

Лабораторная работа №11

Настройка безопасного удалённого доступа по протоколу SSH

Сахно Никита НФИбд-02-23

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	2
3.1	Запрет удалённого доступа по SSH для пользователя root.....	2
3.2	Ограничение списка пользователей для удалённого доступа по SSH	3
3.3	Настройка дополнительных портов для удалённого доступа по SSH	5
3.4	Настройка удалённого доступа по SSH по ключу.....	7
3.5	Организация туннелей SSH, перенаправление TCP-портов	8
3.6	Запуск консольных приложений через SSH	9
3.7	Запуск графических приложений через SSH (X11Forwarding)	10
3.8	Внесение изменений в настройки внутреннего окружения виртуальной машины	10
4	Выводы.....	11

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

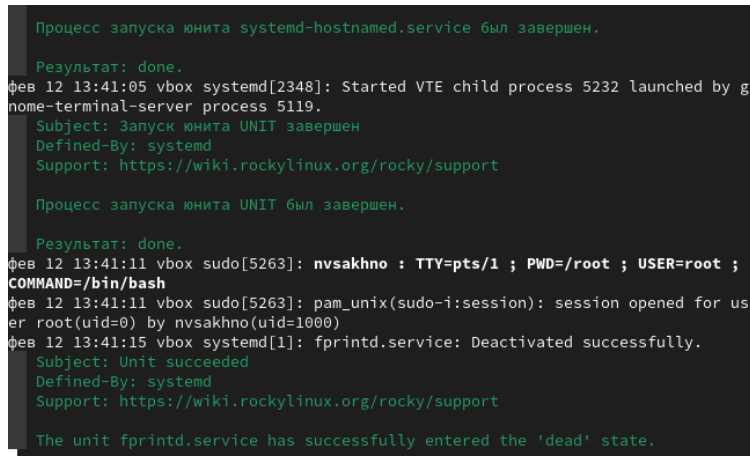
3.1 Запрет удалённого доступа по SSH для пользователя root

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:

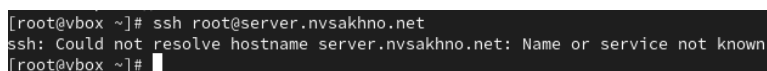
```
sudo -i  
journalctl -x -f
```



```
Процесс запуска юнита systemd-hostnamed.service был завершен.  
  
Результат: done.  
фев 12 13:41:05 vbox systemd[2348]: Started VTE child process 5232 launched by g  
nome-terminal-server process 5119.  
Subject: Запуск юнита UNIT завершен  
Defined-By: systemd  
Support: https://wiki.rockylinux.org/rocky/support  
  
Процесс запуска юнита UNIT был завершен.  
  
Результат: done.  
фев 12 13:41:11 vbox sudo[5263]: nvsakhno : TTY=pts/1 ; PWD=/root ; USER=root ;  
COMMAND=/bin/bash  
фев 12 13:41:11 vbox sudo[5263]: pam_unix(sudo-i:session): session opened for us  
er root(uid=0) by nvsakhno(uid=1000)  
фев 12 13:41:15 vbox systemd[1]: fprintd.service: Deactivated successfully.  
Subject: Unit succeeded  
Defined-By: systemd  
Support: https://wiki.rockylinux.org/rocky/support  
  
The unit fprintd.service has successfully entered the 'dead' state.
```

Мониторинг системных событий

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



```
[root@vbox ~]# ssh root@server.nvsakhno.net  
ssh: Could not resolve hostname server.nvsakhno.net: Name or service not known  
[root@vbox ~]#
```

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`

```
GNU nano 5.6.1
# Logging
#SyslogFacility AUTH
#LogLevel INFO

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

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.nvsakhno.net`

```
[root@vbox ~]# ssh nvsakhno@server.nvsakhno.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 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 `nvsakhno`: `ssh nvsakhno@server.nvsakhno.net`

```
[root@vbox ~]# ssh nvsakhno@server.nvsakhno.net
ssh: Could not resolve hostname server.nvsakhno.net: Name or service not known
[root@vbox ~]#
```

Получение доступа к серверу посредством SSH-соединения

Соединение через пользователя `nvasakhno` произошло успешно.

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

```
GNU nano 5.6.1
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

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

#PubkeyAuthentication yes
```

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 nvaskhno: `ssh nvaskhno@server.nvsakhno.net`

```
icheva.net (192.168.1.1)' can't be established.
ohrUBfplcMfDqV6X3jQFIMqk1/IUBYBydgr4.
Warning: Permanently added 'icheva.net' (ED25519) to the list of known hosts.
sword:
sword:
Access denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

Получение доступа к серверу посредством SSH-соединения

В доступе отказано.

В файле `/etc/ssh/sshd_config` конфигурации sshd внесем следующее изменение:
`AllowUsers vagrant nvaskhno`

```
# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
AllowUsers vagran nvaskhnot
#StrictModes yes
#MaxAuthTries 6

^G Справка ^O Записать ^W Поиск ^K
```

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. Теперь доступ успешно получен, поскольку мы разрешили пользователю nvaskhno 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
```

```
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
Port 22
Port 2022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

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`

```
[root@vbox ~]# ssh nvsakhno@server.nvsakhno.net
ssh: Could not resolve hostname server.nvsakhno.net: Name or service not known
[root@vbox ~]# nano /etc/ssh/sshd_config
[root@vbox ~]# systemctl restart sshd
[root@vbox ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enab
   Active: active (running) since Thu 2026-02-12 13:49:50 MSK; 8s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 5673 (sshd)
    Tasks: 1 (limit: 10513)
   Memory: 1.9M (peak: 2.4M)
      CPU: 31ms
   CGroup: /system.slice/sshd.service
           └─5673 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

фев 12 13:49:50 vbox systemd[1]: Starting OpenSSH server daemon...
фев 12 13:49:50 vbox sshd[5673]: error: Bind to port 2022 on 0.0.0.0 failed: Pe>
фев 12 13:49:50 vbox sshd[5673]: error: Bind to port 2022 on :: failed: Permiss>
фев 12 13:49:50 vbox systemd[1]: Started OpenSSH server daemon.
фев 12 13:49:50 vbox sshd[5673]: Server listening on 0.0.0.0 port 22.
фев 12 13:49:50 vbox sshd[5673]: Server listening on :: port 22.
lines 1-18/18 (END)
```

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.

```

... unit sshd.service завершён: приводится статистика по потребленным
им ресурсам.
фев 12 13:50:43 vbox kernel: SELinux: Converting 623 SID table entries...
фев 12 13:50:43 vbox kernel: SELinux: policy capability network_peer_controls=1
фев 12 13:50:43 vbox kernel: SELinux: policy capability open_perms=1
фев 12 13:50:43 vbox kernel: SELinux: policy capability extended_socket_class=1
фев 12 13:50:43 vbox kernel: SELinux: policy capability always_check_network=0
фев 12 13:50:43 vbox kernel: SELinux: policy capability cgroup_seclabel=1
фев 12 13:50:43 vbox kernel: SELinux: policy capability nnp_nosuid_transition=1
фев 12 13:50:43 vbox kernel: SELinux: policy capability genfs_seclabel_symlinks
=1
фев 12 13:50:45 vbox dbus-broker-launch[800]: avc: op=load_policy lsm=selinux s
eqno=2 res=1
фев 12 13:50:45 vbox dbus-broker-launch[2376]: avc: op=load_policy lsm=selinux
seqno=2 res=1

```

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

```

```

# semanage port -a -t ssh_port_t -p tcp 2022
# 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.

```

фев 12 13:49:50 vbox sshd[5673]: Server listening on :: port 22.
[root@vbox ~]# systemctl restart sshd
[root@vbox ~]# systemctl status -l sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: ena
   Active: active (running) since Thu 2026-02-12 13:52:14 MSK; 9s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 5736 (sshd)
      Tasks: 1 (limit: 10513)
    Memory: 1.8M (peak: 2.0M)
       CPU: 26ms
    CGroup: /system.slice/ssh.service
            └─5736 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

фев 12 13:52:14 vbox systemd[1]: Starting OpenSSH server daemon...
фев 12 13:52:14 vbox sshd[5736]: Server listening on 0.0.0.0 port 2022.
фев 12 13:52:14 vbox sshd[5736]: Server listening on :: port 2022.
фев 12 13:52:14 vbox sshd[5736]: Server listening on 0.0.0.0 port 22.
фев 12 13:52:14 vbox sshd[5736]: Server listening on :: port 22.
фев 12 13:52:14 vbox systemd[1]: Started OpenSSH server daemon.
lines 1-18/18 (END)

```

Расширенный статус работы sshd

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

```
[root@vbox ~]# ssh nvsakhno@server.nvsakhno.net
ssh: Could not resolve hostname server.nvsakhno.net: Name or service not known
[root@vbox ~]# ssh -p2022 nvsakhno@server.nvsakhno.net
ssh: Could not resolve hostname server.nvsakhno.net: Name or service not known
[root@vbox ~]#
```

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 nvsakhno@server.nvsakhno.net`

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`

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
AllowUsers vagrant nvsakhno
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

PubkeyAuthentication yes

# The default is to check both .ssh/authorized_keys and .ssh/authorized_k
# but this is overridden so installations will only check .ssh/authorized
AuthorizedKeysFile .ssh/authorized_keys
```

Редактирование файла

После сохранения изменений в файле конфигурации перезапустим `sshd`.

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

```

Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fr2KR7DBpxbg5i7ufLZy+y+uEG0oofQSXFkTsd7UEPU root@vbox
The key's randomart image is:
+---[RSA 3072]-----+
|      =ooo.      |
|    o + o .      |
|   o o + . E     |
| . . . = = .     |
|+. ++ .SB        |
|ooo. o..+ ..     |
|O. ... .. .      |
| ..o.= .o. .      |
| o+*****o..     |
+---[SHA256]-----+
[root@vbox ~]#

```

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 nvsakhno@server.nvsakhno.net`

При запросе введем пароль пользователя на удалённом сервере.

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

```

The key's randomart image
+---[RSA 3072]-----+
|      =ooo.      |
|    o + o .      |
|   o o + . E     |
| . . . = = .     |
|+. ++ .SB        |
|ooo. o..+ ..     |
|O. ... .. .      |
| ..o.= .o. .      |
| o+*****o..     |
+---[SHA256]-----+
[root@vbox ~]#

```

Копирование открытого ssh ключа и получение доступа к серверу

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

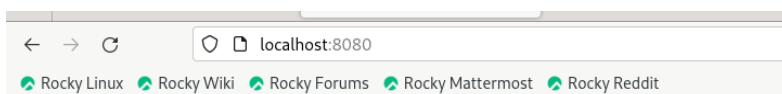
3.5 Организация туннелей SSH, перенаправление TCP-портов

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

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

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

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



localhost:8080

3.6 Запуск консольных приложений через SSH

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

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

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

A screenshot of a terminal window showing the output of the 'ss -tln' command. The output is a table with columns for PID, PPID, STATE, LISTEN, TCP, and PID. The table lists various services and their listening ports.

PID	PPID	STATE	LISTEN	TCP	PID
dovecot	4762	0t0	TCP *:pop3 (LISTEN)	root	23u
dovecot	4762	0t0	TCP *:pop3s (LISTEN)	root	24u
dovecot	4762	0t0	TCP *:pop3s (LISTEN)	root	40u
dovecot	4762	0t0	TCP *:imap (LISTEN)	root	41u
dovecot	4762	0t0	TCP *:imap (LISTEN)	root	42u
dovecot	4762	0t0	TCP *:imaps (LISTEN)	root	43u
dovecot	4762	0t0	TCP *:imaps (LISTEN)	root	13u
master	5049	0t0	TCP *:smtp (LISTEN)	root	3u
sshd	5782	0t0	TCP *:down (LISTEN)	root	4u
sshd	5782	0t0	TCP *:down (LISTEN)	root	5u
sshd	5782	0t0	TCP *:ssh (LISTEN)	root	6u
sshd	5782	0t0	TCP *:ssh (LISTEN)	root	6u

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`

```
#X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
```

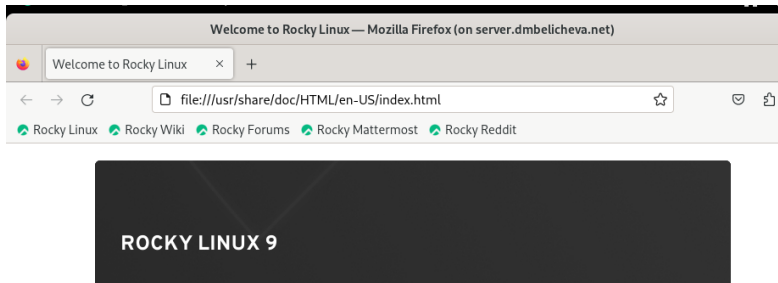
Редактирование файла

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.nvsakhno.net firefox`

```
ssh -YC dmbelicheva@server.nvsakhno.net firefox
Xauthority does not exist
[[0][GFX1-]: glxtest: ManageChildProcess failed
ChildProcess failed

[[0][GFX1-]: glxtest: ManageChildProcess failed
or, error_code=1, request_code=154, minor_code=1 (t=4.72659) [GFX1-]: glxtest:
4, minor_code=1
```

Запуск графических приложений через SSH



Welcome to Rocky Linux

A community Enterprise Operating System

Результат запуска графического приложения через SSH

3.8 Внесение изменений в настройки внутреннего окружения виртуальной машины

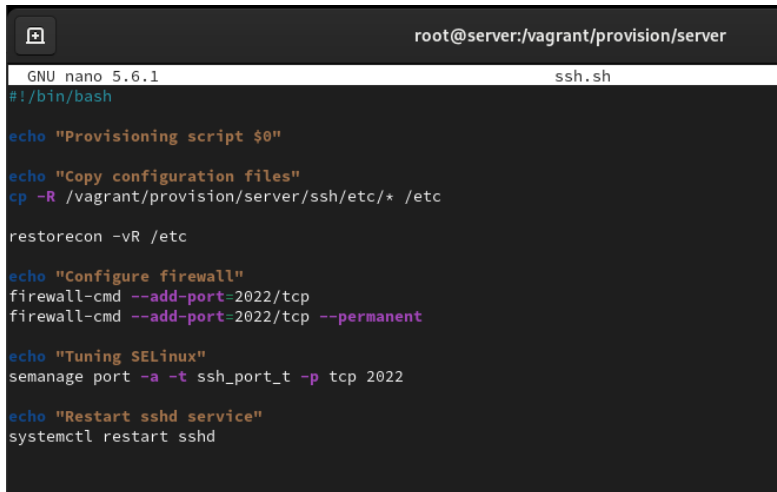
На виртуальной машине `server` перейдем в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создадим в нём каталог `ssh`, в который поместим в соответствующие подкаталоги конфигурационный файл `sshd_config`:

```
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:

A screenshot of a terminal window titled 'root@server:/vagrant/provision/server'. The terminal shows the GNU nano 5.6.1 editor editing the file 'ssh.sh'. The script content is as follows:

```
#!/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
```

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