

优炫数据库repmgr工具使用手册 2.1



UXSINO
优 炫 软 件

优炫数据库repmgr工具使用手册 2.1

版权 © 2016-2023 北京优炫软件股份有限公司

法律声明

优炫数据库管理系统(简称: UXDB) 是由北京优炫软件股份有限公司开发并发布的一款商业性数据库管理系统。

优炫数据库管理系统 (UXDB) 的一切知识产权以及与该软件产品相关的所有信息内容, 包括但不限于: 文字表述及其组合、图标、图饰、图表、色彩、界面设计、版面框架、有关数据、及电子文档等均属北京优炫软件股份有限公司所有。本软件及其文档的任何使用、复制、修改、出租、传播、销售及分发等行为均须经北京优炫软件股份有限公司书面许可。

凡侵犯北京优炫软件股份有限公司知识产权的行为, 北京优炫软件股份有限公司将依法追究其法律责任。

本声明的最终解释权归属于北京优炫软件股份有限公司。



和其他优炫公司商标均为北京优炫软件股份有限公司的商标。

本文档提及的其他所有商标或注册商标, 由各自的所有人拥有。

注意

由于产品版本安装或其他原因, 本文档内容会不定期进行更新。除非另有约定, 本文档仅作为使用指导, 本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

北京优炫软件股份有限公司 (总部)

- 地址: 北京市海淀区学院南路62号中关村资本大厦11层 (邮编: 100081)
 - 网址: <http://www.uxsino.com>
 - 邮箱: <uxdb_support@uxsino.com>
 - 电话: 010-82886998
 - 传真: 010-82886338
 - 服务热线: 400-650-7837
-

目录

前言	v
1. 文档目的	v
2. 文档对象	v
3. 修改记录	v
1. 概述	1
2. 原理	2
3. 依赖软件	3
4. repmgr部署	4
4.1. 准备环境	4
4.2. 准备工作	4
4.2.1. 网络配置	4
4.2.2. 配置sudo免密码	5
4.2.3. ssh免密码	5
4.2.4. 安装rsync	5
4.3. 配置主节点数据库	6
4.4. 配置主节点的repmgr	7
4.5. 配置备节点的repmgr	9
4.6. 启动守护进程	10
4.7. 查看集群信息	10
4.8. 模式切换说明	10
4.9. 配置多个备节点	11
5. repmgr使用命令	13
5.1. 显示节点信息和复制状态	13
5.2. 显示集群注册节点信息	13
5.3. 查看守护进程状态	13
5.4. 注册主节点	13
5.5. 注销主节点	14
5.6. 备节点克隆	14
5.7. 注册备节点	14
5.8. 注销备节点	14
5.9. rejoin	14
5.10. 手动切换主从节点	15
5.11. 启动和停止守护进程	15
6. 高可用场景示例	16
6.1. 一主一备	16
6.1.1. 主库异常	16
6.1.2. 备库异常	17
6.2. 一主多备	18
6.2.1. 主库异常	18
6.2.2. 备库异常	19

表格清单

1. 文档更新记录	v
-----------------	---

前言

1. 文档目的

本文档介绍优炫数据库repmgr工具的部署和使用。

2. 文档对象

- 技术支持工程师
- 维护工程师

3. 修改记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

表 1. 文档更新记录

工具版本	发布日期	修改说明
2.1.1.5C	2022-10-27	第一次正式发布。

第 1 章 概述

repmgr可以为单机集群提供一个高可用方案，高可用集群中包含两个或多个节点，一个为主节点，对外提供服务，其它是备节点。当主节点发生故障时，如数据库服务终止、网络故障、停电时，备节点能够立即接管集群，对外提供服务。在切换过程中对用户是透明的。

repmgr高可用集群搭建完成后，可以在主节点停止服务后将对外服务自动切换到备节点，并可以拉起主节点重新加入集群做为备节点提供服务。在备节点停止服务后也可以拉起备节点继续提供服务。

数据库服务器可以持续对外提供服务，包括发生意外情况后依然可以提供服务，可以通过高可用集群解决单机数据库会因意外而停止服务的情况。

第 2 章 原理

repmgr是一套工具，用于管理UXDB服务器集群中的复制和故障转移。它通过程序增强了uxdb的内置复制功能，这些程序可以设置备用服务器，监视复制并执行管理任务，例如故障转移或切换操作。

故障转移时根据LSN、优先值、id值选择一个最合适的备节点提升为主。先比较LSN值，LSN值大的成为主节点；当LSN值一样时比较优先值，优先值大的成为主节点；当LSN值和优先值都一样时，id值较小的会成为主节点。

repmgr流复制管理工具对集群节点的管理是基于一个分布式的管理方式，有repmgr和repmgrd两个命令。

- repmgr命令实现对集群节点的管理，如注册主/备节点、Clone节点，Promote节点，Follow节点以及Switchover操作等。
- repmgrd命令用来启动repmgr系统的守护进程，用以对集群节点的监控。

第 3 章 依赖软件

集群的系统上需要安装rsync软件。

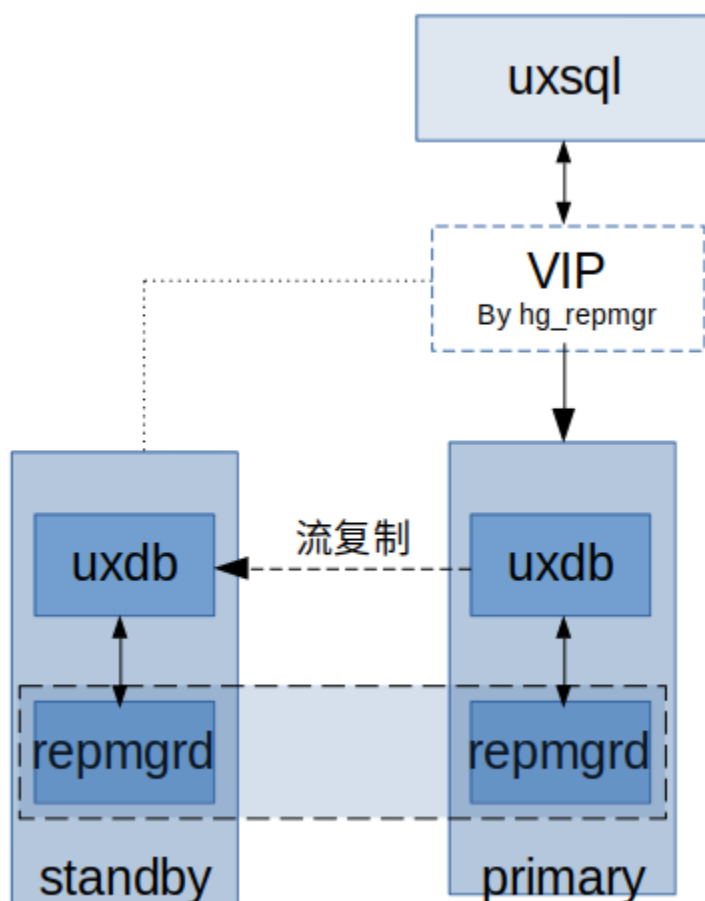
第 4 章 repmgr部署

高可用集群搭建完成后为客户端提供和单机无差别的服务，客户端通过IP和端口访问数据库。

4.1. 准备环境

准备两个虚拟机，一主一备异步复制的环境。

以主节点IP为192.71.0.114，从节点IP为192.71.0.115为例。



4.2. 准备工作

在需要搭建repmgr高可用集群的机器上，进行一些准备工作，包括网络配置、sudo权限、ssh配置等。

4.2.1. 网络配置

配置网络环境, 保证机器之间可以互相ping通，添加防火墙规则，允许访问。

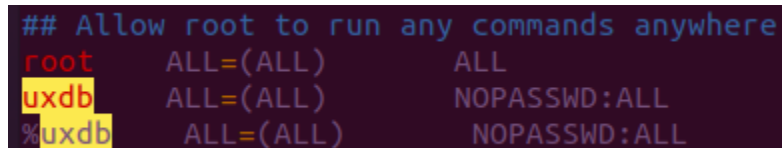
centos上关闭防火墙，命令如下所示。

```
//查看防火墙状态
firewall-cmd --state
//关闭防火墙
systemctl stop firewalld
//禁止防火墙
systemctl disable firewalld
```

4.2.2. 配置sudo免密码

虚拟IP功能中需要sudo命令免密码权限。

```
sudo vim /etc/sudoers
```



```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
uxdb    ALL=(ALL)        NOPASSWD:ALL
%uxdb   ALL=(ALL)        NOPASSWD:ALL
```

4.2.3. ssh免密码

ssh免密操作如下所示(操作系统用户uxdb)。

1. 在每个节点执行如下命令。

```
$ ssh-keygen -t rsa
```

2. 每个节点分别执行如下命令。

```
$ ssh-copy-id -i .ssh/id_rsa.pub uxdb@192.71.0.114
$ ssh-copy-id -i .ssh/id_rsa.pub uxdb@192.71.0.115
```

3. 各自机器上，测试ssh登录另一机器。

```
//114机器
ssh 192.71.0.115
//115机器
ssh 192.71.0.114
```

4. 验证成功后通过exit退出。

4.2.4. 安装rsync

系统上需要安装rsync软件，centos的虚拟机上已安装好。

```
//检查是否安装过rsync，whereis rsync也可以
rpm -qa|grep rsync
//如果未安装，使用yum安装rsync
yum install rsync
```

4.3. 配置主节点数据库

在主节点（192.71.0.114）机器上初始化集群，配置流复制等。

1. 初始化集群。

```
./initdb -W -D datarep
```

设置UXDATA环境变量。

```
sudo vim /etc/profile  
export UXDATA=/home/uxdb/uxdbinstall/dbsql/bin/datarep/  
source /etc/profile
```

2. 配置主节点uxsinodb.conf文件，仅列出修改的部分。

需要在uxdb目录下创建archive目录，用于归档日志。

```
mkdir /home/uxdb/archive
```

修改数据库配置uxsinodb.conf文件。

```
cd /home/uxdb/uxdbinstall/dbsql/bin  
vim datarep/uxsinodb.conf  
  
wal_level=replica  
archive_mode=on  
archive_command='cp %p /home/uxdb/archive/%f'  
max_wal_senders=10  
wal_keep_segments=512  
hot_standby=on  
shared_preload_libraries = 'repmgr'  
full_page_writes = on  
wal_log_hints=on
```

3. 配置主节点ux_hba.conf文件。

将以下配置加入ux_hba.conf文件，其中192.71.0.0/24表示支持192.71.0.1~254间的IP访问。

```
vim datarep/ux_hba.conf
```

```
local replication repmgr trust  
host replication repmgr 127.0.0.1/32 trust  
host replication repmgr 192.71.0.0/24 trust  
local repmgr repmgr trust  
host repmgr repmgr 127.0.0.1/32 trust  
host repmgr repmgr 192.71.0.0/24 trust
```

```
# TYPE DATABASE USER ADDRESS METHOD
local replication repmgr trust
host replication repmgr 127.0.0.1/32 trust
host replication repmgr 192.71.0.0/24 trust
local repmgr repmgr trust
host repmgr repmgr 127.0.0.1/32 trust
host repmgr repmgr 192.71.0.0/24 trust
# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 0.0.0.0/0 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all md5
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

注意

新增的repmgr用户的配置项，放在ux_hba.conf文件中的其他配置项的前面，否则使用repmgr用户建立连接的时候，会出现没有设置密码的错误。

```
[uxdb@song197:~/uxdbinstall/dbsql/bin]$ repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf primary register --force
INFO: connecting to primary database...
ERROR: connection to database failed
DETAIL:
fe_sendauth: no password supplied
DETAIL: attempted to connect using:
user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.197 fallback_application_name=repmgr
```

4. 创建repmgr用户和数据库。

//启动数据库。

./ux_ctl -D datarep/ -l logfile start

//创建用户并输入超级用户密码，当前密码是123456。

createuser -s repmgr

//创建数据库并输入超级用户密码，当前密码是123456。

createdb repmgr -O repmgr

5. 连接测试。

在192.71.0.115上连接192.71.0.114上的数据库。

uxsql 'host=192.71.0.114 user=repmgr dbname=repmgr connect_timeout=2'

4.4. 配置主节点的repmgr

1. 配置文件repmgr.conf

在数据库安装路径bin目录下创建repmgr.conf文件，并至少包含下面的配置。其中IP和数据目录等按照实际的数据库修改。priority表示在故障转移时候的优先级，优先级高的会被提升为主。

repmgr有三个默认文件路径，配置文件repmgr.conf只需要放在其中一个路径下，在bin目录下创建相当于第二个默认路径（./repmgr.conf）。

- a. /home/uxdb/uxdbinstall/dbsql/conf/repmgr.conf
- b. ./repmgr.conf
- c. /etc/repmgr.conf

vim repmgr.conf

```
node_id=1
node_name='node1'
conninfo='host=192.71.0.114 user=repmgr dbname=repmgr connect_timeout=2'
data_directory='/home/uxdb/uxdbinstall/dbsql/bin/datarep'
log_file='/home/uxdb/uxdbinstall/dbsql/bin/logrepmgrd.log'
ux_bindir='/home/uxdb/uxdbinstall/dbsql/bin'
repmgr_bindir='/home/uxdb/uxdbinstall/dbsql/bin'
failover=automatic
priority=100
connection_check_type=ping
reconnect_attempts=3
reconnect_interval=2
promote_command='repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby promote --log-to-file'
follow_command='repmgr standby follow -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf --log-to-file --upstream-node-id=%n'
monitoring_history=yes
monitor_interval_secs=2
standby_disconnect_on_failover=true
service_start_command = '/home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/datarep -l logfile start'
service_stop_command = '/home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/datarep -l logfile stop'
service_restart_command = '/home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/datarep -l logfile restart'
service_reload_command = '/home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/datarep -l logfile reload'
repmgrd_service_start_command = './repmgrd -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf -d'
repmgrd_service_stop_command = 'kill `cat /tmp/repmgrd.pid`'
virtual_ip = '192.71.0.190/32' #虚拟IP地址
network_card = 'ens33' #本机网卡名称
check_brain_split=true
stream_replication_mode='async' #本节点的同步/异步流模式（当前配置代表本节点是异步流复制模式，如果从async修改为sync，需要重启守护进程，使配置生效）
```

2. 注册主节点

启动数据库服务器后可以进行主节点的注册。

- a. 启动数据库。

```
./ux_ctl -D datarep/ start
```

b. 注册主节点。

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf primary register --force
```

4. 5. 配置备节点的repmgr

1. 配置文件repmgr.conf

从主节点（在114机器上执行）复制一个repmgr.conf文件到从节点。

```
scp repmgr.conf uxdb@192.71.0.115:/home/uxdb/uxdbinstall/dbsql/bin/
```

修改repmgr.conf，与从节点匹配（conninfo中是本机IP）。

```
node_id=2  
node_name='node2'  
conninfo='host=192.71.0.115 user=repmgr dbname=repmgr connect_timeout=2'
```

2. 克隆数据库集群

在备节点上执行命令从主节点克隆数据库目录文件，IP是主节点的地址。

```
repmgr -h 192.71.0.114 -U repmgr -d repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf  
standby clone
```

3. 设置UXDATA环境变量

进行如下设置。

```
sudo vim /etc/profile  
export UXDATA=/home/uxdb/uxdbinstall/dbsql/bin/datarep/  
source /etc/profile
```

4. 设置归档日志路径

在uxdb目录下创建archive目录，用于归档日志。

```
mkdir /home/uxdb/archive
```

5. 注册备节点

克隆完成后可以注册备节点了。

先启动备节点的数据库。

```
./ux_ctl -D datarep/ start
```

注册备节点。

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby register
```

4.6. 启动守护进程

//在主节点（192.71.0.114）上启动守护进程。

```
repmgr daemon start -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

//在备节点（192.71.0.115）上启动守护进程。

```
repmgr daemon start -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

4.7. 查看集群信息

repmgr高可用集群已搭建好，可以查看如下信息。

1. 查看集群注册节点信息，可以看到两节点一主一备在正常运行。

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf cluster show
```

```
[uxdb@localhost bin]$ repmgr cluster show -f repmgr.conf
DEBUG: connecting to: "user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.114 fallback_application_name=repmgr"
DEBUG: connecting to: "user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.114 fallback_application_name=repmgr"
DEBUG: connecting to: "user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.115 fallback_application_name=repmgr"
ID | Name | Role | Status | Upstream | Location | Priority | Replication lag | Last replayed LSN
---+---+---+---+---+---+---+---+---
1 | node1 | primary | * running | | default | 100 | n/a | none
2 | node2 | standby | running | node1 | default | 100 | 0 bytes | 0/11008A80
```

2. 在主节点上执行命令ip addr，可以看到虚拟IP已绑定成功。下图中ens33下的 192.71.0.190 就是虚拟IP。

```
[uxdb@localhost bin]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:50:56:2e:dd:f4 brd ff:ff:ff:ff:ff:ff
    inet 192.71.0.114/23 brd 192.71.1.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet 192.71.0.190/32 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::1c0f:a5a5:29c1:7c90/64 scope link
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN qlen 1000
    link/ether 52:54:00:fe:41:90 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN qlen 1000
    link/ether 52:54:00:fe:41:90 brd ff:ff:ff:ff:ff:ff
```

3. 可以在其它虚拟机上（uxdb版本一致）上通过虚拟IP连接到数据库。

```
./uxsql -h 192.71.0.190 -d uxdb
```

4.8. 模式切换说明

配置主节点的数据目录下的uxsinodb.conf文件，增加两个配置就可以将异步改成同步。数据库节点的节点名，由repmgr.conf的node_name定义。

```
synchronous_standby_names = 'node2'
```

```
synchronous_commit = on
```

注意

在配置同步或者异步之后，不建议在已经多次进行主备切换后，在当前环境直接通过修改配置文件修改同步或者异步，因为多次准备切换后有可能破坏了集群的配置，所以应该重新部署环境并配置各项参数，具体内容请参见[第 4 章 repmgr部署](#)

注意

如果使用场景为配置同步之后，又想多次的主备切换测试，参数 `synchronous_standby_names` 应指定所有节点信息，且每个节点都需要配置，如下所示。

```
synchronous_standby_names = 'node1,node2'
```

• 同步模式切换异步模式

配置同步模式多次切换完成之后，如果想要在该环境上改成异步流复制模式，步骤如下：

1. 停止所有节点守护进程及所有节点实例（先主后备）。
2. 修改所有节点的 `repmgr.conf` 文件，将 `stream_replication_mode='sync'` 修改为 `stream_replication_mode='async'`。
3. 修改配置文件 `repmgr.conf`，屏蔽流复制节点相关的配置参数，如下所示。

```
#synchronous_standby_names = 'node1,node2'
#synchronous_commit = on
```

4. 启动主节点、备节点数据库实例，请参见[第 5.11 节 “启动和停止守护进程”](#)。
5. 启动主节点、备节点守护进程，请参见[第 5.11 节 “启动和停止守护进程”](#)。

• 异步模式切换同步模式

配置异步模式多次切换完成之后，如果想要在该环境上改成同步流复制模式，步骤如下：

1. 停止所有节点守护进程及所有节点实例（先主后备）。
2. 修改所有节点的 `repmgr.conf` 文件，将 `stream_replication_mode='async'` 修改为 `stream_replication_mode='sync'`。
3. 修改配置文件 `repmgr.conf`，放开流复制节点相关的配置参数，如下所示。

```
synchronous_standby_names = 'node1,node2'
synchronous_commit = on
```

4. 启动主节点、备节点数据库实例，请参见[第 5.11 节 “启动和停止守护进程”](#)。
5. 启动主节点、备节点守护进程，请参见[第 5.11 节 “启动和停止守护进程”](#)。

4.9. 配置多个备节点

重复备节点的步骤可以搭建一主多备集群。

1. 在新的机器上配置与所有集群中所有机器的ssh互信、配置网络、sudo免密码。
2. 安装同样版本的数据库。
3. 配置文件repmgr.conf。
4. 克隆数据库集群。
5. 注册备节点。

第 5 章 repmgr使用命令

repmgr有三个默认文件路径，当repmgr.conf文件在任意一个路径下时，下面的repmgr命令可以省略掉-f repmgr.conf参数。

1. /home/uxdb/uxdbinstall/dbsql/conf/repmgr.conf
2. ./repmgr.conf
3. /etc/repmgr.conf

5.1. 显示节点信息和复制状态

repmgr -f repmgr.conf node status

```
[uxdb@localhost bin]$ repmgr -f repmgr.conf node status
Node "node1":
  UXsinoDB version: 2.1.1.4
  Total data size: 67 MB
  Conninfo: host=192.71.0.114 port=5432 user=repmgr dbname=repmgr connect_timeout=2
  Role: primary
  WAL archiving: enabled
  Archive command: cp %p /home/uxdb/archive/%f
  WALs pending archiving: 0 pending files
  Replication connections: 0 (of maximal 10)
  Replication slots: 0 physical (of maximal 10; 0 missing)
  Replication lag: n/a
```

5.2. 显示集群注册节点信息

repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf cluster show

```
[uxdb@localhost bin]$ repmgr -f repmgr.conf cluster show
ID | Name | Role | Status | Upstream | Location | Priority | Replication lag | Last replayed LSN
-----+-----+-----+-----+-----+-----+-----+-----+-----
1 | node1 | primary | * running | node1 | default | 100 | n/a | none
2 | node2 | standby | running | node1 | default | 60 | 0 bytes | 0/46042BE8
3 | node3 | standby | ! running as primary | node1 | default | 90 | 167 kB | 0/460000A0

WARNING: following issues were detected
- node "node3" (ID: 3) is registered as standby but running as primary
```

5.3. 查看守护进程状态

repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf daemon status

```
[uxdb@localhost bin]$ repmgr -f repmgr.conf daemon status
ID | Name | Role | Priority | Status | repmgrd | PID | Paused? | Upstream last seen
-----+-----+-----+-----+-----+-----+-----+-----+-----
2 | node2 | primary | 60 | running | not running | n/a | n/a | n/a
3 | node3 | standby | 90 | running | not running | n/a | n/a | n/a
```

5.4. 注册主节点

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf primary register --force
```

```
[uxdb@localhost bin]$ repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf primary register --force
INFO: connecting to primary database...
DEBUG: connecting to: "user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.114 fallback_application_name=repmgr"
NOTICE: attempting to install extension "repmgr"
NOTICE: "repmgr" extension successfully installed
[sudo] uxdb 的密码:
NOTICE: primary node record (id: 1) registered
[uxdb@localhost bin]$
```

5.5. 注销主节点

可以在任何节点上运行，要注销的节点ID为--node-id。

```
repmgr primary unregister -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf --node-id=2
```

```
[uxdb@localhost bin]$ repmgr primary unregister -f repmgr.conf --node-id=2
INFO: node node2 (ID: 2) was successfully unregistered
```

5.6. 备节点克隆

```
repmgr -h 192.71.0.114 -U repmgr -d repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby clone
```

```
[uxdb@uxdev bin]$ repmgr -h 192.71.0.114 -U repmgr -d repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby clone
NOTICE: destination directory "/home/uxdb/uxdbinstall/dbsql/bin/dataprep" provided
INFO: connecting to source node
DETAIL: connection string is: host=192.71.0.114 user=repmgr dbname=repmgr
DETAIL: current installation size is 67 MB
INFO: creating directory "/home/uxdb/uxdbinstall/dbsql/bin/dataprep"...
NOTICE: starting backup (using ux_basebackup)...
HINT: this may take some time; consider using the -c/--fast-checkpoint option
INFO: executing:
    /home/uxdb/uxdbinstall/dbsql/bin/ux_basebackup -l "repmgr base backup" -D /home/uxdb/uxdbinstall/dbsql/bin/dataprep -h 192.71.0.114 -p 5432 -U repmgr -X stream
NOTICE: standby clone (using ux_basebackup) complete
NOTICE: you can now start your UXsnoDB server
HINT: for example: /home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/dataprep -l logfile start
HINT: after starting the server, you need to register this standby with "repmgr standby register"
```

5.7. 注册备节点

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby register
```

```
[uxdb@localhost bin]$ repmgr -f ./repmgr.conf standby register
INFO: connecting to local node "node2" (ID: 2)
INFO: connecting to primary database
WARNING: --upstream-node-id not supplied, assuming upstream node is primary (node ID 3)
INFO: standby registration complete
NOTICE: standby node "node2" (id: 2) successfully registered
```

5.8. 注销备节点

//在本节点上注销

```
repmgr standby unregister -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

//在其它节点上注销

```
repmgr standby unregister -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf --node-id=3
```

5.9. rejoin

把一个节点以备节点的身份重新加入集群。

需要先关闭当前节点的数据库服务,在节点（挂掉且已有备节点升为主）执行，host的IP是已提升为主节点的IP。

```
repmgr -d 'host=192.71.0.115 user=repmgr dbname=repmgr connect_timeout=2' node rejoin --
force-rewind -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

5.10. 手动切换主从节点

在需要升为主的备节点上执行命令：

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby switchover
```

```
NOTICE: standby switchover has completed successfully
[uxdb@localhost bin]$
```

注意

一主多备集群，在切换主节点的时候要添加--siblings-follow参数，将其他备节点跟随到新的主节点上。执行如下命令。

```
repmgr -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf standby
switchover --siblings-follow
```

5.11. 启动和停止守护进程

1. 在repmgr.conf文件里配置repmgrd_service_start_command和repmgrd_service_stop_command的命令。

```
repmgrd_service_start_command = 'repmgrd -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf -d'
repmgrd_service_stop_command = 'kill `cat /tmp/repmgrd.pid`'
```

```
# Used by "repmgr daemon (start|stop)" to control repmgrd
#
repmgrd_service_start_command = 'repmgrd -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf -d'
repmgrd_service_stop_command = 'kill `cat /tmp/repmgrd.pid`'
```

2. 启动守护进程。

```
repmgr daemon start -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

3. 停止守护进程。

```
repmgr daemon stop -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf
```

```
[uxdb@localhost bin]$ repmgr daemon start -f repmgr.conf
NOTICE: executing: "repmgrd -f /home/uxdb/uxdbinstall/dbsql/bin/repmgr.conf -d"
[2020-10-16 16:18:01] [NOTICE] redirecting logging output to "/home/uxdb/uxdbinstall/dbsql/bin/repmgrd.log"
NOTICE: repmgrd was successfully started
```

第 6 章 高可用场景示例

主要展示了高可用场景在部分异常情况下的处理方式。

6.1. 一主一备

主要是针对一主一备在异步流复制的情况下的异常处理。

假设主库地址192.71.0.114，备库地址192.71.0.115，虚拟IP为192.71.0.190，主备库关系和状态如图所示：

```
[uxdb@localhost bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	primary	* running		default	100	n/a	none
2	node2	standby	running	node1	default	100	0 bytes	0/F002720

```
[uxdb@localhost bin]$ repmgr daemon status
```

ID	Name	Role	Priority	Status	repmgrd	PID	Paused?	Upstream last seen
1	node1	primary	100	running	not running	n/a	n/a	n/a
2	node2	standby	100	running	not running	n/a	n/a	n/a

6.1.1. 主库异常

模拟主库异常关闭，例如执行kill -9（主库PID）。

查看主库进程ID，并关闭进程。

```
ps -ef|grep uxdb
Kill -9 2379
```

```
[uxdb@localhost bin]$ ps -ef|grep uxdb
uxdb      2238      1 0 Nov21 ?        00:00:00 /usr/bin/gnome-keyring-daemon --daemonize --login
uxdb      2262    2229 0 Nov21 ?        00:00:01 /usr/libexec/gnome-session-binary --session gnome-classic
uxdb      2271      1 0 Nov21 ?        00:00:00 dbus-launch --sh-syntax --exit-with-session
uxdb      2272      1 0 Nov21 ?        00:00:00 /usr/bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
uxdb      2306      1 0 Nov21 ?        00:00:00 /usr/libexec/lssettings-daemon
uxdb      2310      1 0 Nov21 ?        00:00:00 /usr/libexec/gvfsd
uxdb      2315      1 0 Nov21 ?        00:00:00 /usr/libexec/gvfsd-fuse /run/user/1000/gvfs -f -o big_writes
uxdb      2379      1 0 09:49 ?        00:00:00 /home/uxdb/uxdbinstall/dbsql/bin/uxdb -D /home/uxdb/uxdbinstall/dbsql/
bin/datarep
uxdb      2380    2379 0 09:49 ?        00:00:00 uxdb: logger
uxdb      2382    2379 0 09:49 ?        00:00:00 uxdb: checkpointer
uxdb      2383    2379 0 09:49 ?        00:00:00 uxdb: background writer
uxdb      2384    2379 0 09:49 ?        00:00:00 uxdb: stats collector
uxdb      2397    2262 0 Nov21 ?        00:00:03 /usr/bin/ssh-agent /bin/sh -c exec -l /bin/bash -c "env GNOME_SHELL_SE
SION_MODE=classic gnome-session --session gnome-classic"
uxdb      2419      1 0 Nov21 ?        00:00:00 /usr/libexec/at-spi-bus-launcher
uxdb      2424    2419 0 Nov21 ?        00:00:00 /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/accessi
bility.conf --nofork --print-address 3
```

主库异常关闭后，备库自动提升为主库，虚拟IP漂移绑定到新主库上。

```
[uxdb@uxdev bin]$ repmgr cluster show
ID | Name | Role | Status | Upstream | Location | Priority | Replication lag | Last replayed LSN
---+---+---+---+---+---+---+---+---
1 | node1 | standby | running | node2 | default | 100 | 0 bytes | 0/C001868
2 | node2 | primary | * running | | default | 100 | n/a | none
[uxdb@uxdev bin]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:89:44 brd ff:ff:ff:ff:ff:ff
    inet 192.71.0.115/23 brd 192.71.1.255 scope global noprefixroute ens33
        valid lft forever preferred_lft forever
    inet 192.71.0.190/32 scope global ens33
        valid lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:8944/64 scope link
        valid lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:10:8f:53 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:10:8f:53 brd ff:ff:ff:ff:ff:ff
[uxdb@uxdev bin]$
```

原主库自动同步新主库数据，成为新主库的新备库。

```
NOTICE: ux_rewind execution required for this node to attach to rejoin target node 2
DETAIL: rejoin target server's timeline 52 forked off current database system timeline 51 before current recovery point 0/2B000028
NOTICE: executing ux_rewind
DETAIL: ux_rewind command is "/home/uxdb/uxdbinstall/dbsql/bin/ux_rewind -D '/home/uxdb/uxdbinstall/dbsql/bin/dataprep' --source-server='host=192.71.0.115 port=5432 user=repmgr dbname=repmgr connect_timeout=2'"
ux_rewind: servers diverged at WAL location 0/2A07F7B8 on timeline 51
ux_rewind: rewinding from last common checkpoint at 0/2A070A08 on timeline 51
ux_rewind: Done!
NOTICE: 0 files copied to /home/uxdb/uxdbinstall/dbsql/bin/dataprep
NOTICE: setting node 1's upstream to node 2
WARNING: unable to ping "host=192.71.0.114 port=5432 user=repmgr dbname=repmgr connect_timeout=2"
DETAIL: UXSQLping() returned "UXSQLPING_NO_RESPONSE"
NOTICE: starting server using "/home/uxdb/uxdbinstall/dbsql/bin/ux_ctl -D /home/uxdb/uxdbinstall/dbsql/bin/dataprep -l logfile start"
NOTICE: NODE REJOIN successful
DETAIL: node 1 is now attached to node 2
[2021-11-24 14:59:10] [NOTICE] reconnected to node after 61 seconds, node is now a standby, switching to standby monitoring
[2021-11-24 14:59:10] [NOTICE] former primary has been restored as standby after 61 seconds, updating node record and resuming monitoring
INFO: set_repmgrd_pid(): provided pidfile is /tmp/repmgrd.pid
[2021-11-24 15:04:11] [INFO] node "node1" (node ID: 1) monitoring upstream node "node2" (node ID: 2) in normal state
[2021-11-24 15:04:11] [DETAIL] last monitoring statistics update was 2 seconds ago
5908,1 Bot
```

6.1.2. 备库异常

模拟备库stop，执行如下命令：

```
./ux_ctl -D dataprep stop
```

备库库自动同步主库数据，成为主库的备库，恢复原有的主备关系。

```
[uxdb@uxdev bin]$ repmgr cluster show
ERROR: connection to database failed
DETAIL:
could not connect to server: Connection refused
        Is the server running on host "192.71.0.115" and accepting
        TCP/IP connections on port 5432?
DETAIL: attempted to connect using:
        user=repmgr connect_timeout=2 dbname=repmgr host=192.71.0.115 port=5432 fallback_application_name=repmgr
[uxdb@uxdev bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	primary	* running		default	100	n/a	none
2	node2	standby	running	node1	default	100	12 kB	0/11002EF8

6.2. 一主多备

假设主库地址192.71.0.114，备库1地址192.71.0.115，备库2地址192.71.0.116，虚拟IP为192.71.0.190，主备库关系和状态如图所示：

```
[uxdb@localhost bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	primary	* running		default	100	n/a	none
2	node2	standby	running	node1	default	100	0 bytes	0/13003080
3	node3	standby	running	node1	default	100	0 bytes	0/13003080

6.2.1. 主库异常

1. 模拟主库所在的服务器断网，执行如下命令断网：

```
service network stop
```

2. 主库断网后，备库自动提升为主库，虚拟IP漂移绑定到新主库上。

```
[uxdb@uxdev bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	standby	running	node2	default	100	0 bytes	0/E002218
2	node2	primary	* running		default	100	n/a	none
3	node3	standby	running	node2	default	100	0 bytes	0/E002218

```
[uxdb@uxdev bin]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1d:89:44 brd ff:ff:ff:ff:ff:ff
    inet 192.71.0.115/23 brd 192.71.1.255 scope global noprefixroute ens33
        valid lft forever preferred_lft forever
    inet 192.71.0.190/32 scope global ens33
        valid lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe1d:8944/64 scope link
        valid lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:10:8f:53 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:10:8f:53 brd ff:ff:ff:ff:ff:ff
```

3. 原主库故障恢复。

```
service network start
```

4. 原主库自动同步新主库数据，成为新主库的新备库。

```
[uxdb@uxdev bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	standby	running	node2	default	100	0 bytes	0/140033C8
2	node2	primary	* running		default	100	n/a	none
3	node3	standby	running	node1	default	100	0 bytes	0/140033C8

6.2.2. 备库异常

查看主备关系及状态信息。

repmgr cluster show

```
[uxdb@uxdev bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	standby	running	node2	default	100	0 bytes	0/140033C8
2	node2	primary	* running		default	100	n/a	none
3	node3	standby	running	node1	default	100	0 bytes	0/140033C8

模拟node3节点的备库异常关闭，例如执行kill -9 备库（PID）。

查看备库进程ID，并关闭进程。

ps -ef|grep uxdb
Kill -9 4241

```
uxdb 3417 2259 0 21:00 ? 00:00:00 /usr/libexec/gvfsd-dnssd --spawner :1.4 /org/gtk/gvfs/exec
uxdb 4241 1 0 21:45 ? 00:00:00 /home/uxdb/uxdbinstall/dbsql/bin/uxdb -D datarep
uxdb 4242 4241 0 21:45 ? 00:00:00 uxdb: logger
uxdb 4243 4241 0 21:45 ? 00:00:00 uxdb: startup recovering 0000000A00000000000000014
uxdb 4244 4241 0 21:45 ? 00:00:00 uxdb: checkpointer
uxdb 4245 4241 0 21:45 ? 00:00:00 uxdb: background writer
uxdb 4246 4241 0 21:45 ? 00:00:00 uxdb: stats collector
uxdb 4303 4241 0 21:47 ? 00:00:00 uxdb: walreceiver streaming 0/14003400
```

备库自动同步主库数据，成为主库的备库，恢复原有的主备关系。

```
[uxdb@uxdev bin]$ repmgr cluster show
```

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	standby	running	node2	default	100	0 bytes	0/140034E8
2	node2	primary	* running		default	100	n/a	none
3	node3	standby	? unreachable	node1	default	100	320 MB	0/0

node3节点异常

WARNING: following issues were detected

- unable to connect to node "node3" (ID: 3)
- node "node3" (ID: 3) is registered as an active standby but is unreachable

[uxdb@uxdev bin]\$ repmgr cluster show 稍等片刻后查看状态，node3自动同步，并且正常运行。

ID	Name	Role	Status	Upstream	Location	Priority	Replication lag	Last replayed LSN
1	node1	standby	running	node2	default	100	0 bytes	0/140034E8
2	node2	primary	* running		default	100	n/a	none
3	node3	standby	running	node1	default	100	13 kB	0/140034E8