

(устанавливал Debian с родным DE, т.к. в системе без DE не работало автомонтирование NFS папок в fstab)

Оглавление

1. Установка и настройка SSH.....	1
2. Установка и настройка SUDO	1
3. Установим MC	2
Добавим репозитории.....	2
Разноцветный терминал	2
4. Добавляем диски.....	2
Идентифицируем жесткий диск	2
Создание файловой системы.....	2
5. Создаем пользователей	4
Объединяем пользователей в группу.....	4
6. Создаем папки	4
7. Меняем пользователей и права созданных папок.....	4
8. Установка и настройка NFS.....	4
9. Установка и настройка Samba.....	5
10. Настройка transmission-daemon в Debian 9 (Stretch).....	7
11. Plex.....	11
плагин Кинопоиск для отображения инфо о фильмах с указанием рейтинга Кинопоиска.....	12
перенос настроек Plex с другого сервера (не проверено)	12
12. Яндекс.Диск	12
13. Установка Webmin.....	13
14. Установка NGINX + php-fpm + mysql.....	13
Сначала установим NGINX.....	13
Следующим шагом устанавливаем php.....	14
Осталось установить MySQL.....	15
Автозагрузка NGINX	16
15. Letsencrypt.....	16
Настройка Certbot.....	16
16. Qbittorrent-tox.....	20

В процессе установки Debian задаем пароль для пользователя root - rootpasswd, создаем пользователя username с паролем userpasswd.

1. Установка и настройка SSH

В консоли от пользователя root (можно воспользоваться SU и ввести пароль ROOT)

```
apt update
apt install ssh
```

2. Установка и настройка SUDO

```
su
```


Убедившись в том, что это нужный диск, приступаем. Для начала нужно будет создать новый раздел, для этого опять используем программу *fdisk*:

```
sudo fdisk /dev/sdc
```

Запустится утилита *fdisk*, которая попросит ввести какую-нибудь команду.

```
Welcome to fdisk (util-linux 2.25.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Command (m for help):
```

Вводим одну единственную букву - *n*.

Затем отвечаем на вопрос о типе раздела:

```
Partition type
p primary (0 primary, 0 extended, 4 free)
e extended (container for logical partitions)
```

Если планируется создавать один раздел, или больше (но не более четырех), то выбираем пункт *p*. У *Partition number*, *First sector* и *Last sector* оставляем все по умолчанию и просто нажимаем *Enter*. Если все прошло успешно, то появится сообщение примерно следующего содержания:

```
Created a new partition 1 of type 'Linux' and of size 931.5 GiB.
```

Теперь нужно сохранить эти изменения - вводим команду *w*, которая укажет программе *fdisk* о необходимости выйти, предварительно записав изменения на диск. Если же что-то было сделано не так, то можно выйти без изменений введя команду *q*.

Далее нужно создать файловую систему на жестком диске. Пусть это будет *ext4*:

```
sudo mkfs.ext4 /dev/sdc1
```

Так же может оказаться, что диск уже до этого использовался, и там вполне может быть уже существующая файловая система. В таком случае при форматировании раздела появится следующее сообщение:

```
/dev/sdc1 contains a ntfs file system
Proceed anyway? (y,n)
```

Если на жестком диске нет никаких нужных данных, и не планируется использование файловой системы *NTFS*, то нужно ответить *y*.

Осталось смонтировать (добавить) раздел в систему. Допустим это будет папка */backup*, расположенная в корне. Создаем папку:

```
mkdir /backup
```

Теперь получим идентификатор раздела командой *blkid*:

```
sudo blkid
```

После чего будет выведен список всех разделов в системе и их *UUID*:

```
/dev/sdc1: UUID="360ca23a-300t-asdd-8c8e-a3xdf46dlpf1" TYPE="ext4"
```

Находим нужный, и запоминаем его *UUID*. Теперь открываем */etc/fstab*, и добавляем туда следующую строчку:

```
UUID=360ca23a-300t-asdd-8c8e-a3xdf46dlpf1 /backup ext4 errors=remount-ro 0 1
```

или как вариант, выполнить команду:

```
echo "UUID=360ca23a-300t-asdd-8c8e-a3xdf46dlpf1 /backup ext4 errors=remount-ro 0 1" >> /etc/fstab
```

Этим мы объясняем системе, что при загрузке следует смонтировать раздел с определенным *UUID* как папку */backup* и файловой системой *ext4*.

Для того, чтобы примонтировать раздел прямо сейчас, можно выполнить следующую команду:

```
mount -t ext4 /dev/sdc1 /backup
```

На этом добавление жесткого диска в систему завершено.

5. Создаем пользователей

```
sudo adduser username1
sudo adduser username2
```

(в процессе создания будут созданы домашние каталоги и оболочка, а также запрос на задание пароля)

Объединяем пользователей в группу

создаем группу group1

```
sudo groupadd group1
```

добавляем пользователей в группу

```
sudo usermod -a -G group1 username
```

6. Создаем папки

```
sudo mkdir -p /fold/subfold/subsubfold
```

* - параметр -p создание папок по пути к конечной папки, если их нет

```
sudo mkdir -p /mnt/sdb1/Audio /mnt/sdb1/Files /mnt/sdb1/Ira /mnt/sdb1/Kate
/mnt/sdb1/Photo /mnt/sdb1/Torrent/Auto /mnt/sdc1/Films /mnt/sdc1/Mult
/mnt/sdc1/Serials
```

7. Меняем пользователей и права созданных папок

Пользователи

```
sudo chown username:group1 /fold /fold1/subfold
```

```
sudo chown da2001:family /mnt/sdb1/Audio /mnt/sdb1/Files /mnt/sdb1/Photo
/mnt/sdb1/Torrent /mnt/sdc1/Films /mnt/sdc1/Mult /mnt/sdc1/Serials
sudo chown ira:family /mnt/sdb1/Ira
sudo chown kate:family /mnt/sdb1/Kate
```

Права доступа к папкам и файлам

папки

```
sudo find /home/test -type d -exec chmod 755 {} \;
```

файлы

```
sudo find /home/test -type f -exec chmod 644 {} \;
```

8. Установка и настройка NFS

```
sudo apt install nfs-kernel-server nfs-common
```

Настройки сервера

Все настройки сервера хранятся в файле [/etc/exports](#). Открываем его на редактирование

```
sudo nano /etc/exports
```

и добавляем в конец файла строки вида (строк может быть произвольное количество):

```
/path/to/folder 192.168.1.0/24(rw, sync, no_subtree_check)
```

Настройки клиента

Проверим какие папки можем примонтировать, запросим список шар удаленного сервера

```
showmount -e 192.168.1.8
```

создадим папки, в которые будем монтировать

```
sudo mkdir -p /mnt/nfs/NAS/data1 /mnt/nfs/NAS/data2 /mnt/nfs/NAS/data3
```

в `fstab` по адресу `/etc/fstab`

```
sudo nano /etc/fstab
```

добавим отдельную запись для каждой папки

```
192.168.1.7:/media/data1 /mnt/nfs/NAS/data1 nfs auto 0 0
192.168.1.7:/media/data2 /mnt/nfs/NAS/data2 nfs auto 0 0
192.168.1.7:/media/data3 /mnt/nfs/NAS/data3 nfs auto 0 0
```

!!!ТАКОЕ МОНТИРОВАНИЕ НЕ ЗАРАБОТАЛО НА Debian 9.9 без DE. Установил систему с родным DE (Gnome) и все стало монтироваться при загрузке!!!

9. Установка и настройка Samba

```
sudo apt install samba
```

При помощи терминала сделаем резервную копию начального файла конфигурации:

```
sudo cp /etc/samba/smb.conf{, .bak}
```

Теперь можно отредактировать файл настроек `/etc/samba/smb.conf`,

```
sudo nano /etc/samba/smb.conf
```

Пример конфигурации:

```
##### Global Settings #####

[global]
netbios name = DSERV
obey pam restrictions = yes
dns proxy = no
pam password change = yes
workgroup = HOME
panic action = /usr/share/samba/panic-action %d
syslog = 0
log file = /var/log/samba/log.%m
server role = standalone server
passwd program = /usr/bin/passwd %u
map to guest = bad user
max log size = 1000
passdb backend = tdbsam
passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:* %n\n
*password\supdated\ssuccessfully* .
os level = 20
usershare allow guests = yes
security = user
wins support = no
printing = CUPS
printing name = CUPS

##### Share Definitions #####

[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = yes
```

```
guest ok = no
read only = yes
create mask = 0700
```

```
[print$]
comment = Printer Drivers
path = /var/lib/samba/printers
browseable = yes
read only = yes
guest ok = no
```

```
[Audio]
path = /mnt/sdb1/Audio
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

```
[Files]
path = /mnt/sdb1/Files
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

```
[Ira]
path = /mnt/sdb1/Ira
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

```
[Kate]
path = /mnt/sdb1/Kate
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

```
[Photo]
path = /mnt/sdb1/Photo
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

```
[Films]
```

```
path = /mnt/sdc1/Films
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
[Mult]
path = /mnt/sdc1/Mult
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family

[Serials]
path = /mnt/sdc1/Serials
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family

[Torrent]
path = /mnt/sdb1/Torrent
browseable = yes
read only = no
guest ok = no
create mask = 0644
directory mask = 0755
force group = family
```

Теперь надо разобраться с пользователями.

Samba использует пользователей которые уже есть в системе, возьмем для примера имя `user`, допустим что он уже есть в системе, надо внести его в базу данных SMB и назначить пароль для доступа к общим ресурсам, сделаем это командой:

```
sudo smbpasswd -a user
```

Вам будет предложено ввести пароль, пользователь будет добавлен в базу, теперь необходимо включить этого пользователя.

```
sudo smbpasswd -e user
```

Перезагружаем samba

```
sudo service smbd restart
```

10. Настройка transmission-daemon в Debian 9 (Stretch)

Для Linux существует замечательный легкий BitTorrent-клиент, который называется transmission-daemon. Работает он совершенно безотказно, но вот произвести его первоначальную настройку для многих новичков является нетривиальной задачей. Самое интересное, что некоторые моменты в настройках меняются от версии к версии как самого transmission-daemon, так и дистрибутива. В данном мануале приведу пример настройки transmission-daemon под Debian 9.

Устанавливаем transmission-daemon, если он у вас еще не установлен

```
sudo apt install transmission-daemon
```

Копируем конфигурационные файлы в директорию пользователя, от имени которого будет запускаться [transmission-daemon](#)

```
sudo cp -r /etc/transmission-daemon /home/da2001/.config/
```

Меняем права доступа

```
sudo chown -R da2001:da2001 /home/da2001/.config/transmission-daemon
```

В [/etc/init.d/transmission-daemon](#) меняем

```
USER=da2001
```

В [/etc/default/transmission-daemon](#) меняем

```
CONFIG_DIR="/home/da2001/.config/transmission-daemon"
```

Выполняем команду

```
sudo systemctl edit transmission-daemon.service
```

В открывшемся окне вписываем

```
[Service]
User=da2001
```

Сохраняем изменения и выходим.

Если у вас редактор по умолчанию Nano, то для того чтобы сохранить и выйти необходимо нажать Ctrl+O, затем Ctrl+X, если же Vi, то жмем Esc, затем набираем :wq! и жмем Enter.

Перезагружаемся.

В [/home/da2001/.config/transmission-daemon/settings.json](#) меняем:

если компьютер, на котором будет запускаться transmission-daemon имеет статический ip, то прописываем этот ip в строку

```
"bind-address-ipv4": "192.168.1.8",
```

сюда вписываем директорию, в которую будут загружаться торренты

```
"download-dir": "/mnt/sdb1/Torrent",
```

если хотим указать отдельную директорию для незавершенных загрузок, то меняем значение на true

```
"incomplete-dir-enabled": true,
```

и указываем директорию для незавершенных загрузок

```
"incomplete-dir": "/mnt/sdb1/Torrent/incomplete",
```

устанавливаем количество информации, выдаваемой в [/var/log/syslog](#), значение можно менять от 0 до 9, где 0 - отсутствие записей в логи

```
"message-level": 1,
```

чтобы обращаться к transmission-daemon через web-интерфейс, прописываем true, иначе оставляем false

```
"rpc-enabled": true,
```

сюда прописываем ip-адрес, который указали в поле ["bind-address-ipv4"](#)

```
"rpc-bind-address": "192.168.1.8",
```

если хотим, чтобы запрашивался пароль для доступа через web-интерфейс, то меняем на true, иначе оставляем false

```
"rpc-authentication-required": true,
```

если в предыдущем пункте выбрали true, то устанавливаем имя для подключения к web-интерфейсу

```
"rpc-username": "da2001",
```

в кавычки вписываем свой пароль, после перезагрузки конфигурации его значение будет хэшировано

```
"rpc-password": "63e518bf450175974b02403e657357326a76452f3Io4VSG2a",
```

можно поменять порт, по которому будет доступен web-интерфейс

```
"rpc-port": 1419,
```

чтоб web-интерфейс был доступен только с определенных ip и/или подсетей, то меняем на true

```
"rpc-whitelist-enabled": true,
```

в этой строке указываем необходимые ip и подсети через запятую и без пробела

```
"rpc-whitelist": "127.0.0.1,192.168.1.*",
```

для снижения фрагментации скачанных файлов на диске, резервируем место под них заранее

```
"preallocation": 2,
```

по-умолчанию стоит 1, меняем ее на 2.

Основные настройки на этом закончены. Но можно настроить еще всякие дополнительные функции, например, чтоб `transmission-daemon` подбирал .torrent-файлы из определенной директории и ставил их на закачку, для этого в конец конфигурационного файла дописываем строки

```
"watch-dir": "/mnt/sdb1/Torrent/Auto",
```

```
"watch-dir-enabled": true
```

причем не забываем поставить запятую в конец строки, после которой мы добавили эти строки.

Так же владельцем директории `/mnt/sdb1/Torrent` должен быть da2001 - юзер, от имени которого запускается `transmission-daemon`

```
sudo chown -R da2001:da2001 /mnt/sdb1/Torrent
```

Права на директорию `/mnt/sdb1/Torrent` и файлы в ней должны быть 775

```
sudo chmod -R 775 /mnt/sdb1/Torrent
```

Если Вы настроили автодобавление .torrent-файлов из определенной директории на закачку, то по умолчанию, после того, как это автодобавление произошло, файл автоматически переименовывается из foobar.torrent в foobar.torrent.added. Чтоб файлы .torrent.added автоматически удалялись, то можно добавить соответствующую задачу в cron, а можно в `/home/da2001/.config/transmission-daemon/settings.json` изменить строки на

```
"script-torrent-done-enabled": true,
```

```
"script-torrent-done-filename": "/usr/sbin/transmission-rm-added.sh",
```

затем создать файл `/usr/sbin/transmission-rm-added.sh`

```
sudo touch /usr/sbin/transmission-rm-added.sh
```

со следующим содержимым

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

```
rm /mnt/sdb1/Torrent/Auto/*.added
```

после чего владельцем файла `/usr/sbin/transmission-rm-added.sh` делаем юзера da2001

```
sudo chown da2001:da2001 /usr/sbin/transmission-rm-added.sh
```

ПРИМЕР файла настроек `/home/da2001/.config/transmission-daemon/settings.json`

```
{
  "alt-speed-down": 50,
  "alt-speed-enabled": false,
  "alt-speed-time-begin": 540,
  "alt-speed-time-day": 127,
  "alt-speed-time-enabled": false,
  "alt-speed-time-end": 1020,
  "alt-speed-up": 50,
  "bind-address-ipv4": "0.0.0.0",
  "bind-address-ipv6": "::",
  "blocklist-enabled": false,
  "blocklist-url": "http://www.example.com/blocklist",
  "cache-size-mb": 4,
  "dht-enabled": true,
  "download-dir": "/mnt/sdb1/Torrent",
  "download-limit": 100,
  "download-limit-enabled": 0,
  "download-queue-enabled": true,
  "download-queue-size": 5,
  "encryption": 1,
  "idle-seeding-limit": 30,
  "idle-seeding-limit-enabled": false,
  "incomplete-dir": "/home/da2001/Downloads",
  "incomplete-dir-enabled": false,
  "lpd-enabled": false,
  "max-peers-global": 200,
  "message-level": 1,
  "peer-congestion-algorithm": "",
  "peer-id-ttl-hours": 6,
  "peer-limit-global": 200,
  "peer-limit-per-torrent": 50,
  "peer-port": 51413,
  "peer-port-random-high": 65535,
  "peer-port-random-low": 49152,
  "peer-port-random-on-start": false,
  "peer-socket-tos": "default",
  "pex-enabled": true,
  "port-forwarding-enabled": true,
  "preallocation": 2,
  "prefetch-enabled": true,
  "queue-stalled-enabled": true,
  "queue-stalled-minutes": 30,
  "ratio-limit": 2,
  "ratio-limit-enabled": true,
  "rename-partial-files": true,
  "rpc-authentication-required": true,
  "rpc-bind-address": "0.0.0.0",
  "rpc-enabled": true,
  "rpc-host-whitelist": "",
  "rpc-host-whitelist-enabled": true,
  "rpc-password": "{90998958412461688d13fee3d914779e6a11974ddZge1k/q",
  "rpc-port": 9091,
  "rpc-url": "/transmission/",
  "rpc-username": "da2001",
  "rpc-whitelist": "127.0.0.1",
  "rpc-whitelist-enabled": false,
  "scrape-paused-torrents-enabled": true,
```

```
"script-torrent-done-enabled": false,
"script-torrent-done-filename": "",
"seed-queue-enabled": false,
"seed-queue-size": 10,
"speed-limit-down": 10000,
"speed-limit-down-enabled": true,
"speed-limit-up": 100,
"speed-limit-up-enabled": false,
"start-added-torrents": true,
"trash-original-torrent-files": false,
"umask": 18,
"upload-limit": 100,
"upload-limit-enabled": 0,
"upload-slots-per-torrent": 14,
"utp-enabled": true,
"watch-dir": "/mnt/sdb1/Torrent/Auto",
"watch-dir-enabled": true
}
```

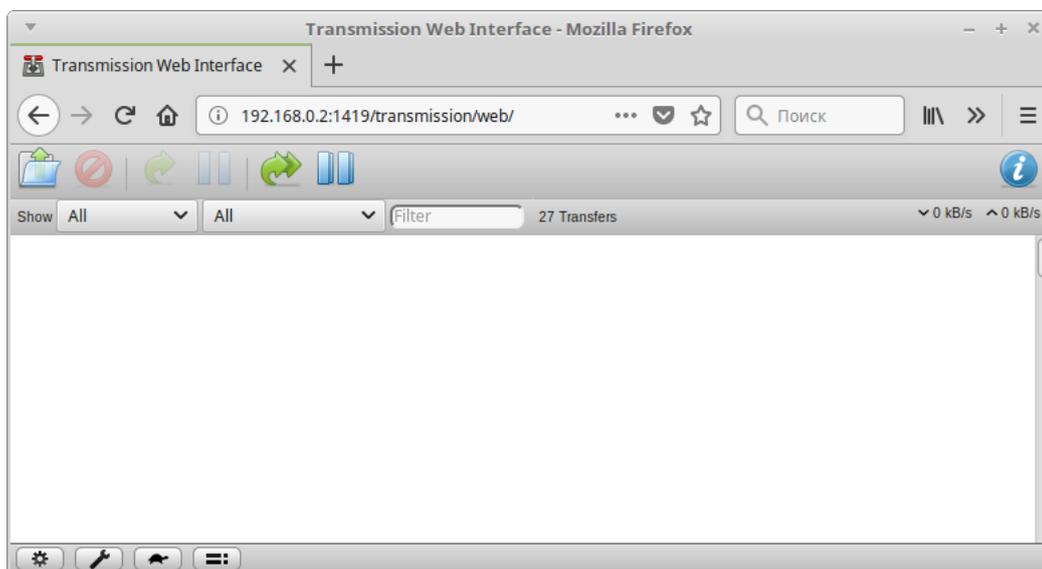
После изменения настроек **ОБЯЗАТЕЛЬНО** выполняем команду

```
sudo invoke-rc.d transmission-daemon reload
```

и только после этого

```
sudo service transmission-daemon restart
```

Все, можно пользоваться transmission-daemon-ом через web-интерфейс по адресу <http://127.0.0.1:1419> с компьютера на котором установлен transmission-daemon или по адресу <http://server-ip:1419> с любого компьютера в вашей локальной сети.



В левом нижнем углу можно поменять настройки transmission-daemon.

А еще transmission-daemon-ом можно управлять при помощи программы Transmission remote GUI <https://github.com/transmission-remote-gui/transgui>

```
sudo apt install transgui
```

11. Plex

Скачиваем с официального сайта установочный файл <https://www.plex.tv/media-server-downloads/#plex-media-server>

```
wget https://downloads.plex.tv/plex-media-server-new/1.15.5.994-4610c6e8d/debian/plexmediaserver_1.15.5.994-4610c6e8d_amd64.deb
```

установим пакет с помощью утилиты dpkg

```
sudo dpkg -i plexmediaserver_1.15.5.994-4610c6e8d_amd64.deb
```

если в процессе установки система скажет, что есть неразрешенные зависимости, вводим

```
sudo apt-get -f install
```

НО, лучше воспользоваться apt. Находясь в папке с пакетом, ввести

```
sudo apt install ./plexmediaserver_1.15.5.994-4610c6e8d_amd64.deb
```

при использовании apt все зависимости установятся автоматически.

В процессе установки, по адресу: etc/apt/sources.list.d/ создается файл `plexmediaserver.list` следующего содержания:

```
# When enabling this repo please remember to add the PlexPublic.Key into the apt
setup.
# wget -q https://downloads.plex.tv/plex-keys/PlexSign.key -O - | sudo apt-key
add -
#deb https://downloads.plex.tv/repo/deb/ public main
```

т.е., для добавления репозитория Plex выполним

```
wget -q https://downloads.plex.tv/plex-keys/PlexSign.key -O - | sudo apt-key add
-
```

и раскомментируем нижнюю строку (удалим #)

```
deb https://downloads.plex.tv/repo/deb/ public main
```

После установки удалим уже не нужный файл

```
rm plexmediaserver_1.15.5.994-4610c6e8d_amd64.deb
```

[плагин Кинопоиск для отображения инфо о фильмах с указанием рейтинга Кинопоиска](#)
устанавливаем git

```
sudo apt update && sudo apt install -y git
```

переходим в папку, где Plex хранит все плагины

```
cd /var/lib/plexmediaserver/Library/Application\ Support/Plex\ Media\
Server/Plug-ins/
```

скачиваем плагин, меняем владельца, перезагружаем Plex

```
sudo git clone https://github.com/amirotin/Kinopoisk.bundle.git
sudo chown -R plex:plex Kinopoisk.bundle/
sudo service plexmediaserver restart
```

перед добавлением библиотек с медиа, нужно настроить Plex, выставить агент Кинопоиск по-умолчанию и указать русский язык. Plex доступен по адресу: <http://server-ip:32400/web>

[перенос настроек Plex с другого сервера \(не проверено\)](#)

скопировать со старого Plex данные из

```
/var/lib/plexmediaserver/Library/Application Support/Plex Media Server/
```

12. Яндекс.Диск

Устанавливается и настраивается по официальной инструкции

<https://yandex.ru/support/disk/cli-clients.html>

Хотя при первоначальной настройке можно указать – запускать при старте системы, у меня это не заработало. Перезагружаемся и проверяем работу клиента

```
yandex-disk status
```

если видим сообщение типа

```
демон не запущен
```

читаем дальше:

Создадим юнит запуска в systemd

```
sudo nano /etc/systemd/system/yandex-disk@.service
```

вставим в него следующее

```
[Unit]
Description=Yandex.Disk
Requires=network.target
After=network.target

[Service]
Type=forking
User=%i
ExecStart=/usr/bin/yandex-disk start
RestartSec=1min
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Добавляем ссылку на автозагрузку юнита

```
sudo systemctl enable yandex-disk@da2001.service
```

13. Установка Webmin

Инструкция с официального сайта <http://www.webmin.com/deb.html>

Добавим репозиторий

```
sudo nano /etc/apt/sources.list
```

и вставим

```
deb https://download.webmin.com/download/repository sarge contrib
```

импортируем ключ GPG

```
wget http://www.webmin.com/jcameron-key.asc
apt-key add jcameron-key.asc
```

и, непосредственно, сама установка

```
sudo apt update
sudo apt install apt-transport-https && sudo apt install webmin
```

после этого открываем в браузере

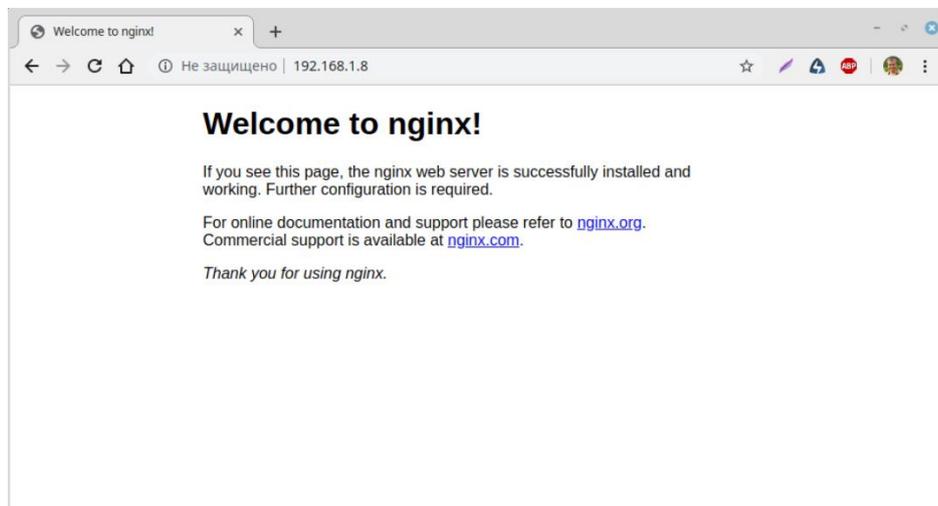
<https://server-ip:10000>

14. Установка NGINX + php-fpm + mysql

Сначала установим NGINX

```
sudo apt update
sudo apt install nginx
```

проверяем работу сервера: вводим в браузере <http://server-ip>



Следующим шагом устанавливаем php

```
sudo apt install php-fpm php-mysql
```

Для настройки nginx для работы с php создадим виртуальный хост и в его настройках сделаем соответствующие настройки.

Создаем папку, где будут храниться все файлы нашего сайта

```
sudo mkdir /var/www/da2001.su
```

Сделаем стартовую страницу сайта

```
sudo nano /var/www/da2001.su/index.html
```

с таким содержимым

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <title>Добро пожаловать на сайт da2001.su!</title>
  </head>
  <body>
    <h1>Отлично! Всё работает!</h1>
    da2001.su
  </body>
</html>
```

Затем создадим конфигурационный файл в каталоге [/sites-available/](#):

```
sudo nano /etc/nginx/sites-available/da2001.su
```

вставим в него следующий текст:

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/da2001.su;
    index index.php index.html index.htm;
    server_name da2001.su www.da2001.su;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ /\.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
        fastcgi_index index.php;
    }
}
```

```

fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
include fastcgi_params;
}
}

```

Теперь необходимо активировать созданную конфигурацию виртуального хоста. Для этого создадим на нее символическую ссылку в папке </etc/nginx/sites-enabled/>

```
sudo ln -s /etc/nginx/sites-available/da2001.su /etc/nginx/sites-enabled/
```

проверяем конфигурацию на наличие синтаксических ошибок:

```
sudo nginx -t
```

если видим

```

nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful

```

значит все в порядке, иначе нужно проверить конфигурацию виртуального хоста.

Когда все ошибки в конфигурации исправлены, перезагружаем nginx, чтобы изменения вступили в силу.

```
sudo service nginx restart
```

Проверим установку php, создадим файл проверки:

```
sudo nano /var/www/da2001.su/info.php
```

вставим в него

```

<?php
phpinfo();
?>

```

Сохраним. После этого в браузере переходим по адресу: <http://da2001.su/info.php>

Если видим, что-то подобное, значит все установлено и настроено верно

PHP Version 7.0.33-0+deb9u3	
System	Linux DNAS 4.9.0-9-amd64 #1 SMP Debian 4.9.168-1+deb9u2 (2019-05-13) x86_64
Build Date	Mar 8 2019 10:01:24
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/fpm
Loaded Configuration File	/etc/php/7.0/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/fpm/conf.d
Additional .ini files parsed	/etc/php/7.0/fpm/conf.d/10-mysqld.ini, /etc/php/7.0/fpm/conf.d/10-opcache.ini, /etc/php/7.0/fpm/conf.d/10-pdo.ini, /etc/php/7.0/fpm/conf.d/20-calendar.ini, /etc/php/7.0/fpm/conf.d/20-type.ini, /etc/php/7.0/fpm/conf.d/20-exif.ini, /etc/php/7.0/fpm/conf.d/20-fileinfo.ini, /etc/php/7.0/fpm/conf.d/20-ftp.ini, /etc/php/7.0/fpm/conf.d/20-gettext.ini, /etc/php/7.0/fpm/conf.d/20-iconv.ini, /etc/php/7.0/fpm/conf.d/20-json.ini, /etc/php/7.0/fpm/conf.d/20-mysqli.ini, /etc/php/7.0/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.0/fpm/conf.d/20-phar.ini, /etc/php/7.0/fpm/conf.d/20-posix.ini, /etc/php/7.0/fpm/conf.d/20-readline.ini, /etc/php/7.0/fpm/conf.d/20-shmop.ini, /etc/php/7.0/fpm/conf.d/20-sockets.ini, /etc/php/7.0/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.0/fpm/conf.d/20-sysvsem.ini, /etc/php/7.0/fpm/conf.d/20-sysvshm.ini, /etc/php/7.0/fpm/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS

Осталось установить MySQL

В Debian 9 по умолчанию в качестве альтернативы используется разработанный сообществом форк проекта MySQL — MariaDB.

```
sudo apt install mysql-server
```

После установки MySQL требует настройки.

Чтобы защитить установку, можно запустить простой скрипт безопасности, который удалит потенциально рискованные настройки. Иницилируйте скрипт:

```
sudo mysql_secure_installation
```

Система предложит ввести root пароль MySQL. Пока что у вас нет такого пароля, потому можно просто нажать Enter. Затем вам будет предложено выбрать пароль root. Нажмите у и выберите пароль.

На остальные вопросы можно нажать Y и Enter. Это удалит анонимных пользователей и тестовые базы данных, отключит удалённый root логин и обновит текущие настройки MySQL.

Автозагрузка NGINX

У меня NGINX не стартовал при загрузке системы, приходилось его запускать вручную. Начал разбираться и выяснилось, что он запускался раньше, чем поднималась сеть, поэтому NGINX закрывался с ошибкой. Необходимо притормозить его и заставить запускаться после того, как установится сетевое соединение.

Для этого в файле

[/lib/systemd/system/nginx.service](#)

в секцию UNIT добавляем

```
[Unit]
After=network-online.target remote-fs.target nss-lookup.target
Wants=network-online.target
```

при этом нужно закомментировать (#) то, что там было в After сохраняем, перезагружаемся.

На Debian 9.9 помогло

15. Letsencrypt

Для начала нужно установить клиент Certbot на сервер.

Certbot очень активно разрабатывается, потому его пакеты в стандартных репозиториях систем быстро устаревают. Однако в Debian 9 можно добавить backports репозиторий, где можно найти актуальные версии пакетов.

Если backports репозиторий не был добавлен при первоначальной настройке сервера, сделаем это сейчас:

```
sudo nano /etc/apt/sources.list
```

Добавьте зеркало репозитория в конец файла:

```
deb http://deb.debian.org/debian stretch-backports main contrib non-free
deb-src http://deb.debian.org/debian stretch-backports main contrib non-free
```

Сохраните и закройте файл.

Обновите индекс пакетов:

```
sudo apt update
```

Теперь установите пакет certbot для Nginx:

```
sudo apt install python-certbot-nginx -t stretch-backports
```

Настройка Certbot.

Регистрация в Let's Encrypt, для нового сервера это нужно сделать лишь один раз:

```
sudo certbot register --email ваша@эл.почта
```

Подправим конфиг

```
sudo nano /etc/letsencrypt/cli.ini
```

вставим

```
authenticator = webroot
webroot-path = /var/www/letsencrypt
post-hook = service nginx reload
text = True
```

Каталог [/var/www/letsencrypt](#) необходимо создать:

```
sudo mkdir /var/www/letsencrypt
```

Чтобы с правами доступа точно все было в порядке:

```
sudo chown -R www-data /var/www/letsencrypt && sudo chmod -R 750 /var/www/letsencrypt
```

Для удобства настройки Nginx создадим пару файлов. Будем их подключать потом в нужных местах конфигов, чтобы не писать одно и тоже сто раз. Первый [/etc/nginx/inc_letsencrypt](#) — необходим для обновления SSL-сертификатов:

```
sudo nano /etc/nginx/inc_letsencrypt
```

Вставим:

```
location ~ /.well-known {
    location ~ /.well-known/acme-challenge/(.*) {
        auth_basic off;
        default_type "text/plain";
        root /var/www/letsencrypt;
    }
}
```

Второй — [/etc/nginx/inc_ssl_da2001](#) — включает все необходимое для работы https на доменах сайта da2001 (в будущем попробую включить сюда и da2001.ru):

```
sudo nano /etc/nginx/inc_ssl_da2001
```

```
ssl on;
ssl_certificate /etc/letsencrypt/live/da2001.su/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/da2001.su/privkey.pem;
ssl_session_timeout 1d;
ssl_session_cache shared:SSL:50m;
ssl_session_tickets off;

ssl_dhparam /etc/letsencrypt/live/da2001.su/dh.pem;

ssl_protocols TLSv1.2;
ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256';
ssl_prefer_server_ciphers on;

#add_header Strict-Transport-Security "max-age=31536000; #includeSubDomains"
always;

ssl_stapling on;
ssl_stapling_verify on;
ssl_trusted_certificate /etc/letsencrypt/live/da2001.su/chain.pem;
resolver 8.8.8.8;
```

Немного расширенный конфиг сайта [/etc/nginx/sites-available/da2001.su](#):

```
sudo nano /etc/nginx/sites-available/da2001.su
```

```
# Вспомогательный блок.
```

```

server {
    listen 80;
    listen [::]:80;
    server_name da2001.su www.da2001.su;

    # Для обновления сертификатов.
    include inc_letsencrypt;

    # Редирект с http на https.
    location / {
        return 301 https://da2001.su$request_uri;
    }
}

# Блок для редиректа с дополнительных доменов.
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name www.da2001.su;

    # Подключаем все необходимое для ssl.
    include inc_ssl_da2001;

    # Редирект на основной домен.
    return 301 https://da2001.su$request_uri;
}

# Основной блок.
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name da2001.su;

    # Подключаем все необходимое для ssl.
    include inc_ssl_da2001;

    root /var/www/da2001.su/;
    index index.php index.html index.htm;

    add_header X-Frame-Options "SAMEORIGIN";
    add_header x-xss-protection "1; mode=block" always;
    add_header X-Content-Type-Options "nosniff" always;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ /\.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    }

    access_log /var/log/nginx/pctuner_access.log;
    error_log /var/log/nginx/pctuner_error.log;
}

```

Изначально нужные сертификаты (файлы) отсутствуют, поэтому Nginx не примет конфиг в таком виде. До того момента пока сертификаты не будут получены необходимо закомментировать подключение файла `inc_ssl_da2001`:

```
# include inc_ssl_da2001;
```

Пример теста получения сертификата:

```
sudo certbot certonly --dry-run -d da2001.su -d www.da2001.su
```

Один сертификат будет на 2 домена.

В одном сертификате имеет смысл объединять все домены, имеющие отношение к одному и тому же сайту. Больше доменов добавляются аналогично:

```
certbot certonly --dry-run -d da2001.su -d www.da2001.su -d da2001.ru -d www.da2001.ru
```

Здесь параметр `--dry-run` отвечает за запуск в тестовом режиме. Это позволяет убедиться, что с настройкой сервера все хорошо и сертификат точно будет получен. Дело в том, что есть ограничение на количество полученных сертификатов за некий промежуток времени, поэтому обязательно вначале делайте тестирование. Если все в порядке, достаточно убрать этот параметр, и тогда сертификат будет получен уже реально.

В случае успешного теста будет ответ:

```
- The dry run was successful.
```

Кроме сертификата необходимо сгенерировать ключ:

```
sudo openssl dhparam -outform PEM -out /etc/letsencrypt/live/da2001.su/dh.pem 2048
```

после проверяем конфигурацию nginx на наличие ошибок

```
sudo nginx -t
```

если ошибок нет

```
sudo service nginx restart
```

На этом все. Включаем обратно подключение файла `inc_ssl_da2001` в конфиге и перезапускаем Nginx.

В будущем если потребуется добавить в сертификат еще один домен, то просто используем выше приведенную команду, указав в ней все нужные домены, в том числе старые и новые, программа сама разберется, что с этим делать.

Для автоматического обновления сертификатов в будущем ничего делать не нужно, файл `/etc/cron.d/certbot` с необходимыми для этого настройками уже был создан. Программа автоматически запускается дважды в сутки. Она видит полученные сертификаты всех сайтов на сервере, при необходимости обновляет их.

	Сервис	Хост
1	da2001.ru	da2001.ru
2		www.da2001.ru
3	da2001.su	da2001.su
4		www.da2001.su
5	PLEX	plex.da2001.ru
6	Transmission	torrent.da2001.ru
7	QBittorrent	qbt.da2001.ru
8	Nextcloud	cloud.da2001.ru
9	Syncthing	sync.da2001.ru
10	OnlyOffice	office.da2001.ru
11	Webmin	webmin.da2001.ru
12	Proxmox	pve.da2001.ru
13	Webmail	webmail.da2001.ru

```
sudo certbot certonly --dry-run -d da2001.su -d www.da2001.su -d da2001.ru -d
www.da2001.ru -d torrent.da2001.ru -d pve.da2001.ru -d sync.da2001.ru -d
webmin.da2001.ru -d plex.da2001.ru -d cloud.da2001.ru -d office.da2001.ru -d
qbt.da2001.ru -d webmail.da2001.ru
```

16. Qbittorrent-tox

Qbittorrent-tox – версия торрент-клиента без GUI, только консоль и WEB. Присутствует в стандартном репозитории.

Устанавливаем

```
sudo apt install qbittorrent-nox
```

Создаем сервис для автозагрузки при старте системы

```
sudo nano /etc/systemd/system/qbittorrent.service
```

с таким содержимым

```
[Unit]
Description=qBittorrent Daemon Service
After=network.target

[Service]
User=da2001
ExecStart=/usr/bin/qbittorrent-nox
ExecStop=/usr/bin/killall -w qbittorrent-nox

[Install]
WantedBy=multi-user.target
```

где `User=da2001` – пользователь, от имени которого будет стартовать сервис

чтобы новый сервис попал в список загрузки, обновляем службу автозагрузки

```
sudo systemctl daemon-reload
```

теперь необходимо создать папку, в которой будут храниться настройки. Папка создается в домашней директории пользователя, от имени которого стартует сервис

```
mkdir /home/da2001/.config/qBittorrent/
```

Первый запуск необходимо выполнить из консоли вручную, т.к. нужно подтвердить согласие на использование.

```
qbittorrent-nox
```

ВИДИМ

*** Официальное уведомление ***

qBittorrent – это программа для обмена файлами. При запуске торрента информация, связанная с ним, становится доступна другим пользователям посредством раздачи. Вы несёте персональную ответственность за всю информацию, которой делитесь с другими пользователями.

Никаких дальнейших уведомлений выводиться не будет.

Нажмите 'y', чтобы принять и продолжить...

Нажимаем `y`

Теперь доступен Web-интерфейс по адресу: <http://127.0.0.1:8080>

Логин: admin, пароль: adminadmin. Сразу необходимо изменить в настройках.

И последнее – включаем автозагрузку сервиса

```
sudo systemctl enable qbittorrent
```

После перезагрузки сервис должен стартовать автоматом.

Если перезагружать лень, можно стартовать вручную

```
sudo systemctl start qbittorrent
```

Проверить состояние

```
sudo systemctl status qbittorrent
```