

Настройка сети, запуск Putty

Настраиваем сеть, то есть прописываем IP-адрес, маску, шлюз, DNS. Всё это можно прописать через специальный файл, который находится в каталоге **/etc/netplan**

1. Заходим в этот каталог:

```
cd /etc/netplan
```

```
ubs@ubspc:~$ cd /etc/netplan
ubs@ubspc:/etc/netplan$
```

2. Посмотрим содержимое этого каталога:

```
ls -a
```

Там есть всего один файл — **01-netcfg.yaml**

```
ubs@ubspc:/etc/netplan$ ls -a
.  ..  01-netcfg.yaml
```

3. Открываем этот файл в текстовом редакторе **Nano** (чтобы делать там изменения, обязательно используем команду **sudo**):

```
sudo nano 01-netcfg.yaml
```

```
ubs@ubspc:/etc/netplan$ sudo nano 01-netcfg.yaml
[sudo] ■■■■■■ ■■■ ubs:
```

4. Вот так выглядит первоначальное содержимое файла:

```
GNU nano 4.8                               Serv1 (Intel(r) AMT KVM) - VNC Vie
01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eno1:
      dhcp4: yes
```

5. Необходимо внести изменения в этот файл. Во-первых, отключаем динамические адреса, прописав **dhcp4: no** (вместо **yes**).

Потом дописываем строки, чтобы получилось вот так:

```
GNU nano 4.8                               Serv1 (Intel(r) AMT KVM) - VNC Viewe
                                           01-netcfg.yaml
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eno1:
      dhcp4: no
      addresses: [191.221.70.147/23]
      gateway4: 91.221.70.1
      nameservers:
        addresses: [91.221.70.1, 91.221.71.11]
```

Все IP-адреса конечно же проставляем те, которые нужны для работы вашего сервера, а не написанные здесь.

Внимание! Декларативный язык YAML, с помощью которого в данном случае пишутся настройки сети, очень чувствителен к пробелам, поэтому команды записываем чётко с такими же отступами, как показаны на скрине.

Расшифровка команд:

network: (блок начало конфигурации)

version: (версия YAML)

renderer: (указываем сетевой менеджер который будем использовать, в данном случае networkd)

ethernets: (этот блок говорит о том, что будем конфигурировать ethernet протокол)

eno1: (это имя адаптера, который будем конфигурировать)

dhcp4: (включает/отключает раздачу динамических IPv4-адресов)

addresses: (здесь пишем свой статический IP, через слеш — маска подсети)

gateway4: (адрес шлюза, пишем без квадратных скобок)

nameservers: (настройка серверов имен, DNS)

addresses: (адреса DNS, все адреса в квадратных скобках пишутся через запятую и пробел).

6. Сохраняем файл. Для этого можно сначала нажать **Ctrl+X** для выхода из программы, и на запрос ответить буквой **Y** (то есть, yes, сохранить), а потом нажать **Enter**.

В VNC Viewer русская кодировка может не отображаться, и будут видны квадраты — ничего страшного, просто выполняем последовательность комбинаций клавиш. Не забываем перевести клавиатуру на английскую раскладку.



После нажатия **Enter** файл сохранится и закроется.

7. Желательно снова набрать команду

```
sudo nano 01-netcfg.yaml
```

чтобы проверить, всё ли правильно сохранилось. После этого просто закрываем файл клавишами **Ctrl+X**

8. Для проверки конфигурационного файла вводим:

```
sudo netplan --debug generate
```

Если всё хорошо, получим примерно такой вывод:

```
ubs@ubspc:/etc/netplan$ sudo netplan --debug generate
DEBUG:command generate: running ['/lib/netplan/generate']
** (generate:1293): DEBUG: 14:37:42.344: starting new processing pass
** (generate:1293): DEBUG: 14:37:42.344: We have some netdefs, pass them through a final round of validation
** (generate:1293): DEBUG: 14:37:42.345: eno1: setting default backend to 1
** (generate:1293): DEBUG: 14:37:42.345: Configuration is valid
** (generate:1293): DEBUG: 14:37:42.345: Generating output files..
** (generate:1293): DEBUG: 14:37:42.345: openswitch: definition eno1 is not for us (backend 1)
** (generate:1293): DEBUG: 14:37:42.345: NetworkManager: definition eno1 is not for us (backend 1)
(generate:1293): GLib-DEBUG: 14:37:42.346: posix_spawn avoided (fd close requested)
(generate:1293): GLib-DEBUG: 14:37:42.348: posix_spawn avoided (fd close requested)
ubs@ubspc:/etc/netplan$
```

9. Чтобы применить настройки, пишем такую команду:

```
sudo netplan --debug apply
```

Если всё нормально, получим примерно такой вывод (на скрине представлена только часть):

```
- 91.221.70.147/23
dhcp4: false
gateway4: 91.221.70.1
nameservers:
  addresses:
    - 91.221.70.1
    - 91.221.71.1
renderer: networkd
version: 2

DEBUG:Link changes: {}
DEBUG:netplan triggering .link rules for lo
DEBUG:netplan triggering .link rules for eno1
** (process:1297): DEBUG: 14:38:11.778: starting new processing pass
** (process:1297): DEBUG: 14:38:11.778: We have some netdefs, pass them through a final round of validation
** (process:1297): DEBUG: 14:38:11.779: eno1: setting default backend to 1
** (process:1297): DEBUG: 14:38:11.779: Configuration is valid
DEBUG:eno1 not found in {}
DEBUG:Merged config:
network:
  ethernets:
    eno1:
      addresses:
        - 91.221.70.147/23
      dhcp4: false
      gateway4: 91.221.70.1
      nameservers:
        addresses:
          - 91.221.70.1
          - 91.221.71.1
      renderer: networkd
      version: 2
ubs@ubspc:/etc/netplan$
```

10. Запускаем службу sshd:

```
sudo systemctl start sshd
```

11. Проверяем статус службы sshd:

```
sudo systemctl status sshd
```

```
ubspc:~$ sudo systemctl status sshd
■ ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-31 14:17:40 MSK; 27min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1008 (sshd)
      Tasks: 1 (limit: 18949)
     Memory: 3.9M
    CGroup: /system.slice/ssh.service
            └─1008 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

■   31 14:17:40 ubspc systemd[1]: Starting OpenBSD Secure Shell server...
■   31 14:17:40 ubspc sshd[1008]: Server listening on 0.0.0.0 port 22.
■   31 14:17:40 ubspc sshd[1008]: Server listening on :: port 22.
■   31 14:17:40 ubspc systemd[1]: Started OpenBSD Secure Shell server.
■   31 14:40:28 ubspc sshd[1351]: Unable to negotiate with 218.92.0.26 port 26196: no matching key exchange method found. Their
■   31 14:43:39 ubspc sshd[1450]: Invalid user blank from 81.82.179.5 port 56356
■   31 14:43:39 ubspc sshd[1450]: pam_unix(sshd:auth): check pass; user unknown
■   31 14:43:39 ubspc sshd[1450]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=81.82
■   31 14:43:41 ubspc sshd[1450]: Failed password for invalid user blank from 81.82.179.5 port 56356 ssh2
■   31 14:43:42 ubspc sshd[1450]: Connection closed by invalid user blank 81.82.179.5 port 56356 [preauth]
lines 1-21/21 (END)
```

Для выхода в ожидающую ввода командную строку нажимает **Ctrl+C**

12. Меняем правила и открываем порты:

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow OpenSSH
```

```
sudo ufw allow ssh
```

```
sudo ufw allow 22
```

```
sudo ufw allow 80/tcp
```

```
sudo ufw allow 443/tcp
```

```
sudo ufw allow 10000/udp
```

```
sudo ufw allow 5349/tcp
```

```
sudo ufw enable
```

13. Смотрим, что получилось:

```
sudo ufw status numbered
```

Вывод в консоли должен быть таким:

```
ubs@ubspc:~$ sudo ufw status numbered
■■■■■■■■: ■■■■■■■■

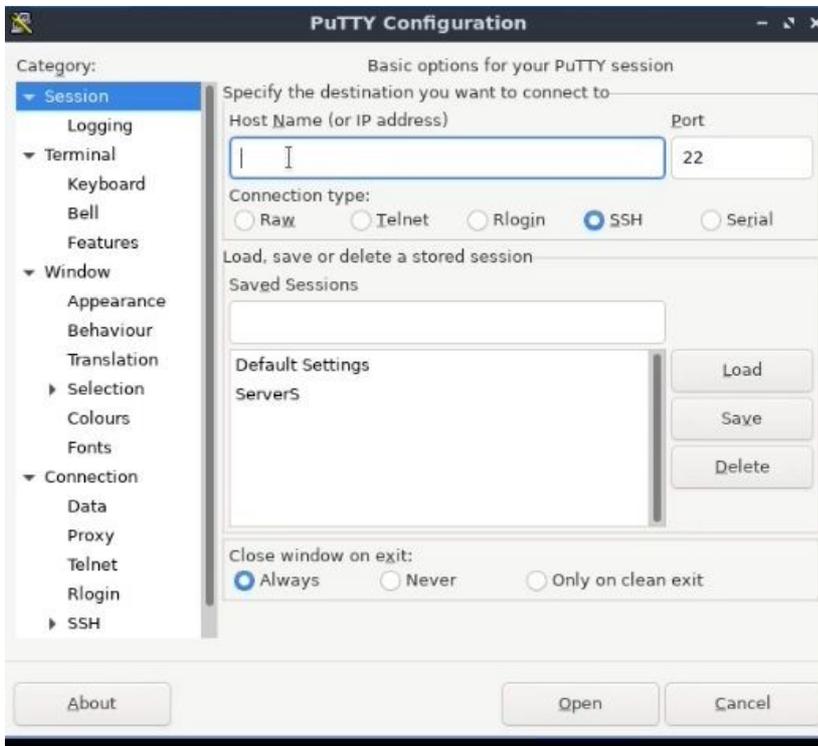
  ■                ■■■■■■■■  ■■
  --                -
[ 11] OpenSSH          ALLOW IN  Anywhere
[ 21] 22/tcp          ALLOW IN  Anywhere
[ 31] 22              ALLOW IN  Anywhere
[ 41] 80/tcp          ALLOW IN  Anywhere
[ 51] 443/tcp         ALLOW IN  Anywhere
[ 61] 10000/udp       ALLOW IN  Anywhere
[ 71] 5349/tcp        ALLOW IN  Anywhere
[ 81] OpenSSH (v6)   ALLOW IN  Anywhere (v6)
[ 91] 22/tcp (v6)    ALLOW IN  Anywhere (v6)
[101] 22 (v6)        ALLOW IN  Anywhere (v6)
[111] 80/tcp (v6)    ALLOW IN  Anywhere (v6)
[121] 443/tcp (v6)   ALLOW IN  Anywhere (v6)
[131] 10000/udp (v6) ALLOW IN  Anywhere (v6)
[141] 5349/tcp (v6)  ALLOW IN  Anywhere (v6)

ubs@ubspc:~$ _
```

14. Перезагружаем компьютер:

```
sudo reboot
```

15. Запускаем на своём компьютере ранее установленную и настроенную на удалённый сервер программу Putty:



16. Вводим имя пользователя и системный пароль:

A terminal window with a dark background. The text 'login as: uhs' is displayed on the first line. The second line shows 'uhs@media3.nbics.net's password:' followed by a green cursor block.

```
login as: uhs
uhs@media3.nbics.net's password: █
```

Доступ через Putty удобнее, чем через VNC Viewer, так как Putty поддерживает буфер обмена между сервером и своим компьютером, а также поддерживает русскую кодировку текста.

17. Установим apt-transport-https:

```
sudo apt install apt-transport-https
```