

# 多租户高能设计

## 让你的数据库整合梦想成真

公益讲座11:00准时开始,请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布,已入群客户请勿重复入群!



20-20

数据库和云讲座群



甲骨文云技术公众号



B站专家系列课程

# 多租户高能设计 让你的数据库整合梦想成真

甲骨文技术公益课 - 数据库专场

2023年3月31日 11:00

线上直播



# 基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：

**400-699-8888**

\* 活动最终解释权归甲骨文公司所有

# 多租户高可用性的优秀实践



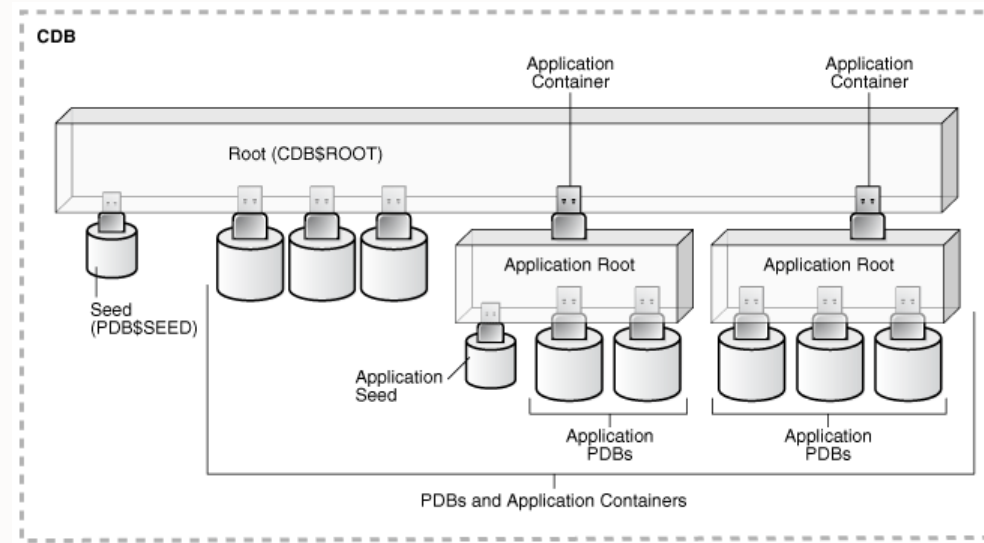
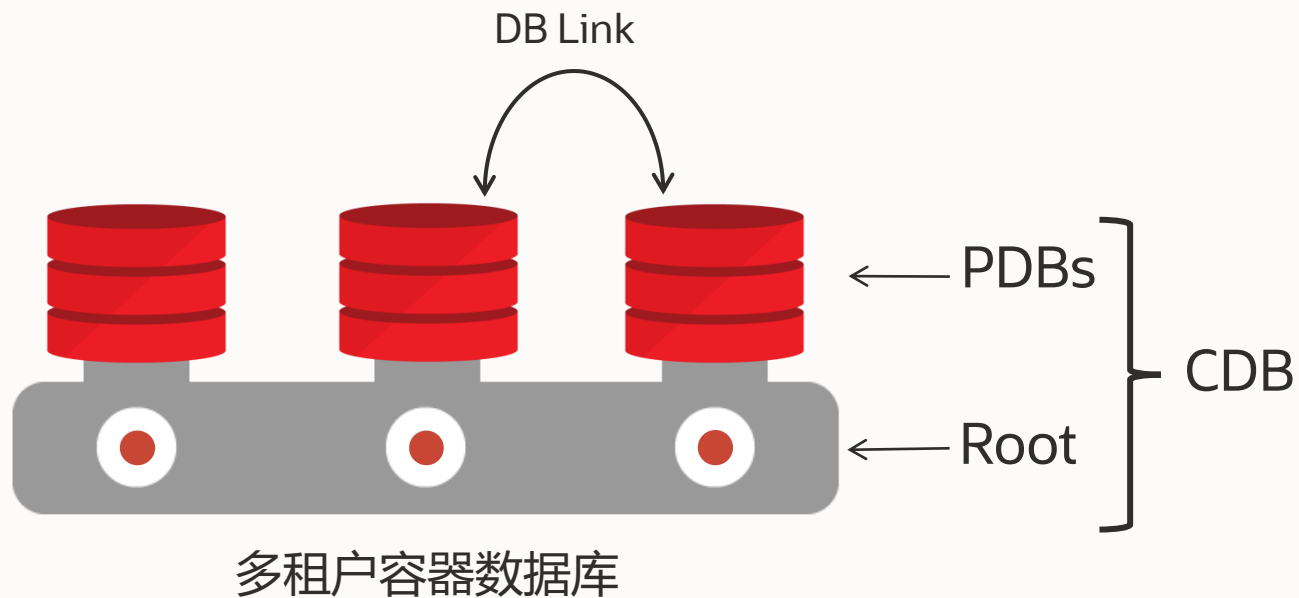
1. 多租户回顾
2. PDB 生命周期管理
3. 热克隆的核心技术
4. 与数据卫士的整合
5. 可刷新的PDB切换
6. 与RAC集群的整合
7. 快照轮盘
8. Transportable RMAN备份



# 多租户回顾

---

# 多租户架构



PDB是一个可移植的模式、模式对象和非模式对象的集合，在Oracle客户端看来是一个非CDB



# 多租户的主要好处



## 最大限度地减少资本支出

- 每台服务器可以部署更多应用



## 最大限度地减少运营成本

- 以一管多 (减少 patching ! )
- 标准化流程和服务水平
- 自助服务供给



## 最大限度地提高敏捷性

- 用于开发和测试的快照克隆
- 通过 "可插拔性" 实现可移植性
- 使用RAC的可扩展性



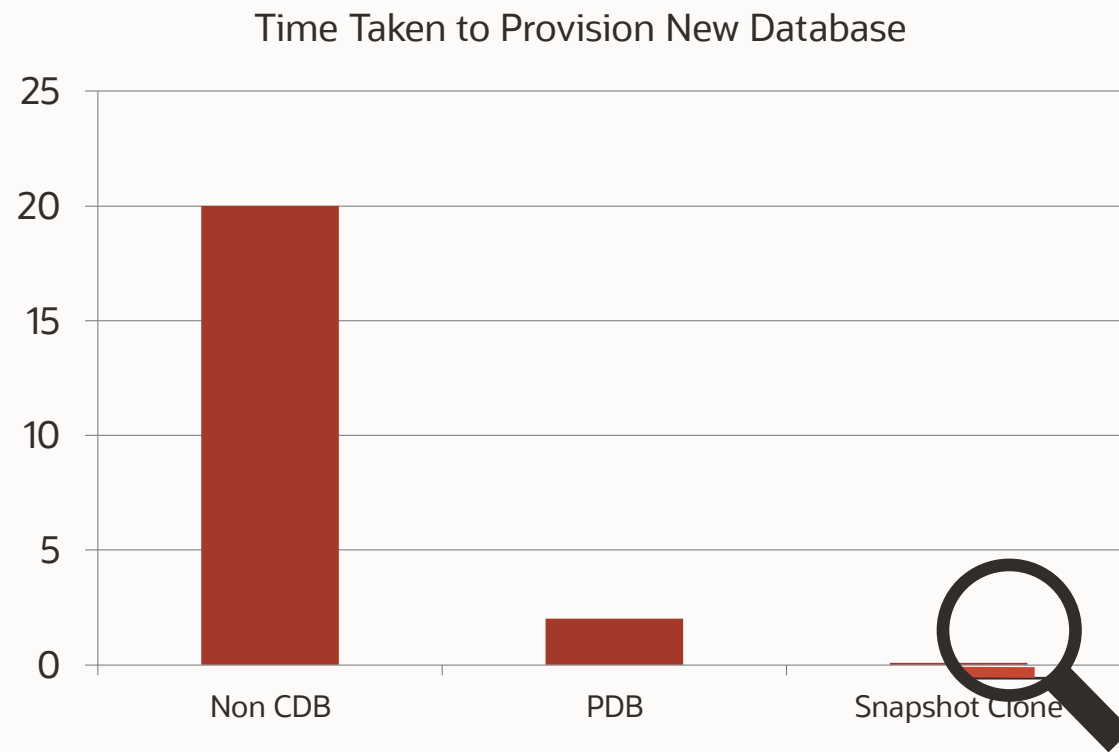
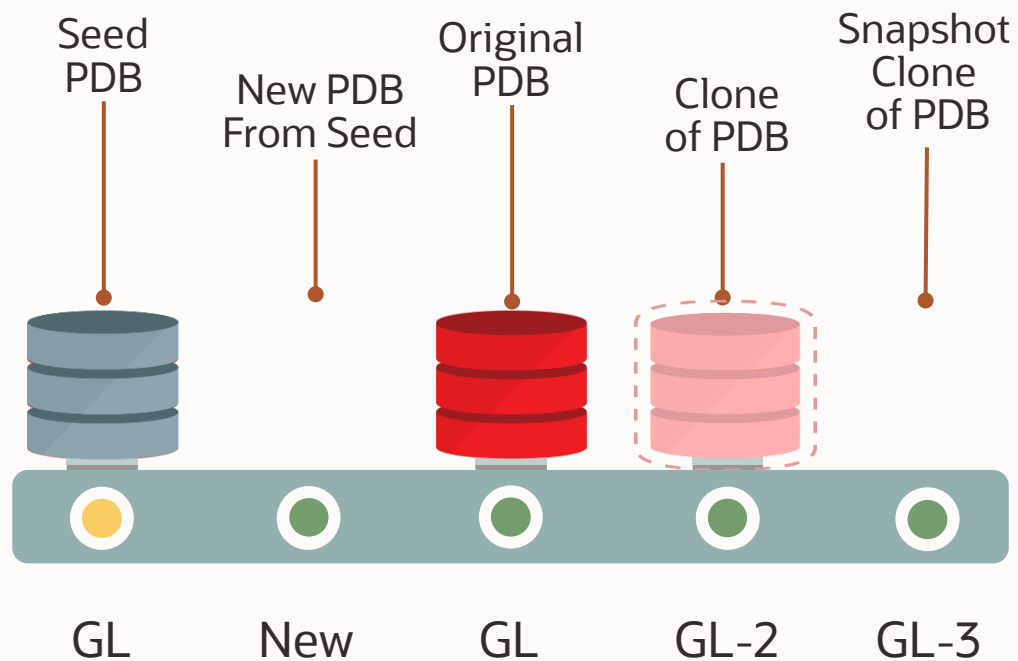
## 简单易用

- 无需改动应用



# 多租户特性为新项目提供敏捷性

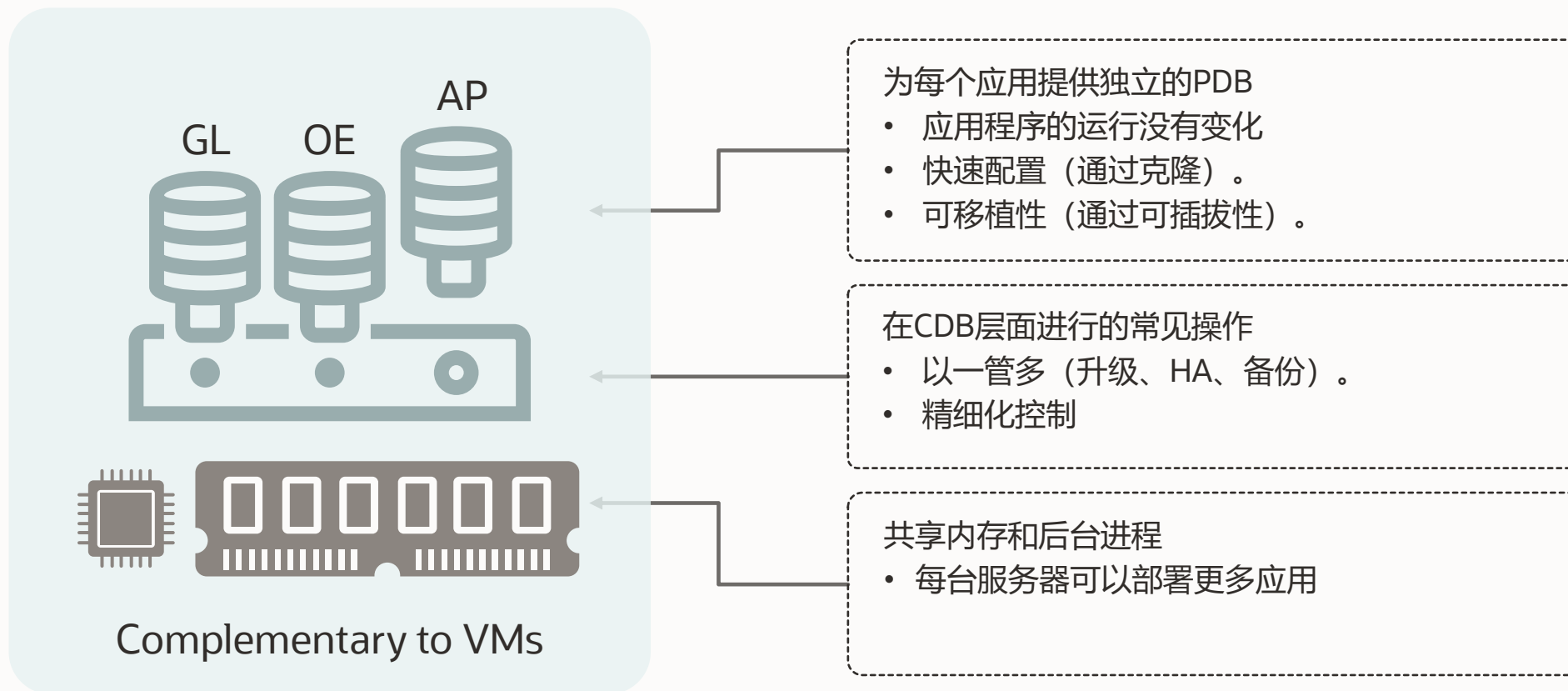
## 快速配置，快照克隆





# 多租户架构的优势

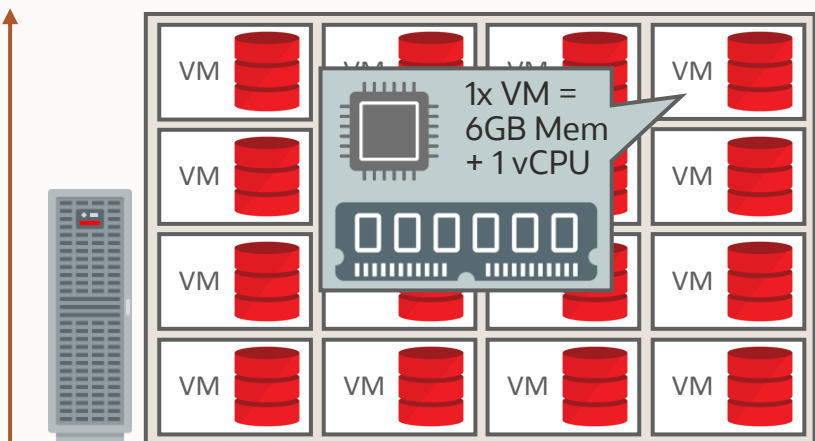
## 规模经济下的隔离和敏捷性



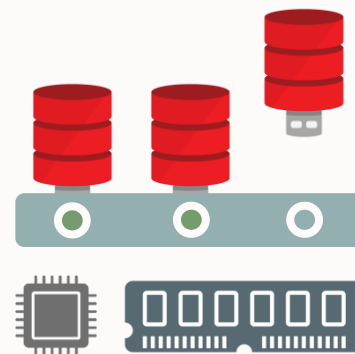
# 比较数据库整合架构



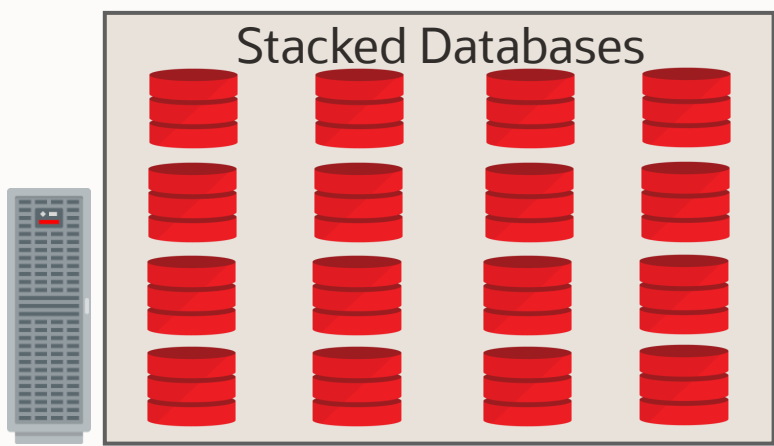
隔离  
& 敏捷性



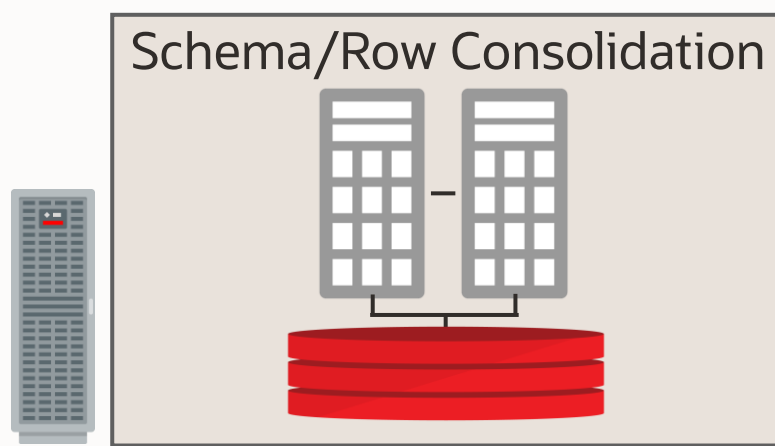
Multitenant



Stacked Databases



Schema/Row Consolidation

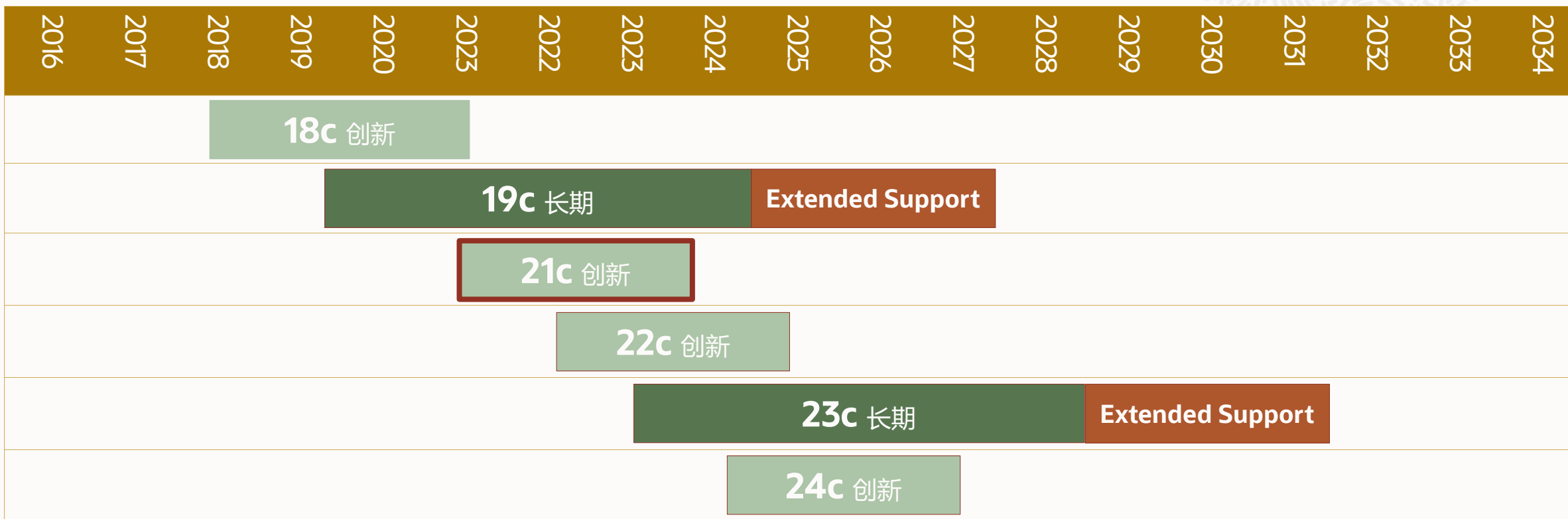


效率



# 预计的数据库发布和支持时间表

(可能有变化)

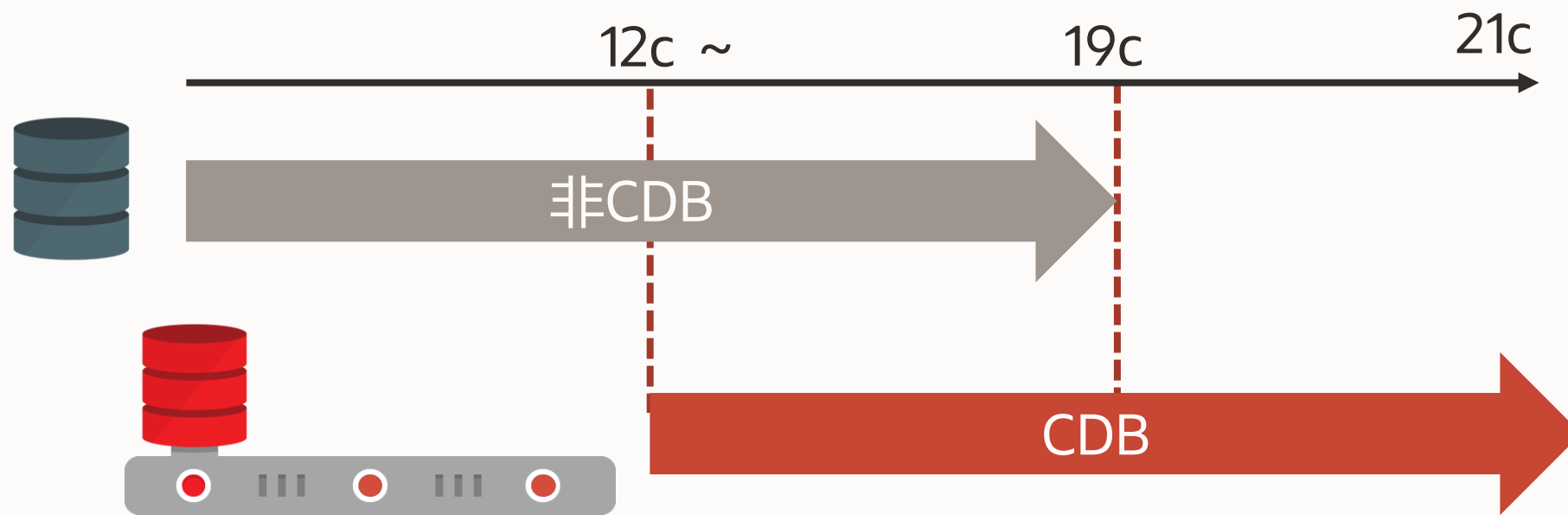


- 创新版 - 2年的 Premier Support
- 长期发布--5年的 Premier Support , 以及3年的 Extended Support



# 未来架构策略

Oracle数据库21c之后只支持CDB配置



# 要使用Oracle多租户

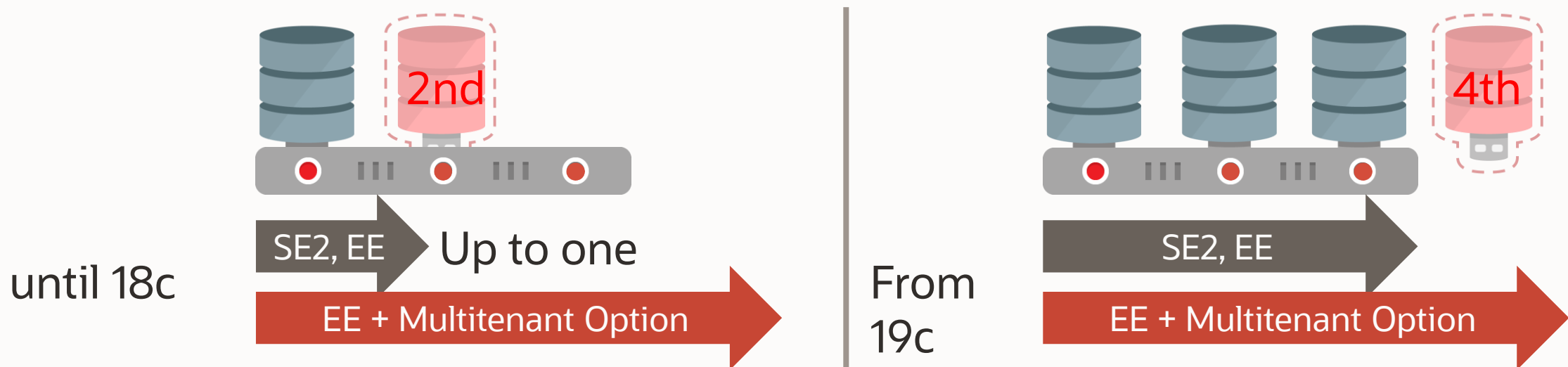
Oracle数据库自19c以来，在标准版中最多可以创建三个PDBs

## 甲骨文数据库标准版2

- 一个CDB上最多可以创建三个PDBs

## 甲骨文数据库企业版

- 每个CDB最多有3个PDB
- 要在一个CDB上创建4个或更多的PDB，需要付费选项Oracle Multitenant



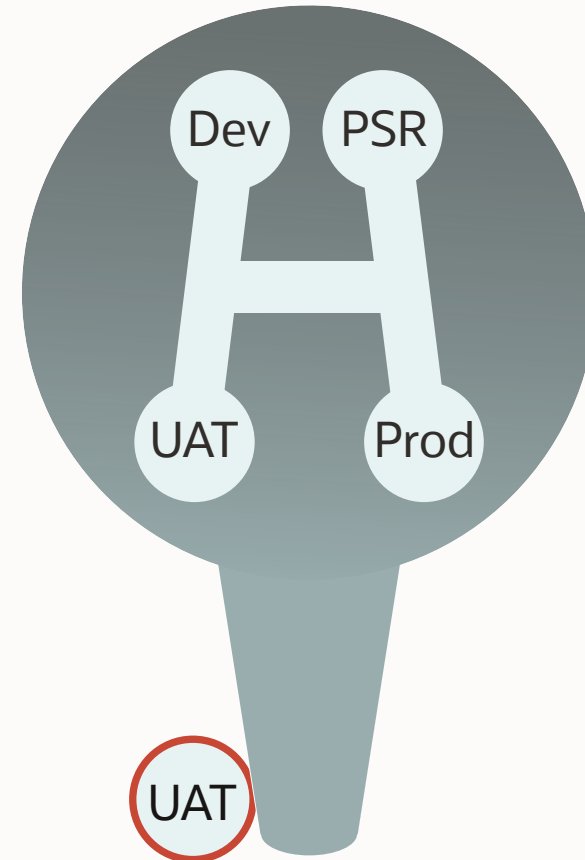
# PDB生命周期管理

---



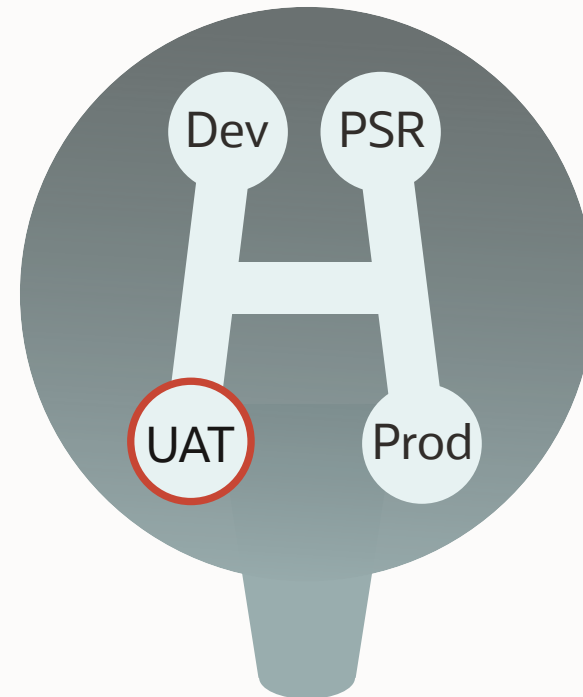
## PDB生命周期管理

- 热克隆
- 可刷新的PDBs
- PDB重定位



## PDB生命周期管理

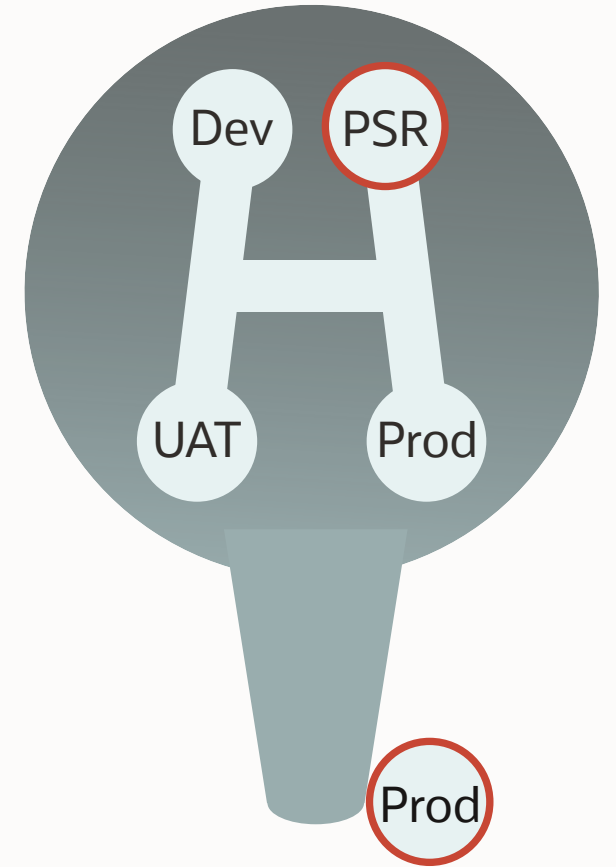
- 热克隆
- 可刷新的PDBs
- PDB重定位





## PDB生命周期管理

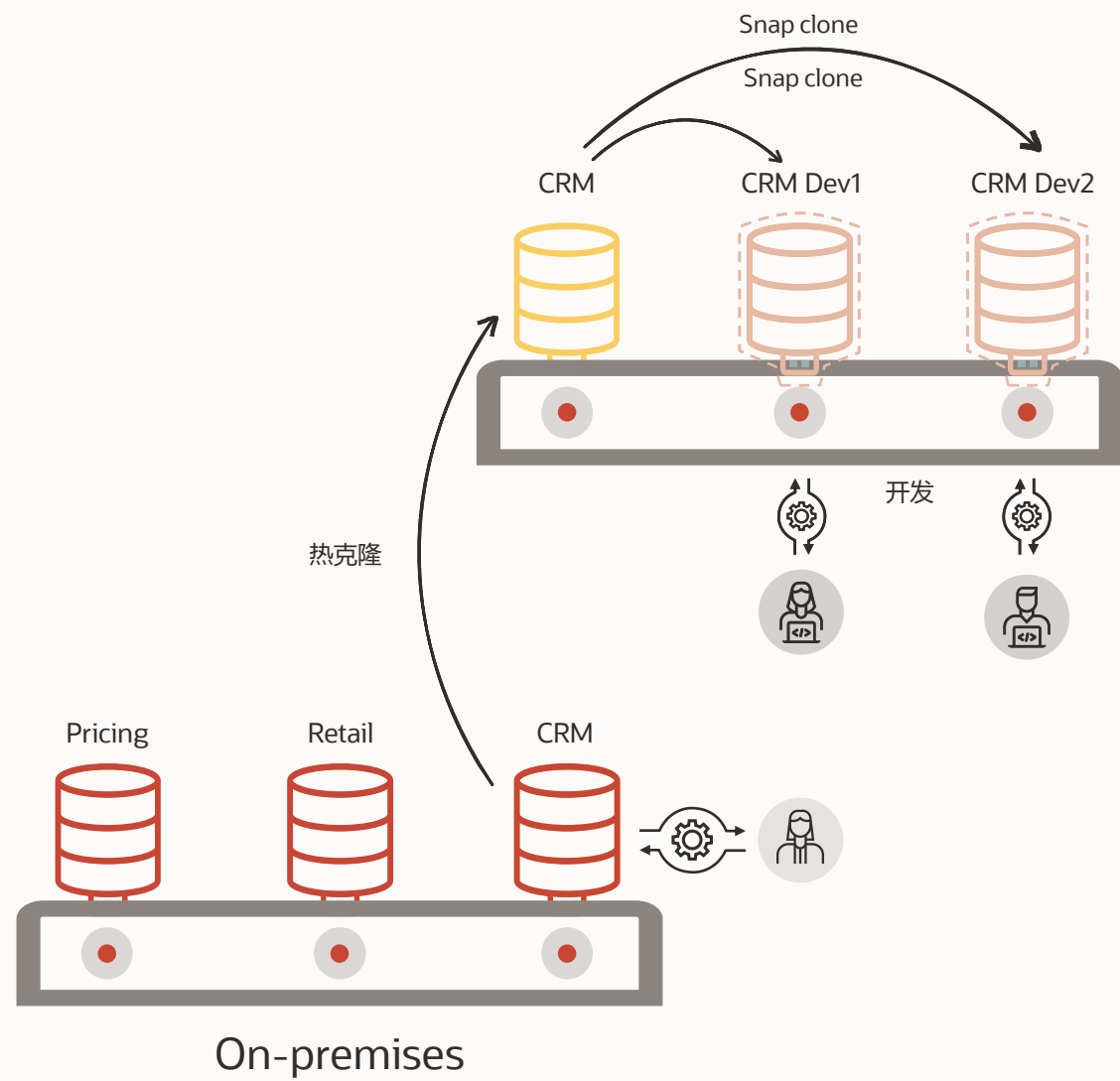
- 热克隆
- 可刷新的PDBs
- PDB重定位



# PDB热克隆

## PDB热克隆

- 在线实例化的黄金副本



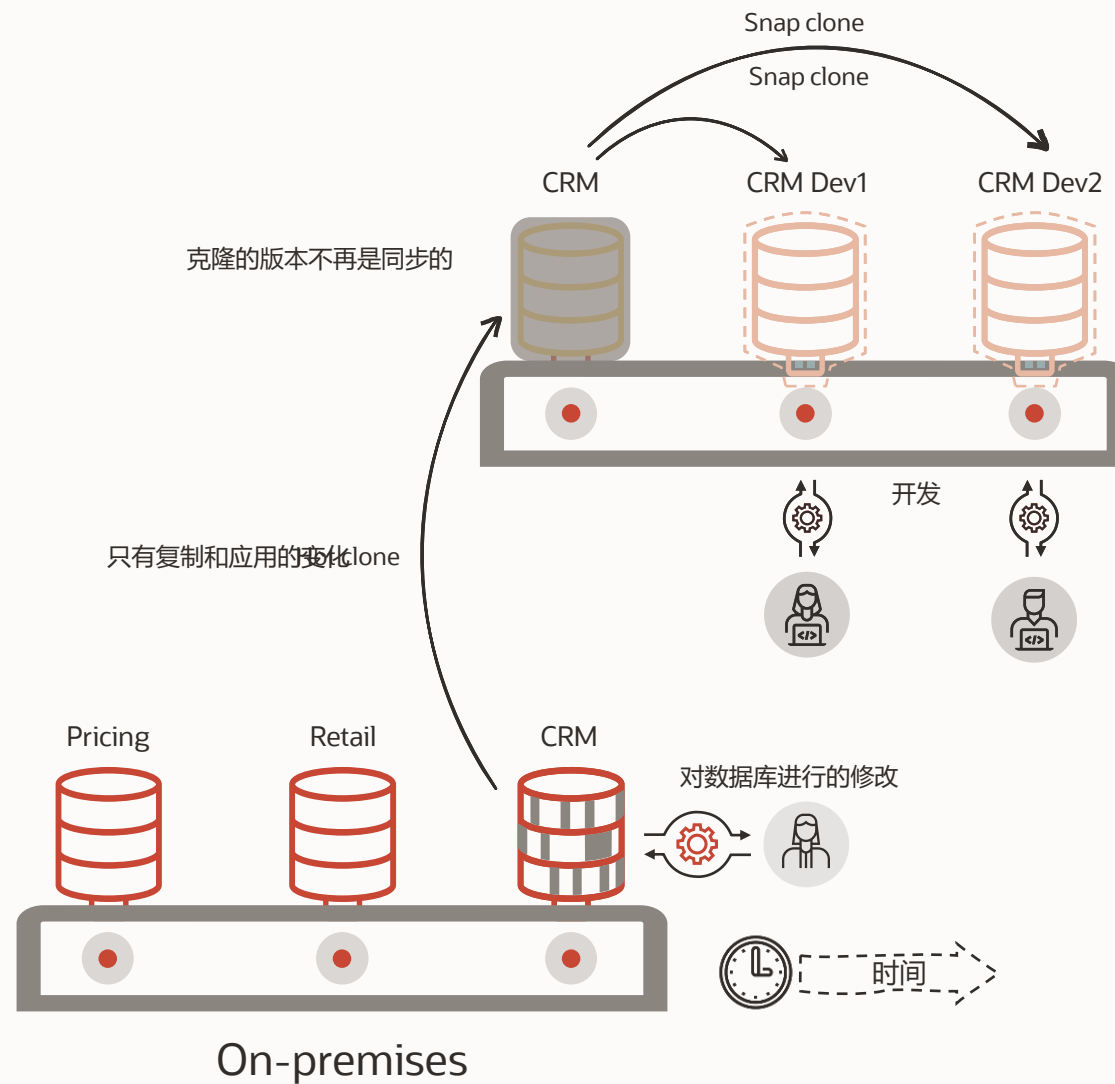
# 可刷新的PDB

## PDB热克隆

- 在线实例化的黄金副本

## 可刷新的PDB

- 用最新的数据对克隆进行增量刷新



# 在线PDB重定位

## PDB热克隆

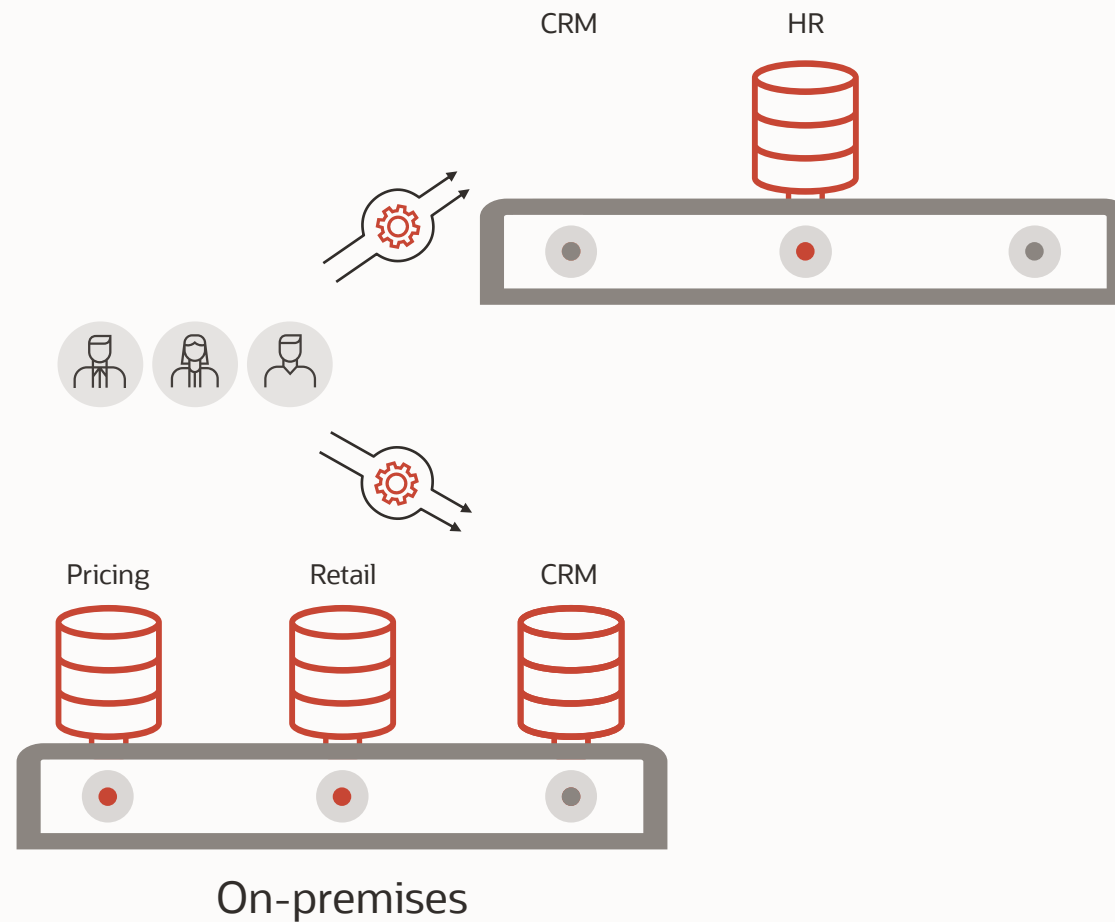
- 在线实例化的黄金副本

## 可刷新的PDB

- 用最新的数据对克隆进行增量刷新

## 在线PDB重定位

- 搬迁时不需要停机



# 热克隆的核心技术

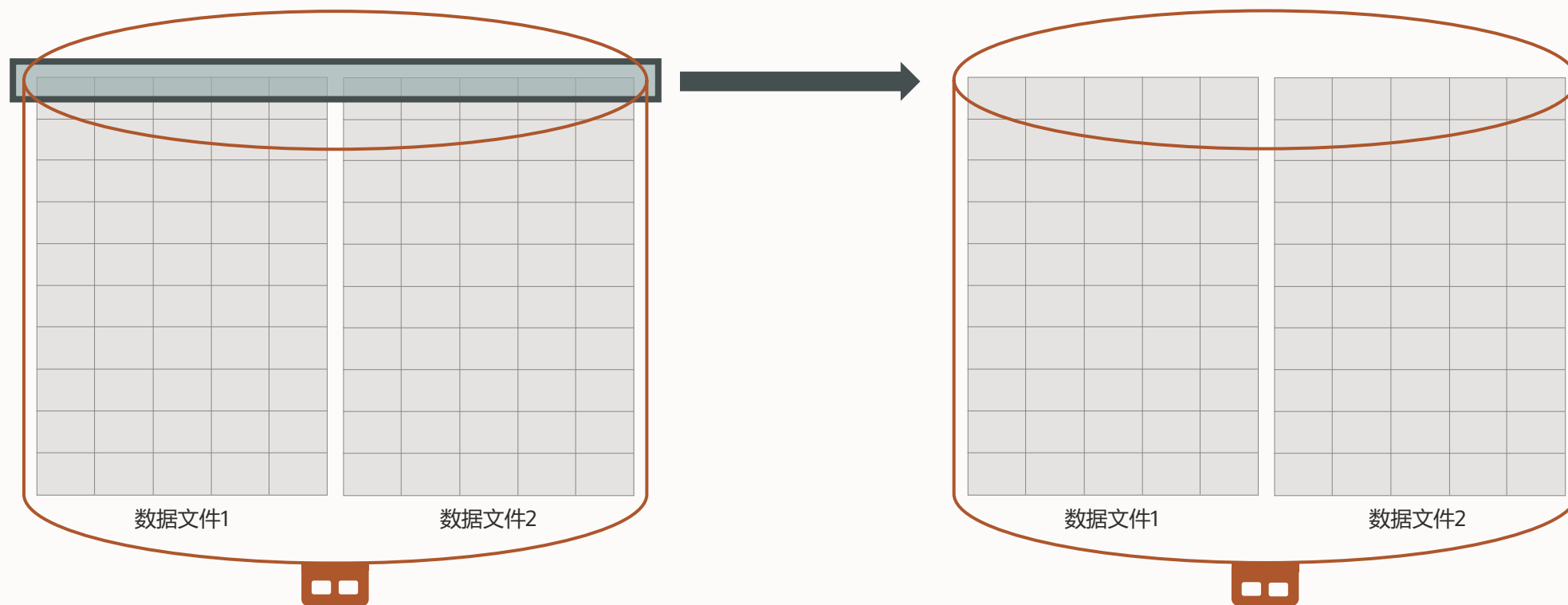
幕后发生了什么...

# 冷克隆

需要停用源PDB



- 源PDB过渡到只读
- 并行读取和复制
- 当完成后打开源数据库读写

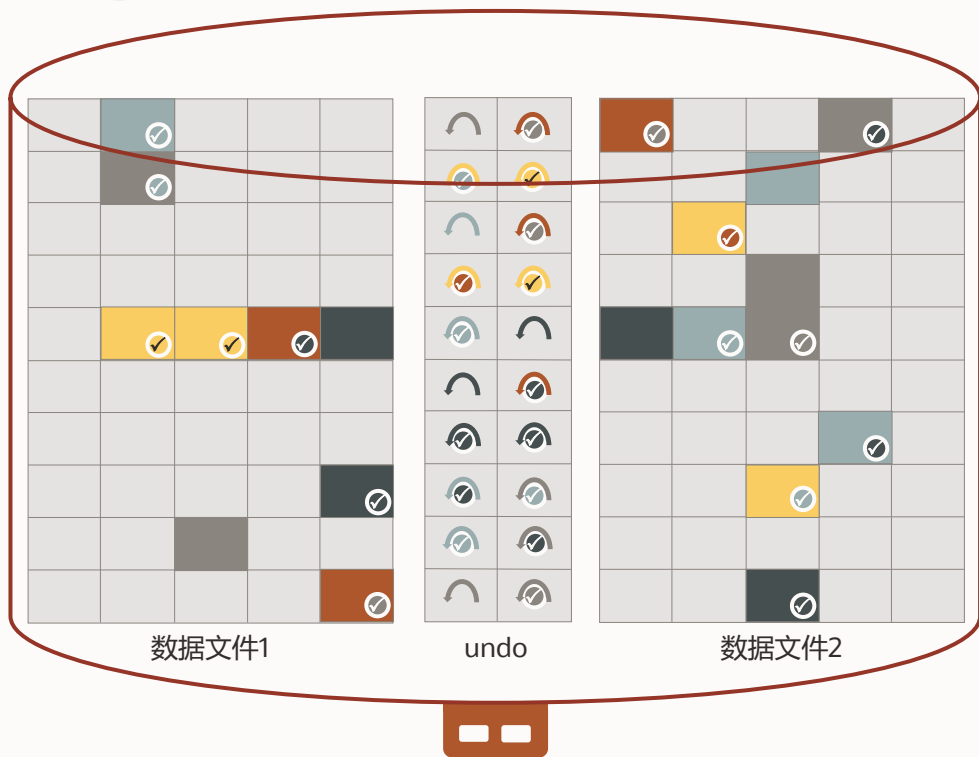
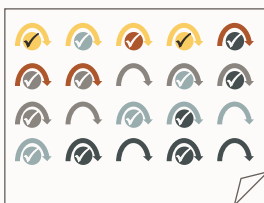


# 冷克隆





## 数据库随时间的变化



redo日志



Legend

-  Uncommitted block changed in interval
-  Block changed in interval  
Committed in interval
-  Uncommitted redo
-  Undo written in interval  
Committed in interval



# 热克隆



redo日志



源PDB保持读/写

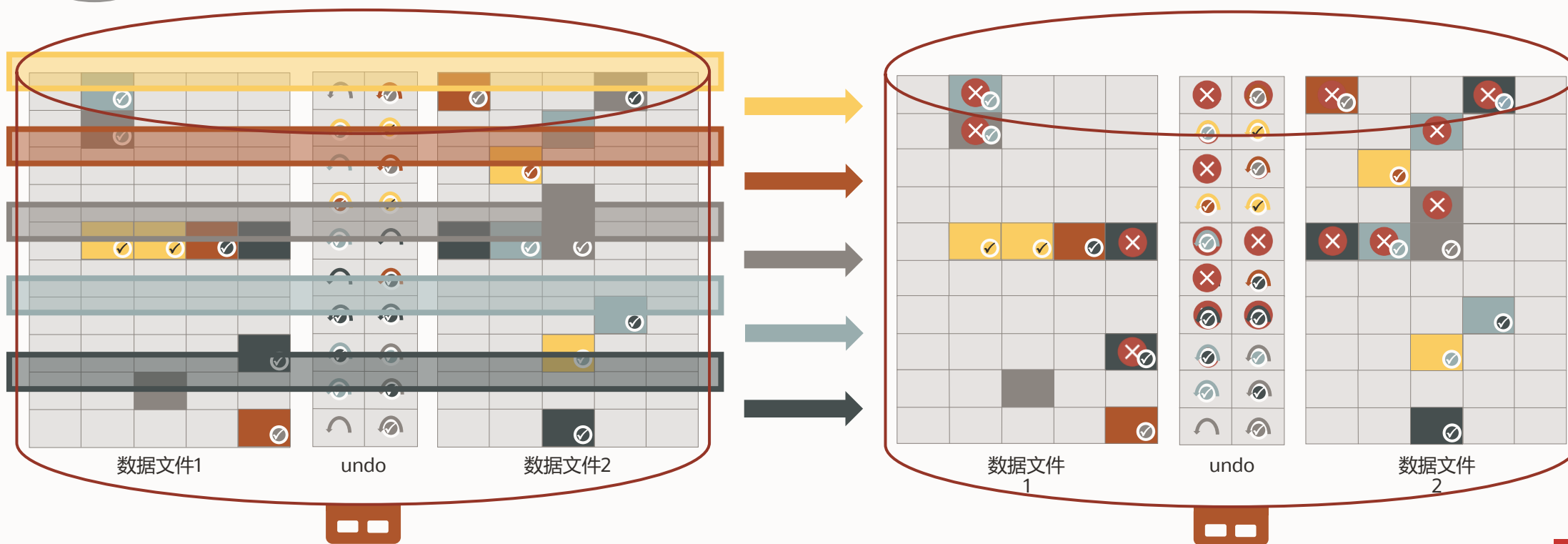
并行读取和复制

正在进行的操作意味着“fuzzy read”

数据变化不包括在初始文件副本中

传输和应用redo以追平源端

应用undo 回滚未提交的事务



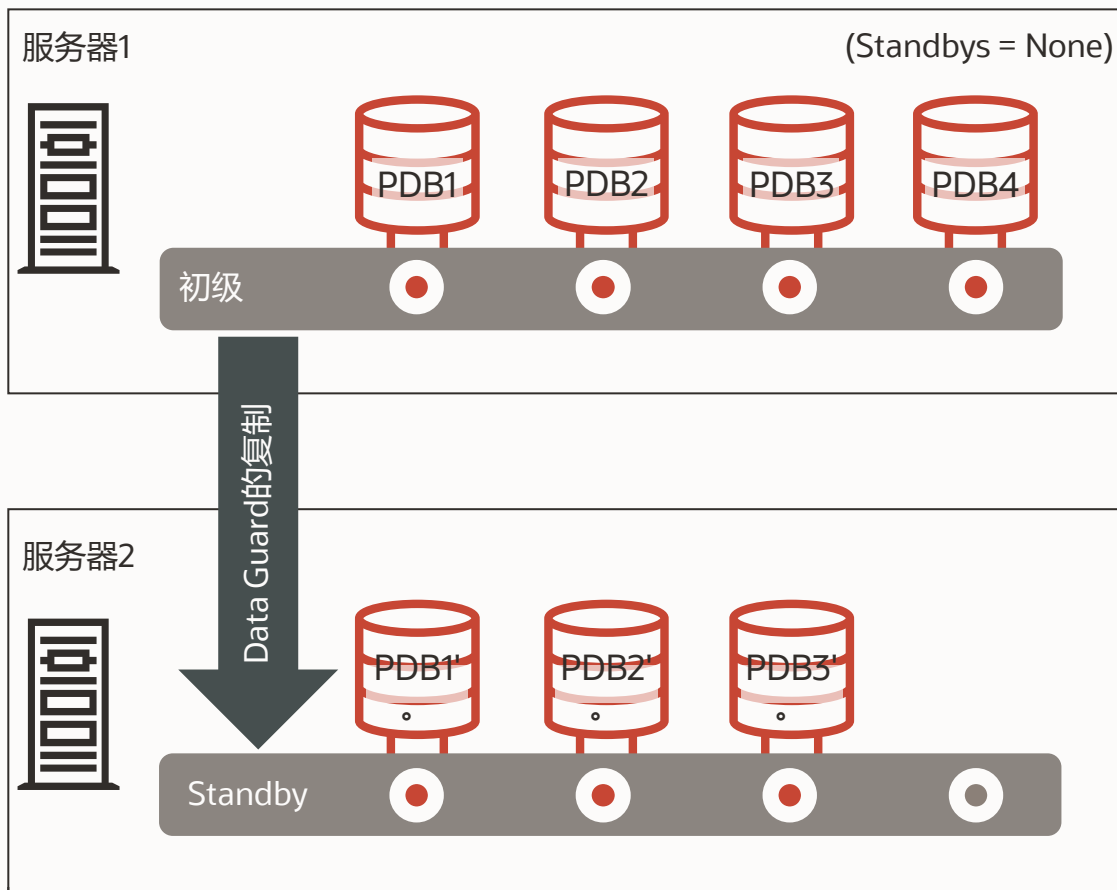


# 与数据卫士的整合

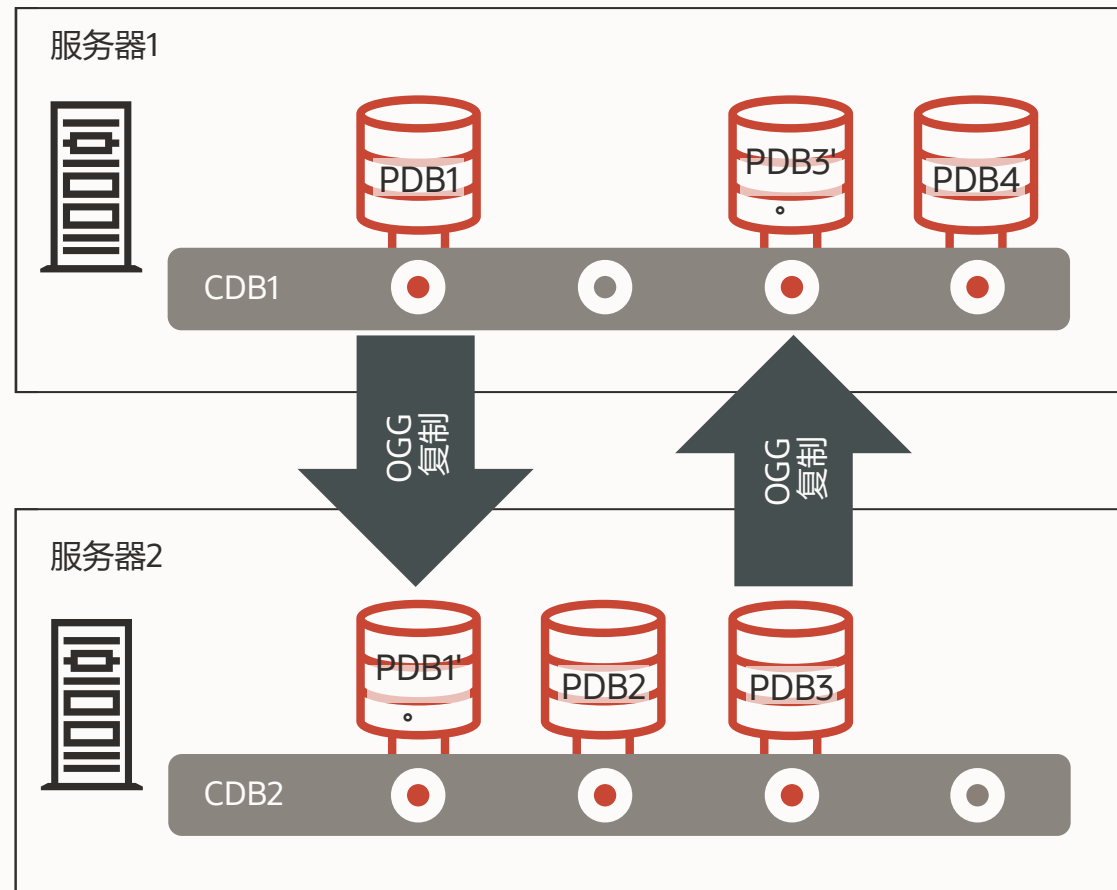
在高可用性配置中配置PDB的考虑因素

# 多租户复制能力

## 通过Data Guard 管理

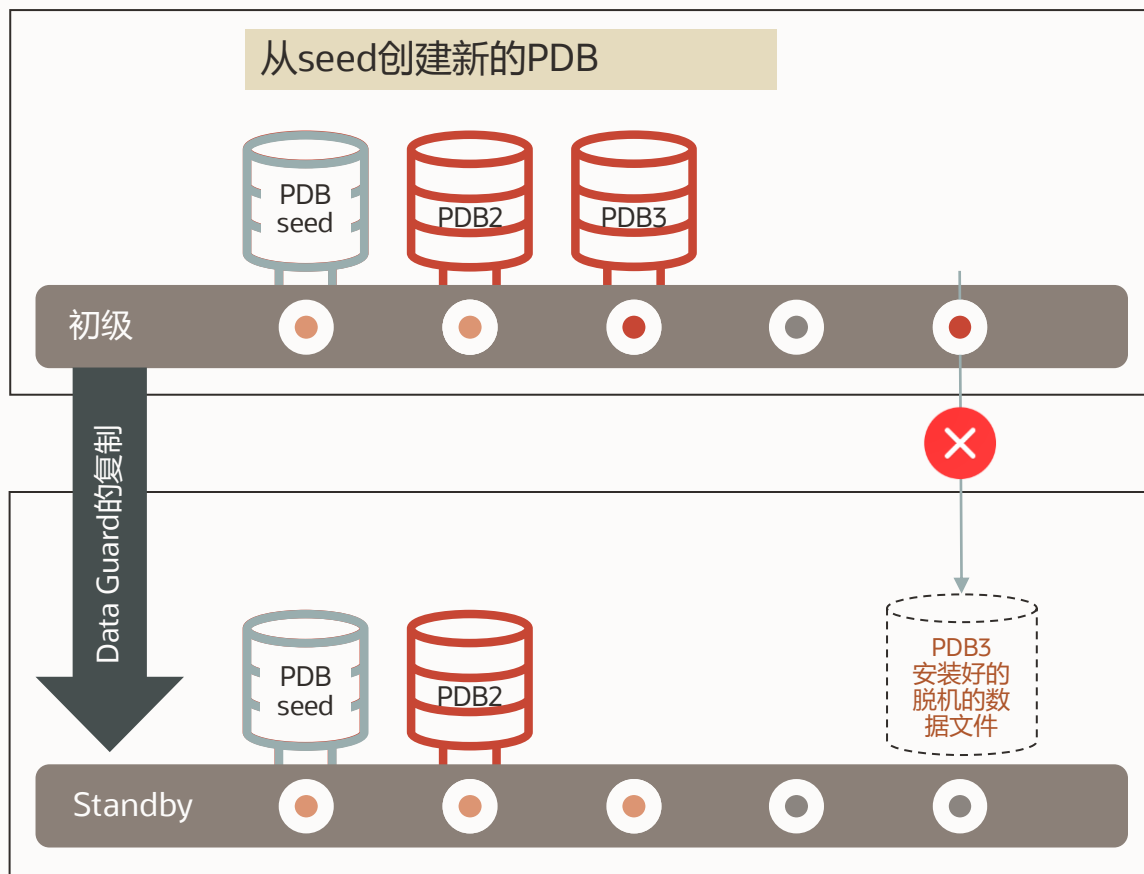


## 使用GoldenGate的PDB之间复制



# Data Guard配置中的PDB配置

简单配置下没有问题；在更复杂的情况下需要额外的步骤



## 从seed创建新的PDB

- PDB\$Seed 在Standby端
- 操作重放成功

## PDB1的本地冷克隆

- PDB1在Standby端
- 操作重放成功

## 热克隆

- 隐式执行，无Standby

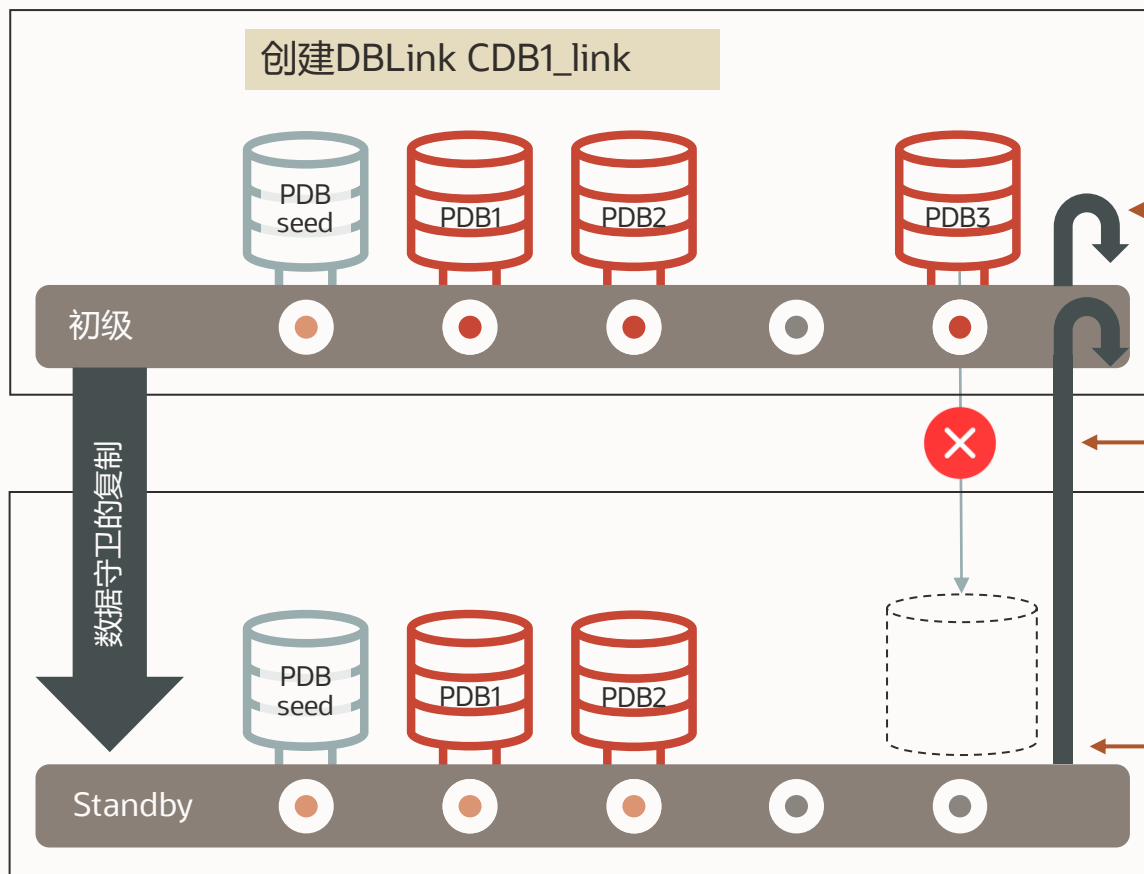
## Plug-in 或远程冷克隆

- Standby端上没有源数据文件，所以操作无法成功重放。
- PDB3可见但在Standby端下无法使用 (数据文件脱机)
- Data guard 恢复被停止



# 加强与Data Guard的整合

额外的一次性Data Guard设置步骤使Standby端可以访问源端上的文件

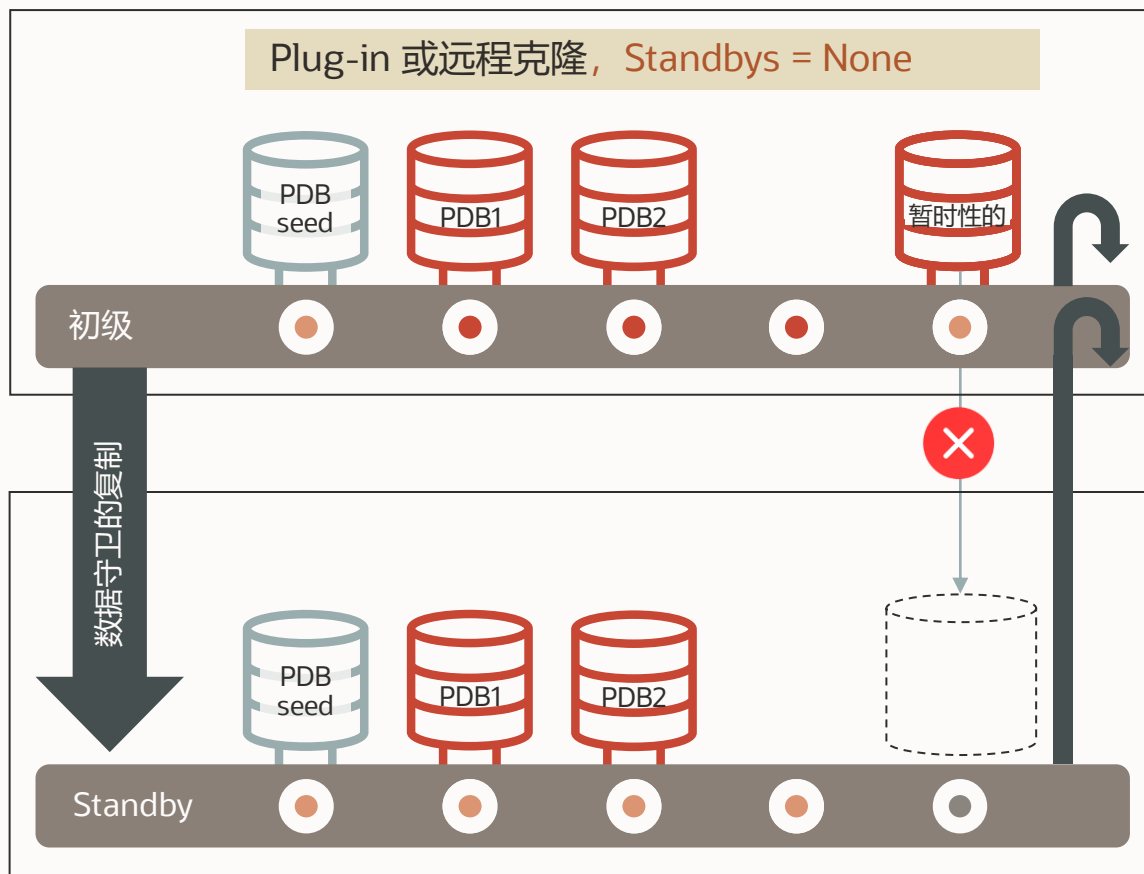


1. 在主服务器上创建自引用的DBLink
  - 在Standby端上进行重放，以创建引用源端的DBLink。
2. Standby端要设置的新参数：
  - `standby_PDB_source_file_Dblink`
  - 标识上一步创建的DBLink
3. 打开Standby端的CDB\$Root为只读
  - 当一个创建PDB的操作在Standby端上重播时，如果所需的数据文件不在Standby端上，它们会通过这个Link从源端复制过来
  - Standby端CDB\$Root可以是只读的，无需Active Data Guard



# 加强与Data Guard的整合

简单的PDB Plug-in 或使用 " transient no-standby PDB "与Data Guard进行远程克隆



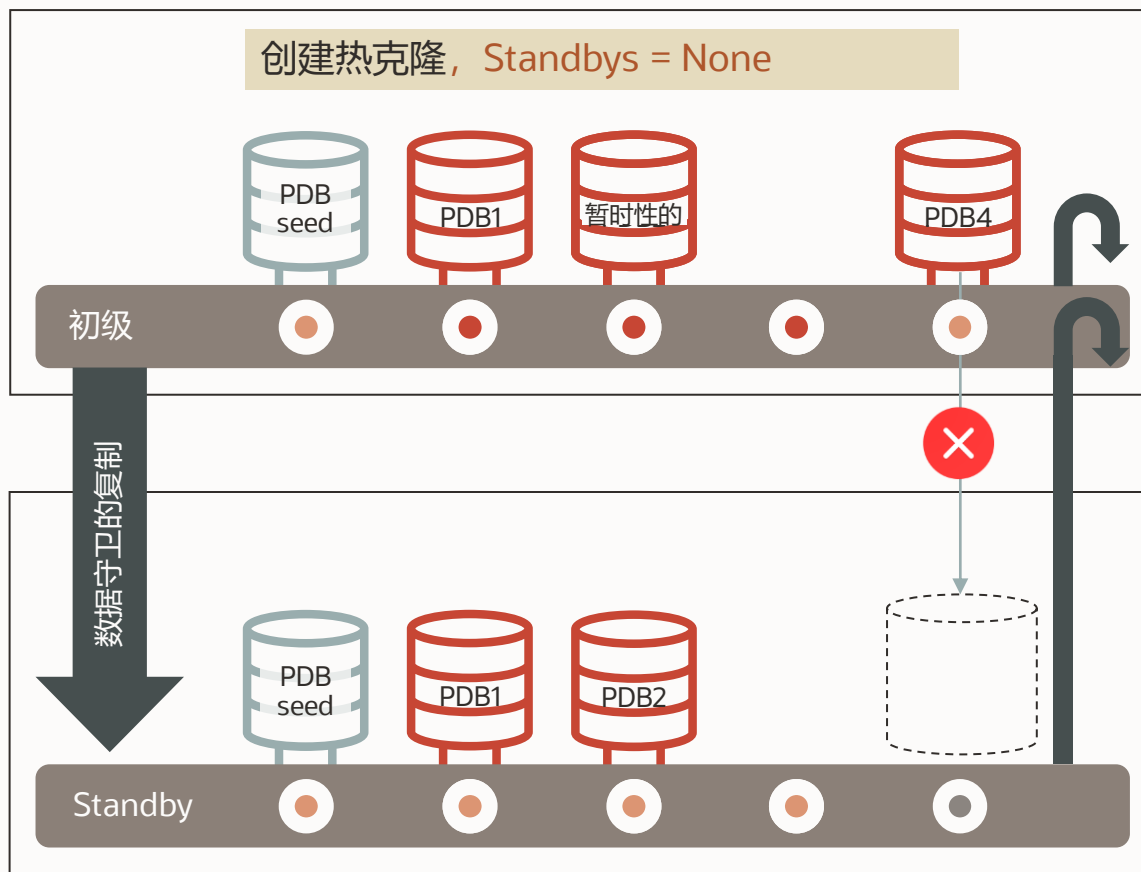
Plug-in 式或远程克隆PDB为PDB3

1. 创建 " transient no-standby PDB "
  - 既可以作为PDB Plug-in , 也可以作为远程克隆。
  - 重要: 在创建可插拔数据库语句中包含子句 "standbys=none"
2. 创建所需的PDB, 作为这个 transient no-standby PDB 的冷克隆。
  - 通过DBLink复制文件, 在Standby端上成功 Replay (Standby端参数 standby\_PDB\_source\_file\_DBLink)
3. transient no-standby PDB 现在可以被放弃了



# 加强与Data Guard的整合

## Data Guard配置中使用 " transient no-standby PDB "的简单热克隆



### 创建PDB4作为PDB2的热克隆

1. 热克隆到 " transient no-standby PDB "
  - 重要：在创建可插拔数据库语句中包含子句 "standbys=none"
  - 注意：热克隆的目的是为了在源端保持在线的情况下，实现克隆，而源端仍然保持online状态
2. 创建所需的PDB，作为这个 transient no-standby PDB 的冷克隆。
  - 通过DBLink复制文件，在Standby端上成功 Replay (Standby端参数 standby\_PDB\_source\_file\_DBLINK)
3. transient no-standby PDB 现在可以被放弃了

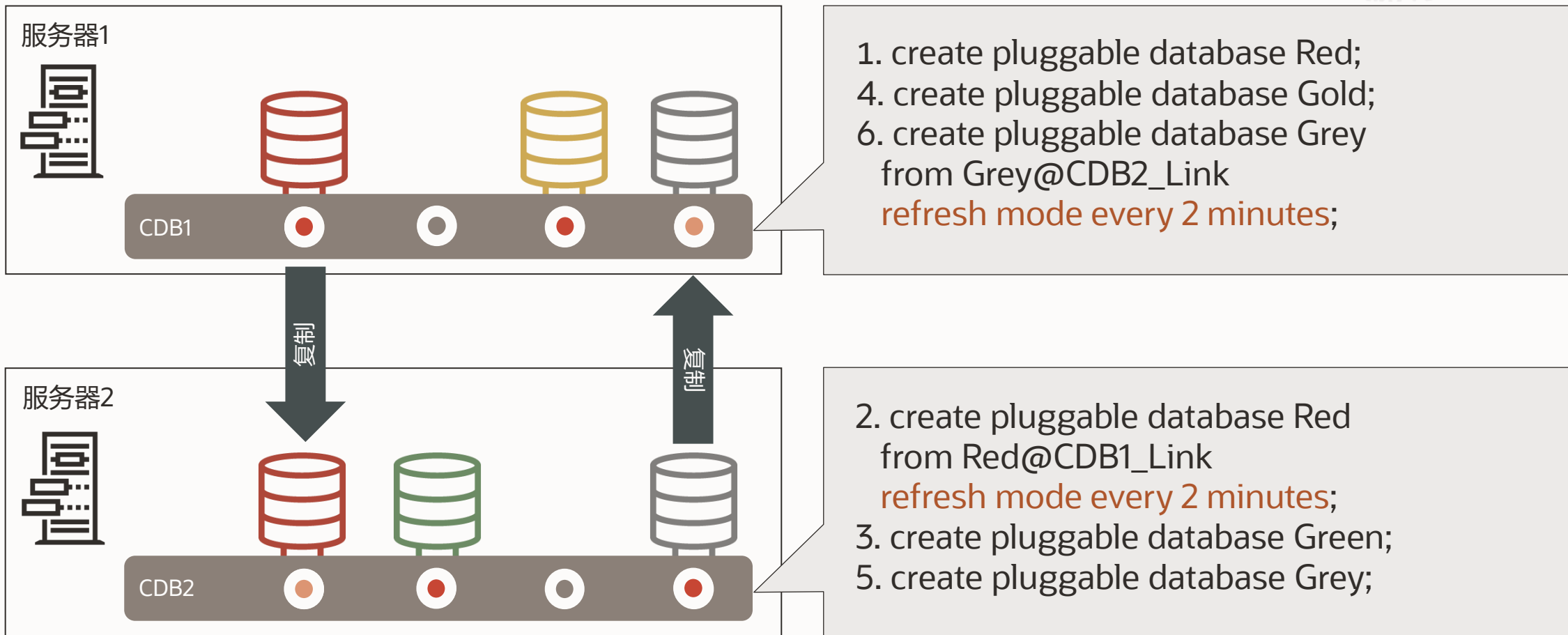


# 可刷新的PDB切换

简单化管理每个PDB的复制

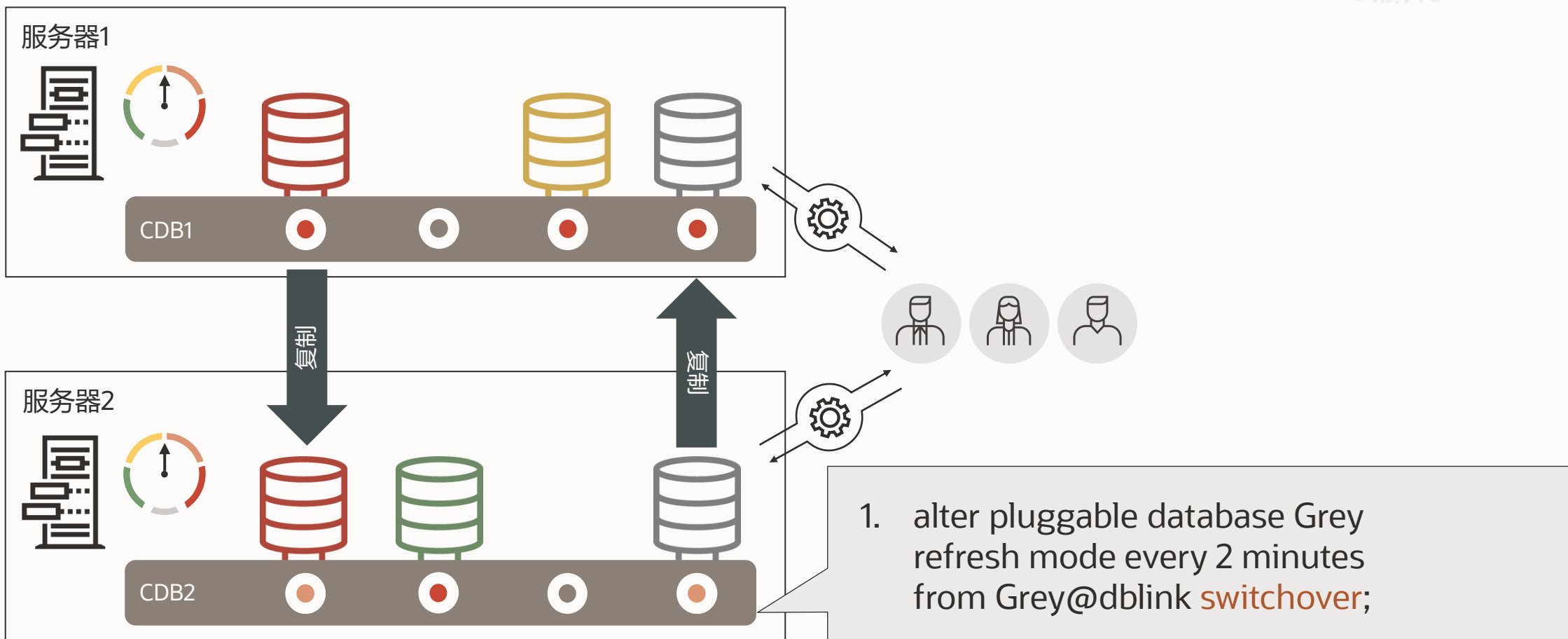


# 可刷新的PDBs作为副本 按PDB复制，仅CDB需要管理!

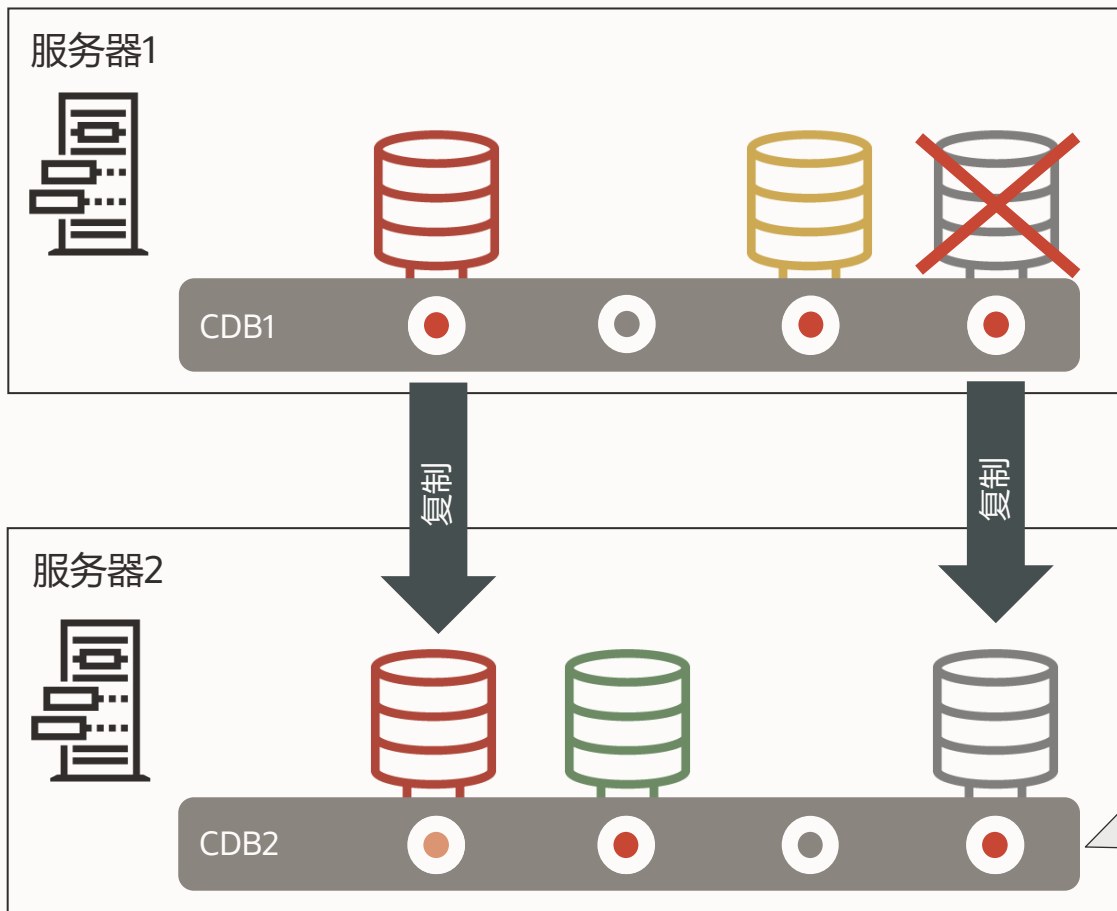




# 可刷新的PDB切换 计划内的转换



# 可刷新的PDB切换 计划外的切换



1. alter pluggable database Grey refresh;
2. alter pluggable database Grey refresh mode none;
3. alter pluggable database Grey open read write;



## 恢复时间目标 (RTOs) 需要多长时间才能恢复?



- 决定性的因素：
  - 在源端产生redo的比率
  - 复制刷新的频率
- 最坏的情况：在下一次刷新之前，源端发生故障
- 最佳做法：
  - 相对较高的刷新频率（几分钟而不是几小时）。
  - 用真实的工作负载进行完整测试



## 恢复点目标 (RPOs) 我将丢失多少数据?



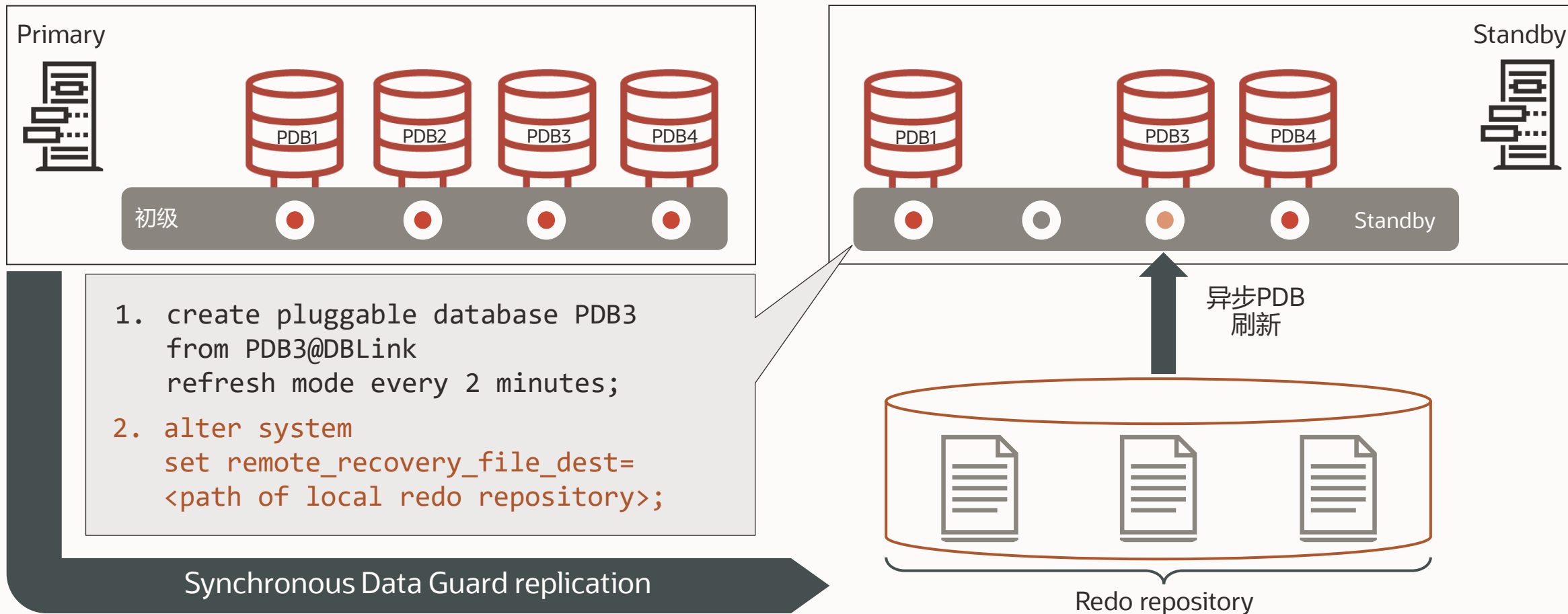
- 孤立于PDB本身的故障可能导致零数据损失
- 最大的数据损失是由复制PDB的刷新率决定的

```
create pluggable database myPDB from myPDB@DBLink  
refresh mode every 2 minutes;
```

- 最大的数据损失是2分钟的交易（如果主CDB故障）。



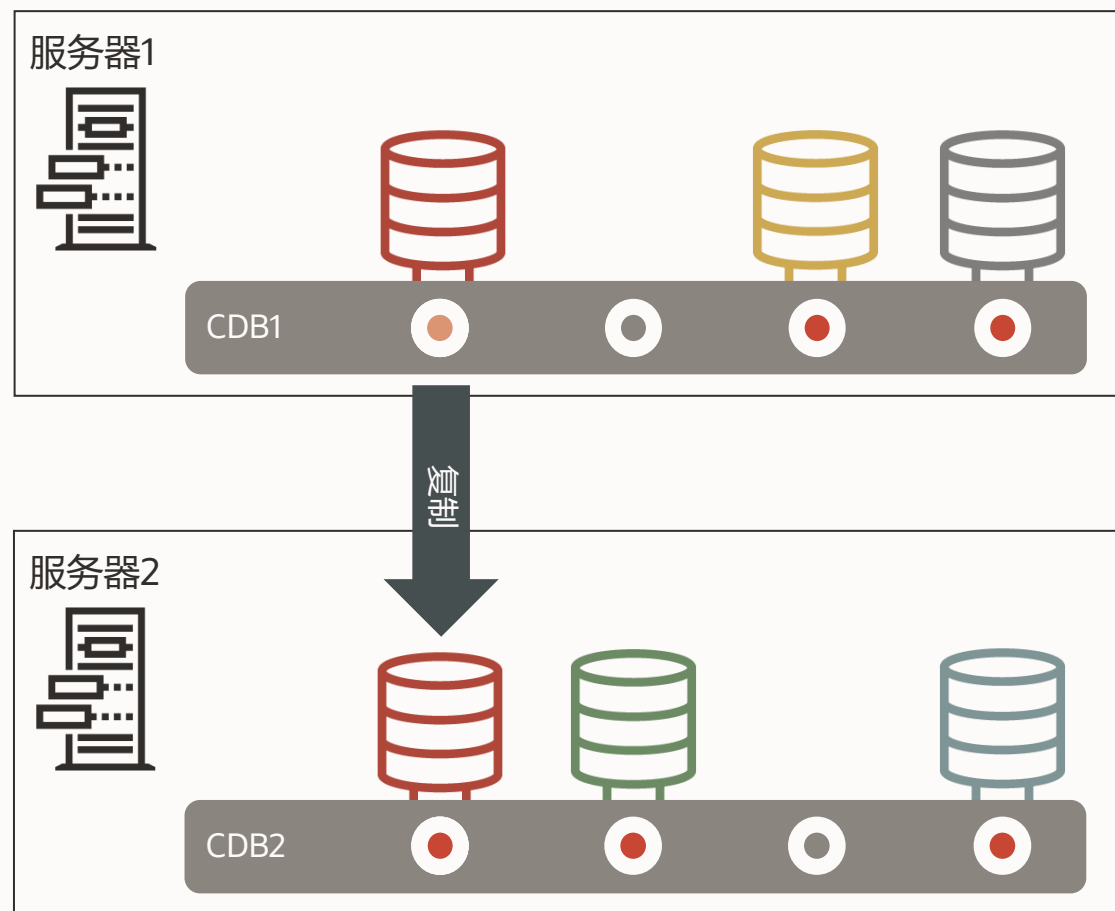
# 与 Data Guard redo repository 集成



# 在线PDB重定位过程中保留灾难恢复功能

在线PDB重定位到Data Guard Primary是有问题的

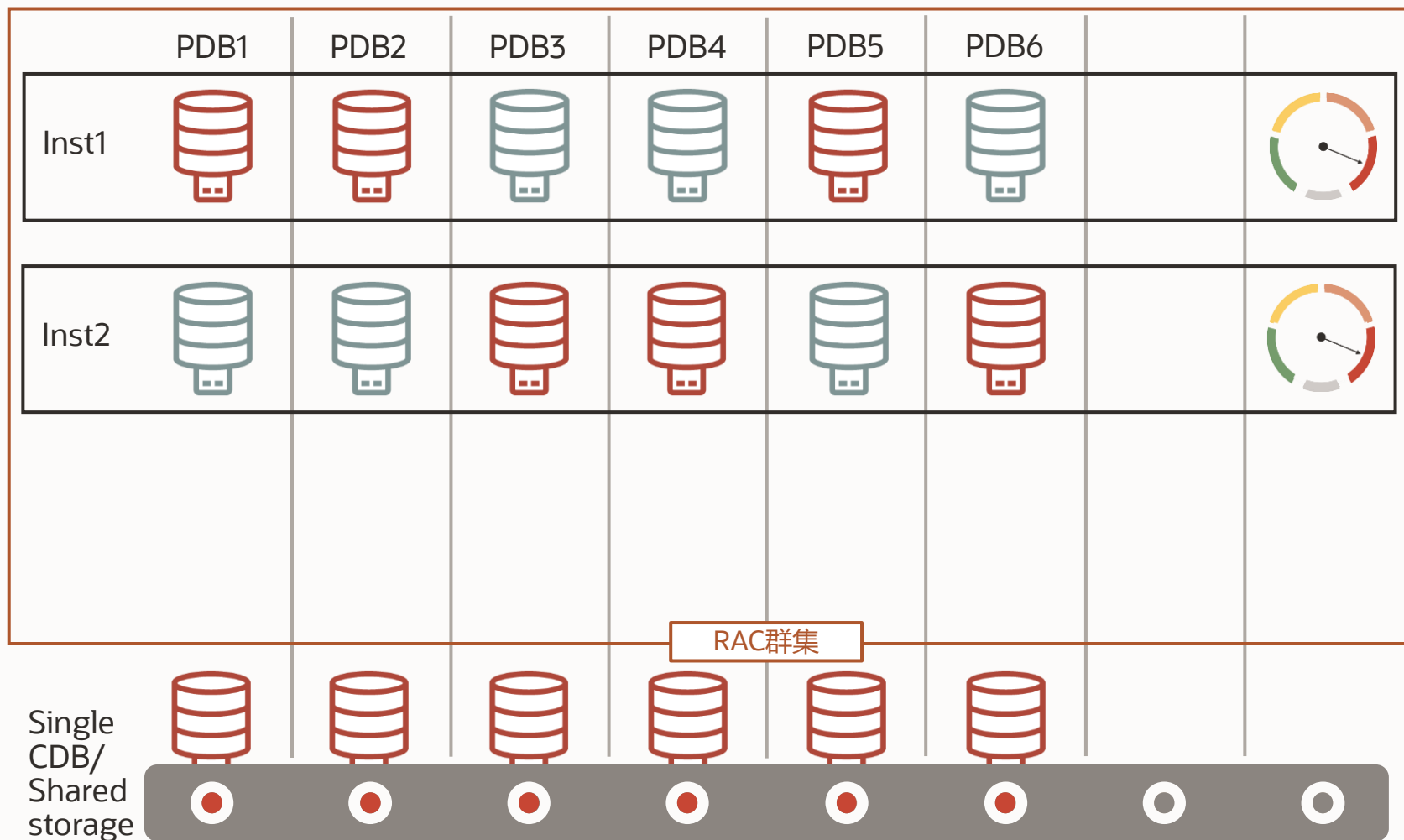
- 使用standbys=none, 以便在最小的中断情况下进行重定位。
    - 但这意味着没有复制的DR
  - 使用 transient no-standby PDB 技术来建立副本
    - 但这涉及到冷克隆: 不是HA
- 用可刷新的PDB切换解决HA/DR的困境



# 与RAC集群的整合

多租户配置中的高可用性和可扩展性

# 多租户和RAC: Affinity

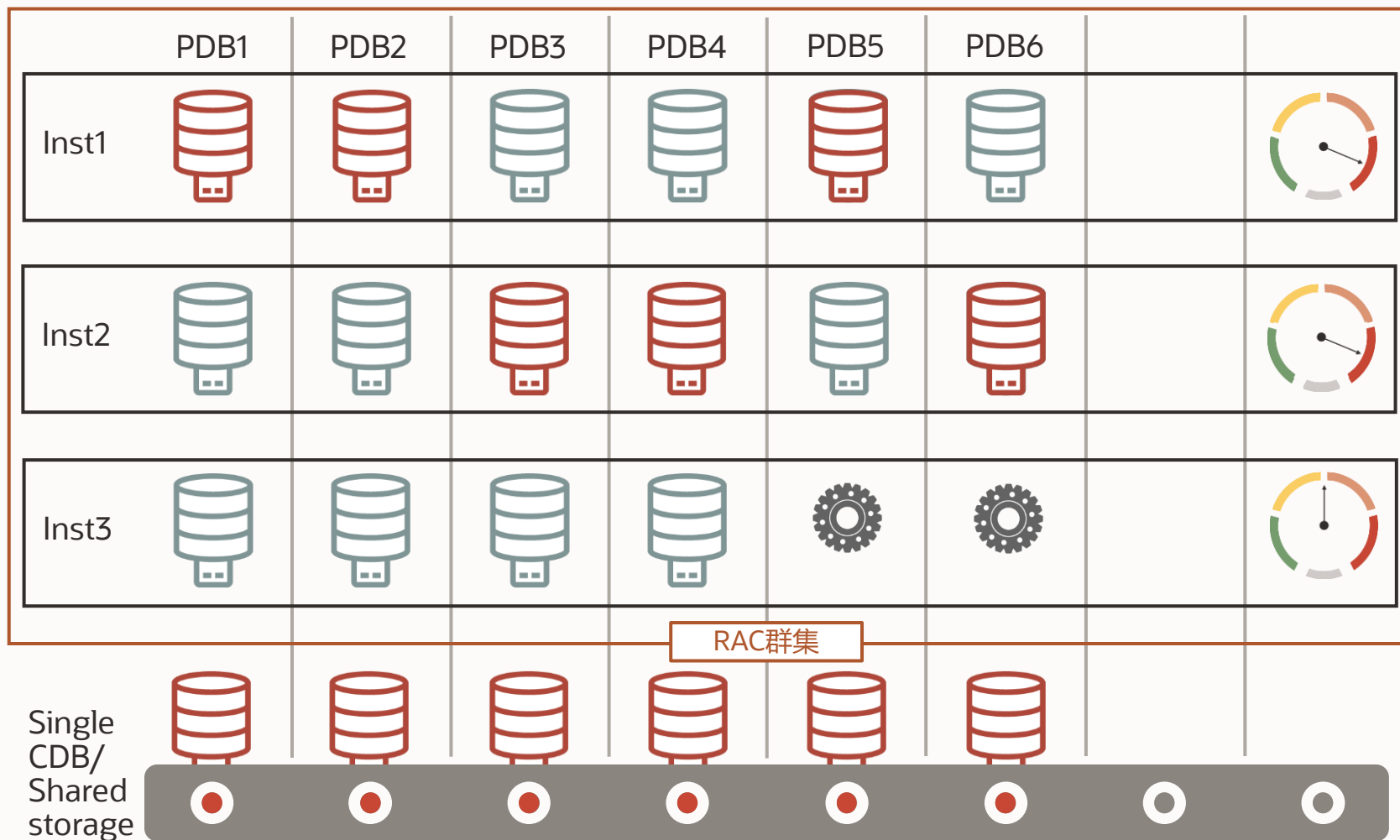


- 单一CDB
- 双节点集群
- 每个节点的单一实例
- PDB对一个节点的 Affinity 通过在那里启动其服务来定义
- 在其他节点中以 "mounted" 状态存在





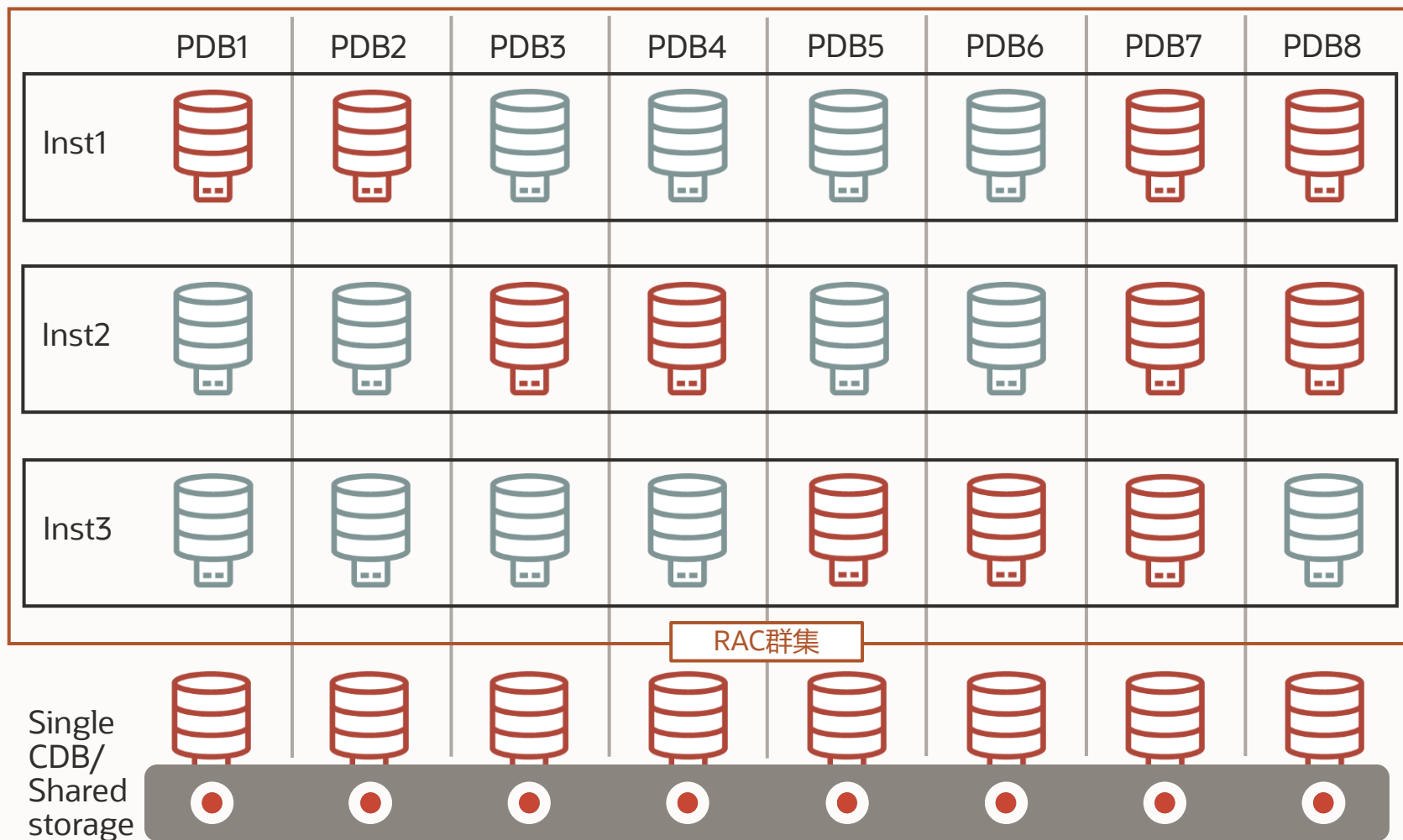
# 多租户和RAC：可扩展性和敏捷性



- 扩展集群
- 重新分配PDBs
  - 通过 Srvctl
  - 或者：  
改变PDB重定位；



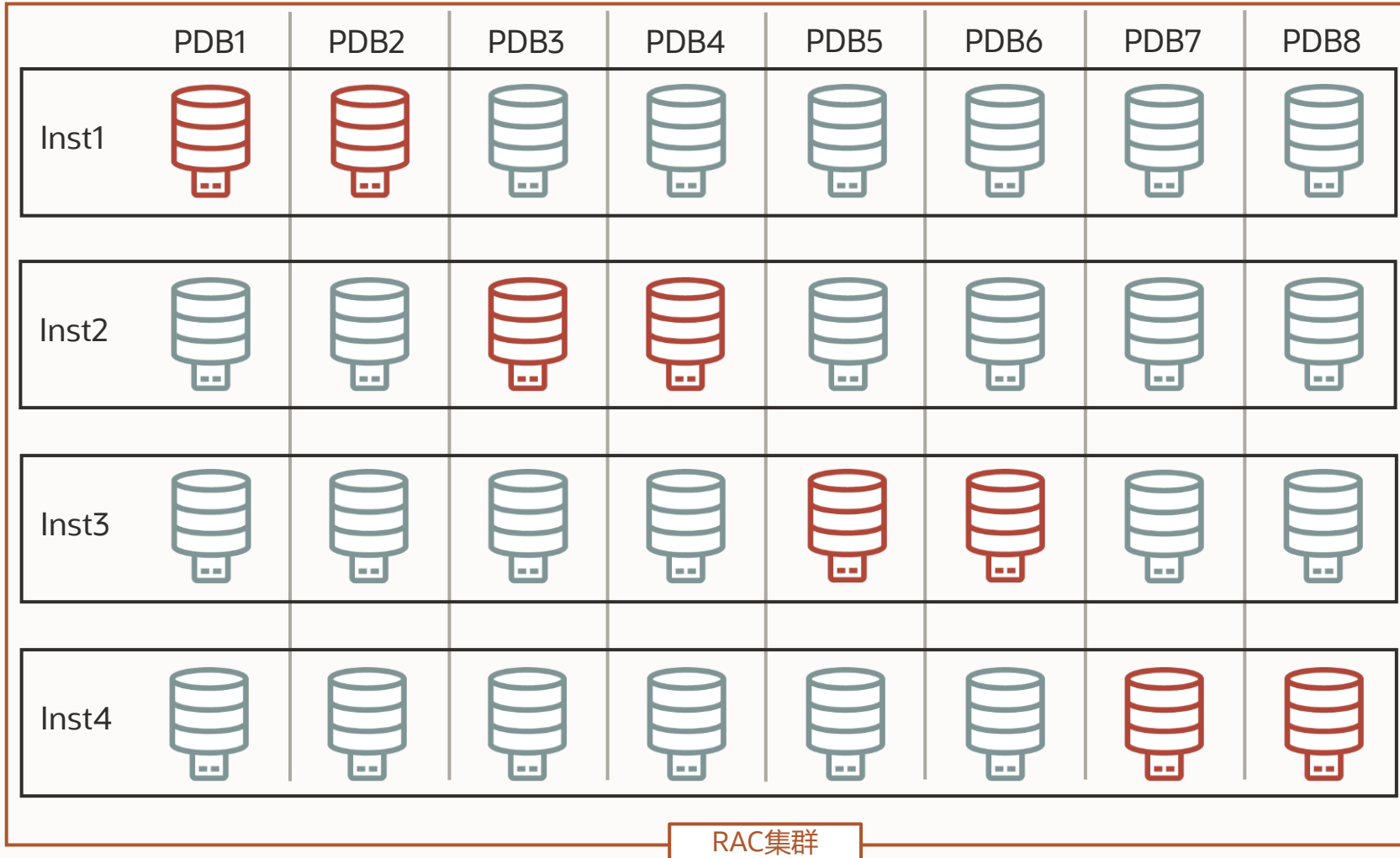
# 多租户和RAC：可扩展性和敏捷性



- 单一CDB
- 每个节点的单一实例
- PDBs可以被配置为与特定节点的 " singleton " affinity
  - 在其他节点中以 " mounted " 状态存在
- PDBs可以在多个节点上打开



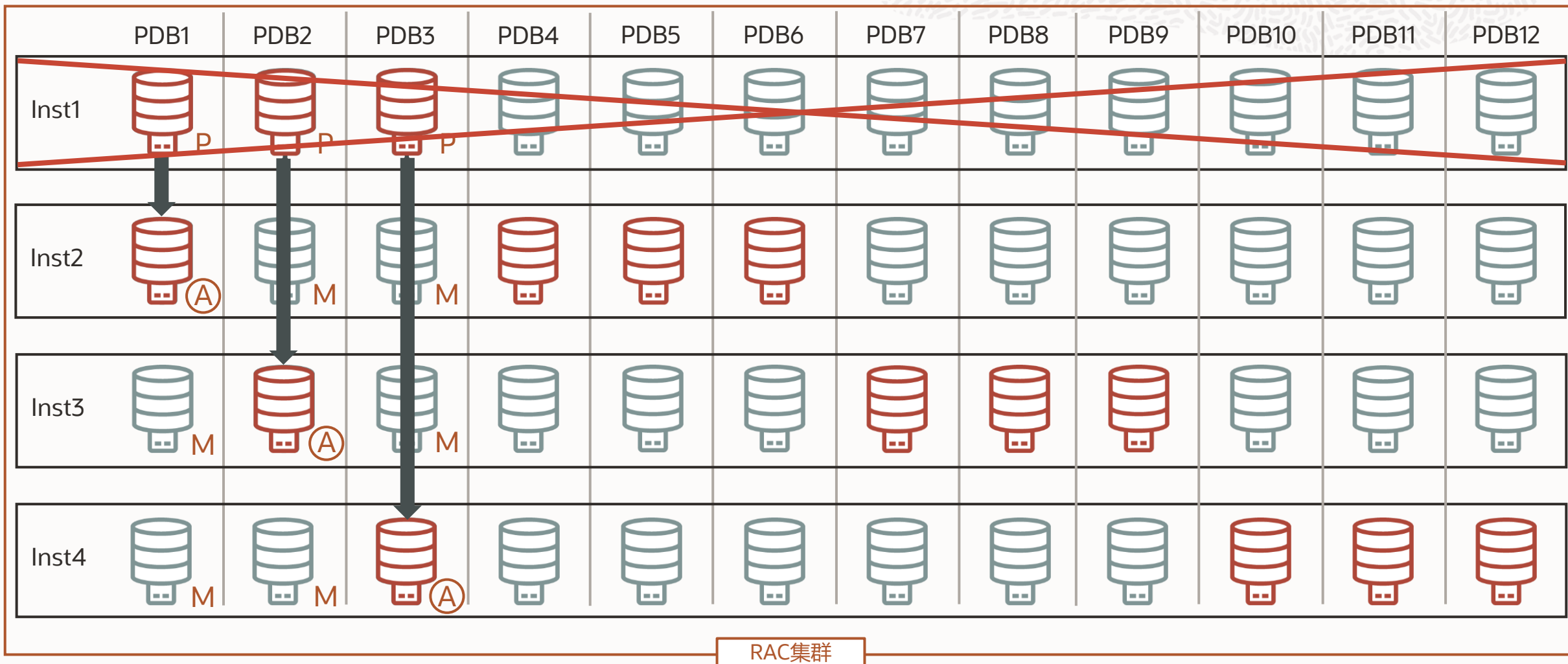
# RAC集群中PDB对实例的 Singleton affinity



- Per-PDB lock domain 提供强大的隔离
- 实例故障后的 **Cache fusion** 可以忽略受影响节点中未打开的PDBs



# 多租户和RAC：可用性



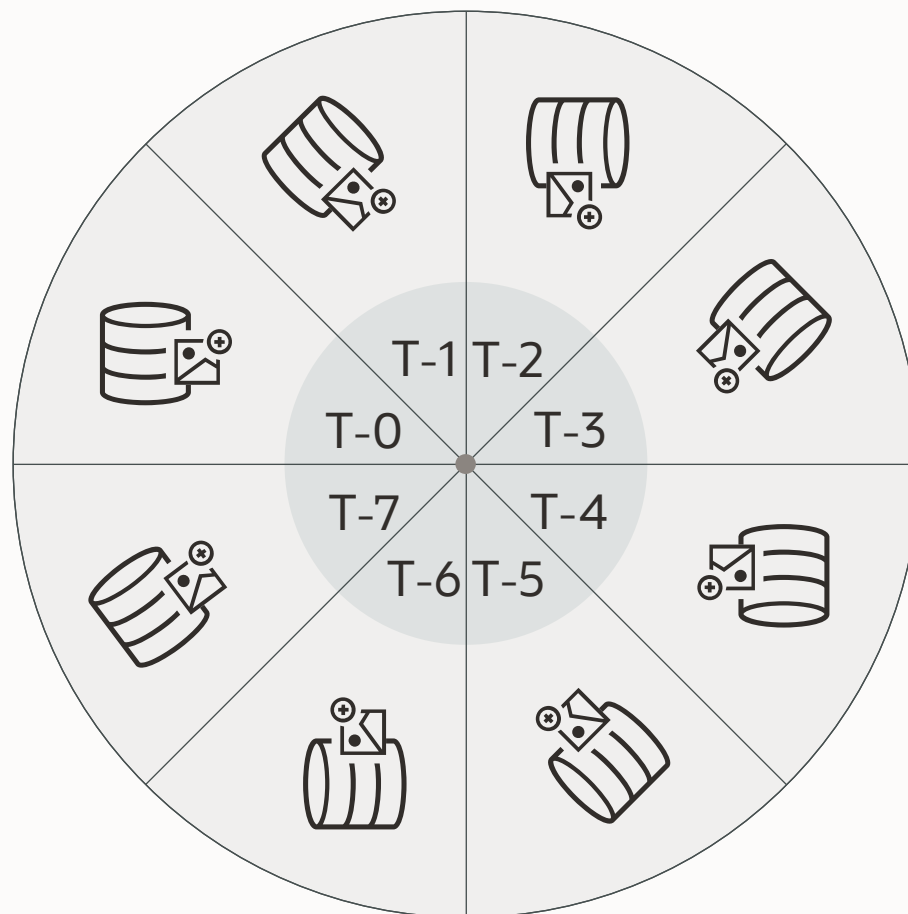
# 快照轮盘

---

Repository for periodic point-in-time copies of a PDB

# 快照轮盘

PDB的周期性point-in-time副本的存储库

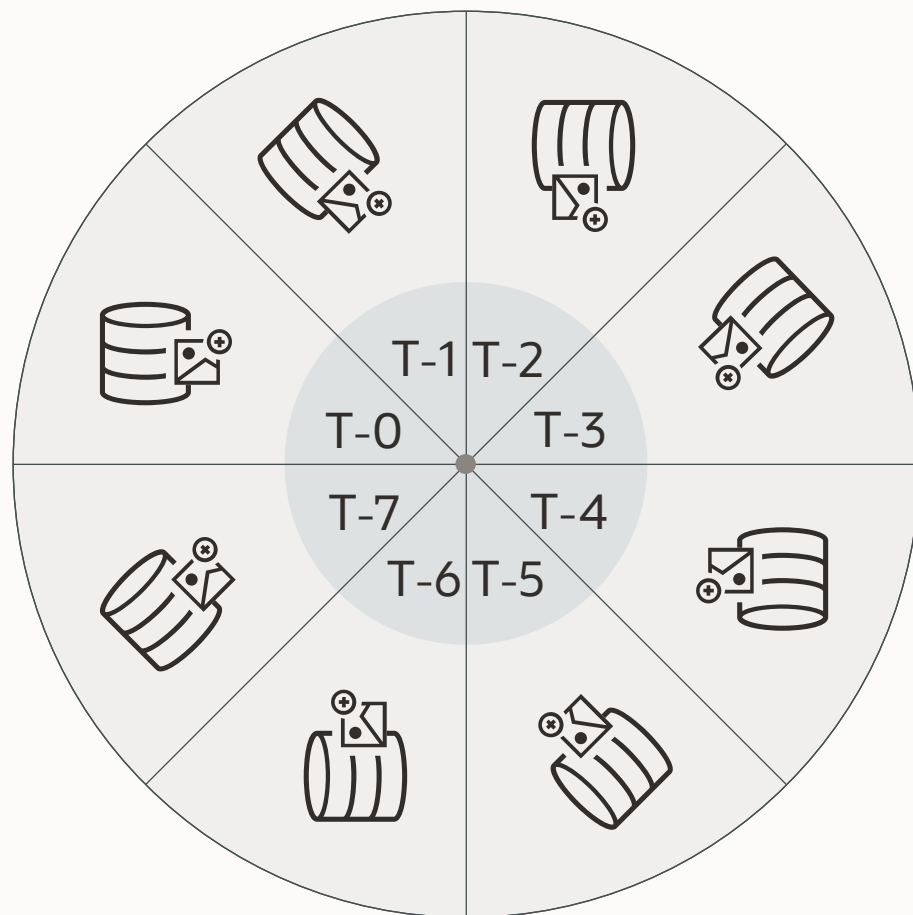
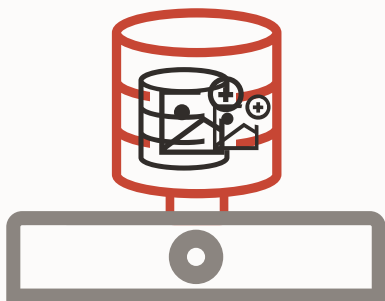


2022年9月  
**12<sup>th</sup>**  
星期一  
#B4PDB 我们需要更多的DB服务器



# 快照轮盘

PDB的周期性point-in-time副本的存储库

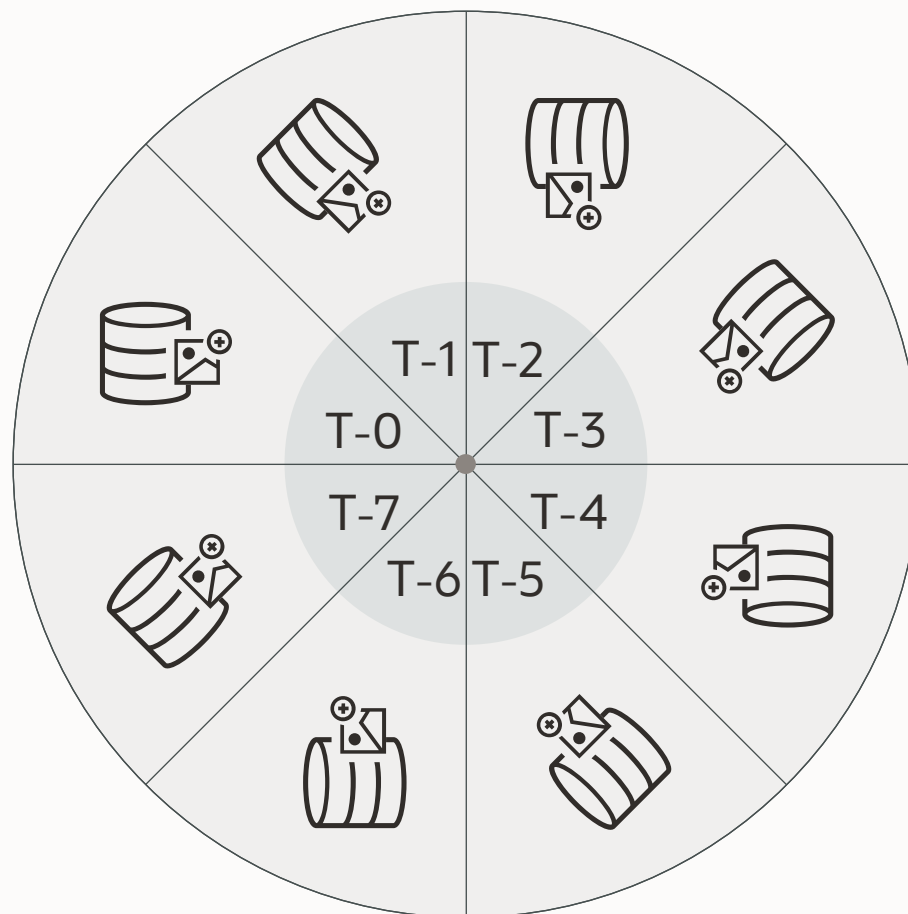


2022年9月  
**13<sup>th</sup>**  
星期二  
#B4PDB 我们单独管理DB



# 快照轮盘

## PDB的周期性point-in-time副本的存储库



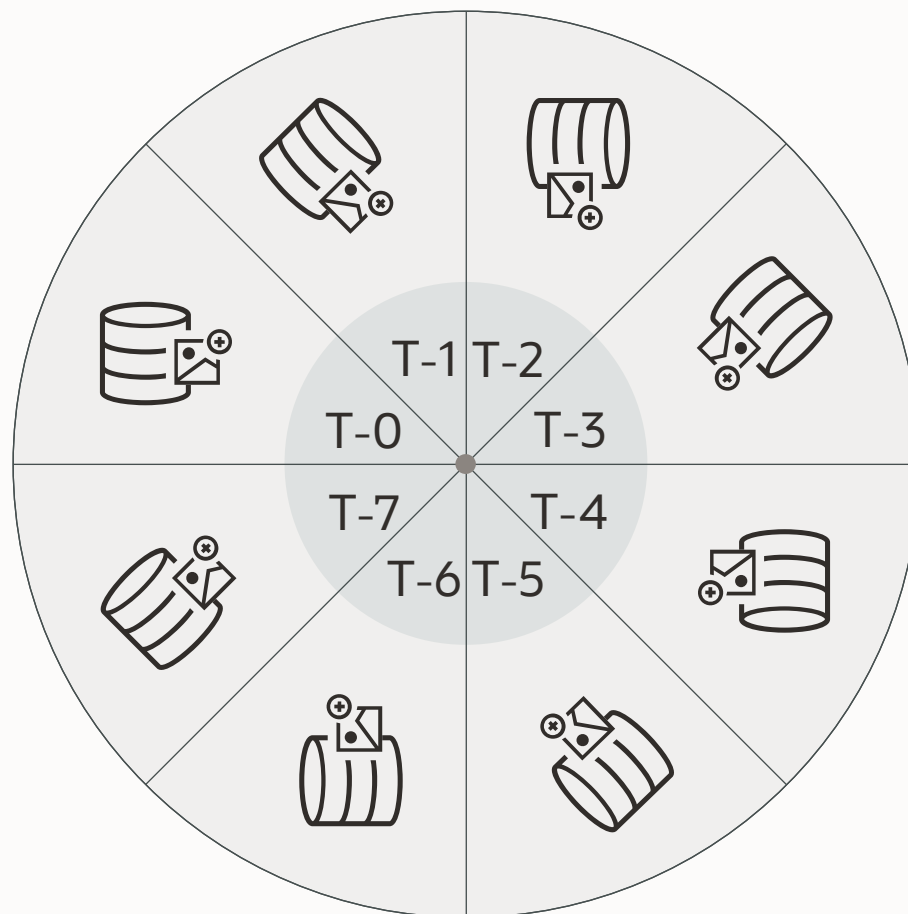
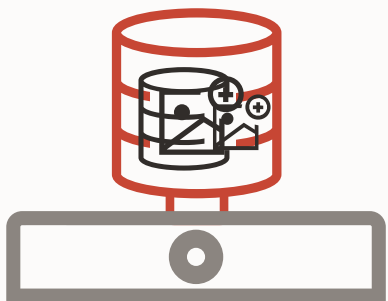
2022年9月  
**14<sup>th</sup>**  
星期三  
#B4PDB 配置新的数据库花了很多时间





# 快照轮盘

PDB的周期性point-in-time副本的存储库

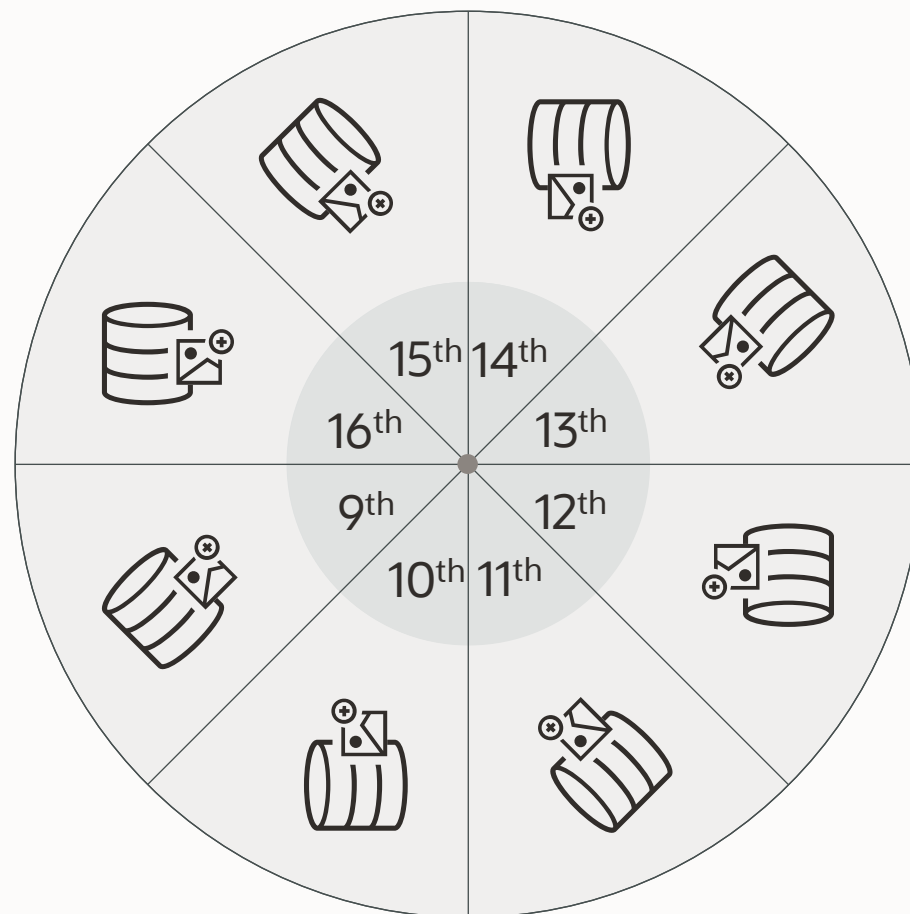
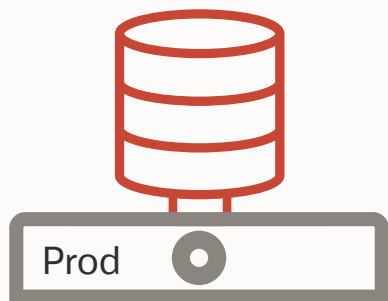


2022年9月  
**15<sup>th</sup>**  
星期四  
#B4PDB 总拥有成本  
高出约2倍



# 快照轮盘

## 调试特定日期问题的理想数据来源

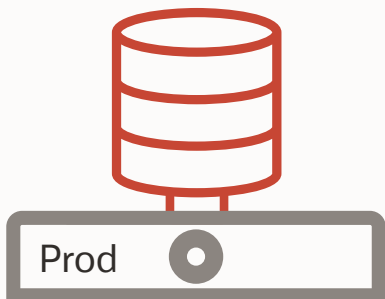


2022年9月  
**16<sup>th</sup>**  
星期五  
#B4PDB生产的克隆  
总是陈旧的



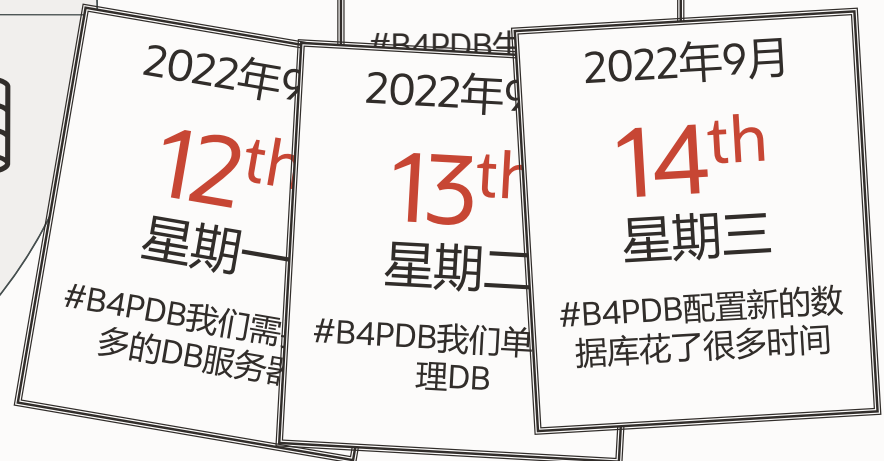
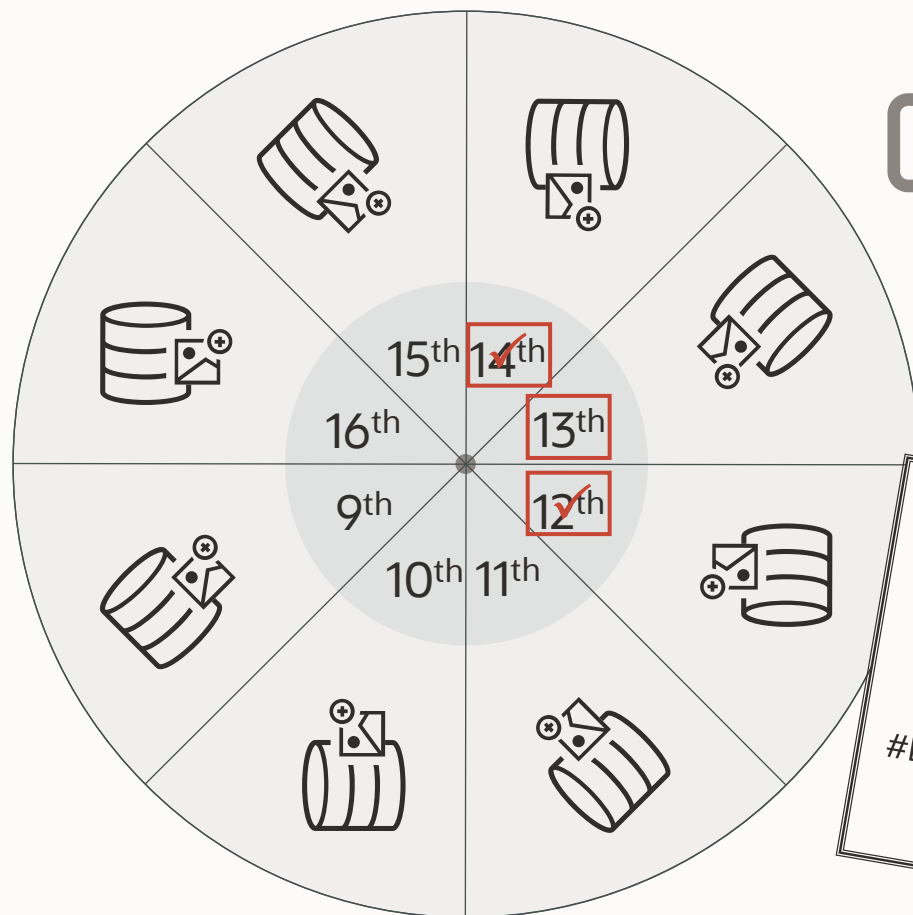
# 快照轮盘

调试特定日期问题的理想数据来源



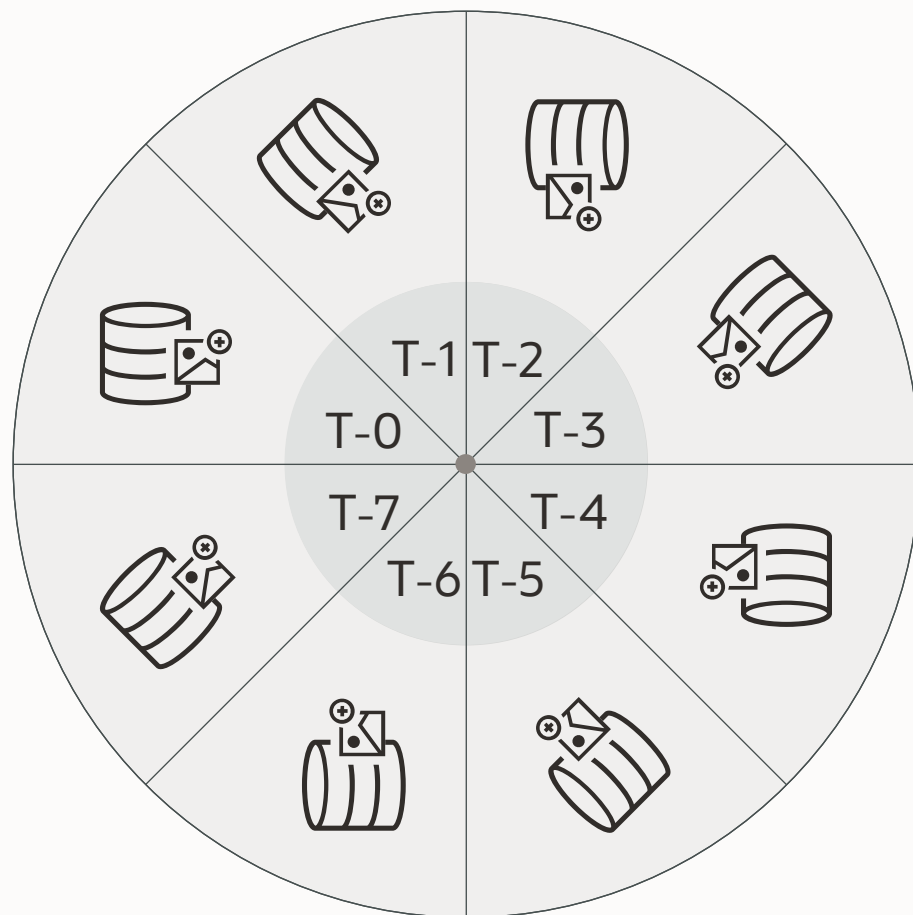
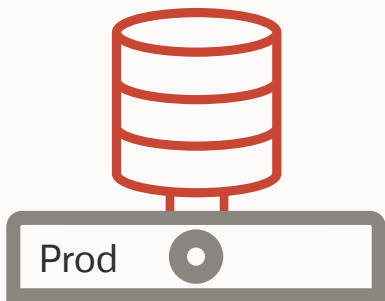
## 特定日期的bug

1. Reported on Friday
2. “Bracket” problem
3. Eliminate Monday
4. Eliminate Wednesday
5. Drill-in on Tuesday



# 快照轮盘

PDB 的point-in-time recovery



2022年9月  
**16<sup>th</sup>**  
星期五  
#B4PDB生产的克隆  
体总是陈旧的



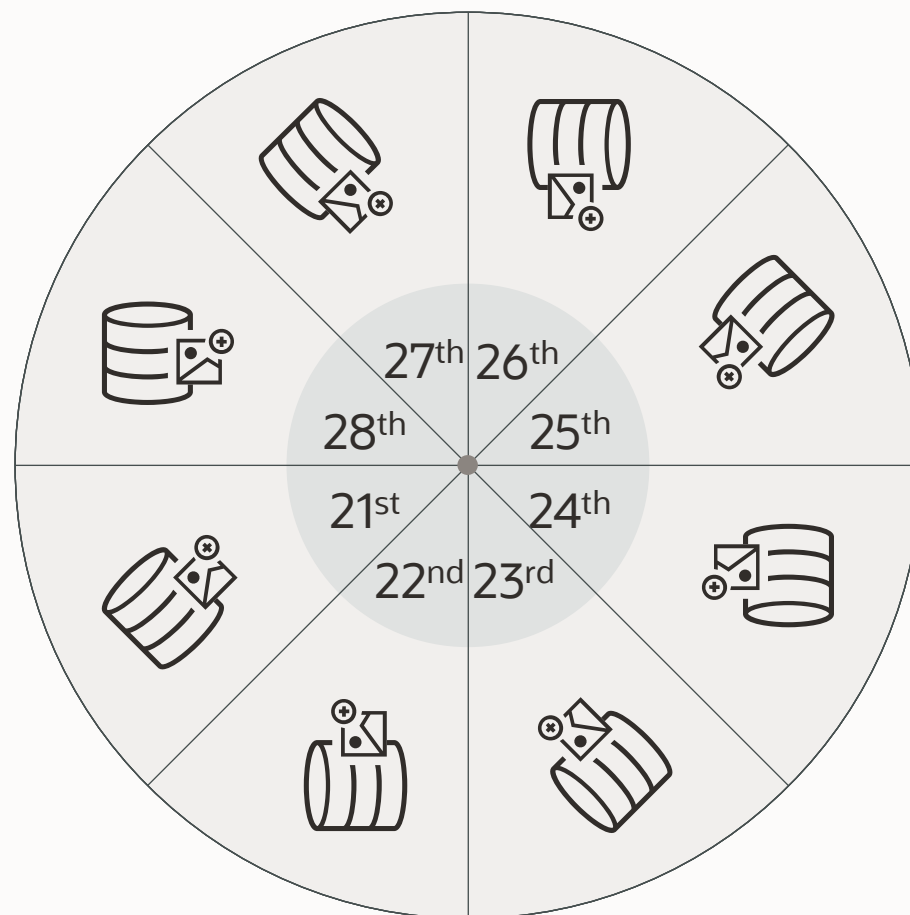
# 快照轮盘

## PDB 的point-in-time recovery



point-in-time恢复

1. 数据错误  
在星期二
2. 从周一的快照中恢复



2022年9月  
**16<sup>th</sup>**  
星期五  
#B4PDB生产的克隆体  
总是陈旧的

用于备份



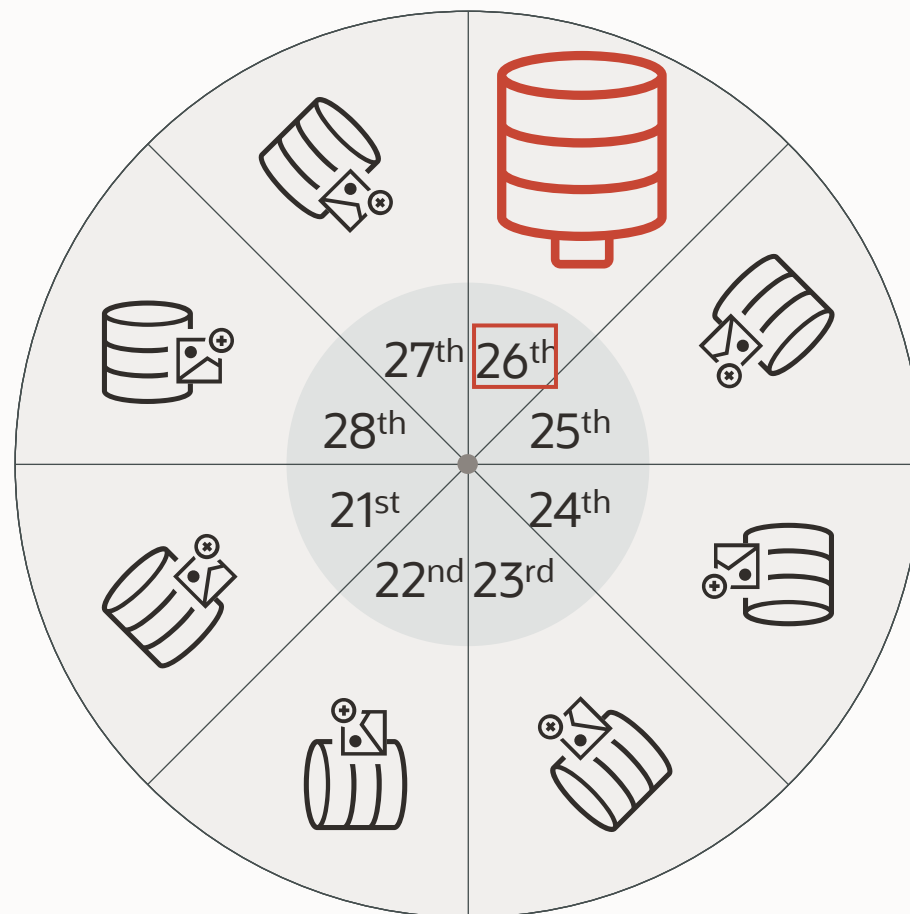
# 快照轮盘

## PDB 的point-in-time recovery



point-in-time恢复

1. 在星期二发生数据错误
2. 从周一的快照中恢复



2022年9月  
**28<sup>th</sup>**  
星期三  
#B4PDB我们没有锁定配置文件



# Transportable 备份

---

消除对可插拔式便携性的障碍

# RMAN: 比较各种备份策略



每周一次的完整备份

原始策略 (糟糕的RPO-周日凌晨2点)

周日凌晨2点

星期一

星期二

星期三

星期四

星期五

星期六

周日凌晨2点

完整备份  
(所有数据文件)

完整备份  
(所有数据文件)





# RMAN: 比较各种备份策略

## 每周一次的完整备份

原始策略 (糟糕的RPO-周日凌晨2点)



## 每周一次的完整备份, 归档日志

可容忍的 (良好的RPO, 不良的RTO)。



# RMAN: 比较各种备份策略

## 每周一次的完整备份

原始策略 (糟糕的RPO-周日凌晨2点)



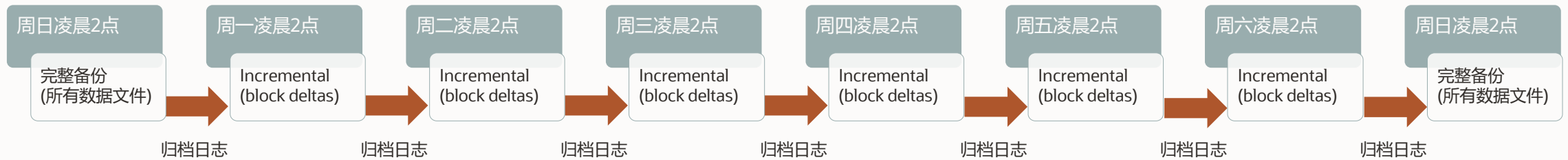
## 每周一次的完整备份, 归档日志

可容忍的 (良好的RPO, 不良的RTO)。



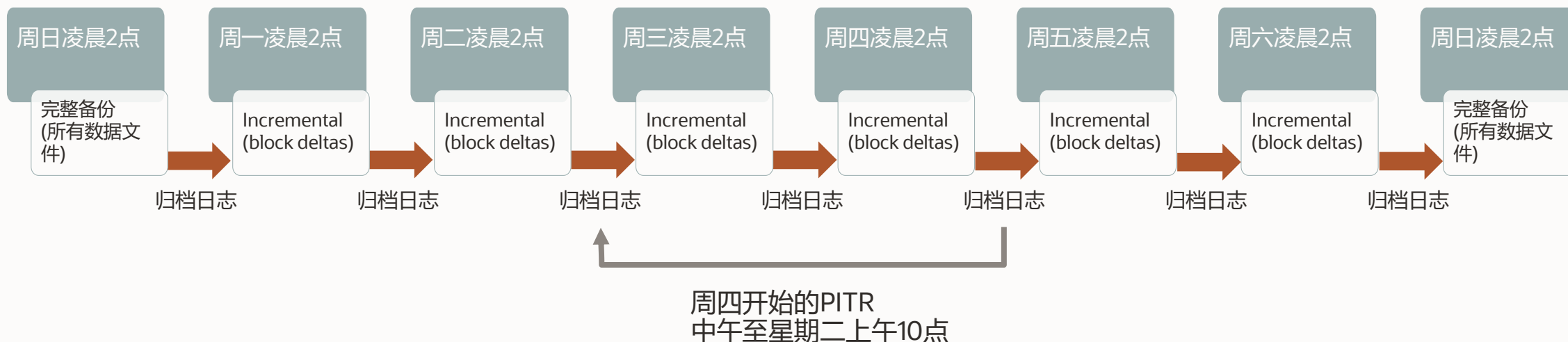
## 每周一次的完整备份, 每天一次的增量备份和归档日志

非常好的 (良好的RPO和良好的RTO)。



# 典型的RMAN用例：时间内点恢复（PITR）

例子：在星期四的午餐时间决定我们要恢复到星期二上午10点

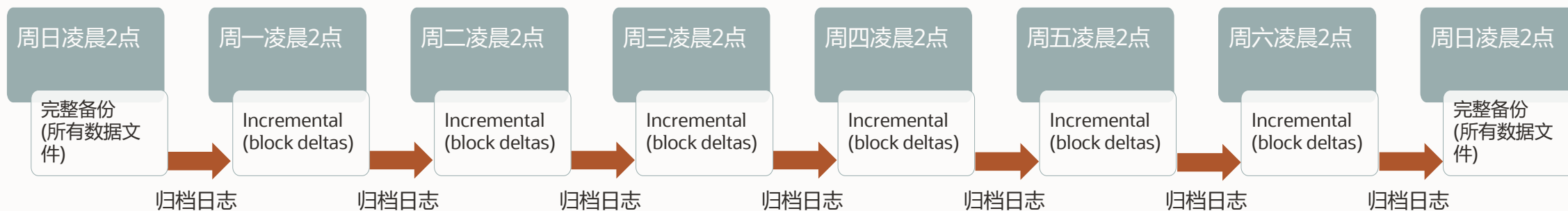


- RMAN恢复将自动：
  - 恢复上周日的完整备份
  - 应用周一和周二的增量
  - 使用归档日志来恢复从星期二凌晨2点到星期二上午10点的数据



# 在18c之前，备份是不能传输的

在实践中，这可能会限制在CDB之间移动PDB的敏捷性



PDB重定位到另一个CDB

- 周一中午:
  - 将PDB移至另一个CDB
  - 这具有使旧的备份成为 "orphaning "

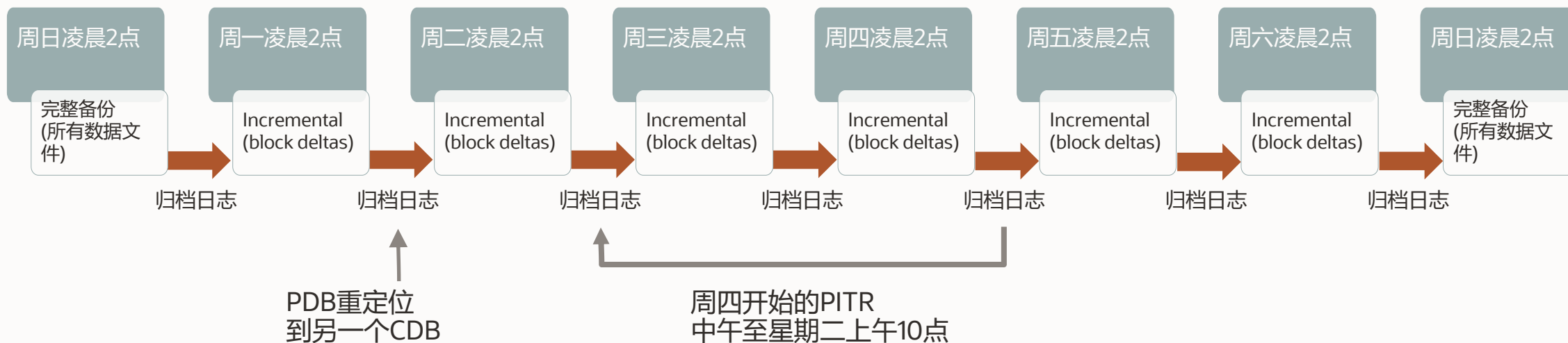
PITR可以、但是很复杂!

- 在这种情况下，PITR更为复杂
  - 恢复到辅助实例
  - 恢复到插拔前
  - 手动移动数据文件到新的CDB
  - 向前滚动到所需的point-in-time



# Transportable 备份

## 消除对可插拔式便携性的障碍



- 尽管在CDB之间移动PDB， RMAN会自动恢复：
  - 恢复上周日的完整备份
  - 应用周一和周二的增量
  - 使用归档日志来恢复从星期二凌晨2点到星期二上午10点的数据
- 无论PDB是通过拔出/插入还是在线重定位来移动，都能发挥作用



# 操作的好处 - PDB克隆



多租户之前

有了多租户

DB尺寸	DB克隆	本地PDB克隆 (同一服务器)	远程PDB克隆 (通过局域网)
5GB	5分钟	几秒钟	几秒钟
100GB	15分钟	1分钟	2分10秒
500GB	1小时20分钟	4分55秒	9分30秒
1000GB	2 H	8分20秒	17分钟

- 提供测试环境的刷新**速度**约为**7倍**
  - DWH从20小时到3小时以下。
  - 整个过程的标准化



# 真实世界的客户案例

## Database-Blog

AN INITIATIVE OF MOBILIAR IT

Home

The DB-Blog Team

Contact

Papers / Docs

About the company

Want to join Mobiliar?

### Consolidating 350 PDB's in less than 6 hours

Before Christmas is usually the time to clean up and finish things so that you can start the new year fully motivated. We also took this...

December 16, 2019 | 4 Comments | [Read article](#)

#### RECENT POSTS

- [Consolidating 350 PDB's in less than 6 hours](#)  
December 16, 2019
- [DOAG 2019 – Save the Date for December 22, 2019](#)
- [So long and thanks for all the fish](#)  
2019



ORACLE



# Oracle 数据库高级安全特性， 许您一个安全的未来



李炜玉

- 资深解决方案工程师
- 16年Oracle工作经历
- 专注金融行业解决方案

## 内容简介

- Oracle数据库安全架构概览
- 19c高级安全特性
- 利用高级安全特性透明地加密数据



Zoom直播

直播时间：4月7日 11:00 - 12:00  
扫描二维码进入直播  
Zoom ID: 976 6962 5763  
密码: 98039717



微信扫一扫预约



数据库和云讲座群

20-20



甲骨文云技术公众号



技术专家1V1深入交流

