Отчёт по лабораторной работе 11

Настройка безопасного удалённого доступа по протоколу SSH

Суннатилло Махмудов

Содержание

# 1 Цель работы

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

# 2 Теоретические сведения

## 2.1 Протокол SSH

**SSH (Secure Shell)** — это сетевой протокол, обеспечивающий безопасный удалённый доступ к серверу. Он заменяет небезопасные методы подключения, такие как Telnet и rlogin, предоставляя шифрованный канал связи между клиентом и сервером.

Основные функции SSH: \* Защищённая аутентификация пользователя; \* Шифрование всего сетевого трафика; \* Контроль целостности данных; \* Возможность туннелирования и переадресации портов.

## 2.2 Архитектура SSH

* **SSH-клиент** — инициирует соединение с сервером и выполняет команды.
* **SSH-сервер** — ожидает входящих подключений и аутентифицирует пользователей.
* **SSH-ключи** — пара файлов (открытый и закрытый ключ), используемых для входа без пароля.

Схема работы: 1. Клиент устанавливает соединение с сервером.  
2. Стороны обмениваются ключами и договариваются о параметрах шифрования.  
3. Выполняется аутентификация пользователя.  
4. Создаётся защищённый канал передачи данных.

## 2.3 Основные параметры конфигурации SSH

Файл конфигурации сервера — **/etc/ssh/sshd\_config**.  
Ключевые параметры: \* **Port** — задаёт номер порта SSH (по умолчанию 22);  
\* **PermitRootLogin** — разрешает или запрещает вход под root;  
\* **AllowUsers / DenyUsers** — ограничивает список пользователей;  
\* **PasswordAuthentication** — включает или выключает вход по паролю;  
\* **PubkeyAuthentication** — разрешает аутентификацию по ключу;  
\* **X11Forwarding** — включает возможность перенаправления графического интерфейса.

# 3 Выполнение лабораторной работы

## 3.1 Запрет удалённого доступа по SSH для пользователя root

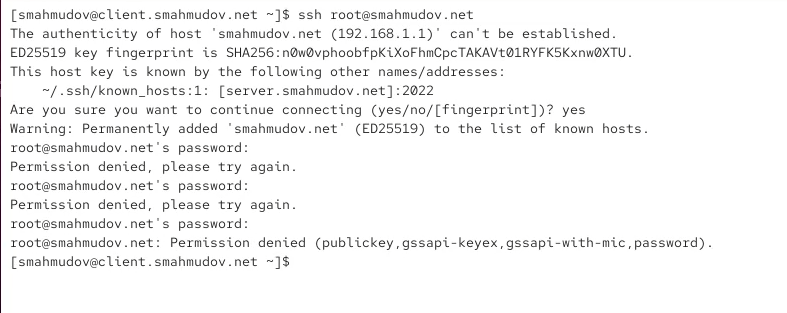
1. На сервере был задан пароль для пользователя **root**, чтобы обеспечить возможность первоначального входа при необходимости:

* sudo -i  
  passwd root

1. В отдельном терминале на сервере был запущен мониторинг системных событий в реальном времени с помощью команды:

* sudo -i  
  journalctl -x -f

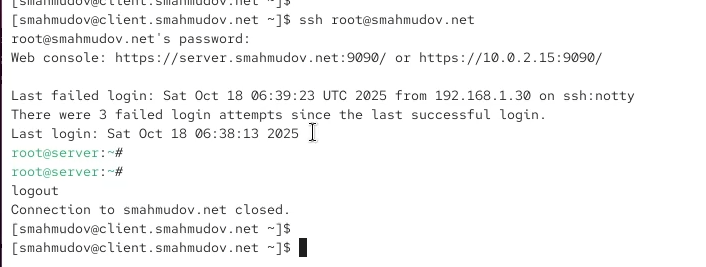
1. С клиента предпринята попытка подключения к серверу по SSH под пользователем **root**:

* ssh root@smahmudov.net
* Система запрашивала пароль, однако вход не был выполнен — сервер отклонил попытку аутентификации. Это указывает на то, что по умолчанию вход под пользователем **root** через SSH запрещён.
* 
* Рис. 1: Попытка подключения по SSH под пользователем root до разрешения

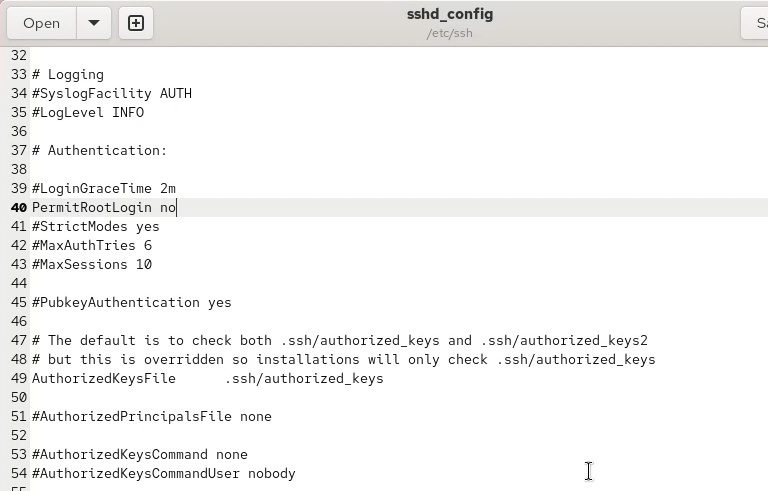
1. Для временного разрешения входа пользователь **root** в файл конфигурации **/etc/ssh/sshd\_config** было добавлено или изменено значение параметра:

* PermitRootLogin yes
* 
* Рис. 2: Изменение параметра PermitRootLogin на yes

1. После сохранения изменений был перезапущен сервис SSH:

* systemctl restart sshd
* Повторная попытка подключения к серверу под пользователем **root** завершилась успешно, что подтверждает корректность внесённых изменений.
* 
* Рис. 3: Успешное подключение к серверу под пользователем root

1. Для восстановления безопасной конфигурации параметр был возвращён в исходное состояние:

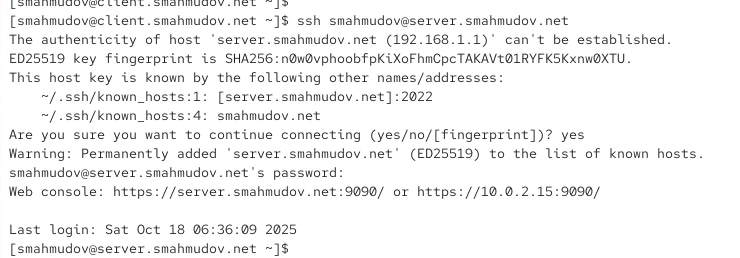
* PermitRootLogin no
* 
* Рис. 4: Возврат параметра PermitRootLogin в состояние no

1. После перезапуска SSH-службы повторная попытка входа по SSH под пользователем **root** вновь завершилась неудачей, что свидетельствует о срабатывании ограничения.

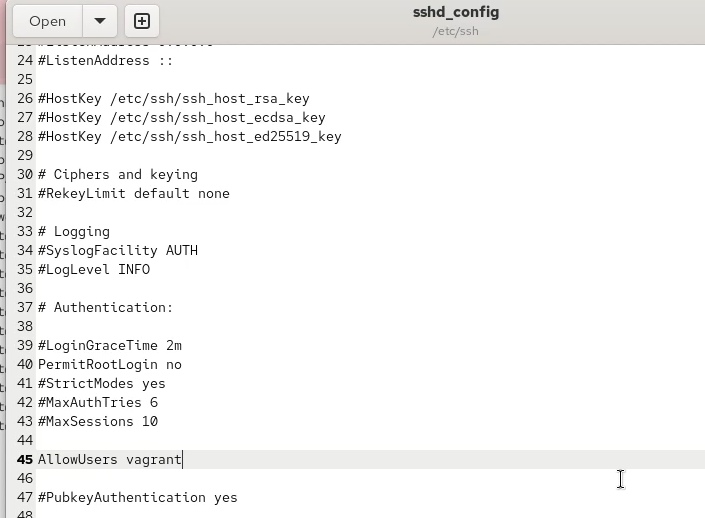
* 
* Рис. 5: Неудачная попытка входа после запрета доступа root

## 3.2 Ограничение списка пользователей для удалённого доступа по SSH

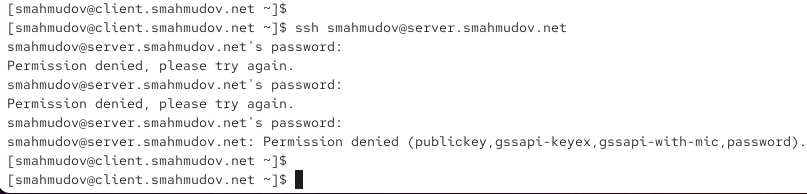
1. С клиента была предпринята попытка подключения к серверу по SSH под пользователем **smahmudov**:

* ssh smahmudov@server.smahmudov.net
* Подключение было успешно выполнено, что свидетельствует о том, что на данный момент вход по SSH для всех пользователей разрешён.
* 
* Рис. 6: Успешное подключение по SSH под пользователем smahmudov

1. На сервере был открыт для редактирования файл конфигурации **/etc/ssh/sshd\_config**, в который была добавлена строка:

* AllowUsers vagrant
* 
* Рис. 7: Добавление параметра AllowUsers vagrant

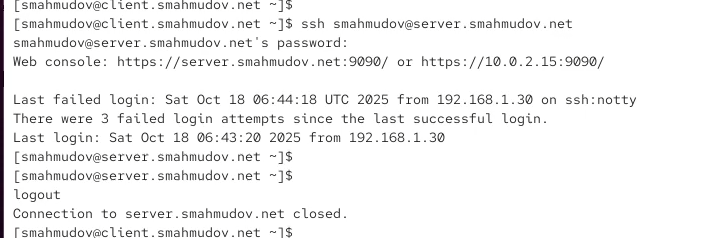
1. После сохранения изменений служба SSH была перезапущена командой:

* systemctl restart sshd
* Повторная попытка подключения с клиента под пользователем **smahmudov** завершилась неудачей. Сервер отказал в доступе, так как в списке разрешённых пользователей указан только **vagrant**.
* 
* Рис. 8: Неудачная попытка подключения при ограничении AllowUsers vagrant

1. В конфигурационный файл **/etc/ssh/sshd\_config** было внесено изменение, разрешающее доступ также пользователю **smahmudov**:

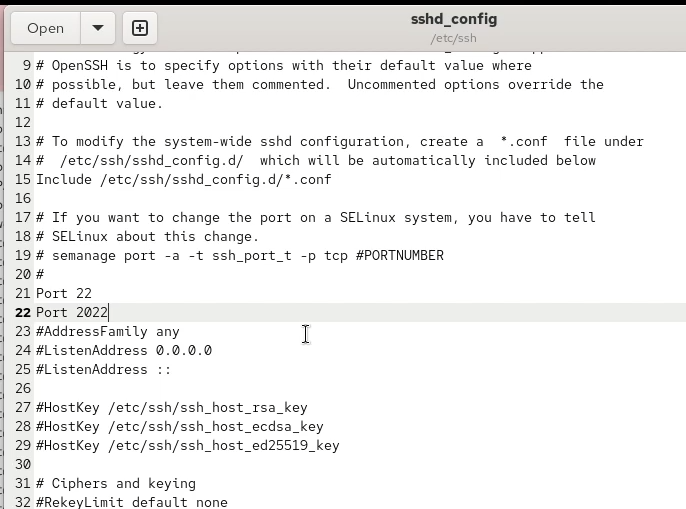
* AllowUsers vagrant smahmudov
* 
* Рис. 9: Расширение списка разрешённых пользователей

1. После сохранения изменений и перезапуска SSH-службы:

* systemctl restart sshd
* Повторная попытка подключения с клиента под пользователем **smahmudov** прошла успешно, что подтверждает корректность настройки параметра **AllowUsers**.
* 
* Рис. 10: Успешное подключение по SSH после добавления пользователя smahmudov

## 3.3 Настройка дополнительных портов для удалённого доступа по SSH

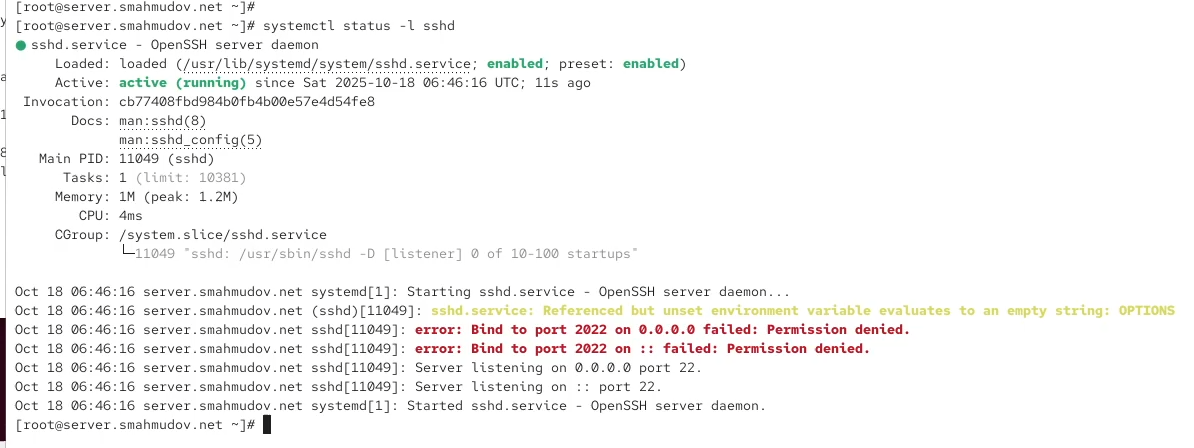
1. На сервере в файле конфигурации **/etc/ssh/sshd\_config** были добавлены строки:

* Port 22  
  Port 2022
* Это позволяет службе **sshd** принимать подключения по двум портам, обеспечивая резервный канал в случае ошибки конфигурации или блокировки стандартного порта.
* 
* Рис. 11: Добавление второго порта SSH в конфигурацию

1. После сохранения изменений была выполнена перезагрузка службы SSH:

* systemctl restart sshd

1. При просмотре расширенного статуса службы **sshd** команда systemctl status -l sshd показала сообщения об ошибке:

* error: Bind to port 2022 failed: Permission denied
* Это означает, что SELinux не разрешил процессу **sshd** использовать новый порт 2022, поскольку для него не была назначена соответствующая метка безопасности.
* 
* Рис. 12: Сообщение об ошибке при запуске sshd на новом порту

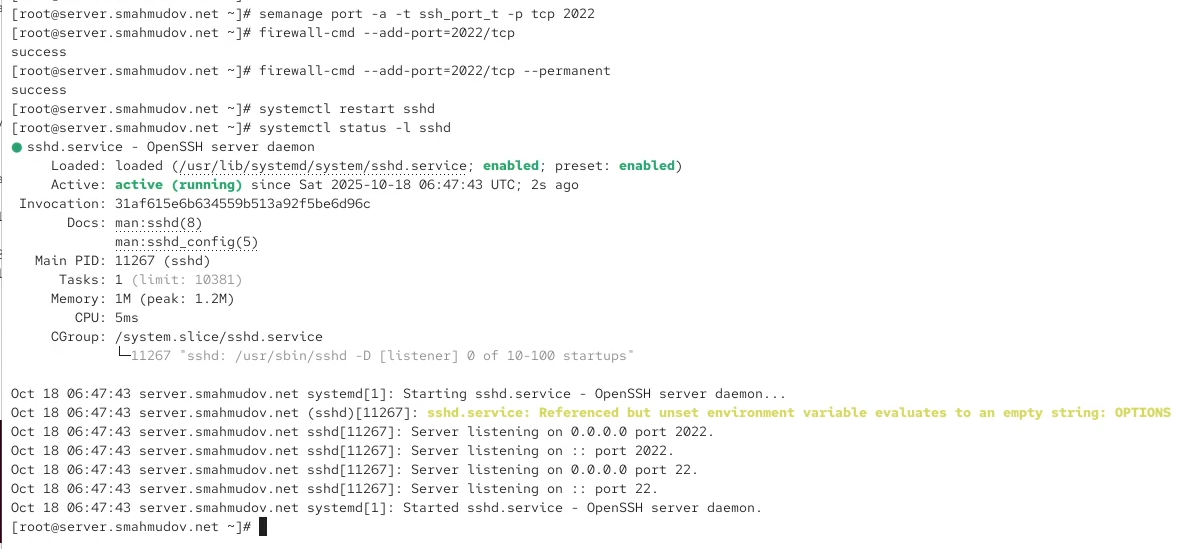
1. Для разрешения работы **sshd** на порту 2022 была выполнена команда:

* semanage port -a -t ssh\_port\_t -p tcp 2022
* После этого порт был добавлен в список разрешённых для типа **ssh\_port\_t**.

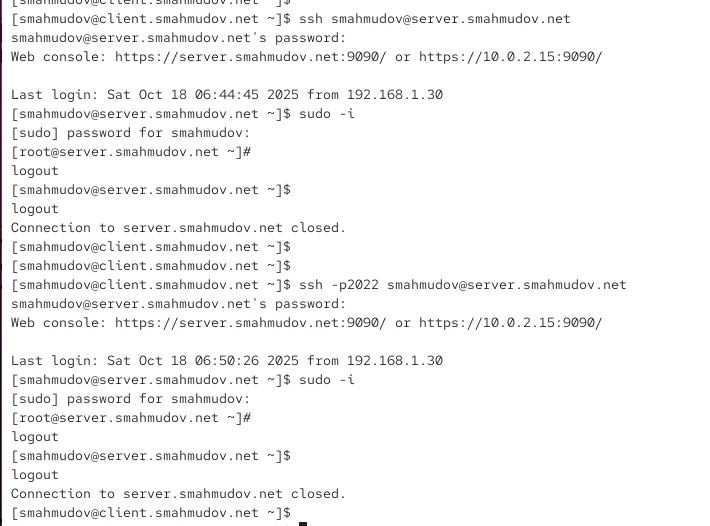
1. Далее в настройках брандмауэра был открыт новый порт для протокола TCP:

* firewall-cmd –add-port=2022/tcp  
  firewall-cmd –add-port=2022/tcp –permanent

1. После повторного перезапуска службы **sshd** её статус показал, что сервер успешно слушает подключения одновременно на портах **22** и **2022**.

* 
* Рис. 13: Успешное прослушивание портов 22 и 2022 службой sshd

1. С клиента выполнено подключение по стандартному порту 22 к серверу под пользователем **smahmudov**. Подключение прошло успешно, после чего был получен доступ администратора с помощью команды **sudo -i**.

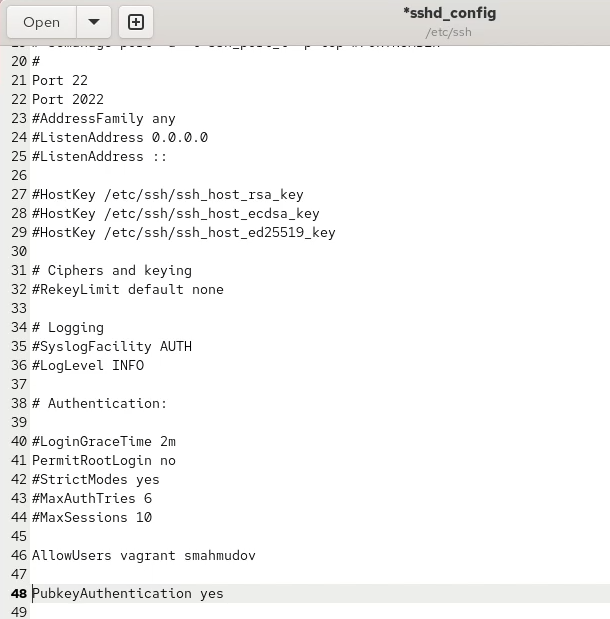
* 
* Рис. 14: Успешное подключение по стандартному порту 22

1. Затем было выполнено подключение с указанием нового порта **2022**:

* ssh -p2022 smahmudov@server.smahmudov.net
* Подключение также прошло успешно, что подтверждает корректную настройку дополнительного SSH-порта.

## 3.4 Настройка удалённого доступа по SSH по ключу

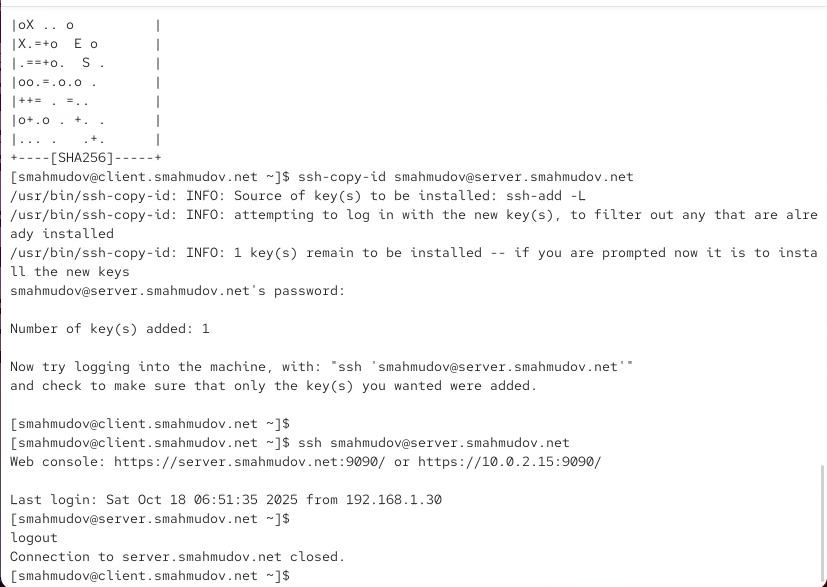
1. На сервере в файле конфигурации **/etc/ssh/sshd\_config** был включён параметр, разрешающий аутентификацию по ключу:

* PubkeyAuthentication yes
* 
* Рис. 15: Разрешение аутентификации по ключу

1. После сохранения изменений служба SSH была перезапущена для применения новых настроек:

* systemctl restart sshd

1. На клиенте была создана пара SSH-ключей при помощи команды **ssh-keygen**. После выполнения команды были сгенерированы два файла:  
   – закрытый ключ ~/.ssh/id\_rsa  
   – открытый ключ ~/.ssh/id\_rsa.pub
2. С помощью команды **ssh-copy-id** открытый ключ был скопирован на сервер:

* ssh-copy-id smahmudov@server.smahmudov.net
* После этого пользователь смог входить на сервер без запроса пароля, что подтверждает успешную настройку аутентификации по ключу.
* 
* Рис. 16: Копирование ключа на сервер и успешная авторизация без пароля

## 3.5 Организация туннелей SSH и перенаправление TCP-портов

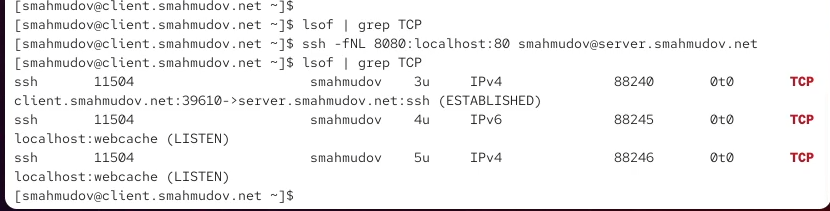
1. На клиенте была выполнена проверка активных TCP-соединений:

* lsof | grep TCP

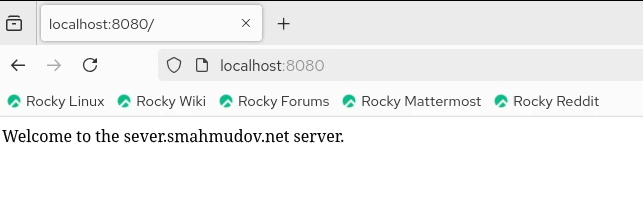
1. Для организации туннеля между клиентом и сервером был перенаправлен порт 80 сервера на локальный порт 8080:

* ssh -fNL 8080:localhost:80 smahmudov@server.smahmudov.net

1. После выполнения команды на клиенте появились новые соединения, подтверждающие создание SSH-туннеля.

* 
* Рис. 17: Проверка активных соединений после создания туннеля

1. При открытии в браузере страницы **localhost:8080** отобразилось приветствие сервера, что подтверждает успешное перенаправление порта.

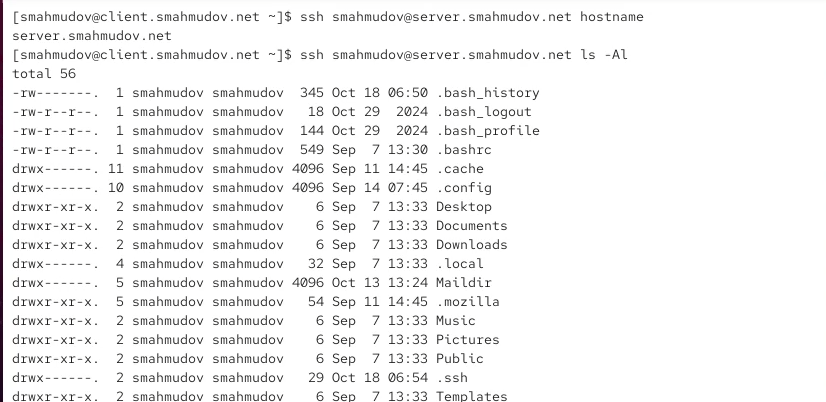
* 
* Рис. 18: Результат перенаправления порта 8080 в браузере

## 3.6 Запуск консольных приложений через SSH

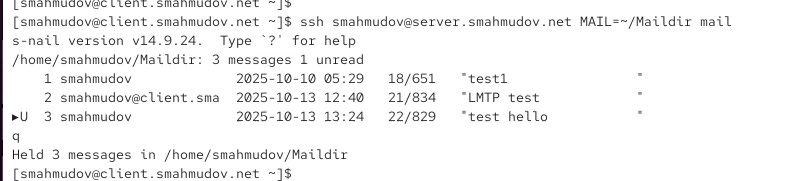
1. С клиента был выполнен запрос имени узла сервера:

* ssh smahmudov@server.smahmudov.net hostname
* Результатом стал вывод имени хоста **server.smahmudov.net**.

1. Далее был просмотрен список файлов в домашнем каталоге пользователя:

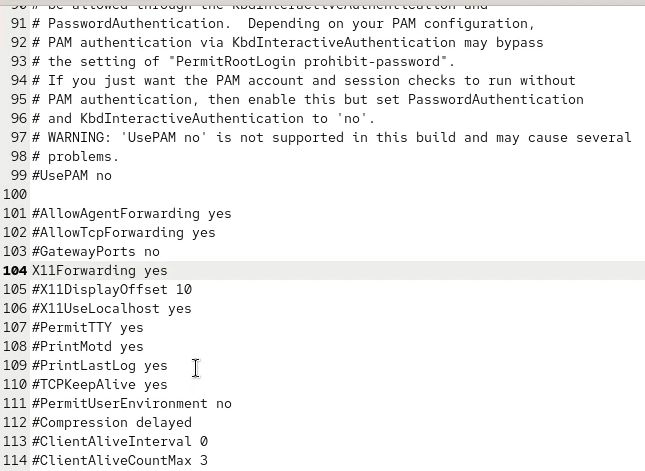
* ssh smahmudov@server.smahmudov.net ls -Al
* 
* Рис. 19: Просмотр списка файлов на сервере через SSH

1. Проверка почтового ящика показала наличие сообщений в каталоге **~/Maildir/**:

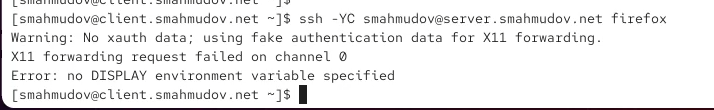
* 
* Рис. 20: Просмотр почты пользователя через SSH

## 3.7 Запуск графических приложений через SSH (X11 Forwarding)

1. На сервере в файле **/etc/ssh/sshd\_config** был разрешён показ графических интерфейсов через параметр:

* X11Forwarding yes
* 
* Рис. 21: Разрешение X11 Forwarding на сервере

1. После перезапуска службы SSH была предпринята попытка запуска браузера Firefox через SSH с параметром перенаправления X11:

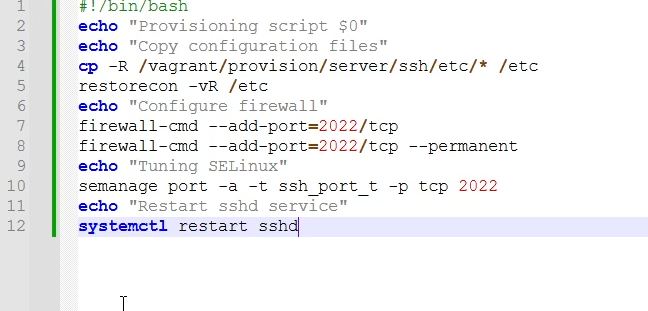
* ssh -YC smahmudov@server.smahmudov.net firefox
* Однако выполнение завершилось ошибкой из-за отсутствия переменной окружения DISPLAY, что свидетельствует об отсутствии графической среды на клиенте.
* 
* Рис. 22: Неудачная попытка запуска графического приложения через SSH

## 3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

1. В каталоге **/vagrant/provision/server/** был создан подкаталог **ssh/etc/ssh**, в который был скопирован конфигурационный файл **sshd\_config**:

* mkdir -p /vagrant/provision/server/ssh/etc/ssh  
  cp -R /etc/ssh/sshd\_config /vagrant/provision/server/ssh/etc/ssh/

1. В этом же каталоге был создан исполняемый скрипт **ssh.sh**, содержащий команды для автоматического применения сетевых настроек и перезапуска SSH-службы.

* 
* Рис. 23: Скрипт автоматической настройки и перезапуска SSH

# 4 Вывод

В ходе лабораторной работы были изучены методы конфигурирования безопасного удалённого доступа по протоколу SSH. Выполнены операции по запрету входа под пользователем root, ограничению списка пользователей, добавлению дополнительных портов для подключения и настройке SELinux и брандмауэра. Реализована аутентификация по ключам, настройка SSH-туннелей, перенаправление портов и запуск консольных приложений на удалённом сервере. Также была продемонстрирована автоматизация конфигурации SSH посредством сценария bash.

# 5 Контрольные вопросы

1. **Вы хотите запретить удалённый доступ по SSH на сервер пользователю root и разрешить доступ пользователю alice. Как это сделать?**  
   Для запрета удалённого входа под пользователем root и разрешения входа пользователю alice необходимо отредактировать файл конфигурации /etc/ssh/sshd\_config, добавив строки:  
   PermitRootLogin no  
   AllowUsers alice  
   После внесения изменений нужно перезапустить службу SSH командой systemctl restart sshd.
2. **Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?**  
   Чтобы SSH-сервер принимал подключения через несколько портов, в файле /etc/ssh/sshd\_config добавляют строки:  
   Port 22  
   Port 2022  
   Затем необходимо настроить SELinux и брандмауэр для разрешения нового порта.  
   Это может потребоваться для повышения отказоустойчивости, тестирования, а также при необходимости использовать альтернативный порт для обхода фильтров или ограничений провайдера.
3. **Какие параметры используются для создания туннеля SSH, когда команда ssh устанавливает фоновое соединение и не ожидает какой-либо конкретной команды?**  
   Для создания туннеля SSH в фоновом режиме используются параметры: -fNL [локальный\_порт]:[удалённый\_хост]:[удалённый\_порт] [пользователь]@[сервер].  
   -f переводит процесс SSH в фоновый режим,  
   -N указывает не выполнять удалённые команды,  
   -L задаёт локальную переадресацию портов.
4. **Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?**  
   Для настройки локальной переадресации необходимо выполнить команду:  
   ssh -L 5555:server2.example.com:80 [пользователь]@[сервер].  
   После выполнения команда создаёт туннель, через который локальный порт 5555 перенаправляется на порт 80 удалённого сервера server2.example.com.
5. **Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?**  
   Чтобы разрешить SSH-серверу использовать порт 2022 при активном SELinux, нужно добавить его в список разрешённых портов командой:  
   semanage port -a -t ssh\_port\_t -p tcp 2022.  
   Проверить результат можно с помощью semanage port -l | grep ssh.
6. **Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?**  
   Для разрешения подключения через порт 2022 необходимо добавить правило в firewalld:  
   firewall-cmd –add-port=2022/tcp  
   firewall-cmd –add-port=2022/tcp –permanent  
   После этого нужно перезагрузить службу брандмауэра командой systemctl restart firewalld.  
   Теперь SSH-сервер сможет принимать подключения как на стандартном порту 22, так и на дополнительном 2022.

# 6 Список литературы

1. Как пользоваться SSH - https://losst.pro/kak-polzovatsya-ssh