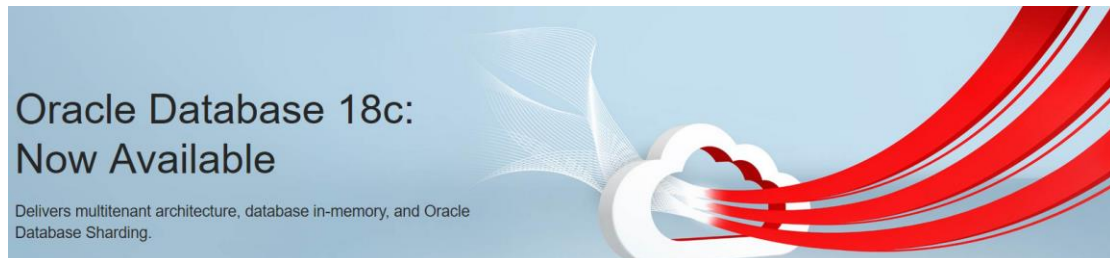


Oracle Database 18c new feature

Oracle Database 18c 是世界上最新一代最受欢迎的数据库，她在高性能、高可用性、安全性、易用性和其他方面都有显著的变化。如果要用一个词形容的话，我觉得“走心”这个词很合适。在介绍 Oracle Database 18c 的新特性之前，在这里先回答几个问题，这也可能是大家比较关心的。



Oracle Database 12c 之后的版本是如何命名的？

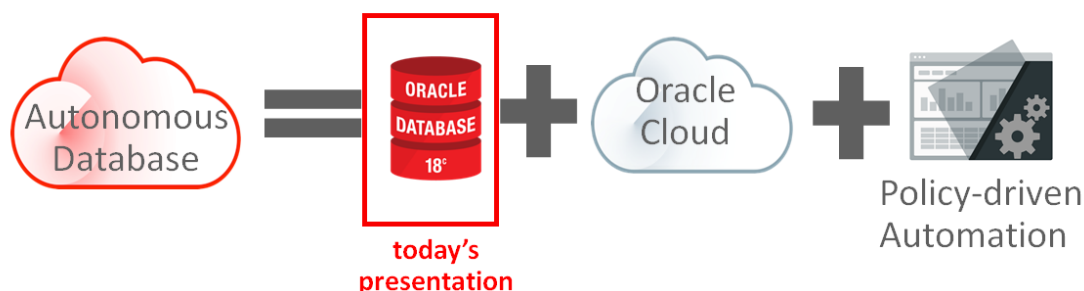
在中国农历戊戌狗年（2018-02-16）的第一天，Oracle 发布了最新的 Oracle Database 18c。她的到来不仅标志着世界上最流行的数据库发布方式的改变，也是一个全新意义上的开始。Oracle Database 18c 是该产品的第一个版本，遵循每年发布的模式。从这里开始，Oracle 数据库每年都会随着季度更新发布，以反映产品发布的年份。您可以通过访问 Oracle MOS(My Oracle Support)来查看文档 2285040.1。下面就是 2018 年的 Oracle Database 18c 的发布日期和相应版本的名称。

Production	April	July	October
18.1.0	18.2.0	18.3.0	18.4.0

Oracle Database 18c 和 Oracle Autonomous Database 一样吗？

在 2017 年的 Oracle Open World 上，Larry Ellison 宣布了全球第一个自治数据库（Autonomous Database）的诞生，她有三个特点 Self-Driving（自治驾驶），Self-Securing（自治安全）和 Self-Repairing（自治修复）。Oracle Autonomous Database Cloud services 是一项云服务，基于 Oracle Database 18c，Oracle 的云基础设施(包括 Cloud at Customer)和云工具。可用性方面达到 99.995%，也就是说平均每月宕机时间小于 2.5 分钟。如果您购买的是部署在 Oracle Cloud 或 Cloud At Customer 上的 Autonomous Database Cloud Service，那么您获得的就是自治数据库服务。下面是 AutonomousDatabase Cloud Service 的示意图：

Autonomous Database and Oracle Database 18c



Oracle Database 18c 包含了改进的自动化功能，性能优化，以及其他增强功能，但是没有 Oracle 云基础设施和云工具提供的功能，她和之前版本的 Oracle Database 一样，不是自治数据库。下面这三种情况不是自治数据库：

- 如果您在任何云供应商平台上安装 Oracle Database 18c，那么您获得的不是自治数据库服务
- 如果您是在本地的数据中心安装 Oracle Database 18c，那么您获得的不是自治数据库服务
- 如果您购买的是部署在 Oracle 公有云或者 Cloud At Customer 上的 DBaaS，那么您获得的不是自治数据库服务

Oracle Database 18c 这个版本稳定可靠吗？

Oracle Database 18c 是 Oracle Database 12cR2 的下一个迭代，如果现在还太不习惯的话，暂且将她叫做 12.2.0.2 也可以，是对 Oracle 数据库 12c 中已有功能的增强和改进，还引入了一些新的功能。截止目前，已经可以 Oracle Cloud and Oracle Engineered Systems 上使用。

如何体验 Oracle Database 18c？

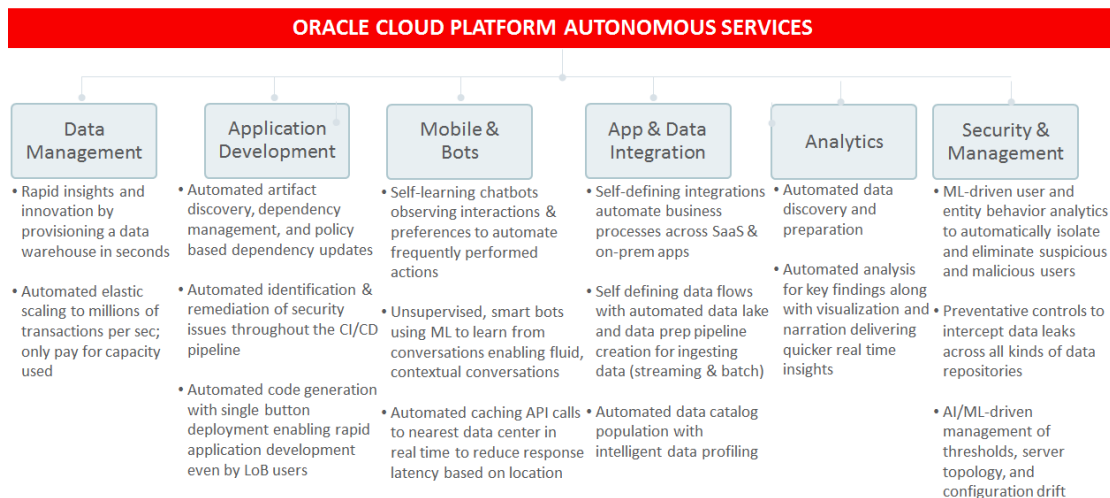
截止目前，Oracle Database 18c 只能在 Oracle 云和 Oracle 工程系统上使用。其他平台的版本后续会陆续的释出。我们可以以下两种方式：

1. 注册 Oracle 云账号（首次注册有 300\$赠送），然后访问 <https://cloud.oracle.com> 登录体验
2. 通过 Livesql 网站 <https://livesql.oracle.com>

Oracle 除了自治数据库云服务外，还有哪些自治服务？

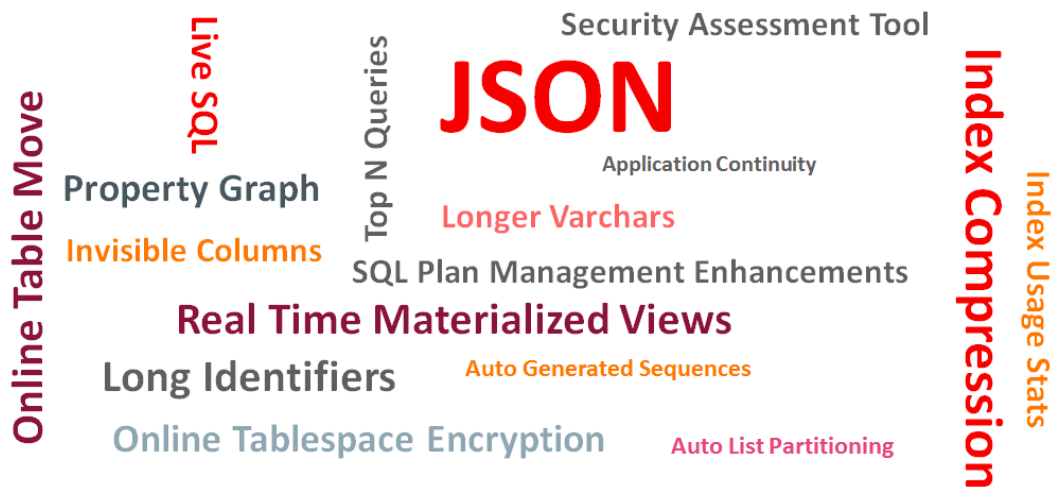
除了自治云数据库服务外。Oracle 在云端还提供了应用开发，移动化和智能聊天机器人，大数据分析，数据整合，以及在安全和运维管理等自治服务。可以为企业节省成本，降低风险，尤其在持续性的创新，持续性的交付和预测性分析方面表现更佳。

Oracle Cloud Platform Autonomous Services



我们知道 Oracle Database 12c 引入了多租户 (Multitenant), 内存数据库 (Database In-Memory), 数据库分片 (Oracle Database Sharding) 这些新功能。据不完全统计, 在 Oracle 12cR2 中已经引入了 600 多项新功能。

Oracle Database 12c “Under the Radar” Features



总的来说, Oracle Database 18c 更接地气, 青出于蓝而胜于蓝。无论在对开发人员和运维人员的支持, 还是在高可用性, 高性能方面都有着不俗的表现, 这也是企业迫切需求的。下面就让我们来一起了解以下这些主要的新功能在 18c 上的体现, 想了解更多请参考 Oracle 的官方白皮书。

先看几个小特性。虽然小, 但是功能很强大, 很实用。

1. SET FEEDBACK ON SQL_ID

启用 SQL_ID 时后, 在执行完一个 SQL 语句时, 会自动的显示这个语句 SQL_ID, SQL_ID 在定位问题和优化时非常有用

```
SQL> SET FEEDBACK ON SQL_ID
```

```

SQL> SELECT * FROM DUAL;

D
-
X

1 row selected.

SQL_ID: a5ks9fhw2v9s1

SQL> COLUMN sql_text FORMAT a50

SQL> SELECT sql_text FROM v$sql WHERE sql_id = '&_sql_id';

SQL_TEXT
-----

SELECT * FROM DUAL

1 row selected.

SQL_ID: cf9bgxbfytv5b

```

2.OPTIMIZER_IGNORE_HINTS, OPTIMIZER_IGNORE_PARALLEL_HINTS

这两个初始化参数很有用。你可能会碰到由于 HINT 造成的各种性能问题，眼睁睁的看着，但没有办法修改。现在有了这个利器，你就偷着乐吧

3.ALTER SYSTEM CANCEL SQL

通过这个命令可以取消当前执行的查询语句，如果是 DML 语句，事务会回滚。可用于终止脱缰的 SQL 语句，这个很不错

```

ALTER SYSTEM CANCEL SQL 'SID, SERIAL, @INST_ID, SQL_ID';

ALTER SYSTEM CANCEL SQL '20, 51142, 8vu7s907prbgr';

```

4.Scalable Sequences

可伸缩序列。有了这个新特性，在大规模并发的环境下可以大大减少了序列和索引块的争用，尤其序列作为表的主键时。下面是创建语句的样本

```

CREATE | ALTER SEQUENCE sequence_name

...

SCALE [EXTEND | NOEXTEND] | NOSCALE

...

```

5.Private Temporary Table

私有临时表。只在内存中存在，是临时对象，在只读的数据库上使用非常方便，比如在 Oracle Active Data Guard 环境下

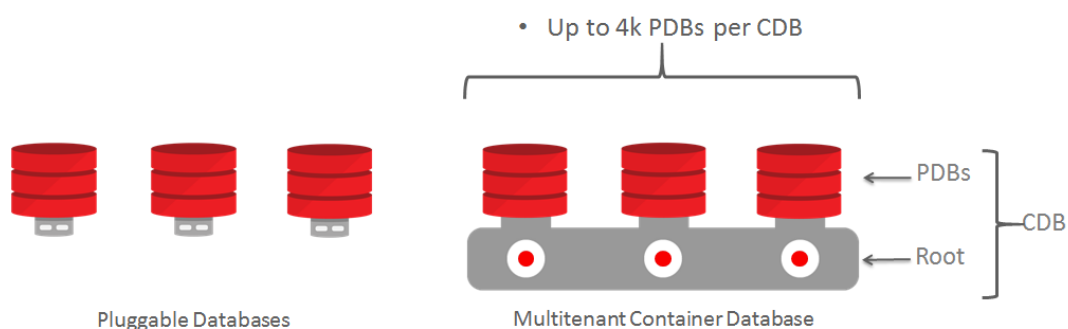
言归正传，下面就让我们一起来看看 Oracle Database 18c 在 Multitenant, Database In-Memory, Sharding, High availability, Data Warehousing and Big Data 和 Security 方面那些比较实用、接地气的新特性。这里只是抛砖引玉，希望大家去发现更多有价值的信息。

Multitenant

在 Oracle Database 12c 中引入了可插拔数据库（pluggable database）的概念，我们最熟悉的可能就是她了，这使用户能够自由地插拔数据库，并将其移动到本地或云中的其他容器中，截止目前在一个容器数据库中最多支持 4096 个可插拔数据库。在这种架构下，大规模的数据库整合、管理、打补丁、备份等操作作为一个整体来对待，节省了成本，大大简化了运维人员的工作量，还提升了系统的灵活性。后来又增加了热克隆（Hot Clone），在线迁移（Online Relocate），在 PDB 的级别上提供 IO、CPU 和内存资源控制的功能，以及非容器中可用的所有特性都可以用于 PDB (Flashback 数据库等)。在 Oracle Database 18c 中又有了新的功能，在这里我们介绍两个非常实用的功能 Refreshable PDB Switchover 和 PDB Snapshot Carousel。

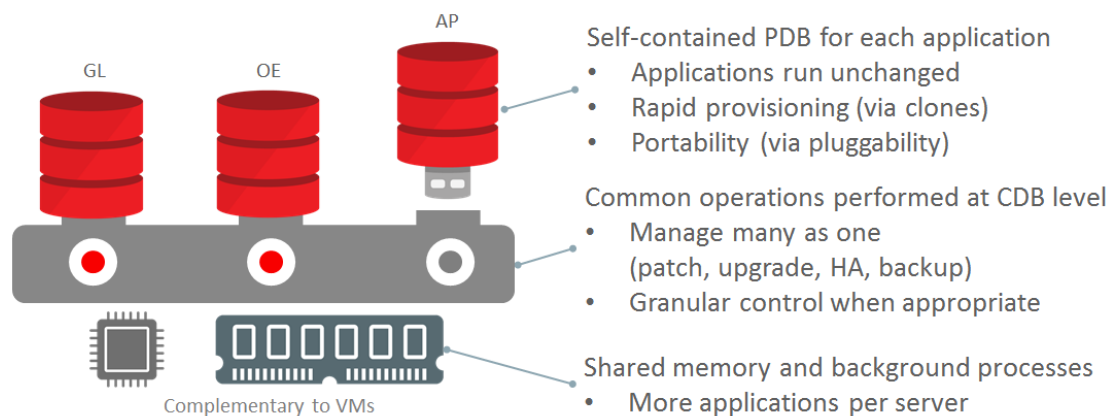
Multitenant Architecture

Components of a Multitenant Container Database (CDB)



Advantages of Multitenant Architecture

Reduced CapEx & OpEx, Increased Agility, Easy to Adopt and Use



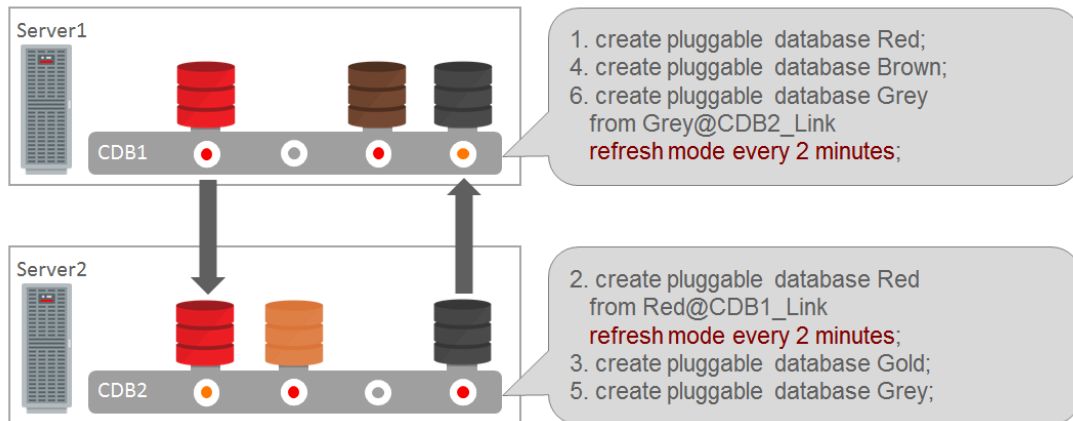
Refreshable PDB Switchover

A refreshable clone PDB 是一个克隆的且只读的可插拔数据库，可以周期性地和它的源 PDB 同步。通过这种方式，就相当于我们为数据库的高可用性增加了一层保障。在这种情况下，两个数据库一个是主库（read write 模式），另外一个是从库（read only 模式），有点类似数

据卫士（Data guard）的意思，并且两个库的角色是可以相互转化的。在正常情况下可以起到负载均衡的作用。在异常情况下，从库还可以变成主库，在最坏的情况下会丢失刷新间隔的数据。还可以结合 Data Guard 的场景使用，更多功能请查看官方白皮书。

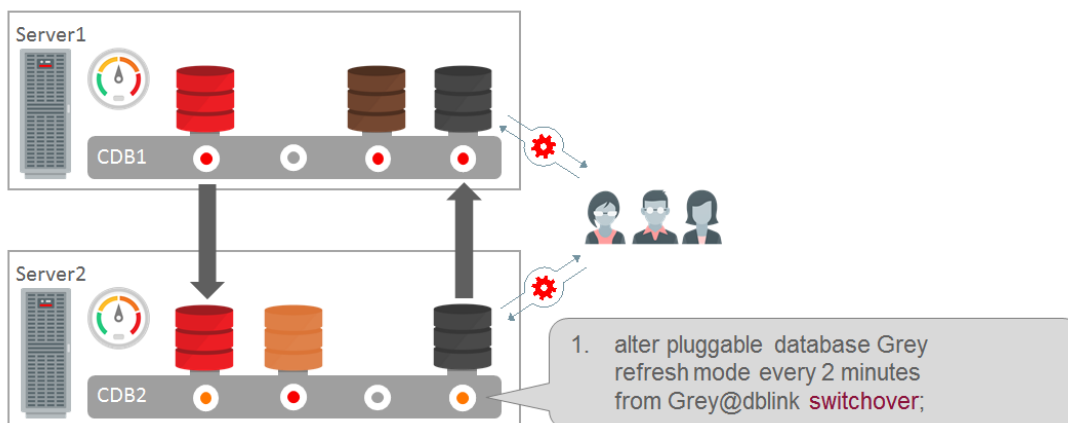
Refreshable PDB Switchover

Per-PDB replica with only two CDBs to manage!



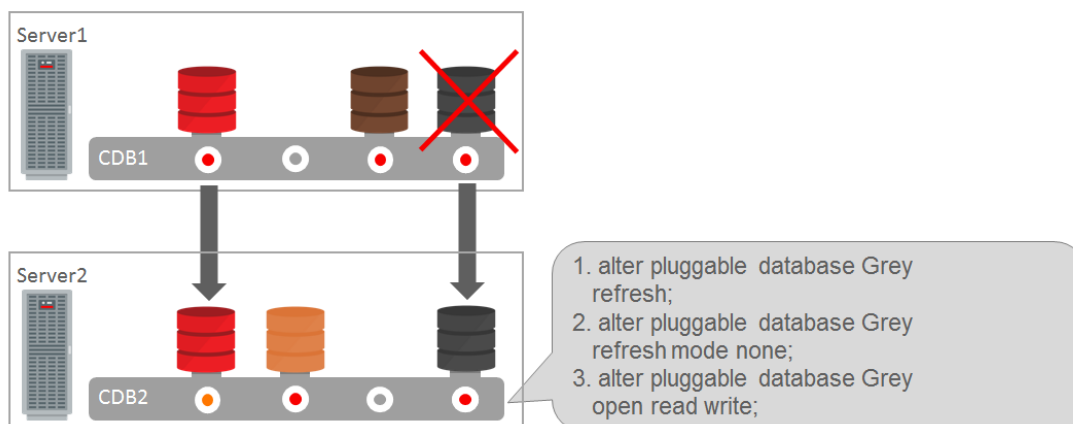
Refreshable PDB Switchover

Planned switchover



Refreshable PDB Switchover

Unplanned switchover



PDB Snapshot Carousel

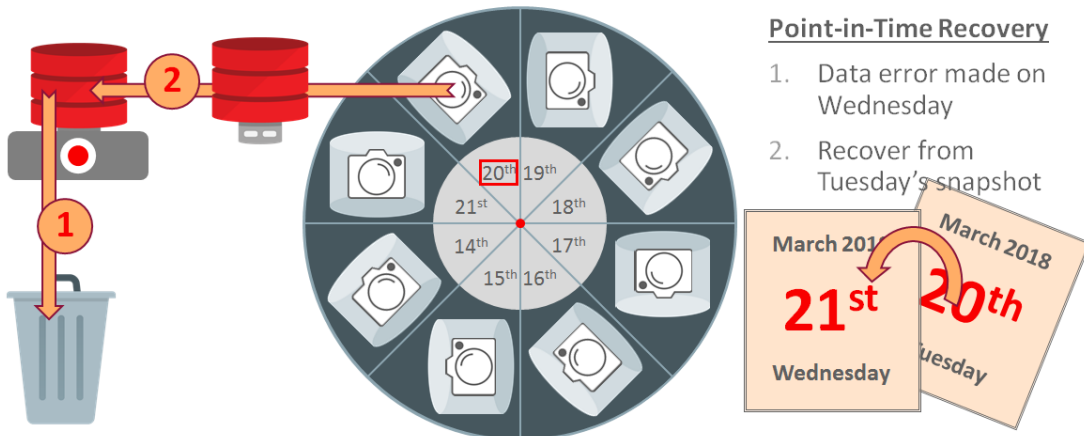
PDB Snapshot 是 PDB 在特定时间点的命名副本。这个功能有点类似虚拟机中的快照。当为 PDB 启用快照功能后，您可以创建最多 8 个（类似一个旋转的转盘一样，当多余 8 个后，最老的会被删除）快照。PDB Snapshot Carousel 维护了一个经常刷新的 PDB 副本库，您可以在 carousel 中执行任何时间点的克隆或时间点恢复。一个典型的用例是克隆一个 PDB 来进行开始或测试，可以每天克隆一个，需要时从克隆的环境中复制一个就好。一个就是用于基于时间点的恢复，假如你准备在生产库上装载数据，在这之前创建了一个快照，装载数据时意外地造成了生产库的损坏，这时我们就可以先创建一个基于快照的数据库，然后重新装载数据，使用起来非常方便。下面是一些脚本样例。

```
CREATE PLUGGABLE DATABASE cdb1_pdb3 FROM cdb1_pdb1
FILE_NAME_CONVERT=('cdb1_pdb1','cdb1_pdb3')
SNAPSHOT MODE EVERY 120 MINUTES;
```

```
ALTER PLUGGABLE DATABASE SNAPSHOT MODE EVERY 24 HOURS;
```

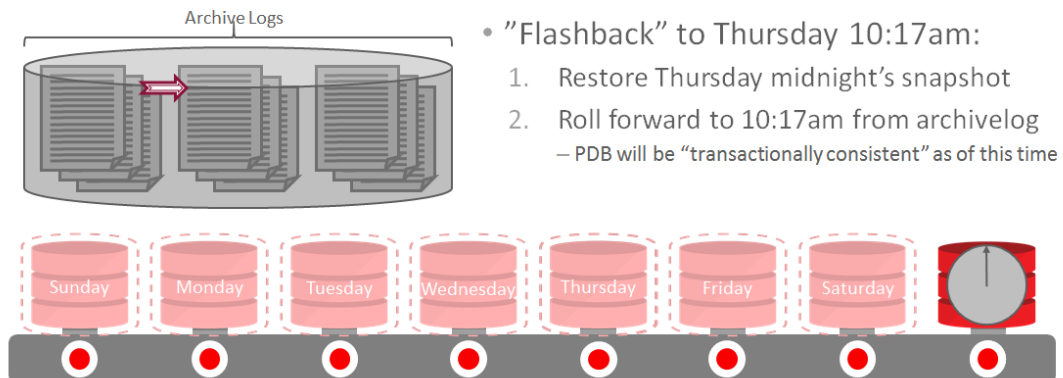
Snapshot Carousel

Convenient source for a simple point-in-time recovery of a PDB



Point-in-Time Recovery with Snapshot Carousel

Carousel stores archive logs along with corresponding snapshot clones



Point-in-Time Cloning with Snapshot Carousel

- “Hot Clone” introduced with 12.2
- Basic hot clone is *transactionally consistent* with source PDB *as of SCN at completion of statement*

```
alter pluggable database Hot_Clone open;
```

- Snapshot Carousel enables hot clone to a specific point in time (or SCN)
 - Restore most recent snapshot from carousel
 - Roll forward to required point in time by performing recovery from corresponding archive log

Transportable Backups

在 Oracle Database 18c 之前，一个容器数据库可能包含了很多 PDB，如果你只想把其中一个 PDB 还原到另外一个数据库，是这样的步骤，我们需要先还原 root，然后是 seed，最后还原我们需要的 PDB

```
restore database root ;
restore database "PDB$SEED";
restore database PDB1;
```

从 Oracle Database 18c 开始，通过 Preplugin Backups 的新特性，我们可以将单个 PDB 还原到一个新的数据库，下面是简单的步骤

```
catalog preplugin <archivelog that are generated after unplug>;
restore pluggable database <pdbname> from preplugin;
recover pluggable database <pdbname> from preplugin;
restore pluggable database <pdbname> skip preplugin;    <<== optional
recover pluggable database <pdbname>;
alter pluggable database <pdbname> open;
```

Database In-memory

Oracle 数据库允许用户采用双内存模型（行格式和列格式）。在这种情况下，OLTP 系统在内存中的行数据可以被高效地更新，而列数据可以被更快地扫描和聚合。这些对于应用程序来说是透明的，无需做任何更改。还可以利用 Software in Silicon 的特性来分析每秒数十亿行的数据，这意味着过去花费数小时的报告现在可以在几秒钟内被完成。

在 18c 以前的版本，我们很难知道哪些 Segment 将从 IM 列存储中获益最多。虽然 In-Memory Advisor 工具可以给我们指导，但实际的使用量是判断的最佳方式。Automatic In-Memory 就可以帮助我们实现这一点，她借助热图数据（Heat Map Data），列统计数据（column statistic）和其他相关统计数据来自动的管理 IM 列存储数据，无需人为干预（之前我们需要通过 alter <> inmemory 手动来实现）。通过初始化参数 INMEMORY_AUTOMATIC_LEVEL 来控制，默认是关闭的（OFF），可以设置为 MEDIUM 或 LOW 来启用。除此之外，还支持 LOB 对象，在对 JSON 的处理，多表连接查询等都有不错的效果。还支持外部表，这个在数据仓库分析方面非常实用。

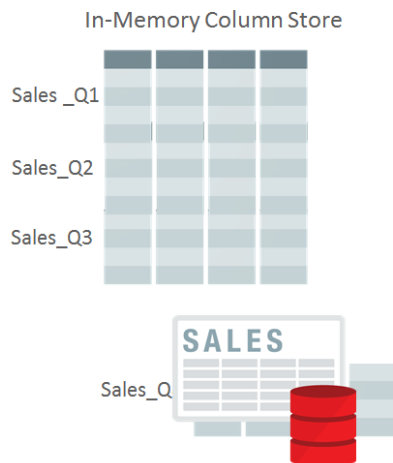
In-Memory 18c Key Features



- Automatic In-Memory Management
- In-Memory For External Tables
- In-Memory for Extreme Capacity NVRAM Memory
- Many performance improvements:
 - In-Memory Dynamic Scans
 - In-Memory Optimized Arithmetic
 - Mixed workload performance improvements

Automatic In-Memory Management

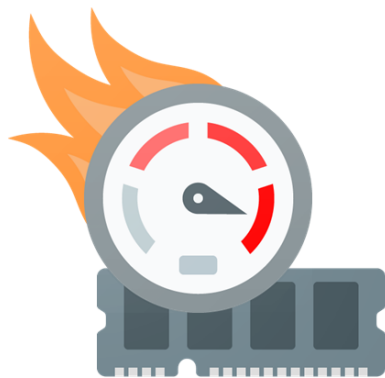
Greater Automation and Reduced Administration for Cloud Services



- In-Memory tables and partitions automatically ranked using access Heat Map (introduced in 12c)
- Hot data automatically chosen for In-Memory population
- Colder In-Memory data automatically evicted
- Useful for managed cloud services since no user intervention required

Database In-Memory Performance Improvements

Multiple Areas of Performance Improvements



- Enhanced Mixed Workload performance
 - Up to 2X Performance Improvement
- Improved aggregation performance
 - Aggregation push-down to Exadata
- Improved JSON Processing
- Faster expression evaluation
- Improved Fast-Start performance
- Improved Join Processing
- Improved Compression at Query High

Oracle Database Sharding

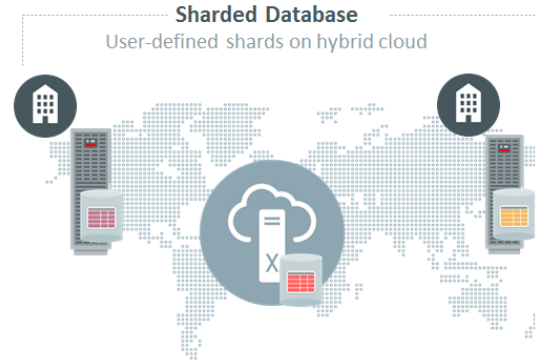
Oracle Sharding 是 Oracle Database 12.2 版本推出的新功能,也就是数据分片。是一种在数据层将数据水平分区存储到不同的数据库上的技术。Sharding 可以实现将一个分区表的不同分区存储在不同的数据库中,每个数据库位于不同的服务器,每一个数据库都称为 Shard, 这些 shard 组成一个逻辑数据库,称为 SDB (Sharded Database), 每个 SDB 中保存该表的不同数据集 (他们有相同的列)。Sharding 实际上是一种 shared-nothing 技术,每个 Shard 数据库使用独立的服务器硬件。我们可以通过增加新的 Shard 节点, 来线性扩展性能, 自动平衡数据。还可以选择根据地理位置不同, 将数据存储在不同的 Shard 节点中。

在 Oracle Database 18c 中, Oracle Sharding 增加了基于 RANGE/LIST 的用户自定义的分片方式, 允许用户显式地指定数据存储到某个特定的分片。出于性能、监管或其他原因, 某些数据需要存储在特定的分片上, 用户可以在特定的分片之间移动数据并进行完全控制。通过用户定义分片的另一个优点是, 在一个分片计划或计划外停机时, 用户确切地知道哪些数据是不可用的。还支持跨地域和混合云自动 Sharding 能力, 非常适合在云端部署。

New Sharding Method: User-defined Sharding

Explicit mapping of data to individual shards for performance & regulations

- Partition shards by RANGE or LIST
 - List or Range of sharding key values are assigned to each chunk by the user
- Supported by Data Guard and Active Data Guard
- Full control on location of data provides:
 - Regulatory compliance
 - Data remains in country of origin
 - Hybrid cloud and cloud bursting
 - Some shards on premises; other shards in the cloud
 - Efficient range queries
 - Precisely controlled data distribution

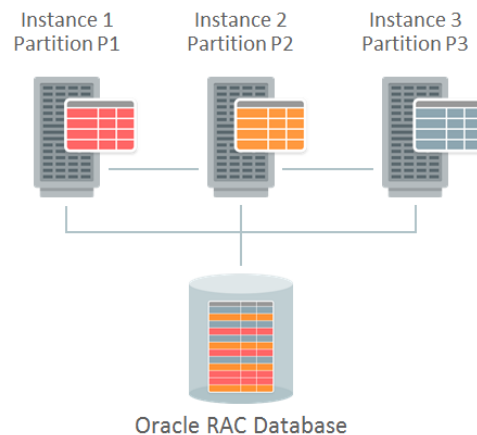


除此之外, Sharding 在 RAC 方面的功能也有增强。我们可以将 Shard 分片添加到 RAC 实例。通过分片密钥可以将请求路由到逻辑上保存分片的 RAC 实例, 提高缓存利用率, 减少实例间的阻塞。不指定分片密钥的请求仍然可以透明地工作。

Sharded RAC

Higher performance for shard-aware RAC applications

- Affinitizes shards to RAC instances
 - Requests that specify sharding key will be routed to the RAC instance that logically holds the shard
 - Affinity gives better cache utilization and reduced block pings across instances
- Requests that don't specify sharding key still work transparently
- Gives Sharded Database performance with minimal application changes
 - Just add sharding key to the most performance intensive operations



High Availability

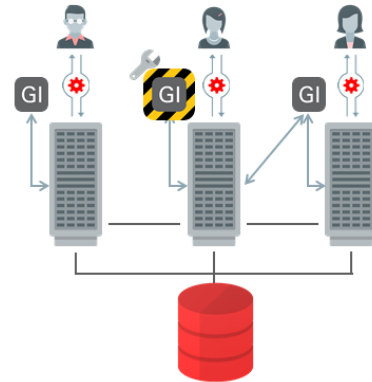
Oracle 的高可用架构包括 Oracle RealApplication Cluster, Oracle Data Guard, Oracle ZDLRA(零数据丢失恢复一体机), Oracle Golden Gate, Oracle Sharding 等。在 Oracle Database 18c 中, 借助 Oracle Flex Cluster 和 Oracle Flex ASM 的特性, 我们给 Grid Infrastructure 打补丁时, 通过滚动的方式, 当前节点上的数据库仍可以正常对外服务, 不再需要停机。还有其他不错的功能, 比如:

- **Read-Only Oracle Home:** 通过这个特性, 可以把日志等文件写入 ORACLE_BASE 中, 更方便管理和维护
- **ASM Database Cloning:** 她的优势在于, 通过 ASM Database Cloning 复制的是完整的数据库, 而不是文件或物理存储块, 可以替代基于存储级别的克隆或复制同步, 更稳定可靠。这些都是基于 ASM Flex Disk Groups 实现的, 也建议大家多采用
- **Zero-Downtime Database Upgrade:** 零停机数据库升级自动化了数据库升级所需的所有步骤。在升级过程中, 它可以最小化甚至消除应用程序停机时间。它还提供了一个回滚路径, 可以在必要时回滚升级。零停机数据库升级降低了数据库升级的工作量、风险和对应用程序的影响

Zero Impact Grid Infrastructure Patching

Never take down a database instance to patch Grid Infrastructure

- Zero Impact Patching enables patching of the Oracle Grid Infrastructure without interrupting database operations.
- Patches are applied out-of-place and in a rolling fashion with one node being patched at a time while the **database instance(s) on that node remain up and running.**
- Zero Impact Patching supports Oracle Real Application Clusters (RAC) databases on clusters with two or more nodes.

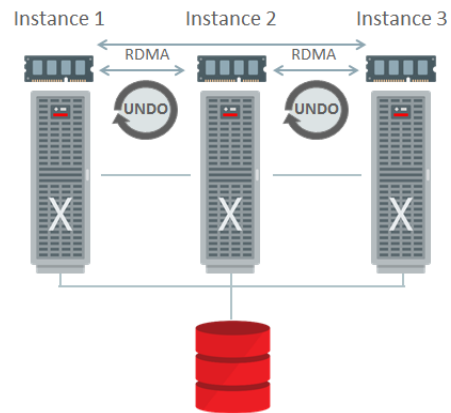


除此之外, Oracle Oracle Real Application Cluster 在 Oracle Exadata 上, 还可以利用 Undo Block RDMA-read 和 Commit Cache 来 LMS 的负载, 通过多种有效的机制来提升系统的高性能和高可用性。

RAC optimizations for Exadata

New RDMA algorithm decreases messages, latency, and CPU time

- Undo Block RDMA-read
 - In some workloads, more than **half of remote reads are for Undo Blocks** to satisfy read consistency
 - **Undo Block RDMA-read** uses RDMA to directly and rapidly access UNDO blocks in remote instances
- Commit Cache
 - The Commit Cache maintains an in-memory table on each instance which records the commit time of transactions
 - Remote LMS directly reads the commit cache and sends back commit times for requested transactions.
 - Replaces having to send entire 8K transaction table block



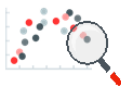
Data Warehousing and Big Data

在数据仓库和大数据方面，Oracle 有最先进的分析引擎，包括分区技术、压缩技术、分析视图，分析 SQL，数据挖掘等。通过使用 Oracle Bigdata SQL 还可以轻松分析 Hadoop 中保存的数据，打通结构化数据和非结构化数据的楚河汉界。当然还有 Oracle 的大数据一体机（Big Data Appliance）。除了这些 Oracle 云端的大数据分析云（OAC）和大数据服务（BDCS）都有更卓越的体验。

在 Oracle Database 18c 中，In-Memory 功能开始支持外部表，增加了许多机器学习（Machine Learning）的算法，在线合并分区，多态表（Polymorphic Tables）还支持多种格式的数据（JSON, XML 等），近似查询在响应时间和精准度方面都有增强。

Data Warehousing and Big Data

Before 18c



- The most advanced analytics engine available today
 - Partitioning, Compression, SQL, Analytical Views, Analytical SQL, Data Mining
- Easily analyze data held in Hadoop with Big Data SQL
- Big Data Appliance

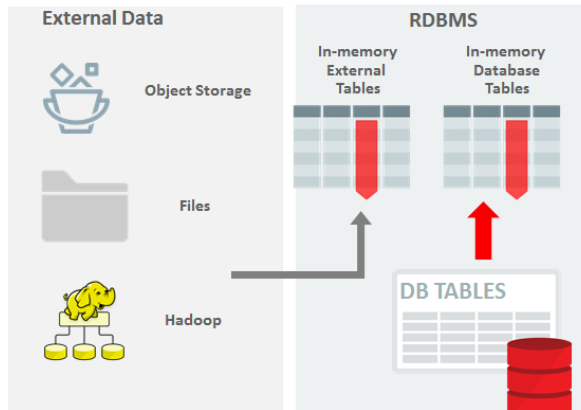
Oracle Database 18c



- In-Memory for external tables
- Automatic propagation of nologged data to standby
- More Machine Learning algorithms
- Polymorphic Table Functions
- Alter Table Merge Partition Online
- Approximate Query Improvements

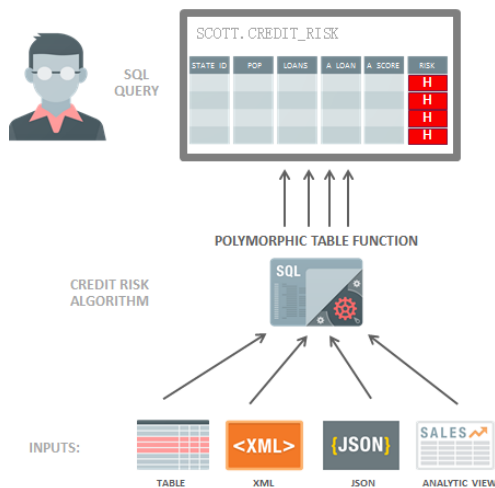
In-Memory For External Tables

Fast Analytics on External Data



- External Tables allow transparent access to data outside the DB
- In-Memory For External Tables builds in-memory column cache of data outside the DB for ultra-fast analytics on external data
- **All In-Memory Optimizations** apply
 - Vector processing, JSON expressions extend transparently to external data
- Up to **100X** faster

Polymorphic Tables: Self-Describing, Fully Dynamic SQL



- Part of ANSI 2016
- Encapsulate **sophisticated algorithms**
 - Hides implementation of algorithms
 - Leverage powerful, dynamic capabilities of SQL
 - Pass in any table-columns for processing
 - Returns SQL rowset (table, JSON, XML doc etc)
 - E.g. return credit score and associated risk level

```
SELECT
  state_id, . . . , AVG(credit_score), risk
FROM CREDIT_RISK(
  tab => scott.customers,
  cols => columns(dob, zip,loan_default),
  outs => columns(credit_score, risk_level))
WHERE risk_level = 'High'
GROUP BY state_id;
```

Top-N approximate aggregation

Interactive response times against terabytes of data

- Approximate results for common top-N queries
 - How many approximate page views did the top five blog posts get last week?
 - What were the top 50 customers in each region and their approximate spending?
- Order of magnitude faster processing with high accuracy (error rate < 0.5%)
- New approximate functions APPROX_COUNT(), APPROX_SUM(), APPROX_RANK()

Top 5 blogs with approximate hits

```
SELECT blog_post, APPROX_COUNT(*)
FROM weblog
GROUP BY blog_post
FETCH FIRST 5 ROWS ONLY;
```

Top 50 customers per region with approximate spending

```
SELECT region, customer_name,
  APPROX_RANK(PARTITION BY region
    ORDER BY APPROX_SUM(sales) DESC) appr_rank,
  APPROX_SUM(sales) appr_sales
FROM sales_transactions
GROUP BY region, customer_name
HAVING APPROX_RANK(..) <=50;
```

Security

几十年来，Oracle 一直是数据安全领域无可争议的领导者，也一直在开发创新的数据安全产品，以帮助组织应对来自不同方面的威胁和攻击。在安全方面，Oracle 有多重安全防护体系，

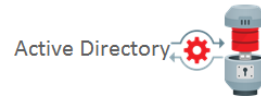
包括数据审核、访问控制、加密、秘钥库、数据掩映(Data Redaction)、数据脱敏(Data Masking)和 Audit Vault Database Firewall (数据库防火墙) 等来保证数据的安全。可以通过 Oracle 的安全评估工具来深入的了解数据库在安全方面的缺陷。在 Oracle Database 18c 中,对域(Active Directory) 用户有了更好的支持, 通过把域用户映射到数据库中的 Schema 和 Roles, 就可以直接访问数据库, 不再需要通过中间层的认证审核。我们还可以创建一个没有密码的 Schema, 这个 Schema 无法远程访问除非修改并指定身份认证方法。在某些情况下, 通过修改用户为无密码模式, 相对来说可以保护数据。适用于管理员账户和客户创建的账户。

Before 18c



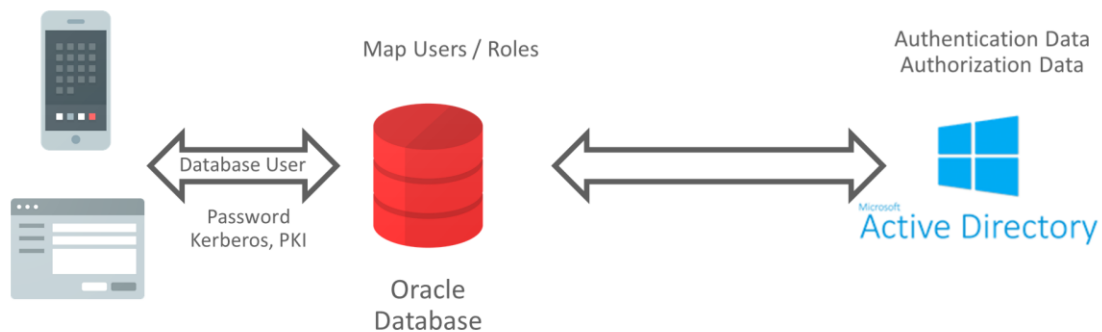
- Security In-Depth
 - Access Controls, Encryption, Redaction, Masking, Auditing, SQL Firewalls
- Key Vault
- Audit Vault Database Firewall
- Security Assessment Tool

Oracle Database 18c

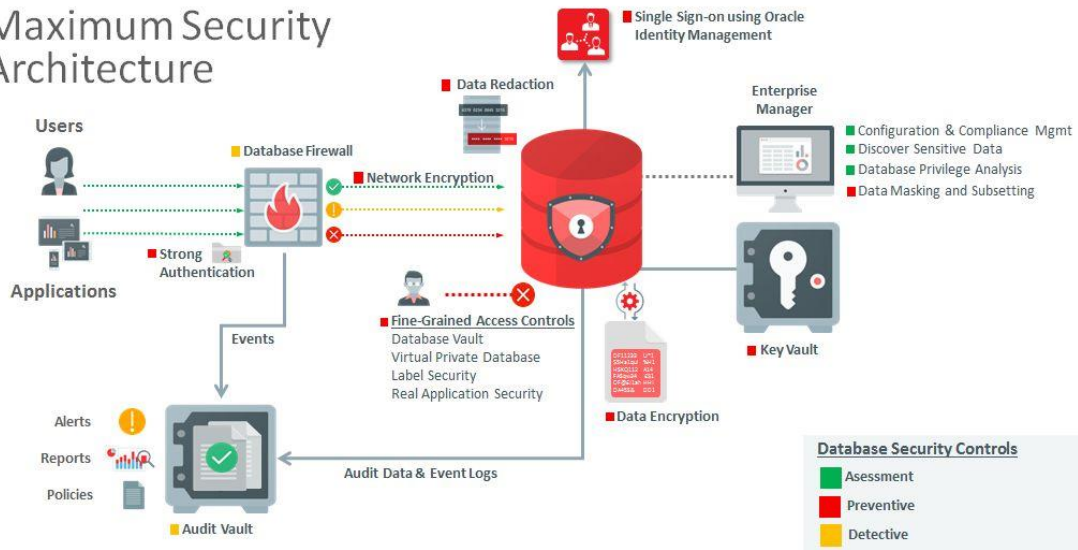


- Integration with Active Directory
 - Authorization to database through Active Directory user/group mappings to database schema users and roles
- Per PDB Key storage
- Password-less schema creation
 - No default passwords

Centrally Managed Users Directly in Active Directory



Maximum Security Architecture



通过以上介绍的新特性，我想大家已对 Oracle Database 18c 有了一定的了解，希望对您在日常工作中有一定的帮助。Oracle Autonomous Database 的目标是通过更少的人工，更低的成本以及提供更高的性能、安全性、可靠性来支持企业级关键业务的工作负载。Oracle 推出的全球第一个自治数据库（Autonomous Database）值得您拥有。

最后，别忘了在 Oracle Database 18c 白皮书后面的相关视频链接，里面有关于 Tom 大叔介绍 Oracle Database 12c 的新特性的视频和 RWP 团队关于性能调优的视频，虽然从制作到现在已有近 4 年的时间，但仍是经典之作。

<https://docs.oracle.com/en/database/oracle/oracle-database/18/videos.html>

<https://docs.oracle.com/en/database/oracle/oracle-database/18/real-world-performance-videos.html>

更多新特性请参考：

<https://docs.oracle.com/en/database/oracle/oracle-database/18/index.html>