Сулейманов Роман Lallison. Домашнее задание. Модуль 2

Namespace Kubernetes: `sre-cource-student-93`
Имя проекта Hub Cloud MTS: `Проект для студента Lallison`

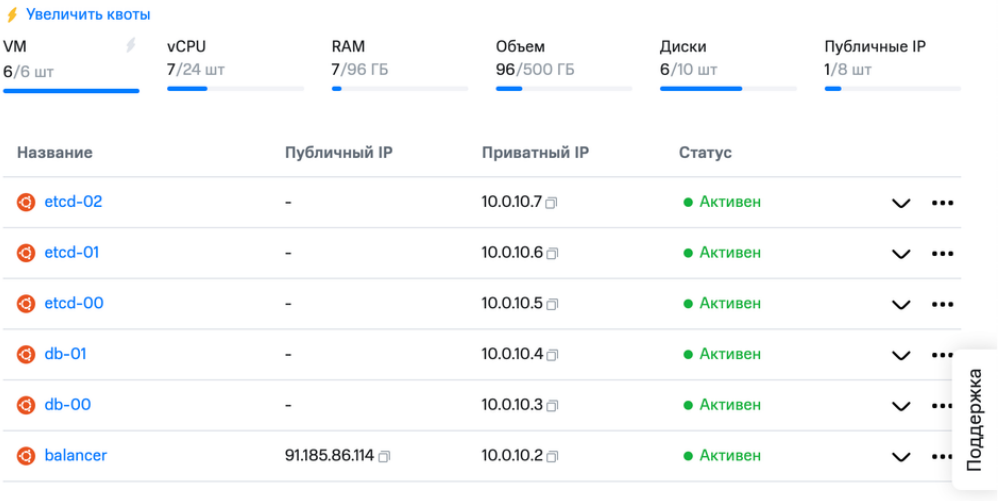
Шаги:

1. На личном ПК сгенерировал SSH ключ

```
ssh-keygen
```

2. В MTS Cloud Hub оформил запрос на увеличение квоты VM до 6 машин

3. Создал 6 VM, указал свой логин и id_rsa.pub личного ПК



Name	Public IP	Private IP	vCPU	RAM	OS	Desc
balancer	91.185.86.114	10.0.10.2	2	2	ubuntu	host jumper
db-00	-	10.0.10.3	1	1	ubuntu	master
db-01	-	10.0.10.4	1	1	ubuntu	replica
etcd-00	-	10.0.10.5	1	1	ubuntu	
etcd-01	-	10.0.10.6	1	1	ubuntu	
etcd-02	-	10.0.10.7	1	1	ubuntu	

4. Подключился к balancer с личного ПК

```
ssh 91.185.86.114
```

```
kferterb@mac ~ % ssh 91.185.86.114
The authenticity of host '91.185.86.114 (91.185.86.114)' can't be established.
ED25519 key fingerprint is SHA256:+1lX7Ycra8L22/sd2jV/4geTsclg/TsBhbp0zamQ1A4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '91.185.86.114' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-86-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Oct  8 06:35:25 AM UTC 2023

  System load:  0.00537109375      Processes:              149
  Usage of /:   34.7% of 14.91GB   Users logged in:        0
  Memory usage: 12%                IPv4 address for ens160: 10.0.10.2
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kferterb@balancer:~$
```

5. На balancer установил pipx, ansible, обновил ENV и переподключился к balancer
   Installing Ansible — Ansible Documentation

```
sudo apt install pipx
pipx install --include-deps ansible
pipx ensurepath
```

```
kferterb@balancer:~$ ansible --version
ansible [core 2.15.4]
  config file = None
  configured module search path = ['/home/kferterb/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /home/kferterb/.local/pipx/venvs/ansible/lib/python3.10/site-packages/ansible
  ansible collection location = /home/kferterb/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/kferterb/.local/bin/ansible
  python version = 3.10.12 (main, Jun 11 2023, 05:26:28) [GCC 11.4.0] (/home/kferterb/.local/pipx/venvs/ansible/bin/python)
  jinja version = 3.1.2
  libyaml = True
kferterb@balancer:~$
```

6. Скопировал ключи с личного ПК на balancer и проверил подключения к остальным хостам

```
scp /Users/kferterb/.ssh/id_rsa /Users/kferterb/.ssh/id_rsa.pub kferterb@91.185.86.114:/home/kferterb/.ssh/
```

7. В HOME balancer создал папку ansible и в ней склонировал репозиторий с готовой ролью PG
   GitHub - vitabaks/postgresql_cluster: PostgreSQL High-Availability Cluster (based on "Patroni" and DCS "etcd" or "consul"). Automating with Ansible.

```
mkdir ansible && cd ansible
git clone https://github.com/vitabaks/postgresql_cluster.git
```

8. Сконфигурировал inventory и проверил успешность подключения

```
kferterb@balancer:~/ansible/postgresql_cluster$ ansible all -m ping -i inventory
0.0.0.0 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.10.7 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.10.6 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.10.3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.10.4 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
10.0.10.5 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
kferterb@balancer:~/ansible/postgresql_cluster$
```

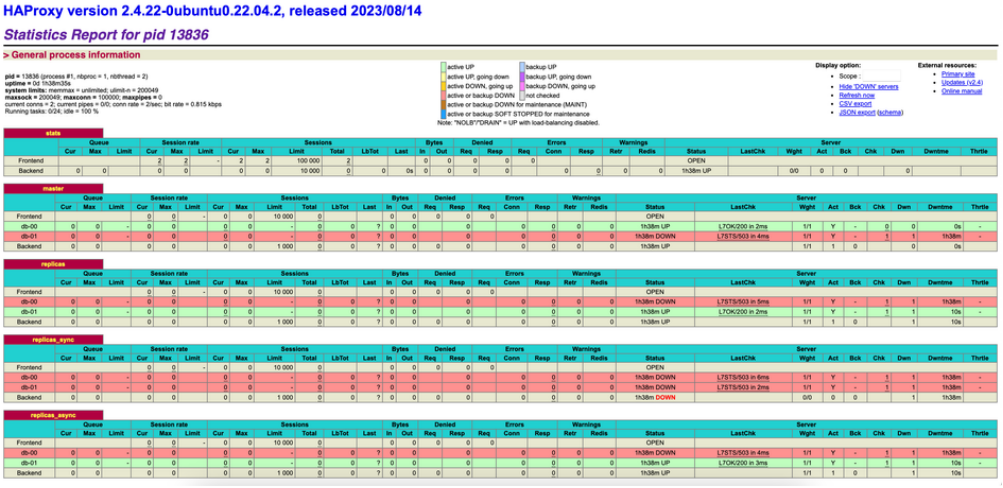
9. В main.yml изменил `with_haproxy_load_balancing: true` и `pgbouncer_install: false`

10. Запустил роль Ansible и убедился в успешности ее выполнения

```
ansible-playbook -i inventory deploy_pgcluster.yml
```

```
PLAY RECAP *********************************************************************************************************************************************************
0.0.0.0                    : ok=28   changed=16   unreachable=0    failed=0    skipped=92   rescued=0    ignored=0
10.0.10.3                  : ok=62   changed=33   unreachable=0    failed=0    skipped=328  rescued=0    ignored=0
10.0.10.4                  : ok=52   changed=31   unreachable=0    failed=0    skipped=310  rescued=0    ignored=0
10.0.10.5                  : ok=24   changed=12   unreachable=0    failed=0    skipped=54   rescued=0    ignored=0
10.0.10.6                  : ok=24   changed=12   unreachable=0    failed=0    skipped=47   rescued=0    ignored=0
10.0.10.7                  : ok=24   changed=12   unreachable=0    failed=0    skipped=47   rescued=0    ignored=0
localhost                  : ok=0    changed=0    unreachable=0    failed=0    skipped=5    rescued=0    ignored=0

kferterb@balancer:~/ansible/postgresql_cluster$
```

11. В браузере перешел на http://91.185.86.114:7000/stat и убедился в корректности HAProxy



12. Так же проверил с помощью NC

```
nc -zv 91.185.86.114 5000
```

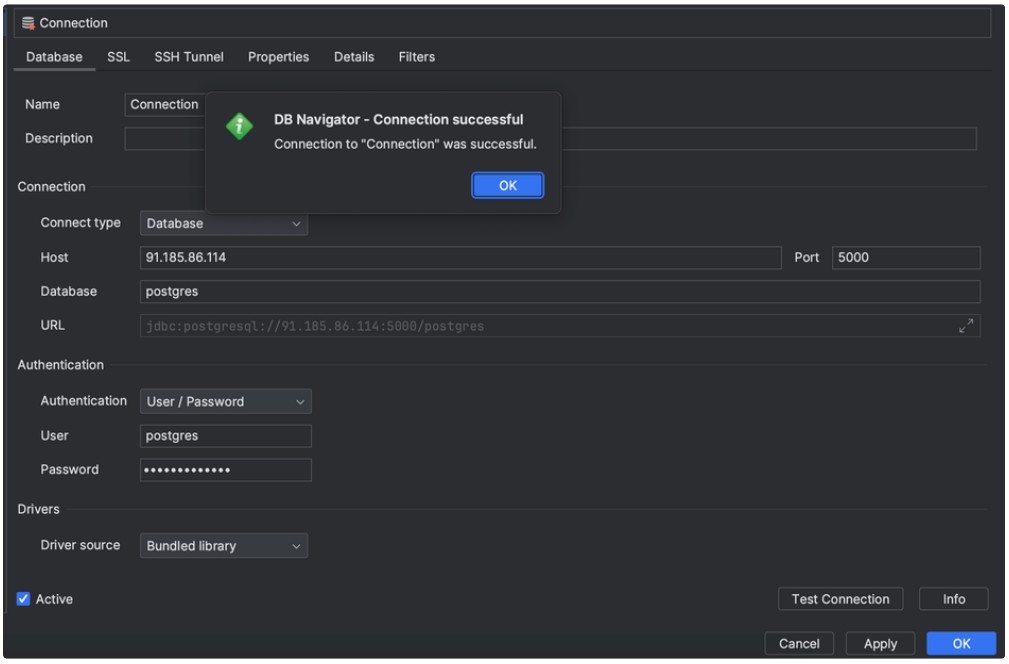
```
~ nc -zv 91.185.86.114 5000
Connection to 91.185.86.114 port 5000 [tcp/commplex-main] succeeded!
~
```

13. Внес balancer в /etc/postgresql/15/main/pg_hba.conf для db-00 и db-01 и перезагрузил PostgreSQL

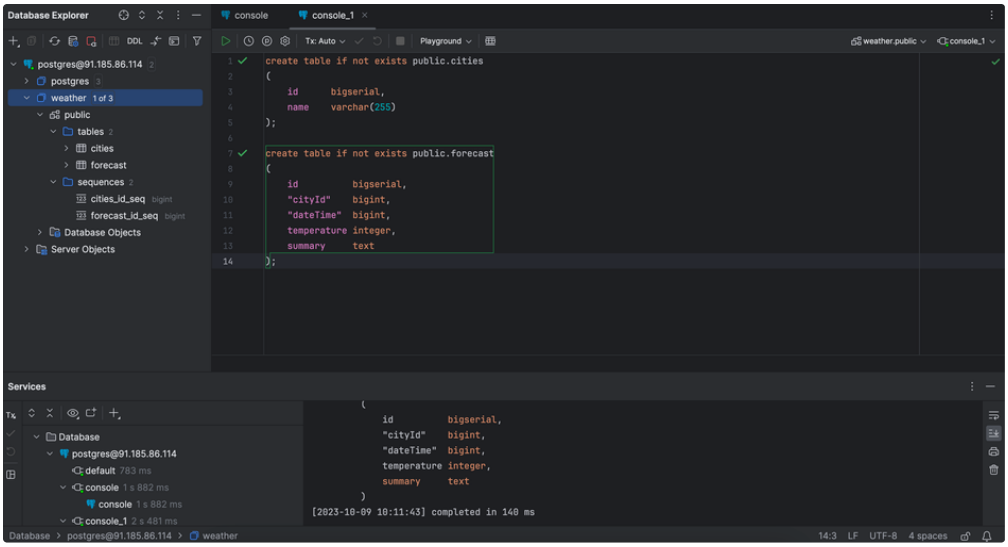
```
# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket

# TYPE      DATABASE                USER                    ADDRESS                 METHOD
  local     all                     postgres                                        trust
  local     replication             postgres                                        trust
  local     all                     all                                             peer
  host      all                     pgbouncer               127.0.0.1/32            trust
  host      all                     all                     127.0.0.1/32            scram-sha-256
  host      all                     all                     ::1/128                 scram-sha-256
  host      all                     all                     10.0.10.3/32        scram-sha-256
  host      all                     all                     10.0.10.4/32        scram-sha-256
  host      all                     all                     10.0.10.2/32        scram-sha-256
# Allow replication connections from localhost, by a user with the
# replication privilege.
  host      replication             replicator              localhost               trust
  host      replication             replicator              10.0.10.3/32        scram-sha-256
  host      replication             replicator              10.0.10.4/32        scram-sha-256
```

```
psql -U postgres
SELECT pg_reload_conf();
```

14. Проверил подключение к базе в IDE



15. Установил схему на базу данных



16. Создал, настроил и проверил Helm chart. Файлы deployment.yaml, values.yaml и configmap.yaml

```
helm create sre-course
helm template sre-course
```

17. Запустил установку Helm с флагом dry и убедился что все корректно

```
helm install sre-course sre-course --kubeconfig=/Users/kferterb/mtsSRE/k8s/kubeconfig/student_lallison@murena.
```

18. Запустил финальную установку Helm. Так же добавил команду для обновления конфигурации

```
helm install sre-course sre-course --kubeconfig=/Users/kferterb/mtsSRE/k8s/kubeconfig/student_lallison@murena.
helm upgrade sre-course sre-course --kubeconfig=/Users/kferterb/mtsSRE/k8s/kubeconfig/student_lallison@murena.
```

19. Проверил корректность деплоя