

**SAMSUNG SDS**

Realize your vision

# Techtonic 2019

Partner



Foresee

Disrupt

2019.11.14 • SAMSUNG SDS Tower B1F  
Magellan Hall / Pascal Hall

Track 4 | OpenSource

# 오픈소스SW로 경쟁력 있는 개발자가 되어 보자!

서성한 프로 (오픈소스플랫폼팀) / 에스코어

# AGENDA

1. 오픈소스SW의 경쟁력
2. 최적의 오픈소스 선정하기
3. 오픈소스를 잘 활용하기
4. 오픈소스와 함께 성장하기

1

---

# 오픈소스SW의 경쟁력

---

# 대표적인 오픈소스SW의 성공 사례 - Linux



Server, PC, Mobile 등 다양한 환경에서 널리 쓰이는 운영체제

## ▶ 압도적인 시장 점유율

- 퍼블릭 클라우드: 90%, 스마트폰: 82%, 슈퍼컴퓨터: 99%

## ▶ IBM이 대표적인 상용 배포판 업체인 Red Hat을 39조원에 인수

## ▶ 다양한 분야의 기업들이 협력과 경쟁을 통해서 지속적인 발전

주요 기여자(기업 위주)	비율
Intel	13.1 %
Red Hat	7.2 %
Linaro	5.6 %
IBM	4.1 %
Samsung	3.2 %

<출처 : [Linux foundation, 2017](#)>

# 4차 산업혁명을 선도하는 오픈소스SW

핵심 기술인 Big Data, Cloud, 인공지능 등 오픈소스SW 주도 하에 발전 중

## Big Data - Hadoop Ecosystem

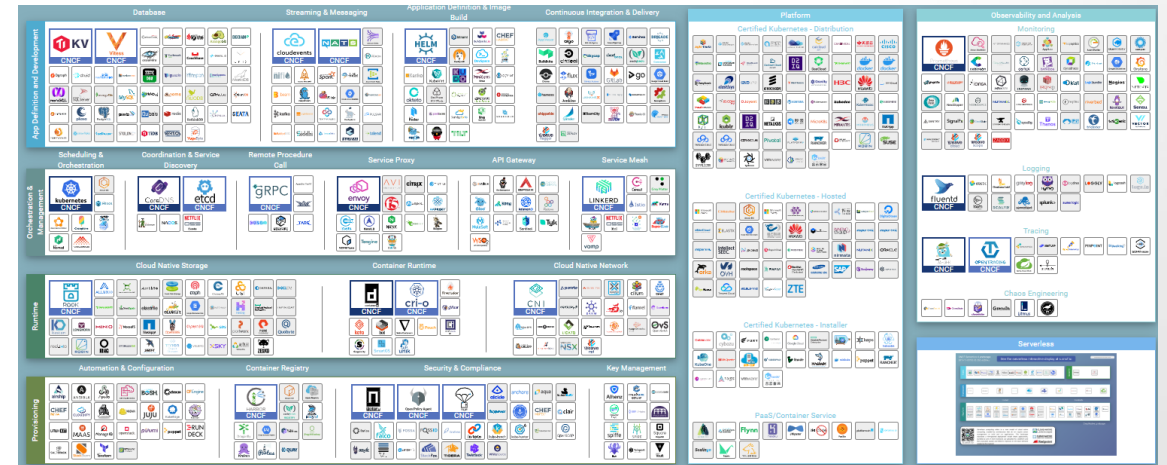
- ▶ 다양한 오픈소스가 최적 조합된 배포판
  - Storage, Processing & Analysis, Management ...
- ▶ 양대 축인 Cloudera와 Hortonworks 합병



<출처 : [Apache Foundation](https://www.apache.org/)>

## Cloud - CNCF\*

- ▶ Container, MSA 관련 오픈소스의 총체
  - 수백여 개의 프로젝트에 대형 IT 기업은 다수 참여



\* CNCF: Cloud Native Computing Foundation

<출처 : [CNCF](https://www.cncf.io/)>

# 기업용 SW로서 오픈소스SW의 경쟁력

안정성이 최우선인 기업용 SW로 쓸 수 있을까?


**비용 절감**  
 라이선스 및 유지  
 /보수 비용 절감



**IT 효율화**  
 고객 요구 사항에  
 따른 빠른 대응



**벤더 종속 탈피**  
 표준 준수 대체 SW로  
 전환 용이



상용(proprietary) SW보다 잘 난 오픈소스들도 많다!

	상용 SW	양질의 오픈소스SW
이슈 처리	<ul style="list-style-type: none"> <li>- 폐쇄적</li> <li>· Vendor에 전적으로 의존</li> <li>· 고객 불만족 시 대안 부재</li> </ul>	<ul style="list-style-type: none"> <li>- 개방적</li> <li>· 누구나 기술력을 확보하면 근본 대응 가능</li> <li>· 투명한 처리 프로세스</li> </ul>
지속적 개선	<ul style="list-style-type: none"> <li>- Vendor 전략에 따라 가변적</li> <li>- 독점적 지위 확보 시 혁신 정체 현상</li> </ul>	<ul style="list-style-type: none"> <li>- 전세계 엔지니어들의 지속적인 개선 수행</li> <li>- 기술변화에 민첩하게 대응</li> </ul>

2

---

# 최적의 오픈소스 선정하기

---



# 최적의 오픈소스 선정하기

Message Queue 선정 과정 사례 : 다양한 오픈소스 중에 무엇을 선택해야 할까?

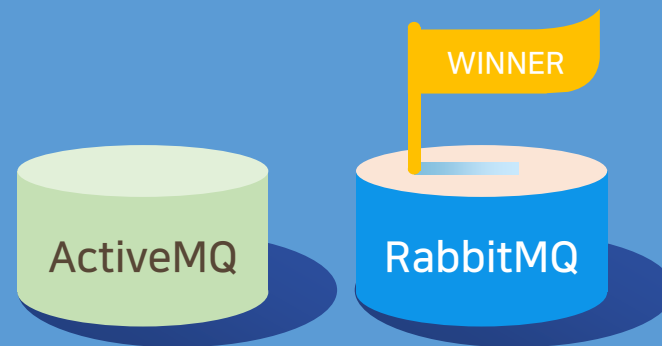
## 1단계 : 후보군 조사

- 요건 검토 : 기능 요구 사항, 고가용성, 성능, 보안 등
- 커뮤니티 활성화 수준 : Committer 수, 이슈 대응 속도, 로드맵 등
- 성숙도 : 히스토리, 유지보수 지속성, 다수의 references 등
- 기술지원 : Enterprise 버전 제공 업체 검토
- 사내 자원 현황 : 개발자/운영자 기술 역량 및 선호, 인프라 구축 현황



## 2단계 : BMT/PoC 수행

- 직접 써보면서 눈으로 확인하기
- 보유한 상용SW가 있으면 같이 비교 가능



기능	95% 이상 호환
성능	유사
가용성	보장
비용	절감 효과 UP!!!

3

---

# 오픈소스를 잘 활용하기

---

# 오픈소스 DBMS 적용 및 튜닝 사례 소개



잘 설계한 **아키텍처**가 쏘이다



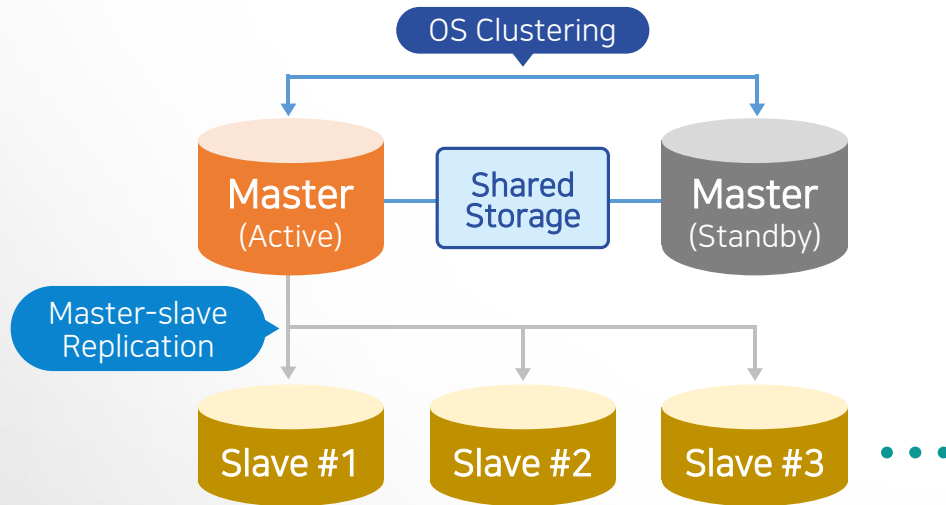
**데이터** 관련 오픈소스 잘 다루기

# 잘 설계한 아키텍처가 半이다.

다양한 고객 요건에 맞추어 최적의 RDBMS 아키텍처 선정 사례

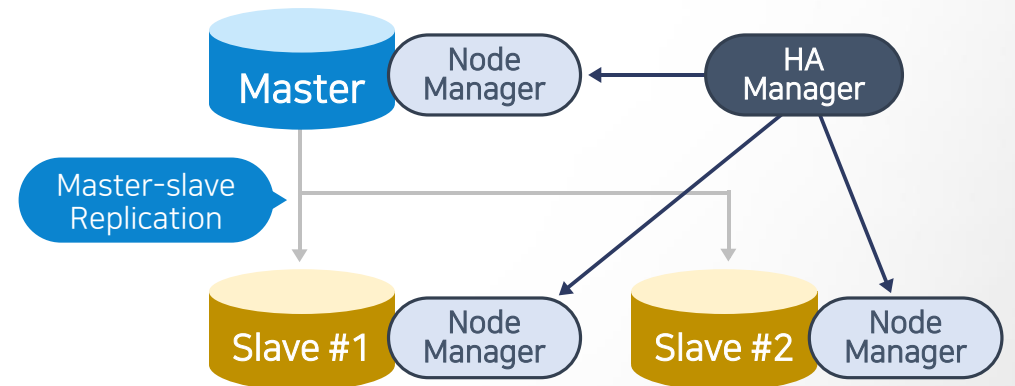
## On-Premises향 아키텍처

- Read 부하 분산 : Single-Master Multiple-Slave 구성
- Master 가용성 보장 : OS clustering + Shared storage



## Public Cloud향 아키텍처

- Cloud에서 제공하는 Managed service 활용 (AWS RDS 등)
- 고비용 Managed service 대안으로 MHA를 활용한 HA 구성



\* AWS RDS: Amazon Web Service Relational Database Service

\* MHA: Master High Availability, DB 고가용성을 위한 오픈소스

# 데이터 관련 오픈소스 잘 다루기

기존 상용 DB를 오픈소스로 전환 시 발생하는 이슈에 대한 대응 사례

## 1단계 운영 중인 구버전 기준 우회 방안 제시

### 상용 DBMS에서 동작하는 SQL

```
SELECT DISTINCT ..... 생략 .....  
FROM TBL1 t1 LEFT OUTER JOIN TBL2 t2 ON t1.COL01= t2.COL02  
WHERE (TBL1.COL03 IN (SUBQUERY))  
AND t1.COL02= 'S020227233050E211513'  
ORDER BY LOWER(t1.COL05) ;
```

서브 쿼리 분리 튜닝

### 오픈소스 구버전 기준 해결 방안

```
SELECT..... 생략 .....  
FROM TBL1 t1 LEFT OUTER JOIN TBL2 t2 ON t1.COL01= t2.COL02  
WHERE 1= 1+(SUBQUERY1)  
AND 1= 1+(SUBQUERY2)  
AND t1.COL02= 'S020227233050E211513'  
ORDER BY LOWER(t1.COL05) ;
```

## 2단계 최신 버전의 추가 기능 활용 근본 조치

### 최신 stable 버전 기준 : WITH AS 구문 사용

```
WITH subq1 (COL03)  
AS (SUBQUERY)  
SELECT..... 생략 .....  
FROM TBL1 t1 LEFT OUTER JOIN TBL2 t2 ON t1.COL01= t2.COL02  
WHERE EXISTS (SELECT 1 from subq1 where subq1.COL03 = t2.COL03)  
AND t1.COL02= 'S020227233050E211513'  
ORDER BY LOWER(t1.COL05) ;
```

### 향후 release 버전 기준 : 성능 저하 없는 SQL 호환

```
SELECT DISTINCT ..... 생략 .....  
FROM TBL1 t1 LEFT OUTER JOIN TBL2 t2 ON t1.COL01= t2.COL02  
WHERE (TBL1.COL03 IN (SUBQUERY))  
AND t1.COL02= 'S020227233050E211513'  
ORDER BY LOWER(t1.COL05) ;
```

4

---

# 오픈소스와 함께 성장하기

---

# 오픈소스와 함께 성장하기

열린 마음으로 기반으로 수용하되, 냉철한 시각으로 판단하기

## 오픈소스 잘 다루기

- 새로운 것을 지속적으로 수용
- 양질의 오픈소스 선정 안목
- 아키텍처링 역량 향상
- 기능 및 내부 구조 잘 이해하기

## 오픈소스 잘 고치기

- 양질의 코드 많이 읽기
- 작은 것부터 기여
- 상호 존중 커뮤니케이션

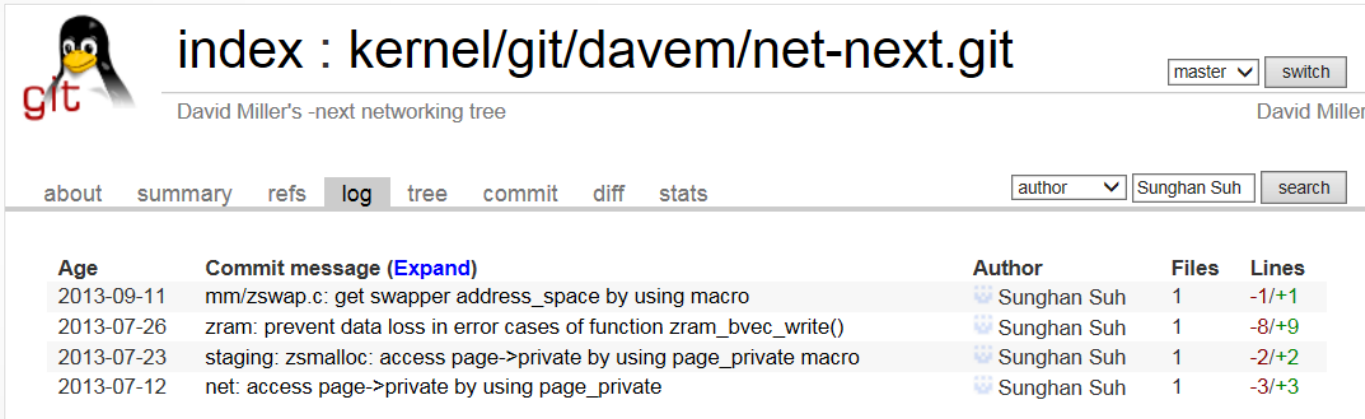
## 오픈소스로 가치 창출하기

- 오픈소스 전문 기업과 일하기
- 고객의 돈을 절약해주기
- 고객에게 최첨단 시스템 제공
- 라이선스에 대한 정확한 이해

# 오픈소스를 잘 고치려면?

“꾸준히, 자유롭게, 즐겁게”

## ➤ 작은 부분부터 차근차근 신뢰 관계 형성



index : kernel/git/davem/net-next.git

David Miller's -next networking tree

about summary refs log tree commit diff stats

Age	Commit message (Expand)	Author	Files	Lines
2013-09-11	mm/zswap.c: get swapper address_space by using macro	Sunghan Suh	1	-1/+1
2013-07-26	zram: prevent data loss in error cases of function zram_bvec_write()	Sunghan Suh	1	-8/+9
2013-07-23	staging: zsmalloc: access page->private by using page_private macro	Sunghan Suh	1	-2/+2
2013-07-12	net: access page->private by using page_private	Sunghan Suh	1	-3/+3

➤ **상처받지 말기** 커뮤니티에는 여러 사람이 거쳐 가길 반복하니, 새로운 사람을 믿기 힘들다.

➤ **언젠가 기회가 온다** 갑자기 새로운 일이 생기거나 기존 maintainer들이 일이 생기거나...



# References

- **오픈소스SW 기본 지식**
  - ✓ 공개소프트웨어 포탈 : <https://www.oss.kr/>
  - ✓ 오픈소스 라이선스 이해 : [https://www.oss.kr/oss\\_license](https://www.oss.kr/oss_license)
- **오픈소스 기여하기**
  - ✓ 꾸준히, 자유롭게, 즐겁게 : <http://osdi.insightbook.co.kr/>
- **분야별 지식**
  - ✓ Cloud Native : <https://www.cncf.io>
  - ✓ 에스코어 오픈소스 기술서비스 : <https://support.s-core.co.kr>



# Thank You



The graphic features the text 'Q & A' in a clean, sans-serif font. The 'Q' and 'A' are white, while the ampersand is a vibrant lime green. To the right of the text, two orange triangles point towards a large, dark blue circle on the far right. The background is a dark blue gradient with a pattern of lighter blue circles on the left side.

Q & A

Partner Disrupt Foresee