

Проекты VK WorkSpace

Отказоустойчивый HAProху

Оглавление

Шаг 1: Подготовка нод (серверов)	3
Шаг 2: Установка HAProxy	3
Шаг 3: Установка keepalived	3
Шаг 4: Настройка keepalived	4
Шаг 5: Настройка базовой конфигурации HAProxy	6

HAProxy — инструмент обеспечения высокой доступности и балансировки нагрузки. В статье описана установка HAProxy и keepalived для ОС Ubuntu Linux 20.04 и 22.04.

Установка состоит из шести этапов:

1. Подготовка нод (серверов)
2. Установка HAProxy
3. Установка keepalived
4. Настройка keepalived
5. Настройка базовой конфигурации HAProxy

Шаг 1: Подготовка нод (серверов)

1. Создайте две ноды (сервера) с последовательно пронумерованными именами хостов:
2. haproxy-server1.your_domain, 192.168.1.1;
3. haproxy-server2.your_domain, 192.168.1.2.
4. Создайте сопоставления имён хостов в DNS-сервере и внесите соответствующую A-запись для виртуального IP:

```
haproxy-server.your_domain A 192.168.1.13
```

Если такой возможности нет, внесите нужные записи в `/etc/hosts`.

5. Включите параметры ядра `net.ipv4.ip_nonlocal_bind`, `net.ipv4.ip_forward`, выполнив следующие команды:

```
sudo echo net.ipv4.ip_nonlocal_bind=1 >> /etc/sysctl.conf
sudo echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf
sudo sysctl -p
```

Шаг 2: Установка HAProxy

1. Установите HAProxy на все узлы, выполнив команду:

```
sudo apt install haproxy -y
```

2. Запустите сервис HAProxy на всех узлах и добавьте его в автозагрузку:

```
sudo systemctl enable --now haproxy
```

Шаг 3: Установка keepalived

Установите keepalived на все узлы:

```
sudo apt update
sudo apt install keepalived
```

Шаг 4: Настройка keepalived

1. Создайте директорию keepalived:

```
sudo mkdir -p /usr/libexec/keepalived
```

2. Создайте файл скрипта haproxy_check.sh для отслеживания работоспособности HAProxy:

```
sudo nano /usr/libexec/keepalived/haproxy_check.sh
```

3. Добавьте в файл haproxy_check.sh скрипт проверки работоспособности HAProxy:

```
#!/bin/bash
/bin/kill -0 $(cat /var/run/haproxy.pid)
```

4. Создайте файл конфигурации keepalived на каждой ноде:

```
sudo nano /etc/keepalived/keepalived.conf
```

5. Добавьте в файл keepalived.conf пример конфигураций keepalived для узла haproxy-server1.your_domain:

```
global_defs {
    router_id haproxy-server1
    enable_script_security
    script_user root
}

vrrp_script haproxy_check {
    script "/usr/libexec/keepalived/haproxy_check.sh"
    interval 2
    weight 2
}

vrrp_instance VI_1 {
    interface eth0
    virtual_router_id 101
    priority 100
    advert_int 2
    state MASTER
    virtual_ipaddress {
        192.168.1.13
    }
    track_script {
        haproxy_check
    }
    authentication {
        auth_type PASS
        auth_pass SecretPassword
    }
}
```

6. Добавьте в файл keepalived.conf пример конфигураций keepalived для узла haproxy-server2.your_domain:

```
global_defs {
    router_id haproxy-server2
    enable_script_security
    script_user root
}

vrrp_script haproxy_check {
    script "/usr/libexec/keepalived/haproxy_check.sh"
    interval 2
    weight 2
}

vrrp_instance VI_1 {
    interface eth0
    virtual_router_id 101
    priority 90
    advert_int 2
    state BACKUP
    virtual_ipaddress {
        192.168.1.13
    }
    track_script {
        haproxy_check
    }
    authentication {
        auth_type PASS
        auth_pass SecretPassword
    }
}
```

Где введённые параметры:

- `router_id` — ID роутера, уникальное значение на каждом узле;
- `script_user` — пользователь, от имени которого будут запускаться скрипты VRRP;
- `interface` — наименование интерфейса, к которому будет привязан keepalived;
- `virtual_router_id` — ID виртуального роутера, общее значение на всех узлах;
- `priority` — приоритет нод внутри виртуального роутера;
- `state` — тип роли ноды в виртуальном роутере;
- `virtual_ipaddress` — виртуальный IP;
- `auth_type` — тип аутентификации в виртуальном роутере;
- `auth_pass` — пароль.

7. Перезапустите keepalived на всех узлах:

```
sudo systemctl restart keepalived
```

Шаг 5: Настройка базовой конфигурации HAProxy

1. Переместите конфигурационный файл по умолчанию:

```
sudo mv /etc/haproxy/haproxy.cfg{, .original}
```

2. Создайте и откройте для редактирования новый конфигурационный файл с помощью команды:

```
sudo nano /etc/haproxy/haproxy.cfg
```

Пример базовой конфигурационный HAProxy для файла haproxy.cfg Базовая часть описывает параметры работы сервера HAProxy: режим работы, таймауты, количество подключений, включение web интерфейса и т.д. В примере приведена необходимая конфигурация параметров для работы сервера HAProxy. Подробнее об используемых параметрах читайте в [официальной документации HAProxy](#).

```
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 haproxy-server.your_domain:7000
    stats enable
    stats uri /

### RabbitMQ ###
### RabbitMQ ###


### S3 MinIO ###
### S3 MinIO ###
```

```
### Redis ###  
### Redis ###
```

3. Перезапустите HAProxy:

```
sudo systemctl restart haproxy
```

 Технический писатель: Белова Ирина

 6 мая 2026 г.