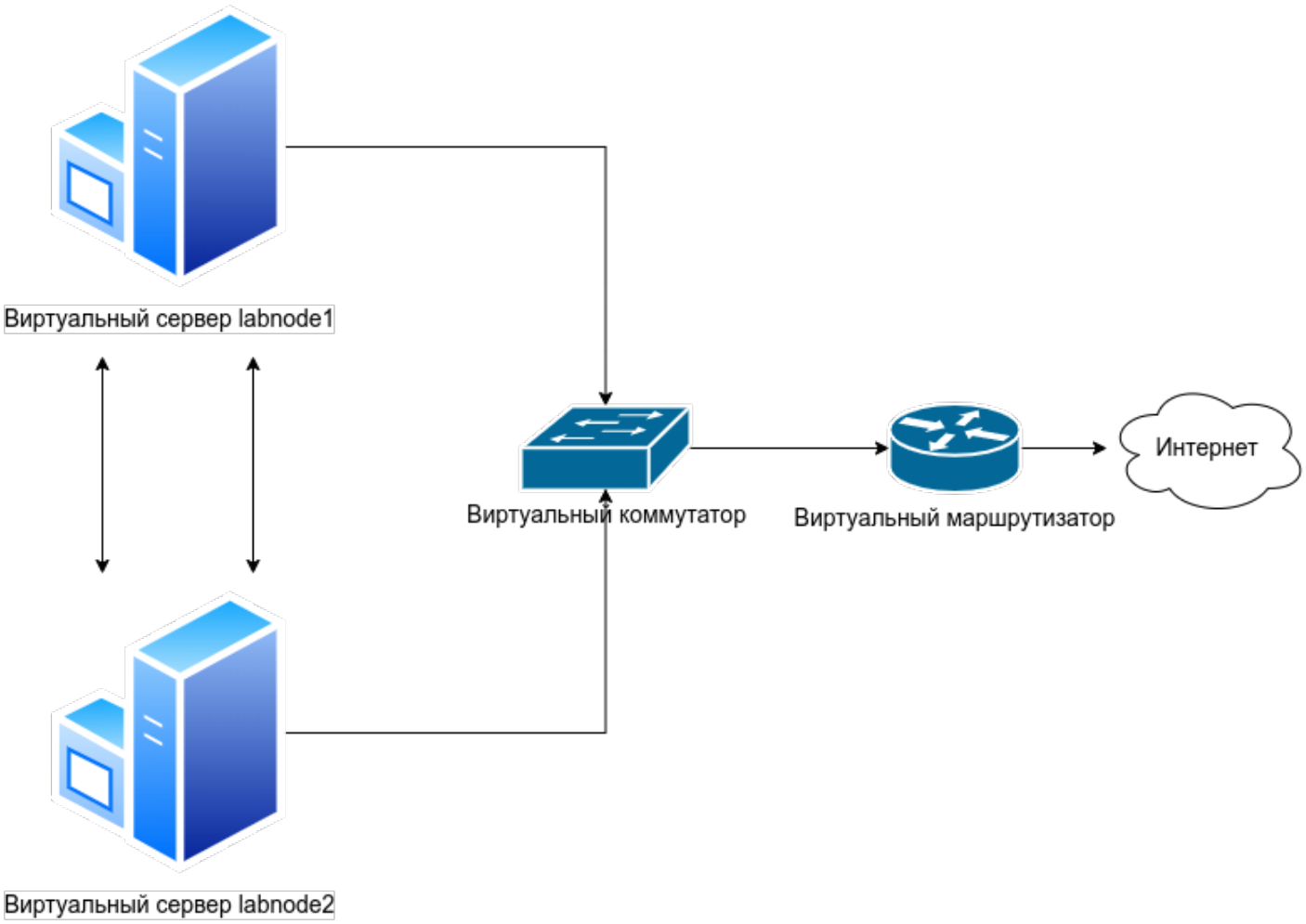
Работа с сетевой подсистемой Linux. Часть 1

Схема виртуального лабораторного стенда



Задание 0. Построение стенда

1. Создать 2 виртуальные сети:

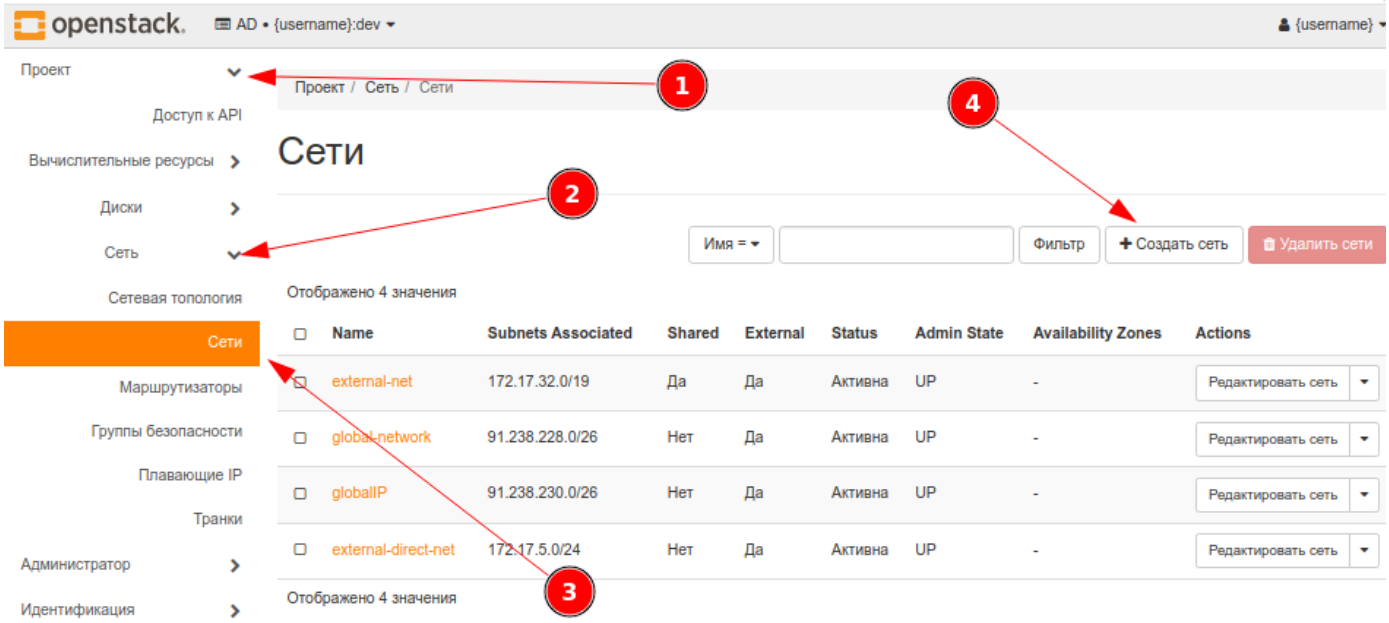
1. lab_net1
2. lab_net2

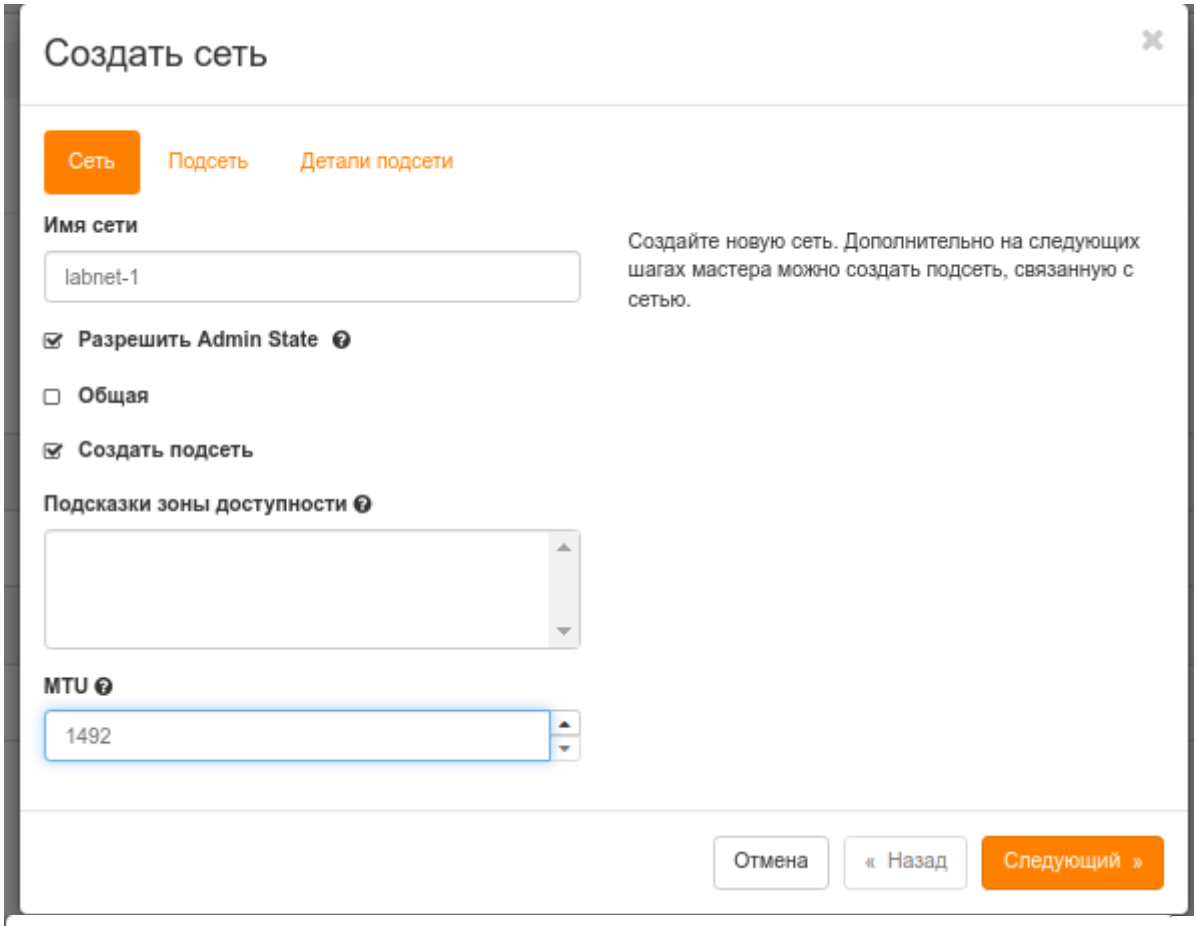
Название сети/подсети	Сетевой адрес	mtu
labnet-1	10.0.12.0/24	1492
labnet-2	10.0.12.0/24	1492

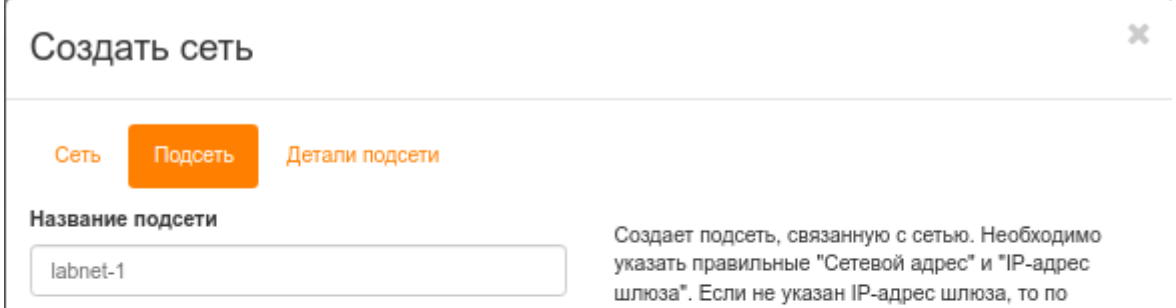
> При создании подсети необходимо выбрать пункт "Запретить шлюз"

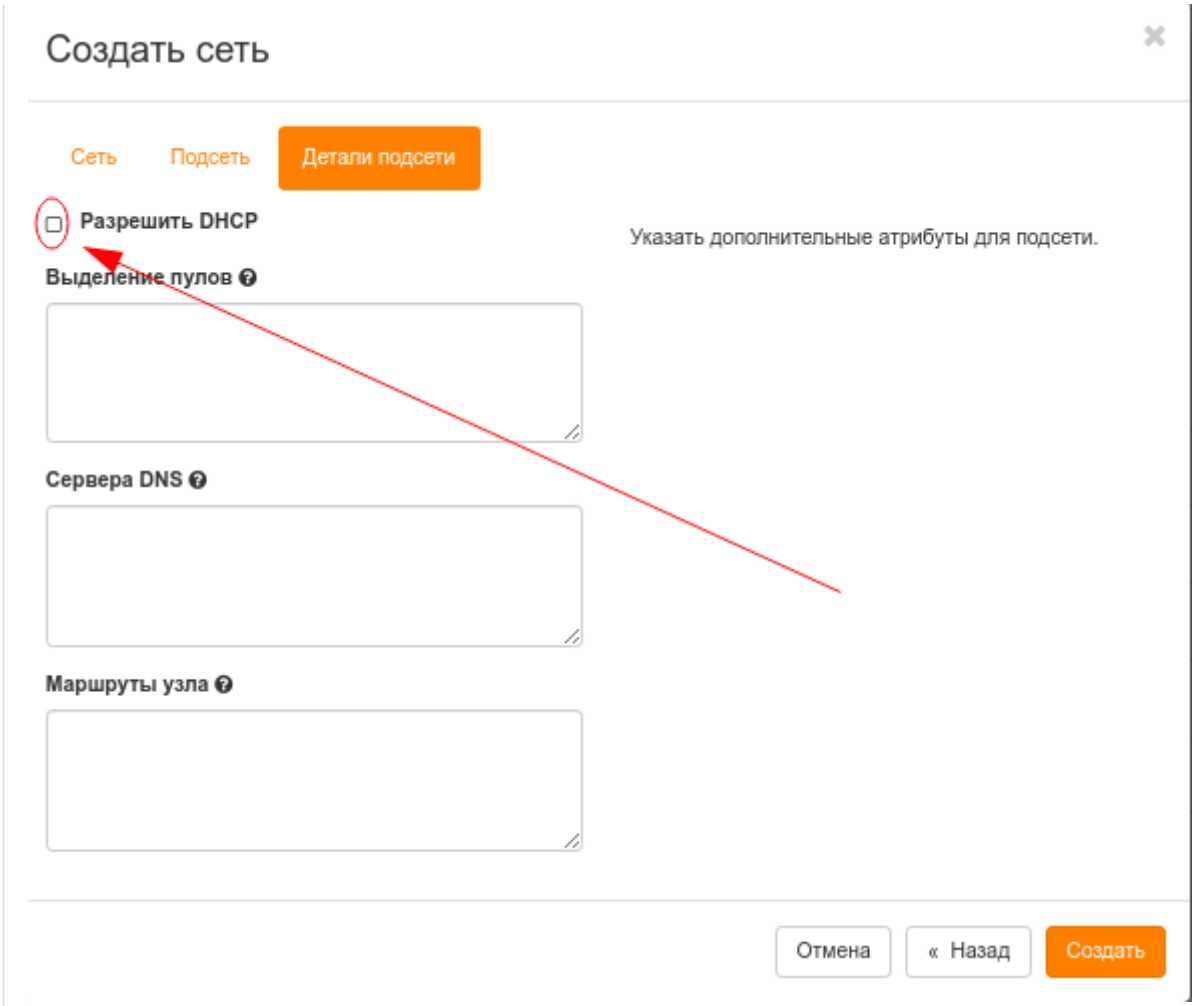
Пример создания сетей

Для создания сети нужно перейти в сети:







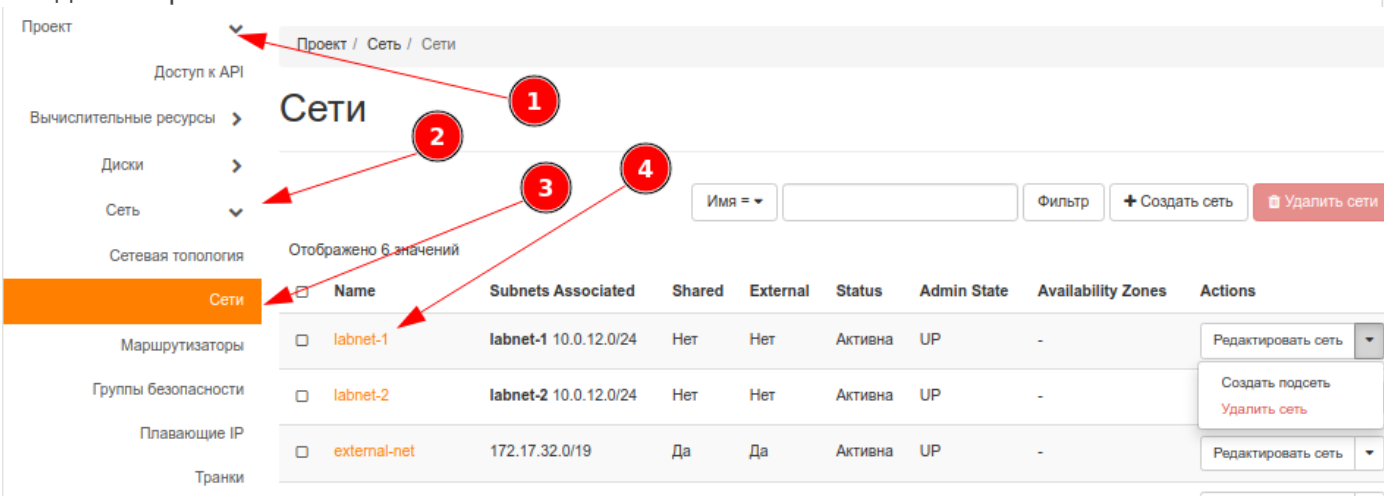


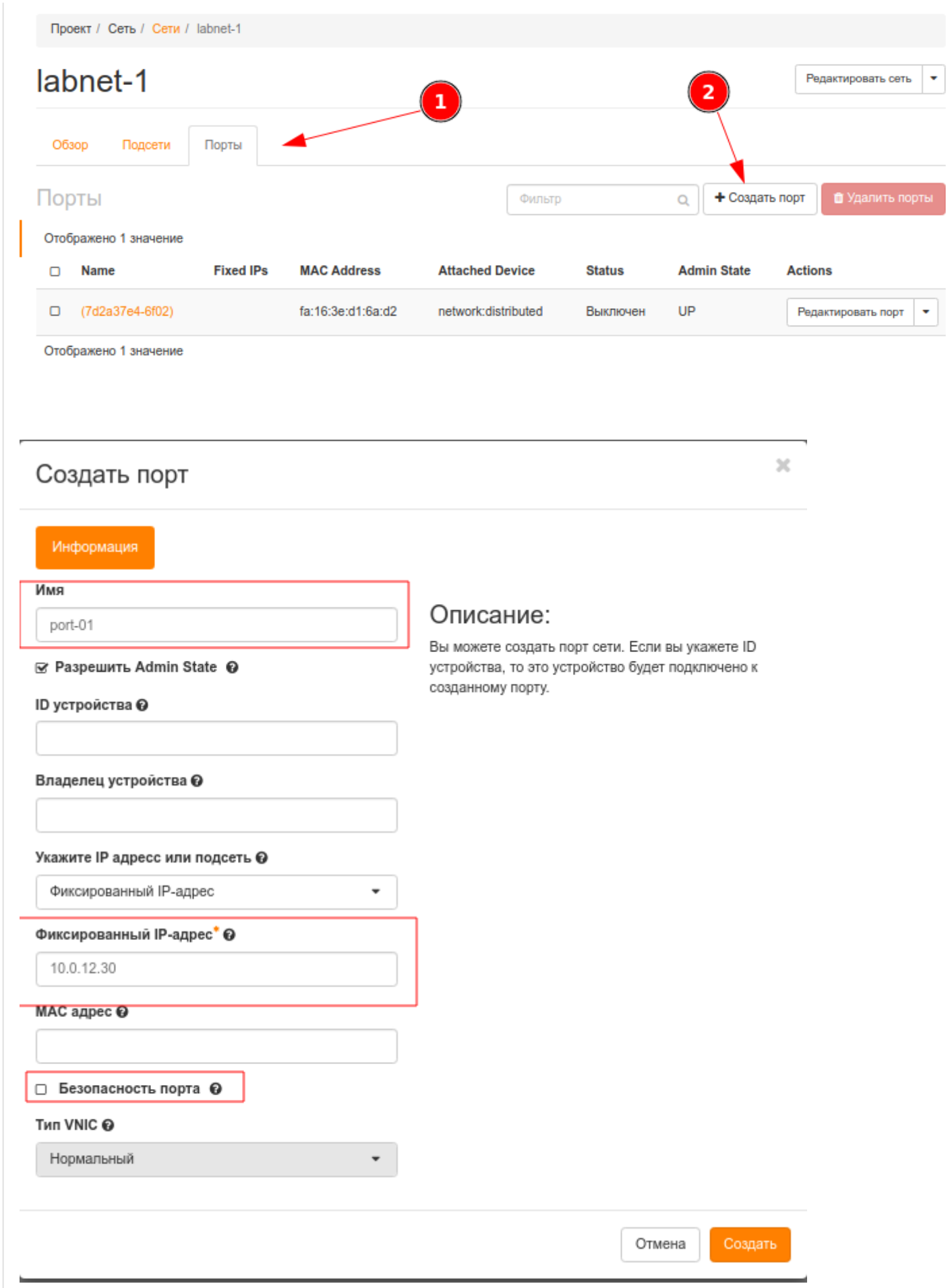
Создать порты для дальнейшего использования

название сети	Название порта	ip адрес	Безопасность порта
labnet-1	port-1-1	10.0.12.20	Отключена
labnet-1	port-1-2	10.0.12.30	Отключена
labnet-2	port-2-1	10.0.12.21	Отключена
labnet-2	port-2-2	10.0.12.31	Отключена

Пример создания порта

Создать порты:



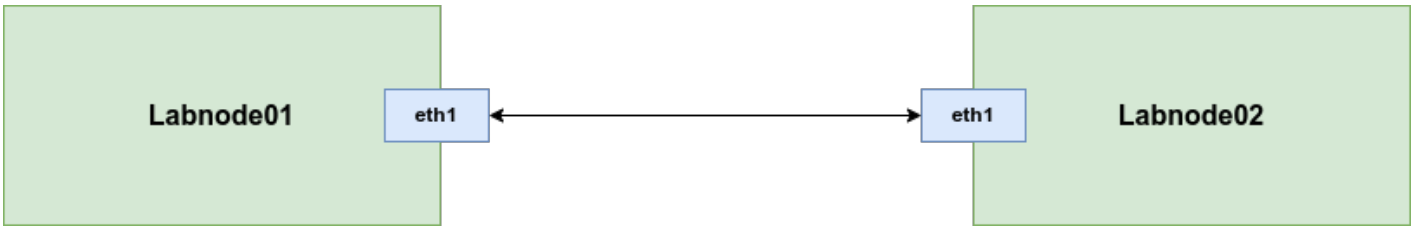

Проект / Сеть / Сети / labnet-1
labnet-1
Редактировать сеть
1
2
Обзор
Подсети
Порты
Порты
Фильтр
+ Создать порт
Удалить порты
Отображено 1 значение
Name
Fixed IPs
MAC Address
Attached Device
Status
Admin State
Actions
(7d2a37e4-6f02)
fa:16:3e:d1:6a:d2
network:distributed
Выключен
UP
Редактировать порт
Отображено 1 значение
Создать порт
Информация
Имя
port-01
Разрешить Admin State
ID устройства
Владелец устройства
Укажите IP адресс или подсеть
Фиксированный IP-адрес
Фиксированный IP-адрес*
10.0.12.30
MAC адрес
Безопасность порта
Тип VNIC
Нормальный
Описание:
Вы можете создать порт сети. Если вы укажете ID устройства, то это устройство будет подключено к созданному порту.
Отмена
Создать

2. Создать виртуальные машины для работы

Название виртуальной машины	Источник	Тип инстанса	Сети	Сетевые порты
labnode-1	Образ-CentOS-7:RECSDS	small	external-net	port-1-1,port-2-1
labnode-2	Образ-Ubuntu-server20.04	small	external-net	port-1-2,port-2-2

Задание 1. Установка статического IP адреса физическому интерфейсу

Подключение должно быть выполнено по следующей схеме (рис. 1).



Воспользоваться утилитой ip. Для того чтобы увидеть существующие в системе интерфейсы, набрать команду:

```
ip address
```

Там же будут отображены основные параметры этих сетевых интерфейсов.
Команда ip позволяет использовать короткие имена команд. В данном случае, вместо `ip address` можно использовать команду

```
ip a
```

В случае правильного выполнения команд (для всех команд кроме ip address) утилита `ip` не будет возвращать никакого значения. В случае, если команда выполнена неправильно, будет возвращена соответствующая ошибка.

Задать интерфейсу `eth1` IP адрес:

```
sudo ip address add 10.0.12.20/24 dev eth1
```

Изменить состояние на `up`.

```
sudo ip link set up dev eth1
```

Посмотреть изменения (состояние устройства eth1 должно измениться на UP):

```
ip address
```

Подключиться к **labnode-2**. Установить интерфейсу **eth1** IP адрес: Необходимо привести конфиг на labnode-2 к его минимальной конфигурации необходимой для подключения:

```
sudo vi /etc/netplan/50-cloud-init.yaml
```

И привести его к следующему виду:

```
network:
    version: 2
    ethernets:
        eth0:
            dhcp4: true
            mtu: 9000
```

Задать адрес интерфейса `eth1`

```
sudo ip address add 10.0.12.30/24 dev eth1
```

Изменить состояние на `up`:

```
sudo ip link set up dev eth1
```

С **labnode-1** проверить доступность **labnode-2**:

```
ping -c 4 10.0.12.30
```

После перезагрузки сервера, или сервиса сети все изменения отменяются. Перезагрузить **оба** сервера, и посмотреть на состояние интерфейсов (необходимо проверить, сохранились ли на интерфейсе адреса, заданные предыдущими командами):

```
reboot
# Дождаться перезагрузки системы
ip address
```

Задание 2. Настройка статического адреса через конфигурационные файлы

Подключение должно быть выполнено по следующей схеме (рис. 1). Для того чтобы изменения оставались в силе, нужно настроить интерфейс через конфигурационный файл. Тогда настройки будут загружаться при старте системы. Создать конфигурацию

интерфейса eth1 на **labnode-1**:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

И привести его к следующему виду:

```
TYPE=Ethernet
DEVICE=eth1
BOOTPROTO=static
IPADDR=10.0.12.20
PREFIX=24
ONBOOT=yes
```

Основные параметры:

1. TYPE - Тип сетевого интерфейса
2. NAME - Имя интерфейса
3. DEVICE - Устройство, которое интерфейс использует
4. BOOTPROTO - если этот параметр static, то интерфейс не будет автоматически получать адрес от сети, маску и другие параметры подключения. В случае необходимости автоматического получения адреса – необходимо указать значение этого параметра -dhcp.
5. ONBOOT - включать ли интерфейс при загрузке.
6. IPADDR - IP-адрес
7. DNS1 - DNS, через который обращаться к доменам. Можно указать несколько параметров: DNS1, DNS2...
8. PREFIX - префикс, другой способ задания маски сети. Для префикса 24 маска будет 255.255.255.0
9. GATEWAY - шлюз

Все остальные параметры являются необязательными в данной лабораторной работе, и должны быть удалены.

Скопировать файл **ifcfg-eth1** с именем **ifcfg-eth2**:

```
sudo cp /etc/sysconfig/network-scripts/ifcfg-eth1 /etc/sysconfig/network-scripts/ifcfg-eth2
```

Отредактировать его с помощью редактора **vi**:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth2
```

И привести его содержимое к следующему виду:

```
TYPE=Ethernet
DEVICE=eth2
BOOTPROTO=static
IPADDR=10.0.12.21
PREFIX=24
ONBOOT=yes
```

Перезагрузить сеть и посмотреть интерфейсы:

```
sudo systemctl restart network
ip address
```

Далее необходимо настроить интерфейсы на узле **labnode-2**используя netplan. Сначала необходимо отредактировать создать конфигурацию интерфейса eth1 на **labnode-2**:

```
sudo vi /etc/netplan/51-eth1.yaml
```

И привести его к следующему виду:

```
network:
    version: 2
    ethernets:
        eth1:
            addresses:
            - 10.0.12.30/24
            mtu: 1492
```

Скопировать файл **51-eth1.yaml** с именем **51-eth2.yaml**:

```
sudo cp /etc/netplan/51-eth1.yaml /etc/netplan/52-eth2.yaml
```

Отредактировать его через vi:

```
sudo vi /etc/netplan/52-eth2.yaml
```

И привести к следующему виду:

```
network:
    version: 2
    ethernets:
        eth2:
```

```
addresses:
- 10.0.12.31/24
mtu: 1492
```

Примените конфигурацию и посмотреть интерфейсы:

```
sudo netplan try
ip address
```

Проверить доступность интерфейсов:

```
(labnode-1) ping -c 4 10.0.12.30
(labnode-1) ping -c 4 10.0.12.31
(labnode-2) ping -c 4 10.0.12.20
(labnode-2) ping -c 4 10.0.12.21
```

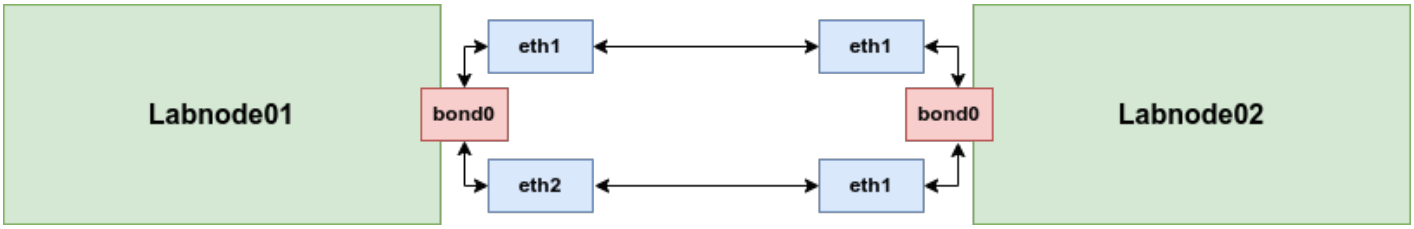
Задание 3. Настройка объединения интерфейсов.

"Объединение" (bonding) сетевых интерфейсов - позволяет совокупно собрать несколько портов в одну группу, эффективно объединяя пропускную способность в одном направлении. Например, вы можете объединить два порта по 100 мегабит в 200 мегабитный магистральный порт.

<I>В некоторых случаях интерфейсы после перезапуска сетевой службы не удаляют ip адреса, что связано с особенностью работы up/down скриптов. В таком случае в системе на разных интерфейсах может присутствовать одинаковый ip адрес (проверить можно командой `ip address`). Для решения этой проблемы необходимо отчистить все адреса на интерфейсе. Сделать это можно как просто удалив конкретный адрес с интерфейса, так и воспользоваться специальной командой, которая очистит все имеющиеся на нём адреса:

```
sudo ip address flush dev eth1
sudo ip address flush dev eth2
```

Подключение должно быть выполнено по следующей схеме (рис. 2).



Для того, чтобы создать интерфейс **bond0**, нужно создать файл конфигурации:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

Конфигурация **bond0** интерфейса на **labnode-1** будет следующей:

```
TYPE=Bond
DEVICE=bond0
BOOTPROTO=static
IPADDR=10.0.12.20
PREFIX=24
BONDING_MASTER=yes
BONDING_OPTS="mode=0 miimon=100"
ONBOOT=yes
```

Это создаст сам **bond0** интерфейс. Но нужно также назначить физические интерфейсы **eth1** и **eth2**, как подчиненные ему. Необходимо изменить конфигурационный файл **eth1**:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth1
```

И привести его к следующему виду:

```
TYPE=Ethernet
DEVICE=eth1
MASTER=bond0
SLAVE=yes
```

То же самое сделать и с **eth2**:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-eth2
```

```
TYPE=Ethernet
DEVICE=eth2
MASTER=bond0
SLAVE=yes
```

Перезагрузить сеть.

```
sudo systemctl restart network
```

Посмотреть, что получилось:

```
ip a
```

В полученном выводе интерфейсы **eth1** и **eth2** должны быть в подчиненном режиме (SLAVE), а интерфейс **bond0** должен иметь ip адрес и находиться в состоянии **UP**. Также необходимо

проверить, что на интерфейсах **eth1** и **eth2** нет никаких ip адресов. Теперь необходимо проделать аналогичные манипуляции на labnode-2.

Удалить содержимое файлов 51-eth1.yaml и 52-eth2.yaml

```
rm -rf /etc/netplan/51-eth1.yaml
rm -rf /etc/netplan/52-eth2.yaml
```

Настроить интерфейс bond0 для создания отказоустойчивого подключения. Для этого открыть новый файл конфигурации 53-bond0.yaml:

```
sudo vi /etc/netplan/53-bond0.yaml
```

Конфигурация **bond0** интерфейса будет следующей:

```
network:
    version: 2
    ethernets:
        eth1: {}
        eth2: {}
    bonds:
        bond0:
            mtu: 1492
            interfaces:
                - eth1
                - eth2
            parameters:
                mode: balance-rr
                mii-monitor-interval: 100
            addresses:
                - 10.0.12.30/24
```

Перечитать конфигурационные файлы сетевых устройств и проверить после этого настройки сетевых интерфейсов:

```
sudo netplan try
```

С **labnode-1** необходимо проверить доступность **labnode-2**:

```
ping -c 4 10.0.12.30
```

Теперь на **labnode-1** необходимо отключить **eth1** и посмотреть его состояние:

```
sudo ip link set down eth1
ip address
```

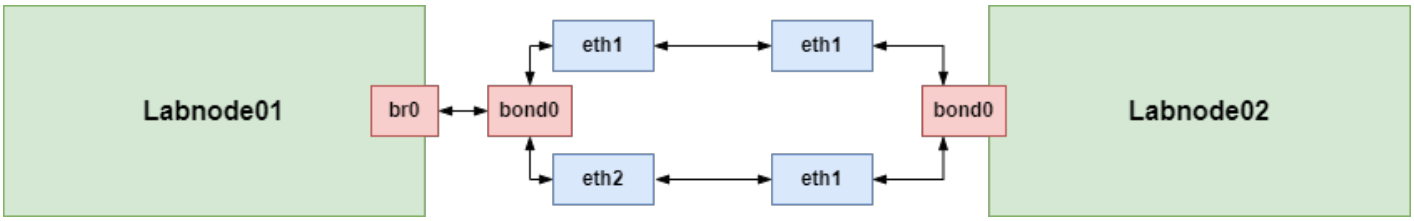
И с **labnode-2** проверить его доступность:

```
ping 10.0.12.20 -c 4
```

Если объединение интерфейсов настроено правильно, то узел будет доступен, даже после выключения одного из интерфейсов.

Задание 4. Настройка bridge интерфейса.

Ядро Linux имеет встроенный механизм коммутации пакетов между интерфейсами, и может функционировать как обычный сетевой коммутатор. Интерфейс Bridge представляет собой как сам виртуальный сетевой коммутатор, так и сетевой интерфейс с ip адресом, назначенным на порт этого коммутатора. Подключение должно быть выполнено по следующей схеме (рис. 3).



На **labnode-1** требуется создать конфиг **ifcfg-br0**:

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-br0
```

Мост будет иметь следующую конфигурацию:sudo

```
TYPE=Bridge
DEVICE=br0
BOOTPROTO=static
IPADDR=10.0.12.20
PREFIX=24
STP=on
ONBOOT=yes
```

Spanning Tree Protocol (STP) нужен, чтобы избежать петель коммутации.

Интерфейс **bond0**, настроенный до этого, может быть интерфейсом этого сетевого коммутатора, но в таком случае ip адрес уже будет назначен на интерфейс виртуального сетевого коммутатора, и все настройки ip с интерфейса **bond0** можно будет убрать. Для этого в конфигурационный файл интерфейса bond0 также нужно добавить параметр BRIDGE=br0. Также удалить из него параметры `BOOTPROTO, IPADDR, PREFIX, ONBOOT` (можно

просто закомментировать с помощью символа #, когда пригодятся, раскомментировать их, убрав символ #):

```
sudo vi /etc/sysconfig/network-scripts/ifcfg-bond0
```

Перезагрузить сеть:

```
sudo systemctl restart network
```

Можно проверить результат. Для этого на **labnode-2** выполнить следующую команду:

```
ip address
```

Проверить, что в результате вывода этой команды ip адрес будет назначен только на интерфейс br0, и он будет в состоянии UP. Если все правильно, то проверить доступность соседнего узла командой:

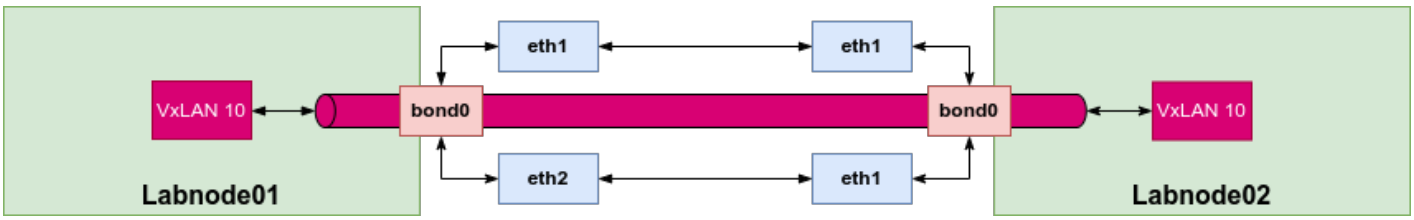
```
ping -c 4 10.0.12.20
```

> Мост может подняться не сразу. Необходимо подождать.

Задание 5. Создание VxLAN интерфейсов.

VxLAN является механизмом построения виртуальных сетей на основаниях тоннелей, поверх реальных сетей, но при этом позволяющим их разграничивать.

Подключение должно быть выполнено по следующей схеме (рис. 4).



На **labnode-1** удалить конфигурацию моста **br0**:

```
sudo rm /etc/sysconfig/network-scripts/ifcfg-br0
```

И привести **bond0** к прежнему виду:

```
TYPE=Bond
DEVICE=bond0
BOOTPROTO=static
```

```
IPADDR=10.0.12.20
PREFIX=24
#BRIDGE=br0
BONDING_MASTER=yes
BONDING_OPTS="mode=0 miimon=100"
ONBOOT=yes
```

И перезагрузить сервер:

```
sudo reboot
```

После загрузки сервера проверить работу сети с узла labnode-2:

```
ping -c 4 10.0.12.20
```

На **labnode-1** добавить интерфейс **vxlan10**:

```
sudo ip link add vxlan10 type vxlan id 10 dstport 0 dev bond0
```

Настроить коммутацию Linux Bridge:

```
  sudo bridge fdb append to 00:00:00:00:00:00 dst 10.0.12.30 dev vxlan10
```

Назначить vxlan10 IP адрес и перевести его в состояние up:

```
sudo ip addr add 192.168.1.20/24 dev vxlan10
sudo ip link set up dev vxlan10
```

VxLAN работает как приложение. Пакеты инкапсулируются в udp, и для работы VxLAN требуется udp порт 8472. Открыть его в фаерволе:

```
sudo firewall-cmd --permanent --add-port=8472/udp
sudo firewall-cmd --reload
```

После нужно сделать все то же самое на *labnode-2*. Добавить интерфейс vxlan10:

```
sudo ip link add vxlan10 type vxlan id 10 dstport 0 dev bond0
```

Настроить коммутацию Linux Bridge:

```
sudo bridge fdb append to 00:00:00:00:00:00 dst 10.0.12.20 dev vxlan10
```

Назначить vxlan10 IP адрес и перевести его в состояние up:

```
sudo ip addr add 192.168.1.30/24 dev vxlan10
sudo ip link set up dev vxlan10
```

Протестировать соединение через vxlan. Для этого на **labnode-1**:

```
ping -c 4 192.168.1.30
```

Посмотреть на arp таблицу. Там можно увидеть соответствие mac адресов с ip адресами.

```
arp
```

Нужно убедиться, что появилось приложение, которое слушает порт 8472.

```
ss -tulpn | grep 8472
```

Теперь необходимо добавить vxlan с другим тегом (20), и убедиться в том, что из него не будет доступа к vxlan с тегом 10 (пакеты будут отбрасываться из-за разных тегов). Перезагрузить **labnode-2**. Текущая настройка vxlan сбросится.

```
sudo reboot
```

На **labnode-2** добавить интерфейс vxlan20 и настроить его:

```
sudo ip link add vxlan20 type vxlan id 20 dstport 0 dev bond0
sudo bridge fdb append to 00:00:00:00:00:00 dst 10.0.12.20 dev vxlan20
sudo ip addr add 192.168.1.30/24 dev vxlan20
sudo ip link set up dev vxlan20
ss -tulpn | grep 8472
```

Протестировать соединение через vxlan. Для этого на **labnode-1**:

```
ping 192.168.1.30 -c 4
```

Версия #12
Тарабанов Илья Федорович создал 7 ноября 2023 19:36:51
Тарабанов Илья Федорович обновил 16 мая 2024 19:12:22