

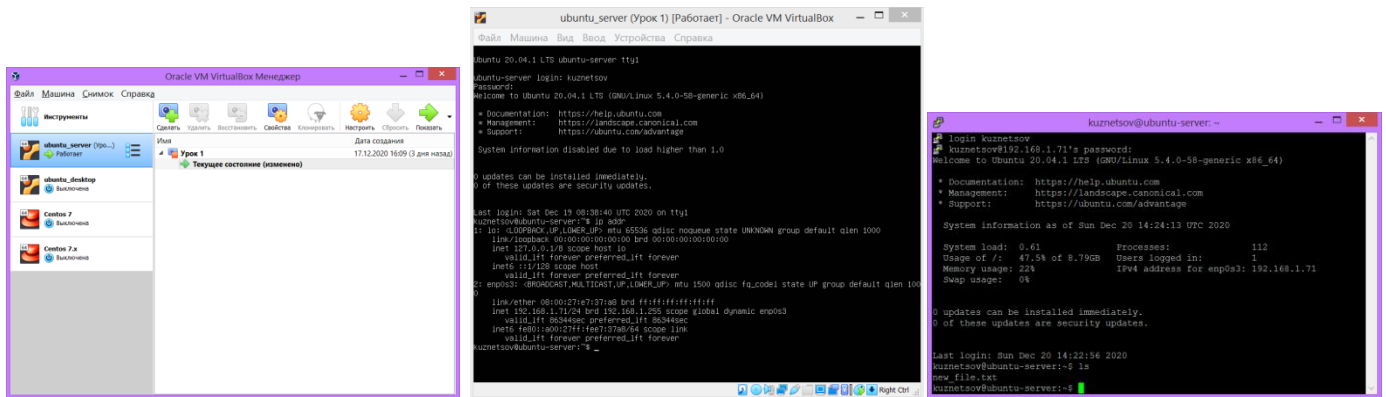
Домашнее задание:

1. **Навигация по файловой системе.** Попрактиковаться в перемещении между каталогами, используя полный и относительный путь. Перечислить, какие параметры команды `cd` позволяют быстро вернуться в домашний каталог, позволяют перейти на уровень выше.
2. **Управление файлами и каталогами и текстовые редакторы.** Создать файл с наполнением, используя несколько способов. Использовать разобранные текстовые редакторы для наполнения файлов данными. Создать копии созданных файлов, создать несколько каталогов с подкаталогами, перенести несколько файлов в созданные каталоги. Перечислить команды и используемые параметры команд.
3. * Используя дополнительный материал, настроить авторизацию по SSH с использованием ключей.

1. Навигация по файловой системе

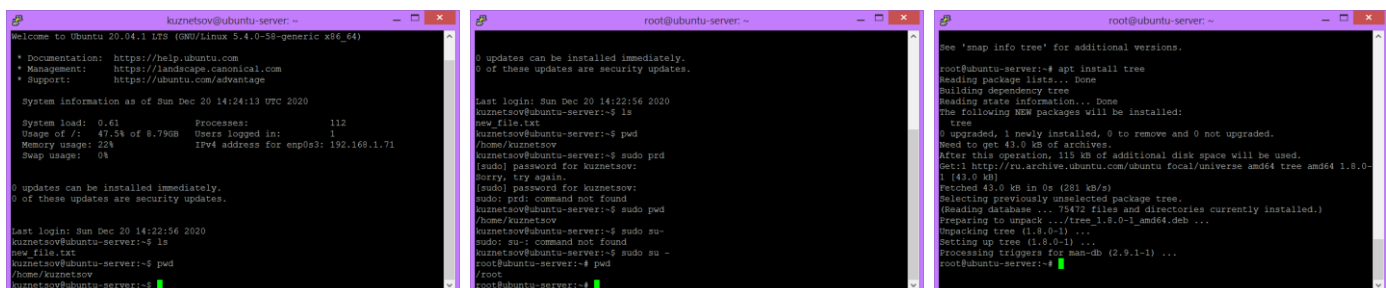
Попрактиковаться в перемещении между каталогами, используя полный и относительный путь.

Запускаем `ubuntu-server`, в терминальном режиме проходим регистрацию и выясняем `ip` (`ip addr`), подключаемся через `putty` (вводим `ip-адрес`), в консоли вводим `login` и `password`. Естественно, в первый раз работы с сервером (вру, с `ubuntu-desktop`) пришлось убедиться что `SSHd` не установлен (`systemctl status sshd`)...установил по команде (`sudo apt install openssh-server -y`)

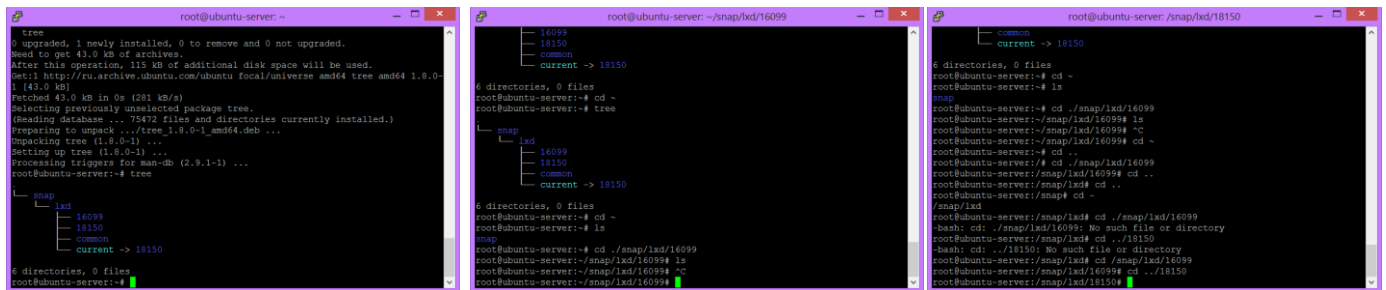


Начнем с простого, `pwd` (print working directory) – показывает домашний каталог пользователя (`/home/kuznetsov`).

Переключимся на `root` (`sudo su -`) и снова запросим домашний каталог (`pwd`) – результат иной (`/root`). Запустил команду (`tree`) для получения дерева каталогов. Пишет, что не установлено, пришлось запустить (`apt install tree`), перед этим (`apt update`) для обновления инфы о пакетах.



Попробуем навигацию по полному пути (`cd /snap/lxd/16099`). Полный путь всегда начинается с корневого каталога. Относительный путь указывается в отношении текущего расположения. К примеру, если мы находимся в каталоге «`/snap/lxd/16099`», для перехода в каталог «`./snap/lxd/18150`» достаточно ввести (`cd ../18150`)



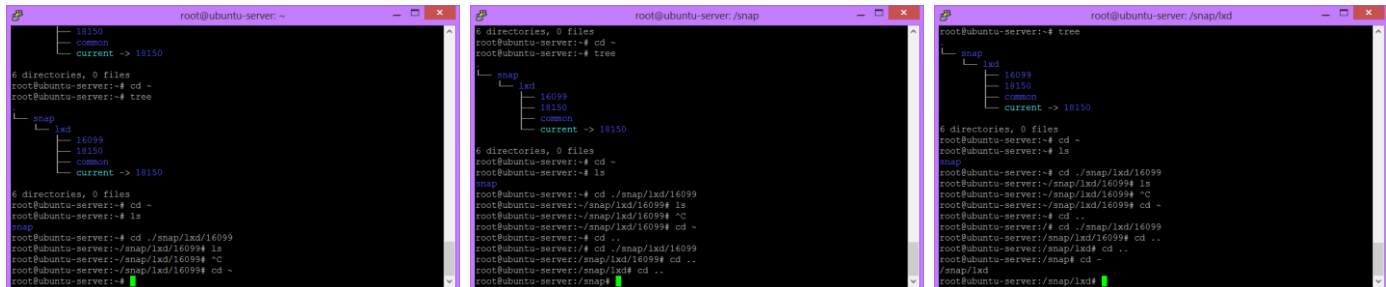
The first screenshot shows the installation of snapd and the creation of a directory tree with snap, lxd, 16099, 18150, and common directories. The second screenshot shows navigating from the root to /snap/lxd/16099 and then to /snap/lxd/18150 using relative paths. The third screenshot shows navigating from /snap/lxd/18150 back to /snap/lxd/16099 and then to the root directory using relative paths.

Перечислить, какие параметры команды `cd` позволят быстро вернуться в домашний каталог, позволят перейти на уровень выше.

Команда (`cd ~`) позволяет быстро вернуться в домашний каталог.

Для перехода на уровень выше, достаточно ввести (`cd ..`)

Для быстрого перехода в предыдущий каталог, достаточно ввести команду (`cd -`)

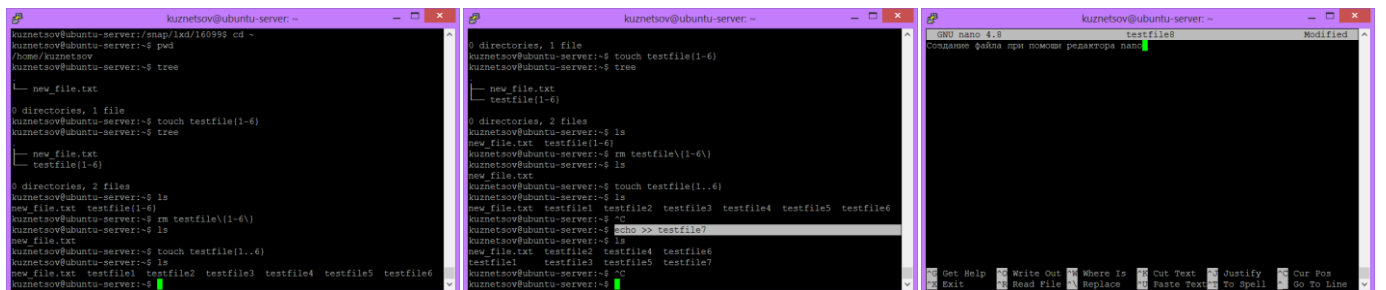


The first screenshot shows navigating to the root directory with 'cd ~'. The second screenshot shows navigating up one level with 'cd ..'. The third screenshot shows navigating back to the previous directory with 'cd -'.

2. Управление файлами и каталогами и текстовые редакторы. *Создать файл с наполнением, используя несколько способов.*

Для создания файлов в ОС Linux есть несколько способов:

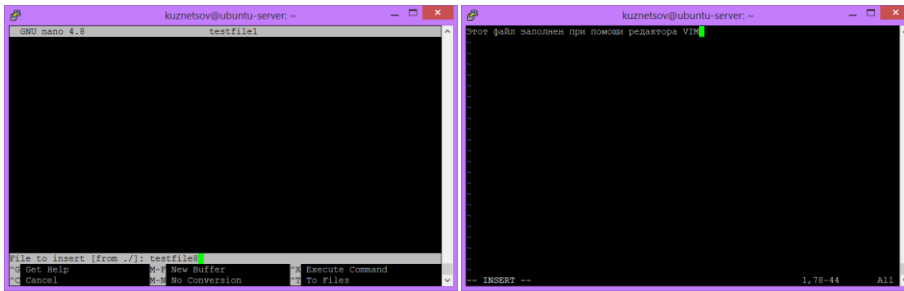
- Используя утилиту `touch` — она создаст пустой файл. Создаем несколько файлов (`touch testfile{1..6}`)
- Используя перенаправление потока вывода, например, с помощью утилиты `echo` (`echo >> testfile7`) (`echo 'Заполним файл при помощи echo...' > testfile8`).
- Используя текстовый редактор (на примере `nano`).



The first screenshot shows creating files with 'touch' and listing them. The second screenshot shows creating a file with 'echo' and listing it. The third screenshot shows creating a file with 'nano' and listing it.

Использовать разобранные текстовые редакторы для наполнения файлов данными.

Рассмотрим на примере редактора `nano` (`nano testfile1`). Вставим содержимое (`Ctrl_R`) из ранее заполненного файла «`testfile8`». Далее используем редактор `VIM` (`vim testfile2`). Для входа в режим редактирования по «`Insert`» нажмем «`i`». После окончания работы с текстом нажимаем клавишу «`Esc`» (выходим из режима редактирования), далее для выхода с сохранением «`:wq`».

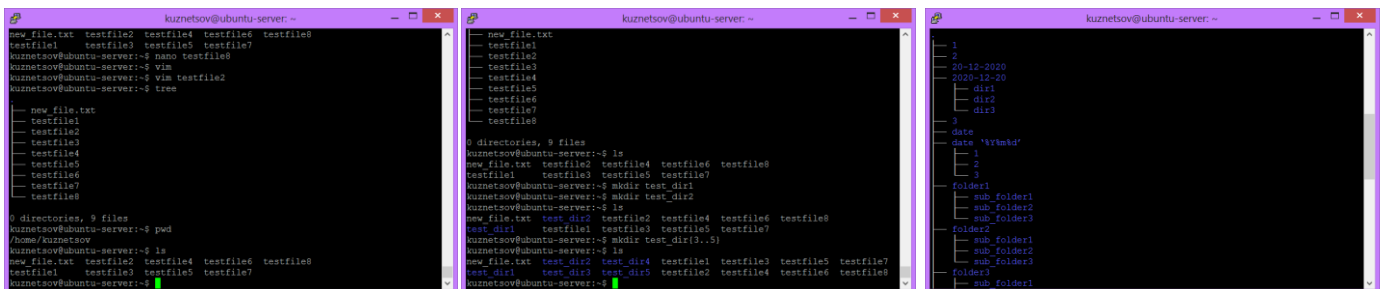


Создать копии созданных файлов, создать несколько каталогов с подкаталогами, перенести несколько файлов в созданные каталоги. Перечислить команды и используемые параметры команд.

Имеем домашний каталог «/home/kuznetsov». В каталоге несколько файлов.

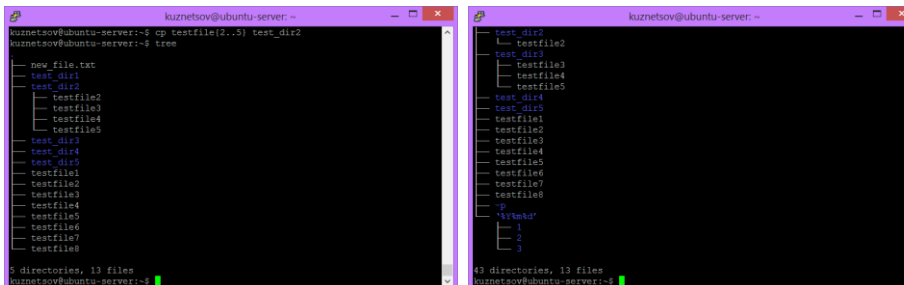
Создадим подкаталоги «mkdir test_dir1», «mkdir test_dir2», «mkdir test_dir3.5».

Поэкспериментируем с каталогами, включающими дату (mkdir -p "\$(date +"%Y-%m-%d")"/dir{1..3}).



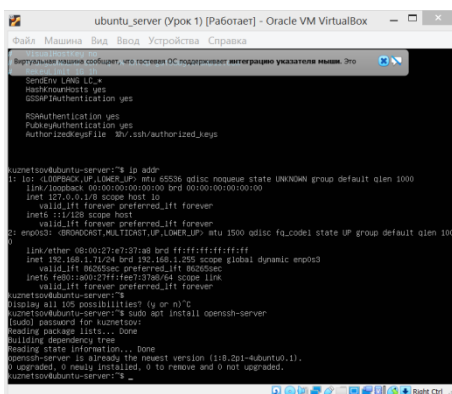
Скопируем несколько файлов по маске в каталог «test_dir2» командой (cp testfile{2..5} test_dir2)

Часть файлов из каталога «test_dir2» перенесем в каталог «test_dir3» командой (mv test_dir2/testfile{2..5} test_dir3)



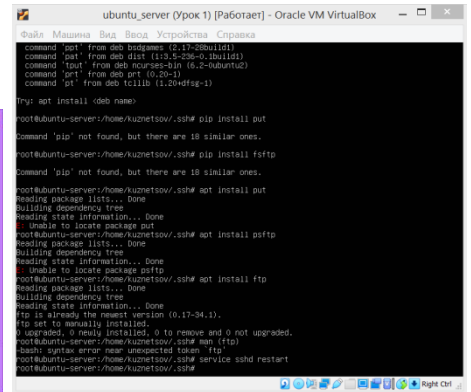
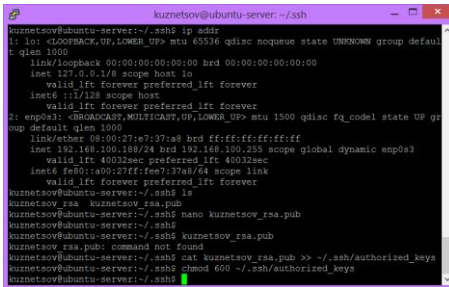
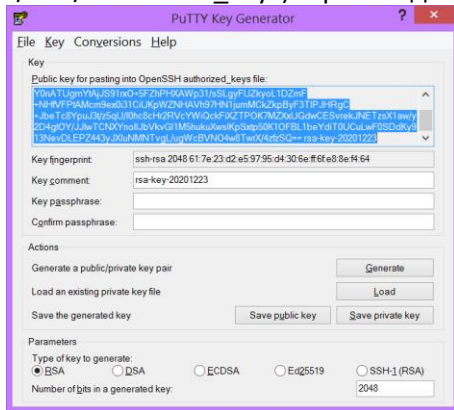
3. Используя дополнительный материал, настроить авторизацию по SSH с использованием ключей.

Обычно, требуется установить SSH сервер. В консоли вводим команду «sudo apt install openssh-server». В нашем случае, уже установлена последняя версия.



Скачиваем «Putty Key Generator», запускаем генерацию ключа и сохраняем его на локальном диске (D:\Python\Курсы\Linux\Key).

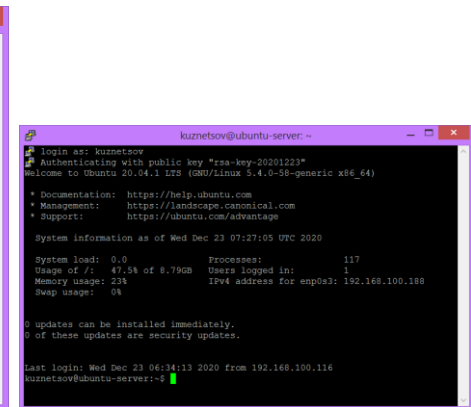
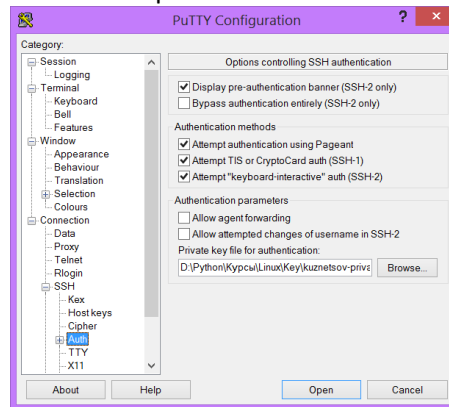
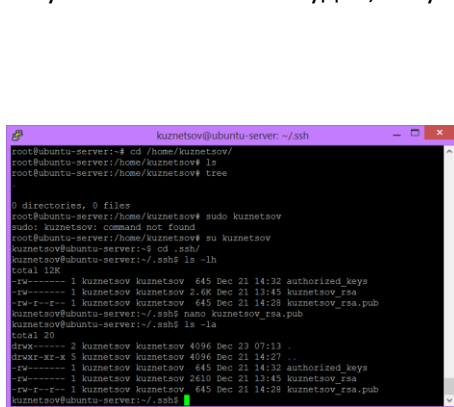
Здесь же, в окне «Putty Key Generator» копируем в буфер обмена публичный ключ. Если его копировать в редакторе, есть вероятность появления символов переноса строки, который придется удалять вручную. После этого, в терминале putty сохраняем содержимое публичного ключа при помощи редактора nano в файле «kuznetsov_rsa.pub». Далее, командой (`cat kuznetsov_rsa.pub >> ~/.ssh/authorized_keys`) установим сгенерированный ключ на сервер. Сразу установим права на хранилище ключей (`chmod 600 ~/.ssh/authorized_keys`). Произведем рестарт сервиса.



Убедимся, что права на каталог .ssh и файл ключа выставлены верные права.

Введем параметры нахождения секретного ключа в putty.

Запускаем сессию и «о чудо», запускается без пароля!



Проблемы, которые встретил (кратко):

- 1) Генерил ключ по разному и на сервере в консоли (`ssh-keygen -t rsa`) и в терминале putty и через PuttyKeyGen. Чуда не случилось. При запуске сессии ругался на секретный ключ «Unable to use key file...» (как позже выяснил, генерил не тот тип ключа, вместо RSA использовал SSH-1(RSA)).
- 2) Абсолютно бессмысленно производил настройку ssh сервера для авторизации по ключу. Открывал под root на редактирование файл /etc/ssh/sshd_config и внесил строки:
 - RSAAuthentication yes
 - PubkeyAuthentication yes
 - AuthorizedKeysFile %h/.ssh/authorized_keys
- 3) Не обращал внимание на права каталога ssh и файла ключей (права 600 и 700 соответственно)

В общем, как и говорят про Linux – «танцы с бубном» завершились победой...в этот раз моей.

