

Министерство цифрового развития,
связи и массовых коммуникаций
Федеральное государственное
образовательное бюджетное учреждение
высшего профессионального образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Методическое пособие практических работ №4-8

Преподаватель:

Андреев А.В.

Оглавление

Оглавление

Практическая работа №4.....	3
Практическая работа №5.....	20
Практическая работа №6.....	29
Практическая работа №7.....	37

Практическая работа №4

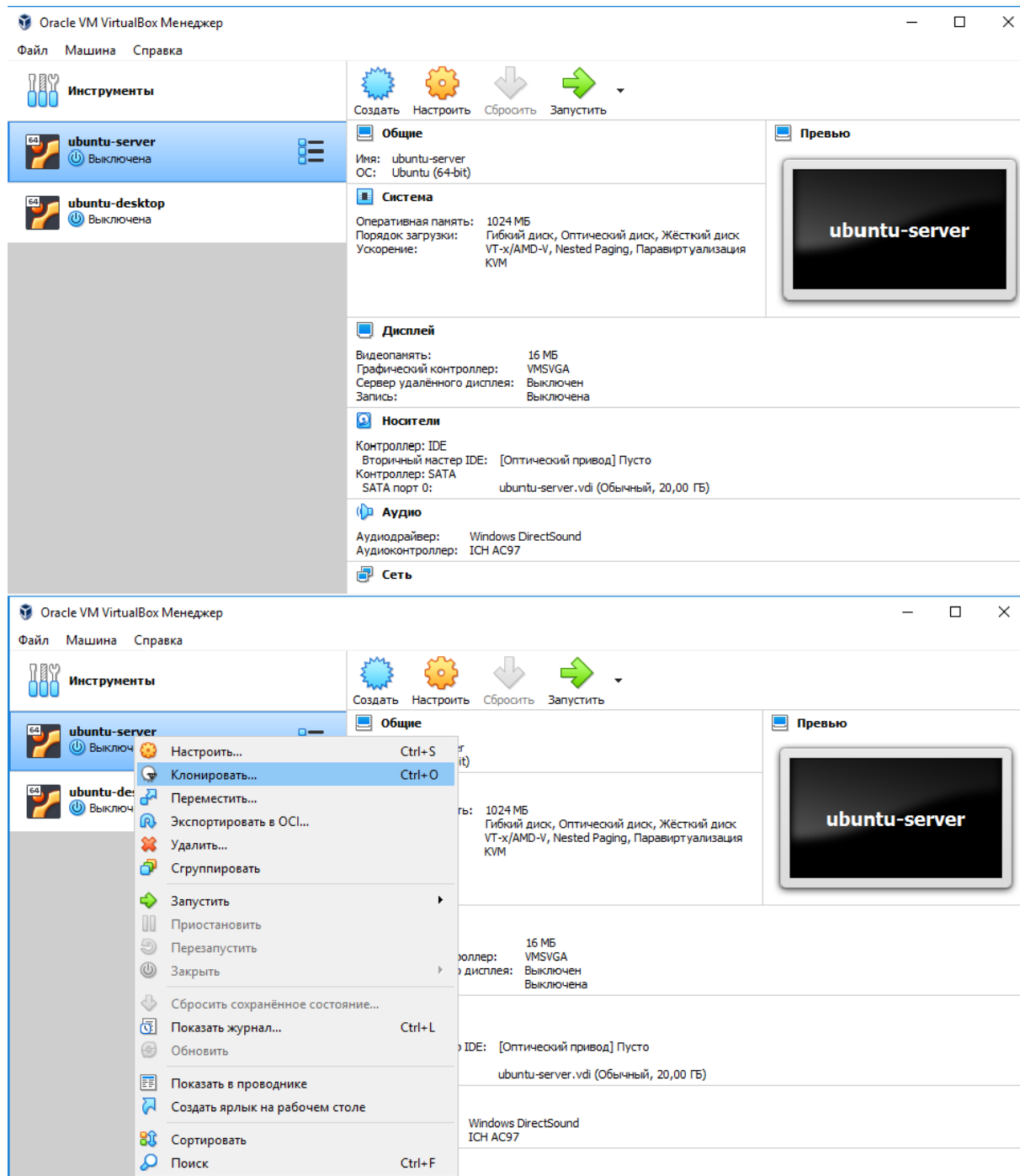
Настройка шлюза локальной сети, на базе Ubuntu 20.04.5

Если у вас есть локальная сеть, то для её клиентов необходимо предоставить доступ в Интернет. Для обеспечения данной возможности необходимо настроить шлюз, который будет принимать запросы клиентов и пересылать их во внешний мир, а поступающие ответы, передавать обратно.

Подготовка

Для выполнения практической работы нам понадобится отдельная виртуальная машина (ВМ) с Ubuntu Server и ВМ с Ubuntu Desktop.

Клонируем необходимые ВМ из шаблонов, подготовленных заранее.



Обратите внимание на указание имени ВМ и выбор политики MAC-адреса.

? ✕

← Клонировать виртуальную машину

Укажите имя и расположение новой машины

Пожалуйста укажите имя и, при необходимости, папку новой виртуальной машины. Эта машина будет клоном машины **ubuntu-server**.

Имя:

Путь:

Политика MAC-адреса:

Дополнительные опции: ☐ Сохранить имена дисков ☐ Сохранить идентификаторы оборудования

? ✕

← Клонировать виртуальную машину

Укажите тип клонирования

Пожалуйста укажите какое клонирование Вы желаете выполнить.

Если Вы выберете **Полное клонирование**, будет создана полная копия клонируемой виртуальной машины (включая все файлы виртуальных жёстких дисков).

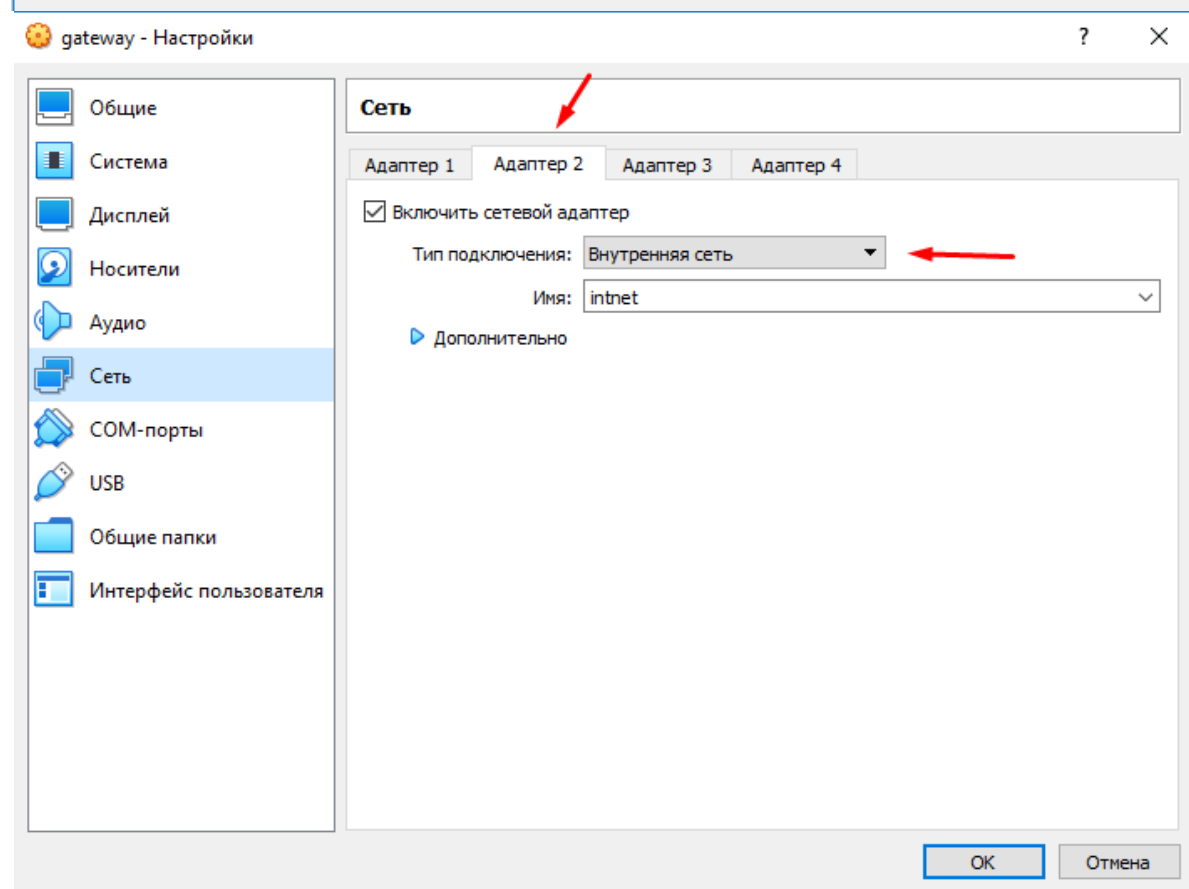
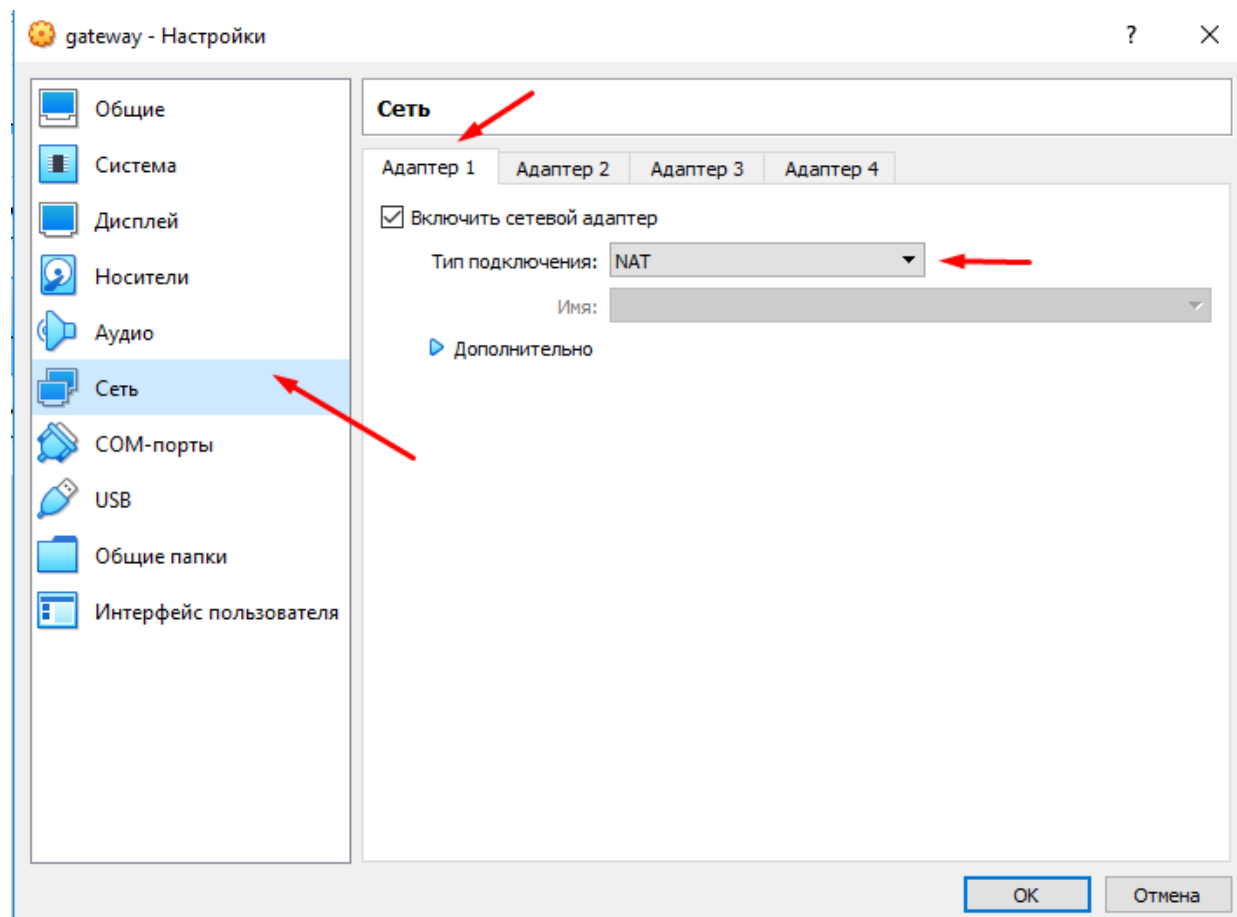
Если Вы выберете **Связное клонирование**, будет создана новая машина, использующая файлы виртуальных жёстких дисков клонируемой машины и Вы не сможете перенести новую машину на другой компьютер без переноса клонируемой.

Если Вы выберете **Связное клонирование**, в клонируемой машине также будет создан новый снимок, являющийся частью процедуры клонирования.

☒ Полное клонирование
☐ Связное клонирование

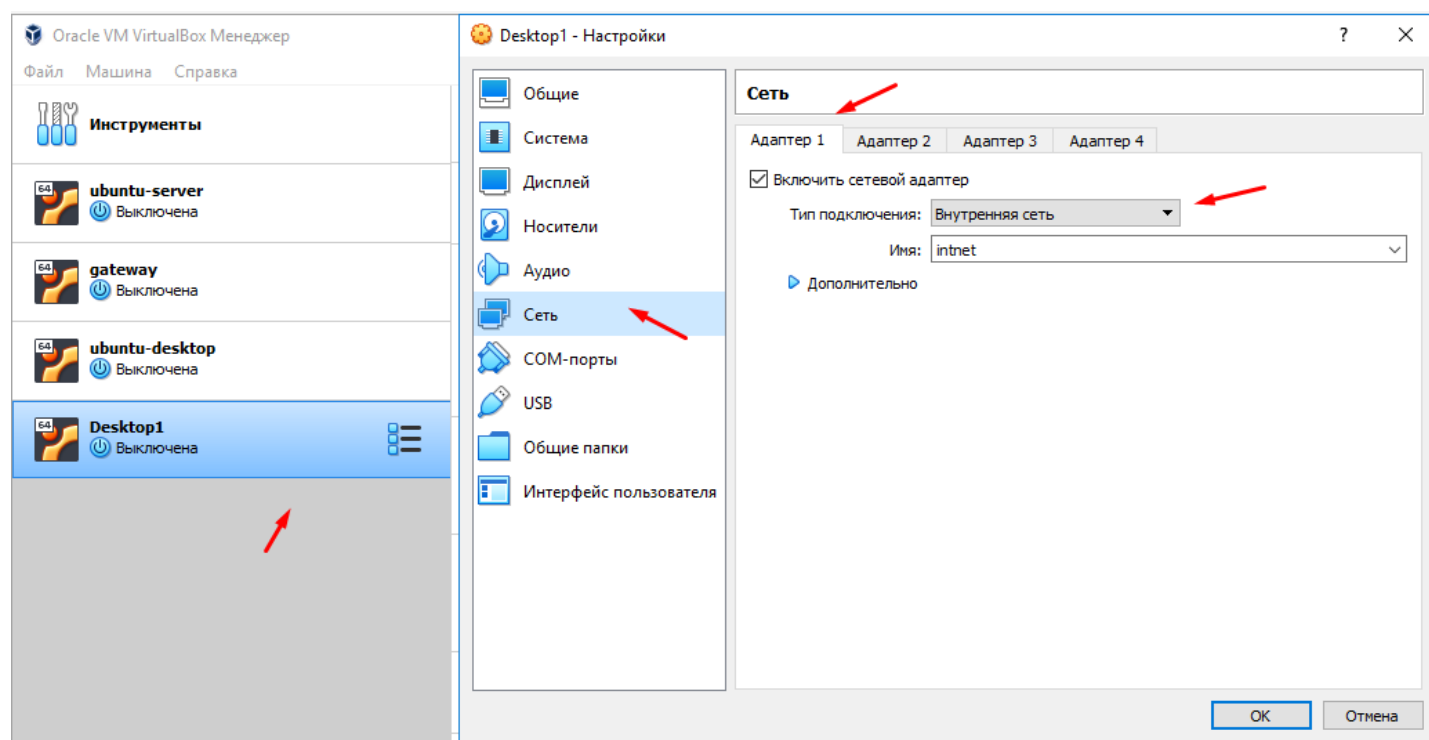
После завершения процесса клонирования заходим в опции ВМ «gateway» и настраиваем сетевые адаптеры:

В новой VirtualBox Менеджер настройка второго сетевого адаптера в расширенных настройках



Подготовка сервера завершена.

Для подготовки пользовательской ВМ повторяем процедуру клонирования уже с шаблона с Ubuntu Desktop, далее настраиваем сетевой интерфейс:

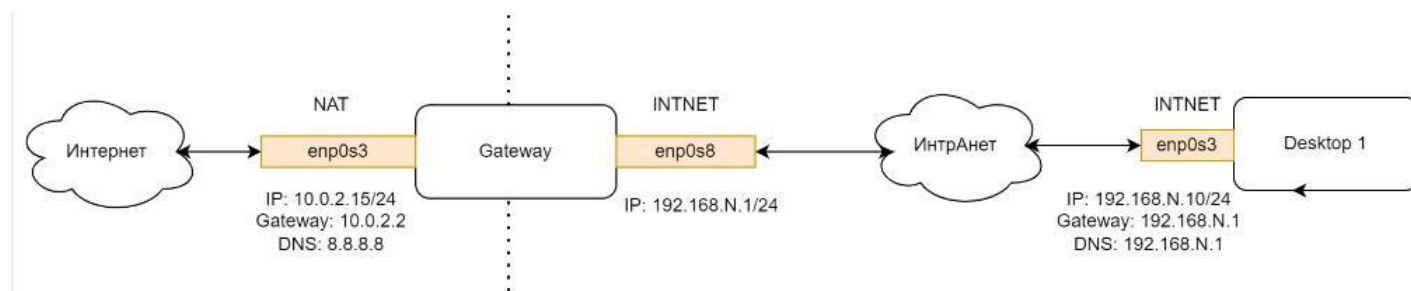


Выполнение

Для работы нам понадобится:

1. Виртуальная машина Ubuntu 20.04.5 Server с двумя виртуальными сетевыми адаптерами. Данная ВМ будет выполнять функцию шлюза.
2. Виртуальная машина Ubuntu Desktop, на которой будет тестироваться работа

шлюза. Схема работы:



Где N – номер студента в журнале.

Предполагается, что операционная система у вас установлена, на сервере, который имеет 2 сетевых интерфейса.

enp0s3 — подключение к Интернету. Может получать IP-адрес динамически, может иметь статический. Для виртуальной машины это тип адаптера «Сетевой мост» или «NAT».

enp0s8 — подключение к локальной сети, будет иметь статический IP **192.168.N.1** и маску **255.255.255.0**. Тип адаптера для виртуальной машины «Внутренняя сеть».

Также, для тестирования нам понадобится клиентская машина.

Первым делом, настраиваем сетевые интерфейсы сервера:

Поднимаем права до **root**

```
sudo su
```

Вводим пароль.

Теперь получим список сетевых интерфейсов и определим их имена для дальнейшей настройки

```
ip a
```

В ответ получаем список сетевых интерфейсов в системе и их параметры:

```
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: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:54:31:6a brd ff:ff:ff:ff:ff:ff
3: enp0s8: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 08:00:27:c2:9f:ac brd ff:ff:ff:ff:ff:ff
```

Обратите внимание на интерфейс №2 с именем **enp0s3** и интерфейс №3 **enp0s8**.

enp0s3 – является внешним интерфейсом для связи с Интернетом, а **enp0s8** внутренним для построения интранет-сети.

Редактируем настройки сетевых интерфейсов:

```
nano /etc/netplan/00-installer-config.yaml
```

Настраиваем **enp0s3** (по которому осуществляется подключение к Интернету).

Обратите внимание, что в файле конфигурации везде используются **пробелы** для отступов, и они **обязательны**.

Вариант №1- Получение IP по DHCP от провайдера:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
```

Это вариант настройки по умолчанию, но нам он не подходит.

Вариант №2-Статический IP

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - XXX.XXX.XXX.XXX/24
      gateway4: YYY.YYY.YYY.YYY
      nameservers:
        addresses: [ZZZ.ZZZ.ZZZ.ZZZ]
```

Где вместо XXX.XXX.XXX.XXX вписываем IP-адрес, который мы получили от провайдера.
Указываем /24 -выписываем маску подсети.
Это соответствует 255.255.255.0.

В качестве YYY.YYY.YYY.YYY указываем IP-адрес шлюза.
Ну и вместо ZZZ.ZZZ.ZZZ.ZZZ вписываем IP-адрес DNS сервера.

При использовании в Virtual BOX внешнего сетевого интерфейса «NAT», параметры будут одинаковые для всех:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses:
        - 10.0.2.15/24
      gateway4: 10.0.2.2
      nameservers:
        addresses: [8.8.8.8]
```

Настраиваем **enp0s8** (по которому подключается локальная сеть)

```
enp0s8:
  dhcp4: no

  addresses:
    - 192.168.N.1/24
```


Обратите внимание что **N** – это ваш номер в журнале. И вы должны заменить его на цифру.

В результате действий у нас должен получиться файл следующего содержания:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no

      addresses:

        - 10.0.2.15/24

    gateway4: 10.0.2.2

  nameservers:

    addresses: [8.8.8.8]

    enp0s8:
      dhcp4: no

      addresses:

        - 192.168.N.1/24
```

Сохраняем изменения, выходим.

Применяем конфигурацию:

```
netplan apply
```

Проверим правильность настройки командой:

```
ip a
```

Результат:

```
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:54:31:6a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe54:316a/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c2:9f:ac brd ff:ff:ff:ff:ff:ff
    inet 192.168.109.1/24 brd 192.168.109.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec2:9fac/64 scope link
        valid_lft forever preferred_lft forever
```

Проверяем наличие доступа в Интернет:

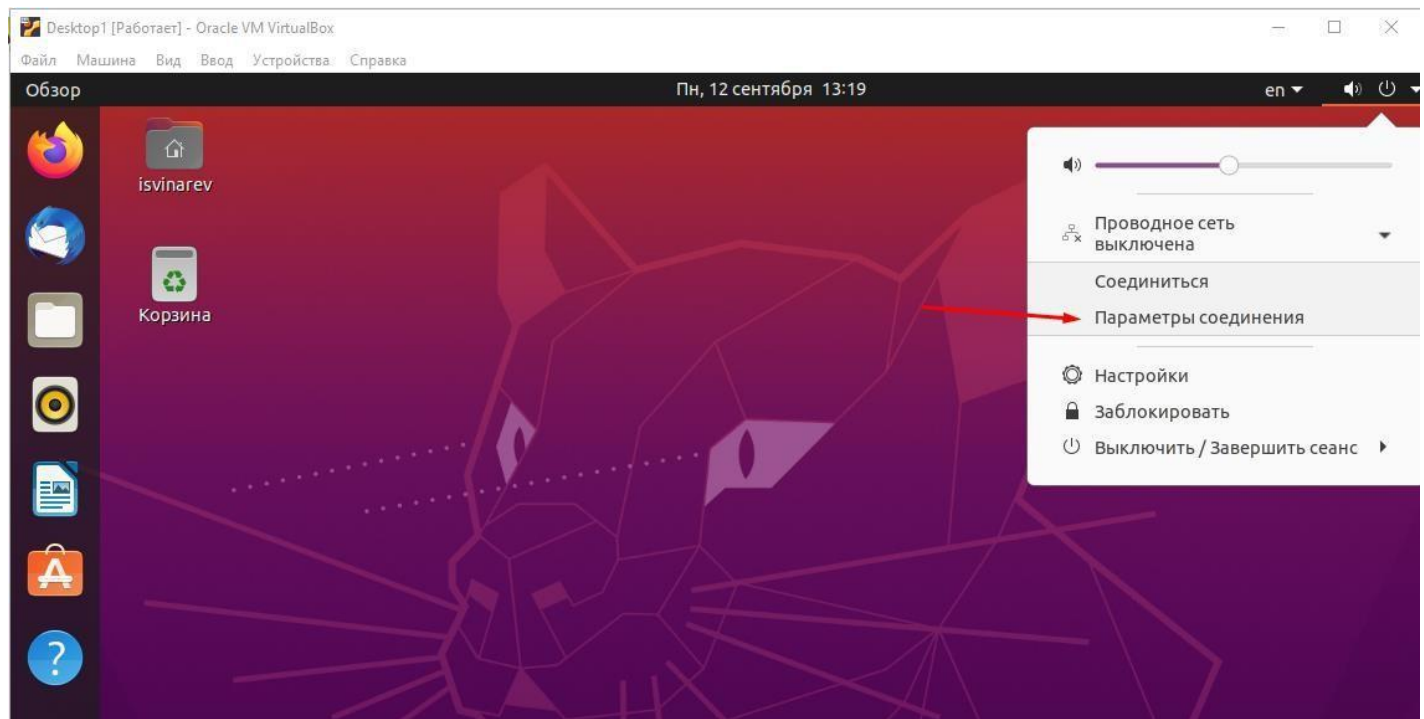
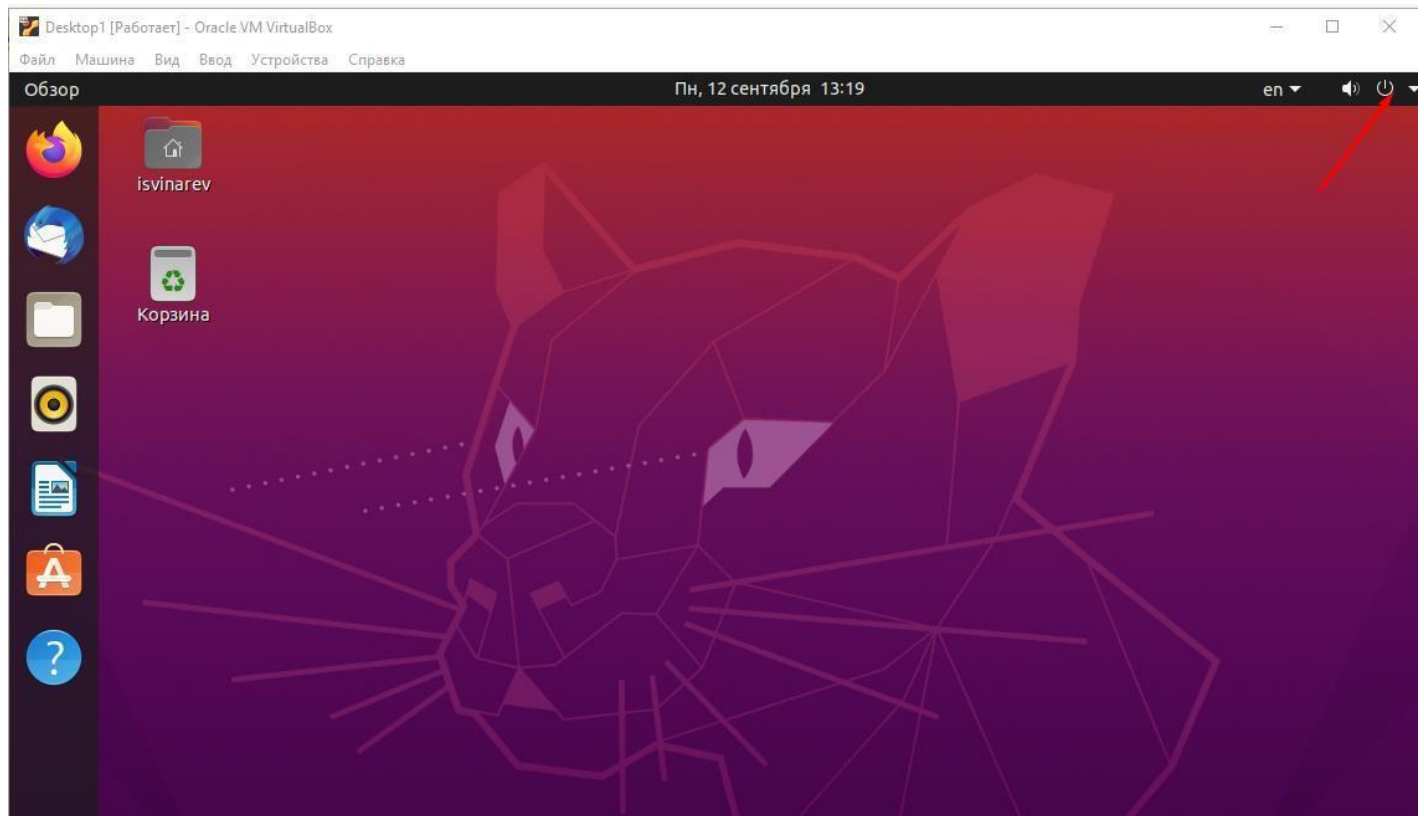
```
ping ya.ru
```

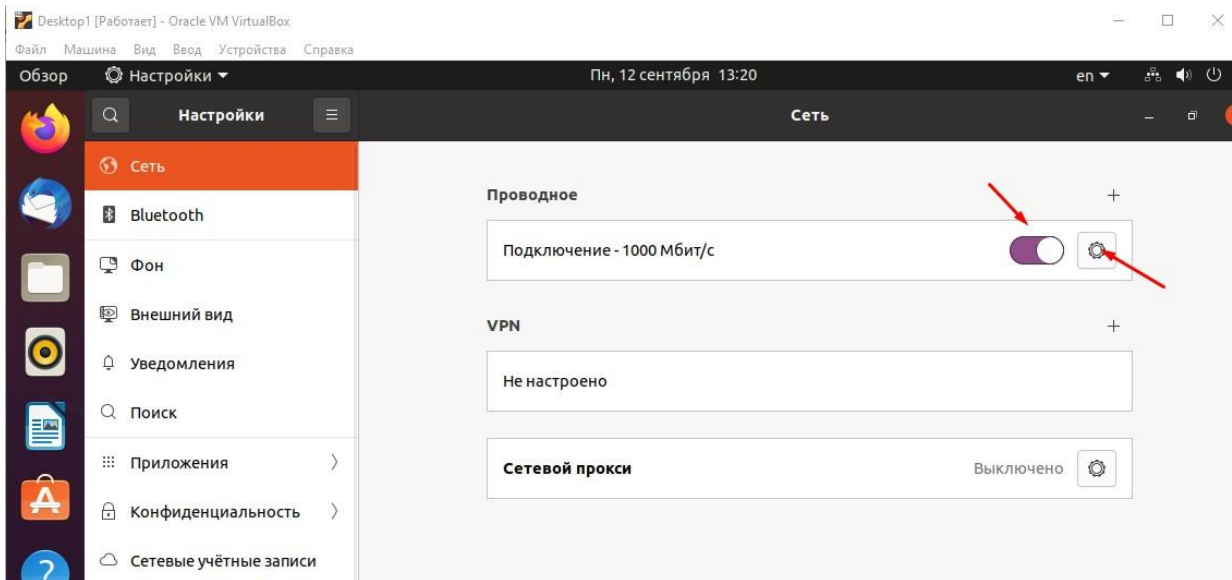
Результат:

```
PING ya.ru (87.250.250.242) 56(84) bytes of data.
64 bytes from ya.ru (87.250.250.242): icmp_seq=1 ttl=242 time=81.3 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=2 ttl=242 time=80.6 ms
64 bytes from ya.ru (87.250.250.242): icmp_seq=3 ttl=242 time=80.5 ms
```

Из написанного ясно, что в локальной сети адрес шлюза и DNS-сервера будет 192.168.N.1. Переходим к нашему тестовому клиенту, т.к. у нас в сети нет DHCP-сервера, то IP-адрес мы будем назначать вручную.

Заходим на ВМ «Desktop1» в настройки сети:





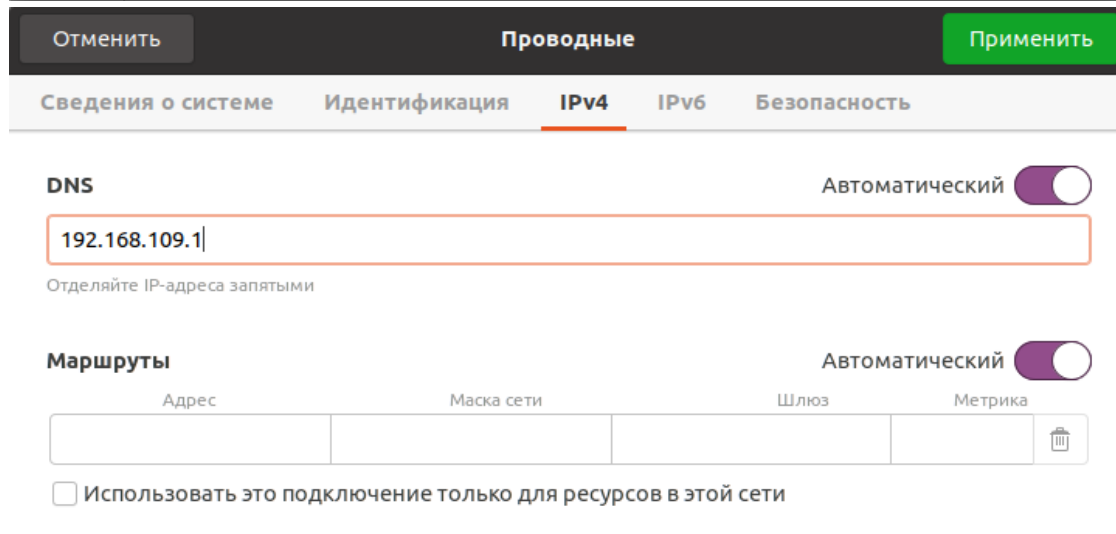
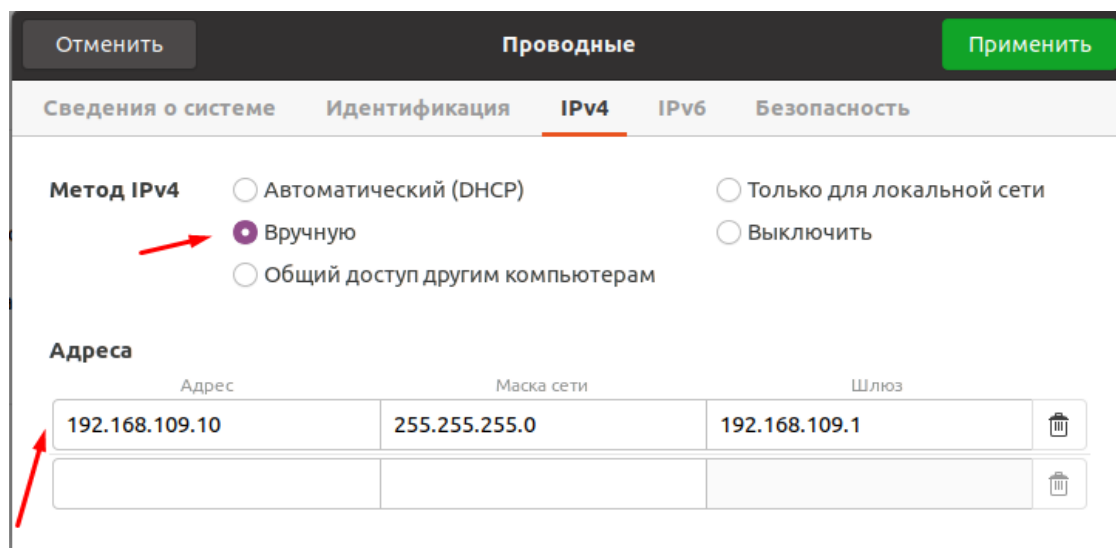
Присваиваем клиенту:

IP 192.168.N.10

Маску подсети 255.255.255.0

Шлюз 192.168.N.1

DNS 192.168.N.1



Пробуем с клиента пинговать 192.168.N.1 — запросы должны бегать нормально.

Переходим на наш шлюз.

Первым делом, разрешаем перенаправление пакетов:

```
nano /etc/sysctl.conf
```

Необходимо найти строку и снять с неё комментарий:

```
net.ipv4.ip_forward=1
```

Сохраняем изменения и выходим.

Для настройки правил перенаправления пакетов мы воспользуемся встроенным межсетевым экраном (брандмауэр) **netfilter**. Все операции с брандмауэром выполняются через командный интерфейс **iptables**.

Предварительно нам нужно загрузить пакет, позволяющий сохранить настройки сетевого экрана после перезагрузки.

```
apt-get update  
apt install iptables-persistent
```

Во время установки, появляются два сообщения с предложением сохранить текущие настройки, в определённых файлах.

Соглашаемся дважды.

Принцип действия **iptables-persistent** заключается в том, что при перезагрузке системы, в брандмауэр автоматически добавляются правила, находящиеся в файлах `/etc/iptables/rules.v4` и `/etc/iptables/rules.v6`.

Далее создаём эти правила:

```
iptables -F  
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE  
iptables -A FORWARD -i enp0s3 -o enp0s3 -j REJECT  
iptables -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

Так же добавим правило для перенаправления DNS запросов из внутренней сети в DNS сервер Google (8.8.8.8).

```
iptables -t nat -A PREROUTING -i enp0s8 -p tcp -m tcp --dport 53 -j DNAT --to-destination 8.8.8.8:53  
iptables -t nat -A PREROUTING -i enp0s8 -p udp -m udp --dport 53 -j DNAT --to-destination 8.8.8.8:53
```

Сохраним созданные правила маршрутизации:

```
iptables-save > /etc/iptables/rules.v4
```

Выполняем перезагрузку сервера

```
reboot
```

Дождёмся загрузки сервера и перейдя к клиентской системе, пробуем открыть любой сайт.

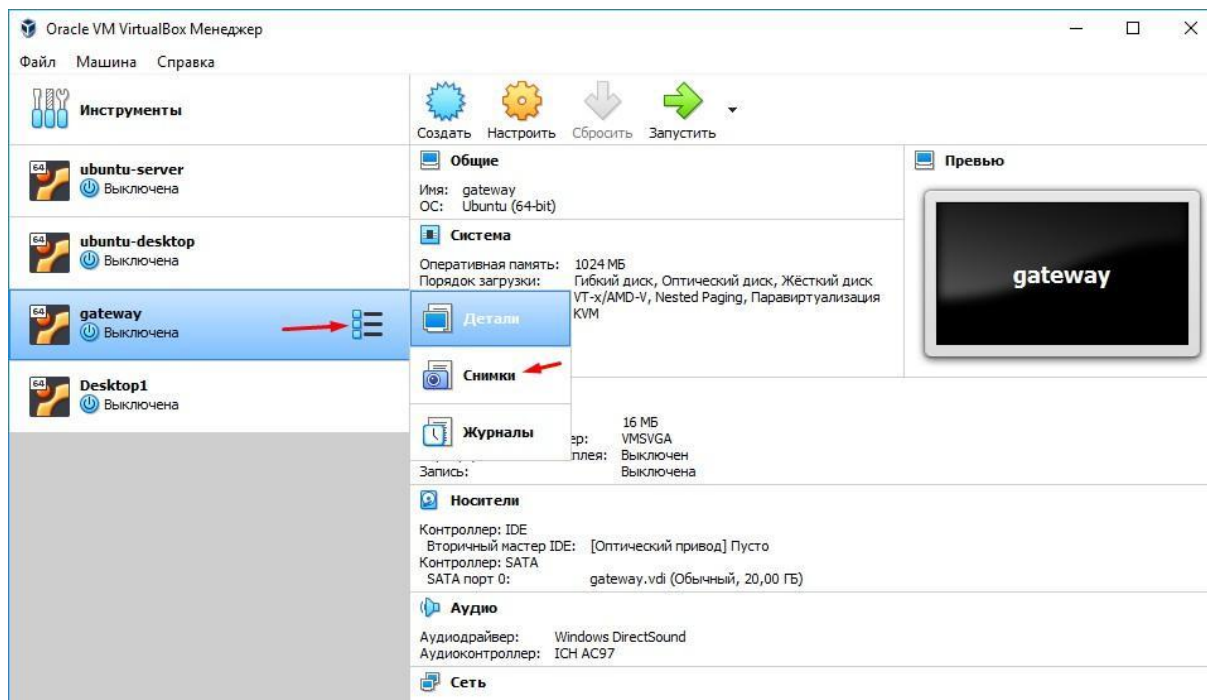
Настройка DHCP-сервера под управлением Ubuntu

Подготовка:

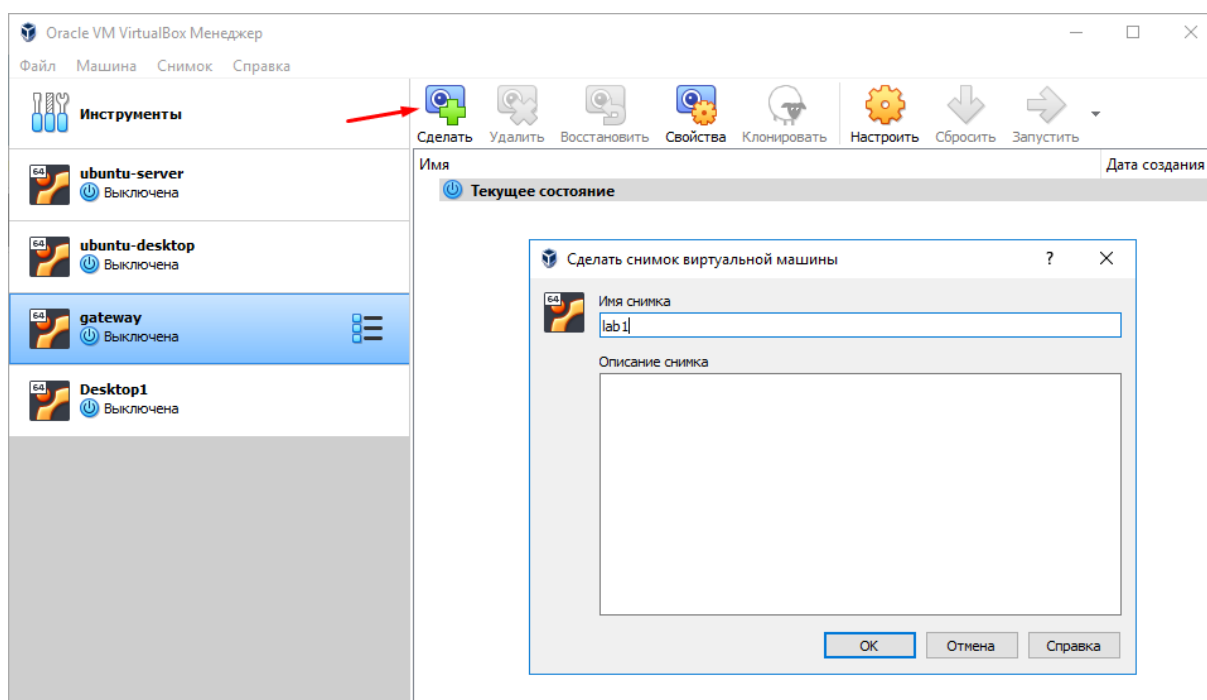
1. Шлюз настроен ранее
2. Создать снимок состояния ВМ.

Снимок ВМ (snapshot) позволяет сохранить состояние и конфигурацию ВМ во времени, что даёт возможность откатить внесённые изменения и получить работающий шлюз.

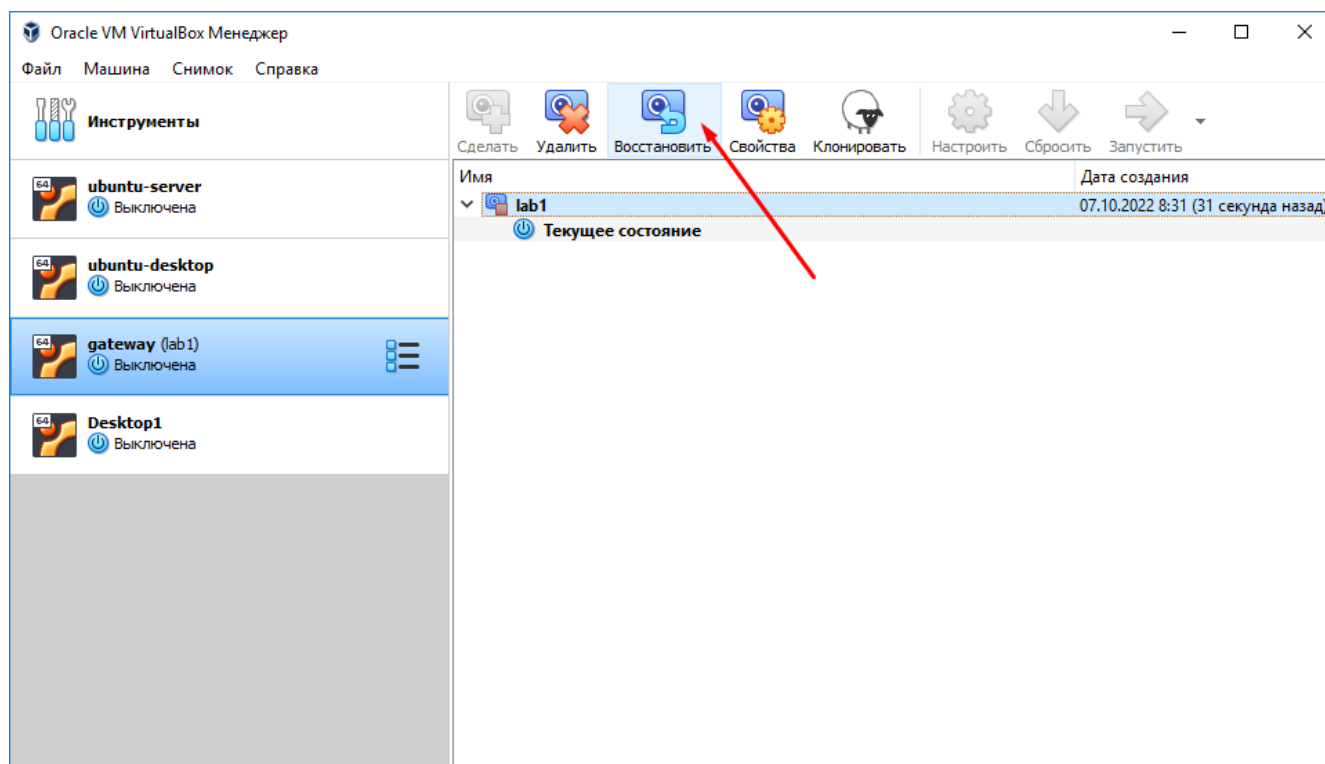
Для создания снимка ВМ переходим в меню снимков.



Создаём снимок, даём ему название «Lab4».



Теперь возможно вернуть состояние VM на момент создания снимка.



Выполнение:

N – номер студента в журнале.

Переходим к настройке.

Поднимаем права до root:

```
sudo su
```

Установим пакет DHCP-сервера:

```
apt-get update  
apt install isc-dhcp-server
```

Зависимости подтянутся автоматически.

Адресное пространство, в нашей локальной сети, будет находиться в диапазоне 192.168.N.0/24 т.е. в нашей подсети может находиться максимум 254 сетевых устройства.

Для начала, укажем на каком интерфейсе будет работать наш DHCP-сервер

```
nano /etc/default/isc-dhcp-server
```

Нас интересует строка **INTERFACESv4** т.к. к локальной сети у нас подключается **enp0s8**, вот его и укажем:

```
INTERFACESv4="enp0s8"
```

Теперь нам необходимо настроить конфигурационный файл DHCP-сервера. Предварительно сделаем его резервную копию (это правило хорошего тона) и удалим из него всю лишнюю информацию:

```
cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.example  
echo "" > /etc/dhcp/dhcpd.conf
```

Открываем файл на редактирование.

```
nano /etc/dhcp/dhcpd.conf
```

Сервер планируется единственным в сети, поэтому будет работать в авторитарном режиме, для этого добавляем строку:

```
authoritative;
```

Теперь создадим нашу подсеть, диапазон IP у нас будет начиная с 192.168.N.10 и заканчивая 192.168.N.254, маска подсети 255.255.255.0 (или 24 bit), в качестве шлюза, DNS-сервера у нас выступает сам сервер, указываем адрес IP-интерфейса enp0s8 192.168.N.1.
Время аренды адреса, указывается в секундах, я указал 7 дней.

```
subnet 192.168.N.0 netmask 255.255.255.0 {  
  range 192.168.N.10 192.168.N.254;  
  option domain-name-servers 192.168.N.1;  
  option routers 192.168.N.1;  
  option broadcast-address 192.168.N.255;  
  default-lease-time 604800;  
  max-lease-time 604800;  
}
```

Сохраняем изменения выходим.

Запускаем сервис DHCP

```
service isc-dhcp-server start
```

Проверяем статус сервиса

```
service isc-dhcp-server status
```


В моём варианте с N=109 получается следующее:

```
root@gateway:/home/isvinarev# service isc-dhcp-server status
• isc-dhcp-server.service - ISC DHCP IPv4 server
  Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2022-10-07 01:42:01 UTC; 34s ago
  Docs: man:dhcpcd(8)
  Main PID: 2358 (dhcpcd)
  Tasks: 4 (limit: 1066)
  Memory: 5.8M
  CGroup: /system.slice/isc-dhcp-server.service
          └─2358 dhcpcd -user dhcpcd -group dhcpcd -f -4 -pf /run/dhcp-server/dhcpcd.pid -cf
            /etc/dhcp/dhcpcd.conf enp0s8

Oct 07 01:42:01 gateway sh[2358]: PID file: /run/dhcp-server/dhcpcd.pid
Oct 07 01:42:01 gateway dhcpcd[2358]: Wrote 0 leases to leases file.
Oct 07 01:42:01 gateway sh[2358]: Wrote 0 leases to leases file.
Oct 07 01:42:01 gateway dhcpcd[2358]: Listening on
LPF/enp0s8/08:00:27:c2:9f:ac/192.168.109.0/24 Oct 07 01:42:01 gateway sh[2358]: Listening on
LPF/enp0s8/08:00:27:c2:9f:ac/192.168.109.0/24 Oct 07 01:42:01 gateway sh[2358]: Sending on
LPF/enp0s8/08:00:27:c2:9f:ac/192.168.109.0/24
Oct 07 01:42:01 gateway dhcpcd[2358]: Sending on
LPF/enp0s8/08:00:27:c2:9f:ac/192.168.109.0/24
Oct 07 01:42:01 gateway dhcpcd[2358]: Sending on Socket/fallback/fallback-net
Oct 07 01:42:01 gateway sh[2358]: Sending on Socket/fallback/fallback-net
Oct 07 01:42:01 gateway dhcpcd[2358]: Server starting service.
```

Выйти из просмотра можно клавишей «Q» или комбинацией Ctrl+C.

По тексту можно увидеть, что конфигурация верна, DHCP сервис запустился и работает на интерфейсе enp0s8 с сетью 192.168.109.0/24.

Переходим к нашей клиентской ВМ, устанавливаем в настройках сетевого соединения - получение IP-адреса от DHCP-сервера. Получаем настройки сети, проверяем доступ в Интернет.

Если есть необходимость в резервировании IP-адреса за определенной машиной, то бегать к клиентскому ПК, чтобы забить там статический IP, нет необходимости, да и это совершенно неправильно. Гораздо удобнее выполнить резервацию этого IP-адреса на DHCP-сервере. После выполнения резервации данный IP-адрес будет выдаваться только тому MAC-адресу, за которым он зарегистрирован.

Делается это очень просто:

В dhcpcd.conf добавляется следующее:

```
host testhost {
    hardware ethernet 00:01:8a:e3:s8:92;
    fixed-address 192.168.N.51;
}
```

Где:

hardware ethernet – указываем MAC-адрес сетевой карты клиента.

Если понадобилось посмотреть, какие адреса были выданы, а также узнать их статус (свободен/занят), то идём в:

```
nano /var/lib/dhcp/dhcpd.leases
```

Траблшутинг:

В случае возникновения проблем, то их причины нужно узнавать из окошка статуса сервиса или из логов. По умолчанию isc-dhcp-server кидает записи в syslog, который находится в /var/log/syslog.

Например, я забыл добавить «;» к параметру authoritative в файле конфигурации.

Статус сервиса покажет следующее:

```
# service isc-dhcp-server status
isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled;
   vendor preset: enabled)
   Active: failed (Result: exit-code) since Fri 2022-10-07 01:49:48 UTC; 5min ago
```

Понятно, что есть ошибка, но неясно, где её искать. Для этого посмотрим содержимое файла /var/log/syslog. Файл довольно объёмный, поэтому сразу посмотрим последние 50 строк файла.

```
tail -50 /var/log/syslog
```

В выводе мы сможем увидеть такие строки:

```
Oct 7 01:49:48 gateway sh[2408]: /etc/dhcp/dhcpd.conf line 2: semicolon expected.
```

Становится ясно что в 1 или 2 строке файла /etc/dhcp/dhcpd.conf что-то не так.

Открываем файл, правим его, перезапускаем сервис и повторно проверяем на наличие ошибок.

```
service isc-dhcp-server restart
service isc-dhcp-server status
```

В результате этой практической работы у нас должна быть связь **Сервера с Desktopом**. Проверить это можно командой

На **Desktop** пишем **ping 192.168.N.1** (N – ваш номер в списке группы)

Также на Сервере и Desktopе должен быть Интернет

Для этого мы и там и там пропишем:

```
ping ya.ru
```

Если все идёт, то всё хорошо и можно переходить к следующей практической работе.

Если Интернета нет на Сервере, то проверяем все действия до написания файла(включительно)

```
nano /etc/netplan/00-installer-config.yaml
```

Частые ошибки – опечатки в написании файла выше (расположение или текст файла), при написании самого файла. Возможна неправильная настройка сервера в виртуальной машине.

P.S. если после написания **neplan apply** появляются **warnings**, не пугаемся. Ничего страшного

Если нет связи Десктопа с Сервером. Проверяем настройки в самом Desktop (правильно ли вы указали IPv4(вручную). Адрес, Маска подсети, шлюз, DNS

Если нет Интернета на Десктопе, но есть связь с Сервером. Рекомендую переписать правила **NAT**

P.S. Пока вы на этом этапе, если что-то не получается или не работает, будет проще, если начать все заново (начиная с клонирования)

Практическая работа №5

Настройка DNS+DHCP-сервера для локальной сети + динамическое обновление DNS зон.

Подготовка

1. Шлюз настроен в практической работе №4.
2. DHCP-сервер настроен в практической работе №4.

Обозначения вашего варианта:

N – номер студента в журнале

STUDENT.GROUP.local – вместо STUDENT фамилия студента транслитом, например ivanov.ia131.local Первым делом, научимся переименовывать hostname в операционной системе. Для этого выполняем команды (Не забываем поднять права до root):

```
hostnamectl set-hostname SERVER_HOSTNAME
```

Где **SERVER_HOSTNAME** заменяете на имя вашей

виртуальной машины. Далее поправим имя сервера в файле

/etc/hosts:

```
nano /etc/hosts
```

```
127.0.0.1 localhost  
127.0.1.1 SERVER_HOSTNAME
```

Повторите тоже самое для виртуальной машины Desktop.

Далее необходимо удалить правила маршрутизации, которые были установлен в практической работе №4, т. к. надобность в них отпадает, вместо этого будет использоваться Bind9.

```
nano /etc/iptables/rules.v4
```

Удаляем сточки из файла:

```
-A PREROUTING -i enp0s8 -p tcp -m tcp --dport 53 -j DNAT --to-destination 8.8.8.8:53  
-A PREROUTING -i enp0s8 -p udp -m udp --dport 53 -j DNAT --to-destination 8.8.8.8:53
```

Сохраняем изменения и перезагружаем сервер.

```
reboot
```

Установка DNS сервера

Установим Bind:

```
apt-get update
apt install bind9

apt install dnsutils
```

Переходим к настройкам Bind9, для начала отредактируем файл **named.conf.options**:

```
nano /etc/bind/named.conf.options
```

Туда добавляем следующие строки:

```
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no; # conform to RFC1035
    listen-on {
        127.0.0.1;
        192.168.N.1;
    };
};
```

Рассмотрим более подробно:

В пункте **forwarders** мы указали вышестоящие DNS сервера, куда будет передаваться запрос в случае, если информации о запрошенном URL не будет найдено в собственной базе. В нашем случае, это DNS сервер **google**, но можно указать те, которые нужны вам, например, DNS сервер вашего провайдера.

В пункте **listen-on** указываем IP-адреса, которые будет обслуживать наш DNS сервер, это localhost и интерфейс по которому подключена наша локальная сеть 192.168.N.1, запросы на другие IP-адреса обслуживаться не будут, эту функцию можно рассматривать как дополнительную возможность по снижению нагрузки на наш сервер т.к. запросы из вне-не обслуживают.

Переходим к редактированию файла **named.conf.local**:

```
nano /etc/bind/named.conf.local
```

Удалите все его текущее содержимое. Добавим туда следующее:

```
include "/etc/bind/rndc.key"; controls {  
    inet 127.0.0.1 allow { localhost; } keys { rndc-key; };  
};  
  
zone "STUDENT.GROUP.local" IN { type master;  
    file "/var/lib/bind/forward.db"; allow-update { key rndc-key; };  
};  
  
zone "N.168.192.in-addr.arpa" IN { type master;  
    file "/var/lib/bind/reverse.db"; allow-update { key rndc-key; };  
};
```

Рассмотрим написанное, более подробно:

Пункт **include key** содержит информацию о ключе, который позволит настройку автообновление зоны дальше по ходу практической работы.

Пункт **zone «STUDENT.GROUP.local»**-зона, которую обслуживает наш DNS сервер, в нашем случае **STUDENT.GROUP.local**. Также там описывается тип зоны master и путь к файлу, где будут храниться данные зоны, последний пункт разрешает обновление данного файла, только с использованием ключа.

Пункт **zone «N.168.192.in-addr.arpa»** отвечает за создание зоны обратного просмотра.

Теперь нам необходимо создать сами файлы, в которых будут храниться данные зоны

SERVER_HOSTNAME.STUDENT.GROUP.local.

Они будут располагаться в **/var/lib/bind/**, сделано это по одной простой причине - группа **bind** не имеет права на запись в **/etc/bind/**, а начинать менять права на системные директории, - не очень хорошая идея.

Переходим к созданию файла настроек зоны, создадим зону прямого просмотра и назовём его **forward.db**:

```
nano /var/lib/bind/forward.db
```

С содержимым:

```
$TTL 86400
STUDENT.GROUP.local IN SOA SERVER_HOSTNAME.STUDENT.GROUP.local.
admin.STUDENT.GROUP.local (
    20110103      ; Serial - increment after save
    604800        ; Refresh
    86400         ; Retry
    2419200       ; Expire
    604800 )      ; Negative Cache TTL
;

IN NS SERVER_HOSTNAME.STUDENT.GROUP.local
IN A 192.168.N.1
localhost IN A 127.0.0.1

SERVER_HOSTNAME IN A 192.168.N.1
```

Обратите внимание что в вторая строка пишется слитно без абзаца:

```
STUDENT.GROUP.local.IN SOA
                        SERVER_HOSTNAME.STUDENT.GROUP.local
```

1.

```
admin.STUDENT.GROUP.local. (
SERVER_HOSTNAME – Имя
сервера
```

Из содержимого понятно, что наша зона **STUDENT.GROUP.local**, в которой присутствует DNS сервер (он же шлюз) с именем **SERVER_HOSTNAME**, который имеет IP-адрес **192.168.N.1**. Других записей делать не будем т.к. их будет добавлять DHCP-сервер автоматически.

Если вдруг вам понадобится создать запись вручную, то достаточно добавить строку с именем и IP-адресом узла локальной сети.

Теперь создадим файл зоны обратного просмотра, чтобы не выдумывать ничего, назовём его **reverse.db**

```
nano /var/lib/bind/reverse.db
```

С содержимым:

```
TTL 86400 ; 1 day
N .168.192.in-addr.arpa. IN SOA STUDENT.GROUP.local. STUDENT.GROUP.local. (
                                20110104
; Serial
                                10800
; Refresh
                                3600
; Retry
                                604800
; Expire
                                3600 )
; Minimum
                                IN NS
SERVER_HOSTNAME.STUDENT.GROUP.local.
1 IN PTR STUDENT.GROUP.local.
1 IN PTR
SERVER_HOSTNAME.STUDENT.GROUP.local.
```

Выходим и запускаем bind

```
systemctl restart bind9
```

Теперь нам необходимо проверить работоспособность нашего сервера локально. Необходимо поправить запись dns-nameservers в настройках сети:

```
nano /etc/netplan/00-installer-config.yaml
```



```
network:
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 10.0.2.15/24
      gateway4: 10.0.2.2
    enp0s8:
      dhcp4: no
      addresses:
        - 192.168.N.1/24
      nameservers:
        addresses: [192.168.N.1]
        search:
          [STUDENT.GROUP.local]
```

Перезагрузите сервер.

Проверим самоопределение имени нашего сервера:

```
nslookup SERVER_HOSTNAME
```

В ответ получим

```
Server: 127.0.0.1
Address: 127.0.0.53#53

Name: gateway
Address: 127.0.1.1
```

Эту операцию, также, можно выполнить и через **host**, сделав запрос вида:

```
host SERVER_HOSTNAME
```

Если имя преобразовано в IP значит зона прямого просмотра работает нормально. Теперь проверим зону обратного просмотра:

```
nslookup 192.168.N.1
```

Нам выдаст:

```
Server: 127.0.0.1
```

```
Address: 127.0.0.1#53
```

```
1.N.168.192.in-addr.arpa name = SERVER_HOSTNAME.STUDENT.GROUP.local
```

```
1.N.168.192.in-addr.arpa name = STUDENT.GROUP.local
```

Если ответ получен, значит обратное разрешение имён работает.

Отладка

Если желаемого результата не получилось, то перезапускаем сервис bind.

```
systemctl restart bind9
```

Файл с системными логами поможет найти опечатки в конфигурационных файлах.

```
tail -40 /var/log/syslog
```

Настраиваем динамическое обновление зон DHCP-сервером

Скопируем файл с ключами из Bind, так как у сервиса dhcp не хватит прав доступа к каталогу Bind.

```
cp -Rr /etc/bind/rndc.key /etc/dhcp/ddns-keys
```

Отредактируем конфигурационный файл DHCP-сервера:

```
nano /etc/dhcp/dhcpd.conf
```

```
subnet 192.168.N.0 netmask 255.255.255.0 {  
    range 192.168.N.10 192.168.N.254;  
    option domain-name-servers 192.168.N.1;  
    option domain-name "STUDENT.GROUP.local";  
    option routers 192.168.N.1;  
    option broadcast-address 192.168.N.255;  
    default-lease-time 604800;  
    max-lease-time 604800;  
}
```

Заодно добавим строки, которые отвечают за автоматическое создание прямой и обратной зоны для клиентов с динамическими IP, без нее записи придется создавать вручную, а мы этот процесс

автоматизируем.

```
include "/etc/dhcp/ddns-keys/rndc.key"; ddns-updates on;

ddns-update-style standard;
ddns-domainname "STUDENT.GROUP.local";


zone STUDENT.GROUP.local. { primary 192.168.N.1;
key rndc-key;
}

zone N.168.192.in-addr.arpa. { primary 192.168.N.1;
key rndc-key;
}
```

В результате всех действий, у вас должен получиться конфигурационный файл вида:

Перезапустим DNS и DHCP-сервер:

```
authoritative;

include "/etc/dhcp/ddns-keys/rndc.key"; ddns-updates on;
ddns-update-style standard;
ddns-domainname "STUDENT.GROUP.local";


zone STUDENT.GROUP.local. { primary 192.168.N.1;
key rndc-key;
}

zone N.168.192.in-addr.arpa. { primary 192.168.N.1;
key rndc-key;
}


subnet 192.168.N.0 netmask 255.255.255.0 { range 192.168.N.10 192.168.N.254;
option domain-name-servers 192.168.N.1; option domain-name "STUDENT.GROUP.local"; option routers
192.168.N.1;

option broadcast-address 192.168.N.255; default-lease-time 604800;

max-lease-time 604800;
}
```

```
systemctl restart bind9  
service isc-dhcp-server restart
```

Подключаем наш клиентский хост к локальной сети, предположим, что его имя DESKTOP_NAME01, он получит IP от нашего DHCP-сервера, а DHCP-сервер создаст DNS запись вида DESKTOP_NAME01.STUDENT.GROUP.local.
Пробуем выполнить запрос по имени

```
nslookup DESKTOP_NAME01
```

Нам должен возвращаться IP-адрес, который получила клиентская машина от DHCP-сервера. Ну и преобразование IP-адреса в имя, также должно работать.

Во время работы данной связки, у нас будут создаваться файлы в директории **/var/lib/bind** с расширением. **jnl**, они создаются автоматически, создавать или удалять их не нужно.

Отладка

Если желаемого результата не получилось, то перезапускаем сервис dhcp.

```
service isc-dhcp-server restart
```

Файл с системными логами поможет найти опечатки в конфигурационных файлах.

```
tail -40 /var/log/syslog
```

P.s. Рекомендую сделать снимок этой части на сервере, поскольку далее работа будет идти с этой точки, а откатываться вы вполне возможно будете

Практическая работа №6

Облачное файловое хранилище Seafile.

Подготовка

1. Клонировать **отдельную** виртуальную машину из золотого образа. Имя VM: **seafile**.(Ubuntu 22.04)
Сгенерировать новый MAC-адрес при клонировании.
2. Настроить **статический** IP на VM **seafile**.

Редактируем файл с настройками

```
nano /etc/netplan/00-installer-config.yaml
```

```
network:
ethernets:

    enp0s3:
        dhcp4: no
        addresses:
            - 192.168.N.4/24

        gateway4: 192.168.N.1
        nameservers:
            addresses: [192.168.N.1]
            search: [STUDENT.GROUP.local]
```

p.s. если после написания этого у вас пропадет Интернет, перепроверьте написание файла. Если Интернета так и не будет – верните настройки этого файла как в 4 или 5 работе.

Сохраняем изменения, выходим.

Применяем конфигурацию:

```
netplan apply
```

Проверим правильность настройки командой:

```
ping gateway
ping ya.ru
```

3. **Переименовать** виртуальный сервер в seafile.

Для этого выполняем команды (Не забываем поднять права до root):

```
hostnamectl set-hostname seafile
```

Далее поправим имя сервера в файле /etc/hosts:

```
nano /etc/hosts
```

```
127.0.0.1 localhost  
127.0.1.1 seafile
```

Перезагружаем сервер seafile.

4. Добавить **прямую и обратную** запись в домен

[STUDENT.GROUP.local](#). Подключаемся к серверу **gateway**, который является

DNS сервером.

Редактируем файл

```
nano /var/lib/bind/forward.db
```

Добавим строку в конце файла

```
seafile    IN      A      192.168.N.4
```

Выходим и запускаем bind

```
systemctl restart bind9
```

Проверяем работу DNS

```
nslookup seafile
```

Выполнение

Повышаем права до root:

```
sudo su
```

Устанавливаем необходимые пакеты:

```
apt-get update  
apt install python3 python3-setuptools python3-pip libmysqlclient-dev
```

С помощью менеджера пакетов python также установим:

```
pip3 install --timeout=3600 django==3.2.* Pillow pylibmc captcha jinja2  
sqlalchemy==1.4.3 django-pylibmc django-simple-captcha python3-ldap  
mysqlclient pycryptodome==3.12.0 cffi==1.14.0
```

В качестве сервера баз данных будем использовать MariaDB. Устанавливаем БД следующей командой:

```
apt install mariadb-server
```

Установим пароль для учётной записи root MySQL:

```
mysqladmin -u root password
```

Система запросит новый пароль. Его нужно ввести дважды.

Чтобы наш пароль применился, нужно сбросить привилегии в СУБД. Для этого заходим в оболочку sql:

```
mysql
```

И вводим:

```
flush privileges;  
\q;
```

Разрешаем автозапуск демона СУБД:

```
systemctl enable mariadb
```

Мы готовы к переходу к установке самого сервера seafile.

Для начала создадим директорию для сервера, назовём её **seafile**:

```
mkdir /opt/seafile
```

перейдём в неё:

```
cd /opt/seafile
```

Скачиваем свежую версию дистрибутива

```
wget https://s3.eu-central-1.amazonaws.com/download.seadrive.org/seafile-server_9.0.9_x86-64.tar.gz
```

p.s. возможно на момент выполнения практической работы версия устареет. Поставьте более новую
Распаковываем скаченное:

```
tar -xzf seafile-server_9.0.9_x86-64.tar.gz
```

Назначим в качестве владельца каталога /seafile нашего пользователя.

```
chown -R STUDENT:STUDENT /opt/seafile/
```

Где STUDENT - ваш пользователь в системе.

Запускаем скрипт установки seafile сервера:

```
cd ./seafile-server-9.0.9/  
./setup-seafile-mysql.sh
```

Вводите название сервера: seafile вводите IP сервера: 192.168.N.4 Принимаем порт 8082 по умолчанию.

Создаём новую базу данных:

```
-----  
[1] Create new ccnet/seafile/seahub databases  
[2] Use existing ccnet/seafile/seahub databases [ 1  
or 2 ] 1
```

Адрес по умолчанию для сервера MySQL:

```
What is the host of mysql server?  
[ default "localhost" ]
```

Укажите порт для сервера MySQL:

```
What is the port of mysql server?  
[ default "3306" ]
```

Введите пароль от root MySQL:

```
What is the password of the mysql root user?  
[ root password ]
```

Далее будут созданы пользователь MySQL для seafile и БД, принимаем все значения по умолчанию. На этом установка seafile закончена.

Дополнительно к серверу БД и seafile нам необходим веб-сервер.

Установим web-сервер Nginx на нашу систему.

```
apt install nginx -y
```

Создадим файл с настройками для сервера seafile.

```
touch /etc/nginx/sites-available/seafile.conf
```

Создадим симлинк на наш файл конфигурации.

```
ln -s /etc/nginx/sites-available/seafile.conf /etc/nginx/sites-enabled/seafile.conf
```

Удалим дефолтный файл конфигурации Nginx (если он есть).

```
rm /etc/nginx/sites-enabled/default
```

Далее отредактируем файл с настройками

```
nano /etc/nginx/sites-enabled/seafile.conf
```

```
server {  
    listen 192.168.N.4:80;  
    server_name seafile.lan;  
    index index.html;  
    location / {  
        proxy_pass http://127.0.0.1:8000;  
        proxy_set_header Host $host;  
        proxy_set_header X-Forwarded-For  
$proxy_add_x_forwarded_for;  
        proxy_set_header X-Real-IP $remote_addr;  
        proxy_buffering off;  
    }  
}
```

*Обратите внимание что **N** - ваш номер варианта, а `server_name seafile.lan`; пишется без изменений. Перезагружаем Web-сервер Nginx.

```
service nginx restart
```

Теперь можно запустить сервис seafile.

```
/opt/seafile/seafile-server-9.0.9/seafile.sh start  
/opt/seafile/seafile-server-9.0.9/seahub.sh start
```

При первом запуске будет предложено создать аккаунт администратора `admin@STUDENT.GROUP.local`

Далее настроим автозапуск приложения.

```
nano /etc/systemd/system/seafile.service

[Unit]
Description=Seafile
After= mariadb.service
After=network.target

[Service]

Type=forking

ExecStart=/opt/seafile/seafile-server-9.0.9/seafile.sh start

ExecStop=/opt/seafile/seafile-server-9.0.9/seafile.sh stop

[Install]

WantedBy=multi-user.target
```

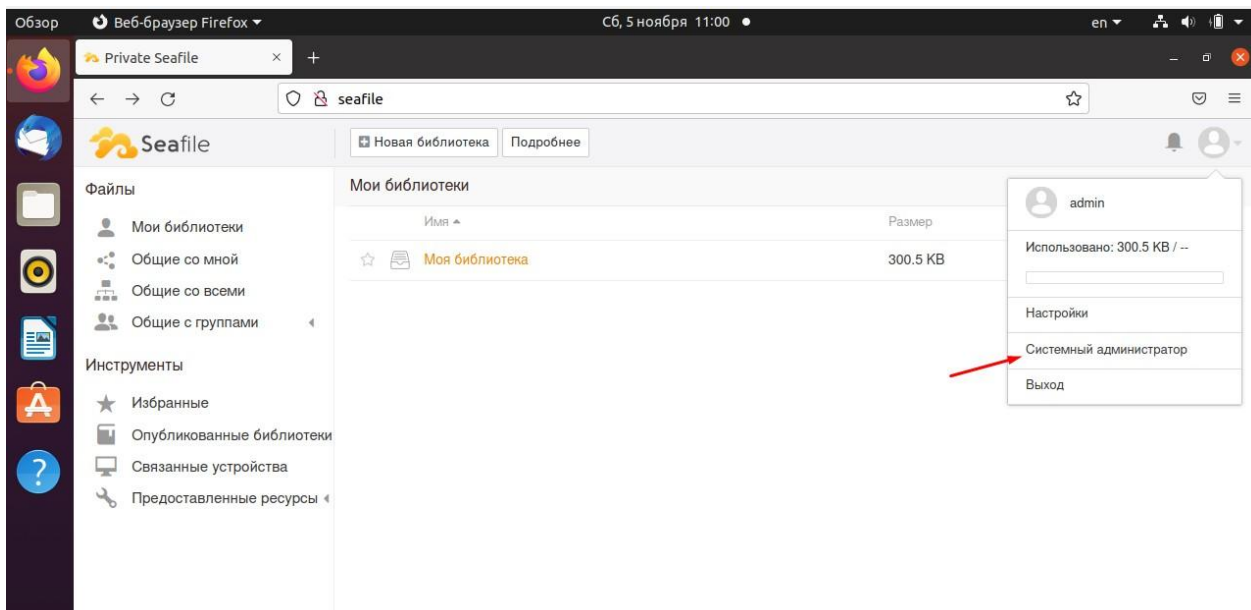
Для работы с сервером необходим доступ через веб-консоль, подключаемся к ВМ десктопу и запускаем браузер.

Переходим по адресу:

```
http://seafile
```

Используем учётную запись администратора admin@STUDENT.GROUP.local, созданного заранее.

Создаём пользователя:



Скачиваем и устанавливаем клиент seafile на Desktop

Задача:

Настроить клиент Seafile на Desktop, синхронизировать библиотеки, продемонстрировать результат.

```
sudo su  
add-apt-repository ppa:seafile/seafile-client  
apt-get update  
apt-get install seafile-gui
```

Системный администратор

- Инфо
- Устройства
- Настройки
- Библиотеки
- Пользователя(ей)**
- Группы
- Уведомления
- Ссылки

Имя / Email		Статус	Использовано пространства / Квота	Создано / Последний вход / Последний доступ
<input type="checkbox"/>	admin admin@svinarev.ia131.local	Активный	300.5 KB / --	2022-11-05 09:37 / минуту назад минуту назад
<input type="checkbox"/>	user1 user1@svinarev.ia131.local	Активный	309.0 KB / --	2022-11-05 09:51 / час назад час назад

< 1 > 25 / Страница ▼

Практическая работа №7

Электронная почта.

Подготовка

1. Клонировать из золотого образа новую виртуальную машину с 1 сетевым интерфейсом «Внутренняя сеть intnet». (Ubuntu 22.04)
2. Настроить **статический** IP. Нужно отредактировать файл

```
nano /etc/netplan/01-netcfg.yaml

network:
version: 2

renderer: networkd
ethernets:

    enp
    os3:
    dhcp4: no

    addresses: [192.168.N.5/24]
    gateway4: 192.168.N.1
```

Где N – Номер варианта в журнале.
После этого применяем изменения
сети

```
netplan apply
```

Проверить можно командой ip и уже известными вам командами ping и nslookup

```
ip a
```

3. **Переименовать** виртуальный сервер в **mail**.
4. Добавить **прямую и обратную** запись в домен.

Логинимся в систему, обновляем списки пакетов:

```
sudo apt-get update
```

Редактируем файл hosts:

```
nano /etc/hosts
```

Получаем:

```
127.0.0.1    mail.STUDENT.GROUP.local mail localhost
```

```
192.168.N.5 mail.STUDENT.GROUP.local mail
```

Скачиваем **iRedMail**

```
wget https://github.com/iredmail/iRedMail/archive/refs/tags/1.6.2.tar.gz
```

Распаковываем архив:

```
tar xvf 1.6.2.tar.gz
```

Переходим в директорию, созданную после распаковки:

```
cd iRedMail-1.6.2
```

Выкачиваем пакеты необходимые для установки:

```
cd ./pkgs/
```

```
chmod +x get_all.sh
```

```
./get_all.sh
```

Приступаем к установке:

```
cd ..
```

```
chmod +x iRedMail.sh
```

```
./iRedMail.sh
```

После этого начнётся загрузка необходимых пакетов.
В открывшемся меню установки выбираем **Yes**

[illegible]

Выбираем тип веб-сервера: Nginx

```

Preferred web server
Choose a web server you want to run.

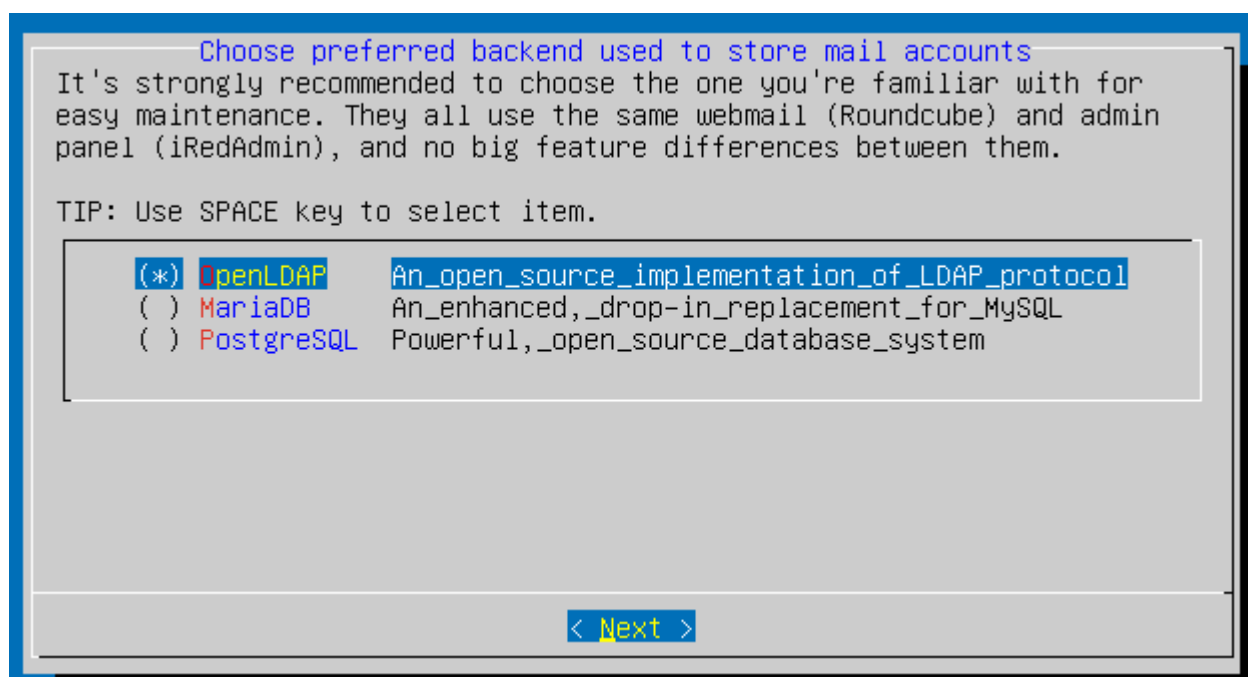
TIP: Use SPACE key to select item.

(*) Nginx           The fastest web server
( ) No web server    I don't need any web applications on this server

< Next >

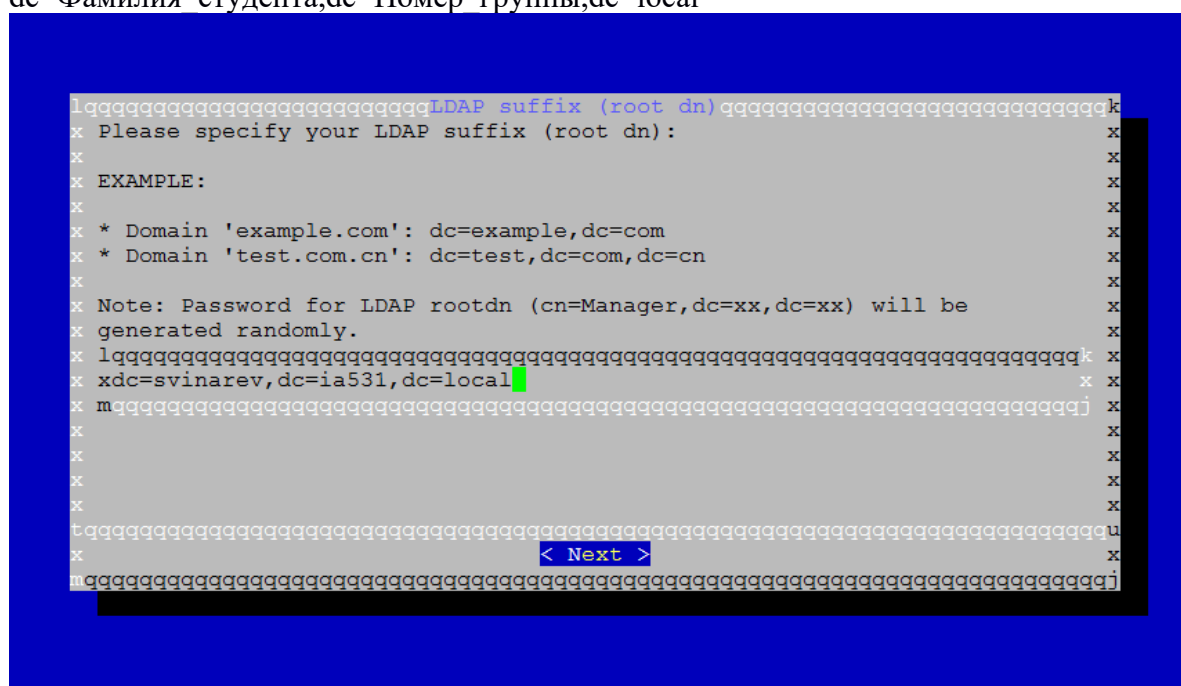
```

Выбираем тип базы данных: OpenLdap



Указываем имя домена:

dc=Фамилия студента,dc=Номер группы,dc=local



Указываем пароль администратора базы данных


```

lqqqqqqqqqqqqqqqqqqqqqqYour first mail domain nameqqqqqqqqqqqqqqqqqqqqqqk
x Please specify your first mail domain name. x
x x x
x EXAMPLE: x
x x x
x * example.com x
x x x
x WARNING: x
x x x
x It can *NOT* be the same as server hostname: mail.svinarev.ia531.local. x
x x x
x We need Postfix to accept emails sent to system accounts (e.g. root), if x
x your mail domain is same as server hostname, Postfix won't accept any x
x email sent to this mail domain. x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x xsvinarev.ia531.local x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x < Next > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj

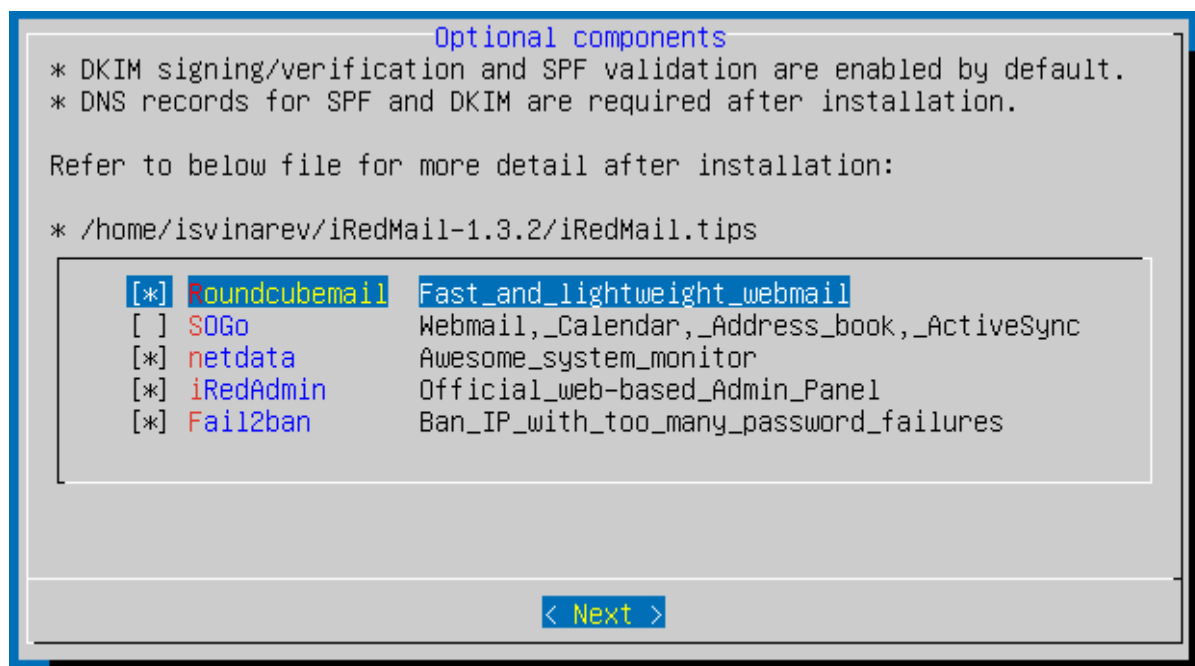
```

Устанавливаем пароль почтового администратора:

```

lqqqqqqqqqqqqqqqqqqqqqqPassword for the mail domain administratorqqqqqqqqqqqqqqqqqqqqk
x Please specify password for the mail domain administrator: x
x x x
x * postmaster@svinarev.ia531.local x
x x x
x You can login to webmail and iRedAdmin with this account. x
x x x
x WARNING: x
x x x
x * Do *NOT* use special characters in password right now. e.g. $, #, @. x
x * EMPTY password is *NOT* permitted. x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x***** x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
x x x
x x x
x x x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
x < Next > x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj

```



На оставшиеся вопросы отвечаем утвердительно и перезагружаем сервер по завершению установки.

С этим разобрались, теперь необходимо перейти по адресу **https://mail.Имя_домена/iredadmin**, далее заходим в систему управления нашим почтовым сервером, имя пользователя **postmaster@Имя_домена**, пароль тот что был указан при установке.

Создайте пользователя и перейдите в консоль почтового ящика по адресу **https://mail.Имя_домена/mail**.

Задание: Создать пользователя и отправить электронное письмо себе на почту либо между двумя пользователями почтового сервера.

Практическая работа №8

Система управления контентом WordPress.

Подготовка

1. Клонировать **отдельную** виртуальную машину Ubuntu Server. (22.04)
2. Настроить **статический** IP 192.168.N.6.
3. **Переименовать** виртуальный сервер в **wordpress**.
4. Добавить **прямую и обратную** запись в домен.

Выполнение

Все команды выполняются на ВМ wordpress.

Установка LAMP

Повышаем права до root:

```
sudo su
```

Обновляем список пакетов.

```
apt-get update
```

Устанавливаем LAMP (Linux-Apache-MySQL-PHP) сервер на базе Ubuntu, в который входят Apache 2, PHP 5, и MySQL 5.

```
apt-get -y install tasksel  
tasksel install lamp-server
```

Откроем файл /etc/apache2/apache2.conf командой:

```
nano /etc/apache2/apache2.conf
```

P.s. если у вас возникла проблема, попробуйте установить Apache2 вручную

```
sudo apt install apache2
```

И в конец всего содержимого впишем без каких-либо изменений:

```
ServerName localhost
```

Сохраним. Перезапустим Apache 2 командой:

```
systemctl restart apache2.service
```

Изменим права на содержимое каталога www:

```
cd /var  
chown -R ваш_логин_в_системе:ваш_логин_в_системе www  
chmod -R 755 /var/www
```

Где указываете имя вашего пользователя, например isvinarev:isvinarev.

Создание базы данных

Для управления и хранения информации о сайтах и пользователях WordPress использует реляционную базу данных.

У нас установлен MySQL, который и выполняет эту функцию, но нужно создать базу данных и пользователя, с которыми будет работать WordPress.

Для начала залогиньтесь в root-аккаунт (административный аккаунт) MySQL при помощи следующей команды:

```
mysql -u root -password
```

Создаём отдельную базу данных **wordpress**, которой WordPress может управлять.

Все операторы MySQL должны заканчиваться точкой с запятой (;), поэтому в случае возникновения проблем прежде всего проверьте этот момент.

Затем создадим отдельный пользовательский аккаунт MySQLc именем **author** и паролем **P@ssw0rd**, который мы будем использовать исключительно для работы с новой базой данных.

```
mysql> create database wordpress character set utf8 collate utf8_bin;

Query OK, 1 row affected (0.00 sec)

mysql> create user 'author'@'localhost' identified by 'P@ssw0rd';

mysql> grant all privileges on wordpress.* to author@localhost;

Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;

Query OK, 0 rows affected (0.00 sec)
mysql> exit
```

Загрузка WordPress

Теперь мы загрузим файлы WordPress с вебсайта этой программы.

```
cd ~
wget https://ru.wordpress.org/latest-ru_RU.tar.gz
```

В ваш домашний каталог загрузится сжатый файл, который содержит заархивированные каталоги файлов WordPress.

При помощи следующей команды мы можем извлечь файлы для восстановления нужного нам каталога WordPress:

```
tar xzvf latest-ru_RU.tar.gz
```

В вашем домашнем каталоге будет создан каталог под названием **wordpress**.

Конфигурация WordPress

Перейдите в каталог WordPress, который вы недавно распаковали:

```
cd ~/wordpress
```

Эталонный файл конфигурации, который практически полностью совпадает с необходимой нам конфигурацией, включен туда по умолчанию. Однако нам нужно скопировать его в стандартное местоположение файла конфигурации, чтобы WordPress распознал этот файл. Для этого введите следующее:

```
cp wp-config-sample.php wp-config.php
```

Теперь у нас есть файл конфигурации. Откройте его в текстовом редакторе:

```
nano wp-config.php
```

Единственное, что нам нужно изменить, – это параметры, содержащие информацию о нашей базе данных.

Нам нужно найти настройки для `DB_NAME`, `DB_USER` и `DB_PASSWORD`, чтобы WordPress правильно подключился к созданной нами базе данных и опознал ее.

В качестве значений этих параметров введите информацию о созданной базе данных. Вот так все должно выглядеть:

```
* * MySQL settings
```

```
* Secret keys
```

```
* Database table prefix
```

```
* ABSPATH
```

```
@link https://codex.wordpress.org/Editing\_wp-config.php
```

```
@package WordPress
```

```
*/
```

```
// ** MySQL settings - You can get this info from your web host ** //
```

```
/** The name of the database for WordPress */
```

```
define('DB_NAME', 'wordpress');
```

```
/** MySQL database username */
```

```
define('DB_USER', 'author');
```

```
/** MySQL database password */
```

```
define('DB_PASSWORD', 'P@ssw0rd');
```

```
/** MySQL hostname */
```


Теперь, когда мы конфигурировали наше приложение, нам нужно скопировать его в корень документа Apache, где он будет предоставлен посетителям вебсайта.

Один из самых простых и надежных способов переноса файлов из каталога в каталог - при помощи команды rsync. При использовании такого способа сохраняются разрешения и поддерживается целостность данных.

В руководстве по Ubuntu LAMP корень документа расположен по адресу: /var/www/html/. Чтобы перенести туда файлы WordPress, введите:

```
rsync -avP ~/wordpress/ /var/www/html/
```

Также удалите файл index.html из этой же директории

```
rm /var/www/html/index.html
```

В корень документа будут безопасно скопирован весь контент из распакованного вами каталога.

Завершение установки через веб-интерфейс

Теперь, когда все файлы размещены в нужных местах и программное обеспечение конфигурировано, вы можете завершить установку через веб-интерфейс.

В своем веб-браузере перейдите к доменному имени своего сервера:

http://wordpress

Вы увидите страницу начальной конфигурации WordPress, на которой вы создадите начальную учетную запись администратора:

Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

Password, twice
A password will be automatically generated for you if you leave this blank.

Strength indicator
Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers, and symbols like ! " ? \$ % ^ & .

Your E-mail
Double-check your email address before continuing.

Privacy ☒ Allow search engines to index this site.

Поля на скриншоте: Имя сайта, Имя пользователя, Пароль (ввести дважды), Электронная почта, Конфиденциальность (Разрешить поисковым системам индексировать этот сайт)

Введите информацию, касающуюся вашего сайта и создаваемой вами учетной записи администратора. По завершении нажмите на клавишу «Установить WordPress» в низу страницы. WordPress подтвердит установку и затем пригласит вас войти в созданную вами учетную запись:




Success!

WordPress has been installed. Were you expecting more steps? Sorry to disappoint.

Username admin

Password *Your chosen password.*

Нажмите на клавишу внизу страницы и введите следующую информацию:

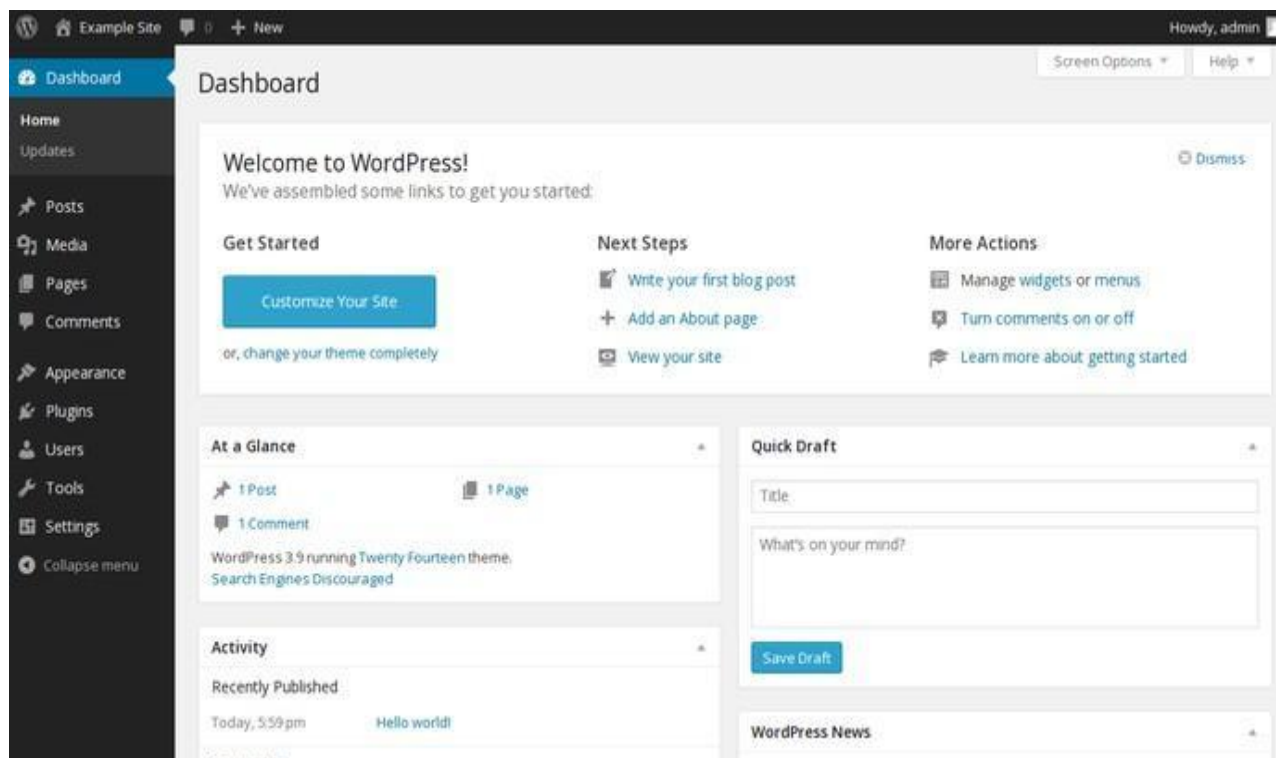


Username

Password

☐ Remember Me

Вы увидите интерфейс WordPress:



Задача: Создайте несколько записей в блоге.

Просмотреть их как простой пользователь можно выйдя из аккаунта и открыв <http://wordpress>.

PrivateBin – сервер передачи паролей и информации.

Подготовка:

1. Клонировать из золотого образа новую виртуальную машину “privatebin” с 1 сетевым интерфейсом «Внутренняя сеть intnet».
2. Настроить **статический IP**
Нужно отредактировать файл

```
nano /etc/netplan/00-installer-config.yaml

network: version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.N.7/24]
      gateway4: 192.168.N.1
      nameservers:
        addresses: [192.168.N.1]
        search: [STUDENT.GROUP.local]
```

где N – Номер варианта в журнале.

После этого применяем изменения сети

```
netplan apply
```

Проверить можно командой ip и уже известными вам командами ping и nslookup

```
ip a
```

3. **Переименовать** виртуальный сервер в privatebin, отредактировав файлы /etc/hostname и /etc/hosts. Далее перезагрузить ВМ.
4. Добавить прямую и обратную запись в домен.

Выполнение

Все команды выполняются на виртуальной машине «privatebin».

Устанавливаем веб-сервер Apache и PHP:

```
apt update  
apt install -y apache2 php php-xml php-mbstring php-mysql php-json php-pdo
```

Нам потребуется создать самоподписанный SSL-сертификат для возможности подключения через протокол HTTPS к сервису privatebin, так как самостоятельно он не может его создать, в отличие от сервисов в предыдущих работах.

Запускаем веб-сервер

```
systemctl enable --now apache2.service
```

Включаем mod_ssl для него:

```
a2enmod ssl
```

Перезагружаем веб-сервер

```
systemctl restart apache2
```

Создаем сертификат и располагаем его в каталоге /etc/ssl/.

Обратите внимание, что эта команда пишется в одну строку.

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -  
out /etc/ssl/certs/apache-selfsigned.crt
```

Заполняем опросник:

```
Country Name (2 letter code) [XX]:RU  
State or Province Name (full name) []:Novosibirsk  
Locality Name (eg, city) [Default City]: Novosibirsk  
Organization Name (eg, company) [Default Company Ltd]:Sibsutis  
Organizational Unit Name (eg, section) []:IA131  
Common Name (eg, your name or your server's hostname) []:privatebin.student.group.local
```

Настраиваем веб-сервер на работу по HTTPS (порт 443) и перенаправлении с HTTP

(порт 80) на HTTPS.

```
nano /etc/apache2/sites-available/privatebin.conf
```

В самом файле добавляем строки:

```
<VirtualHost *:80> ServerName 192.168.N.7

Redirect / https://192.168.N.7/

</VirtualHost>

<VirtualHost *:443> ServerName 192.168.N.7

DocumentRoot /var/www/html/PrivateBin/ SSLEngine on

SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt

SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

ErrorLog ${APACHE_LOG_DIR}/privatebin-error.log

CustomLog ${APACHE_LOG_DIR}/privatebin-access.log combined

<Directory /var/www/html/PrivateBin>

AllowOverride All

</Directory>

</VirtualHost>
```

Где N – Номер варианта в журнале.

Активируем конфигурацию и перезапускаем веб-сервер:

```
a2ensite privatebin.conf
systemctl reload
```

Скачиваем и устанавливаем PrivateBin из репозитория git

```
cd /var/www/html/ && git clone https://github.com/PrivateBin/PrivateBin.git
```

И назначаем пользователя владельцем каталога:

Обратите внимание, что вы указываете имя своего пользователя

```
chown -R "Ваш пользователь":"Ваш пользователь" PrivateBin/
chmod 777 -R PrivateBin/
```

Например, у меня выходит так:

```
chown -R isvinarev:isvinarev PrivateBin/  
chmod 777 -R PrivateBin/
```

Далее переместим данные в корень каталога

```
rsync -avP ./PrivateBin/ /var/www/html/
```

Также удалите файл index.html из этой же директории

```
rm /var/www/html/index.html
```

Для доступа к сайту используем ВМ **Ubuntu Desktop**.

Теперь вы можете открыть веб-сервер по адресу <https://privatebin.student.group.local> или короткое <https://privatebin> и использовать его для передачи паролей в виде ссылок.

Задача:

В паре с практической работой №7 переслать пароль между двумя пользователями используя электронную почту и сервер передачи паролей PrivateBin.