Лабораторная работа №1
«Основы сетевого администрирования»

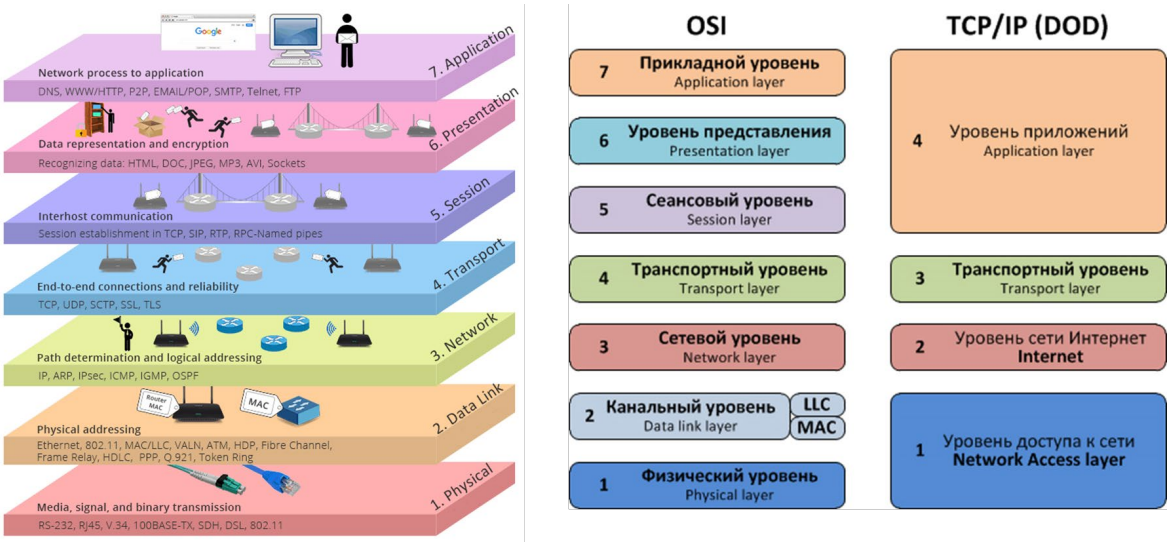
1. Теоретическая часть

1.1. Введение в сетевое администрирование

Локальная вычислительная сеть (ЛВС, локальная сеть; англ. Local Area Network, LAN) — компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий (дом, офис, фирму, институт).

В сети должен происходить обмен данными между вычислительными устройствами — компьютерами, серверами, маршрутизаторами, принтерами и другим оборудованием. Для передачи информации могут быть использованы различные среды передачи данных (витая пара, оптоволокно, радиоволны).

Всю работу сети можно проанализировать как по модели OSI (англ. The **O**pen **S**ystems **I**nterconnection model) , так и по модели TCP/IP.



В обязанности сетевого администратора входит построение ЛВС, её настройка, поддержка и улучшение. И это лишь малая часть, чем может заниматься специалист в данной области. Обязанности могут различаться в

зависимости от масштабов сетей, специфики работы какой-либо компании и т.д.

1.2. Роль Linux в управлении ЛВС

Несмотря на то, что в локальную сеть могут входить пользовательские компьютеры, где чаще всего используется операционная система Windows для выполнения повседневных задач, администрирование сети происходит с помощью устройств (серверов) под управлением операционной системы на базе ядра Linux.

Используются как широко распространённые дистрибутивы GNU/Linux с графическим интерфейсом, так и дистрибутивы, предоставляющие для управления только командную строку. Это связано с тем, что использовать GNU/Linux на серверах выгодно, потому что эта операционная система бесплатна и можно сразу же развернуть нужный образ GNU/Linux на сервере. Это обеспечивает надёжность, гибкость и масштабируемость.

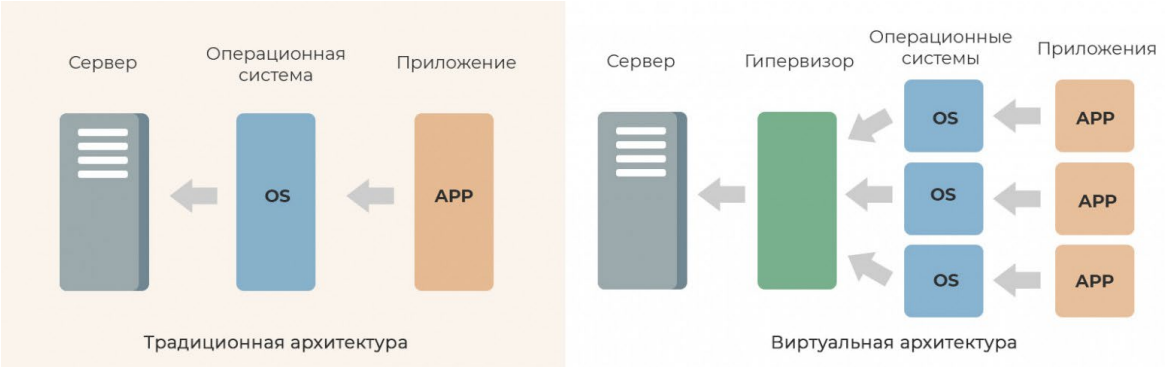
1.3. Виртуальная машина

Каждому администратору сети приходится иметь дело с настройкой виртуальных машин. Они необходимы для разработки и тестирования приложений, а также хранения данных. Преимуществами виртуальных машин можно назвать гибкость в выборе операционной системы и возможность дублирования рабочего пространства.

Виртуальная машина (VM, от англ. virtual machine) — программная или аппаратная система, эмулирующая аппаратное обеспечение компьютера и исполняющая программы для guest-платформы (guest — гостевая платформа) на host-платформе (host — хост-платформа, платформа-хозяин).

На одном хост-компьютере может быть множество гостевых ВМ. Хотя виртуальная машина создается с помощью ПО, она использует физические

ресурсы хост-машины, такие как ЦП, ОЗУ и место в хранилище на жёстком диске. На своём хост-компьютере можно настроить столько виртуальных машин, сколько нужно, но придётся разделить физические аппаратные ресурсы между ними всеми. Однако большинство ВМ будут работать медленнее, чем физический компьютер, просто из-за дополнительных уровней абстракции, которые они должны пройти для выполнения функции.



Видов виртуализации существует несколько:

Программная виртуализация – вид виртуализации, который использует различные библиотеки ОС, транслируя вызовы виртуальной машины в вызовы ОС. Например, DOSBox, Virtualbox, VirtualPC.

Аппаратная виртуализация – вид виртуализации, который предусматривает специализированную инструкцию аппаратной части, а конкретно инструкций процессора. Позволяет исполнять запросы в обход гостевой ОС прямо на аппаратном обеспечении.

Программная виртуализация обеспечивается гипервизором.

Гипервизор (англ. Hypervisor) или монитор виртуальных машин — программа или аппаратная схема, обеспечивающая одновременное, параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере.

Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами.

1.4. Виртуализация QEMU/KVM в Astra Linux

Для исполнения прямых аппаратных запросов в ОС должна иметься библиотека, которая направляла бы эти запросы аппаратной части напрямую. На платформах базы Linux долгое время никакой встроенной системы виртуализации (встроенного гипервизора), просто не существовало. Каждый производитель ПО для виртуализации, который поддерживало технологию аппаратной виртуализации, вынуждены были создавать собственные модули для ядра Linux.

Однако со временем был разработан свободный базовый гипервизор **KVM** или **Kernel-based Virtual Machine**, который представляет из себя загружаемый модуль ядра Linux и предназначен для обеспечения виртуализации на платформ **x86** с поддержкой одной из технологий аппаратной виртуализации — **Intel VT** либо **AMD SVM**.

Сам по себе KVM не выполняет эмуляции. Вместо этого сторонняя программа, работающая в пространстве пользователя, использует интерфейс **/dev/kvm** для настройки адресного пространства гостя виртуальной машины. Например, свободная программа **QEMU** (Quick Emulator) использует KVM для эмуляции аппаратного обеспечения различных платформ.

В Astra Linux по умолчанию используется менеджер виртуальных машин **virt-manager**, представляющий собой пользовательский интерфейс для управления виртуальными машинами на рабочем столе с помощью API **libvirt**. В нём представлена сводная информация о запущенных доменах, их текущей производительности и статистике использования ресурсов. Мастера позволяют

создавать новые домены, а также настраивать распределение ресурсов домена и виртуальное оборудование.

1.4.1. Управление виртуальными машинами с помощью утилиты virsh

virsh — утилита, из состава библиотеки libvirt, для командной строки Linux, предназначенная для управления виртуальными машинами и гипервизорами KVM и Xen. Для подключения к гипервизору необходимо указать параметр --connect qemu:///system.

Основные команды:

```
virsh start <VM name> – запустить виртуальную машину;
virsh suspend <VM name> – остановить виртуальную машину;
virsh resume <VM name> – возобновить работу виртуальной машины;
virsh shutdown <VM name> – выключить виртуальную машину;
virsh reboot <VM name> – перегрузить виртуальную машину;
virsh list --all – вывести список всех виртуальных машин;
virsh snapshot-create-as <VM name> <SN name> – создать снимок;
virsh snapshot-list <VM name> – вывести список всех снимков;
virsh snapshot-revert <VM name> <SN name> – восстановиться из снимка;
virsh snapshot-delete <VM name> <SN name> – восстановиться из снимка;
```

1.5. Понятие службы в Linux

В операционной системе GNU/Linux, кроме обычных программ, существуют постоянно работающие утилиты, работающие в фоне и предоставляющие определенные функции пользователям или системе. Такие программы называются **службами**.

Например, для подключения к компьютеру по протоколу **ssh** (Secure Shell) необходимо постоянно запущенное приложение сервера, принимающее и обрабатывающее запросы. В этом качестве выступает служба **sshd**. Для каждого нового соединения создаётся её экземпляр, который выполняет обмен ключами, шифрование, аутентификацию, выполнение команд и обмен данными.

Для работы со службами используется утилита **systemctl**, имеющий следующий синтаксис:

```
$ systemctl опции команда служба...
```

Опции служат для настройки поведения запущенной программы. Основные команды перечислим ниже:

- start - запустить службу linux;
- stop - остановить службу linux;
- reload - попросить службу перечитать свою конфигурацию из файловой системы;
- restart - перезапустить службу;
- enable - добавить службу в автозагрузку;
- disable - удалить службу из автозагрузки.

Например, с помощью команды:

```
systemctl start sshd
```

возможно запустить службу для работы с ssh-соединениями.

Для определения месторасположения службы (или любой другой стандартной утилиты) возможно воспользоваться командой whereis

```
whereis ssh
```

1.6. Базовые утилиты для работы с сетью

Для работы с сетевыми соединениями необходимо владеть несколькими базовыми инструментами, которые помогут при настройке и анализе сети.

Приведем их в виде таблицы. В столбце «Пример работы» указаны наиболее часто используемые параметры. Более подробно вы можете изучить в документации на каждую из утилит.

Название утилиты	Описание работы	Пример работы	Комментарий
ifconfig	Настройка сетевых интерфейсов	`ifconfig eth0`	Просмотр текущих сетевых соединений Не всегда установлена. В AL выполняется только с правами администратора.
ip	Замена для ifconfig, более мощная утилита для работы с сетевыми интерфейсами.	`ip a` `ip route`	Просмотр текущих сетевых соединений. Просмотр таблицы маршрутизации

ping	Проверка доступности удалённого хоста	`ping google.com` `ping 192.168.122.1`	Возможно указывать IP адрес или доменное имя
netstat/ss	Показ активных соединений и портов	`netstat -tulpn` `ss -tulpn`	Удобно проверять, какие порты заняты какими приложениями. Для указания имен приложений требуются права администратора
traceroute	Показ маршрута до удалённого хоста	`traceroute google.com`	Отслеживает путь, по которому пакеты данных проходят от компьютера до целевого хоста. Помогает выявить, где могут возникать задержки или разрывы связи
nslookup / dig	Проверка DNS-запросов	`nslookup google.com` `dig google.com`	По умолчанию не установлены. Расположены в пакете *dnsutils*.
curl / wget	Получение данных с веб-серверов	`wget http://example.com/file.zip` `curl -I http://example.com`	wget по умолчанию сохраняет загруженные файлы на диск, может рекурсивно загружать сайты
nmap	Сканирование сети и открытых портов	`nmap 192.168.1.1`	По умолчанию не установлен. Расположен в пакете *nmap*.

1.7. Базовые настройки системы GNU/Linux

После установки какого-либо дистрибутива GNU/Linux для подключения к локальной или глобальной сети необходимо произвести базовые настройки. Это возможно сделать несколькими способами – вручную, настраивая конфигурационные файлы или используя сторонние утилиты.

Рассмотрим первый способ настройки. Основные параметры компьютера устанавливаются в следующих конфигурационных файлах:

Путь к файлу	Назначение
/etc/hostname	Настройка имени компьютера
/etc/hosts	Настройка разрешения доменных имен
/etc/resolv.conf	Настройка адресов серверов имен, к которым имеет доступ данная система
/etc/network/interfaces	Настройка сетевых интерфейсов
/etc/apt/sources.list	Настройка списка репозиториев

Рассмотрим эти файлы подробнее:

1. Файл **/etc/hostname** предназначен для настройки имени компьютера. Текущее имя компьютера можно узнать командой **hostname** или по переменной

окружения **HOSTNAME**. После редактирования файла необходимо выполнить перезагрузку системы.

2. Файл **/etc/hosts** представляет собой список доменных имён. В качестве аналогии можно привести телефонный справочник, только вместо номера телефона указывается IP-адрес, а вместо имени человека – доменное имя. При использовании команды **ping** указывается либо IP-адрес устройства, либо его доменное имя. При указании имени система обращается к файлу **/etc/hosts**, определяет по имени соответствующий IP-адрес, затем происходит обращение к целевому устройству по протоколу ICMP.

3. Файл **/etc/resolv.conf** хранит доменное имя и IP-адрес DNS-сервера. В случае, если при указании имени система не находит соответствующий IP-адрес в локальном файле **/etc/hosts**, она делает запрос в DNS-сервер. Если и там нет информации об указанном доменном имени, то выдаётся ошибка разрешения имён.

4. В файле **/etc/network/interfaces** хранятся параметры сетевых интерфейсов (например, Ethernet или WiFi). По умолчанию содержит следующие строки:

1) **source /etc/network/interfaces.d/***. Команда **source** вставляет в текущий файл **interfaces** содержимое папки **interfaces.d** (обычно изначально пустого). Удобен для соблюдения принципа модульности.

2) **auto lo**. Директива **auto** означает, что интерфейс, указанный справа, должен быть автоматически запущен во время загрузки системы.

lo (loopback, обратная петля) - виртуальный сетевой интерфейс, не связанный с каким-либо оборудованием, но при этом полностью интегрированный во внутреннюю сетевую инфраструктуру системы. Любой трафик, который посылается программой на интерфейс loopback, тут же получается тем же интерфейсом.

Он может быть использован сетевым клиентским программным обеспечением, чтобы общаться с серверным приложением, расположенным на том же компьютере. То есть если на компьютере, на котором запущен веб-сервер, указать в веб-браузере URL http://127.0.0.1/ или http://localhost/, то он попадает на веб-сайт этого компьютера.

Этот механизм работает без какого-либо активного подключения, поэтому он полезен для тестирования служб, не подвергая их безопасность риску, как при удаленном сетевом доступе. Подобным образом, пингование адреса loopback — это основной тест функционирования IP стека в операционной системе.

3) **iface lo inet loopback**. Директива **iface** описывает сетевой интерфейс **lo**, **inet** означает семейство интернет-протоколов (IPv4).

4) **auto eth0**. Директива **auto** предписывает автоматически включить сетевой интерфейс eth0 во время загрузки. В Astra Linux используется по умолчанию традиционная схема именования сетевых Ethernet интерфейсов: eth0, eth1 и т.д.

5) **iface eth0 inet dhcp**. Описание интерфейса **eth0** с присвоением IP-адреса по протоколу DHCP, то есть автоматически при обращении к DHCP-серверу.

Пользователь может самостоятельно назначать интерфейсу статический IP-адрес. Для этого в конфигурационном файле **/etc/network/interfaces** необходимый интерфейс описывается в следующем формате:

```
auto IFACE_NAME
iface IFACE_NAME inet static
address A.B.C.D
netmask A.B.C.D
gateway A.B.C.D
```

При этом **static** означает, что интерфейсу присваивается адрес статически, **address A.B.C.D** – это назначаемый IP-адрес интерфейса, **netmask A.B.C.D** – маска подсети, **gateway A.B.C.D** – шлюз по умолчанию.

После сохранения изменений необходимо перезагрузить интерфейс для установления введённых параметров. Для этого используются команды **ifdown IFACE_NAME** и **ifup IFACE_NAME**.

Установившиеся настройки можно посмотреть с помощью команды **ifconfig** (с применением **sudo**) или с помощью стандартной команды **ip a**.

Ниже приведен пример настройки сетевого интерфейса:

```
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
```

С помощью указанной конфигурации настраивается статический IP адрес со значением 192.168.1.2. Маска подсети – 255.255.255.0, адрес шлюза – 192.168.1.1.

Обратите внимание! Для работы с конфигурационным файлом /etc/network/interfaces используется служба networking.service. При запущенной утилите NetworkManager конфигурационный файл interfaces не перечитывается. Для корректной работы убедитесь, что служба NetworkManager остановлена и удалена из автозагрузки, а служба networking – запущена.

5. Установка ПО и настройка списка репозиториев

Существует несколько способов установки программного обеспечения на систему с ОС Linux. Основными из них являются сборка из исходных кодов и установка с использованием пакетных менеджеров.

Первый способ является трудозатратным и используется, когда установка из стандартных репозиториев невозможна. Администратору для установки будет необходимо самостоятельно скачать все зависимости, которые требует исходная программа, например компилятор или сторонние библиотеки нужных версий, корректно установить программу и в дальнейшем вручную обновлять в случае выхода новых версий. Основное достоинство подхода – для установки

не требуются права sudo. Этот способ подходит для сложных и нестандартных программ или для пользователей, у которых нет sudo-прав.

Второй способ является более оптимальным вариантом. Для установки по используется отдельная программа – пакетный менеджер. Он автоматически загружает и устанавливает саму программу и все её зависимости. Загрузка происходит из репозитория, который может быть расположен на удаленном сервере, так и локально. Для удобства работы с пакетным менеджером существуют утилиты, позволяющие автоматически загружать программы из репозиториев и устанавливать его со всеми необходимыми зависимостями. В Debian и Astra Linux в качестве пакетного менеджера выступает программа dpkg и дополнительные утилиты, такие как apt-get, apt-cache и другие.

Список всех репозиториев, из которых возможно производить установку с помощью пакетного менеджера расположен в файле /etc/apt/sources.list

На рисунке указаны две записи со ссылками на репозитории – Debian и Astra Linux. Формат записи следующий:

deb <путь_к_корневому_каталогу_репозитория> <код_дистрибутива> <список_компонент>

В указанный файл возможно добавить другие источники загрузки, например CD диск, локальный репозиторий и т.п.

После каждого внесения изменений в файлы с описанием репозиториев и перед установкой ПО следует обновить списки пакетов. Для этого необходимо выполнить команду:

```
sudo apt update
```

Для установки пакета воспользуйтесь командой:

```
sudo apt-get install <имя пакета>
```

Для удаления пакета с очисткой всех настроек воспользуйтесь командой:

```
sudo apt-get purge <имя пакета>
```

Ниже приведем несколько полезных команд для работы с пакетным менеджером и установленными программами.

Обновить пакеты программного обеспечения с соблюдением всех зависимостей

```
sudo apt-get dist-upgrade
```

Обновить конкретный пакет

```
sudo apt-get install <имя пакета> --only-upgrade
```

Очистка кэша apt

```
sudo apt-get clean
```

Просмотр всех доступных пакетов

```
apt-cache <имя пакета>
```

Определение имени пакета по описанию. Ключевые слова указываются через пробел. Словосочетания заключаются в одинарные кавычки.

```
apt-cache search <ключевое слово>
```

Просмотр всех установленных пакетов

```
dpkg -l
```

Определить, какому пакету принадлежит установленная программа или файл

```
dpkg -S <полный путь до файла>
```

Определить какие файлы входят в пакет

```
dpkg -L <имя пакета>
```

1.8. Настройка сети с использованием утилиты Network Manager.

Конфигурационные файлы, отвечающие за настройку сетевого оборудования, могут отличаться в разных дистрибутивах Linux. Утилита Network Manager представляет собой универсальный способ настройки сети. Основной командой для управления является nmcli. Она позволяет создавать, изменять, удалять, активировать и деактивировать сетевые подключения.

Синтаксис команды:

```
nmcli опции объект команда
```

Где:

- опции — дополнительные параметры для команды, такие как -t для вывода без форматирования или -p для вывода с использованием форматирования.
- объект — сущность, с которой вы хотите взаимодействовать

Чаще всего в nmcli используются следующие объекты:

  - **device** - управление сетевыми интерфейсами (lo, enp1s0 и др.);
  - **connection** - управление соединениями;
  - **networking** - управление сетью в целом;
  - **general** - показывает состояние всех сетевых протоколов и NetworkManager в целом;
  - **radio** - управление сетевыми протоколами, wifi, ethernent и т д.
- команда — действие, которое вы хотите выполнить над объектом (например, show, up, down, add, edit, delete).

Перед началом работы с Network Manager убедитесь, что служба NetworkManager.service запущена, а служба networking.service – отключена.

За настройку службы отвечает конфигурационный файл /etc/NetworkManager/NetworkManager.conf. Проверьте, что параметр managed в разделе [ifupdown] равен true. Если нет, то исправьте это.

Примеры работы:

Проверка статуса работы службы

```
nmcli networking
```

Определение сетевых интерфейсов

```
nmcli device
```

Определение сетевых подключений

```
nmcli connection show
```

Добавить сетевое подключение my-eth через интерфейс enp1s0

```
nmcli connection add type ethernet con-name my-eth ifname enp1s0
```

Удалить сетевое подключение

```
nmcli connection delete my-eth
```

Назначить соединению статический метод конфигурации IP

```
nmcli connection mod my-eth ipv4.method manual
```

Назначить соединению динамический метод конфигурации IP

```
nmcli connection mod my-eth ipv4.method auto
```

Назначить соединению статический ip адрес

```
nmcli connection mod my-eth ipv4.addresses 192.168.122.3/24
```

Назначить соединению шлюз по-умолчанию

```
nmcli connection mod my-eth ipv4.gateway 192.168.122.1
```

Назначить соединению IP-адреса DNS-сервера

```
nmcli connection mod my-eth ipv4.dns '8.8.8.8'
```

Включить сетевой интерфейс

```
nmcli connection up my-eth
```

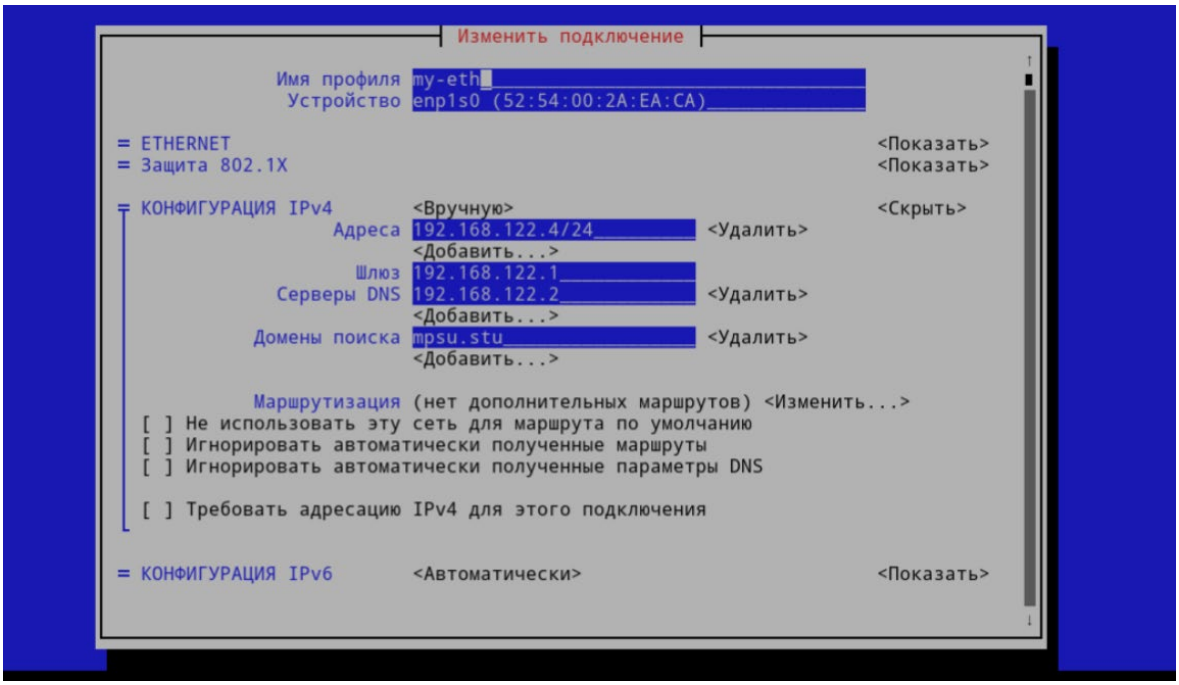
Отключить сетевой интерфейс

```
nmcli connection down my-eth
```

Созданные настройки сохраняются в виде конфигурационного файла в директории /etc/NetworkManager/system-connections/. При необходимости возможно отредактировать его вручную. После редактирования необходимо перегрузить службу NetworkManager.

Для большего удобства настройки существует утилита nmtui с псевдографическим интерфейсом. С его помощью возможно настраивать основные параметры сети не прибегая к командам.

nmtui



1.9. Подключение к системе по SSH

SSH (англ. Secure Shell — «безопасная оболочка») — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой

SSH работает в режиме точка-многоточка. В этом режиме есть некоторый узел, к которому могут подключаться другие узлы — сервер. Все клиенты подключаются к этому серверу.

В Linux системах обычно используется OpenSSH, где сервер обозначается sshd, что означает SSH демон (Daemon), а клиент SSH — ssh.

Запуск / статус / остановка / перезапустить / перечитать конфигурацию / включить / исключить автозагрузку:

systemctl start/status/stop/restart/reload/enable/disable sshd.service

Для аутентификации по ключам спользуется алгоритм асимметричного шифрования RSA. Пара ключей представлена публичным и приватным ключами. Ключи на клиенте хранятся в каталоге ~/.ssh в файлах :

- id_rsa – приватный ключ пользователя
- id_rsa.pub – публичный ключ пользователя
- known_hosts – публичные ключи ssh-серверов

На сервере командой **ssh-keygen** производится генерация ключей.

Командой **ssh-copy-id adminstd@A.B.C.D** передаётся публичный ключ на клиент по его IP-адресу A.B.C.D.

После этого клиент командой **ssh adminstd@A.B.C.D** может удалённо подключиться по протоколу SSH к серверу с IP-адресом A.B.C.D.

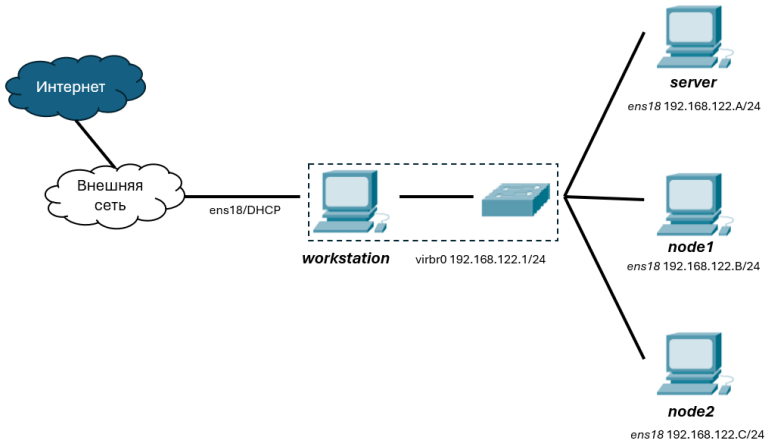
Команда scp (*Secure CoPy*) - это утилита, которая работает по протоколу SSH.

Она позволяет копировать файлы с клиента на сервер напрямую без использования протоколов FTP или SMB. Имеет следующий синтаксис:

```
$ scp опции пользователь1@хост1:файл пользователь2@хост2:файл
```

2. Практическая часть

На рисунке представлена схема локальной сети. Основной машиной является workstation, внутри которой должны быть развернуты три виртуальных машины – server, node1, node2.



На машине workstation настроены два сетевых интерфейса – ens18 для выхода в интернет, адрес которого выдается динамически и интерфейс virbr0, исполняющий роль сетевого моста между подсетью, выводящей в интернет и трех внутренних виртуальных машин. Адрес сетевого моста – 192.168.122.1/24, адреса виртуальных машин должны быть выбраны из той же подсети в соответствии с вашим вариантом (см. далее).

1. Основная настройка

Подключитесь к виртуальной машине. Откройте вкладку терминала. Перейдите в раздел «Настройка -> Настроить профиль». В открывшейся панели вы можете настроить терминал под свой вкус.

Рекомендуем указать следующие параметры:

- Вкладки -> Удаленный сеанс. Изменить значение на %h. Это позволит вывести вместо номера вкладки имя подключаемой по ssh машины.
- Мышь -> Установить галочку «Копировать при выделении».

2. Работа с виртуальными машинами

Работу с виртуальными машинами возможно осуществлять с использованием командной строки с помощью утилиты virsh или с использованием графического приложения «Менеджер виртуальных машин».

Для работы необходимо наличие трех виртуальных машин – server, node1, node2. Создайте их путем клонирования существующей машины alse18. Для

этого войдите в приложение «Менеджер виртуальных машин» и щелкните правой кнопкой мыши по строке alse18. В выпадающем меню выберете пункт «Клонировать». Название машины должно совпадать с названием со схемы (server, node1, node2).

Перед выполнением всех заданий создайте снимок каждой из машин. Дайте им осмысленные названия. После окончания каждой из работ создавайте снимки данных.

Запустите машины. После этапа загрузки операционной системы появится окно для ввода пароля пользователя **adminstd**. Введите пароль по умолчанию **qwerty22**. Используя эмулятор терминала Fly, выполните следующие задания.

2.1. Задание 1

2.1.1. Создайте на каждой из машин пользователя. Имя пользователя должно иметь формат: фамилияИО. Например, ivanovii. Все дальнейшие действия должны производиться из-под этого пользователя.

2.1.2. Добавьте пользователя в группы sudo и во все группы, в которые входит пользователь adminstd.

2.1.3. Выдайте созданному пользователю максимальную метку целостности (63).

2.1.4. Войдите в систему из-под созданного пользователя и на сервере установите утилиту, позволяющую работать по протоколу ftp. (Подсказка, в качестве ключевых слов поиска используйте lightweight, efficient, ‘ftp server’)

2.2. Задание 2

2.2.1. Выясните текущие сетевые настройки каждой машины. Проверьте соединение между машинами, доступность выхода в интернет.

2.2.2. Используя параметры, приведённые в таблице ниже, настройте сеть, дополнив необходимые конфигурационные файлы. Server и Node1 настройте, используя конфигурационные файлы, Node2 – используя Network Manager.

Machine name	Server	Node1	Node2
Host name	server	node1	node2
IP address	192.168.122.(№ в группе + 1)	192.168.122.(№ в группе + 2)	192.168.122.(№ в группе + 3)
Mask	255.255.255.0		
Default gateway	192.168.122.1		

2.2.3. Проверьте соединение между workstation и каждой из машин используя утилиту ping. Проверьте соединение между машинами.

2.2.4. Попробуйте отправить **ping** между машинами используя доменное имя. Получилось ли это сделать? Если нет, то исправьте это.

2.3. Задание 3.

2.3.1. На машине workstation запустите службу ssh и добавьте её в автозагрузку. Настройте аутентификацию по ключам с сервером и каждой из node.

2.3.2. Подключитесь из разных вкладок workstation к каждой из машин, используя созданного пользователя и доменное имя машины.

2.3.3. Создайте на машине workstation файл, содержащее сообщение «Привет, сервер!» и передайте его на server используя утилиту scp.

2.3.4. Создайте на сервере в директории tmp файл и выгрузите его на машину workstation. Команду выполнить с workstation.

2.4. Задание 4.

2.4.1. Выключите машины и создайте снимок каждой из них. Дайте им осмысленное название.

2.4.2. Подключитесь к машине node1 и выполните команду sudo rm -rf /. К какому результату это привело?

2.4.3. Подключитесь к машине node2 и выполните команду :(){ :|:& };: К какому результату это привело? Объясните результат выполнения команды (см. https://www.cyberciti.biz/faq/understanding-bash-fork-bomb/).

2.4.4. Восстановите машины, используя созданные ранее снимки.

Контрольные вопросы

1. Что такое ЛВС? Из каких устройств может состоять?
2. Из каких уровней состоит модель OSI? Для чего её создали?
3. Какие операционные системы могут использоваться на серверах?
4. Что такое виртуальная машина? Зачем она нужна?
5. Какие существуют способы передачи файлов по сети?

Список рекомендуемой литературы: