[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=23916997" \l "page-metadata-end)

* 알 수 없는 사용자 (22980)님이 작성, [2020-05-06](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=23916997&selectedPageVersions=2&selectedPageVersions=3)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=23916997#page-metadata-start)

* 1[요약](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-truecircle%EC%9A%94%EC%95%BD)
  + 1.1[오픈소스DB 선택 기준](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EC%84%A0%ED%83%9D%EA%B8%B0%EC%A4%80)
* 2[목적](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EB%AA%A9%EC%A0%81)
* 3[우리가 사용하고 있는 DB](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%9A%B0%EB%A6%AC%EA%B0%80%EC%82%AC%EC%9A%A9%ED%95%98%EA%B3%A0%EC%9E%88%EB%8A%94DB)
  + 3.1[핵심 업무에 ORACLE 을 가장 많이 사용하고 있는 이유](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%ED%95%B5%EC%8B%AC%EC%97%85%EB%AC%B4%EC%97%90ORACLE%EC%9D%84%EA%B0%80%EC%9E%A5%EB%A7%8E%EC%9D%B4%EC%82%AC%EC%9A%A9%ED%95%98%EA%B3%A0%EC%9E%88%EB%8A%94%EC%9D%B4%EC%9C%A0)
  + 3.2[오픈소스DB 선택 기준이 있었나요?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EC%84%A0%ED%83%9D%EA%B8%B0%EC%A4%80%EC%9D%B4%EC%9E%88%EC%97%88%EB%82%98%EC%9A%94?)
* 4[오픈소스DB 안전한가요?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EC%95%88%EC%A0%84%ED%95%9C%EA%B0%80%EC%9A%94?)
  + 4.1[오픈소스DB 는 왜 성장할까?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EB%8A%94%EC%99%9C%EC%84%B1%EC%9E%A5%ED%95%A0%EA%B9%8C?)
  + 4.2[어떤 오픈소스DB를 선택해서 사용해야 할까?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%96%B4%EB%96%A4%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EB%A5%BC%EC%84%A0%ED%83%9D%ED%95%B4%EC%84%9C%EC%82%AC%EC%9A%A9%ED%95%B4%EC%95%BC%ED%95%A0%EA%B9%8C?)
    - 4.2.1[검토1. 안전한가?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EA%B2%80%ED%86%A01.%EC%95%88%EC%A0%84%ED%95%9C%EA%B0%80?)
    - 4.2.2[검토2. 레퍼런스는 충분한가?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EA%B2%80%ED%86%A02.%EB%A0%88%ED%8D%BC%EB%9F%B0%EC%8A%A4%EB%8A%94%EC%B6%A9%EB%B6%84%ED%95%9C%EA%B0%80?)
    - 4.2.3[검토3. 우리가 만들려는 서비스를 담아 낼 수 있는가?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EA%B2%80%ED%86%A03.%EC%9A%B0%EB%A6%AC%EA%B0%80%EB%A7%8C%EB%93%A4%EB%A0%A4%EB%8A%94%EC%84%9C%EB%B9%84%EC%8A%A4%EB%A5%BC%EB%8B%B4%EC%95%84%EB%82%BC%EC%88%98%EC%9E%88%EB%8A%94%EA%B0%80?)
    - 4.2.4[검토4. 개발 경험이 있는가?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EA%B2%80%ED%86%A04.%EA%B0%9C%EB%B0%9C%EA%B2%BD%ED%97%98%EC%9D%B4%EC%9E%88%EB%8A%94%EA%B0%80?)
* 5[우리회사 오픈소스DB 활용 방향성](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%9A%B0%EB%A6%AC%ED%9A%8C%EC%82%AC%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1)
  + 5.1[방향성1. 오픈소스DB 선택 기준](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EB%B0%A9%ED%96%A5%EC%84%B11.%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EC%84%A0%ED%83%9D%EA%B8%B0%EC%A4%80)
  + 5.2[방향성2. 서비스 유형 별 운영 기준](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EB%B0%A9%ED%96%A5%EC%84%B12.%EC%84%9C%EB%B9%84%EC%8A%A4%EC%9C%A0%ED%98%95%EB%B3%84%EC%9A%B4%EC%98%81%EA%B8%B0%EC%A4%80)
  + 5.3[방향성3. 장애 예방 아키텍처 ( HA, 백업, 모니터링 )](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EB%B0%A9%ED%96%A5%EC%84%B13.%EC%9E%A5%EC%95%A0%EC%98%88%EB%B0%A9%EC%95%84%ED%82%A4%ED%85%8D%EC%B2%98(HA,%EB%B0%B1%EC%97%85,%EB%AA%A8%EB%8B%88%ED%84%B0%EB%A7%81))
* 6[오픈소스DB가 궁금해 ( Q/A )](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EA%B0%80%EA%B6%81%EA%B8%88%ED%95%B4(Q/A))
  + 6.1[Q. 오픈소스를 상업적으로 사용하면 GPL 라이선스에 따라 소스를 모드 공개해야 한다는데 맞나요?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-Q.%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4%EB%A5%BC%EC%83%81%EC%97%85%EC%A0%81%EC%9C%BC%EB%A1%9C%EC%82%AC%EC%9A%A9%ED%95%98%EB%A9%B4GPL%EB%9D%BC%EC%9D%B4%EC%84%A0%EC%8A%A4%EC%97%90%EB%94%B0%EB%9D%BC%EC%86%8C%EC%8A%A4%EB%A5%BC%EB%AA%A8%EB%93%9C%EA%B3%B5%EA%B0%9C%ED%95%B4%EC%95%BC%ED%95%9C%EB%8B%A4%EB%8A%94%EB%8D%B0%EB%A7%9E%EB%82%98%EC%9A%94?)
  + 6.2[Q. 우리가 무료로 사용하던 오픈소스DB를 특정 단체에 의해 상용화 되면 어떻게 되나요?](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-Q.%EC%9A%B0%EB%A6%AC%EA%B0%80%EB%AC%B4%EB%A3%8C%EB%A1%9C%EC%82%AC%EC%9A%A9%ED%95%98%EB%8D%98%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EB%A5%BC%ED%8A%B9%EC%A0%95%EB%8B%A8%EC%B2%B4%EC%97%90%EC%9D%98%ED%95%B4%EC%83%81%EC%9A%A9%ED%99%94%EB%90%98%EB%A9%B4%EC%96%B4%EB%96%BB%EA%B2%8C%EB%90%98%EB%82%98%EC%9A%94?)
  + 6.3[Q. 오픈소스DB 비용 비교 사례 ( IDC/Cloud, 유지보수 여부 )](https://wiki.gsretail.com/pages/viewpage.action?pageId=19403798#id-%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%ED%99%9C%EC%9A%A9%EB%B0%A9%ED%96%A5%EC%84%B1-Q.%EC%98%A4%ED%94%88%EC%86%8C%EC%8A%A4DB%EB%B9%84%EC%9A%A9%EB%B9%84%EA%B5%90%EC%82%AC%EB%A1%80(IDC/Cloud,%EC%9C%A0%EC%A7%80%EB%B3%B4%EC%88%98%EC%97%AC%EB%B6%80))

요약

**오픈소스DB 선택 기준**

* **IT인프라운영팀으로 운영 이관 시 권고하는 오픈소스DB 리스트이며,**
* **개발하려는 서비스에 더 최적화된 오픈소스DB 가 있다면 사용해도 무방합니다.**

| **DB 유형** | **위치** | **중요도** | **오픈소스DB** | **現 사용 서비스** |
| --- | --- | --- | --- | --- |
| 관계형  ( RDBMS ) | IDC | 영업 시스템 | MySQL Community + 유지보수(bug 조치 등) | 상품 구조개선, 방송넷 개선 사용 예정 |
| 업무지원 시스템 | MySQL Community | 이벤트, PMS, TMS, BMS, JIRA, 매장, LOGDB, LMS 등 |
| Cloud | 영업 시스템 | 1. EC2 MySQL Community + 유지보수(bug 조치 등)  2. AWS - RDS Aurora  ( TOAST - RDS MySQL ) | 브리즈(나이키, 노스페이스), 신물류, I-Value(Mariadb) |
| 업무지원 시스템 | EC2 MySQL Community |  |
| 문서형  ( NoSQL ) | IDC |  | MongoDB | 51Scrum 서비스 ( 심의포탈 외 17개 ) |
| Cloud | DynamoDB | 이노랩-SMIM, 핑퐁, 상품평 사용 예정 |

\*\* 배치 서버 : PostgreSQL

* **장애 예방 아키텍처 ( HA : High Availability ) 는 기본 2대(Master-Slave) 이상으로 구성하여 최소의 안정성을 확보하며**
* **효과적인 장예 예방 아키텍처 구성을 위해 사전에 아래 장애 복구 시간(RTO, RPO) 에 대한 논의가 필요합니다.**
  + **데이터 유실을 감수할 수 있는 최대 시점( RPO )   
      ex ) 장애 시점으로 부터 [ 0초, 1시간, 24시간 ] 전**
  + **시스템이 다운되었을 경우 복구되어햐 하는 최소 시간 ( RTO )  
     ex ) 장애 시점으로 부터 [ 0초, 1시간, 24시간 ] 후**

목적

* **GSSHOP 에서 사용하고 있는 DBMS 현황 공유**
* **오픈소스DB 선택 기준**
* **안정적인 운영을 위해 IDC / Cloud 에서의 오픈소스DB 아키텍처**
* **오픈소스DB 궁금증 Q/A**

우리가 사용하고 있는 DB

우리 회사에서는 다양한 DB 를 사용하고 있습니다.

업무 성격으로 분류해 보면 주문, 상품 등의 핵심서비스는 ORACLE 로, 마케팅과 내부 임직원 업무지원 시스템은 대부분 오픈소스(MYSQL, MariaDB)를 사용합니다.

( 상세 :  <http://dataportal.gsshop.com/database/databaseList/> )

| **구분** | **DBMS** | **DB 개수** | **서비스** |
| --- | --- | --- | --- |
| **상용** | ORACLE | 20 | 주문, 상품, 고객 등 핵심 서비스 |
| MSSQL | 2 | 업무 지원 |
| SYBASEIQ | 1 | DW |
| **오픈소스** | MYSQL | 18 | 마케팅, 업무지원, 브리즈(나이키) |
| AURORA | 1 | 신물류 |
| MARIADB | 3 | 업무지원, I-Value |
| DYNAMODB | 2 | 업무지원 |
| Couchbase | 1 | 업무지원 |
| MONGODB | 1 | 업무지원 |
| POSTGRE | 1 | 업무지원 |

**핵심 업무에 ORACLE 을 가장 많이 사용하고 있는 이유**

성능과 안정성 모든 부분에서 상용인 ORACLE 이 가장 뛰어나다는 것은 부정할 수는 없습니다.

이에 반해 과거 오픈소스 DB는 안정성과 레퍼런스가 부족하여 장애에 민감한 핵심 서비스에 사용하기는 어려웠습니다.

그러나 현재 MySQL, MariaDB 등 오픈소스DB의 완성도가 검증되고 대형 사이트에서의 사용 레퍼런스도 많아져,

정보전략부문에서는 2017년 부터 신규로 구축되는 모든 시스템은 오픈소스DB를 사용하고 있으며 기존 ORACLE 구축되었던 서비스도 오픈소스DB로 전환할 계획을 갖고 있습니다.

**오픈소스DB 선택 기준이 있었나요?**

현재 우리 회사는 관계형DB인 MySQL 부터 NoSQL인 DynamoDB 까지 7가지 이상 오픈소스 DB를 사용하고 있습니다.

현재까지는 오픈소스DB를 선택하는 기준은 없었습니다. 핵심 서비스에는 ORACLE 을 못 밖아 놓은 상태에서 JIRA, LOG DB, PMS 등 리스크가 적은 업무지원 시스템에는 MySQL과 MariaDB는 사용하였고,

나머지는 도입되는 S/W 에 함께 패키징 되어 있는 DB를 사용하였죠.

오픈소스DB 안전한가요?

현재 데이터베이스 시장에서 오픈소스DB 현황을 살펴보면 오픈소스DB 는 MySQL, PostgreSQL, MongoDB 를 필두로 수십개가 존재합니다.

아직 데이터베이스 시장에서  상용 DB 인 ORACLE 을 가장 많이 사용하고 있지만 아래 그림에서 오픈소스DB 사용이 지속적으로 증가하고 있음을 확인 할 수 있습니다.

오픈소스DB 의 시장 점유율이 높아지는 것은 이제 오픈소스DB 제품의 완성도가 충분히 검증되었고 비즈니스에 적용해 안정적으로 운영 및 관리할 수 있는 생태계가 형성되었다는 것을 말해줍니다.

( 참고 : <https://db-engines.com/en/ranking> )

**오픈소스DB 는 왜 성장할까?**

궁금증이 생깁니다. 오픈소스DB는 왜 성장할까요?

오픈소스DB의 수요와 성장의 가장 큰 이유는 상업용 DB 제품 도입에 필요한 높은 라이선스 비용이 아닐까 생각합니다.

또한 라이선스를 구입하더라도 몇 년에 한 번씩 업그레이드하거나 갱신을 필요로 하는데, 이때도 많은 비용을 필요로 합니다.

**다양한 서비스를 빠르게 실험하고 빠르게 구축해야 하는 우리 회사와 같은 상황에서도 높은 비용을 지불하는 상용DB 보다는 필요할 때 바로 적용해 볼 수 있는 오픈소스DB가 당연히 유리하겠죠**

**어떤 오픈소스DB를 선택해서 사용해야 할까?**

그럼, 어떤 오픈소스DB를 사용해야 할까요?

**다양한 오픈소스DB 를 경험하여 활용하는 것도 좋지만, 장애 없이 안정적으로 시스템을 운영하기 위해서는 적합한 DB를 선택하여 집중하는 것이 필요합니다.**

오픈소스DB 선택 시 고려해야 할 점은 아래의 4가지로 요약할 수 있습니다.

* **검토1. 안전한가?**
* **검토2. 레퍼런스는 충분한가?**
* **검토3. 우리가 만들려는 서비스를 담아 낼 수 있는가?**
* **검토4. 경험이 있는가?**

***검토1. 안전한가?***

오픈소스DB 는 무료로 사용할 수 있다는 큰 장점 반대편에 언제든지 발생 가능한 Bug, 장애 등을 회사 내 맨파워로  해결해야 한다는 리스크를 갖고 있습니다.

물론 장애 영향도가 적은 시스템이라면 시간을 갖고 해결 할 수 있지만, 1초의 다운타임도 용납되지 않는 주문DB 에 비정상적인 Bug, 하드웨어 Fault 등의 장애가 발생된다면 상상하기도 두렵습니다.

이 때문에 매출과 관련된 업무시스템 DB는 DownTime 제로를 위해 최소한의 안전 장치가 필요하며. 대부분의 경쟁사도 핵심 서비스에 대해서 최소한의 안정성을 보장 받을 수 있는 유지보수 계약을 하고 있습니다.  
(그래도 비용 측면에서 보면 상용DB 1/10 정도입니다.)

저희 회사 역시 주문, 상품 등의 핵심 시스템의 **오픈소스DB 선택 시 안정성이 검증되고, 유지보수가 가능한 오픈소스DB를 검토**하고 있습니다.  
(장애 영향도가 비교적 작은 내부 업무 지원 시스템에 대해서는 FULL 자사 인력으로 운영하고 있습니다)

***검토2. 레퍼런스는 충분한가?***

국외, 국내에 우리와 유사한 서비스로 구축되어 안정적으로 운영되고 있는 타 사 레퍼런스가 있는지 확인이 필요합니다.

| **회사** | **업무** | **OpenSource** | **유지보수 계약** |
| --- | --- | --- | --- |
| 쿠팡 | Coupay 결제 | MySQL | O |
| 카카오 | 메시지 처리 | MySQL  MariaDB | O |
| 티켓몬스터 | 메인 DB 조회 분산 | MariaDB  MySQL | O |
| SSG.COM | 글로벌 DB | MySQL | O |
| 롯데홈쇼핑 | OneTV | MySQL | O |
| LF몰 | 매장 POS관리 | MySQL | O |
| 씨제이올리브영 | 고객플랫폼DB | MySQL | O |

***검토3. 우리가 만들려는 서비스를 담아 낼 수 있는가?***

MySQL, PostgreSQL, MongoDB, DynamoDB 등의 오픈소스DB는 이미 안정성과 기능이 검증되어 활용되고 있어 상용 솔루션의 역할을 충분히 대신 할 수 있습니다.

중요한 점은 오픈소스DB를 상용DB 와 동일한 방식으로 사용하면 안된다는 점입니다.

오픈소스DB의 기본적인 아키텍처는 Scale-UP 이 아닌 Scale-Out 입니다. 다시 말하면,  CPU, Memory 증설이 아닌 DB 개수를 늘려서 성능 확보를 하는 개념입니다.

하지만, DB 개수만 늘린다고 될까요? 당연히 어플리케이션에서 늘어난 DB로 트랜잭션을 분리할 수 있는 구조로 설계되어야 합니다.

트랜잭션을 분리하기 위해서는 프로그램만 바뀌면 될까요? 아닙니다. 데이터 모델도 함께 변경되어야 합니다.

결국 오픈소스DB를 전환, 그리고 100% 활용하기 위해서는 오픈소스DB 성격( 데이터 모델, 샤딩, 트랜잭션 처리, 배치, 튜닝, SQL 등 )에 맞게 개발되어야 하며,

기존 상용DB환경을 오픈소스DB로 전환하는 경우에도 역시 동일합니다

* MySQL 과 MariaDB : Online 서비스DB 에 최적화
* PostgreSQL : 데이터 분석, DW 부분에서 뛰어난 성능 (멀티 프로세스)
* Aurora RDS : Cloud 환경에서 성능, 고가용성, Backup 보장
* MongoDB / DynamoDB : 각각 IDC / Cloud 에서 활용되는 대표 NoSQL

***검토4. 개발 경험이 있는가?***

아마도 ORACLE 에 익숙한 개발자 분들이 MySQL 과 같은 오픈소스DB 를 어렵게 생각할 수 있습니다.

ORACLE, MySQL, MariaDB, PostgreSQL 등의 대부분의 관계형 DB는 모두 ANSI SQL 을 사용하고 있어 사용하다 보면 금방 익숙해 집니다. 걱정안하셔도 됩니다.

물론 각 DB 마다 아주 편리한 고유 기능들이 있지만, 저희가 Oracle 기능에 묻혀 탈오라클을 못하고 있다는 점을 생각해 볼 때  Sequence, Procedure, Function, Package, DB Link 등의 특정 DB에 디펜던시가 있는 기능 사용은 지양해야 합니다.

우리회사 오픈소스DB 활용 방향성

**방향성1. 오픈소스DB 선택 기준**

| **DB 유형** | **위치** | **중요도** | **오픈소스DB** | **現 사용 서비스** |
| --- | --- | --- | --- | --- |
| 관계형  ( RDBMS ) | IDC | 영업 시스템 | MySQL Community + 유지보수 | 방송넷 개선, 상품 구조개선 사용 예정 |
| 업무지원 시스템 | MySQL Community | 이벤트, PMS, TMS, BMS, JBP |
| Cloud | 영업 시스템 | EC2 MySQL Community + 유지보수  RDS Aurora | 브리즈(나이키, 노스페이스), 신물류 |
| 업무지원 시스템 | EC2 MySQL Community |  |
| 문서형  ( NoSQL ) | IDC |  | MongoDB | 51Scrum 서비스 |
| Cloud | DynamoDB | 상품평 |

배치 서버 : PostgreSQL

**방향성2. 서비스 유형 별 운영 기준**

| **서비스 유형** | **유지 보수** | **유지 보수 내용** | **대상 서비스** |
| --- | --- | --- | --- |
| 영업 시스템  ( 매출 영향 ) | Community + 상용HA | * Bug 조치 / SR * 장애 시 HA 안정성 보장 * Backup / 모니터링 | 상품 구조개선, 방송넷DB 구조개선에 적용 예정 |
| 업무지원 시스템 | 자체 운영 | 없음 | * 이벤트DB * TMS * PMS |

**방향성3. 장애 예방 아키텍처 ( HA, 백업, 모니터링 )**

대표적으로 MySQL, MariaDB, PostgreSQL 등 오픈소스DB 의 고가용성(HA : High Availability) 방법은 저희가 사용하고 있는 상용 ORACLE 과는 완전히 다릅니다.

간단하게 말하면, ORACLE 은 하나의 저장소를 이용하여 고가용성을 구현하는 반면, MySQL 과 같은 오픈소스DB는 여러 개의 저장소를 동기화하는 방법으로 고가용성을 확보합니다.

때문에 저장소 사이의 동기화 지연, 데이터 정합성 오류라는 ORACLE 에서는 없었던 새로운 걱정이 생기게 되죠.

아래 그림에서 확인 할 수 있듯이 ORACLE은 하나의 저장소를 이용하기 때문에 운영자가 데이터 정합성 관련하여 따로 걱정할 부분이 없는 반면에 ( 이것이 Oracle 의 기술이죠 )

저장소를 각각 갖고 MySQL 아키텍처는 운영자가 데이터 Sync 보장을 위해 3'rd Party 솔루션을 사용하고, 지속적인 모니터링을 필요로 합니다.

참고로 AWS 와 같은 Cloud 내 RDS 는 자체적인 Availability Zone (AZ) 개념으로 데이터 동기화를 지원하여 이 역시 ORACLE 과 같이 데이터 정합성은 운영자가 걱정할 부분이 없습니다. ( 당연히 RDS 사용은 추가 비용이 들겠죠.. )

아래는 ORACLE 과 MySQL 그리고 Cloud 에서의 HA Concept 을 보여줍니다.

| **IDC** | | **Cloud** |
| --- | --- | --- |
| **ORACLE** | **MYSQL** | **RDS MySQL** |
|  |  |  |

그럼, 우리 회사에서는 사용하고 있는 / 검토하고 있는 오픈소스DB 아키텍처를 살펴볼까요?

| **구분** | **복제 방식** | **중요도** | **장점** | **단점** | **비용** | **적용 서비스** | **아키텍처** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Community  버전 | SQL 복제 | 업무지원 시스템 | * 구성이 간단함 | * 장애 복구가 수동으로 이루어짐 * Master 와 Slave 간 데이터 동기화 지연 발생 가능성 존재 * 엔진 레벨 Bug 발생 시 대응 지연 | 없음 | * EC-IN * TMS * PMS * LOGDB * JBP |  |
| 디스크 복제  ( DRBD Sync ) | 업무지원 시스템 | * 장애 복구가 자동으로 이루어짐 * Query Sync 보다 데이터 동기화가 빠름 | * Standby Server 사용 불가 * 대량 트랜잭션 발생 시 Corruption 발생 * 엔진 레벨 Bug 발생 시 대응 지연 | 없음 | * 알림톡 * LMS * 이벤트 * BMS * 홈페이지 * JIRA * SSO * 제도개선 * 자막기 |  |
| MySQL Community + 상용HA | SQL 복제 + Cluster | 영업 시스템  ( 매출 관련 ) | * Bug 대응 * 데이터 Loss 가 없으며 데이터 일관성 보장 * 장애 복구가 자동으로 이루어짐 | * Master 와 Slave 간 데이터 동기화 지연 발생 가능성 존재 | 있음 | 상품 구조개선,  방송넷DB 구조개선에 적용 예정 |  |

오픈소스DB가 궁금해 ( Q/A )

**Q. 오픈소스를 상업적으로 사용하면 GPL 라이선스에 따라 소스를 모드 공개해야 한다는데 맞나요?**

오픈소스DB는 GPL 라이선스에 따라 상업적으로 판매(배포)하는 경우 소스를 모두 공개해야 합니다.

여기서 "상업적으로 판매(배포)" 부분이 중요한데요, 예를 들어 설명하면,

1. 우리 회사가 응용SW와 MySQL 커뮤니티 버전(GPL 2.0)을 패키징하여 A 회사에 납품하였을 경우   
   → A 회사에서 소스코드 요청 시 공개의 의무를 갖습니다.
2. 우리가 직접 MySQL 커뮤니티 버전(GPL 2.0)을 설치 하여 개발했을 경우  
   → 의무사항이 발생하지 않습니다.

2번 예와 같이 다른 회사로 판매(배포)하지 않는 이상 아무런 의무사항이 없습니다.

**Q. 우리가 무료로 사용하던 오픈소스DB를 특정 단체에 의해 상용화 되면 어떻게 되나요?**

* MySQL 의 경우 2010년 오라클에게 인수되었지만, Community 버전은 GPL 라이선스 규칙에 따라 모든 소스가 공개되어 있고 ORACLE 사가 저작권을 갖고 있지는 않습니다.  
  Oracle 사에서 저작권을 갖고 있는 제품은 MySQL Commercial 버전으로 Community 기본 기능에 Backup, HA(High Availability), Monitoring, SR 등을 함께 제공하고 이를 Standard, Enterprise 라고 명명합니다.
* 만약 Oracle 사에서 MySQL Community 버전 지원 중단 또는, MariaDB 상용화가 되더라도 앞으로 DB 구축 시 ANSI SQL 사용, 특정 DB에만 특수화된 기능 사용 지양을 기본 원칙으로 하고 있기 때문에 다른 오픈소스DB로 쉽게 이관 가능합니다.

**Q. 오픈소스DB 비용 비교 사례 ( IDC/Cloud, 유지보수 여부 )**

ORACLE 과 MySQL Community, HA 라이선스만을 대상으로 하였고,

2 Socket 4 Core 의 쓸만한 DB서버에 성능 확보를 위한 Partition 기능을 사용한다고 가정했을 때를 비교하였습니다.

( 천원 )

|  | **버전** | **Bug  & 장애대응** | **Y+0** | **Y+1** | **Y+2** | **Y+3** | **Y+4** | **총비용** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ORACLE | Oracle Enterprise  Partition Option | 유지보수 계약  Bug / 보안 패치  가용성 보장 | 288,000 | 57,600 | 57,600 | 57,600 | 57,600 | 518,400 |
| MySQL | MySQL Community | 자사 인력 대응  커뮤니티 패치 대기 | 0 | 0 | 0 | 0 | 0 | 0 |
| MySQL Community + HA  ( 서버 3대 기준 ) | 유지보수 계약  Bug / 보안 패치  가용성 보장 | 18,000 | 18,000 | 18,000 | 18,000 | 18,000 | 90,000 |

감사합니다. 궁금한 내용은 IT인프라운영팀 DB 파트에 언제든지 문의주세요.

[Skip to end of metadata](https://wiki.gsretail.com/display/DB/AWS+RDS+vs+MySQL+on+EC2#page-metadata-end)

* 작성자 : [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh) - [2020-11-16](https://wiki.gsretail.com/pages/viewpreviousversions.action?pageId=41885025)

[Go to start of metadata](https://wiki.gsretail.com/display/DB/AWS+RDS+vs+MySQL+on+EC2#page-metadata-start)

Amazon의 RDS와 EC2에 MySQL (MySQL Enterprise Edition) 상용 버전을 설치하여 사용 할 때의 장단점이 기술된 문서입니다.

해당 문서는 오라클 사에서 작성하였으므로, 제약이 존재하는 RDS보다는 DBA조직이 있다면 EC2에 MySQL을 설치하여 더 유연하게 사용하는게 좋다는 백서입니다.

**MySQL-Enterprise-Edition-on-Amazon-KR\_v1.1.doc**

**요약**

여러분의 조직에서 퍼블릭 클라우드 업체로 옮기겠다는 결정을 할 것입니다.  
그 결정은 ‘Database-as-a Service’ (MySQL의 포크버전인 RDS 위에) 를 선택하거나 혹은 MySQL Enterprise Edition을 ‘Infrastructure-as-a-Service’ (EC2)에서 실행하는 것 입니까?  
이 백서에서는 이 두가지 옵션을 살펴보고 위험 및 이점에 대해 논의할 것입니다.

**MySQL이란?**

MySQL은 세계에서 가장 널리 사용되는 오픈 소스 데이터베이스로, 성능, 안정성 및 사용 편의성을 제공함으로써  
MySQL은 Facebook, Twitter, YouTube, Yahoo 등 유명 웹 사이트에서 사용되는 웹 기반 응용 프로그램의 주요 데이터베이스로 자리 매김했습니다.

오라클에서는 가장 큰 MySQL 엔지니어링 및 지원 조직을 제공하며 차세대 웹, 클라우드, 모바일 및 임베디드 애플리케이션에 새로운 기능을 제공하는 MySQL 혁신을 추진합니다.

**Amazon Relational Database Service (Amazon RDS)는 무엇일까요?**

Amazon Relational Database Service (Amazon RDS) 는 클라우드에서 관계형 데이터베이스를 쉽게 설정, 작업, 및 확장할 수 있도록 해주는 웹 서비스입니다.  
이는 업계 표준 관계형 데이터베이스에 대해 비용 효율적이고, 변경 가능한 용량을 제공하며 일반적인 데이터베이스 관리 작업을 관리합니다.

RDS를 통해 Amazon은 백업, 소프트웨어 패치, 자동 장애 감지 및 복구와 같은 관계형 데이터베이스에 필요한 많은 복잡하고 지루한 관리 작업을 대신 수행합니다.  
그러나 이런 관리 서비스 경험을 제공하기 위해 Amazon RDS는 기본 운영 체제에 대한 셸(shell) 액세스를 제공하지 않으며 데이터베이스 내의 고급 권한이 요구되는 특정 시스템 프로시저 / 테이블에 대한 액세스를 제한합니다.

**Amazon EC2는 무엇일까요?**

Amazon Elastic Compute Cloud (Amazon EC2) 는 Amazon Web Services (AWS) 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다.  
Amazon EC2를 사용하면 하드웨어에 대한 투자가 필요 없으므로 애플리케이션을 신속하게 개발하고 배포 할 수 있습니다.  
Amazon EC2를 사용하여 필요한 만큼 가상 서버를 시작하고, 보안을 설정하고, 네트워킹을 구성하고, 스토리지를 관리 할 수 있습니다.  
Amazon EC2를 사용하면 요건 변경 사항이나 트래픽 급상승에 대한 변경 사항을 처리하기 위해 스케일 확장 또는 축소 할 수 있으므로 트래픽에 대한 사전 예측을 할 필요가 줄어 듭니다.

EC2를 사용하면 플랫폼에 개별적으로MySQL 소프트웨어를 배포해야 합니다.  
Amazon에서 설치, 설정 그리고 지원 또한 제공하지 않습니다.

**RDS vs EC2에 대한 고려 사항**

기업 / 조직의 내부 IT시스템, 플랫폼, 개발 및 지원팀에 대해 잘 이해하는 것이 중요합니다.  
만일 엄격하고 심지어 규제 요구사항이 있는 규모가 크고 성숙한 시스템을 경험이 풍부한 DBA그룹이 운영하고 있다면 이를 관리 서비스로 옮기는 것은 좋지 않은 선택일 것 입니다.  
다른 한편, 대부분 경험있는 데이터베이스 관리자(DBA)가 없는 스타트업이나 새로운 시스템을 설계하는 경우 관리 서비스가 아마도 중요한 고려 사항으로 될 수 있습니다.

RDS를 사용하는 장점은 배포의 용이성, 유연한 사용, 수평 (읽기) 확장성이 필요하고 사용자가 경험있는 내부 인력이 많지않은 경우입니다.  
왜냐면 이런 것들을 RDS에서 전부 제공하고 있기 때문입니다.

RDS의 단점은 기업이 장기적으로 더 많은 비용을 들이게 된다는 것입니다.  
이것은 사용자가 전체 액세스 제어 권한을 가지고 있지 않고 추상화 레이어(예: 인스턴스의 OS를 제어할 수 없음)에서 작업하도록 제한되어 있기 때문입니다.  
이것은 솔루션의 특정 중요한 부분이 없어지거나 비 활성화되는 결과를 초래하게 됩니다.  
그리고 무엇인가 잘못된 경우, 기업에서는 MySQL전문가가 무슨 일이 발생하고 있는지 파악해야 합니다.  
하지만 고객들은 실제로 데이터베이스 분석 데이터를 볼 수 없기 때문에 운영의 복잡성을 초래하게 됩니다.  
만일 이 솔루션이 미션 크리티컬하거나 고성능 시스템이 연중 무휴로 운영되는 경우 RDS가 백업 및 패치를 제어한다는 사실은 큰 어려움을 초래하게 됩니다.

따라서, DBA없이 쉽게 배포할 수 있도록 제공하는 유연성과 운영 제어에 대한 트레이드오프는 장기적으로 봤을때 그다지 비용효과적이지 않을 수 있으며 여전히 DBA팀이 필요할 수 있습니다.  
반대로 DBA팀이 IaaS상 데이터베이스에 대한 자동 배포 코드를 생성하는 것은 어렵지 않습니다.  
고객은 효과적으로 두가지 장점을 모두 누릴 수 있습니다.

미션 크리티컬 시스템을 RDS로 옮길 때 가장 중요한 고려사항은 이런 선택이 기업으로 하여금 솔루션을 사용할 때 유연성이 부족하고 근본적으로 문제를 해결할 수 없는 상황이 발생할 수 있습니다.  
대신에 미리 패키지화 된 데이터베이스를 IaaS로 가져 가면 훨신 더 나은 유연성과 제어 기능을 제공합니다.

**Amazon EC2에서나 Amazon RDS에서 MySQL은** **동일합니까?**

아닙니다, RDS는 MySQL의 포크(fork) 입니다. (포크는 소스 코드 작성자로부터 지원되지 않는 원 소스 코드에서 수정된 버전입니다.)  
이것은 MySQL의 모든 기능을 사용하지는 않습니다. 또한 MySQL Enterprise Edition은 RDS에서 사용할 수 없습니다.  
Amazon RDS는 현재 다음과 같은 MySQL 기능을 지원하지 않습니다:

* MySQL Enterprise Authentication
* MySQL Enterprise TDE
* MySQL Enterprise Encryption
* MySQL Enterprise Masking
* MySQL Enterprise Firewall
* MySQL Enterprise Audit
* MySQL Shell
* MySQL InnoDB Cluster
* MySQL Document Store

결과적으로 이런 기능들이 없기 때문에 다음과 같은 작업을 수행할 수 없습니다.

* MySQL Document Store 기능 활용 (X Plugin을 로드할 수 없음).
* InnoDB Cluster의 고가용성 옵션 또는 semi-sync복제 기능.
* MySQL 보안 기능들. RDS는 Oracle MySQL보안 플러그인을 로드 할 수 없으며 암호 검증을 위한 Oracle MySQL 오픈소스 플러그인을 허용하지 않기때문에 보안 관련된 취약점이 있습니다.

추가로 Amazon에서 지원하는 MySQL 버전을 사용하면 다음 버전으로 업그레이드 할 시기 또는 운영할려고 하는 특정 버전을 여러분이 결정할 수 없습니다.  
그리고 설정을 제어할 수도 없고 “set persist”를 이용하여 MySQL 인스턴스를 설정할 수도 없습니다.

**MySQL 데이터베이스** **지원**

앞에서 설명드린 바와 같이, Amazon RDS는 커뮤니티 에디션의 무료 오픈소스 데이터베이스의 포크된 버전을 사용하고 있습니다.  
이것은 미션 크리티컬하지 않은 개발 프로젝트에 대해서는 괜찮지만, Amazon RDS 환경에 대한 지원은MySQL 엔진에 대한 지원을 포함하지 않습니다.  
고객들은 RDS환경의 데이터베이스 엔진의 MySQL데이터베이스 버전 및 패치 업그레이드는 완전히 Amazon에 의존하고 있습니다.  
결과적으로 Amazon RDS에서 사용하는 데이터베이스 엔진은 Oracle MySQ에 대한 상용 지원 서비스와는 전혀 관계가 없기때문에 MySQL에서 발견된 버그에 대해서 우회방법만을 제안할 수 밖에 없습니다.  
MySQL에서 지원되는 모든 핫 픽스는 RDS에 배포할 수 없습니다.  
버그는 단 한번만 Oracle MySQL의 오픈 소스 버전에 수정되며, Amazon RDS에서 이 릴리스를 채택할지 말지는 결정합니다.

**Amazon RDS서비스의 제약사항**

RDS는 관계형 데이터베이스 서비스입니다. 그리고Amazon이 사용 가능한 기능들과 새로운 기능을 추가할 시기를 제어합니다.  
Amazon은 퍼블릭 클라우드 제공자가 기대하는 모든 보안 인증을 제공할 수 있다 하더라도 Amazon RDS에서 어떤 기능을 사용할 수 있는지에 대한 통제력이 부족한 것이 그것 자체로 보안 리스크입니다.  
규제 대상 산업의 경우, 특히 공공 부문에서는 요구되어지는 기능만 사용하도록 하는 규제사항이 있기때문에 보안 위협과 정보유출에 대해 공격 대상의 영역을 제한하게 됩니다.

**크로스 클라우드 개발 문제**

RDS와 같은 포크(fork)버전의 MySQL버전을 사용할 때의 또 다른 문제점은 온프레미스 환경에서의 설정과 다르며 다른 클라우드 벤더사이에도 일관성이 없다는 것입니다.  
모든 크로스 클라우드 개발에서는 내부 개발팀과 운영팀이 이러한 차이점을 인식해야 합니다.  
하이브리드 환경에서 MySQL을 배포하는 경우(온프레미스와 다른 클라우드 벤더), 동일한 릴리스/툴을 사용하여 배포를 통일할 것을 권장합니다.  
EC2를 사용하는 경우(혹은 비슷한 IaaS플랫폼), 예를 들어 Ansible혹은 Docker환경을 사용하여 MySQL을 오케스트레이션/설치하면 모든 MySQL설치가 동일해 집니다.

**Amazon RDS는 현재 Oracle MySQL최신 버전을 사용할까요?**

보안 및 최강의 성능을 위해서는 모든 소프트웨어의 사용 가능한 패치 업데이트를 유지해야 합니다.  
그러나 이 글을 쓰는 시점(2019년 5월)에 RDS 의 MySQL 버전은 최대 10개월이나 차이가 납니다.(MySQL 8.0 - 3 개월, MySQL 5.7 - 4 개월, MySQL 5.6 – 10 개월, MySQL 5.5 – 10 개월).  
EC2에서 MySQL Enterprise Edition을 사용하면 애플리케이션이 보안 리스크에 노출 되는 문제를 피할 수 있습니다.

**MySQL Enterprise Edition은 GDPR, PCI 등과** **같은** **규제** **준수** **요구** **사항을** **어떻게** **만족합니까?**

많은 기업들은 GDPR, PCI, HIPPA 등과 같이 끊임없이 증가하는 규제 요구 사항 및 데이터 보호 정책을 준수하기 위해 애쓰고 있습니다.  
MySQL Enterprise Edition은 이러한 컴플라이언스 요구 사항을 충족시킬 수있는 강력한 도구들을 제공합니다. 이런 요구사항은 4가지 주요 영역으로 구성됩니다:

* **평가**: MySQL Enterprise Monitor를 사용하여 취약한 암호 정책과 같은 보안 취약성을 식별합니다.
* **방어**: 탄력적인 데이터베이스 인프라를 만들고, 사용하지 않는 데이터뿐만 아니라 사용 중인 데이터에 대해서도 암호화와 같은 적절한 안전 조치를 구현하는 툴을 제공합니다.
* **감지**: MySQL Enterprise Audit은 누가 어떤 정보에 액세스하였는지를 식별함으로 사이버 보안 이벤트를 적시에 발견하는 데 도움이 됩니다.
* **복구**: MySQL 은 일반적인 복구 운영 작업으로 사이버 보안 이벤트의 영향을 줄입니다.

**Security Architecture**

**Amazon EC2에서** **MySQL Enterprise Edition을 실행하면 어떤 이점이 있을까요?**

수백만 건의 설치를 통해 MySQL Enterprise Edition은 오늘날의 대부분의 웹 애플리케이션에서 사용되는 선도적인 오픈 소스 데이터베이스 솔루션의 상용 버전입니다.  
MySQL Enterprise Edition에는 가장 포괄적인 고급 기능, 관리 도구 및 기술 지원이 포함되어 있어 고객이 최고 수준의 MySQL 확장성, 보안, 안정성 및 가동 시간을 달성 할 수 있습니다.  
이것은 비즈니스적으로 크리티컬한  MySQL 애플리케이션을 개발, 배포 및 관리 할 때 발생하는 리스크, 코스트  및 복잡성을 줄여줍니다.

**WhatMySQL Enterprise Edition에 다음과 같은 기능이 포함됩니다**.

**MySQL Enterprise Backup**

MySQL Enterprise Backup은 MySQL 데이터베이스의 온라인, 넌-블로킹 백업을 제공합니다.  
MySQL은 모든 트랜잭션이 실행되는 동안에도 모든 InnoDB 데이터에 대해 전체, 증분 및 부분 백업을 수행합니다.  
모든 백업 작업은 빠른 결과를 위해 병열로 실행되고 백업 이미지의 사이즈를 90% 줄일 수 있는 압축 옵션도 지원합니다.  
복구 옵션에는 이전 버전과 호환 가능한 전체 복구, 한번 클릭으로 특정 시점 복구및 특정 객체 집합을 위한 부분 복구가 제공합니다.

**MySQL Enterprise High Availability**

MySQL Enterprise 고가용성은 가장 까다로운 미션 크리티컬한 애플리케이션의 가용성 요구 사항을 충족할 수 있게 합니다.

MySQL InnoDB Cluster는 그룹 복제, MySQL Router, 및 MySQL Shell과 함께 MySQL서버를 사용하여 데이터베이스에 통합된 고유의 HA 솔루션을 제공합니다.  
이것은 InnoDB, GTID, 바이너리 로그, 멀티-스레드 슬레이브 실행, 멀티-소스 복제 및 Performance Schema등 검증된 MySQL기능을 활용합니다.  
MySQL InnoDB Cluster 는 5분 내에 설치 가능하며 MySQL Shell의 스크립트 가능한 관리 API를 사용하여 관리할 수 있습니다.

**MySQL Enterprise Scalability**

MySQL Enterprise 확장성은 사용자, 쿼리 및 데이터 로드가 증가됨에 따른 지속적인 성능 및 확장성 요구사항을 충족시킬 수 있습니다.  
MySQL 스레드 풀은 효율적인 스레드 처리 모델을 제공하고, 클라이언트 컨넥션 및 sql구문 실행 스레드를 관리하는 오버헤드를 줄이도록 설계 되였습니다.

**MySQL Enterprise Authentication**

MySQL Enterprise Authentication은 외부 인증 모듈을 사용할 준비가 되어 있어 Linux Pluggable Authentication Modules (PAM) 및  Windows Active Directory 를 비롯한 기존의 보안 인프라와 쉽게 통합할 수 있습니다.  
MySQL Enterprise Authentication을 사용하면 조직에서 싱글 사인온 (SSO) 메커니즘을 구현하고 기존 보안 규칙을 활용하여 중앙 집중식 디렉토리에서 처리할 수 있습니다.

**MySQL Enterprise Transparent Data Encryption (TDE)**

MySQL Enterprise Transparent Data Encryption (TDE)는 데이터베이스의 물리 파일을 암호화하여data-at-rest 암호화를 가능하게 합니다.  
데이터는 데이터 스토리지에 쓰기 전에 자동 암호화되고 저장 스토리지에서 읽을 때 복호화됩니다.  
결과적으로 해커와 악의적인 사용자는 테이블 스페이스 파일, 데이터베이스 백업, 또는 디스크에서 중요한 데이터를 읽을 수 없습니다.  
MySQL Enterprise TDE는 마스터 키와 테이블 스페이스 키로 구성된 2 계층 암호화 키 구조를 사용합니다. 이런 아키텍처로 쉬운 키 관리와 순환(rotation)을 제공하고 있습니다.

**MySQL Enterprise Encryption**

중요한 데이터를 보호하기 위해MySQL Enterprise Encryption은 비대칭 암호화 (공개 키 암호화)를 위한 산업 표준 기능을 제공합니다.  
MySQL Enterprise Encryption는 민감한 데이터를 보호하고 HIPAA, Sarbanes-Oxley 및 PCI 데이터 보안 표준과 같은 규제 요구 사항을 준수하는 데 도움이 되는 암호화, 키 생성, 디지털 서명 및 기타 암호화 기능을 제공합니다.

**MySQL Enterprise Masking and De-identification**

MySQL Enterprise Masking 및 De-identification은 사용하기 쉽고 내장된 데이터베이스 솔루션을 제공하여 실제 값을 숨기거나 대체하여 권한이 없는 사용자로 부터 민감한 데이터를 보호할 수 있습니다.  
모든 주요 산업 규정에서는 PII (개인 식별 정보), PAN (기본 계정 번호) 및 기타 기밀 데이터에 대한 데이터 마스킹을 요구하고 있어서 권한 있는 사람만 해당 데이터에 접근할 수 있도록 해야 합니다.

**MySQL Enterprise Firewall**

MySQL Enterprise Firewall은 중요한 개인 및 금융 데이터의 손실을 초래할 수있는 SQL 주입 공격을 차단합니다.  
화이트리스트를 생성하고, 실시간 위협 모니터링, SQL 문 차단 및 경고 기능을 통해 DBA로 하여금 데이터 자산을 보호 할 수 있습니다.  
침입 탐지 시스템 역할을 하는 MySQL Enterprise Firewall은 관리자에게 승인된 화이트 리스트와 일치하지 않는 SQL문에 대해 알려줍니다.

**MySQL Enterprise Audit**

MySQL Enterprise Audit을 사용하면 기존 애플리케이션에 정책 기반 감사 준수를 빠르고 원활하게 추가 할 수 있습니다. 사용자 수준의 활동 기록을 동적으로 활성화하고, 활동 기반 정책을 구현하고, 감사 로그 파일을 관리하고, MySQL 감사를 Oracle 및 타사 솔루션과 통합 할 수 있습니다.

**MySQL Enterprise Monitor**

MySQL Enterprise Monitor 및 MySQL Query Analyzer를 사용하면 MySQL 인스턴스, 해당 인스턴스를 사용하는 애플리케이션 및 지원 인프라의 성능 및 가용성을 향상시킬 수 있습니다.  
MySQL Enterprise Monitor는 지속적으로 MySQL 쿼리 및 성능 관련 서버 지표을 모니터링하고 개발자와 DBA에게 기본 성능 흐름과 큰 차이가 있음을 경고합니다.  
Replication Dashboard는 MySQL 인스트루먼트(instrumentation) 정보를 표시하고 토폴로지 뷰는 복제 그룹의 현재 구성을 표시하므로 각 노드 및 각 복제 하위 시스템의 상태를 신속하게 확인할 수 있습니다.  
Best practice Advisors는 성능을 향상시키기 위해 설정 및 변수 설정을 변경할 것을 제안합니다.  
흐름 분석을 활용하는 MySQL Enterprise Monitor는 문제가 심각해지기 전에 문제를 경고하고 향후 용량 요구사항을 정확하게 예측할 수 있습니다.

**Oracle Premier Support**

오라클은 MySQL에 대한 24x7 글로벌 지원을 제공합니다. MySQL 지원팀은 데이터베이스 전문가인 MySQL 개발자로 구성되어 있으며 여러분이 직면한 문제와 도전을 잘 이해하고 있습니다.  
Oracle Enterprise Edition에 대한 Oracle Premier Support에는 다음과 같은 기능들이 포함되어 있습니다.

* 24x7 제품 지원
* 횟수 제안없이 지원
* 기술 자료
* 유지보수 릴리스, 버그 수정, 패치 및 업데이트
* MySQL 컨설팅 지원

**Oracle MySQL은** **제품** **업데이트에** **투자하고** **있습니까?**

MySQL8.0에는 400 개가 넘는 기능 (worklogs)이 추가되었으며, 500 개의 새로운 테스트가 추가되었으며 5,000 개 이상의 버그가 수정되었습니다.  
또한 미국, 인도 및 스웨덴에 개발 팀을두고 있습니다. 스웨덴은 MySQL의 탄생지이며 조직 내에서 초창기부터 참여했던MySQL 개발자가 여전히 많은 곳입니다.  
오라클은 MySQL을 인수한 이후로 개발 및 기술지원팀이 두 배 이상 증가했으며 QA / 테스트 지원팀도 3 배 증가했습니다.  
솔루션 컨설턴트, 기술지원 컨설턴트 및 엔지니어들을 통해 Oracle MySQL은 향후 릴리스에 추가할 고객의 요구 사항 및 기능 요청을 지속적으로 듣고 있습니다.  
더 나아가서 상용 고객들은 자문 위원회에 참여함으로써 시니어 개발자 및 MySQL 이사회와 만나 개발 로드맵과 향후 계획에 대해 논의할 수 있는 기회를 갖게 됩니다.

**SUMMARY**

이 백서에서는 Amazon의 RDS와 EC2 클라우드 오퍼링위헤 MySQL (MySQL Enterprise Edition) 상용 버전을 사용할때의 장점에 대해 자세히 살펴 보았습니다.  
하지만 시작 부분에서 우리는 'Database-as-a Service'(MySQL의 포크 버전을 사용하는 RDS)를 선택해야 하는지 아니면  'Infrastructure-as-a-Service'(EC2)에서 MySQL Enterprise Edition을 실행해야 하는지에 대해 질문을 하였습니다.

명확한 것는 Amazon EC2는 가장 융통성 있는 옵션을 제공하고 MySQL Enterprise Edition과 함께 제공되는 경우 특정 요구 사항을 충족시키면서 비용과 자원을 최소화하는 맞춤형 데이터베이스 인프라 환경에 대한 문을 열어줍니다.

그러나 정답은 여러분의 요구 사항을 면밀히 검토하는 것이 가장 중요합니다.  
여러분의 조직에서 최신 업계 표준 규정을 준수하는 고급 보안 기능을 필요로 합니까?  
성능을 최적화하고 제품 업데이트 관리를 돕고 클라우드 환경과 온프레미스 환경을 모두 지원해야 하는 MySQL 전문가가 필요합니까?

[MySQL비용검토(IDC vs Cloud with DB라이선스)](https://wiki.gsretail.com/pages/viewpage.action?pageId=41885448)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=41885448#page-metadata-end)

* [정관호 Kyle [X]](https://wiki.gsretail.com/display/~junggh)님이 작성, [2020-11-17](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=41885448&selectedPageVersions=1&selectedPageVersions=2)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=41885448#page-metadata-start)

**MySQL E.E 및 S.E 라이선스 비용**

**년 서브스크립션 방식으로 대상장비의 CPU상관없이 서버당 가격으로 산정**

[09. 기타](https://wiki.gsretail.com/pages/viewpage.action?pageId=25952418)

[Skip to end of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=25952418#page-metadata-end)

* 알 수 없는 사용자 (22980)님이 작성, [2020-05-25](https://wiki.gsretail.com/pages/diffpagesbyversion.action?pageId=25952418&selectedPageVersions=2&selectedPageVersions=3)에 최종 변경

[Go to start of metadata](https://wiki.gsretail.com/pages/viewpage.action?pageId=25952418#page-metadata-start)

Python CLI

DQL 이라는 Python CLI 를 통해 쉽게 접근/변경 가능

**참고 :**[**https://dql.readthedocs.io/en/latest/topics/getting\_started.html**](https://dql.readthedocs.io/en/latest/topics/getting_started.html)

관계형 모델링 가이드

**참고 :**[**https://docs.aws.amazon.com/ko\_kr/amazondynamodb/latest/developerguide/bp-modeling-nosql-B.html**](https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/bp-modeling-nosql-B.html)