

Домашнее задание:

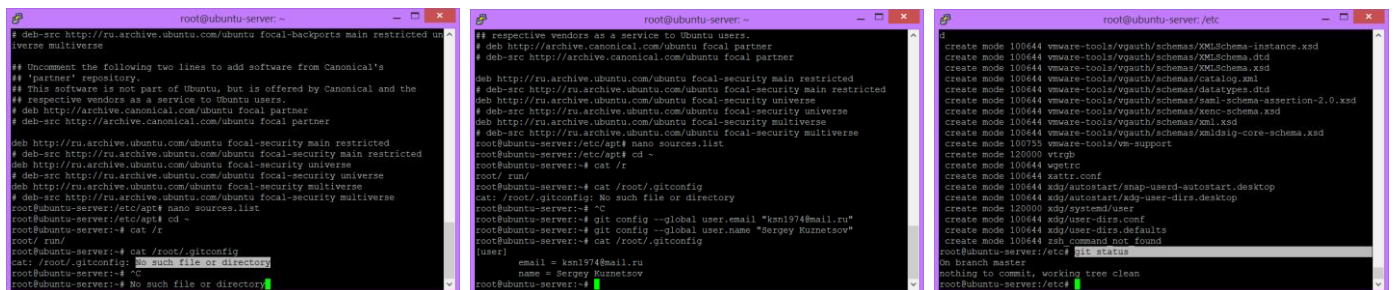
1. Подключить репозиторий с nginx любым удобным способом, установить nginx и потом удалить nginx, используя утилиту dpkg.
2. Установить пакет на свой выбор, используя snap.
3. Настроить iptables: разрешить подключения только на 22-й и 80-й порты.
4. * Настроить проброс портов локально с порта 80 на порт 8080.

1. Подключить репозиторий с nginx любым удобным способом, установить nginx и потом удалить nginx, используя утилиту dpkg.

Подключим GIT для более наглядной демонстрации работы с репозиториями (установками/удалениями пакетов). Убедимся, что GIT не установлен (`cat /root/.gitconfig`) – «No such file or directory».

Введем почту (`git config --global user.email "ksn1974@mail.ru"`) и имя пользователя (`git config --global user.name "Sergey Kuznetsov"`). Проверим (`cat /root/.gitconfig`)

Запускаем GIT командой (`cd /etc/ && git init && git add . && git commit -a -m 'init'`). Перейти в отслеживаемый каталог «`cd /etc/`», инициализируем «`git init`», добавляем все что есть «`git add .`» и делаем commit. Проверим статус (`git status`) – все изменения сделаны и «закомичены». Впоследствии, для commit, будет достаточно команды (`cd /etc/ && git add . && git commit -a -m 'update'`)

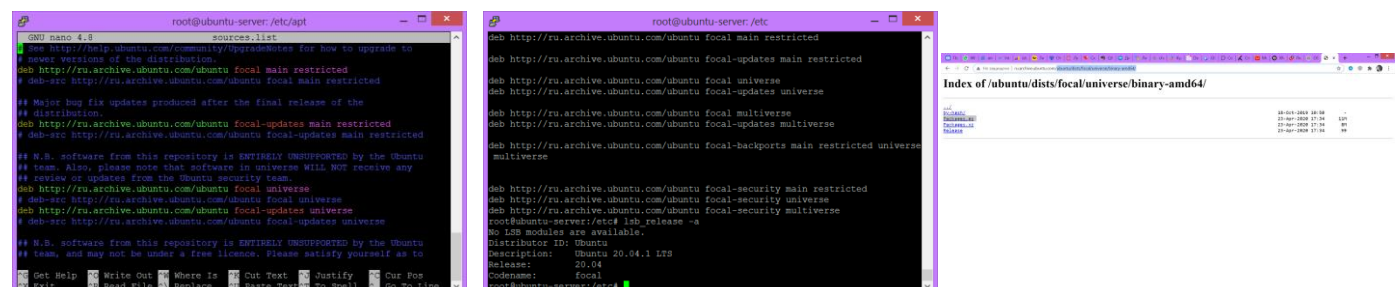


```
root@ubuntu-server:~# cat /etc/apt/sources.list > /root/.gitignore
root@ubuntu-server:~# cd /etc/ && git init
root@ubuntu-server:/etc# git add .
root@ubuntu-server:/etc# git commit -a -m 'init'
root@ubuntu-server:/etc# git config --global user.email "ksn1974@mail.ru"
root@ubuntu-server:/etc# git config --global user.name "Sergey Kuznetsov"
root@ubuntu-server:/etc# cat /root/.gitconfig
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    worktree = /etc
[remote "origin"]
    url = http://ru.archive.ubuntu.com/ubuntu
    fetch = +refs/heads/*:refs/remotes/origin/*
[user]
    email = ksn1974@mail.ru
    name = Sergey Kuznetsov
```

Под «root» посмотрим список подключенных репозиториях, просмотрев файл «sources.list» в каталоге «/etc/apt/» (`nano /etc/apt/sources.list`). Без комментариев, как любил выражаться мой бывший босс «в сухом остатке», просмотреть можно (`grep -v "^#" /etc/apt/sources.list`).

Текущий релиз сервера «20.04.1» (`lsb_release -a`).

Зайдем на сайт «<http://ru.archive.ubuntu.com/ubuntu>» («Codename» со значением «focal» указывает на него, как репозиторий «main»). Скачаем архив «Packages.gz», расположенный по адресу «ubuntu.dists.focal.universe.binary-amd64/» и убедимся, что в нем ...



```
root@ubuntu-server:/etc# nano /etc/apt/sources.list
deb http://ru.archive.ubuntu.com/ubuntu focal main restricted
deb http://ru.archive.ubuntu.com/ubuntu focal universe
deb http://ru.archive.ubuntu.com/ubuntu focal-security main restricted
deb http://ru.archive.ubuntu.com/ubuntu focal-security universe
deb http://ru.archive.ubuntu.com/ubuntu focal-security multiverse
deb http://ru.archive.ubuntu.com/ubuntu focal-updates main restricted
deb http://ru.archive.ubuntu.com/ubuntu focal-updates universe
deb http://ru.archive.ubuntu.com/ubuntu focal-updates multiverse
deb http://ru.archive.ubuntu.com/ubuntu focal-backports main restricted universe multiverse

root@ubuntu-server:/etc# lsb_release -a
LSB Version: 20.04
Distributor ID: Ubuntu
Description: Ubuntu 20.04.1 LTS
Release: 20.04
Codename: focal

root@ubuntu-server:/etc# curl -O http://ubuntu.dists.focal.universe.binary-amd64/Packages.gz
```

Подключим репозиторий «*nginx*» командой (*apt-add-repository ppa:nginx/stable*). Проверим версию (*apt search nginx | grep '^nginx'*) – стала новее «*nginx/focal 1.18.0-3ubuntu1+focal2 all*»

Чтобы убедиться в старте «nginx», введем команду (`netstat -nap | grep ngi`), пришлось установить (`apt install net-tools`) и снова ввести (`netstat -nap | grep ngi`) – процессы «nginx» стартовали.

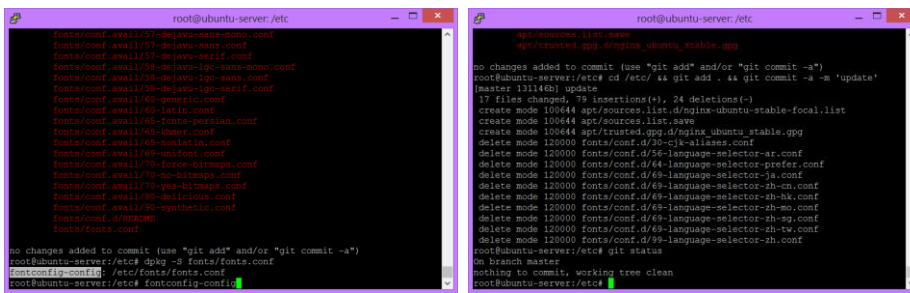
[illegible]

Воспользуемся (apt autoremove) - пытается почистить те пакеты, которые не используются. ... Смотрим, что получилось (dpkg --get-selections | grep nginx). «ii» стало «rc», что говорит об удалении пакетов, но оставленных конфигурациях. Это «нужная» политика стадий удаления для переустановки пакетов без потери настроек.

Для окончательной чистки конфигурации воспользуемся «apt purge nginx...» с перечислением объектов (apt purge nginx libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip nginx-common). Вводим (dpkg --get-libs | grep nginx) – все чисто!

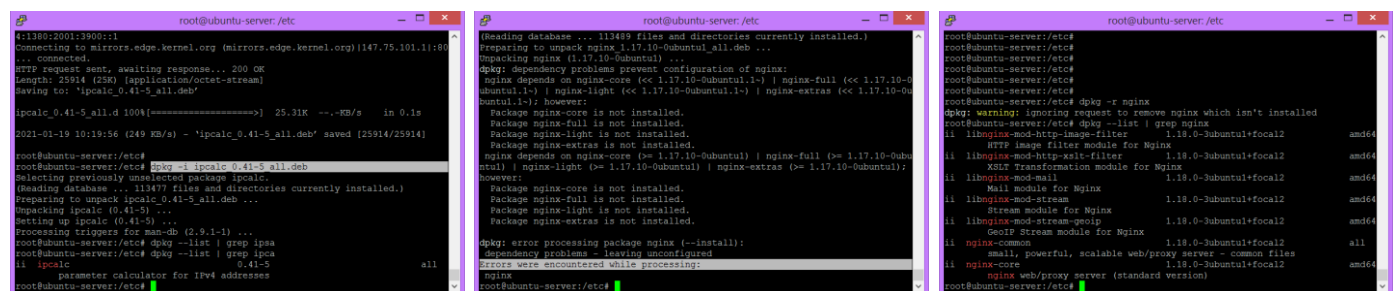
The image displays three terminal windows from the 'root@ubuntu-server:/etc' directory, showing the process of installing NGINX on Ubuntu 18.04. The first window shows the removal of the old 'nginx' package and the installation of the new one. The second window shows the removal of the old 'nginx-core' package and the installation of the new one. The third window shows the removal of the old 'nginx-common' package and the installation of the new one. The output of the 'dpkg-query -f='\${Package} \${Version} \${Architecture}\n' command is shown in the first window, listing the installed packages and their versions.

Удалим этот пакет (`apt purge fontconfig-config`). Также сделаем и с остальными пакетами (`apt purge fontconfig-config`), (`apt purge fonts-dejavu-core`)... Сделаем `commit` (`cd /etc/ && git add . && git commit -a -m 'update'`)

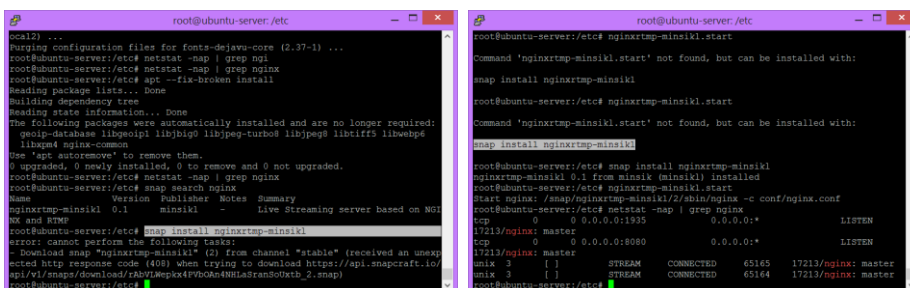


Если пробуем скачать (wget http://mirrors.kernel.org/ubuntu/pool/main/n/nginx/nginx_1.17.10-0ubuntu1_all.deb) и установить пакет «nginx» (dpkg -i nginx_1.17.10-0ubuntu1_all.deb), на стадии установки возникают проблемы. Дело в том, что «dpkg» не отслеживает связи...

Исправим ошибки, возникшие при инсталляции (`apt --fix-broken install`). Запустим (`netstat -nap | grep ngi`) - все же остались какие-то процессы... Удалим наши пакеты с использованием «dpkg». Чтобы удалить пакет, не затрагивая зависимости, используют команду (`dpkg -r nginx`) - аналог «apt remove».



Очистим после прошлых экспериментов (`apt purge fonts-dejavu-core`), (`apt --fix-broken install`), (`netstat -nap | grep nginx`)... Все ок. Ищем пакет с «nginx» (`snap search nginx`) и инсталлируем (`snap install nginxrtmp-minsikl`). Пакет скачался и был установлен.



3. Настроить iptables: разрешить подключения только на 22-й и 80-й порты.

Уточнимся по текущему адресу нашей машины (ip addr) – «192.168.100.188».

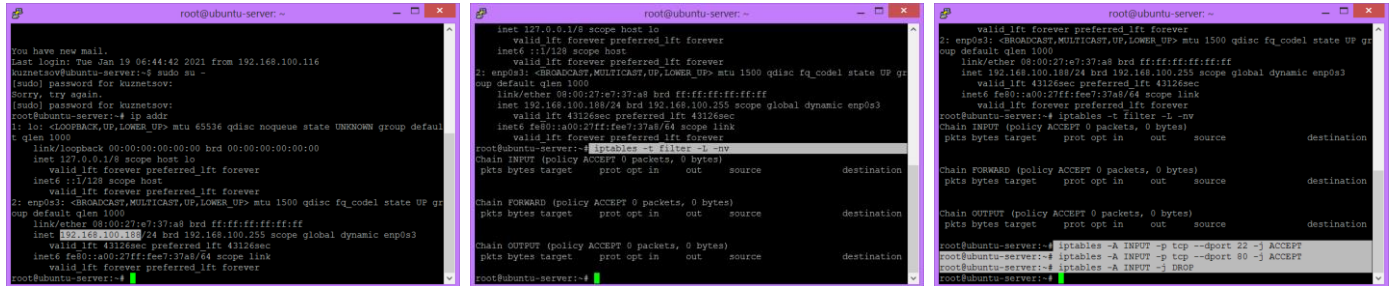
Удалим все «старые» правила (iptables -F) и посмотрим текущие правила фильтрации (iptables -t filter -L -nv) – все разрешено.

Запретим все порты, за исключением «22» и «80» командами:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
iptables -A INPUT -j DROP
```



```
root@ubuntu-server:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e7:37:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.188/24 brd 192.168.100.255 scope global dynamic eth0
        valid_lft 43126sec preferred_lft 43126sec
    inet6 fe80::a00:27ff:fe07:37a8/64 scope link
        valid_lft forever preferred_lft forever

root@ubuntu-server:~# iptables -F
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in  out  source      destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in  out  source      destination

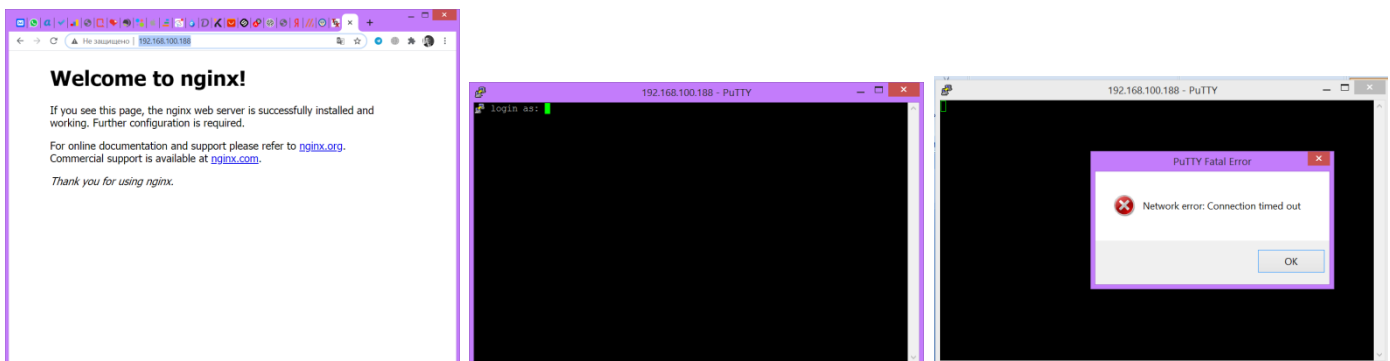
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in  out  source      destination

root@ubuntu-server:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
root@ubuntu-server:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@ubuntu-server:~# iptables -A INPUT -j DROP
```

Убедимся в браузере, что порт «80» открыт (<http://192.168.100.188:80/>).

И регистрация по «22» порту также открыта.

В противном случае, если удалить разрешающее правило (iptables -D INPUT -p tcp --dport 22 -j ACCEPT), регистрация закрыта.



4. Настроить проброс портов локально с порта 80 на порт 8080

Удалим все «старые» правила (iptables -F) и (iptables -t nat -F). Проверим отсутствие маршрутизаций (iptables -t nat -L -nv).

Текущий адрес нашей машины (ip addr) – «192.168.1.71».

Посмотрим, какой порт слушает «nginx» (netstat -nap | grep nginx) – «80».

```
root@ubuntu-server:~# iptables -F
Chain OUTPUT (policy ACCEPT 95 packets, 14213 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
  pkts bytes target     prot opt in     out     source            destination
root@ubuntu-server:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ee:73:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.71/24 brd 192.168.1.255 scope global dynamic ens3
        valid_lft 86265sec preferred_lft 86265sec
    inet6 fe80::a00:27ff:fe73:a8/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu-server:~# netstat -nap | grep nginx
tcp        0      0 0.0.0.0:*               LISTEN      734/nginx: master p
tcp6       0      0 ::::8000               LISTEN      734/nginx: master p
unix 3      [ ]          STREAM  CONNECTED 24564 734/nginx: master p
unix 3      [ ]          STREAM  CONNECTED 24563 734/nginx: master p
root@ubuntu-server:~#
```

По умолчанию, при заходе на сайт используется «80» порт. Убедимся, что браузер (http:// 192.168.1.71/) - отвечает...

А вот порт «8080», через браузер не работает (http:// 192.168.1.71:8080/).

Насколько понимаю, по аналогии с классной работой, в задании мы должны показать, как после проброса порта, он должен начать работать через браузер. Поэтому мы сделаем не проброс порта «80» на «8080» по «PREROUTING», а наоборот: (iptables -t nat -A PREROUTING -p tcp -m tcp --dport 8080 -j REDIRECT --to-port 80) Убедимся, что это правило появилось в таблице «nat» (iptables -t nat -L -nv) - таблица для трансляции адресов и портов.

```
root@ubuntu-server:~# iptables -t nat -A PREROUTING -p tcp -m tcp --dport 8080 -j REDIRECT --to-port 80
Chain PREROUTING (policy ACCEPT 2 packets, 156 bytes)
  pkts bytes target     prot opt in     out     source            destination
2 104 REDIRECT tcp -- *      *      0.0.0.0/0      0.0.0.0/0
Chain INPUT (policy ACCEPT 9 packets, 260 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain OUTPUT (policy ACCEPT 1 packets, 152 bytes)
  pkts bytes target     prot opt in     out     source            destination
Chain POSTROUTING (policy ACCEPT 1 packets, 152 bytes)
  pkts bytes target     prot opt in     out     source            destination
root@ubuntu-server:~#
```

Убедимся, теперь через браузер «» слушает оба порта «80» и «8080». Цель достигнута!

```
root@ubuntu-server:~# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ee:73:a8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.71/24 brd 192.168.1.255 scope global dynamic ens3
        valid_lft 86265sec preferred_lft 86265sec
    inet6 fe80::a00:27ff:fe73:a8/64 scope link
        valid_lft forever preferred_lft forever
root@ubuntu-server:~# netstat -nap | grep nginx
tcp        0      0 0.0.0.0:*               LISTEN      734/nginx: master p
tcp6       0      0 ::::8000               LISTEN      734/nginx: master p
unix 3      [ ]          STREAM  CONNECTED 24564 734/nginx: master p
unix 3      [ ]          STREAM  CONNECTED 24563 734/nginx: master p
root@ubuntu-server:~#
```

Естественно, не забыть настроить маскардинг (NAT-маскировку) IP для внешней сети:

iptables -t nat -A POSTROUTING -s 127.0.0.1 /24 ! -d 127.0.0.1 /24 -j MASQUERADE