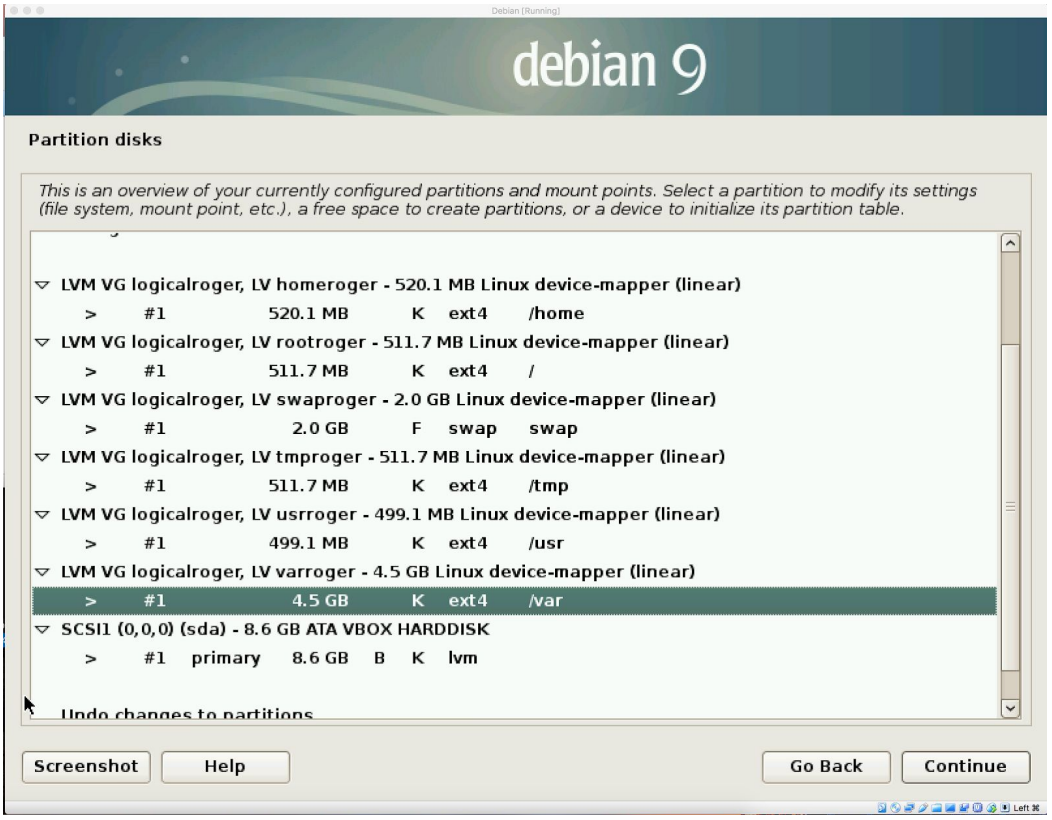
Статьи

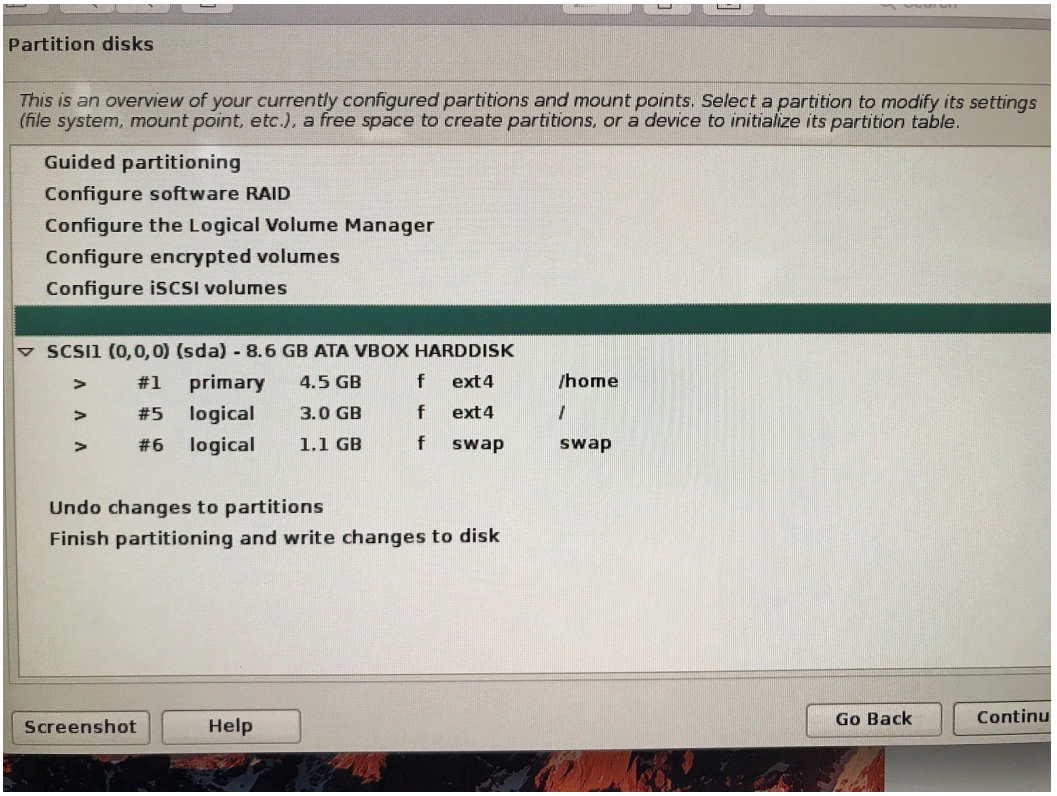
1) Базовая настройка Debian - https://serveradmin.ru/debian-nastroyka-servera/
2) Iptables: https://serveradmin.ru/nastroyka-iptables-v-centos-7/
3) Partition: https://www.ibm.com/developerworks/ru/library/l-lpic1-v3-102-1/
4) Маска подсети: https://www.networkcenter.info/inform/netmask
5) Сайт с расчетом масок и бродкаста: http://jodies.de/ipcalc?host=194.236.125.221&mask1=&mask2=30
6) Полная установка web-сервера: https://d1mon.com/n/1404
7) Ssh: https://putty.org.ru/articles/change-default-sshd-port.html
8) Ssh: https://habr.com/ru/post/331348/
9) Ssh connection: https://mordeniuss.ru/enable-ssh-debian/
10) Компьютерные сети - https://proglib.io/p/computer-networking
11) Автозагрузка демонов: https://habr.com/ru/post/141003/
12) Корневые серверы DNS: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BD%D0%B5%D0%B2%D1%8B%D0%B5_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%8B_DNS
13) Все про DNS: http://webcache.googleusercontent.com/search?q=cache:http://www.bog.pp.ru/work/dns_info.html
14) Хосты: https://ktonanovenkogo.ru/vokrug-da-okolo/hosting/fajl-hosts-gde-on-naxoditsya-v-windows-chto-delat-kak-udalit-virus.html
15) Полная инструкция по настройке сетей в Debian: http://www.aitishnik.ru/linux02.html

Установка виртуалки

Image:

Настройки по дебиану:


Debian [Running]
debian 9
Partition disks
This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.
LVM VG logicalroger, LV homeroger - 520.1 MB Linux device-mapper (linear)
> #1 520.1 MB K ext4 /home
LVM VG logicalroger, LV rootroger - 511.7 MB Linux device-mapper (linear)
> #1 511.7 MB K ext4 /
LVM VG logicalroger, LV swaproger - 2.0 GB Linux device-mapper (linear)
> #1 2.0 GB F swap swap
LVM VG logicalroger, LV tmproger - 511.7 MB Linux device-mapper (linear)
> #1 511.7 MB K ext4 /tmp
LVM VG logicalroger, LV usrroger - 499.1 MB Linux device-mapper (linear)
> #1 499.1 MB K ext4 /usr
LVM VG logicalroger, LV varroger - 4.5 GB Linux device-mapper (linear)
> #1 4.5 GB K ext4 /var
SCSI1 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
> #1 primary 8.6 GB B K lvm
Undo changes to partitions
Screenshot
Help
Go Back
Continue


Partition disks
This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.
Guided partitioning
Configure software RAID
Configure the Logical Volume Manager
Configure encrypted volumes
Configure iSCSI volumes
SCSI1 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK
> #1 primary 4.5 GB f ext4 /home
> #5 logical 3.0 GB f ext4 /
> #6 logical 1.1 GB f swap swap
Undo changes to partitions
Finish partitioning and write changes to disk
Screenshot
Help
Go Back
Continu

Создание юзера

Создание sudo-user, проверка разделения дисков, установка нужного, проверка активных сервисов:
Разделение дисков (disk partition) - посмотреть

```
lsblk
```

Sudo для нового юзера:

```
su
apt-get install sudo
sudo adduser sschmele sudo
su sschmele
```

Установка-обновление

```
sudo apt-get install net-tools
sudo apt-get update
```

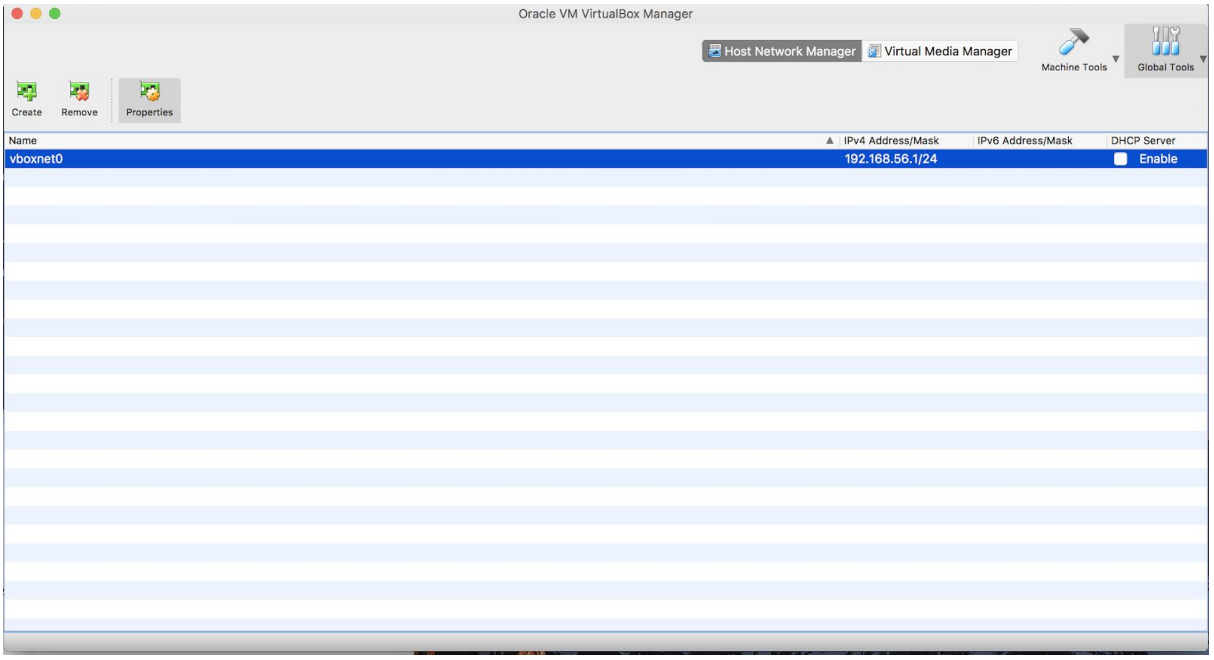
Просмотр статуса сервисов

```
sudo service --status-all
systemctl list-unit-files --type service --state enabled
```

РАБОТА НАД IP

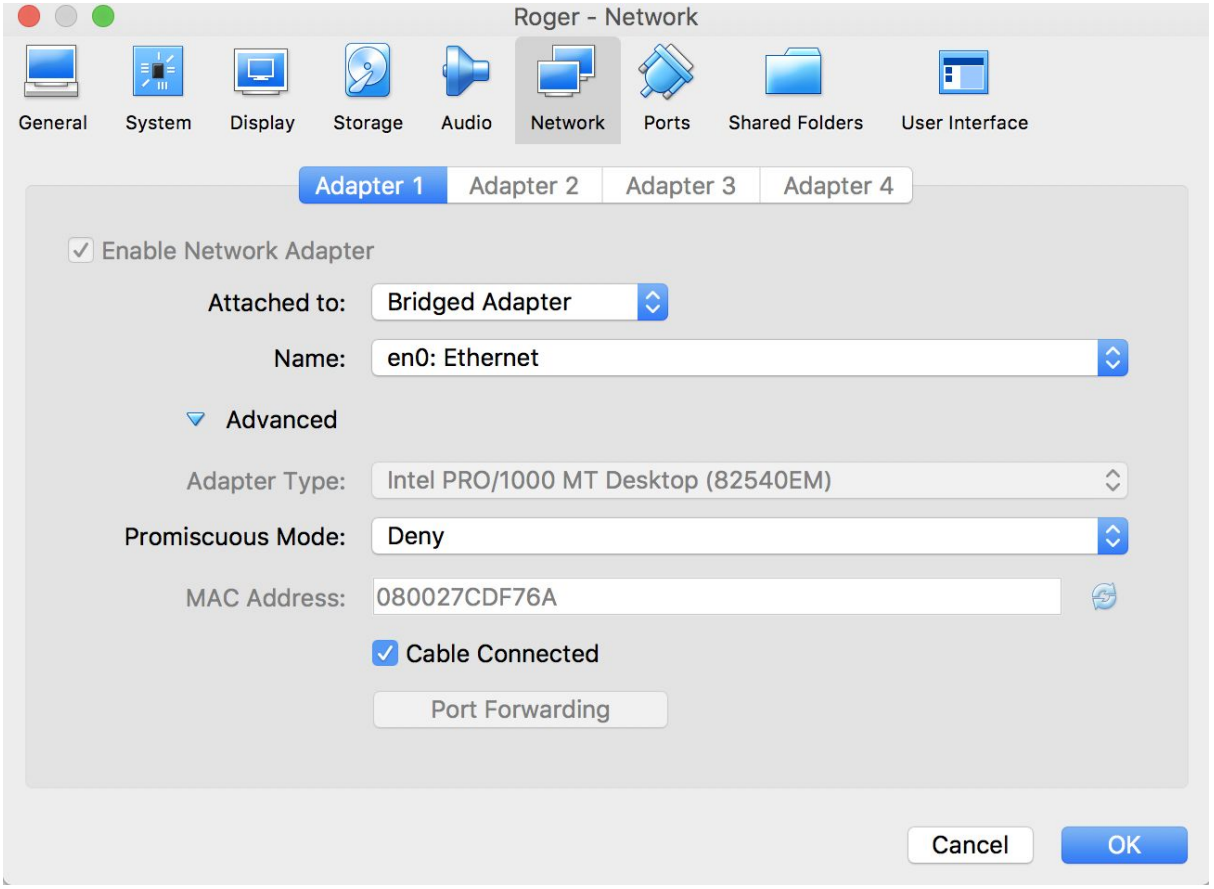
Global tools -> Host Network Manager -> Create

По дефолту появляется vboxnet0. Можно добавить столько адаптеров подобного типа, сколько нужно (если есть необходимость в настройке нескольких отдельных сетей типа "Виртуальный адаптер хоста").



Отключение DHCP Server - галочка в Enable

в конце концов, после NAT и Host-only, я решила делать через bridge:



Через bridge:

Change on bridge with eth0
IP was 192.168.2*.2*8

1) sudo ifdown enp0s3
2) change file /etc/network/interfaces:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
address 192.168.20.249
netmask 255.255.255.252
gateway 192.168.20.1
dns-nameserver 8.8.8.8

auto enp0s3
```

Added static instead of dhcp in :
iface enp0s3 inet dhcp

added address, netmask, gateway, dns-nameserver, auto enp0s3 (Есть вопрос к этому действию, были комментарии, что это не совсем правильно - прописывать gateway). Но без этого не работало.

3) check gateway: netstat -rn:

```
sschmele@roger:~$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.20.1    0.0.0.0         UG        0 0          0 enp0s3
192.168.20.248  0.0.0.0         255.255.255.252 U         0 0          0 enp0s3
```

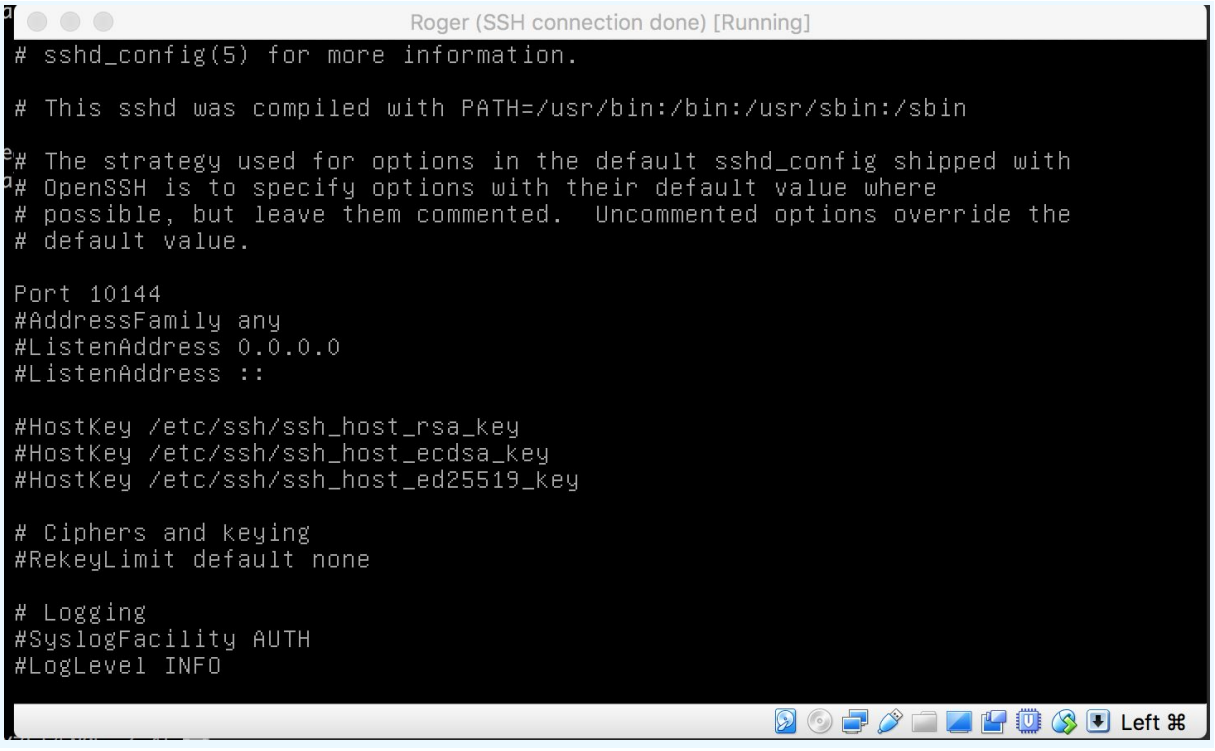
4) check network:
ping 127.0.0.1 -c 2
ping 8.8.8.8 -c 2

SSH work

sudo systemctl status ssh

To check all the services installed: sudo apt list --installed
To update ssh: sudo apt-get install openssh-server

sudo ssh -p [port_nb] [user_name]@[public IP-address]
Default port nb is 22 but we change for some other - по заданию нужно заменить дефолтный порт
Port is stated in /etc/ssh/sshd_config:



Check the ports used - проверить использованные порты (проверяет не все):
Какие-то порты могут быть установлены дефолтными для пакетов, которые будут скачиваться после. Поэтому при необходимости открыть порты, лучше выбирать пятизначные значения.

lsof -i:

```
sschmele@roger:~$ sudo lsof -i
COMMAND PID        USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
exim4   617 Debian-exim    3u  IPv4  12074      0t0  TCP localhost:smtp (LISTEN)
exim4   617 Debian-exim    4u  IPv6  12078      0t0  TCP localhost:smtp (LISTEN)
sshd    759        root    3u  IPv4  13320      0t0  TCP *:10144 (LISTEN)
sshd    759        root    4u  IPv6  13336      0t0  TCP *:10144 (LISTEN)
sschmele@roger:~$
```

Or
netstat -tulpn:

```
sschmele@roger:~$ netstat -tulpn
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
-
tcp        0      0 0.0.0.0:10144           0.0.0.0:*               LISTEN
-
tcp6       0      0 ::1:25                  :::*                    LISTEN
-
tcp6       0      0 :::10144                :::*                    LISTEN
-
```

For ports it is better to use ports with 5 numbers - the highest limit is 65535.

Генерация RSA ключей

Than on your main machine from that you connect to a server (MAC) you generate keys:

sudo ssh-keygen -t rsa (-b 2048)

The default is 2048, the minimum is 1024, "-t" specify the type of key

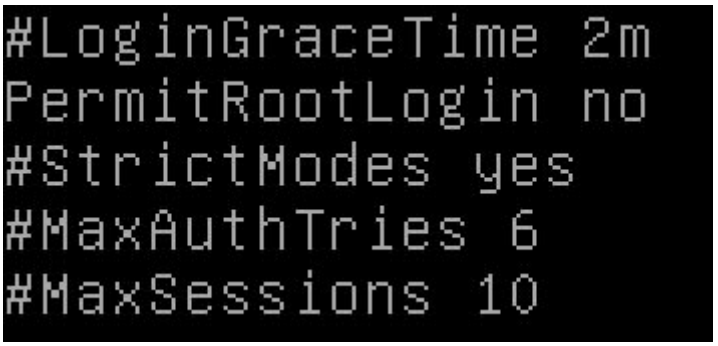
It asks you where to save your keys, default in the file /home/user/.ssh/id_rsa and the public in /home/user/.ssh/id_rsa.pub
Shows you the fingerprint: SHA256: TyQZnNrw8...
And shows the key's randomart image

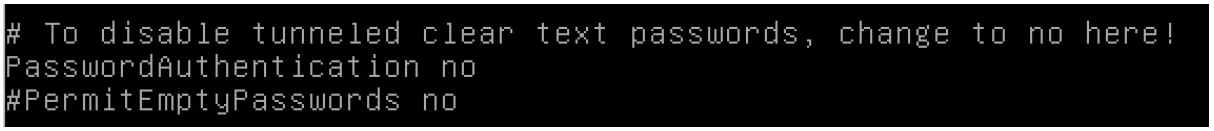
Перенос ключей по SSH

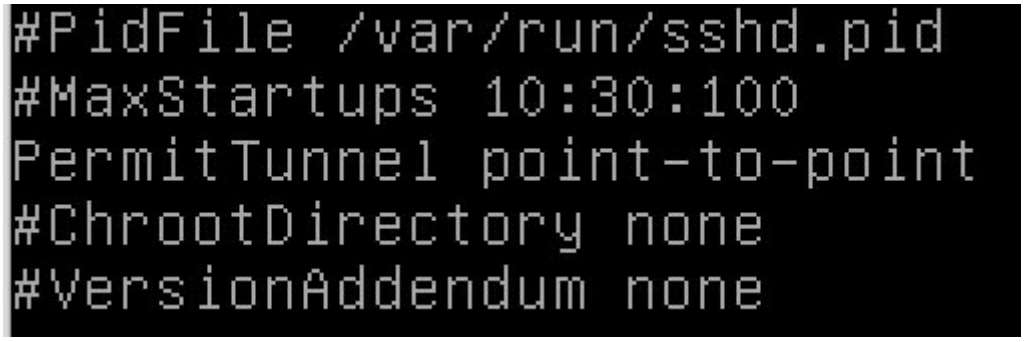
Than connection to the machine with server is needed and the copy of the public key is needed:
ssh-copy-id -i ~/.ssh/id_rsa.pub sschmele@192.168.20.249 -p 10144 where -i - path to the file to copy through ssh

```
oa-j4% ssh-copy-id -i ~/.ssh/id_rsa.pub sschmele@192.168.20.249 -p 10144
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/Users/sschmele/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new key
s
sschmele@192.168.20.249's password:

Number of key(s) added:        1

Now try logging into the machine, with:   "ssh -p '10144' 'sschmele@192.168.20.249'"
and check to make sure that only the key(s) you wanted were added.

oa-j4% ssh sschmele@192.168.20.249 -p 10144
Linux roger 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u3 (2019-06-16) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
```

And then according to the exercise the change in access rights is needed - нужно поменять доступ по заданию Roger (подключение только по ключам, только к юзеру - к root нельзя): /etc/ssh/sshd_config

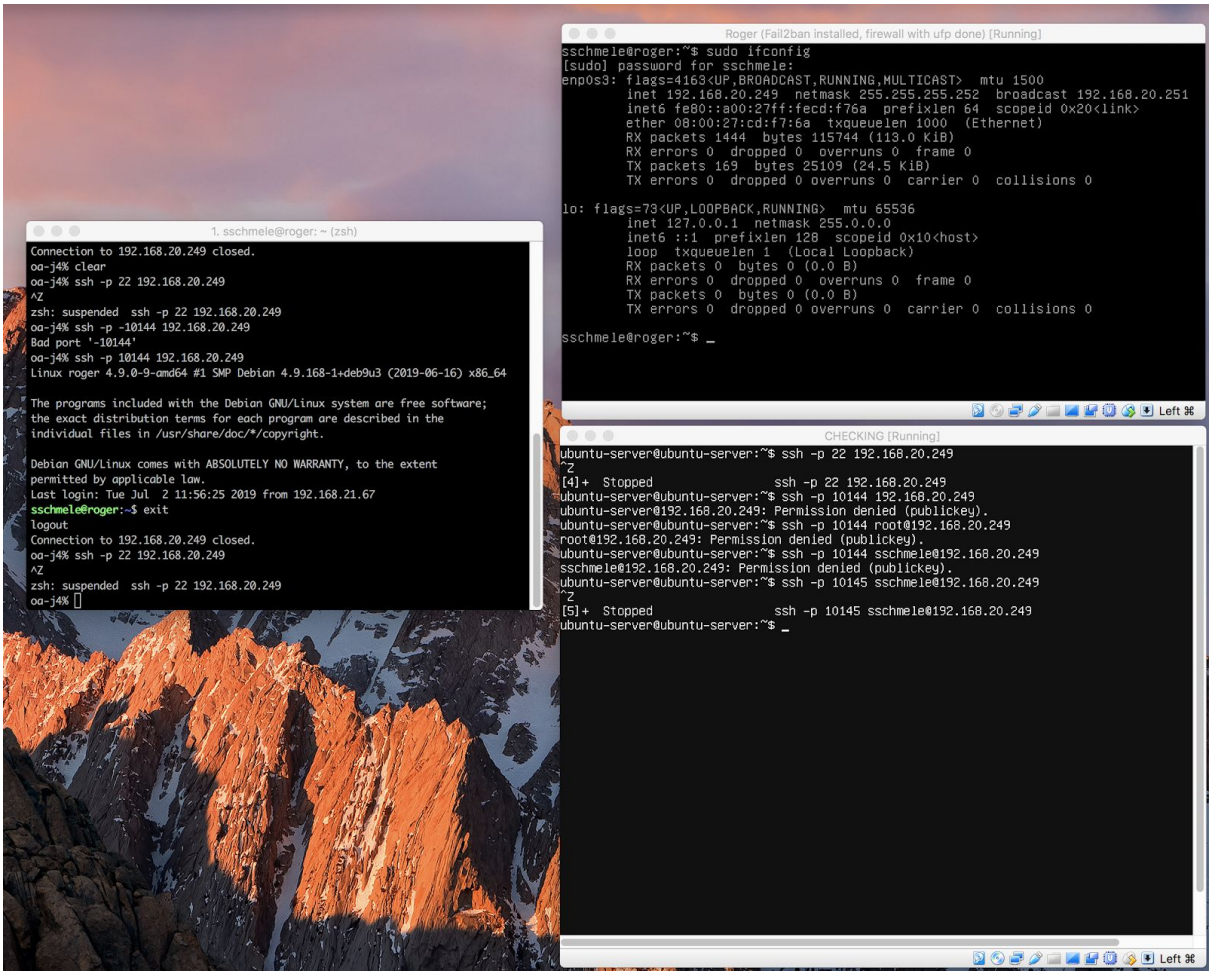






After that the access by password should be closed. После этой настройки закрывается доступ по паролю.

from the machine with public key we can enter and from machine without (Ubuntu) - no ways. Порт 22 теперь отключен.



Firewall

Задача: перекрыть все порты кроме 80 http, 443 для https и наш для ssh.
Можно сделать двумя способами: UFW или через пропись IP-tables

UFW:
http://blog.sedicomm.com/2018/07/06/kak-nastroit-brandmauer-ufw-na-ubuntu-i-debian/

sudo ufw status
List the ufw application profiles by typing:
sudo ufw app list
You will see a list of the application profiles:
Output
Available applications:
AIM
Bonjour
CIFS
. . .
WWW
WWW Cache
WWW Full

WWW Secure
. . .

The Apache profiles begin with WWW:

- WWW: This profile opens only port 80 (normal, unencrypted web traffic)
- WWW Cache: This profile opens only port 8080 (sometimes used for caching and web proxies)
- WWW Full: This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic)
- WWW Secure: This profile opens only port 443 (TLS/SSL encrypted traffic)

It is recommended that you enable the most restrictive profile that will still allow the traffic you've configured. Since we haven't configured SSL for our server yet in this guide, we will only need to allow traffic on port 80:
sudo ufw allow 'WWW'

You can verify the change by typing:
sudo ufw status

You should see HTTP traffic allowed in the displayed output:
Output
Status: active

```
To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
WWW                        ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
WWW (v6)                   ALLOW       Anywhere (v6)
```

As you can see, the profile has been activated to allow access to the web server.

IP-tables:

https://eax.me/iptables/

iptables -F - удаление всех правил
iptables -A INPUT DROP - изменение политики на прием сообщений
iptables -A INPUT -p TCP --dport [nb port] -j ACCEPT

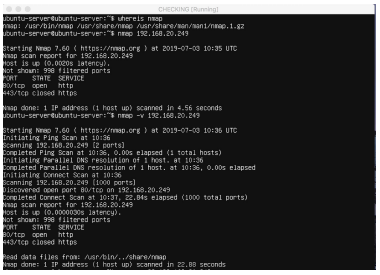
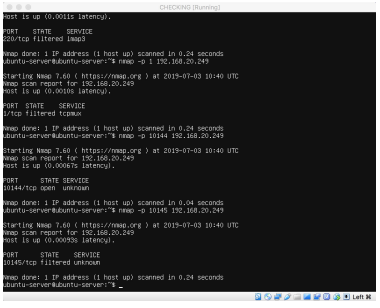
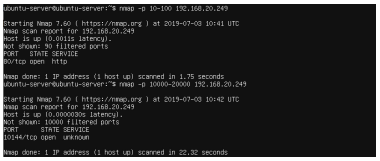
Scans protection

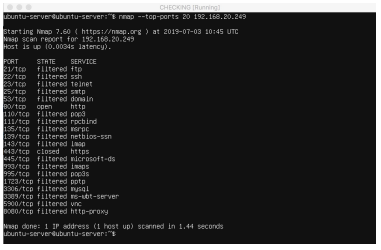
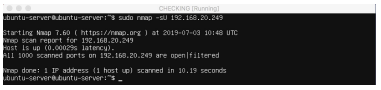
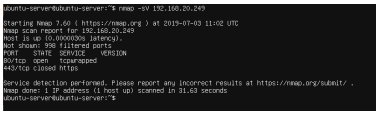
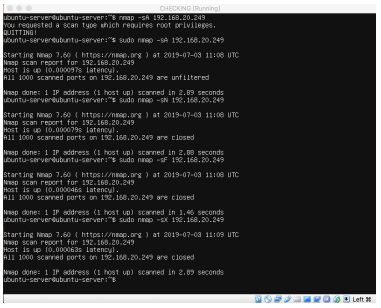
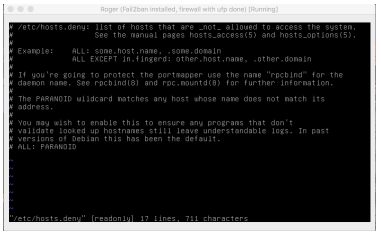
НУЖНО СКАЧАТЬ И НАСТРОИТЬ fail2ban - файл /etc/fail2ban/jail.conf - у меня это сделано не было

Portsenrty

Сканеры портов для проверки:
https://habr.com/ru/company/hosting-cafe/blog/281943/ - выбрала nmap
Nmap - https://xn--90aeniddllys.xn--p1ai/nmap-testiruem-bezopasnost-servera/
Nmap - https://www.shellhacks.com/ru/20-nmap-examples/

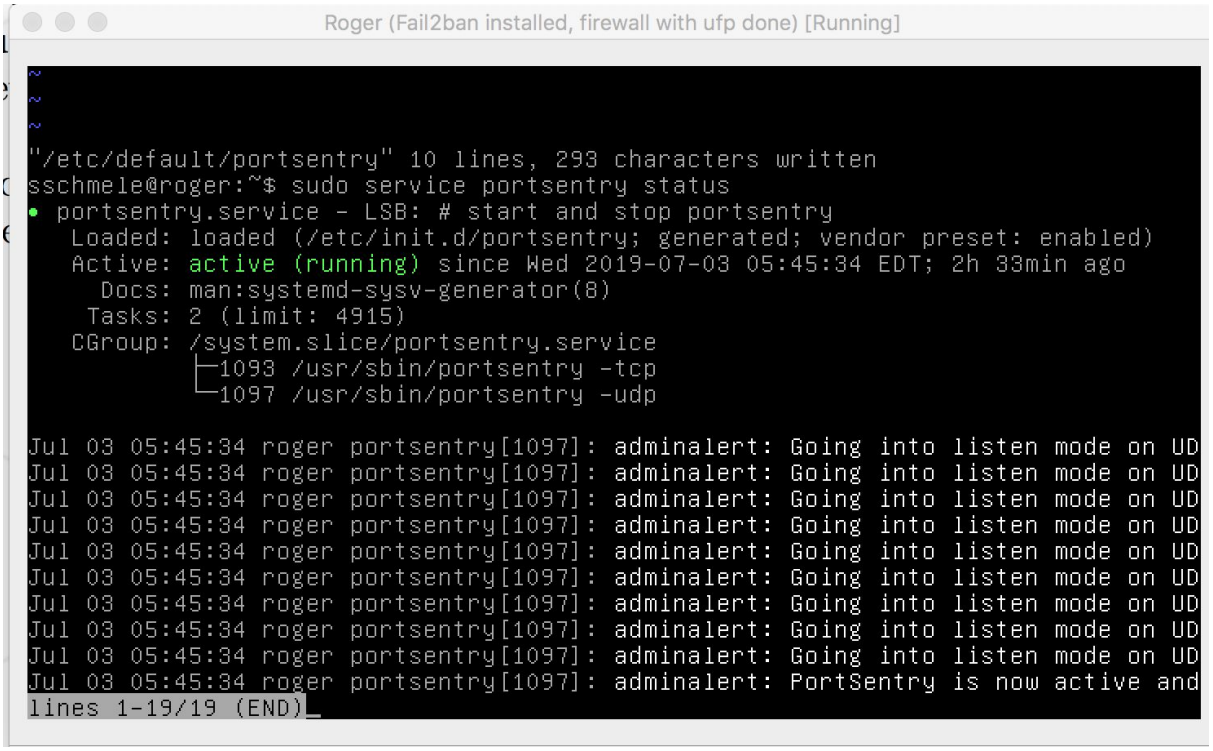
Nmap распознаёт шесть состояний портов:
open - открыт
closed - закрыт
filtered - порт недоступен, скорее всего фильтруется файрволом
unfiltered - порт доступен, но состояние определить не удалось
open|filtered - открыт или фильтруется файрволом
closed|filtered - закрыт или фильтруется файрволом

проверка детализированная: nmap -v 192.168.20.249	
проверка конкретных портов: nmap -p [port number] 192.168.20.249 22, 220, 1, 10144, 80, 443	
проверка диапазона портов: nmap -p 10-100 192.168.20.249 nmap -p 10000-20000 192.168.20.249	

проверка самых распространенных портов: nmap --top-ports 20 192.168.20.249	
сканирование UDP портов: sudo nmap -sU 192.168.20.249	
сканирование названия и версии сервиса на порту опцией -sV: nmap -sV 192.168.20.249	
сканирование на дыры в firewall: nmap -sA 192.168.20.249 nmap -sN 192.168.20.249 (NULL) nmap -sF 192.168.20.249 nmap -sX 192.168.20.249	
nmap -sT -O 192.168.20.249	
etc/hosts.deny до:	

Меняю файл: /etc/portsentry/portsentry.conf - прописываю рекомендованные порты и добавляю свой 10144 порт
Активирую скрипты для запуска работы portsentry: в файле /etc/default/portsentry прописываю:
TCP_MODE=”atcp”
UDP_MODE=”audp”

sudo service portsentry status



Конфигурационный файл /etc/portsentry/portsentry.modes

Файл служит для определения режима работы PortSentry. В версии portsentry-1.1 доступны следующие шесть опций, предназначенных для задания режимов работы:
•tcp – основной режим обнаружения сканирования для протокола TCP;
•udp – основной режим обнаружения сканирования для протокола UDP;
•stcp – «незаметный» режим обнаружения сканирования для протокола TCP;
•sudp – «незаметный» режим обнаружения сканирования для протокола UDP;
•atcp – расширенный «незаметный» режим обнаружения сканирования для протокола TCP;
•audp – расширенный «незаметный» режим обнаружения сканирования для протокола UDP.

Сервисы

Проверка работы сервисов, нужных для данного проекта:
Убрала exim4:
sudo service exim4 stop - на время сессии
sudo systemctl enable [] - добавить в автозагрузку
sudo systemctl disable [] - добавить в автозагрузку

```
sschmele@roger:~$ systemctl list-unit-files --type service --state enabled
UNIT FILE                  STATE
apache2.service            enabled
autovt@.service            enabled
console-setup.service      enabled
cron.service               enabled
getty@.service             enabled
keyboard-setup.service     enabled
networking.service         enabled
rsync.service              enabled
rsyslog.service            enabled
ssh.service                enabled
sshd.service               enabled
syslog.service             enabled
systemd-timesyncd.service  enabled
ufw.service                enabled
```

systemctl list-unit-files --type service --state enabled

```
sschmele@roger:~$ sudo service --status-all
 [ + ]  apache-htcacheclean
 [ + ]  apache2
 [ - ]  console-setup.sh
 [ + ]  cron
 [ + ]  dbus
 [ - ]  exim4
 [ - ]  hwclock.sh
 [ - ]  keyboard-setup.sh
 [ + ]  kmod
 [ + ]  networking
 [ + ]  portsentry
 [ + ]  procps
 [ - ]  rsync
 [ + ]  rsyslog
 [ + ]  ssh
 [ - ]  sudo
 [ + ]  udev
 [ + ]  ufw
sschmele@roger:~$ cat /etc/init.d
```

sudo service --status-all

Просмотр сервиса:

sudo systemctl [name]

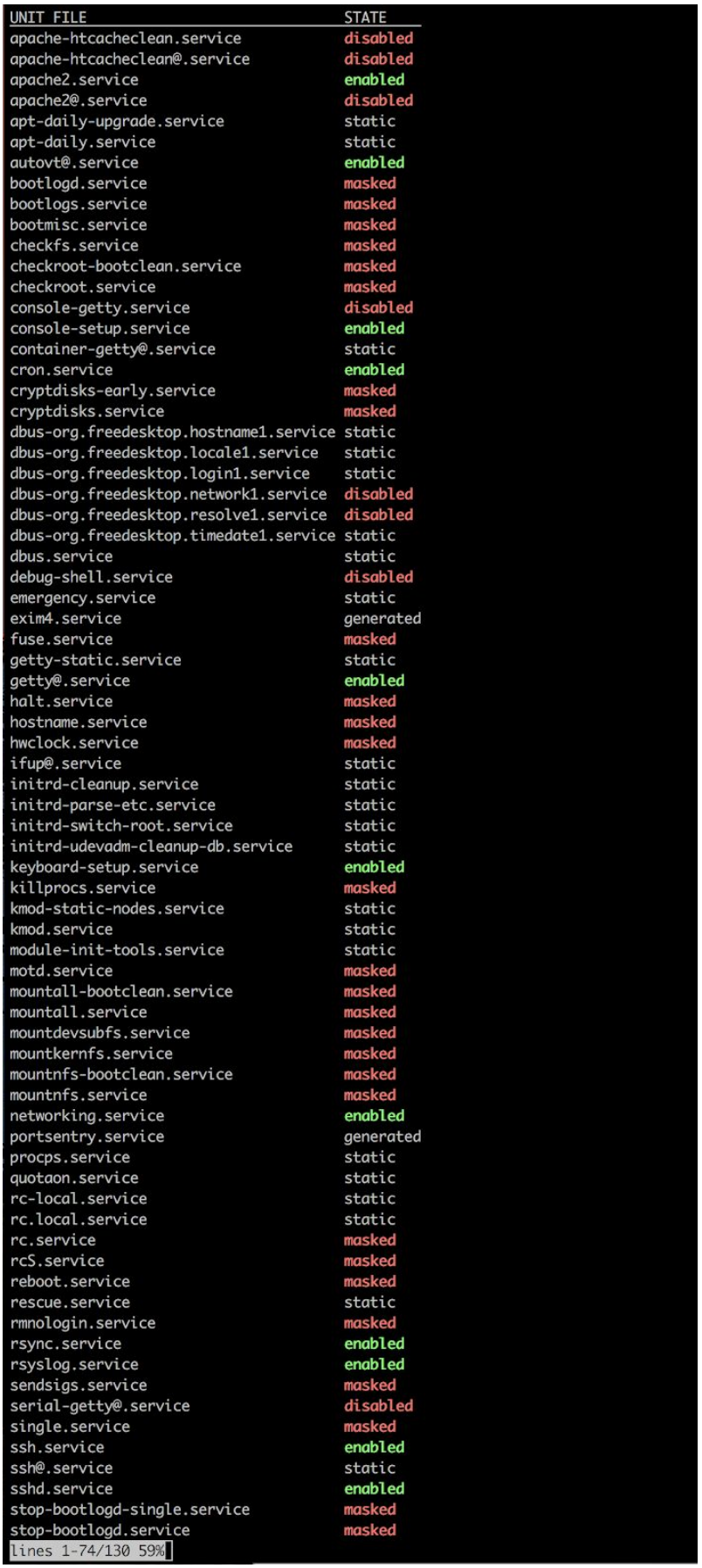
Юниты лежат в директории: /etc/systemd/system
Наличие суффикса (@) означает, что стартует не сам по себе сервис, а один из его инстансов.

Объяснение существующих, но непонятных для меня сервисов:

getty@.service и autovt@.service - https://habr.com/ru/post/304594/
rsyslog.service -синхронизация файлов и каталогов
syslog.service - журналирование событий

dbus - для общения приложений между собой
exim4 - почтовый сервис
hwclock.sh - управление аппаратными часами
kmod - управление модулями ядра Linux
procps - программы для мониторинга и завершения системных процессов
udev - управление устройствами

systemctl list-unit-files --type service



```
UNIT FILE                                  STATE
apache-htcacheclean.service                disabled
apache-htcacheclean@.service               disabled
apache2.service                            enabled
apache2@.service                           disabled
apt-daily-upgrade.service                  static
apt-daily.service                          static
autovt@.service                            enabled
bootlogd.service                           masked
bootlogs.service                           masked
bootmisc.service                           masked
checkfs.service                            masked
checkroot-bootclean.service                masked
checkroot.service                          masked
console-getty.service                      disabled
console-setup.service                      enabled
container-getty@.service                   static
cron.service                               enabled
cryptdisks-early.service                   masked
cryptdisks.service                         masked
dbus-org.freedesktop.hostname1.service     static
dbus-org.freedesktop.locale1.service       static
dbus-org.freedesktop.login1.service        static
dbus-org.freedesktop.network1.service      disabled
dbus-org.freedesktop.resolve1.service      disabled
dbus-org.freedesktop.timedate1.service     static
dbus.service                               static
debug-shell.service                        disabled
emergency.service                          static
exim4.service                              generated
fuse.service                               masked
getty-static.service                       static
getty@.service                             enabled
halt.service                               masked
hostname.service                           masked
hwclock.service                            masked
ifup@.service                              static
initrd-cleanup.service                     static
initrd-parse-etc.service                   static
initrd-switch-root.service                 static
initrd-udevadm-cleanup-db.service          static
keyboard-setup.service                     enabled
killprocs.service                          masked
kmod-static-nodes.service                  static
kmod.service                               static
module-init-tools.service                  static
motd.service                               masked
mountall-bootclean.service                 masked
mountall.service                           masked
mountdevsubfs.service                      masked
mountkernfs.service                        masked
mountnfs-bootclean.service                 masked
mountnfs.service                           masked
networking.service                         enabled
portsentry.service                         generated
procps.service                             static
quotaon.service                            static
rc-local.service                           static
rc.local.service                           static
rc.service                                 masked
rcS.service                                masked
reboot.service                             masked
rescue.service                             static
rmnologin.service                          masked
rsync.service                              enabled
rsyslog.service                            enabled
sendsigs.service                           masked
serial-getty@.service                      disabled
single.service                             masked
ssh.service                                enabled
ssh@.service                               static
sshd.service                               enabled
stop-bootlogd-single.service               masked
stop-bootlogd.service                      masked
lines 1-74/130 59%
```

Скрипты

Обновления:

1) sudo apt-get update

just updates the list of available packages from the repositories configured in /etc/apt/source.list and /etc/apt/source.list.d

2) sudo apt-get upgrade

updates installed software packages

Установка времени в системе linux

Утилита timedatectl доступна в системах, использующих systemd.

timedatectl list-timezones - показывает все временные зоны

sudo timedatectl set-timezone Europe/Moscow - установка временной зоны

date - проверка

Скрипт по обновлениям пакетов

```
sschmele@roger:~$ cat script_pack_update
#!/bin/bash
file="/var/log/update_script.log"
TIME="$(date)"
N=$'\n'

if [ -n "$(tail -1 $file)" ]
then
        sudo echo "$N" >> $file
fi
echo "$TIME$N" >> $file
echo -e "\033[32;01mUPDATING the sources of packages\033[0m"
sudo apt-get update >> $file
echo -e "\033[32;01mUPDATING packages\033[0m"
sudo apt-get upgrade >> $file
echo -e "\033[32;01mDONE\033[0m"
```

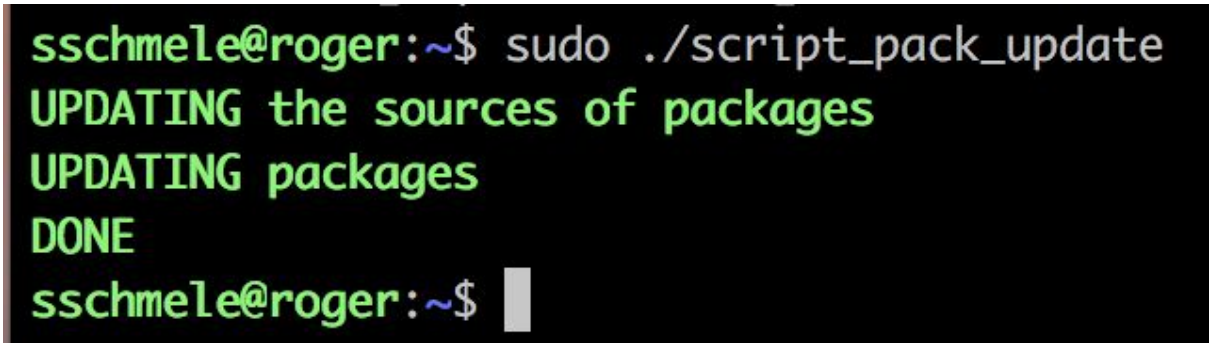
На всякий случай:

```
#!/bin/bash
file="/var/log/update_script.log"
TIME="$(date)"
N=$'\n'
```

```
if [ -n "$(tail -1 $file)" ]
then
        sudo echo "$N" >> $file
fi
echo "$TIME$N" >> $file
echo -e "\033[32;01mUPDATING the sources of packages\033[0m"
sudo apt-get update >> $file
echo -e "\033[32;01mUPDATING packages\033[0m"
sudo apt-get upgrade >> $file
echo -e "\033[32;01mDONE\033[0m"
```

Вывод скрипта:

sudo ./script_pack_update



Результат в файле /var/log/update_script.log:

```
sschmele@roger:~$ cat /var/log/update_script.log
Tue Jul  9 15:16:55 MSK 2019

Ign:1 http://ftp.us.debian.org/debian stretch InRelease
Hit:2 http://security.debian.org/debian-security stretch/updates InRelease
Hit:3 http://ftp.us.debian.org/debian stretch-updates InRelease
Hit:4 http://ftp.us.debian.org/debian stretch Release
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.


Tue Jul  9 15:18:28 MSK 2019

Hit:1 http://security.debian.org/debian-security stretch/updates InRelease
Ign:2 http://ftp.us.debian.org/debian stretch InRelease
Hit:3 http://ftp.us.debian.org/debian stretch-updates InRelease
Hit:4 http://ftp.us.debian.org/debian stretch Release
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.


Tue Jul  9 15:21:36 MSK 2019

Hit:1 http://security.debian.org/debian-security stretch/updates InRelease
Ign:2 http://ftp.us.debian.org/debian stretch InRelease
Hit:3 http://ftp.us.debian.org/debian stretch-updates InRelease
Hit:4 http://ftp.us.debian.org/debian stretch Release
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sschmele@roger:~$
```

Правило crontab

http://blog.sedicomm.com/2017/07/24/kak-dobavit-zadanie-v-planirovshhik-cron-v-linux-unix/

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO = root

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
@reboot         root    /home/sschmele/script_pack_update
00 4    * * 1   root    /home/sschmele/script_pack_update
@daily          root    /home/sschmele/script_cron_monito
#
```

Added:
@reboot root /home/sschmele/script_pack_update
00 4 * * 1 root /home/sschmele/script_pack_update
@daily root /home/sschmele/script_cron_monito

Легко запоминаемый формат:

```
* * * * * Команда, которая будет выполнена
- - - - -
| | | | |
| | | | - День недели (0 - 7) (воскресенье = 0 или 7)
| | | --- Месяц (1 - 12)
| | --- День месяца (1 - 31)
| ---- Час (0 - 23)
----- Минута (0 - 59)
```

Специальная строка	Значение
@reboot	Запускается только один раз при запуске.
@yearly	Запускается раз в год, «0 0 1 1 *».
@annually	Такое же, как у @yearly
@monthly	Запускается раз в месяц, «0 0 1 * *».

@weekly	Выполняется раз в неделю, «0 0 * * 0».
@daily	Выполняется один раз в день, «0 0 * * *».
@midnight	Такое же, как у @daily.
@hourly	Запускается один раз в час, «0 * * * *».

Проверка работы после перезагрузки:
Файл /var/log/update_script.log

```
Wed Jul 10 16:51:22 MSK 2019

Hit:1 http://security.debian.org/debian-security stretch/updates InRelease
Ign:2 http://ftp.us.debian.org/debian stretch InRelease
Hit:3 http://ftp.us.debian.org/debian stretch-updates InRelease
Hit:4 http://ftp.us.debian.org/debian stretch Release
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.


Wed Jul 10 16:51:47 MSK 2019

Hit:1 http://security.debian.org/debian-security stretch/updates InRelease
Ign:2 http://ftp.us.debian.org/debian stretch InRelease
Hit:3 http://ftp.us.debian.org/debian stretch-updates InRelease
Hit:4 http://ftp.us.debian.org/debian stretch Release
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sschmele@roger:/var/mail$
```

```
0 upgraded, 0 newly installed, 0 to remo
sschmele@roger:/var/mail$ sudo reboot
```

Получилось:

```
Wed Jul 10 18:34:55 MSK 2019

Get:1 http://security.debian.org/debian-security stretch/updates InRelease [94.3
 kB]
Ign:2 http://ftp.us.debian.org/debian stretch InRelease
Get:3 http://ftp.us.debian.org/debian stretch-updates InRelease [91.0 kB]
Hit:4 http://ftp.us.debian.org/debian stretch Release
Fetched 185 kB in 1s (131 kB/s)
Reading package lists...
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
The following package was automatically installed and is no longer required:
  libgnutls-openssl27
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
sschmele@roger:~$ _
```

Хеш-функции

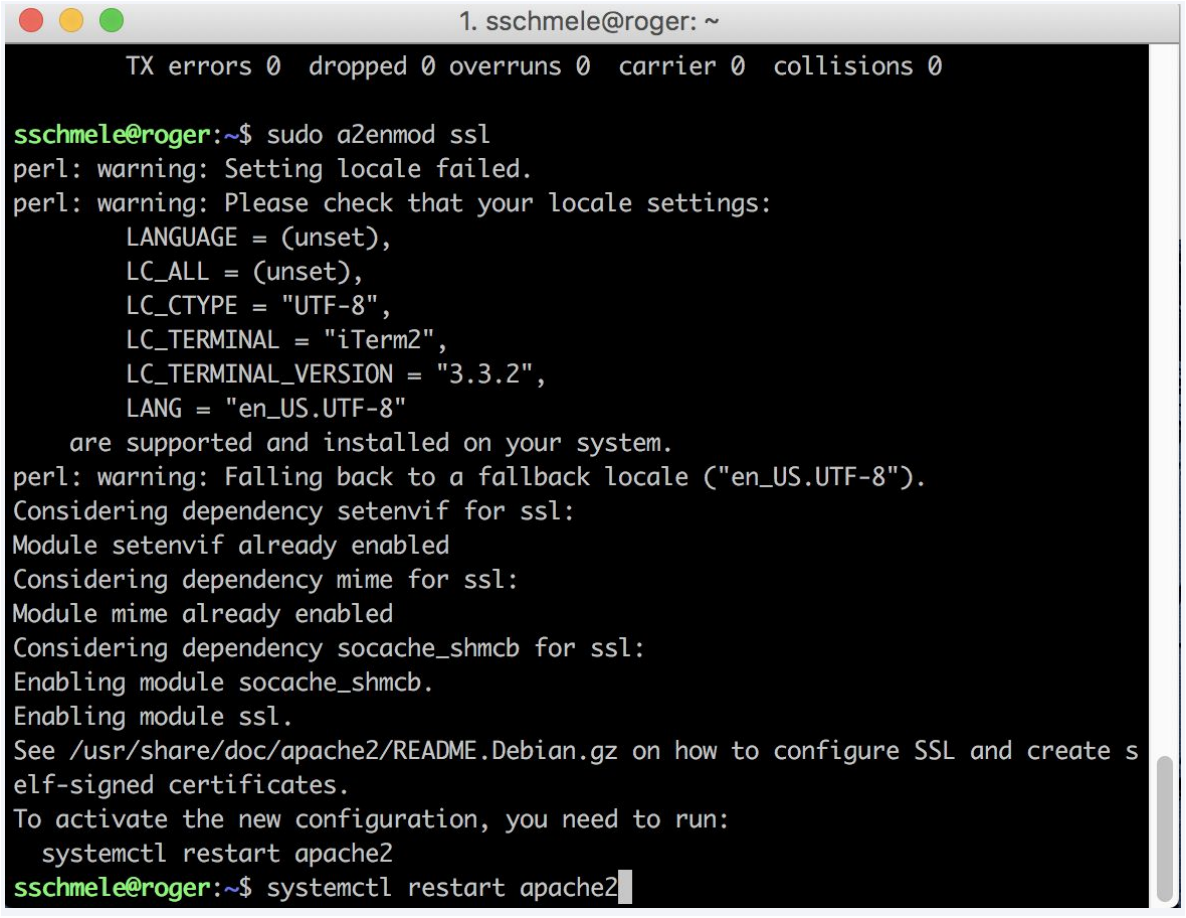
https://habr.com/ru/post/113127/

Скрипт по обновлениям через хеш-функции

```
sschmele@roger:~$ cat script_cron_monitor
#!/bin/bash
FILE="/home/sschmele/cron_hash.log"
CHECK="/etc/crontab"
LAST="$(tail -1 $FILE)"
NEW="$(md5sum $CHECK | awk '{print $1}')"

#echo "LAST = $LAST" | cat -e
#echo "NEW = $NEW" | cat -e

if [ $LAST !=  $NEW ]
then
        (echo "To: root@localhost";
        echo "Reply-To: root@localhost";
        echo "Subject: Changes in /etc/crontab";
        echo ""
        echo -e "LAST HASH-VALUE: $LAST\n\nTHE NEW ONE: $NEW\n\n") | sudo sendmail root@localhost
        echo "" >> $FILE
        echo "There are changes in the /etc/crontab file" >> $FILE
        date >> $FILE
        md5sum $CHECK | awk '{print $1}' >> $FILE
fi
```

```
#!/bin/bash
```

```
FILE="/home/sschmele/cron_hash.log"
CHECK="/etc/crontab"
LAST="$(tail -1 $FILE)"
NEW="$(md5sum $CHECK | awk '{print $1}')"

#echo "LAST = $LAST" | cat -e
#echo "NEW = $NEW" | cat -e

if [ $LAST !=  $NEW ]
then
        (echo "To: root@localhost";
        echo "Reply-To: root@localhost";
        echo "Subject: Changes in /etc/crontab";
        echo ""
        echo -e "LAST HASH-VALUE: $LAST\n\nTHE NEW ONE: $NEW\n\n") | sudo
sendmail root@localhost
        echo "" >> $FILE
        echo "There are changes in the /etc/crontab file" >> $FILE
        date >> $FILE
```




```
        md5sum $CHECK | awk '{print $1}' >> $FILE
fi
```

В /home/sschmele/cron_hash.log лежат hash-значения /etc/crontab файла
Сообщения приходят на root и перенаправляются на юзера sschmele:

```
sschmele@roger:~$ mail
"/var/mail/sschmele": 1 message 1 new
>N   1 Cron Daemon        Wed Jul 10 18:34  24/865   Cron <root@roger> /home/sschmele/script_pack_update
? 
```

Знак вопроса - это выбор опции. Если письмо одно, проще нажать “1”

```
sschmele@roger:~$ mail
"/var/mail/sschmele": 1 message 1 new
>N   1 Cron Daemon        Wed Jul 10 18:34  24/865   Cron <root@roger> /home/sschmele/script_pack_update
? 1
Return-path: <root@roger.ssroger>
Envelope-to: root@roger.ssroger
Delivery-date: Wed, 10 Jul 2019 18:34:59 +0300
Received: from root by roger.ssroger with local (Exim 4.89)
        (envelope-from <root@roger.ssroger>)
        id 1hlEcV-0000FF-H1
        for root@roger.ssroger; Wed, 10 Jul 2019 18:34:59 +0300
From: root@roger.ssroger (Cron Daemon)
To: root@roger.ssroger
Subject: Cron <root@roger> /home/sschmele/script_pack_update
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin>
X-Cron-Env: <MAILTO=root>
X-Cron-Env: <HOME=/root>
X-Cron-Env: <LOGNAME=root>
Message-Id: <E1hlEcV-0000FF-H1@roger.ssroger>
Date: Wed, 10 Jul 2019 18:34:59 +0300

UPDATING the sources of packages
UPDATING packages
DONE
```

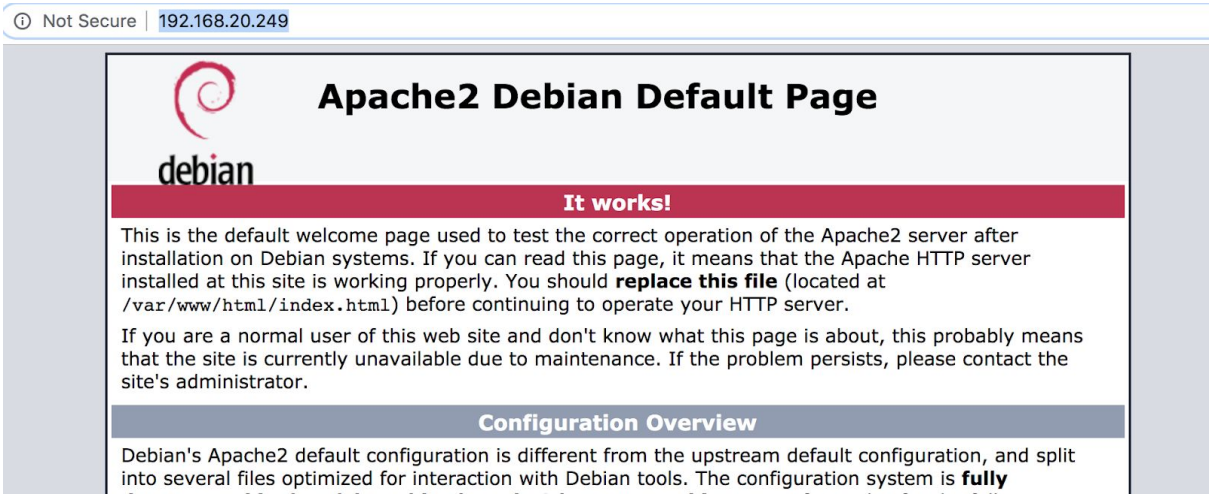
Для завершения прочитывания сообщения:
^D:

```
?
No applicable message
?
Saved 1 message in /home/sschmele/mbox
Held 0 messages in /var/mail/sschmele
```

Установка web-сервера - Web Part

https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-debian-9

После установки веб-сервера на протоколе http появляется лежащий в /var/www/html/ index.html файл:
http://192.168.20.249



Написание веб-сайта:
обязателен DOCTYPE
https://www.youtube.com/watch?v=uAcnIvKhfQQ

SSL - Создание самоподписанного сертификата

Самоподписанный сертификат не сможет подтвердить подлинности сервера, так как он не подписан проверенным центром сертификации (ЦС); тем не менее, такой сертификат позволит шифровать взаимодействие с веб-клиентами. Самоподписанный сертификат подходит пользователям, у которых пока что нет доменного имени. При наличии домена рекомендуется обратиться за подписью сертификата к одному из надёжных ЦС. Также можно получить бесплатный доверенный сертификат от сервиса Let’s Encrypt.
Поэтому обозначение:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

X.509 - стандарт открытого ключа
-x509 - делает подпись сама
-nodes - убирает защиту паролем

-keyout - место для файла ключа
-out - куда поместить созданный сертификат

Настройка связей: где что лежит

Сниппет Apache в каталоге etc/apache2/conf-available (фрагмент исходного текста или кода)

```
sudo nano /etc/apache2/conf-available/ssl-params.conf
```

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLHonorCipherOrder On
# Disable preloading HSTS for now.  You can use the commented out header
line that includes

# the "preload" directive if you understand the implications.
# Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
# Requires Apache >= 2.4.11
SSLSessionTickets Off
```

Далее настройка основного конфигурационного файла:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

В этом проекте файл был скопирован и основным файлом (который был запущен как хост) являлся файл 192.168.20.249.conf

Основные изменения в файле:

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin sschemele@roger.com
ServerName 192.168.20.249
DocumentRoot /var/www/html/192.168.20.249

DirectoryIndex roger.html index.html - эта фраза помогает подгружать файл
не только имени index из указанного root: сперва ищется файл с указанным
названием, если не находится - подгружается index.html

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
SSLEngine on
```

```
SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
<FilesMatch "\.(cgi|shtml|phtml|php)$">
SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
SSLOptions +StdEnvVars
</Directory>
</VirtualHost>
</IfModule>
```

Переадресация с 80 порта:

```
sudo nano /etc/apache2/sites-available/192.168.20.249.conf
```

```
<VirtualHost *:80>
. . .
Redirect permanent "/" "https://your_domain_or_IP/"
. . .
</VirtualHost>
```

Как теперь выглядит главный конфигурационный файл:

```
<IfModule mod_ssl.c>
        <VirtualHost *:80>
                Redirect permanent "/" "https://192.168.20.249/"
        </VirtualHost>
        <VirtualHost _default_:443>
                ServerAdmin   sschmele@roger.com
                ServerName    192.168.20.249

                DocumentRoot /var/www/html/192.168.20.249/
                DirectoryIndex roger.html index.html

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

               SSLEngine on
               SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
              SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key


                <FilesMatch "\.(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
                <Directory /usr/lib/cgi-bin>
                                SSLOptions +StdEnvVars
                </Directory>
        </VirtualHost>
</IfModule>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Инфа:
https://www.8host.com/blog/sozdanie-samopodpisannogo-ssl-sertifikata-dlya-apache-v-debian-9/

Обновление настроек Apache и запуск

mod_ssl : `sudo a2enmod ssl`
mod_headers (для сниппета): `sudo a2enmod headers`

Виртуальный хост:

```
sudo a2ensite 192.168.20.249
```

Файл ssl-params.conf:

```
sudo a2enconf ssl-params
```

Синтаксис на наличие ошибок:

```
sudo apache2ctl configtest - Syntax OK
```

Перезапуск сервера:

```
sudo systemctl restart apache2
```

Копирование файлов через ssh:

scp -P 10144 -r ~/Downloads/Login_v20 sschmele@192.168.20.249:/home/sschmele/page

-r - флаг для переноса папки

Автоматизация деплоя (git-hooks)

Инфа про деплой и типы хостинга:
https://ru.hexlet.io/courses/php-mvc/lessons/deploy/theory_unit

Объяснение гит-хуков:
https://www.digitalocean.com/community/tutorials/how-to-use-git-hooks-to-automate-development-and-deployment-tasks
Общая инфа по настройке: https://habr.com/ru/post/127213/
Конечная версия, на которую ориентировалась:
https://www.digitalocean.com/community/tutorials/how-to-set-up-automatic-deployment-with-git-with-a-vps

На сервере продолжаем работать с папкой, в которой лежит проект, а также заводим гит-репозиторий, через который будут проходить изменения и выгружаться в виде изменений на сайте.

Репозиторий на сервере

Папка ~/repo.git
Создание хука по деплою при появлении изменений:

```
sschmele@roger:~$ cat repo.git/hooks/post-receive
#!/bin/sh
git --work-tree=/var/www/html/192.168.20.249 --git-dir=/home/sschmele/repo.git checkout -f
sschmele@roger:~/repo.git/hooks$ chmod +x post-receive
```

Репозиторий на основной машине

Папка ~/work/Roger_project
    git init
    все файлы
Установка удаленного репозитория:
    git remote add prod ssh://sschmele@192.168.20.249:10144/home/sschmele/repo.git
Если что, изменения записываются в файл: ~/work/Roger_project/.git/config
    git add . && git commit -m “New”
    git push prod master

Чек-лист

VM Part

Here we evaluate the VM Part.

Checks

We will check in this section that the instructions
have been respected. If only one of these points is wrong you must
end the evaluation and give a score of 0.
- The VM runs well on a Linux OS.
- Technos such as Traefik as well as that Docker/Vagrant/etc.
containers are not used in this project.

Yes | No

Install and Update

We will check in this section that the VM is correctly
installed and updated. To validate the evaluation of the VM,
the following tests must be passed. If this one fails,
the VM evaluation is failed and finished.

- The size of the VM disk is 8 GB.

- There is at least one 4.2 GB partition.

- From the shell of the VM, run the command that lets you
know if the OS and packages are up to date.
If you discover that the OS or packages are not up to date,
this test has failed.

- From the shell of the VM, run the command that allows
to know which packages are installed. If you discover that
docker/vagrant/traefik type packages are installed,
this test has failed.

Yes | No

Network and Security Part

Here we evaluate the Network and Security Part.

Network and Security

We are going to check in this section if the VM network is correctly
configured. To validate this part, those seven tests has to be successed.
If one of those tests is failed then all the Network Part is failed and
stopped.

- Ask the evaluated person to create a user with SSH key to be able to
connect to the VM. He must be part of the sudo group. If it's not
in this case, this test is failed.

- If this user executes the sudo command he must be able
to use commands that require special rights. If this is not
the case, this test is failed.

- Check that the DHCP service of the VM is deactivated If
not, this test is failed.

- Choose a different netmask than /30, ask the evaluated person to
configure a network connection with this netmask on the host and guest side.
The evaluated person will choose the IPs. If it is not successful, this test is failed.

- From a shell on the VM, check that the port of the SSH has been successfully
been modified. SSH access MUST be done with publickeys.
The root user should not be able to connect in SSH. If this is not
the case, this test is failed.

- From a shell on the VM, run the command that lists all firewall rules.
If no rules are in place or that it is not sufficient in relation to the
request from the subject, then this test is failed.

- From a shell on your computer, run the command that allows you to
to test a DOS (Slowloris or other). Check that everything is still working.
In addition, make sure that a Fail2Ban service (or similar service) is installed on the VM.
If this is not the case, this test is failed.

- From a shell on the VM, run the command that
lists the open ports. Check that the open ports
correspond to the subject's request. If not, this
test is failed.

- Check if the active services of the machine are only those
necessary for its proper functioning. If not, this test
has failed.

- Check that there is a script to update all sources
of package, packages, which log into the right file and that it is in cron.
If this is not the case, this test is failed.

- Check that there is a script to monitor the changes in the file
/etc/crontab and sends an email to root if it has been modified.
You must therefore receive an email showing that the file has changed, either locally with
the mail order, either in your own mailbox. If not, this test
has failed.

You must therefore receive an email showing that the file has changed, either locally with the mail order, either in your own mailbox. If not, this test has failed.

- Check that there is self-signed SSL on all services. If this is not the case, this test is failed.

Yes | No

Web Part

Here we evaluate the Web Part.

Web Server

We will check in this section that the web server is implemented on the VM. To validate this part, the next three tests must be passed. If at least one of them of them fail, the Web server evaluation is failed and finished. Proceed to the next section of the scale.

- From a shell of the VM, check that the package of a Web server is installed. If this is not the case, this test is failed.

- From a shell of the VM, check that there is only one active configuration on the web server and not the default one. In addition, it should not "Listen" on the localhost of the VM. If it is not respected, this test has failed.

- Check that the web application corresponds to what is required in the subject and is available on any browser on the IP of the VM or a host (init.login.fr for example). If it's not the in this case, this test is failed.

Yes | No

Deployment Part

Here we evaluate the Deployment Part of the VM.

Deployment Part

We will check in this section that the deployment is going well. To validate this part, the following two tests must be succeed.

- Ask the student to explain how he chose to do the deployment and why he chose this solution.

- Make a minor modification on the site to ensure that the deployment is working well.

Yes | No

Bloopers (заметки об ошибках)

Git hooks:

Гит пуш может зависнуть на определенном проценте, если папкой гита со стороны сервера сделать папку в неизменяемые обычным пользователем директории.
Обновление гита вне директории:

```
git -C /var/www/html/192.168.20.249 pull roger master
```