

# **Лабораторная работа №11**

**Настройка безопасного удалённого доступа по протоколу SSH**

**Шубина София Антоновна**

# Содержание

<b>1 Цель работы</b>	<b>5</b>
<b>2 Задание</b>	<b>6</b>
<b>3 Выполнение лабораторной работы</b>	<b>7</b>
3.1 Запрет удалённого доступа по SSH для пользователя root . . . . .	7
3.2 Ограничение списка пользователей для удалённого доступа по SSH	10
3.3 Настройка дополнительных портов для удалённого доступа по SSH	12
3.4 Настройка удалённого доступа по SSH по ключу . . . . .	16
3.5 Организация туннелей SSH, перенаправление TCP-портов . . . . .	18
3.6 Запуск консольных приложений через SSH . . . . .	20
3.7 Запуск графических приложений через SSH (X11Forwarding) . . . . .	22
3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины . . . . .	24
<b>4 Выводы</b>	<b>27</b>

# Список иллюстраций

3.1 Установка пароля . . . . .	7
3.2 Мониторинг системных событий . . . . .	8
3.3 Получение доступа к серверу посредством SSH-соединения . . . . .	8
3.4 Редактирование файла . . . . .	9
3.5 Перезапуск sshd . . . . .	9
3.6 Получение доступа к серверу посредством SSH-соединения . . . . .	9
3.7 Получение доступа к серверу посредством SSH-соединения . . . . .	10
3.8 Редактирование файла . . . . .	10
3.9 Перезапуск sshd . . . . .	10
3.10 Получение доступа к серверу посредством SSH-соединения . . . . .	11
3.11 Редактирование файла . . . . .	11
3.12 Получение доступа к серверу посредством SSH-соединения . . . . .	11
3.13 Редактирование файла . . . . .	12
3.14 Перезапуск . . . . .	12
3.15 Расширенный статус работы sshd . . . . .	13
3.16 Мониторинг системных событий . . . . .	13
3.17 исправление меток . . . . .	13
3.18 Настройка межсетевого экрана . . . . .	14
3.19 Расширенный статус работы sshd . . . . .	14
3.20 Получение доступа к серверу посредством SSH-соединения . . . . .	14
3.21 logout . . . . .	15
3.22 Получение доступа к серверу посредством SSH-соединения через порт 2022 . . . . .	15
3.23 logout . . . . .	15
3.24 Редактирование файла . . . . .	16
3.25 Перезапуск . . . . .	16
3.26 Формирование ключа ssh . . . . .	17
3.27 Копирование ключа . . . . .	17
3.28 получение доступа к серверу . . . . .	18
3.29 logout . . . . .	18
3.30 просмотр какие службы запущены . . . . .	19
3.31 перенаправим порт 80 на 8082 . . . . .	19
3.32 Перенаправление на порт 8082 . . . . .	19
3.33 localhost:8082 . . . . .	20
3.34 hostname . . . . .	20
3.35 список файлов на сервере . . . . .	21
3.36 просмотр с клиента почту на сервере . . . . .	21

3.37 Редактирование файла . . . . .	22
3.38 Перезапуск sshd . . . . .	22
3.39 установка . . . . .	23
3.40 Запуск графических приложений через SSH . . . . .	23
3.41 Результат запуска графического приложения через SSH . . . . .	24
3.42 Создание файла . . . . .	25
3.43 Редактирование файла . . . . .	25
3.44 Редактирование файла . . . . .	26

# **1 Цель работы**

Приобрести практические навыки по настройке удалённого доступа к серверу с помощью SSH.

## **2 Задание**

1. Настроить запрет удалённого доступа на сервер по SSH для пользователя root.
2. Настроить разрешение удалённого доступа к серверу по SSH только для пользователей группы vagrant и вашего пользователя.
3. Настроить удалённый доступ к серверу по SSH через порт 2022.
4. Настроить удалённый доступ к серверу по SSH по ключу.
5. Организовать SSH-туннель с клиента на сервер, перенаправив локальное соединение с TCP-порта 80 на порт 8080.
6. Используя удалённое SSH-соединение, выполнить с клиента несколько команд на сервере.
7. Используя удалённое SSH-соединение, запустить с клиента графическое приложение на сервере.
8. Написать скрипт для Vagrant, фиксирующий действия по настройке SSH-сервера во внутреннем окружении виртуальной машины server. Соответствующим образом внести изменения в Vagrantfile.

# 3 Выполнение лабораторной работы

## 3.1 Запрет удалённого доступа по SSH для пользователя **root**

На сервере зададим пароль для пользователя root, если этого не было сделано ранее, я обновила пароль:

```
sudo -i  
passwd root
```

```
[root@server.sashubina.net ~]# passwd root  
New password:  
Retype new password:  
passwd: password updated successfully  
[root@server.sashubina.net ~]#
```

Рис. 3.1: Установка пароля

На сервере в дополнительном терминале запустим мониторинг системных событий:

```
sudo -i  
journalctl -x -f
```

```
[sashubina@server.sashubina.net ~]$ sudo -i
[sudo] password for sashubina:
[root@server.sashubina.net ~]# journalctl -x -f
Nov 02 18:54:43 server.sashubina.net kernel: traps: VBoxClient[16424] trap int3 ip:41dd4b sp:7fac1635cd0 error:0 in VBo
xClient[16d4b, 400000+bb000]
Nov 02 18:54:43 server.sashubina.net systemd[1]: Process 16421 (VBoxClient) of user 1001 terminated abnorma
lly with signal 5/TRAP, processing...
Nov 02 18:54:43 server.sashubina.net systemd[1]: Started systemd-coredump@303-16425-0.service - Process Core Dump (PID 1
6425/UID 0).
Subject: A start job for unit systemd-coredump@303-16425-0.service has finished successfully
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support

A start job for unit systemd-coredump@303-16425-0.service has finished successfully.

The job identifier is 12619.
Nov 02 18:54:43 server.sashubina.net systemd-coredump[16426]: [.] Process 16421 (VBoxClient) of user 1001 dumped core.

Module libXau.so.6 from rpm libXau-1.0.11-8.el10.x86_64
Module libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64
Module libX11.so.6 from rpm libX11-1.8.10-1.el10.x86_64
Module libffi.so.8 from rpm libffi-3.4.4-9.el10.x86_64
Module libwayland-client.so.0 from rpm wayland-1.23.0-2.el
10.x86_64
```

Рис. 3.2: Мониторинг системных событий

Процесс VBoxClient аварийно завершился с сигналом SIGTRAP, вызвав отладочное прерывание int3 в ядре. Система создала дамп памяти через systemd-coredump, зафиксировав используемые библиотеки (libX11, libxcb, libwayland и др.). Ошибка может быть связана с конфликтом версий библиотек или багом в гостевых дополнениях VirtualBox, но это не влияет на дальнейшую работу

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя root: `ssh root@server.sashubina.net`

```
[sashubina@client.sashubina.net ~]$ ssh root@server.sashubina.net
ssh: connect to host server.sashubina.net port 22: Connection refused
[sashubina@client.sashubina.net ~]$ 
```

Рис. 3.3: Получение доступа к серверу посредством SSH-соединения

В доступе отказано.

На сервере откроем файл `/etc/ssh/sshd_config` конфигурации `sshd` для редактирования и запретим вход на сервер пользователю `root`, установив:

`PermitRootLogin no`

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

Рис. 3.4: Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd:

```
systemctl restart sshd
```

```
[root@server.sashubina.net ~]# systemctl restart sshd
[root@server.sashubina.net ~]#
```

Рис. 3.5: Перезапуск sshd

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя root: `ssh root@server.sashubina.net`

```
[sashubina@client.sashubina.net ~]$ ssh root@server.sashubina.net
ssh: connect to host server.sashubina.net port 22: Connection refused
[sashubina@client.sashubina.net ~]$
```

Рис. 3.6: Получение доступа к серверу посредством SSH-соединения

В доступе с клиента к серверу посредством SSH соединения через пользователя root отказано. Так и должно быть, ведь мы запретили вход на сервер пользователю root.

## 3.2 Ограничение списка пользователей для удалённого доступа по SSH

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя sashubina: `ssh sashubina@server.sashubina.net`

```
[root@client.sashubina.net ~]# ssh sashubina@server.sashubina.net
kex_exchange_identification: read: Connection reset by peer
Connection reset by 192.168.1.20 port 22
```

Рис. 3.7: Получение доступа к серверу посредством SSH-соединения

Соединение через пользователя sashubina произошло успешно.

На сервере откроем файл `/etc/ssh/sshd_config` конфигурации sshd на редактирование и добавим строку `AllowUsers vagrant`

```
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
AllowUsers vagrant
```

Рис. 3.8: Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd:  
`systemctl restart sshd`

```
[root@server.sashubina.net ~]# systemctl restart sshd
[root@server.sashubina.net ~]#
```

Рис. 3.9: Перезапуск sshd

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя sashubina: `ssh sashubina@server.sashubina.net`

```
[sashubina@server.sashubina.net ~]$ ssh sashubina@server.sashubina.net
ssh: connect to host server.sashubina.net port 22: No route to host
[sashubina@server.sashubina.net ~]$ █
```

Рис. 3.10: Получение доступа к серверу посредством SSH-соединения

В доступе отказано. Тк нам надо прописать еще и пользователя sashubina  
В файле /etc/ssh/sshd\_config конфигурации sshd внесем следующее изменение:

```
AllowUsers vagrant sashubina
```

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#      X11Forwarding no
#      AllowTcpForwarding no
#      PermitTTY no
#      ForceCommand cvs server
AllowUsers vagrant sashubina█
```

Рис. 3.11: Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd  
и вновь попытаемся получить доступ с клиента к серверу посредством SSH-  
соединения через пользователя user.

```
[root@client.sashubina.net ~]# ssh sashubina@server.sashubina.net
sashubina@server.sashubina.net's password:
Web console: https://server.sashubina.net:9090/ or https://192.168.1.1:9090/
Last failed login: Sun Nov  2 20:32:00 UTC 2025 from 192.168.1.30 on ssh :notty
There were 2 failed login attempts since the last successful login.
Last login: Sun Nov  2 18:29:15 2025
[sashubina@server.sashubina.net ~]$ █
```

Рис. 3.12: Получение доступа к серверу посредством SSH-соединения

Теперь доступ успешно получен, поскольку мы разрешили пользователю  
sashubina доступ к серверу посредством ssh.

### 3.3 Настройка дополнительных портов для удалённого доступа по SSH

На сервере в файле конфигурации sshd /etc/ssh/sshd\_config найдем строку Port и ниже этой строки добавим:

Port 22

Port 2022

```
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 22
Port 2022
#AddressFamily any
#ListenAddress 0.0.0.0
```

Рис. 3.13: Редактирование файла

Эта запись сообщает процессу sshd о необходимости организации соединения через два разных порта, что даёт гарантию возможности открыть сеансы SSH, даже если была сделана ошибка в конфигурации.

После сохранения изменений в файле конфигурации перезапустим sshd:

```
systemctl restart sshd
```

```
[root@server.sashubina.net ~]# systemctl restart sshd
[root@server.sashubina.net ~]#
```

Рис. 3.14: Перезапуск

Посмотрим расширенный статус работы sshd: `systemctl status -l sshd`

```

root@server.sashubina.net ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled, preset: enabled)
   Active: active (running) since Mon 2025-10-27 16:06:32 UTC; 27s ago
     Invocation: d846a6a7be444e9ab198aedb4c0cd16b
   Docs: man:sshd(8)
         man:sshd_config(5)
 Main PID: 31319 (sshd)
   Tasks: 1 (limit: 10407)
    Memory: 1M (peak: 1.3M)
      CPU: 13ms
     CGroup: /system.slice/sshd.service
             └─31319 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Oct 27 16:06:32 server.sashubina.net systemd[1]: Starting sshd.service - OpenSSH server daemon...
Oct 27 16:06:32 server.sashubina.net (sshd)[31319]: sshd.service: Referenced but unset environment variable evaluates to an empty string.
Oct 27 16:06:32 server.sashubina.net sshd[31319]: error: Bind to port 2022 on 0.0.0.0 failed: Permission denied.
Oct 27 16:06:32 server.sashubina.net sshd[31319]: error: Bind to port 2022 on :: failed: Permission denied.
Oct 27 16:06:32 server.sashubina.net systemd[1]: Started sshd.service - OpenSSH server daemon.
Oct 27 16:06:32 server.sashubina.net sshd[31319]: Server listening on 0.0.0.0 port 22.
Oct 27 16:06:32 server.sashubina.net sshd[31319]: Server listening on :: port 22.

[lines 1-20/20: (END)]

```

Рис. 3.15: Расширенный статус работы sshd

Система сообщает нам об отказе в работе sshd через порт 2022. Дополнительно посмотрим сообщения в терминале с мониторингом системных событий.

```

***** Модуль bind_ports предлагает (точность 92.2) *****
Если вы хотите разрешить /usr/sbin/sshd для привязки к сетевому порту $PORT_ЧИСЛО
То вам need to modify the port type.
Сделать
# semanage port -a -t PORT_TYPE -p tcp 2022
  где PORT_TYPE может принимать значения: ssh_port_t, vnc_port_t, xserver_port_t.

***** Модуль catchall_booleans предлагает (точность 7.83) *****
Если хотите allow nis to enabled
То вы должны сообщить SELinux об этом, включив переключатель «nis_enabled».

Сделать
setsebool -P nis_enabled 1

***** Модуль catchall предлагает (точность 1.41) *****
Если вы считаете, что sshd должен быть разрешено name_bind доступ к порту 2022 tcp_socket по умолчанию.
То рекомендуется создать отчет об ошибке.
Чтобы разрешить доступ, можно создать локальный модуль политики.
Сделать
разрешить этот доступ сейчас, выполнив:

```

Рис. 3.16: Мониторинг системных событий

Видно, что отказ происходит из-за запрета SELinux на работу с этим портом. Исправим на сервере метки SELinux к порту 2022: `semanage port -a -t ssh_port_t -p tcp 2022`

```

[root@server.sashubina.net ~]# semanage port -a -t ssh_port_t -p tcp 2022
Port tcp/2022 already defined, modifying instead
[root@server.sashubina.net ~]#

```

Рис. 3.17: исправление меток

В настройках межсетевого экрана откроем порт 2022 протокола TCP:

`firewall-cmd --add-port=2022/tcp`

```
firewall-cmd --add-port=2022/tcp --permanent
```

```
[root@server.sashubina.net ~]# firewall-cmd --add-port=2022/tcp
Warning: ALREADY_ENABLED: '2022:tcp' already in 'public'
success
[root@server.sashubina.net ~]# firewall-cmd --add-port=2022/tcp --permanent
Warning: ALREADY_ENABLED: 2022:tcp
success
[root@server.sashubina.net ~]#
```

Рис. 3.18: Настройка межсетевого экрана

Вновь перезапустим sshd и посмотрим расширенный статус его работы. Статус должен показать, что процесс sshd теперь прослушивает два порта.

```
[root@server.sashubina.net ~]# systemctl restart sshd
[root@server.sashubina.net ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2025-11-02 20:40:41 UTC; 18s ago
   Invocation-ID: 5622ac19d2914516873b5288e7bd7681
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 31479 (sshd)
      Tasks: 1 (limit: 10407)
        Memory: 1M (peak: 1.2M)
          CPU: 1ms
         CGroup: /system.slice/sshd.service
                   └─31479 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Nov 02 20:40:41 server.sashubina.net sshd[1]: Starting sshd.service - OpenSSH server daemon...
Nov 02 20:40:41 server.sashubina.net (sshd)[31479]: sshd.service: Referenced but unset environment variable evaluates to an empty string.
Nov 02 20:40:41 server.sashubina.net sshd[31479]: Server listening on 0.0.0.0 port 2022.
Nov 02 20:40:41 server.sashubina.net sshd[31479]: Server listening on :: port 2022.
Nov 02 20:40:41 server.sashubina.net sshd[31479]: Started sshd service - OpenSSH server daemon.
Nov 02 20:40:41 server.sashubina.net sshd[31479]: Server listening on 0.0.0.0 port 22.
Nov 02 20:40:41 server.sashubina.net sshd[31479]: Server listening on :: port 22.
[lines 1-20/20 (END)]
```

Рис. 3.19: Расширенный статус работы sshd

С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя sashubina: ssh sashubina@server.sashubina.net

```
[sashubina@server.sashubina.net ~]$ ssh sashubina@server.sashubina.net
Web console: https://server.sashubina.net:9090/ or https://192.168.1.1:9090/

Last login: Sun Nov  2 20:33:38 2025 from 192.168.1.30
[sashubina@server.sashubina.net ~]$ ssh sashubina@server.sashubina.net
Web console: https://server.sashubina.net:9090/ or https://192.168.1.1:9090/

Last login: Sun Nov  2 20:34:41 2025 from 192.168.1.20
[sashubina@server.sashubina.net ~]$
```

Рис. 3.20: Получение доступа к серверу посредством SSH-соединения

После открытия оболочки пользователя введем sudo -i для получения доступа root. Отлогинемся от root и нашего пользователя на сервере, введя дважды logout

```
[sashubina@server.sashubina.net ~]$ sudo -i  
[sudo] password for sashubina:  
Sorry, try again.  
[sudo] password for sashubina:  
[root@server.sashubina.net ~]#  
logout  
[sashubina@server.sashubina.net ~]$  
logout  
Connection to server.sashubina.net closed.  
[sashubina@server.sashubina.net ~]$
```

---

Рис. 3.21: logout

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя user, указав порт 2022: ssh sashubina@server.sashubina.net

```
[sashubina@server.sashubina.net ~]$ ssh -p2022 sashubina@server.sashubina.net  
Web console: https://server.sashubina.net:9090/ or https://192.168.1.1:9090/  
  
Last login: Sun Nov 2 20:45:04 2025 from 192.168.1.20  
[sashubina@server.sashubina.net ~]$
```

Рис. 3.22: Получение доступа к серверу посредством SSH-соединения через порт 2022

После открытия оболочки пользователя введем sudo -i для получения доступа root. Отлогинемся от root и нашего пользователя на сервере, введя дважды logout

```
[sashubina@server.sashubina.net ~]$ sudo -i  
[sudo] password for sashubina:  
[root@server.sashubina.net ~]#  
logout  
[sashubina@server.sashubina.net ~]$  
logout  
Connection to server.sashubina.net closed.  
[sashubina@server.sashubina.net ~]$
```

---

Рис. 3.23: logout

## 3.4 Настройка удалённого доступа по SSH по ключу

В этом упражнении создадим пару из открытого и закрытого ключей для входа на сервер.

На сервере в конфигурационном файле /etc/ssh/sshd\_config зададим параметр, разрешающий аутентификацию по ключу: PubkeyAuthentication yes

```
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10  
  
PubkeyAuthentication yes
```

Рис. 3.24: Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd.

```
[root@server.sashubina.net ~]# systemctl restart sshd
```

Рис. 3.25: Перезапуск

На клиенте сформируем SSH-ключ, введя в терминале под пользователем sashubina: ssh-keygen

Когда спросят, хотим ли мы использовать кодовую фразу, нажмем Enter, чтобы использовать установку без пароля. При запросе имени файла, в котором будет храниться закрытый ключ, примем предлагаемое по умолчанию имя файла (~/.ssh/id\_rsa). Когда попросят ввести кодовую фразу, нажмем Enter дважды.

```
[sashubina@server.sashubina.net ~]$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/sashubina/.ssh/id_ed25519):
/home/sashubina/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase for "/home/sashubina/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sashubina/.ssh/id_ed25519
Your public key has been saved in /home/sashubina/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ryHx6qyF8XUWeuloLU/wfadbkprmJ3oVNsaHhy4B7U sashubina@server.sash
ubina.net
The key's randomart image is:
+--[ED25519 256]--+
|          . |
|         . o . |
|        . =... |
|       . oo=.o |
|      . . S oo.E* |
|     + + o =o+oo|
|    . + o .o=B+oo|
|   o o o.++*o |
|  .o+ . ...=o. |
+---[SHA256]---+
[sashubina@server.sashubina.net ~]$
```

Рис. 3.26: Формирование ключа ssh

Закрытый ключ теперь будет записан в файл `~/.ssh/id_rsa`, а открытый ключ записывается в файл `~/.ssh/id_rsa.pub`.

Скопируем открытый ключ на сервер, введя на клиенте: `ssh-copy-id sashubina@server.sashubina.net`

При запросе введем пароль пользователя на удалённом сервере.

```
[sashubina@server.sashubina.net ~]$ ssh-copy-id sashubina@server.sashubi
na.net
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/sas
hubina/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to
filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you ar
e prompted now it is to install the new keys
sashubina@server.sashubina.net's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'sashubina@server.sashubina
.net'"
and check to make sure that only the key(s) you wanted were added.

[sashubina@server.sashubina.net ~]$
```

Рис. 3.27: Копирование ключа

Попробуем получить доступ с клиента к серверу посредством SSH-соединения:

```
ssh sashubina@server.sashubina.net
```

```
[sashubina@server.sashubina.net ~]$ ssh sashubina@server.sashubina.net
Web console: https://server.sashubina.net:9090/ or https://192.168.1.1:9
090/
Last login: Sun Nov  2 20:47:13 2025 from 192.168.1.20
[sashubina@server.sashubina.net ~]$ █
```

Рис. 3.28: получение доступа к серверу

Теперь мы прошли аутентификацию без ввода пароля для учётной записи удалённого пользователя. Отлогинимся от сервера

```
[sashubina@server.sashubina.net ~]$
logout
Connection to server.sashubina.net closed.
[sashubina@server.sashubina.net ~]$
logout
Connection to server.sashubina.net closed.
[root@client.sashubina.net ~]# █
```

Рис. 3.29: logout

## 3.5 Организация туннелей SSH, перенаправление TCP-портов

На клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: `lsof | grep TCP`

```
[root@client.sashubina.net ~]# lsof | grep TCP
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1001/gvfs
      Output information may be incomplete.
lsof: WARNING: can't stat() fuse.portal file system /run/user/1001/doc
      Output information may be incomplete.

systemd          root    80u   IPv6        0:11    0t0    TCP *:webcam (LISTEN)
cupid       1228     root    7u   IPv6        1:0062  0t0    TCP localhost:ipp (LISTEN)
cupid       1228     root    8u   IPv4        1:0063  0t0    TCP localhost:ipp (LISTEN)
sshd       1225     root    7u   IPv4        1:0784  0t0    TCP *:ssh (LISTEN)
sshd       1225     root    8u   IPv6        1:0787  0t0    TCP *:ssh (LISTEN)
master     1366     root   13u   IPv4        1:1142  0t0    TCP localhost:smtp (LISTEN)
firefox    12493   sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12521 Firefox    sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
firefox    12493 12522 RaylandPr sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12523 pool-spaw sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12524 gmain sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12525 dconfx2d sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12526 clean_dia sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12530 IPCv201/ sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12536 Timer sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
firefox    12493 12537 Netlink sashubina 60u   IPv4        251175 0t0    TCP client.sashubina.net:49134->93.243.107.34.bc.googleuser
content.com:https (ESTABLISHED)
```

Рис. 3.30: просмотр какие службы запущены

Перенаправим порт 80 на server.sashubina.net на порт 8082, тк порт 8080 занят

```
ssh -fNL 8082:localhost:80 sashubina@server.sashubina.net
```

```
[root@client.sashubina.net ~]# ssh -fNL 0.0.0.0:8082:server.sashubina.net:80 sashubina@server.sashubina.net
sashubina@server.sashubina.net's password:
```

Рис. 3.31: перенаправим порт 80 на 8082

Вновь на клиенте посмотрим, запущены ли какие-то службы с протоколом TCP: Я использовала именно эту команду, а не `lsof | grep TCP`, потому что там не были отображены изменения `lsof -i:8082`

```
[root@client.sashubina.net ~]# lsof -i:8082
COMMAND  PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
ssh    35462 root    4u   IPv4 438316      0t0    TCP *:us-cli (LISTEN)
```

Рис. 3.32: Перенаправление на порт 8082

Команда демонстрирует настройку SSH-туннеля для проброса портов, что позволяет безопасно обращаться к веб-серверу на удаленной машине через локальный порт.

На клиенте запустим браузер и в адресной строке введем `localhost:8082`. Убедимся, что отобразится страница с приветствием «Welcome to the `server.sashubina.net` server».

Welcome to the server.sashubina.net server

Рис. 3.33: localhost:8082

## 3.6 Запуск консольных приложений через SSH

На клиенте откроем терминал под пользователем sashubina. Посмотрим с клиента имя узла сервера: `ssh sashubina@server.sashubina.net hostname`

```
[sashubina@client.sashubina.net ~]$ ssh sashubina@server.sashubina.net h  
ostname  
sashubina@server.sashubina.net's password:  
server.sashubina.net  
[sashubina@client.sashubina.net ~]$
```

Рис. 3.34: hostname

Установлено SSH-соединение с сервером server.sashubina.net под пользователем sashubina и выполнена команда `hostname`, которая подтвердила, что подключение действительно происходит к целевому серверу.

Посмотрим с клиента список файлов на сервере: `ssh sashubina@server.sashubina.net ls -Al`

```
[sashubina@client.sashubina.net ~]$ ssh sashubina@server.sashubina.net ls -Al
sashubina@server.sashubina.net's password:
total 56
-rw-----. 1 sashubina sashubina 3873 Nov  2 20:56 .bash_history
-rw-r--r--. 1 sashubina sashubina 18 Oct 29 2024 .bash_logout
-rw-r--r--. 1 sashubina sashubina 144 Oct 29 2024 .bash_profile
-rw-r--r--. 1 sashubina sashubina 549 Sep 19 13:19 .bashrc
drwx-----. 11 sashubina sashubina 4096 Sep 20 14:32 .cache
drwx-----. 10 sashubina sashubina 4096 Sep 20 15:36 .config
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Desktop
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Documents
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Downloads
drwx-----. 4 sashubina sashubina 32 Sep 19 18:03 .local
drwx-----. 5 sashubina sashubina 4096 Oct 20 22:01 Maildir
drwxr-xr-x. 5 sashubina sashubina 54 Sep 19 18:04 .mozilla
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Music
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Pictures
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Public
drwx-----. 2 sashubina sashubina 88 Nov  2 20:53 .ssh
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Templates
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-clipboard-tty2-control.pid
-rw-r----. 1 sashubina sashubina 7 Nov  2 21:56 .vboxclient-clipboard-tty2-service.pid
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-draganddrop-tty2-control.pid
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-hostversion-tty2-control.pid
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-seamless-tty2-control.pid
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-vmsvga-session-tty2-control.pid
-rw-r----. 1 sashubina sashubina 6 Nov  2 18:29 .vboxclient-vmsvga-session-tty2-service.pid
drwxr-xr-x. 2 sashubina sashubina 6 Sep 19 18:03 Videos
[sashubina@client.sashubina.net ~]$
```

Рис. 3.35: список файлов на сервере

С того же клиента выполнен вход на сервер и запущена команда `ls -Al` для просмотра содержимого домашней директории. В выводе видны стандартные каталоги пользователя, скрытые конфигурационные файлы, а также присутствует специализированная почтовая директория `Maildir`, что указывает на использование сервером почтовой системы формата `Maildir`.

Посмотрите с клиента почту на сервере: `ssh sashubina@server.sashubina.net`  
`MAIL=~/Maildir/ mail`

```
[sashubina@client.sashubina.net ~]$ ssh sashubina@server.sashubina.net M
AIL=~/Maildir/ mail
sashubina@server.sashubina.net's password:
s-mail version v14.9.24. Type `?' for help
/home/sashubina/Maildir: 4 messages 3 unread
  1 Sofia          2025-10-12 15:23  18/630  "1
  =
▶U 2 Sofia          2025-10-12 17:16  18/630  "3
  =
U 3 Super User      2025-10-20 16:58  18/605  "LMTP test
  =
U 4 Super User      2025-10-20 22:01  18/605  "LMTP test
  =
```

Рис. 3.36: просмотр с клиента почту на сервере

Через SSH-сессию запущена почтовая программа `mail` (`s-mail`), которая настроена на работу с директорией `Maildir`. Программа показывает почтовый ящик с 4 сообщениями, 3 из которых не прочитаны. Сообщения пришли от

отправителей “Sofia” и “Super User”, причем два из них с темой “LMTP test”, что свидетельствует о тестировании работы почтового сервера по протоколу LMTP.

## 3.7 Запуск графических приложений через SSH (X11Forwarding)

На сервере в конфигурационном файле /etc/ssh/sshd\_config разрешим отображать на локальном клиентском компьютере графические интерфейсы X11:  
X11Forwarding yes

```
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
X11UseLocalhost no
```

Рис. 3.37: Редактирование файла

После сохранения изменения в конфигурационном файле перезапустим sshd.

```
[root@server.sashubina.net ~]# systemctl restart sshd
[root@server.sashubina.net ~]#
```

Рис. 3.38: Перезапуск sshd

Для начала необходимо загрузить xorg на сервер и клиент, только после этого у меня получилось запустить firefox

```

xorg-x11-xauth      x86_64      1:1.1.2-8.el10      appstream      34 k
Installing dependencies:
libXmu              x86_64      1.1.4-8.el10      appstream      76 k
libXt               x86_64      1.3.0-5.el10      appstream     180 k

Transaction Summary
=====
Install 3 Packages

Total download size: 291 k
Installed size: 678 k
Downloading Packages:
(1/3): xorg-x11-xauth-1.1.2-8.el10.x86_64      11 kB/s | 34 kB      00:03
(2/3): libXmu-1.1.4-8.el10.x86_64.rpm          24 kB/s | 76 kB      00:03
(3/3): libXt-1.3.0-5.el10.x86_64.rpm           57 kB/s | 180 kB     00:03
-----
Total                                         44 kB/s | 291 kB      00:06

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing      :                                                 1/1
Installing    : libXt-1.3.0-5.el10.x86_64             1/3
Installing    : libXmu-1.1.4-8.el10.x86_64             2/3
Installing    : xorg-x11-xauth-1:1.1.2-8.el10.x86_64   3/3
Running scriptlet: xorg-x11-xauth-1:1.1.2-8.el10.x86_64 3/3

Installed:
libXmu-1.1.4-8.el10.x86_64                  libXt-1.3.0-5.el10.x86_64
xorg-x11-xauth-1:1.1.2-8.el10.x86_64

```

Рис. 3.39: установка

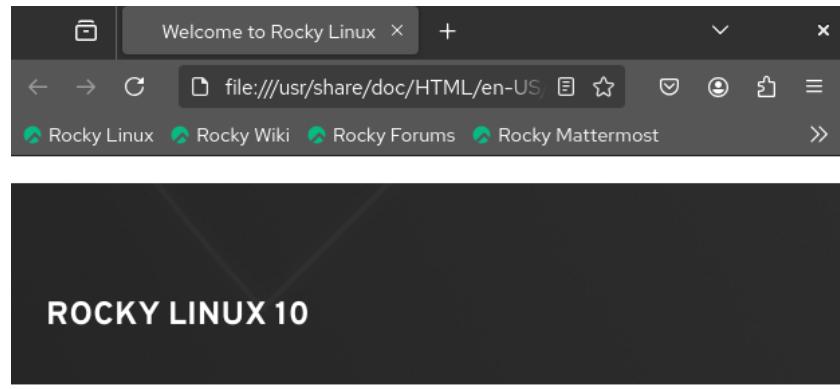
Попробуем с клиента удалённо подключиться к серверу и запустить графическое приложение, например firefox: `ssh -YC sashubina@server.sashubina.net firefox`

```

[sashubina@client.sashubina.net ~]$ ssh -YC sashubina@server.sashubina.net firefox
sashubina@server.sashubina.net's password:
[sashubina@client.sashubina.net ~]$ █

```

Рис. 3.40: Запуск графических приложений через SSH



## Welcome to Rocky Linux

### A community Enterprise Operating System

Rocky Linux is an Enterprise Linux distribution based on the sources available from the CentOS Project and Red Hat, Inc. used for CentOS Stream and Red Hat Enterprise Linux. Rocky Linux intends to be binary compatible with the upstream distributions. The source code can be found at our [Rocky Linux Git Service](#).

Rocky Linux is developed by a diverse team of core developers, with various backgrounds and skill sets that allow them to build, test, and provide Rocky Linux to the world. The core developers are supported by an active user community as well, which includes many testers from the Rocky Linux Testing Team, system administrators, system engineers,

Рис. 3.41: Результат запуска графического приложения через SSH

На сервере у нас появилось окно с Firefox

## 3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог ssh, в который поместим в соответствующие подкаталоги конфигурационный файл sshd\_config:

```
cd /vagrant/provision/server  
mkdir -p /vagrant/provision/server/ssh/etc/ssh  
cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
```

В каталоге /vagrant/provision/server создадим исполняемый файл ssh.sh:

```
cd /vagrant/provision/server  
touch ssh.sh  
chmod +x ssh.sh
```

```
[root@server.sashubina.net ~]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# mkdir -p /vagrant/provision/server/ssh/etc/ssh  
[root@server.sashubina.net server]# cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/  
[root@server.sashubina.net server]# cd /vagrant/provision/server  
[root@server.sashubina.net server]# touch ssh.sh  
[root@server.sashubina.net server]# chmod +x ssh.sh  
[root@server.sashubina.net server]# ^C  
[root@server.sashubina.net server]# nano ssh.sh
```

Рис. 3.42: Создание файла

Открыв его на редактирование, пропишем в нём следующий скрипт:

```
GNU nano 8.1  
#!/bin/bash  
echo "Provisioning script $0"  
echo "Copy configuration files"  
cp -R /vagrant/provision/server/ssh/etc/* /etc  
restorecon -vR /etc  
echo "Configure firewall"  
firewall-cmd --add-port=2022/tcp  
firewall-cmd --add-port=2022/tcp --permanent  
echo "Tuning SELinux"  
semanage port -a -t ssh_port_t -p tcp 2022  
echo "Restart sshd service"  
systemctl restart sshd
```

Рис. 3.43: Редактирование файла

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile необходимо добавить в разделе конфигурации для сервера:

```
server.vm.provision "server ssh",
  type: "shell",
  preserve_order: true,
  path: "provision/server/ssh.sh"
```

```
server.vm.provision "server ssh",
  type: "shell",
  preserve_order: true,
  path: "provision/server/ssh.sh"
```

Рис. 3.44: Редактирование файла

## **4 Выводы**

В процессе выполнения данной лабораторной работы я приобрела практические навыки по настройке удалённого доступа к серверу с помощью SSH.