Для настройки отказоустойчивости PostgreSQL будет использоваться следующее ПО:

* OS: Deiban 11 или 12
* Patroni
* Etcd
* Keepalived
* Pgbouncer
* Haproxy

Тестовый стенд развернут на системе виртуализации VirtualBox, ip-адресация по-умолчанию

Кластер будет построен на 3-х нодах. Будет рассмотрена конфигурация 1 master и 1 replica сервер СУДБ PostgreSQL. Но эту конфигурацию без труда можно переделать в 1 master и 2 replicas

Таким образом, ПО на хостах будет следующим:

|  |  |
| --- | --- |
| 1  2  3 | db1-1: etcd, keepalived, PostgreSQL, Partoni, PGbouncer, HAproxy  db1-2: etcd, keepalived, PostgreSQL, Partoni, PGbouncer, HAproxy  db1-3: etcd |

Подготовка хостов

Обновим ОС и установим (по желанию) базовый набор ПО

|  |  |
| --- | --- |
| 1  2  3 | $ sudo apt update && sudo apt upgrade -y  $ sudo apt -y install nano curl bind9-utils dnsutils telnet wget net-tools traceroute git tcpdump rsync open-vm-tools mlocate htop tar zip unzip cloud-guest-utils gdisk  $ sudo apt autoremove -y |

На всех 3-х нодах приводим файл hosts к виду:

|  |  |
| --- | --- |
| 1  2  3  4 | $ sudo nano /etc/hosts  10.0.2.15 db1-1  10.0.2.16 db1-2  10.0.2.17 db1-3 |

Переименуем каждую ноду через hostnamectl

|  |  |
| --- | --- |
| 1 | $ sudo hostnamectl set-hostname db1-1 # для каждой ноды свой hostname |

Установка Etcd (All nodes)

Скачиваем дистрибутив и распаковываем. На момент подготовки стенда, финальная версия etcd была: 3.5.5

|  |  |
| --- | --- |
| 1  2  3 | $ cd /tmp  $ wget https://github.com/etcd-io/etcd/releases/download/v3.5.5/etcd-v3.5.5-linux-amd64.tar.gz  $ tar xzvf etcd-v3.5.5-linux-amd64.tar.gz |

Перемещаем бинарники в каталог /usr/local/bin

|  |  |
| --- | --- |
| 1 | $ sudo mv /tmp/etcd-v3.5.5-linux-amd64/etcd\* /usr/local/bin/ |

Проверяем

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | $ etcd --version  etcd Version: 3.5.5  Git SHA: 19002cfc6  Go Version: go1.16.15  Go OS/Arch: linux/amd64  $ etcdctl version  etcdctl version: 3.5.5  API version: 3.5  $ etcdutl version  etcdutl version: 3.5.5  API version: 3.5 |

Предварительная настройка etcd (All nodes)

Создаем системную группу и пользователя

|  |  |
| --- | --- |
| 1  2 | $ sudo groupadd --system etcd  $ sudo useradd -s /sbin/nologin --system -g etcd etcd |

Создаем директории, назначаем владельца, права доступа

|  |  |
| --- | --- |
| 1  2  3  4 | $ sudo mkdir /opt/etcd  $ sudo mkdir /etc/etcd  $ sudo chown -R etcd:etcd /opt/etcd  $ sudo chmod -R 700 /opt/etcd/ |

Создаем конфиги для etcd (для каждой ноды разный)

db1-1:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $ sudo nano /etc/etcd/etcd.conf  ETCD\_NAME="etcd1"  ETCD\_LISTEN\_CLIENT\_URLS="http://10.0.2.15:2379,http://127.0.0.1:2379"  ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.0.2.15:2379"  ETCD\_LISTEN\_PEER\_URLS="http://10.0.2.15:2380"  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.0.2.15:2380"  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-postgres-cluster"  ETCD\_INITIAL\_CLUSTER="etcd1=http://10.0.2.15:2380,etcd2=http://10.0.2.16:2380,etcd3=http://10.0.2.17:2380"  ETCD\_INITIAL\_CLUSTER\_STATE="new"  ETCD\_DATA\_DIR="/var/lib/etcd"  ETCD\_ELECTION\_TIMEOUT="10000"  ETCD\_HEARTBEAT\_INTERVAL="2000"  ETCD\_INITIAL\_ELECTION\_TICK\_ADVANCE="false"  ETCD\_ENABLE\_V2="true" |

db1-2

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $ sudo nano /etc/etcd/etcd.conf  ETCD\_NAME="etcd2"  ETCD\_LISTEN\_CLIENT\_URLS="http://10.0.2.16:2379,http://127.0.0.1:2379"  ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.0.2.16:2379"  ETCD\_LISTEN\_PEER\_URLS="http://10.0.2.16:2380"  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.0.2.16:2380"  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-postgres-cluster"  ETCD\_INITIAL\_CLUSTER="etcd1=http://10.0.2.15:2380,etcd2=http://10.0.2.16:2380,etcd3=http://10.0.2.17:2380"  ETCD\_INITIAL\_CLUSTER\_STATE="new"  ETCD\_DATA\_DIR="/var/lib/etcd"  ETCD\_ELECTION\_TIMEOUT="10000"  ETCD\_HEARTBEAT\_INTERVAL="2000"  ETCD\_INITIAL\_ELECTION\_TICK\_ADVANCE="false"  ETCD\_ENABLE\_V2="true" |

db1-3

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | $ sudo nano /etc/etcd/etcd.conf  ETCD\_NAME="etcd3"  ETCD\_LISTEN\_CLIENT\_URLS="http://10.0.2.17:2379,http://127.0.0.1:2379"  ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.0.2.17:2379"  ETCD\_LISTEN\_PEER\_URLS="http://10.0.2.17:2380"  ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.0.2.17:2380"  ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-postgres-cluster"  ETCD\_INITIAL\_CLUSTER="etcd1=http://10.0.2.15:2380,etcd2=http://10.0.2.16:2380,etcd3=http://10.0.2.17:2380"  ETCD\_INITIAL\_CLUSTER\_STATE="new"  ETCD\_DATA\_DIR="/var/lib/etcd"  ETCD\_ELECTION\_TIMEOUT="10000"  ETCD\_HEARTBEAT\_INTERVAL="2000"  ETCD\_INITIAL\_ELECTION\_TICK\_ADVANCE="false"  ETCD\_ENABLE\_V2="true" |

Создаем Systemd Unit на каждой из нод

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | $ sudo nano /etc/systemd/system/etcd.service  [Unit]  Description=Etcd Server  Documentation=https://github.com/etcd-io/etcd  After=network.target  After=network-online.target  Wants=network-online.target    [Service]  User=etcd  Type=notify  #WorkingDirectory=/var/lib/etcd/  WorkingDirectory=/opt/etcd/  EnvironmentFile=-/etc/etcd/etcd.conf  User=etcd  # set GOMAXPROCS to number of processors  ExecStart=/bin/bash -c "GOMAXPROCS=$(nproc) /usr/local/bin/etcd"  Restart=on-failure  LimitNOFILE=65536  IOSchedulingClass=realtime  IOSchedulingPriority=0  Nice=-20    [Install]  WantedBy=multi-user.target |

Добавляем сервис в автозагрузку и запускаем

|  |  |
| --- | --- |
| 1  2  3 | $ sudo systemctl daemon-reload  $ sudo systemctl enable etcd  $ sudo systemctl start etcd |

Управление Etcd

Смотрим ноды кластера

|  |  |
| --- | --- |
| 1 | $ ETCDCTL\_API=2 etcdctl member list |

Проверяем кто лидер

|  |  |
| --- | --- |
| 1 | $ etcdctl endpoint status --cluster -w table |

Здоровье кластера

|  |  |
| --- | --- |
| 1 | $ etcdctl endpoint health --cluster -w table |

Другие команды проверки статуса

|  |  |
| --- | --- |
| 1  2  3  4  5 | $ ETCDCTL\_API=2 etcdctl member list  $ ETCDCTL\_API=3 etcdctl -w table --endpoints=db1-1:2379,db1-2:2379,db1-3:2379 endpoint status  $ ETCDCTL\_API=3 etcdctl endpoint status --cluster -w table  $ etcdctl endpoint status --cluster  $ etcdctl endpoint health --cluster |

Установка Postgresql 15 из репозитория (db1-1, db1-2)

Добавляем репозиторий и устанавливаем PostgreSQL

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $ sudo apt -y install gnupg2  $ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'  $ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -  $ sudo apt update  $ sudo apt -y install postgresql-15  $ sudo ln -s /usr/lib/postgresql/15/bin/\* /usr/sbin/ |

Добавляем пользователя replicator

|  |  |
| --- | --- |
| 1  2 | $ sudo -u postgres psql  =# create user replicator replication login encrypted password 'passwd'; |

Задаем пароль для пользователя postgres

|  |  |
| --- | --- |
| 1  2  3 | =# \password postgres;  Enter new password for user "postgres": passwd  Enter it again: passwd |

Подключаем расширения

|  |  |
| --- | --- |
| 1  2 | =# CREATE EXTENSION pg\_stat\_statements;  =# LOAD 'auto\_explain'; |

Создаем пользователя pgbouncer

|  |  |
| --- | --- |
| 1  2 | =# create user pgbouncer password '39xYw2KcwV';  =# \q |

Редактируем pg\_hba.conf

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | $ sudo nano /etc/postgresql/15/main/pg\_hba.conf  # Database administrative login by Unix domain socket  local all postgres peer  # TYPE DATABASE USER ADDRESS METHOD  # "local" is for Unix domain socket connections only  local all all peer  # IPv4 local connections:  host all all 127.0.0.1/32 md5  host all all 10.0.2.0/24 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 peer  host replication replicator localhost trust  host replication all 127.0.0.1/32 md5  host replication all ::1/128 md5  host replication replicator 10.0.2.0/24 md5 |

На второй и ноде удаляем (от пользователей root или postgres) содержимое каталога /var/lib/postgresql/15/main, т.к. этот каталог реплицируется после запуска Patroni

|  |  |
| --- | --- |
| 1  2  3 | $ sudo su  # rm -rf /var/lib/postgresql/15/main/\*  # exit |

Установка KeepAlived (db1-1, db1-2)

Установим ПО

|  |  |
| --- | --- |
| 1 | $ sudo apt -y install keepalived |

Правим конфиг

db1-1:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | $ sudo nano /etc/keepalived/keepalived.conf  global\_defs {  router\_id ocp\_vrrp  enable\_script\_security  script\_user root  }    vrrp\_script haproxy\_check {  script "/usr/libexec/keepalived/haproxy\_check.sh"  interval 5 # check every 5 seconds  weight 2 # add 2 points of prio if OK  }    vrrp\_instance VI\_1 {  interface enp0s3  virtual\_router\_id 11  priority 101 # 101 on master, 100 on backup  advert\_int 10  state MASTER  virtual\_ipaddress {  10.0.2.11  }  track\_script {  haproxy\_check  }  authentication {  auth\_type PASS  auth\_pass ehr0wg1chww8  }  } |

db1-2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | $ sudo nano /etc/keepalived/keepalived.conf  global\_defs {  router\_id ocp\_vrrp  enable\_script\_security  script\_user root  }    vrrp\_script haproxy\_check {  script "/usr/libexec/keepalived/haproxy\_check.sh"  interval 5 # check every 5 seconds  weight 2 # add 2 points of prio if OK  }    vrrp\_instance VI\_1 {  interface enp0s3  virtual\_router\_id 11  priority 100 # 101 on master, 100 on backup  advert\_int 10  state BACKUP  virtual\_ipaddress {  10.0.2.11  }  track\_script {  haproxy\_check  }  authentication {  auth\_type PASS  auth\_pass ehr0wg1chww8  }  } |

В данном конфиге ip 10.0.2.11 - кластерный (плавающий) ip

Создаем каталог и скрипт проверки HAproxy (до установки и настройки HAproxy не будет отрабатывать)

|  |  |
| --- | --- |
| 1  2  3  4 | $ sudo mkdir -p /usr/libexec/keepalived/  $ sudo nano /usr/libexec/keepalived/haproxy\_check.sh  #!/bin/bash  /bin/kill -0 `cat /var/run/haproxy/haproxy.pid` |

Назначаем права

|  |  |
| --- | --- |
| 1  2 | $ sudo chmod 700 /usr/libexec/keepalived/haproxy\_check.sh  $ sudo chmod +x /usr/libexec/keepalived/haproxy\_check.sh |

Запускаем сервис, добавляем в автозагрузку, проверяем

|  |  |
| --- | --- |
| 1  2  3 | $ sudo systemctl start keepalived  $ sudo systemctl enable keepalived  $ sudo systemctl status keepalived |

Установка Patroni (db1-1, db1-2)

Устанавливаем ПО

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | $ sudo apt -y install python3 python3-pip python3-dev python3-psycopg2 libpq-dev  $ sudo pip3 install --upgrade pip  $ sudo pip3 install psycopg2  $ sudo pip3 install psycopg2-binary  $ sudo pip3 install wheel  $ sudo pip3 install patroni  $ sudo pip3 install python-etcd |

Создаем каталог /etc/patroni/

|  |  |
| --- | --- |
| 1 | $ sudo mkdir /etc/patroni/ |

Создаем конфиг patroni.yml

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158 | $ sudo nano /etc/patroni/patroni.yml  ---  scope: postgres-cluster # одинаковое значение на всех узлах  name: db1-1 # разное значение на всех узлах  namespace: /service/ # одинаковое значение на всех узлах  restapi:  listen: 10.0.2.15:8008 # разное значение на всех узлах  connect\_address: 10.0.2.15:8008 # разное значение на всех узлах  authentication:  username: patroni  password: 'passwd'  etcd:  hosts: 10.0.2.15:2379,10.0.2.16:2379,10.0.2.17:2379 # список всех узлов, на которых установлен etcd  bootstrap:  method: initdb  dcs:  ttl: 60  loop\_wait: 10  retry\_timeout: 27  maximum\_lag\_on\_failover: 2048576  master\_start\_timeout: 300  synchronous\_mode: true  synchronous\_mode\_strict: false  synchronous\_node\_count: 1  # standby\_cluster:  # host: 127.0.0.1  # port: 1111  # primary\_slot\_name: patroni  postgresql:  use\_pg\_rewind: false  use\_slots: true  parameters:  max\_connections: 800  superuser\_reserved\_connections: 5  max\_locks\_per\_transaction: 64  max\_prepared\_transactions: 0  huge\_pages: try  shared\_buffers: 512MB  work\_mem: 128MB  maintenance\_work\_mem: 256MB  effective\_cache\_size: 4GB  checkpoint\_timeout: 15min  checkpoint\_completion\_target: 0.9  min\_wal\_size: 2GB  max\_wal\_size: 4GB  wal\_buffers: 32MB  default\_statistics\_target: 1000  seq\_page\_cost: 1  random\_page\_cost: 4  effective\_io\_concurrency: 2  synchronous\_commit: on  autovacuum: on  autovacuum\_max\_workers: 5  autovacuum\_vacuum\_scale\_factor: 0.01  autovacuum\_analyze\_scale\_factor: 0.02  autovacuum\_vacuum\_cost\_limit: 200  autovacuum\_vacuum\_cost\_delay: 20  autovacuum\_naptime: 1s  max\_files\_per\_process: 4096  archive\_mode: on  archive\_timeout: 1800s  archive\_command: cd .  wal\_level: replica  wal\_keep\_segments: 130  max\_wal\_senders: 10  max\_replication\_slots: 10  hot\_standby: on  hot\_standby\_feedback: True  wal\_log\_hints: on  shared\_preload\_libraries: pg\_stat\_statements,auto\_explain  pg\_stat\_statements.max: 10000  pg\_stat\_statements.track: all  pg\_stat\_statements.save: off  auto\_explain.log\_min\_duration: 10s  auto\_explain.log\_analyze: true  auto\_explain.log\_buffers: true  auto\_explain.log\_timing: false  auto\_explain.log\_triggers: true  auto\_explain.log\_verbose: true  auto\_explain.log\_nested\_statements: true  track\_io\_timing: on  log\_lock\_waits: on  log\_temp\_files: 0  track\_activities: on  track\_counts: on  track\_functions: all  log\_checkpoints: on  logging\_collector: on  log\_statement: mod  log\_truncate\_on\_rotation: on  log\_rotation\_age: 1d  log\_rotation\_size: 0  log\_line\_prefix: '%m [%p] %q%u@%d '  log\_filename: 'postgresql-%a.log'  log\_directory: /var/log/postgresql  initdb: # List options to be passed on to initdb  - encoding: UTF8  - locale: en\_US.UTF-8  - data-checksums  pg\_hba: # должен содержать адреса ВСЕХ машин, используемых в кластере  - host all all 0.0.0.0/0 md5  - host replication replicator 127.0.0.1/32 md5  - host replication replicator 10.0.2.0/24 md5  postgresql:  listen: 10.0.2.15,127.0.0.1:5432 # разное значение на всех узлах  connect\_address: 10.0.2.15:5432 # разное значение на всех узлах  use\_unix\_socket: true  data\_dir: /var/lib/postgresql/14/main  bin\_dir: /usr/lib/postgresql/14/bin  config\_dir: /etc/postgresql/14/main  pgpass: /var/lib/postgresql/.pgpass\_patroni  authentication:  replication:  username: replicator  password: passwd  superuser:  username: postgres  password: passwd  parameters:  unix\_socket\_directories: /var/run/postgresql  stats\_temp\_directory: /var/lib/pgsql\_stats\_tmp  remove\_data\_directory\_on\_rewind\_failure: false  remove\_data\_directory\_on\_diverged\_timelines: false  # callbacks:  # on\_start:  # on\_stop:  # on\_restart:  # on\_reload:  # on\_role\_change:  create\_replica\_methods:  - basebackup  basebackup:  max-rate: '100M'  checkpoint: 'fast'  watchdog:  mode: off # Allowed values: off, automatic, required  device: /dev/watchdog  safety\_margin: 5  tags:  nofailover: false  noloadbalance: false  clonefrom: false  nosync: false  # specify a node to replicate from (cascading replication)  # replicatefrom: (node name) |

Назначаем права

|  |  |
| --- | --- |
| 1  2 | $ sudo chown postgres:postgres -R /etc/patroni  $ sudo chmod 700 /etc/patroni |

Создаем каталог, назначаем владельца

|  |  |
| --- | --- |
| 1  2 | $ sudo mkdir /var/lib/pgsql\_stats\_tmp  $ sudo chown postgres:postgres /var/lib/pgsql\_stats\_tmp |

Тестируем старт (сразу на двух нодах)

|  |  |
| --- | --- |
| 1 | $ sudo -u postgres patroni /etc/patroni/patroni.yml |

Должен создастся файл .pgpass\_patroni, назначаем владельца, права

|  |  |
| --- | --- |
| 1  2 | $ sudo chown postgres:postgres /var/lib/postgresql/.pgpass\_patroni  $ sudo chmod 0600 /var/lib/postgresql/.pgpass\_patroni |

Создаем Sustemd Unit (на каждой ноде)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | $ sudo nano /etc/systemd/system/patroni.service  [Unit]  Description=High availability PostgreSQL Cluster  After=syslog.target network.target  [Service]  Type=simple  User=postgres  Group=postgres  # Read in configuration file if it exists, otherwise proceed  EnvironmentFile=-/etc/patroni\_env.conf  # Start the patroni process  ExecStart=/usr/local/bin/patroni /etc/patroni/patroni.yml  # Send HUP to reload from patroni.yml  ExecReload=/bin/kill -s HUP $MAINPID  # only kill the patroni process, not it's children, so it will gracefully stop postgres  KillMode=process  # Give a reasonable amount of time for the server to start up/shut down  TimeoutSec=60  # Do not restart the service if it crashes, we want to manually inspect database on failure  Restart=no  [Install]  WantedBy=multi-user.target |

Добавляем сервис а автозагрузку, стартуем, смотрим статус

|  |  |
| --- | --- |
| 1  2  3  4 | $ sudo systemctl daemon-reload  $ sudo systemctl start patroni  $ sudo systemctl status patroni  $ sudo systemctl enable patroni |

Смотрим состояние кластера

|  |  |
| --- | --- |
| 1 | $ sudo patronictl -c /etc/patroni/patroni.yml list |

Ошибка:  
== CRITICAL: system ID mismatch, node db1-2 belongs to a different cluster

Решение: Выполнить команды

|  |  |
| --- | --- |
| 1  2  3  4 | $ sudo systemctl stop patroni  $ ETCDCTL\_API=2 etcdctl rm /service/postgres-cluster/initialize  $ sudo systemctl restart patroni  $ sudo systemctl status patroni |

Создадим файла с настройками по умолчанию, что позволит не указывать настройки подключения для patronictl

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | $ sudo mkdir -p ~/.config/patroni/  $ sudo nano ~/.config/patroni/patronictl.yaml  dcs\_api:  etcd://localhost:2379  namespace: /service/  scope: postgres-cluster  authentication:  username: patroni  password: passwd |

Проверяем

|  |  |
| --- | --- |
| 1  2 | $ patronictl show-config  $ patronictl list |

Если на других нодах PostgreSQL был запущен раньше, удалите каталог данных, чтобы заработала реплика

|  |  |
| --- | --- |
| 1  2  3 | $ sudo systemctl stop patroni  $ sudo rm -rf /var/lib/postgresql/14/main/\*  $ sudo systemctl start patroni |

Ошибка:

==patroni: fatal the database system is starting up  
Ошибка в логах PostgreSQL: fatal the database system is starting up

Решение:

|  |  |
| --- | --- |
| 1  2  3 | $ sudo systemctl stop patroni  $ sudo rm -rf /var/lib/postgresql/15/main/\*  $ sudo systemctl start patroni |

Установка PGbouncer (db1-1, db1-2)

Устанавливаем PGBouncer, правим конфиг

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | $ sudo apt -y install pgbouncer  $ sudo mv /etc/pgbouncer/pgbouncer.ini /etc/pgbouncer/pgbouncer.ini.origin  $ sudo nano /etc/pgbouncer/pgbouncer.ini  [databases]  postgres = host=127.0.0.1 port=5432 dbname=postgres  \* = host=127.0.0.1 port=5432  [pgbouncer]  logfile = /var/log/postgresql/pgbouncer.log  pidfile = /var/run/postgresql/pgbouncer.pid  listen\_addr = \*  listen\_port = 6432  unix\_socket\_dir = /var/run/postgresql  auth\_type = md5  #auth\_type = trust  auth\_file = /etc/pgbouncer/userlist.txt  auth\_user = postgres  auth\_query = SELECT usename, passwd FROM pg\_shadow WHERE usename=$1  #admin\_users = pgbouncer, postgres  admin\_users = postgres  ignore\_startup\_parameters = extra\_float\_digits,geqo,search\_path  pool\_mode = session  #pool\_mode = transaction  server\_reset\_query = DISCARD ALL  max\_client\_conn = 10000  #default\_pool\_size = 20  reserve\_pool\_size = 1  reserve\_pool\_timeout = 1  max\_db\_connections = 1000  #max\_client\_conn = 900  default\_pool\_size = 500  pkt\_buf = 8192  listen\_backlog = 4096  log\_connections = 1  log\_disconnections = 1  # Documentation https://pgbouncer.github.io/config.html |

Создадим файл userlist.txt

|  |  |
| --- | --- |
| 1  2  3 | $ sudo nano /etc/pgbouncer/userlist.txt  "postgres" "passwd"  "pgbouncer" "passwd" |

Перезапускаем PGbouncer на нодах

|  |  |
| --- | --- |
| 1 | $ sudo systemctl restart pgbouncer |

Файл /etc/pgbouncer/userlist.txt содержит имена пользователей и пароли, с которыми PGbouncer подключается к базе. Пароли не обязательно хранить в открытым виде, можно так:

|  |  |
| --- | --- |
| 1  2 | ...  "pgbouncer" "md576a2173be6393254e72ffa4d6df1030a" |

Хэш 76a2173be6393254e72ffa4d6df1030a посчитан как MD5 от пароля

|  |  |
| --- | --- |
| 1 | $ echo -n 'passwd' | md5sum |

Тестируем

|  |  |
| --- | --- |
| 1  2 | $ sudo su - postgres  $ psql -p 6432 -h 127.0.0.1 -U postgres postgres |

Установка и настройка HAproxy (db1-1, db1-2)

Установим ПО

|  |  |
| --- | --- |
| 1 | $ sudo apt -y install haproxy |

Правим конфиг

db1-1:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | $ sudo nano /etc/haproxy/haproxy.cfg  global  maxconn 100000  log /dev/log local0  log /dev/log local1 notice  chroot /var/lib/haproxy  stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners  stats timeout 30s  user haproxy  group haproxy  daemon  defaults  mode tcp  log global  retries 2  timeout queue 5s  timeout connect 5s  timeout client 60m  timeout server 60m  timeout check 15s  listen stats  mode http  bind 10.0.2.15:7000  stats enable  stats uri /  listen postgres\_master  bind \*:5000  maxconn 10000  option tcplog  option httpchk OPTIONS /master  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 4 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas  bind \*:5001  maxconn 10000  option tcplog  option httpchk OPTIONS /replica  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas\_sync  bind \*:5002  maxconn 10000  option tcplog  option httpchk OPTIONS /sync  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas\_async  bind \*:5003  maxconn 10000  option tcplog  option httpchk OPTIONS /async  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008 |

db1-2:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | $ sudo nano /etc/haproxy/haproxy.cfg  global  maxconn 100000  log /dev/log local0  log /dev/log local1 notice  chroot /var/lib/haproxy  stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners  stats timeout 30s  user haproxy  group haproxy  daemon  defaults  mode tcp  log global  retries 2  timeout queue 5s  timeout connect 5s  timeout client 60m  timeout server 60m  timeout check 15s  listen stats  mode http  bind 10.0.2.16:7000  stats enable  stats uri /  listen postgres\_master  bind \*:5000  maxconn 10000  option tcplog  option httpchk OPTIONS /master  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 4 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas  bind \*:5001  maxconn 10000  option tcplog  option httpchk OPTIONS /replica  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas\_sync  bind \*:5002  maxconn 10000  option tcplog  option httpchk OPTIONS /sync  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008  listen postgres\_replicas\_async  bind \*:5003  maxconn 10000  option tcplog  option httpchk OPTIONS /async  balance roundrobin  http-check expect status 200  default-server inter 3s fastinter 1s fall 3 rise 2 on-marked-down shutdown-sessions  server db1-1 10.0.2.15:6432 check port 8008  server db1-2 10.0.2.16:6432 check port 8008 |

Проверка конфига

|  |  |
| --- | --- |
| 1 | $ sudo haproxy -f /etc/haproxy/haproxy.cfg -c |

Перезапускаем HAproxy, смотрим статус

|  |  |
| --- | --- |
| 1  2 | $ sudo systemctl restart haproxy  $ sudo systemctl status haproxy |

Настройка отказоустойчивого кластера завершена

Типовые кейсы Patroni

Если master упал, сделать реплику лидером

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | $ sudo patronictl -c /etc/patroni/patroni.yml list  $ sudo patronictl -c /etc/patroni/patroni.yml failover  Candidate ['db1-1'] []: db1-1  Current cluster topology  +--------+-----------+---------+---------+----+-----------+  | Member | Host | Role | State | TL | Lag in MB |  + Cluster: postgres-cluster (7154050747629718791) --------+  | db1-1 | 10.0.2.15 | Replica | running | 9 | 16 |  +--------+-----------+---------+---------+----+-----------+  Are you sure you want to failover cluster postgres-cluster? [y/N]: y  2022-10-14 15:46:19.53515 Successfully failed over to "db1-1"  +--------+-----------+--------+---------+----+-----------+  | Member | Host | Role | State | TL | Lag in MB |  + Cluster: postgres-cluster (7154050747629718791) -------+  | db1-1 | 10.0.2.15 | Leader | running | 9 | |  +--------+-----------+--------+---------+----+-----------+ |

Сменить лидера

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | $ sudo patronictl -c /etc/patroni/patroni.yml list  $ sudo patronictl -c /etc/patroni/patroni.yml switchover  Master [db1-1]: db1-1  Candidate ['db1-2'] []: db1-2  When should the switchover take place (e.g. 2022-10-15T02:36 ) [now]: now  Current cluster topology  +--------+-----------+--------------+---------+----+-----------+  | Member | Host | Role | State | TL | Lag in MB |  + Cluster: postgres-cluster (7154050747629718791) -+-----------+  | db1-1 | 10.0.2.15 | Leader | running | 12 | |  | db1-2 | 10.0.2.16 | Sync Standby | running | 12 | 0 |  +--------+-----------+--------------+---------+----+-----------+  Are you sure you want to switchover cluster postgres-cluster, demoting current master db1-1? [y/N]: y  2022-10-15 01:36:46.92463 Successfully switched over to "db1-2"  +--------+-----------+---------+---------+----+-----------+  | Member | Host | Role | State | TL | Lag in MB |  + Cluster: postgres-cluster (7154050747629718791) --------+  | db1-1 | 10.0.2.15 | Replica | stopped | | unknown |  | db1-2 | 10.0.2.16 | Leader | running | 12 | |  +--------+-----------+---------+---------+----+-----------+  cfgadmin@db1-1:~$ sudo patronictl -c /etc/patroni/patroni.yml list  +--------+-----------+--------------+---------+----+-----------+  | Member | Host | Role | State | TL | Lag in MB |  + Cluster: postgres-cluster (7154050747629718791) -+-----------+  | db1-1 | 10.0.2.15 | Sync Standby | running | 13 | 0 |  | db1-2 | 10.0.2.16 | Leader | running | 13 | |  +--------+-----------+--------------+---------+----+-----------+ |

Смотрим историю переключений

|  |  |
| --- | --- |
| 1 | $ sudo patronictl -c /etc/patroni/patroni.yml history |

Отключаем авто переключение Replica-Master

|  |  |
| --- | --- |
| 1 | sudo patronictl -c /etc/patroni/patroni.yml pause |

Реинициализировать кластер

|  |  |
| --- | --- |
| 1 | sudo patronictl -c /etc/patroni/patroni.yml reinit postgres-cluster |

postgres-cluster - задается в настройках patroni (scope: ...)

Реинициализировать ноду

|  |  |
| --- | --- |
| 1 | $ sudo patronictl -c /etc/patroni/patroni.yml reinit postgres-cluster db1-2 |

Редактируем конфиг patroni

|  |  |
| --- | --- |
| 1 | $ sudo patronictl -c /etc/patroni/patroni.yml edit-config |

Перезапустить кластер (обычно после редактирования конфига patroni)

|  |  |
| --- | --- |
| 1  2 | $ sudo patronictl -c /etc/patroni/patroni.yml restart postgres-cluster  #$ sudo patronictl -c /etc/patroni/patroni.yml reload postgres-cluster |

Статусы сервисов

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | $ sudo patronictl -c /etc/patroni/patroni.yml list  $ etcdctl endpoint status --cluster  $ etcdctl endpoint health --cluster  $ sudo systemctl status haproxy  $ sudo systemctl status patroni  $ sudo systemctl status etcd  $ sudo systemctl status keepalived |