

Общая версия Linux. Уровень 1

# Установка операционной системы и знакомство с ней



# Оглавление

[Выбор дистрибутива. Сравнение Debian-подобных и RedHat-подобных дистрибутивов](#)

[Типы дистрибутивов](#)

[Версии](#)

[Обзор главного окна инсталлятора](#)

[Разметка диска. Завершение установки ОС](#)

[Утилита sudo, административные действия в ОС](#)

[Настройка сети, используя утилиты ip и nmcli](#)

[Изменение или настройка сети с использованием утилиты ip](#)

[Изменение или настройка сети с использованием утилит ifconfig и route из пакета net-tools](#)

[Изменение или настройка сети с использованием утилиты nmcli](#)

[Обновление системы. Установка дополнений VM в случае необходимости](#)

[Установка дополнений гостевой ОС \(VirtualBox\)](#)

[Установка OpenSSH Server и файлового менеджера Midnight Commander](#)

[Дополнительные материалы](#)

[Практическое задание](#)

[Используемые источники](#)

## На этом уроке

1. Посмотрим, какие типы дистрибутивов ОС Linux существуют. Научимся выбирать дистрибутив операционной системы под свои нужды. Узнаем, в чём основное различие между RedHat-подобными и Debian-подобными ОС.
2. Выясним, какие возможности нам предоставляет инсталлятор.
3. Узнаем способы разбивки на разделы пространства жёсткого диска под различные нужды.
4. Освоим основные действия, связанные с настройкой свежеставленной ОС.

## Глоссарий

**Виртуализация** — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

**Виртуальная машина** — программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы (target — целевая, или гостевая платформа) и исполняющая

программы для target-платформы на host-платформе (host — платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы.

**VirtualBox** (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS.

**VMWare Player** (VMware Workstation Player) — бесплатный для некоммерческого использования программный продукт на основе виртуальной машины VMware Workstation, но с ограниченной функциональностью, предназначенный для запуска образов виртуальных машин.

**Раздел** — часть долговременной памяти жёсткого диска или флеш-накопителя, выделенная для удобства работы и состоящая из смежных блоков. На одном устройстве хранения может быть несколько разделов.

**Logical Volume Manager (LVM)** — система управления томами с данными для Linux. Она позволяет создавать поверх физических разделов (или даже неразбитых жёстких дисков) логические тома, которые в самой системе будут видны как обычные блочные устройства с данными, то есть как обычные разделы.

**Bash** (сокр. от англ. *Bourne again shell*, каламбур «*Born again*» *shell* — «возрождённый» *shell*) — усовершенствованная и модернизированная вариация командной оболочки Bourne shell. Одна из наиболее популярных современных разновидностей командной оболочки UNIX. Особенно популярна в среде Linux, где она часто используется в качестве предустановленной командной оболочки.

**IP-адрес** — уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP.

**SSH** (сокр. от англ. *Secure Shell* — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). Схож по функциональности с протоколами TELNET и RLOGIN, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

**Точка монтирования** — каталог или, в редких случаях, файл, с помощью которого обеспечивается доступ к разделу. Раздел присоединяется в точку монтирования.

## Выбор дистрибутива. Сравнение Debian-подобных и RedHat-подобных дистрибутивов

Всё богатство мира дистрибутивов Linux можно разделить на несколько типов:

1. Дистрибутивы, основанные на ОС Debian (Ubuntu, Mint и т. д.).
2. Дистрибутивы, основанные на ОС Red Hat (CentOS, Fedora, Oracle Linux).
3. Отдельно, несмотря на то, что есть схожесть с типом пакетного менеджера RPM с Red Hat, можно выделить дистрибутивы SUSE и OpenSUSE.
4. Slackware.
5. Gentoo.

Это абсолютно условное деление на группы, более подробная информация о существующих типах дистрибутивов Linux есть [на сайте DistroWatch](#). Эта группировка основана на двух составляющих:

1. Пакетный менеджер и формат пакетов. Например, deb-пакеты и менеджер пакетов dpkg в Debian-дистрибутивах, RPM-пакеты и менеджер пакетов RPM в RedHat-дистрибутивах.
2. Общий дистрибутив-предок. Например, для Ubuntu и Mint общим предком является [Debian](#), а для Centos и Oracle Linux — [Red Hat Linux](#).

Наш курс будет основан на изучении [дистрибутива Ubuntu](#).

Ubuntu — один из наиболее популярных продуктов как для домашнего использования, так и для серверных решений. Он прекрасно задокументирован: практически на любой вопрос вы можете найти ответ либо в официальной документации, либо на страницах сообществ. Дистрибутив регулярно обновляется и зарекомендовал себя как стабильная операционная система. Для получения наиболее актуальной версии ОС Ubuntu необходимо перейти на официальный сайт в раздел [Download](#), и дальше вам предстоит выбрать версию. Предлагаются следующие варианты:

1. Ubuntu Desktop (LTS и не LTS).
2. Ubuntu Server (LTS и не LTS).

## Типы дистрибутивов

Ubuntu Desktop предназначен для использования на персональных ПК и сразу после установки включает в себя базовый набор программного обеспечения, который необходим обычному пользователю ПК: удобная графическая оболочка, видео- и аудиоплееры, простейшие программы для просмотра фото, офисные приложения. В отличие от версии Desktop, Server больше предназначен для установки на серверы, и почти ничего из того, что мы используем на домашних ПК, там не нужно. Как правило, отсутствует графическая оболочка и присутствует более гибкий инсталлятор. Но Linux не был бы Linux, если бы из любого десктоп-решения нельзя было бы сделать сервер, а из сервера — десктоп-решение.

## Версии

На 2020 год в обоих типах нам доступны две версии. Это 20.x LTS — версии дистрибутива с длительной поддержкой (LTS — сокр. от англ. Long Term Support), то есть данные дистрибутивы будут получать обновления достаточно долгое время, порядка пяти лет. Дистрибутив включает в себя только стабильные пакеты. Версии LTS выпускаются раз в два года. Не LTS версии содержат в себе

версии пакетов, которые не попадают в LTS версию из-за их возможной нестабильности. Версии выпускают где-то каждые девять месяцев, и они включают в себя самые новые пакеты.

Для домашнего или рабочего ПК вполне допустимо, даже лучше, использовать не LTS версии, а вот для сервера рекомендуется использовать LTS хотя бы из соображений стабильности.

**Внимание!** Вы можете использовать любой удобный для вас дистрибутив, но в рамках курса мы будем работать с последней доступной LTS версией Ubuntu Server.

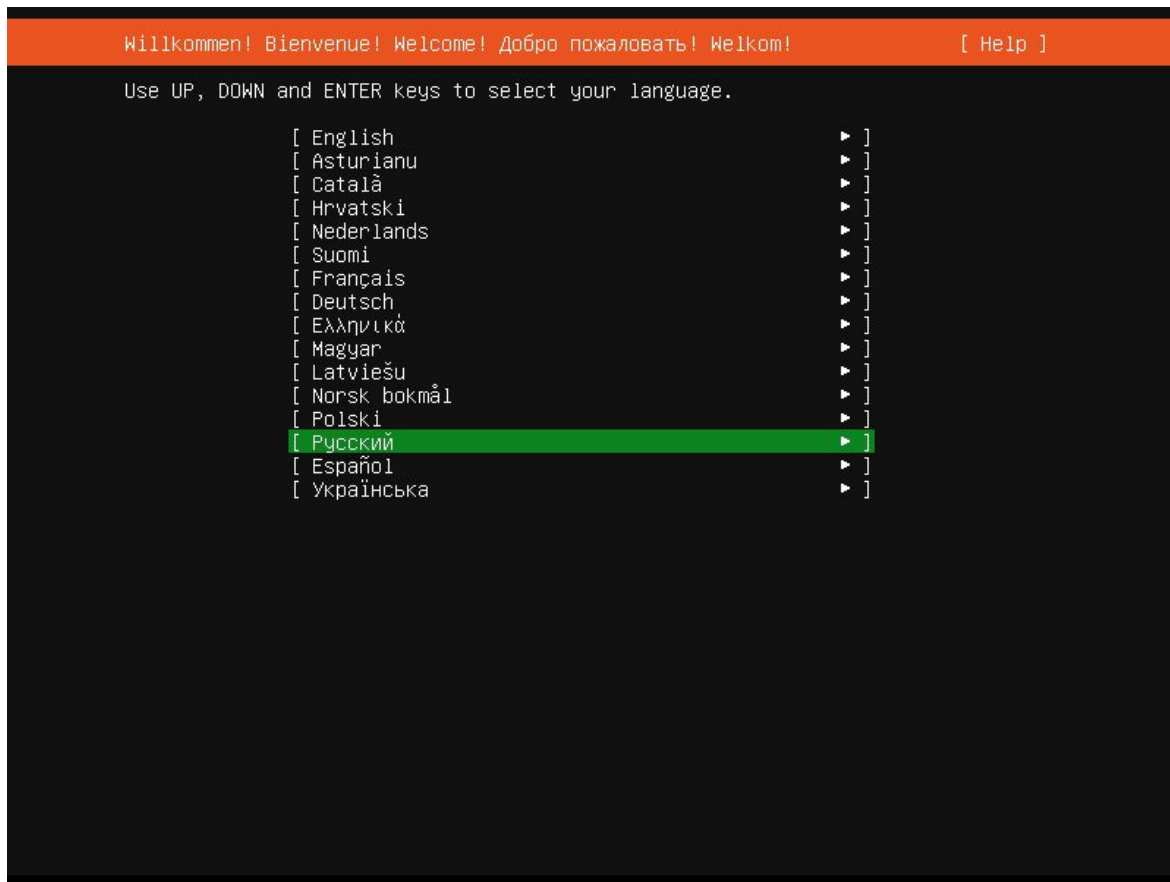
## Обзор главного окна инсталлятора

Для начала работы нам необходимы следующие инструменты:

1. Готовая виртуальная машина со следующими параметрами: ЦПУ — 1, ОЗУ — 2 GB, жёсткий диск — не менее 10 GB. Создание и настройку виртуальной машины в VirtualBox и VMware для целей курса вы можете [посмотреть по ссылке](#).
2. Скачанный образ последней LTS версии Ubuntu Server.

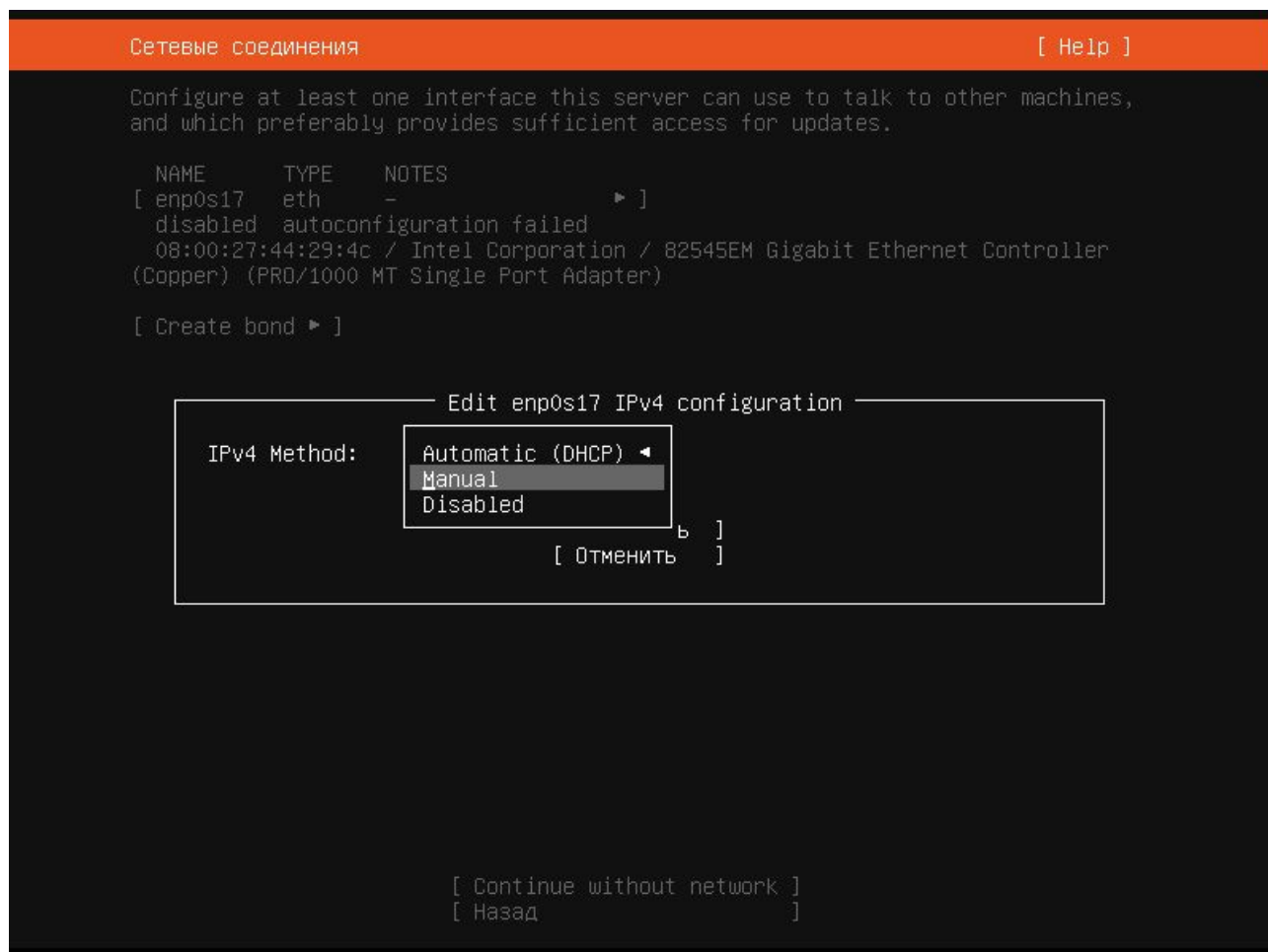
Подключаем образ к виртуальной машине, как это показано в инструкции и запускаем её.

После старта мы увидим первое отличие серверной версии Ubuntu от десктопной — отсутствие графического интерфейса у инсталлятора. Управление простое: клавиши «Вверх», «Вниз», «Влево», «Вправо» — перемещение по меню, клавиша «Пробел» — выбор пункта, Enter применяет выбранное.



Следующий шаг — выбор раскладки. Здесь оставляем всё по умолчанию, выбираем клавишу «Готово» и переходим к пункту настройки сети. Выбираем сетевой интерфейс и жмём Enter. Нам предлагается несколько путей назначения IP-адреса нашему серверу:

1. Используя протокол DHCP (Automatic DHCP) — этот вариант наиболее удобен, в частности, если у вашей виртуальной машины стоит тип сетевого подключения NAT.
2. Вручную (Manual) — используется, когда мы знаем настройки сети для нашего сервера, и этот вариант будет использован далее в процессе разбора установки.
3. Отключена (Disabled) — вариант, когда сеть не используется.



Выбираем вариант Manual, жмём Enter и вводим подсеть (Subnet) в формате x.x.x.x/mask (192.168.0.0/24), IP-адрес, шлюз по умолчанию и DNS-сервер (используем DNS Google 8.8.8.8). Нажимаем «Сохранить». Переходим к следующему шагу.

## Разметка диска. Завершение установки ОС

Прежде чем провести разметку диска на разделы, разберём структуру файловой системы в Linux. Файловая система Linux, в отличие от Windows, имеет древовидную структуру. Есть корневой раздел (/ -root), в котором содержатся файлы и каталоги, которые в свою очередь могут быть точками монтирования для других разделов.

Чтобы система загрузилась, зачастую достаточно двух разделов: корневого раздела «/» и раздела SWAP.

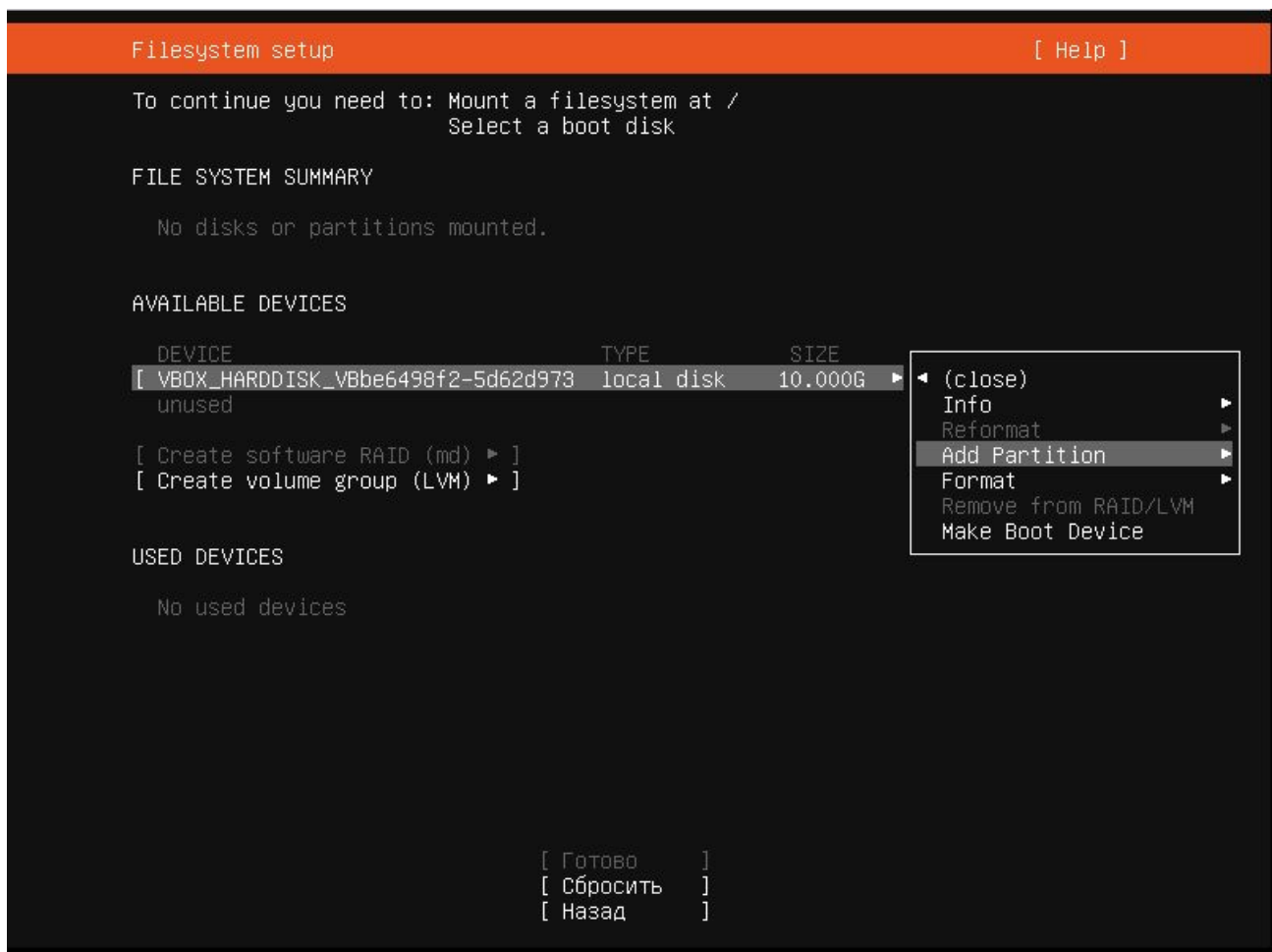
**SWAP** — один из механизмов виртуальной памяти, при котором отдельные фрагменты памяти (обычно неактивные) перемещаются из ОЗУ во вторичное хранилище (отдельный раздел или файл),



освобождая ОЗУ для загрузки других активных фрагментов памяти. Если мы разворачиваем сервер под Kubernetes, то раздел SWAP не нужен.

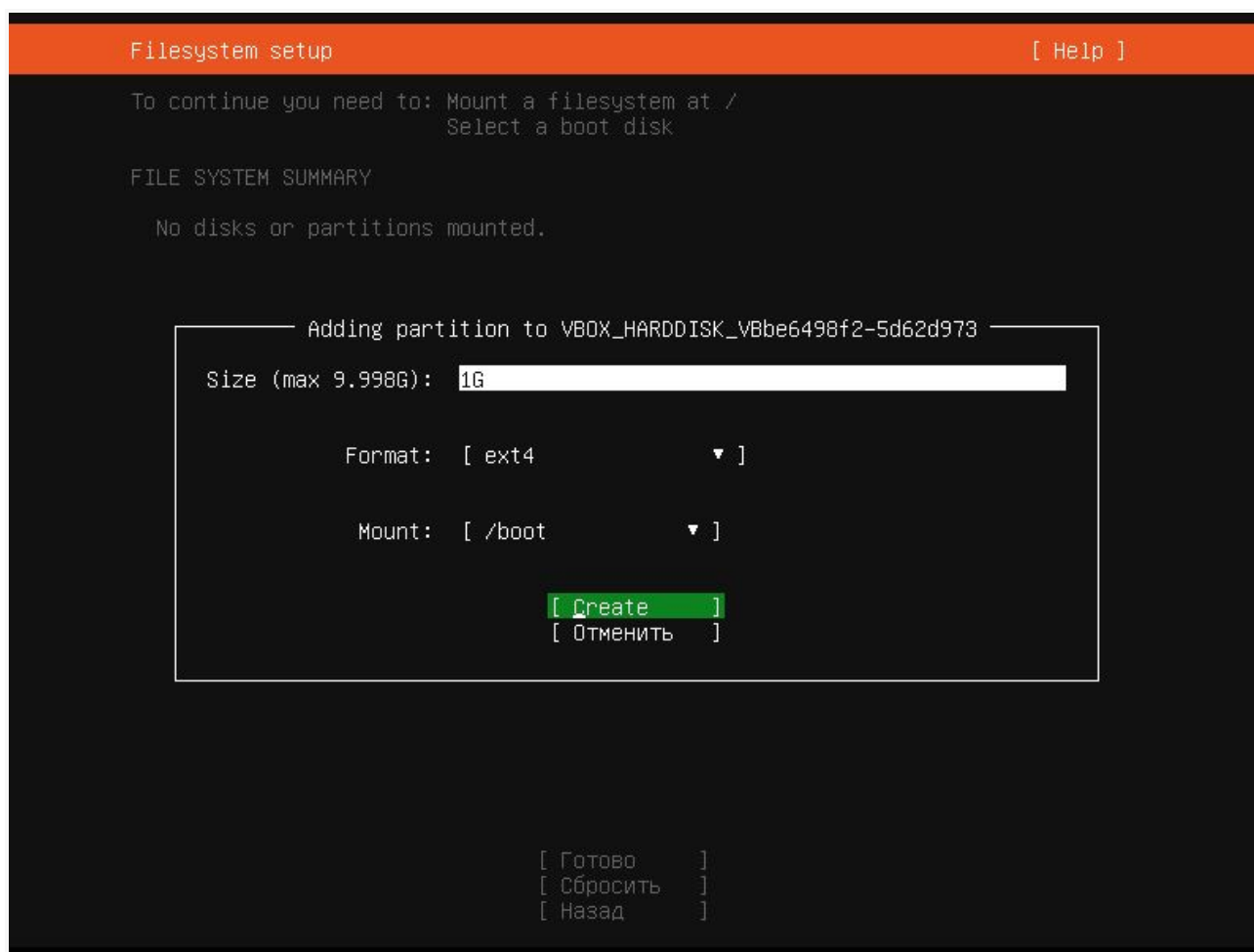
Если нет каких-то особых требований, можно доверить разметку диска инсталлятору, но на практике часто бывают случаи, когда разметка, предлагаемая инсталлятором, не удовлетворяет требованиям к серверу, например, домашний каталог пользователей должен иметь иной тип файловой системы. Второе отличие серверного варианта Ubuntu от десктопного — разметка дисков осуществляется вручную.

В меню Filesystem setup выбираем диск и нажимаем Enter, появляется меню разметки:



Выбираем пункт Add Partition, нажимаем Enter и заполняем поля следующим образом:

1. Size (размер в GB) — 1G — объём пространства под раздел.
2. Format — ext4 — тип [файловой системы](#), в которой будет отформатирован раздел.
3. Mount — /boot — точка монтирования. Раздел /boot содержит загрузочные файлы, необходимые для старта системы (загрузчик, ядро и т. д.). Его рекомендуется выносить в отдельный раздел, в частности, если мы используем [LVM](#) при разметке дисков, так как ни BIOS, ни EFI не умеют работать с LVM-разделами.



Прежде чем продолжить разбивку, дадим определение понятию LVM.

[Logical Volume Manager \(LVM\)](#) — это система управления томами с данными для Linux. Она позволяет создавать поверх физических разделов или даже неразмеченных жёстких дисков логические тома, которые в самой системе будут видны как обычные блочные устройства с данными, то есть как обычные разделы. Рекомендуется использовать её в случае установки серверной версии Ubuntu, так как она позволяет более гибко работать с имеющимся дисковым пространством: увеличить или уменьшить раздел без потери данных и с минимальным простоем системы.

Оставшиеся разделы разметим с использованием LVM. Для этого выбираем свободное пространство, нажимаем Enter. В пункте Size либо ничего не указываем (инсталлятор использует всё доступное пространство), либо указываем размер. Тип файловой системы выбираем «Оставить неформатированным» (Leave unformatted) и нажимаем Create.

Filesystem setup

[ Help ]

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[ /boot	1.000G	new ext4	new partition of local disk ► ]

Adding partition to VBOX\_HARDDISK\_VBbe6498f2-5d62d973

Size (max 8.997G): 8.997G

Format: [ Leave unformatted ▼ ]

Mount: [ / ▼ ]

[ Create ]

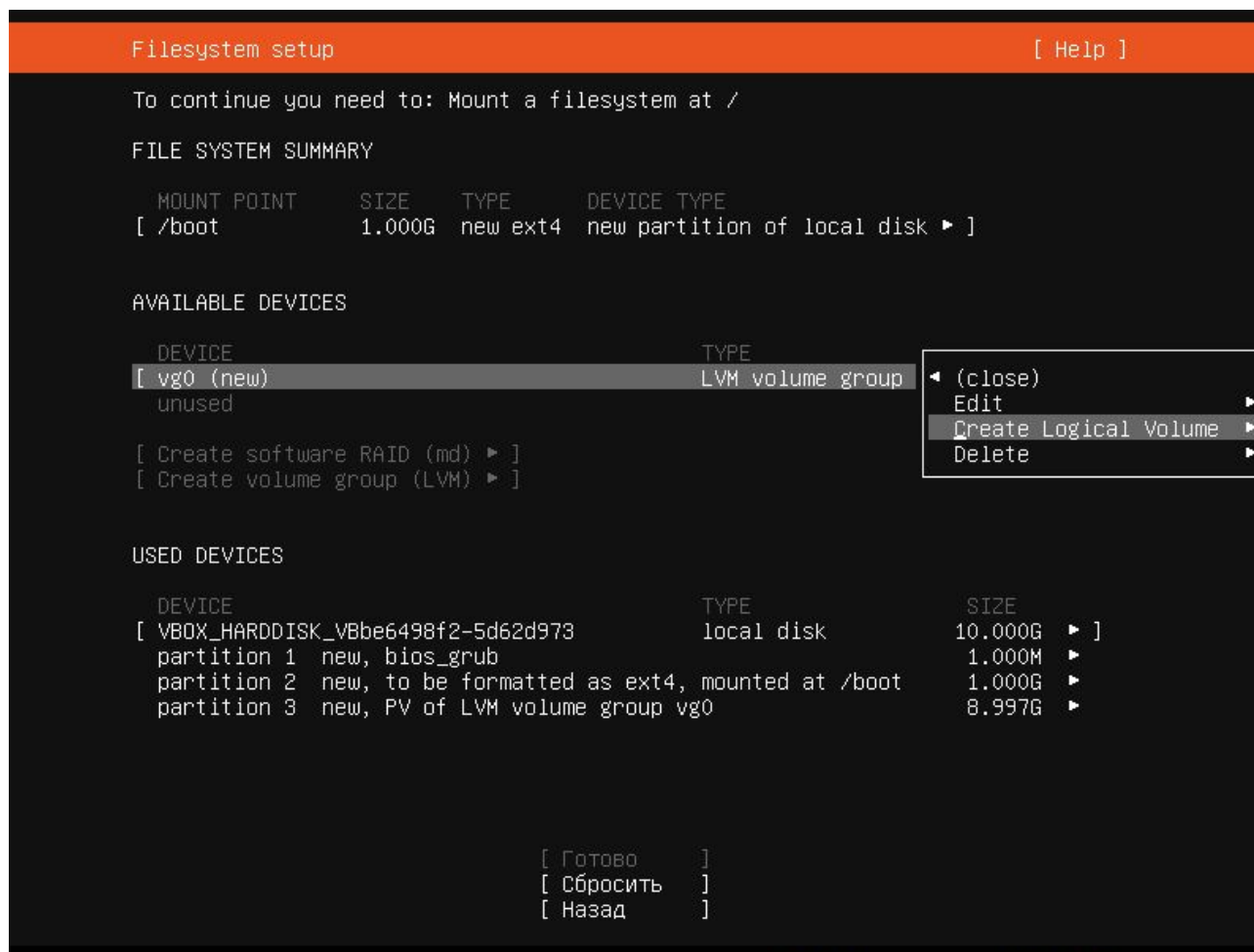
[ Отменить ]

[ Готово ]

[ Сбросить ]

[ Назад ]

Следующий шаг: создаём Volume Group. Выбираем пункт Create volume group, нажимаем Enter, имя группы оставляем по умолчанию и нажимаем Create. Volume Group с точки зрения LVM — это абстракция, которая объединяет в себе физические тома (реальные жёсткие диски). В результате мы получаем единое дисковое пространство, которое уже можем разбивать на своё усмотрение. В меню разметки выбираем имя группы и нажимаем Enter. Открывается меню, в котором нас интересует пункт Create Logical Volume.



Создаём первый Logical Volume. С точки зрения операционной системы это обычный раздел. Этот раздел в нашем случае будет [SWAP](#). Есть несколько подходов к выбору раздела SWAP:

1. Самый распространённый: SWAP равен количеству оперативной памяти.
2. Второй подход основан на [рекомендации компании Oracle](#): если в системе до 16 GiB оперативной памяти, то размер SWAP выставляем равным объёму ОЗУ, если выше 16 GiB, то размер SWAP равен 16 GiB.
3. И третий подход: если мы готовим систему под кластер Kubernetes, то раздел SWAP вообще не нужен.

В нашем случае выставляем размер раздела равным количеству оперативной памяти, выделенной виртуальной машине:

Filesystem setup

[ Help ]

To continue you need to: Mount a filesystem at /

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[ /boot	1.000G	new ext4	new partition of local disk ► ]

Adding logical volume to vg0

Name: lv-0

Size (max 8.996G): 2G

Format: [ swap ▼ ]

Mount: [ / ▼ ]

[ Create ]

[ Отменить ]

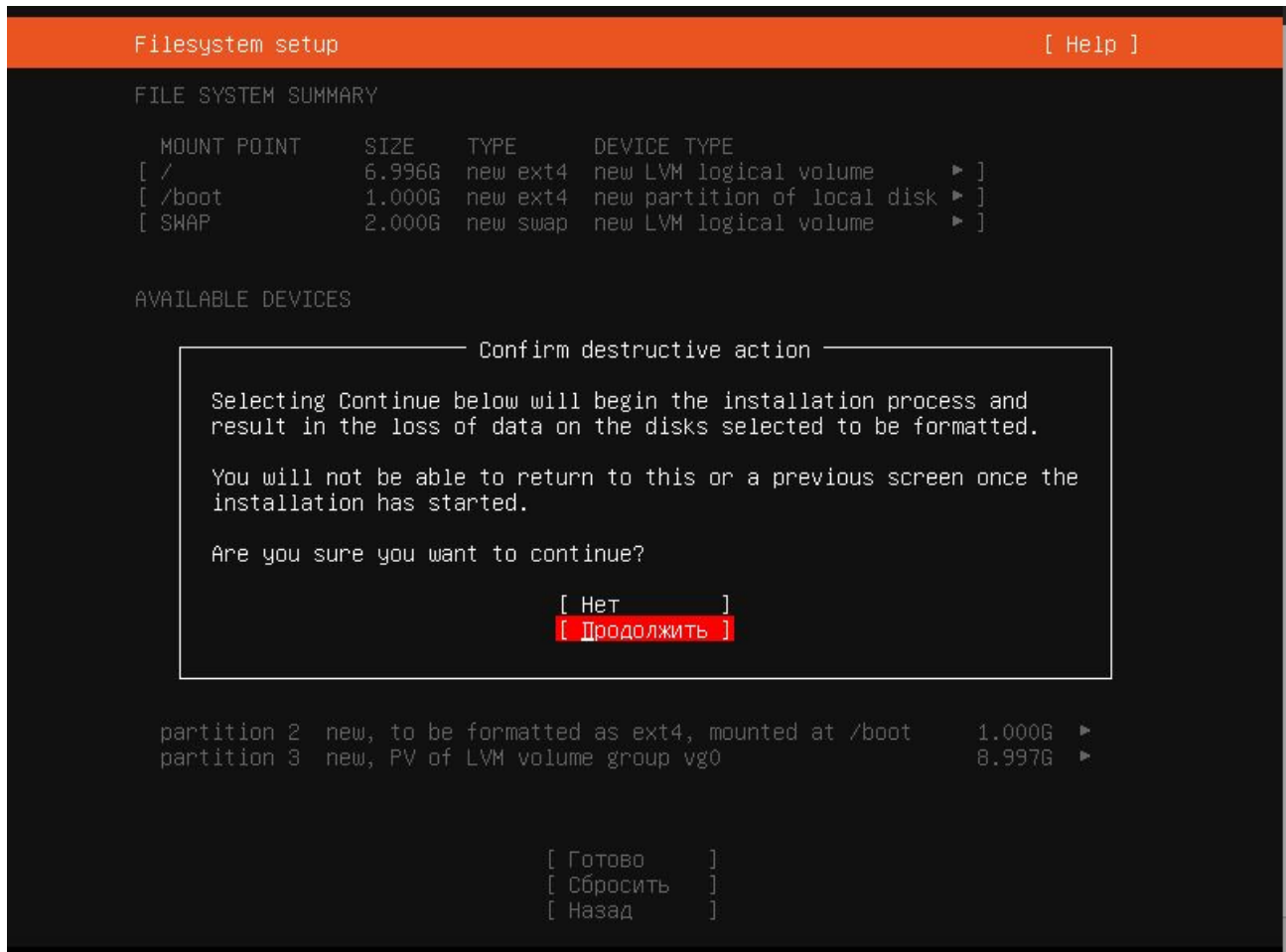
[ Готово ]

[ Сбросить ]

[ Назад ]

И завершаем разметку диска разделом «/», под который отдаём всё оставшееся место. Для этого повторяем те же шаги, что и для раздела SWAP.

Заканчиваем разметку, нажав «Готово», и далее выбираем пункт «Продолжить»:



После этого переходим к завершению установки. Создаём пользователя, который будет основным пользователем системы или администратором компьютера, задаём пароль, задаём имя сервера:

Profile setup

[ Help ]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:   
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

[ Готово ]

Следующий шаг — настройка SSH. В нашем случае мы активируем только пункт Install SSH server:

SSH Setup

[ Help ]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

[X] Install OpenSSH server

Import SSH identity: [ No ▼ ]  
You can import your SSH keys from Github or Launchpad.

Import Username:

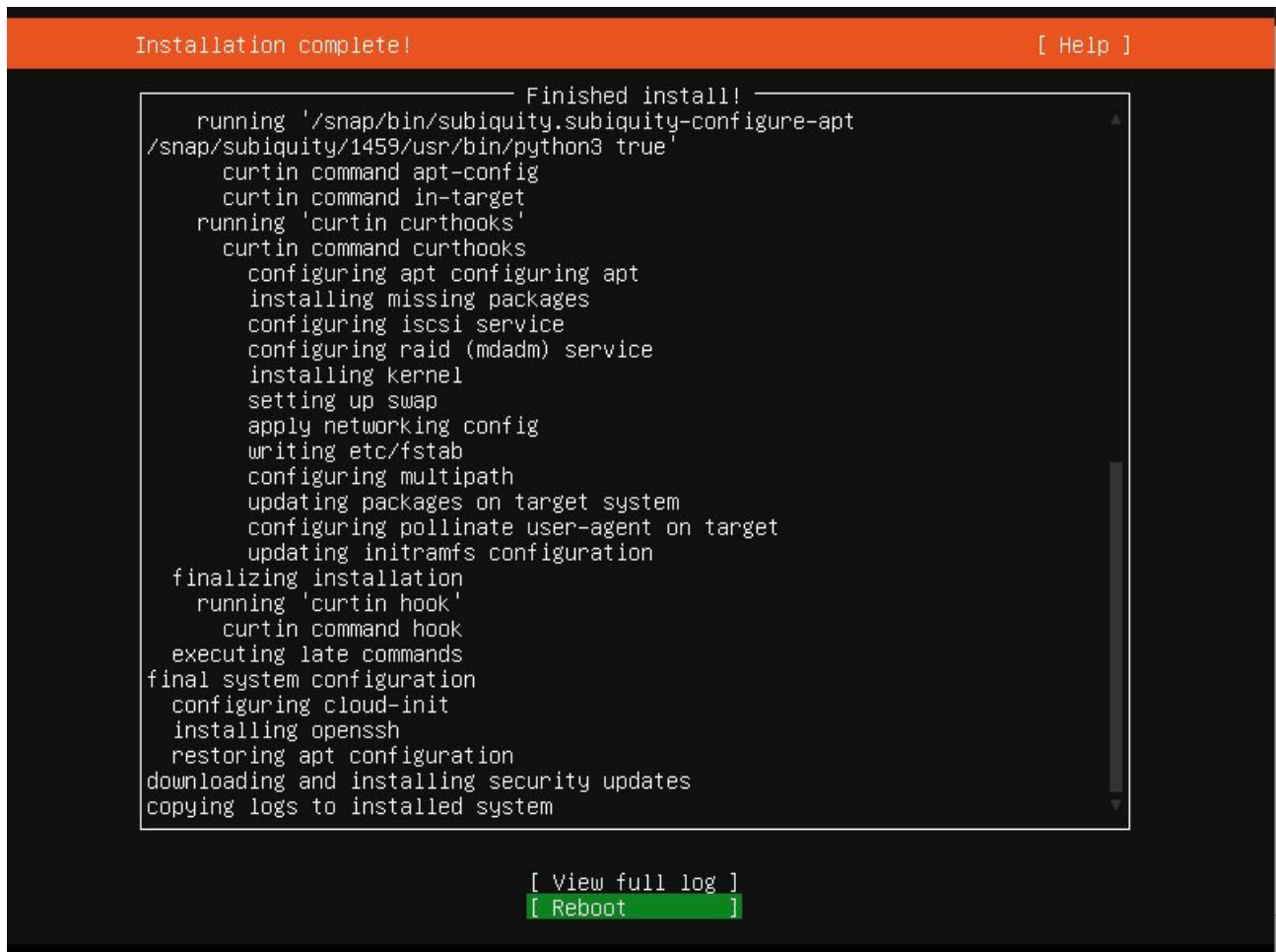
[X] Allow password authentication over SSH

[ Готово ]

[ Назад ]



Далее нам предлагается установить дополнительные возможности для сервера, этот пункт пропускаем, жмём «Готово» и ждём окончания установки ОС.



The screenshot shows a terminal window from the Ubuntu installer. The title bar is orange and says "Installation complete!" on the left and "[ Help ]" on the right. The terminal text is as follows:

```
----- Finished install! -----
running '/snap/bin/subiquity.subiquity-configure-apt
/snap/subiquity/1459/usr/bin/python3 true'
curtin command apt-config
curtin command in-target
running 'curtin curthooks'
curtin command curthooks
configuring apt configuring apt
installing missing packages
configuring iscsi service
configuring raid (mdadm) service
installing kernel
setting up swap
apply networking config
writing etc/fstab
configuring multipath
updating packages on target system
configuring pollinate user-agent on target
updating initramfs configuration
finalizing installation
running 'curtin hook'
curtin command hook
executing late commands
final system configuration
configuring cloud-init
installing openssh
restoring apt configuration
downloading and installing security updates
copying logs to installed system
```

At the bottom of the terminal window, there are two options: "[ View full log ]" and "[ Reboot ]", with the latter highlighted in green.

## Утилита **sudo**, административные действия в ОС

После установки ОС нам необходимо произвести ряд действий, связанных с настройкой и последующим администрированием системы. В Linux все действия, связанные с администрированием, а также внесением критических изменений в операционную систему, отданы особому пользователю — **root**. Любые действия, выполненные от этого пользователя, необратимы и в случае неаккуратного или неправомерного использования могут привести систему в неработоспособное состояние. Чтобы избежать возможных деструктивных действий или взлома системы, работа от имени пользователя **root** во многих дистрибутивах запрещена. Чтобы совершать административные действия, существует утилита **sudo**.

[Sudo](#) — это утилита, предоставляющая привилегии root для выполнения административных операций в соответствии со своими настройками. Она позволяет контролировать доступ к важным приложениям в системе.

Для дальнейших действий, связанных с администрированием системы, мы будем пользоваться этой утилитой.

## Настройка сети, используя утилиты `ip` и `nmcli`

Первое, что мы разберём, — это способы настройки сети. В Ubuntu Server и выше за настройку сети отвечает Netplan. В то же время изменить конфигурацию сети мы можем и классическим способом, используя инструменты командной строки: `ip` из пакета `iproute2`, `nmcli` (в случае с десктоп-версией Ubuntu) или `ifconfig` из пакета `net-tools`.

### Важно:

1. Изменение настроек сети с использованием утилит командной строки обычно «живёт» до первой перезагрузки системы.
2. В Ubuntu Desktop за настройку сети отвечает NetworkManager (утилита `nmcli` для управления из командной строки) или скрипт `ifupdown` (конфигурационный файл `/etc/network/interfaces` и служба `networking`). Netplan позволяет структурировать все указанные способы.
3. Изменение настроек сети — действие, требующее прав суперпользователя. При выполнении действий, связанных с изменением конфигурации, принято использовать `sudo`.

Разберем утилиты по порядку. Все рассматриваемые утилиты — консольные приложения, то есть для работы с ними необходима командная строка. В Linux стандартное приложение для работы с командной строкой — терминал, более подробно мы будем рассматривать его на дальнейших уроках. В работе с консольными приложениями используется следующий формат: **команда параметры\_команды**, при этом для передачи параметра могут использоваться символы `-`, `--`. Получить справку по синтаксису многих команд и возможным параметрам можно, используя следующий формат записи: **команда --help**.

## Изменение или настройка сети с использованием утилиты `ip`

1. Утилита `ip` — консольное приложение, которое можно использовать для получения информации о конфигурации сетевых интерфейсов, настройки сетевых интерфейсов, а также изменения таблицы маршрутизации.
2. Просмотр доступных сетевых интерфейсов и текущих настроек (не требует прав суперпользователя): `ip a`.
3. Изменение или присвоение IP-адреса сетевому интерфейсу:

```
sudo ip addr add {ip_addr/mask} dev {interface}
```

Пример:

```
sudo ip addr add 192.168.0.110/24 dev enp0s25
```

Имя интерфейса можем узнать из вывода `ip a`. Оно необходимо, чтобы операционная система могла сопоставить конкретное физическое устройство и присваиваемые этому устройству программные свойства (IP-адрес).

#### 4. Установка маршрута по умолчанию:

```
sudo ip route add default via {ip_gateway}
```

Пример:

```
sudo ip route add default via 192.168.0.1
```

## Изменение или настройка сети с использованием утилит `ifconfig` и `route` из пакета `net-tools`

1. Просмотр доступных интерфейсов и текущих настроек (не требует прав суперпользователя):  
`ifconfig -a`.
2. Изменение или присвоение IP-адреса интерфейсу:

```
sudo ifconfig {interface} {ip_addr} netmask {netmask}
```

Пример:

```
sudo ifconfig eth0 192.168.0.100 netmask 255.255.255.0
```

#### 3. Установка маршрута по умолчанию:

```
sudo route add default gw {ip_gateway}
```

Пример:

```
sudo route add default gw 192.168.0.1
```

## Изменение или настройка сети с использованием утилиты nmcli

Nmcli — утилита для управления службой NetworkManager из командной строки. Её можно использовать в случае работы с десктоп-версией Ubuntu.

1. Просмотр доступных интерфейсов и текущих настроек (не требует прав суперпользователя):  
`nmcli device` — покажет существующие устройства, `nmcli -p device show` — покажет настройки сети на всех имеющихся интерфейсах.
2. Настройка сетевого подключения, используя nmcli (добавляем статический адрес):

```
sudo nmcli connection add con-name "{name_connection}" ifname {device}
autoconnect {yes/no} type ethernet ip4 {ip_addr} gw4 {ip_gateway}
```

`{Name_connection}` — имя соединения (лучше использовать одно слово на английском языке), `{device}` — сетевой интерфейс, `autoconnect {yes/no}` — разрешить подключение при старте системы автоматически или нет, `{ip_addr}` — IP-адрес, `{ip_gateway}` — ip шлюза.

Пример:

```
sudo nmcli connection add con-name "connection1" ifname enp0s2 autoconnect
yes type ethernet ip4 192.168.0.100 gw4 192.168.0.1
```

## Обновление системы. Установка дополнений VM в случае необходимости

Следующий шаг, который необходимо сделать после установки операционной системы, — установить дополнения для виртуальной машины и обновить операционную систему. Дополнения, в частности, необходимы, если мы используем виртуальную машину под управлением VirtualBox.

### Установка дополнений гостевой ОС (VirtualBox)

Дополнения гостевой ОС содержат в себе набор приложений и драйверов, необходимых для оптимизации, повышения производительности и удобства эксплуатации ОС.

1. В меню управления виртуальной машиной выбираем пункт «Устройства» и в конце списка нажимаем «Подключить образ дополнений гостевой ОС».
2. Установка приложений требует прав суперпользователя, поэтому все последующие действия будем совершать, используя утилиту `sudo`. В установленной ОС выполняем следующую команду: `sudo apt update`. Apt — это утилита для управления пакетами, иными словами,

приложениями в Ubuntu. Параметр `update` говорит утилите, что необходимо обновить информацию о доступных пакетах в репозиториях (репозиторий — хранилище пакетов).

- Используя утилиту `apt`, скачаем и установим следующие пакеты: `sudo apt install perl make gcc -y`. Здесь параметр `install` говорит утилите `apt`, что необходимо поставить три пакета, и параметр `-y` означает, что `apt` будет устанавливать пакеты без подтверждения от пользователя.
- В Ubuntu Server не включена функция автомонтирования (подключения) внешних носителей. Следующий шаг — монтируем подключенный образ дополнений в ОС (требует прав суперпользователя): `sudo mount /dev/cdrom /mnt`, где `mount` — утилита монтирования дисковых устройств в Linux, `/dev/cdrom` — это виртуальное устройство чтения компакт-дисков. Оно же может быть и реальным, физическим. Каталог `/mnt` — точка монтирования `cdrom`.
- Переходим в каталог `/mnt`: `cd /mnt`. `Cd` (Change Directory) — команда для перемещения между каталогами в Linux. Запускаем скрипт установки дополнений: `sudo ./VBoxLinuxAdditions.run`.

После установки дополнений необходимо перезагрузить операционную систему, но, прежде чем это сделать, необходимо установить обновления. Для этого выполним команду `sudo apt upgrade -y`. Параметры `upgrade -y` передают утилите `apt`, что нужно провести обновление системы, не спрашивая подтверждения у пользователя. По завершении обновления перезагружаем ОС: `sudo reboot`.

## Установка OpenSSH Server и файлового менеджера Midnight Commander

В Ubuntu Server OpenSSH Server ставится автоматически, в случае использования Ubuntu Desktop его необходимо установить самостоятельно: `sudo apt install openssh-server -y`. Также для удобства навигации и работы с файлами рекомендуется поставить файловый менеджер Midnight Commander: `sudo apt install mc -y`.

## Практическое задание

- Повторить процесс установки ОС, но в качестве дистрибутива использовать десктоп-версию Ubuntu.
- \* Провести установку Centos 7.x и сравнить отличия в процессе установки.
- \* Провести сравнение дистрибутивов.

# Дополнительные материалы

<https://losst.ru/ustanovka-centos-7> — установка CentOS 7.

<https://losst.ru/kak-polzovatsya-virtualbox> — как пользоваться VirtualBox.

# Используемые источники

<https://help.ubuntu.com/20.04/installation-guide/amd64/install.en.pdf>