

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра теории вероятностей и кибербезопасности

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №11

дисциплина: Администрирование сетевых подсистем

Студент: Пакавира Арсениу Висенте Луиш

Студ. билет № 1032225105

Группа: НФИбд-02-23

МОСКВА

2025 г.

Цель работы:

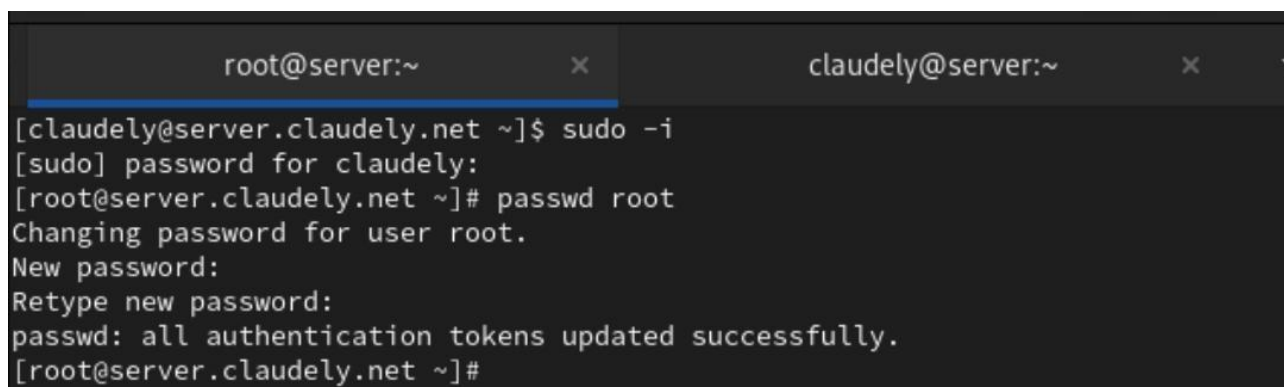
Целью данной работы является приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

Выполнение работы:

На сервере зададим пароль для пользователя root (Рис. 1.1):

```
ssudo -i
```

```
passwd root
```

The image shows a terminal window with two tabs. The active tab is titled 'root@server:~'. The terminal content shows a user 'claudely' at 'server.claudely.net' running 'sudo -i'. This prompts for a password for 'claudely'. After successful authentication, the prompt changes to root level. Then, the command 'passwd root' is entered, which prompts for a new password for the 'root' user, followed by a retype confirmation. The final message is 'passwd: all authentication tokens updated successfully.' and the prompt returns to root level.

```
root@server:~ x claudely@server:~ x
[claudely@server.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@server.claudely.net ~]# passwd root
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server.claudely.net ~]#
```

Рис. 1.1. Открытие режима суперпользователя на виртуальной машине server и создание пароля для пользователя root.

На сервере в дополнительном терминале запустим мониторинг системных событий (Рис. 1.2):

```
sudo -i journalctl
```

```
-x -f
```

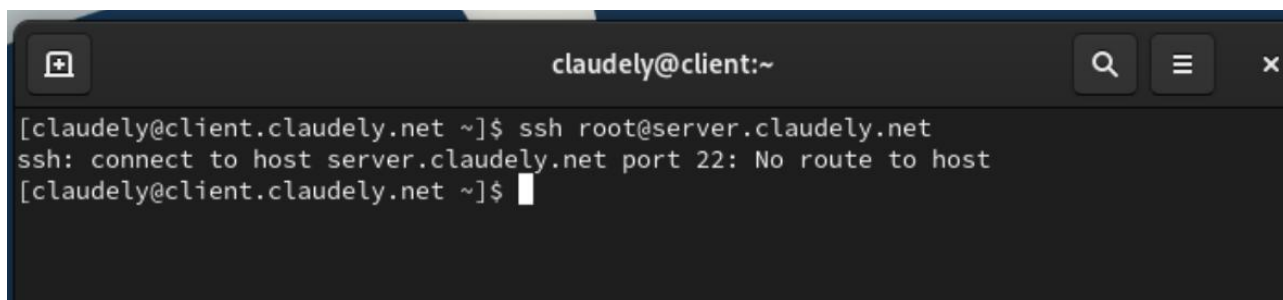
The image shows a terminal window with a dark theme. At the top, there's a title bar with 'root@server:~' and standard window controls. Below the title bar, there are two tabs, both labeled 'root@server:~'. The main content area displays a series of system logs and user actions. It starts with a prompt '[claudely@server.claudely.net ~]\$ sudo -i', followed by a password prompt '[sudo] password for claudely:'. Then, a root prompt '[root@server.claudely.net ~]# journalctl -x -f' is shown. The subsequent lines are system logs: a network unreachable message, a password change for 'root' (highlighted in green), a failed password update for 'gkr-pam' (highlighted in yellow), and a successful sudo session for 'claudely' (highlighted in green). The logs continue with systemd-journald messages about journal rotation and rsyslogd messages about journal file changes and hostname service startup. The terminal text is as follows:

```
[claudely@server.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@server.claudely.net ~]# journalctl -x -f
Dec 22 08:33:29 server.claudely.net named[852]: network unreachable resolving 'content-
signature-chains.prod.autograph.services.mozaws.net/AAAA/IN': 2600:9000:5306:9900::1#53
Dec 22 08:33:29 server.claudely.net passwd[10337]: pam_unix(passwd:chauthtok): password
changed for root
Dec 22 08:33:29 server.claudely.net passwd[10337]: gkr-pam: couldn't update the login k
eyring password: no old password was entered
Dec 22 08:33:52 server.claudely.net sudo[10384]: claudely : TTY=pts/1 ; PWD=/root ; USE
R=root ; COMMAND=/bin/bash
Dec 22 08:33:52 server.claudely.net sudo[10384]: pam_unix(sudo-i:session): session open
ed for user root(uid=0) by claudely(uid=1002)
Dec 22 08:33:52 server.claudely.net systemd-journald[454]: Data hash table of /run/log/
journal/bb344eeb9a684c4bafal2b93fea4dac6/system.journal has a fill level at 75.1 (2632
of 3505 items, 2019328 file size, 767 bytes per hash table item), suggesting rotation.
Dec 22 08:33:52 server.claudely.net systemd-journald[454]: /run/log/journal/bb344eeb9a6
84c4bafal2b93fea4dac6/system.journal: Journal header limits reached or header out-of-da
te, rotating.
Dec 22 08:33:52 server.claudely.net rsyslogd[1322]: imjournal: journal files changed, r
eloading... [v8.2310.0-4.el9 try https://www.rsyslog.com/e/0 ]
Dec 22 08:33:52 server.claudely.net systemd[1]: Starting Hostname Service...
Subject: A start job for unit systemd-hostnamed.service has begun execution
Defined-By: systemd
Support: https://wiki.rockylinux.org/rocky/support
```

Рис. 1.2. Запуск в дополнительном терминале мониторинга системных событий.

С клиента попытаемся получить доступ к серверу посредством SSHсоединения через пользователя root (Рис. 1.3):

```
ssh root@server.claudely.net
```

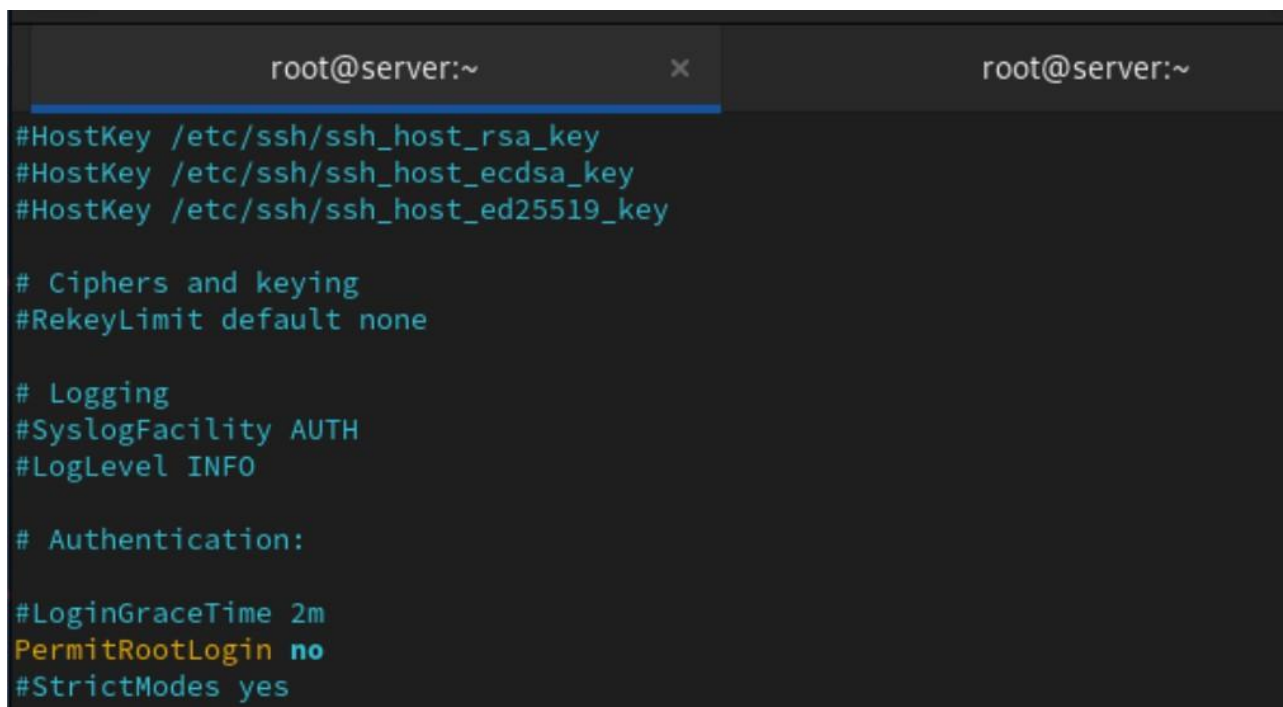


```
claudely@client:~  
[claudely@client.claudely.net ~]$ ssh root@server.claudely.net  
ssh: connect to host server.claudely.net port 22: No route to host  
[claudely@client.claudely.net ~]$
```

Рис. 1.3. Попытка получить с клиента доступ к серверу посредством SSHсоединения через пользователя root.

На сервере откроем файл `/etc/ssh/sshd_config` конфигурации `sshd` для редактирования и запретим вход на сервер пользователю `root`, установив (Рис. 1.4):

`PermitRootLogin no`



```
root@server:~  
#HostKey /etc/ssh/ssh_host_rsa_key  
#HostKey /etc/ssh/ssh_host_ecdsa_key  
#HostKey /etc/ssh/ssh_host_ed25519_key  
  
# Ciphers and keying  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
#LogLevel INFO  
  
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin no  
#StrictModes yes
```

Рис. 1.4. Открытие на сервере файла `/etc/ssh/sshd_config` конфигурации `sshd` для редактирования и запрет входа на сервер пользователю `root`.

После сохранения изменений в файле конфигурации перезапустим `sshd`

(Рис. 1.5):

```
systemctl restart sshd
```



```
[root@server.claudely.net ~]# nano /etc/ssh/sshd_config  
[root@server.claudely.net ~]# systemctl restart sshd  
[root@server.claudely.net ~]#
```

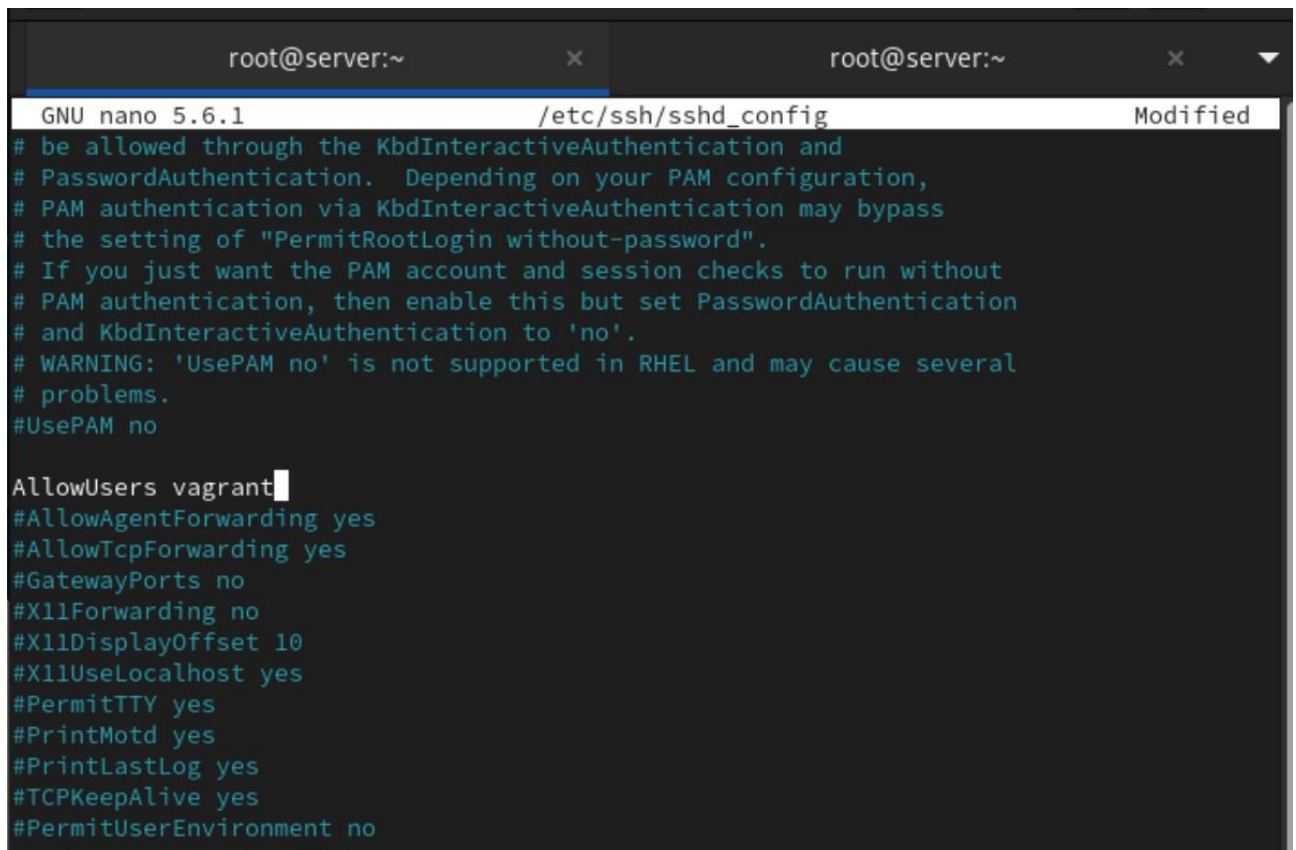
Рис. 1.5. Перезапуск sshd.

С клиента попытаемся получить доступ к серверу посредством SSHсоединения через пользователя claudely (Рис. 2.1):

```
ssh claudely@server.claudely.net
```

На сервере откроем файл /etc/ssh/sshd_config конфигурации sshd на редактирование и добавим строку (Рис. 2.2):

```
AllowUsers vagrant
```



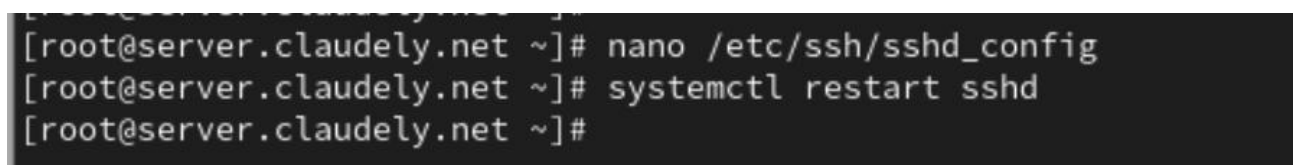
```
root@server:~ x root@server:~ x
GNU nano 5.6.1 /etc/ssh/sshd_config Modified
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in RHEL and may cause several
# problems.
#UsePAM no

AllowUsers vagrant
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
```

Рис. 2.2. Открытие на сервере файла /etc/ssh/sshd_config конфигурации sshd на редактирование и добавление нужной строки.

После сохранения изменений в файле конфигурации перезапустим sshd (Рис. 2.3):

```
systemctl restart sshd
```



```
[root@server.claudely.net ~]# nano /etc/ssh/sshd_config
[root@server.claudely.net ~]# systemctl restart sshd
[root@server.claudely.net ~]#
```

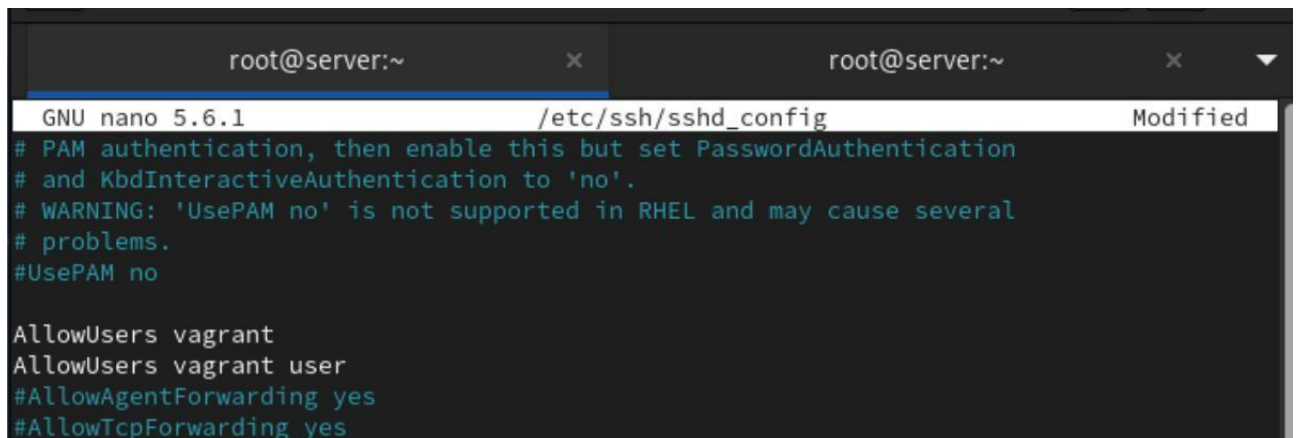
Рис. 2.3. Перезапуск sshd.

Повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя claudely (Рис. 2.4):

```
ssh claudely@server.claudely.net
```

В файле `/etc/ssh/sshd_config` конфигурации `sshd` внесём следующее изменение (Рис. 2.5):

`AllowUsers vagrant claudely`



```
root@server:~ x root@server:~ x
GNU nano 5.6.1 /etc/ssh/sshd_config Modified
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in RHEL and may cause several
# problems.
#UsePAM no

AllowUsers vagrant
AllowUsers vagrant user
#AllowAgentForwarding yes
#AllowTcpForwarding yes
```

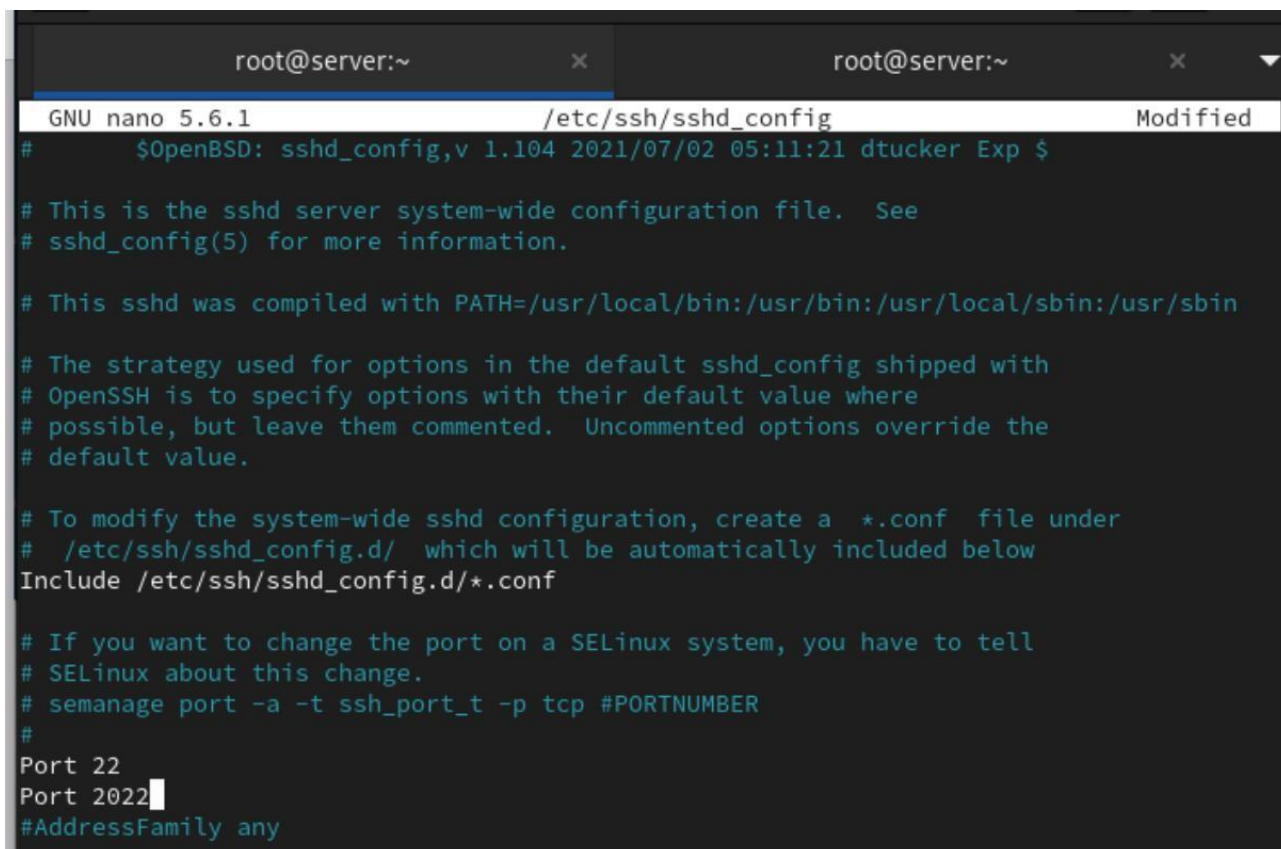
Рис. 2.5. Внесение изменения в файле `/etc/ssh/sshd_config` конфигурации `sshd`.

После сохранения изменений в файле конфигурации перезапустим `sshd` и вновь попытаемся получить доступ с клиента к серверу посредством SSHсоединения через пользователя `claudely` (Рис. 2.6):

На сервере в файле конфигурации `sshd` `/etc/ssh/sshd_config` найдём строку `Port` и ниже этой строки добавим (Рис. 3.1):

`Port 22`

`Port 2022`



```
root@server:~ x root@server:~ x
GNU nano 5.6.1 /etc/ssh/sshd_config Modified
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

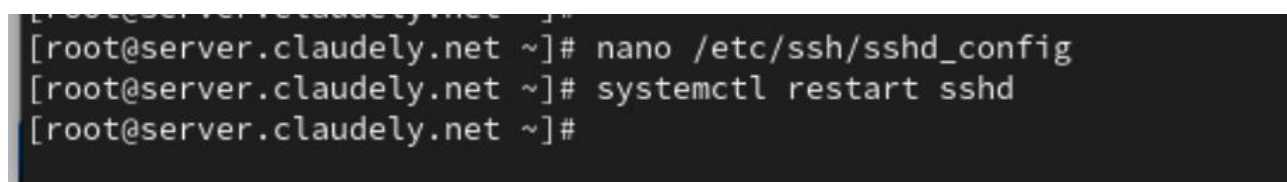
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# 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
```

Рис. 3.1. Добавление ниже строки Port записей в файле конфигурации sshd /etc/ssh/sshd_config на сервере.

После сохранения изменений в файле конфигурации перезапустим sshd:

systemctl restart sshd



```
root@server.claudely.net ~]# nano /etc/ssh/sshd_config
root@server.claudely.net ~]# systemctl restart sshd
root@server.claudely.net ~]#
```

И посмотрим расширенный статус работы:

systemctl status -l sshd

Система сообщила нам об отказе в работе sshd через порт 2022 (Рис. 3.2):

Дополнительно посмотрим сообщения в терминале с мониторингом системных событий (Рис. 3.3):

```
root@server:~ x root@server:~ x claudely@server:~ — ... x
[claudely@server.claudely.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 2024-12-22 09:00:08 UTC; 21s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 10630 (sshd)
     Tasks: 1 (limit: 4553)
    Memory: 860.0K
       CPU: 12ms
    CGroup: /system.slice/sshd.service
           └─10630 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 22 09:00:08 server.claudely.net systemd[1]: Starting OpenSSH server daemon...
Dec 22 09:00:08 server.claudely.net sshd[10630]: error: Bind to port 2022 on 0.0.0.0 f>
Dec 22 09:00:08 server.claudely.net sshd[10630]: error: Bind to port 2022 on :: failed>
Dec 22 09:00:08 server.claudely.net sshd[10630]: Server listening on 0.0.0.0 port 22.
Dec 22 09:00:08 server.claudely.net sshd[10630]: Server listening on :: port 22.
Dec 22 09:00:08 server.claudely.net systemd[1]: Started OpenSSH server daemon.
lines 1-18/18 (END)
```

Рис. 3.2. Перезапуск sshd и просмотр расширенного статуса работы.

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
```

Вновь перезапустим sshd и посмотрим расширенный статус его работы (статус показывает, что процесс sshd теперь прослушивает два порта) (Рис. 3.4):

```
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# semanage port -a -t ssh_port_t -p tcp 2022
[root@server.claudely.net ~]#
[root@server.claudely.net ~]# firewall-cmd --add-port=2022/tcp
success
[root@server.claudely.net ~]# firewall-cmd --add-port=2022/tcp --permanent
success
[root@server.claudely.net ~]#
```

Рис. 3.4. Исправление на сервере метки SELinux к порту 2022, открытие в настройках межсетевого порта 2022 протокола TCP, повторный перезапуск sshd и просмотр расширенного статуса его работы.

Теперь повторим попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя claudely, указав порт 2022:

```
ssh -p2022 claudely@server.claudely.net
```

После открытия оболочки пользователя введём `sudo -i` для получения доступа root (рис. 3.6):

На сервере в конфигурационном файле `/etc/ssh/sshd_config` зададим параметр, разрешающий аутентификацию по ключу (рис. 4.1):

```
PubkeyAuthentication yes
```

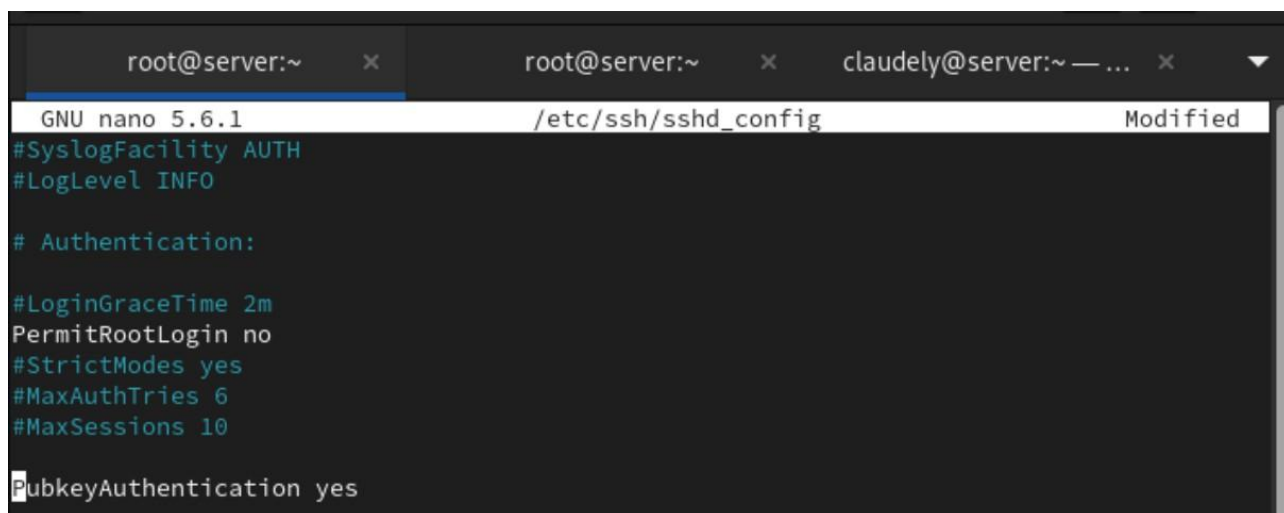


Рис. 4.1. Настройка параметра на сервере в конфигурационном файле `/etc/ssh/sshd_config`, разрешающего аутентификацию по ключу.

После сохранения изменений в файле конфигурации перезапустим sshd (рис. 4.2):

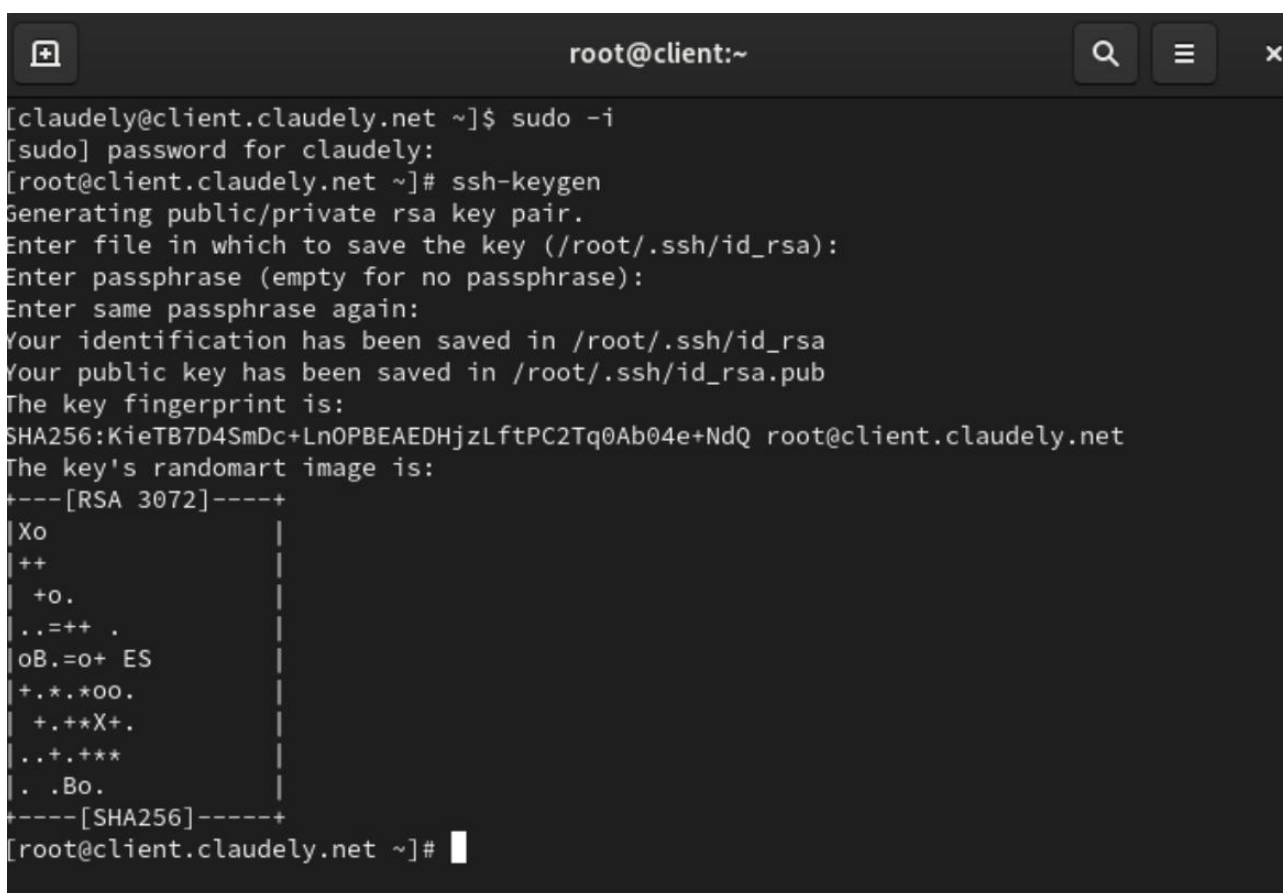
```
[root@server.claudely.net ~]# nano /etc/ssh/sshd_config
[root@server.claudely.net ~]# systemctl restart sshd
[root@server.claudely.net ~]#
```

Рис. 4.2. Перезапуск sshd.

На клиенте сформируем SSH-ключ, введя в терминале под пользователем claudely `ssh-keygen`

Далее скопируем открытый ключ на сервер, введя на клиенте (рис. 4.3):

`ssh-copy-id claudely@server.claudely.net`



```
root@client:~
[claudely@client.claudely.net ~]$ sudo -i
[sudo] password for claudely:
[root@client.claudely.net ~]# ssh-keygen
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:KieTB7D4SmDc+Ln0PBEAEDHjzLftPC2Tq0Ab04e+NdQ root@client.claudely.net
The key's randomart image is:
+---[RSA 3072]-----+
|Xo                    |
|++                   |
|+o.                  |
|..=++ .              |
|oB.=o+ ES            |
|+.*.*oo.             |
|+.+*X+.              |
|..+.***              |
|. .Bo.               |
+---[SHA256]-----+
[root@client.claudely.net ~]#
```

Рис. 4.3. Формирование на клиенте SSH-ключа и копирование открытого ключа на сервер.

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

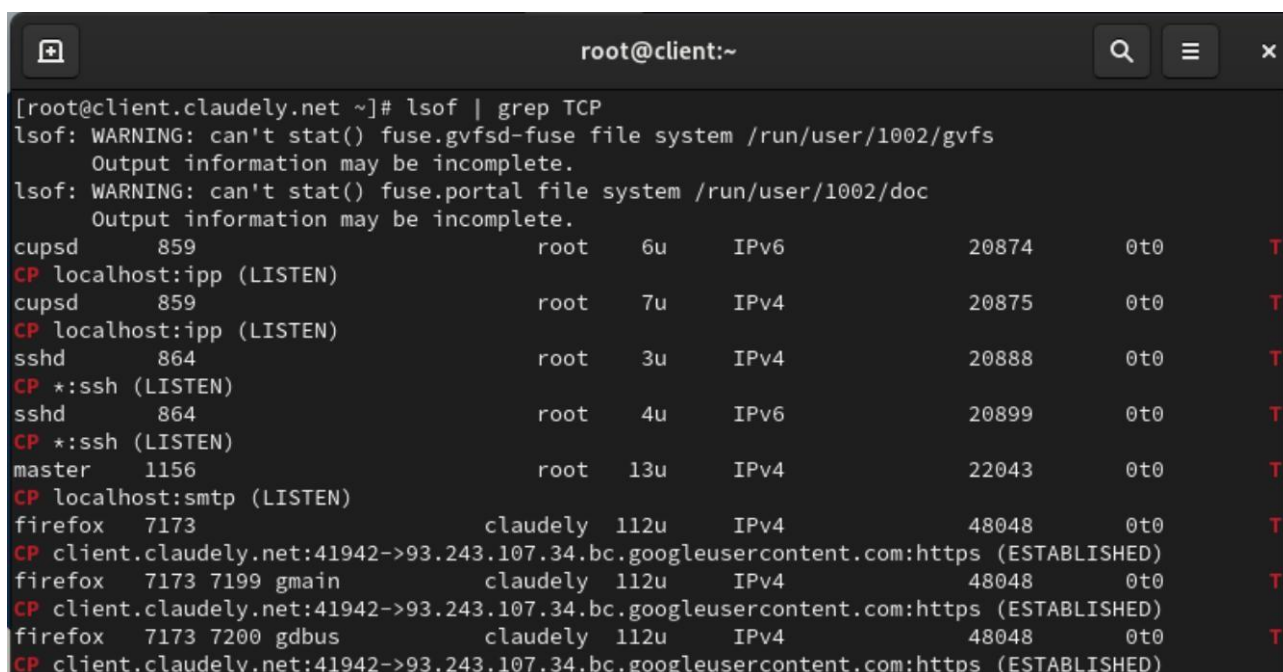
```
lsof | grep TCP
```

После чего перенаправим порт 80 на server.claudely.net на порт 8080 на локальной машине (рис. 5.1):

```
ssh -fNL 8080:localhost:80 claudely@server.claudely.net
```

Вновь на клиенте посмотрим, запущены ли какие-то службы с протоколом TCP (рис. 5.2):

```
lsof | grep TCP
```



```
root@client:~  
[root@client.claudely.net ~]# lsof | grep TCP  
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1002/gvfs  
Output information may be incomplete.  
lsof: WARNING: can't stat() fuse.portal file system /run/user/1002/doc  
Output information may be incomplete.  
cupsd      859          root      6u        IPv6      20874      0t0      T  
CP localhost:ipp (LISTEN)  
cupsd      859          root      7u        IPv4      20875      0t0      T  
CP localhost:ipp (LISTEN)  
sshd       864          root      3u        IPv4      20888      0t0      T  
CP *:ssh (LISTEN)  
sshd       864          root      4u        IPv6      20899      0t0      T  
CP *:ssh (LISTEN)  
master    1156         root     13u        IPv4     22043      0t0      T  
CP localhost:smtp (LISTEN)  
firefox    7173        claudely 112u       IPv4     48048      0t0      T  
CP client.claudely.net:41942->93.243.107.34.bc.googleusercontent.com:https (ESTABLISHED)  
firefox    7173 7199 gmain    claudely 112u       IPv4     48048      0t0      T  
CP client.claudely.net:41942->93.243.107.34.bc.googleusercontent.com:https (ESTABLISHED)  
firefox    7173 7200 gdbus    claudely 112u       IPv4     48048      0t0      T  
CP client.claudely.net:41942->93.243.107.34.bc.googleusercontent.com:https (ESTABLISHED)
```

Рис. 5.2. Повторный просмотр на клиенте запущенных служб с протоколом TCP.

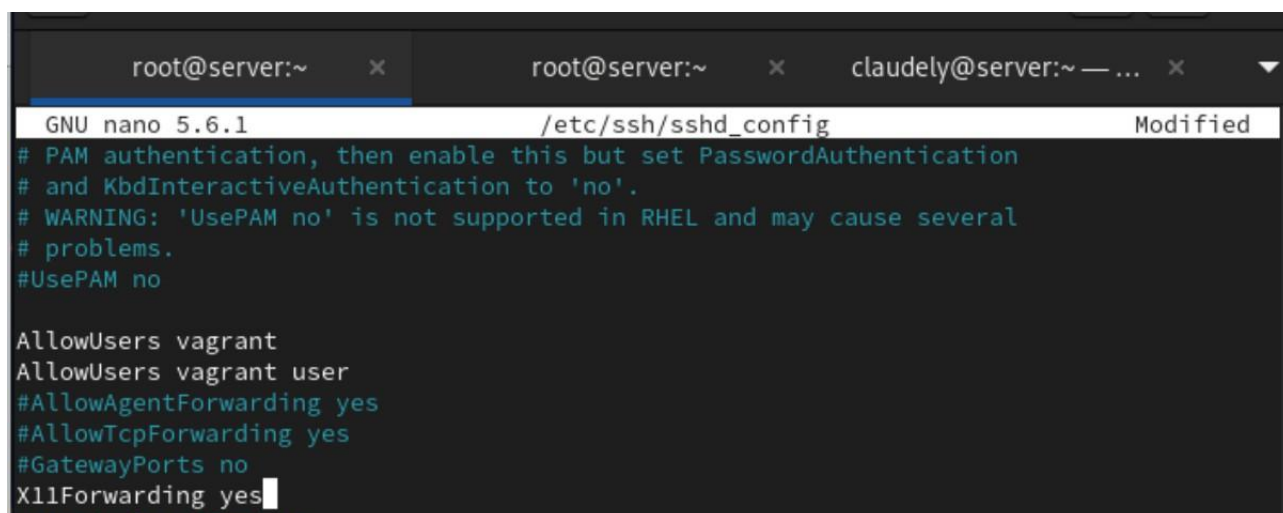
Посмотрим с клиента почту на сервере (рис. 6):

```
ssh claudely@server.claudely.net MAIL=~/.Maildir/ mail
```

На сервере в конфигурационном файле /etc/ssh/sshd_config разрешим отображать на локальном клиентском компьютере графические интерфейсы X11

(рис. 7.1):

X11Forwarding yes

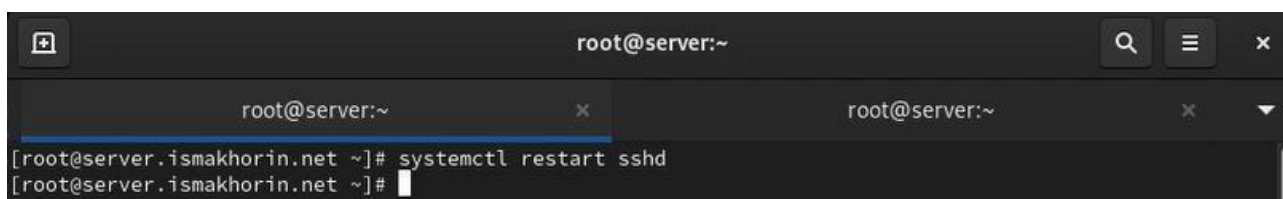


```
GNU nano 5.6.1 /etc/ssh/sshd_config Modified
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
# WARNING: 'UsePAM no' is not supported in RHEL and may cause several
# problems.
#UsePAM no

AllowUsers vagrant
AllowUsers vagrant user
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
```

Рис. 7.1. Разрешение отображать на сервере в конфигурационном файле `/etc/ssh/sshd_config` на локальном клиентском компьютере графические интерфейсы X11.

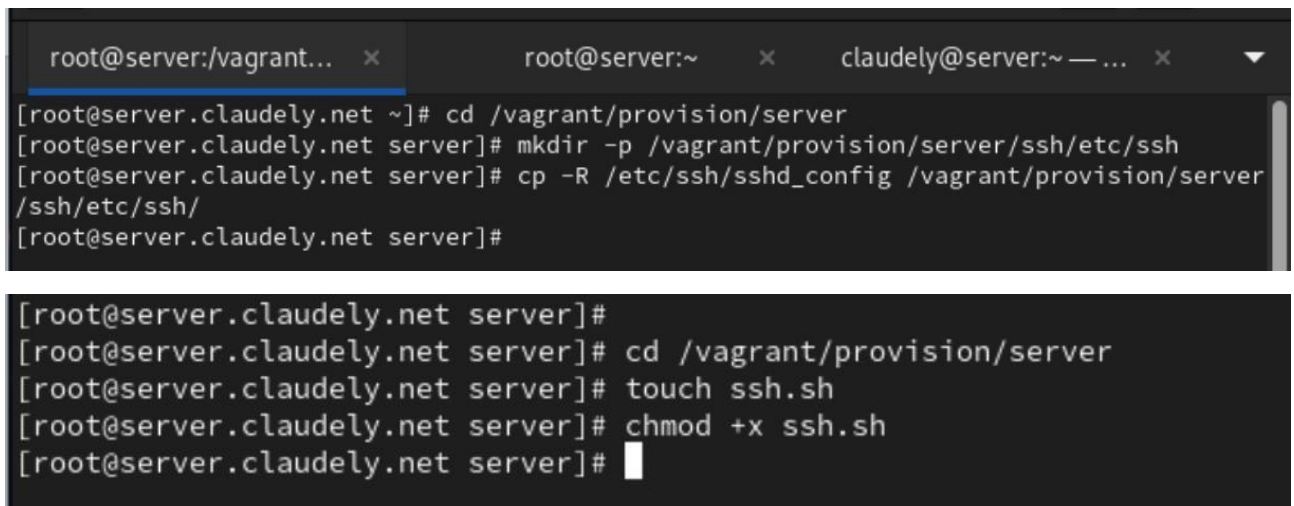
После сохранения изменения в конфигурационном файле перезапустим `sshd` (рис. 7.2):



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

Рис. 7.2. Перезапуск `sshd`.

На виртуальной машине `server` перейдём 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`. В каталоге `/vagrant/provision/server` создадим исполняемый файл `ssh.sh` (рис. 8.1):

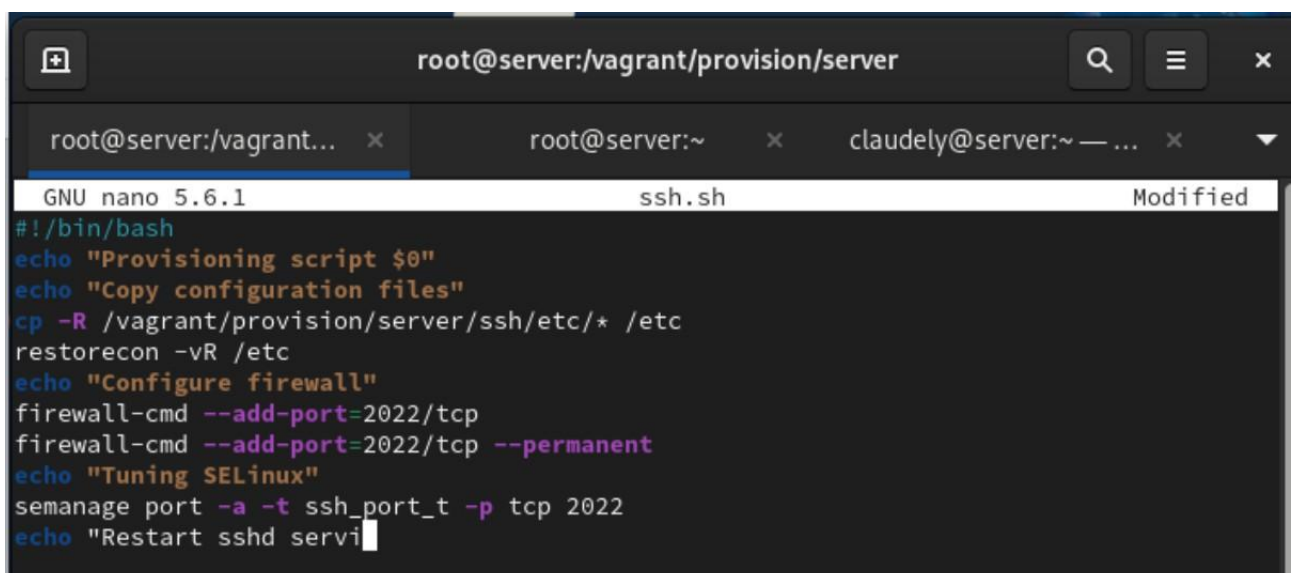


```
root@server:/vagrant... x root@server:~ x claudely@server:~ — ... x
[root@server.claudely.net ~]# cd /vagrant/provision/server
[root@server.claudely.net server]# mkdir -p /vagrant/provision/server/ssh/etc/ssh
[root@server.claudely.net server]# cp -R /etc/ssh/sshd_config /vagrant/provision/server/ssh/etc/ssh/
[root@server.claudely.net server]#

[root@server.claudely.net server]#
[root@server.claudely.net server]# cd /vagrant/provision/server
[root@server.claudely.net server]# touch ssh.sh
[root@server.claudely.net server]# chmod +x ssh.sh
[root@server.claudely.net server]#
```

Рис. 8.1. Переход на виртуальной машине server в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создание в нём каталога `ssh`, в который поместили в соответствующие подкаталоги конфигурационный файл `sshd_config`. Создание в каталоге `/vagrant/provision/server` исполняемого файла `ssh.sh`.

Открыв его на редактирование, пропишем в нём скрипт из лабораторной работы (Рис. 8.2):



```
root@server:/vagrant/provision/server
GNU nano 5.6.1 ssh.sh Modified
#!/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 servi
```

Рис. 8.2. Открытие файла на редактирование и написание в нём скрипта.

Для отработки созданного скрипта во время загрузки виртуальной машины server в конфигурационном файле Vagrantfile добавим в разделе конфигурации для сервера (рис. 8.3):

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

server.vm.provider :virtualbox do |v|
```

Рис. 8.3. Редактирование конфигурационного файла Vagrantfile.

Вывод:

В ходе выполнения лабораторной работы были приобретены практические навыки по настройке удалённого доступа к серверу с помощью SSH.

Ответы на контрольные вопросы:

1. Вы хотите запретить удалённый доступ по SSH на сервер пользователю root и разрешить доступ пользователю alice. Как это сделать? – **В конфигурационном файле SSH /etc/ssh/sshd_config:**

Запрет удалённого доступа пользователю root

PermitRootLogin no

Разрешение доступа пользователю alice

AllowUsers alice

После внесения изменений, необходимо перезапустить службу SSH:

sudo service ssh restart

2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться? –

В конфигурационном файле /etc/ssh/sshd_config добавьте строки:

Первый порт (по умолчанию 22)

Port 22

Второй порт

Port 2222

После изменений перезапустите службу SSH. Это может быть полезно для повышения безопасности, а также для избежания конфликтов с другими службами, использующими порт 22.

3. Какие параметры используются для создания туннеля SSH, когда команда ssh устанавливает фоновое соединение и не ожидает какойлибо конкретной команды? - **ssh -N -f -L**

local_port:destination_host:remote_port user@ssh_server

-N: Не выполнять команду на удаленном хосте.

-f: Перевести ssh в фоновый режим после установки туннеля.

4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com? – **ssh -L 5555:server2.example.com:80 user@ssh_server**

Теперь, при подключении к локальному порту 5555, трафик будет перенаправляться через SSH к порту 80 на сервере server2.example.com.

5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022? – **sudo semanage port -a -t ssh_port_t -p tcp 2022**

Данная команда добавляет правило SELinux, разрешая использование порта 2022 для сервиса ssh.

6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022? – **sudo firewall-cmd --permanent --add-port=2022/tcp**
sudo firewall-cmd --reload

Эти команды добавляют правило в межсетевой экран для разрешения входящих подключений по SSH через порт 2022 и перезагружают конфигурацию межсетевого экрана.