

Курс «Системы виртуализации».

Методические указания по выполнению домашнего задания «Установка DRBD»

Автор курса: Павел Семенец

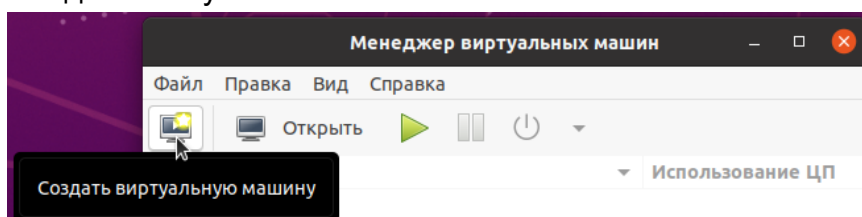
Автор методического пособия: Антон Трифонцов

1. Создание и настройка виртуальной машины (VM).

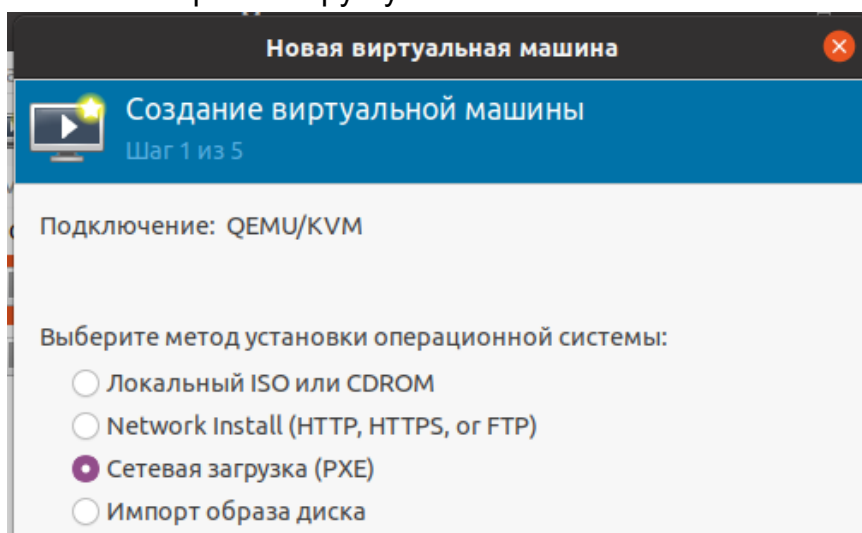
Создайте новую VM через Менеджер виртуальных машин (QEMU/KVM). Всего надо будет создать две VM.

Установка гипервизора KVM описана по [ссылке](#).

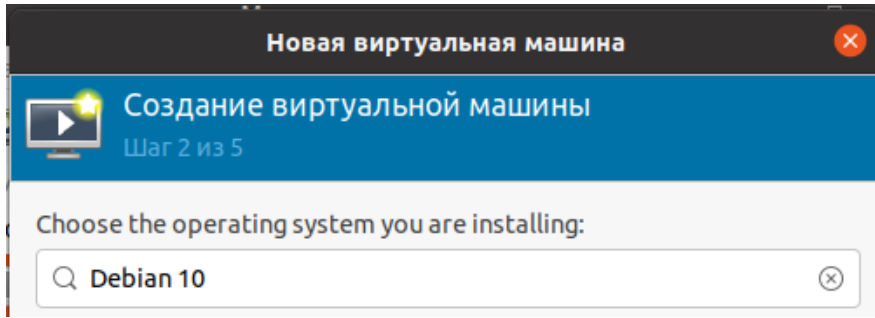
Создаем новую VM:



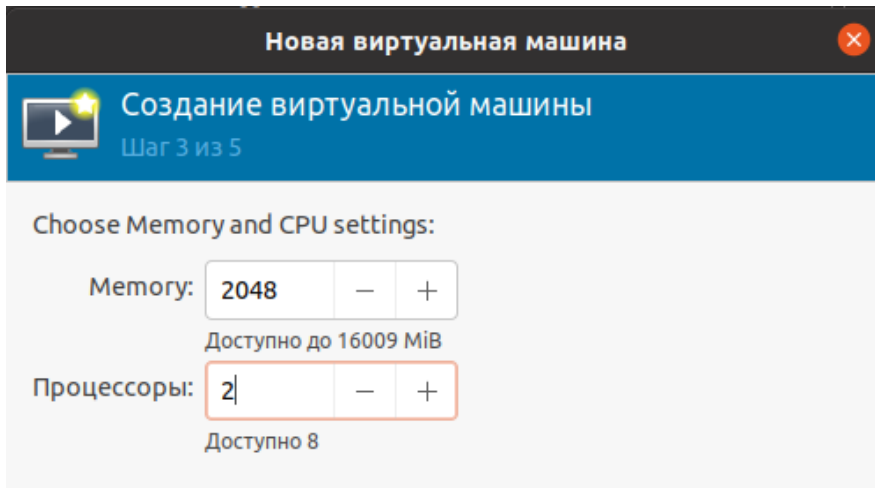
Шаг 1. Выбираем загрузку по сети



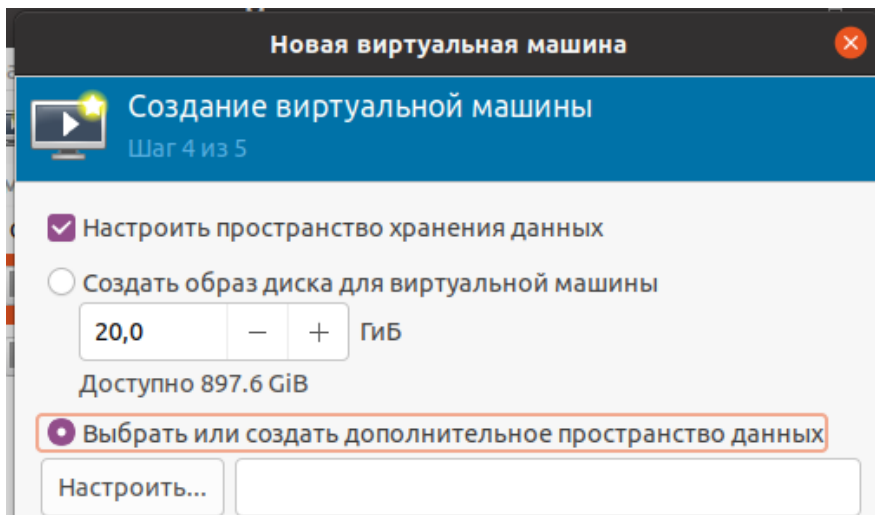
Шаг 2. Выбираем операционную систему Debian 10



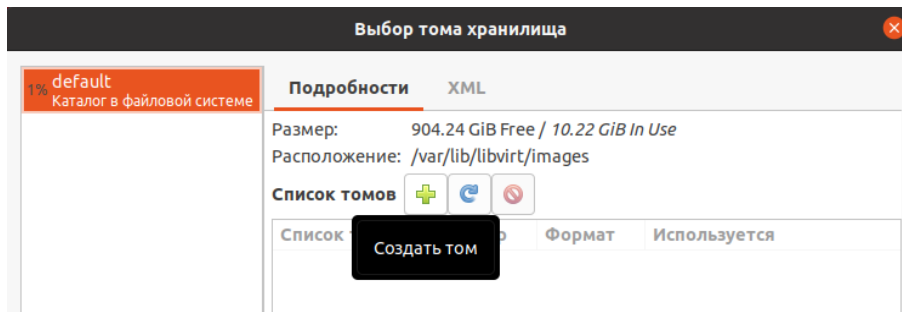
Шаг 3. Устанавливаем размер оперативной памяти и количество процессоров



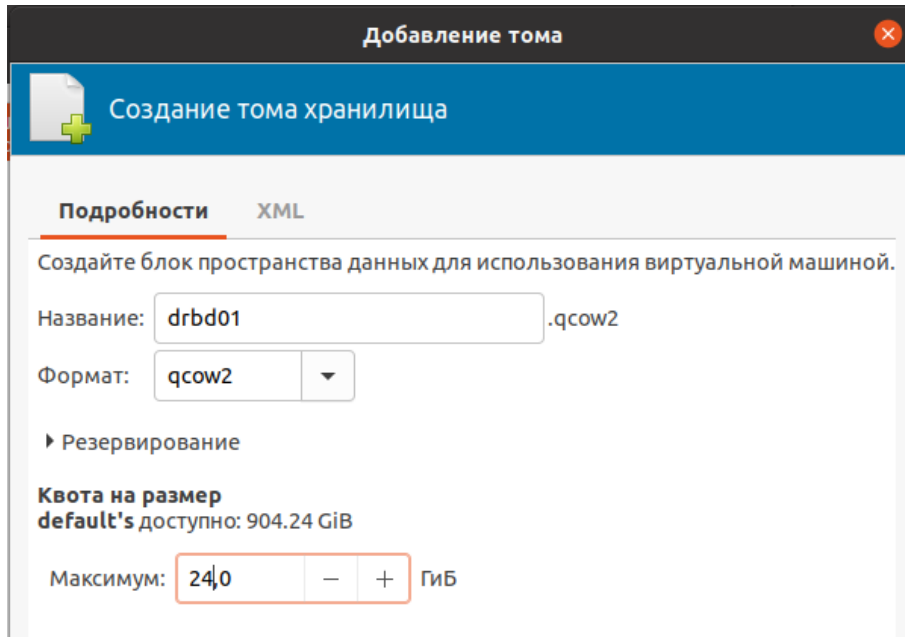
Шаг 4. Выбираем дополнительное пространство данных



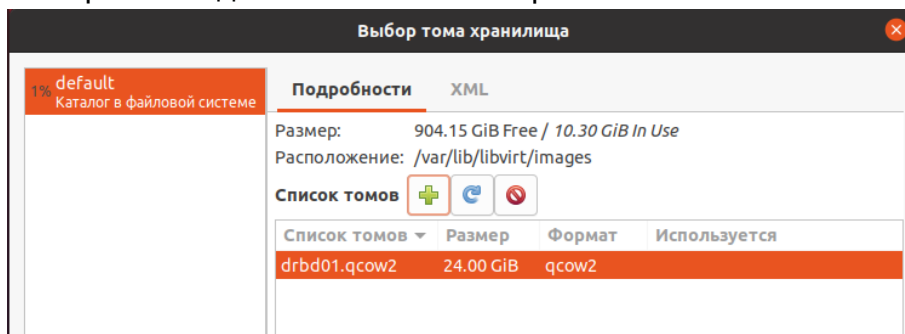
Создаем новый том



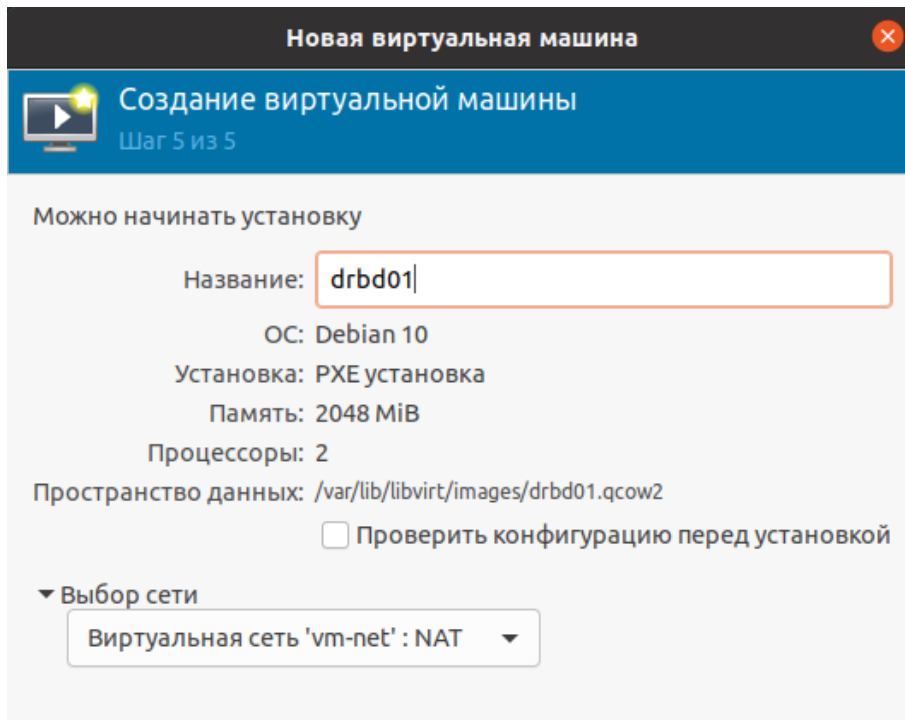
Указываем название тома drbd01 и размер диска 24 ГБ



Выбираем созданный том: drbd01.qcow2

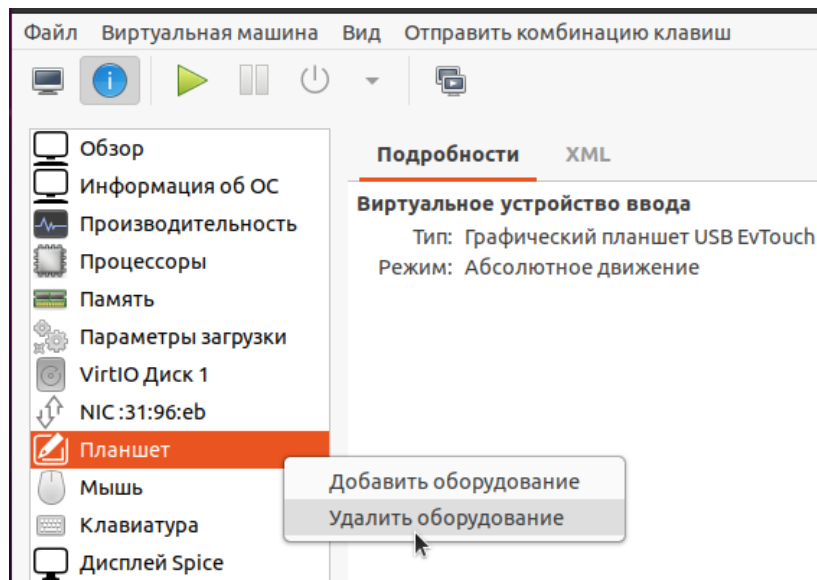


Шаг 5. Указываем название VM: drbd01



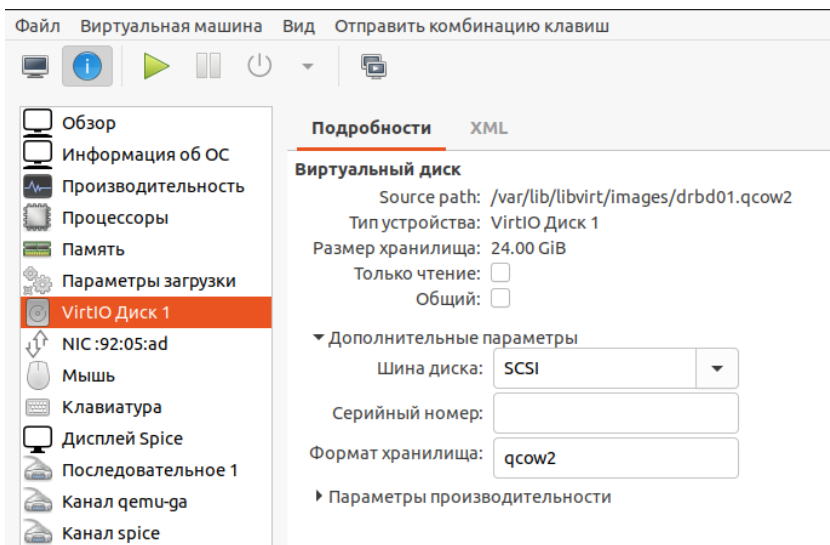
Создание VM закончено. Нажимаем кнопку Готово. Далее идет запуск VM. Выключаем VM принудительно.

Заходим в настройки VM. Удаляем неиспользуемое оборудование.

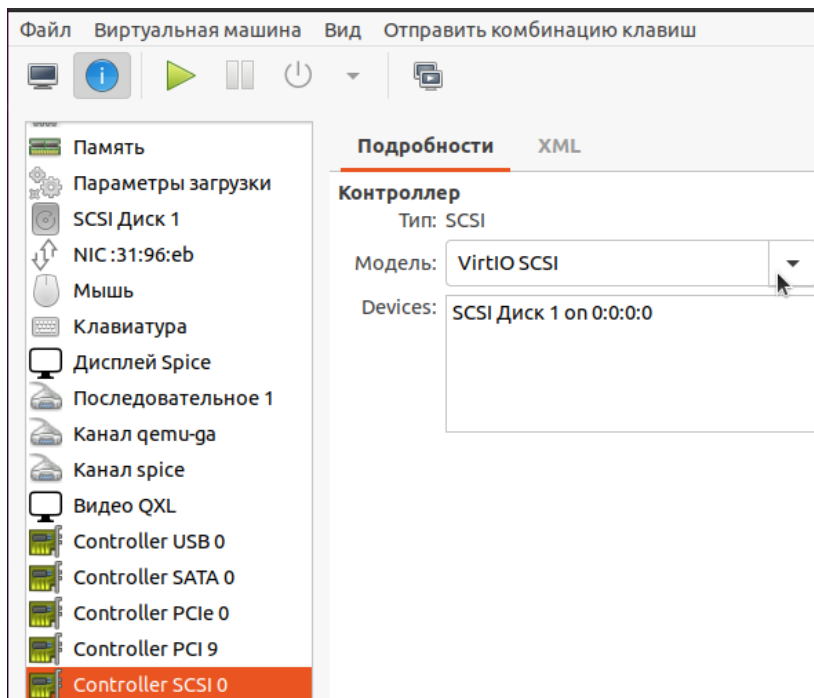




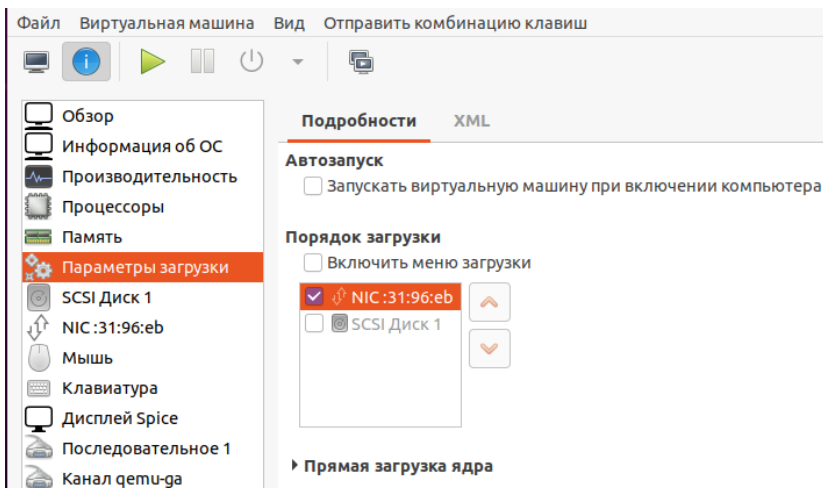
Для оборудования VirtIO Диск 1 устанавливаем шину SCSI. Нажимаем кнопку Применить.



Появится новое оборудование Controller SCSI 0. Модель контроллера установить: VirtIO SCSI (вместо Isilogic).



В Параметрах загрузки выставляем первым загрузку по сети (оборудование NIC).



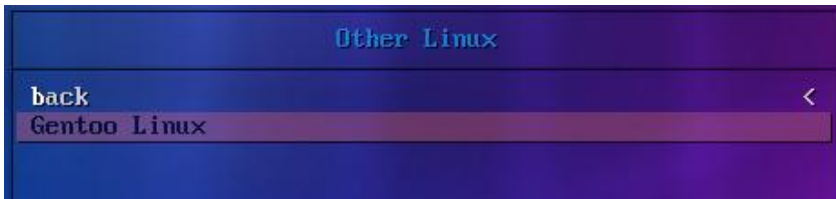
Запускаем VM.

2. Установка операционной системы.

После запуска VM, происходит загрузка по сети из образа oVirt-toolsSetup.iso (необходимо скачать заранее). Выбираем Other Linux



Затем Gentoo Linux



После загрузки ОС переходим к созданию разделов на диске.

Набираем в консоле: `# gdisk /dev/sda`

```
livecd ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.3
```

```
Partition table scan:
```

```
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present
```

```
Creating new GPT entries.
```

Далее команда `o` – удаляем все разделы (если таковые были).

Команда `n` – создаем новый раздел:

- первый раздел grub (Enter, Enter, +1M, ef02)

```
Command (? for help): n
Partition number (1-128, default 1):
First sector (34-50331614, default = 2048) or {+-}size{KMGTP}:
Last sector (2048-50331614, default = 50331614) or {+-}size{KMGTP}: +1M
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): ef02
Changed type of partition to 'BIOS boot partition'
```

- второй раздел swap (Enter, Enter, +1G, 8200)

```
Command (? for help): n
Partition number (2-128, default 2):
First sector (34-50331614, default = 4096) or {+-}size{KMGTP}:
Last sector (4096-50331614, default = 50331614) or {+-}size{KMGTP}: +1G
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300): 8200
Changed type of partition to 'Linux swap'
```

- третий раздел root (Enter, Enter, +8G, Enter)

```
Command (? for help): n
Partition number (3-128, default 3):
First sector (34-50331614, default = 1052672) or {+-}size{KMGTP}:
Last sector (1052672-50331614, default = 50331614) or {+-}size{KMGTP}: +8G
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
```

- четвертый раздел (Enter, Enter, Enter, Enter)

```
Command (? for help): n
Partition number (4-128, default 4):
First sector (34-50331614, default = 17829888) or {+-}size{KMGTP}:
Last sector (17829888-50331614, default = 50331614) or {+-}size{KMGTP}:
Current type is 'Linux filesystem'
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'
```

В результате должны получиться следующие разделы:

Number	Start (sector)	End (sector)	Size	Code	Name
1	2048	4095	1024.0 KiB	EF02	BIOS boot partition
2	4096	2101247	1024.0 MiB	8200	Linux swap
3	2101248	18878463	8.0 GiB	8300	Linux filesystem
4	18878464	50331614	15.0 GiB	8300	Linux filesystem

Далее записываем произведенные действия, подтверждаем и перезагружаемся.

```
Command (? for help): w

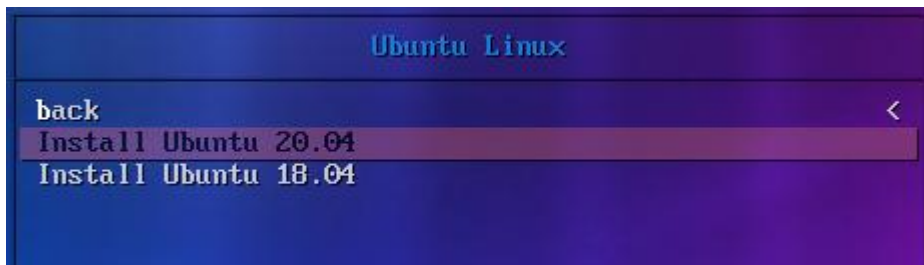
Final checks complete. About to write GPT data. THIS WILL OVERWRITE EXISTING
PARTITIONS!!

Do you want to proceed? (Y/N): Y
OK; writing new GUID partition table (GPT) to /dev/sda.
The operation has completed successfully.
livecd ~ # reboot_
```

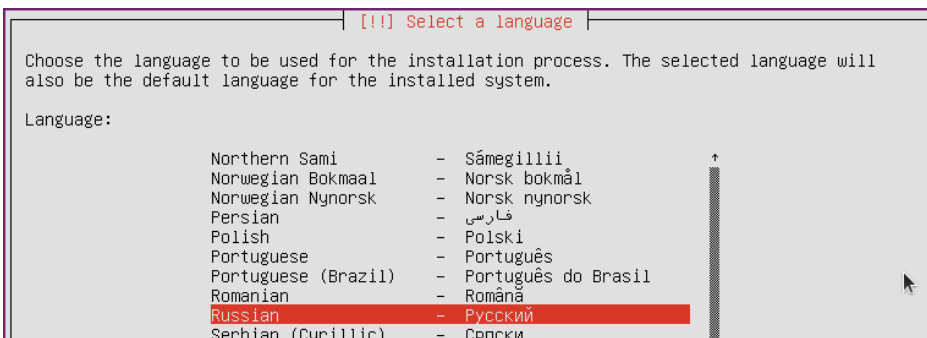
В меню загрузки выбираем Ubuntu Linux.



Затем Install Ubuntu 20.04



Устанавливаем Ubuntu 20.04



Вводим название компьютера: drbd01

[!] Настройка сети

Введите имя этого компьютера.

Имя компьютера -- это одно слово, которое идентифицирует вашу систему в сети. Если вы не знаете каким должно быть имя вашей системы, то посоветуйтесь с администратором вашей сети. Если вы устанавливаете вашу собственную домашнюю сеть, можете выбрать любое имя.

Имя компьютера:

drbd01

<Вернуться> <Продолжить>

[!] Выбор зеркала архива Ubuntu

Выберите зеркало архива Ubuntu, расположенное в ближайшей к вам сети. Имейте в виду, что зеркало в ближайшей стране (или даже в вашей собственной) не всегда будет наилучшим выбором.

Страна, в которой расположено зеркало архива Ubuntu:

Пуэрто-Рико
Республика Корея
Республика Македония
Реюньон
Российская Федерация

[!] Выбор зеркала архива Ubuntu

Выберите зеркало архива Ubuntu. Если вы не знаете, с каким зеркалом у вас наилучшая связь, выберите находящееся в вашей стране или регионе.

Обычно <код вашей страны>.archive.ubuntu.com является хорошим выбором.

Зеркало архива Ubuntu:

ru.archive.ubuntu.com

<Вернуться>

Выбор зеркала архива Ubuntu оставляем пустым:

[!] Выбор зеркала архива Ubuntu

Если вам необходимо использовать HTTP-прокси для доступа к внешнему миру, укажите в этом поле информацию о прокси. Если нет -- оставьте поле пустым.

Информацию о прокси следует вводить в стандартном виде
http://[пользователь][:пароль]@узел[:порт]/

Информация о HTTP-прокси (если прокси нет -- не заполняйте):

<Вернуться> <Продолжить>

Вводим полное имя пользователя:

[!] Настройка учётных записей пользователей и паролей

Будет создана учётная запись пользователя, которая будет использоваться вместо учётной записи суперпользователя (root) для выполнения всех действий, не связанных с администрированием.

Введите реальное имя этого пользователя. Эта информация будет использована в письмах в поле "От кого", посылаемых этим пользователем, а также всеми программами, которые показывают или используют реальное имя пользователя в своей работе. Ваше имя и фамилия вполне подходят.

Введите полное имя нового пользователя:

Tech

<Вернуться> <Продолжить>

И имя учетной записи:

[!] Настройка учётных записей пользователей и паролей

Выберите имя пользователя (учётную запись), под которым вы будете известны в системе. В качестве учётной записи может быть использовано ваше реальное имя. Учётная запись должна начинаться со строчной латинской буквы, за которой может следовать любое количество строчных латинских букв или цифр.

Имя вашей учётной записи:

tech

Вводим пароль для учетной записи:

[!] Настройка учётных записей пользователей и паролей

Хороший пароль представляет из себя смесь букв, цифр и знаков препинания, и должен периодически меняться.

Введите пароль для нового пользователя:

jjjjjjj

Show Password in Clear

И подтверждаем его.

[!] Настройка учётных записей пользователей и паролей

Проверка правильности ввода осуществляется путём повторного ввода пароля и сравнения результатов.

Введите пароль ещё раз:

jjjjjjj

Показывать вводимый пароль

[!] Настройка времени

Based on your present physical location, your time zone is Europe/Moscow.

If this is not correct, you may select from a full list of time zones instead.

Is this time zone correct?

Разметку дисков делаем вручную:

[!] Разметка дисков

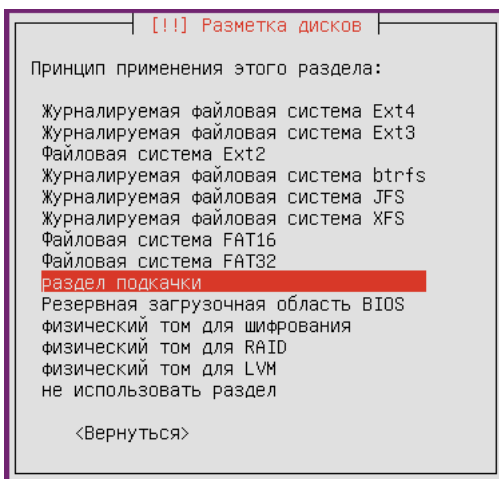
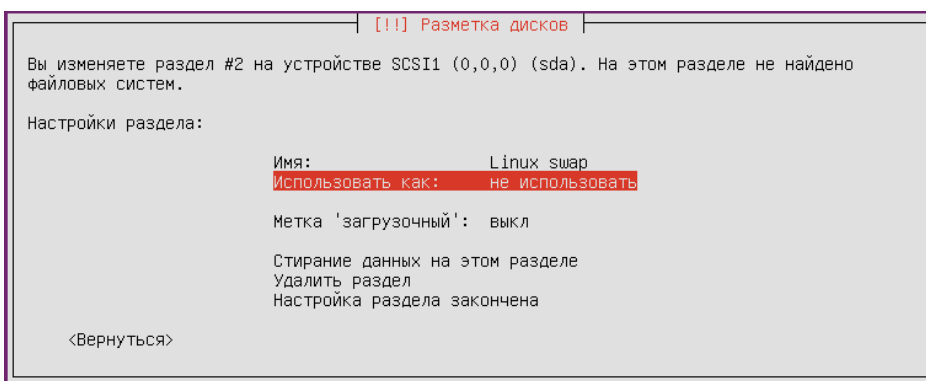
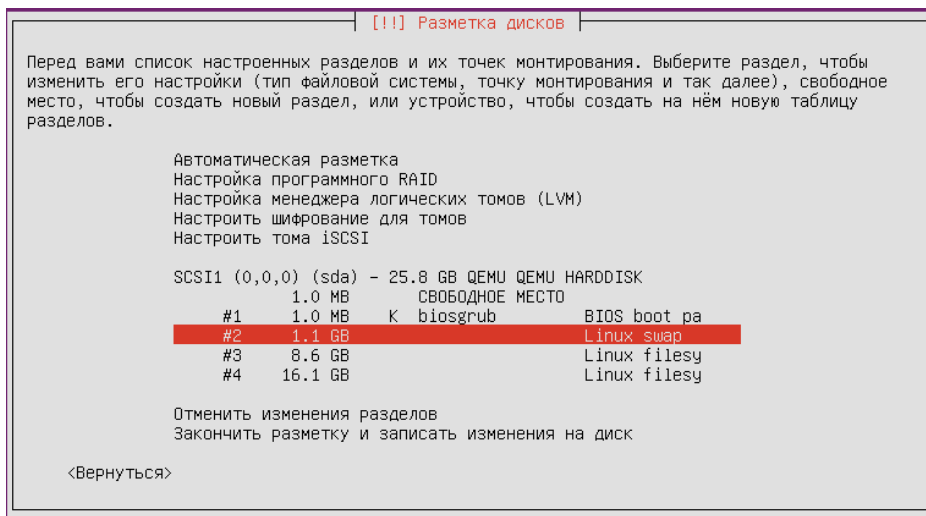
Программа установки может провести вас через процесс разметки диска (предлагая разные стандартные схемы) на разделы, либо это можно сделать вручную. Если выбрать использование инструмента управления разметкой, у вас всё равно будет возможность позже посмотреть и подправить результат.

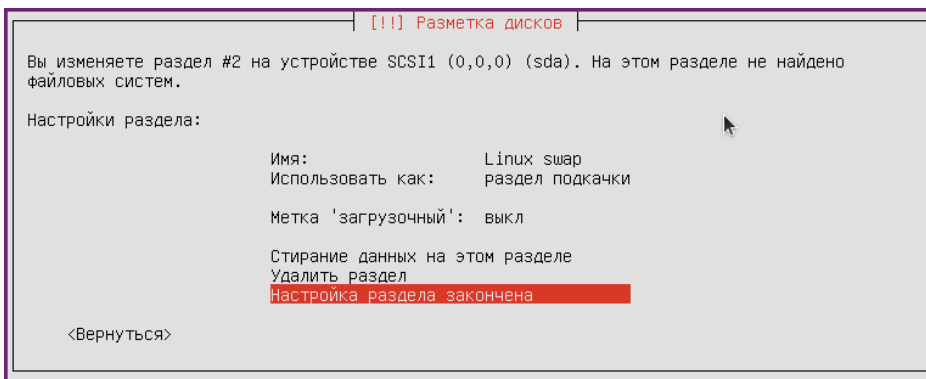
Если выбрать использование инструмента управления разметкой всего диска, то далее вас попросят указать нужный диск.

Метод разметки:

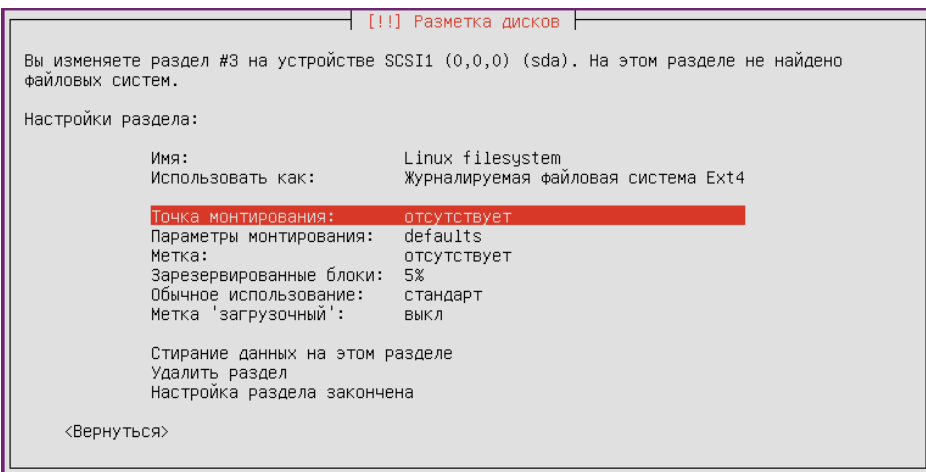
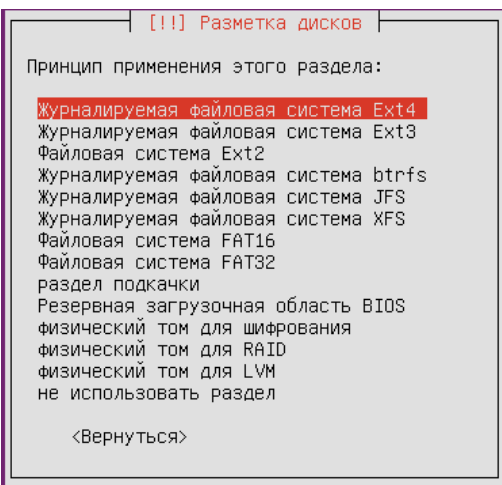
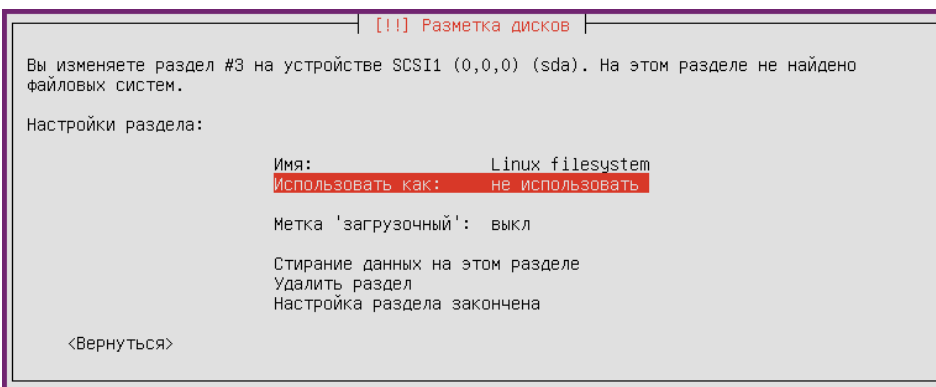
- Авто - использовать весь диск
- Авто - использовать наибольшее свободное место
- Авто - использовать весь диск и настроить LVM
- Авто - использовать весь диск с шифрованным LVM
- Вручную**

Первый и четвертый разделы не трогаем. Начинаем настройку со второго раздела:





Настраиваем третий раздел:



```

[!!!] Разметка дисков

Точка монтирования этого раздела:

/ -- корневая файловая система (root file system)
/boot -- статические файлы системного загрузчика
/home -- домашние каталоги пользователей
/tmp -- временные файлы
/usr -- статичные данные
/var -- изменяемые данные
/scr -- данные служб, предоставляемых системой
/opt -- дополнительные пакеты программного обеспечения
/usr/local -- локальные каталоги
Ввести вручную
Не монтировать этот раздел

<Вернуться>

```

```

[!!!] Разметка дисков

Вы изменяете раздел #3 на устройстве SCSI1 (0,0,0) (sda). На этом разделе не найдено
файловых систем.

Настройки раздела:

Имя: Linux filesystem
Использовать как: Журналируемая файловая система Ext4

Точка монтирования: /
Параметры монтирования: defaults
Метка: отсутствует
Зарезервированные блоки: 5%
Обычное использование: стандарт
Метка 'загрузочный': выкл

Стирание данных на этом разделе
Удалить раздел
Настройка раздела закончена

<Вернуться>

```

В параметрах монтирования отмечаем noatime и nodiratime

```

[!!!] Разметка дисков

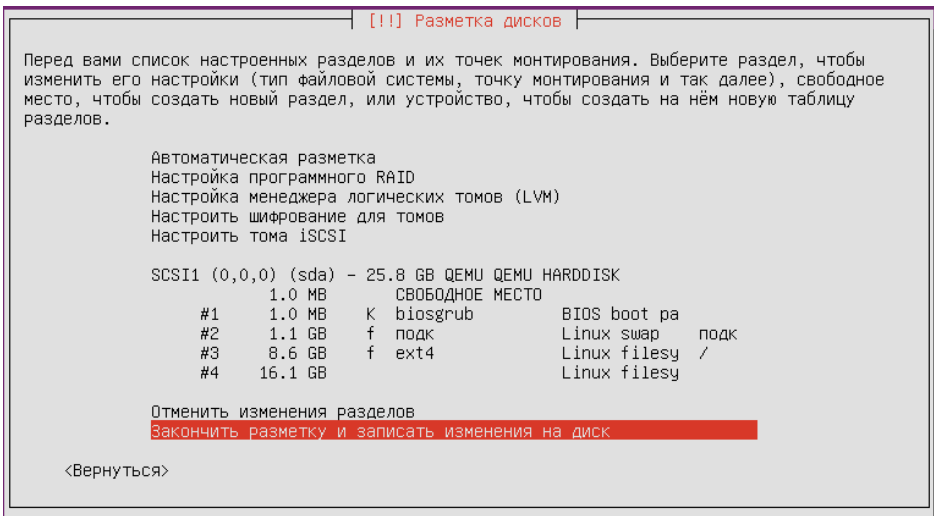
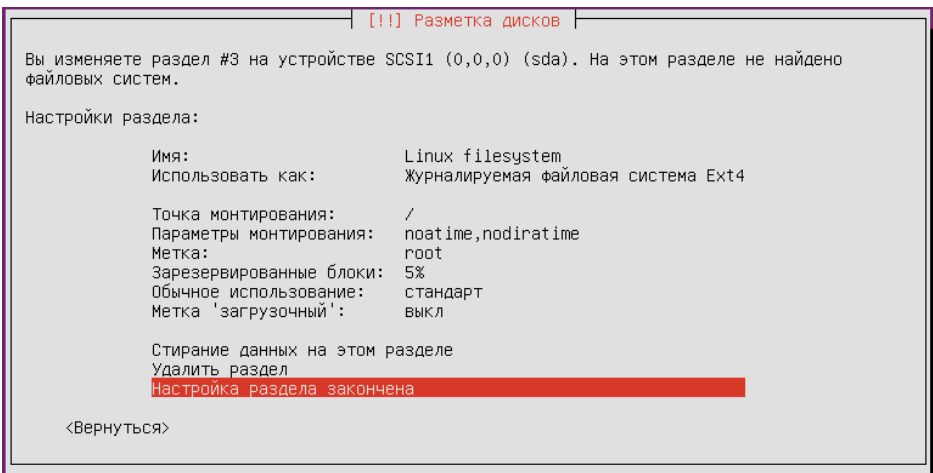
Параметры монтирования позволяют подстроить поведение файловой системы.

Параметры монтирования:

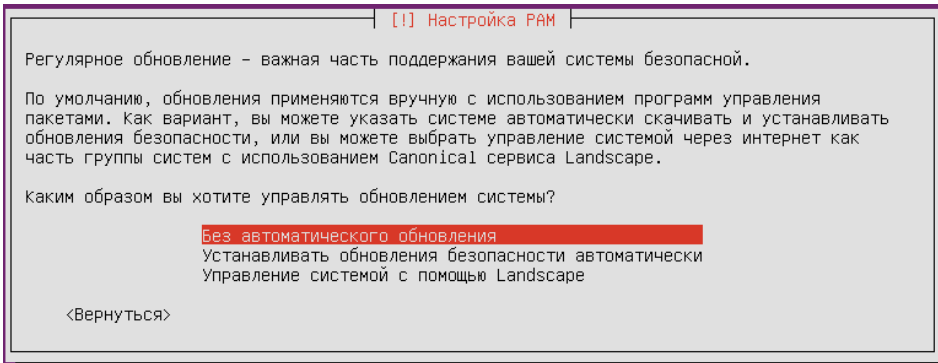
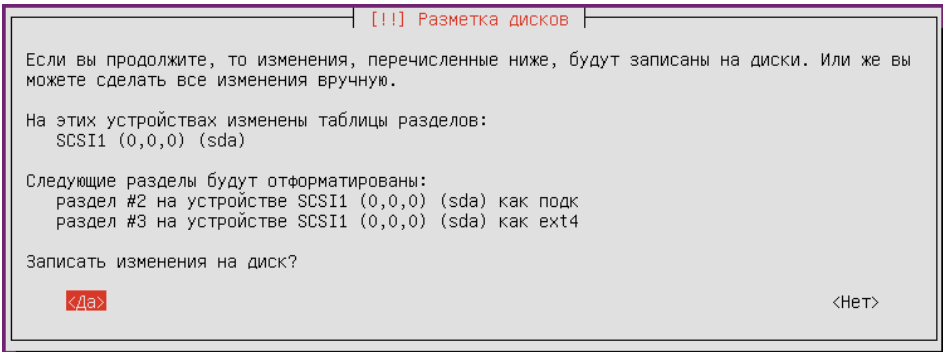
[ ] discard -- обрезать освободившиеся блоки на нижележащем блочном устройстве
[*] noatime -- не обновлять время доступа к inode
[*] nodiratime -- не обновлять время доступа к inode каталога
[ ] relatime -- обновлять время доступа к inode при изменениях
[ ] noexec -- не поддерживать символьные и блочные устройства
[ ] nosuid -- игнорировать биты SUID и SGID
[ ] noexec -- запретить выполнение любых программ
[ ] ro -- монтировать файловую систему в режиме 'только для чтения'
[ ] sync -- включить синхронный ввод-вывод в файловой системе
[ ] usrquota -- разрешить учёт дисковых квот пользователей
[ ] grpquota -- разрешить учёт дисковых квот групп
[ ] user_xattr -- включить расширенные пользовательские атрибуты

<Вернуться> <Продолжить>

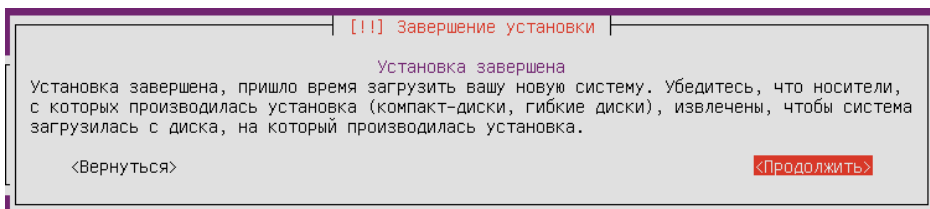
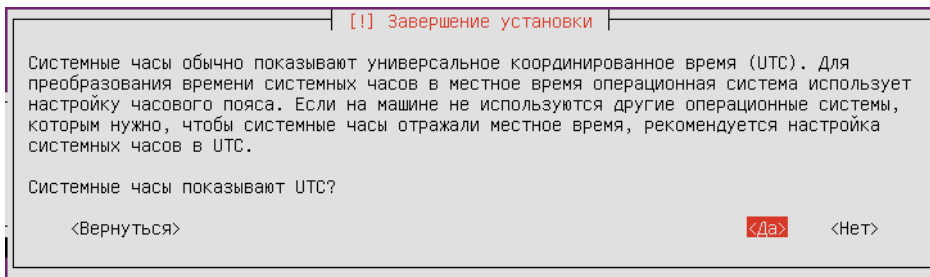
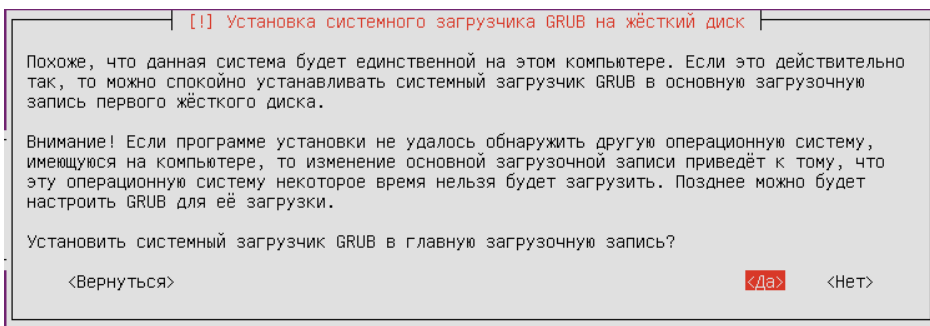
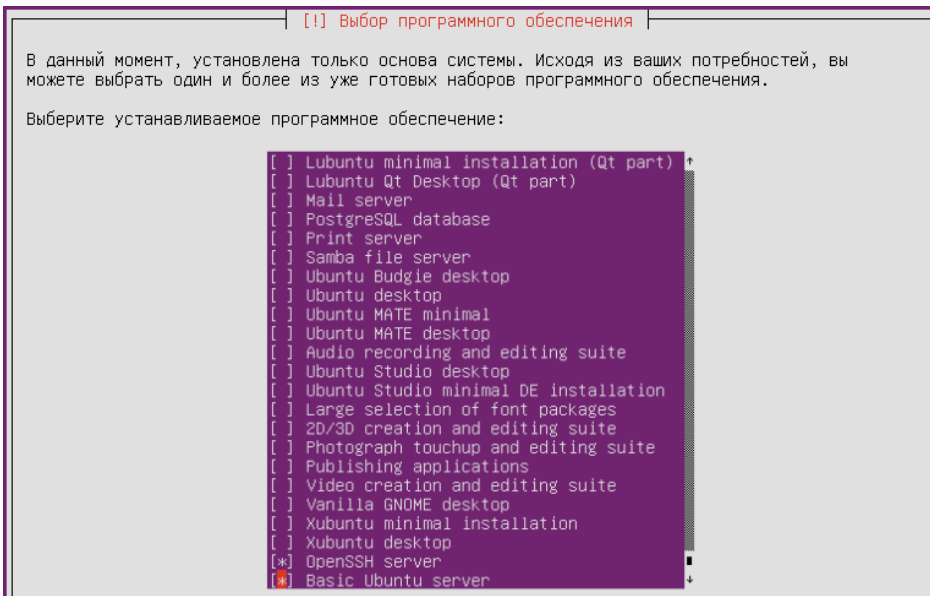
```



Подтверждаем изменения. Разметка дисков закончена.

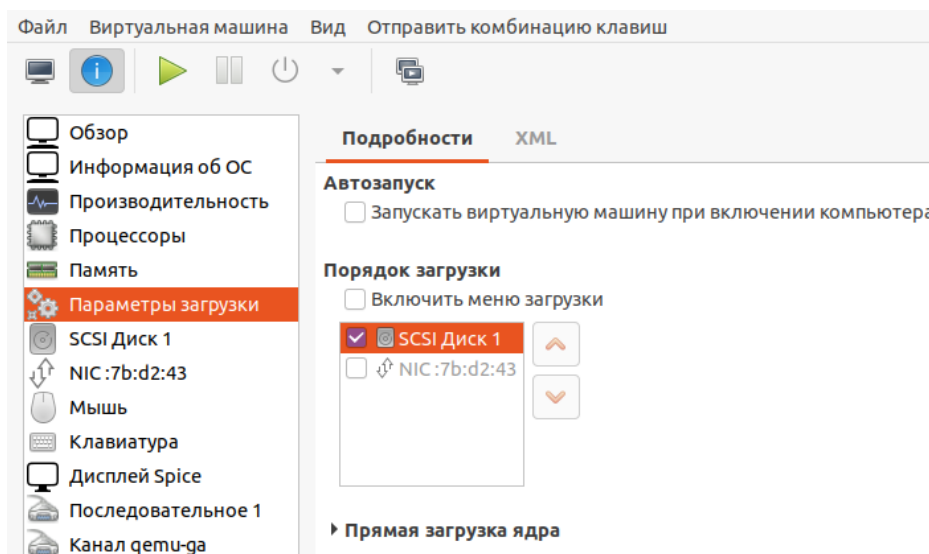


Отмечаем для установки OpenSSH server и Basic Ubuntu server:



Установка ОС завершена. После нажатия кнопки Продолжить, система перезагрузится. Принудительно выключаем VM.

Заходим в настройки VM. В Параметрах загрузки устанавливаем первым SCSI Диск 1.



Нажимаем кнопку Применить. Запускаем VM.

После загрузки должна появиться консоль терминала. Если экран остается черным – нажимаем Alt+F2. Авторизуемся. Установка ОС завершена.

По такому же принципу создаем еще одну VM: drbd02.

3. Настройка и запуск DRBD.

3.1 Настройка сети через netplan.

Выведем все наши сетевые интерфейсы, в том числе отключенные:

```
root@drbd01:~# ifconfig -a
```

Настроим сеть с помощью netplan на каждой VM:

```
root@drbd01:~# vim /etc/netplan/01-netcfg.yaml
```

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      addresses: [10.100.10.13/26]
      gateway4: 10.100.10.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
      dhcp4: no
```

Применим сохраненную конфигурацию:

```
root@drbd01:~# netplan generate
```

```
root@drbd01:~# netplan apply
```

Проверим доступность остальных ВМ:

```
root@drbd01:~# ping 10.100.10.14
```

```
root@drbd02:~# ping 10.100.10.13
```

3.2 Настройка беспарольного доступа на ВМ.

Создаем пару ключей на каждой ВМ командой ssh-keygen (Enter, Enter, Enter):

```
root@drbd01:~# ssh-keygen
```

```
root@drbd02:~# ssh-keygen
```

```
root@drbd01:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:N6Sx/eqieG8S7ezQZW9b/ns1DhEg0cZ7GRHy6q17JH8 root@drbd01
The key's randomart image is:
+---[RSA 3072]-----+
|
| .o.++
| +=. .
| o+.o .
| B.o .
| .S.B .
| ..o=. . ..|
| .+..+. + o|
| .o.= .E ..|
| ...*o= o .o+|
+----[SHA256]-----+
root@drbd01:~#
```

На host-машине создаем пару ключей из под учетной записи пользователя:

```
user@host:~/vm-net$ ssh-keygen
```

На host-машине должен быть установлен OpenSSH сервер:

```
user@host:~/vm-net$ sudo apt install openssh-server
```

На каждой ВМ смотрим содержимое файла id_rsa.pub:

```
root@drbd01:~# cat /root/.ssh/id_rsa.pub
```

```
root@drbd02:~# cat /root/.ssh/id_rsa.pub
```

```
root@drbd01:~# cat /root/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQDT6wNzCbPa7u0RnrH1fG3a59g7WBGp/R048bnfhBzCc545nVvhi3vutddwbb0n
EvRuocCFVIiXR3MojcDjs92cYfIppZZR0Q8qqA0XnEcE9hf/3yJafu50UZn2ssuE3SvPxBQxLv5242JmZQNG2QbM1KHLEJeqRhf
anXvFnp0V/PcPwNB62ezlqfwC8LRRZ1A3sLe9evQN6ibnfjKtw/Gch2aXJXNBxfuhoY297L8u0XSqMLcIQk4Pu9JM9HQ0AnYn9My
sy+x0UJKlrzVnZV/lbBsBd/BGAbXLbcYwrv8DlbEHaYNFCsChhs/bcTlvqFL8wdDlrxum6u4UvdZns0Z1VU0LXhjgiDPq/RJGX37
gZqf0kLL+tIVDQvUswJkTQd1ZMuIAPSKNudo9pKLSkmvq+LzX00QfPvpC1JcEC2LZfsKQxgNITc3Ct6fG5jYhnye7K50KR4S5gd5
TjZGE0g5j0/kRKYLZvA00TSXFfLKT7LJEKbgIYogx0kK9iZP= root@drbd01
```

На ВМ drbd01 создаем файл authorized_keys и добавляем в него ключи из drbd02 и host-машины:

```
root@drbd01:~# cat > /root/.ssh/authorized_keys
```

```

root@drbd01:~# cat > /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDGSC6c/KIXEYvLqSEn6zvUhzBYJMtGiuPVoDG1nogZ0PAF03cagsh4mhRkSh
2L79sJtGZkIR56dT1HShRuvCVGcRfZZtDJO/QzquH6rsaEEBcpIZXpavrbcnWvjGufA00B9Y3FkVdPxKfKcizFw7joQkfhGEPcc
UatB1TFz5CjX+2DuLkHGER8CvzAYK1ADLPB32YeZZ4ALXpQg+o9QStuE2LmJuULdym43XCU/mlch98/TvV1bM2sI1uXr07es/qdL
UVorGFu3Y3AB7mgtMKE2DE01P5h9JVCNA0PdghgKyhp6xWFR4kAJKEA/iYpQ8756zftGa/yyUQM/6zPAh9AzCbNImKvncVNQU+u5
aVZDZDo0M+Kccg54X+A+hqVGxGUiRrkC10Q9/LGFuen1cleA50BDIUdLqGtNcG4K5cpsvubkcvDdL/4DMdk+23BgMTp+K4jLCCFe
sTNFPTuyFh0ovIANBzgpKbHisL1f33RXkryNQgOJLVsiYlKZLQ0= anton@ux58
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGD6wNzCbPa7u0RnrH1fG3a59g7WBGp/R048bnfhBzCc545nVvhi3vutddWbb0n
EvRuocCFVIiXR3MojcDjs92cYfIppZROQ8qqA0XnEcE9hf/3yJafu50UZN2ssuE3SvPxBQxLv5242JMZQNG2QbMb1KHELJeqrRhF
anXvFnp0V/PcPwNB62ezlqfwC8LRRZ1A3sLe9evQN6ibnfjKtw/Gch2aXJXNBxfuhoY297L8u0XSqMLcIQk4Pu9JM9HQ0AnYn9My
sy+x0UJKlrzVnZv/lbBsBd/BGAbXLbcYwrv8DlbeHaYNFCScHhs/bcTlvqFL8wdDLrxum6u4UvdZns0Z1VU0LXHjgiDPq/RJGX37
gZqf0kLL+tIVDQvUswJkTQd1ZMuiAPSKNudo9pKLSkmvq+Lzx00QfPvpC1JcEC2LZfsKQxgNITc3Ct6fG5jYhnye7K50KR4S5g5
TjzGE0g5j0/kRKYLzVA00TSXFFLKT7LJEKbqgIYogxOkKK9iZPs= root@drbd01
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDUTiFvjEEZ+jFk4IYbZE+4Dcsc0EVoZZnrIuNC99PVQSGvRKiJ+Lz510djIVm/
p06poUlbtuuezAo/IoDjytZG0108rY/Gi/B4XcH+VqAOUTpLkAvCeJV0dZr6y3lnHehHzVm+lpPcTlHAX/HXq5LHyuJt76aPjFE
1S/v+8G90RnTf5hZXKH2CCcpY0vLsC2i21CZxERSqUrF7D7xrvvevE22pMQYLavTEPJWBADSaKp2BnDeuw5hsC0QD7bVRCA8STY4
7Q13Xi79EJ2CkVxRG/Bqtr57yV0ug03dq90xwjB4BET3fj0UgnRw3uyI4vBa52AzbZTdBFRlmWziMINCY+T0Y7q6UDFZ/d/iXfpu
4a0wfig2HyhXXLECGZdGgJshHfb9xwAZnzc6i14bczk2+72f63k47DLyWzTNkgiXvZPTDzv+h0Dq6PadkeAILWYyunNauM1m5GPY
1/vnQ9r/Q00rExdAKauNzQ3Yy0B6FFTRTJtM1Sjx2HbTXCxm+Y8= root@drbd02

```

Файл `authorized_keys` создан, копируем его на `drbd02`:

```

root@drbd02:~# scp 10.100.10.13:~/.ssh/authorized_keys
/root/.ssh/

```

На `host`-машине открываем файл `authorized_keys` и вставляем ключи через буфер обмена:

```

root@host:~# cat > /root/.ssh/authorized_keys

```

На `VM drbd01` отредактируем файл `ssh_config`. Необходимо раскомментировать параметр **StrictHostKeyChecking** и изменить его значение на **no**.

```

root@drbd01:~# vim /etc/ssh/ssh_config

```

```

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking no
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519

```

Сохраняем файл и копируем его на `drbd02`:

```

root@drbd02:~# scp 10.100.10.13:/etc/ssh/ssh_config /etc/ssh/

```

Отредактируем файл /etc/hosts:

```
root@drbd01:~# vim /etc/hosts
```

```
127.0.0.1    localhost
127.0.1.1    ubuntu

10.100.10.13 drbd01
10.100.10.14 drbd02
```

Сохраняем файл и копируем его на drbd02:

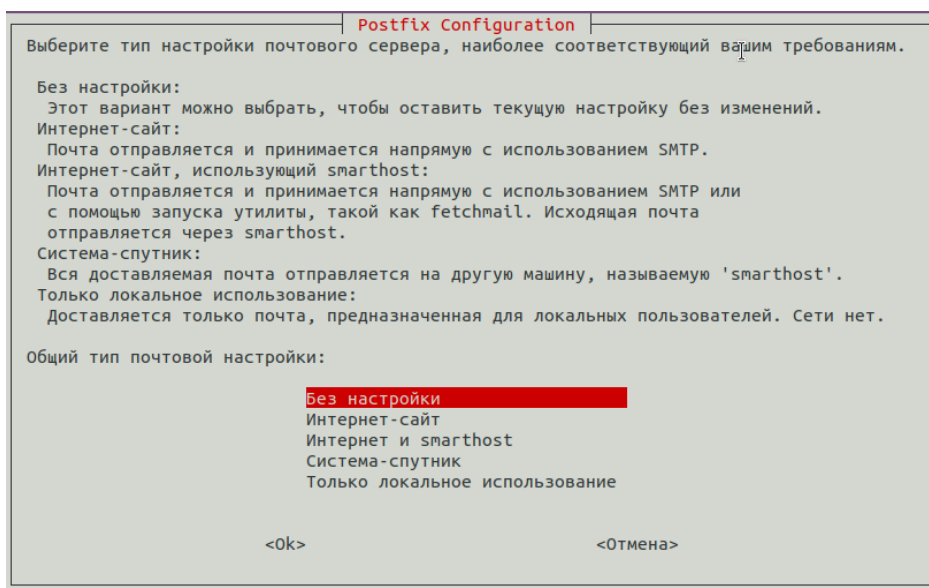
```
root@drbd02:~# scp 10.100.10.13:/etc/hosts /etc/
```

3.3 Установка пакетов DRBD и Pacemaker.

Установим следующие пакеты:

```
root@drbd01:~# apt install drbd-utils pacemaker crmsh
```

```
root@drbd02:~# apt install drbd-utils pacemaker crmsh
```



Проверим, запущен ли сервис drbd (должен быть выключен):

```
root@drbd01:~# systemctl status drbd
```

```
root@drbd02:~# systemctl status drbd
```

Останавливаем, если запущен:

```
root@drbd01:~# systemctl stop drbd
```

```
root@drbd02:~# systemctl stop drbd
```

Убираем из автозагрузки:

```
root@drbd01:~# systemctl disable drbd
```

```
root@drbd02:~# systemctl disable drbd
```

Проверим наличие скрипта запуска DRBD:

```
root@drbd01:~# ls -la /usr/lib/ocf/resource.d/linbit/
```

```
root@drbd01:~# ls -la /usr/lib/ocf/resource.d/linbit/
итого 64
drwxr-xr-x 2 root root 4096 ноя 27 02:30 .
drwxr-xr-x 5 root root 4096 ноя 27 02:30 ..
-rwxr-xr-x 1 root root 44519 мар 21 2020 drbd
-rw-r--r-- 1 root root 10464 мар 21 2020 drbd.shellfuncs.sh
root@drbd01:~#
```

Отредактируем файл общей конфигурации DRBD:

```
root@drbd01:~# vim /etc/drbd.d/global_common.conf
```

```
global {
    usage-count no;
}

common {
    syncer {
        rate 50M;
    }
    handlers {
    }

    startup {
        wfc-timeout 15;
        degr-wfc-timeout 60;
    }

    options {
    }

    disk {
    }

    net {
        cram-hmac-alg sha1;
        shared-secret "dY5t07M5eyT9X8w"; # пароль
        придумать свой
    }
}
```

Сохраняем файл и копируем его на drbd02:

```
root@drbd02:~# scp 10.100.10.13:/etc/drbd.d/global_common.conf
/etc/drbd.d/
```

Далее создадим файл конфигурации ресурсов DRBD:

```
root@drbd01:~# vim /etc/drbd.d/r0.res
```

```
resource r0 {
    protocol C;
    device /dev/drbd0;
    disk /dev/sda4;
    meta-disk internal;
    on drbd01 {
        address 10.100.10.13:7788;
    }
    on drbd02 {
        address 10.100.10.14:7788
    }
}
```

Сохраняем файл и копируем его на drbd02:

```
root@drbd02:~# scp 10.100.10.13:/etc/drbd.d/r0.res /etc/drbd.d/
```

Создадим само устройство DRBD:

```
root@drbd01:~# drbdadm create-md r0
```

```
root@drbd01:~# vim /etc/drbd.d/r0.res
root@drbd01:~# drbdadm create-md r0
initializing activity log
initializing bitmap (480 KB) to all zero
Writing meta data...
New drbd meta data block successfully created.
root@drbd01:~#
```

Устройство успешно создано.

Создадим устройство DRBD на второй ноде:

```
root@drbd02:~# drbdadm create-md r0
```

Запустим сервис DRBD на первой и второй ноде:

```
root@drbd01:~# systemctl start drbd
```

```
root@drbd02:~# systemctl start drbd
```

Определим роль второй ноды как primary:

```
root@drbd02:~# drbdadm -- --overwrite-data-of-peer primary all
```

```
root@drbd01:~# cat /proc/drbd
```

```
root@drbd01:~# cat /proc/drbd
version: 8.4.11 (api:1/proto:86-101)
srcversion: FC3433D849E3B88C1E7B55C
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
   ns:0 nr:1265216 dw:1265216 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:14460840
   [>.....] sync'ed: 8.1% (14120/15356)M
   finish: 0:06:44 speed: 35,764 (36,148) want: 45,560 K/sec
root@drbd01:~#
```

Запустился процесс синхронизации.

Останавливаем сервис DRBD на первой и второй ноде:

```
root@drbd01:~# systemctl stop drbd
```

```
root@drbd02:~# systemctl stop drbd
```

Настроим сервис Pacemaker.

Отредактируем файл конфигурации corosync.conf:

```
root@drbd01:~# vim /etc/corosync/corosync.conf
```

```
totem {
    version: 2

    cluster_name: debian

    crypto_cipher: none
    crypto_hash: none
}

logging {
    fileline: off
    to_stderr: yes
    to_logfile: yes
    logfile: /var/log/corosync/corosync.log
    to_syslog: yes
    debug: off
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

quorum {
    provider: corosync_votequorum
    expected_votes: 1
    two_node: 1
}

nodelist {

    node {
        name: drbd01
        nodeid: 1
        ring0_addr: 10.100.10.13
    }
    node {
        name: drbd02
        nodeid: 2
        ring0_addr: 10.100.10.14
    }
}
```

Настройка файла закончена, сохраняем.

Останавливаем сервисы:

```
root@drbd01:~# systemctl stop pacemaker
```

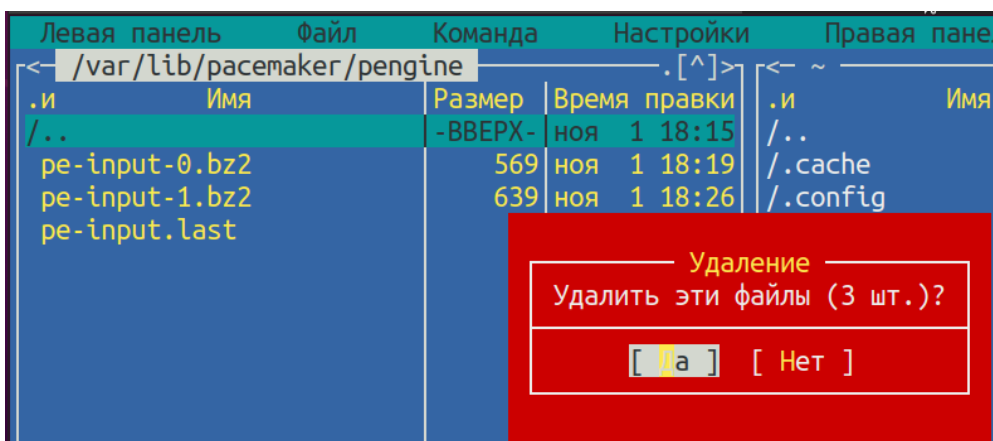
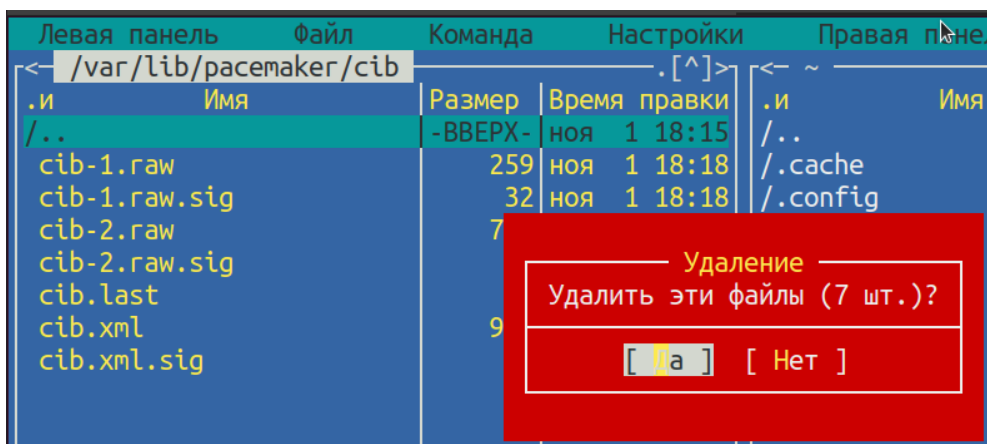
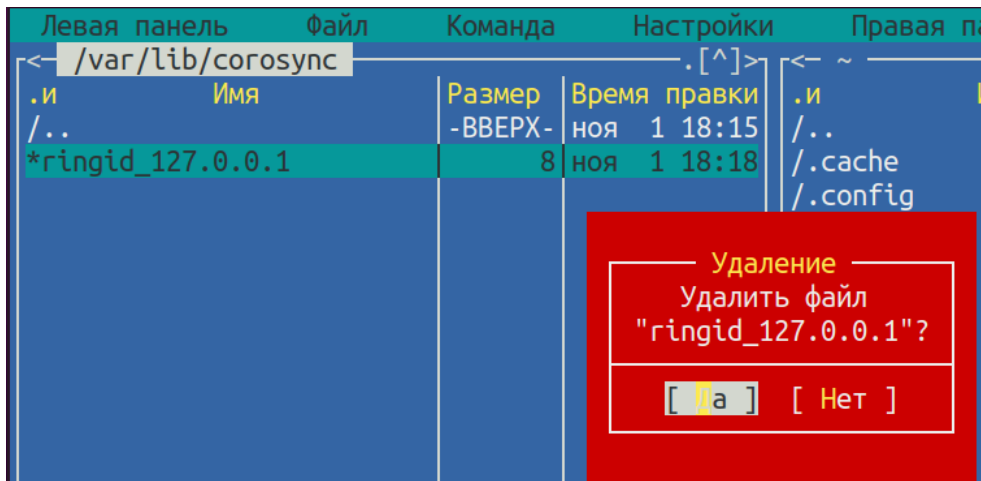
```
root@drbd01:~# systemctl stop corosync
```

Удаляем все файлы из следующих директорий (для удобства можно воспользоваться Midnight Commander: `$ sudo apt install mc`):

```
/var/lib/corosync
```

```
/var/lib/pacemaker/cib
```

```
/var/lib/pacemaker/pengine
```



Копируем файл конфигурации corosync.conf на вторую ноду:

```
root@drbd01:~# scp /etc/corosync/corosync.conf 10.100.10.14:
/etc/corosync/
```

Останавливаем сервисы на второй ноде:

```
root@drbd02:~# systemctl stop pacemaker
root@drbd02:~# systemctl stop corosync
```

Удаляем все файлы из следующих директорий:

```
/var/lib/corosync
/var/lib/pacemaker/cib
/var/lib/pacemaker/pengine
```

Добавляем в автозагрузку сервис Corosync и Pacemaker:

```
root@drbd01:~# systemctl enable corosync
root@drbd02:~# systemctl enable corosync
```

```
root@drbd01:~# systemctl enable pacemaker
root@drbd02:~# systemctl enable pacemaker
```

Запускаем сервисы на первой ноде:

```
root@drbd01:~# systemctl start corosync
root@drbd01:~# systemctl start pacemaker
```

Проверяем состояние нашего кластера с помощью утилиты crm_mon:

```
root@drbd01:~# crm_mon -fAr
```

```
Cluster Summary:
* Stack: corosync
* Current DC: drbd01 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Nov 27 14:27:10 2021
* Last change: Sat Nov 27 14:21:52 2021 by hacluster via crmd on drbd01
* 2 nodes configured
* 0 resource instances configured

Node List:
* Node drbd02: UNCLEAN (offline)
* Online: [ drbd01 ]

Full List of Resources:
* No resources

Migration Summary:
```

Видим, что первая нода работает.

Запускаем сервисы на второй ноде:

```
root@drbd02:~# systemctl start corosync
root@drbd02:~# systemctl start pacemaker
```

```

Cluster Summary:
* Stack: corosync
* Current DC: drbd01 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Nov 27 14:31:43 2021
* Last change: Sat Nov 27 14:21:52 2021 by hacluster via crmd on drbd01
* 2 nodes configured
* 0 resource instances configured

Node List:
* Online: [ drbd01 drbd02 ]

Full List of Resources:
* No resources

Migration Summary:

```

Видим, что все ноды работают.

3.4 Настройка ресурса Stonith.

Отредактируем конфигурацию нашего кластера, причем теперь это можно сделать на любой ноде, добавив строку `no-quorum-policy=ignore`:

```
root@drbd02:~# crm configure edit
```

```

node 1: drbd01
node 2: drbd02
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian

```

Установим следующие пакеты:

```
root@drbd01:~# apt install libvirt-clients
```

```
root@drbd02:~# apt install libvirt-clients
```

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```

node 1: drbd01
node 2: drbd02

primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"

primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"

property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian

```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Cluster Summary:
* Stack: corosync
* Current DC: drbd01 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Nov 27 15:36:06 2021
* Last change: Sat Nov 27 15:32:05 2021 by root via cibadmin on drbd01
* 2 nodes configured
* 2 resource instances configured

Node List:
* Online: [ drbd01 drbd02 ]

Full List of Resources:
* f_drbd01 (stonith:external/libvirt): Started drbd01
* f_drbd02 (stonith:external/libvirt): Started drbd02

Migration Summary:
```

Далее необходимо разнести ресурсы stonith по разным нодам.

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02
primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"

location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02

property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Cluster Summary:
* Stack: corosync
* Current DC: drbd01 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Nov 27 15:48:13 2021
* Last change: Sat Nov 27 15:48:10 2021 by root via cibadmin on drbd01
* 2 nodes configured
* 2 resource instances configured

Node List:
* Online: [ drbd01 drbd02 ]

Full List of Resources:
* f_drbd01 (stonith:external/libvirt): Started drbd02
* f_drbd02 (stonith:external/libvirt): Started drbd01

Migration Summary:
```

Запустим DRBD.

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02

primitive p_drbd ocf:linbit:drbd \
    params drbd_resource=r0

ms ms_drbd p_drbd \
    meta master-max=1 master-node-max=1 clone-max=2 clone-
node-max=1 notify=true

primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Cluster Summary:
* Stack: corosync
* Current DC: drbd01 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Nov 27 16:07:56 2021
* Last change: Sat Nov 27 16:07:53 2021 by root via cibadmin on drbd01
* 2 nodes configured
* 4 resource instances configured

Node List:
* Online: [ drbd01 drbd02 ]

Full List of Resources:
* f_drbd01 (stonith:external/libvirt): Started drbd02
* f_drbd02 (stonith:external/libvirt): Started drbd01
* Clone Set: ms_drbd [p_drbd] (promotable):
* Masters: [ drbd02 ]
* Slaves: [ drbd01 ]

Node Attributes:
* Node: drbd01:
* master-p_drbd : 1000
* Node: drbd02:
* master-p_drbd : 10000

Migration Summary:
```

3.5 Настройка Linux SCSI target.

Установим следующие пакеты:

```
root@drbd01:~# apt install tgt
root@drbd02:~# apt install tgt
```

Проверим файл конфигурации tgt, который должен иметь следующий вид:

```
root@drbd01:~# cat /etc/tgt/targets.conf
```

```
root@drbd01:~# cat /etc/tgt/targets.conf
# Empty targets configuration file -- please see the package
# documentation directory for an example.
#
# You can drop individual config snippets into /etc/tgt/conf.d
include /etc/tgt/conf.d/*.conf
```

Проверим директорию /etc/tgt/conf.d , – она должна быть пустой:

```
root@drbd01:~# ls -la /etc/tgt/conf.d
```

Проверим статус сервиса tgt, что он запущен:

```
root@drbd01:~# systemctl status tgt
```

```
root@drbd01:~# systemctl status tgt
● tgt.service - (i)SCSI target daemon
   Loaded: loaded (/lib/systemd/system/tgt.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-11-27 18:24:37 MSK; 11min ago
     Docs: man:tgtd(8)
  Main PID: 9941 (tgtd)
    Status: "Starting event loop..."
     Tasks: 1
    Memory: 960.0K
    CGroup: /system.slice/tgt.service
           └─9941 /usr/sbin/tgtd -f

ноя 27 18:24:37 drbd01 systemd[1]: Starting (i)SCSI target daemon...
ноя 27 18:24:37 drbd01 tgtd[9941]: tgtd: user_ib_init(3431) Failed to initialize RDMA; load kernel
ноя 27 18:24:37 drbd01 tgtd[9941]: tgtd: work_timer_start(146) use timer_fd based scheduler
ноя 27 18:24:37 drbd01 tgtd[9941]: tgtd: bs_init(387) use signalfd notification
ноя 27 18:24:37 drbd01 systemd[1]: Started (i)SCSI target daemon.
```

Добавим сервис tgt в автозагрузку:

```
root@drbd01:~# systemctl enable tgt
root@drbd02:~# systemctl enable tgt
```

Присвоим ip-адрес сервису tgt:

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02

primitive p_tgt_ip IPaddr2 \
    params ip=10.100.10.16 cidr_netmask=26 nic=enp1s0 \
    op monitor interval=30s

primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
```

```
primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive p_drbd ocf:linbit:drbd \
    params drbd_resource=r0
ms ms_drbd p_drbd \
    meta master-max=1 master-node-max=1 clone-max=2 clone-
node-max=1 notify=true
location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Node List:
* Node drbd01: online:
* Resources:
* f_drbd02 (stonith:external/libvirt): Started
* p_drbd (ocf::linbit:drbd): Slave
* p_tgt_ip (ocf::heartbeat:IPaddr2): Started
* Node drbd02: online:
* Resources:
* f_drbd01 (stonith:external/libvirt): Started
* p_drbd (ocf::linbit:drbd): Master
```

Видим, что ip tgt запустился на первой ноде.

Сконфигурируем сам сервис tgt.

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02

primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive p_drbd ocf:linbit:drbd \
    params drbd_resource=r0
primitive p_tgt_ip IPaddr2 \
    params ip=10.100.10.16 cidr_netmask=26 nic=enp1s0 \
    op monitor interval=30s
primitive p_tgt iSCSITarget \
    params implementation=tgt iqn="iqn.2021-
11.int.example:storage.drbd" allowed_initiators="10.100.10.13
10.100.10.14" portals="10.100.10.16:3260" tid=1
```

```
ms ms_drbd p_drbd \
    meta master-max=1 master-node-max=1 clone-max=2 clone-
node-max=1 notify=true
location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Node List:
* Node drbd01: online:
  * Resources:
    * f_drbd02 (stonith:external/libvirt): Started
    * p_drbd (ocf::linbit:drbd): Slave
    * p_tgt_ip (ocf::heartbeat:IPAddr2): Started
* Node drbd02: online:
  * Resources:
    * f_drbd01 (stonith:external/libvirt): Started
    * p_drbd (ocf::linbit:drbd): Master
    * p_tgt (ocf::heartbeat:iSCSITarget): Started
```

Видим, что сервис tgt запустился на второй ноде, а его ip все еще на первой ноде.

Определим tgt и ip tgt в одну группу.

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02

primitive f_drbd01 stonith:external/libvirt \
    params hostlist="drbd01"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive f_drbd02 stonith:external/libvirt \
    params hostlist="drbd02"
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive p_drbd ocf:linbit:drbd \
    params drbd_resource=r0
primitive p_tgt iSCSITarget \
    params implementation=tgt iqn="iqn.2021-
11.int.example:storage.drbd" allowed_initiators="10.100.10.13
10.100.10.14" portals="10.100.10.16:3260" tid=1
primitive p_tgt_ip IPAddr2 \
    params ip=10.100.10.16 cidr_netmask=26 nic=enp1s0 \
    op monitor interval=30s
group rg_tgt p_tgt_ip p_tgt
colocation c_rg_tgt_drbd inf: rg_tgt ms_drbd:Master
order o_msdrbd_bfore_rgtgt Mandatory: ms_drbd:promote
rg_tgt:start
```

```
ms ms_drbd p_drbd \
    meta master-max=1 master-node-max=1 clone-max=2 clone-
node-max=1 notify=true
location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Node List:
* Node drbd01: online:
* Resources:
* f_drbd02 (stonith:external/libvirt): Started
* p_drbd (ocf::linbit:drbd): Slave
* Node drbd02: online:
* Resources:
* f_drbd01 (stonith:external/libvirt): Started
* p_drbd (ocf::linbit:drbd): Master
* p_tgt_ip (ocf::heartbeat:IPAddr2): Started
* p_tgt (ocf::heartbeat:iSCSITarget): Started
```

Видим, что ip tgt перешел на вторую ноду.

Отредактируем конфигурацию нашего кластера:

```
root@drbd01:~# crm configure edit
```

```
node 1: drbd01
node 2: drbd02
primitive f_drbd01 stonith:external/libvirt \
    params hostlist=drbd01
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive f_drbd02 stonith:external/libvirt \
    params hostlist=drbd02
hypervisor_uri="qemu+ssh://root@10.100.10.1/system"
primitive p_drbd ocf:linbit:drbd \
    params drbd_resource=r0
primitive p_tgt iSCSITarget \
    params implementation=tgt iqn="iqn.2021-
11.int.example:storage.drbd" allowed_initiators="10.100.10.13
10.100.10.14" portals="10.100.10.16:3260" tid=1
primitive p_tgt_ip IPAddr2 \
    params ip=10.100.10.16 cidr_netmask=26 nic=enp1s0 \
    op monitor interval=30s
primitive p_tgt_lun iSCSILogicalUnit \
    params implementation=tgt target_iqn="iqn.2021-
11.int.example:storage.drbd" lun=1 path="/dev/drbd0"
group rg_tgt p_tgt_ip p_tgt p_tgt_lun
ms ms_drbd p_drbd \
    meta master-max=1 master-node-max=1 clone-max=2 clone-
```

```
node-max=1 notify=true
colocation c_rg_tgt_drbd inf: rg_tgt ms_drbd:Master
location lf_drbd01 f_drbd01 -inf: drbd01
location lf_drbd02 f_drbd02 -inf: drbd02
order o_msdrbd_bfore_rgtgt Mandatory: ms_drbd:promote
rg_tgt:start
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=2.0.3-4b1f869f0f \
    no-quorum-policy=ignore \
    cluster-infrastructure=corosync \
    cluster-name=debian
```

Проверим состояние нашего кластера:

```
root@drbd01:~# crm_mon -fAr
```

```
Node List:
* Node drbd01: online:
  * Resources:
    * f_drbd02 (stonith:external/libvirt): Started
    * p_drbd (ocf::linbit:drbd): Slave
* Node drbd02: online:
  * Resources:
    * f_drbd01 (stonith:external/libvirt): Started
    * p_drbd (ocf::linbit:drbd): Master
    * p_tgt_ip (ocf::heartbeat:IPaddr2): Started
    * p_tgt (ocf::heartbeat:iSCSITarget): Started
    * p_tgt_lun (ocf::heartbeat:iSCSILogicalUnit): Started
```

Видим, что iSCSILogicalUnit запущен на второй ноде.

Проверим все подключенные к порталу цели и предоставляемые ими блочные устройства:

```
root@drbd01:~# tgtadm -op show -mode target
```

```
root@drbd02:~# tgtadm --op show --mode target
Target 1: iqn.2021-11.int.example:storage.drbd
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
  LUN information:
    LUN: 0
      Type: controller
      SCSI ID: IET 00010000
      SCSI SN: beaf10
      Size: 0 MB, Block size: 1
      Online: Yes
      Removable media: No
      Prevent removal: No
      Readonly: No
      SWP: No
      Thin-provisioning: No
      Backing store type: null
      Backing store path: None
      Backing store flags:
```

```

LUN: 1
  Type: disk
  SCSI ID: p_tgt_lun
  SCSI SN: 8aee5696
  Size: 16103 MB, Block size: 512
  Online: Yes
  Removable media: No
  Prevent removal: No
  Readonly: No
  SWP: No
  Thin-provisioning: No
  Backing store type: rdwr
  Backing store path: /dev/drbd0
  Backing store flags:
Account information:
ACL information:
  10.100.10.13
  10.100.10.14
root@drbd02:~# █

```

Конфигурация tgt закончена.

Попробуем подключиться к порталу и получить список доступных целей:

```
root@drbd01:~# iscsiadm -m discovery -t st -p 10.100.10.16:3260
```

```

root@drbd01:~# iscsiadm -m discovery -t st -p 10.100.10.16:3260
10.100.10.16:3260,1 iqn.2021-11.int.example:storage.drbd

```

Теперь можно подключить цель:

```
root@drbd01:~# iscsiadm -m node -login
```

```

root@drbd01:~# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2021-11.int.example:storage.drbd, portal: 10.100.10.16,3260] (multiple)
Login to [iface: default, target: iqn.2021-11.int.example:storage.drbd, portal: 10.100.10.16,3260] successful.

```

Проверяем:

```
root@drbd01:~# ls -la /dev/sd*
```

```

root@drbd01:~# ls -la /dev/sd*
brw-rw---- 1 root disk 8, 0 ноя 26 03:29 /dev/sda
brw-rw---- 1 root disk 8, 1 ноя 26 03:29 /dev/sda1
brw-rw---- 1 root disk 8, 2 ноя 26 03:29 /dev/sda2
brw-rw---- 1 root disk 8, 3 ноя 26 03:29 /dev/sda3
brw-rw---- 1 root disk 8, 4 ноя 27 16:07 /dev/sda4
brw-rw---- 1 root disk 8, 16 ноя 27 22:17 /dev/sdb
root@drbd01:~#

```

Видим, что появилось новое блочное устройство /dev/sdb.