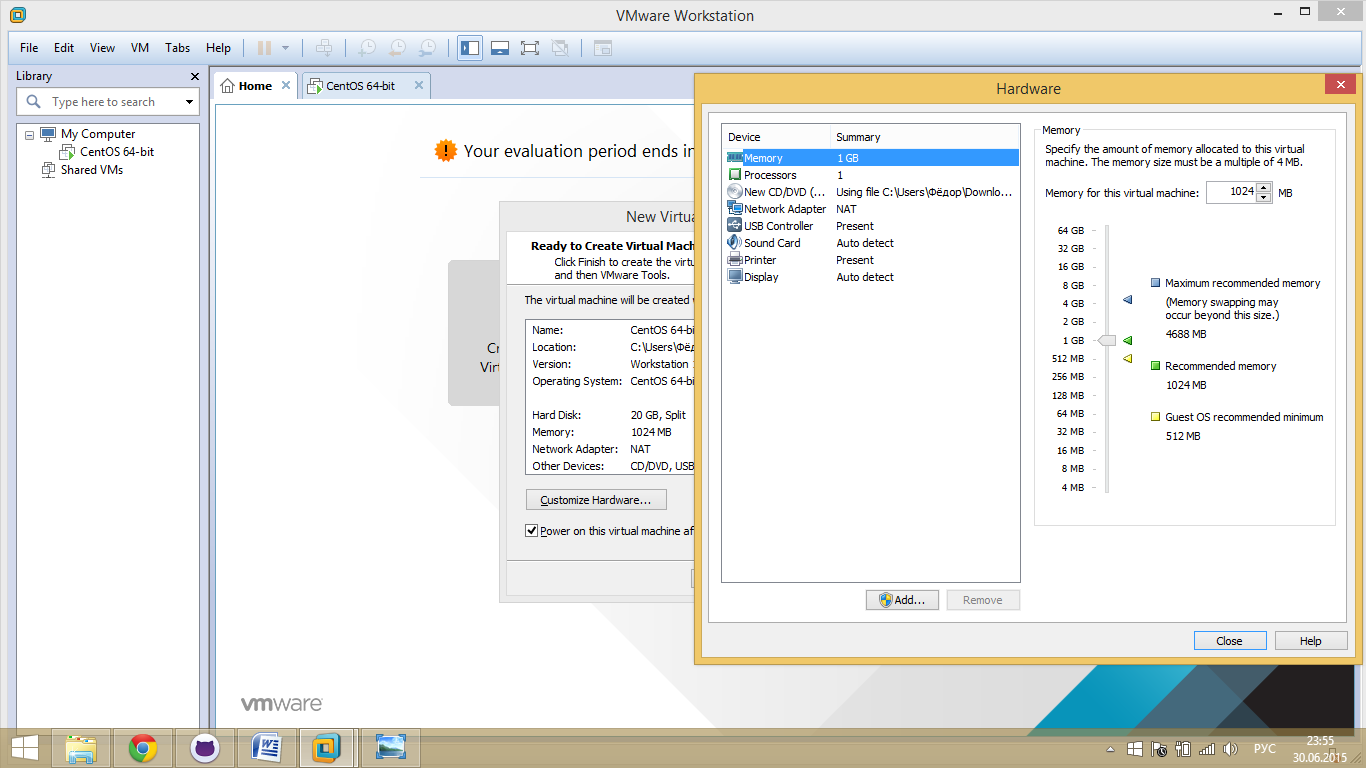
Практика. 1 курс 2 семестр.

1. Установить дистрибутив CentOS на VMware Workstation.

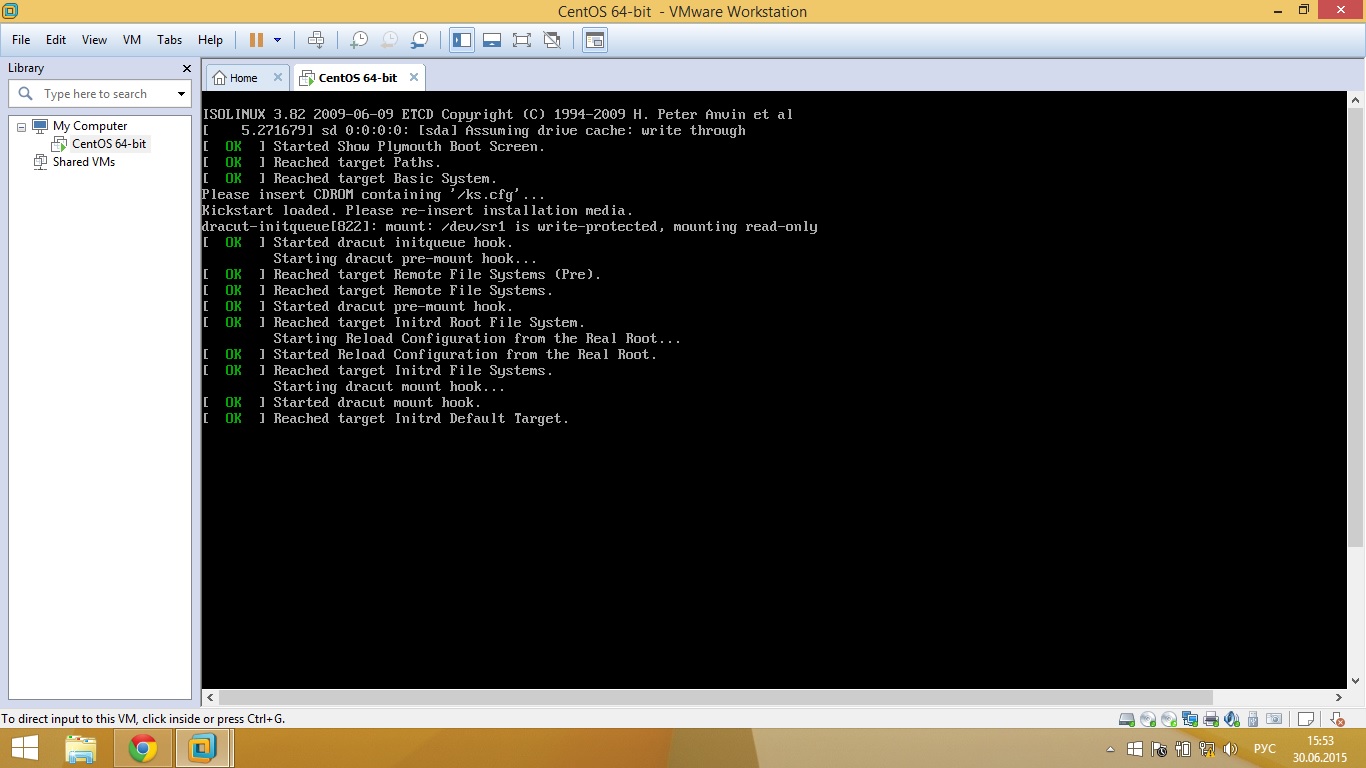
CentOS (Community ENTerprise Operating System) — дистрибутив Linux, основанный на коммерческом Red Hat Enterprise Linux  компании Red Hat и совместимый с ним. Срок поддержки каждой версии CentOS составляет 10 лет. Каждая версия обновляется каждые 6 месяцев для поддержки новых аппаратных средств, данная операционная система использует программу yum (Yellow dog Updater, Modified )для скачивания и установки обновлений из репозитория CentOS Mirror Network.

VMware Workstation — программное обеспечение виртуализации, предназначенное для компьютеров х86-64 операционных систем Windows и Linux. Позволяет пользователю установить одну или более виртуальных машин на один компьютер и запускать их параллельно с ним. Каждая виртуальная машина может выполнять свою операционную систему. Данное ПО поддерживает возможность объединения нескольких виртуальных машин в группу, которую можно включать, выключать, приостанавливать или возобновлять как единый объект, что является полезным для тестирования технологий клиент-сервер.

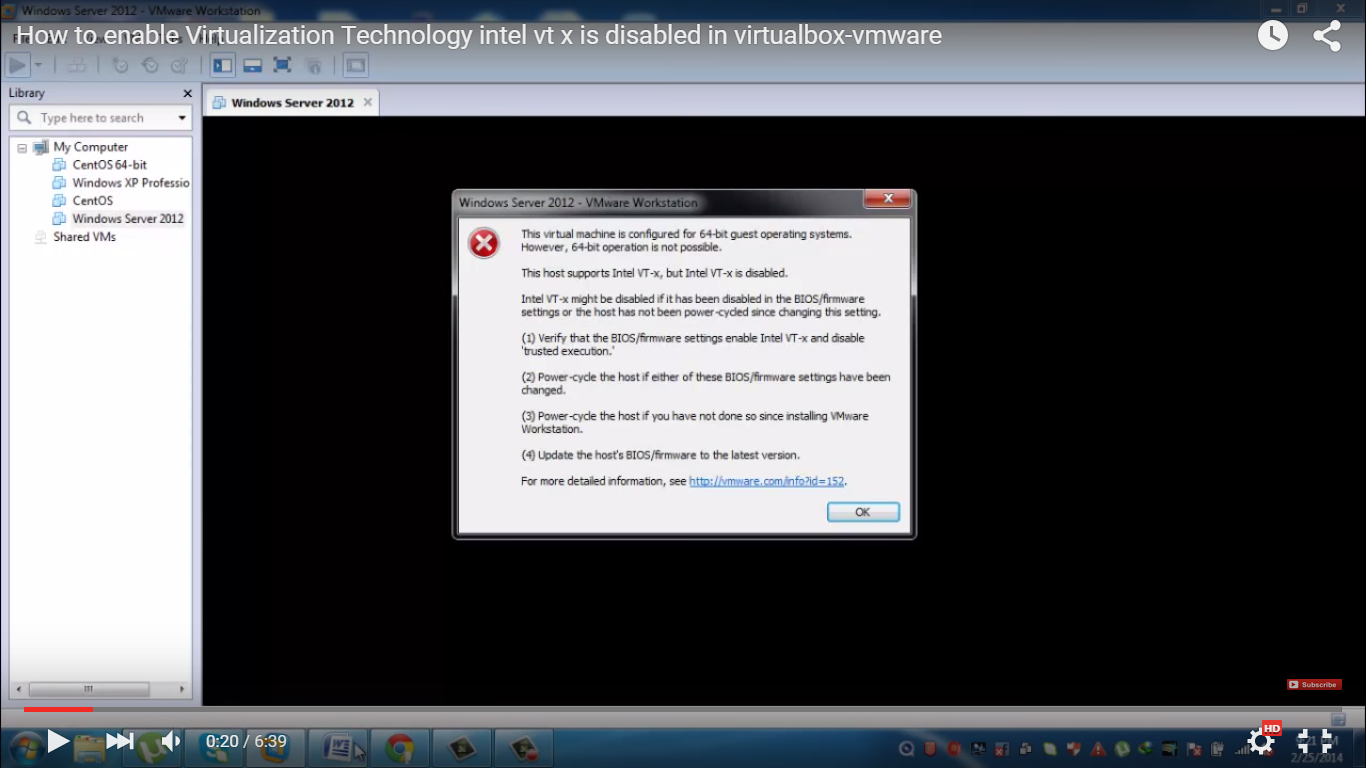
* Скачиваем дистрибутив СentOS( минимальная версия x86 архитектура) с официального сайта [сentos.org](http://centos.org/)
* В VMware Workstation создаём новую виртуальную машину: File → New Virtual Machine(Ctrl+N); следуя инструкциям устанавливаем интересующий нас дистрибутив , в процессе установки потребуется заполнить поля User name и Password, имя пользователя и пароль понадобятся нам в дальнейшем при использовнии ОС Linux. Далее выбираем название виртуальной машины и определяем место и размер выделяемый этой ВМ на жёстком диске . Воспользовавшись пунктом Customize Hardware можно настроить дополнительные параметры для нашей виртуалки(например, объём оперативной памяти выделяемой под неё, взаимодействие с USB-контролерами, сетевыми адаптерами и т.д.)

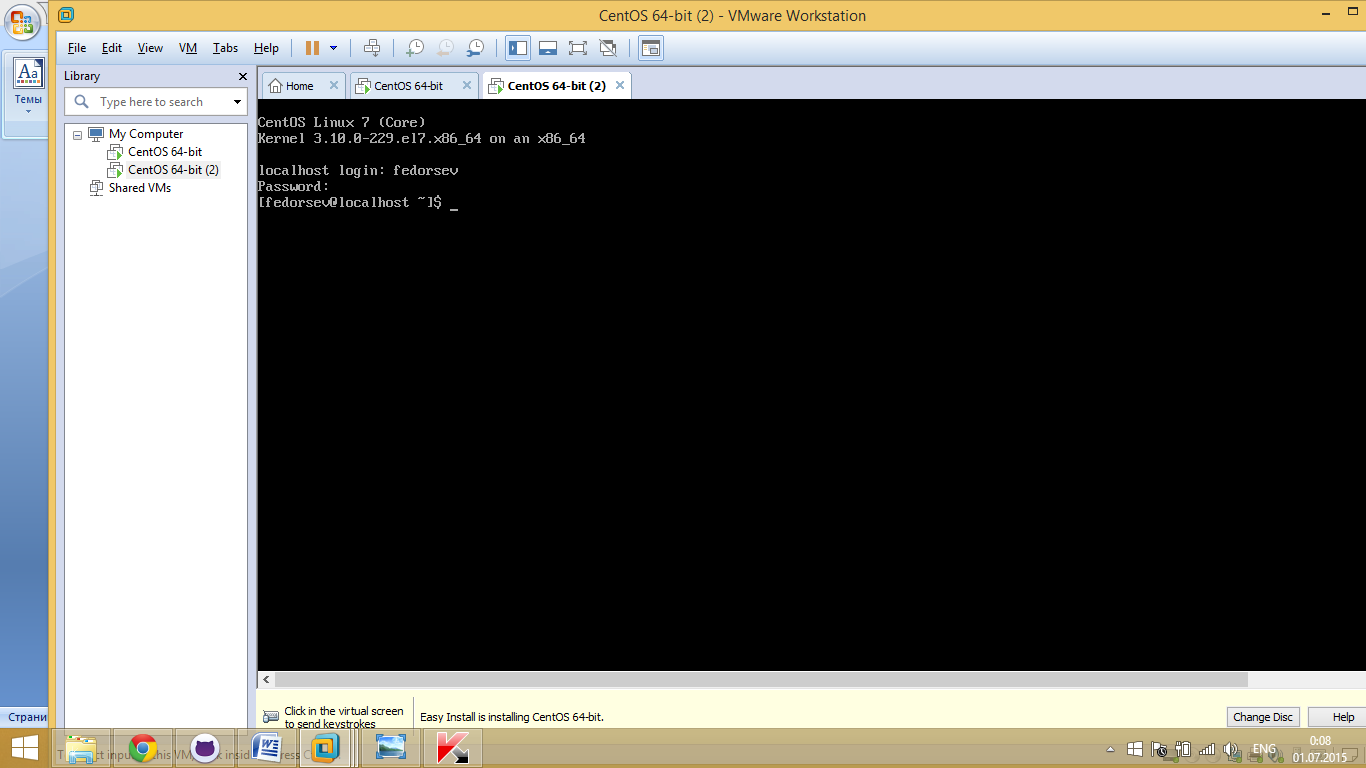


* Виртуальная машина готова к установке дистрибутива, дальше следует просто немного подождать, пока система установится на виртуалку.



* При запуске установки ОС на ВМ может возникнуть проблема, связанная с тем, что в большинстве компьютеров стандартно отключена поддержка виртуализации. Требуется включить этот параметр в BIOSе системы.



* Если все сказанное выше выполнено, система готова к использованию, всё, что от нас требуется ввести логин и пароль

2. Освоить основные консольные команды и определить различия между рутом(root) и обычным пользователем, заходим в систему с правами рут пользователя.

При работе с консолью Linux используется большое количество команд, вот список некоторых из них( в основном для работы с директориями и файлами):

* Для начала приведем общий вид консольной команды в системе Linux:

команда(название) опции аргументы

*ls --all -l* *"путь к каталогу"*

*ls -a -l "путь к каталогу"*

*ls -al "путь к каталогу" -* аналогичные записи той же команды

*ls -la "путь к каталогу"*

Для различных команд существует большое количество опций, чтобы получить справку по какой-то конкретно интересующей нас команде, достаточно написать в консоли *man "команда".*

Для длинных опций используется -- , для коротких - . Для большинства длинных опций существуют сокращения, так же возможна запись различных опций под одним знаком - , при этом не важно в каком порядке они указываются.

* Теперь продемонстрируем несколько базовых команд для работы с директориями, файлами и самой системой:

*pwd (print working directory) -* выводит на экран текущую директорию

*ls(list directory content) -* выводит на экран содержание директории(полезные параметры -a выводит все папки, включая скрытые и -l вывод на экран в виде списка)

*mkdir "путь до директории"-* создает директорию(с параметром -p создаются все промежуточные(родительские) директории)

*touch "путь до файла" -* создаёт файл

*cd "путь до директории" -* войти в директорию

*rm "путь до файла" -* удаляет файл *( rm -r* удаляет директорию, если в директории содержатся промежуточные директории или какие-то файлы система выдаст несколько дополнительных вопросов, для подтверждения удаления этих директорий, файлов, чтобы избежать это используется опция *rm -f)*

*cp "путь до файла1" "путь до файла2" -* копирует один файл в другой, для копирования директории так же используется опция *cp -r "путь к директории1" "путь к директории2"*

*mw "путь1" "путь2" - перемещает файл или директорию*

*cat -* на экран выводится содержимое указанного файла (или нескольких файлов, если их имена последовательно задать в качестве аргументов команды), если вывод команды cat перенаправить в файл, то можно получить копию какого-то файла: cat file1 > file2

собственно, первоначальное предназначение команды cat как раз и предполагало перенаправление вывода, так как эта команда создана для конкатенации, т. е. объединения нескольких файлов в один: cat file1 file2 ... fileN > new-file

Именно возможности перенаправления ввода и вывода этой команды и используются для создания новых файлов. Для этого на вход команды cat направляют данные с клавиатуры, а вывод команды — в новый файл: cat > newfile

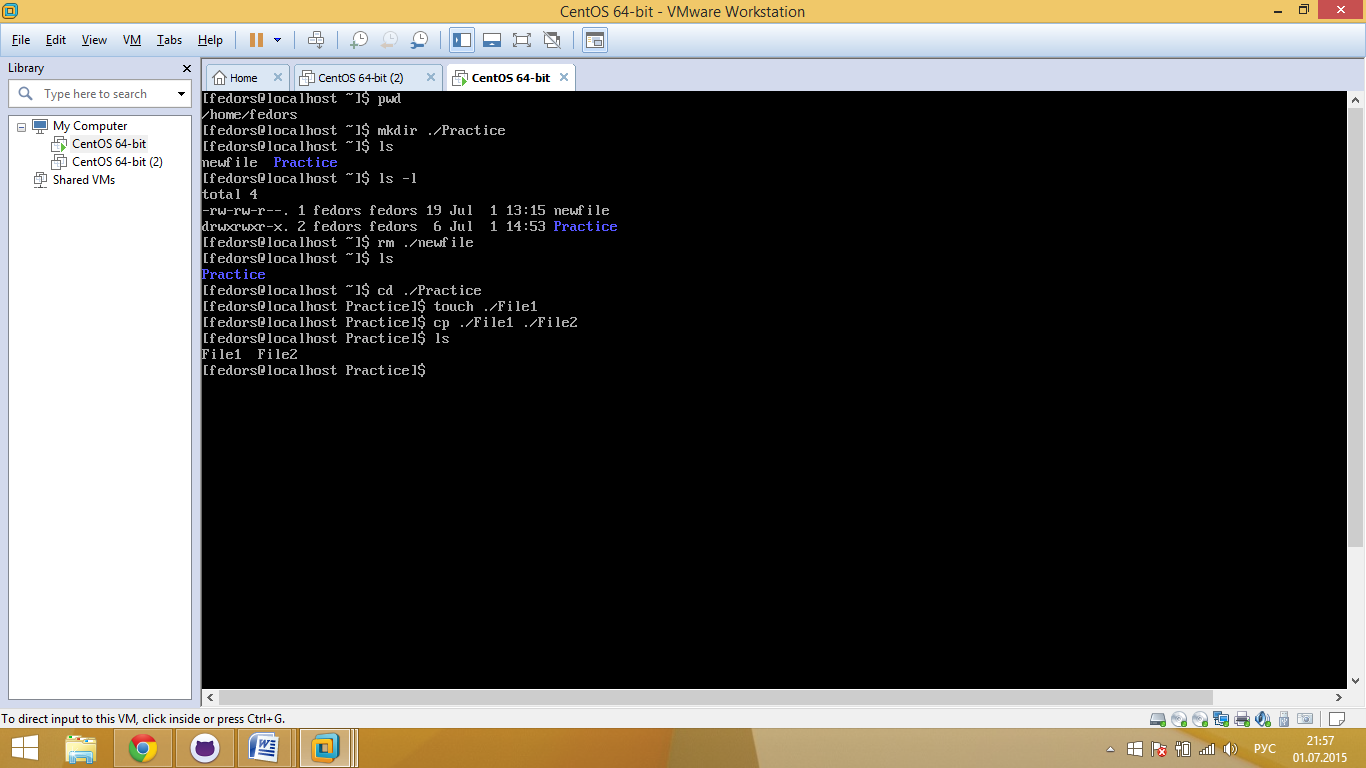
*vi -* вызывает стандартный текстовый редактор системы Linux vi(visual editor), не очень удобный для использования, но гарантированно присутствует в любой системе UNIX

*clear -* очищает консоль

*exit -* выходим из операционки

*poweroff-* выключаем ВМ

*reboot -* перезапускаем ВМ



* Устройство пути в Linux несколько отличается от аналогичного в Windows , в ОС Windows, если мы находимся в какой-то папке и поднимаемся всё выше и выше, то в конечном итоге мы придём к какому-то из дисков(С:\ ; D:\ и т.д.), в Linux же это корневая директория(обозначается "/"), поэтому существует два способа задать путь:

1)Абсолютный - путь от главной(корневой) директории: "/" /home/fedors/Practice/...

2)Относительный - путь от текущей директории

Practice/...

Кроме того в путях можно использовать специальные символы:

. - текущая директория

.. - директория на уровень выше

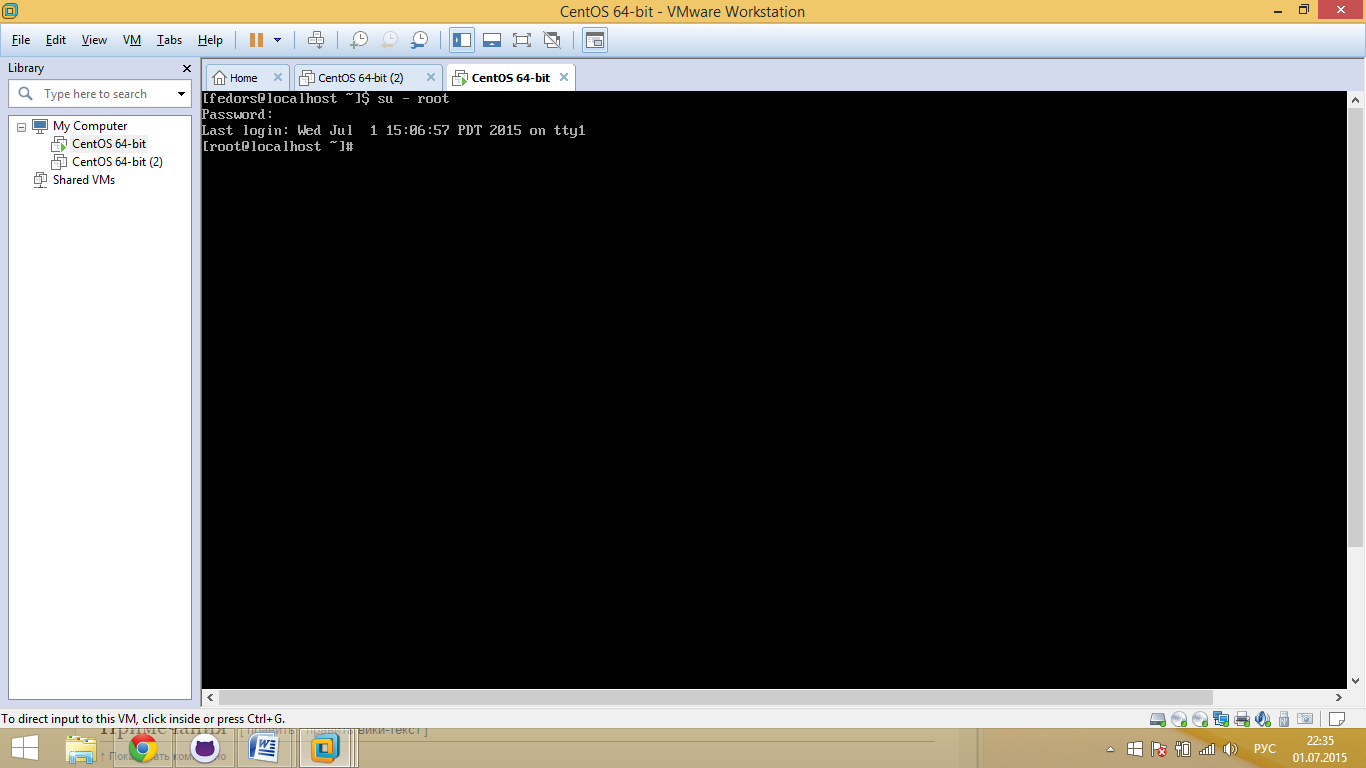
~ - домашняя директория

\* - любое количество любых символов

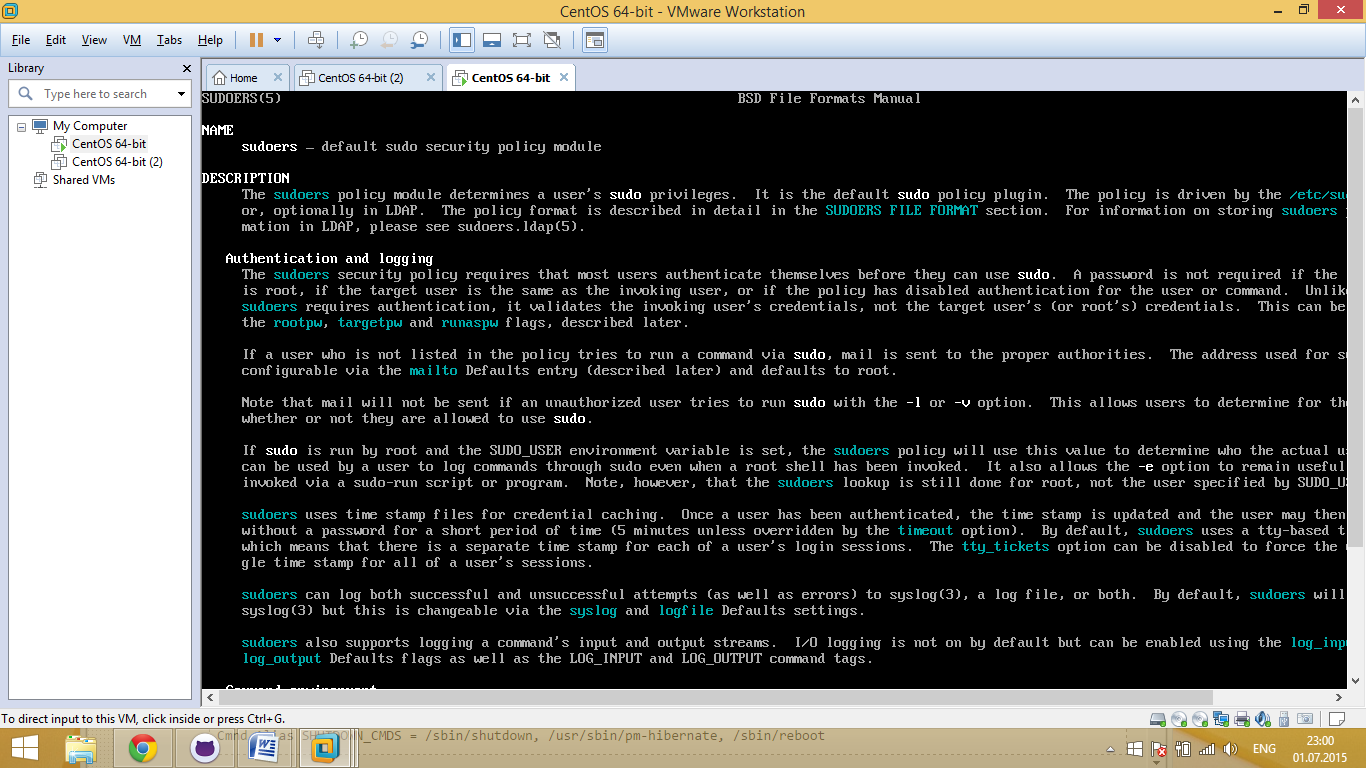
? - ровно один любой символ

* Различия между рутом и обычным пользователем заключаются в том, что в системе Linux любые действия пользователя возможны только после его идентификации с какой-либо из регистрационных записей. Linux предоставляет пользователю возможности, определяемые его правами на работу с файлами и директориями. Считается, что каждый файл принадлежит определенному пользователю или определенной группе пользователей. С каждым файлом также связана таблица прав, указывающая, какие действия (чтение, запись, исполнение программы, открытие директории и т.д.) может предпринимать владелец файла, член группы, владеющей файлом, и произвольный пользователь. Кроме регистрационных записей обычных пользователей существует также регистрационная запись суперпользователя с зарезервированным именем root. Суперпользователь может выполнять любые допустимые действия с любым файлом, независимо от того, кто является его владельцем и какова таблица прав этого файла. Кроме того, суперпользователь может изменять информацию о владельце любого файла и его таблицу прав. А также суперпользователь обладает значительно более широкими правами, не связанными с файловыми операциями.

Обычный пользователь не в состоянии повредить или удалить никакие файлы, имеющие критически важное значение для работоспособности системы в целом, а суперпользователь - может. Стоит отметить, что операции удаления и перемещения файлов выполняются в Linux значительно менее формально, без многочисленных предупреждений, привычных для пользователей Windows.

Для того, чтобы стать рутом, нам требуется воспользоваться командой смены пользователя( *su - имя пользователя(*в данном случае *su - root))* и ввести пароль

В некоторых дистрибутивах Linux в целях безопасности отсутствует возможность перехода к root -пользователю, поэтому стоит так же рассмотреть вариант использование sudo (это утилита, предоставляющая привилегии root для выполнения административных операций в соответствии со своими настройками, она позволяет легко контролировать доступ к важным приложениям в системе), обычно sudo используют для запуска приложений требующих наличия прав администратора(*sudo "имя команды"),* чтобы получить справку по всему функционалу этой утилитыпользуемся известной командой *man sudoers*.

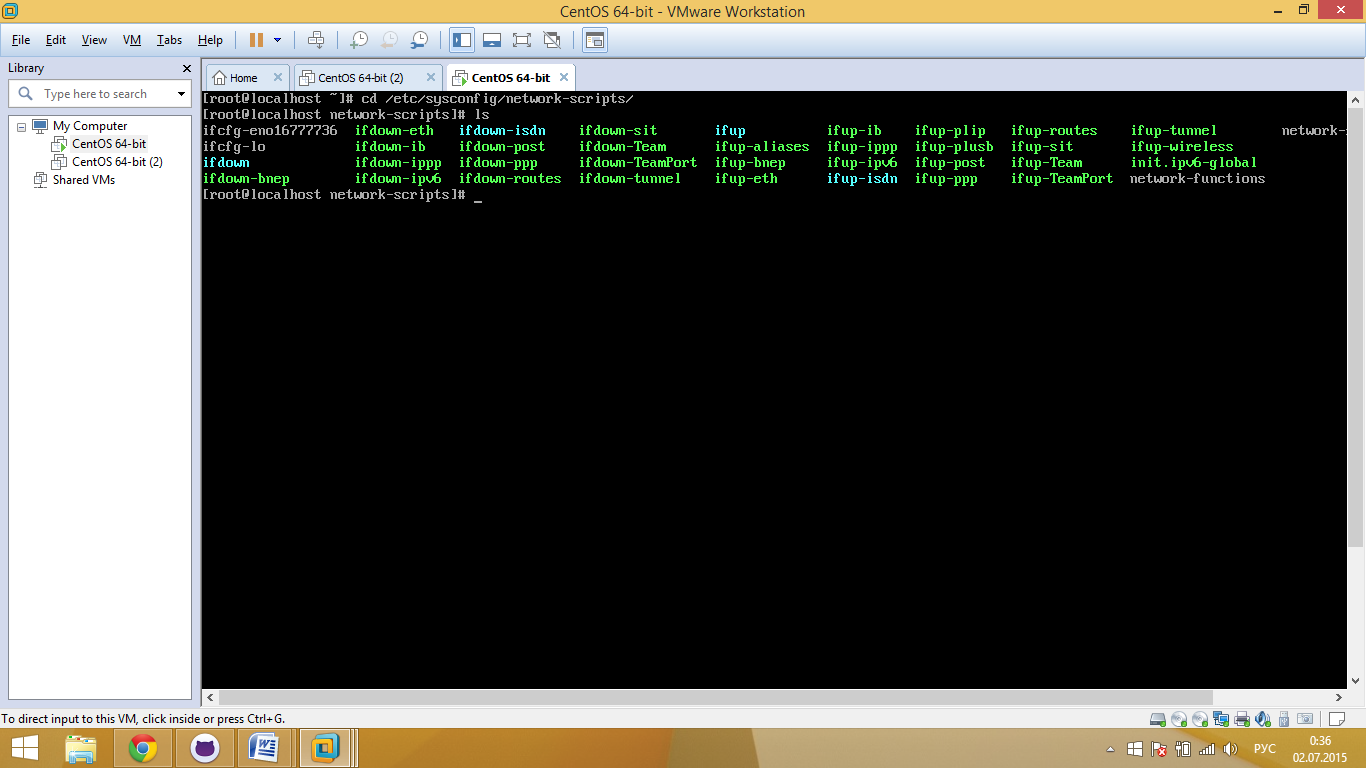


3.Настроить сеть и установить несколько программ( текстовый редактор, файловый менеджер и браузер).

Стоит отметить, что в CentOS для скачивания и установки обновлений из репозиториев используется открытый консольный менеджер PRM-пакетов yum(Yellow dog Updater, Modified), он представляет из себя консольный менеджер, использующий интерфейс командной строки. Однако существуют и надстройки, предоставляющие

GUI.

* Для начала зайдём под правами рута, так как нам потребуется вносить изменения в системные файлы
* Для настройки сетевого подключения требуется внести изменения в файл конфига, так как все аппаратные и программные устройства в Linux, а так же различные параметры самой операционной системы хранятся именно в конфигурационных файлах.
* Конфиг сетевого адаптера находится в */etc/sysconfig/network-scripts/,* такие файлы обычно называются *ifcfg "имя" ,* находим этот файл



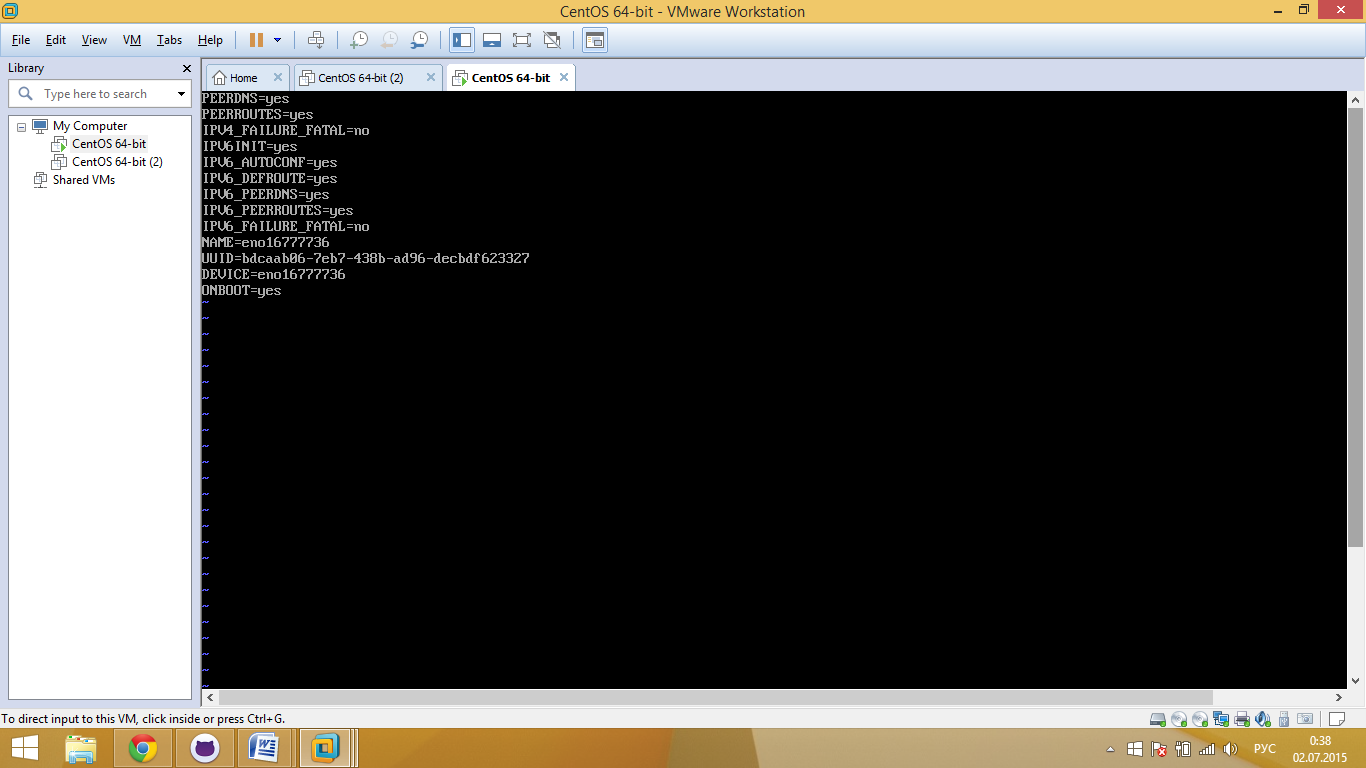
* редактируем его через встроенный текстовый редактор vi, для сети использующей DHCP соответственно нужно проверить или внести изменения в следующий пункты:

DEVICE = "имя"

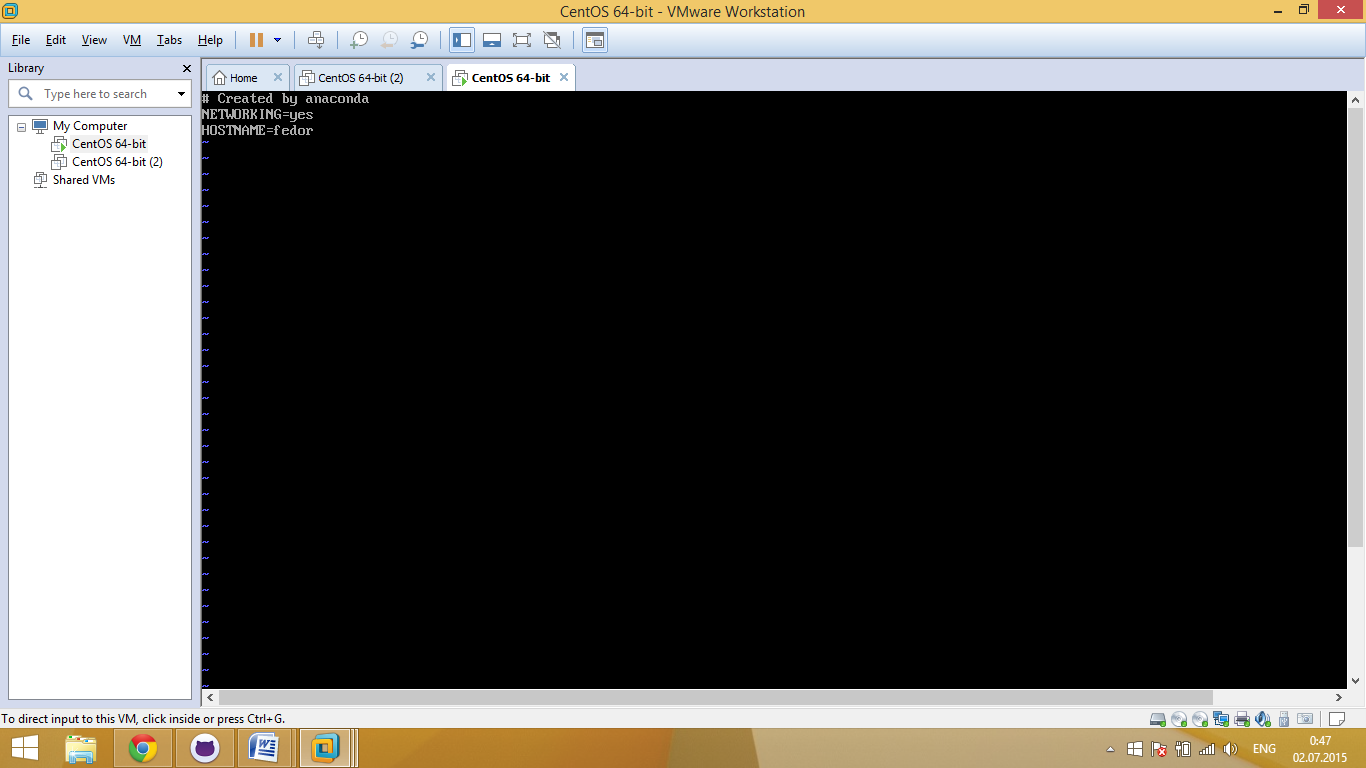
BOOTPROTO = "dhcp"

ONBOOT = "yes" - этот параметр отвечает за активирование устройства во время загрузки

Для редактирования файла нажимаем 'i'; для сохранения изменений: сначала 'esc' для возвращения в командный режим, а затем 'wq' для сохранения внесенных поправок



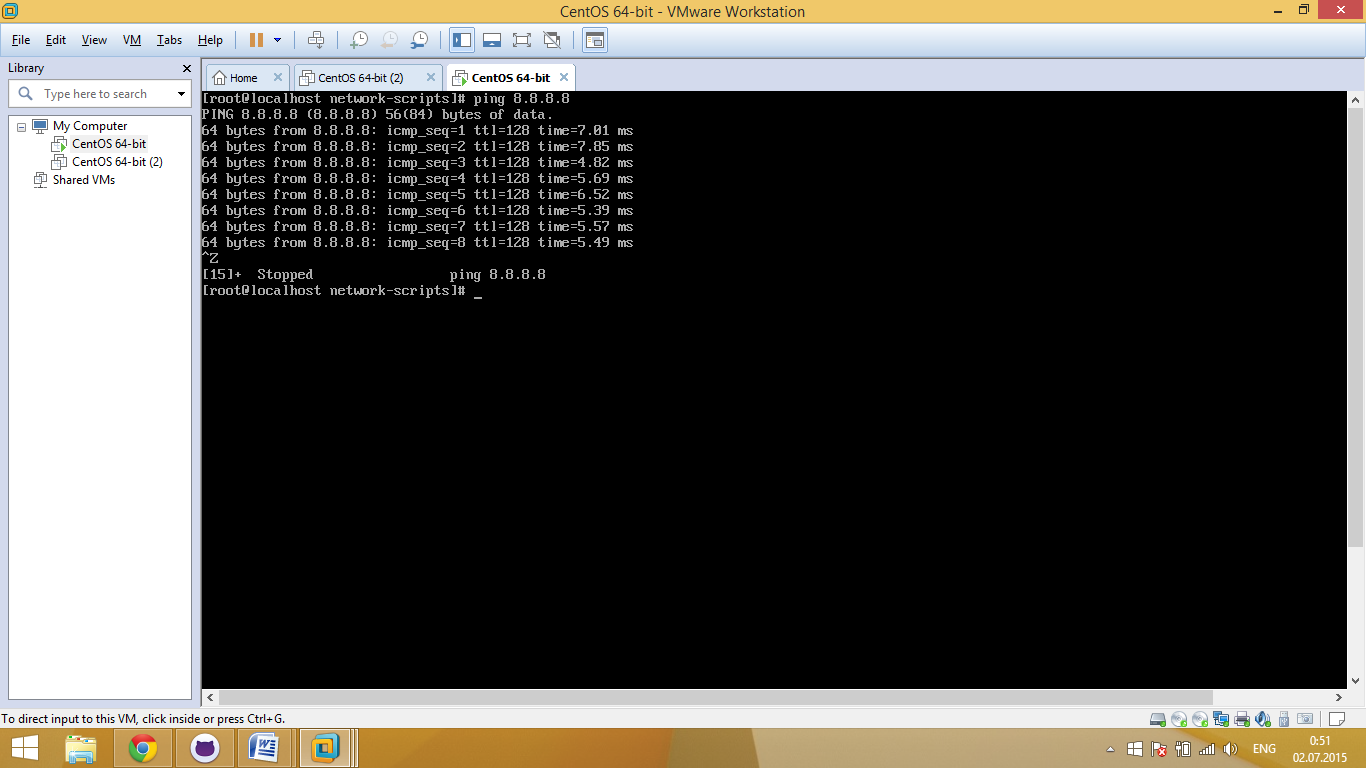
* редактируем сетевой конфигурационный файл vi /etc/sysconfig/network , записываем туда следующий строчки(изначально файл пустой):

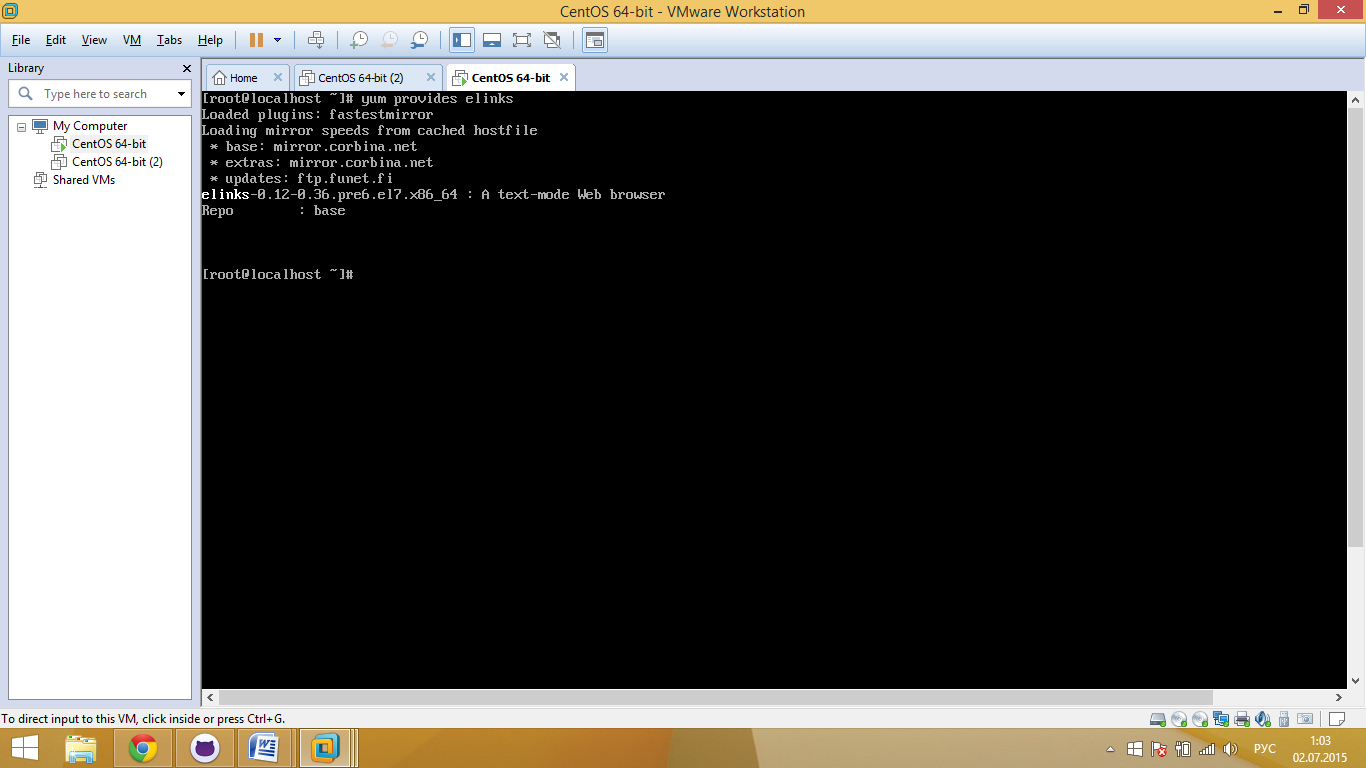
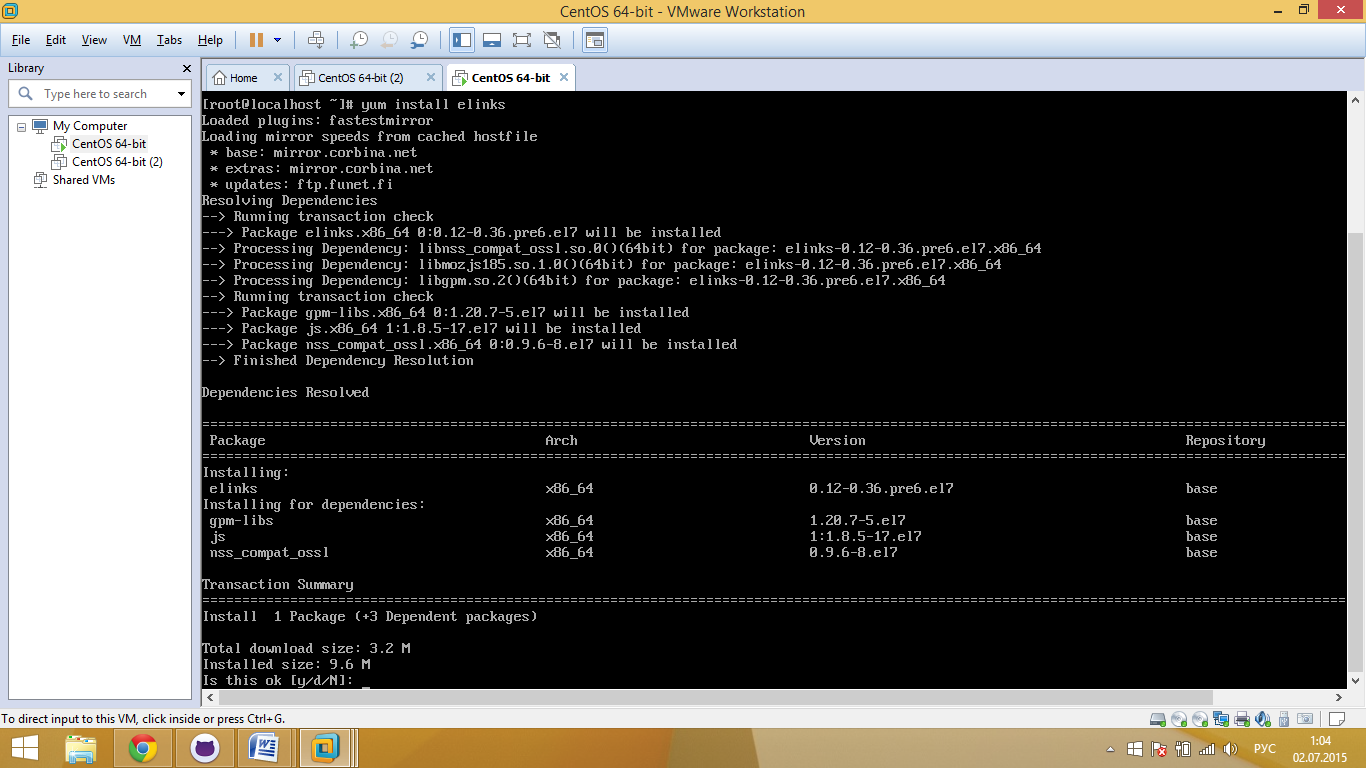


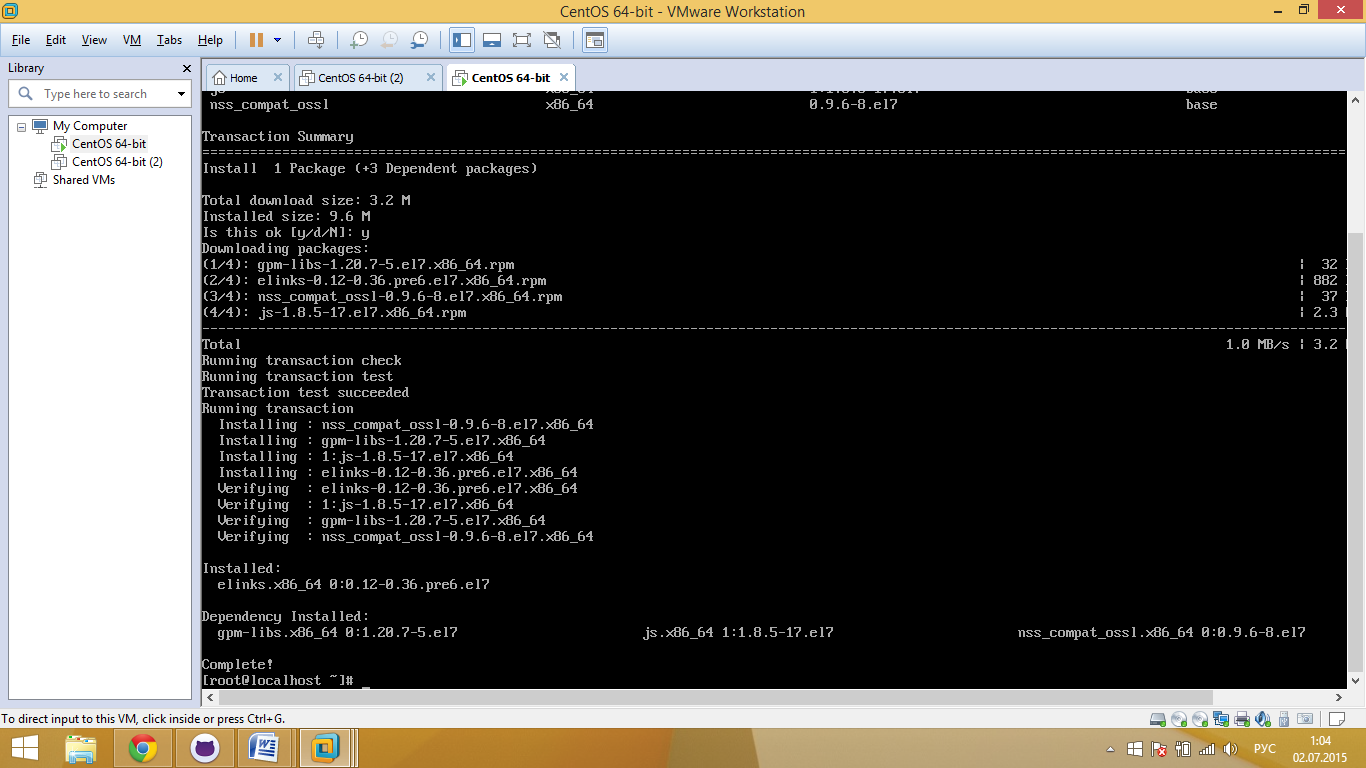
* для корректного применения настроек выполняем перезапуск сети

/etc/init.d/network restart

* проверяем наличие подключения командой ping 8.8.8.8



* Для установки приложений пользуемся yum( так же требуются права рута), делается это с помощью пары команд, первой - yum provides "имя файла" находим сам файл и зеркало с которого будем его скачивать, соответственно видим название файла, его версию, размер
* второй командой yum install "имя файла" устанавливаем , появляется запрос на скачивание, подтверждаем его

Программа готова к использованию

* Аналогичным образом в добавок к elinks(браузер) устанавливаем на нашу систему файловый менеджер mc и текстовый редактор nano

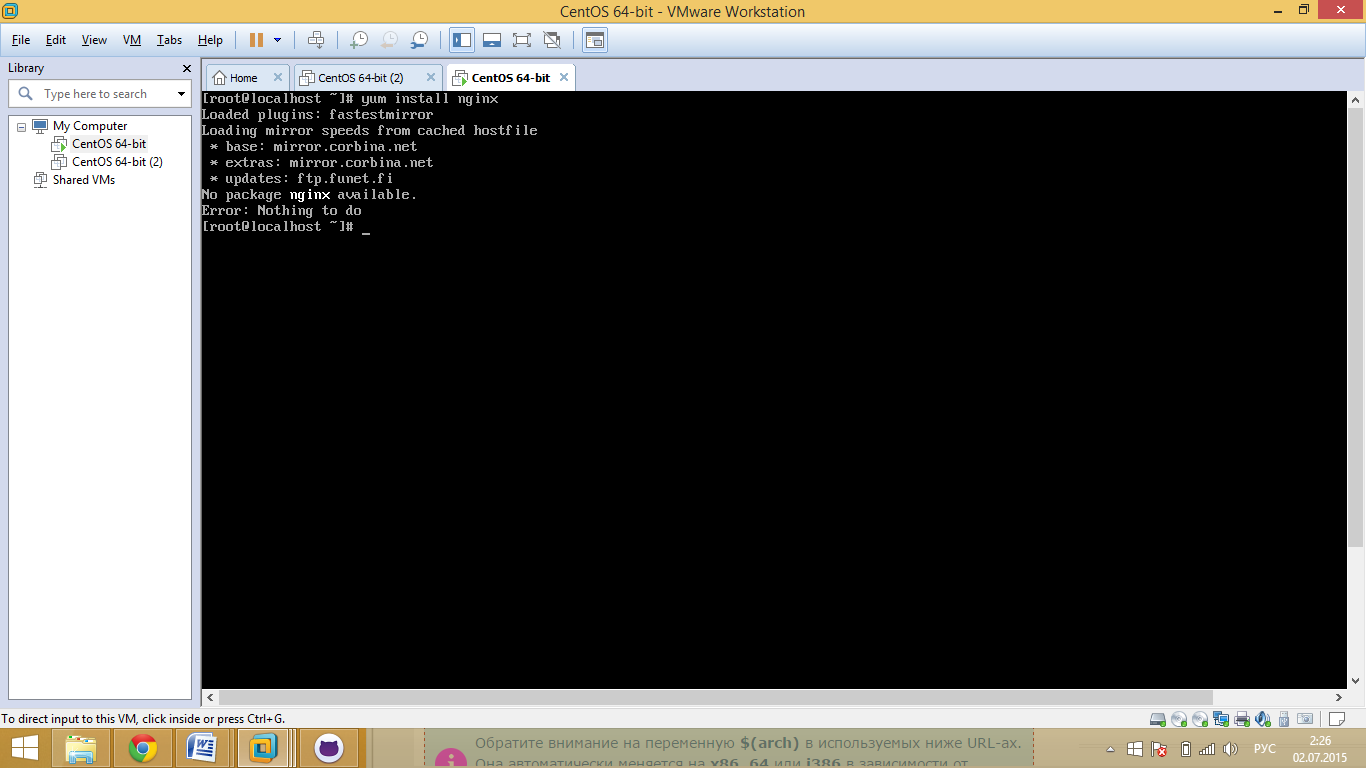
4. Устанавливаем на CentOS веб-сервер nginx.

Веб-сервер - сервер, принимающий http-запросы от клиентов(обычно веб-бразуер) и выдающий http-ответы, содержащие html-страницы, различные файлы, картинки и т.д. Под веб-сервером принято понимать как программное обеспечение выполняющее функции сервера, так и непосредственно [компьютер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) , на котором это программное обеспечение работает.

nginx - веб-сервер и почтовый прокси-сервер, работающий на UNIX-подобных операционных системах, разработанный Игорем Сысоевым, применение nginx целесообразно прежде всего для статических веб-сайтов и как прокси-сервера перед динамическими сайтами

УСТАНОВКА nginx

* При попытке установить nginix с помощью стандартных манипуляций

*yum install nginx* получаем ошибку 

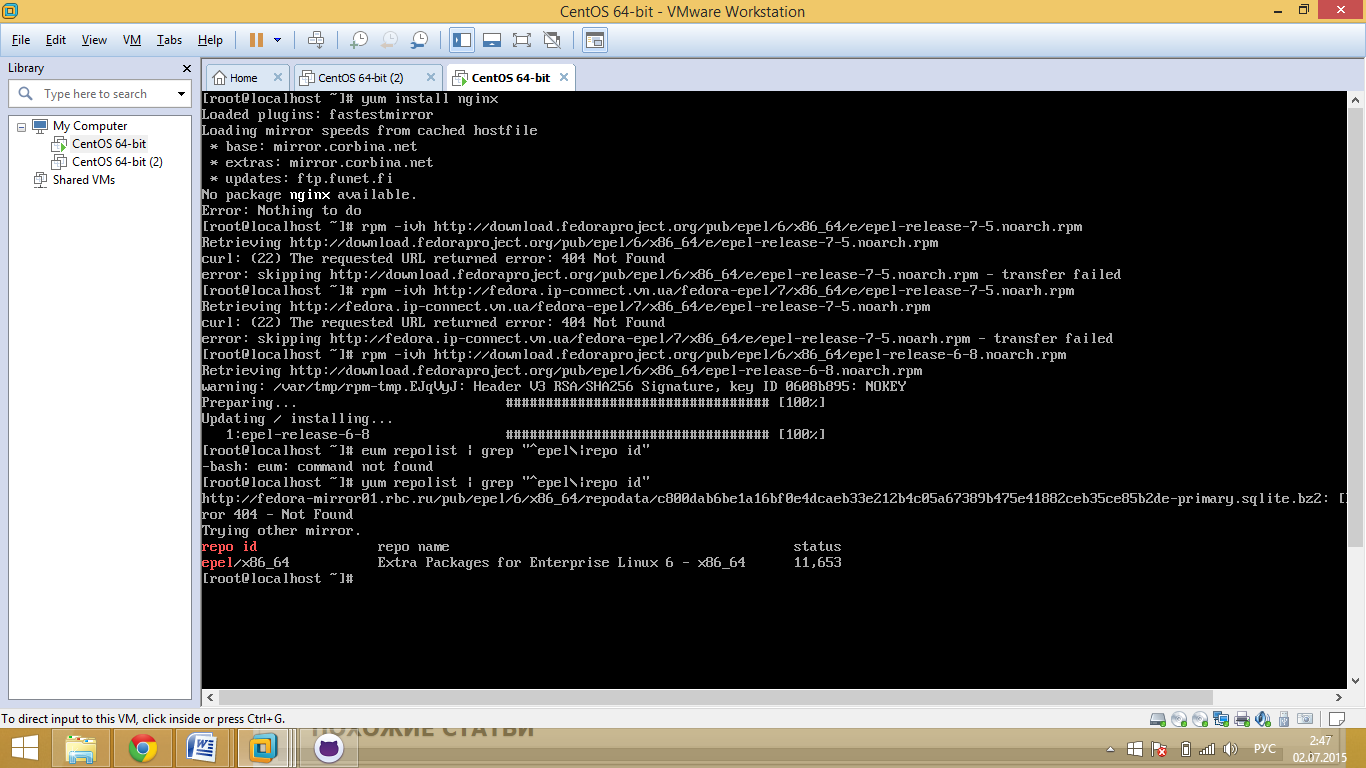
Данная ошибка связана с тем, что веб-сервер nginix хранится не в стандартном репозитории, а в EPEL (Extra Packages for Enterprise Linux) - это основанный сообществом открытый и бесплатный проект репозитория, предоставляемый командой Fedora, который содержит высококачественные пакеты дополнительного программного обеспечения для Linux дистрибутивов.

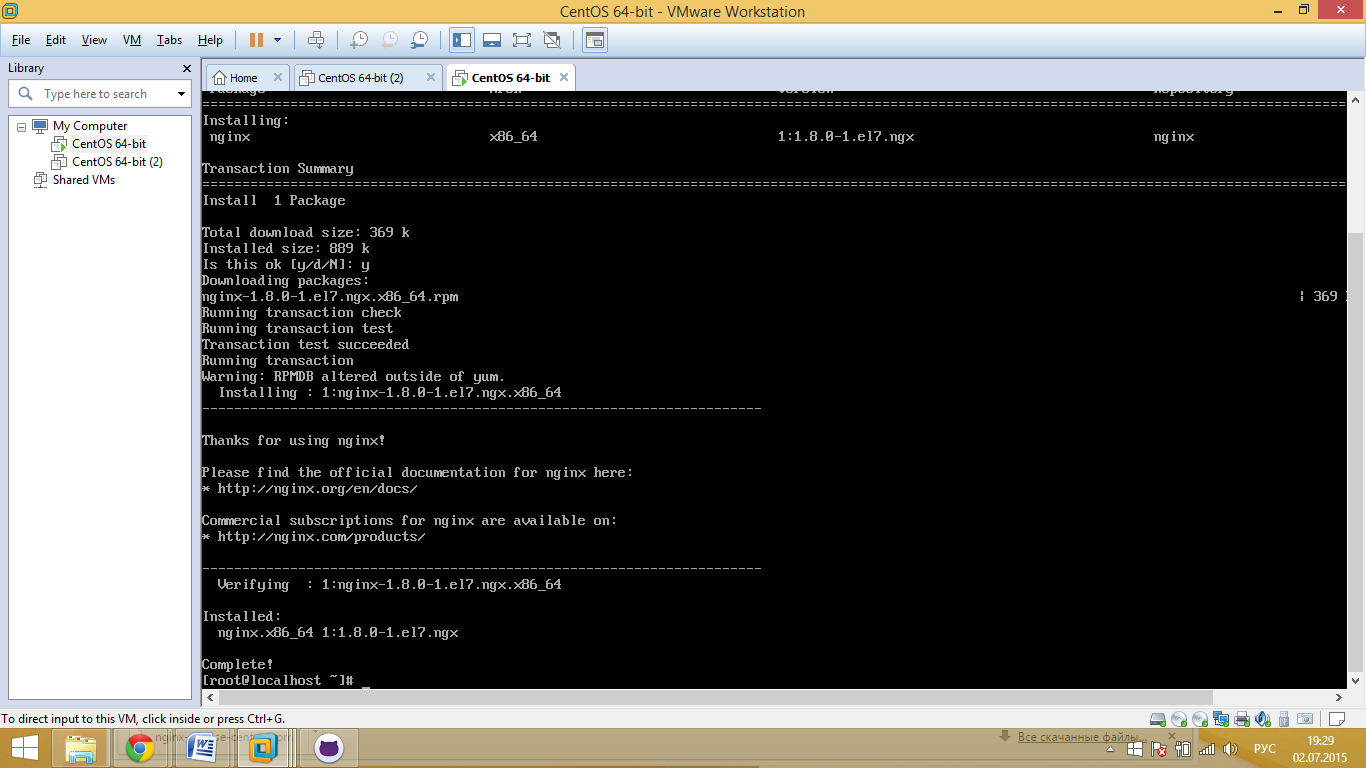
Добавляем репозиторий EPEL, что в дальнейшем через yum можно было с ним взаимодействовать. Делаем мы это следующей командой :

*rpm -ivh http://download.fedoraproject.org/pub/epel/6/$(arch)/epel-release-6-8.noarch.rpm*

Для проверки добавления репозитория используем команду

*yum repolist | grep "^epel\|repo id"*

**

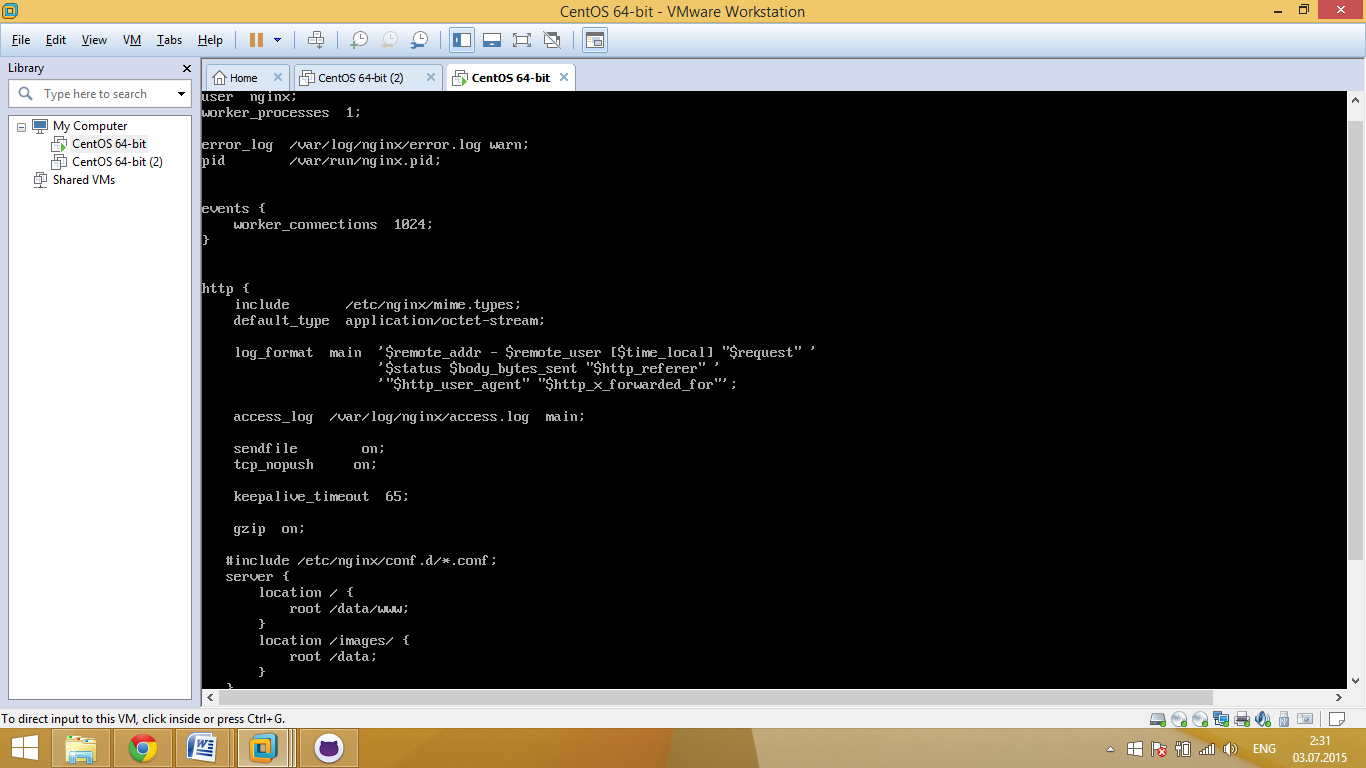
* Теперь можно установить nginx с помощью стандартной команды *yum install nginx*
* возможен вариант альтернативной установки веб-сервера nginx(с EPEL недокачивает несколько файлов, программа не запускается), заключается он в скачивание напрямую rpm-пакета с сайта nginx.com, с помощью команды *rpm - ihv http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm*

после установки пакета, опять же с помощью команды *yum install nginx*, производим установку

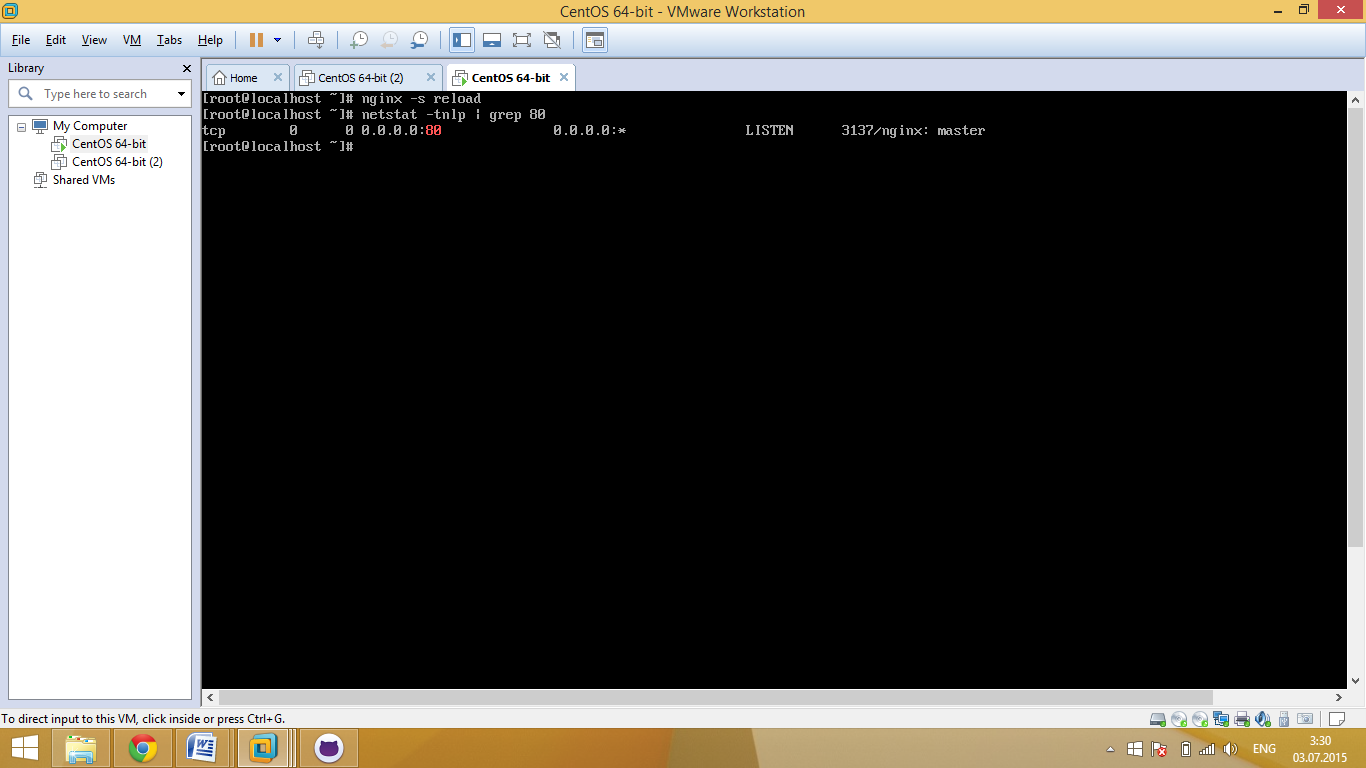
* в любом случае добавление репозитория EPEL является полезным, так как данный сервис предоставляет множество программ с открытым кодом для установки через yum, не содержит пакетов, конфликтующих с базовыми пакетами ядра; нет проблем с совместимостью; пакеты из EPEL чаще обновляются

НАСТРОЙКА nginx

* для настройки веб-сервера nginx редактируем файл конфига, находящийся в */etc/nginx/nginx.conf ,* добавляем в директиву http директиву server c вложенными в неё директивами location
* в директиве location указываем путь к файлу, находящемуся в локальной файловой системе (это может быть html-страница, текстовый документ, картинка и т.д.), префикс location сравнивается с URI из запроса, при совпадении нескольких блоков location nginx выбирает блок с самым длинным префиксом
* добавляем путь к заранее созданной директиве и хранящемуся в ней текстовому файлу *root /data/www*, соответственно при запросе по адресу http://localhost/ nginx отправит файл хранящийся по указанному пути
* мы получаем готовую конфигурацию сервера, слушающего на стандартном 80 порту (для замены порта со стандартного 80-го на любой другой можно дописать в директорию server команду *listen "номер порта"*) и находящемуся на локальном компьютере по адрессу *http://localhost/*



* для применения новой конфигурации воспользуемся командой *nginx -s reload b* и проверим слушается ли 80 порт nginx утилитой *netstat -tnlp | grep 80*
* утилита netstat(в минимальной версии centOS отсутствует, пришлось загружать её), предназначается для получения сведений о состоянии сетевых соединений и слушаемых на данном компьютере портах TCP и UDP, а также, для отображений статистических данных по сетевым интерфейсам и протоколам



* http://localhost/ - IP-адрес **(**127.0.0.1**)**, с помощью которого компьютер может обратиться по сети к самому себе, независимо от наличия у него подключения к сети

5. SSH-соединение.SSH-ключи. Настраиваем SSH-сервер на СentOS. Заходим с одной ВМ на другую, используя SSH-соединение. Учимся управлять ВМ с помощью утилиты putty и обмениваться файлами между реальной машиной и ВМ с помощью pscp.

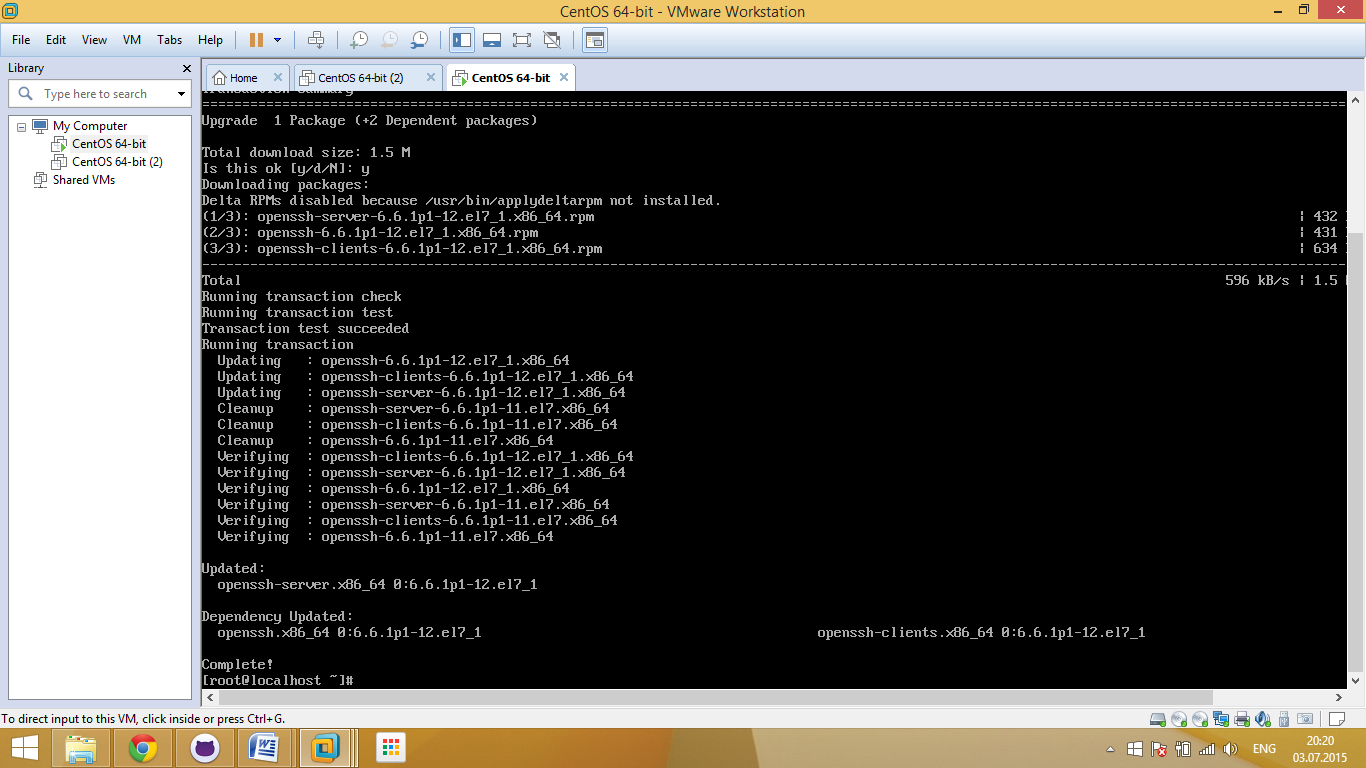
SSH — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и передачу файлов, по функциональности схож с протоколами Telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли, SSH допускает выбор различных алгоритмов шифрования, SSH-клиенты и SSH-серверы имеются для большинства операционных систем

SSH позволяет безопасно передавать в незащищённой среде практически любой другой [сетевой протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), таким образом, можно не только удалённо работать на компьютере через [командную оболочку](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BE%D0%BB%D0%BE%D1%87%D0%BA%D0%B0), но и передавать по шифрованному каналу звуковой поток или видео

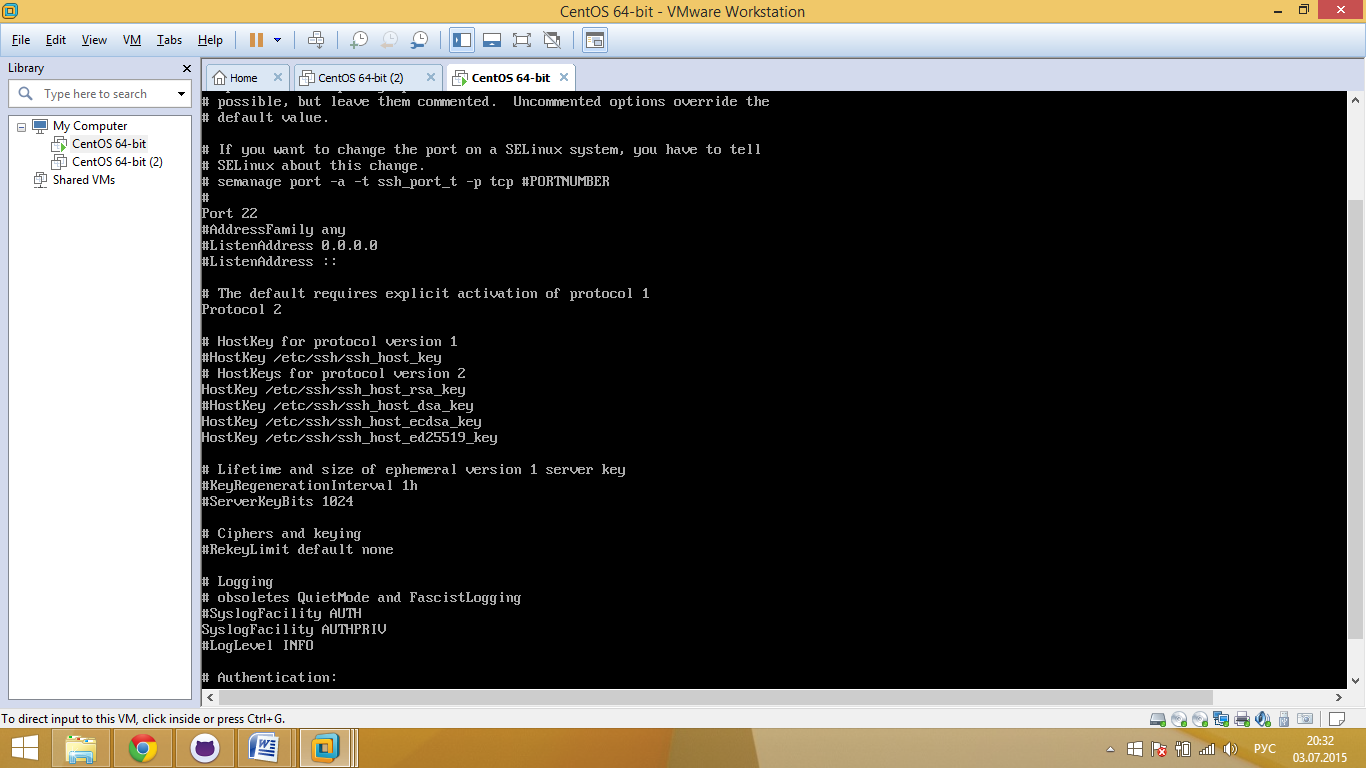
PuTTy - это популярный SSH-клиент , программа для безопасного подключения к удаленному компьютеру (или к серверу) и выполнения на нем различных команд, PuTTY ведет логи, позволяет настраивать шрифты, цвета и разрешение консоли, допускает сохранение в своей памяти ключей авторизации, поддерживает работу через прокси-сервер

pscp - клиент для SCP (удаленное копирование файлов по шифрованному протоколу scp с управлением из командной строки)

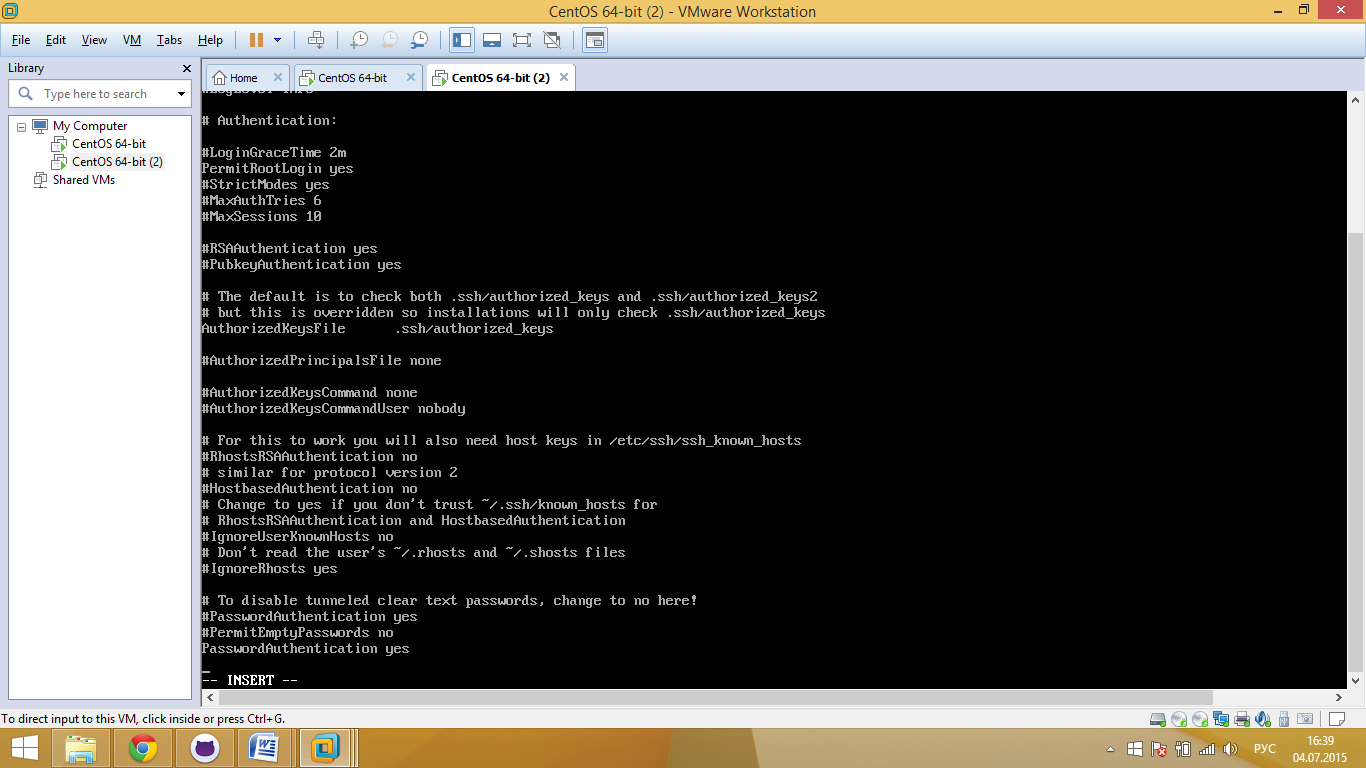
* создаём вторую виртуальную машину, производим соответсвующие настройки сети и установку софта, аналогично первой ВМ
* для настройкии ssh-соединения между ВМ используем openssh-server, устанавливаем на обе ВМ пользуюсь командой *yum install openssh-server*



* после установки идём в конфиг прграммы
* первый параметр – Port, по умолчанию используется 22 порт, можно изменить его на нестандартный порт 2203 – это избавит наш сервер от сетевых роботов, которые автоматически сканируют интернет в поиске открытых портов и пытаются через них подключиться (раскомментируем данный параметр в конфигах на обеих машинах)
* **#ListenAddress:: #ListenAddress 0.0.0.0 -** эти строки отвечают за настройку разграничений по сетевым интерфейсам, сетевому адресу или имени компьютера, по умолчанию сервер слушает (принимает подключения) на всех сетевых интерфейсах (оставляем без изменений)
* следующий параметр отвечает за версию протокола SSH (по умолчанию значение 2), оставляем 2, так как первая версия протокола SSH считается не безопасной (раскомментируем данный параметр в конфигах на обеих машинах)
* строки HostKey необходимы для второй версии протокола SSH и отвечают за названия файлов ключей и их расположение, эти ключи используются при аутентификации с ключом хоста



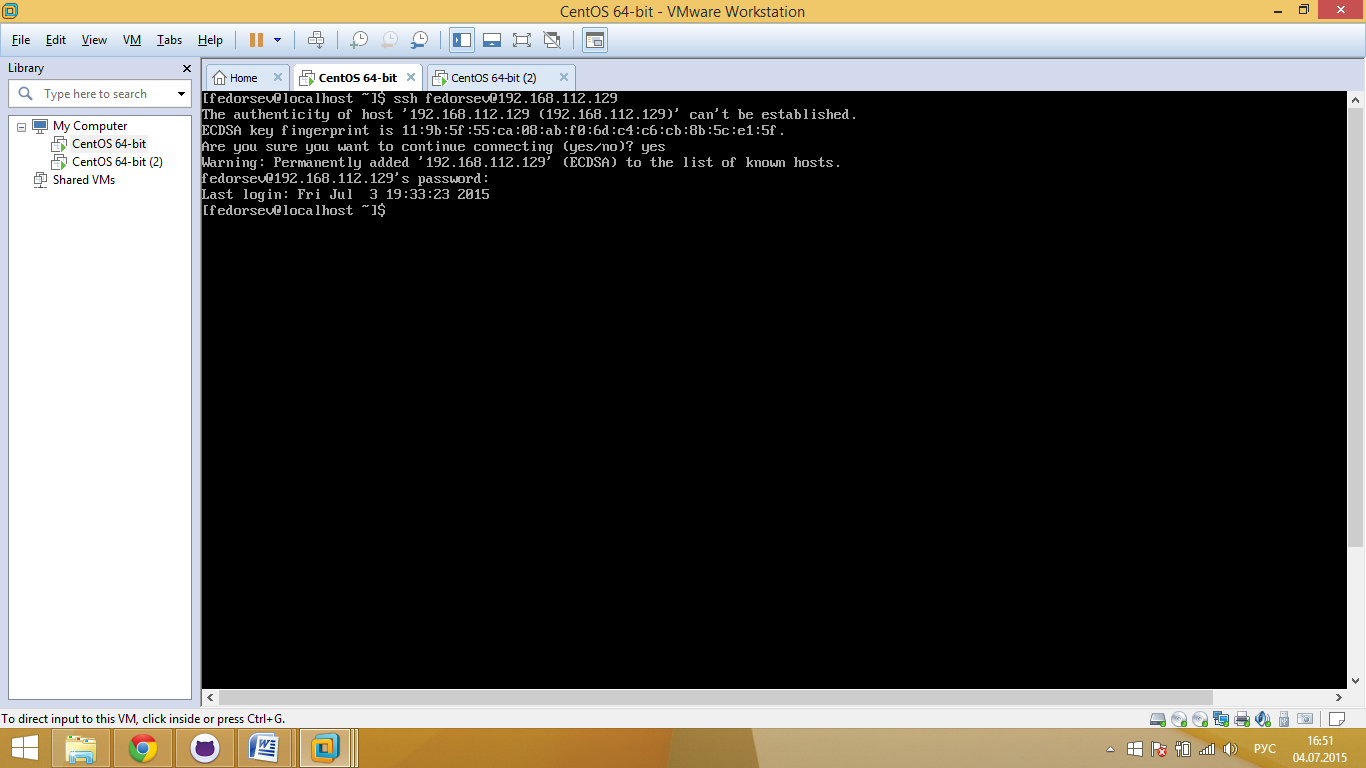
* следующая группа параметров относится к аутентификации, первый параметр означает, что соединение будет разорвано через указанное количество секунд, если пользователь не войдёт в систему **LoginGraceTime 2m**
* Второй параметр разрешает или запрещает вход по SSH под рутом (запрещаем) **PermitRootLogin no**
* Добавляем параметр AllowUsers, которого нет в конфигурационном файле по умолчанию. Этот параметр разрешает доступ к серверу по протоколу SSH только для перечисленных пользователей **AllowUsers fedors**



* теперь с основной системы соединяемся удалённым сервером с помощью команды

*ssh имя\_удалённого\_сервера@айпи\_удалённого\_сервера*

(айпи предварительно узнаем пользуюсь командой *ip -a)*

* далее получаем предупреждение, что подлинность данного хоста не установлена, видим ESCDA ключ и подтверждаем подключение, полуаем уведомление о том, что наш айпи запомнен, вводим пароль удалённого сервера и таким образом получаем доступ к удаленному хосту**
* Пара советов по настройке конфига, для обеспечения большей безопасности соединения :

1) Запрещение удалённого [*root*](https://ru.wikipedia.org/wiki/Root)-доступа.

2) Запрещение подключения с пустым паролем или отключение входа по паролю.

3) Выбор нестандартного порта для *SSH*-сервера.

4) Ограничение списка [IP-адресов](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81), с которых разрешён доступ (например, настройкой [файервола](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D1%80%D0%B2%D0%BE%D0%BB" \o "Персональный файрвол)).

5) Запрещение доступа с некоторых потенциально опасных адресов.

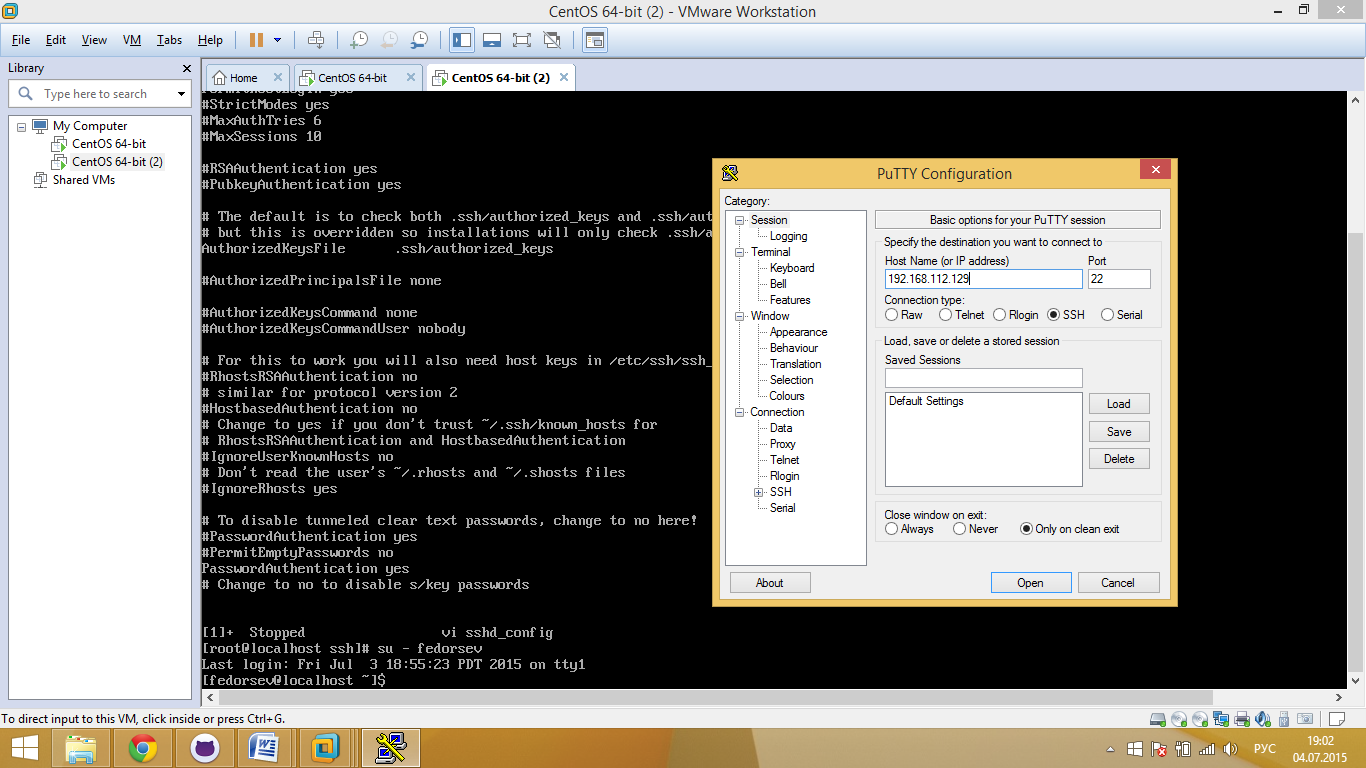
6)Регулярный просмотр сообщений об ошибках [аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F).

7)Установка [систем обнаружения вторжений](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BE%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D1%82%D0%BE%D1%80%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) *(IDS)*.

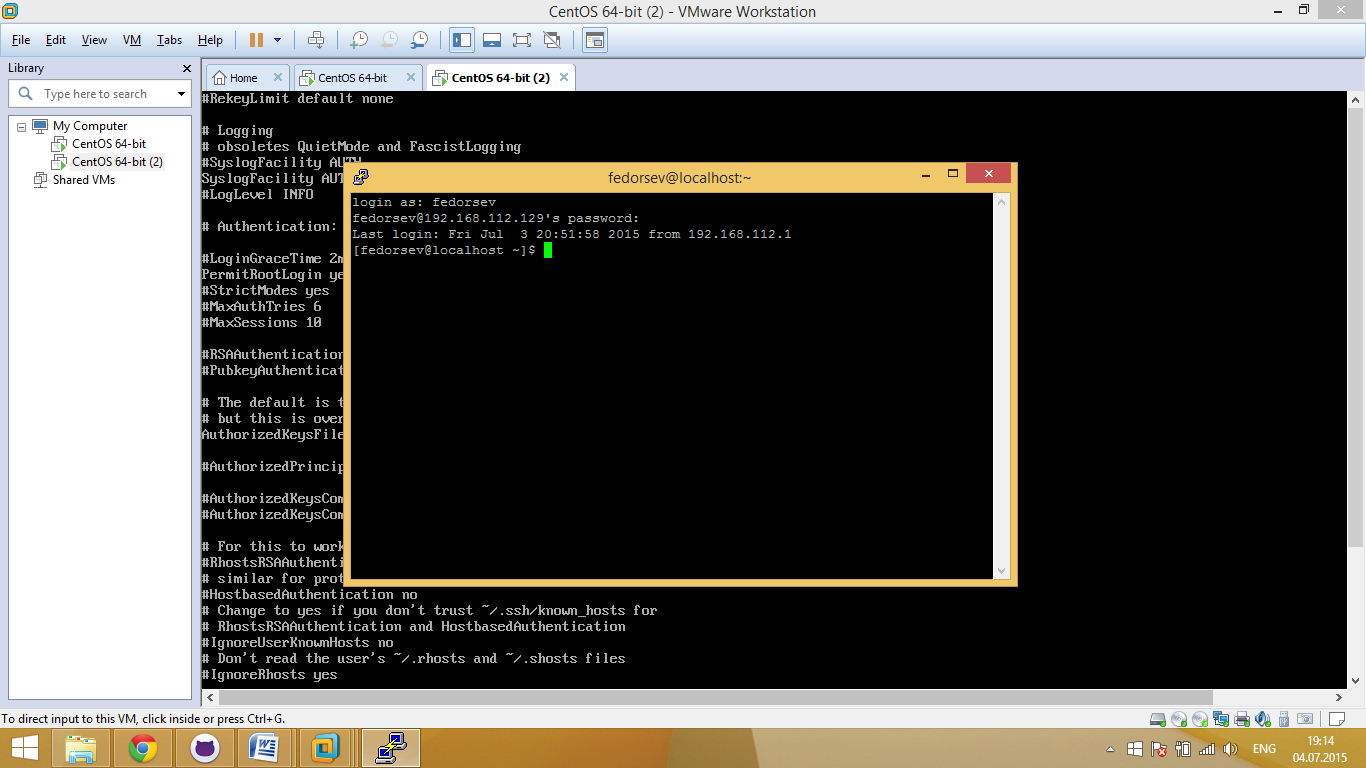
* теперь научимся пользоваться утилитой putty, подключимся к ВМ с реальной машины:

скачиваем putty с официального сайта программы *http://www.putty.org/*

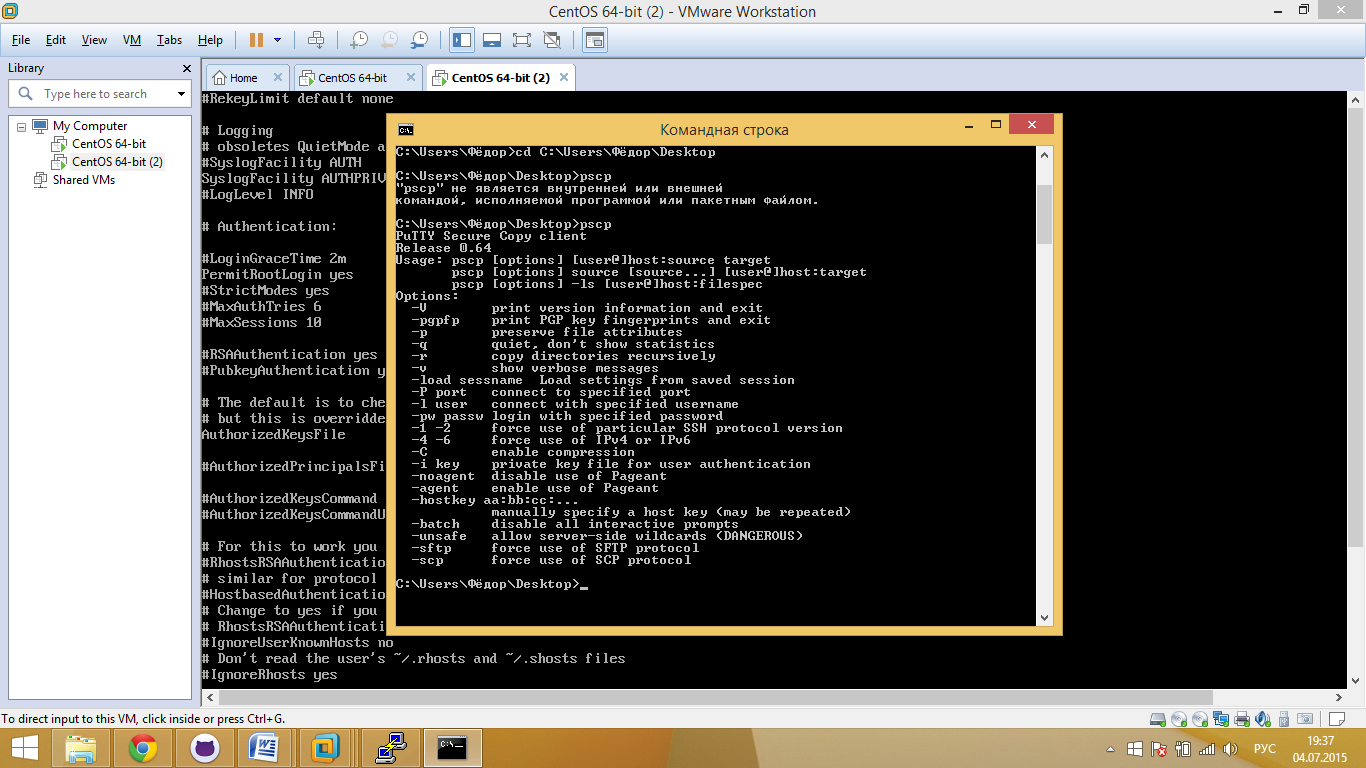
запускаем putty, в поле host name(IP adress) вводим айпишник виртуалки, нажимаем open



откроется консоль, в которой вводим логин и пароль, доступ получен



* используя утилиту pscp входящую в состав пакета putty, передадим какой-нибудь файл с реальной машины виртуальную, для этого запустим на реальной машине(в командной строке Windows) pscp.exe



* соответственно если мы хотим скопировать файл с реальной машины на виртуалку, пользуемся командой *pscp С:\\адресс файла имя\_удалённого\_хоста@айпи\_удалённого\_хоста:/папка в которой будет сохренён файл*
* для копирования же файла с виртуалки просто *пишем pscp имя\_удалённого\_хоста@айпи\_удалённого\_хоста:/папка в которой харанится файл*

SSH-ключи

Ключи SSH служат средством идентификации при подключении к серверу SSH с использованием [криптосистемы с открытым ключом](http://en.wikipedia.org/wiki/ru:%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC) и [аутентификации вызов-ответ](http://en.wikipedia.org/wiki/ru:%D0%92%D1%8B%D0%B7%D0%BE%D0%B2-%D0%BE%D1%82%D0%B2%D0%B5%D1%82_(%D0%B0%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F)). Одним из непосредственных достоинств этого метода перед традиционной идентификацией с помощью пароля является то, что вы можете быть авторизованы на сервере без регулярной необходимости отсылать ваш пароль через сеть. Даже если кто-либо будет прослушивать ваше соединение, у него не будет возможности перехватить и взломать ваш пароль, поскольку фактически он никогда не передается. Также использование для идентификации ключей SSH устраняет риск, связанный с брут-форс атаками, за счет существенного уменьшения шанса атакующего угадать правильные учетные данные.

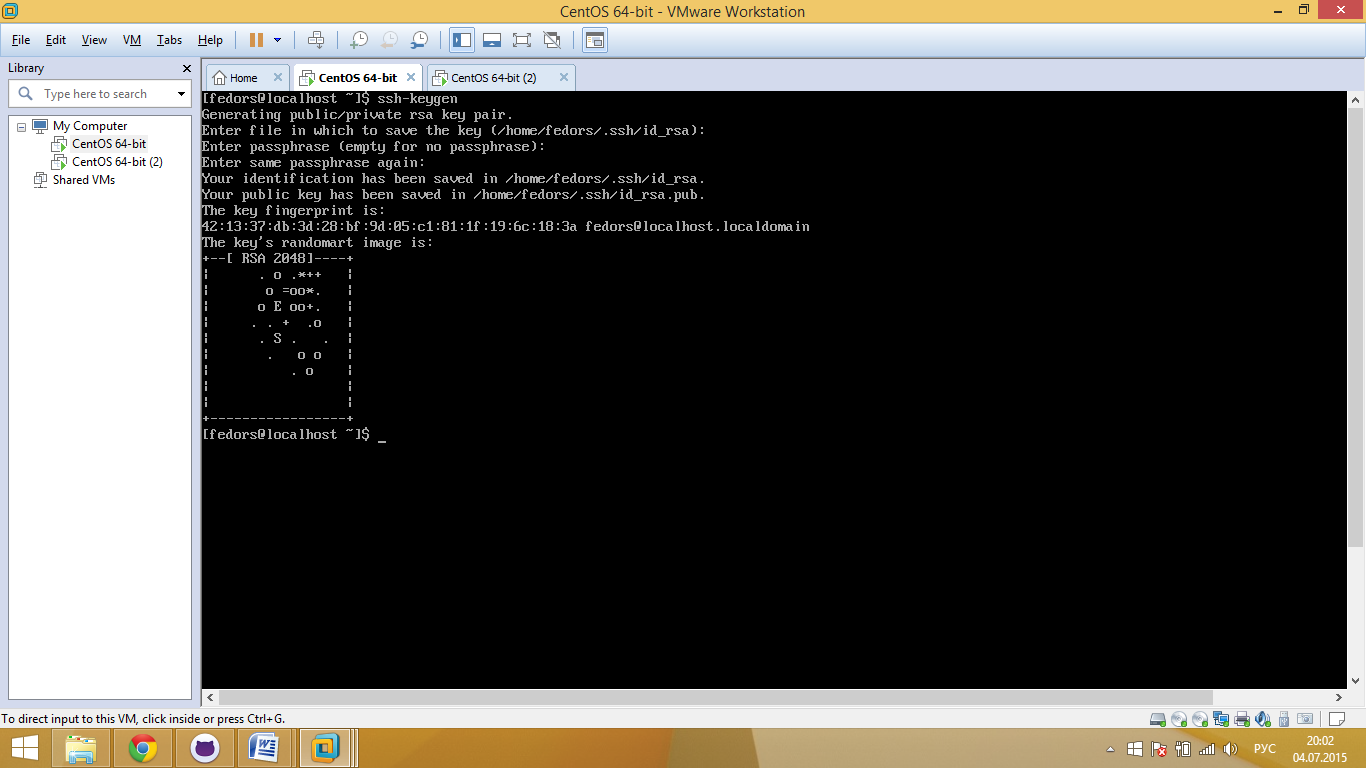
Ключи SSH являются парными: один из них - закрытый, другой - открытый. Закрытый ключ известен только нам. Открытый ключ может свободно раздаваться с любого сервера SSH, к которому вы хотите подключиться.

Когда у сервера SSH есть ваш открытый ключ в файле, и он видит, что вы запрашиваете соединение, он использует этот открытый ключ, чтобы создать и отправить вам вызов. Этот вызов является чем-то вроде зашифрованного сообщения, на которое должен поступить соответствующий ответ, чтобы сервер предоставил вам доступ. Безопасным это сообщение делает тот факт, что оно может быть прочитано только кем-то, у кого есть закрытый ключ. Открытый ключ может быть использован для зашифровки сообщения, но расшифровать то же самое сообщение он не сможет. Только вы, держатель закрытого ключа, будете иметь возможность корректно принять вызов и создать соответствующий ответ.

Этот этап вызов-ответ проходит незаметно для пользователя. До тех пор, пока у вас есть закрытый ключ, который обычно хранится в каталоге ~/.ssh/, ваш клиент SSH будет иметь возможность отправить правильный ответ серверу.

Поскольку закрытые ключи считаются конфиденциальной информацией, обычно они хранятся на диске в зашифрованном виде. По этой причине, когда запрашивается закрытый ключ, необходимо ввести пароль для расшифровки этого ключа. Внешне это может быть похоже на ввод пароля непосредственно на сервере SSH, но это не так: этот пароль используется только для расшифровки закрытого ключа в вашей локальной системе. Этот пароль не передается и не должен передаваться через сеть.

* генерируются эти пароли с помощью команды ssh-keygen



* Меняем в конфигурационном файле ssh следующие пункты: *PubkeyAuthentication yes  
  AuthorizedKeysFile "путь до файла с ключом"*

Дальнейшее подключение к удалённой машине происходит аналогично, может потребоваться указать пароль и секретную фразу для ключа, если вы указывали её при генерации.