

Основы DevOps

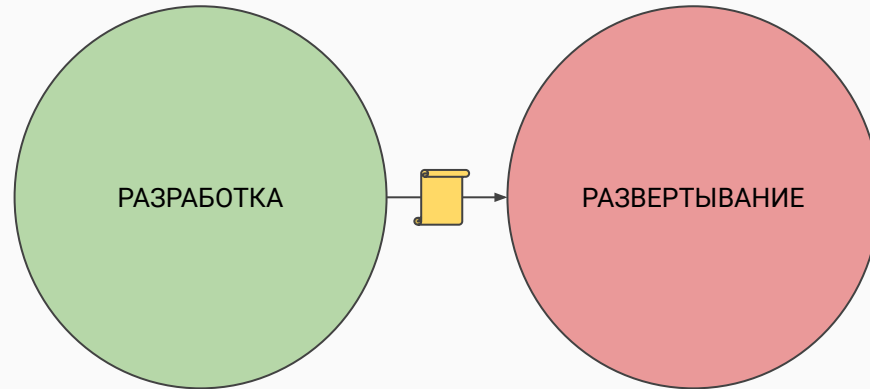
Введение



История развития разработки программного обеспечения

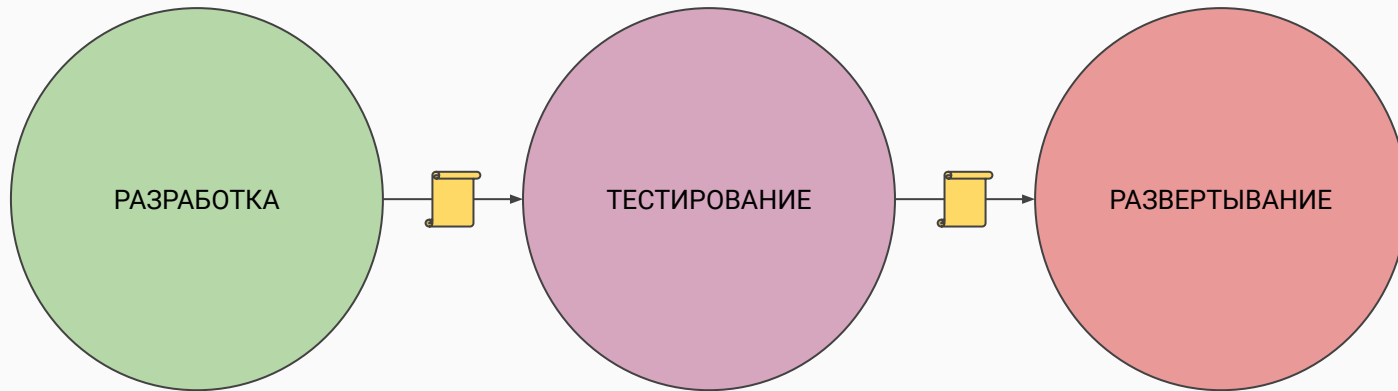
Развитие разработки программного обеспечения - “каменный век”

На заре создания современных подходов к разработке очень часто тестирование вообще отсутствовало и внедрением программного обеспечения занимались сами разработчики. Это могло приводить к настоящим трагедиям, например, широко известен случай с аппаратом лучевой терапии Therac-25 (США, 1985 - 1987 г.), когда в результате ошибки программиста погибло несколько пациентов различных больниц.



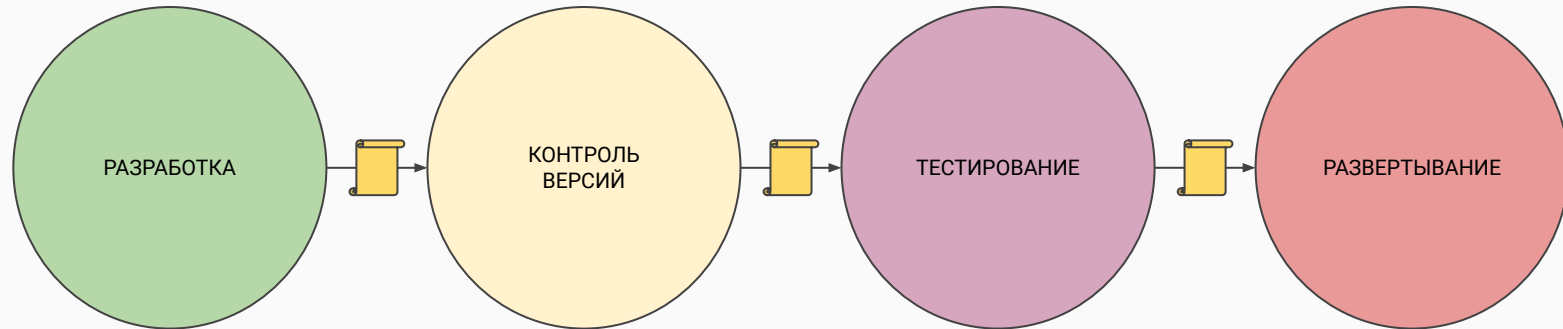
Развитие разработки программного обеспечения - “античность”

Введение стадии тестирования в качестве обязательного компонента процесса разработки благотворно повлияло на качество и безопасность программного обеспечения. Однако часто возникала другая проблема - после успешного внедрения и использования некого продукта могла отсутствовать документация или описание процессов разработки. В результате такого недосмотра, например, у нас нет информации по поводу некоторых технологических решений ракеты “Сатурн-5”, с помощью которой был осуществлен первый полет человека на Луну.



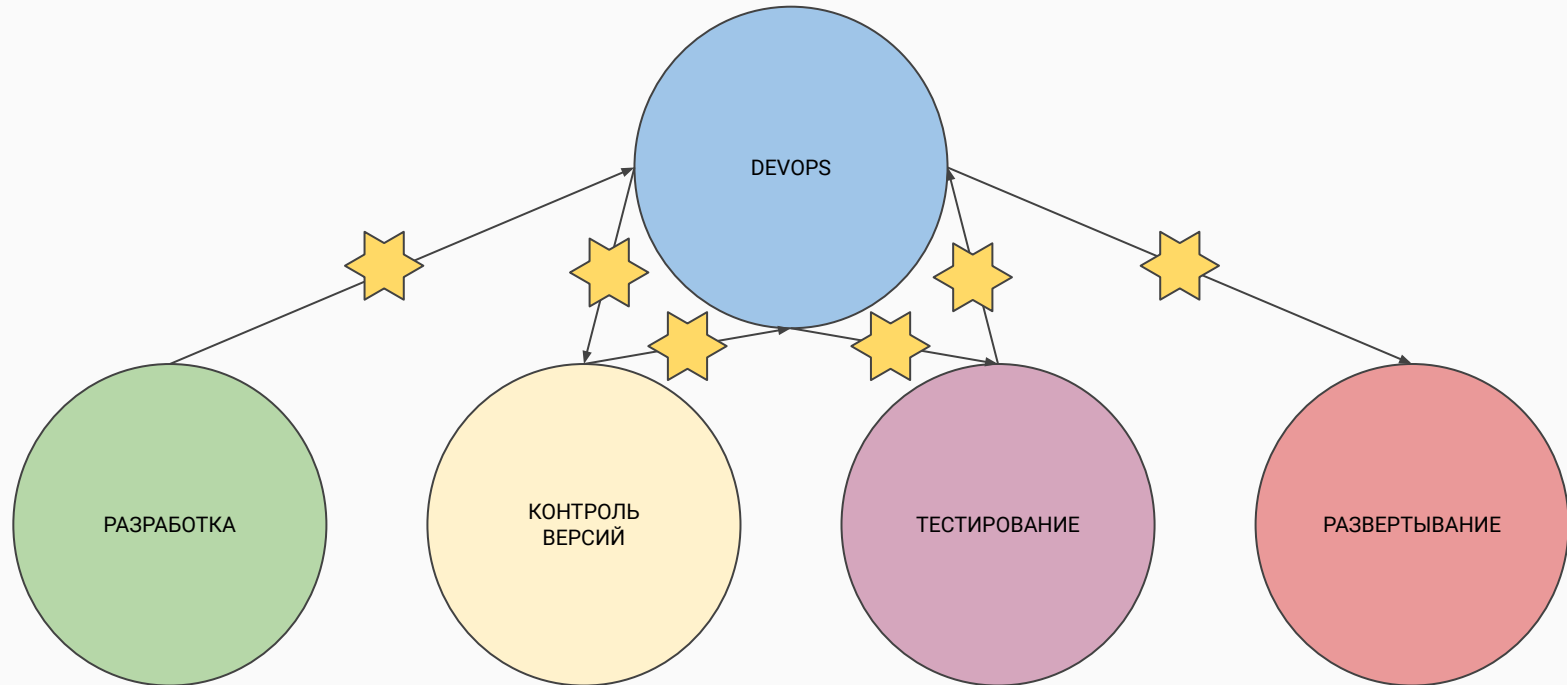
Развитие разработки программного обеспечения - “новое время”

Решением проблемы отсутствия документации о ходе разработки стало создание т.н. систем контроля версий - специальных программ, которые позволяли хранить несколько версий одного и того же документа, возвращаться к более ранним версиям, а также фиксировать, кто, когда и почему сделал то или иное изменение. На данный момент самой популярной такой системой является Git (ее автор - Линус Торвальдс, создатель ядра Linux). Однако это не решило проблему взаимодействия разных команд разработчиков и администраторов, т.к. их область деятельности слишком разрослась, и они просто перестали понимать друг друга - как в техническом, так и в организационном плане.



Развитие разработки программного обеспечения - “новейшее время”

Самым последним нововведением в разработке является методология и даже отдельная профессия DevOps. Специалисты в этой области координируют работу остальных команд и автоматизируют создание, тестирование и обновление продуктов. Это позволяет поддержке, администраторам и разработчикам сконцентрироваться на своей работе, а сам продукт попадает к конечному пользователю гораздо быстрее и в гораздо более лучшем качестве.



Предпосылки и причины появления DevOps

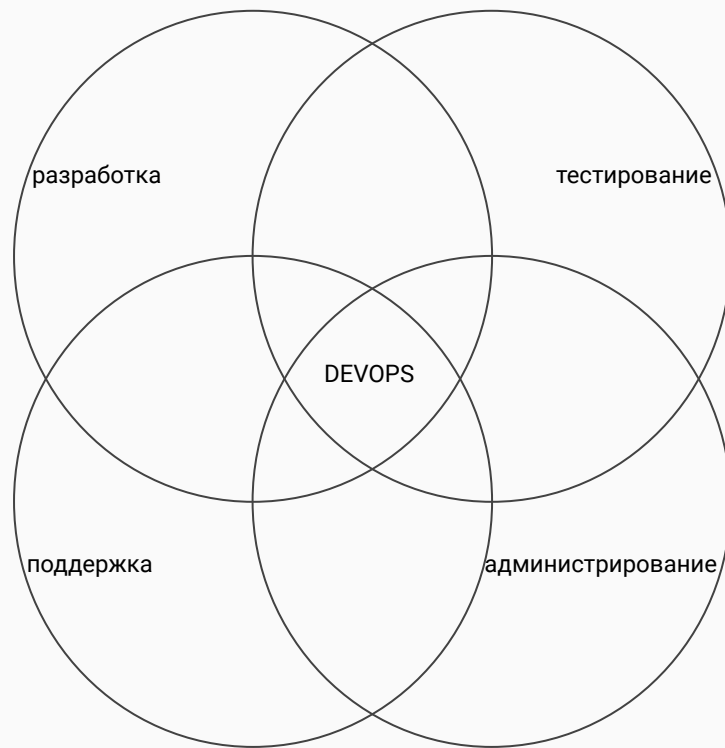
- Методология **DevOps** появилась относительно недавно - в 2009 году, и ее появление было связано с техническим прогрессом и вытекающим из этого усложнением разработки программного обеспечения (ПО).
- Крайне остро встал вопрос взаимодействия разработчиков, тестировщиков, системных администраторов, которые отвечали за развертывание и обновление ПО, а также службы поддержки, которая отвечала за стабильную работу ПО.
- Область деятельности каждой из вышеперечисленных групп разрослась настолько, что начались конфликты интересов (разработчикам выгодно быстро выпускать обновления, поддержка считает, что частые обновления угрожают стабильности ПО и т.д.).
- Техническая сложность развертывания и обновления ПО также сильно возросла - разработчики и администраторы просто перестали понимать друг друга, а установочные инструкции часто не учитывали специфику окружения и отличались от версии к версии.
- Чтобы пресечь конфликты, а также решить технические проблемы, и была создана методология и даже отдельная профессия **DevOps** - представители этой профессии стали посредниками и своего рода “клеем” между разными командами разработки.

Профессия DevOps инженер

Термин DevOps

DevOps - это методология разработки и управления информационными технологиями, которая объединяет разработчиков (development) и операционных специалистов (operations) с целью автоматизации и улучшения процессов разработки, тестирования, поставки и обслуживания программного обеспечения. Главная цель **DevOps** - сокращение времени и рисков при внесении изменений в приложения, а также обеспечение более стабильной и быстрой поставки новых функций и обновлений.

Как профессия, **DevOps** инженер (**DevOps** engineer) - это IT-специалист, который специализируется на создании и поддержке автоматизированных процессов разработки, тестирования, развертывания и управления инфраструктурой для программных приложений. **DevOps** инженеры объединяют в себе знания и навыки как разработчика (development), так и системного администратора (operations).



Непрерывная интеграция и непрерывная доставка (CI/CD)

Основной задачей DevOps специалиста является продумывание и создание **CI/CD** конвейера (**pipeline**), посредством которого результат труда разработчиков (как правило, программный код) будет автоматически подвергаться тестированию и, если потребуется, автоматически доставляться конечному пользователю в уже собранном виде.

- **Непрерывная интеграция (CI)** - это практика, при которой разработчики с помощью систем контроля версий регулярно загружают свой код (делают т.н. коммит) в общий репозиторий (директория или некий контейнер системы контроля версий, настроенные для отслеживания изменений файлов). Каждый раз, когда код загружается, автоматически запускаются тесты, чтобы убедиться, что новые изменения не ломают существующую функциональность. Это помогает выявлять и устранять проблемы на ранних этапах разработки.
- **Непрерывная доставка (CD)** - это продолжение **CI**. После успешного прохождения тестов код автоматически готовится к выпуску. Это может включать в себя создание установочных пакетов, обновление серверов и автоматическую установку на рабочие серверы. Суть в том, что после каждого изменения кода вы можете быстро и надежно доставлять его конечному потребителю (часто используется термин “в продакшн”), что позволяет быстрее реагировать на изменения и улучшать продукт.

Чем конкретно занимается DevOps инженер?

- Создание скриптов с использованием Bash, Python или других языков программирования для настройки конфигурации приложений и их зависимостей.
- Написание программ для автоматизация тестирования при каждом коммите (загрузке кода) в репозиторий.
- Настройка CI/CD конвейеров с использованием систем контроля версий и таких инструментов как Jenkins, GitLab CI или Github Actions.
- Использование систем Terraform или Ansible для создания и управления инфраструктурой (операционными системами и сетями).
- Создание Docker-образов приложений и их зависимостей, чтобы упростить развертывание в тестовых и боевых средах.
- Настройка веб-серверов, такие как Apache HTTP Server, Nginx, Tomcat и другие. Эти серверы применяются для обслуживания веб-приложений и сайтов.
- Настройка и управления серверами баз данных, такими как MySQL, PostgreSQL, MongoDB, Microsoft SQL Server и другими. Эти серверы хранят и обрабатывают данные приложений.
- Настройка инструментов мониторинга, таких как Prometheus и Grafana, для отслеживания состояния приложений и инфраструктуры.

Список базовых технологий для получения работы

- Знать хотя бы один популярный язык программирования (Python, Go, PHP, Java и т.д.)
- Разбираться в операционных системах (в CI/CD процессе главные позиции занимают ОС на ядре Linux - производные от Debian, Red Hat и т.д.)
- Понимать устройство компьютерных сетей, уметь взаимодействовать с этими сетями
- Ориентироваться в инструментах, связанных с системами контроля версий (Git, Github, Bitbucket и т.д.)
- Иметь опыт в настройке web-серверов - Apache, Nginx, Tomcat и т.д.
- Уметь настраивать сервера баз данных - MySQL, PostgreSQL и т.д.
- Понимать для чего нужен Docker и уметь использовать эту технологию
- Иметь хотя бы представление об облачных платформах (AWS, Google Cloud, Azure и т.д.)
- И многое, многое, многое, многое, многое, многое, многое, очень многое другое ...

Обучение практикам DevOps на курсах

- **Изучить с нуля профессию DevOps (чтобы затем с высокой долей вероятности получить работу) на курсах невозможно!**
- DevOps опирается на работу с фундаментальными вещами - операционными системами, сетями и программированием - для начала на курсах можно освоить именно эти вещи
- Если человеку интересна сфера IT, но он еще не определился, что именно хочет, то курсы позволят познакомиться с разными темами, что поможет конкретизировать свой выбор
- При должном прилежании слушатель DevOps курсов с изначально нулевыми знаниями IT сможет повысить свою компьютерную грамотность и претендовать на должность, которая предшествует позиции DevOps
- Практики DevOps распространились и укоренились настолько, что их знание на базовом требуют уже от простых разработчиков - курсы могут обеспечить этот базовый уровень
- Человек, уже имеющий опыт в IT, но застрявший в устаревших технологиях и подходах, на курсах сможет узнать о самых последних достижениях в своей области, перейти на новый уровень знаний и даже получить повышение в своей компании

1. Администрирование ОС на базе ядра Linux (на примере Ubuntu)
2. Основы компьютерных сетей
3. Практическое применение знаний о компьютерных сетях и операционных системах
4. Системы контроля версий (на примере Git)
5. Контейнеризация при помощи Docker
6. Непрерывная интеграция и доставка (CI/CD) на базе Jenkins
7. Альтернативные способы непрерывной интеграции и доставки
8. Мониторинг операционных систем и приложений
9. *Облачные технологии (на примере AWS - Amazon Web Services)*
10. *Инфраструктура как код в контексте систем Ansible и Terraform*

Подготовка к установке Oracle VirtualBox и ОС Ubuntu

Почему необходимо начинать освоение DevOps с изучения ОС на ядре Linux?

- В DevOps практикуется использование инструментов командной строки, скриптов и контейнеризации для создания и развертывания приложений. Эти навыки легче развивать на Linux, где доступ к командной строке более естественен.
- Linux предоставляет пользователю полный контроль над системой и ее настройками. Вы можете настраивать систему под свои нужды.
- Linux широко используется на серверах по всему миру. Он является основой многих серверных систем, облачных платформ (например, AWS, Azure, Google Cloud), контейнерных технологий (Docker) и инструментов автоматизации (Ansible). Поэтому понимание Linux является фундаментом для работы в DevOps сфере.
- Многие инструменты, используемые в DevOps (например, Jenkins или Terraform) лучше поддерживаются или легко настраиваются на Linux. Знание Linux позволяет уверенно работать с этими инструментами.
- Linux имеет огромное сообщество пользователей и разработчиков, что означает, что всегда можно найти помощь и ресурсы для обучения.
- Linux распространяется по лицензии с открытым исходным кодом (чаще всего GNU GPL), что позволяет использовать его бесплатно.

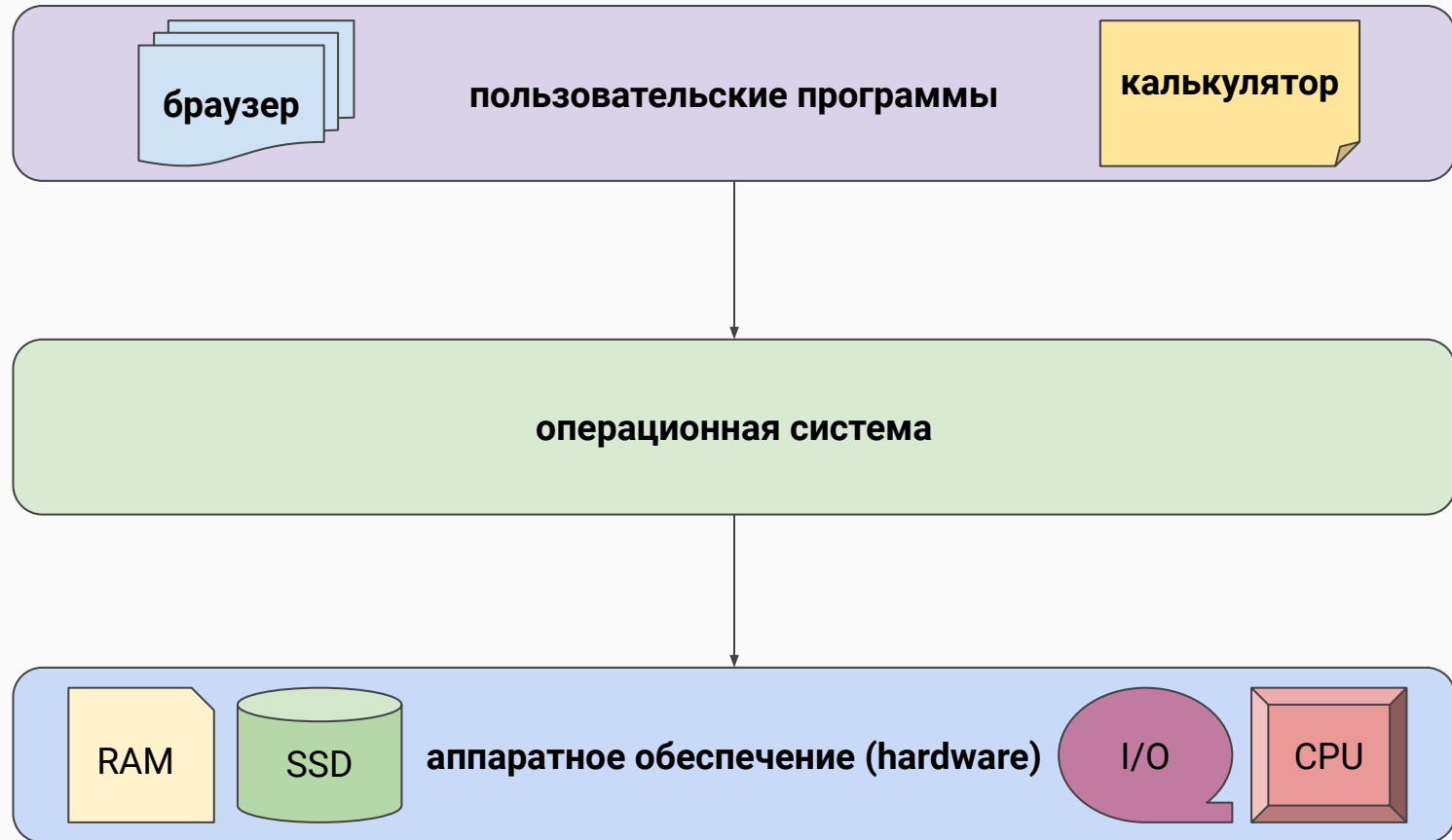
Базовые способы установки ОС на ядре Linux на свой компьютер

- **Установка на физический компьютер в качестве единственной или дополнительной ОС (с разделением диска на несколько разделов).** Обеспечивает максимальную производительность (в каждый момент времени компьютер предоставляет свои ресурсы только одной ОС) и полное погружение в работу с Linux. Однако данный вариант является опасным для обучения, т.к. существует вероятность, что при экспериментах будет повреждена либо сама система, либо загрузчик ОС. Это может привести к большим проблемам, особенно если у учащегося нет второго компьютера.
- **Использование WSL (Windows Subsystem for Linux).** Технология WSL позволяет запускать ОС на ядре Linux поверх ОС Windows (при этом обе ОС будут работать одновременно, но изолированно). Данный подход обладает достаточным быстродействием (хотя и медленнее, чем предыдущий вариант), позволяет работать с инструментами Linux из командной строки Windows, а также предоставляет бесшовный доступ к файлам и директориям на диске Windows. Однако WSL плохо поддерживает графический интерфейс, кроме того, прямой доступ к данным Windows таит в себе определенную угрозу, т.к. при обучении Linux можно случайно уничтожить файлы Windows.
- **Использование виртуальной машины (например, Oracle VirtualBox).** Данная программа для виртуализации также позволяет запускать Linux поверх Windows, однако уступает WSL в производительности и не имеет прямого доступа к файловой системе Windows. В то же время Oracle VirtualBox поддерживает графический интерфейс (что может быть важно новичку) и более строго изолирована от Windows, что позволяет безопасно экспериментировать с DevOps инструментами. Таким образом, виртуальная машина является хотя и самым медленным, но одновременно самым надежным и удобным средством для знакомства с ОС на ядре Linux.

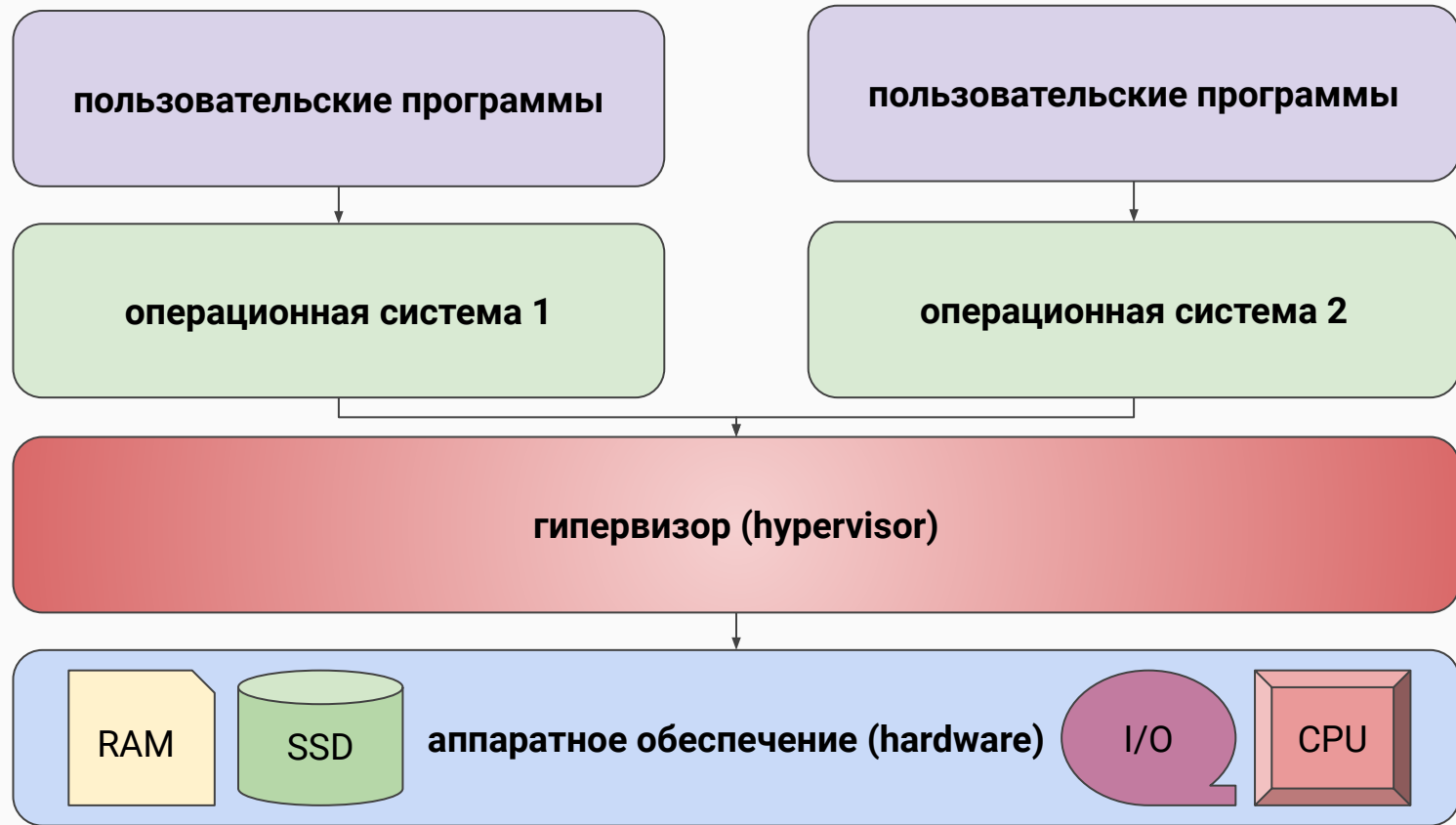
Технологии для установки Linux поверх Windows (виртуализация и гипервизор)

- **Аппаратная виртуализация** - это способность процессора (CPU) одновременно поддерживать несколько виртуальных машин на одном или нескольких физических ядрах. Чтобы воспользоваться возможностями этой технологии, нужно иметь подходящее оборудование и ПО. Для начала следует узнать, включена виртуализация или нет - для этого в Windows следует нажать **Ctrl+Shift+Esc**, перейти во вкладку, посвященную процессору, а затем найти надпись **Virtualization**, где будет указано, включена ли данная технология или нет. Для включения виртуализации сразу после запуска компьютера следует перейти в **BIOS (UEFI)**, найти настройку **Virtualization** (название может отличаться, но так или иначе там должен упоминаться термин "виртуализация") и перевести ее в режим **Enabled**. На разных компьютерах переход в **BIOS (UEFI)** осуществляется при помощи разных клавиш (**F2**, **Fn+F2**, **F6**, **DEL** и т.д.).
- **Гипервизор** - это программное обеспечение, которое можно использовать для запуска нескольких виртуальных машин в одной физической машине. Каждая виртуальная машина имеет свою операционную систему и приложения. Гипервизор по мере необходимости выделяет вычислительные ресурсы, такие как мощности CPU и память, для отдельных виртуальных машин. Таким образом поддерживается оптимальное использование ресурсов. Для виртуализации WSL (его современной второй версии) используется специальная выделенная часть (Virtual Machine Platform) встроенного гипервизора Windows (Hyper-V), а Oracle VirtualBox сама по себе является гипервизором.

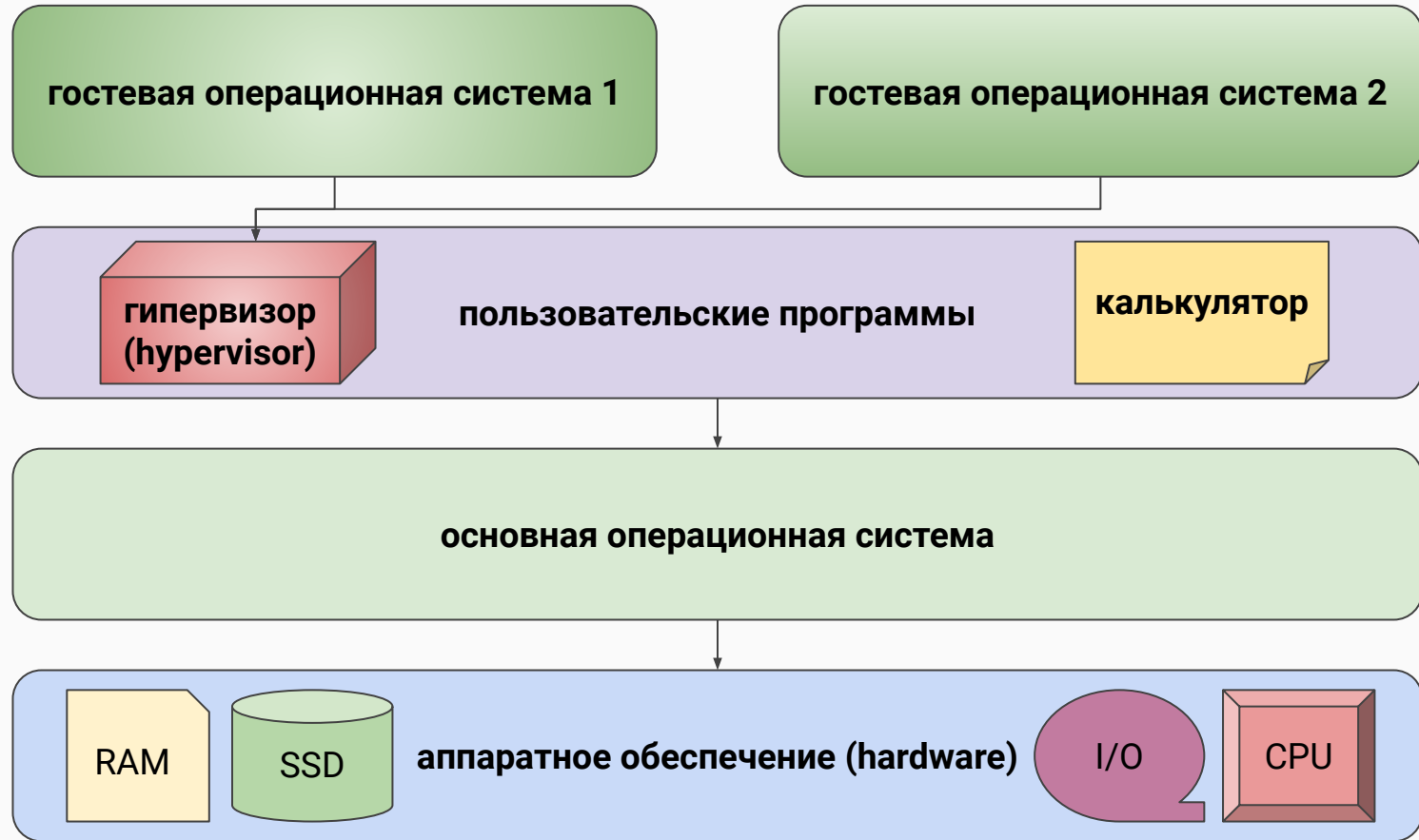
Схема функционирования операционной системы без виртуализации



Виртуализация (гипервизор первого типа)



Виртуализация (гипервизор второго типа)



Выбор дистрибутива Linux для обучению DevOps практикам

Ubuntu рекомендуется как один из лучших дистрибутивов Linux для обучения по нескольким причинам:

- **Популярность и распространенность** - Ubuntu является одним из самых популярных дистрибутивов Linux. Это означает, что с Ubuntu вы сможете найти больше информации для обучения и поддержки. Столкновение с проблемами будет проще решить, так как сообщество Ubuntu очень активно.
- **Простота установки и использования** - Ubuntu предоставляет простой процесс установки и настройки, что делает его привлекательным для новичков. Команда apt-get (или более новая apt) облегчает установку и управление пакетами, что упрощает работу с DevOps-инструментами и приложениями.
- **Широкий выбор DevOps-инструментов** - в репозиториях Ubuntu можно легко найти и установить большинство инструментов, используемых в DevOps, такие как Ansible, Terraform и многие другие. Это позволяет быстро начать практическое обучение и развертывание инструментов.
- **Поддержка долгосрочных версий** - Ubuntu предоставляет долгосрочную поддержку (LTS) для определенных версий, что обеспечивает стабильность и обновления безопасности в течение длительного времени.
- **Кроссплатформенность** - многие DevOps-инструменты и платформы могут быть установлены и использованы как на Ubuntu, так и на других дистрибутивах Linux, что обеспечивает универсальность ваших навыков и знаний.

Установка Oracle VirtualBox

Чтобы начать работать с **Oracle VirtualBox**, надо сперва загрузить эту программу на компьютер. Для этого следует перейти по ссылке <https://www.virtualbox.org/wiki/Downloads> и выбрать версию для вашей операционной системы (например, **"Windows hosts"**) - загрузка начнется сразу же:

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.1 packages, see [VirtualBox 6.1 builds](#). Version 6.1 will remain supported until December 2023.

VirtualBox 7.0.10 platform packages

- [Windows hosts](#)
- [macOS / Intel hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

The binaries are released under the terms of the GPL version 3.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

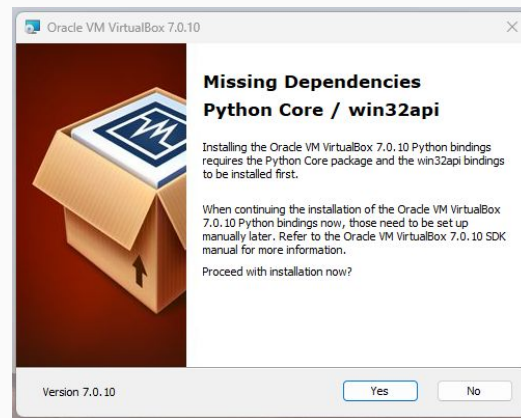
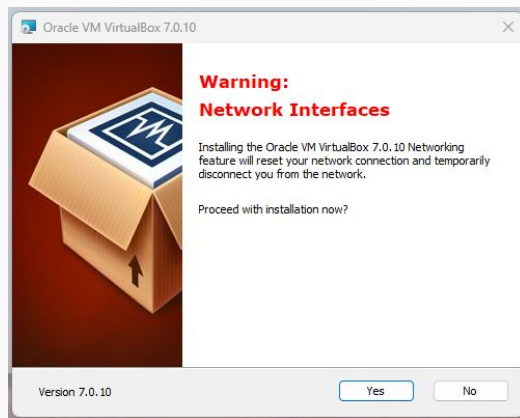
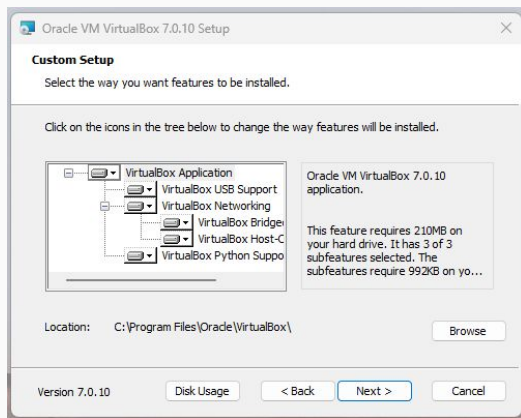
- [SHA256 checksums](#), [MD5 checksums](#)

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

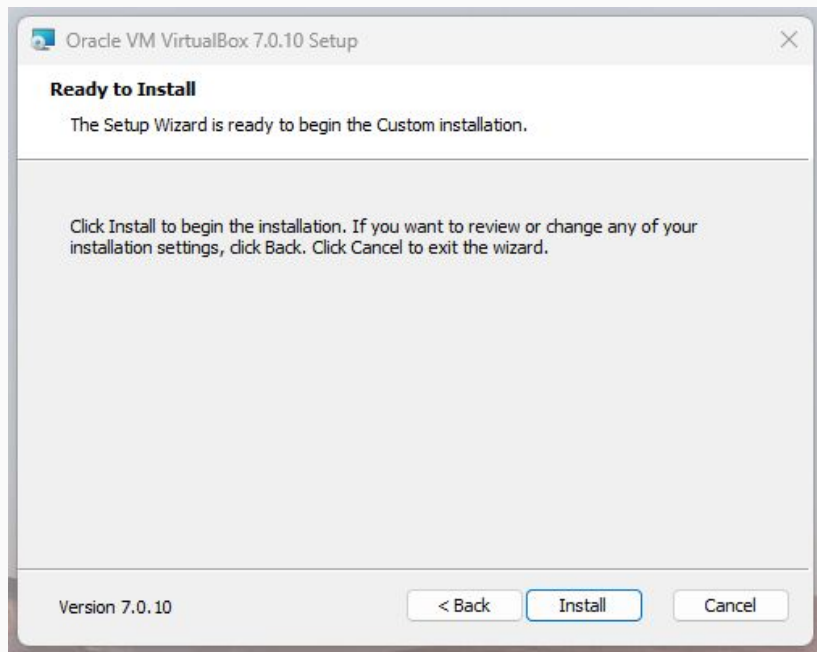
После загрузки установочного файла перейдите в файловом менеджере в директорию с этим файлом, а затем произведите двойной клик по нему - должны начаться установка **Oracle VirtualBox** и появиться окошко, представленное ниже:



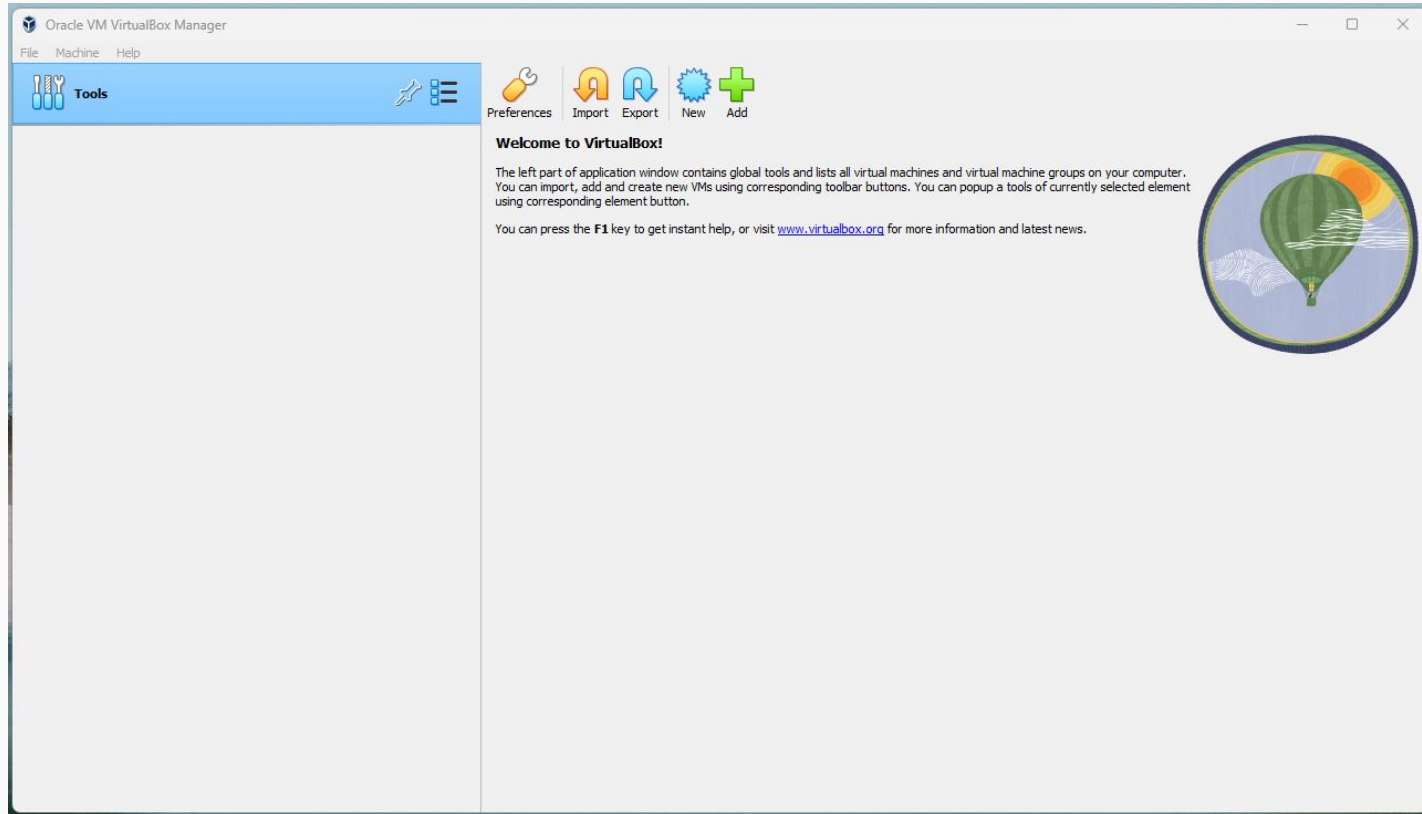
В процессе установки сначала подтвердите, что хотите установить компоненты **Oracle VirtualBox**, а затем согласитесь с отключениями сети во время установки программы. Также подтвердите, что согласны с отсутствием **Python Core / win32api** (если понадобится, то мы установим это потом):



Подтвердите свое финальное согласие с началом процесса установки, нажав кнопку **“Install”** - после этого должны немедленно начаться установка **Oracle VirtualBox**. После установки галочку с предложением открыть программу оставляем открытой и нажимаем кнопку **“Finish”**:



Если все было сделано правильно, нам сопутствовала удача и правильно сошлись звезды (куда же без этого), мы должны увидеть визуальный интерфейс **Oracle VirtualBox**, посредством которого можно создавать виртуальные машины и устанавливать на них операционные системы:



Установка ОС Ubuntu

Перед тем, как установить **Ubuntu**, надо загрузить ее образ к себе на компьютер с официального сайта: <https://ubuntu.com/download/desktop>. Необходимо выбрать **LTS (Long Term Support)** версию, нажать на зеленую кнопку “**Download 22.04.3**” и подтвердить начало загрузки:

Canonical

ubuntu®

Enterprise ▾ Developer ▾ Community ▾ Download ▾

We are hiring Products ▾

Downloads

Desktop Server IoT Cloud

Search 🔍 Sign in

Download Ubuntu Desktop

The open-source desktop operating system that powers millions of PCs and laptops around the world. Find out more about Ubuntu's features and how we support developers and organisations below.

[Ubuntu Desktop homepage](#) [Visit the Ubuntu Desktop blog](#)

Ubuntu 22.04.3 LTS

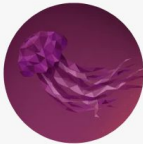
The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

[Ubuntu 22.04 LTS release notes](#)

Recommended system requirements:

- 2 GHz dual-core processor or better
- 4 GB system memory
- 25 GB of free hard drive space

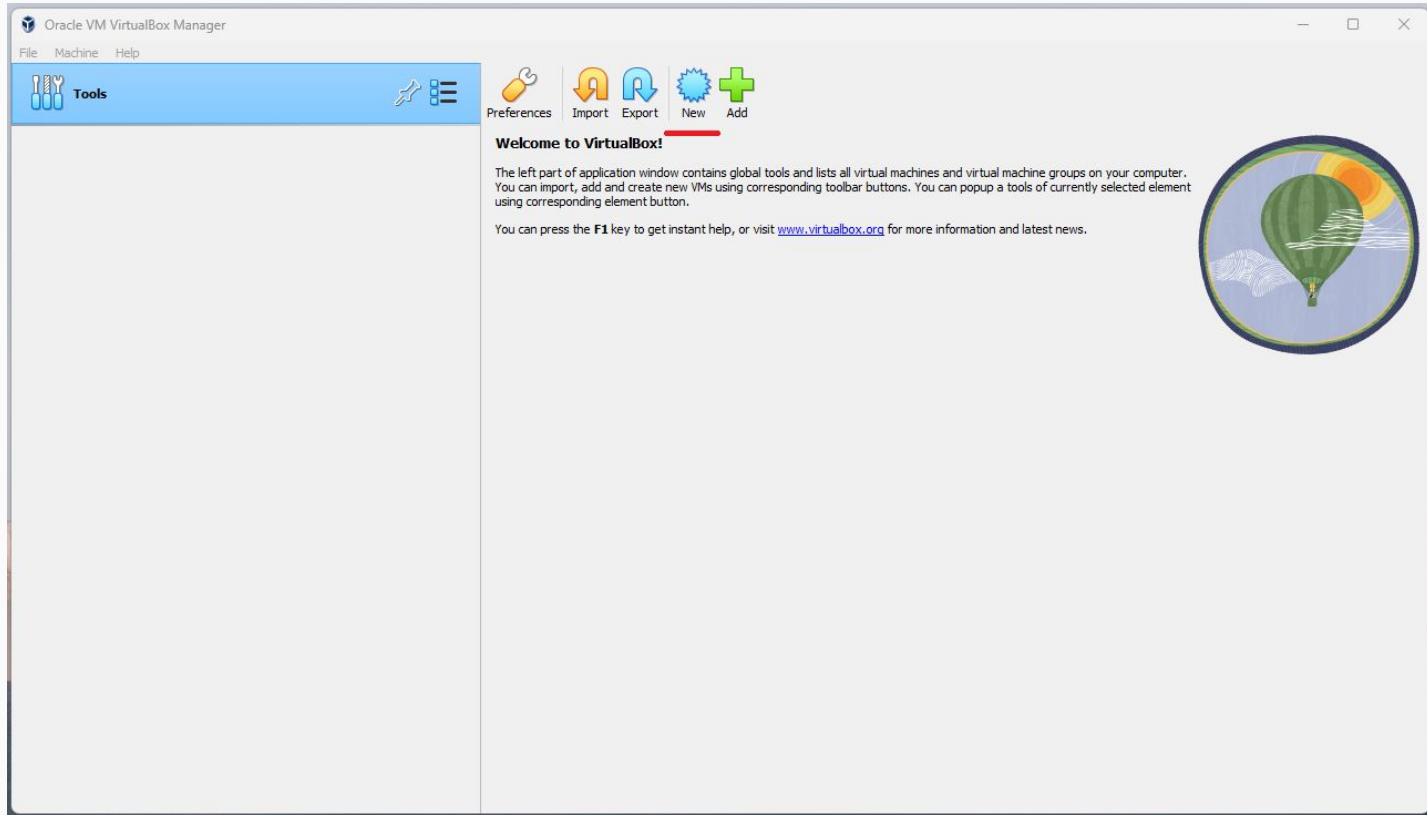
- Internet access is helpful
- Either a DVD drive or a USB port for the installer media



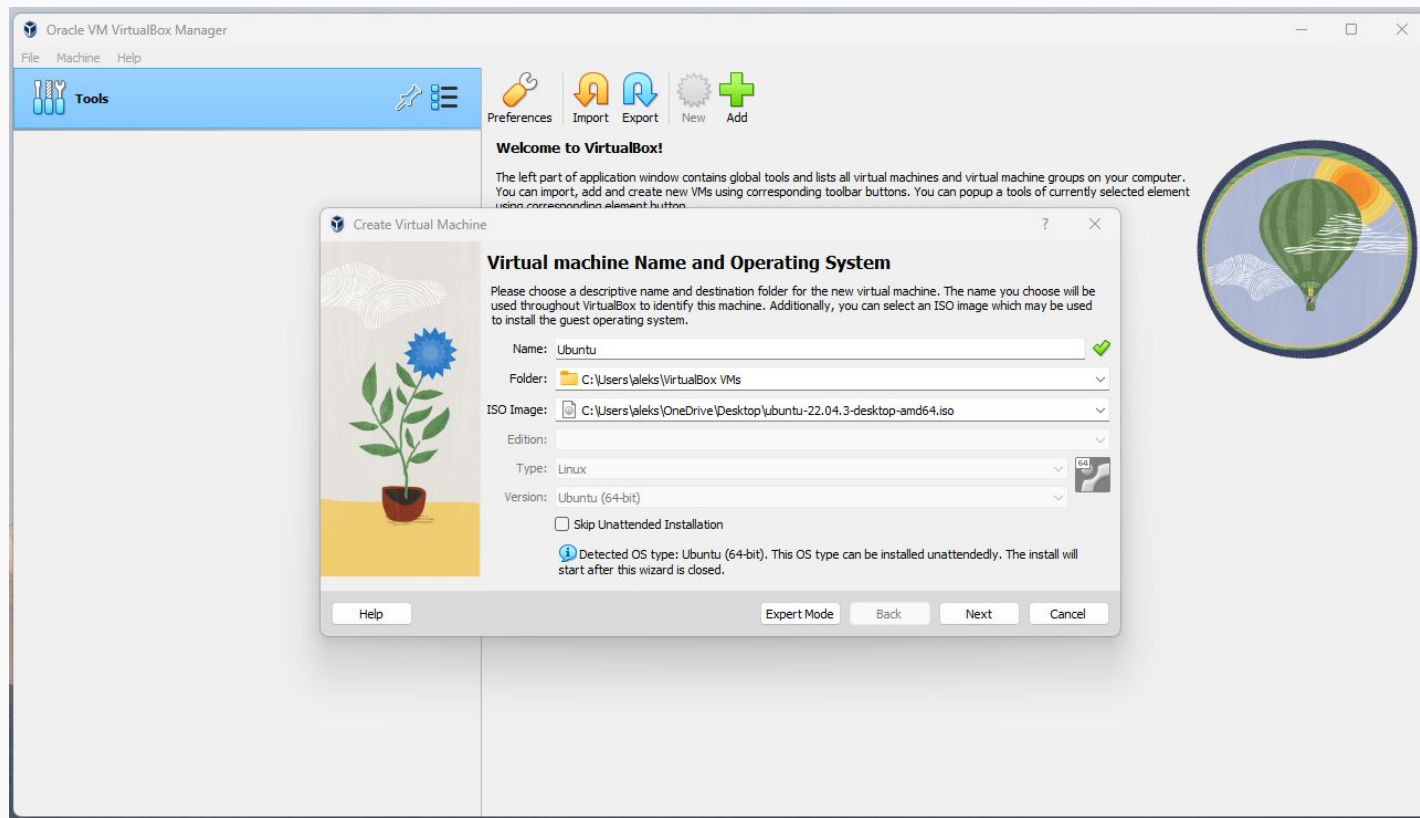
[Download 22.04.3](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors and past releases see [our alternative downloads](#).

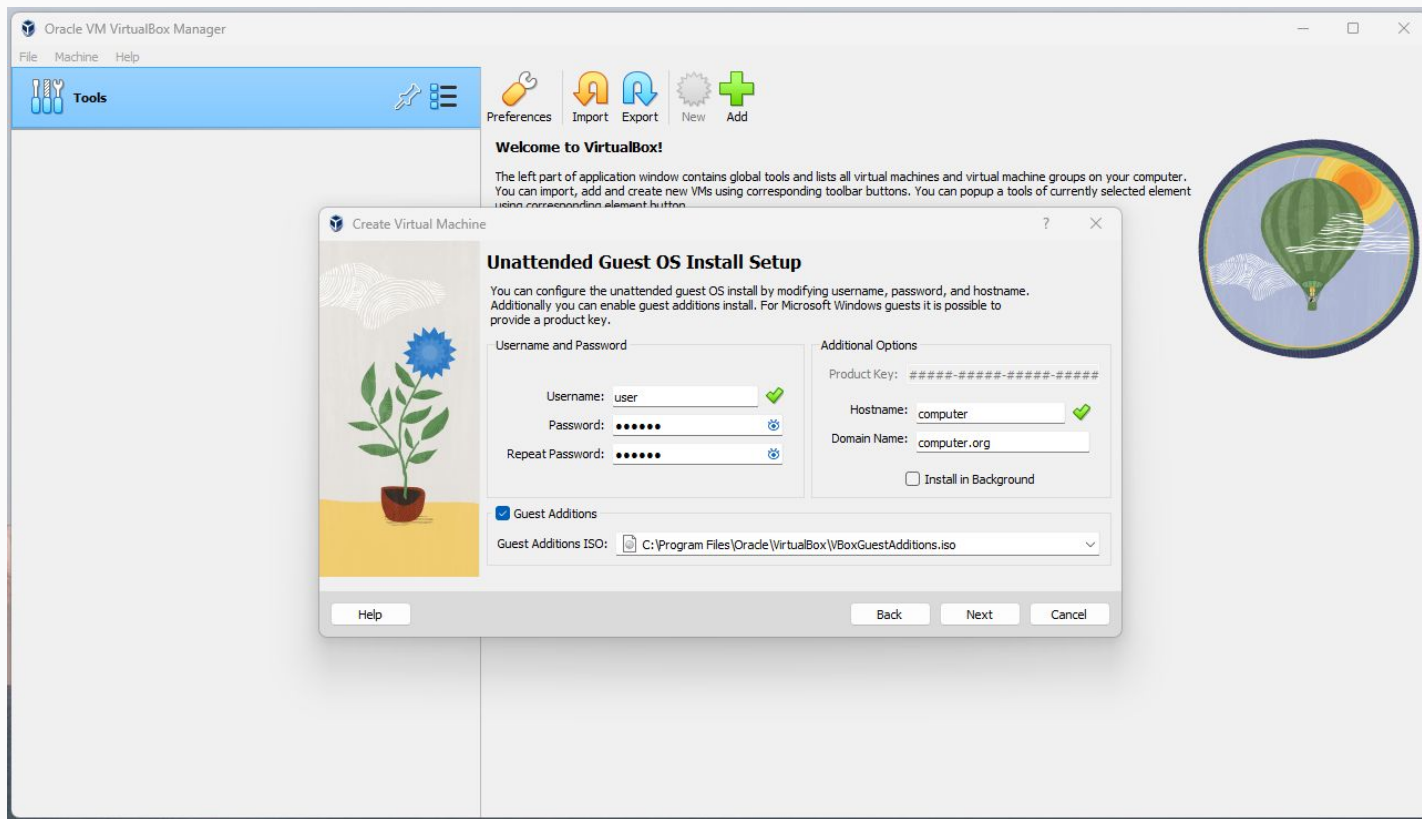
После загрузки образа Ubuntu к себе на компьютер надо начать установку этой операционной системы. Для этого следует запустить **Oracle VirtualBox** (если он еще не запущен), разыскать большую синюю кнопку в виде снежинки с надписью “**New**” и нажать на нее:



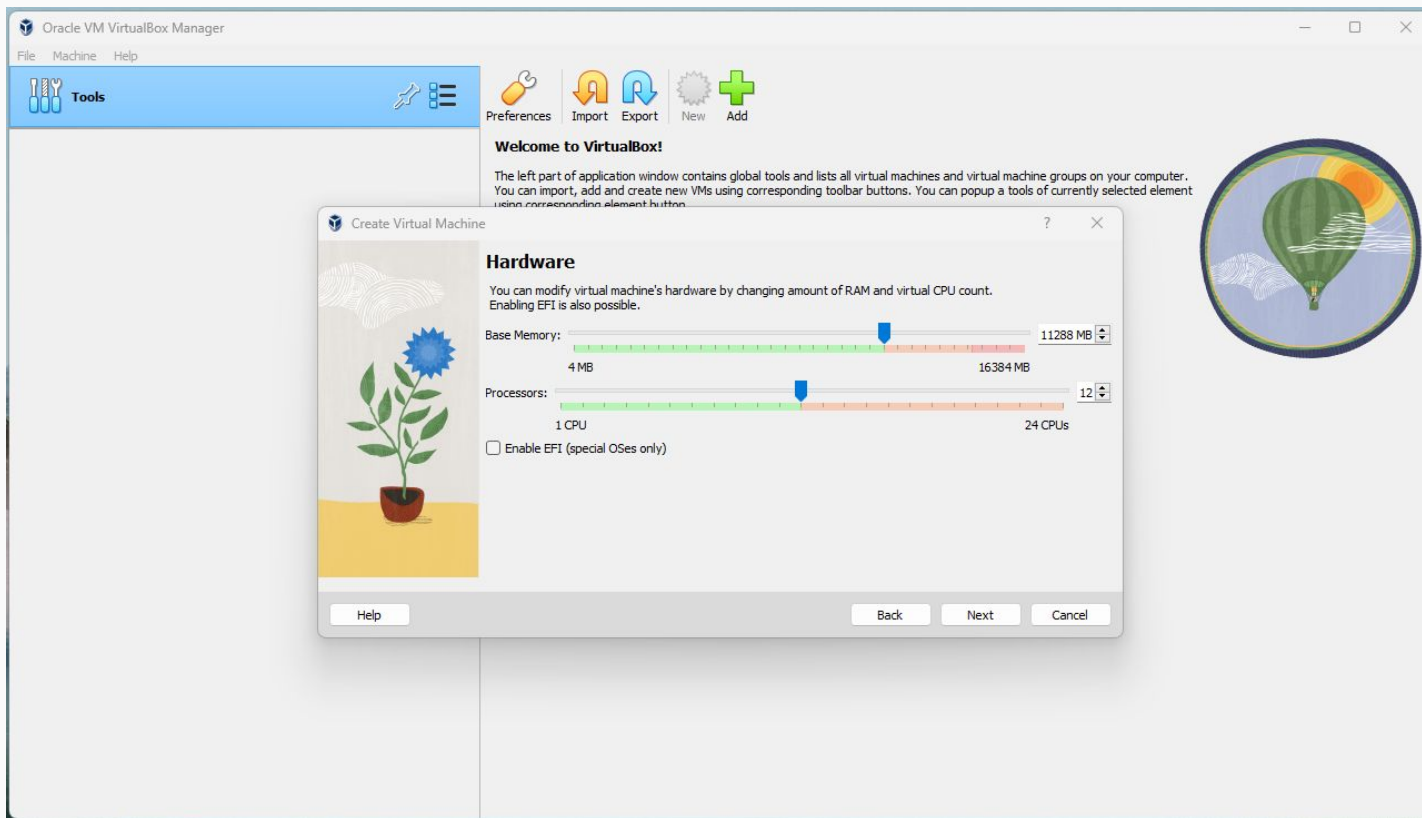
В открывшемся после предыдущего шага окне в поле **“Name”** следует вписать слово **Ubuntu**, а в поле **ISO Image** выбрать путь к загруженному образу **Ubuntu**. Больше никаких настроек трогать не надо. Затем надо нажать кнопку **“Next”**:



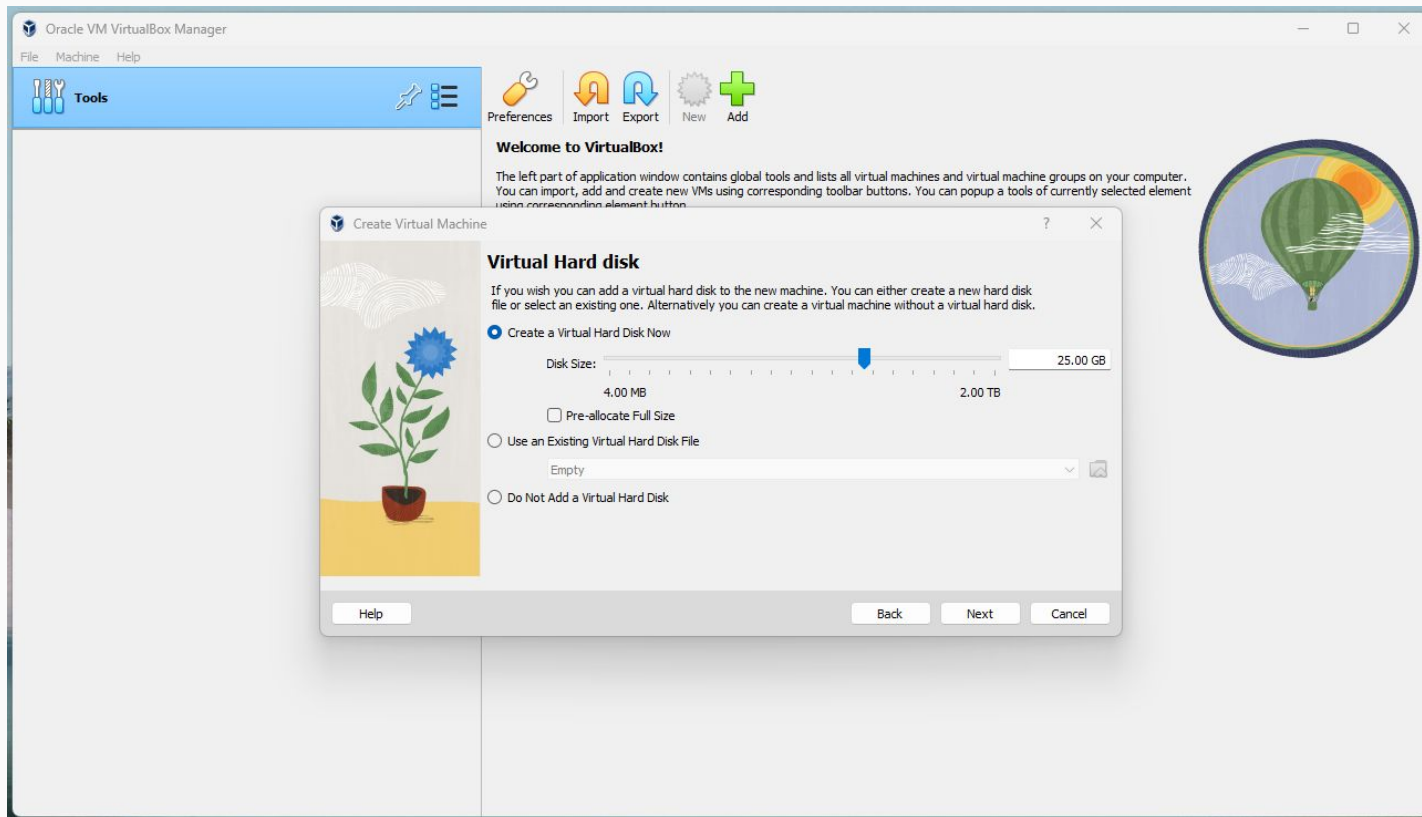
В новом окне в поле “**Username**” следует вписать слово **user**, а слово **secret** установить в качестве пароля. В поле **Hostname** впишем слово **computer**, а в поле **Domain Name** - **computer.org**. Затем сделаем активной галочку **Guest Additions** и нажмем кнопку “**Next**”:



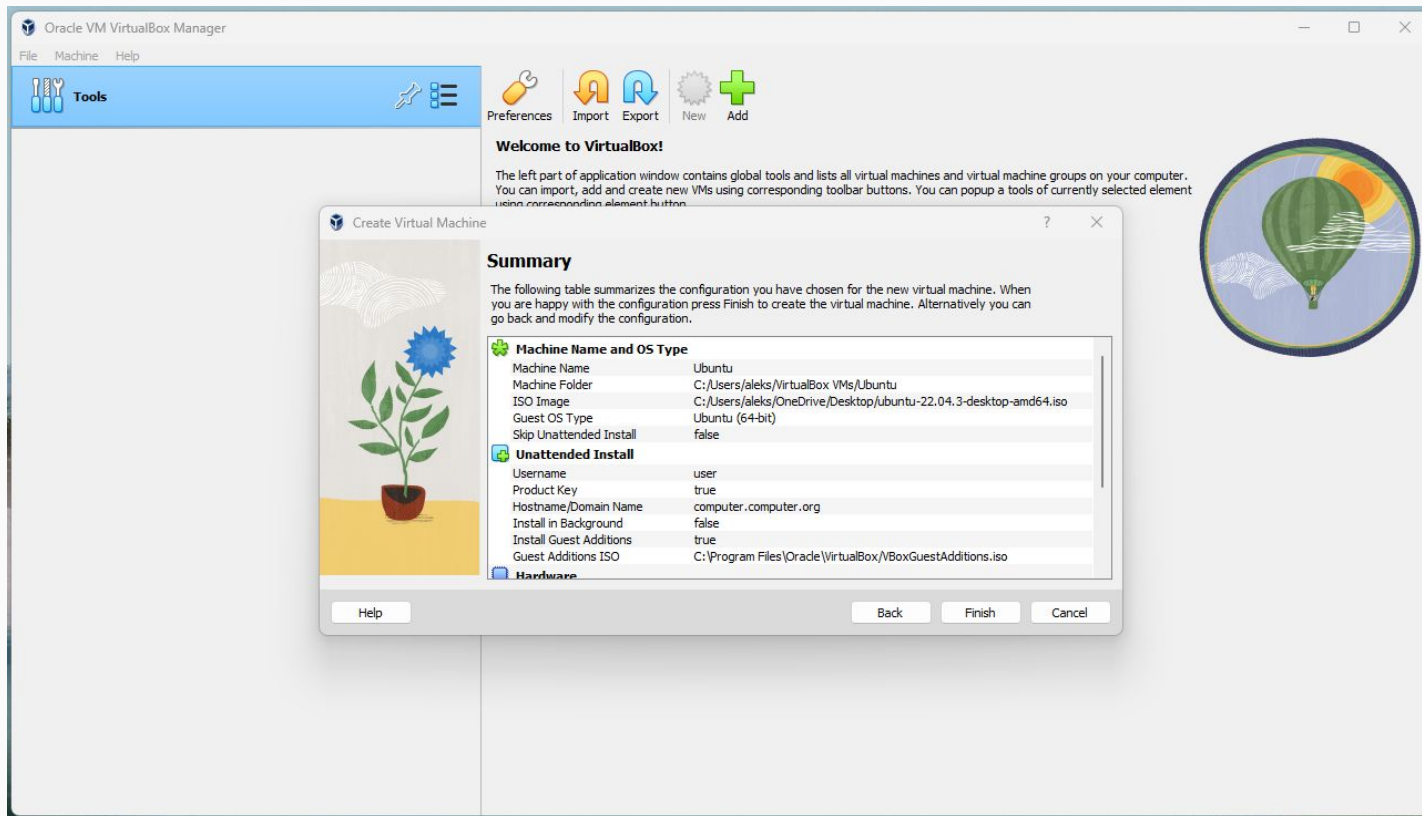
В следующем окошке надо установить разрешенные значения для оперативной памяти (**Base Memory**) и ядер процессора (**Processors**) в контексте нашей виртуальной машины. Лучше всего установить такие значения, которые полностью покроют зеленую полосу, а затем нажать “**Next**”:



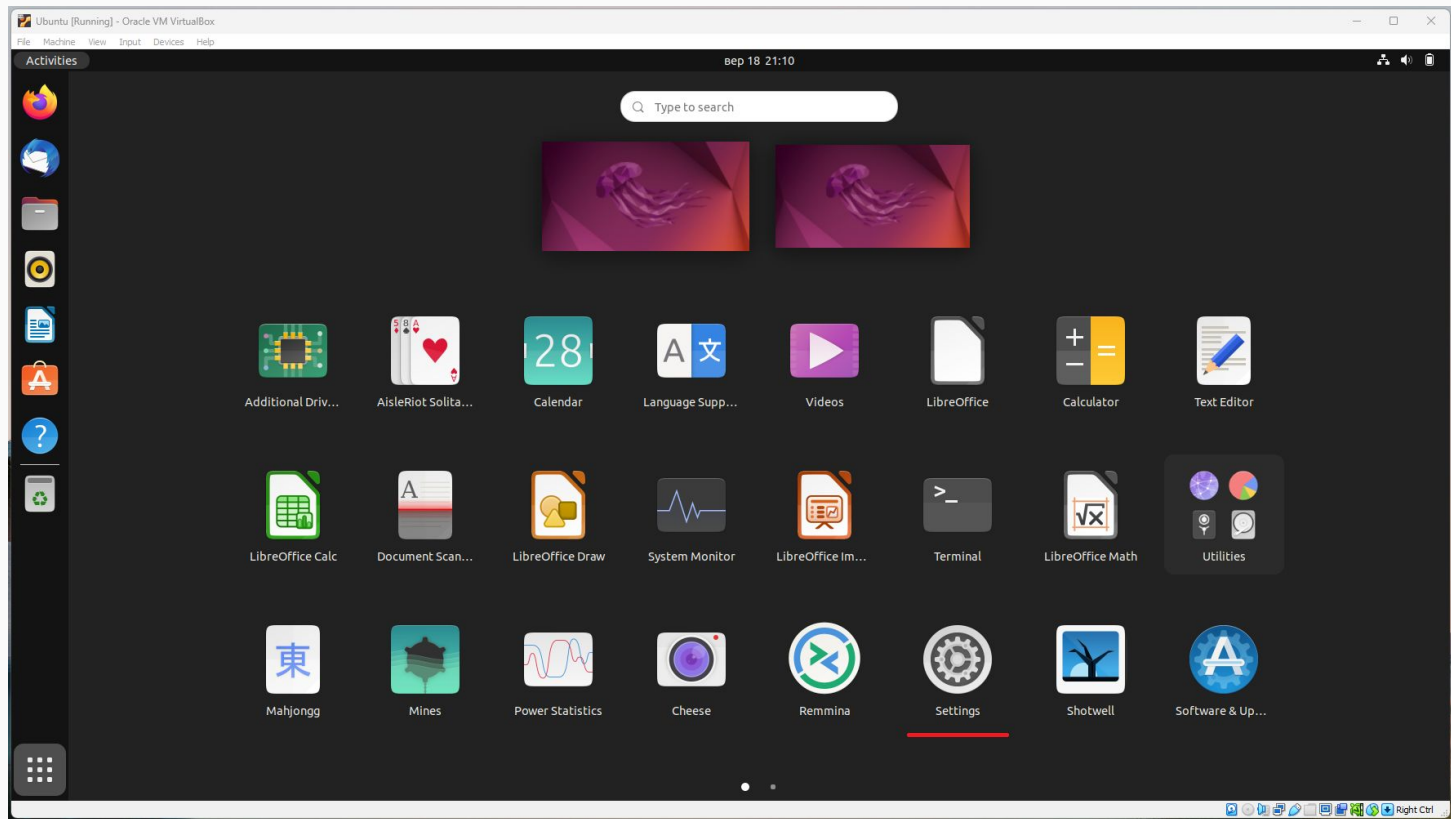
На следующем шаге нам необходимо выбрать размер диска (объем постоянной памяти) для нашей виртуальной машины. Выберем вариант **Create a Virtual Disk Now** и установим минимум 25 GB (а лучше - больше) в качестве размера. Затем нажмем кнопку **“Next”**.



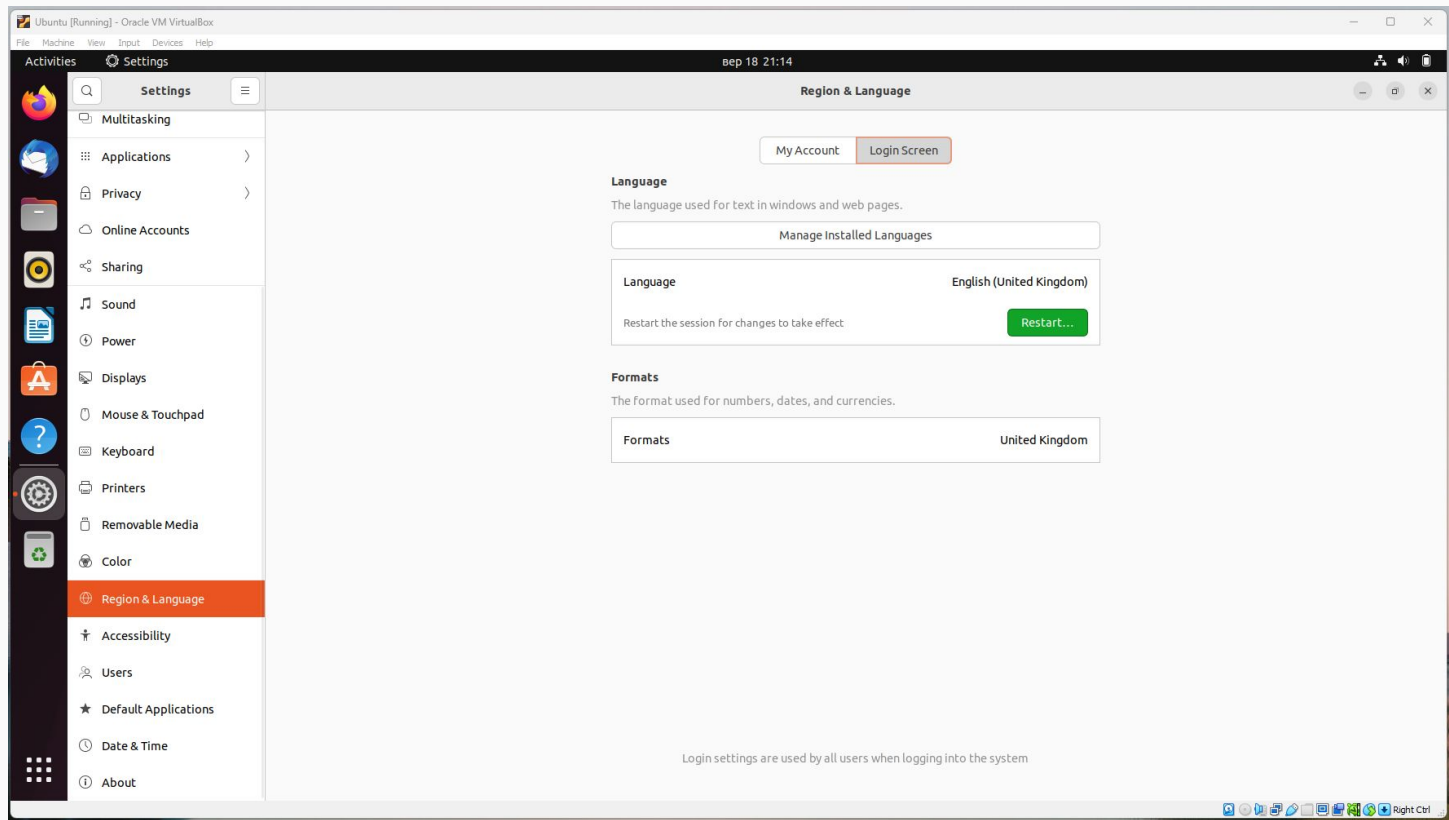
Теперь перед нами сводная таблица всех наших предыдущих действий. Внимательно проверяем, что все заполнено именно так, как нам надо. После этого нажимаем кнопку **“Finish”** - после этого начнется установка операционной системы.



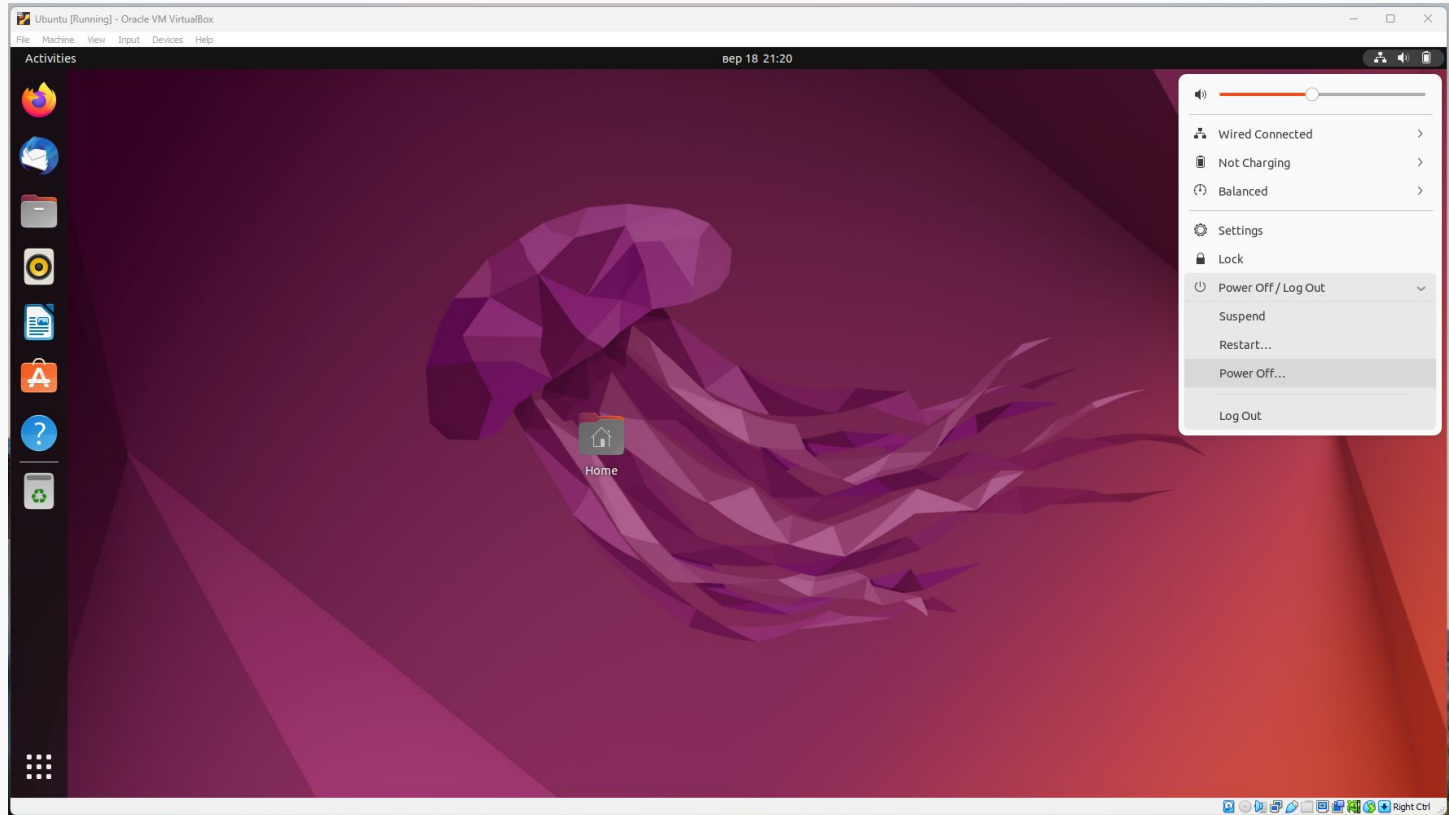
После установки системы перед нами предстанет окно авторизации - выберем пользователя **user**, введем пароль **secret**, а затем окажемся внутри Ubuntu. После пропуска приветственных экранов нажмем в нижнем левом углу квадратик с точками и в открывшемся списке выбираем “**Settings**”:



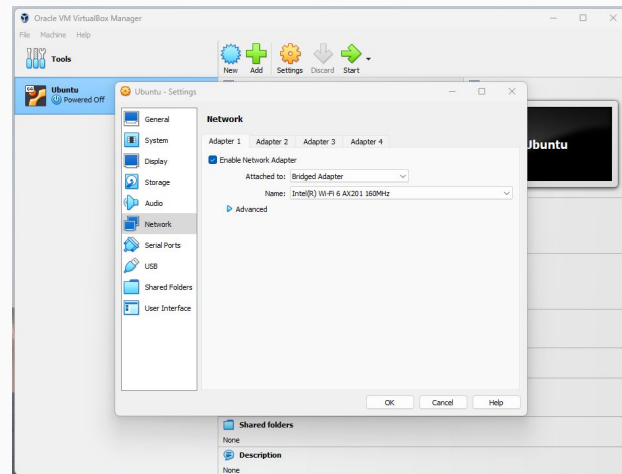
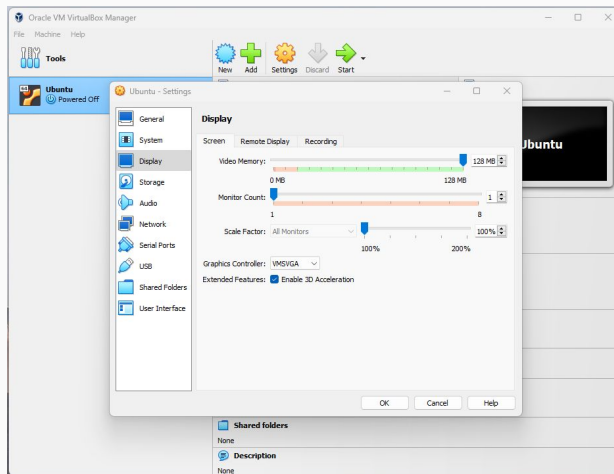
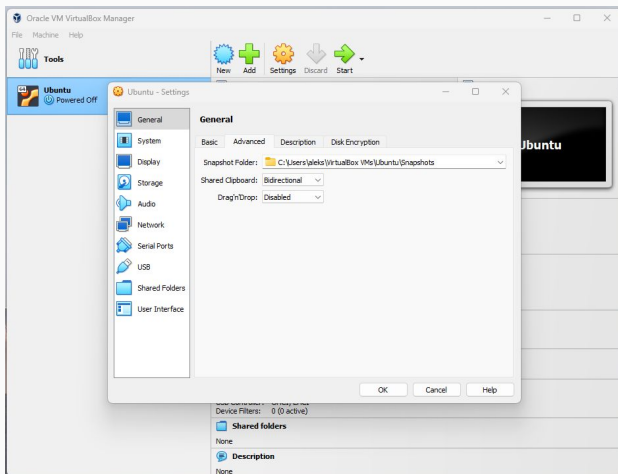
В списке слева выбираем пункт “**Region & Language**” (если его не видно, прокручиваем список вниз), а затем всем опциям (**Language** и **Formats**) как для раздела **My Account**, так и для раздела **Login Screen** выбираем язык **English (United Kingdom)**:



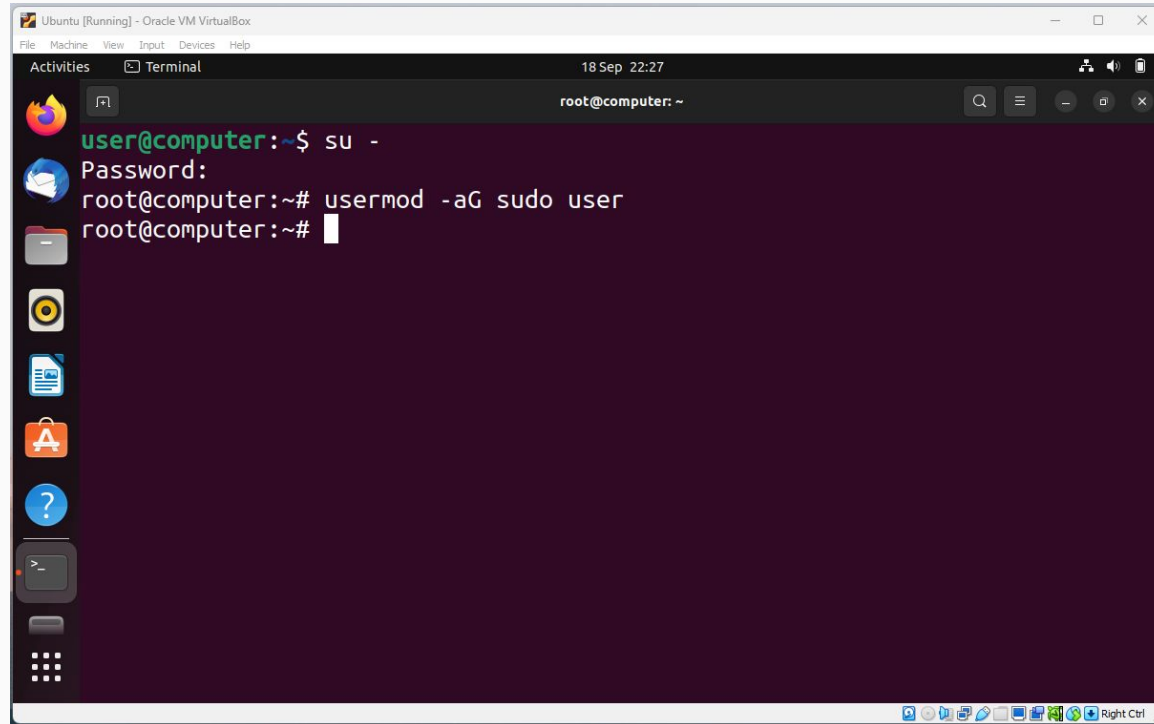
После всех вышеописанных манипуляций нажимаем на опции в верхнем правом углу экрана операционной системы, открываем пункт “**Power Off / Log Out**”, затем выбираем вариант “**Power Off...**” и подтверждаем свой выбор. Это выключит **Ubuntu**, и мы вернемся в **Oracle VirtualBox**:



Нажмем правой кнопкой мыши на пункт **Ubuntu** в списке слева и выберем **“Settings”**. Во вкладке **General/Advanced** в пункте **Shared Clipboard** выберем **“Bidirectional”**, а во вкладке **Display/Screen** выдаем максимально возможное значение настройке **“Video Memory”** и включаем опцию **“Enable 3D Acceleration”**. Затем во вкладке **Network/Adapter 1** в пункте **Attached to** выбираем **“Bridged Adapter”**. После всех этих действий подтверждаем наш выбор нажатие кнопки **“OK”**:



Затем вновь заходим в нашу ОС - двойной клик на надписи **Ubuntu**, после загрузки выбираем пользователя **user** и вводим пароль **secret**. Внутри **Ubuntu** вызываем командную строку комбинацией клавиш **Ctrl+Alt+T**. Внутри командной строки вводим команду **su -** и на предложение ввести пароль пишем **secret** (буквы отображаться не будут) и нажимаем **Enter**. Затем пишем команду **usermod -aG sudo user** и опять нажимаем **Enter**.



Как только мы закончили работу в командной строке, мы должны нажать на опции в правом верхнем углу, выбрать пункт “**Power Off / Log Out**”, затем выбираем вариант “**Restart...**” и подтвердить свой выбор. Когда система перезагрузится, и мы аутентифицируемся как **user** с паролем **secret**, уже можно полноценно пользоваться всеми инструментами **Ubuntu**. Если все прошло успешно - поздравляем! Если возникли какие-либо проблемы - не стесняйтесь, обращайтесь к преподавателю!

