

EVDOKIMOVA DARYA 21205
HAProxy SERVER aka ОТКАЗОУСТОЙЧИВЫЙ КЛАСТЕР

РЕКОМЕНДАЦИЯ: Всё делать с использованием мобильного интернета (короче говоря, телефон - точка доступа). Иначе сделаете из-под домашнего роутера, придёте в нгу - и там все слетит, потому что ip адреса другие, сеть другая. А с мобилой всегда одни и те же.

Здесь рассказываю про идею ha кластера. Все команды - делать по инструкции [из группы в вк](#). При возникновении проблем - гугол + может какие-то советы из моего дока помогут.

!!!ATTENTION!!!: все команды делать из-под режима 'su -l' .

Оборудование: VirtualBox (я делала на ubuntu, вот [инструкция](#) по скачиванию);
3 сервера:

- Node1 на [debian](#) (amd64)
- Node2 на [debian](#) (amd64)
- DNSServak на [ubuntu server](#) (потому что почему-то с debian пошло что-то не так, но может у вас норм будет).

Тип - подключения - сетевой мост(!!!) (иначе у нас 3 машины будут отображаться на какой-то один адрес, который потом принимает/отдает данные.

При клонировании установленной первой машины генерация новых mac-адресов - обязательна!!! (иначе все 3 машины будут с одним и тем же ip-адресом, а нам этого не надо).

Установка машины (если чо) - как в инструкции (пункт "с. Установка OS").

Важно! Нужно прописать статические адреса на узлах (Node1 and Node2). [А зачем?](#)

ТЕРМИНОЛОГИЯ

- Кластер — группа серверов (вычислительных единиц), объединенных каналами связи.
- Отказоустойчивость (Fault Tolerance, FT) — способность системы к дальнейшей работе после выхода из строя какого-либо её элемента.
- Высокая доступность (High Availability, HA) — в случае выхода из строя узла пользователь какое-то время не будет получать услугу, однако восстановление системы произойдёт автоматически; время простоя минимизируется.
- КВД — кластер высокой доступности, HA-кластер.

Что значит "[кластер высокой доступности](#)"?

[Принцип работы ha cluster](#).

[Про corosync and pacemaker](#)

- **Corosync** — программный продукт, который позволяет создавать единый кластер из нескольких аппаратных или виртуальных серверов. Corosync отслеживает и передает состояние всех участников (нод) в кластере. Этот продукт позволяет:
 - мониторить статус приложений;
 - оповещать приложения о смене активной ноды в кластере;
 - отправлять идентичные сообщения процессам на всех нодах;
 - предоставлять доступ к общей базе данных с конфигурацией и статистикой;
 - отправлять уведомления об изменениях, произведенных в базе.
-
- **Pacemaker** - менеджер ресурсов кластера. Он позволяет использовать службы и объекты в рамках одного кластера из двух или более нод. Вот вкратце его достоинства:
 - позволяет находить и устранять сбои на уровне нод и служб;
 - не зависит от подсистемы хранения: можем забыть общий накопитель, как страшный сон;
 - не зависит от типов ресурсов: все, что можно прописать в скрипты, можно кластеризовать;
 - поддерживает STONITH (Shoot-The-Other-Node-In-The-Head), то есть умершая нода изолируется и запросы к ней не поступают, пока нода не отправит сообщение о том, что она снова в рабочем состоянии;
 - поддерживает кворумные и ресурсозависимые кластеры любого размера;
 - поддерживает практически любую избыточную конфигурацию;
 - может автоматически реплицировать конфиг на все узлы кластера — не придется править все вручную;
 - можно задать порядок запуска ресурсов, а также их совместимость на одном узле;
 - поддерживает расширенные типы ресурсов: клоны (когда ресурс запущен на множестве узлов) и дополнительные состояния (master/slave и подобное) — актуально для СУБД (MySQL, MariaDB, PostgreSQL, Oracle);
 - имеет единую кластерную оболочку CRM с поддержкой скриптов.

ИДЕЯ

(*Здесь рассказываю про идею на кластера. Все команды - делать по инструкции. При возникновении проблем - гугол + может какие-то советы из моего дока помогут. Ну и еще уточнения команд будут, зачем что делали и тд).

Идея заключается в том, что с помощью Corosync мы построим кластер (грубо говоря, "объединим" хосты), а следить за его состоянием будем с помощью Pacemaker (его

главная задача — достижение максимальной доступности ресурсов, которыми он управляет, и защита их от сбоев).

[Пример](#) из жизни сисадмина, поможет понять, зачем нужен ha cluster.

ТЕПЕРЬ ДЕТАЛЬКИ

Хороший [сайт](#) с объяснениями про pacemaker and corosync.

Про corosync: Смотреть презентацию в обсуждении задания к лабе.

Вот [здесь](#) хорошо написано.

Corosync uses the totem protocol for "heartbeat" like monitoring of the other node's health. A token is passed around to each node, the node does some work (like acknowledge old messages, send new ones), and then it passes the token on to the next node. This goes around and around all the time. Should a node not pass it's token on after a short time-out period, the token is declared lost, an error count goes up and a new token is sent. If too many tokens are lost in a row, the node is declared lost/dead.

Once the node is declared lost, the remaining nodes reform a new cluster. If enough nodes are left to form quorum, then the new cluster will continue to provide services. In two-node clusters, quorum is disabled so each node can work on it's own.

В директиве totem изменяем (либо если его нет, то создаем) interface на следующий:

```
interface {  
    ringnumber: 0  
    bindnetaddr: 192.168.106.0  
    broadcast: yes  
}
```

Bindnetaddr – адрес сети нашего кластера (зависит от ip-адреса нашей виртуальной машины)

В директиве nodelist добавляем следующее:

```
node {  
    name: node1  
    nodeid: 1  
    ring0_addr: 192.168.106.104  
}  
node {  
    name: node2  
    nodeid: 2  
    ring0_addr: 192.168.106.105  
}
```

ring0_addr – это ip-адреса 1 и 2 узла

*Все настройки corosync -все по инструкции.

!При копировании ключа authkey возможна ошибка permission denied.

Совет: скопировать из-под рута (с Node1) на user (Node2) в корневую папку. А потом уже в нужную (/etc/corosync/).

Once the node is declared lost, the remaining nodes reform a new cluster. If enough nodes are left to form quorum, then the new cluster will continue to provide services. In two-node clusters, quorum is disabled so each node can work on it's own. (Это как раз наш случай)

*тут плавно перейдем к настройке Pacemaker

Читаем про [кворум](#).

Что такое кворум? Говорят, что кластер имеет кворум при достаточном количестве «живых» узлов кластера. Достаточность количества «живых» узлов определяется по формуле: $n > N/2$, где n – количество живых узлов, N – общее количество узлов в кластере.

Читаем про команды pacemaker-a.

В режиме конфигурации “crm configure” пишем следующие команды:

- **`property no-quorum-policy=ignore`**

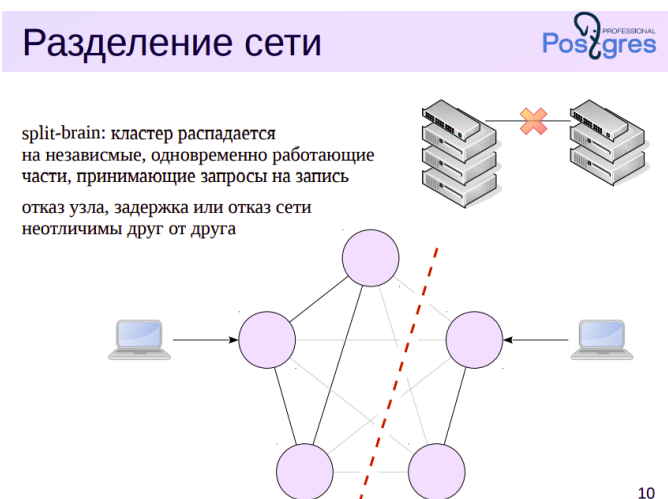
Это означает, что если если нода может достигаться более, чем до 50% нод, то нужно убить другую группу нод. В данном случае при отказе 1 ноды вторая не будет работать, т.к. не набрано 50%.

В кластере, состоящем из двух узлов, при сбое на одном из 2-х узлов не будет кворума. По умолчанию, если нет кворума, Pacemaker останавливает ресурсы. Чтобы этого избежать, нужно при настройке Pacemaker указать ему, чтобы наличие или отсутствие кворума не учитывалось.

(*Кворум - количество нод).

ВОПРОС: зачем нам вообще нужен кворум? Почему без него нельзя обойтись.

ОТВЕТ: Нужно избежать split brain - ситуация, когда кластер распадается на



независимые, одновременно работающие части, принимающие запросы на запись. Возникает рассогласование данных, и при восстановлении данные одной из групп будут вынуждено потеряны. Читаем [слайд номер 10](#). Еще [вот эта статья](#) полезная. Оттуда сейчас будет пример и объяснение.

Итак.

Важно понимать, что смерть узла системы и потеря с ним

связи не эквивалентны.

Разница проста — умерший узел не может совершать действий, которые «по неведению» окажутся деструктивными по отношению к остальной системе. Всегда можно устроить так, чтобы он умирал навсегда, или приходя в себя первым делом консультировался с остальной системой, чтобы ненароком ничего не сломать.

При разрыве связи все печальнее. Узел не может позволить себе умереть, потому что возможно только он и обеспечивает работоспособность (без связи с другими узлами невозможно понять, есть ли еще кто-то живой). Не может он и синхронизировать свою работу с остальными — связи-то нет. Остается действовать вслепую, на свой страх и риск.

Собственно, пример.

Допустим, в нашей системе ровно 2 совершенно одинаковых узла — А и В. Каждый из них хранит копию данных второго и может независимо обрабатывать запросы извне. Каждый, обрабатывая запрос, уведомляет второго об изменениях, чтобы данные оставались согласованы.

Вариант 1: узел А умирает.

Система продолжает работать как ни в чем не бывало — В продолжает обрабатывать запросы. Когда А придет в чувство, он первым делом синхронизируется с В и они вдвоем продолжат работать дальше. Ни доступность, ни согласованность не страдают.

Вариант 2: А и В живы, но связь между ними прервана.

При этом каждый из них продолжает принимать запросы извне, но не может уведомить второго об изменениях. Для каждого узла все выглядит так, будто второй узел умер и он действует в одиночку. Эту ситуацию часто называют «split-brain» — мозг разделился на два полушария, каждое из которых считает себя единственным хозяином ситуации. Система заболела шизофренией.

Если в этот момент на А был обработан запрос на удаление некой записи R, а на В был обработан запрос на модификацию той же самой записи, то данные стали несогласованы. Когда связь между А и В восстановится, при синхронизации всплывет конфликт — удалить R или оставить модифицированную версию? Тут можно выкручиваться разными стратегиями разрешения конфликтов, но consistency мы уже потеряли.

ТО ЕСТЬ решение о том, что делать с записью R определяется БОЛЬШИНСТВОМ голосов (The cluster quorum is the majority of voting nodes in the active cluster membership plus a witness vote). И именно отсюда и вытекает ответ на вопрос: “Почему отключаются ресурсы именно при более 50 процентах?”. А не 30, например.

Еще [статья полезная](#). И [еще](#). [Про CAP-теорему](#) (для общего развития, так сказать).

- `property stonith-enabled=false`

STONITH = Shoot-TheOther-Node-In-The-Head

When pacemaker is told that membership has changed because a node died, it looks to see what services might have been lost. Once it knows what was lost, it looks at the rules it's been given and decides what to do.

Generally, the first thing it does is "stonith" (aka "fence") the lost node. This is a process where the lost node is powered off, called power fencing, or cut off from the network/storage, called fabric fencing. In either case, the idea is to make sure that the lost node is in a known state. If this is skipped, the node could recover later and try to provide cluster services, not having realized that it was removed from the cluster. This could cause problems from confusing switches to corrupting data.

В нашем случае каждая нода будет пытаться сделать STONITH другой ноде. Ничего работать не будет, потому что все ноды умрут.

- primitive VIP ocf:heartbeat:IPaddr2 params ip=192.168.0.11 cidr_netmask=24 op monitor interval=1s

Создаём виртуальный ip-адрес(VIP, то есть создаем ресурс); с помощью op monitor interval=1s кластер будет следить за состоянием ресурса каждую секунду. По-умолчанию кластер не обеспечивает работоспособность ресурсов, т.е. не следит после запуска, жив ли ресурс. Чтобы это происходило, нужно добавлять операцию monitor, тогда кластер будет следить за состоянием ресурса. Параметр interval этой операции - интервал с каким делать проверку.

- primitive HAP lsb:haproxy op monitor interval=1s

Создаём ресурс haproxy(HAP)

- colocation CLC inf: VIP HAP

Это принуждение к размещению ресурсов на одном узле или наоборот на разных узлах. При отказе одной ноды, вторая должна иметь необходимые ресурсы для продолжения работы, поэтому необходимо разместить ресурсы на *каждой* ноде.

Иногда нужно, чтобы некоторый набор ресурсов выполнялся на одном узле, но не все эти ресурсы являются примитивами, поэтому их нельзя сгруппировать. В этом случае используется colocation - это принуждение к размещению ресурсов на одном узле или наоборот на разных узлах.

- order ORD inf: VIP HAP

Определяем порядок запуска ресурсов (сначала VIP, а потом HAP, зависящий от VIP)

- commit

Сохраняем настройки.

- /sbin/reboot

Перезапускаем машину.

*Когда установим pacemaker на Node2 и напишем 'crm configure status', то будут те же данные, что и на Node1 (на которой мы изначально задавали все настройки).

ПРО ВЗАИМОСВЯЗЬ МЕЖДУ COROSYNC-ОМ И PACEMAKER-ОМ

<https://github.com/voxpupuli/puppet-corosync/issues/32>

HAPROXY

Теперь перейдем к haproxy. HAProxy - это балансировщик нагрузки.

Отличная [статья](#). [Документация](#).

Как работает [балансировка](#)?

Команда `'net.ipv4.ip_nonlocal_bind=1'` [зачем](#) нужна? This command allows processes to bind() to non-local IP addresses, which can be quite useful for application such as load balancer.

То есть у нас есть floating point - он же frontend - адрес 192.68.106.111 - виртуальный адрес, который мы привязываем в наш кластер.

APACHE2

Читаем вот [это](#).

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

Этот каталог содержит файлы настроек для **виртуальных сетевых узлов** (Virtual Hosts) Apache2. Виртуальные сетевые узлы позволяют настраивать Apache2 на множество сайтов с отдельными конфигурациями.

- **ports.conf**: содержит инструкции, которые определяют какие TCP порты прослушивает Apache2.
- Инструкция **Listen** определяет порт, и в общем случае IP адрес, на которых Apache2 должен ожидать соединения. Если IP адрес не определен, Apache2 будет прослушивать все IP адреса, которые назначены компьютеру, где он запущен. Значение по умолчанию для **Listen** 80. Замените его на 127.0.0.1:80 чтобы Apache2 прослушивал только интерфейс внутренней петли, что сделает его недоступным из интернета; на 81 (например) для изменения порта доступа или оставьте как есть для стандартного функционирования. Эта инструкция может быть найдена и изменена в единственном файле `/etc/apache2/ports.conf`.
- 4. Инструкция **ServerName** необязательная и определяет на какой адрес FQDN ваш сайт должен отвечать. Изначальный виртуальный хост не имеет ServerName, поэтому отвечает на все запросы не соответствующие директивам ServerName других виртуальных хостов. Если вы приобрели доменное имя `ubunturocks.com` и хотите прописать его на вашем Ubuntu сервере, значение ServerName для файла настроек вашего виртуального хоста должно быть `ubunturocks.com`. Добавьте эту инструкцию в файл нового виртуального хоста, который вы создавали ранее (`/etc/apache2/sites-available/mynewsite`).
- Инструкция **DocumentRoot** определяет где Apache2 будет искать файлы, которые являются содержимым сайта. По умолчанию используется значение `/var/www`, как определено в `/etc/apache2/sites-available/default`. Если желаете, можете изменить это значение в файле сайта вашего виртуального хоста и не забудьте создать этот каталог, если необходимо!

ПРО DNS

Статья на [Хабре](#).

=====

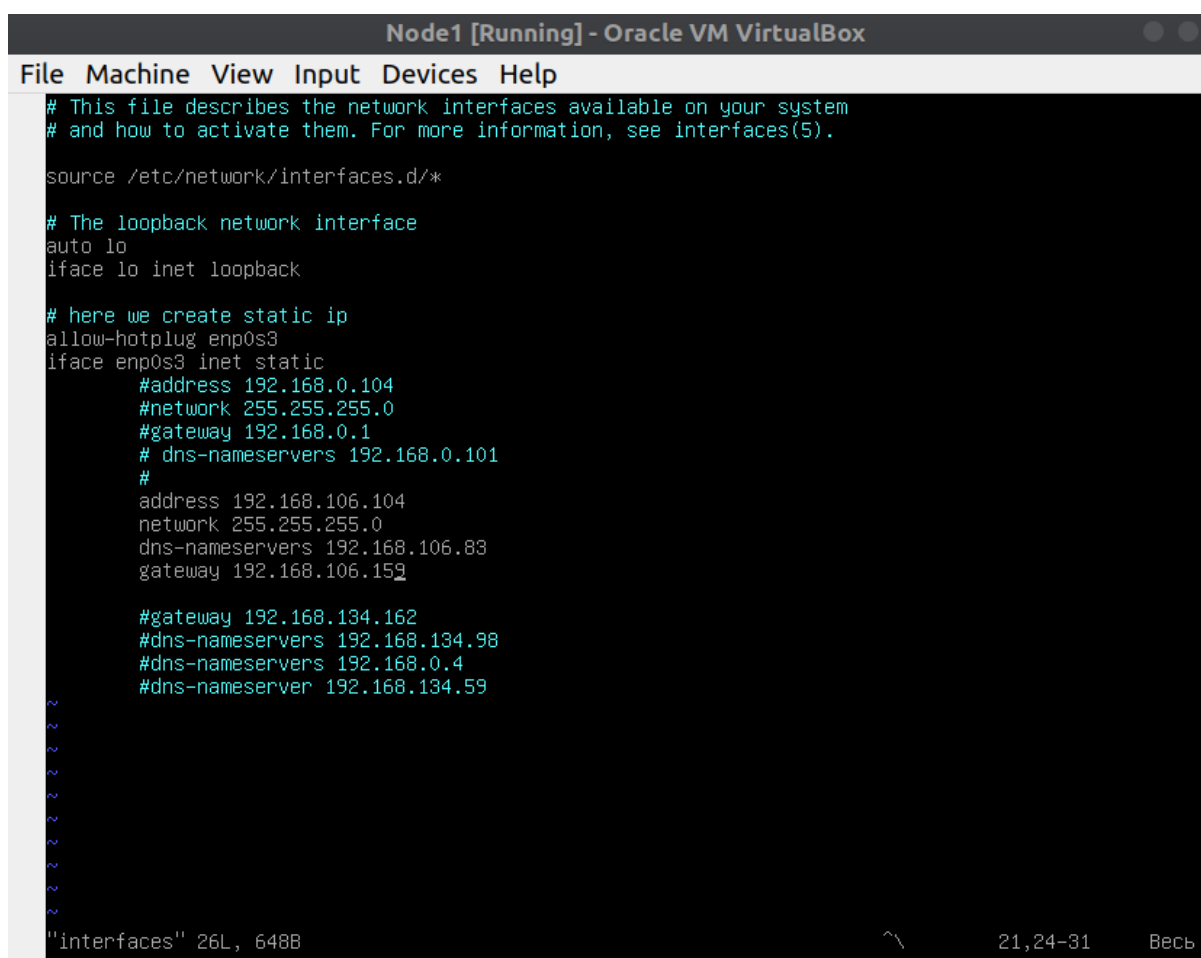
ПРО КЭШИРОВАНИЕ И ХЭШИРОВАНИЕ!!!

[Кайфовая статья](#) на английском.

(Украдена из инструкции)

ip a

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



/etc/corosync/corosync.conf

```
# Please read the corosync.conf.5 manual page
totem {
    version: 2

    # Corosync itself works without a cluster name, but DLM needs one.
    # The cluster name is also written into the VG metadata of newly
    # created shared LVM volume groups, if lvmlockd uses DLM locking.
    cluster_name: debian

    # crypto_cipher and crypto_hash: Used for mutual node authentication.
    # If you choose to enable this, then do remember to create a shared
    # secret with "corosync-keygen".
    # enabling crypto_cipher, requires also enabling of crypto_hash.
    # crypto works only with knet transport
    crypto_cipher: none
    crypto_hash: none

    interface {
        ringnumber: 0
        #bindnetaddr: 192.168.0.0
        bindnetaddr: 192.168.106.0
        broadcast: yes
    }
}

logging {
    # Log the source file and line where messages are being
    # generated. When in doubt, leave off. Potentially useful for
    # debugging.
    fileline: off
    # Log to standard error. When in doubt, set to yes. Useful when
    # running in the foreground (when invoking "corosync -f")
    to_stderr: yes
    # Log to a log file. When set to "no", the "logfile" option
    # must not be set.
```

1,39

Навєрxy

```
    # must not be set.
    to_logfile: yes
    logfile: /var/log/corosync/corosync.log
    # Log to the system log daemon. When in doubt, set to yes.
    to_syslog: yes
    # Log debug messages (very verbose). When in doubt, leave off.
    debug: off
    # Log messages with time stamps. When in doubt, set to hires (or on)
    #timestamp: hires
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

quorum {
    # Enable and configure quorum subsystem (default: off)
    # see also corosync.conf.5 and votequorum.5
    provider: corosync_votequorum
}

nodelist {
    # Change/uncomment/add node sections to match cluster configuration
    #
    node {
        # Hostname of the node
        name: node1
        # Cluster membership node identifier
        nodeid: 0
        # Address of first link
        ring0_addr: 127.0.0.1
        # When knet transport is used it's possible to define up to 8 links
        #ring1_addr: 192.168.1.1
    }
    # ..3
```

40,15-22

70%

```

# ..3
#

node {
    name: node1
    nodeid: 1
    ring0_addr: 192.168.106.104
}

node {
    name: node2
    nodeid: 2
    ring0_addr: 192.168.106.105
}

```

84,1

Внизу

/etc/haproxy/haproxy.cfg

Node1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```

global
    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

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermediate
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

defaults
    log          global
    mode         http
    option       httplog
    option       dontlognull
    timeout connect 5000
    timeout client  50000
    timeout server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http

```

1,1

Наверх

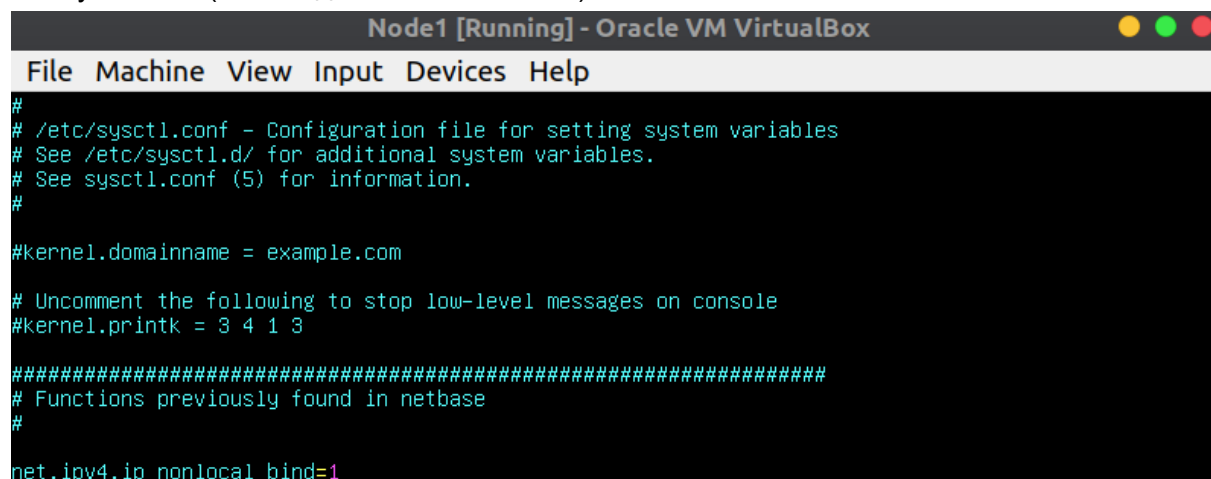
```

        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend front
    #bind 192.168.0.11:80
    bind 192.168.106.111:80
    default_backend back
backend back
    balance roundrobin
    server node1 192.168.106.104:81 check
    server node2 192.168.106.105:81 check

```

/etc/sysctl.conf (только добавленная часть)



```

Node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com

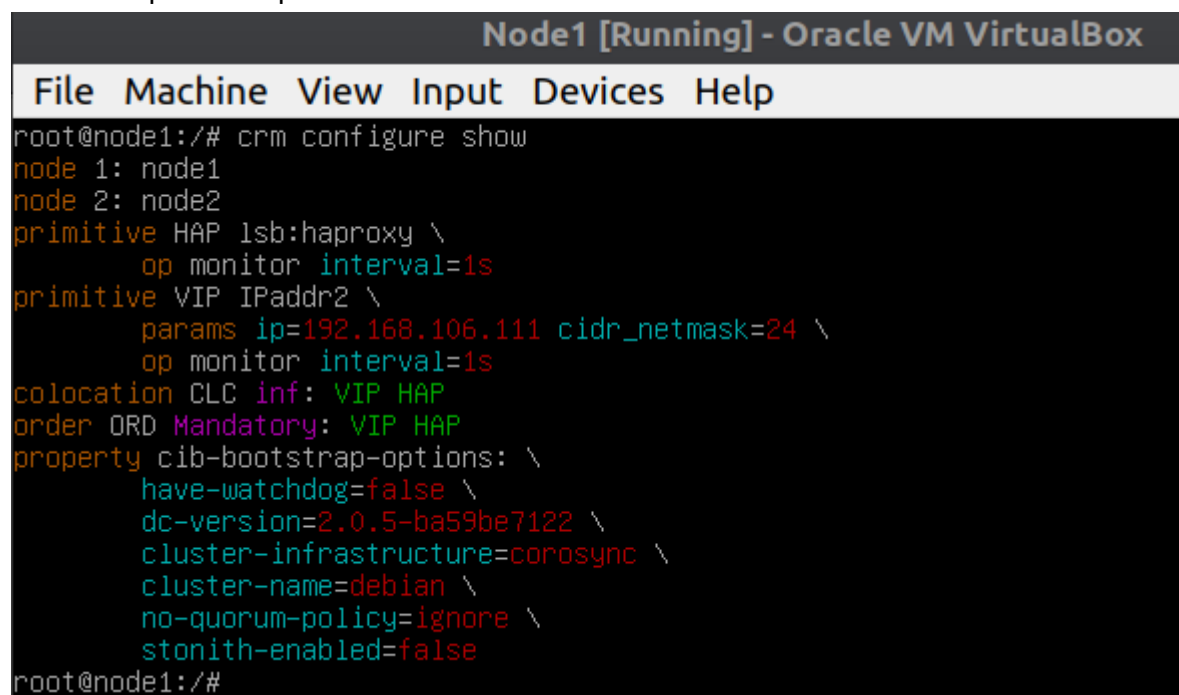
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

#####
# Functions previously found in netbase
#
net.ipv4.ip_nonlocal_bind=1

```

Про pacemaker

Вставляю прост настройки



```

Node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@node1:/# crm configure show
node 1: node1
node 2: node2
primitive HAP lsb:haproxy \
    op monitor interval=1s
primitive VIP IPaddr2 \
    params ip=192.168.106.111 cidr_netmask=24 \
    op monitor interval=1s
colocation CLC inf: VIP HAP
order ORD Mandatory: VIP HAP
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.5-ba59be7122 \
    cluster-infrastructure=corosync \
    cluster-name=debian \
    no-quorum-policy=ignore \
    stonith-enabled=false
root@node1:/#

```

```
/etc/apache2/ports.conf
```

```
Node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

#Listen 80 #так было
NameVirtualHost *:81
Listen 81

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

#NameVirtualHost *:81

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

```
/etc/apache2/sites-available/dashasite.com.conf
```

```
Node1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
<VirtualHost 192.168.106.104:81>
  ServerName fit.nsu
  DocumentRoot /var/www/fit.nsu
</VirtualHost>

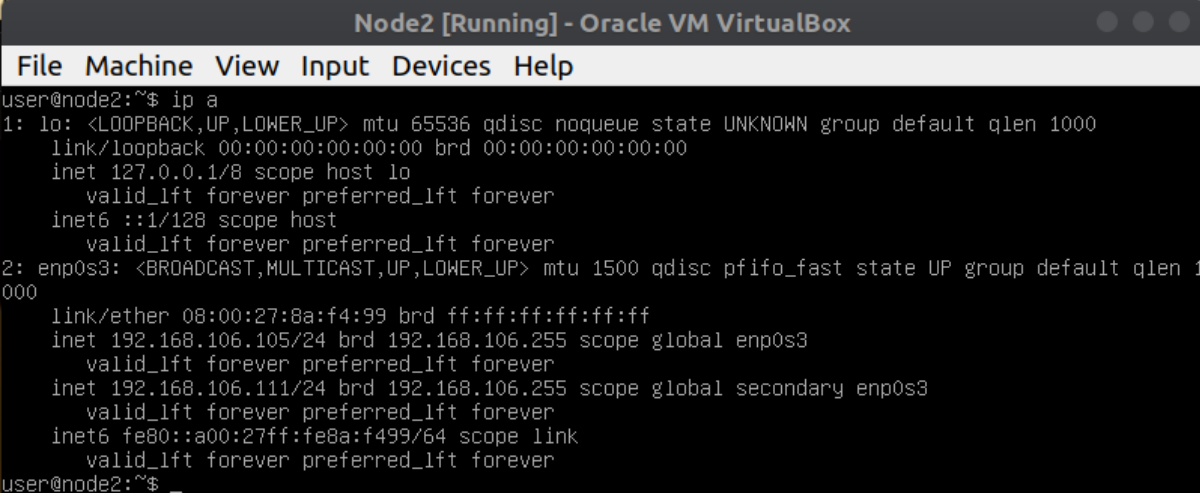
<VirtualHost 192.168.106.104:81>
  ServerName fit2.nsu
  DocumentRoot /var/www/fit2.nsu
</VirtualHost>

...

/etc/apache2/sites-available/dashasite.com.conf" 9L, 201B
```

NODE2

ip a



```
Node2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
user@node2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:8a:f4:99 brd ff:ff:ff:ff:ff:ff
    inet 192.168.106.105/24 brd 192.168.106.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.106.111/24 brd 192.168.106.255 scope global secondary enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe8a:f499/64 scope link
        valid_lft forever preferred_lft forever
user@node2:~$
```

Может сюда надо было б еще че-то вставить, но уже не помню, что именно ((

SERVAK

```
/etc/bind/db.nsu
```

```

DNSServak [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.nsu.          root.nsu. (
                        2           ; Serial
                        604800       ; Refresh
                        86400        ; Retry
                        2419200      ; Expire
                        604800 )     ; Negative Cache TTL
;
ns        IN      NS       ns.nsu.
fit       IN      A        192.168.106.83
fit2      IN      A        192.168.106.111
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~/etc/bind/db.nsu" [только для чтения] 15L, 302B
15,1
Весь

```

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

```
DNSServak [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.localhost";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

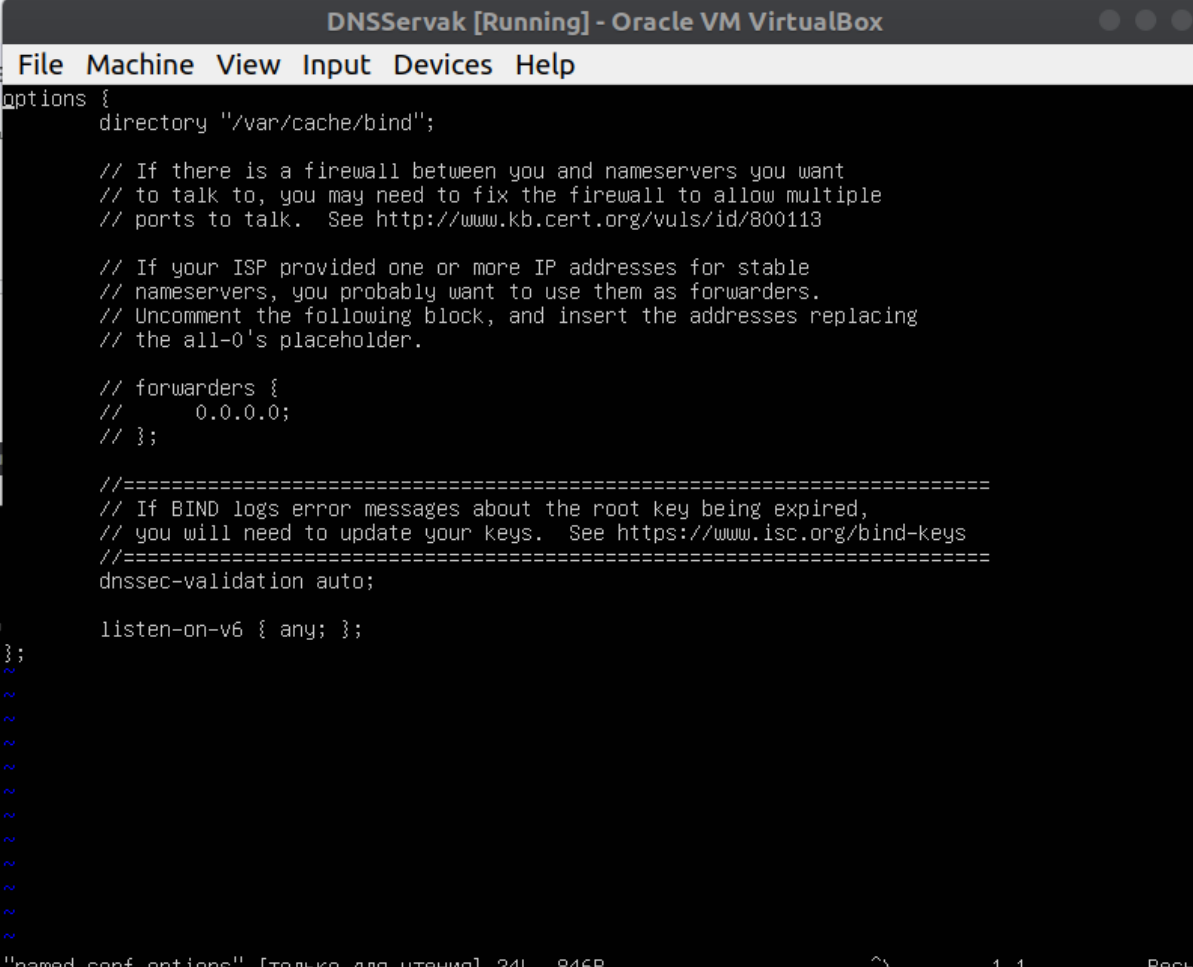
zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

zone "nsu" {
    type master;
    file "/etc/bind/db.nsu";
};

~
~
"/etc/bind/named.conf.default-zones" [только для чтения] 34L, 554B  ^\ 1,1  Весь
```

Не помню, нужен ли был этот файл, но вставлю на всякий случай



```
DNSServak [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};

"named.conf.options" [только для чтения] 24L, 846B 1,1 Весь
```


Вот ток для настройки того, чтоб с ноута был доступ к сайтам.

[illegible]