Лабораторная работа №11

Настройка безопасного удалённого доступа по протоколу SSH

Беличева Дарья Михайловна

Содержание

# 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

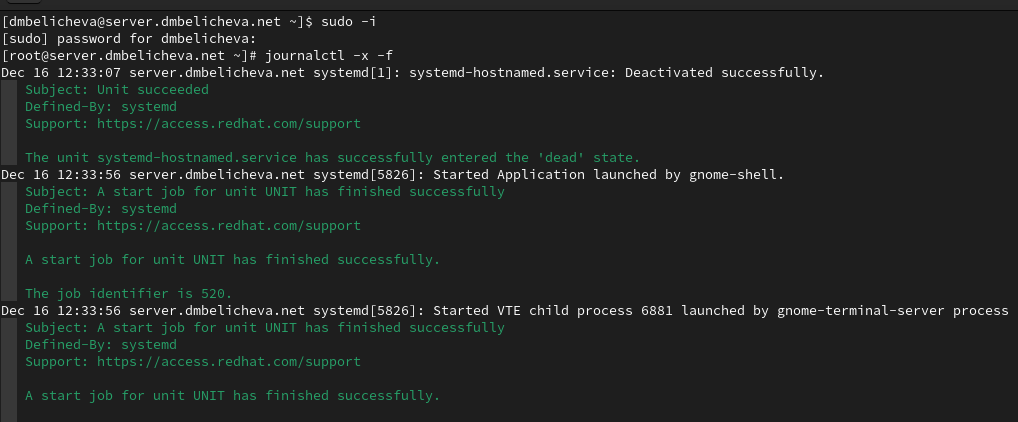
## 3.1 Запрет удалённого доступа по SSH для пользователя root

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

sudo -i  
passwd root

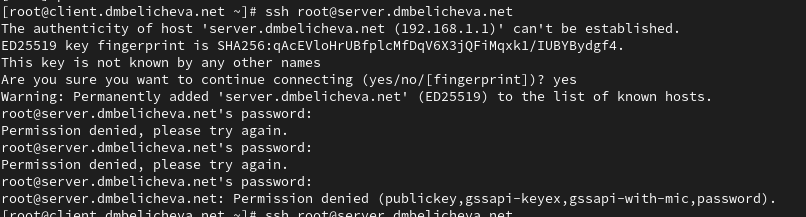
На сервере в дополнительном терминале запустим мониторинг системных событий:

sudo -i  
journalctl -x -f



Мониторинг системных событий

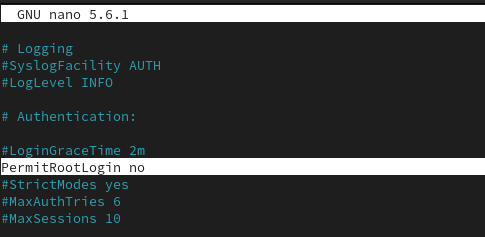
С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя root: ssh root@server.dmbelicheva.net



Получение доступа к серверу посредством SSH-соединения

В доступе отказано.

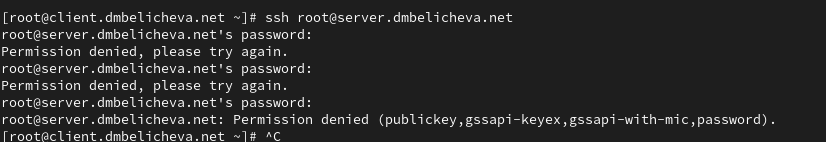
На сервере откроем файл /etc/ssh/sshd\_config конфигурации sshd для редактирования и запретим вход на сервер пользователю root, установив: PermitRootLogin no



Редактирование файла

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

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

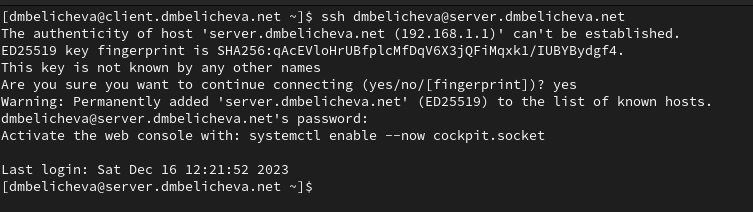


Получение доступа к серверу посредством SSH-соединения

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

## 3.2 Ограничение списка пользователей для удалённого доступа по SSH

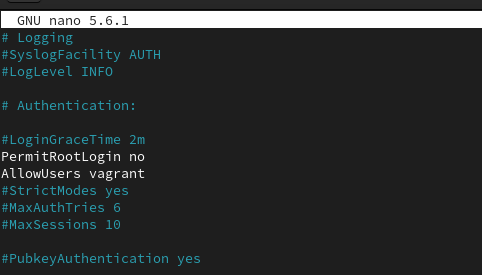
С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя dmbelicheva: ssh dmbelicheva@server.dmbelicheva.net



Получение доступа к серверу посредством SSH-соединения

Соединение через пользователя dmbelicheva произошло успешно.

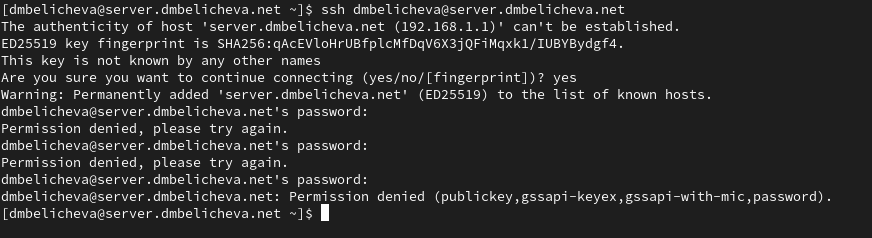
На сервере откроем файл /etc/ssh/sshd\_config конфигурации sshd на редактирование и добавим строку AllowUsers vagrant



Редактирование файла

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

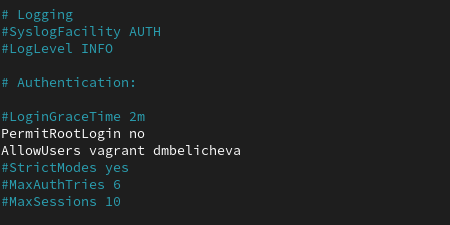
Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя dmbelicheva: ssh dmbelicheva@server.dmbelicheva.net



Получение доступа к серверу посредством SSH-соединения

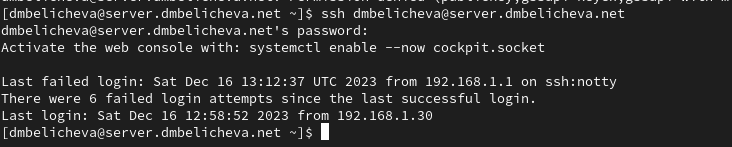
В доступе отказано.

В файле /etc/ssh/sshd\_config конфигурации sshd внесем следующее изменение: AllowUsers vagrant dmbelicheva



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.



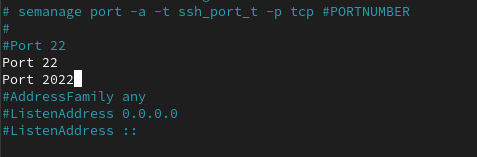
Получение доступа к серверу посредством SSH-соединения

Теперь доступ успешно получен, поскольку мы разрешили пользователю dmbelicheva 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

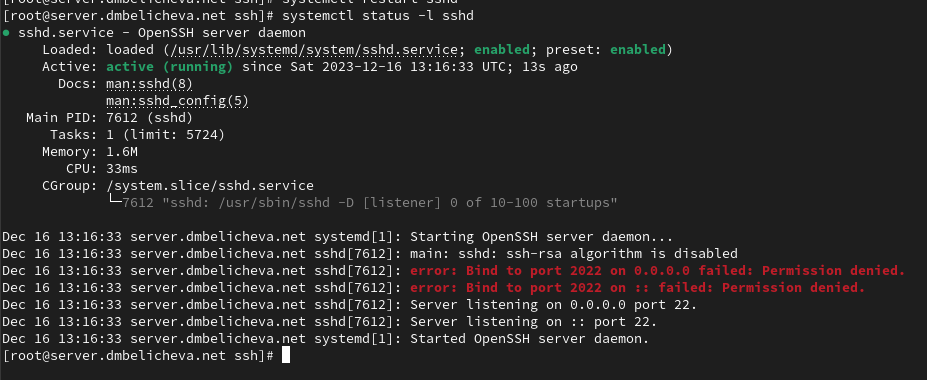


Редактирование файла

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.

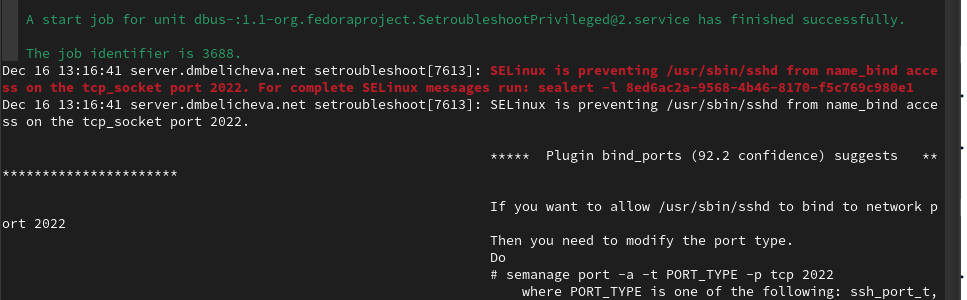
После сохранения изменений в файле конфигурации перезапустим sshd: systemctl restart sshd

Посмотрим расширенный статус работы sshd: systemctl status -l sshd



Расширенный статус работы sshd

Система должна сообщить вам об отказе в работе sshd через порт 2022.Дополнительно посмотрим сообщения в терминале с мониторингом системных событий.



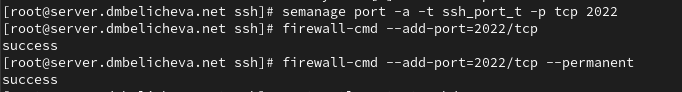
Мониторинг системных событий

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

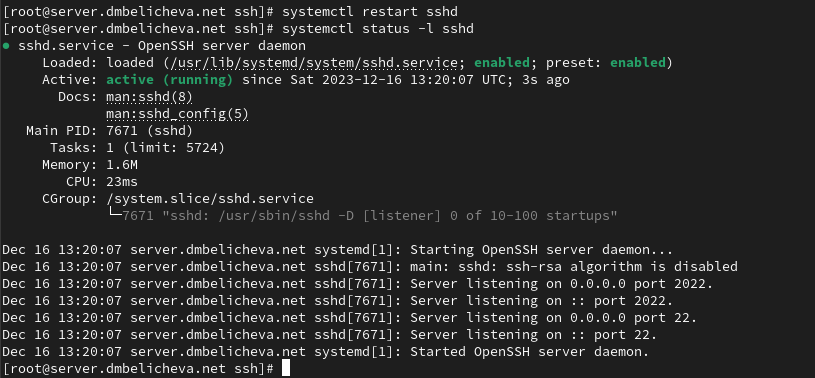
В настройках межсетевого экрана откроем порт 2022 протокола TCP:

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



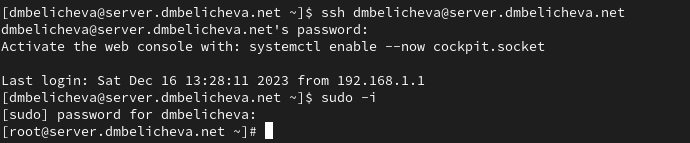
Настройка межсетевого экрана

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



Расширенный статус работы sshd

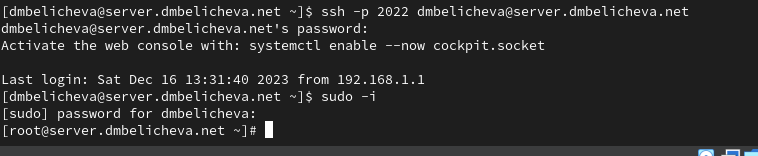
С клиента попытаемся получить доступ к серверу посредством SSH-соединения через пользователя dmbelicheva: ssh dmbelicheva@server.dmbelicheva.net



Получение доступа к серверу посредством SSH-соединения

После открытия оболочки пользователя введем sudo -i для получения доступа root.

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



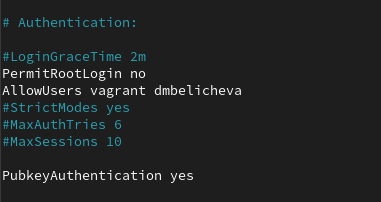
Получение доступа к серверу посредством SSH-соединения через порт 2022

После открытия оболочки пользователя введем sudo -i для получения доступа root.

## 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

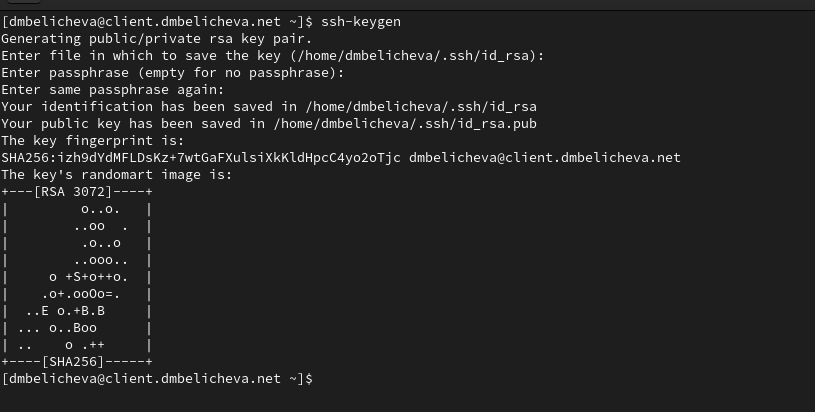


Редактирование файла

После сохранения изменений в файле конфигурации перезапустим sshd.

На клиенте сформируем SSH-ключ, введя в терминале под пользователем dmbelicheva: 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.



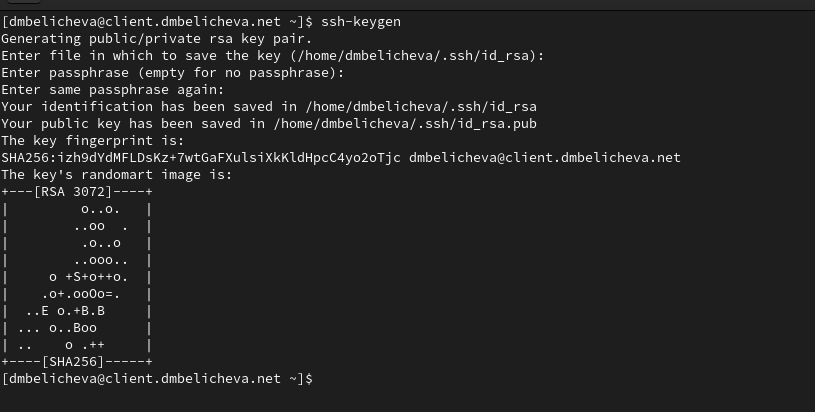
Формирование ключа ssh

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 dmbelicheva@server.dmbelicheva.net

При запросе введем пароль пользователя на удалённом сервере.

Попробуем получить доступ с клиента к серверу посредством SSH-соединения: ssh dmbelicheva@server.dmbelicheva.net



Копирование открытого ssh ключа и получение доступа к серверу

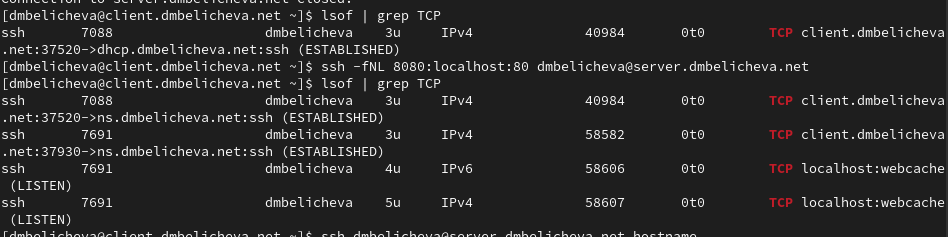
Теперь пройдем аутентификацию без ввода пароля для учётной записи удалённого пользователя.

## 3.5 Организация туннелей SSH, перенаправление TCP-портов

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

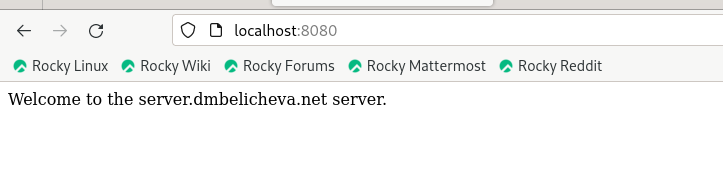
Перенаправим порт 80 на server.dmbelicheva.net на порт 8080 на локальной машине: ssh -fNL 8080:localhost:80 dmbelicheva@server.dmbelicheva.net

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



Перенаправление на порт 8080

На клиенте запустим браузер и в адресной строке введем localhost:8080. Убедимся, что отобразится страница с приветствием «Welcome to the server.dmbelicheva.net server».



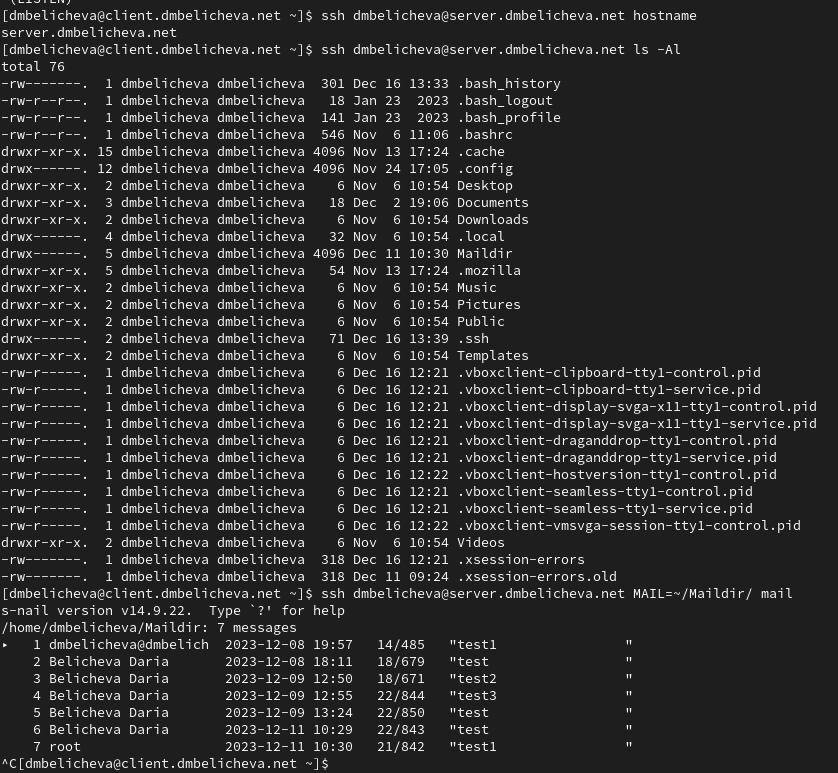
localhost:8080

## 3.6 Запуск консольных приложений через SSH

На клиенте откройте терминал под пользователем dmbelicheva. Посмотрите с клиента имя узла сервера: ssh dmbelicheva@server.dmbelicheva.net hostname

Посмотрите с клиента список файлов на сервере: ssh dmbelicheva@server.dmbelicheva.net ls -Al

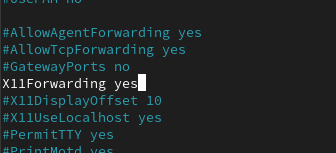
Посмотрите с клиента почту на сервере: ssh dmbelicheva@server.dmbelicheva.net MAIL=~/Maildir/ mail



Запуск консольных приложений через SSH

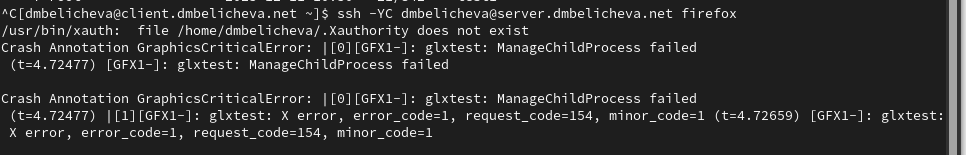
## 3.7 Запуск графических приложений через SSH (X11Forwarding)

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

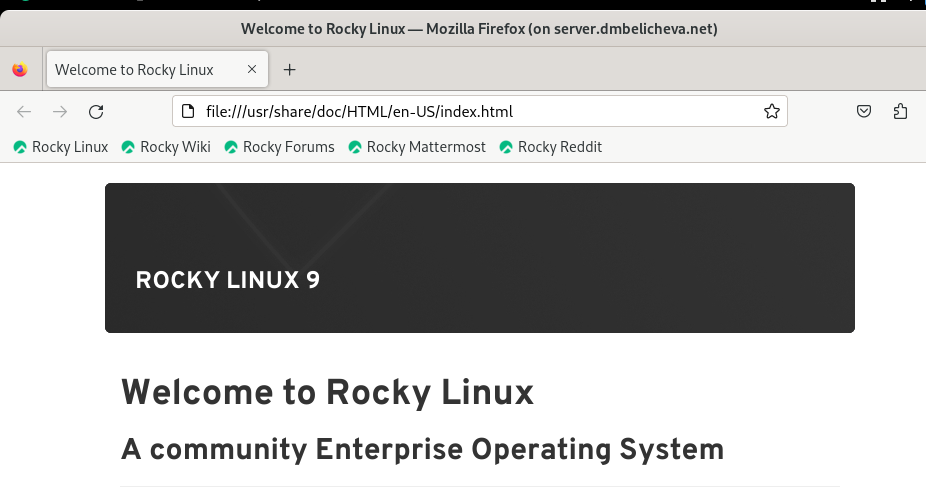


Редактирование файла

После сохранения изменения в конфигурационном файле перезапустим sshd. Попробуем с клиента удалённо подключиться к серверу и запустить графическое приложение, например firefox: ssh -YC user@server.dmbelicheva.net firefox



Запуск графических приложений через SSH



Резултат запуска графического приложения через SSH

## 3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

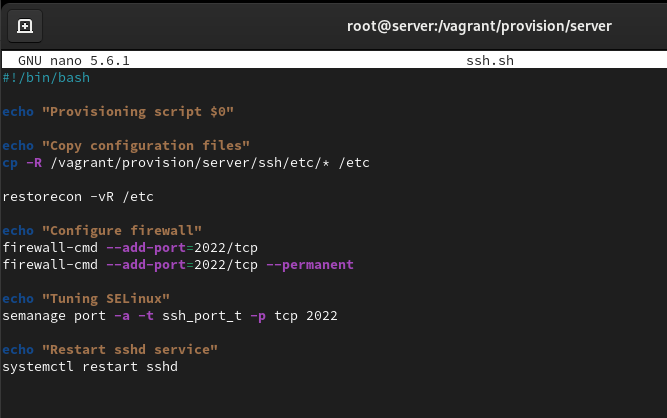
На виртуальной машине server перейдем в каталог для внесения изменений в настройки внутреннего окружения /vagrant/provision/server/, создадим в нём каталог ssh, в который поместим в соответствующие подкаталоги конфигурационный файл sshd\_config:

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

Открыв его на редактирование, пропишем в нём следующий скрипт:



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"

# 4 Выводы

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