# Отчёт по лабораторной работе 11

**Настройка безопасного удалённого доступа по протоколу SSH**

Заур Мустафаев

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

# 2 Выполнение работы

## 2.1 Запрет удалённого доступа по SSH для пользователя root

1. На сервере был установлен пароль для пользователя root.
2. Запущен мониторинг системных событий с помощью journalctl.
3. С клиента выполнена попытка подключения по SSH под пользователем root.


Рис. 2.1: Попытка подключиться по SSH под root

сервер запрашивает пароль, но вход не выполняется — попытки аутентификации завершаются ошибкой.

4. На сервере в файле конфигурации `/etc/ssh/sshd_config` изменён параметр `PermitRootLogin` на значение `no`.

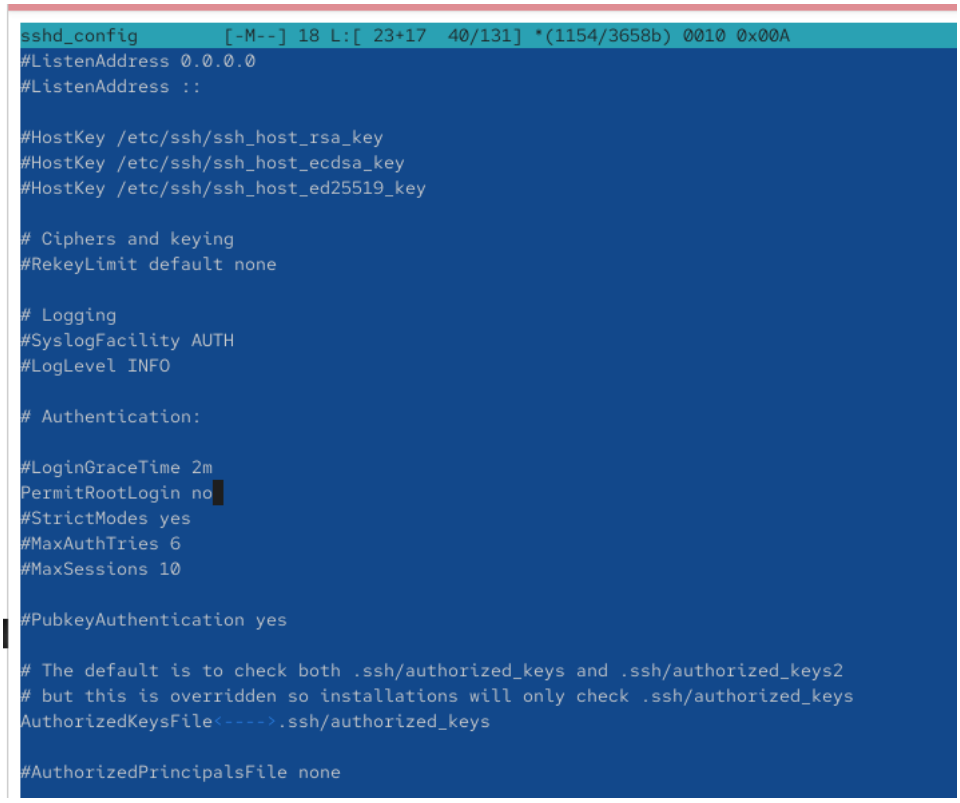


Рис. 2.2: Редактирование параметра PermitRootLogin no

5. Произведён перезапуск службы SSH.

6. Повторная попытка подключения к серверу под пользователем root.

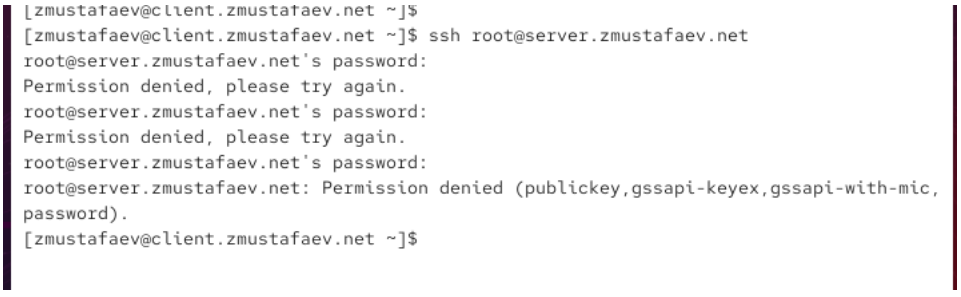


Рис. 2.3: Повторная попытка входа под root после изменения

удалённый вход под root стал невозможен — сервер блокирует попытку

подключения, выводя ошибку `Permission denied`.

Это подтверждает, что вход root по SSH запрещён.

## 2.2 Ограничение списка пользователей для удалённого доступа по SSH

1. С клиента выполнена попытка подключения по SSH под обычным пользователем.


Рис. 2.4: Успешная авторизация пользователем zmustafaev

вход выполнен успешно, пользователь получает доступ к серверу.

2. В файле `/etc/ssh/sshd_config` добавлена строка `AllowUsers vagrant`.

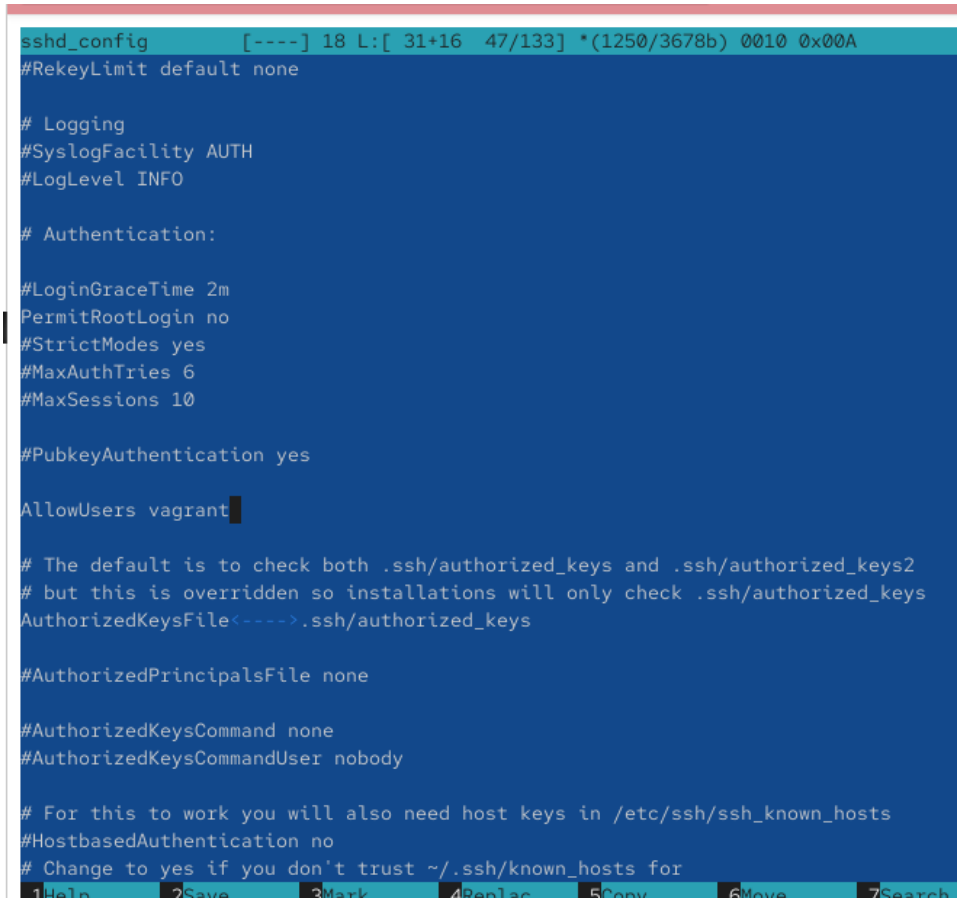


Рис. 2.5: Добавление AllowUsers vagrant

3. Служба SSH была перезапущена.

4. Выполнена повторная попытка подключения под пользователем zmustafaev.

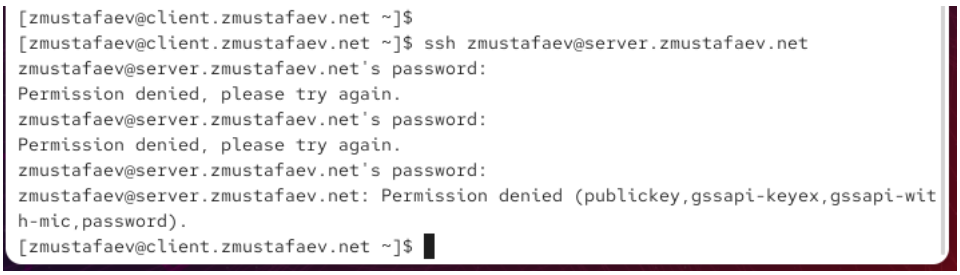


Рис. 2.6: Отказ при подключении после AllowUsers vagrant

доступ отклонён, так как в конфигурации разрешён только пользователь

vagrant.

5. В файл /etc/ssh/sshd_config добавлены два пользователя:

   AllowUsers vagrant zmustafaev

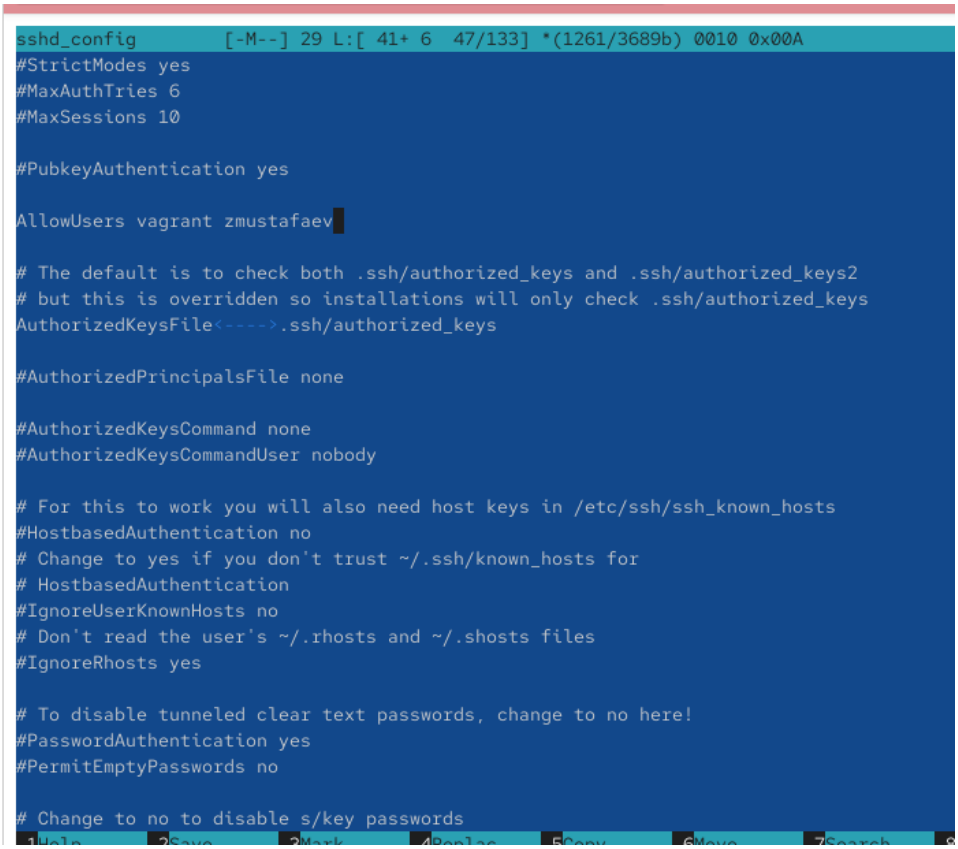


Рис. 2.7: Добавление второго пользователя в AllowUsers

6. После перезапуска службы SSH выполнена попытка входа под пользователем zmustafaev.

```
[zmustafaev@client.zmustafaev.net ~]$
[zmustafaev@client.zmustafaev.net ~]$ ssh zmustafaev@server.zmustafaev.net
zmustafaev@server.zmustafaev.net's password:
Web console: https://server.zmustafaev.net:9090/ or https://192.168.1.1:9090/

Last failed login: Mon Nov 10 12:05:08 UTC 2025 from 192.168.1.30 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Mon Nov 10 12:02:57 2025 from 192.168.1.30
[zmustafaev@server.zmustafaev.net ~]$
[zmustafaev@server.zmustafaev.net ~]$
```

Рис. 2.8: Успешный вход после добавления пользователя

теперь пользователь входит успешно, так как он добавлен в список разрешённых пользователей.

## 2.3 Настройка дополнительных портов для удалённого доступа по SSH

1. На сервере в файле конфигурации `/etc/ssh/sshd_config` добавлены строки:

   - Port 22

   - Port 2022
     Это позволяет службе SSH работать на двух портах одновременно, обеспечивая возможность подключения даже при ошибке в конфигурации одного из них.

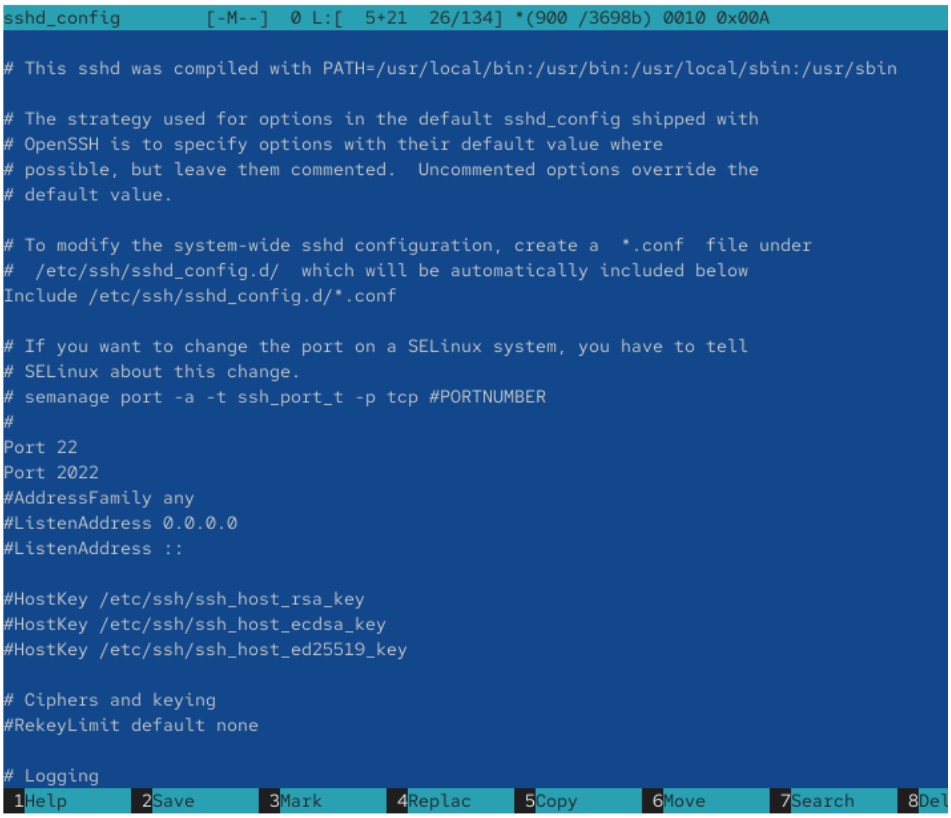


Рис. 2.9: Добавление порта 2022 в sshd_config

2. После сохранения изменений служба SSH была перезапущена, затем проверен её статус.

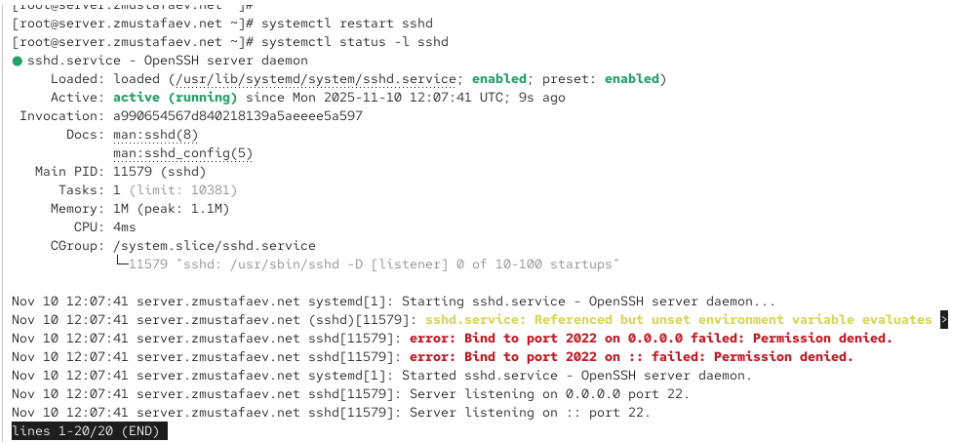


Рис. 2.10: Перезапуск и проверка статуса службы SSH

служба SSH запущена, но система сообщает об ошибке `Bind to port 2022`

`failed: Permission denied`.

Это означает, что SELinux не разрешает процессу sshd прослушивать новый порт 2022.

3. Для устранения ошибки были назначены корректные метки SELinux для порта 2022 с типом `ssh_port_t`.

4. После этого межсетевой экран был настроен на разрешение входящих соединений через порт 2022 протокола TCP.

5. Служба SSH была перезапущена повторно, и её состояние проверено повторно.

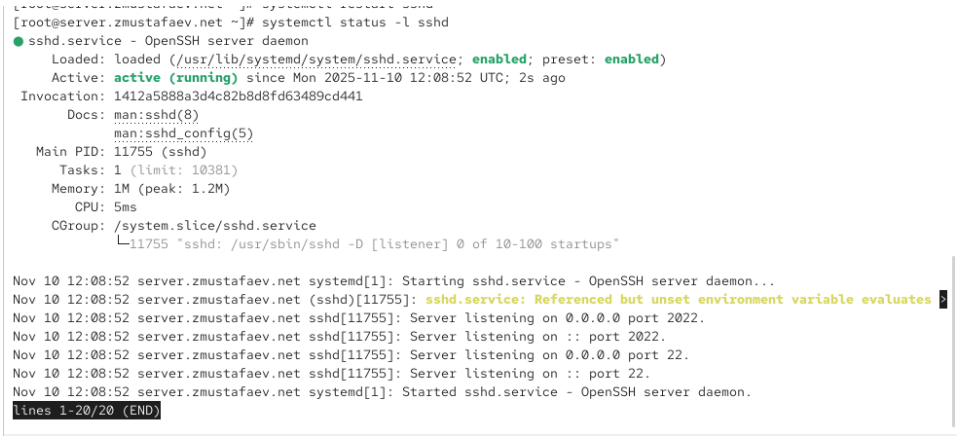


Рис. 2.11: Проверка статуса после исправления SELinux и firewall

теперь процесс SSH успешно слушает оба порта — 22 и 2022, что подтверждается сообщениями журнала.

6. С клиента выполнено подключение к серверу по SSH без указания порта.

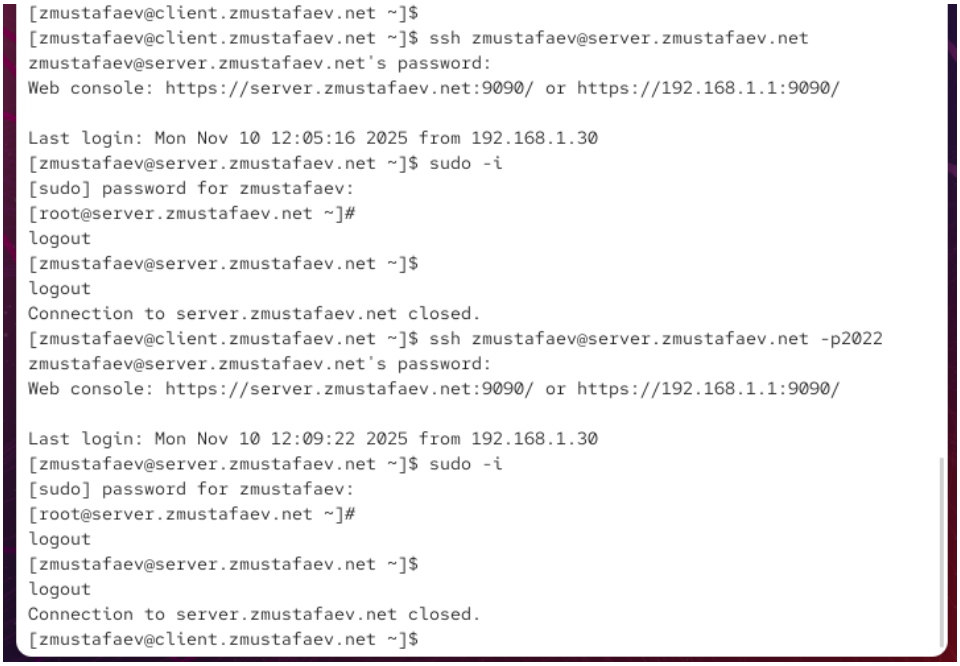


Рис. 2.12: Подключение к серверу по стандартному порту 22

соединение установлено успешно, доступ к серверу получен, пользователь смог получить права root через sudo.

7. После выхода из системы выполнена повторная попытка подключения, но с указанием нового порта 2022.
   Подключение прошло успешно, подтверждая корректную настройку дополнительного порта SSH.

## 2.4 Настройка удалённого доступа по SSH по ключу

1. На сервере в файле `/etc/ssh/sshd_config` включена аутентификация по ключу с помощью параметра `PubkeyAuthentication yes`.

2. Служба SSH была перезапущена, чтобы изменения вступили в силу.

3. На клиенте под пользователем был сгенерирован SSH-ключ (`ssh-keygen`). Созданы два файла: закрытый ключ `id_rsa` и открытый ключ `id_rsa.pub`.

4. Открытый ключ был скопирован на сервер с клиента.

5. Выполнено SSH-подключение к серверу без ввода пароля.

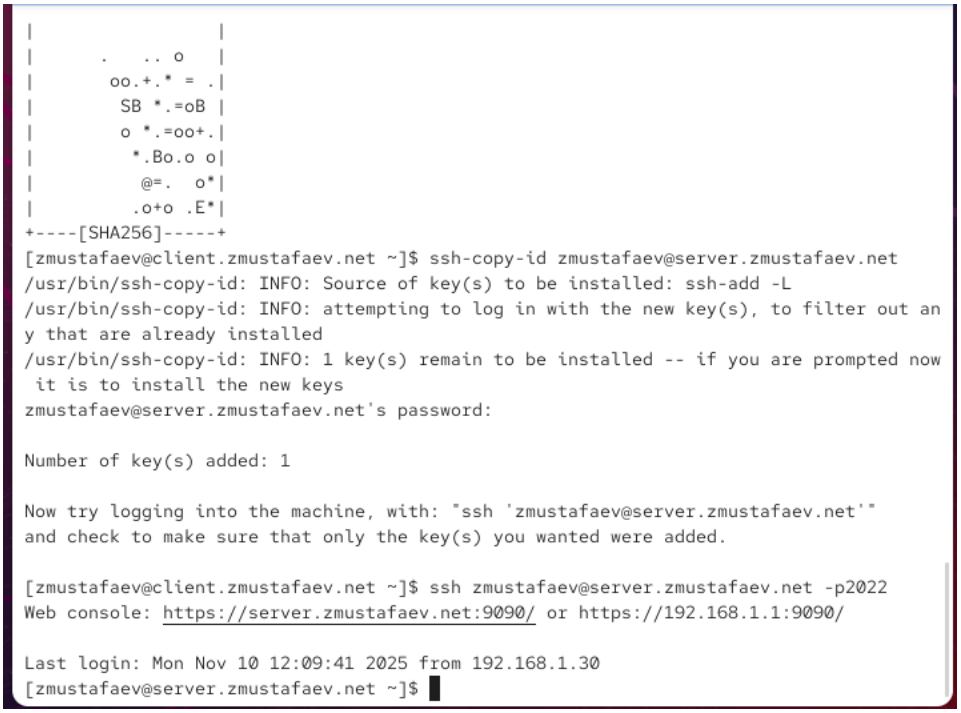


Рис. 2.13: Успешное подключение по ключу

вход на сервер по SSH теперь выполняется без запроса пароля, что подтверждает успешную настройку SSH-аутентификации по ключу.

## 2.5 Организация SSH-туннеля (перенаправление TCP-портов)

1. На клиенте был запущен просмотр активных TCP-подключений.

2. Выполнено создание SSH-туннеля: локальный порт 8080 был связан с портом 80 на сервере.


Рис. 2.14: Создание SSH-туннеля

3. Повторный вывод списка TCP-соединений показал, что теперь локальный порт 8080 находится в состоянии LISTEN.

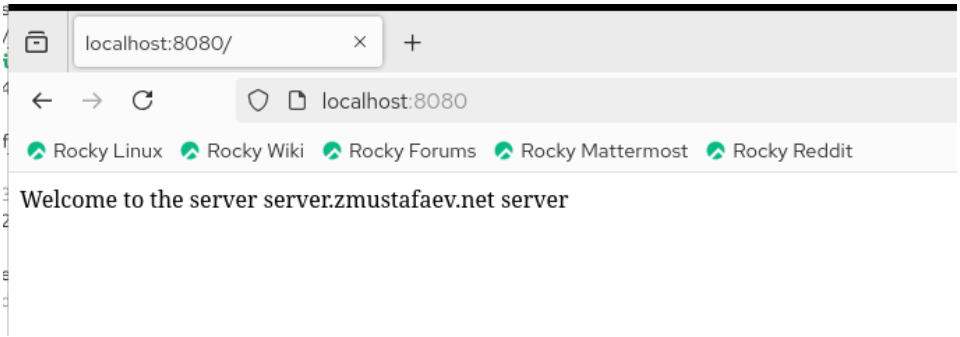


Рис. 2.15: Проверка работы туннеля

в выводе видно, что процесс `ssh` слушает локальный порт 8080, перенаправляя весь трафик на порт 80 сервера.

4. В браузере на клиенте в адресной строке введён `localhost:8080`.

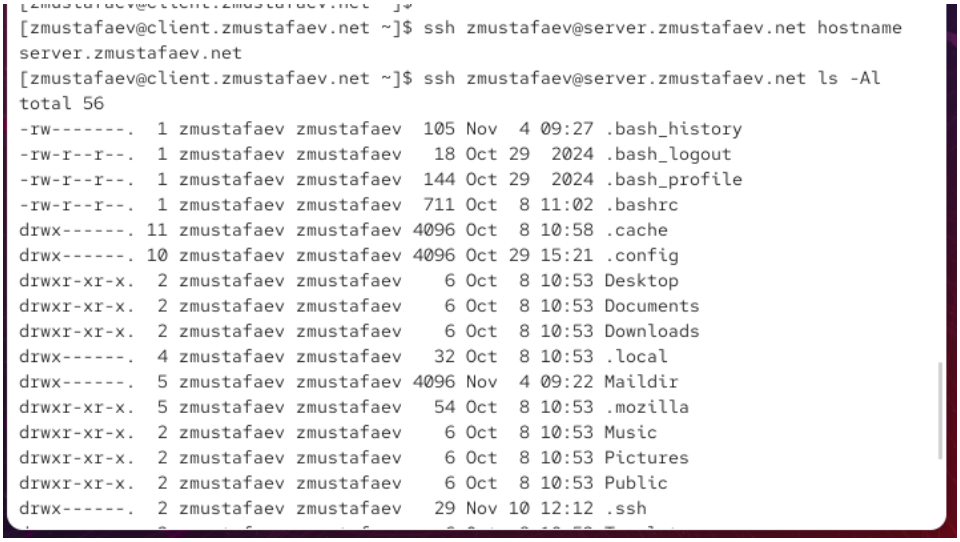


Рис. 2.16: Страница, доступная через SSH-туннель

веб-страница загружается через SSH-туннель, что подтверждает корректную работу перенаправления портов.

## 2.6 Запуск консольных приложений на сервере через SSH

1. С клиента было выполнено удалённое выполнение команды для определения имени хоста сервера.
2. Дальше выполнена команда для вывода списка файлов на сервере.
3. После этого был просмотр почтового ящика пользователя на сервере.


Рис. 2.17: Пример выполнения удалённых команд через SSH

команды выполняются на сервере без открытия полноценной интерактивной оболочки. Это позволяет управлять сервером и запускать консольные приложения напрямую из клиентского терминала.

## 2.7 Запуск графических приложений через SSH (X11Forwarding)

1. На сервере в конфигурационном файле `/etc/ssh/sshd_config` была разрешена возможность удалённого вывода графического интерфейса X11 с помощью параметра `X11Forwarding yes`.

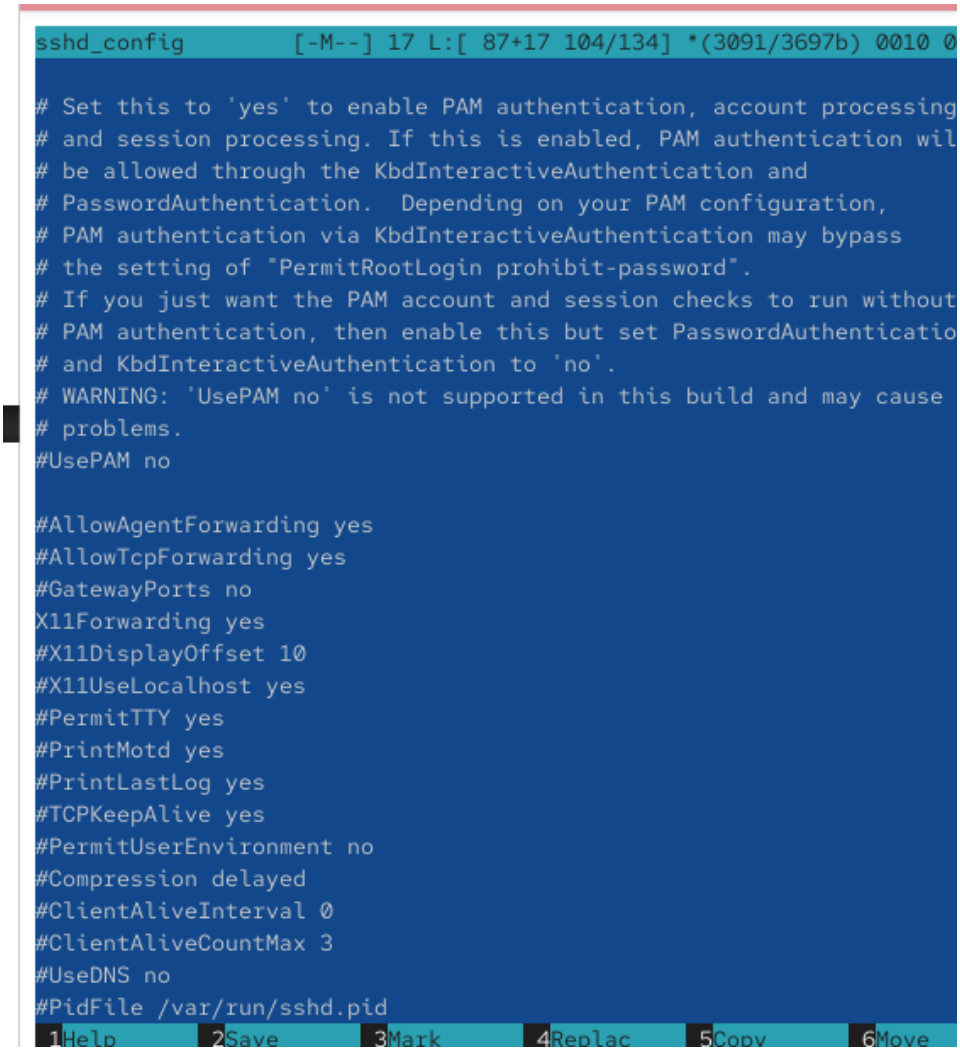


Рис. 2.18: Включение X11Forwarding в sshd_config

2. Служба SSH была перезапущена для применения изменений.

3. На клиенте выполнена попытка удалённого запуска графического приложения (firefox) с использованием опции X11-перенаправления.

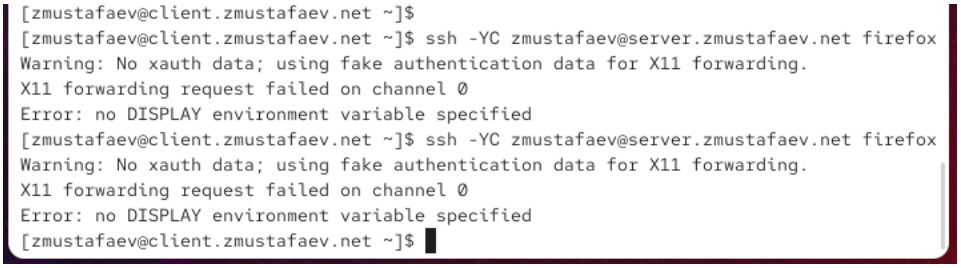


Рис. 2.19: Попытка запуска графического приложения через SSH

X11-перенаправление не выполнено. В сообщении указано, что переменная окружения DISPLAY не определена. Это означает, что на клиентской стороне не настроена X11-среда (например, отсутствует X-server).

## 2.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На сервере был выполнен переход в каталог `/vagrant/provision/server/` и создана структура каталогов для хранения конфигурационных файлов SSH.
   Затем файл `sshd_config` был скопирован в новую структуру каталогов.


Рис. 2.20: Создание каталога для конфигурации SSH

2. В каталоге `/vagrant/provision/server/` был создан исполняемый файл `ssh.sh`, в который внесён скрипт для:

   - копирования конфигурационных файлов SSH в `/etc`,
   - настройки SELinux для работы порта 2022,
   - открытия порта 2022 в firewall,
   - перезапуска службы sshd.

```
#!/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. 2.21: Содержимое скрипта ssh.sh

теперь при автоматическом запуске виртуальной машины происходит применение настроек SSH, включая открытие порта 2022, корректную настройку SELinux и перезапуск службы sshd. Это позволяет быстро развернуть виртуальную машину с заранее подготовленной конфигурацией.

# 3 Вывод

В ходе работы была выполнена настройка удалённого доступа по SSH: запрещён вход под пользователем root, ограничен доступ списком разрешённых пользователей, добавлен дополнительный порт для подключения и настроены правила SELinux и межсетевого экрана. Реализована аутентификация по ключу и создан SSH-туннель с локальным перенаправлением порта. Развернут автоматизационный скрипт, позволяющий повторно применять конфигурацию.

# 4 Контрольные вопросы

1. **Вы хотите запретить удалённый доступ по SSH на сервер пользователю root и разрешить доступ пользователю alice. Как это сделать?**
   В файле `/etc/ssh/sshd_config` установить:
   - `PermitRootLogin no` — запрет входа root по SSH,
   - `AllowUsers alice` — разрешение доступа только пользователю alice, затем перезапустить sshd.
2. **Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?**
   В файле `sshd_config` указать несколько строк `Port`, например:
   - `Port 22`
   - `Port 2022`

     Это позволяет подключаться по альтернативному порту при ошибках настройки или для повышения безопасности.
3. **Какие параметры используются для создания туннеля SSH, когда команда ssh устанавливает фоновое соединение и не ожидает конкретной команды?**
   Используются параметры:
   - `-f` — перевод ssh в фоновый режим,
   - `-N` — не запускать удалённую команду,
   - `-L` — локальное перенаправление порта.

4. **Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?**

   Использовать команду: `ssh -fNL 5555:localhost:80 user@server2.example.com`

Теперь обращение к `localhost:5555` перенаправляется на порт 80 сервера.

5. **Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?**

   Добавить порт в список разрешённых для ssh: `semanage port -a -t ssh_port_t -p tcp 2022`

6. **Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?**

   Внести порт в правила firewalld:

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

Затем перезапустить правила либо службу firewalld.