

# Amazon Web Services에서의 Oracle Database 실행 모범 사례

*Abdul Sathar Sait*

2014년 12월



# 목차

요약	3
서론	3
Oracle 라이선싱 고려 사항	4
Amazon RDS 라이선스 포함	4
기존 보유 라이선스 사용(BYOL)	5
Oracle Database 를 위해 Amazon RDS 와 Amazon EC2 중 선택하기	6
보안 및 성능을 위한 설계	7
네트워크 구성	7
Amazon EC2 인스턴스 유형	9
데이터베이스 스토리지	10
Oracle AMI	12
결론	12
참고 문헌	13

## 요약

Amazon Web Services(AWS)는 호스팅된 IaaS 환경에서 Oracle Database를 실행하는 기능을 제공합니다. AWS에서의 Oracle Database 실행은 데이터 센터에서 Oracle Database를 실행하는 것과 매우 흡사합니다. 데이터베이스 관리자나 개발자에게 두 가지 환경은 차이가 없습니다. 다만 AWS에서의 Oracle Database 구현에서 최상의 결과를 얻는 데 도움이 되는 보안, 스토리지, 컴퓨팅 구성, 관리, 모니터링과 관련된 AWS 플랫폼 고려 사항이 여러 가지 있습니다. 이 백서에서는 AWS에서 Oracle Database를 실행하면서 최상의 성능, 가용성, 안정성을 달성하고 총 소유 비용(TCO)을 절감할 수 있는 모범 사례를 제시합니다. 이 백서는 AWS에서 Oracle Database를 실행하려는 데이터베이스 관리자, 엔터프라이즈 아키텍트, 시스템 관리자 및 개발자를 대상으로 합니다.

## 서론

Amazon Web Services(AWS)는 안정적이고 안전한 AWS 클라우드 인프라에 Oracle Database를 배포할 수 있는 종합적인 서비스 및 도구 세트를 제공합니다. AWS는 AWS에서 Oracle Database를 실행할 수 있는 다음 두 가지 옵션을 제공합니다.

- Oracle Database 프로비저닝 및 관리를 간소화하는 관리형 데이터베이스 서비스인 [Oracle용 Amazon Relational Database Service\(Amazon RDS\)](#) 사용. Amazon RDS로 설치, 디스크 프로비저닝, 관리, 패치 적용, 마이너 버전 업그레이드, 실패한 인스턴스 교체, 백업 및 복구 작업 자동화를 통해 클라우드상에서 간편하게 관계형 데이터베이스를 설치, 운영, 조정할 수 있습니다.

Amazon RDS의 다중 AZ 기능은 동기 복제를 사용하여 여러 가용 영역에서 2개의 데이터베이스를 운영하므로 자동 장애 조치가 포함된고가용성 환경을 만들 수 있습니다. Amazon RDS의 푸시 버튼식 조정을 통해 데이터베이스 인스턴스를 손쉽게 확장하거나 축소하여 비용 관리와 성능을 개선할 수 있습니다. Amazon RDS에서는 시간당 사용별 지불이 가능한 [라이선스 포함 옵션](#)도 제공합니다.

- Amazon Elastic Compute Cloud(Amazon EC2)에서 자체 관리형 Oracle Database 직접 실행. 이 옵션을 사용하면 인프라 설정과 데이터베이스 환경을 완벽하게 제어할 수 있습니다. Amazon EC2에서의 데이터베이스 실행은 자체 서버에서 데이터베이스를 실행하는 것과 매우 흡사합니다. 데이터베이스를 완벽히 제어할 수 있고 운영 체제 수준 액세스가 가능하므로 모니터링 및 관리 에이전트를 실행하고 데이터 복제, 백업, 복원 도구를 선택해 사용할 수 있습니다. 뿐만 아니라 Oracle Database에서 사용 가능한 모든 옵션 모듈을 사용할 수 있습니다. 다만 이 옵션을 사용하려면 Amazon EC2 인스턴스, 스토리지 볼륨, 확장성, 네트워킹, 보안 등 모든 구성 요소를 AWS 아키텍처 모범 사례에 기반하여 설정, 구성, 관리, 조정해야 합니다.

Amazon EC2에서 자체 관리형 Oracle Database를 실행하건 Oracle용 완전 관리형 Amazon RDS를 실행하건 이 백서에 설명된 모범 사례에 따르면 AWS에서의 Oracle Database 구현을 최대한 활용하는 데 도움이 됩니다. Oracle 라이선싱 옵션, Oracle Database 구현을 위한 Amazon EC2 또는 Amazon RDS 선택 시 고려 사항 및 구현에서 네트워크 구성, 인스턴스 유형, 데이터베이스 스토리지를 최적화하는 방법에 대해 설명합니다.

## Oracle 라이선싱 고려 사항

[Oracle 설명서](#)에 나온 것처럼 Oracle은 AWS에서의 Oracle Database 배포를 완벽 지원합니다. AWS에서의 Oracle Database 라이선싱은 데이터베이스가 설치되는 인스턴스 크기를 기준으로 합니다. Oracle Database 라이선싱에 대한 자세한 내용은 Oracle 웹사이트의 [Licensing Oracle Software in the Cloud Computing Environment](#)를 참조하십시오. 몇 가지 중요 사항:

- 라이선싱 목적상 Amazon EC2 인스턴스의 가상 코어 수는 물리적 코어 수와 같다고 간주합니다. 각 Amazon EC2 인스턴스 유형의 가상 코어 수를 알려면 [Amazon EC2 및 RDS DB 인스턴스 유형별 가상 코어](#)를 참조하십시오.
- Oracle Database Standard Edition은 최대 16개의 가상 코어가 있는 Amazon EC2 인스턴스에서만 라이선스를 받을 수 있습니다.
- Oracle Standard Edition One은 최대 8개의 가상 코어가 있는 Amazon EC2 인스턴스에서만 라이선스를 받을 수 있습니다.
- Oracle Database Standard Edition One 또는 Standard Edition의 경우, 가상 코어가 4개 이하인 Amazon EC2 인스턴스가 소켓 1개로 계산됩니다.
- Oracle Database Enterprise Edition의 경우, 가상 코어가 2개 이하인 Amazon EC2 인스턴스가 소켓 1개로 계산됩니다.

이 백서에 나온 Oracle 라이선싱 정책과 비용에 대한 모든 논의는 오직 정보 제공을 위한 것이며, 발행 시점에 사용할 수 있는 정보를 기준으로 한 것입니다. 보다 구체적인 정보는 자체 Oracle 라이선스 계약을 참고해야 합니다.

## Amazon RDS 라이선스 포함

라이선스 포함 서비스 모델을 사용하는 경우, Oracle Database 라이선스 비용을 Amazon RDS 서비스 시간당 요금에 포함시킬 수 있습니다. 이 경우, Oracle Database 소프트웨어 라이선스를 AWS에서 획득했기 때문에 Oracle 라이선스를 별도로 구매할 필요가 없습니다. 라이선스 포함 시간당 요금에는 소프트웨어, 기본 하드웨어 리소스 및 Amazon RDS 관리 기능이 포함됩니다. 이것은 라이선스 비용을 최적화하는 좋은 방법이며, Amazon RDS 인스턴스를 확장하거나 축소할 때 유연성을 제공합니다. 선결제 요금이나 장기 계약 없이 시간당 비용을 활용할 수 있습니다. 또한 1년 또는 3년 예약 조건으로 Amazon RDS 예약 인스턴스를 구입할 수도 있습니다. 예약 인스턴스 방식을 선택하면 각각의 데이터베이스 인스턴스에 대해 저렴한 요금을 일시불로 선결제 한 다음 대폭 할인된 시간당 사용 요금을 지불할 수 있습니다.

**참고:** Amazon RDS에서 라이선스 포함 모델의 시간당 라이선스는 Oracle Standard Edition One에서만 이용할 수 있습니다. Amazon RDS의 다른 Oracle Database 에디션과 Amazon EC2의 모든 Oracle Database 에디션에서는 다음 섹션에 설명된 것처럼 기존 보유 라이선스를 사용해야 합니다(즉, Oracle에서 라이선스 획득).

Amazon RDS 서비스를 사용하는 시간에 대해서만 Oracle 라이선스 비용을 지불하기 때문에 업무 시간 중에만 활성화되는 개발 및 테스트 환경의 전체 비용을 줄이는 데는 라이선스 포함 옵션이 도움이 될 수 있습니다. 대부분의 기업에서는 주당 총 업무 시간( $10 \times 5 = 50$ 시간)이 주당 총 시간( $24 \times 7 = 168$ 시간)의 약 30%에 불과하므로 이 서비스 모델로 상당한 액수를 절감할 수 있습니다.

이 서비스 모델은 인스턴스 비용에 라이선스가 포함되므로 필요에 따라 인스턴스 크기를 조정할 수 있는 유연성도 제공합니다. 평소의 용량 요구 사항이 주기적이고 예측 가능한 급증보다 훨씬 작은 경우, 이 서비스 모델을 통해 필요한 추가 용량을 소화하도록 확장하고 그렇지 않을 때는 축소하여 비용을 절감할 수 있습니다. 예를 들어 한달 중 마지막 3일을 제외한 대부분의 날에는 db.m3.large 인스턴스의 성능이 필요한 데이터베이스가 있을 수 있습니다. 한달 중 마지막 3일 중에는 급여 처리와 월간 마감 때문에 데이터베이스 사용이 크게 늘어날 수 있습니다. 이 시나리오에서는 한달 동안 db.m3.large 인스턴스 유형을 기준으로 Amazon RDS에서 Oracle Database를 사용하고 마지막 3일 동안은 db.m3.2xlarge로 확장했다가 다시 축소할 수 있습니다. 이렇게 하면 한달 내내 db.m3.2xlarge 인스턴스를 사용하는 경우에 비해 65% 이상 비용을 절감할 수 있습니다.

## 기존 보유 라이선스 사용(BYOL)

자체 Oracle Database 라이선스를 이미 보유하고 있다면 BYOL 서비스 모델을 사용하여 Amazon RDS에서 Oracle 데이터베이스를 실행할 수 있습니다. 이렇게 하면 Oracle 라이선스 비용이 포함되지 않으므로 Amazon RDS 인스턴스 비용이 저렴해집니다. BYOL 모델은 기존 Oracle Database 라이선스를 사용하거나 Oracle에서 직접 새 라이선스를 구매하기를 선호하는 고객들을 위해 만들어졌습니다.

Amazon RDS에 Oracle Database Enterprise Edition 또는 Standard Edition을 사용하거나 Amazon EC2에서 자체 관리형 Oracle Database를 실행하려는 경우, BYOL이 유일하게 지원되는 옵션입니다.

## Oracle 라이선스의 AWS 이전

구체적인 라이선스 계약 조건에 따라 Oracle 라이선스를 AWS로 이전할 수도 있습니다. 다시 말해 기존 라이선스를 이전하여 AWS에서 사용할 수 있습니다. 여기에는 다음이 포함됩니다.

- 서버 기반 라이선스(사용되는 CPU 기준)
- ELA(Enterprise License Agreement)

- ULA(Unlimited License Agreement)
- BPO(Business Process Outsourcing) 라이선스
- OPN(Oracle PartnerNetwork) 라이선스
- Named User Plus 라이선스

AWS로 이전되는 라이선스에는 추가 조건이나 제한(가능한 비용 포함)이 적용될 수 있습니다. 추가 정보와 제한은 구체적인 라이선스 계약을 확인하십시오.

Amazon RDS의 Oracle Database와 Amazon EC2의 Oracle Database에 적용되는 Oracle 라이선스는 비슷하지만 시간당 라이선스는 Amazon RDS에서만 사용할 수 있습니다.

## Oracle Database를 위해 Amazon RDS와 Amazon EC2 중 선택하기

Amazon RDS와 Amazon EC2 모두 Oracle Database를 실행할 수 있지만 장점은 서로 다릅니다. Amazon RDS는 Amazon EC2에서 Oracle Database를 실행하는 것보다 설정, 관리, 유지가 더 쉽고, Oracle Database의 일상적 관리 외의 업무에 집중할 수 있습니다. 그 대신 Amazon EC2에서 Oracle Database를 실행하면 제어, 유연성, 선택의 폭이 커집니다. 애플리케이션과 요구 사항에 따라 두 가지 중 하나를 선호할 수 있습니다.

여러 Oracle 데이터베이스를 AWS로 이전하는 경우, 어떤 데이터베이스는 Amazon RDS에 적합하고 어떤 데이터베이스는 Amazon EC2에서 직접 실행하는 것이 적합할 수 있습니다. 많은 AWS 고객들은 Oracle Database 워크로드에 Amazon RDS와 Amazon EC2의 조합을 사용합니다.

Amazon RDS가 보다 나은 선택일 수 있는 경우는 다음과 같습니다.

- 비즈니스와 애플리케이션에 집중하고 데이터베이스 프로비저닝, 백업 및 복구 작업 관리, 보안 패치 관리, 마이너 Oracle 버전 업그레이드, 스토리지 관리 같은 과중한 업무는 AWS가 처리하기를 원하는 경우.
- 고가용성 데이터베이스 솔루션이 필요하고 스탠바이 데이터베이스를 수동으로 설정하고 유지 관리할 필요 없이 Amazon RDS가 제공하는 푸시 버튼식 동기 Multi-AZ 복제를 활용하기를 원하는 경우.
- Oracle Database Standard 또는 Standard One 에디션의 고가용성을 위해 스탠바이 인스턴스에 대한 동기 복제를 원하는 경우.
- 대규모 선행 투자 대신 시간당 인스턴스 비용의 일부로 Oracle 라이선스 비용을 지불하기를 원하는 경우.
- 데이터베이스 크기가 2.5TB 미만이고 최대 IOPS 수요가 30,000 미만인 경우.

- 데이터베이스 백업과 (가장 중요하게는) 특정 시점으로의 복구를 관리하기 원치 않는 경우.
- 데이터베이스의 일상적 관리보다는 성능 튜닝 및 스키마 최적화 같은 상위 수준 작업에 집중하려는 경우.
- 관련된 라이선싱 및 복잡성에 신경쓰지 않고 워크로드 패턴을 기준으로 인스턴스를 확장 및 축소하려는 경우.

Amazon EC2가 보다 나은 선택일 수 있는 경우는 다음과 같습니다.

- **Sys/System** 사용자 액세스를 포함하여 데이터베이스에 대한 완전한 제어가 필요하거나 운영 체제 수준에서의 액세스가 필요한 경우.
- 데이터베이스 크기가 현재 Amazon RDS에서 가능한 최대 데이터베이스 크기(백서 발행 시점 기준 3TB)의 80%를 초과하는 경우.
- Oracle Enterprise Manager 또는 그 밖의 타사 도구를 사용하여 데이터베이스 인스턴스를 관리해야 하는 경우.
- 현재 Amazon RDS가 지원하지 않는 Oracle 기능이나 옵션을 사용해야 하는 경우.
- 데이터베이스 IOPS 수요가 30,000 이상인 경우.
- Amazon RDS가 지원하지 않는 특정 Oracle Database 버전이 필요한 경우.

## 보안 및 성능을 위한 설계

Oracle Database를 Amazon RDS에서 실행하건 Amazon EC2에서 실행하건 인프라의 모든 구성 요소를 최적화하면 보안, 성능, 안정성이 향상됩니다. 다음 섹션에서는 AWS에서 Oracle Database를 구현할 때 네트워크 구성, 인스턴스 유형, 데이터베이스 스토리지를 최적화할 수 있는 모범 사례를 살펴봅니다.

### 네트워크 구성

Amazon Virtual Private Cloud(VPC)를 통해 계정 전용인 AWS 클라우드에서 논리적으로 격리된 섹션을 프로비저닝할 수 있습니다. IP 주소 범위 선택, 서브넷 생성, 라우팅 테이블 및 네트워크 게이트웨이 구성, 보안 설정 등 가상 네트워킹 환경을 완벽하게 제어할 수 있습니다.

*서브넷*은 Amazon VPC에서의 IP 주소 범위입니다. 선택한 서브넷으로 AWS 리소스를 시작할 수 있습니다. 인터넷에 연결되어야 하는 리소스에는 퍼블릭 서브넷을 사용하고, 인터넷에 연결되지 않는 리소스에는 프라이빗 서브넷을 사용하십시오.

각 서브넷의 AWS리소스를 보호하기 위해 보안 그룹 및 네트워크 액세스 통제 목록(ACL)을 비롯한 여러 보안 계층을 사용할 수 있습니다.

다음 표에서는 보안 그룹과 네트워크 ACL의 근본적인 차이를 설명합니다.

보안 그룹	네트워크 ACL
인스턴스 수준에서 적용(1차 보안 계층)	서브넷 수준에서 적용(2차 보안 계층)
허용 규칙만 지원	허용 규칙 및 거부 규칙 지원
상태 저장: 규칙에 관계없이 반환 트래픽이 자동으로 허용됨	상태 비저장: 반환 트래픽이 규칙에 의해 명시적으로 허용되어야 함
트래픽 허용 여부를 결정하기 전에 모든 규칙을 평가	트래픽 허용 여부 결정 시 규칙을 번호순으로 처리
인스턴스 시작 시 누군가 보안 그룹을 지정하거나 나중에 보안 그룹을 인스턴스와 연결하는 경우에만 인스턴스에 적용됨	연결된 서브넷에서 모든 인스턴스에 자동 적용됨(백업 보안 계층이므로 보안 그룹을 지정하는 사람에게 의존할 필요 없음)

Amazon VPC는 격리, 추가 보안 및 Amazon EC2 인스턴스를 서브넷으로 분리하는 기능을 제공하며, 프라이빗 IP 주소 사용을 허용합니다. 데이터베이스 구현에서는 이 모두가 중요합니다. 프라이빗 서브넷에 Oracle Database 인스턴스를 배포하고 Amazon VPC 내의 애플리케이션 서버 또는 Amazon VPC 내의 Bastion 호스트만 데이터베이스 인스턴스에 액세스하도록 허용합니다. 지정된 포트를 통해 특정 IP 주소에 대한 액세스만 허용하도록 적절한 보안 그룹을 생성합니다. 이러한 권장 사항은 Amazon RDS를 사용하건 Amazon EC2를 사용하건 상관없이 Oracle Database에 적용됩니다.

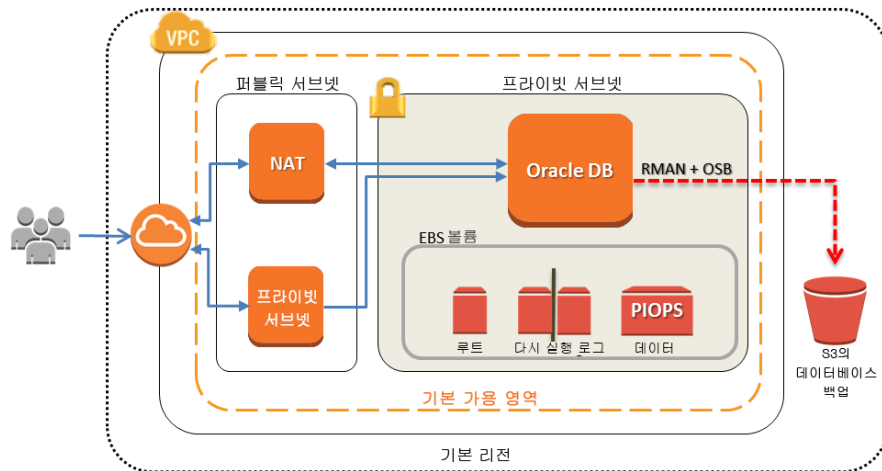


그림 1: Amazon VPC의 프라이빗 서브넷에 있는 Oracle Database



## Amazon EC2 인스턴스 유형

AWS에는 사용할 수 있는 수많은 Amazon EC2 인스턴스 유형이 있으므로 워크로드에 가장 적합한 인스턴스 유형을 선택할 수 있습니다. 하지만 사용할 수 있는 모든 인스턴스 유형이 Oracle Database 실행에 가장 적합한 것은 아닙니다.

Oracle Database에 Amazon RDS를 사용하는 경우, AWS는 모범 사례에 기반하여 일부 인스턴스 유형을 필터링하고 다음 옵션을 제공합니다.

- db.t1.micro
- db.m1.small
- db.m3.medium
- db.m3.large
- db.m3.xlarge
- db.m3.2xlarge

엔터프라이즈 데이터베이스 워크로드에는 db.m3 기반 Amazon RDS 인스턴스를 선택하는 것이 좋습니다. Amazon RDS 인스턴스 유형을 선택할 때는 데이터베이스 워크로드와 사용할 수 있는 Oracle Database 라이선스가 기준이 되어야 합니다.

Amazon EC2에서 자체 관리형 데이터베이스를 실행하는 경우, 사용할 수 있는 Amazon EC2 인스턴스 유형의 선택의 폭이 넓어집니다. 이것은 흔히 사용자가 Amazon RDS 대신 Amazon EC2에서 Oracle Database를 실행하는 이유 중 하나입니다. Oracle Database는 CPU 사용량 면에서 리소스 집약적이기 때문에 아주 작은 인스턴스 유형은 적합하지 않습니다. 메모리 공간이 이보다 큰 인스턴스는 캐싱을 향상시키고 더 큰 시스템 글로벌 영역(SGA)을 제공하여 데이터베이스 성능 개선에 도움이 됩니다. 메모리와 CPU가 적절히 균형을 이룬 인스턴스를 선택하는 것이 좋습니다. 사용하려는 Oracle Database 라이선스 및 구현하려는 아키텍처에 일치하는 인스턴스 유형을 선택하십시오. 비즈니스 요구에 가장 적합한 아키텍처는 [AWS에서의 Oracle Database 고급 아키텍처](#) 백서를 참조하십시오.

Oracle Database는 읽기/쓰기 작업에 디스크 스토리지를 많이 사용하므로 Amazon Elastic Block Store(Amazon EBS)에 최적화된 인스턴스만을 사용하는 것이 좋습니다. Amazon EBS에 최적화된 인스턴스는 Amazon EC2와 Amazon EBS 사이에서 전용 처리량을 전달합니다. 대역폭과 스토리지 하위 시스템에 대한 처리량은 좋은 데이터베이스 성능에 매우 중요하므로 보다 좋은 데이터베이스 성능을 위해서는 네트워크 성능이 더 높은 인스턴스를 선택하십시오.

Amazon EC2에서 Oracle Database를 실행하는 데 가장 적합한 인스턴스 유형은 다음과 같습니다. 라이선싱 목적상 각 인스턴스 유형의 가상 코어를 결정하려면 [가상 코어 표](#)를 참조하십시오.

인스턴스 유형	vCPU	메모리	인스턴스 스토리지
m3.xlarge	4	15	2 x 40 SSD
m3.2xlarge	8	30	2 x 80 SSD
c3.4xlarge	16	55	2 x 160 SSD
r3.large	2	15.25	1 x 32 SSD
r3.xlarge	4	30.5	1 x 80 SSD
r3.2xlarge	8	61	1 x 160 SSD
r3.4xlarge	16	122	1 x 320 SSD
r3.8xlarge	32	244	2 x 320 SSD
i2.xlarge	4	30.5	1 x 800 SSD
i2.2xlarge	8	61	2 x 800 SSD
i2.4xlarge	16	122	4 x 800 SSD
i2.8xlarge	32	244	8 x 800 SSD

## 데이터베이스 스토리지

데이터 스토리지로는 일반적으로 Amazon EBS를 사용합니다. 일부 초고성능 아키텍처에는 인스턴스 스토리지 SSD를 사용할 수 있지만 안정적인 지속성을 위해서는 Amazon EBS 스토리지를 추가해야 합니다. 이 아키텍처의 세부 사항은 [Amazon EC2에서의 Oracle Database 고급 아키텍처](#) 백서를 참조하십시오.

높고 일관된 IOPS 및 데이터베이스 성능을 위해서는 General Purpose(GP2) 볼륨이나 Provisioned IOPS(PIOPS) 볼륨을 사용하는 것이 좋습니다. GP2 및 PIOPS 볼륨은 Amazon EC2와 Amazon RDS에서 사용할 수 있습니다. GP2는 볼륨당 최대 3,000 IOPS를 제공할 수 있고, PIOPS는 볼륨당 최대 4,000 IOPS를 제공할 수 있습니다. GP2 볼륨은 대부분의 데이터베이스 요구에 탁월한 비용과 성능의 균형을 제공합니다. GP2가 제공할 수 있는 것보다 높은 IOPS가 데이터베이스에 필요한 경우, PIOPS 볼륨을 선택하는 것이 좋습니다.

PIOPS 볼륨의 경우, 볼륨 생성 시 IOPS 비율을 지정하면 Amazon EBS는 지정된 연도 중 99.9%의 시간 동안 Provisioned IOPS 성능의 10% 이내를 제공합니다. Provisioned IOPS(SSD) 볼륨의 크기는 4GB에서 1TB까지 될 수 있고, 볼륨당 최대 4,000 IOPS를 프로비저닝할 수 있습니다. 프로비저닝되는 IOPS 대 필요한 볼륨 크기의 비율은 최대 30입니다. 예를 들어 3,000 IOPS의 볼륨을 얻으려면 볼륨 크기가 100GB 이상이 되어야 합니다.

GP2 볼륨도 PIOPS 볼륨과 비슷하게 SSD 기반이지만 GP2 볼륨에서 얻는 IOPS는 기준 IOPS부터 볼륨당 최대 버스트 가능한 3,000 IOPS까지 달라질 수 있습니다. 이는 대부분의 데이터베이스 워크로드에 매우 효과적인데, 데이터베이스에서 필요로 하는 IOPS 성능은 부하와 실행되는 쿼리의 양에 따라 일정 기간 동안 여러 번 달라지기 때문입니다.

범용(SSD) 볼륨 성능은 볼륨 크기에 따라 정해지고 볼륨의 기준 성능 수준 및 I/O 크레딧이 얼마나 빨리 누적되는지를 결정합니다. 볼륨이 클수록 기준 성능 수준이 높아지고 I/O 크레딧이 빨리 누적됩니다. I/O 크레딧이란 기준 성능 이상이 필요할 때 대용량 I/O를 버스트하는 데 **General Purpose(SSD)** 볼륨이 사용할 수 있는 가용 대역폭을 의미합니다. 볼륨에 I/O 크레딧이 많을수록 더 오랜 기간 동안 볼륨이 기준 성능 수준 이상으로 버스트할 수 있고 더 높은 성능이 필요할 때 더 좋은 성능을 발휘할 수 있습니다.

데이터베이스에 일관되게 필요한 IOPS 양을 추산하여 그 정도의 IOPS를 얻기에 충분한 GP2 스토리지를 할당하는 것이 좋습니다. 주기적 급증에 필요한 추가적인 IOPS는 가용 크레딧에 기반한 버스트 성능으로 충당해야 합니다. Oracle Database의 IOPS 요구를 추산하는 방법은 [AWS에서 Oracle Database의 IOPS 요구 파악](#) 백서를 참조하십시오.

볼륨의 버스트 지속 시간은 볼륨의 크기, 필요한 버스트 IOPS 및 버스트가 시작될 때의 크레딧 잔고에 의해 결정됩니다. 볼륨 성능이 자주 기준 수준 한도에 도달하는 경우(빈 I/O 크레딧 잔고로 인해) 더 큰 **General Purpose(SSD)** 볼륨(기준 성능 수준이 보다 높은)을 사용하거나 IOPS 성능을 3,000 IOPS 이상으로 유지해야 하는 워크로드인 경우, **Provisioned IOPS(SSD)** 볼륨으로 전환하는 것을 고려해야 합니다. GP2 볼륨에 대한 더 자세한 내용은 [Amazon EBS 사용 설명서](#)를 참조하십시오.

Amazon RDS의 경우, **General Purpose(SSD)** 스토리지는 프로비저닝된 GB당 일관되게 3 IOPS를 제공하며 최대 3,000 IOPS까지 버스트할 수 있습니다. Amazon RDS용 **Magnetic Storage**를 이미 사용하고 있다면 **General Purpose(SSD)** 스토리지로 전환할 수 있지만 그 과정에서 잠깐 가용성에 영향이 있을 수 있습니다. **Provisioned IOPS**를 사용하면 최대 3TB의 스토리지와 데이터베이스 인스턴스당 30,000 IOPS를 프로비저닝할 수 있습니다. m2.4xlarge 인스턴스에서 실행되는 쓰기 50%와 읽기 50%로 이루어진 워크로드의 경우, Oracle Database에서 최대 25,000 IOPS를 실현할 수 있습니다. 하지만 이 한도 이상을 프로비저닝하면 지연 시간을 낮추고 처리량을 높일 수 있습니다. 데이터베이스 워크로드, 인스턴스 유형, 데이터베이스 엔진 선택에 따라 프로비저닝한 양과 실제로 실현되는 IOPS가 달라질 수 있습니다. *Amazon RDS 사용 설명서*의 [실현되는 IOPS 속도에 영향을 미치는 요인](#)을 참조하십시오.

Amazon EC2의 Oracle Database에서 IOPS와 용량을 늘리려면 여러 볼륨을 함께 스트라이핑하십시오. 서로 다른 데이터 파일에 여러 Amazon EBS 볼륨을 개별적으로 사용할 수도 있지만 이들을 함께 스트라이핑하면 밸런싱과 확장성을 개선할 수 있습니다. 스트라이핑에 **Oracle Automatic Storage Management(ASM)**를 사용할 수 있습니다. 데이터 파일, 로그 파일, 이진수를 별도의 Amazon EBS 볼륨에 두고, 정기적으로 로그 파일 볼륨의 스냅샷을 생성하십시오. 로컬 SSD 스토리지가 있는 인스턴스 유형을 선택하면 **Smart Flash Cache**를 사용하고(운영 체제가 Oracle Linux인 경우) 임시 파일 및 테이블 공간에 로컬 스토리지를 사용하여 데이터베이스 성능을 높일 수 있습니다.

대부분의 Oracle Database 사용자는 정기적인 핫 백업 및 콜드 백업을 생성합니다. 콜드 백업은 데이터베이스가 종료되어 있을 때 생성되는 반면 핫 백업은 데이터베이스가 활성화되어 있을 때 생성됩니다. 높은 내구성과 간편한 액세스를 위해서는 핫 백업과 콜드 백업을 Amazon Simple Storage Service(Amazon S3)에 저장하십시오. AWS Storage Gateway 또는 Oracle Secure Backup Cloud Module을 사용하면 Amazon S3에 데이터베이스를 직접 백업할 수 있습니다. Amazon S3에서 수명 주기 정책을 사용하여 오래된 백업을 Amazon Glacier로 옮겨 보관할 수 있습니다.

## Oracle AMI

Amazon 머신 이미지(AMI)는 클라우드상의 가상 서버인 인스턴스를 시작하는 데 필요한 정보를 제공합니다. 인스턴스를 시작할 때 AMI를 지정해야 하며, AMI에서 필요한 만큼 많은 인스턴스를 시작할 수 있습니다.

Oracle은 AWS에서 Oracle Database를 비롯한 일부 Oracle 제품의 공식 AMI를 정기적으로 제공합니다. 하지만 Oracle이 제공하는 데이터베이스 AMI가 늘 최신 버전인 것은 아닙니다. 뿐만 아니라 Oracle이 제공하는 AMI는 Oracle Linux 운영 체제에 기반하기 때문에 사용자의 운영 체제에 따라서는 최선의 옵션이 아닐 수도 있습니다.

Amazon EC2에 Oracle Database를 설치하고 사용하기 위해 Oracle이 제공하는 AMI를 꼭 사용할 필요는 없습니다. 운영 체제 AMI를 사용하여 Amazon EC2 인스턴스를 시작한 다음 물리적 서버와 똑같이 Oracle 웹사이트에서 Oracle Database 소프트웨어를 다운로드하고 설치할 수 있습니다. AWS에서의 Oracle 워크로드에 권장되는 운영 체제는 [Amazon EC2에서 Oracle 워크로드 운영 체제 선택](#) 백서를 참조하십시오.

필요한 모든 Oracle 소프트웨어가 포함된 첫 번째 환경을 설정하고 나면 후속 설치를 위한 사용자 지정 AMI를 만들 수 있습니다. [AWS Marketplace](#)에서 직접 AMI를 시작할 수도 있습니다. 타사가 제공하는 커뮤니티 AMI는 사용하기 전에 보안 및 안정성을 면밀히 검토해야 합니다. AWS는 이러한 AMI의 보안이나 안정성에 대해 책임을 지지 않습니다.

## 결론

사용 시나리오에 따라 Oracle Database용 Amazon RDS를 사용하거나 Amazon EC2에서 자체 관리형 Oracle Database를 실행할 수 있습니다. 무엇을 선택하든 이 백서에 나온 모범 사례에 따르면 AWS에서의 Oracle Database 구현에서 최선의 결과를 얻는 데 도움이 될 것입니다.

## 참고 문헌

AWS에서의 Oracle 워크로드 실행에 대한 추가 정보는 다음 리소스를 참조하십시오.

AWS에서의 Oracle Database:

- Amazon EC2에서의 Oracle Database 고급 아키텍처  
<http://d0.awsstatic.com/enterprise-marketing/Oracle/AWSAdvancedArchitecturesforOracleDBonEC2.pdf>
- Oracle Database를 AWS로 마이그레이션하는 전략  
<http://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf>
- Amazon EC2에서 Oracle 워크로드 운영 체제 선택  
<http://d0.awsstatic.com/whitepapers/choosing-os-for-oracle-workloads-on-ec2.pdf>
- AWS에서 Oracle Database의 IOPS 요구 파악  
<http://d0.awsstatic.com/whitepapers/determining-iops-needs-for-oracle-database-on-aws.pdf>
- AWS 사례 연구: Amazon.com Oracle DB Backup to Amazon S3  
<http://aws.amazon.com/solutions/case-studies/amazon-oracle/>

AWS에서의 Oracle

- <http://aws.amazon.com/oracle/>
- <http://aws.amazon.com/rds/oracle/>

AWS FAQ에서의 Oracle

- <http://www.oracle.com/technetwork/topics/cloud/faq-098970.html>

AWS 테스트 드라이브에서의 Oracle

- <http://aws.amazon.com/solutions/global-solution-providers/oracle/labs/>

AWS에서의 Oracle 라이선싱

- <http://www.oracle.com/us/corporate/pricing/cloud-licensing-070579.pdf>

Oracle RMAN 백업 및 Amazon S3 시작하기:

- <http://aws.amazon.com/backup-recovery/getting-started/>

## AWS 서비스 세부 정보

- <http://aws.amazon.com/products/>
- <http://aws.amazon.com/documentation/>
- <http://aws.amazon.com/whitepapers/>

## AWS 요금 정보:

- <http://aws.amazon.com/pricing/>
- <http://calculator.s3.amazonaws.com/index.html>

© 2014, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

**고지 사항**

이 문서는 정보 제공 목적으로만 제공됩니다. 본 문서의 발행일 기준으로 AWS 제품 및 실행방법을 설명하며, 예고 없이 변경될 수 있습니다. 고객은 본 문서에 포함된 정보나 AWS 제품 또는 서비스의 사용을 독립적으로 평가할 책임이 있으며, 각 정보 및 제품은 명시적이든 묵시적이든 어떠한 종류의 보증 없이 "있는 그대로" 제공됩니다. 본 문서는 AWS, 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 보증, 표현, 계약 약속, 조건 또는 보증을 구성하지 않습니다. 고객에 대한 AWS의 책임 및 의무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.