

PostgreSQL 17 主从部署、配置优化及备份脚本最佳实践

www

2025-02-08

933

PostgreSQL 17 主从部署、配置优化及备份脚本最佳实践

PostgreSQL 作为一款功能强大且广泛应用的开源关系数据库管理系统，在企业级应用场景中发挥着重要作用。主从部署不仅能显著提升数据库的可用性，还可实现读写分离与数据备份。本文将深入介绍 PostgreSQL 17 主从部署的详细步骤、配置文件的优化策略，同时提供实用的备份脚本。

一、环境准备

在开展主从部署工作之前，务必确保满足以下条件：

- 操作系统**：推荐使用支持 PostgreSQL 运行的 Linux 系统，如 CentOS 7 或更高版本。
- 服务器**：至少准备两台服务器，分别作为主库（Master）和从库（Standby）。
- 软件安装**：在主库和从库上都完成 PostgreSQL 17 的安装。

安装 PostgreSQL 17

以 CentOS 系统为例，可通过 yum 进行安装，具体步骤如下：

```
# 安装 PostgreSQL 官方 yum 源
yum install -y https://download.postgresql.org/pub/repos/yum/reposrms/EL-7-x86_64/pgdg-redh

# 安装 PostgreSQL 17
yum install -y postgresql17-server

# 初始化数据库
/usr/pgsql-17/bin/postgresql-17-setup initdb

# 启动 PostgreSQL 服务
systemctl start postgresql-17

# 设置开机自启
systemctl enable postgresql-17
```

二、主库配置

1. 创建复制用户

在主库上创建专门用于复制的用户，例如 `replica`，并赋予 `REPLICATION` 权限。

```
CREATE ROLE replica LOGIN REPLICATION ENCRYPTED PASSWORD 'your_password';
```



www

26

文章

7

粉丝

4K+

浏览量

获得了 10 次点赞

内容获得 2 次评论

获得了 16 次收藏

热门文章

mysql 主从切换

2025-01-11

770浏览

解决Oracle数据库ORA-01113和ORA-01110错误

2025-01-13

487浏览

如何查看 Oracle 数据库中的隐藏参数

2025-01-13

424浏览

oracle序列重置（非删除重建）

2025-03-11

220浏览

Oracle 19c 审计机制变化：从 DBA_AUDIT_SESSION 到统一审计

2025-02-10

194浏览

在线实训环境入口



PostgreSQL在线实训环境

查看详情 »

最新文章

mgr异常处理1

2025-05-06

25浏览

db拉起脚本

2025-04-02

169浏览

dg库拉起脚本

2025-04-02

77浏览

oracle序列重置（非删除重建）

2025-03-11

220浏览

dg修复

2025-03-05

103浏览

2. 配置 pg_hba.conf

编辑 pg_hba.conf 文件，添加从库的 IP 地址和复制用户的访问权限。

```
# 允许从库通过 replica 用户进行复制连接
host      replication      replica      从库 IP/32      md5
```

3. 配置 postgresql.conf

修改 postgresql.conf 文件，启用流复制相关参数。

```
# 监听所有 IP 地址
listen_addresses = '*'

# 启用归档模式
archive_mode = on
archive_command = 'cp %p /var/lib/pgsql/17/data/archive/%f' # 归档路径可自行调整

# 设置流复制相关参数
wal_level = replica
max_wal_senders = 10 # 根据实际需求调整最大复制连接数
wal_keep_segments = 100 # 保留的 WAL 段数量，用于故障恢复时的数据同步
```

4. 重启主库

修改完配置文件后，重启 PostgreSQL 服务使配置生效。

```
systemctl restart postgresql-17
```

三、从库配置

1. 停止从库服务

在从库上停止 PostgreSQL 服务。

```
systemctl stop postgresql-17
```

2. 清空数据目录

清空从库的数据目录，为从主库同步数据做准备。

```
rm -rf /var/lib/pgsql/17/data/*
```

3. 使用 pg_basebackup 同步数据

使用 pg_basebackup 工具从主库同步数据到从库。

```
su - postgres
pg_basebackup -h 主库 IP -U replica -D /var/lib/pgsql/17/data -P -R
```

- h ：主库的 IP 地址。
- U ：复制用户。
- D ：从库的数据目录。
- P ：显示进度。
- R ：生成 recovery.conf 配置文件（PostgreSQL 17 中为 standby.signal 和 postgresql.auto.conf ）。

目录

- PostgreSQL 17 主从部署、配置优化...
 - 一、环境准备
 - 安装 PostgreSQL 17
 - 二、主库配置
 - 1. 创建复制用户
 - 2. 配置 pg_hba.conf
 - 3. 配置 postgresql.conf
 - 4. 重启主库
 - 三、从库配置
 - 1. 停止从库服务
 - 2. 清空数据目录

4. 配置 postgresql.conf

编辑 postgresql.conf 文件，启用从库相关参数。

```
# 启用热备模式
hot_standby = on

# 从库可接受的最大连接数
max_connections = 200 # 根据实际需求调整
```

5. 启动从库

完成配置后，启动 PostgreSQL 服务。

```
systemctl start postgresql-17
```

四、验证主从部署

1. 在主库上检查复制状态

连接到主库的 PostgreSQL，执行以下 SQL 语句查看复制状态。

```
SELECT * FROM pg_stat_replication;
```

若一切正常，会看到从库的相关信息，包括复制状态、同步进度等。

2. 在从库上检查接收状态

连接到从库的 PostgreSQL，执行以下 SQL 语句查看 WAL 接收状态。

```
SELECT * FROM pg_stat_wal_receiver;
```

确保 state 字段显示为 streaming，表示从库正在正常接收主库的 WAL 日志。

五、主从切换

在某些情况下，如主库故障或计划维护，需要进行主从切换，步骤如下：

1. 提升从库为主库

在从库上执行以下命令，将从库提升为主库。

```
su - postgres
pg_ctl promote
```

2. 重新配置原主库为从库

若原主库恢复正常，需将其重新配置为从库。首先停止原主库服务，清空数据目录，然后使用 pg_basebackup 重新同步数据，并配置 postgresql.conf 文件为从库模式。

六、配置文件优化

1. 内存相关参数优化

shared_buffers

该参数表示数据库服务器用于缓存数据块的共享内存区域大小，增大此值可减少磁盘 I/O，提高查询性能。一般可设置为系统内存的 25% - 50%，例如系统有 8GB 内存，可设置为：

```
shared_buffers = '2GB'
```

work_mem

决定了在排序操作和哈希表创建时使用的内存大小。对于复杂查询，适当增大该值可减少磁盘临时文件的使用，提高查询速度。通常设置为几百 MB，例如：

```
work_mem = '64MB'
```

maintenance_work_mem

用于维护操作（如 VACUUM、CREATE INDEX 等）的内存大小。增大该值可加快这些操作的执行速度，一般设置为系统内存的 5% - 10%，例如：

```
maintenance_work_mem = '512MB'
```

2. 磁盘 I/O 相关参数优化

wal_buffers

WAL（Write - Ahead Logging）缓冲区的大小，用于存储未写入磁盘的 WAL 记录。适当增大该值可减少 WAL 日志的磁盘 I/O，通常设置为 shared_buffers 的 3% - 5%，例如：

```
wal_buffers = '64MB'
```

fsync

控制是否强制将数据同步写入磁盘。设置为 on 可保证数据的持久性，但会增加磁盘 I/O 开销；设置为 off 可提高性能，但在系统崩溃时可能会丢失部分数据。若对数据的持久性要求不是非常高，或者使用了电池备份的 RAID 控制器等可靠的存储设备，可考虑设置为 off：

```
fsync = off
```

synchronous_commit

控制事务提交时是否等待 WAL 记录写入磁盘。on 表示同步提交，off 表示异步提交。若对数据一致性要求不是特别高，为提高性能，可设置为 off：

```
synchronous_commit = off
```

3. 并发相关参数优化

max_connections

表示允许的最大客户端连接数。需根据系统性能和业务需求进行调整，一般小型应用可设置为几十到几百，大型应用可能需设置为上千，例如：

```
max_connections = 200
```

max_worker_processes

表示系统中可以同时运行的后台工作进程的最大数量。通常可设置为 CPU 核心数的 2 - 4 倍，例如服务器有 4 个 CPU 核心，可设置为：

```
max_worker_processes = 8
```

max_parallel_workers_per_gather

控制每个查询可以使用的并行工作进程的最大数量。可根据系统资源和查询复杂度进行调整，一般设置为 max_worker_processes 的一半左右，例如：

```
max_parallel_workers_per_gather = 4
```

4. 日志相关参数优化

log_destination

指定日志的输出目标，为方便后续的日志分析，建议使用 csvlog：

```
log_destination = 'csvlog'
```

logging_collector

控制是否启用日志收集器，设置为 on 以启用日志收集：

```
logging_collector = on
```

log_statement

指定哪些 SQL 语句需要记录到日志中。在开发和测试环境中，可设置为 'all' 以便全面记录 SQL 语句；在生产环境中，为减少日志量，可设置为 'ddl' 或 'mod'：

```
log_statement = 'mod'
```

5. 复制相关参数优化（主从部署场景）

wal_level

指定 WAL 日志的详细级别，对于流复制，需设置为 replica 或更高：

```
wal_level = 'replica'
```

max_wal_senders

表示主库可以同时向从库发送 WAL 日志的最大连接数。根据从库的数量和并发复制需求进行调整，一般设置为比从库数量略大的值，例如有 3 个从库，可设置为：

```
max_wal_senders = 5
```

wal_keep_segments

指定主库保留的 WAL 段数量，用于从库在网络中断等情况下恢复数据同步。根据网络稳定性和业务需求进行调整，一般设置为几十到几百，例如：

```
wal_keep_segments = 100
```

修改配置文件后，需重启 PostgreSQL 服务使配置生效：

```
sudo systemctl restart postgresql-17
```

七、PostgreSQL 备份脚本

以下是一个简单的 PostgreSQL 全量备份脚本示例，可将其保存为 `pg_backup.sh`，并添加执行权限：

```
#!/bin/bash

# 备份目录
BACKUP_DIR="/var/backup/postgresql"

# 数据库用户名
DB_USER="postgres"

# 数据库名称
DB_NAME="your_database_name"

# 日期格式
DATE=$(date +%Y%m%d%H%M%S)

# 创建备份目录（如果不存在）
mkdir -p $BACKUP_DIR

# 执行备份
pg_dump -U $DB_USER $DB_NAME > $BACKUP_DIR/$DB_NAME-$DATE.sql

# 检查备份是否成功
if [ $? -eq 0 ]; then
    echo "备份成功：$BACKUP_DIR/$DB_NAME-$DATE.sql"
else
    echo "备份失败"
fi
```

使用方法

```
chmod +x pg_backup.sh
./pg_backup.sh
```

八、最佳实践建议

- 定期备份**：无论是主库还是从库，都应定期进行全量和增量备份，以防止数据丢失。可使用上述备份脚本结合 `cron` 任务实现定期备份。
- 监控与告警**：设置监控系统，实时监控主从库的状态、复制延迟等指标，并配置告警机制，以便及时发现和处理问题。
- 性能优化**：根据实际业务需求，合理调整主从库的参数，如 `max_connections`、`shared_buffers` 等，以提升数据库性能。
- 测试与演练**：定期进行主从切换测试和故障恢复演练，确保在实际故障发生时能够快速、准确地进行处理。

通过以上步骤和最佳实践建议，你可以成功部署并管理 PostgreSQL 17 的主从架构，同时优化数据库配置，提高数据库的可用性和性能。在实际应用中，应根据具体业务场景和需求进行适当调整和优化。

🔗 墨力计划 postgresql

「喜欢这篇文章，您的关注和赞赏是给作者最好的鼓励」

关注作者 赞赏

【版权声明】本文为墨天轮用户原创内容，转载时必须标注文章的来源（墨天轮），文章链接，文章作者等基本信息，否则作者和墨天轮有权追究责任。如果您发现墨天轮中有涉嫌抄袭或者侵权的内容，欢迎发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

评论

分享你的看法，一起交流吧~



星星之火



支持 PostgreSQL 运行的 Linux 系统，如 CentOS 7 或更高版本

6月前 点赞 评论

相关阅读

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享 (2025/7/25期)

墨天轮小助手 472次阅读 2025-07-25 15:54:18

PostgreSQL DDL变更的坑和巧妙方案

liuzhilong62 330次阅读 2025-07-19 11:50:29

IvorySQL 专家顾问委员会成立啦！

IvorySQL开源数据库社区 239次阅读 2025-07-15 10:15:33

ACDU周度精选 | 本周数据库圈热点 + 技术干货分享 (2025/8/8期)

墨天轮小助手 227次阅读 2025-08-08 15:29:45

PostgreSQL初/中/高级认证考试 (7.26) 通过考生公示

开源软件联盟PostgreSQL分会 163次阅读 2025-07-31 09:44:43

PostgreSQL优化器提示扩展——pg_hint_plan

chirpyli 126次阅读 2025-07-29 21:22:24

PostgreSQL 会重写表的几种情况

墨香溪 122次阅读 2025-07-26 08:12:45

PITR玩砸了，看DBA如何删库跑路

NickYoung 110次阅读 2025-08-04 20:31:40

【PostgreSQL】统计信息：查询优化与运行监控的基础支柱

神经蛙 104次阅读 2025-07-17 23:08:06

PostgreSQL数据库pg_hint_plan实现原理解析

chirpyli 95次阅读 2025-07-30 22:25:23