

# AWS云上数据库迁移

Henry Huang, AWS专业服务咨询顾问

2016年4月18日

# 分享摘要：

- **AWS**：分享云上数据库迁移的最佳实践和注意事项，以及AWS数据库迁移工具（DMT）的使用方法
- **听众**：了解AWS关系型数据库服务（RDS），学习如何在AWS RDS上执行数据库迁移以及数据库迁移工具（DMT）的使用示例
- **目标**：了解数据库迁移上云，降低风险，更好地实现业务价值

# 客户为何迁移数据库上云

# 客户为何迁移数据库上云

- 成本
- 灵活性
- 托管服务
- 性能和可扩展性
- 容错能力和可用性

# 迁移注意事项

# 迁移注意事项

- 业务影响
- 现有数据库的复杂性
- 应用程序的复杂性
- 集成要求（延迟、连接性、安全性等）
- RPO/RTO
- 所需技能

# 迁移目标

# AWS上的关系型数据库



**Amazon  
RDS**



**Amazon EC2  
数据库服务器**



**Amazon  
Redshift**



# 目标1：RDS

- 六个数据库引擎
- 易于管理（托管服务）
- 可用性和持久性
- 安全性
- 多达 32 个 vCPU 以及 244GB 内存
- 价格低廉：EC2 + 10%
- 无法访问RDS实例的操作系统

# RDS数据库引擎

商用

ORACLE®



开源



PostgreSQL



MariaDB

Amazon Aurora

Amazon  
Aurora

# 托管服务的限制

- 完全托管的实例和操作系统
  - 无法访问数据库实例的操作系统
  - 有限的数据库配置管理能力
  - 提供的功能不能依赖于实例操作系统的配置
- 完全托管存储
  - 最大存储限制
    - Microsoft SQL Server - 4 TB
    - MySQL , MariaDB , PostgreSQL , Oracle - 6 TB
    - Aurora - 64TB
  - 数据库的扩展

# 目标2：Amazon EC2 数据库服务器



PostgreSQL



MariaDB

ORACLE®



SYBASE®  
An SAP Company



# 目标1和目标2的比较（以SQL Server为例）




## Amazon RDS for SQL Server

|                      |
|----------------------|
| App optimization     |
| Scaling              |
| High availability    |
| Database backups     |
| DB software patches  |
| DB software installs |
| OS patches           |
| OS installation      |
| Server maintenance   |
| Rack and stack       |
| Power, HVAC, net     |

## SQL Server on Amazon EC2

|                      |
|----------------------|
| App optimization     |
| Scaling              |
| High availability    |
| Database backups     |
| DB software patches  |
| DB software installs |
| OS patches           |
| OS installation      |
| Server maintenance   |
| Rack and stack       |
| Power, HVAC, net     |

### Legend

|   |                  |   |             |   |                  |
|---|------------------|---|-------------|---|------------------|
|  | Customer managed |  | AWS managed |  | Multiple options |
|---|------------------|---|-------------|---|------------------|

# 二者选择的思路

- 首先考虑 RDS
- 要求超出了 RDS 限制，则考虑 EC2 数据库服务器
- 需要 RDS 无法支持的功能，则考虑 EC2 数据库服务器
- 为了完全控制操作系统及软件的访问和管理，则考虑 EC2 数据库服务器
- EC2 数据库服务器需要自行管理备份、复制和集群的问题

# 目标3：RDS Oracle

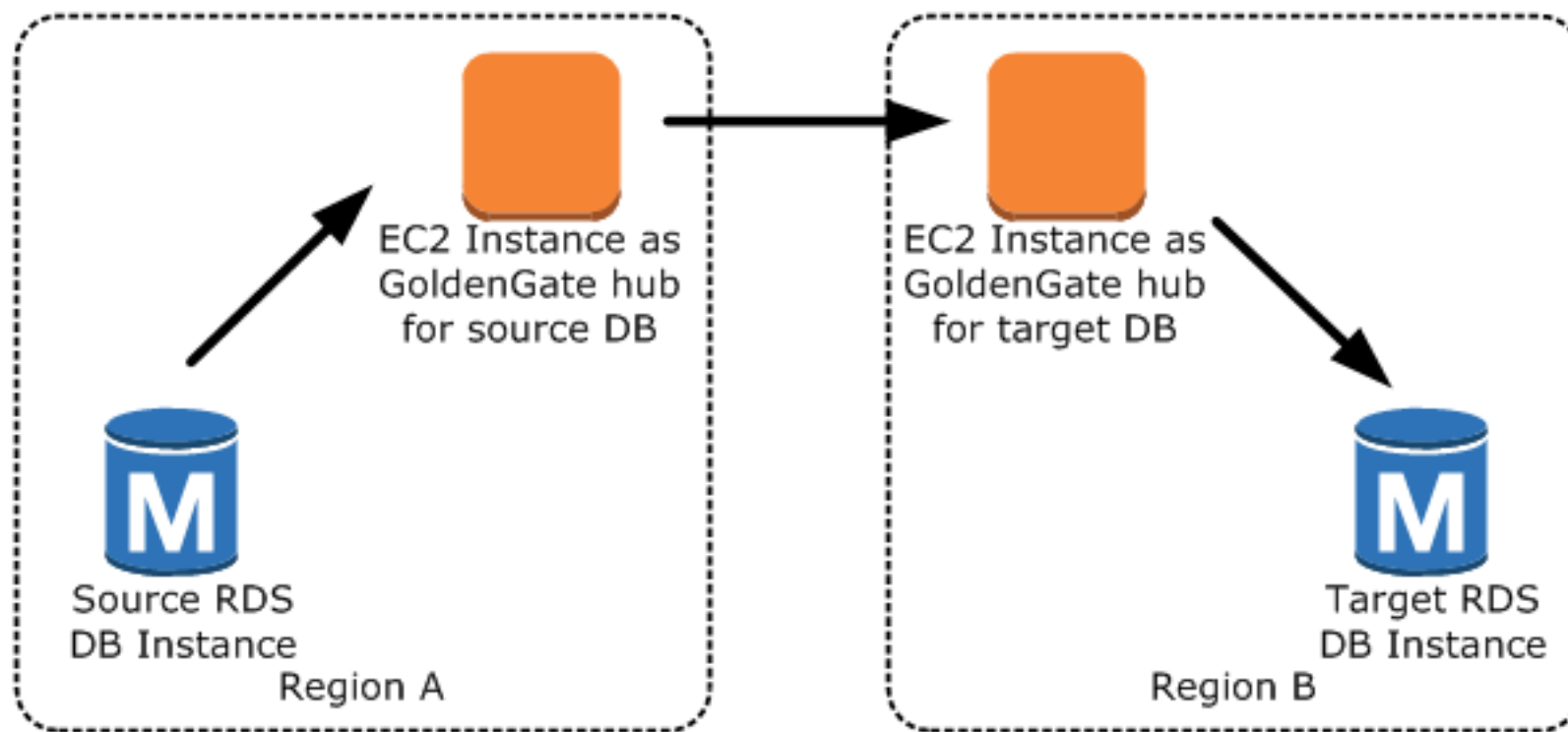
- Oracle SE，多达 16 个 vCPU
- Oracle SE One 和 Oracle SE Two，多达 32 个 vCPU
- Oracle EE
- 版本 11g 和 12c，(11.2.0.4 – 12.1.0.2)
- 主备多可用区
- 单个实例支持自动主机更换（硬件故障）

# 目标3：RDS Oracle支持的功能

- Enterprise Manager Database Control/Express
- XML DB
- 透明磁盘加密 (TDE)
- Oracle GoldenGate 11.2.1：运行 11.2.0.3/4 的 RDS Oracle BYOL 版本，还可与 TDE 配合使用
- SSL 和 Oracle 本地网络加密 (NNE)
- 不支持：RAC、Data Guard、Vault、Spatial、Streams 或 Java



# GoldenGate 与 RDS 结合使用



# 目标3：RDS Oracle 许可

- 附带许可或 BYOL

例如，对于已授权云环境中的 *Database Enterprise Edition* 许可：由 8 个虚拟内核 (英特尔多核芯片) 组成的单个实例上授予 *Oracle Database Enterprise Edition* 许可，则需要  $8 * 0.5 = 4$  个处理器许可证 (每个虚拟内核被视为一个物理内核)。

从 *Oracle* 产品许可的角度看，这意味着每个虚拟处理器相当于一个物理内核，且适用标准的 *Oracle* 处理器规格定义。

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

- 虚拟内核不是vCPU。 注意，1 个 vCPU == 1 个超线程 == 0.5 个虚拟内核。

来源：<https://aws.amazon.com/ec2/virtualcores/>

# 目标4：RDS SQL Server

- SQL Server 2008 R2 引擎
- 从 2008 R2 直接升级上来的 SQL Server 2012 引擎
- 两种引擎均适用 Express、Web、Standard 和 Enterprise 版

# 目标4：RDS SQL Server 支持的功能

- Windows 身份验证
- 适用于传输中加密的 SSL
- 适用于静态加密的透明数据加密 (TDE)
- 2008 和 2012 版中的 SSMS、SSMA、安全 CLR 和全文搜索支持
- 2012 EE 版中的 Columnstore 索引
- SQL Server 镜像多可用区部署，仅限 Standard 和 EE 版
- 不支持原生 Windows/SQL 集群或 WSFC

# 目标5：RDS MySQL，MariaDB 和 PostgreSQL

- MySQL v5.1、5.5-7，MySQL GA 版推出后 3-5 个月
- PostgreSQL v9.3.5 以上版本
- MariaDB v10.0.17 以上版本
- 针对每个实例大小预配置参数
- 多可用区只读副本 (MySQL)

# 目标6：Redshift

- 快速、可扩展、PB 级数据仓库
- 完全托管
- < 1000 美元/PB/年
- 静态加密和传输中加密

# 迁移场景

# 同构迁移：Oracle（工具）

- Oracle SQL Developer < 200MB
- Materialized Views < 500MB；仅限 EE
- Oracle SQL\*Loader < 10GB
- Oracle Data Pump < 5TB
- Oracle Recovery Manager (RMAN) 任意大小
- Oracle GoldenGate 任意大小

## AWS 白皮书：

<https://d0.awsstatic.com/whitepapers/strategies-for-migrating-oracle-database-to-aws.pdf>



# 同构迁移：Oracle（方法）

- 一步式迁移
  - 关闭、提取、传输、导入，然后进行验证
  - 停机影响较大，但比较简单
- 两步式迁移
  - 在非高峰时间点提取数据。数据库保持联机状态
  - 在新数据库中传输、导入和验证数据
  - 源数据库脱机，并迁移自初始迁移后开始出现的增量数据
  - 将停机时间减至最短，且相对简单
- 零宕机
  - 通过 Oracle GoldenGate、Dbvisit Replicate 或 Attunity Replicate 复制数据  
零停机，但很复杂

# 同构迁移：SQL Server

- 生成和发布向导 < 1GB
- 导入和导出向导 > 1GB
- 批量复制功能 > 1GB (最快)

# 同构迁移：MySQL 和 MariaDB

- 纯数据文件 > 20GB  
将数据文件复制到 EC2，然后通过 LOAD DATA LOCAL INFILE 进行导入，速度快
- SQL 文件 < 20GB  
使用 mysqldump sql 文件，速度慢且不可重新开始  
在导入期间禁用索引、外键约束和二进制日志
- 创建只读副本以保持最新，切换时提升只读副本

# 同构迁移：PostgreSQL

- 批量：pg\_dump/pg\_restore
- 增量式：提前写入日志



# AWS Database Migration Service



在 10 分钟或更短的时间内开始您的首次迁移

迁移过程中您的应用程序可持续运行

在 Amazon EC2 或 RDS 之内或之间进行复制

将数据迁移到同一个或不同的数据库引擎

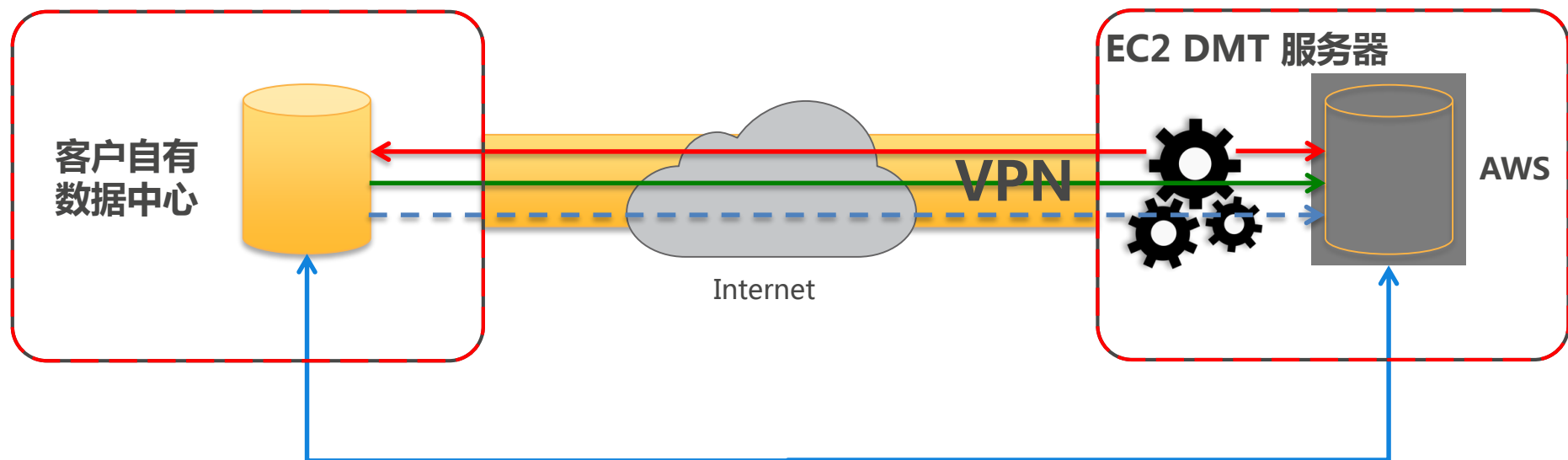
# Database Migration Tool (DMT)

*“这是一个强大的实用程序，用最少的停机时间将企业内部和EC2上的数据库服务器迁移到RDS、Amazon Redshift和Amazon Aurora”*

支持的功能：

- 事务性变更数据捕获（CDC）和应用，对源数据库的性能影响较小
- 异构迁移（例如Oracle到MySQL）和同类迁移（例如Oracle到Oracle）
- 支持的源数据库包括：
  - Oracle，SQL Server，MySQL，Aurora，ODBC，Postgres
- 支持的目标数据库包括：
  - 包括Redshift，Oracle，SQL Server，MySQL，Aurora，Postgres

# DMT的工作原理



启动 EC2 DMT服务器

连接源和目的数据库

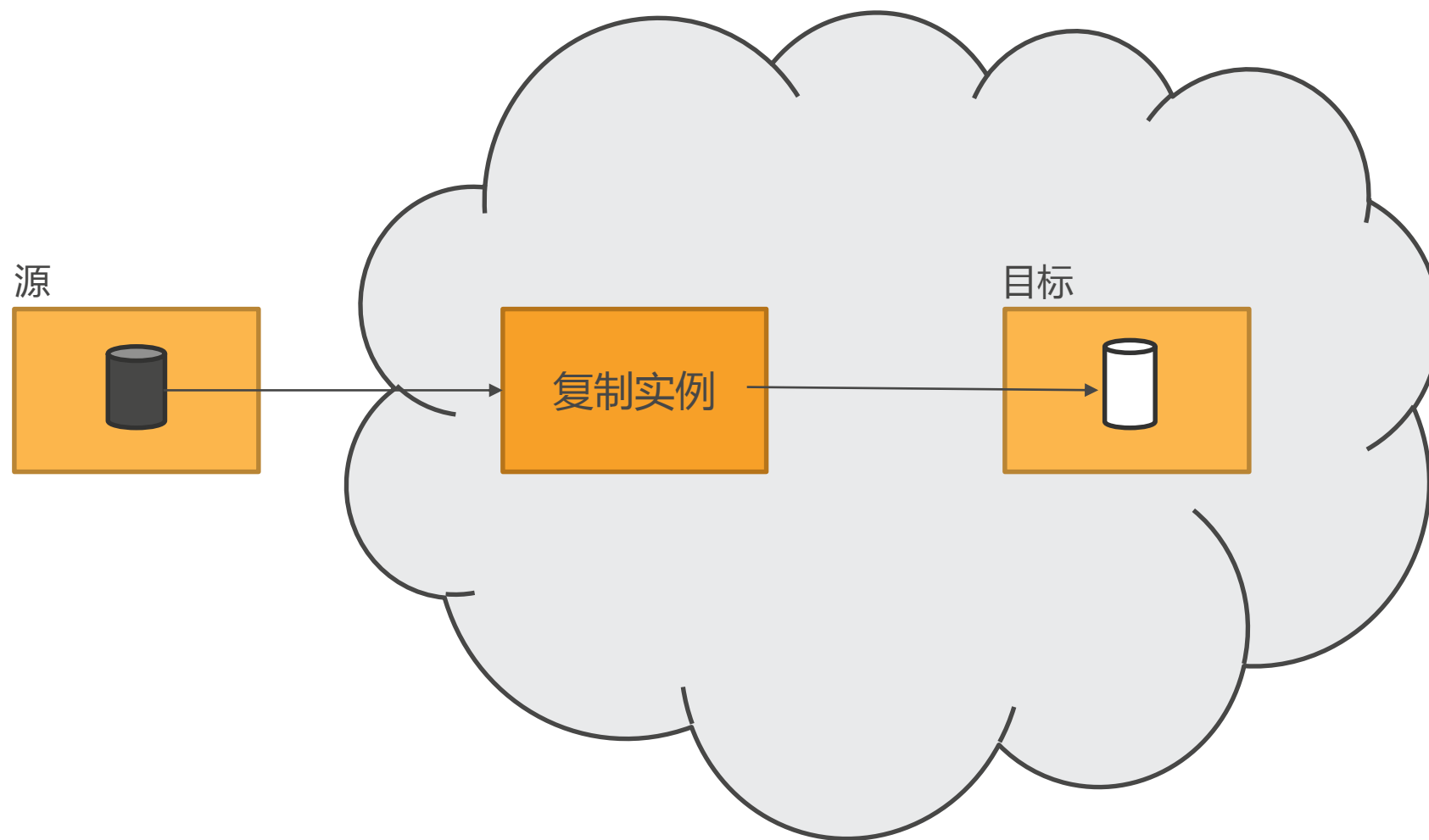
选择相应的表、Schema 或数据库

让DMT服务创建表，加载数据并保持同步

合适的时机将应用程序的访问切换到目标数据库上

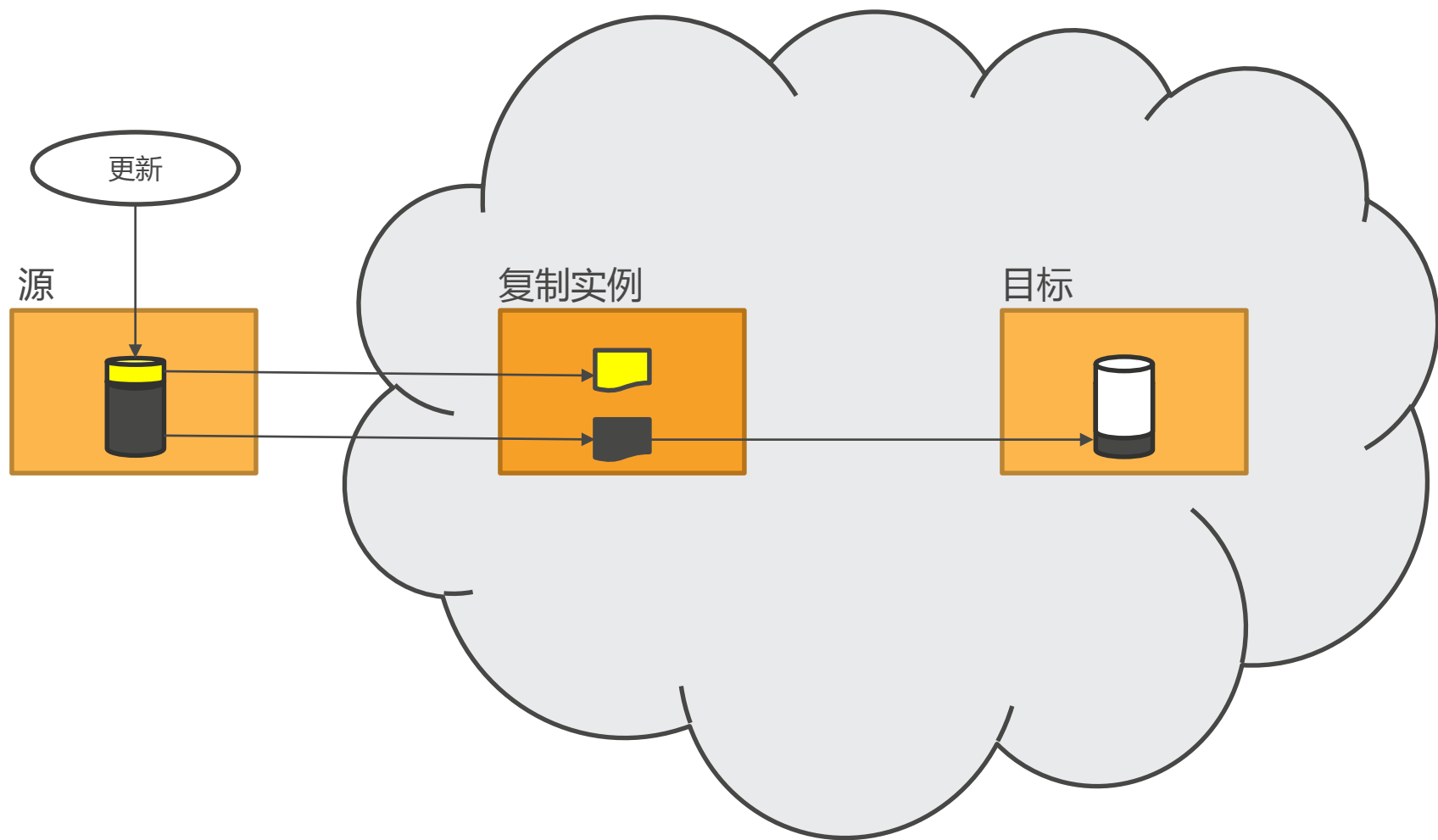


# 开始完全加载

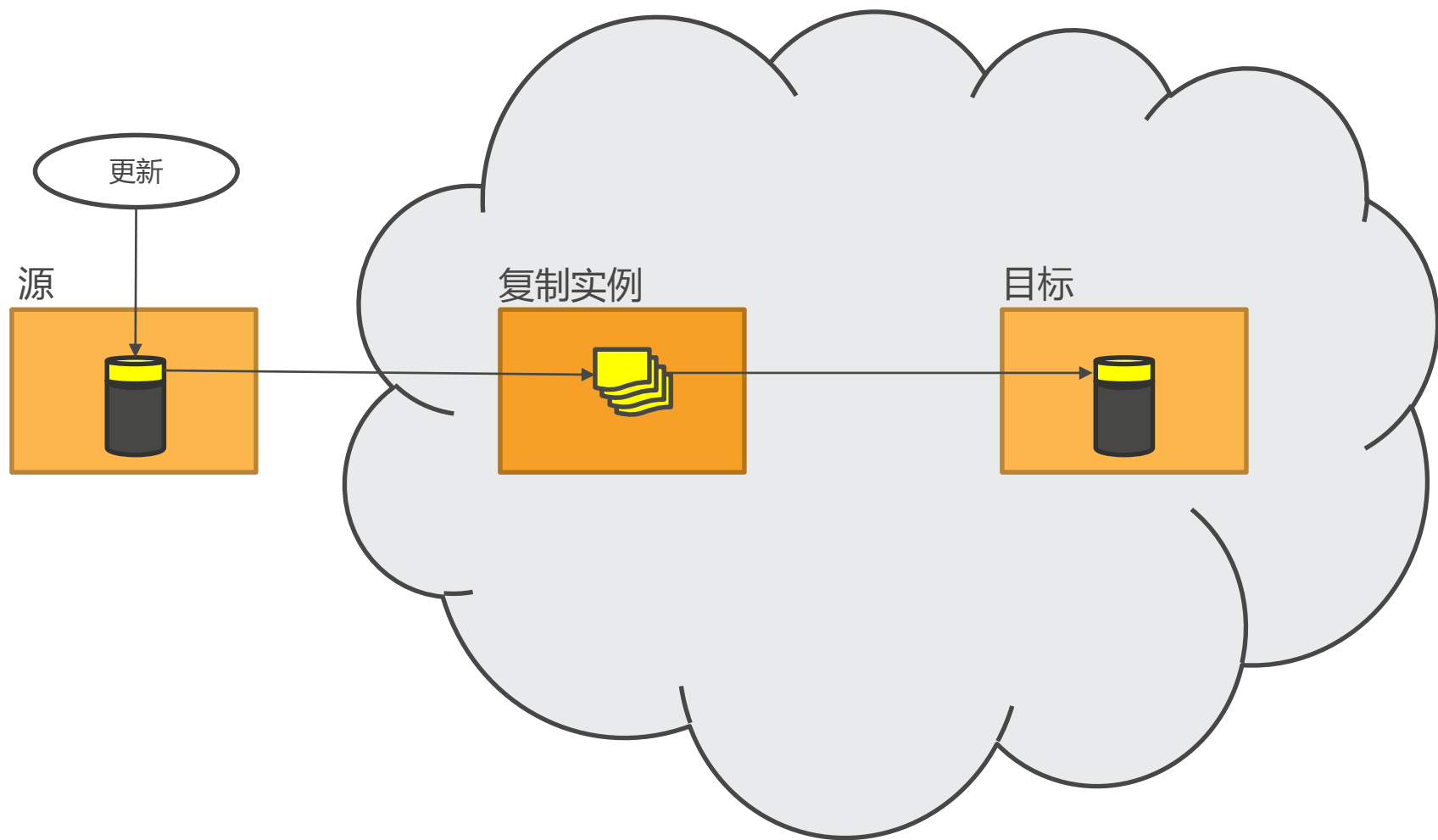




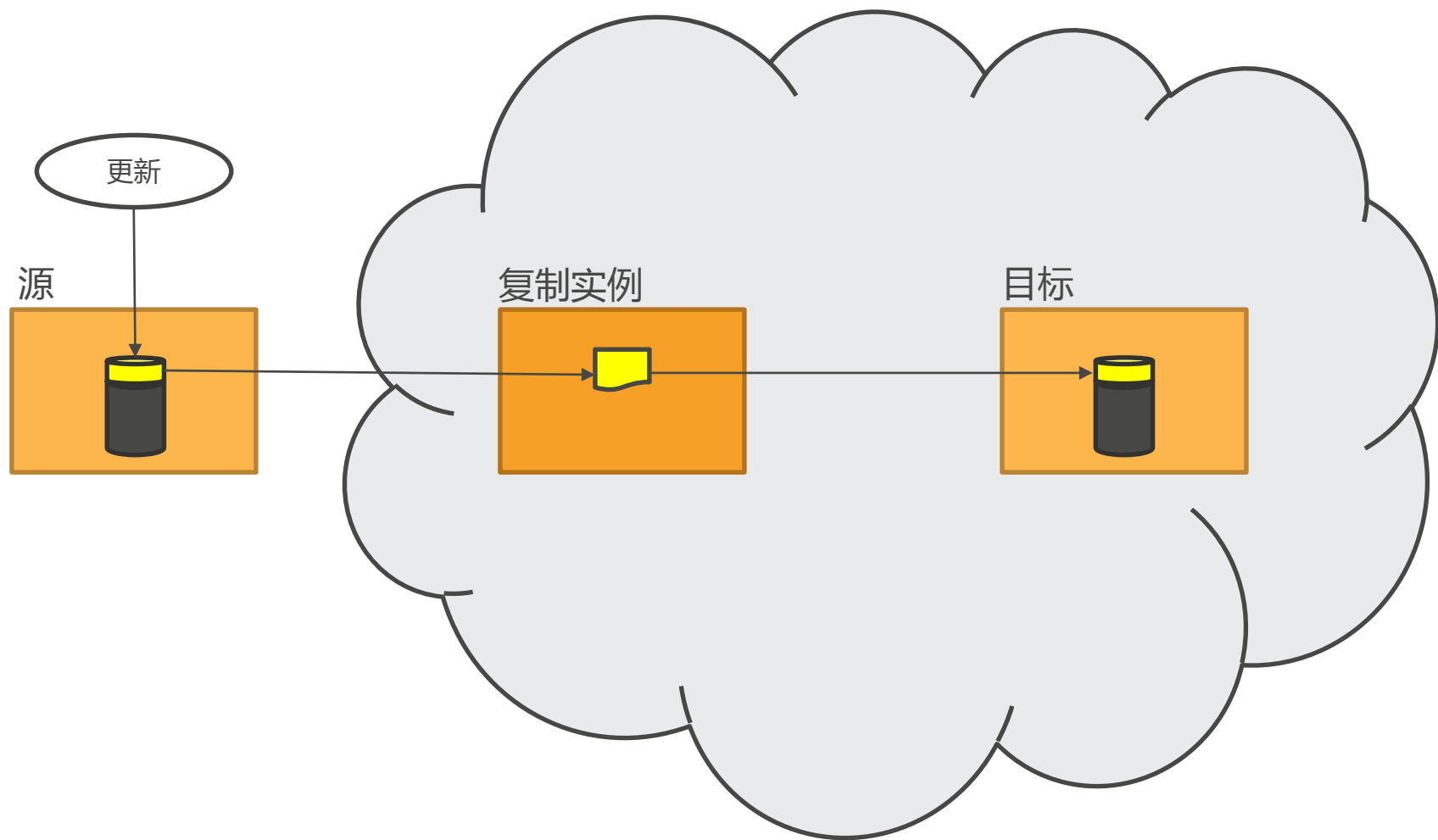
# 在加载数据的同时捕获更改



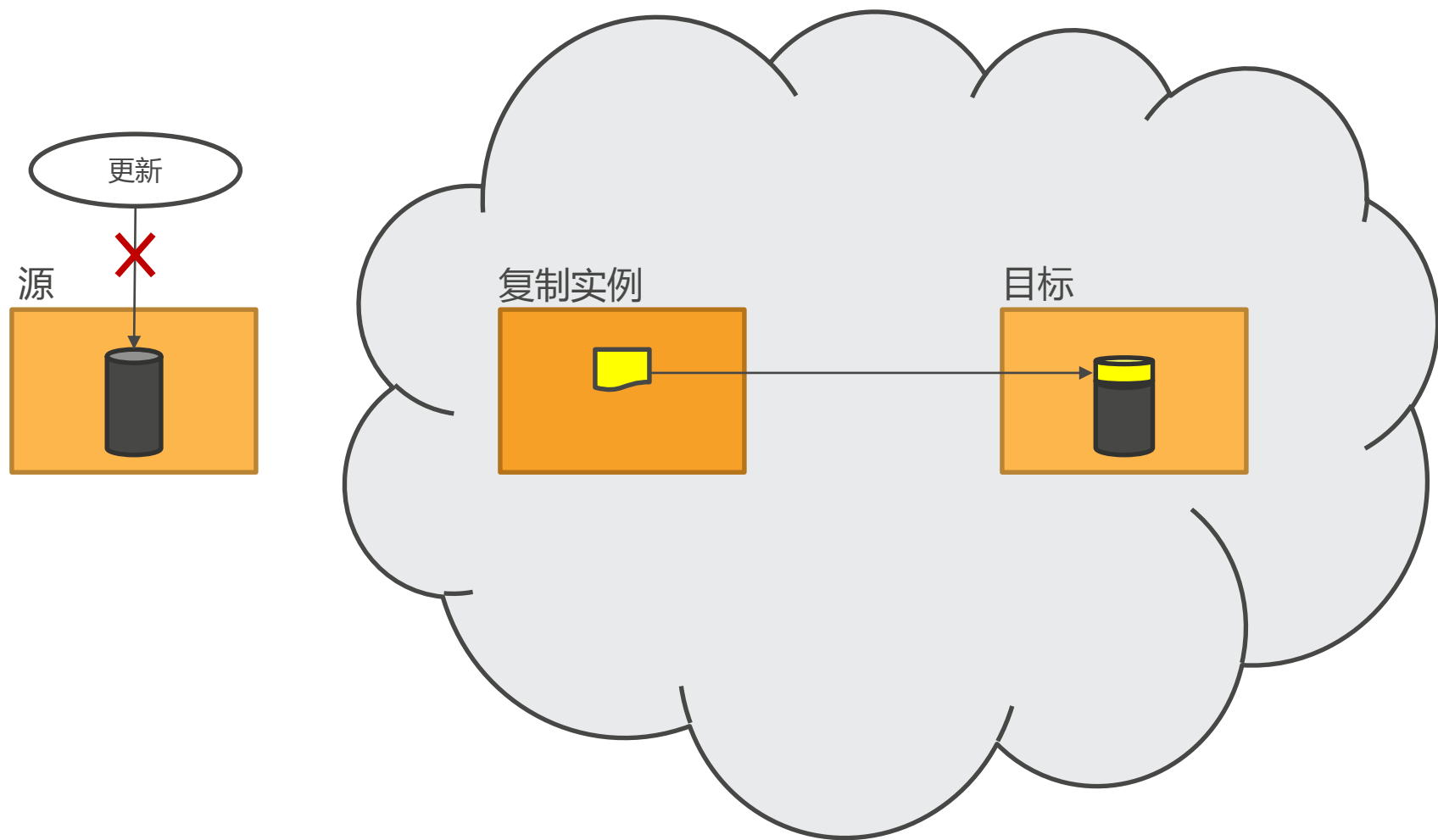
# 加载完成 – 应用捕获的更改



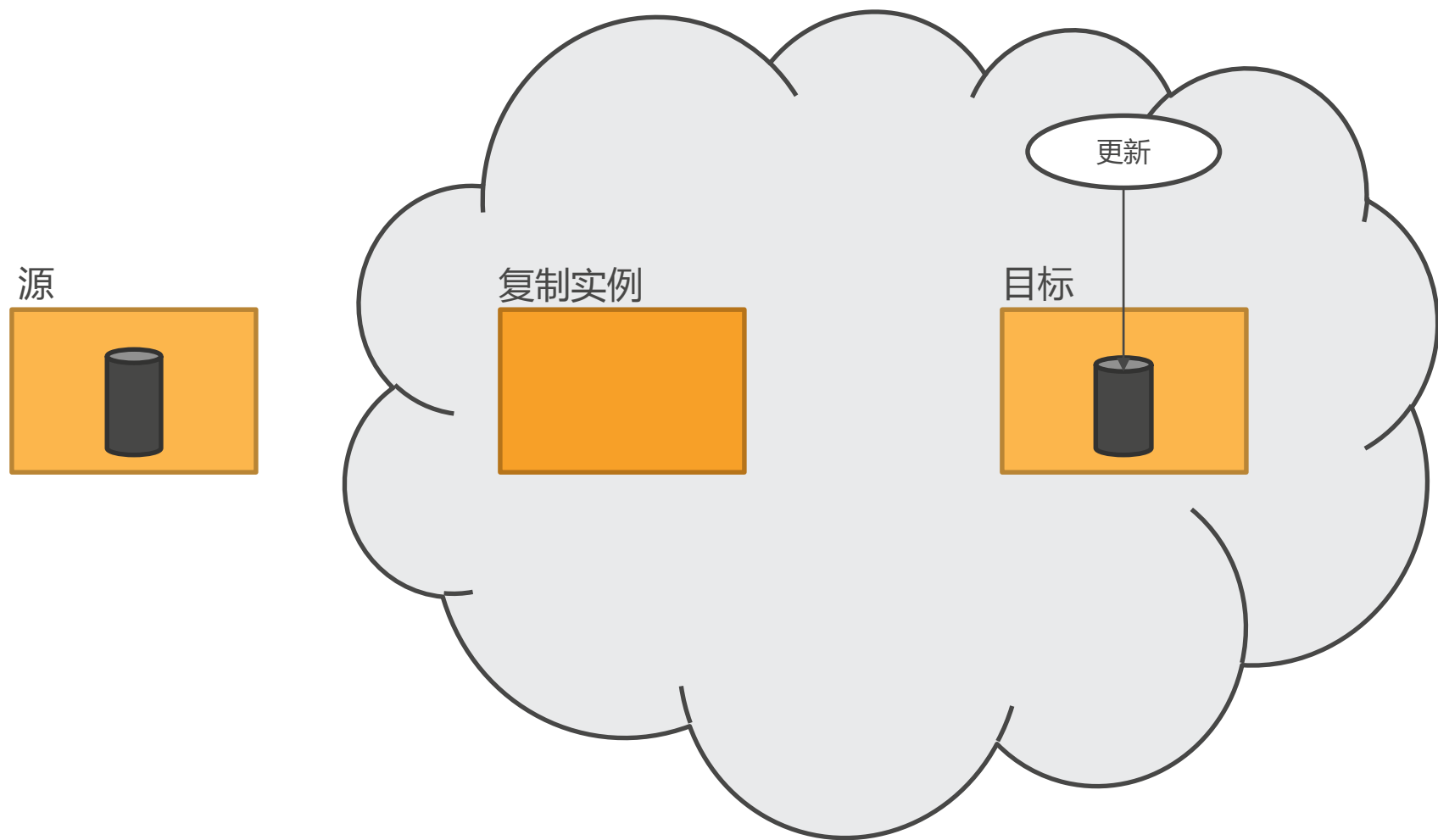
# 更改达到稳定状态



# 转换 – 关闭应用程序并应用其余更改



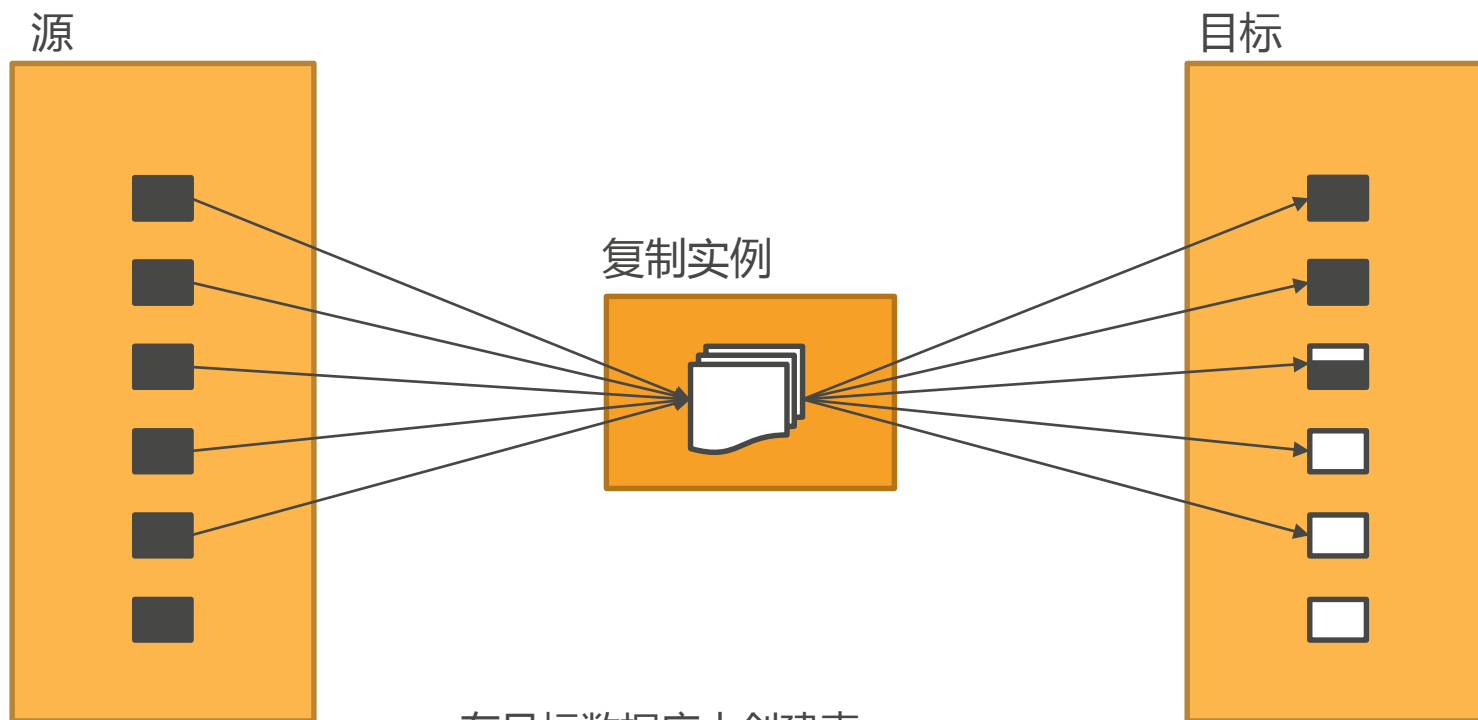
# 开始运行！



# 迁移阶段

- 应用/创建空的数据库架构，去除外键约束和二级索引
- 利用 DMT 迁移数据
- 应用其余的架构对象
- 开始运行

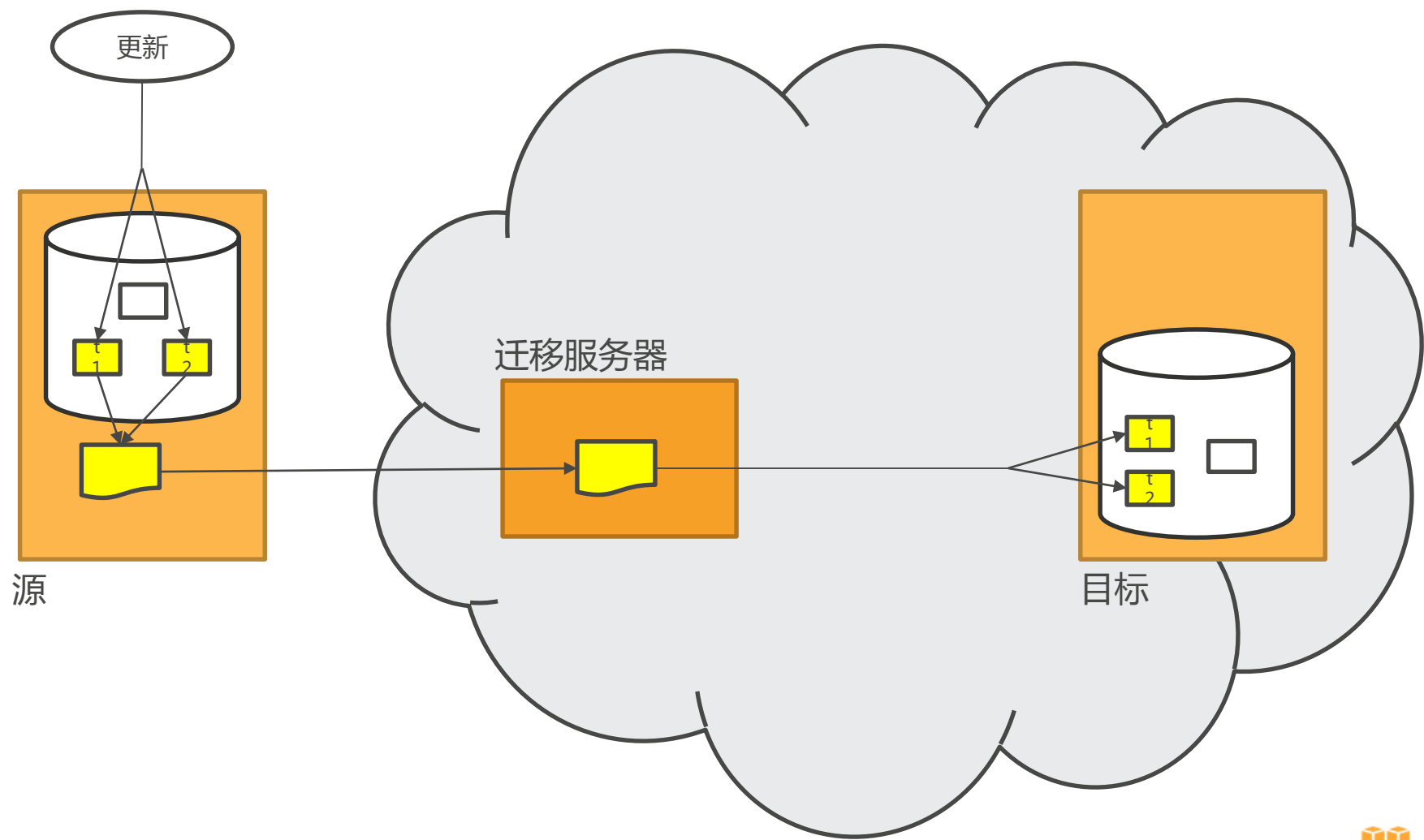
# 加载是逐表进行的



在目标数据库中创建表  
在目标数据库中设置所需的元数据  
从源数据库中填充数据  
每个进程加载一个完整的表  
可以使用多个进程  
可暂停

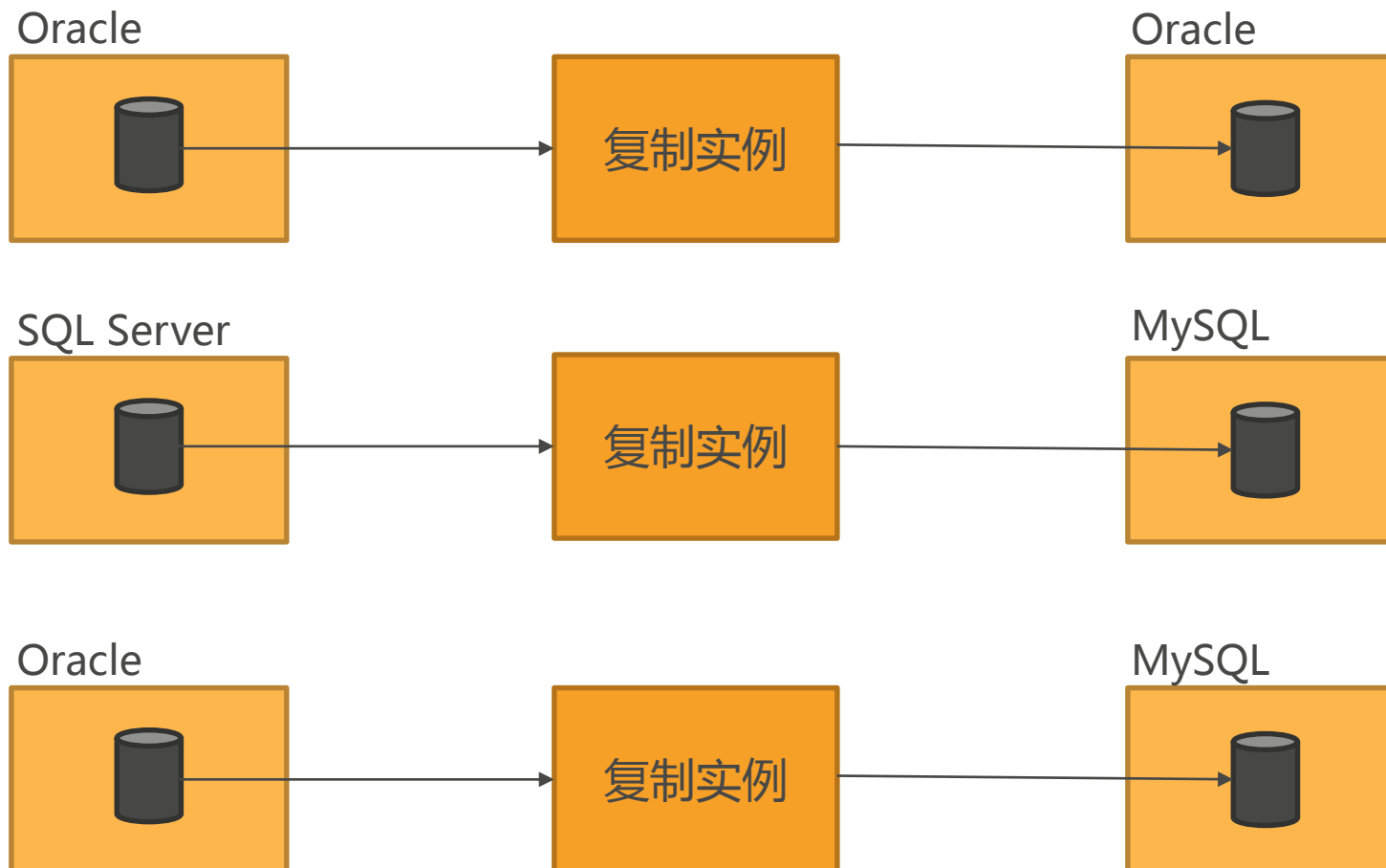
- 重新开始后，将从以前停止处继续
- 将重新加载当时正在处理的所有表

# 更改是事务性的，来自日志





# 同构或异构



# Database Migration Tool 源和目标数据库

## 源数据库

- MySQL, RDS MySQL, Aurora (1)
- Oracle, RDS Oracle
- SQL Server, RDS SQL Server (2)
- ODBC 3.0 sources
- PostgreSQL, RDS PostgreSQL



## 目标数据库

- MySQL, RDS MySQL, Aurora (1)
- Oracle, RDS Oracle
- SQL Server, RDS SQL Server
- Amazon Redshift
- PostgreSQL, RDS PostgreSQL

- ( 1 ) - Amazon Aurora源和目标使用与RDS MySQL相同的访问机制
- ( 2 ) - 批量提取模式支持RDS SQL Server源数据库；尚不支持CDC模式
- ( 3 ) - 没有带RDS或Amazon品牌前缀的数据库类型，表示企业内部部署或使用EC2安装
- ( 4 ) - 源和目标数据库不能同在企业内部

# 支持的所有源和目标数据库

| Endpoint                | Database Versions                 | Database Editions                          |
|-------------------------|-----------------------------------|--|
| Oracle Source           | 10g, 11g or 12c                   | Enterprise, Standard, Standard One         |
| Oracle Target (EC2)     | 10g, 11g or 12c                   | Enterprise, Standard, Standard One         |
| Oracle Target (RDS)     | 11g or 12c                        |  |
| SQL Server Source       | 2005, 2008, 2008R2, 2012 and 2014 | Enterprise, Standard, Workgroup, Developer |
| SQL Server Target (EC2) | 2005, 2008, 2008R2, 2012 and 2014 | Enterprise, Standard, Workgroup, Developer |
| SQL Server Target (RDS) | 2008R2, 2012                      | Enterprise, Standard                       |
| MySQL Source            | 5.5 and 5.6                       |  |
| MySQL Target            | 5.5 and 5.6                       |  |
| Amazon Aurora Target    | 支持兼容MySQL的数据源                     |  |
| Amazon Redshift Target  |                                   |  |
| ODBC 3.0 Source         |                                   |  |

# 数据类型的限制

| Endpoint          | Limitations   |
|-------------------|---|
| Oracle            | BFILE, ROWID, REF, UROWID, Nested Table, User-defined data types  |
| SQL Server Source | CURSOR, SQL_VARIANT, TABLE, GEOGRAPHY, GEOMETRY, HIERARCHYID  |
| MySQL Source      | CURVE, GEOMETRY, GEOMETRYCOLLECTION, LINE, LINEARRING, LINESTRING, MULTICURVE, MULTILINESTRING, MULTIPOINT, MULTIPOLYGON, MULTISURFACE, POINT, POLYGON, SURFACE |

# 何时使用 DMT ?

- 迁移方向：
  - 本地迁移到云或反向
  - 从 EC2 迁移到 RDS或反向
- 异构迁移
- 同构迁移
- 重构数据库，帮助系统的拆分或整合

# Database Migration Tool

## 可做

- 基础的架构迁移 (表、主键)
- 批量数据加载
- 变更数据捕获与应用 (CDC)

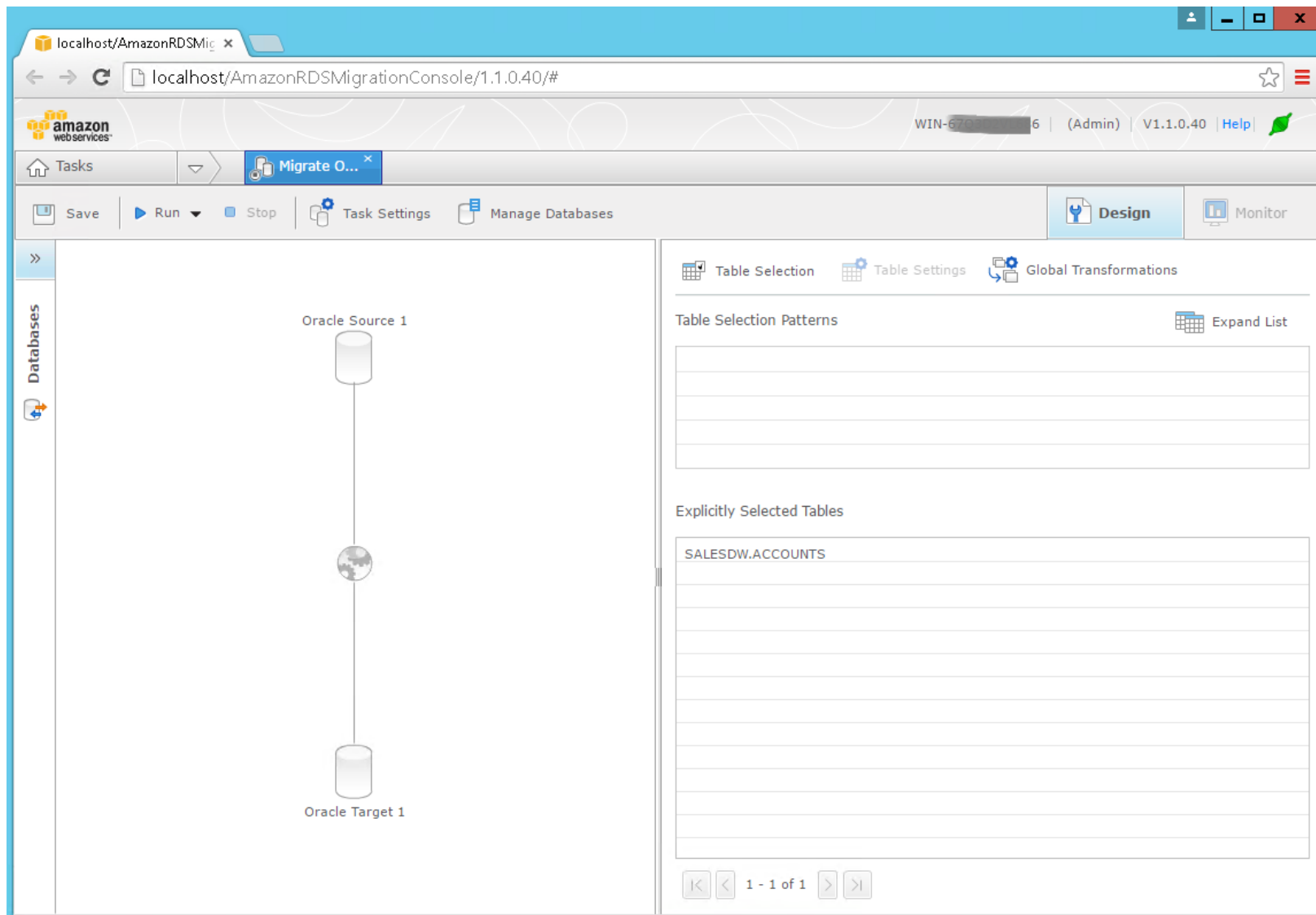
## 不可做

- 复杂的架构迁移 (存储过程、用户帐户等)
- 代码转换

# 注意事项

- DMT服务器与源数据库、目标数据库之间需要具备网络连接，该服务才能正常工作
- 该服务是一种数据迁移服务，并非架构迁移服务。因此，它仅在目标数据库上创建迁移数据所需的内容。稍后我们将讨论 Schema Conversion Tool
- 虽然从技术角度而言，该服务是一种复制服务，但是目前不建议客户使用该服务执行持续复制
- DMT服务器实例类型，建议至少使用一个XLarge（M3或R3）实例
  - 记住：从XLarge实例开始设置数据源和迁移任务，并在实际数据迁移之前将其扩展为更大的实例类型。

# 直观的用户体验

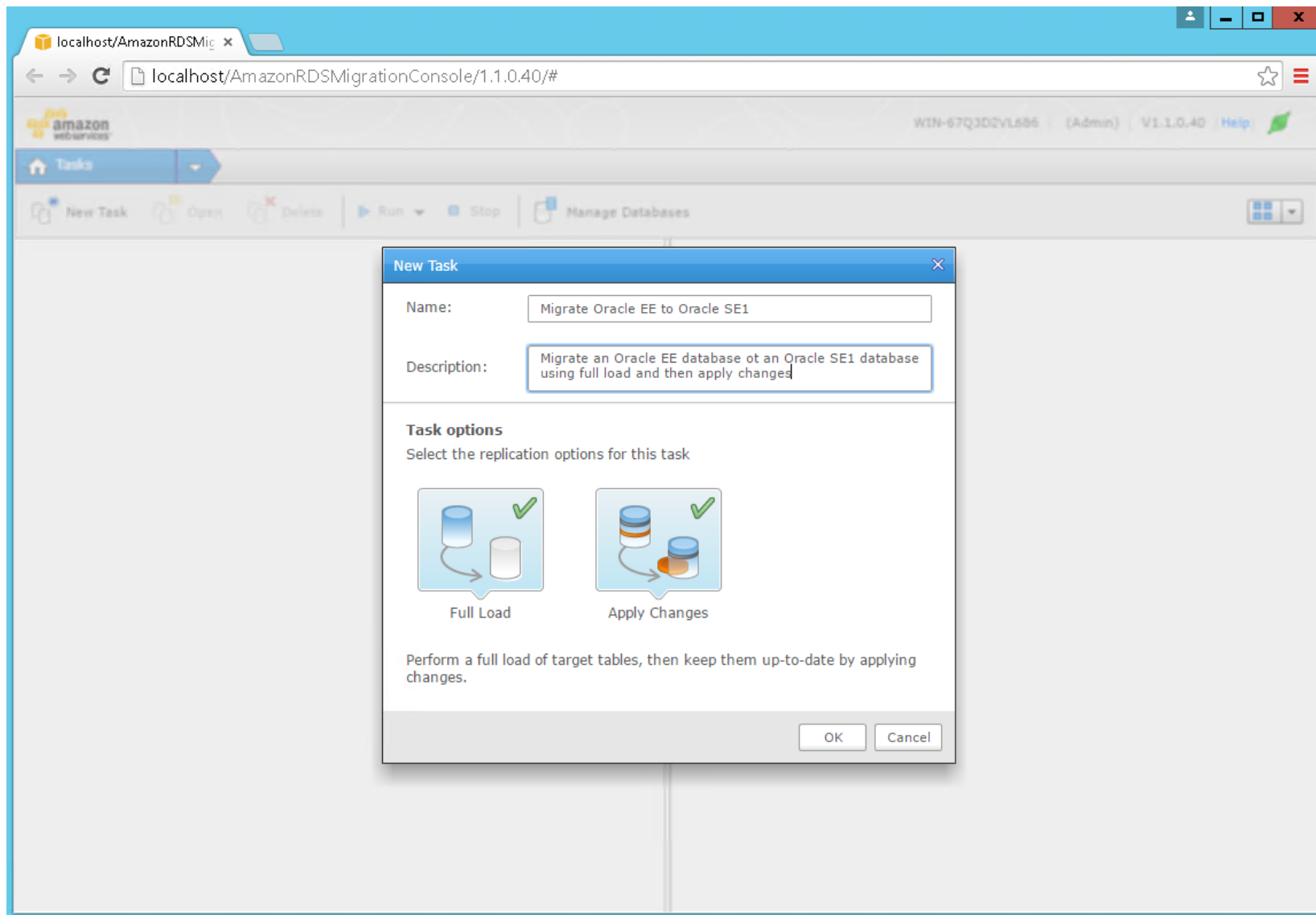


- 直观的用户体验简化了迁移到AWS所需的步骤
- 定义来源和目标
- 创建迁移任务

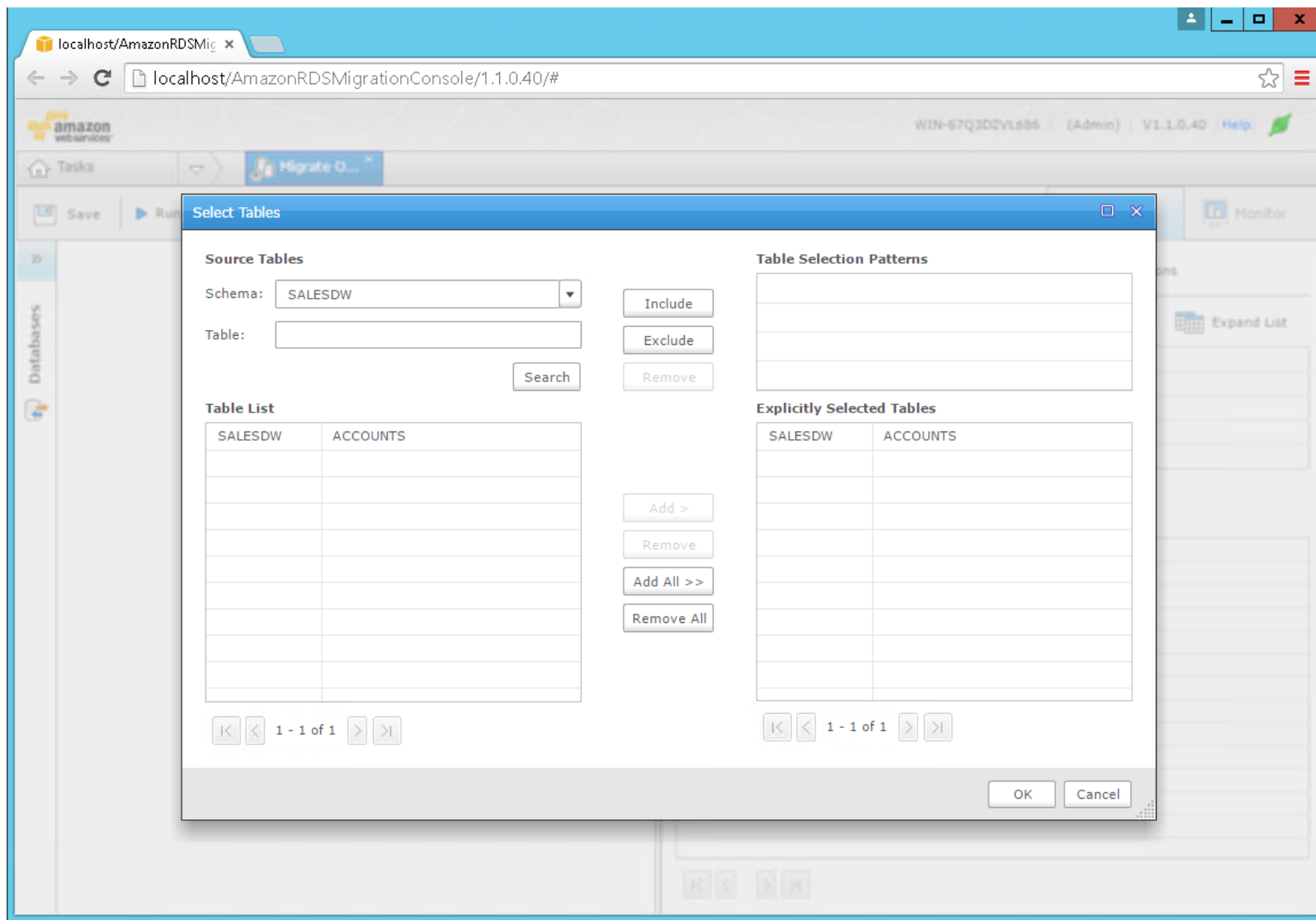


# 支持完全加载和应用捕获更新

选择Full Load，然后选择Apply Changes

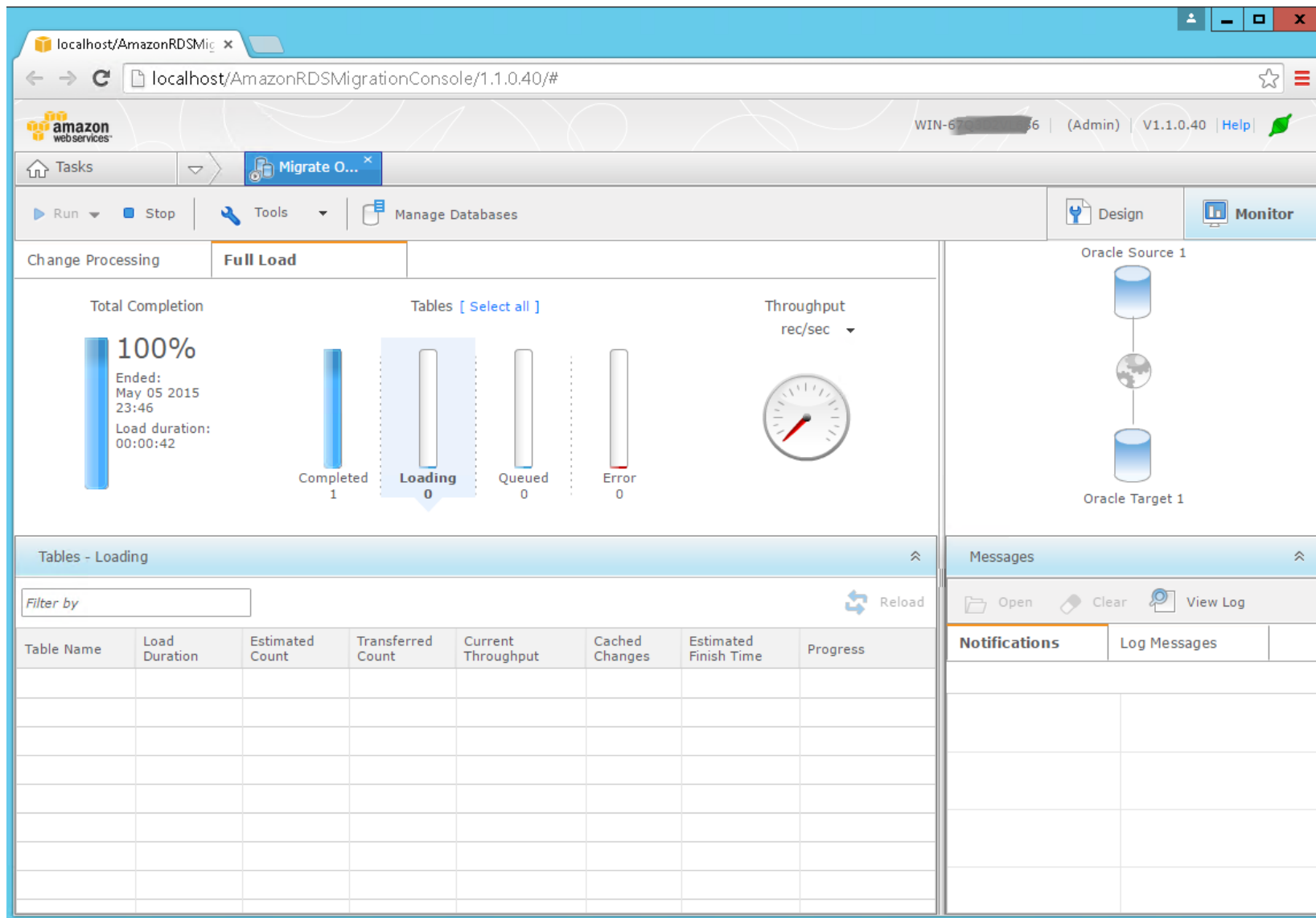


# 选择表和指定的行数据进行迁移



- 选择要迁移的表
- 为行创建过滤条件
- 定义轻量级变换

# 使用仪表板和告警监控



- 监控吞吐量、已完成和排队中表的数量
- 查看成功和失败消息

# 定价和条款

# ¥ 0

## for software license

### 定价

- ◆ 免费的软件许可
- ◆ EC2，EBS以及与RDS迁移工具结合使用的其他AWS服务

### 工具使用范围

- ◆ 支持源、目标数据库或两者都在AWS中
- ◆ 不允许数据库在企业内部迁移
- ◆ 不允许从AMI复制该工具

[Additional Terms and Conditions](#)

# 相关资料

- **客户常见问题**

- 回答初始客户问题，解释工具的价值，可以从[这里](#)下载，不需要NDA

- **附加条款和条件**

- 这些条款规定了RDS迁移工具的使用范围，可以从[这里](#)下载

- **RDS迁移工具的AMI**

- 请联系您的Account Manager 获得许可

- **快速入门指南**

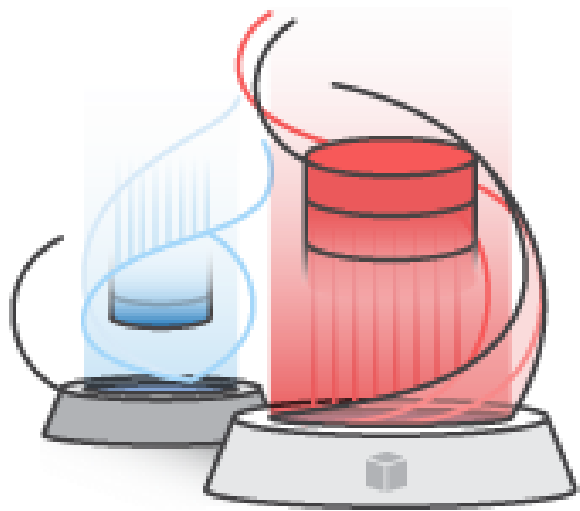
- 推荐在启动迁移实例后的第一步，可以从[这里](#)下载

- **用户指南**

- 技术文档可从[这里](#)下载

# Database Migration Tool : 演示

- 数据库示例
  - MySQL Sample Database:  
[https://github.com/datacharmer/test\\_db.git](https://github.com/datacharmer/test_db.git)
- 同构数据库迁移  
// 其中本地MySQL由不同VPC的EC2 MySQL代替
  - 本地 MySQL 迁移到 RDS MySQL
  - 本地 MySQL 迁移到 EC2 MySQL
  - RDS MySQL 迁移到 EC2 MySQL
- 注意 binlog 的设置 ( 本地MySQL、RDS、 EC2 MySQL )
  - log-bin=mysql-bin
  - binlog\_format=RAW
  - binlog\_checksum=None



## AWS Schema Conversion Tool



从 Oracle 和 SQL Server 迁出

将您的表、视图、存储过程和 DML 迁移到  
MySQL、MariaDB 和 Amazon Aurora

建议在需要时手动进行更改

# AWS Schema Conversion Tool

## 评估报告

- SCT 对源数据库进行分析，然后提供一个报告，其中包含推荐的目标引擎及有关自动和手动转换的信息

## 代码浏览器和建议

- 提示需要进行手动编辑地方，并提供架构和设计指南



# 有关异构迁移的注意事项

- 许多客户希望了解 “商业转开源”
- 考虑数据库的复杂性  
移植存储过程和函数中的代码可能需要手动更改代码
- 考虑支持目标数据库所需的技能  
在没有 GUI 的情况下，SQL Server 管理员可能会困惑
- 性能测试  
如果不确定，则使用较大的实例，之后再进行调整

# 重新设计架构

- 考虑将读取和写入分离
- 考虑将冷、热数据等进行分离
- 考虑为商业智能和报告使用数据仓库
- 考虑将 NoSQL 用于非关系型数据
- 请勿低估重新设计架构的成本和复杂性

# 合作伙伴解决方案

- Dataguise : 数据屏蔽和加密
- Informatica : 数据监管
- Delphix : 数据屏蔽和虚拟化
- Racemi : 虚拟机迁移
- CloudEndure : 虚拟机迁移
- Attunity CloudBeam
  
- 迁移合作伙伴包括...  
LogicWorks、CloudNexa、DataPipe、Pythian、Slalom、  
2<sup>nd</sup> Watch、Apps Associates、UST Global 等等...

# 回顾

- 为何要迁移数据库？
- RDS 和 EC2 数据库服务器应如何选择？
- Oracle GoldenGate 和 RDS 结合使用的几种模式？
- DMT 工具的工作原理
- DMT CDC 是怎么做到的？
- DMT 可支持的源和目标数据库
- DMT 的使用场景和注意事项
- SCT 的作用
- 异构迁移的注意事项
- 重新架构的注意事项
- 合作伙伴的解决方案

# 谢谢