[[](https://habr.com/ru/users/SyCraft/)](https://habr.com/ru/users/SyCraft/" \o "SyCraft)

[**SyCraft**](https://habr.com/ru/users/SyCraft/)27 июн 2019 в 15:35

Работа с кластером Proxmox: установка, настройка сети, ZFS, решение распространенных проблем

**13 мин**

**111K**

[Настройка Linux\*](https://habr.com/ru/hub/linux/)[\*nix\*](https://habr.com/ru/hub/nix/)[Виртуализация\*](https://habr.com/ru/hub/virtualization/)

Туториал

За последние несколько лет я очень тесно работаю с кластерами Proxmox: многим клиентам требуется своя собственная инфраструктура, где они могут развивать свой проект. Именно поэтому я могу рассказать про самые распространенные ошибки и проблемы, с которыми также можете столкнуться и вы. Помимо этого мы конечно же настроим кластер из трех нод с нуля.  
  
  
Proxmox кластер может состоять из двух и более серверов. Максимальное количество нод в кластере равняется 32 штукам. Наш собственный кластер будет состоять из трех нод на мультикасте (в статье я также опишу, как поднять кластер на уникасте — это важно, если вы базируете свою кластерную инфраструктуру на Hetzner или OVH, например). Коротко говоря, мультикаст позволяет осуществлять передачу данных одновременно на несколько нод. При мультикасте мы можем не задумываться о количестве нод в кластере (ориентируясь на ограничения выше).  
  
Сам кластер строится на внутренней сети (важно, чтобы IP адреса были в одной подсети), у тех же Hetzner и OVH есть возможность объединять в кластер ноды в разных датацентрах с помощью технологии Virtual Switch (Hetzner) и vRack (OVH) — о Virtual Switch мы также поговорим в статье. Если ваш хостинг-провайдер не имеет похожие технологии в работе, то вы можете использовать OVS (Open Virtual Switch), которая нативно поддерживается Proxmox, или использовать VPN. Однако, я рекомендую в данном случае использовать именно юникаст с небольшим количеством нод — часто возникают ситуации, где кластер просто “разваливается” на основе такой сетевой инфраструктуры и его приходится восстанавливать. Поэтому я стараюсь использовать именно OVH и Hetzner в работе — подобных инцидентов наблюдал в меньшем количестве, но в первую очередь изучайте хостинг-провайдера, у которого будете размещаться: есть ли у него альтернативная технология, какие решения он предлагает, поддерживает ли мультикаст и так далее.

Установка Proxmox

Proxmox может быть установлен двумя способами: ISO-инсталлятор и установка через shell. Мы выбираем второй способ, поэтому установите Debian на сервер.  
  
Перейдем непосредственно к установке Proxmox на каждый сервер. Установка предельно простая и описана в официальной документации здесь.  
  
Добавим репозиторий Proxmox и ключ этого репозитория:

echo "deb http://download.proxmox.com/debian/pve stretch pve-no-subscription" > /etc/apt/sources.list.d/pve-install-repo.list

wget http://download.proxmox.com/debian/proxmox-ve-release-5.x.gpg -O /etc/apt/trusted.gpg.d/proxmox-ve-release-5.x.gpg

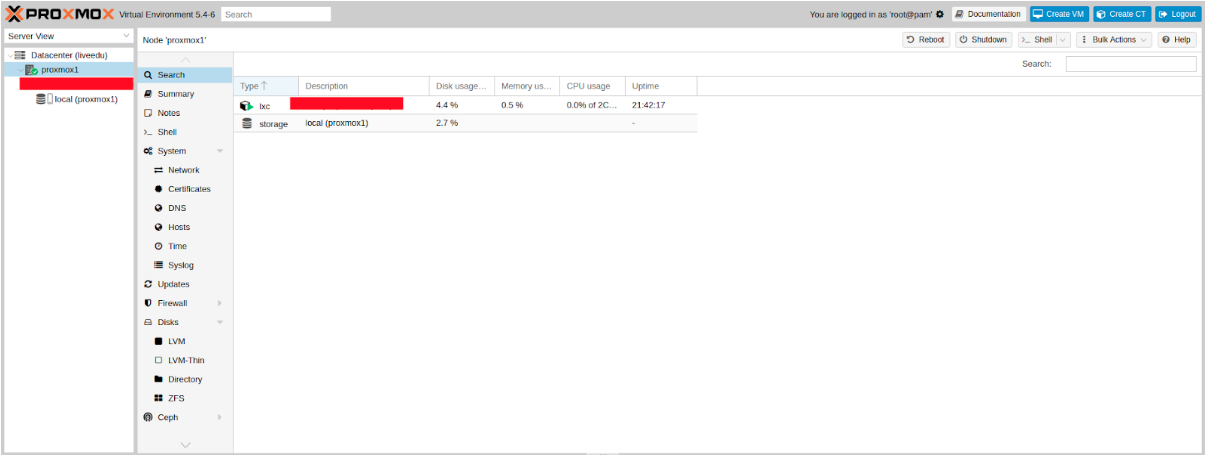
chmod +r /etc/apt/trusted.gpg.d/proxmox-ve-release-5.x.gpg # optional, if you have a changed default umask

Обновляем репозитории и саму систему:

apt update && apt dist-upgrade

После успешного обновления установим необходимые пакеты Proxmox:

apt install proxmox-ve postfix open-iscsi

**Заметка**: во время установки будет настраиваться Postfix и grub — одна из них может завершиться с ошибкой. Возможно, это будет вызвано тем, что хостнейм не резолвится по имени. Отредактируйте hosts записи и выполните apt-get update  
  
С этого момента мы можем авторизоваться в веб-интерфейс Proxmox по адресу https://<внешний-ip-адрес>:8006 (столкнетесь с недоверенным сертификатом во время подключения).  
  
  
**Изображение 1.**Веб-интерфейс ноды Proxmox

Установка Nginx и Let’s Encrypt сертификата

Мне не очень нравится ситуация с сертификатом и IP адресом, поэтому я предлагаю установить Nginx и настроить Let’s Encrypt сертификат. Установку Nginx описывать не буду, оставлю лишь важные файлы для работы Let’s encrypt сертификата:

/etc/nginx/snippets/letsencrypt.conf

Команда для выпуска SSL сертификата:

certbot certonly --agree-tos --email sos@livelinux.info --webroot -w /var/lib/letsencrypt/ -d proxmox1.domain.name

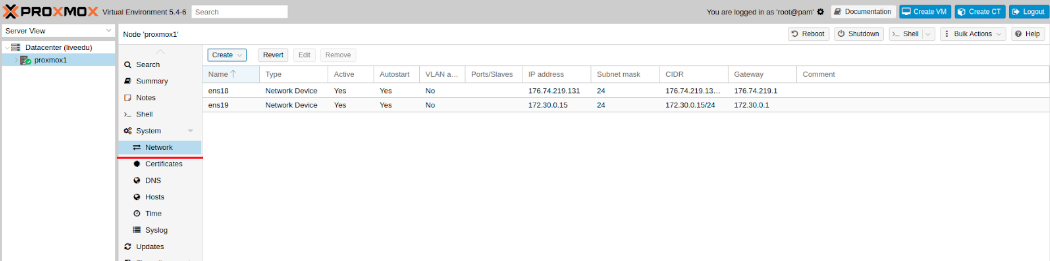
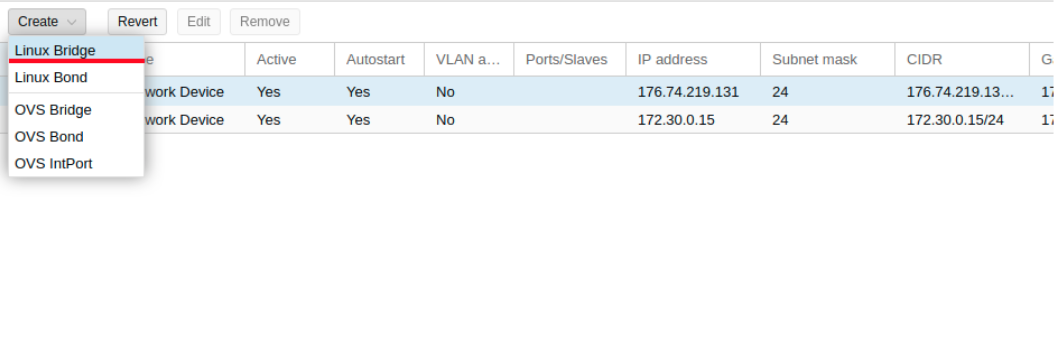
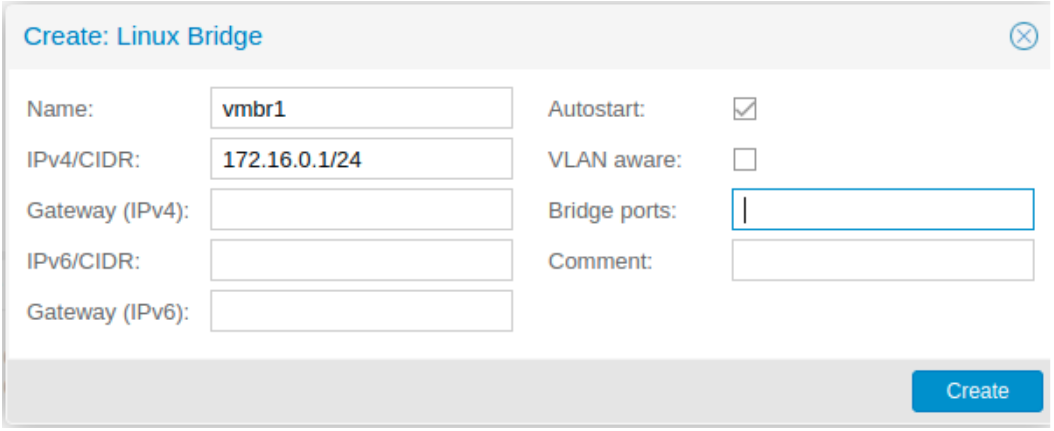
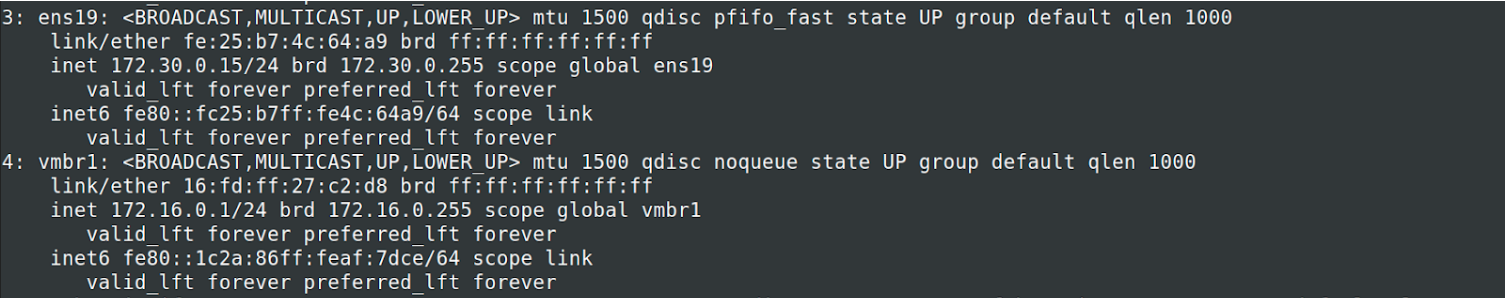
Конфигурация сайта в NGINX

Не забываем после установки SSL сертификата поставить его на автообновление через cron:

0 \*/12 \* \* \* /usr/bin/certbot -a \! -d /run/systemd/system && perl -e 'sleep int(rand(3600))' && certbot -q renew --renew-hook "systemctl reload nginx"

Отлично! Теперь мы можем обращаться к нашему домену по HTTPS.  
  
**Заметка**: чтобы отключить информационное окно о подписке, выполните данную команду:

sed -i.bak "s/data.status !== 'Active'/false/g" /usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js && systemctl restart pveproxy.service

**Сетевые настройки**  
  
Перед подключением в кластер настроим сетевые интерфейсы на гипервизоре. Стоит отметить, что настройка остальных нод ничем не отличается, кроме IP адресов и названия серверов, поэтому дублировать их настройку я не буду.  
  
Создадим сетевой мост для внутренней сети, чтобы наши виртуальные машины (в моем варианте будет LXC контейнер для удобства) во-первых, были подключены к внутренней сети гипервизора и могли взаимодействовать друг с другом. Во-вторых, чуть позже мы добавим мост для внешней сети, чтобы виртуальные машины имели свой внешний IP адрес. Соответственно, контейнеры будут на данный момент за NAT’ом у нас.  
  
Работать с сетевой конфигурацией Proxmox можно двумя способами: через веб-интерфейс или через конфигурационный файл /etc/network/interfaces. В первом варианте вам потребуется перезагрузка сервера (или можно просто переименовать файл interfaces.new в interfaces и сделать перезапуск networking сервиса через systemd). Если вы только начинаете настройку и еще нет виртуальных машин или LXC контейнеров, то желательно перезапускать гипервизор после изменений.  
  
Теперь создадим сетевой мост под названием vmbr1 во вкладке network в веб-панели Proxmox.  
  
  
**Изображение 2.** Сетевые интерфейсы ноды proxmox1  
  
  
**Изображение 3.** Создание сетевого моста  
  
  
**Изображение 4.** Настройка сетевой конфигурации vmbr1  
  
Настройка предельно простая — vmbr1 нам нужен для того, чтобы инстансы получали доступ в Интернет.  
  
Теперь перезапускаем наш гипервизор и проверяем, создался ли интерфейс:  
  
  
**Изображение 5.** Сетевой интерфейс vmbr1 в выводе команды ip a  
  
Заметьте: у меня уже есть интерфейс ens19 — это интерфейс с внутренней сетью, на основе ее будет создан кластер.  
  
Повторите данные этапы на остальных двух гипервизорах, после чего приступите к следующему шагу — подготовке кластера.  
  
Также важный этап сейчас заключается во включении форвардинга пакетов — без нее инстансы не будут получать доступ к внешней сети. Открываем файл sysctl.conf и изменяем значение параметра net.ipv4.ip\_forward на 1, после чего вводим следующую команду:

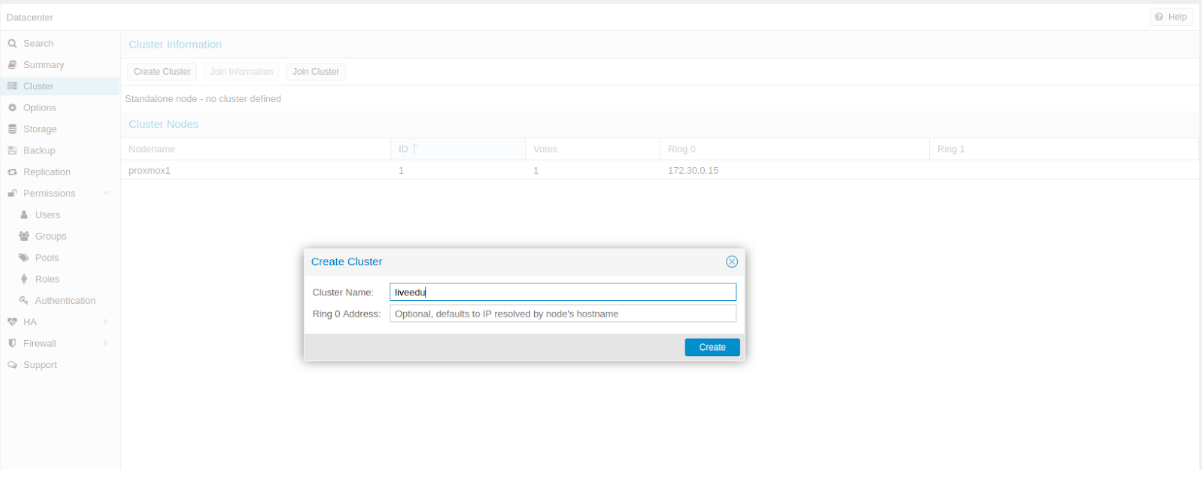
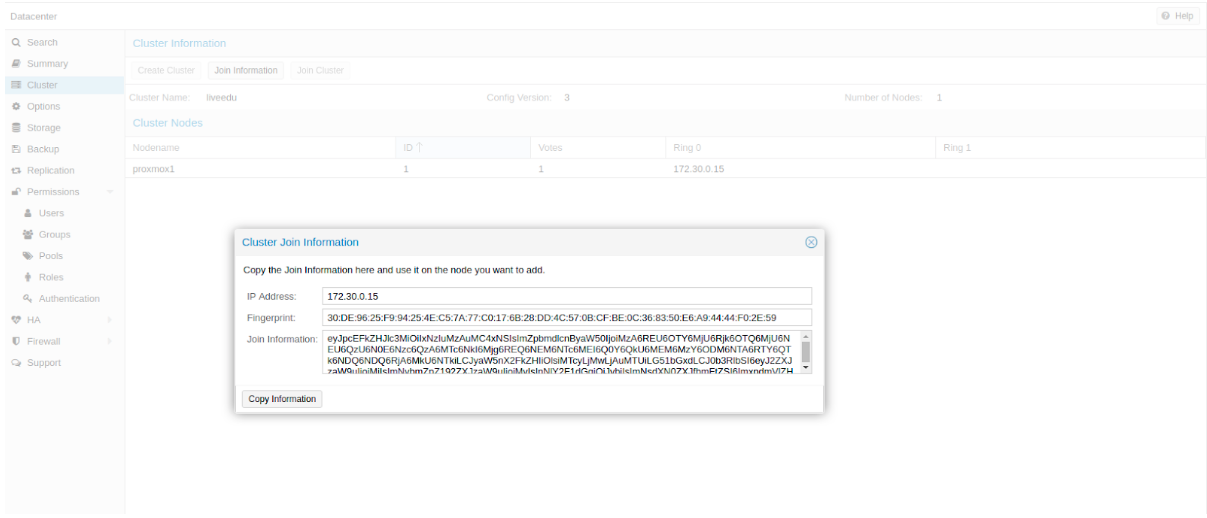
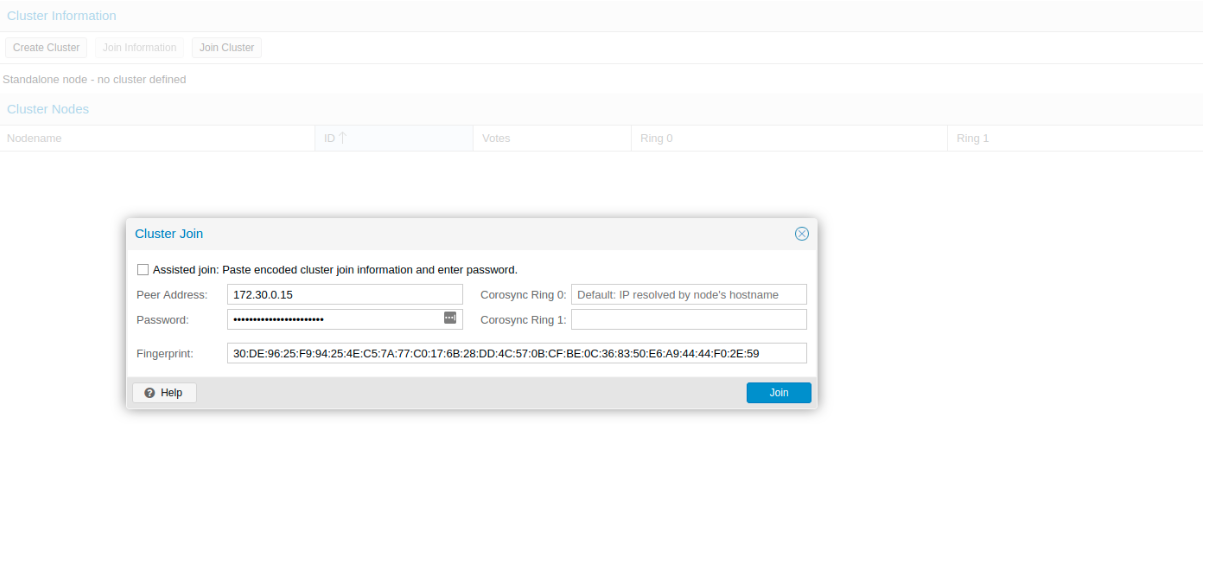
sysctl -p

В выводе вы должны увидеть директиву net.ipv4.ip\_forward (если не меняли ее до этого)  
  
**Настройка Proxmox кластера**  
  
Теперь перейдем непосредственно к кластеру. Каждая нода должна резолвить себя и другие ноды по внутренней сети, для этого требуется изменить значения в hosts записях следующих образом (на каждой ноде должна быть запись о других):

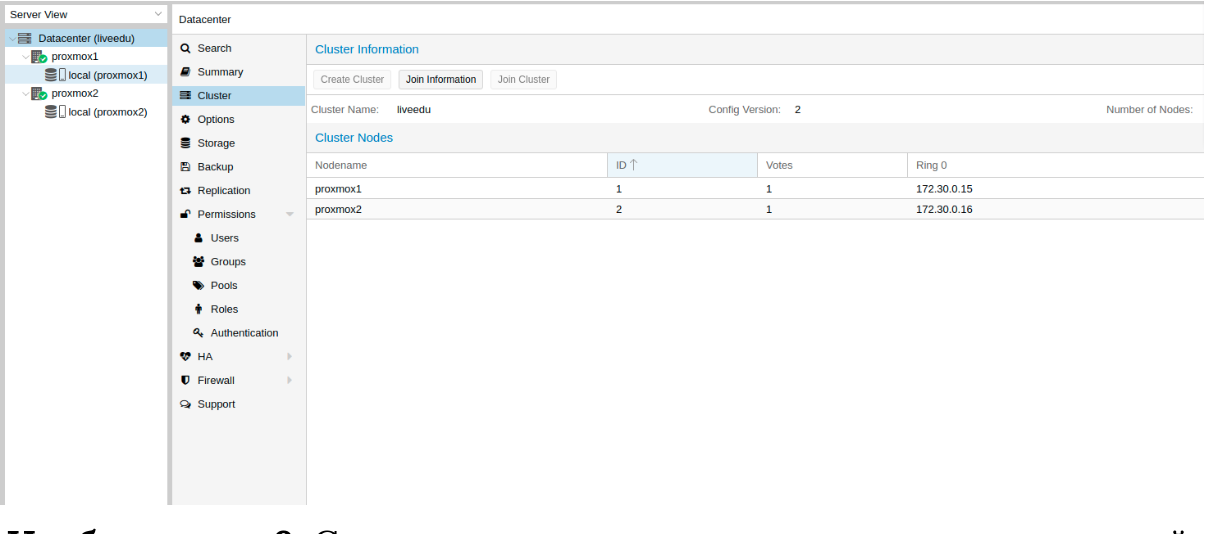
172.30.0.15 proxmox1.livelinux.info proxmox1

172.30.0.16 proxmox2.livelinux.info proxmox2

172.30.0.17 proxmox3.livelinux.info proxmox3

Также требуется добавить публичные ключи каждой ноды к остальным — это требуется для создания кластера.  
  
Создадим кластер через веб-панель:  
  
  
**Изображение 6.** Создание кластера через веб-интерфейс  
  
После создания кластера нам необходимо получить информацию о нем. Переходим в ту же вкладку кластера и нажимаем кнопку “Join Information”:  
  
  
**Изображение 7.** Информация о созданном кластере  
  
Данная информация пригодится нам во время присоединения второй и третьей ноды в кластер. Подключаемся к второй ноде и во вкладке Cluster нажимаем кнопку “Join Cluster”:  
  
  
**Изображение 8.** Подключение к кластеру ноды  
  
Разберем подробнее параметры для подключения:

1. **Peer Address:**IP адрес первого сервера (к тому, к которому мы подключаемся)
2. **Password:** пароль первого сервера
3. **Fingerprint:** данное значение мы получаем из информации о кластере

  
**Изображение 9.** Состояние кластера после подключения второй ноды  
  
Вторая нода успешно подключена! Однако, такое бывает не всегда. Если вы неправильно выполните шаги или возникнут сетевые проблемы, то присоединение в кластер будет провалено, а сам кластер будет “развален”. Лучшее решение — это отсоединить ноду от кластера, удалить на ней всю информацию о самом кластере, после чего сделать перезапуск сервера и проверить предыдущие шаги. Как же безопасно отключить ноду из кластера? Для начала удалим ее из кластера на первом сервере:

pvecm del proxmox2

После чего нода будет отсоединена от кластера. Теперь переходим на сломанную ноду и отключаем на ней следующие сервисы:

systemctl stop pvestatd.service

systemctl stop pvedaemon.service

systemctl stop pve-cluster.service

systemctl stop corosync

systemctl stop pve-cluster

Proxmox кластер хранит информацию о себе в sqlite базе, ее также необходимо очистить:

sqlite3 /var/lib/pve-cluster/config.db

delete from tree where name = 'corosync.conf';

.quit

Данные о коросинке успешно удалены. Удалим оставшиеся файлы, для этого необходимо запустить кластерную файловую систему в standalone режиме:

pmxcfs -l

rm /etc/pve/corosync.conf

rm /etc/corosync/\*

rm /var/lib/corosync/\*

rm -rf /etc/pve/nodes/\*

Перезапускаем сервер (это необязательно, но перестрахуемся: все сервисы по итогу должны быть запущены и работать корректно. Чтобы ничего не упустить делаем перезапуск). После включения мы получим пустую ноду без какой-либо информации о предыдущем кластере и можем начать подключение вновь.

Установка и настройка ZFS

ZFS — это файловая система, которая может использоваться совместно с Proxmox. С помощью нее можно позволить себе репликацию данных на другой гипервизор, миграцию виртуальной машины/LXC контейнера, доступ к LXC контейнеру с хост-системы и так далее. Установка ее достаточно простая, приступим к разбору. На моих серверах доступно три SSD диска, которые мы объединим в RAID массив.  
  
Добавляем репозитории:

nano /etc/apt/sources.list.d/stretch-backports.list

deb http://deb.debian.org/debian stretch-backports main contrib

deb-src http://deb.debian.org/debian stretch-backports main contrib

nano /etc/apt/preferences.d/90\_zfs

Package: libnvpair1linux libuutil1linux libzfs2linux libzpool2linux spl-dkms zfs-dkms zfs-test zfsutils-linux zfsutils-linux-dev zfs-zed

Pin: release n=stretch-backports

Pin-Priority: 990

Обновляем список пакетов:

apt update

Устанавливаем требуемые зависимости:

apt install --yes dpkg-dev linux-headers-$(uname -r) linux-image-amd64

Устанавливаем сам ZFS:

apt-get install zfs-dkms zfsutils-linux

Если вы в будущем получите ошибку fusermount: fuse device not found, try ‘modprobe fuse’ first, то выполните следующую команду:

modprobe fuse

Теперь приступим непосредственно к настройке. Для начала нам требуется отформатировать SSD и настроить их через parted:

Настройка /dev/sda

Аналогичные действия необходимо произвести и для других дисков. После того, как все диски подготовлены, приступаем к следующему шагу:  
  
zpool create -f -o ashift=12 rpool /dev/sda4 /dev/sdb4 /dev/sdc4  
  
Мы выбираем ashift=12 из соображений производительности — это рекомендация самого zfsonlinux, подробнее про это можно почитать в их вики: [github.com/zfsonlinux/zfs/wiki/faq#performance-considerations](https://github.com/zfsonlinux/zfs/wiki/faq#performance-considerations)  
  
Применим некоторые настройки для ZFS:

zfs set atime=off rpool

zfs set compression=lz4 rpool

zfs set dedup=off rpool

zfs set snapdir=visible rpool

zfs set primarycache=all rpool

zfs set aclinherit=passthrough rpool

zfs inherit acltype rpool

zfs get -r acltype rpool

zfs get all rpool | grep compressratio

Теперь нам надо рассчитать некоторые переменные для вычисления zfs\_arc\_max, я это делаю следующим образом:

mem =`free --giga | grep Mem | awk '{print $2}'`

partofmem=$(($mem/10))

echo $setzfscache > /sys/module/zfs/parameters/zfs\_arc\_max

grep c\_max /proc/spl/kstat/zfs/arcstats

zfs create rpool/data

cat > /etc/modprobe.d/zfs.conf << EOL

options zfs zfs\_arc\_max=$setzfscache

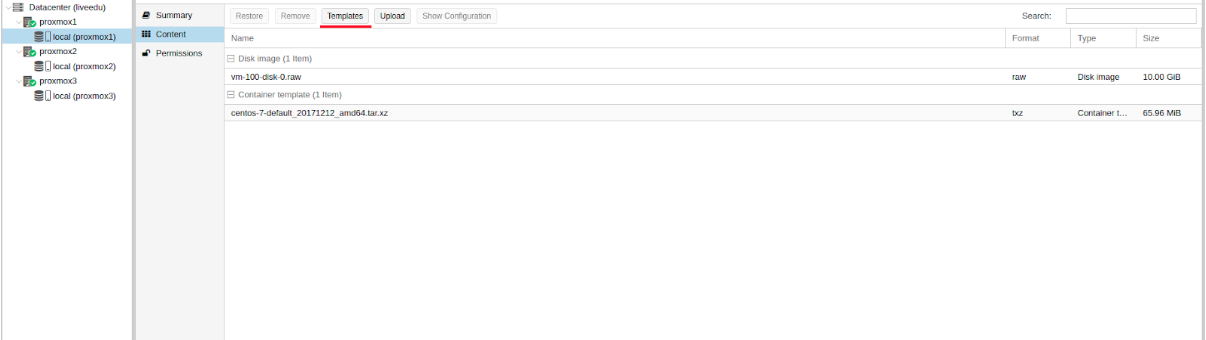
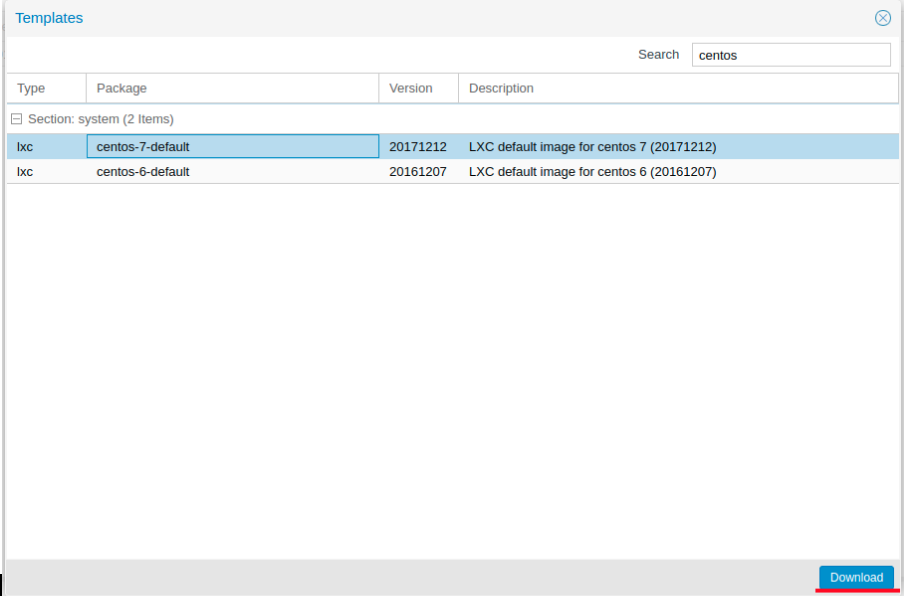
EOL

echo $setzfscache > /sys/module/zfs/parameters/zfs\_arc\_max

grep c\_max /proc/spl/kstat/zfs/arcstats

В данный момент пул успешно создан, также мы создали сабпул data. Проверить состояние вашего пула можно командой zpool status. Данное действие необходимо провести на всех гипервизорах, после чего приступить к следующему шагу.  
  
Теперь добавим ZFS в Proxmox. Переходим в настройки датацентра (именно его, а не отдельной ноды) в раздел «Storage», кликаем на кнопку «Add» и выбираем опцию «ZFS», после чего мы увидим следующие параметры:  
  
ID: Название стораджа. Я дал ему название local-zfs  
ZFS Pool: Мы создали rpool/data, его и добавляем сюда.  
Nodes: указываем все доступные ноды  
  
Данная команда создает новый пул с выбранными нами дисками. На каждом гипервизоре должен появится новый storage под названием local-zfs, после чего вы сможете смигрировать свои виртуальные машины с локального storage на ZFS.

Репликация инстансов на соседний гипервизор

В кластере Proxmox есть возможность репликации данных с одного гипервизора на другой: данный вариант позволяет осуществлять переключение инстанса с одного сервера на другой. Данные будут актуальны на момент последней синхронизации — ее время можно выставить при создании репликации (стандартно ставится 15 минут). Существует два способа миграции инстанса на другую ноду Proxmox: ручной и автоматический. Давайте рассмотрим в первую очередь ручной вариант, а в конце я предоставлю вам Python скрипт, который позволит создавать виртуальную машину на доступном гипервизоре при недоступности одного из гипервизоров.  
  
Для создания репликации необходимо перейти в веб-панель Proxmox и создать виртуальную машину или LXC контейнер. В предыдущих пунктах мы с вами настроили vmbr1 мост с NAT, что позволит нам выходить во внешнюю сеть. Я создам LXC контейнер с MySQL, Nginx и PHP-FPM с тестовым сайтом, чтобы проверить работу репликации. Ниже будет пошаговая инструкция.  
  
Загружаем подходящий темплейт (переходим в storage —> Content —> Templates), пример на скриншоте:  
  
  
**Изображение 10.** Local storage с шаблонами и образами ВМ  
  
Нажимаем кнопку “Templates” и загружаем необходимый нам шаблон LXC контейнера:  
  
  
**Изображение 11.** Выбор и загрузка шаблона  
  
Теперь мы можем использовать его при создании новых LXC контейнеров. Выбираем первый гипервизор и нажимаем кнопку “Create CT” в правом верхнем углу: мы увидим панель создания нового инстанса. Этапы установки достаточно просты и я приведу лишь конфигурационный файл данного LXC контейнера:

arch: amd64

cores: 3

memory: 2048

nameserver: 8.8.8.8

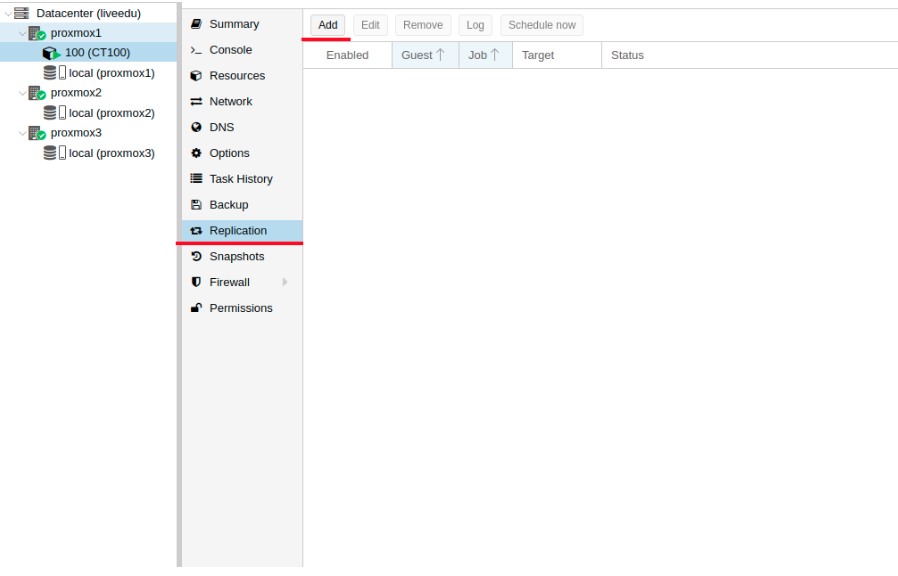
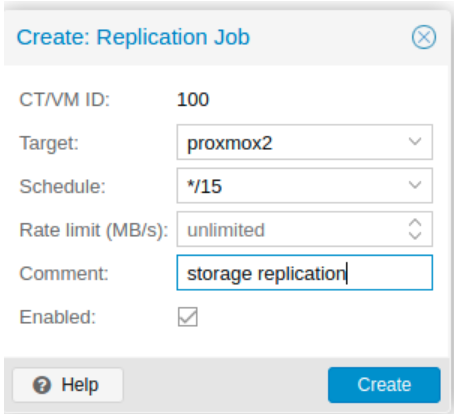
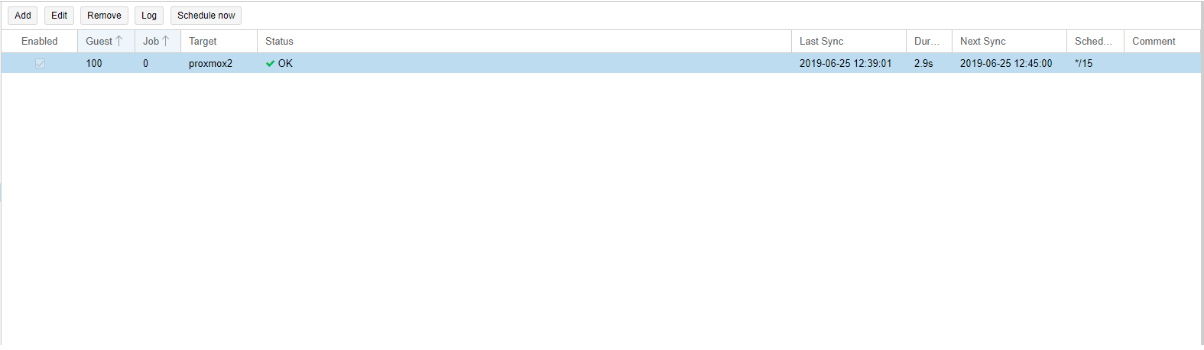
net0: name=eth0,bridge=vmbr1,firewall=1,gw=172.16.0.1,hwaddr=D6:60:C5:39:98:A0,ip=172.16.0.2/24,type=veth

ostype: centos

rootfs: local:100/vm-100-disk-1.raw,size=10G

swap: 512

unprivileged:

Контейнер успешно создан. К LXC контейнерам можно подключаться через команду pct enter , я также перед установкой добавил SSH ключ гипервизора, чтобы подключаться напрямую через SSH (в PCT есть небольшие проблемы с отображением терминала). Я подготовил сервер и установил туда все необходимые серверные приложения, теперь можно перейти к созданию репликации.  
  
Кликаем на LXC контейнер и переходим во вкладку “Replication”, где создаем параметр репликации с помощью кнопки “Add”:  
  
  
**Изображение 12.** Создание репликации в интерфейсе Proxmox  
  
  
**Изображение 13.**Окно создания Replication job  
  
Я создал задачу реплицировать контейнер на вторую ноду, как видно на следующем скриншоте репликация прошла успешно — обращайте внимание на поле “Status”, она оповещает о статусе репликации, также стоит обращать внимание на поле “Duration”, чтобы знать, сколько длится репликация данных.  
  
  
**Изображение 14.** Список синхронизаций ВМ  
  
Теперь попробуем смигрировать машину на вторую ноду с помощью кнопки “Migrate”  
  
Начнется миграция контейнера, лог можно просмотреть в списке задач — там будет наша миграция. После этого контейнер будет перемещен на вторую ноду.  
  
**Ошибка “Host Key Verification Failed”**  
  
Иногда при настройке кластера может возникать подобная проблема — она мешает мигрировать машины и создавать репликацию, что нивелирует преимущества кластерных решений. Для исправления этой ошибки удалите файл known\_hosts и подключитесь по SSH к конфликтной ноде:

/usr/bin/ssh -o 'HostKeyAlias=proxmox2' root@172.30.0.16

Примите Hostkey и попробуйте ввести эту команду, она должна подключить вас к серверу:

/usr/bin/ssh -o 'BatchMode=yes' -o 'HostKeyAlias=proxmox2' root@172.30.0.16

Особенности сетевых настроек на Hetzner

Переходим в панель Robot и нажимаем на кнопку “Virtual Switches”. На следующей странице вы увидите панель создания и управления интерфейсов Virtual Switch: для начала его необходимо создать, а после “подключить” выделенные сервера к нему. В поиске добавляем необходимые сервера для подключения — их не не нужно перезагружать, только придется подождать до 10-15 минут, когда подключение к Virtual Switch будет активно.  
  
После добавления серверов в Virtual Switch через веб-панель подключаемся к серверам и открываем конфигурационные файлы сетевых интерфейсов, где создаем новый сетевой интерфейс:

auto enp4s0.4000

iface enp4s0.4000 inet static

address 10.1.0.11/24

mtu 1400

vlan-raw-device enp4s0

Давайте разберем подробнее, что это такое. По своей сути — это VLAN, который подключается к единственному физическому интерфейсу под названием enp4s0 (он у вас может отличаться), с указанием номера VLAN — это номер Virtual Switch’a, который вы создавали в веб-панели Hetzner Robot. Адрес можете указать любой, главное, чтобы он был локальный.  
  
Отмечу, что конфигурировать enp4s0 следует как обычно, по сути он должен содержать внешний IP адрес, который был выдан вашему физическому серверу. Повторите данные шаги на других гипервизорах, после чего перезагрузите на них networking сервис, сделайте пинг до соседней ноды по IP адресу Virtual Switch. Если пинг прошел успешно, то вы успешно установили соединение между серверами по Virtual Switch.  
  
Я также приложу конфигурационный файл sysctl.conf, он понадобится, если у вас будут проблемы с форвардингом пакетом и прочими сетевыми параметрами:

net.ipv6.conf.all.disable\_ipv6=0

net.ipv6.conf.default.disable\_ipv6 = 0

net.ipv6.conf.all.forwarding=1

net.ipv4.conf.all.rp\_filter=1

net.ipv4.tcp\_syncookies=1

net.ipv4.ip\_forward=1

net.ipv4.conf.all.send\_redirects=0

**Добавление IPv4 подсети в Hetzner**  
  
Перед началом работ вам необходимо заказать подсеть в Hetzner, сделать это можно через панель Robot.  
  
Создадим сетевой мост с адресом, который будет из этой подсети. Пример конфигурации:

auto vmbr2

iface vmbr2 inet static

address ip-address

netmask 29

bridge-ports none

bridge-stp off

bridge-fd 0

Теперь переходим в настройки виртуальной машины в Proxmox и создаем новый сетевой интерфейс, который будет прикреплен к мосту vmbr2. Я использую LXC контейнер, его конфигурацию можно изменять сразу же в Proxmox. Итоговая конфигурация для Debian:

auto eth0

iface eth0 inet static

address ip-address

netmask 26

gateway bridge-address

Обратите внимание: я указал 26 маску, а не 29 — это требуется для того, чтобы сеть на виртуальной машине работала.  
  
**Добавление IPv4 адреса в Hetzner**  
  
Ситуация с одиночным IP адресом отличается — обычно Hetzner дает нам дополнительный адрес из подсети сервера. Это означает, что вместо vmbr2 нам требуется использоваться vmbr0, но на данный момент его у нас нет. Суть в том, что vmbr0 должен содержать IP адрес железного сервера (то есть использовать тот адрес, который использовал физический сетевой интерфейс enp2s0). Адрес необходимо переместить на vmbr0, для этого подойдет следующая конфигурация (советую заказать KVM, чтобы в случае чего возобновить работу сети):

auto enp2s0

iface enp2s0 inet manual

auto vmbr0

iface vmbr0 inet static

address ip-address

netmask 255.255.255.192

gateway ip-gateway

bridge-ports enp2s0

bridge-stp off

bridge-fd 0

Перезапустите сервер, если это возможно (если нет, перезапустите сервис networking), после чего проверьте сетевые интерфейсы через ip a:

2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast master vmbr0 state UP group default qlen 1000

link/ether 44:8a:5b:2c:30:c2 brd ff:ff:ff:ff:ff:ff

Как здесь видно, enp2s0 подключен к vmbr0 и не имеет IP адрес, так как он был переназначен на vmbr0.  
  
Теперь в настройках виртуальной машины добавляем сетевой интерфейс, который будет подключен к vmbr0. В качестве gateway укажите адрес, прикрепленный к vmbr0.

В завершении

Надеюсь, что данная статья пригодится вам, когда вы будете настраивать Proxmox кластер в Hetzner. Если позволит время, то я расширю статью и добавлю инструкцию для OVH — там тоже не все очевидно, как кажется на первый взгляд. Материал получился достаточно объемным, если найдете ошибки, то, пожалуйста, напишите в комментарии, я их исправлю. Всем спасибо за уделенное внимание.  
  
*Автор: Илья Андреев, под редакцией Алексея Жадан и команды «Лайв Линукс»*