.군구별로 정보 변경에 대한 일원화 및 실시간으로 제공하기 위함

✓ 구성 환경





Thank you

- **For SharePlex product**
 - <https://www.quest.com/products/SharePlex>
- **For Quest Software Korea Blog**
 - <http://questblog.co.kr>



Quest®