



PostgreSQL 中文社区



PostgreSQL 中文社区

2021

PostgreSQL China Conference

主办：PostgreSQL 中文社区

# 第 11 届 PostgreSQL 中国技术大会

开源论道 × 数据驱动 × 共建数字化未来

## 逻辑复制在数据迁移中的应用

艾超 腾讯云高级工程师





# CONTENT



## 逻辑复制概念和演进

逻辑复制的功能和演进历史



## 数据迁移应用架构

应用场景和技术架构



## 逻辑复制原理详解

快照读、复制槽、解码插件、发布/订阅



## 数据迁移应用场景详解

真实世界的应用场景和解决方案



## 挑战和应对

安全性、碎片化、跟随社区



# 逻辑复制概念和演进

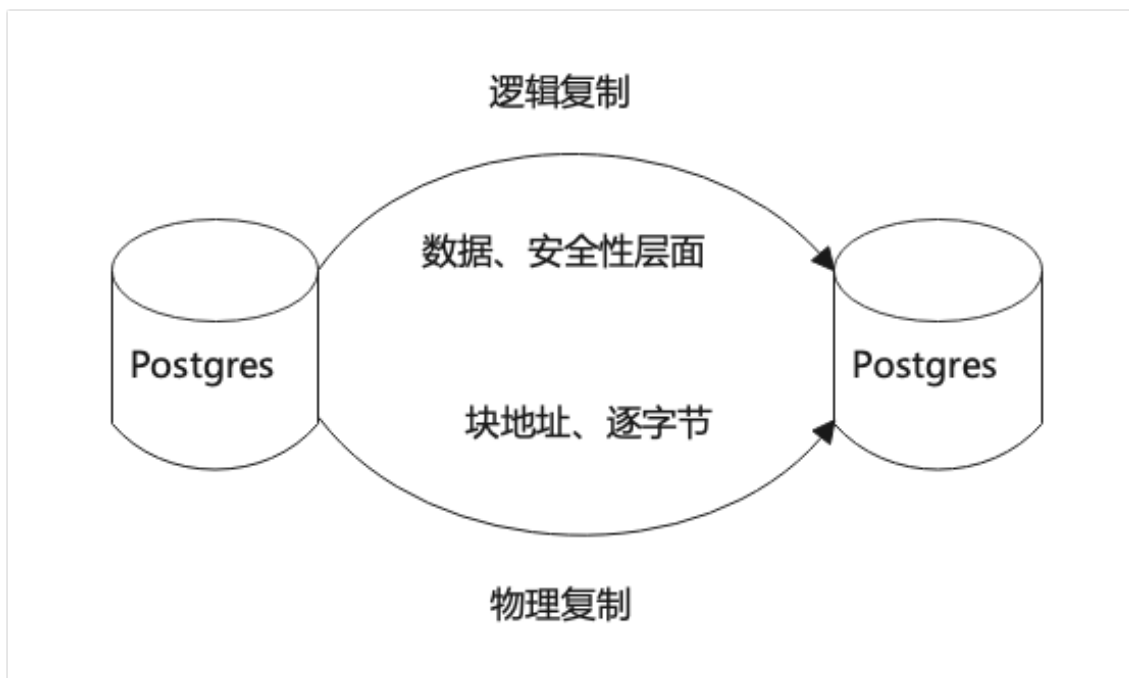
逻辑复制的功能和演进历史





# 逻辑复制 (Logical Replication)

逻辑复制是PG数据库的一种基于数据对象的复制标志 (Replica Identity)，来复制数据对象及数据变化的方法。之所以称为“逻辑”复制，是为了和“物理”复制加以区分。PostgreSQL支持两种复制机制，物理复制使用准确的块地址以及逐字节的复制方式，逻辑复制则允许在数据复制和安全性上更细粒度的控制。

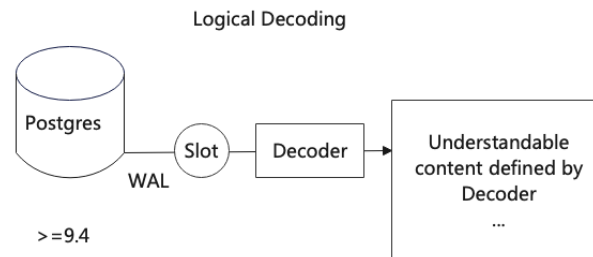




# 逻辑复制 (Logical Replication)

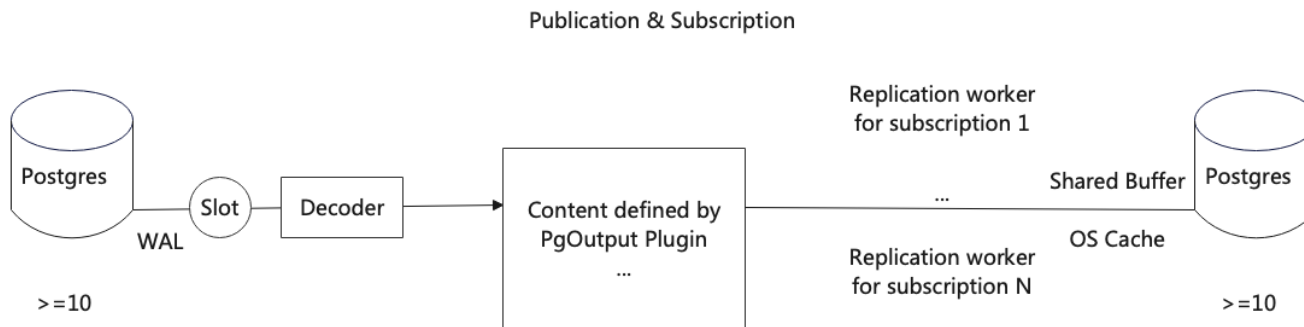
## 9.4以上及10以下版本

在9.4及以上、10以下版本，PostgreSQL为DML带来的数据修改流动到外部提供了基础设施，可用来实现复制解决方案和审计等目的。数据修改以流的形式发出，通过逻辑复制槽(Replication Slot)识别，配合解码插件(Decoding Plugin)，每一份数据修改在每个流中输出有且仅有一次。



## 10及以上版本:发布和订阅

在10及以上版本的逻辑复制使用了一种发布和订阅模型，发布和订阅可以是多对多的关系，例如允许有一个或者更多订阅者订阅一个发布这节点上的一个或多个发布。订阅者以与发布者相同的顺序应用数据，因此能在一个订阅中保证发布的事务一致性，这种数据复制的方式有时候也被称为事务性复制。





# 数据迁移应用架构

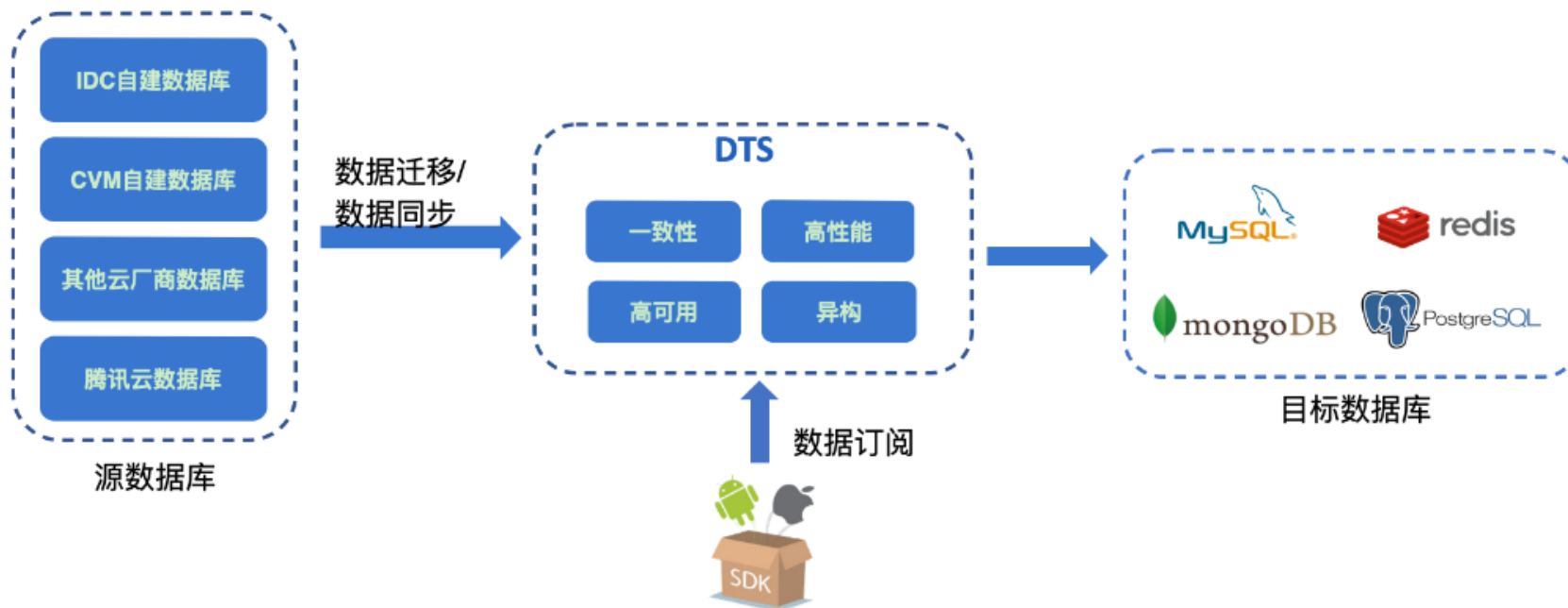
应用场景和技术架构





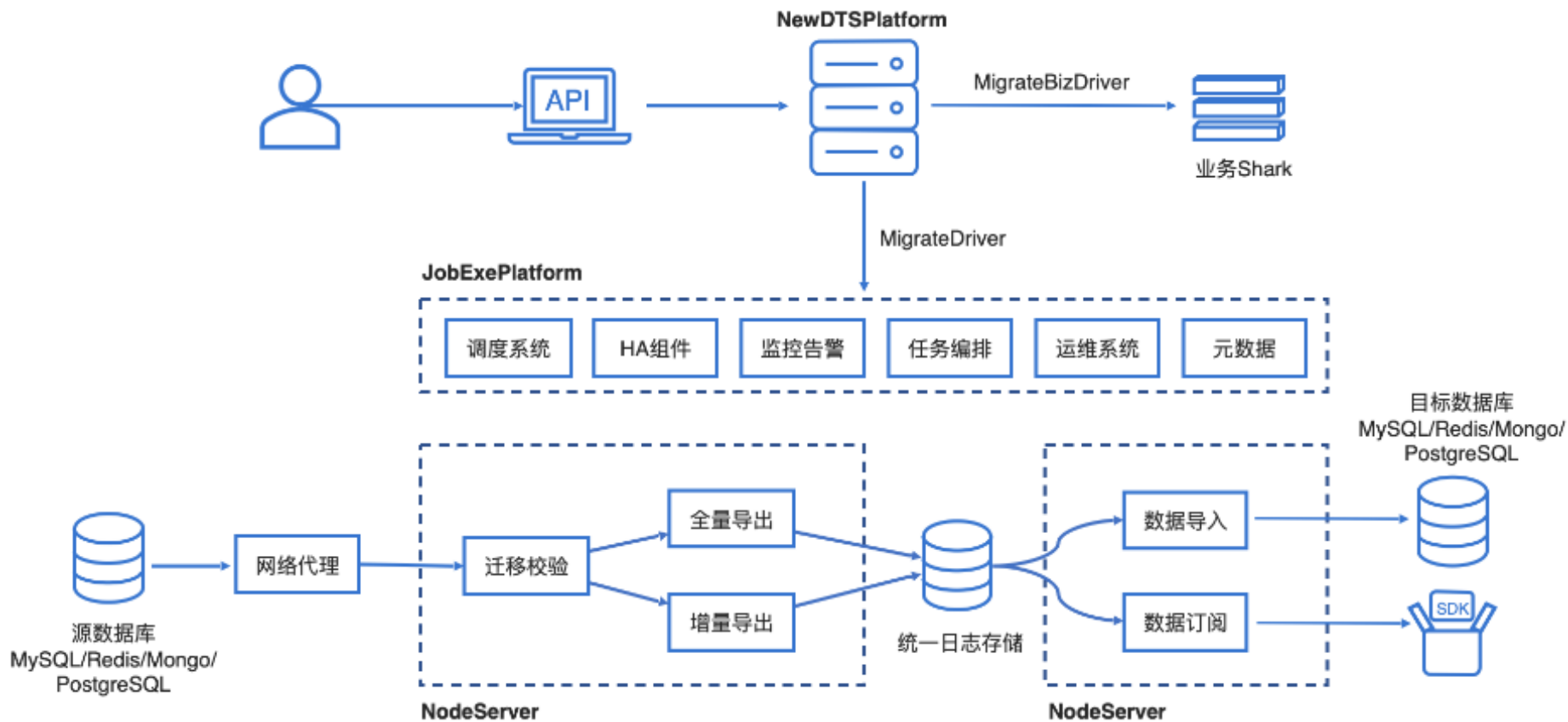
## 数据迁移的应用场景

云上的数据迁移产品能够帮助用户在业务不停服的前提下轻松完成数据库迁移上云，利用实时同步通道轻松构建高可用的数据库容灾架构，通过数据订阅来满足商业数据挖掘、业务异步解耦等场景需求。





# 数据迁移的技术架构







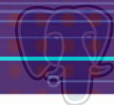
2021 PostgreSQL China Conference  
第 11 届 PostgreSQL 中国技术大会



PostgreSQL 中文社区

# 迁移原理详解

快照读、复制槽、解码插件、发布/订阅



# 复制标识 (Replica Identity)

复制标识 (Replica Identity)，是一个表级属性，用于控制记录被删除或修改时写入WAL的内容，只在逻辑复制中生效，只能通过ALTER TABLE来修改。至少有一个列的值在该记录的新旧版本发生了变化了，才会有记录发生。

DEFAULT

(用户表上的默认值) 在数据变更是将记录主键 (如有) 列的旧值。

USING  
INDEX

数据变更时记录指定索引列的旧值，索引必须是唯一键，且不是部分索引 (partial index)，不可延期 (deferrable)，并仅包含非空列。

FULL

在数据变更时记录所有列的旧值。

NOTHING

在数据变更时不记录任何旧值 (系统表上的默认值)。

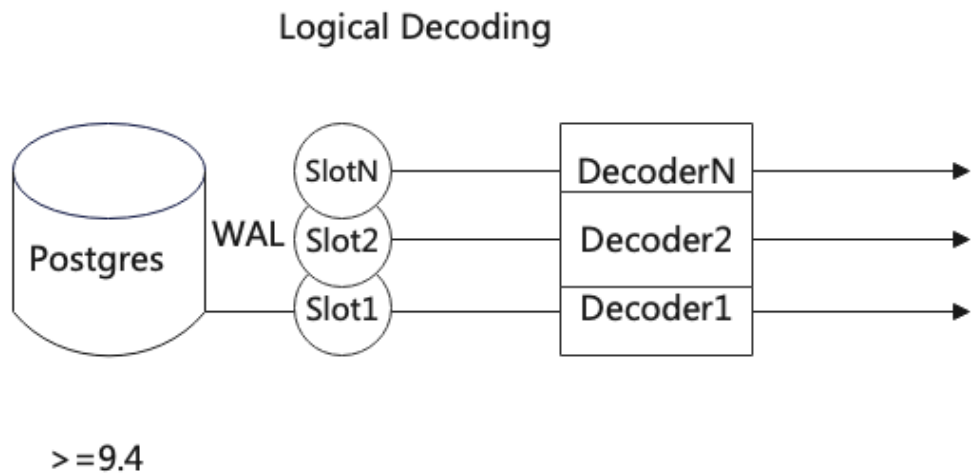




# 复制槽 (Replication Slot)

在逻辑复制中，一个槽表示一段数据变更流，它的顺序与源实例上数据发生变化的顺序一致，且槽中的每个数据变化不会重复。

不同的复制槽之间，消费位点是独立记录的，任一时刻，一个复制槽只能被一个客户端消费。





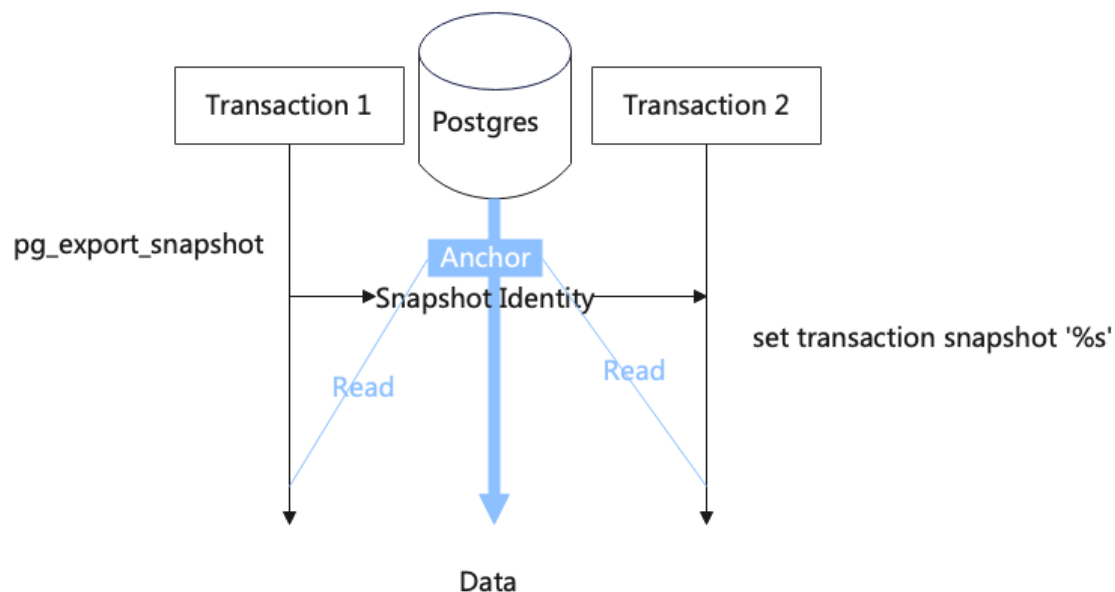
# 快照读 (Snapshot)

当两个或更多会话需要查看数据库中相同的内容时，快照同步是必须的。PG允许一个事务**export**它正在使用的快照，只要事务保持打开，其他事务可以**import**它的快照，来保障这些事务看到的数据与第一个事务看到的完全相同。

创建快照有2种方式：

- 1、利用CREATE\_REPLICATION\_SLOT在创建复制槽时自动**创建快照**，并保证事务一致性
- 2、使用系统函数pg\_export\_snapshot导出快照

在一个事务中获得快照的字符串标识后，事务结束前，在其他事务中导入快照，从而查看完全相同的数据。







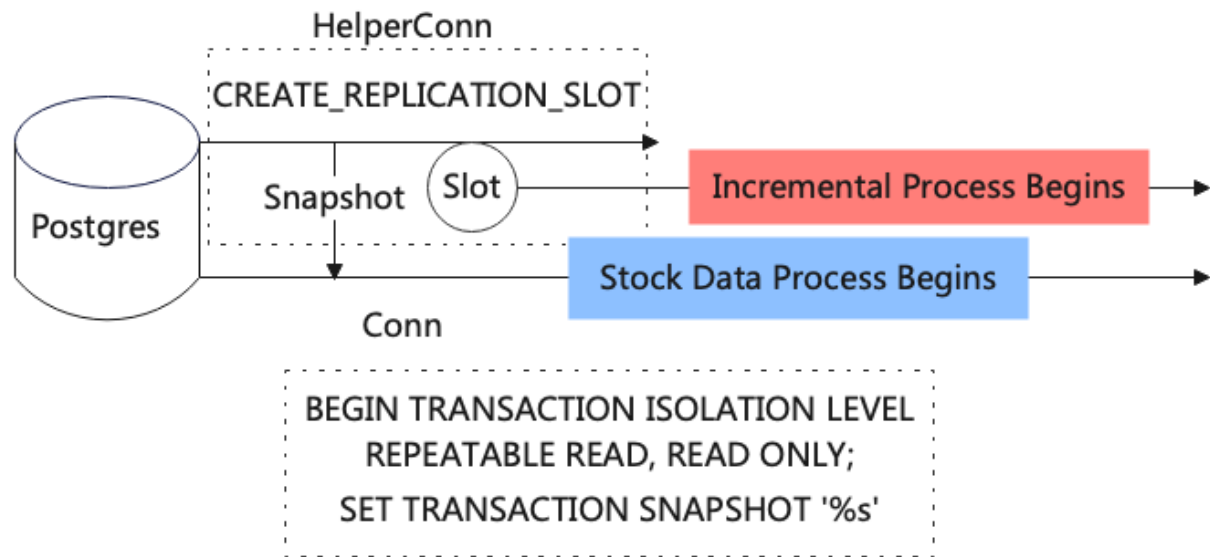
## 存量与增量的启动

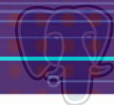
### 增量开启时:

创建连接, 创建复制槽, 导出快照, 记录快照标识。

### 然后开启全量:

新创建连接Conn, 开启事务, 导入快照, 进行数据读取和传输。





# 输出插件 (Output Plugins)

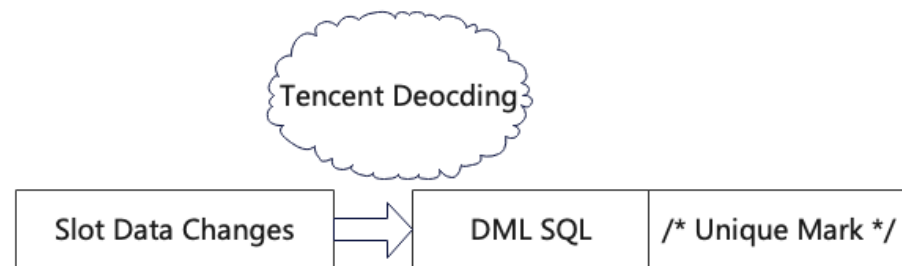
输出插件将WAL日志的内部格式，转换为复制槽的消费者需要的格式。  
插件的实现上，需要为函数\_PG\_output\_plugin\_init提供回调函数。

...

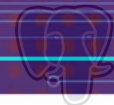
```
typedef struct OutputPluginCallbacks
{
    LogicalDecodeStartupCB startup_cb;
    LogicalDecodeBeginCB begin_cb;
    LogicalDecodeChangeCB change_cb;
    LogicalDecodeCommitCB commit_cb;
    LogicalDecodeShutdownCB shutdown_cb;
} OutputPluginCallbacks;
```

```
typedef void (*LogicalOutputPluginInit) (struct OutputPluginCallbacks *cb);
...
```

contrib/test\_decoding是PG内置一个插件实现，对数据变更的值做简单的打印。  
在此基础上做修改，我们将结构和数值合并为可执行的SQL语句，便于直接执行回放。





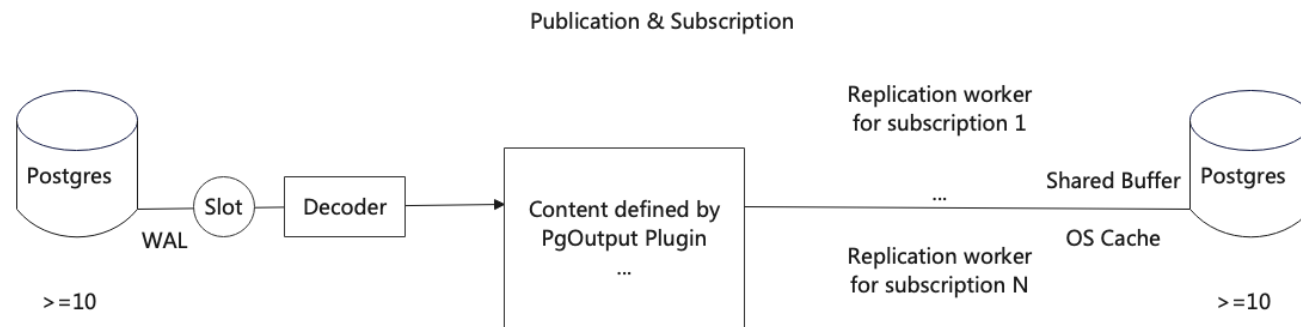


# 发布/订阅

在10以上版本，将SLOT的消费和回放封装为了发布订阅，让用户免于操作快照读、解码插件、复制槽等一系列底层功能，方便地获得稳定的全量+增量迁移通道。

需要注意：

- FOR ALL TABLES的发布需要superuser权限才能执行
- 对于没有主键的源表，依然使用DEFAULT复制标识，会导致表不可写
- 受限于版本，版本10支持Insert/Delete/Update，版本11+添加支持Truncate
- DDL变更无法自动复制
- 序列无法自动复制
- 分区表的父表，外表，大对象等无法自动复制





2021 PostgreSQL China Conference  
第 11 届 PostgreSQL 中国技术大会



PostgreSQL 中文社区

# 数据迁移应用场景详解

真实世界的应用场景和解决方案

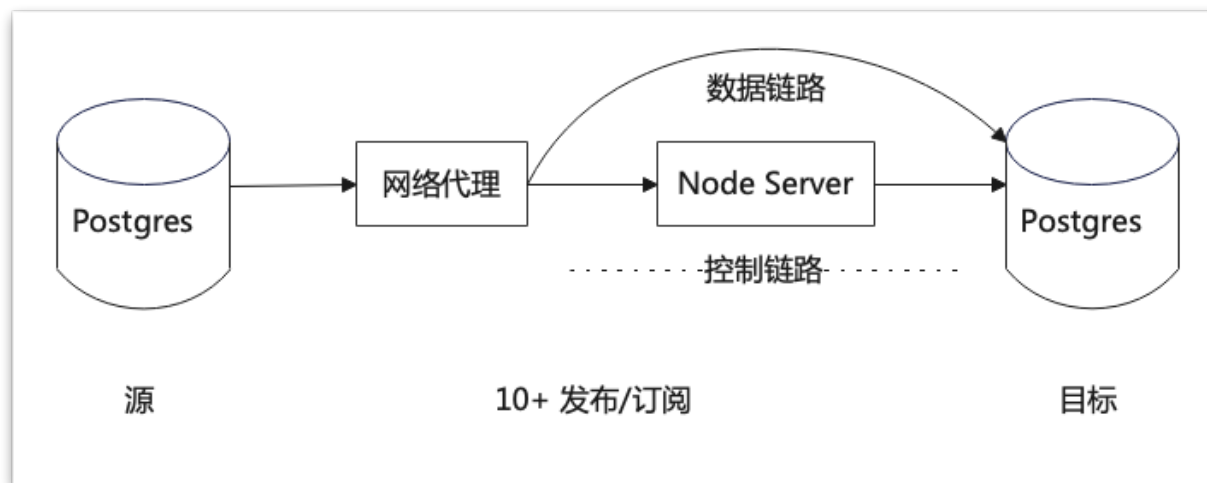
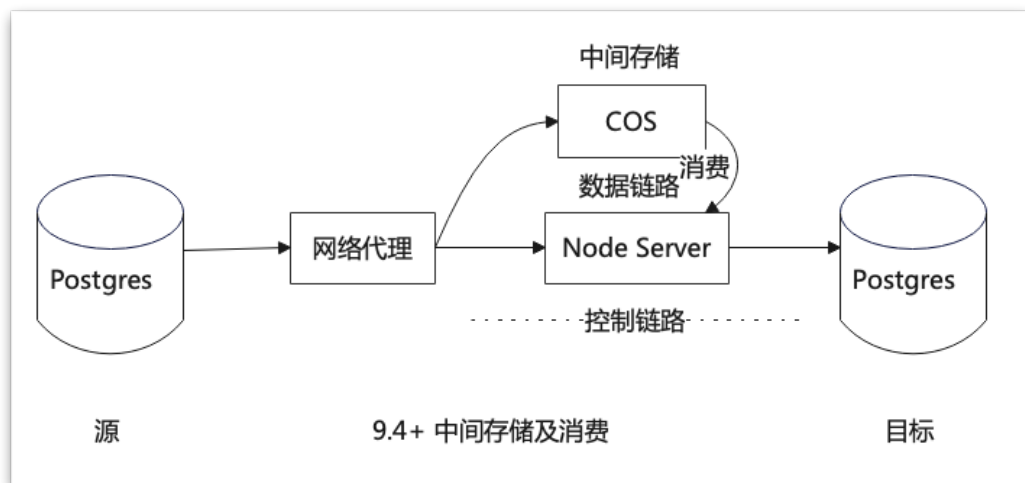
开源论道 × 数据驱动 × 共建数字化未来





# 数据迁移之网络连通性

云上数据库为了保证安全性，一般情况无法让实例直接访问外部网络。这样在迁移外部公网实例入云时，通过逻辑复制，首先就会遇到网络连通性问题。





# 数据迁移之无主键表

对于无主键(或唯一键)的表,

- 1、如果不修改复制标识, 将无法输出数据变更。
- 2、将Replica Identity设为FULL, 又将无法保证数据一致性。

实际迁移中, 将在预检查中提示用户修改表结构, 遵循合理的数据库设计规范。

否则, 因为存在数据不一致的风险, 将重点提示用户。

Query Editor Query History

```
1  -- 表结构
2  postgres=# create table tb(id int, name varchar);
3  postgres=# select ctid,* from tb;
4  ctid | id | name
5  -----+-----+-----
6  (0,1) |  1 | Alice
7  (0,2) |  1 | Alice
8  (0,3) |  1 | Alice
9  -- 源端
10 postgres=# alter table tb replica identity full;
11 postgres=# delete from tb where ctid = '(0,1)'; -- 仅删除第一条记录
12
13 -- 目标端
14 postgres=# delete from tb where id = 1 and name = 'Alice'; -- 删除了全部3条记录
```





# 数据迁移之DDL自动同步

逻辑复制在机制上无法自动同步DDL变更，但我们有其他方式来实现手动或半自动的DDL同步。

1、迁移应用自动探测源和目标端的结构差异，生成差异DDL变更至目标。

优点: 不需要superuser权限，不需要额外插件安装。

缺点: 无法保证数据的事务一致性，只能应用较简单的场景。

2、使用事件触发器追踪表结构变更，并记录至一张特殊的表，逻辑复制监听该表内容，发现更新即变更到目标。

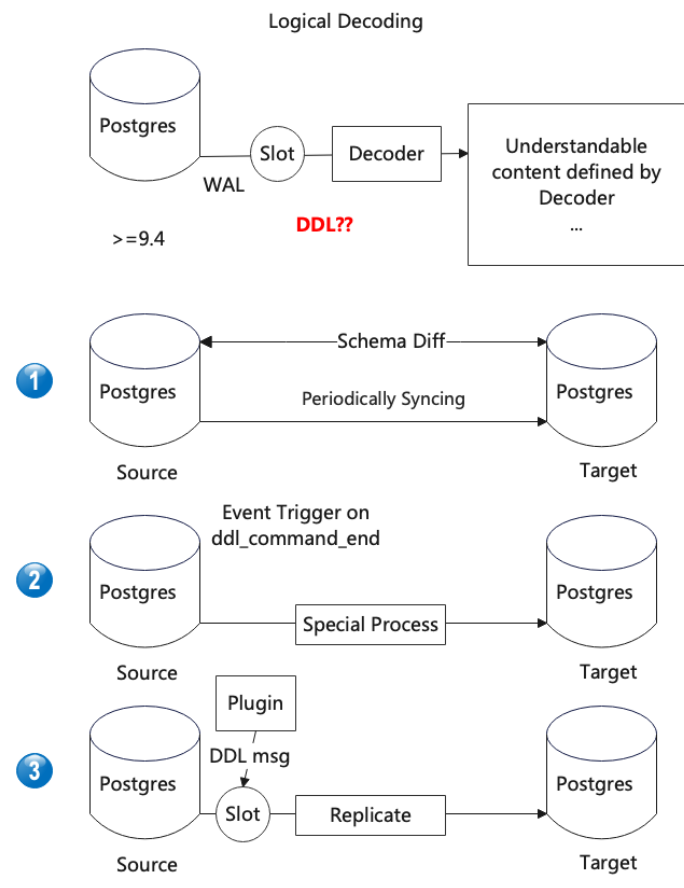
优点: 可初步保证事务一致性。

缺点: 对源账号有要求，需要superuser权限。

3、安装外部插件解析WAL，在Slot中添加DDL信息，用原始的逻辑复制变更至目标。

优点: 可保证事务一致性，

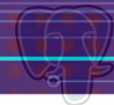
缺点: 需要源实例的运维和插件安装权限。





# 挑战和应对

安全性、碎片化、跟随社区



# 挑战和应对

## 安全性

随着云上数据库技术的不断发展演进，各个厂商也锤炼出了针对自身定制的网络安全规则和数据库用户权限体系，相较于传统迁移场景的完全复制，我们需要按源和目标各自的安全规则，来定制迁移策略。例如：在不同厂商之间迁移，数据库账号权限点不尽相同，考虑到各个厂商的安全策略，为了顺利完成迁移，就需要对账号权限点做适当裁剪。

## 碎片化

PG是一个开源、商业友好的数据库，基于PG的数据库发行厂商也非常多，在内核架构上也有着大大小小的差异性。为了提供更好的服务，我们必须持续对各个发行厂商的产品进行适配，给用户带来更好的使用体验。

## 跟随社区

PG有一个强大的社区，在社区版本迭代中，会有持续不断的特性变更和演进。以迁移场景为例，社区11版本删除了abstime/reftime/tinterval数据类型，逻辑复制增加了对truncate的支持，在社区13版本中逻辑复制增加了对partitioned table的支持，因此版本间的迁移链路也需要随之不断迭代。我们将持续跟进社区，和云数据库自身内核演进一起，做更好的数据迁移解决方案。





2021 PostgreSQL China Conference  
第 11 届 PostgreSQL 中国技术大会



PostgreSQL 中文社区

# THANKS

谢谢观看

开源论道 × 数据驱动 × 共建数字化未来