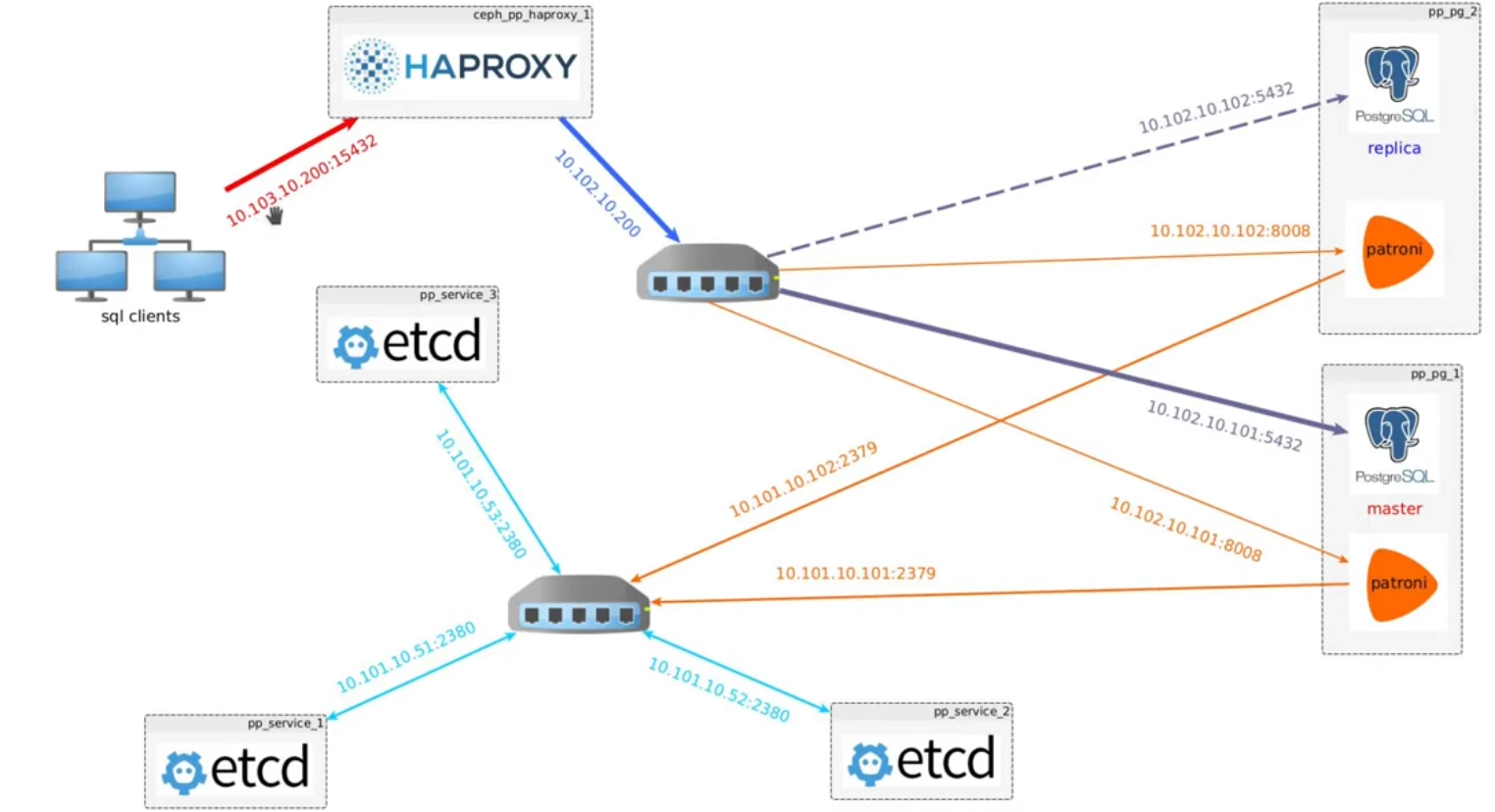
Patroni - настройка кластера

практическая работа



\* конфигурационные файлы представлены в git-репозитории :

<https://github.com/big-town/ha-cluster/tree/master/pg_cluster>

1. Подготовка инфраструктуры

Список ВМ

* 1. etcd1 — 'ppservice1' [10.101.10.51]
  2. etcd2 — 'ppservice2' [10.101.10.52]
  3. etcd3 — 'ppservice3' [10.101.10.53]
  4. postgres1 — 'pppg1' [10.101.10.101]
  5. postgres2 — 'pppg2' [10.101.10.102]
  6. haproxy1 — 'cephpphaproxy1' [dhcp,10.101.10.200]

*\* AstraLinux — имена хостов не должны содержать символ '\_',* т. е. pp\_service\_1 → ppservice1

\* Конфигурация: ОЗУ=1024, console mode, сетевые репозитории /etc/apt/sources.list

<https://github.com/big-town/ha-cluster/blob/master/pg_cluster/etcd/hosts>

1. Кластер **etcd** для создания распределённого хранилища на хостах etcd1, etcd2, etcd3
   1. Установка **etcd**

$ sudo apt install etcd

* 1. Инициализация кластера **etcd**

$ sudo rm -rf /var/lib/etcd/default/member/snap/\*

$ sudo rm -rf /var/lib/etcd/default/member/wal/\*

<https://github.com/big-town/ha-cluster/tree/master/pg_cluster/etcd>

$ systemctl restart etcd.service

Проверка **etcd**

$ etcdctl cluster-health

$ etcdctl member list

Параметры **/etc/default/etcd**:

$ **cat /etc/default/etcd**

* + ETCD\_DATA\_DIR - указывает расположение каталога данных кластера

ETCD\_NAME="ppservices1"

ETCD\_LISTEN\_CLIENT\_URLS="http://0.0.0.0:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS="http://10.101.10.51:2379"

ETCD\_LISTEN\_PEER\_URLS="http://0.0.0.0:2380"

ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS="http://10.101.10.51:2380"

ETCD\_INITIAL\_CLUSTER\_TOKEN="etcd-cluster"

ETCD\_INITIAL\_CLUSTER="ppservices1=http://10.101.10.51:2380,ppservices2=http://10.101.10.52:2380,ppservices3=http://10.101.10.53:2380"

ETCD\_INITIAL\_CLUSTER\_STATE="new"

ETCD\_DATA\_DIR="/var/lib/etcd/default"

ETCD\_ELECTION\_TIMEOUT="5000"

ETCD\_HEARTBEAT\_INTERVAL="1000"

* + ETCD\_LISTEN\_PEER\_URLS - задаёт схему и точку подключения для остальных узлов кластера, по шаблону scheme://IP:port. Схема может быть http, https. Альтернатива, unix:// или unixs:// для юникс сокетов. Если в качестве IP адреса указано 0.0.0.0, то указанный порт будет прослушиваться на всех интерфейсах.
  + ETCD\_LISTEN\_CLIENT\_URLS - задаёт схему и точку подключения для клиентов кластера. В остальном совпадает с ETCD\_LISTEN\_PEER\_URLS.
  + ETCD\_NAME - человекочитаемое имя этого узла кластера. Должно быть уникально в кластере. Для первого узла может быть любым. Для последующих должно совпадать с именем, указанным при добавлении узла.
  + ETCD\_HEARTBEAT\_INTERVAL - время в миллисекудах, между рассылками лидером оповещений о том, что он всё ещё лидер. Рекомендуется задавать с учётом сетевой задержки между узлами кластера.
  + ETCD\_ELECTION\_TIMEOUT - время в миллисекундах, которое проходит между последним принятым оповещением от лидера кластера, до попытки захватить роль лидера на ведомом узле. Рекомендуется задавать его в несколько раз большим, чем ETCD\_HEARTBEAT\_INTERVAL. Более подробно о этих параметрах можно прочесть в документации.
  + ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS - Список равноправных URL-адресов, по которым его могут найти остальные узлы кластера. Эти адреса используются для передачи данных по кластеру. По крайней мере, один из этих адресов должен быть маршрутизируемым для всех членов кластера. Могут содержать доменные имена. Используется только при первом запуске нового узла кластера.
  + ETCD\_ADVERTISE\_CLIENT\_URLS - Список равноправных URL-адресов, по которым его могут найти остальные узлы кластера. Эти адреса используются для передачи данных по кластеру. По крайней мере, один из этих адресов должен быть маршрутизируемым для всех членов кластера. Могут содержать доменные имена.
  + ETCD\_INITIAL\_CLUSTER - Список узлов кластера на момент запуска. Используется только при первом запуске нового узла кластера.
  + ETCD\_INITIAL\_CLUSTER\_TOKEN - Токен кластера. Должен совпадать на всех узлах кластера. Используется только при первом запуске нового узла кластера.
  + ETCD\_INITIAL\_CLUSTER\_STATE - может принимать два значения "new" и "existing". Значение "new" используется при первом запуске первого узла в кластере. При значении "existing", узел при старте будет пытаться установить связь с остальными узлами кластера.

1. Postgresql на pg1 и pg2

$ apt install postgresql

$ sudo systemctl stop postgresql

$ sudo systemctl **disable** postgresql

$ sudo rm -rf /var/lib/postgresql/11/main/\*

1. Patroni на pg1 и pg2

Установка:

sudo apt -y install python3-psycopg2

sudo apt -y install python3-pip

sudo pip3 install patroni[etcd]

1. YML

Создаем файл описание сервиса patroni

$ touch /etc/patroni.yml

Копировать на соотвествующий сервер конфигурацию Patroni:

<https://github.com/big-town/ha-cluster/blob/master/pg_cluster/patroni/pp_pg_1/patroni.yml>

<https://github.com/big-town/ha-cluster/blob/master/pg_cluster/patroni/pp_pg_2/patroni.yml>

1. Unit: Создать юнит сервиса Patroni на pg1, pg2

# touch /etc/systemd/system/patroni.service

или

# systemctl edit --full --force patroni.service

Пример unit-a:

<https://github.com/big-town/ha-cluster/blob/master/pg_cluster/patroni/patroni.service>

выполнить перезагрузку сервиса Patroni и старт при запуске

# systemctl start patroni

# systemctl **enable** patroni

1. Astra Linux — отключение контроля меток безопасности несуществующих пользователей

# /etc/parsec/mswitch.conf

zero\_if\_notfound: yes

1. Проверка состояния Patroni кластера на pg1, pg2

# patronictl -c /etc/patroni.yml list

1. Возможно потребуется очистить конфигурацию кластера Patroni на компонентах etcd,

# etcdctl rm /service/pg-ha-cluster --recursive

1. HA-Proxy на ***haproxy1***

$ sudo apt install haproxy

<https://github.com/big-town/ha-cluster/blob/master/pg_cluster/ha-proxy/ceph_pp_haproxy_1/haproxy.cfg>

Конфигурацию отредактировать и скопировать в

/etc/haproxy/haproxy.cfg

Проверить состояние статистических данных **HAProxy** через порт 7000:

1. Тестирование:
   1. Попытаться создать базу данных на **pg1**

root@pg1 # psql -Upostgres -c "create database dbtest1"

* 1. Попытаться создать базу данных на **pg2**

root@pg2 # psql -Upostgres -c "create database dbtest2"

Посмотреть вывод

* 1. Попытаться создать таблицу через **haproxy**:

root@haproxy# psql -h 10.102.10.200 -p15432 -U postgres dbtest1 -c 'create table tabletest1( id serial, dt timestamp default NOW() )';

root@haproxy# psql -h 10.102.10.200 -p15432 -U postgres dbtest1 -c '**select \* from tabletest1**';

* 1. Проверить существование таблицы на **pg1**

root@**pg1**# psql -U postgres dbtest1 -c '**select \* from tabletest1**';

* 1. Проверить существование таблицы на **pg2**

root@**pg2**# psql -U postgres dbtest1 -c '**select \* from tabletest1**';

* 1. Проверить работу кластера при выключении ноды **pg1**

**http://cephpphaproxy1**

* 1. WAL-файлы — reinit — в случае если мастер-нода долго была выключена

**root**@**pg1**# patronictl -c /etc/patroni/patroni.yml reinit pg-ha-cluster **pp**\_pg\_1

Сопутствующие источники

1. <https://latl.ru/sysadm/databases/patroni/howto_make_patroni_cluster_astra_linux_edir/>

2. [Высокодоступный кластер PostgreSQL на базе Patroni. С DNS точкой клиентского доступа, с поддержкой геораспределения](https://github.com/IlgizMamyshev/pgsql_cluster)