

Задание номер 6:
«Установка и настройка кластера PostgreSQL (Patroni, etcd, HAProxy)»
Выполнил:
Слушатель курса «OTUS».«Linux Administrator. Advanced»
Бурнейка Артурас Витаутасович

Для целей настоящей работы в Yandex Cloud создаются 5 (пять) ВМ:

- etcd-single
- postgresql-first
- postgresql-second
- postgresql-third
- haproxy-single

База данных ETCD для хранения конфигурации демона Patroni, будет развернута и настроена в единственном экземпляре, без отказоустойчивости.

В результате применения terraform-кода все ВМ успешно созданы.
Вывод продемонстрирован ниже:

```
Apply complete! Resources: 5 added, 0 changed, 0 destroyed.

Outputs:

addresses = [
  "178.154.205.5",
  "178.154.201.142",
  "178.154.205.161",
  "51.250.83.167",
  "158.160.119.137",
]
names = [
  "etcd-single",
  "haproxy-single",
  "postgresql-first",
  "postgresql-second",
  "postgresql-third",
]

arturas@oinzzi:~/Documents/otus/task-06/terraform$
```

Статус Yandex Cloud консоли, раздел Виртуальные машины:

Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания
postgresql-first	Running	Ubuntu 22.04 LTS	Intel Cascade Lake	2	5%	1 ГБ	Нет	20 ГБ	ru-central1-a	10.128.0.17	178.154.205.161	29.01.2024, в 18:56
etcd-single	Running	Ubuntu 22.04 LTS	Intel Cascade Lake	2	5%	1 ГБ	Нет	20 ГБ	ru-central1-a	10.128.0.19	178.154.205.5	29.01.2024, в 18:56
postgresql-second	Running	Ubuntu 22.04 LTS	Intel Cascade Lake	2	5%	1 ГБ	Нет	20 ГБ	ru-central1-a	10.128.0.29	51.250.83.167	29.01.2024, в 18:56
postgresql-third	Running	Ubuntu 22.04 LTS	Intel Cascade Lake	2	5%	1 ГБ	Нет	20 ГБ	ru-central1-a	10.128.0.9	158.160.119.137	29.01.2024, в 18:56
haproxy-single	Running	Ubuntu 22.04 LTS	Intel Cascade Lake	2	5%	1 ГБ	Нет	20 ГБ	ru-central1-a	10.128.0.14	178.154.201.142	29.01.2024, в 18:56

С помощью Ansible-роли etcd:

- устанавливается пакет etcd
- копируется преднастроенный конфигурационный файл

- происходит перезагрузка сервиса etcd

Результат применения роли для сервера etcd-single представлен на рисунке ниже:

```
root@etcd:~# Jun 29 16:32:15 2024 10.128.0.19.8000
ubuntu@etcd-single:~$ sudo systemctl status etcd.service
● etcd.service - etcd - highly-available key value store
   Loaded: loaded (/lib/systemd/system/etcd.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-01-29 16:32:52 UTC; 37s ago
     Docs: https://etcd.io/docs
           man:etcd
      Main PID: 2428 (etcd)
         Tasks: 7 (limit: 1012)
        Memory: 4.5M
          CPU: 235ms
        CGroup: /system.slice/etcd.service
                  └─2428 /usr/bin/etcd
```

С помощью инструмента командной строки etcdctl производится проверка здоровья кластера и выводится состав его участников:

```
_=/usr/bin/printenv
ubuntu@etcd-single:~$ etcdctl cluster-health
member 8e9e05c52164694d is healthy: got healthy result from http://10.128.0.19:2379
cluster is healthy
ubuntu@etcd-single:~$ etcdctl member list
8e9e05c52164694d: name=etcd-single peerURLs=http://localhost:2380 clientURLs=http://10.128.0.19:2379 isLeader=true
ubuntu@etcd-single:~$
```

Перварительно произведена установка переменной окружения для работы с более современной версией API:

```
ubuntu@etcd-single:~$ export ETCDCTL_API=3
ubuntu@etcd-single:~$ echo $ETCDCTL_API
3
```

Таким образом etcd (single mode) развернут и настроен для работы с Patroni.

С помощью Ansible-роли postgresql:

- устанавливается пакет postgresql
- устанавливается пакет postgresql-contrib
- копируется конфигурационный файл pg_hba.conf

Результат работы роли (для сервера postgresql-first) представлен на рисунке ниже. Служба postgresql в работе:

```
ubuntu@postgresql-first:~$ sudo systemctl status postgresql.service
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Mon 2024-02-05 15:41:05 UTC; 6min ago
     Process: 3465 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 3465 (code=exited, status=0/SUCCESS)
       CPU: 2ms

Feb 05 15:41:05 postgresql-first systemd[1]: Starting PostgreSQL RDBMS...
Feb 05 15:41:05 postgresql-first systemd[1]: Finished PostgreSQL RDBMS.
```

Вместе с СУБД PostgreSQL в системе появилась учётная запись её администратора (суперпользователя):

```
postgres:x:114:120:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
ubuntu@postgresql-first:/var/lib/postgresql$ cat /etc/passwd | grep postgres
postgres:x:114:120:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
```

Для работы с инструментом командной строки `psql` в демонстрационных целях режим аутентификации для суперпользователя (`postgres`) изменен с `PEER` на `TRUST`:

#	# Database administrative login by Unix domain socket	trust		
TYPE	DATABASE	USER	ADDRESS	METHOD

Настроочные файлы для СУБД PostgreSQL находятся в каталоге `/etc/postgresql/14/main/`:

```
ubuntu@postgresql-first:~$ ls -la /etc/postgresql/14/main/
total 68
drwxr-xr-x 3 postgres postgres 4096 Feb  5 16:22 .
drwxr-xr-x 2 postgres postgres 4096 Feb  5 15:40 ..
drwxr-xr-x 2 postgres postgres 4096 Feb  5 15:41 conf.d
-rw-r--r-- 1 postgres postgres 315 Feb  5 15:41 environment
-rw-r--r-- 1 postgres postgres 143 Feb  5 15:41 pg_ctl.conf
-rw-r----- 1 postgres postgres 5003 Feb  5 16:22 pg_hba.conf
-rw-r----- 1 postgres postgres 1636 Feb  5 15:41 pg_ident.conf
-rw-r--r-- 1 postgres postgres 29048 Feb  5 15:41 postgresql.conf
-rw-r--r-- 1 postgres postgres 317 Feb  5 15:41 start.conf
ubuntu@postgresql-first:~$
```

Успешное подключение к СУБД PostgreSQL через клиента `psql` после установки приводится ниже:

```
ubuntu@postgresql-first:~$ psql --username=postgres
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
Type "help" for help.

postgres=#
```

С помощью Ansible-роли `patroni`:

- устанавливаются необходимые python-пакеты (`python3-pip`, `python3-dev`, `libpq-dev`)
- обновляется пакетный менеджер `pip3`
- устанавливается демон `Patroni` и необходимые для его работы пакеты (`patroni`, `python-etcd`, `psycopg2`)
- генерируется конфиг файл из шаблона (`patroni.yaml`)
- копируется `unit`-файл для `systemd`
- производится перезагрузка `systemd`
- удаляется стандартный кластер `postgresql` (`14/main`)

В результате создан кластер, управляемый демоном `Patroni` и расположенный по пути `/var/lib/postgresql/14/patroni/`:

```
Feb 23 19:38:47 postgresql-first patroni[9938]: 2024-02-23 19:38:44 UTC [9938] FATAL: no pg_hba.conf entry for replication connection
ubuntu@postgresql-first:~$ sudo ls -la /var/lib/postgresql/14/patroni/
total 192
drwx----- 19 postgres postgres 4096 Feb 23 19:37 .
drwxrwxrwx  3 postgres postgres 4096 Feb 23 15:32 ..
drwx-----  6 postgres postgres 4096 Feb 23 16:11 base
drwx-----  2 postgres postgres 4096 Feb 23 19:37 global
-rw-----  1 postgres postgres 388 Feb 23 19:37 patroni_dynamic.json
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_commit_ts
drwx-----  1 postgres postgres 4096 Feb 23 15:37 pg_dynshmem
-rw-----  1 postgres postgres 4095 Feb 23 15:37 pg_hba.conf
-rw-----  1 postgres postgres 4095 Feb 23 19:37 pg_hba.conf.backup
-rw-----  1 postgres postgres 1636 Feb 23 19:37 pg_ident.conf
-rw-----  1 postgres postgres 1636 Feb 23 19:37 pg_ident.conf.backup
drwx-----  4 postgres postgres 4096 Feb 23 19:37 pg_logical
drwx-----  4 postgres postgres 4096 Feb 23 15:37 pg_multixact
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_notify
drwx-----  4 postgres postgres 4096 Feb 23 15:37 pg_shmat
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_serial
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_snapshots
drwx-----  2 postgres postgres 4096 Feb 23 19:37 pg_stat
drwx-----  2 postgres postgres 4096 Feb 23 19:38 pg_stat_tmp
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_subtrans
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_tblspc
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_twophase
-rw-----  1 postgres postgres 4096 Feb 23 19:37 pg_xlog
drwx-----  3 postgres postgres 4096 Feb 23 19:37 pg_wal
drwx-----  2 postgres postgres 4096 Feb 23 15:37 pg_xact
-rw-----  1 postgres postgres  88 Feb 23 15:37 postgresql.auto.conf
-rw-----  1 postgres postgres 28885 Feb 23 15:37 postgresql.base.conf
-rw-----  1 postgres postgres 28885 Feb 23 19:37 postgresql.base.conf.backup
-rw-----  1 postgres postgres  826 Feb 23 19:37 postgresql.conf
-rw-----  1 postgres postgres  826 Feb 23 19:37 postgresql.conf.backup
-rw-----  1 postgres postgres  468 Feb 23 19:37 postmaster.opts
-rw-----  1 postgres postgres 119 Feb 23 19:37 postmaster.pid
```

Кластер Patroni создан на трёх машинах:

+-----+					
Member	Host	Role	State	TL	Lag in MB
postgresql-first	10.128.0.17	Replica	streaming	1	0
postgresql-second	10.128.0.29	Leader	running	1	
postgresql-third	10.128.0.9	Replica	streaming	1	0

В случае необходимости удаления информации о кластере из распределенного хранилища конфигурации ETCD нужно выполнить команду:
`sudo patronictl -c /etc/patroni.yaml remove patroni`.

Выполнена проверка создания базы данных на машине, не являющейся лидером кластера:

```
postgres=# create database cultureobjects with owner=postgres;
ERROR:  cannot execute CREATE DATABASE in a read-only transaction
postgres=#
```

Результат: операцию невозможна так как данная машина находится в read-only режиме.

Выполнена проверка создания базы данных на машине-лидере кластера:

```
postgres=# create database cultureobjects with owner=postgres;
CREATE DATABASE
postgres=# \l
          List of databases
   Name    | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----+
cultureobjects | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
                           |          |          |          |          | postgres=CTc/postgres
                           |          |          |          |          | =c/postgres      +
                           |          |          |          |          | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
                           |          |          |          |          | postgres=CTc/postgres
(4 rows)
```

Результат: база данных успешно создана.

На машину-лидер скачивается предварительно заготовленный CSV-файл с данными для загрузки в Postgres:

```
and/or open issues at https://savannah.gnu.org/bugs/?func=additem&group=pyget.
ubuntu@postgresql-second:~$ wget https://storage.yandexcloud.net/culture-objects/cultobjects.csv
--2024-02-23 20:30:20-- https://storage.yandexcloud.net/culture-objects/cultobjects.csv
Resolving storage.yandexcloud.net (storage.yandexcloud.net)... 213.180.193.243, 2a02:6b8::1d9
Connecting to storage.yandexcloud.net (storage.yandexcloud.net)|213.180.193.243|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24109 (24K) [text/csv]
Saving to: 'cultobjects.csv'

cultobjects.csv           100%[=====] 23.54K --.-KB/s   in 0s

2024-02-23 20:30:20 (310 MB/s) - 'cultobjects.csv' saved [24109/24109]
```

С помощью SQL-команды CREATE TABLE создаётся рабочая таблица с данными (объектами культуры).

После успешного создания с помощью команды COPY FROM производится зашрузка предварительно подготовленных данных.

Результаты представлены на рисунке ниже:

```
ubuntu@postgresql-second:~$ psql -U postgres -d cultureobjects
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
Type "help" for help.

cultureobjects=# create table cultureobjects (
    id serial,
    native_id int,
    native_name text,
    region_id int,
    region_name text,
    full_address text,
    latitude real,
    longitude real,
    primary key (id)
);
CREATE TABLE
cultureobjects=# copy cultureobjects(native_id, native_name, region_id, region_name, full_address, latitude, longitude)
from '/var/lib/postgresql/14/cultobjects.csv'
delimiter '|'
csv header;
COPY 100
```

С помощью SQL-команды SELECT выводится содержимое таблицы с данными (объектами культуры):

147449	Дом, в котором в 1917 г. находился Совет рабочих и солдатских депутатов и Военно-революционный комитет. Здесь в 1925 г. выступал вождь немецких
147450	Дом Ф.Д. Маштакова
147454	Дом жилой
147455	Памятник Александру Невскому
147457	Здание городского торгового корпуса, где 14 декабря 1917 года была провозглашена Советская власть
147467	Военное собрание
147468	Дом со службами А.Я.Балацова
147488	Гостиница «Грандотель» Н.Ф.Мальцева
147508	Здание музея им С.Т. Коненкова
147509	Дом жилой
147510	Здание клуба купеческого общества (ныне областная библиотека имени В.И.Ленина)
147517	Новосибирский вокзал
147528	Здание реального училища
147529	Здание земской управы
147530	Жилой дом, начало XX в.
147532	Усадьба священника Малого: дом жилой, начало XX в.

Таким образом создана база данных (в кластерном исполнении) и в неё загружены все необходимые данные для работы бэкэнда.

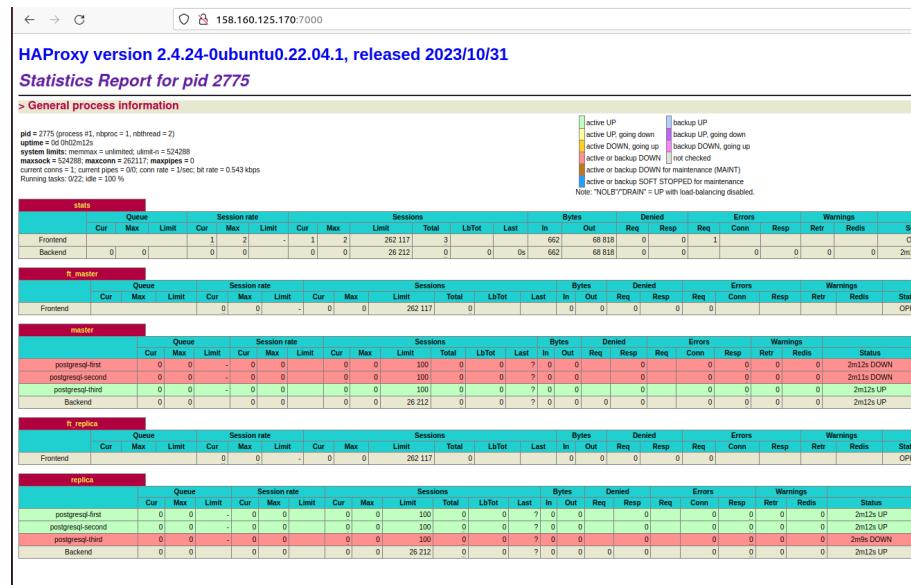
С помощью ansible-роли HAProxy:

- производится установка HAProxy
- копируется конфигурационный файл

В результате балансировщик HAProxy запущен в работу:

```
root@nginx-mon:~# ps aux | grep haproxy
ubuntu@haproxy-single:~$ sudo systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2024-02-26 11:29:21 UTC; 52s ago
       Docs: man:haproxy(1)
             File:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 1681 (haproxy)
      Tasks: 3 (limit: 1012)
     Memory: 69.3M
        CPU: 143ms
       CGroup: /system.slice/haproxy.service
               └─1681 /usr/sbin/haproxy -Ns -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haproxy-master.sock
Feb 26 11:29:21 haproxy-single systemd[1]: Starting HAProxy Load Balancer...
Feb 26 11:29:21 haproxy-single haproxy[1681]: [NOTICE] (1681) : New worker #1 (1683) forked
Feb 26 11:29:21 haproxy-single systemd[1]: Started HAProxy Load Balancer.
ubuntu@haproxy-single:~$
```

Топология кластера PostgreSQL-Patroni видна на веб-панели для мониторинга от HAProxy (доступ по публичному адресу):



Проводится проверка подключения к кластеру PostgreSQL-Patroni через узел с балансировщиком HAProxy (`haproxy-single.ru-central1.internal`):

```
ubuntu@postgresql-first:~$ psql --host=haproxy-single.ru-central1.internal --port=15432 --username=postgres
Password for user postgres:
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \l
              List of databases
   Name    | Owner | Encoding | Collate | Ctype | Access privileges
---+-----+-----+-----+-----+-----+
cutureobjects | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
readme_to_recover | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
                                         | postgres=CTc/postgres
                                         | =c/postgres      +
                                         | postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
                                         | postgres=CTc/postgres
(5 rows)

postgres=#
```

Таким образом кластер PostgreSQL под управлением демона Patroni настроен и запущен.
Доступ к кластеру возможен через сервер с балансировщиком HAProxy.

Спасибо OTUS за интересное задание и новый навык!