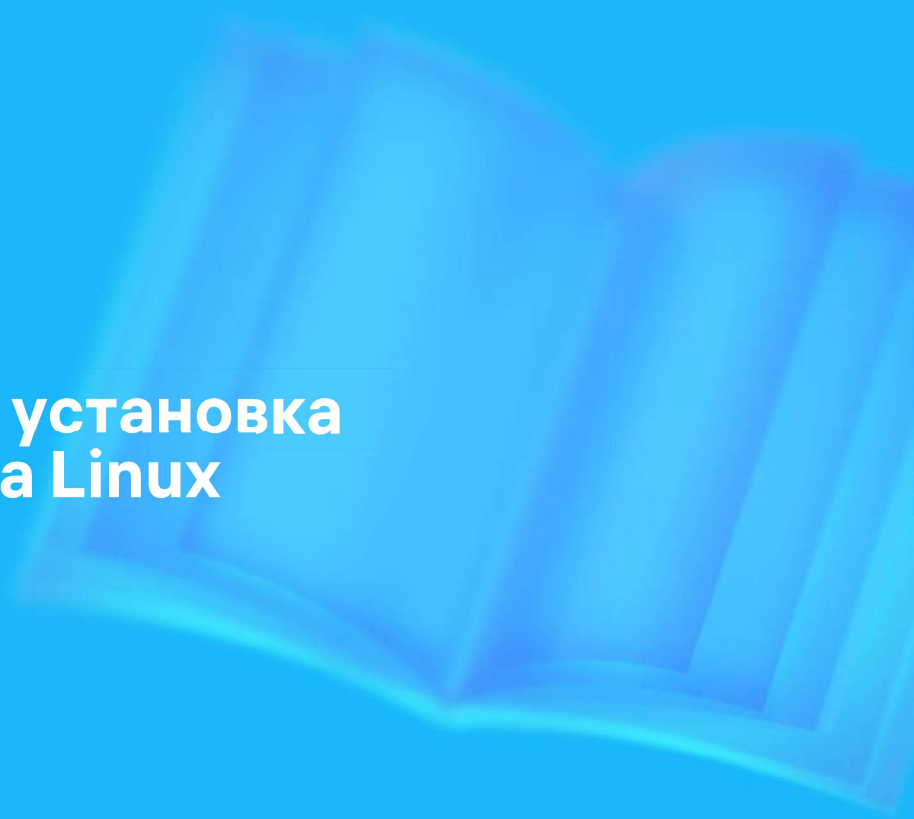




Вебинар №2. Первоначальная установка и настройка Astra Linux



Что будет сегодня



Узнаете об установке
ОС Astra Linux



Познакомитесь с работой
в терминалах,
с командной строкой
и основами работы в ней

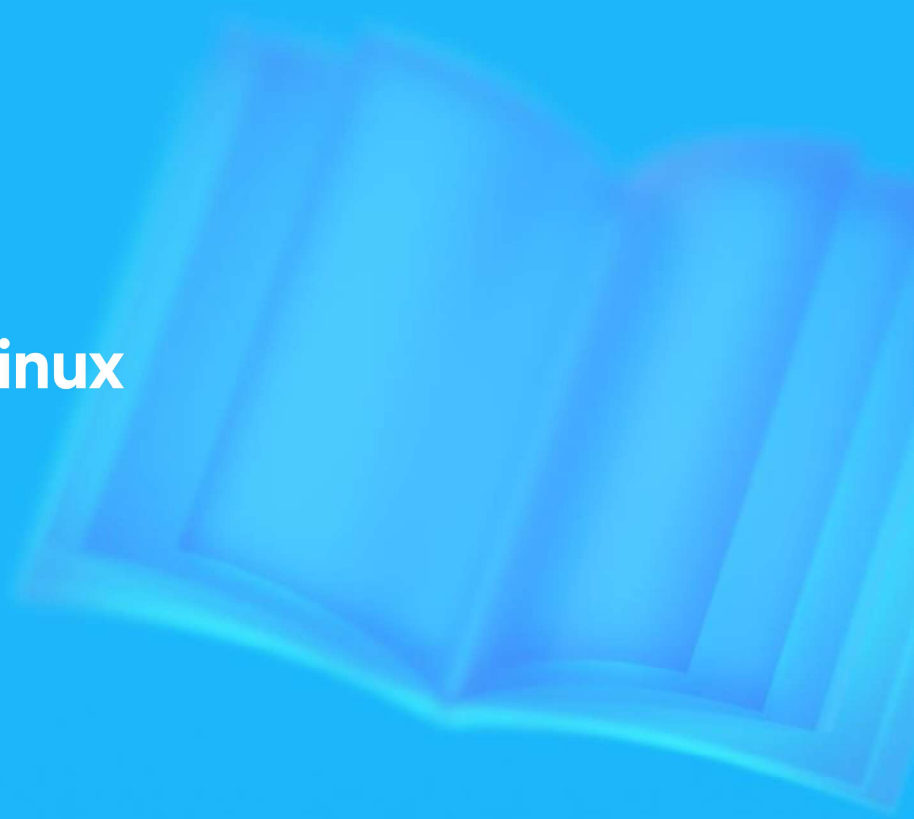


Научитесь настраивать
сеть в Linux, управлять
и настраивать порты



Познакомитесь
с файловыми системами
ОС Unix/Linux и научитесь
управлять ими

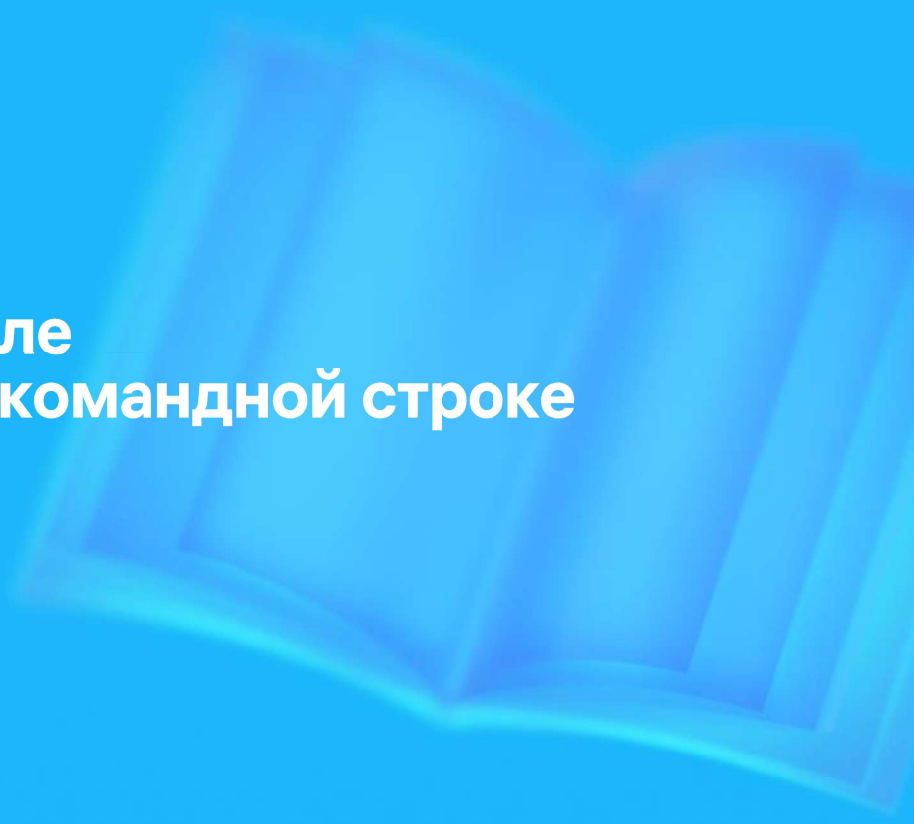
Установка Astra Linux





Работа в терминале

Основы работы в командной строке



Типы терминалов

Аппаратный

Физическое устройство, которое подключается к компьютеру и используется для ввода и вывода текста

Виртуальный

Виртуальное окно на компьютере, которое используется для работы с командной строкой.

Псевдотерминал

Виртуальный терминал, который эмулирует работу физического терминала.

Используется для связи между процессами, которые работают в разных терминалах.

- Для удалённого доступа к серверу через SSH,
- для запуска графических приложений в терминале,
- для запуска скриптов, которые требуют ввода с клавиатуры, и т. д.

ВХОД И ВЫХОД ИЗ СИСТЕМЫ

Для входа в систему необходимо ввести логин и пароль. Ввод пароля в Linux не отображается и всегда остаётся скрытым.

Для выхода из системы можно использовать команду `Exit` или сочетание клавиш `Ctrl + D`.

ДОПОЛНЕНИЕ И ОТСЛЕЖИВАНИЕ КОМАНД И ИМЁН ПЕРЕМЕННЫХ

В Linux для дополнения команд и имён переменных используется утилита `bash-completion`. Она позволяет автоматически дополнять команды и имена переменных при наборе в терминале.

Дополнение команд и имён переменных может быть выполнено с помощью клавиши `Tab`.



СТРУКТУРА КОМАНДНОЙ СТРОКИ

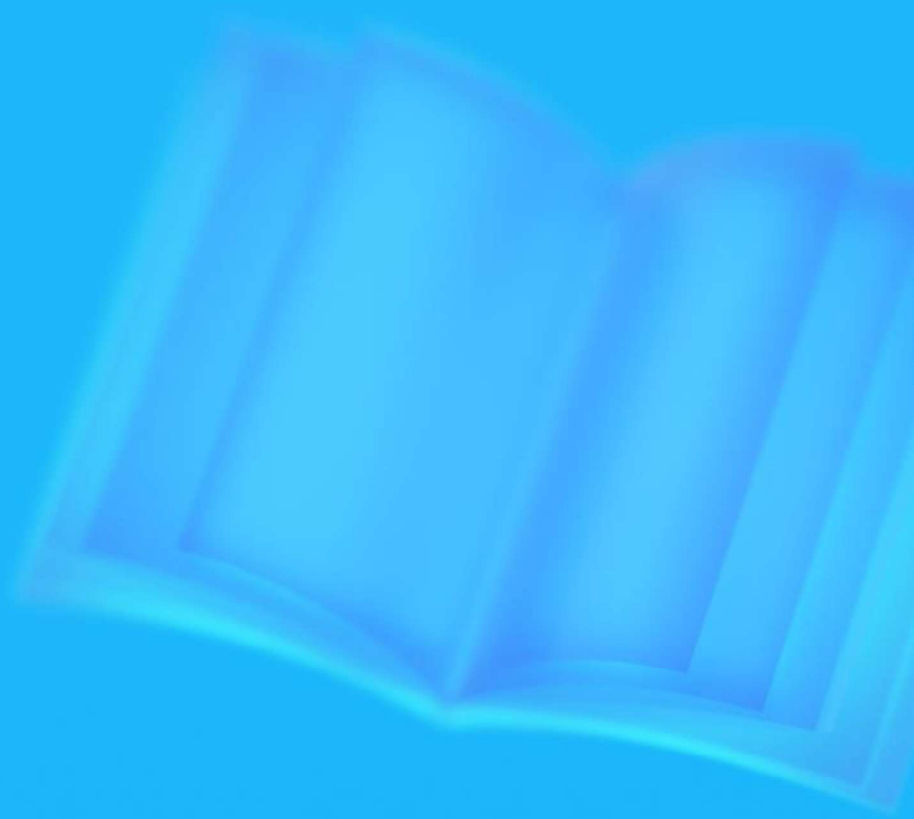
Командная строка состоит из имени команды, параметров и опций.

РАБОТА С ПЕРЕМЕННЫМИ

Для Unix-совместимых ОС интерфейсом являются оболочки (shell), которые позволяют запускать программы и получать их вывод. Самая популярная оболочка для Linux — это bash.

Каждый раз, когда пользователь подключается к своему Linux, запускается сеанс оболочки. Каждый экземпляр оболочки в момент старта получает набор данных и настроек, которые и называются переменными окружения.

Настройка сети




В Linux сетевые интерфейсы — это способ подключения компьютера к сети. Они используются для обеспечения связи между различными устройствами в сети, а также для предоставления доступа к локальным и удалённым ресурсам.

В Linux сетевые интерфейсы могут быть настроены как статически, так и динамически с помощью таких инструментов, как `ifconfig`, `ip` и `Network Manager`.

Linux поддерживает множество протоколов сетевого уровня, включая Ethernet, Wi-Fi, Bluetooth и другие.

Существует возможность применения различных сетевых настроек, таких как `bonding` и `teaming`.



Настройка сетевых интерфейсов с помощью ifup/ifdown (служба networking)

Конфигурационные файлы сетевых интерфейсов обычно находятся в директории `/etc/network/interfaces`. Для просмотра состояния интерфейсов можно использовать команду `ifconfig` или `ip a`.

Службу `networking` в Linux можно использовать для настройки сетевых интерфейсов с помощью команд `ifup` и `ifdown`.

Служба Network Manager — это удобный и гибкий инструмент для настройки сетевых интерфейсов с помощью командной строки в Linux.

Как использовать инструмент `nmtui` в командной строке Linux для настройки сетевых интерфейсов с помощью Network Manager?

Для настройки сетевых интерфейсов также можно использовать `nmtui` — псевдографический инструмент для Network Manager, который позволяет настраивать несколько соединений, включая Ethernet, Wi-Fi, VPN и т. д.

Команды диагностики сети

Команды диагностики сети в Linux помогают в анализе сетевого трафика и определении наличия проблем в сети.

Ping: отправляет ICMP-пакет на заданный хост и ждёт ответа. Результат показывает время отклика и количество потерянных пакетов.

Сетевые порты

В Linux сетевой порт — это числовой идентификатор сетевого соединения или сервиса, который позволяет множеству приложений использовать сетевую подсистему. Каждый порт — это 16-битное число от 0 до 65535. Многие порты зарезервированы для известных служб, таких как HTTP (порт 80), HTTPS (порт 443), SSH (порт 22), FTP (порт 21) и т. д.

Команда `netstat` выводит список открытых сетевых портов.

Понимание сетевых портов в Linux может помочь:

- в настройке и конфигурации сетевых служб;
- заблокировать уязвимые порты, повышая безопасность сети и устойчивость к угрозам;
- в обнаружении проблем в сети и в настройке правил сетевой защиты, чтобы предотвратить несанкционированный доступ к системе.

Контроль открытых портов может помочь в улучшении производительности сетевых сервисов. Например, закрытие ненужных портов может снизить нагрузку на систему и улучшить её производительность.

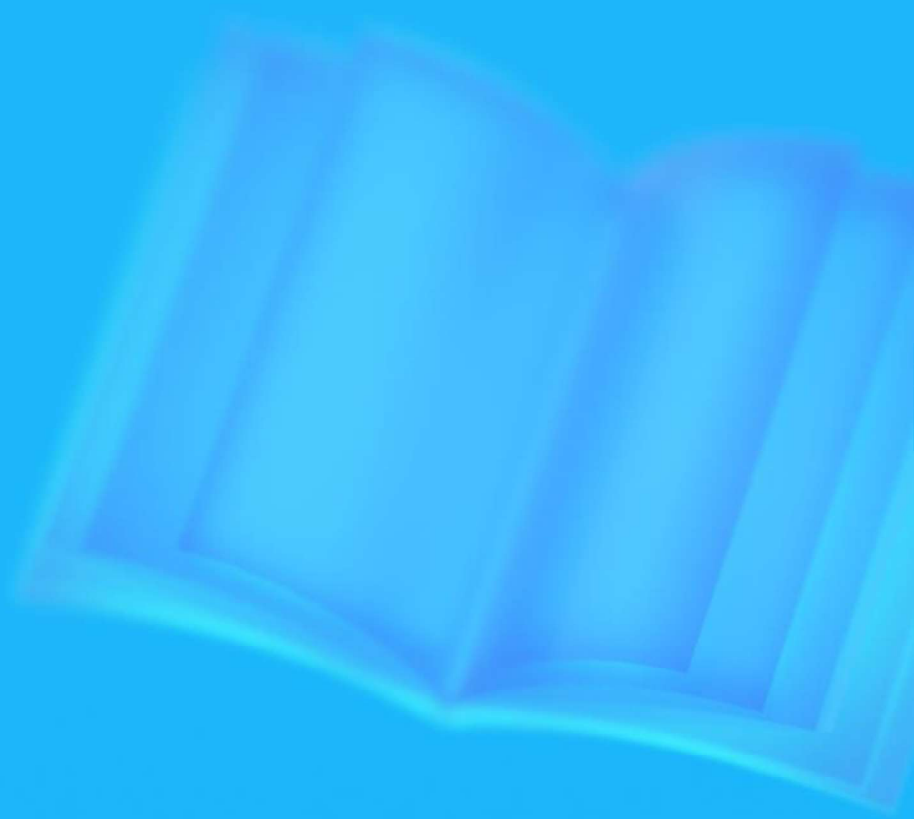
Настройка сервиса блокирования сетевых портов iptables

Конфигурация сетевого экрана iptables может быть сложной задачей, но, следуя некоторым базовым правилам, мы можем с лёгкостью настроить правила блокирования портов в Linux. Разберём процессы:

1. Создание правила блокирования порта.
2. Сохранение правил iptables.
3. Настройка правил iptables для автозагрузки системы.

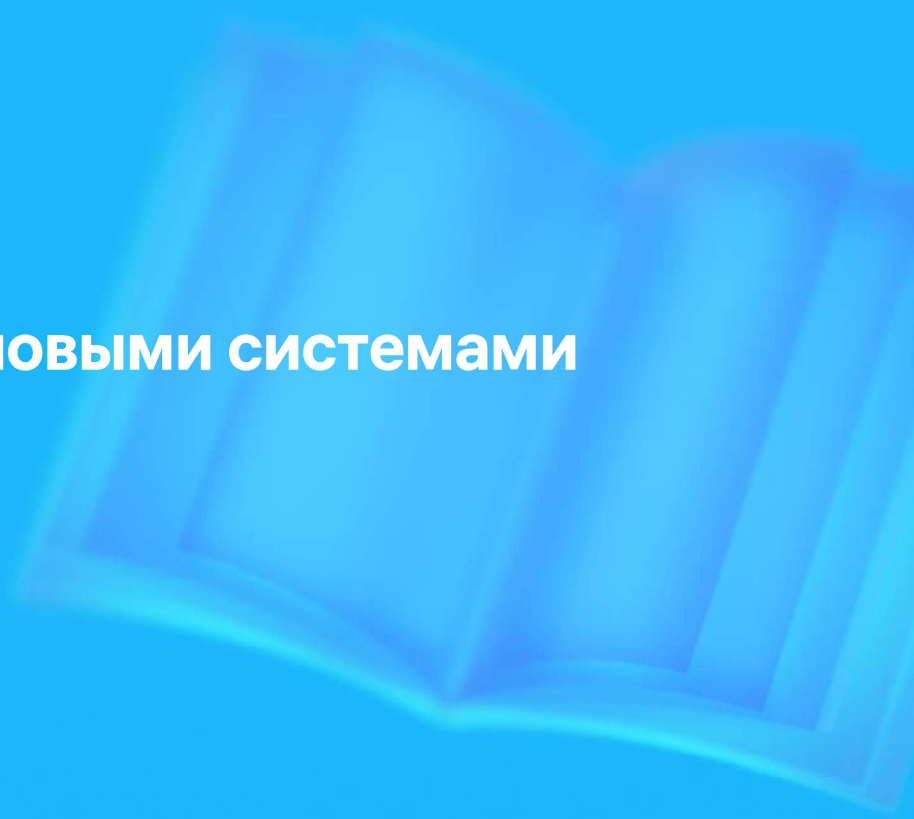
После перезагрузки системы можно использовать команду iptables -L для проверки действующих правил iptables. Если правила настроены верно, соединение с заблокированным портом должно отбрасываться.

Вопросы





Управление файловыми системами



Архитектура подсистемы хранения данных

Подсистема хранения данных в Linux состоит из нескольких компонентов, каждый из которых выполняет свою специфическую функцию.

1. Файловые системы.
2. Драйверы устройств.
3. Утилиты командной строки.
4. Графические интерфейсы.
5. Системные вызовы.

Настройка и контроль работы дисковых устройств

Настройка и контроль работы дисковых устройств в Astra Linux включает в себя несколько этапов:

1. Разметка диска.
2. Форматирование разделов.
3. Монтирование разделов.
4. Контроль состояния дисков.
5. Мониторинг дисков.
6. Шифрование дисков.

Именованние файлов дисковых устройств

В Linux файлы дисковых устройств обычно именуются в соответствии с соглашением об именовании устройств. Это соглашение определяет, как имена файлов дисковых устройств будут выглядеть и как они будут связаны с физическими устройствами.

В Linux дисковые устройства именуются в соответствии с шаблоном. Имена файлов дисковых устройств могут отличаться в зависимости от дистрибутива Linux и конфигурации системы.

Поддерживаемые типы ФС в Astra Linux

Список файловых систем, которые поддерживаются ядром Linux, находится в файле `/proc/filesystems`.

Файловые системы семейства ext:

EXT — расшифровывается как Extended File System, расширенная файловая система.

Семейство ext включает в себя несколько версий:

Ext2 — поддерживает файлы размером до 2 ТБ и объем раздела до 32 ТБ, не имеет журналирования.

Ext3 — поддерживает файлы размером до 2 ТБ и объем раздела до 32 ТБ, имеет журналирование.

Ext4 — поддержка файлов размером до 16 ТБ и объем раздела до 1 экзбайта, имеет множество новых функций:

- имеет журналирование;
- поддерживает расширяемые атрибуты файлов;
- блокирование файлов на уровне ядра;
- быстрое создание файлов и многое другое.

Структура файловых систем ext

Файловая система ext состоит из:

- Блоков, которые делятся на группы.
 - Каждая группа содержит таблицу дескрипторов блоков (inode table).
 - Дескрипторные блоки хранят информацию о файлах и каталогах, а также данные блоков, которые содержат сами файлы и каталоги.
- Дерево каталогов для организации файлов и каталогов. Каждый каталог может содержать подкаталоги и файлы, а также ссылки на другие файлы и каталоги.

Создание разделов

Создание разделов на дисках в Linux может быть выполнено с помощью утилиты `fdisk` или `parted`. `Fdisk`, как правило, предустановлена на Astra Linux, однако если её нет, выполните команду:

```
sudo apt install fdisk.
```

Для `parted` — команду:

```
sudo apt install parted.
```



Утилита `fdisk`

Утилита `fdisk` позволяет создавать и управлять разделами на диске.

Для использования этой утилиты необходимо запустить её с правами суперпользователя (через `sudo`).

После запуска утилиты необходимо:

- Выбрать диск, на котором нужно создать разделы, с помощью команды «`fdisk /dev/sdX`», где `X` - буква диска;
- затем нужно использовать команды для создания разделов, изменения их размеров и типов файловых систем.

Создание файловых систем (форматирование). Утилита mkfs

После создания разделов необходимо отформатировать их с помощью соответствующей утилиты для выбранной файловой системы.

Все файловые системы создаются утилитой `mkfs.<FS_NAME>` (`**mkfs.ext4**`, `**mkfs.xfs**`, ..).

Монтирование файловых систем вручную. Параметры монтирования файловых систем

Чтобы смонтировать файловые системы при запуске компьютера вручную, необходимо:

1. Создать точку монтирования, которая будет использоваться для подключения файловой системы.
2. Определить устройство, на котором находится нужная файловая система. Для этого можно использовать команду «`sudo fdisk -l`» или «`sudo blkid`».
3. Смонтировать файловую систему в созданную точку монтирования с помощью команды «`sudo mount /dev/device /mnt/point`».
4. Проверить, что файловая система успешно примонтирована, используя команду «`mount`».

Монтирование файловых систем автоматически при загрузке компьютера. Параметры монтирования файловых систем

Чтобы смонтировать файловые системы при запуске компьютера автоматически, необходимо:

1. Определить устройство, на котором находится нужная файловая система, используя команду «`sudo fdisk -l`» или «`sudo blkid`».
2. Отредактировать файл `/etc/fstab` с помощью текстового редактора, добавив строку с параметрами монтирования для нужной файловой системы. Например, для монтирования файловой системы `ext4` на устройстве `/dev/sda1` в точку монтирования `/mnt/point` при загрузке компьютера добавьте следующую строку:
`/dev/sda1 /mnt/point ext4 defaults 0 0`.
3. Сохранить изменения в файле `/etc/fstab`.
4. Перезагрузить компьютер и проверить, что файловая система успешно примонтирована, используя команду «`mount`».

С помощью утилиты `lsblk` можно увидеть добавленный диск и общий список дисков на машине.

С флагом `-f` будут выведены также файловые системы на дисках (`lsblk -f`).

Управление логическими томами (Logical Volume Manager)

Logical Volume Manager или lvm — это система управления логическими дисками на одном или нескольких физических дисках без переразметки физических дисков.

Преимущества использования логических дисков под управлением lvm перед разделами физического диска:

- Изменение размера.
- Расширяемость пространства.
- Зеркалирование данных.
- Резервное копирование.
- Читаемые имена.

Три сущности LVM

LVM управляет тремя сущностями:

1. Физический раздел (может включать в себя несколько физических дисков) — Physical Volume (PV).
2. Группа физических дисков — Volume Group (VG).
3. Логический диск — Logical Volume (LV).

Создание физических томов:

`$ sudo pvcreate /dev/vdb1 /dev/vdb2` — просмотр с помощью `pvs` или `pvdisplay`.

Создание групп томов:

`$ sudo vgcreate my_vg /dev/vdb1 /dev/vdb2` — просмотр с помощью `vgs` или `vgdisplay`.

Создание логических томов:

`$ sudo lvcreate -l +100%FREE -n my_vol my_vg` — просмотр с помощью `lvs` или `lvdisplay`.

Изменение размеров логических томов и файловых систем. Дополнительные возможности LVM (трюки)

```
$ sudo vgextend my_vg /dev/vdb3  
$ sudo lvextend -l +100%FREE my_vg/my_vol  
$ sudo lvreduce -L-500M /dev/my_vg/my_vol
```

Перемещение раздела (при переносе со старого на новый диск):

```
$ pvmove -n my_vol /dev/sda1
```

Резервное копирование логического раздела:

```
$ lvcreate -s -n snap -L 5g my_vg/my_vol
```

Откат:

```
$ lvconvert --merge my_vol/snap
```

Удаление:

```
$ lvremove /dev/my_vg/my_vol
```

Шифрование дисков

Шифрование дисков в Linux — это процесс защиты данных на жестком диске путём преобразования их в неразборчивый вид для посторонних лиц. Шифрование может быть выполнено как для всего диска, так и для отдельных разделов.

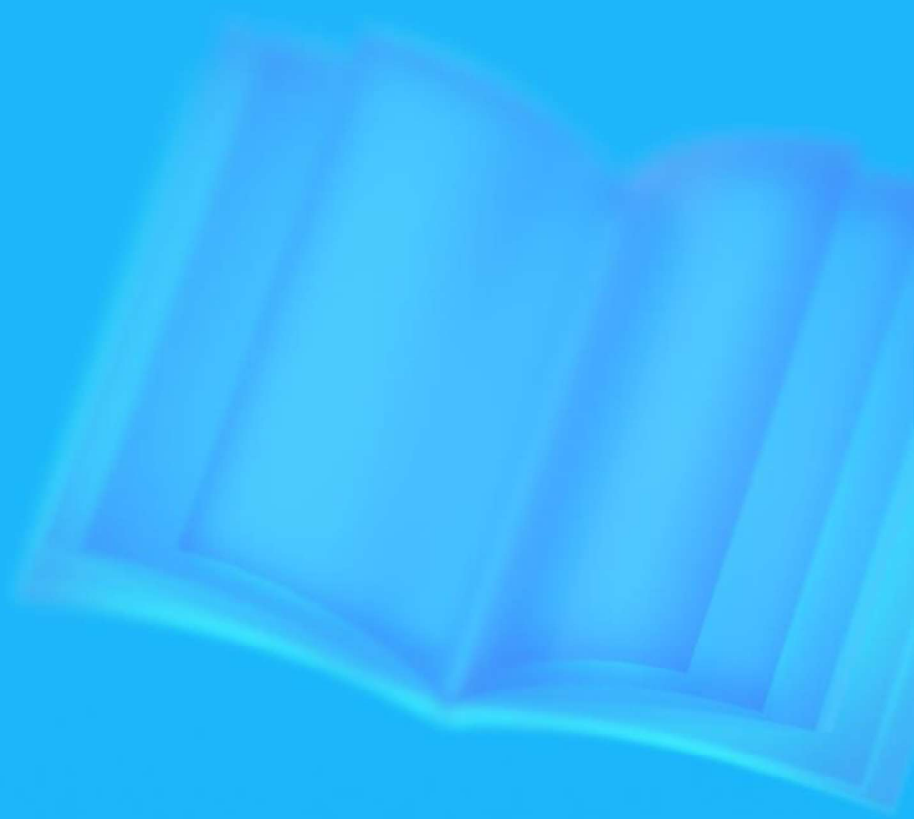
В Linux существует несколько инструментов для шифрования дисков:

- Dm-crypt.
- LUKS (Linux Unified Key Setup).
- TrueCrypt.
- VeraCrypt.

Для шифрования диска или раздела в Linux необходимо выполнить следующие действия:

1. Установить программное обеспечение для шифрования (например, dm-crypt и LUKS).
2. Создать новый зашифрованный раздел или шифровать существующий раздел.
3. Назначить пароль или ключ шифрования.
4. Смонтировать зашифрованный раздел для доступа к данным.

Вопросы



Спасибо за внимание!

