

Стенд:

altserver1 с сетевым интерфейсом ens19 / ens20

altserver2 с сетевым интерфейсом ens19

altserver3 с сетевым интерфейсом ens19

### Шаги настройки:

#### 1. Переименовать машины

```
# hostnamectl set-hostname altserver{1,2,3}
```

#### 2. Настроить /etc/hosts

```
# cat >> /etc/hosts << EOF
<IP altserver1> altserver1
<IP altserver2> altserver2
<IP altserver3> altserver3
EOF
```

Проверка:

```
# ping -c 3 altserver1; ping -c 3 altserver2; ping -c 3 altserver3
```

#### 3. Установить пакеты на **все машины**

```
# apt-get install -y postgresql13 postgresql13-server postgresql13-contrib pacemaker pcs pcs-pcsd
fence-agents-virsh resource-agents-paf corosync resource-agents
```

#### 4. Выполнить установку пароля для пользователя hacluser на **всех трёх узлах**

```
# echo "hacluser" | passwd --stdin hacluser
```

#### 5. Запустить и добавить в автозагрузку службу pcsd на всех трёх узлах

```
# systemctl enable --now pcsd; sleep 5; systemctl status pcsd
```

#### 6. Настроить аутентификацию pcs на **altserver1**

```
# pcs host auth -u hacluser -p "hacluser" altserver1 altserver2 altserver3
```

Вывод:

```
altserver2: Authorized
altserver3: Authorized
altserver1: Authorized
```

#### 7. Создать кластер на **altserver1**

```
# pcs cluster setup sng-cluster altserver1 addr=$(grep altserver1 /etc/hosts | cut -d ' ' -f 1)
altserver2 addr=$(grep altserver2 /etc/hosts | cut -d ' ' -f 1) altserver3 addr=$(grep altserver3
/etc/hosts | cut -d ' ' -f 1) transport udpu
```

Вывод:

```
Destroying cluster on hosts: 'altserver1', 'altserver2', 'altserver3'...
altserver1: Successfully destroyed cluster
altserver2: Successfully destroyed cluster
altserver3: Successfully destroyed cluster
Requesting remove 'pcsd settings' from 'altserver1', 'altserver2', 'altserver3'
```

```
altserver1: successful removal of the file 'pcsd settings'
altserver2: successful removal of the file 'pcsd settings'
altserver3: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'altserver1', 'altserver2', 'altserver3'
altserver2: successful distribution of the file 'corosync authkey'
altserver2: successful distribution of the file 'pacemaker authkey'
altserver1: successful distribution of the file 'corosync authkey'
altserver1: successful distribution of the file 'pacemaker authkey'
altserver3: successful distribution of the file 'corosync authkey'
altserver3: successful distribution of the file 'pacemaker authkey'
Sending 'corosync.conf' to 'altserver1', 'altserver2', 'altserver3'
altserver2: successful distribution of the file 'corosync.conf'
altserver1: successful distribution of the file 'corosync.conf'
altserver3: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.
```

Проверка md5-суммы **corosync.conf** на **всех хостах**:

```
# md5sum /etc/corosync/corosync.conf
```

8. Запустить кластер на **altserver1**

```
# pcs cluster start --all
```

Вывод:

```
altserver2: Starting Cluster...
altserver3: Starting Cluster...
altserver1: Starting Cluster...
```

Проверить статус кластера:

```
# pcs status --full
```

Вывод:

```
Cluster name: sng-cluster

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
* Stack: corosync
* Current DC: altserver3 (3) (version 2.1.4-alt2-dc6eb4362) - partition with quorum
* Last updated: Tue Dec  6 13:03:27 2022
* Last change:  Tue Dec  6 13:02:40 2022 by hacluster via crmd on altserver3
* 3 nodes configured
* 0 resource instances configured

Node List:
* Online: [ altserver1 (1) altserver2 (2) altserver3 (3) ]

Full List of Resources:
* No resources

Migration Summary:

Tickets:

PCSD Status:
altserver1: Online
altserver2: Online
```

```
altserver3: Online
```

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

## 9. Выполнить настройку PostgreSQL на **altserver1** (Primary Server)

Настроить конфигурационные файлы:

```
# /etc/init.d/postgresql initdb  
  
# cat >> /var/lib/pgsql/data/postgresql.conf << EOF  
listen_addresses = '*'  
primary_conninfo = 'host=10.X.X.X port=5432 user=repli application_name=altserver1'  
wal_keep_size = 100  
max_wal_senders = 10  
EOF  
  
# cat >> /var/lib/pgsql/data/pg_hba.conf << EOF  
host    replication    all            $(hostname -s)    reject  
host    replication    all            127.0.0.0/8       reject  
host    replication    all            0.0.0.0/0         trust  
host    all             all            0.0.0.0/0         trust  
EOF  
  
# systemctl start postgresql; sleep 5; systemctl status postgresql
```

Создать пользователя для репликации с паролем **hacluster**:

```
# createuser -U postgres repli -P --replication
```

Настроить аутентификацию:

```
# cat > ~postgres/.pgpass << EOF  
10.X.X.X:5432:replication:repli:hacluster  
EOF  
  
# chown postgres: ~postgres/.pgpass  
# chmod 600 ~postgres/.pgpass
```

Добавить IP на второй интерфейс:

```
# ip addr add 10.X.X.X/21 dev ens20
```

## 10. Выполнить настройку PostgreSQL на **altserver2**, **altserver3**

```
# cat > ~postgres/.pgpass << EOF  
10.X.X.X:5432:replication:repli:hacluster  
EOF  
  
# chown postgres: ~postgres/.pgpass  
  
# chmod 600 ~postgres/.pgpass  
  
# su - postgres -s /bin/sh -c 'pg_basebackup --host 10.X.X.X --username repli --pgdata  
/var/lib/pgsql/data/ --progress'
```

Вывод:

```
25018/25018 kB (100%), 1/1 tablespace
```

Выполнить на **altserver2**, **altserver1**:

```
# sed -e "s/\\(primary_conninfo = .* application_name=\\).*'/\\1$(hostname -s)'" -i /var/lib/pgsql/data/postgresql.conf

# sed -e "s/\\(altserver1\\)\\(\\s*reject\\)/$(hostname -s)\\2/" -i /var/lib/pgsql/data/pg_hba.conf

# touch /var/lib/pgsql/data/standby.signal

# chown postgres:postgres /var/lib/pgsql

# systemctl start postgresql; sleep 5; systemctl status postgresql
```

Проверить на **altserver1**:

```
# psql -U postgres -xc "TABLE pg_stat_replication" | head
```

Вывод:

```
-[ RECORD 1 ]-----+-----
pid          | 6929
usesysid     | 16384
username     | repli
application_name | altserver3
client_addr  | 10.X.X.X
client_hostname | altserver3
client_port  | 57928
backend_start | 2022-12-06 16:11:37.859155+03
backend_xmin  |
```

11. Остановить и убрать из автозагрузки PostgreSQL на **всех хостах**

```
# systemctl disable --now postgresql
```

12. Удалить интерфейс на **altserver1**

```
# ip addr del 10.X.X.X/21 dev ens20
```

13. Отключить STONISH на **altserver1**

```
# pcs property set stonith-enabled=false
```

14. Отключить Symmetric Cluster на **altserver1**

```
# pcs property set symmetric-cluster=false
```

15. Создать ресурс на **altserver1**

```
# pcs resource create pgsql ocf:heartbeat:pgsql \
master_ip=10.X.X.X 'node_list=altserver1 altserver2' \
rep_mode=sync restart_on_promote=false \
pgctl=/usr/bin/pg_ctl pgdata=/var/lib/pgsql/data \
psql=/usr/bin/psql
```

16. Создать продвигаемый (promotable) ресурс на **altserver1**

```
# pcs resource promotable pgsql promoted-max=1 promoted-node-max=1 clone-max=2 notify=true clone-
node-max=1
```

17. Создать Cluster Virtual IP Resource Agent на **altserver1**

```
# pcs resource create vip-master ocf:heartbeat:IPaddr2 \  
cidr_netmask=21 ip=10.X.X.X
```

18. Добавить VIP-Master в группу master-group на **altserver1**

```
# pcs resource group add master-group vip-master
```

19. Закончить настройку кластера, добавив ограничения и включив опцию Symmetric Cluster на **altserver1**

```
# pcs constraint colocation add master-group with Master pgsql-clone  
# pcs constraint order promote pgsql-clone then start master-group symmetrical=false score=INFINITY  
# pcs constraint order demote pgsql-clone then stop master-group symmetrical=false score=0  
# pcs constraint location master-group avoids altserver3  
# pcs property set symmetric-cluster=true
```

20. Остановить кластер:

```
# pcs cluster stop --all
```

Ожидаемый результат

Успешная остановка кластера.

Фактический результат

**Кластер не остановлен.** Зависает при выполнении данной команды. Помогает выполнение на всех хостах:

```
# systemctl kill corosync
```

И процесс оживает и идёт дальше:

```
altserver2: Stopping Cluster (pacemaker)...  
altserver1: Stopping Cluster (pacemaker)...  
altserver3: Stopping Cluster (pacemaker)...  
altserver2: Stopping Cluster (corosync)...  
altserver1: Stopping Cluster (corosync)...  
altserver3: Stopping Cluster (corosync)...
```