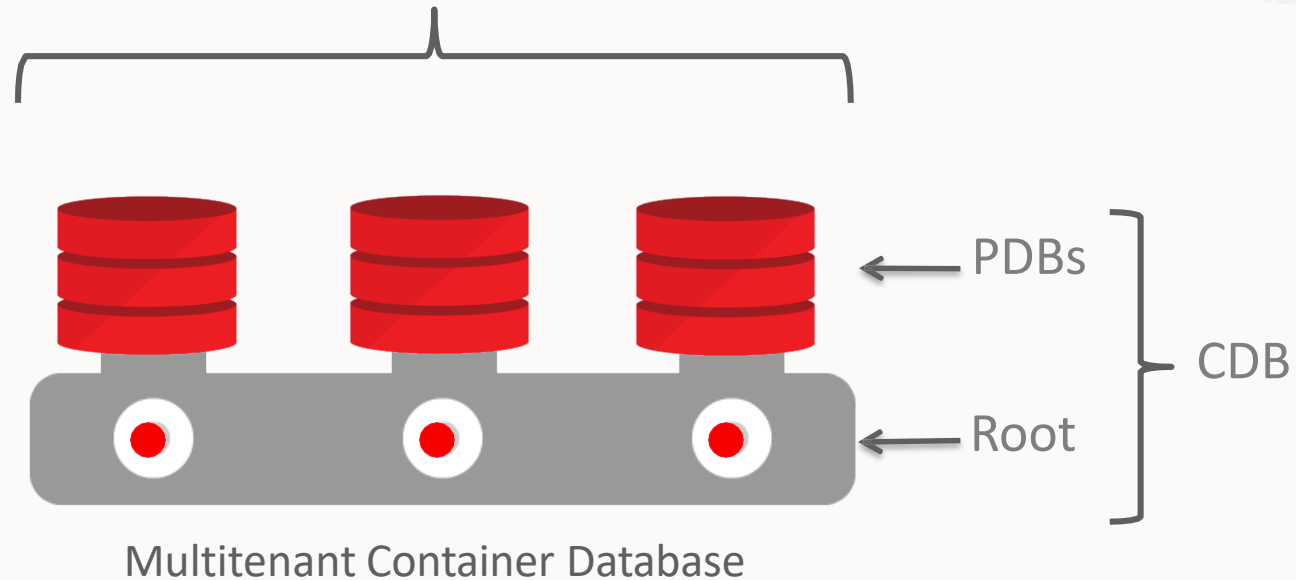


# Multitenant 개념

경북산업직업전문학교 [1-SEP-2022]  
- 무단배포를 금지합니다 -

# What is Multitenant?

- Up to 4k PDBs per CDB



- Oracle 12c에 소개된 DB Option
- 하나의 CDB안에 여러 개의 PDB를 포함 → DB consolidation에 유용

# Database Consolidation **Business Goals**



- **Reduce Costs**

- 여러개의 작은 시스템을 몇개의 하나 또는 소수의 시스템으로 통합
- 자원 공유에 따른 자원 효율성 증가

- **Simplify Operations**

- 시스템 관리 단순화
- 데이터 베이스 표준화

- **Improve Security**

- 하나의 표준화된 플랫폼을 통한 통일된 보안체계적용

# Multitenant Key Benefits

<b>Minimize CapEx</b>	<ul style="list-style-type: none"><li>• DB 통합에 따른 자원공유로 비용 최소화</li></ul>
<b>Minimize OpEx</b>	<ul style="list-style-type: none"><li>• 여러개의 DB를 하나의 DB처럼 관리</li><li>• 관리 표준화</li></ul>
<b>Maximize Agility</b>	<ul style="list-style-type: none"><li>• “pluggability” 를 통한 빠른 provisioning</li><li>• Snapshot cloning (개발 테스트 서버)</li></ul>
<b>Easy</b>	<ul style="list-style-type: none"><li>• To Adopt: 어플리케이션 변화 없이 통합</li><li>• To Use: SQL로 PDB간 interface</li></ul>

# Multitenant 대상업무

## IT비용절감

### Consolidation을 통한 IT 인프라 절감

- 스토리지/서버이용 절감-전기비용,상면 공간절감, 효율적인 자원 사용

### Manage Many as One

-컨테이너 데이터베이스 하나만 관리, 패치,백업 단순화, 운영비용 절감

다수의 DB를 보유하고 이를 관리하기  
위한 시간과 비용이 많이 소요되는 고객

업무가 다양하고 그에 따른 여러가지  
DB를 운영하고 있는 고객

## 유연성 증대

### 비즈니스 요구 사항에 대한 빠른 대응

- 신속한 데이터베이스 프로비저닝  
- PDB 간 Hot clone, Snap clone ( On-prem ↔ cloud )

### Designed for the Cloud

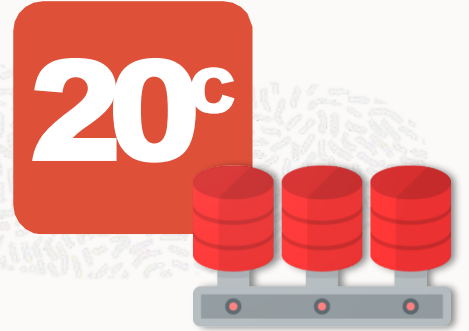
- 시스템 자원을 효율적으로 사용하면서도 어플리케이션 변경없이  
클라우드에 다수의 데이터베이스를 Consolidation

신속한 비즈니스 요구 적용이 필요한  
고객

여러개의 개발 ,테스트 DB를 보유해야만  
하는 고객

운영은 On-prem 이지만 Cloud DR  
구축을 고려하는 고객

# 지원 종료 : Non-Container Database Architecture



- Oracle Database 21C 이후부터는 Non-CDB 지원하지 않음
- Oracle Database 19C 이상에서 **PDB 3개** 까지 무료 지원

# Database Consolidation Challenges

- **Availability**

- 시스템 통합에 따른 Single point failure로 인한 잠재적 risk가 더 커질수 있음
- 각 DB별 유지보수 일정 표준화의 어려움

- **Performance**

- 데이터베이스의 성능저하 위험성 증가
- 시스템 리소스에 대한 수요증가(CPU, I/O)
- Noise Neighbor

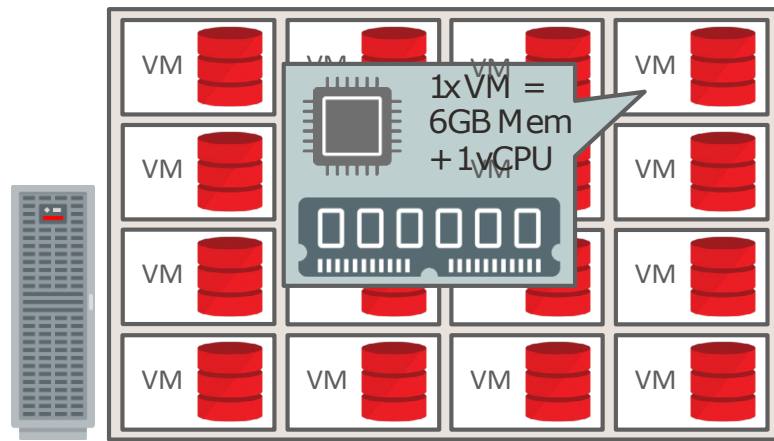
# Why is isolation Required?

물리적인 위치	PROD vs DR , 사용자 위치, Network Bandwidth
DBA 역할	Multiple DBA Teams
보안 분리	Quarantine Networks, 민감 데이터 관리
유지보수	데이터베이스 버전, upgrade 스케줄
장애	장애 범위
자원관리	Noisy Neighbor 관리

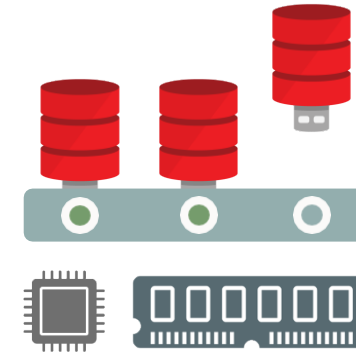


# Comparing Database Consolidation Architectures

Isolation  
& Agility



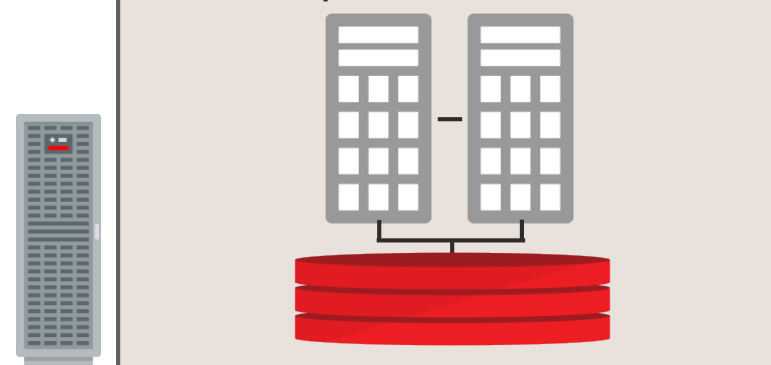
Multitenant



Stacked Databases



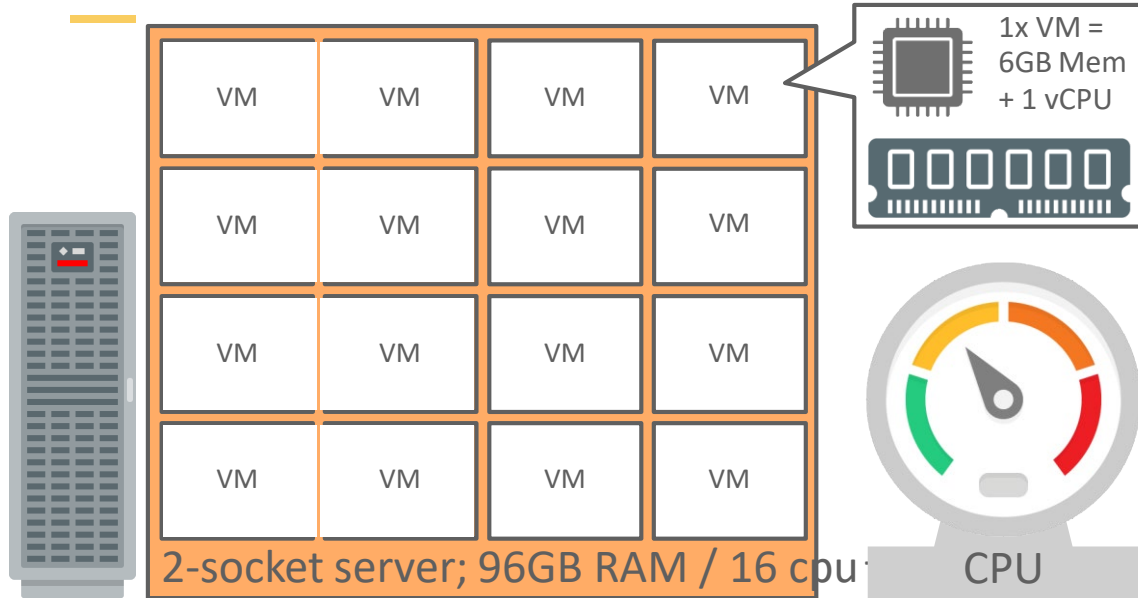
Schema/Row Consolidation



Efficiency

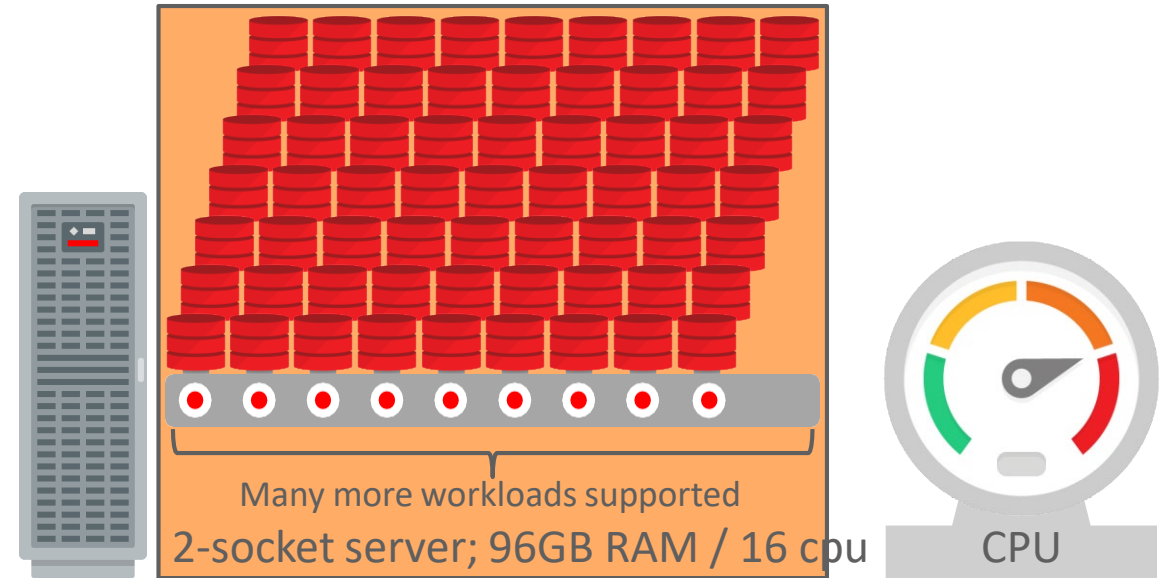
# VM-base Model과 자원사용비교

## VM-Based Model



- VM 별 Fixed된 공간을 차지 하고 있음
- Workload가 적은 VM은 자원이 남지만 Workload가 큰 VM은 자원 부족
- CPU 사용율은 낮고 메모리만 부족한 경우 → 서버를 추가해야 하므로 CPU 증가에 따른 SW 라이선스 비용 증가

## Multitenant Model (same server)

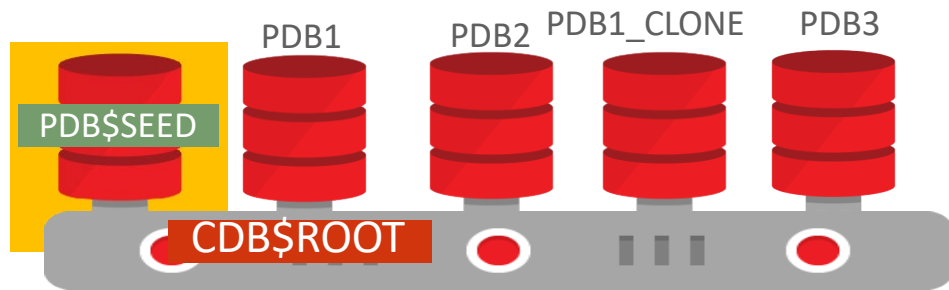


- PDB들이 자원을 공유 하고 있어 VM보다 많은 PDB 생성 가능
- 자원 공유로 workload 차이에 따른 자원 부족 현상 해결 가능

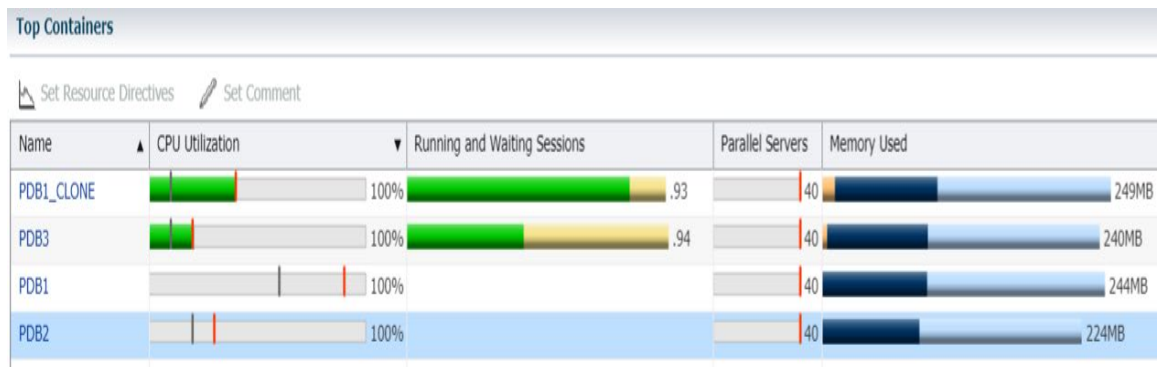
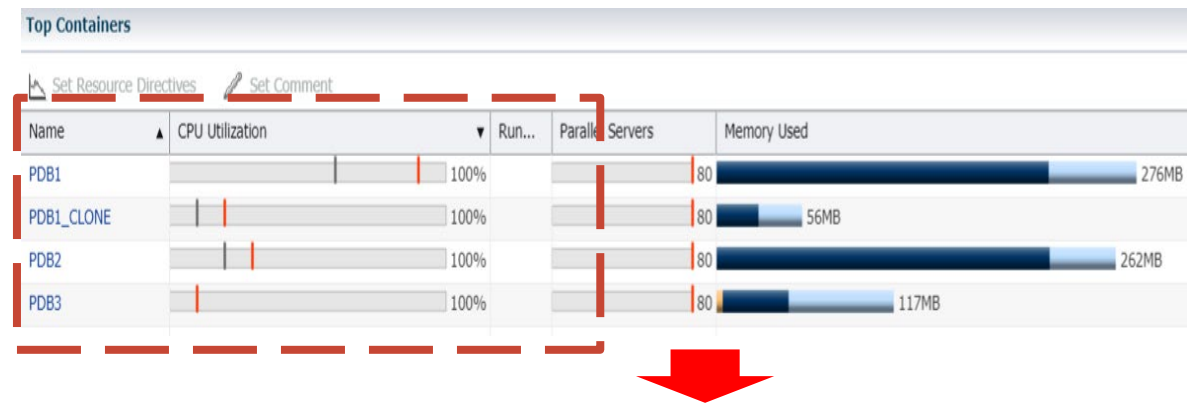
# Pluggable Database - Managing resource

CDB 레벨에서 복수개의 PDB의 CPU 리소스 할당 및 제한.

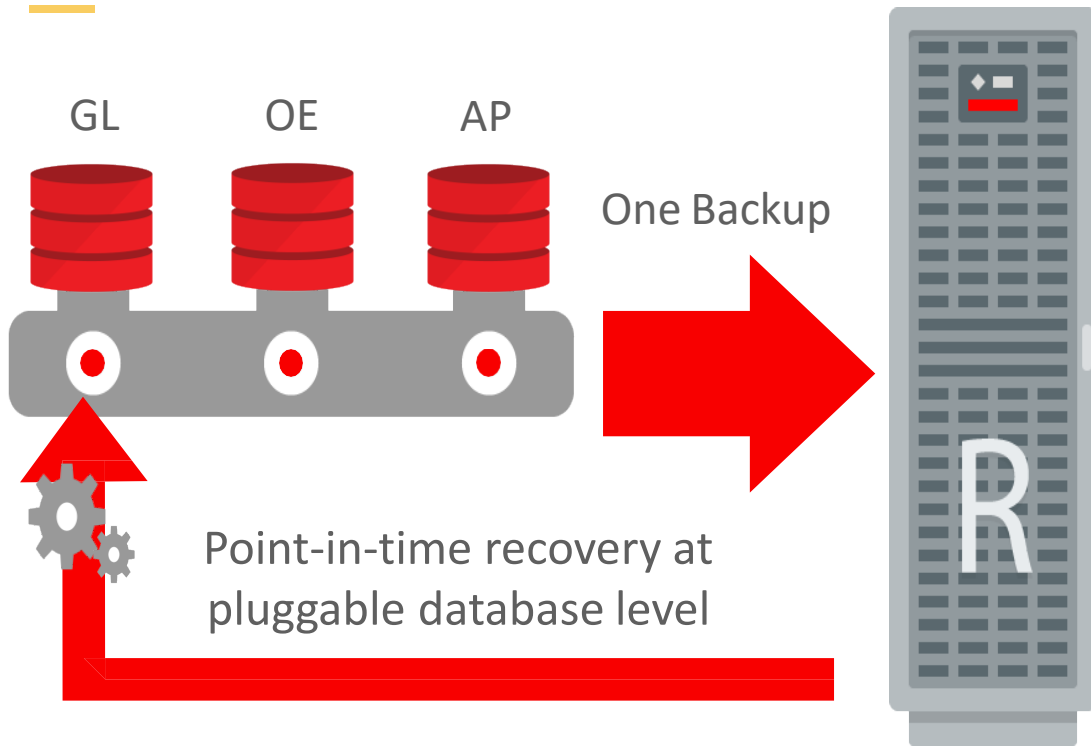
리소스 매니저의 설정 중에 Share 수치에 의해서 전체 pdb 에서 보장받는 cpu 비율 설정.



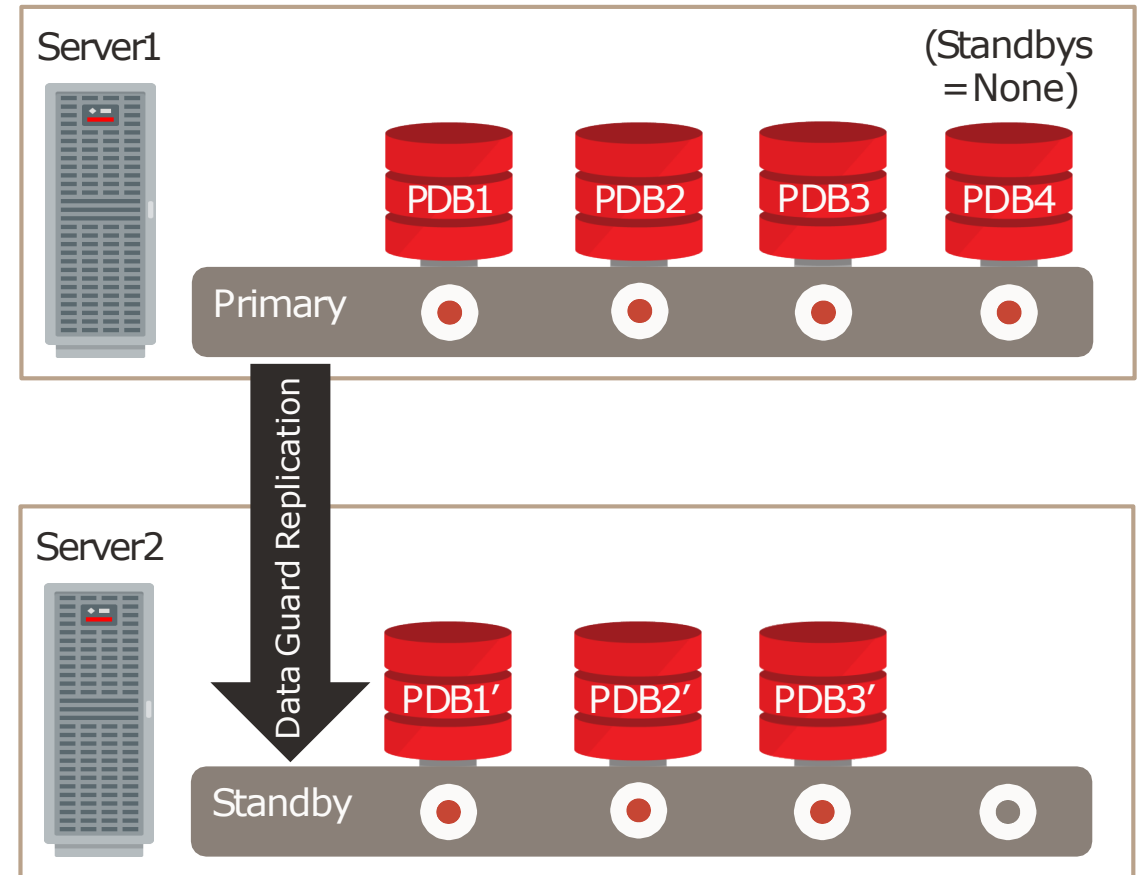
Pluggable database	Shares	Guaranteed CPU	Maximum CPU
PDB1	6	6/10 =60%	90%
PDB2	2	2/10 =20%	30%
PDB1_CLONE	1	1/10 =10%	40%
PDB3	1	1/10 =10%	20%



# 활용: Manage Many database as One with Multitenant



- CDB level로 한번에 여러 PDB 백업
- PDB별 복구 가능



- CDB level 별 ADG 구성

# 활용: Manage Many database as One with Multitenant

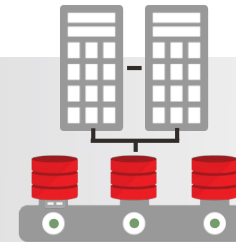
**GOLD**

RAC, Data Guard



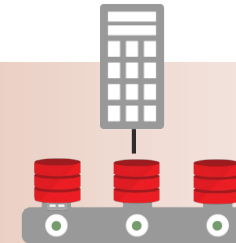
**SILVER**

RAC

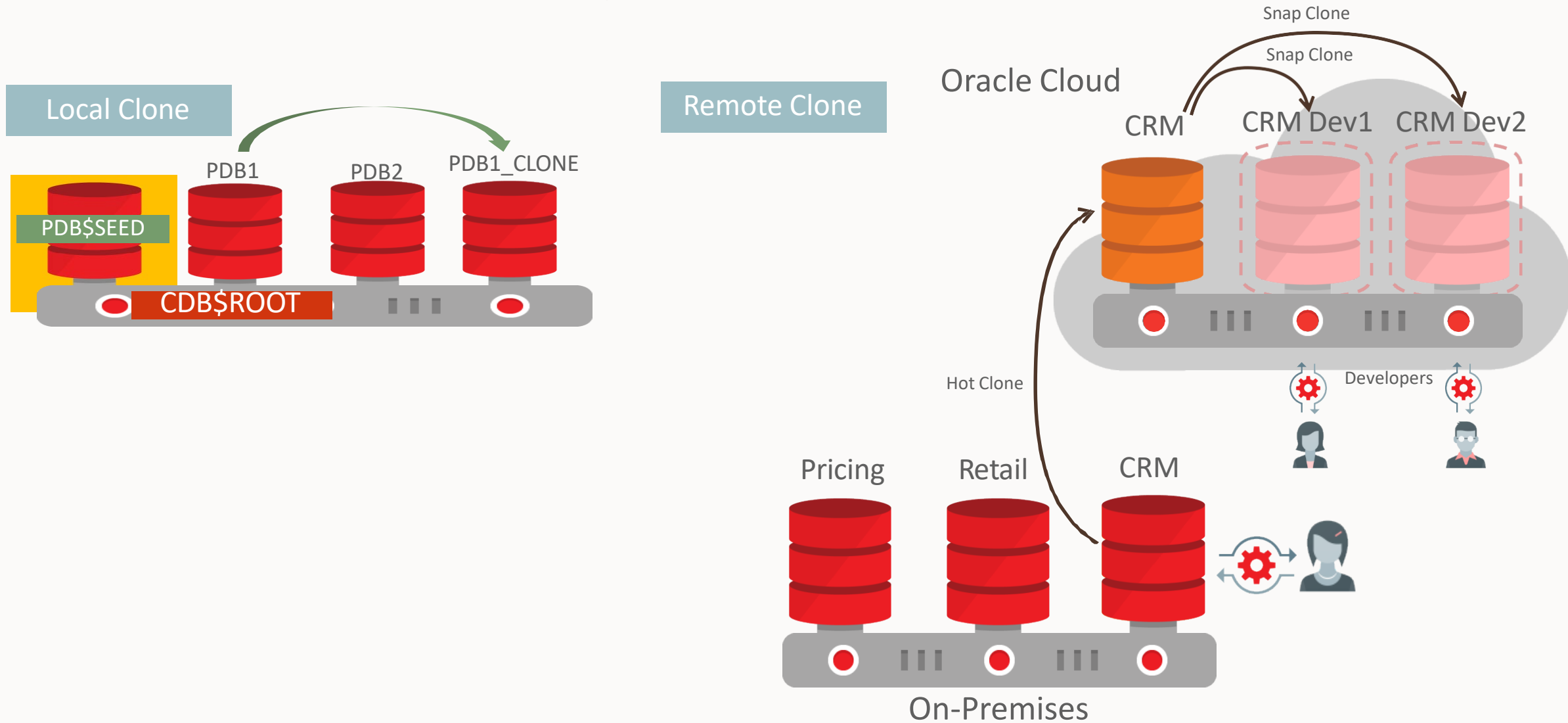


**BRONZE**

Backups



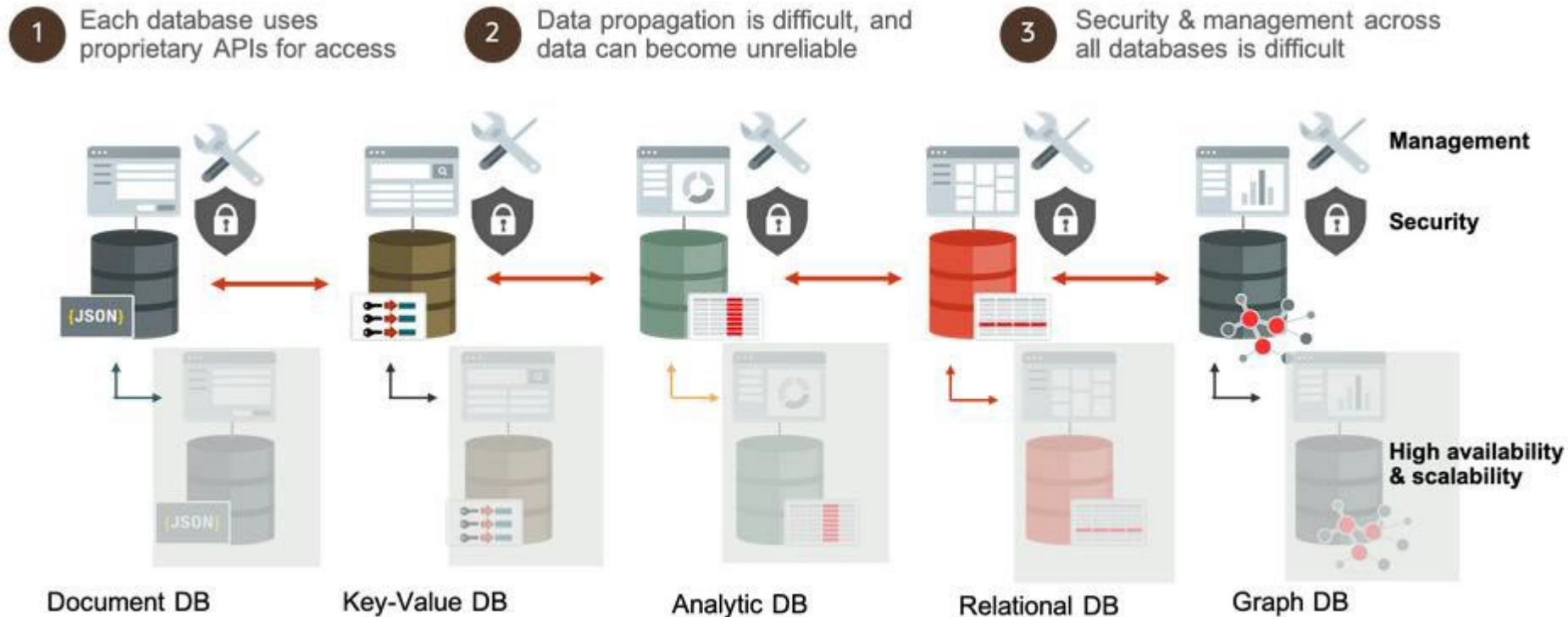
# 활용:Fast provisioning을 이용한 개발, 테스트 , DR



# 활용: Converged database

multiple different ways

- 데이터 모델 형태별 DBMS 구성 → 복잡한 아키텍처
- 통합 데이터 관리의 어려움 → 관리 비용 증가



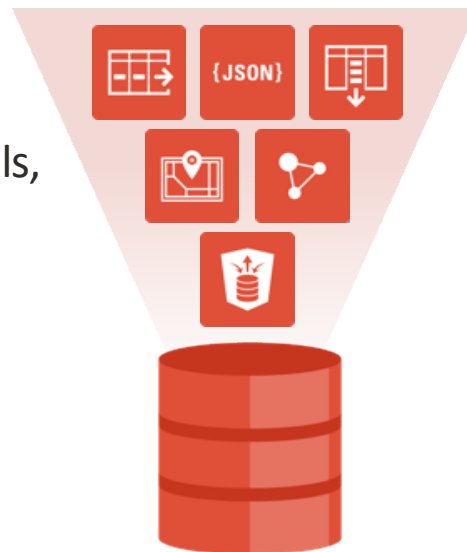
# 활용: Converged database

Multiple Single-Purpose vs **One Converged Database**

## One Converged Database vs. Several Specialized Databases Multi-Purpose System vs. Single-Purpose System

# ORACLE

Support Multiple Data Store & Models,  
Integrate Polyglot Environment  
(Multi-Model Polyglot)  
Security Standardization  
Simple to manage



# aws

Single Model Polyglot Data Fragmentation  
Complex Application Complex to Manage  
(Interface, Security Model, Recovery Procedure ...)



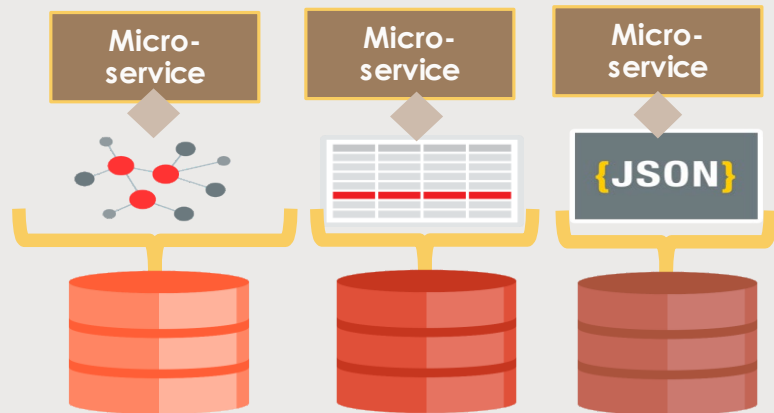
# 활용: Converged database

## Microservice 활용

업무별 데이터를 컨테이너 기반으로 독립적으로 저장하면서 통합 관리

### 단일 목적 데이터 베이스

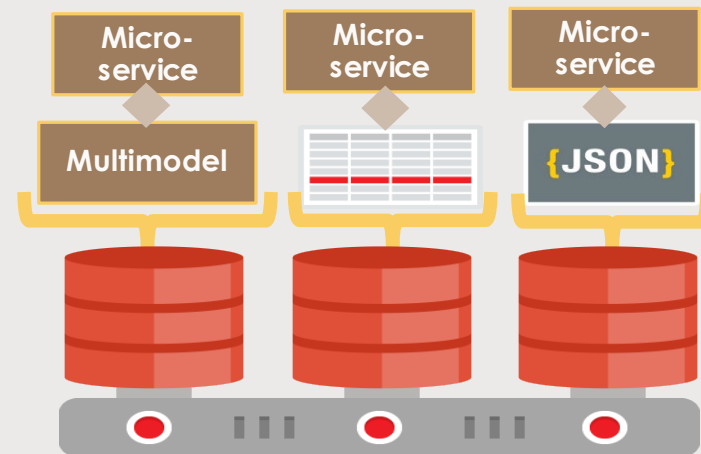
- ✓ Micro Service별로 단일 목적의 데이터베이스사용,
- ✓ 사용된 데이터베이스 종류만큼의 운영관리자 필요



VS

### 컨테이너 기반 데이터 베이스

- ✓ 컨테이너 데이터베이스기능을 활용하여 각 Micro Service별로 독립된 데이터 저장소 제공
- ✓ 통합 관리자에의해 운영



# Multitenant 활용방안 3가지

---

