

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №1
«Организация удалённого доступа в ОС Linux.
»

Практическая работа
по дисциплине «Современные технологии программирования»
студента 1 курса группы ПИ-б-о-231(2)
Аметов Кемран Ленверович
направления подготовки 09.03.04 «Программная инженерия»


Симферополь, 2024

Цель:

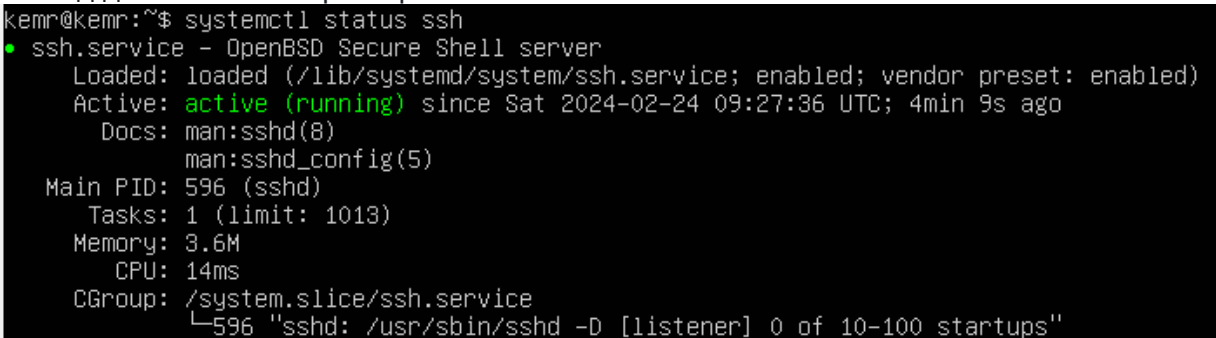
Ознакомиться на практике со средствами удаленного управления в операционной системе Linux. Приобрести опыт и навыки управления удаленным доступом посредством ssh.

Ход выполнения задания.

Каждую выполненную вам команду и результат её работы поместите в отчёт в виде скриншота.

1. Буду использовать существующую виртуальную машину Ubuntu Desktop, с пользователем kemrag.
2. Буду использовать существующую виртуальную машину Ubuntu Server, с пользователем kemr.
3. Обновлю индексы пакетов и обновлю устаревшие:

4. Затем убедитесь, что ssh-сервер установлен и запущен: `systemctl status ssh`.

Вывод должен быть примерно такой:

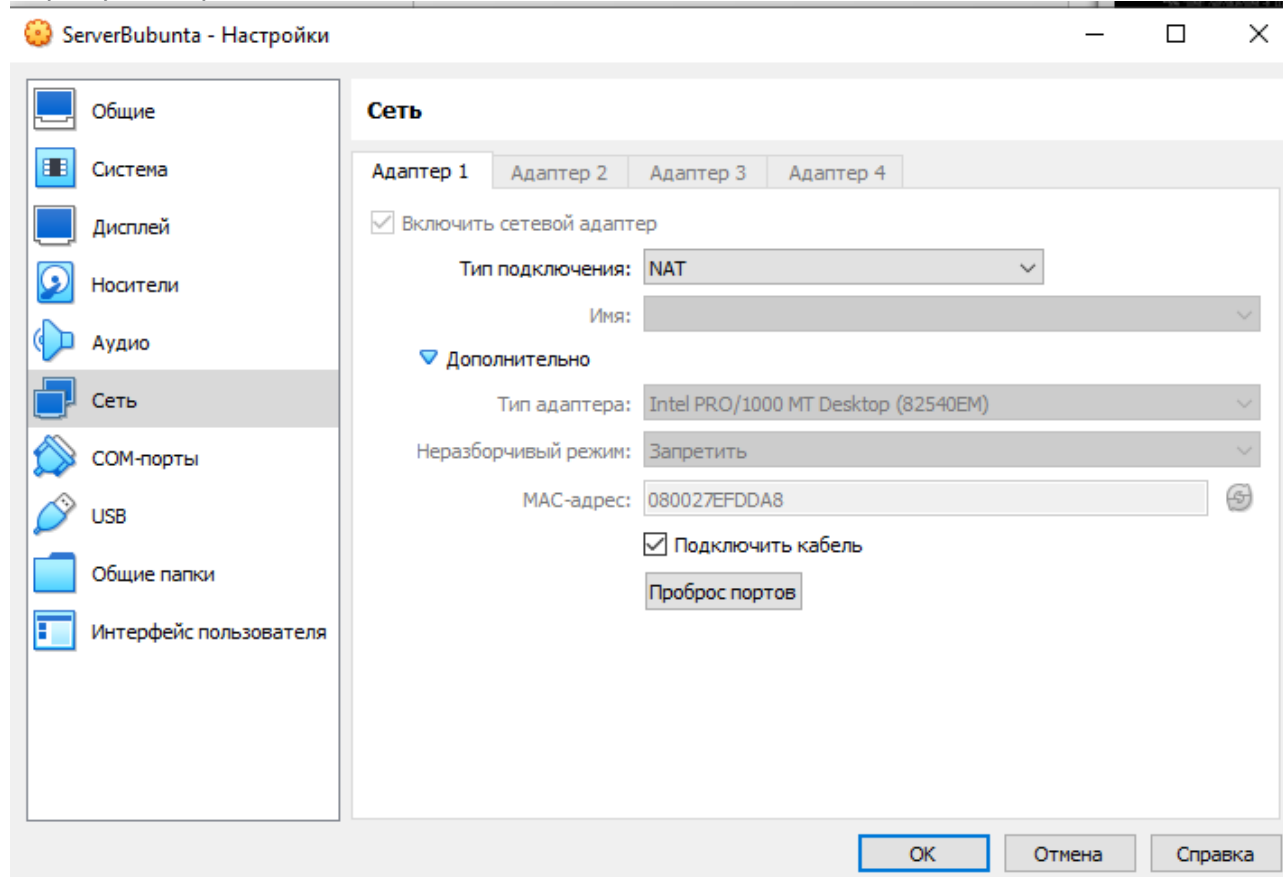


```
kemr@kemr:~$ systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2024-02-24 09:27:36 UTC; 4min 9s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 596 (sshd)
      Tasks: 1 (limit: 1013)
     Memory: 3.6M
        CPU: 14ms
    CGroup: /system.slice/ssh.service
            └─596 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

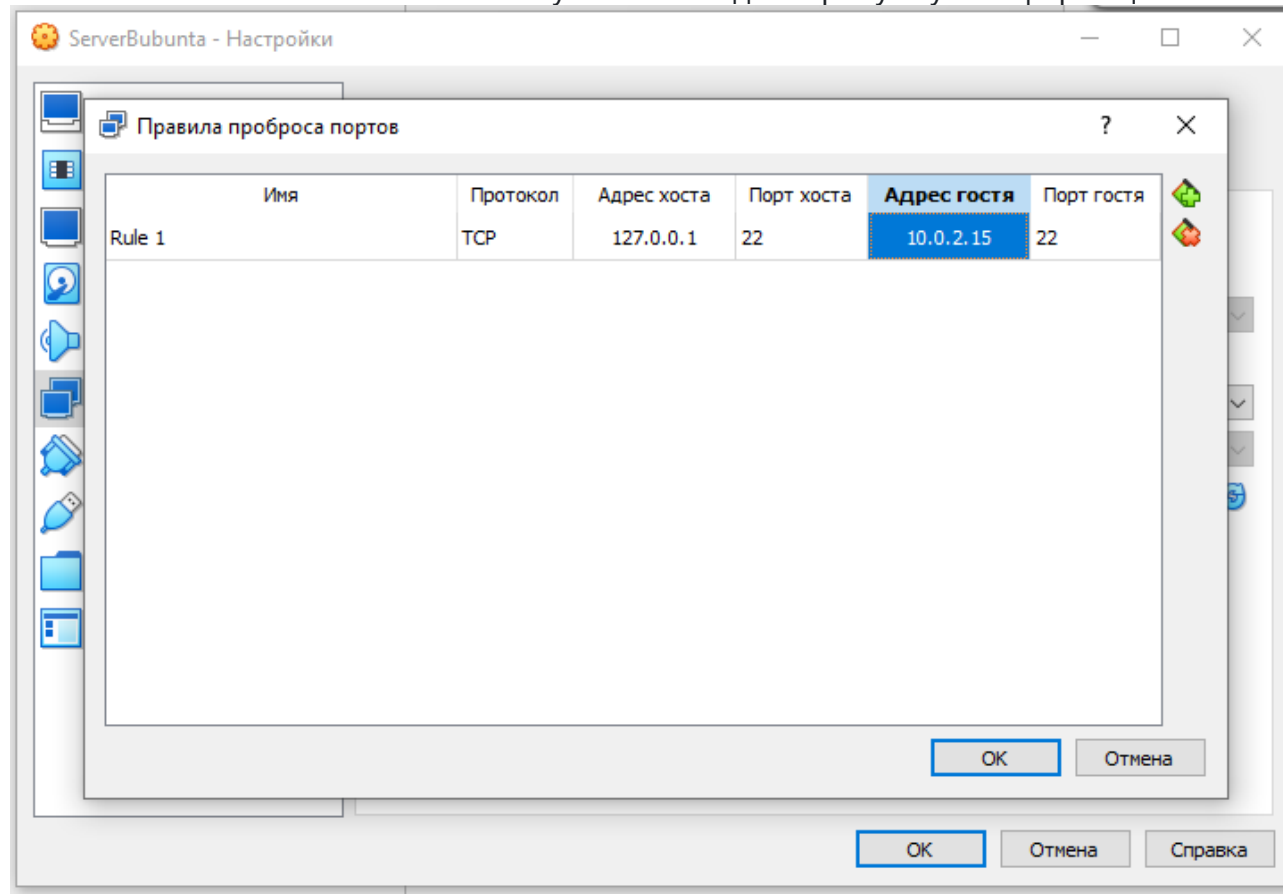
Доступ за NAT через проброс портов

8. В VirtualBox открою настройки машины с Ubuntu Server и на вкладке "Сеть" в настройках первого адаптера, в разделе "Дополнительно" нажмю кнопку

"Проброс портов":

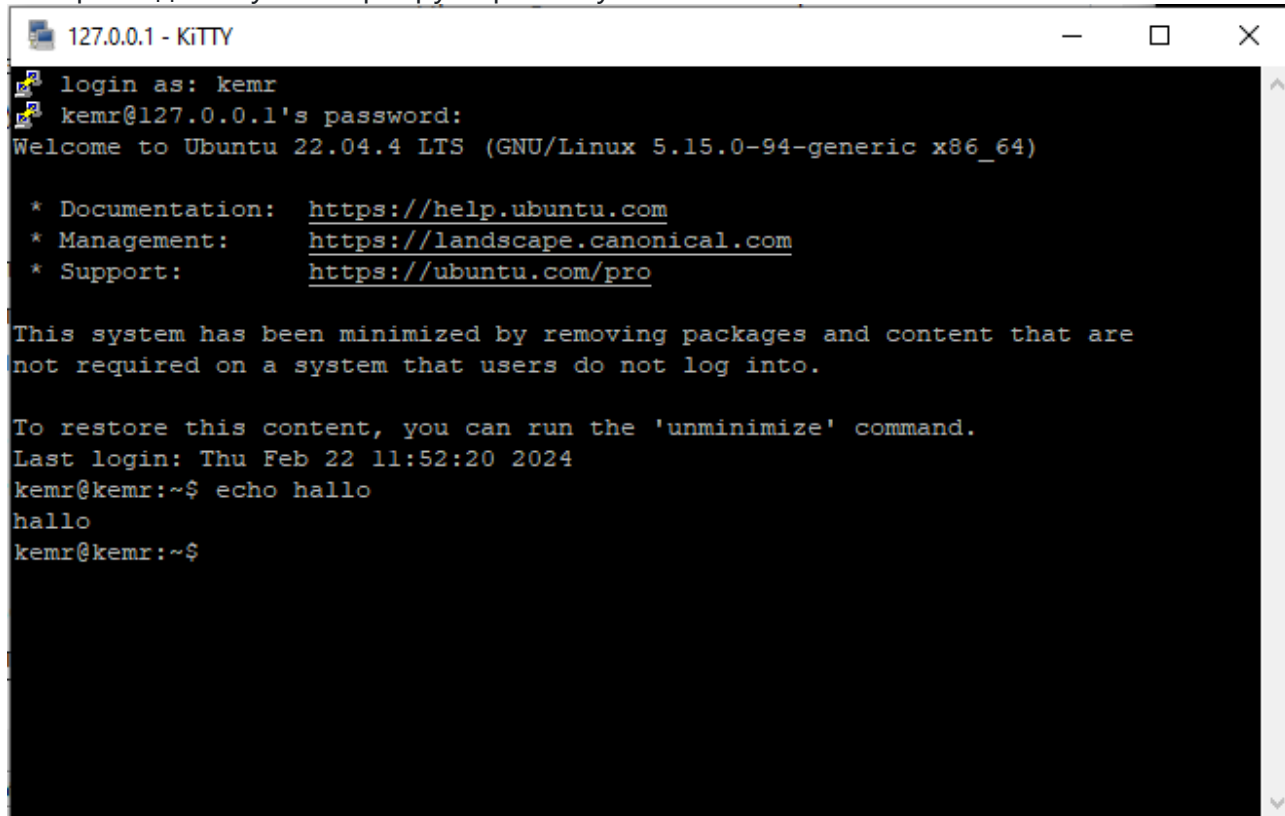


9. В появившемся окне нажмите иконку с "+" и введите требуемую информацию:



Проверка подключения по ssh

10. Теперь подключусь к серверу через kitty



```
127.0.0.1 - KITTY
login as: kemr
kemr@127.0.0.1's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

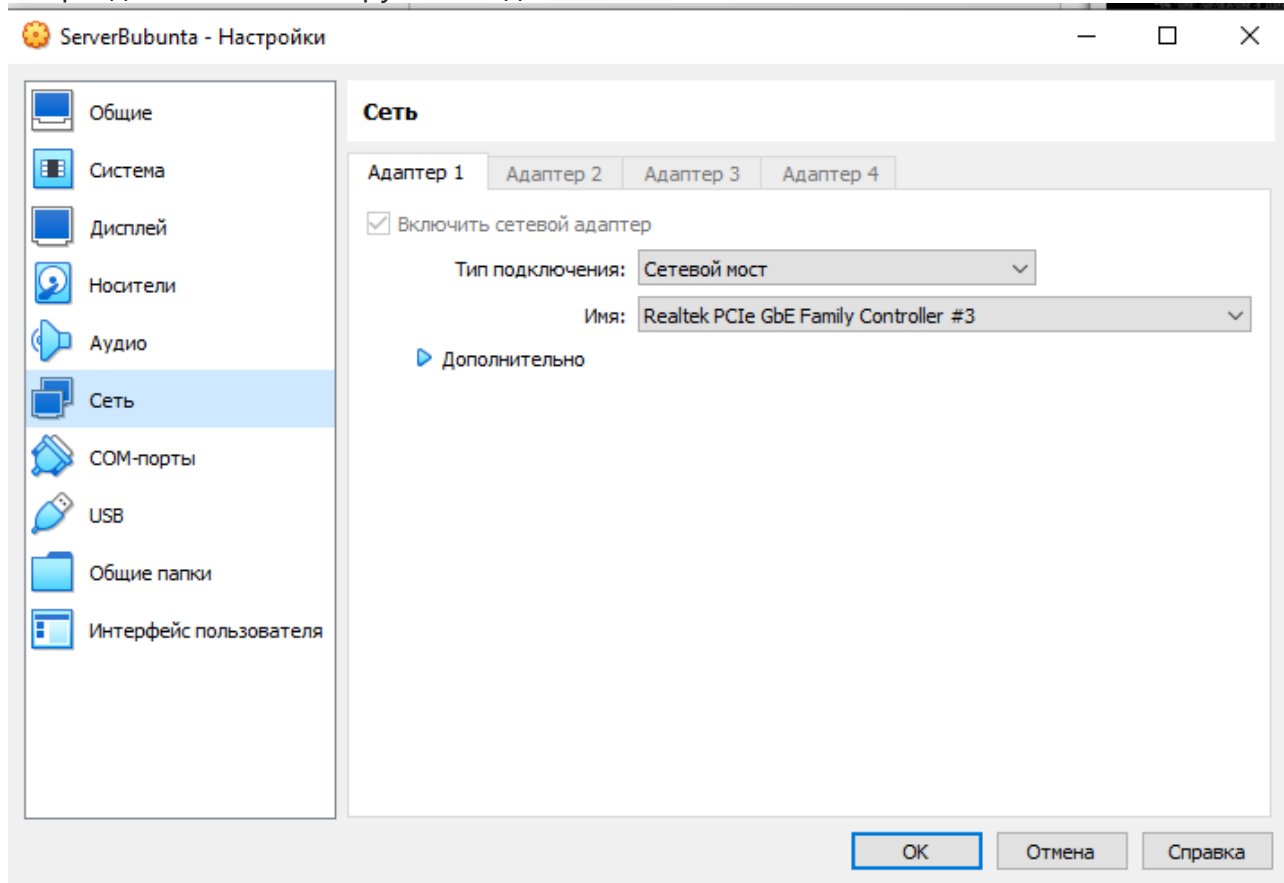
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Feb 22 11:52:20 2024
kemr@kemr:~$ echo hallo
hallo
kemr@kemr:~$
```

Доступ к серверу в сети без NAT

14. Изменяю тип сетевого подключения виртуальной машины на **Сетевой мост**.
Чтобы это сделать в окне VirtualBox открою "Настройки" виртуальной машины

и в разделе "Сеть" выберу Тип подключения "Сетевой мост":



15. Посмотрите список сетевых интерфейсов: `ip a` чтобы узнать новый ip-адрес Ubuntu Server-a. В моём случае роутер выдал серверу такой ip:

```
kemr@kemr:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ef:dd:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 metric 100 brd 192.168.0.255 scope global dynamic enp0s3
        valid_lft 6686sec preferred_lft 6686sec
    inet6 fe80::a00:27ff:feef:dda8/64 scope link
        valid_lft forever preferred_lft forever
```

16. Проверю что сервер доступен из хостовой системы

```
C:\Users\USER>ping 192.168.0.102

Обмен пакетами с 192.168.0.102 по 32 байтами данных:
Ответ от 192.168.0.102: число байт=32 время<1мс TTL=64
Ответ от 192.168.0.102: число байт=32 время<1мс TTL=64
Ответ от 192.168.0.102: число байт=32 время<1мс TTL=64
Ответ от 192.168.0.102: число байт=32 время<1мс TTL=64

Статистика Ping для 192.168.0.102:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)

Приблизительное время приема-передачи в мс:
    Минимальное = 0мсек, Максимальное = 0 мсек, Среднее = 0 мсек

C:\Users\USER>
```

17. Подключусь к серверу через ssh

```
C:\Users\USER>ssh kemr@192.168.0.101
kemr@192.168.0.101's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Last login: Sat Feb 24 09:47:06 2024
```

18. Выполню команду ls.

```
Выполню команду ls.
kemr@kemr:~$ ls
kemr@kemr:~$
```

Обмен файлами с удалённым сервером

Простое копирование (scp)

19. На сервере в домашнем каталоге создам папку "info".

```
kemr@kemr:~$ mkdir info
kemr@kemr:~$ ls
info
```

20. На хостовой машине создам файл "get_info.py" со следующим содержимым:

```
import platform
print(platform.uname())
```

Я создам его в C:/Users/USER.

```
get_info.py X
C: > Users > USER > get_info.py
1 import platform
2 print(platform.uname())
3
```

21. Воспользуюсь утилитой scp чтобы скопировать файл "get_info.py" с хостовой машины на сервер в каталог "info".

```
C:\Users\USER>scp get_info.py kemr@192.168.0.101:/home/kemr/info/
kemr@192.168.0.101's password:
get_info.py 100% 42 42.2KB/s 00:00
C:\Users\USER>
```

22. Проверю, что файл "get_info.py" появился на сервере в каталоге "info" и его содержимое соответствует исходному.

```
kemr@kemr:~$ ls
info
kemr@kemr:~$ cd info
kemr@kemr:~/info$ ls
get_info.py
kemr@kemr:~/info$ _
```

23. Перейду в каталог "info" и запущу скрипт, при этом результат его работы перенаправьте в файл "log.txt":

```
kemr@kemr:~$ cd info
kemr@kemr:~/info$ python3 get_info.py > log.txt
```

24. Теперь выполню копирование файлов с сервера на хостовую машину. Для этого снова перейду в терминал на хостовой машине и выполню команду:

```
C:\Users\USER>scp -r kemr@192.168.0.101:/home/kemr/info .
kemr@192.168.0.101's password:
log.txt                                100% 144 144.5KB/s 00:00
get_info.py                           100% 42 0.0KB/s 00:00
```

25. Открою файл "log.txt" на хостовой системе.

log.txt – Блокнот

Файл Правка Формат Вид Справка

```
uname_result(system='Linux', node='kemr', release='5.15.0-94-generic', version='#104-Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024', machine='x86_64')
```

Обмен файлами с сервером в интерактивном режиме (sftp)

28. На хостовой машине модифицирую файл "get_info.py" со следующим содержимым:

```
import platform
uname = platform.uname()
print(f"System: {uname.system}")
print(f"Node Name: {uname.node}")
print(f"Release: {uname.release}")
print(f"Version: {uname.version}")
print(f"Machine: {uname.machine}")
print(f"Processor: {uname.processor}")
```

```
Users > USER > get_info.py > ...
import platform
uname = platform.uname()
print(f"System: {uname.system}")
print(f"Node Name: {uname.node}")
print(f"Release: {uname.release}")
print(f"Version: {uname.version}")
print(f"Machine: {uname.machine}")
print(f"Processor: {uname.processor}")
```

29. Подключусь к серверу при помощи утилиты sftp.

```
C:\Users\USER>sftp kemr@192.168.0.101
kemr@192.168.0.101's password:
Connected to 192.168.0.101.
```

30. Введу команду help и изучу справку по доступным в sftp командам.

```

sftp> help
Available commands:
bye                               Quit sftp
cd path                           Change remote directory to 'path'
chgrp [-h] grp path              Change group of file 'path' to 'grp'
chmod [-h] mode path             Change permissions of file 'path' to 'mode'
chown [-h] own path              Change owner of file 'path' to 'own'
df [-hi] [path]                  Display statistics for current directory or
                                  filesystem containing 'path'
exit                              Quit sftp
get [-afpR] remote [local]       Download file
help                             Display this help text
lcd path                          Change local directory to 'path'
lls [ls-options] [path]]         Display local directory listing
lmkdir path                      Create local directory
ln [-s] oldpath newpath          Link remote file (-s for symlink)
lpwd                             Print local working directory
ls [-lafhlNrSt] [path]           Display remote directory listing
lumask umask                     Set local umask to 'umask'
mkdir path                       Create remote directory
progress                         Toggle display of progress meter
put [-afpR] local [remote]       Upload file
pwd                              Display remote working directory
quit                             Quit sftp
reget [-fpR] remote [local]      Resume download file
rename oldpath newpath           Rename remote file
reput [-fpR] local [remote]      Resume upload file
rm path                          Delete remote file
rmdir path                       Remove remote directory
symlink oldpath newpath          Symlink remote file
version                          Show SFTP version
!command                         Execute 'command' in local shell
!                                Escape to local shell
?                                Synonym for help
sftp>

```

31. Использую команду put, чтобы скопировать локальный файл "get_info.py" на сервер.

```

sftp> put get_info.py
Uploading get_info.py to /home/kemr/get_info.py
get_info.py                               100% 260    0.3KB/s   00:00
sftp>

```

32. Запущу его через обычный ssh-терминал также как и ранее.

```

kemr@kemr:~$ python3 get_info.py >> log.txt

```

33. Использую команду get в терминале с запущенным sftp, чтобы забрать с сервера файл "log.txt" с выводом программы. И завершу сеанс командой exit.

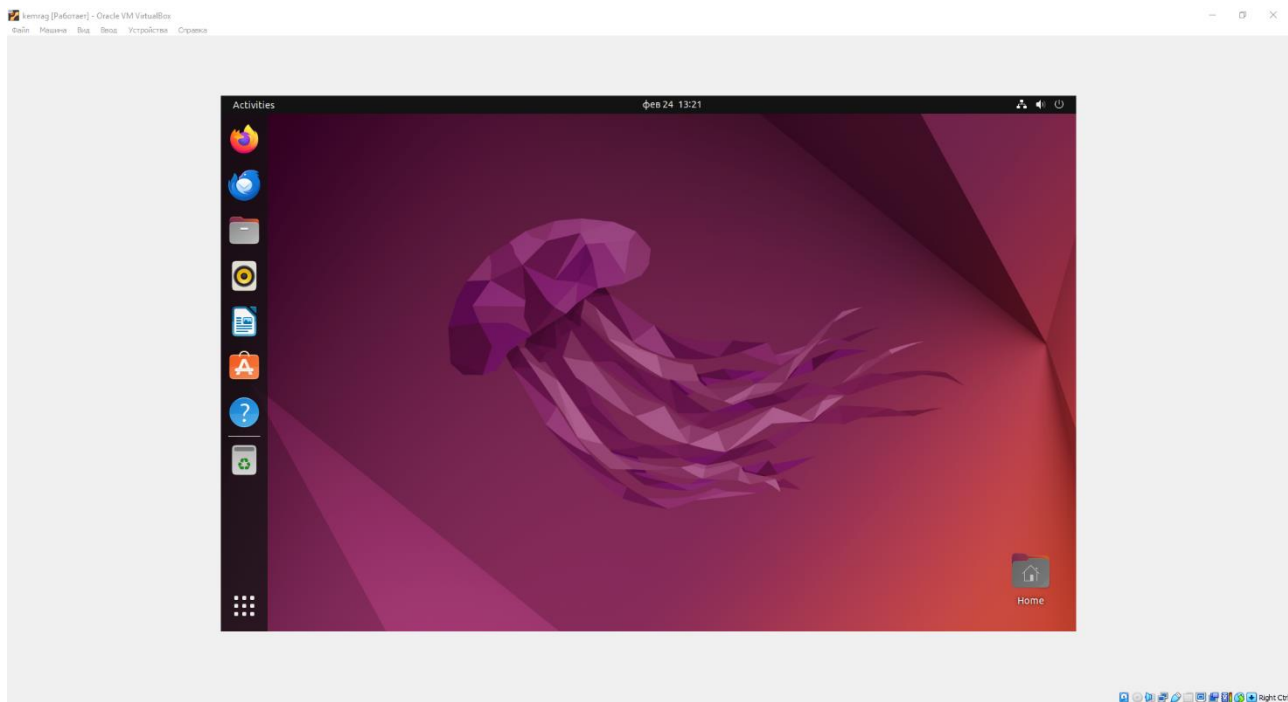
```

sftp> get log.txt
Fetching /home/kemr/log.txt to log.txt
/home/kemr/log.txt                        100% 144   144.2KB/s   00:00
sftp> exit

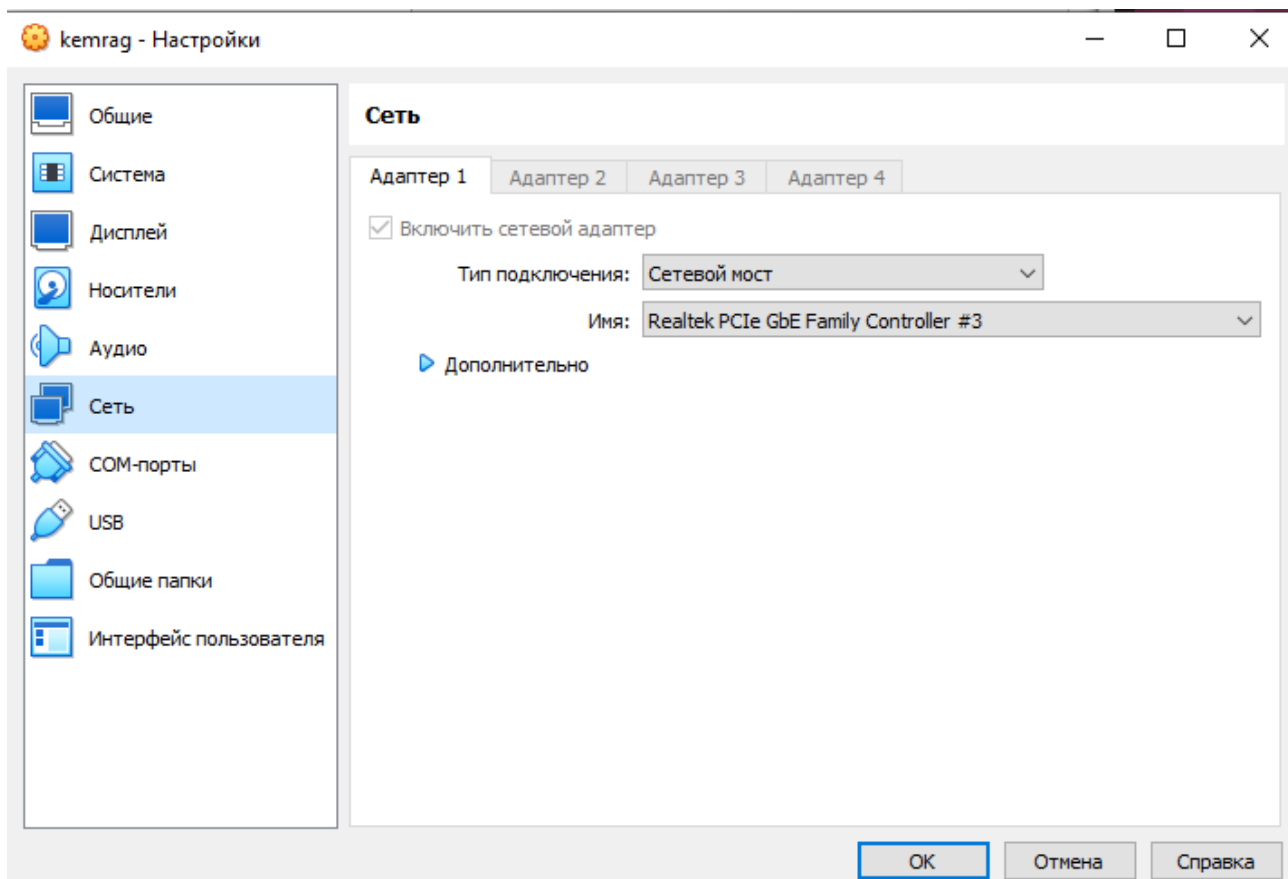
```

Обмен файлами в графическом режиме (sftp)

35. Запущу виртуальную машину с Ubuntu Desktop;

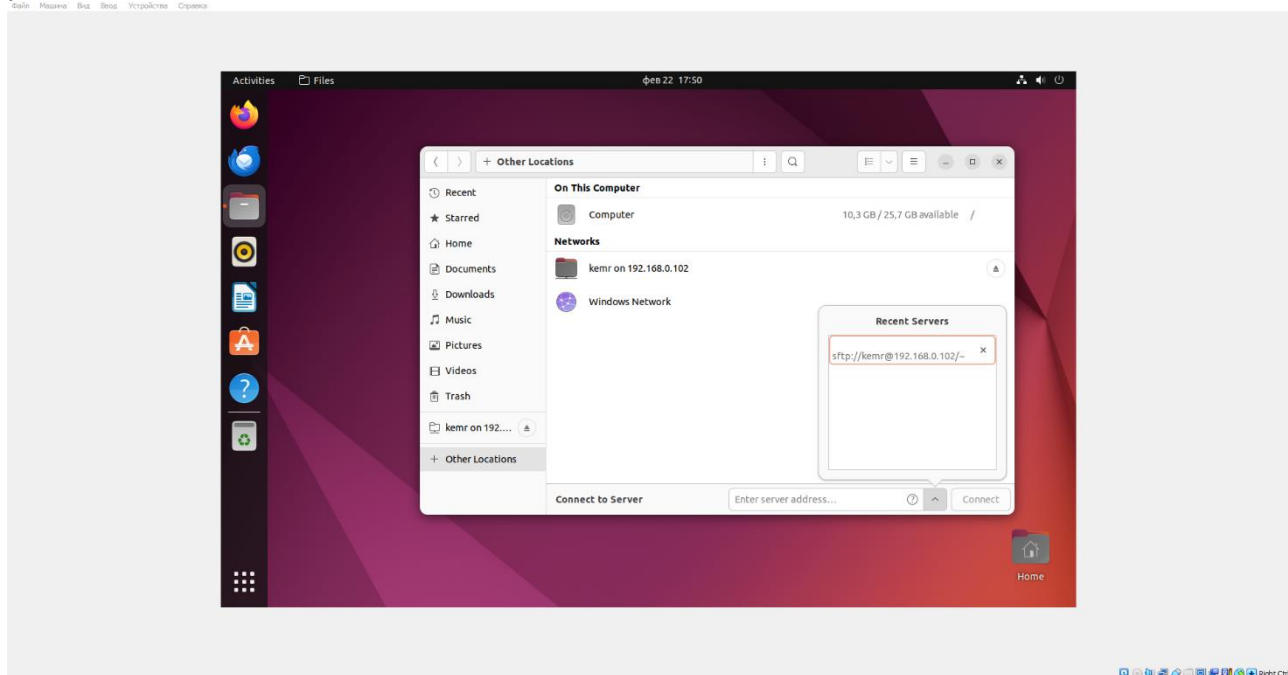


36. В настройках VirtualBox, для этой машины, устанавливаю "Тип подключения" - "Сетевой мост";

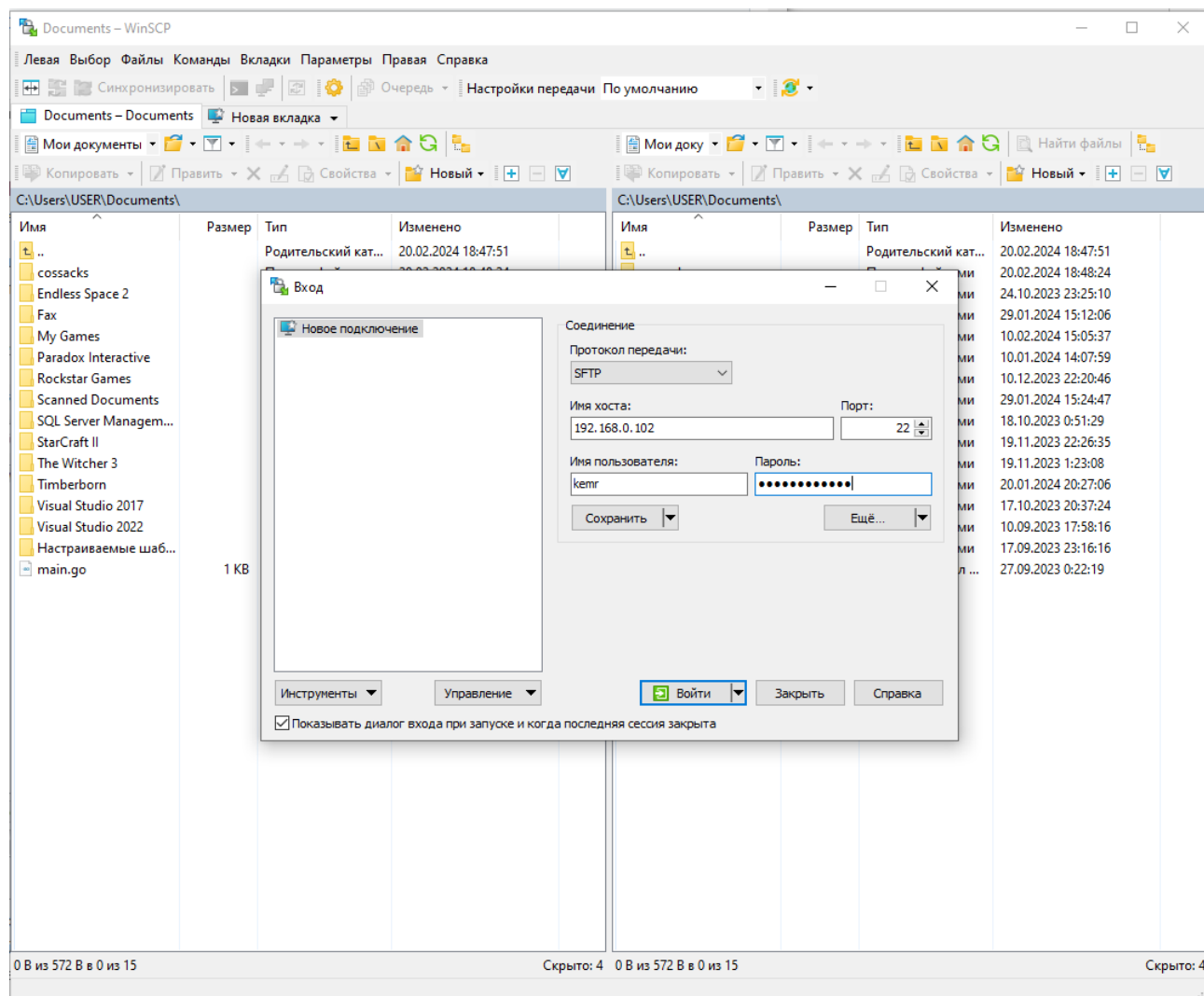


37. Встроенный файловый менеджер Ubuntu Desktop может без проблем подключиться к удалённому серверу и с удалёнными файлами можно будет

работать также, как и со своими собственными:



38. Рассмотрю вариант решения этой задачи под Windows. Проще всего воспользоваться программой WinSCP.



Настройка ssh подключения

40. Теперь рассмотрим следующую ситуацию:

У нас есть:

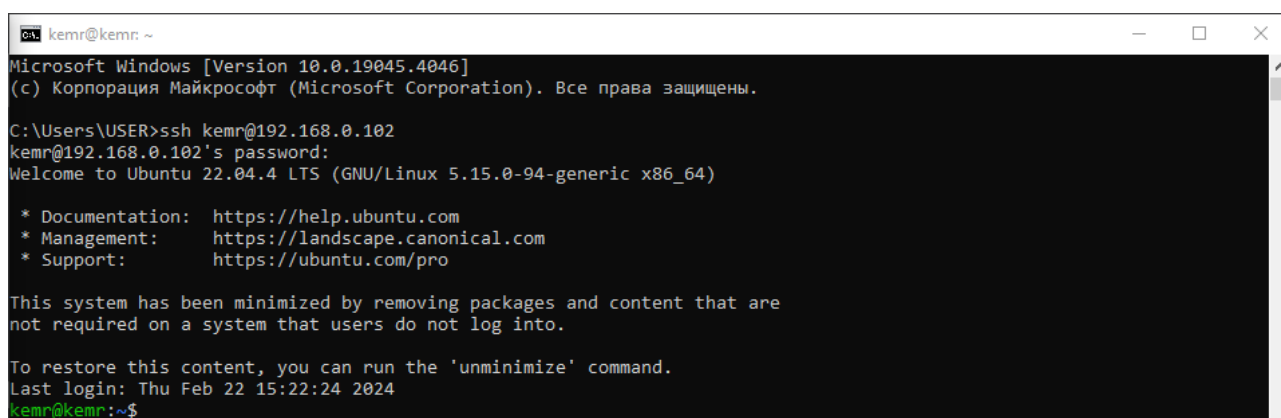
- Свежий удалённый Ubuntu **Server** с установленным ssh-сервером. Предполагается, что он запущен, но физического доступа к нему у нас нет, только через ssh.
- Машина с которой мы **администрируем** сервер. Машина всегда одна и та же и у неё фиксированный ip-адрес. В нашем случае это будет хвостовая машина (т.е. та, на которой установлен VitrualBox).
- **Пользователь** которому нужно дать доступ к серверу. Пользователь может работать с разных машин и у него динамический ip. Это будет виртуальная машина с Ubuntu Desktop.

Мы хотим:

- Настроить сервер так, чтобы доступ к пользователю-админу был только с нашей админской машины;
- Изменить способ доступа с админской машины на доступ по ключу, а не по паролю (для безопасности).
- Дать доступ к серверу обычному пользователю с любой машины по паролю. Предполагается, что у него ограниченные права, поэтому не проблема даже если пароль утечёт.

41. Запустите виртуальную машину с Ubuntu Server-ом, если она остановлена (логиниться не обязательно).

42. Из хостовой машины подключитесь к серверу по ssh через любой клиент и залогиньтесь.



```
kemr@kemr: ~  
Microsoft Windows [Version 10.0.19045.4046]  
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
  
C:\Users\USER>ssh kemr@192.168.0.102  
kemr@192.168.0.102's password:  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-94-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/pro  
  
This system has been minimized by removing packages and content that are  
not required on a system that users do not log into.  
  
To restore this content, you can run the 'unminimize' command.  
Last login: Thu Feb 22 15:22:24 2024  
kemr@kemr:~$
```

43. Для начала выполню базовые настройки безопасности ssh-сервера. Для этого отредактирую конфигурационный файл `"/etc/ssh/sshd_config"`
До:

```
root@kali: /etc/ssh nano 6.2 sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 6543
AddressFamily any
ListenAddress 0.0.0.0
ListenAddress ::

HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#Key /etc/ssh/ssh_host_rsa_key
#Key /etc/ssh/ssh_host_ecdsa_key
#Key /etc/ssh/ssh_host_ed25519_key

# Logging
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostBasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostBasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
```

После

```
GNU nano 6.2 sshd_config *

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 6543
Protocol 2
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
PermitEmptyPasswords no

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^_ Where Is   ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Теперь сохраню файл и выполню команду `sudo sshd -t`. Вывода нет, значит тестовый запуск ssh-сервера прошёл успешно и в конфиге ошибок нет.

```
kemr@kemr:/etc/ssh$ sudo sshd -t
kemr@kemr:/etc/ssh$
```

44. Перезапущу ssh-сервер, чтобы настройки вступили в силу: `sudo systemctl reload sshd`.

```
kemr@kemr:/etc/ssh$ sudo systemctl reload sshd
```

45. Теперь настрою доступ для админской машины. Для начала ограничу доступ для админского аккаунта (у меня это "kemr") только своим ip-адресом (у меня это 192.168.0.105). Это можно делать, т.к. по условию у нас фиксированный ip и он внезапно не изменится.

Для этого снова открою файл `/etc/ssh/sshd_config` и добавлю строку:

```
AllowUsers = kemr@192.168.0.105
```

46. Завершите подключение к серверу (команда `exit`), а затем снова попробую подключиться к нему по новому порту.

```
kemr@kemr:~$ exit
```

47. Следующим шагом настроим доступ для админского хоста 192.168.0.105 по ключу, а не по паролю. Эта процедура будет выполняться в несколько шагов:

Сгенерируем ключи

```
C:\Users\USER>ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\USER\.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\USER\.ssh/id_rsa.
Your public key has been saved in C:\Users\USER\.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:xnQoK9mUFUP1f+F69RVB3101pGbvnh2+93VXv3ambWY user@DESKTOP-1TMBL6L
The key's randomart image is:
+---[RSA 4096]---+
|      . = O .   O = + |
|      o o .   . B |
|      + o .   . + o + |
|      + = .   o . o |
|     o o S     . o = |
|      . .      O . * |
|      .      . + * |
|      .      + o E |
|      .      o % O |
+-----[SHA256]-----+
C:\Users\USER>
```

- Теперь когда у нас есть ключи нужно передать серверу **публичный** ключ. В этом случае сервер автоматически переключится на авторизацию по ключу. Т.е. он будет шифровать пакеты этим ключом и только тот, у кого есть приватный ключ сможет их расшифровать.

Для этого скопирую публичный ключ на сервер в каталог `~/`

```
sftp> put id_rsa.pub
Uploading id_rsa.pub to /home/kemr/id_rsa.pub
id_rsa.pub                                100% 747 749.8KB/s
sftp>
```

- Содержимое ключа нужно добавить в файл `~/.ssh/authorized_keys` (по умолчанию), только тогда ssh-сервер будет о нём знать.

```
kemr@kemr:~$ cat id_rsa.pub >> ~/.ssh/authorized_keys
```

48. Теперь можно завершить старую ssh-сессию и попробовать подключиться заново.

```
kemr@kemr:~$ exit
```

49. Выполню подключение в серверу через ssh при помощи ключа.

```
C:\Users\USER>ssh kemr@192.168.0.101 -p 6543 -i ~/.ssh/prod_rsa
Warning: Identity file C:\Users\USER/.ssh/prod_rsa not accessible: No such file or directory.
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 5.15.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sat Feb 24 11:34:59 2024
kemr@kemr:~$
```

50. После того, как я успешно подключился к серверу отключу возможность авторизации по паролю.

Для этого открою файл "/etc/ssh/sshd_config" и заменю PasswordAuthentication yes на PasswordAuthentication no. Сохраню файл и перезапустите службу ssh: sudo systemctl reload sshd.

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
PermitEmptyPasswords no
```

```
kemr@kemr:/etc/ssh$ sudo systemctl reload sshd
```

51. Настройки доступа для админа закончены. Теперь нужно выдать доступу обычному пользователю.

Для этого создадим нового пользователя, например с именем "proger": sudo adduser proger.

```
kemr@kemr:~$ sudo adduser proger
Adding user `proger' ...
Adding new group `proger' (1001) ...
Adding new user `proger' (1001) with group `proger' ...
Creating home directory `/home/proger' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for proger
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
kemr@kemr:~$
```

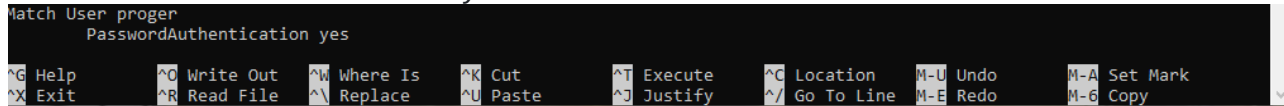
52. Сделаем для пользователя "proger" исключение и разрешим ему доступ по паролю, при этом для всех остальных пользователей (если их создадут и добавляет в список AllowUser) всё равно будет запрашиваться ключ.

Для этого открою файл "/etc/ssh/sshd_config" и добавлю пользователя "proger" в строку AllowUser через пробел:

```
AllowUser = boss@192.168.1.38 proger
AllowUsers = kemr@192.168.0.105 proger
```

Для этого пользователя нет ограничений по ip-адресам.
Затем пролистайте файл до самого низа и **в конце** добавьте:

```
Match User proger
    PasswordAuthentication yes
```



Match User позволяет переопределять значения параметров для указанных пользователей. Пользователи перечисляются через запятую.

53. Сохраню файл, проверю наличие ошибок через тестовый запуск `sudo sshd -t` и перезапущу службу, так как все хорошо.

```
kemr@kemr:/etc/ssh$ sudo sshd -t
kemr@kemr:/etc/ssh$ sudo systemctl reload sshd
```

54. Выполню подключение в серверу под пользователем "proger" через новый терминал ssh при помощи пароля. Под пользователем "proger" клонирую в свой домашний каталог репозиторий: <https://github.com/VladimirChabanov/google.git>.

```
C:\Users\USER>ssh proger@192.168.0.101 -p 6543
proger@192.168.0.101's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

proger@kemr:~$ git clone https://github.com/VladimirChabanov/google.git
Cloning into 'google'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
proger@kemr:~$ cd google
```


55. Чтобы запустить скрипт нужно будет установить зависимости через pip, но pip не установлен, а у пользователя "proger" не достаточно прав, чтобы пользоваться утилитой apt. Под админом выполните установку pip и venv.

```
kemr@kemr:~/test$ sudo apt install python3.10-venv
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kemr@kemr:/etc/ssh$ sudo apt install python3-pip
```

56. Под пользователем "proger" создам в клонированном репозитории виртуальное окружение, активирую его, скачаю зависимости перечисленные в файле "requirements.txt" и запущу скрипт "main.py".
Чтобы проверить его работоспособность выполню любой запрос, например 13

```
proger@kemr:~$ cd google
proger@kemr:~/google$ python3 -m venv ./
proger@kemr:~/google$ source ./bin/activate

(google) proger@kemr:~/google$ pip3 install -r requirements.txt
Collecting beautifulsoup4==4.11.1
  Downloading beautifulsoup4-4.11.1-py3-none-any.whl (128 kB)
    _____ 128.2/128.2 KB 842.1 kB/s eta 0:00:00
Collecting google==3.0.0
  Downloading google-3.0.0-py2.py3-none-any.whl (45 kB)
    _____ 45.3/45.3 KB 4.0 MB/s eta 0:00:00
Collecting soupsieve==2.3.2.post1
  Downloading soupsieve-2.3.2.post1-py3-none-any.whl (37 kB)
Installing collected packages: soupsieve, beautifulsoup4, google
Successfully installed beautifulsoup4-4.11.1 google-3.0.0 soupsieve-2.3.2.post1
(google) proger@kemr:~/google$ python3 main.py
Input request: 12
https://en.wikipedia.org/wiki/12_(number)
https://en.wikipedia.org/wiki/12_(number)#Name
https://en.wikipedia.org/wiki/12_(number)#Mathematical_properties
https://en.wikipedia.org/wiki/12_(number)#Religion
https://en.wikipedia.org/wiki/12_(number)#In_the_arts
https://en.wikipedia.org/wiki/12
https://www.imdb.com/title/tt0488478/
https://www.apple.com/am/iphone-12/specs/
https://www.n12.co.il/
https://www.gsmarena.com/apple_iphone_12-10509.php
(google) proger@kemr:~/google$
```

Ответы на вопросы.

- 1) SSH (Secure Shell) - это криптографический протокол, который обеспечивает безопасную коммуникацию между двумя удаленными устройствами через небезопасную сеть, такую как интернет. Он широко используется для удаленного администрирования компьютеров и передачи данных между ними.

- 2) В файле конфигурации `sshd_config`, который обычно располагается по пути `/etc/ssh/sshd_config` . Найти строчку `Port` (ваш порт). Порт по умолчанию – 22.
- 3) Добавить в `sshd_config` строчку `DenyUsers` (пользователь которому нужно запретить соединение по ssh)
- 4) Публичный - серверу, приватный - пользователю
- 5) `scp -r -i ~/.ssh/id_rsa www admin@site:/var/www/`

Вывод:

В ходе выполнения данной работы я познакомился с ssh, методами взаимодействия с ним и организацией удаленного доступа в ОС Linux.