# Резервное копирование

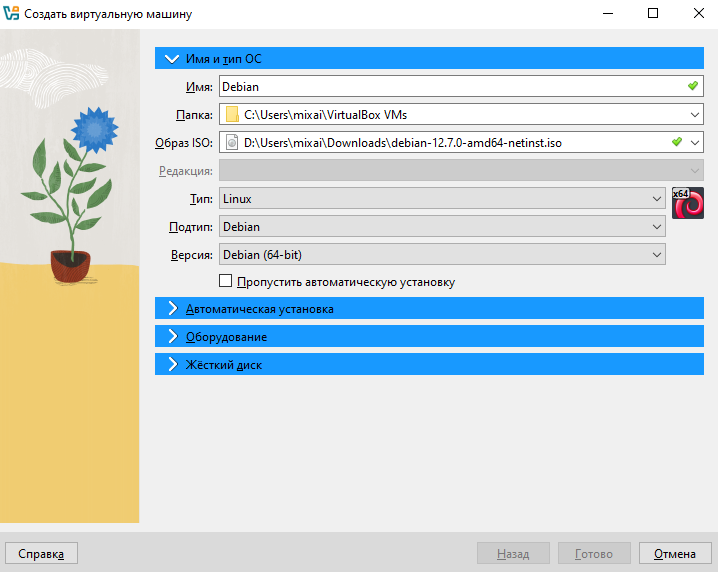
1) Установка виртуальной машины VirtualBox:

<https://www.virtualbox.org/wiki/Downloads>

Просто следуем инструкции.

2) Установка Debian (с Ubuntu были проблемы при установке):

<https://www.debian.org/download>



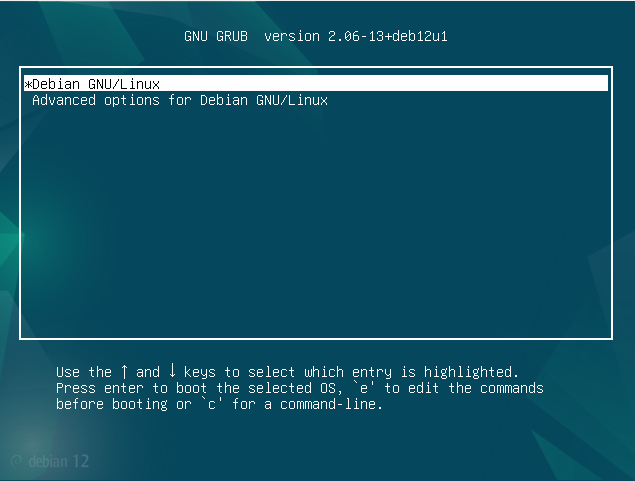
В виртуальной машине с помощью скачанного образа устанавливаем Debian.

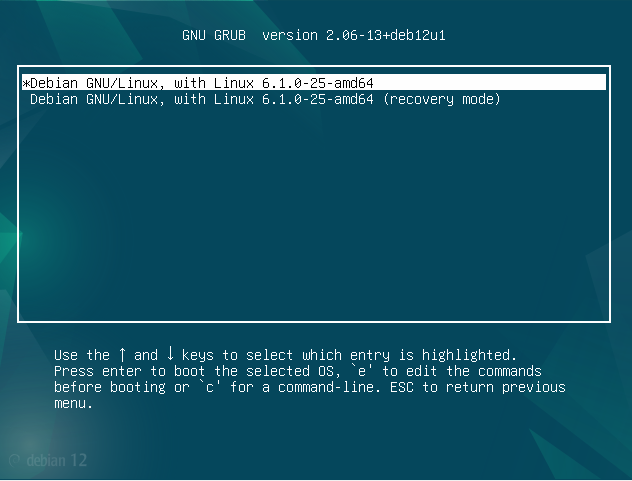
3) Установка пользователя root.

Для работы с терминалом в виртуальной машине Debian мне нужны были права root, но во время установки нигде мне поставить пароль для этого пользователя не предлагалось.

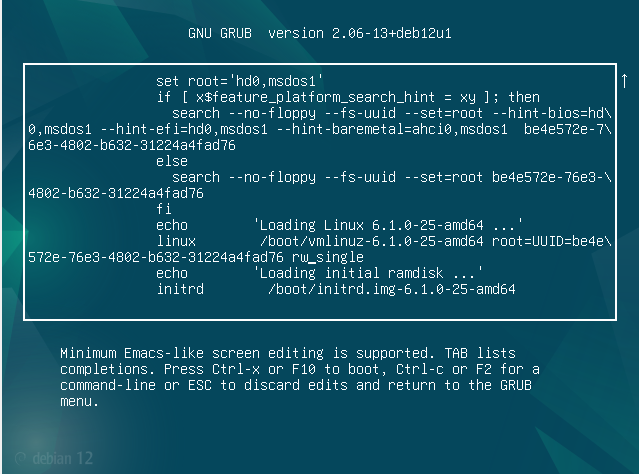
1. Перезагружаем виртуальную машину и во время запуска нажимаем ‘shift’.

2. В GRUB выбираем Advanced options (вторая строка).

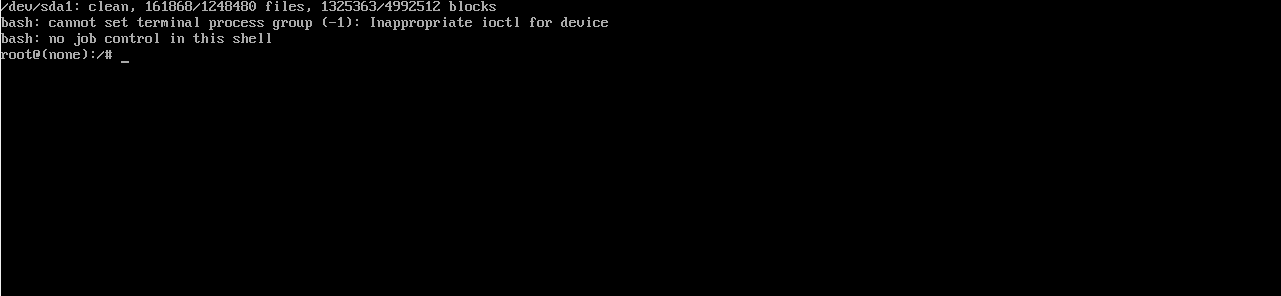


3. В следующем окне выбираем recovery mode (вторая строка) и жмем ‘e’

4. В меню настройки запуска в строке начинающейся с “linux” меняем ro на rw и в конец строки дописываем init=/bin/bash. Затем жмем ‘ctrl’ + ‘x’



5. Попадаем в консоль.



Внутри нее вводим passwd и вводим новый пароль для root дважды.

Дальше можем добавить изначального пользователя в группу sudo (дать права администратора):

usermod -aG sudo vboxuser

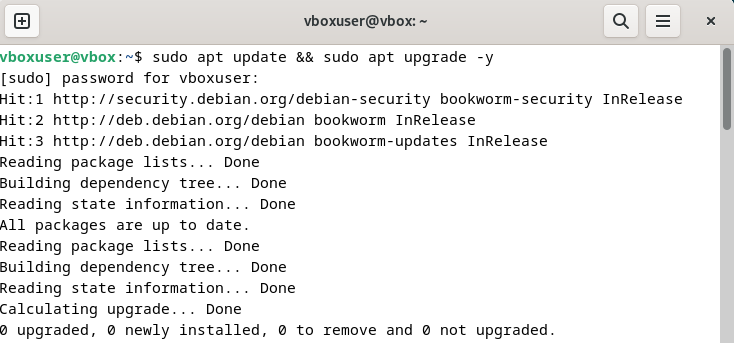
И перезапускаем систему:

exec /sbin/init

4) Обновление системы

Открываем терминал. Нажимаем на Activities и вводим в верхней строке поиска Terminal.

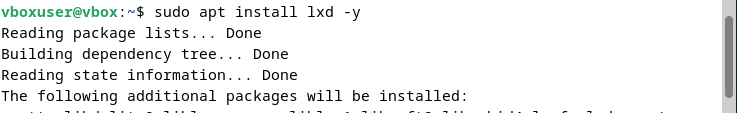
Проверяем обновление системы:  
sudo apt update && sudo apt upgrade –y

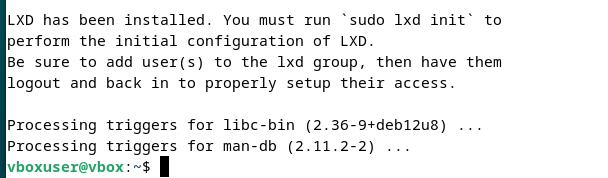


5) Устанавливаем LXD и настраиваем контейнеры

**Установка LXD**:

sudo apt install lxd –y

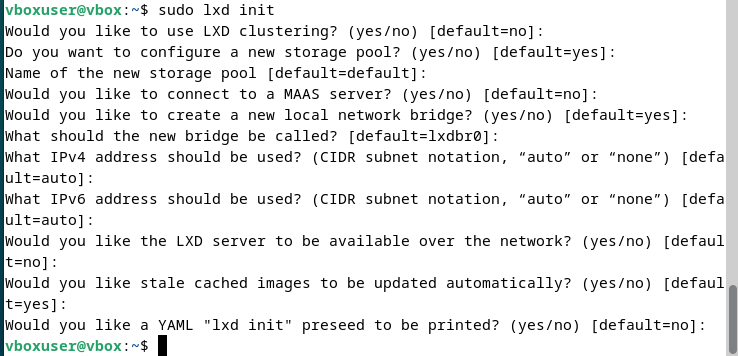




**Инициализация LXD**:

sudo lxd init

Везде выбираем ответы по умолчанию:



**Очистка iptables:**

sudo apt install iptables –y

sudo iptables --flush -t mangle



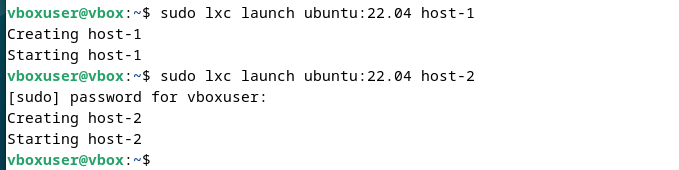
**Перезапуск службы LXD:**



**Создание контейнеров:**

sudo lxc launch images:ubuntu/22.04 host-1

sudo lxc launch images:ubuntu/22.04 host-2



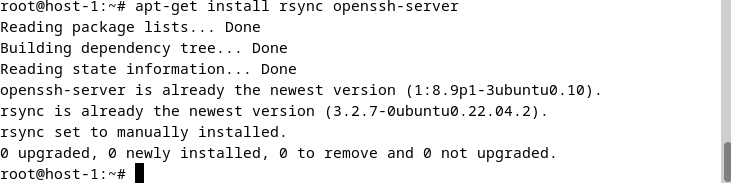
**Вход в контейнер:**

sudo lxc exec host-1 -- /bin/bash



**Устанавливаем openssh-server и rsync:**

apt-get install rsync openssh-server

****

**Настройка сервера, чтобы заходил только по ключу:**

echo "PasswordAuthentication no" >> /etc/ssh/sshd\_config

****

**Перезагружаем openssh:**

systemctl reload sshd

****

**Создаем пользователя для логирования:**

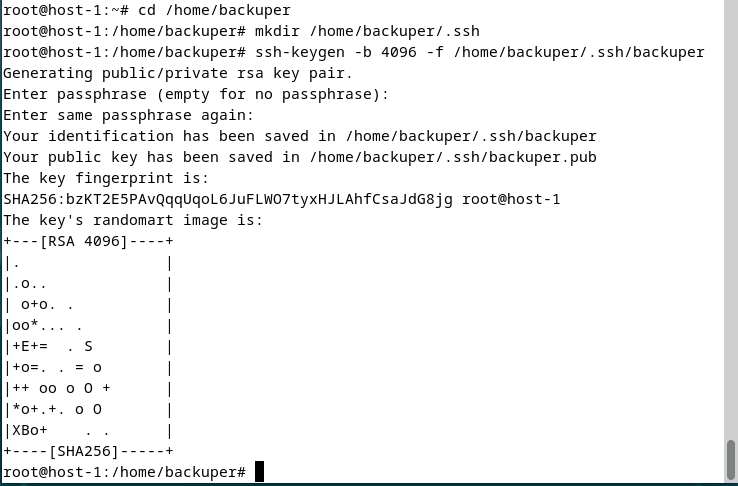
useradd backuper --create-home --home-dir /home/backuper

**Создаем ключи для доступа:**

cd /home/backuper

mkdir /home/backuper/.ssh

ssh-keygen -b 4096 -f /home/backuper/.ssh/backuper

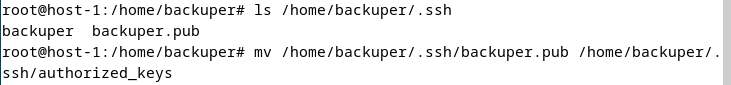
****

Пароль для разблокировки закрытого ключа **backup**.

**Перекидываем открытый ключ с расширением .pub на сервер:**

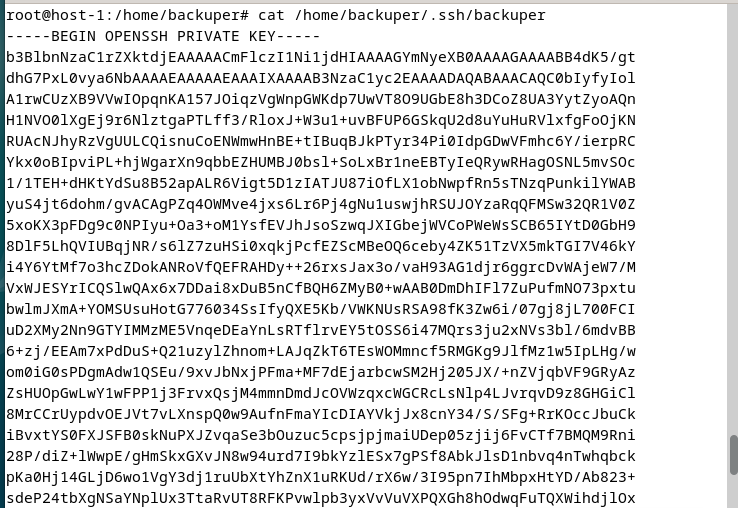
ls /home/backuper/.ssh

mv /home/backuper/.ssh/backuper.pub /home/backuper/.ssh/authorized\_keys



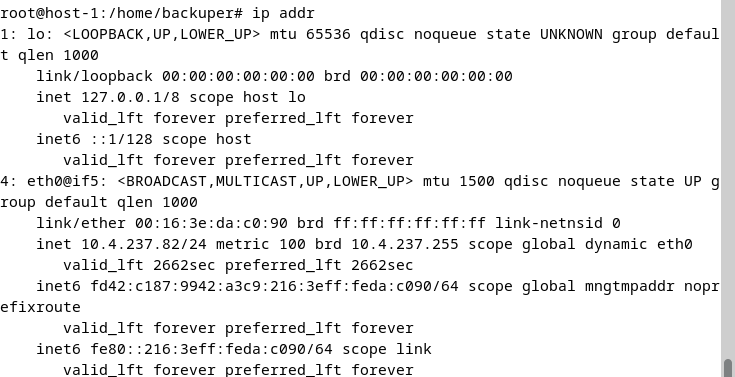
**Копируем закрытый ключ:**

cat /home/backuper/.ssh/backuper



**Проверим адрес машины:**

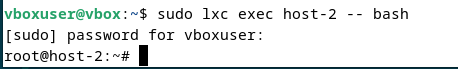
ip addr

****

**Настройка клиента:**

Выходим из консоли через ctr+d

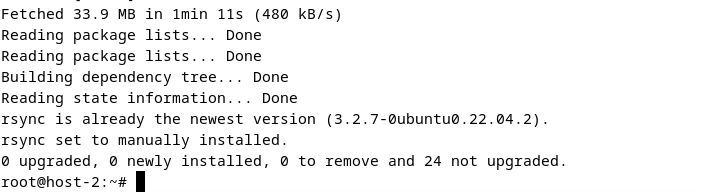
sudo lxc exec host-2 – bash



**Обновляем список пакетов и устанавливаем rsync:**

apt-get update && apt-get install rsync

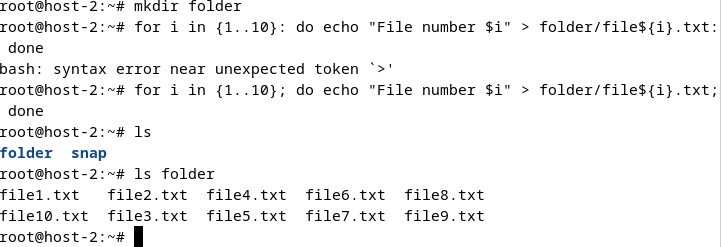




**Создаем папку и несколько файликов:**

mkdir folder

for i in {1..10}; do echo "File number $i" > folder/file${i}.txt; done



**Переносим ключ:**

mkdir ~/.ssh

cat > ~/.ssh/backuper

Через ctr+shift+d вставляем наш ключ и нажимаем ctr+d.



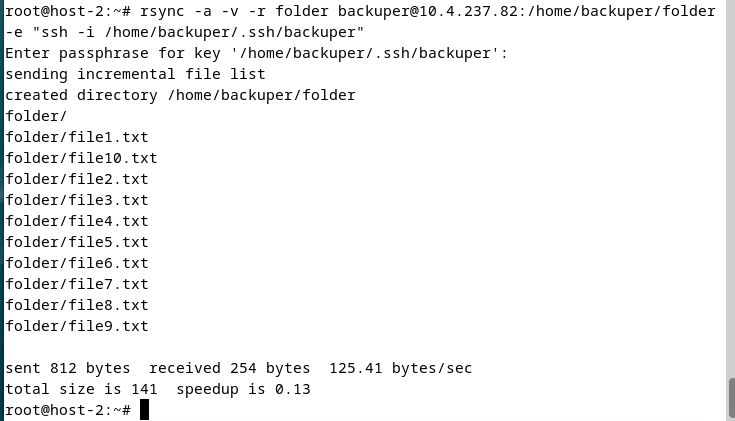


**Настройка доступа:**

chmod 600 ~/.ssh/backuper

**Копируем папку:**

rsync -a -v -r folder backuper@10.4.237.82:host-2 -e "ssh –i ~/.ssh/backuper"

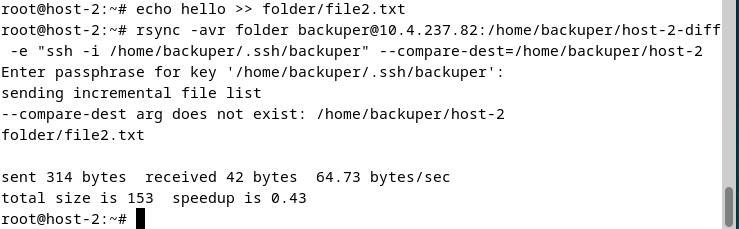


**Делаем частичное копирование:**

Изменили файл:

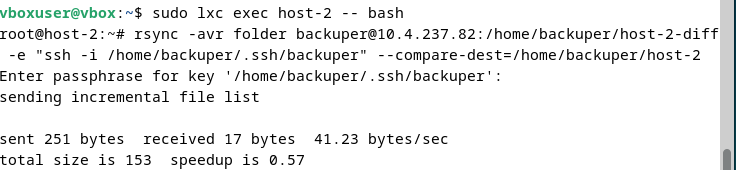
echo hello >> folder/file1.txt

Копирование:  
rsync -avr folder backuper@10.4.237.82:/home/backuper/host-2-diff -e "ssh -i /home/backuper/.ssh/backuper" --compare-dest=/home/backuper/host-2



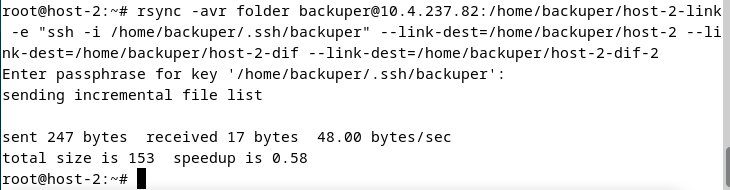
Просмотр разностной копии:

rsync -avr folder backuper@10.4.237.82:/home/backuper/host-2-diff -e "ssh -i /home/backuper/.ssh/backuper" --compare-dest=/home/backuper/host-2



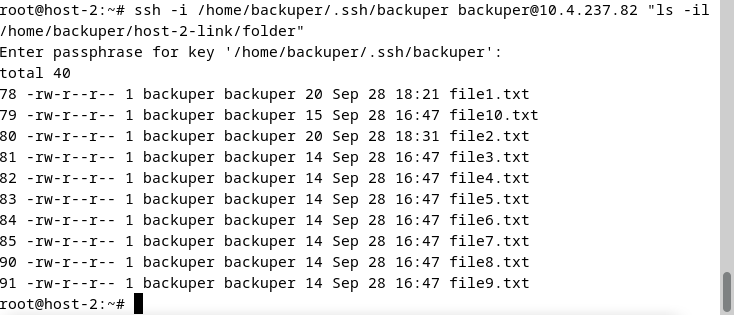
**Жесткие ссылки:**

rsync -avr folder backuper@10.4.237.82:/home/backuper/host-2-link -e "ssh -i /home/backuper/.ssh/backuper" --link-dest=/home/backuper/host-2 --link-dest=/home/backuper/host-2-dif --link-dest=/home/backuper/host-2-dif-2



Проверка содержимого папки:

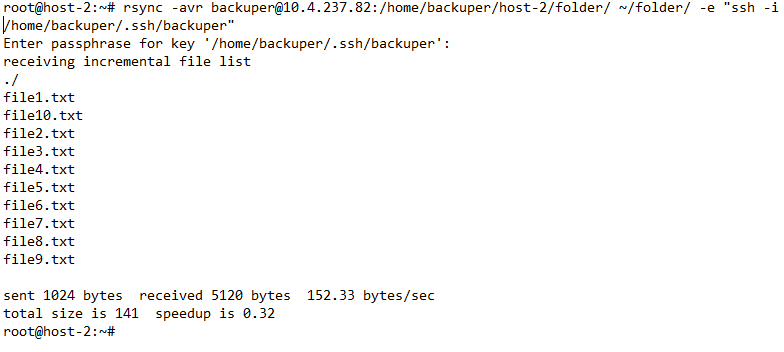
ssh -i /home/backuper/.ssh/backuper backuper@10.4.237.82 "ls -il /home/backuper/host-2-link/folder"



**Восстановление данных:**

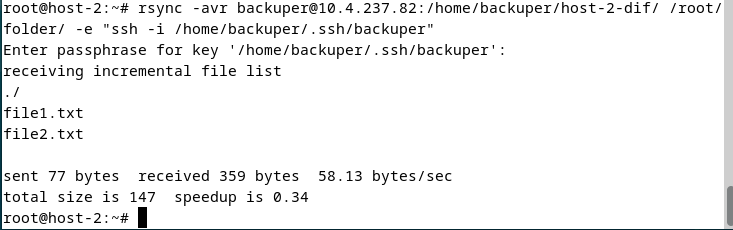
Восстанавливаем файлы из первой копии:

rsync -avr backuper@10.4.237.82:/home/backuper/host-2 / /root/folder/ -e "ssh -i /home/backuper/.ssh/backuper"

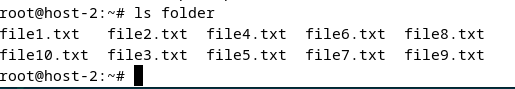


Восстанавливаем файлы из разностной копии:

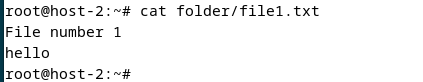
rsync -avr backuper@10.4.237.82:/home/backuper/host-2-dif/ /root/folder/ -e "ssh -i /home/backuper/.ssh/backuper"

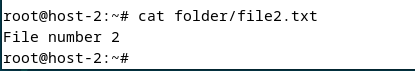


Просмотр папки:



Просмотр файла:





# Вопросы:

**1. Как создаются полная, разностная и инкрементная копии простым языком**

**Полная копия:**

Полная копия включает все файлы и папки из исходной директории, независимо от их изменений.

Процесс: Просто копируем все файлы и папки с оригинального расположения на новое. Она занимает столько же места, сколько и оригинал.

Пример: Если у вас есть папка docs с пятью файлами, то полная копия будет включать все эти пять файлов, даже если они не изменялись.

**Разностная копия:**

Разностная копия содержит только те файлы, которые изменились или были добавлены с момента создания последней полной копии.

Процесс: Сравниваются файлы с последней полной копией и копируются только измененные или новые файлы.

Пример: Если вы сделали полную копию вчера, а сегодня изменили два файла и добавили один новый, разностная копия будет включать только эти три файла.

**Инкрементная копия:**

Инкрементная копия содержит только те файлы, которые изменились или добавлены с момента создания последней копии — это может быть полная или разностная копия.

Процесс: Сравниваются файлы с последней любой копией (полной или разностной) и копируются только измененные или новые файлы.

Пример: Если у вас есть полная копия и три разностные копии, инкрементная копия будет включать только изменения, которые произошли после последней разностной копии.

**2. Как восстанавливаются полная, разностная и инкрементная копии простым языком**

**Восстановление полной копии:**

Вы берете последнюю полную копию и копируете её на нужное место. Это как если бы вы просто вернули все файлы в том виде, в каком они были на момент создания полной копии.

Процесс: Перемещаем или копируем все файлы из полной копии на исходное место.

**Восстановление разностной копии:**

Сначала восстанавливается последняя полная копия, а затем поверх неё накатывается разностная копия.

Процесс: Сначала копируем все файлы из полной копии, затем добавляем или заменяем файлы из разностной копии, чтобы они соответствовали последнему состоянию.

**Восстановление инкрементной копии:**

Сначала восстанавливается последняя полная копия, затем последовательно накладываются все разностные и инкрементные копии.

Процесс: Сначала восстанавливается полная копия, затем применяются разностные копии в порядке их создания, и в конце — инкрементные копии.

**3. Какие есть проблемы с удалёнными файлами при создании и восстановлении разностной и инкрементной копии? Как решается?**

**Проблемы:**

Если файл был удалён в исходной директории после создания полной копии, то в разностной или инкрементной копии он может остаться.

При восстановлении такой копии, файл, который был удалён в оригинале, появится снова.

Это может создать путаницу, так как восстанавливаются "лишние" файлы.

**Решение:**

Ведение специального журнала удалённых файлов: если файл был удалён, информация об этом сохраняется, и при восстановлении он не будет копироваться.

Использование опции --delete в rsync: если указать эту опцию при копировании, она удаляет на целевом месте файлы, которые были удалены в исходной директории.

**4. Оптимизация разностной/инкрементной копии с использованием жёстких ссылок (hard link). Что такое жёсткая и мягкая/символическая ссылка?**

**Жёсткие ссылки (hard link):**

Жёсткая ссылка — это указатель на один и тот же файл на диске. У файла есть уникальный номер (inode), и жёсткие ссылки позволяют нескольким файлам указывать на один и тот же inode.

Преимущество: файлы могут находиться в разных местах файловой системы, но указывать на одно и то же физическое место на диске. Это экономит место.

Пример: если у вас есть файл file1.txt и вы создадите на него жёсткую ссылку, то изменения в одном файле автоматически отразятся в другом, так как это один и тот же файл.

**Мягкие или символические ссылки (symlink):**

Мягкая ссылка — это файл, который указывает на другой файл по его имени и пути. Это как ярлык в Windows.

Преимущество: позволяет легко переходить к файлу или директории из другого места в файловой системе.

Недостаток: если файл, на который указывает символическая ссылка, удалён, то ссылка становится "битой" и ведёт в никуда.

Пример: если вы создадите символическую ссылку link.txt на file1.txt, а затем удалите file1.txt, то link.txt будет вести в никуда.

**Как выглядит файл-жёсткая ссылка?**

Файл с жёсткой ссылкой выглядит как обычный файл. Отличить его можно только по количеству ссылок (link count), которое показывает, сколько файлов ссылаются на один и тот же inode.

Команда ls -l покажет количество ссылок рядом с правами доступа (например, -rw-r--r-- 2 user group), где число "2" указывает на две жёсткие ссылки.

**Как выглядит файл-мягкая ссылка?**

Символическая ссылка отображается с помощью команды ls -l как стрелка (->), указывающая на оригинальный файл.

Пример: lrwxrwxrwx 1 user group 11 Sep 28 20:10 link.txt -> file1.txt, где link.txt — это мягкая ссылка, указывающая на file1.txt.