

# ????

- [Создание программного роутера Debian](#)

# ????????? ????????????????

## ????????? Debian

За основу взята [данная статья](#).

Все действия выполняются на виртуальной машине в Proxmox VE.

Сначала требуется чтобы у виртуальной машины, которая будет выступать в роли роутера, было минимум два сетевых интерфейса, в данном случае они представляют собой два Linux Bridge, в качестве системы виртуализации используется Proxmox. Первый мост – стандартный vmbr0, на нем будет внешняя сеть роутера, второй мост vmbr1 – на нем будет внутренняя сеть роутера, в настройках сетевых устройств Proxmox настроен CIDR: 10.0.1.0/24

После установки Debian в виртуалку настраиваем сетевые интерфейсы:

В данном примере система имеет два сетевых интерфейса: enp6s18 и enp6s19.

enp6s18 – это vmbr0 и будет внешней сетью

enp6s19 – это vmbr1 и будет внутренней сетью.

Редактируем файл /etc/networking/interfaces, содержимое должно иметь примерно следующий вид:

```
auto enp6s18
iface enp6s18 inet static
address 192.168.1.100
netmask 255.255.255.0
gateway 192.168.1.1

auto enp6s19
iface enp6s19 inet static
address 10.0.1.1
netmask 255.255.255.0
```

Далее перезапускаем сетевую службу:

```
sudo systemctl restart networking.service
```

????????? ???????? ???????????

Создадим каталог /etc/firewall и скрипт firewall.sh в нём:

```
sudo mkdir /etc/firewall
sudo nano /etc/firewall/iptables.sh
```

В нем следует создать следующее содержимое:

```
#!/bin/sh

sysctl -w net.ipv4.ip_forward=1

iptables -F
iptables -t nat -A POSTROUTING -o enp6s18 -j MASQUERADE
```

Делаем скрипт исполняемым:

```
sudo chmod +x /etc/firewall/iptables.sh
```

????????? ???????? rc.local

Создаем сервис для службы:

```
sudo nano /etc/systemd/system/rc-local.service
```

В него добавляем следующий код:

```
[Unit]
Description=/etc/rc.local
ConditionPathExists=/etc/rc.local
[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99
[Install]
WantedBy=multi-user.target
```

Создадим файл /etc/rc.local

```
sudo nano /etc/rc.local
```

Помещаем туда следующий код:

```
#!/bin/sh -e

/etc/firewall/iptables.sh

exit 0
```

Делаем его исполняемым:

```
sudo chmod +x /etc/rc.local
```

## ????????????? dnsmasq

Устанавливаем:

```
sudo apt install dnsmasq
```

Открываем конфиг:

```
sudo nano /etc/dnsmasq.conf
```

Добавляем в конец файла следующее:

```
bind-interfaces
domain-needed
bogus-priv
interface=enp6s19
resolv-file=/etc/resolv.conf
dhcp-range=10.0.1.154,10.0.1.254,24h
cache-size=150
```

Этим самым мы установили и сконфигурировали DHCP-сервер и добавили диапазон из 100 адресов с 10.0.1.154 до 10.0.1.254 в качестве адресов, которые будут назначаться автоматически, значения можно поменять на свой вкус.

В качестве интерфейса был назначен enp6s19, который соответствует локальной сети роутера и подключен к мосту vmbri1!

Перезагружаем виртуалку: `sudo reboot now`

Проверяем работу и радуемся.