РЕД СЛЁРМ

+

Southbridge

База данных PostgreSQL: кластеризация на базе Pacemaker

Вадим Исаканов, инженер Southbridge

PostgreSQL. Для чего?

PostgreSQL - OLTP база данных.

Реляционная транзакционная база данных реального времени.

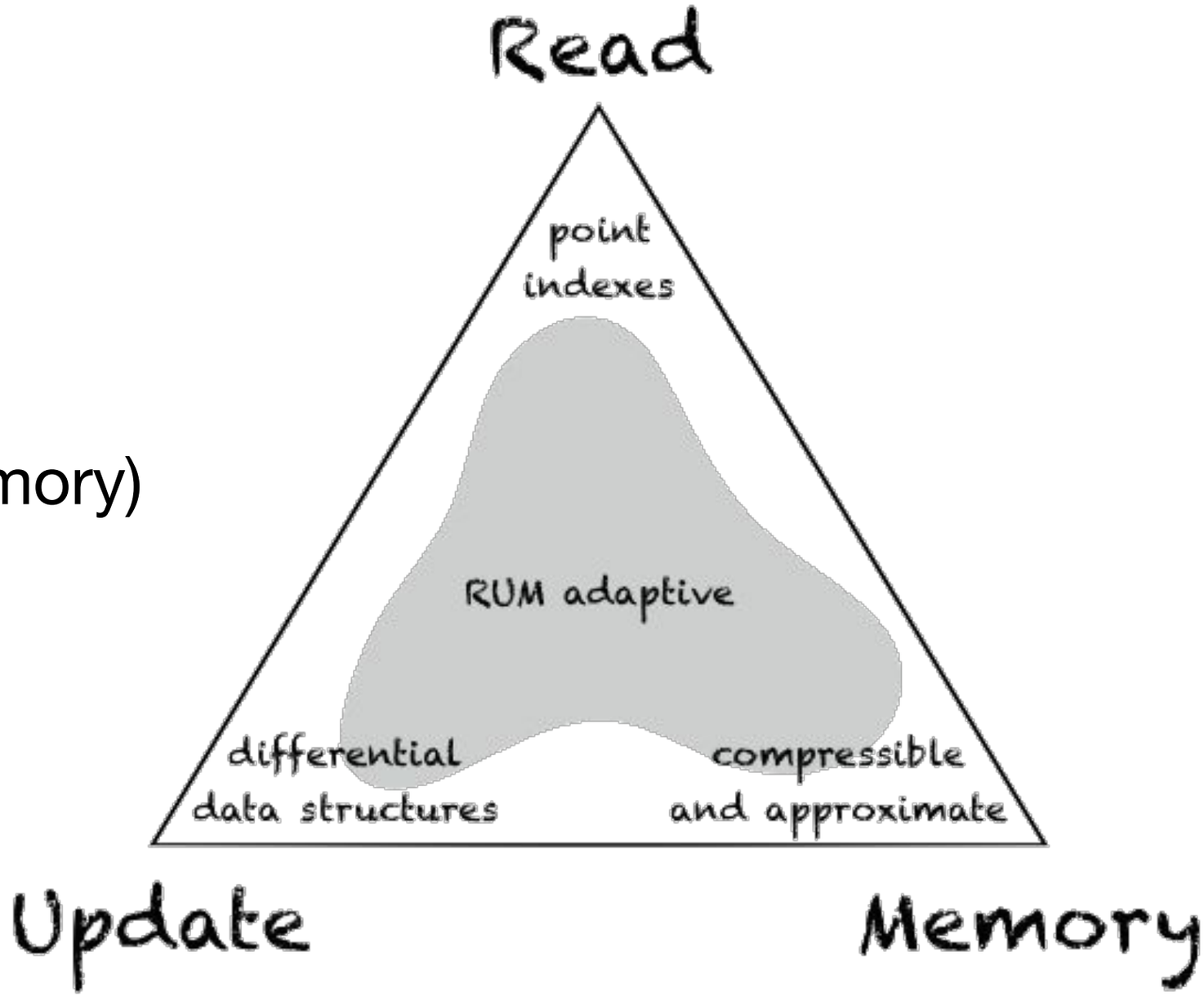
«Часто пишем» и «часто читаем» небольшими порциями данных.

https://www.postgresql.org/docs/manuals/
https://postgrespro.ru/docs
https://uwdc.ru/lectures/backend/postgresql-in-your-eyes

PostgreSQL. Тип нагрузки.

RUM теорема:

R (read) U (update) M (memory) overheads



PostgreSQL. Для чего?

PostgreSQL - OLTP база данных.

Реляционная транзакционная база данных реального времени.

«Часто пишем» и «часто читаем» небольшими порциями данных.

https://www.postgresql.org/docs/manuals/
https://postgrespro.ru/docs
https://uwdc.ru/lectures/backend/postgresql-in-your-eyes

Pacemaker

- Opensource **ПО для кластеризации**.
- Не привязано к конкретному ПО.
- Использует **Corosync** или **Heartbeat.**
- Оперирует понятием **ресурсов** и **ресурс-агентов.**
- Имеет большую гибкость, мы рассматриваем **управление нодами с Linux & PostgreSQL.**

Почитать: https://habr.com/post/107837/

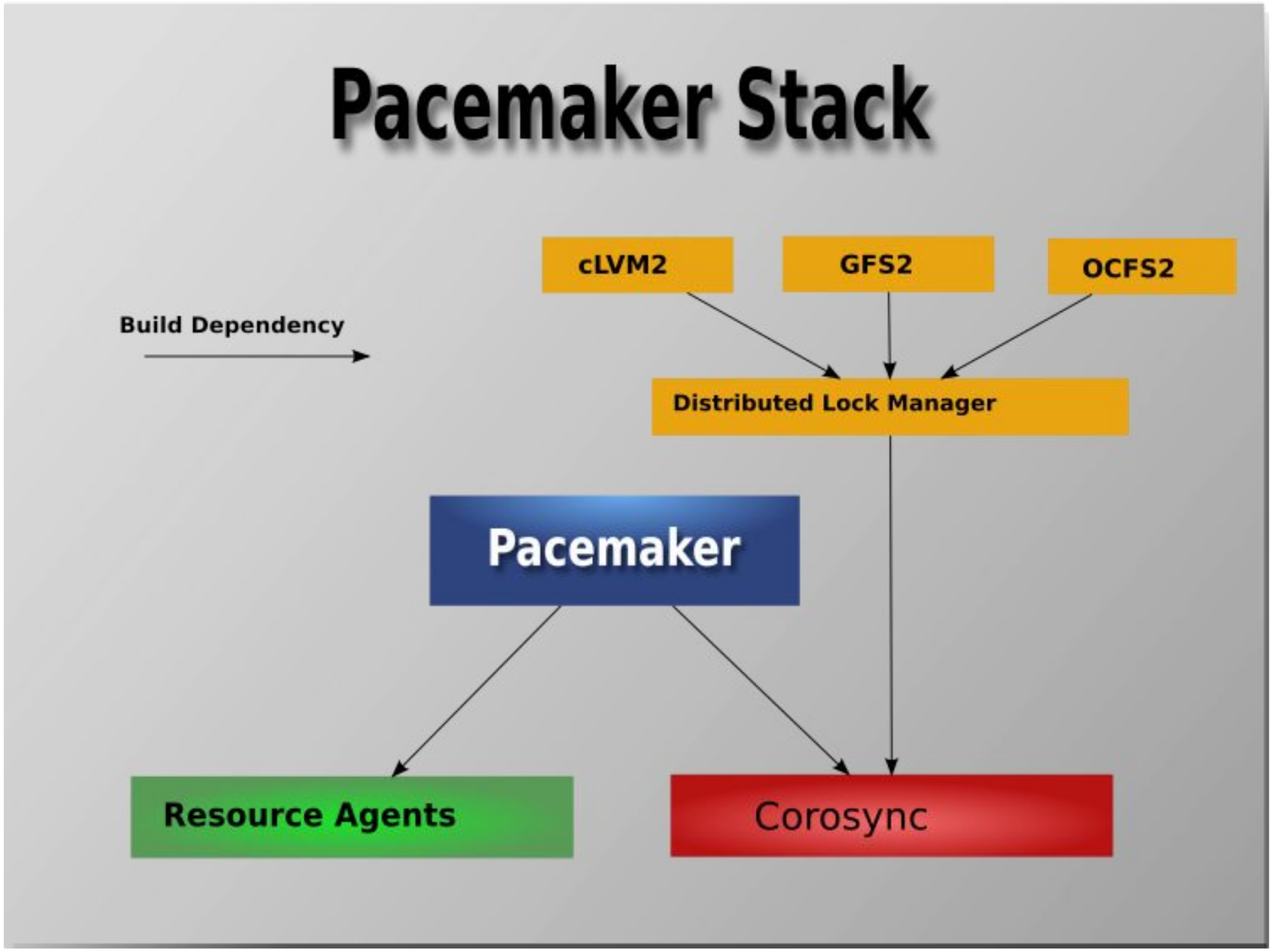
Pacemaker

ноды

corosync

pcs

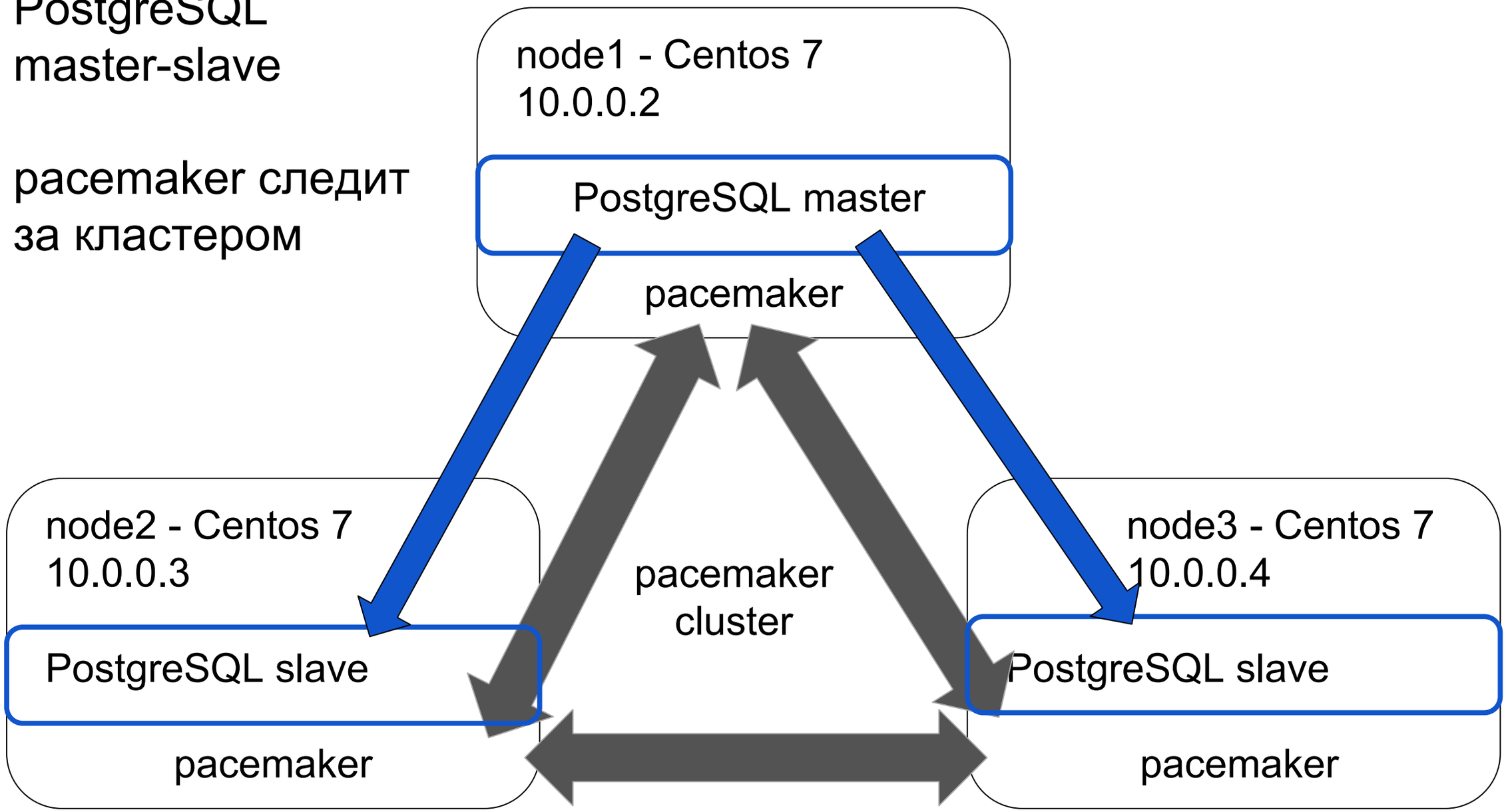
ресурс-агенты



Pacemaker + PostgreSQL

PostgreSQL
master-slave

pacemaker следит
за кластером



Установка

Используем ноды:

- Centos 7.4
- по 1-му сетевому интерфейсу на каждую
- сетевые интерфейсы в одном L2 сегменте
- нет SELINUX
- iptables остановлен

Ноды:

node1.pgcluster	**172.20.5.2**/24	Начальный мастер
node2.pgcluster	**172.20.5.3**/24	Реплика
node3.pgcluster	**172.20.5.4**/24	Реплика

Установка, шаг 0

/etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
172.20.5.2 node1.pgcluster node1
172.20.5.3 node2.pgcluster node2
172.20.5.4 node3.pgcluster node3

Установка, шаг 1

Установка ПО

[all]

```
setenforce 0
sed -i -r 's|(SELINUX=).*|\1disabled|g' /etc/selinux/config

systemctl disable --now iptables.service
```

Без SELINUX & iptables (это долго).

[all] - выполняем на всех узлах
[master] - на pgsql master
[slave] - на pgsql slave
[node1..3] - на одной из хост-нод

Установка, шаг 2

Установка ПО

[all]

yum install 'https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86_64/pgdg-centos10-10-2.noarch.rpm'

yum install postgresql10-server postgresql10-contrib pacemaker pcs resource-agents wget

Установка, шаг 3

Установка ресурс-агента pgsql

[all]

```
mkdir -p /usr/lib/ocf/resource.d/southbridge
wget 'https://raw.githubusercontent.com/ClusterLabs/
  resource-agents/master/heartbeat/pgsql' -O
/usr/lib/ocf/resource.d/southbridge/pgsql

chmod 0755 /usr/lib/ocf/resource.d/southbridge/pgsql
```

Установка, шаг 4

Настройка master. Копируем скрипт настройки ресурсов.

[master]

```
mkdir -p /srv/southbridge/scripts

wget 'https://gitlab.slurm.io/red/slurm/raw/master/practice/
  6.pgsql/pgsql.pcs.sample' -O /srv/southbridge/scripts/pgsql.pcs

chmod 0755 /srv/southbridge/scripts/pgsql.pcs
```

Инициализация кластера

[all]

```
echo PASSWORD | passwd --stdin hacluster

systemctl enable pcsd.service
systemctl start pcsd.service
```

[master]

```
pcs cluster auth node1.pgcluster node2.pgcluster node3.pgcluster -u hacluster
-p PASSWORD --force
pcs cluster setup --force --name PGCLUSTER node1.pgcluster
node2.pgcluster node3.pgcluster
pcs cluster start --all
```

Статус кластера

pcs quorum status
pcs cluster status
pcs status

```
[root@node1 ~]# pcs status
Cluster name: PGCLUSTER
Stack: corosync
Current DC: node2.pgcluster (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum
Last updated: Tue Oct 16 19:55:21 2018
Last change: Tue Oct 16 09:07:37 2018 by root via crm_resource on node2.pgcluster

3 nodes configured
5 resources configured

Online: [ node1.pgcluster node2.pgcluster node3.pgcluster ]

Full list of resources:

 Master/Slave Set: PG-MASTER [PGSQL]
     Masters: [ node1.pgcluster ]
     Slaves: [ node2.pgcluster node3.pgcluster ]
 Resource Group: MASTER-GROUP
     VADDR-MAIN (ocf::heartbeat:IPaddr2):       Started node1.pgcluster
     VADDR-REPL (ocf::heartbeat:IPaddr2):       Started node1.pgcluster

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Инициализируем базу

[master]

sudo -iu postgres /usr/pgsql-10/bin/initdb -D /var/lib/pgsql/10/data

vi /var/lib/pgsql/10/data/postgresql.conf

```
listen_addresses              = '*'
port                        = 5432
wal_level                     = replica
wal_log_hints                 = on
max_wal_senders                 = 10
wal_keep_segments               = 32
hot_standby                   = on
wal_receiver_status_interval    = 2
restart_after_crash           = false
```

vi /var/lib/pgsql/10/data/pg_hba.conf

```
host    replication     replicator      172.20.5.0/24           md5
```

Настройка PGSQL master

Запускаем PostgreSQL, создаем пользователя для репликации

[master]

```
sudo -iu postgres
/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data start

psql -c "CREATE ROLE replicator WITH LOGIN REPLICATION CONNECTION
  LIMIT 10 PASSWORD 'REPLICATOR_PASSWORD';"
```

Настройка PGSQL slave

[all]

```
sudo -iu postgres
echo '*:*:replication:replicator:REPLICATOR_PASSWORD' >> .pgpass
chmod 0600 .pgpass
mkdir -p 10/pg_archive 10/tmp
```

[slaves]

```
sudo -iu postgres
rm -rf /var/lib/pgsql/10/data
mkdir -m 0700 /var/lib/pgsql/10/data
pg_basebackup --host=172.20.5.2 --username=replicator --pgdata=/var/lib/pgsql/10/data
--status-interval=2 --progress

cat > 10/data/recovery.conf <<'EOF'
standby_mode = 'on'
primary_conninfo = 'host=172.20.5.2 port=5432 user=replicator'
restore_command = 'cp /var/lib/pgsql/10/pg_archive/%f %p'
EOF

/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data start
```

Проверяем PGSQL Master-Slave

[master]

```
sudo -iu postgres psql -c 'SELECT client_addr, state, sent_lsn, write_lsn,
  flush_lsn, replay_lsn FROM pg_stat_replication;'
```

Если всё хорошо -- останавливаем реплики и мастер (именно в этом порядке).

[slaves]
[master]

```
sudo -iu postgres /usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data stop
```

Настройка и запуск кластера

Правим ip-адреса на свои, запускаем скрипт настройки кластера

[master]

/srv/southbridge/scripts/pgsql.pcs

Проверяем статус:

crm_mon -Afr ->

```
Stack: corosync
Current DC: node2.pgcluster (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum
Last updated: Thu Aug 30 10:55:04 2018
Last change: Thu Aug 30 10:40:34 2018 by root via crm_attribute on node1.pgcluster

3 nodes configured
5 resources configured

Online: [ node1.pgcluster node2.pgcluster node3.pgcluster ]

Full list of resources:

 Master/Slave Set: PG-MASTER [PGSQL]
     Masters: [ node1.pgcluster ]
     Slaves: [ node2.pgcluster node3.pgcluster ]
 Resource Group: MASTER-GROUP
     VADDR-MAIN (ocf::heartbeat:IPaddr2):       Started node1.pgcluster
     VADDR-REPL (ocf::heartbeat:IPaddr2):       Started node1.pgcluster

Node Attributes:
* Node node1.pgcluster:
    + PGSQL-data-status               : LATEST
    + PGSQL-master-baseline           : 000000000040000D0
    + PGSQL-receiver-status           : normal (master)
    + PGSQL-status                    : PRI
    + master-PGSQL                    : 1000
* Node node2.pgcluster:
    + PGSQL-data-status               : STREAMING|ASYNC
    + PGSQL-receiver-status           : normal
    + PGSQL-status                    : HS:async
    + master-PGSQL                    : -INFINITY
* Node node3.pgcluster:
    + PGSQL-data-status               : STREAMING|ASYNC
    + PGSQL-receiver-status           : normal
    + PGSQL-status                    : HS:async
    + master-PGSQL                    : -INFINITY

Migration Summary:
* Node node1.pgcluster:
* Node node2.pgcluster:
* Node node3.pgcluster:
```

Тесты. writer

[master]
sudo -iu postgres

psql << 'EOF'
CREATE ROLE test LOGIN PASSWORD 'testpassword';
CREATE DATABASE test OWNER test;
\c test
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA public GRANT ALL ON TABLES TO test;
CREATE TABLE test (time timestamp with time zone);
EOF

[all]
sudo -iu postgres
echo "host   test   test   0.0.0.0/0   md5" >> 10/data/pg_hba.conf
/usr/pgsql-10/bin/pg_ctl -D /var/lib/pgsql/10/data reload

[node2]
sudo -iu postgres
echo '**172.20.5.6**:5432:test:test:testpassword' >> .pgpass
setsid bash -c \
"while true; do psql -h **172.20.5.6** -U test -qc \"INSERT INTO test VALUES (current_timestamp)\" test 2>/dev/null;
sleep 1; done"

Тесты. writer

Проверяем:

[master]

psql -Upostgres

SELECT client_addr, state, sent_lsn, write_lsn, flush_lsn, replay_lsn FROM pg_stat_replication;

Увидим, что LSN меняются (т. е. данные передаются на слейвы).

Тесты. PGSQL Slave off

ssh node3
reboot

crm_mon -Afr ->

```
Online: [ node1.pgcluster node2.pgcluster ]
OFFLINE: [ node3.pgcluster ] <--

Full list of resources:

 Master/Slave Set: PG-MASTER [PGSQL]
     Masters: [ node1.pgcluster ]
     Slaves: [ node2.pgcluster ]
 --> Stopped: [ node3.pgcluster ]
 Resource Group: MASTER-GROUP
     VADDR-MAIN (ocf::heartbeat:IPaddr2):       Started node1.pgcluster
     VADDR-REPL (ocf::heartbeat:IPaddr2):       Started node1.pgcluster

Node Attributes:
* Node node1.pgcluster:
    + PGSQL-data-status               : LATEST
    + PGSQL-master-baseline           : 000000000400000D0
    + PGSQL-receiver-status           : normal (master)
    + PGSQL-status                    : PRI
    + master-PGSQL                    : 1000
* Node node2.pgcluster:
    + PGSQL-data-status               : STREAMING|ASYNC
    + PGSQL-receiver-status           : normal
    + PGSQL-status                    : HS:async
    + master-PGSQL                    : -INFINITY
```

Тесты. Смена PGSQL Master

Запретим PGSQL Master на node3:

```
pcs resource ban PG-MASTER node3.pgcluster --master
pcs constraint show
```

Отключим node1:

```
ssh node1
reboot
```

crm_mon -AFR ->

```
Node Attributes:
* Node node2.pgcluster:
    + PGSQL-data-status                : LATEST
    + PGSQL-master-baseline            : 0000000004104F10
    + PGSQL-receiver-status            : normal (master)
    + PGSQL-status                     : PRI
    + master-PGSQL                     : 1000
* Node node3.pgcluster:
    + PGSQL-data-status                : STREAMING|ASYNC
    + PGSQL-receiver-status            : normal
    + PGSQL-status                     : HS:async
    + master-PGSQL                     : -INFINITY
```

Тесты. node3 ban/clear

Запретим выполнение ресурса PGSQL на node:

pcs resource ban PGSQL node3.pgcluster

```
Node Attributes:
* Node node2.pgcluster:
    + PGSQL-data-status                : LATEST
    + PGSQL-master-baseline            : 0000000004104F10
    + PGSQL-receiver-status            : normal (master)
    + PGSQL-status                     : PRI
    + master-PGSQL                     : 1000
* Node node3.pgcluster:
    + PGSQL-data-status                : DISCONNECT
    + PGSQL-status                     : STOP
    + master-PGSQL                     : -INFINITY
```

Разбаним:

pcs resource clear PGSQL node3.pgcluster

Тесты. Остановка кластера PGSQL

Отключим node3:

ssh node3
reboot

```
Master/Slave Set: PG-MASTER [PGSQL]
     Stopped: [ node1.pgcluster node2.pgcluster node3.pgcluster ]
 Resource Group: MASTER-GROUP
     VADDR-MAIN (ocf::heartbeat:IPaddr2):       Stopped
     VADDR-REPL (ocf::heartbeat:IPaddr2):       Stopped

Node Attributes:
* Node node2.pgcluster:
    + PGSQL-data-status                 : LATEST
    + PGSQL-status                      : STOP
    + master-PGSQL                      : -INFINITY
```

После загрузки node3 вернется в кластер.

Тесты. Возврат node1/old master

У бывшего master **PGSQL-status : STOP**
exitreason='My data may be inconsistent. You have to remove /var/lib/pgsql/10/tmp/PGSQL.lock file to force start.

```
Node Attributes:
* Node node1.pgcluster:
    + PGSQL-data-status                : DISCONNECT
    + PGSQL-receiver-status            : normal
    + PGSQL-status                     : STOP
    + master-PGSQL                     : -INFINITY
* Node node2.pgcluster:
    + PGSQL-data-status                : STREAMING|ASYNC
    + PGSQL-receiver-status            : normal
    + PGSQL-status                     : HS:async
    + master-PGSQL                     : -INFINITY
* Node node3.pgcluster:
    + PGSQL-data-status                : LATEST
    + PGSQL-master-baseline            : 000000000DC7C9D8
    + PGSQL-receiver-status            : normal (master)
    + PGSQL-status                     : PRI
    + master-PGSQL                     : 1000
```

Тесты. Возврат node1/old master

Если уникальных данных на бывшем master нет - вернем его в работу.

pcs node maintenance node1.pgcluster

В Node Attributes появится строка + maintenance : on.

[node1]

```
sudo -iu postgres
rm -rf /var/lib/pgsql/10/data
mkdir -m 0700 /var/lib/pgsql/10/data
pg_basebackup --host=172.20.5.6 --username=replicator --pgdata=/var/lib/pgsql/10/data
  --status-interval=2 --progress
rm /var/lib/pgsql/10/tmp/PGSQL.lock

pcs node unmaintenance node1.pgcluster
pcs resource cleanup PGSQL --node node1.pgcluster
```

crm_mon покажет нам полностью здоровый кластер.
Удалим ограничение:

```
pcs resource clear PG-MASTER node3.pgcluster
```

Addendum

Установить режим обслуживания на все ноды можно командой

```
pcs node maintenence --all
```

целиком на кластер:

```
pcs property set maintenance-mode=true
```

Удалить все ресурсы кластера можно командами

```
pcs resource ban PGSQL node1.pgcluster
pcs resource ban PGSQL node3.pgcluster
pcs resource ban PGSQL node2.pgcluster
pcs resource delete MASTER-GROUP
pcs resource delete PGSQL
```

РЕД СЛЁРМ

+

Southbridge

Практика

slurm.io