

# Создание HA-кластера с балансировкой нагрузки

## 1) Подготовка

- a. Поставить виртуальную машину без графической оболочки  
<https://www.debian.org/CD/netinst/>. Для виртуальной машины достаточно 1Гб RAM 3Гб HDD. При установке выбрать web server, ssh server, standard system utilities; язык - английский; страна - Россия, зеркало - mirror.yandex.ru
- a. Клонировать машину 2 раза (с генерацией новых MAC).

## 2) Настройка кластера

- a. Для удобства можно установить mc – удобный файловый менеджер  
[apt-get install mc](#)
- b. Если вторую виртуальную машину получили копированием первой, либо при установке не меняли имя хоста, или просто хотите изменить имя машин, то открываем файлы и меняем имена

[nano /etc/hostname](#)

```
GNU nano 2.2.6 File: /etc/hostname
cluster2
```

[nano /etc/hosts](#)

```
GNU nano 2.2.6 File: /etc/hosts
127.0.0.1 localhost
127.0.1.1 cluster2

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- c. На обе машины устанавливаем **corosync** и **pacemaker**.

[sudo apt-get install corosync pacemaker](#)

Если устанавливался Debian Wheezy { для установки haproxy надо добавить репозиторий:

[echo deb http://httpredir.debian.org/debian wheezy-backports-sloppy main |](#)  
[sed 's/\(.\\*\)-sloppy \(.\\*\)/&@1 2/' | tr @ '\n' | tee /etc/apt/sources.list.d/backports.list](#)

обновить:

[apt-get update](#)

установить:

[apt-get install haproxy -t wheezy-backports-sloppy](#)

}

- d. Конфигурируем **corosync** (выполняет функции коммуникатора-транспортёра в нашем виртуальном кластере, т.е. обеспечивает возможность коммуникации между нодами кластера):

- i. Узнаем ip адрес машины(В настройках VMware можно менять ip):

[ifconfig](#)

- ii. [nano /etc/corosync/corosync.conf](#)

Ищем строку **bindnetaddr**: и меняем на адрес сети нашего кластера (ip данной машины, только последняя цифра 0)

- iii. Теперь расшариваем настройки на другой компьютер

[scp /etc/corosync/corosync.conf ip\\_adress:/etc/corosync/corosync.conf](#)

Где ip\_adress – ip другого компьютера

- iv. Генерируем ключ:

[corosync-keygen](#)

- v. Расшариваем ключ на другой компьютер

[scp /etc/corosync/authkey ip\\_adress:/etc/corosync/authkey](#)

- vi. Добавляем corosync в автозапуск:  
`nano /etc/default/corosync`  
 Меняем START=no на START=yes
  - vii. `reboot`  
 Поднялся corosync
- e. Теперь нужно сконфигурировать **haproxy**:
- i. `nano /etc/haproxy/haproxy.cfg`  
 Дописываем в конце файла эти строчки:  
`frontend front` (font – имя, можно свои придумать)  
`bind 192.168.197.11:80` (ip адрес:порт виртуального интерфейса P.S. любой незанятый)  
`default_backend back`  
  
`backend back` (back – имя, можно свои придумать)  
`balance roundrobin`  
`server cluster1 192.168.197.128:81` (ip адрес первой машины)  
`server cluster2 192.168.197.129:81` (ip адрес второй машины)  
 80 порт для haproxy, 81 – для apache2  
 Порты для машин должны отличаться от порта для виртуального адреса, чтобы каждый слушал свой порт. haproxy будет слушать запросы и динамически направлять их на backend.
  - ii. Проверка валидности настроек haproxy:  
`haproxy -f /etc/haproxy/haproxy.cfg -c`
  - iii. Расшариваем эти настройки на вторую машину  
`scp /etc/haproxy/haproxy.cfg ip_adress:/etc/haproxy/haproxy.cfg`
  - iv. Haproxy должен быть доступен на исполнение. Ищем его и smod +x haproxy.  
*Но делаем это, только если кластер будет ругаться на него!!!*
  - v. `sevice haproxy start/stop/reload/status` – полезные команды
- f. Теперь нужно сконфигурировать **pacemaker** (управляет нашим кластером)
- i. Заходим в режим конфигурации:  
`crm configure`  
 Вводим следующее:
  - ii. `property no-quorum-policy="ignore"`  
`property stonith-enabled="false"` (отключаем то, что на кластере из 2-ух узлов вызовет дедлоки)  
`primitive VIP ocf:heartbeat:IPaddr2 params ip="ip_frontend" cidr_netmask="24(маска сети)" op monitor interval="1s"` (добавляем скрипт на виртуальный IP нашему pacemaker)  
`primitive HAP lsb:haproxy op monitor interval="1s"`  
 (добавляем скрипт на haproxy, который уже у сконфигурирован и будет слушать виртуальный айпи на фронтенде по 80-ому порту каждую секунду)
  - iii. `colocation CLC inf: VIP HAP` (определяем размещение, т.е. виртуальный ip и haproxy в нашем кластере всегда будут стартовать на одной машине, иначе haproxy не увидит VIP)
  - iv. `order ORD inf: VIP HAP` (определяем порядок запуска: сначала VIP потом HAP, который слушает VIP)
  - v. Теперь выполняем команду:  
`commit`
  - vi. `exit, end, status, show` – полезные команды в режиме конфигурации  
`delete` удалять примитивы и прочее. Например: delete ORD(это имя)
  - vii. `reboot`  
 Поднялся pacemaker.
  - viii. Чтобы зайти в командную оболочку pacemaker нужно набрать: `crm`  
`crm_mon` или `crm status` – монитор нашего кластера (показывает состояние)  
`crm configure show` – посмотреть конфигурацию pacemaker

```

root@cluster2:/home/dudaev# crm configure show
node cluster1
node cluster2
primitive HAP lsb:haproxy \
    op monitor interval="1s"
primitive VIP ocf:heartbeat:IPaddr2 \
    params ip="192.168.197.11" cidr_netmask="24" \
    op monitor interval="1s"
colocation CLC inf: VIP HAP
order ORD inf: VIP HAP
property $id="cib-bootstrap-options" \
    dc-version="1.1.7-ee0730e13d124c3d58f00016c3376a1de5323cff" \
    cluster-infrastructure="openais" \
    expected-quorum-votes="2" \
    no-quorum-policy="ignore" \
    stonith-enabled="false"
root@cluster2:/home/dudaev# _

```

#### g. Настройка **apache2**

##### i. Устанавливаем:

`apt-get install apache2`

##### ii. Открываем конфигурационный файл:

`nano /etc/apache2/ports.conf`

Находим и меняем, чтобы не было проблем с haproxy.

NameVirtualHost \*: (меняем на порт «для apache2», который указывали ранее)

Listen (аналогично)

```

NameVirtualHost *:81
Listen 81

```

##### iii. Создаем конфигурационный файл для нашего сайта (никакой зависимости имени файла и имени сайта нет, после этих действий доступ все равно будет осуществляться по ip):

`nano /etc/apache2/sites-available/mysite.com.conf`

И пишем туда минимальные данные:

`<VirtualHost 192.168.197.128:81 >` (ip адрес машины и порт для apache2)

`ServerName mysite.com`

`DocumentRoot /var/www/mysite.com` (Расположение в файловой системе)

`</VirtualHost>`

##### iv. Создаем папку для сайта

`mkdir /var/www/mysite.com`

##### v. Создаем стартовую страницу (либо копируем дефолтную (которая лежит рядом с папкой, например через mc), и меняем для узнаваемости)

`nano /var/www/cluster.org/index.html`

##### vi. Запускаем сайт

`a2ensite mysite.com.conf`

##### vii. Запускаем apache2

`service apache2 start`

##### viii. Повторяем для второго компьютера (Только ip другой) (Имена «сайтов» могут быть разными, они никак не связаны)

#### h. Теперь по VIP модно ходить на сайты с **apache2**, при этом каждый раз балансировщик будет бросать на следующий компьютер.

Если привязать к кукам, то лазить надо будет с разных браузеров. Для этого надо настроить **sticky sessions**:

Открываем настройки **haproxy**:

`nano /etc/haproxy/haproxy.cfg`

И меняем:

`balance roundrobin`

`server cluster1 192.168.197.128:81`

`server cluster2 192.168.197.129:81`

На:

```
balance roundrobin
cookie SERVERID insert indirect nocache
server cluster1 192.168.197.128:81 cookie cluster1 check
server cluster2 192.168.197.129:81 cookie cluster2 check
```

3) Проверка кластера:

Открываем браузер на третьей виртуалке и вбиваем виртуальный ip в адресную строку.

При обновлении страницы содержимое будет меняться (если на серверах лежат разные сайты)

Если отключить один сервер, то сайт должен дальше работать, но отображать только сайт другого кластера

4) Настраиваем ДНС (сайт у нас уже есть, доступный по виртуальному ip)

- a. Создаем четвертую машину без графического интерфейса, либо можно использовать третью с графическим

- b. Устанавливаем bind9:

```
apt-get install bind9
```

- c. Настройка dns-сервера

Создаем новую зону

```
nano /etc/bind/db.mysite.com
```

И пишем туда:

```
$TTL 604800
```

```
@ IN SOA ns.mysite.com. root.mysite.com. (
```

```
2016060201 ; Serial (при каждом изменении информации нужно менять на
единичку годмесяцденьномер)
```

```
10800 ; Refresh
```

```
3600 ; Retry
```

```
3600000 ; Expire
```

```
604800) ; Negative Cache TTL
```

```
;
```

```
@ IN NS ns.mysite.com.
```

```
ns IN A 192.168.197.130 (ip данной машины, с bind9)
```

```
www IN A 192.168.197.11
```

- d. Настройка файла с зонами:

```
nano /etc/bind/named.conf.default-zones
```

Добавляем туда:

```
zone "mysite.com" {
```

```
type master;
```

```
file "/etc/bind/db.mysite.com";
```

```
};
```

- e. Проверка файла зон на наличие ошибок:

```
named-checkconf -z
```

- f. Если никаких ошибок нет, обновим информацию о зонах:

```
rndc reload
```

- g. Настроим пути, до ДНСов:

```
nano /etc/network/interfaces
```

Допишем

```
dns-nameservers 192.168.197.130 (ip данной машины, с bind9)
```