Remote base skillup

for automated checks

Admins External Learning

0. Подготовка окружения

Описание:

Инструкции в этом шаге предполагают, что ты получил письмо с ссылкой на этот документ и доступами к серверам, на которых будет выполняться обучающее задание. Первым шагом будет подключение к серверам через SSH клиент. Для этого нам понадобится следующая информация из письма.

Сбор информации:

Обратите внимание на IP адреса серверов и имена ssh пользователей. Они нам понадобится позднее, когда мы будем подключаться к серверам через SSH клиент:

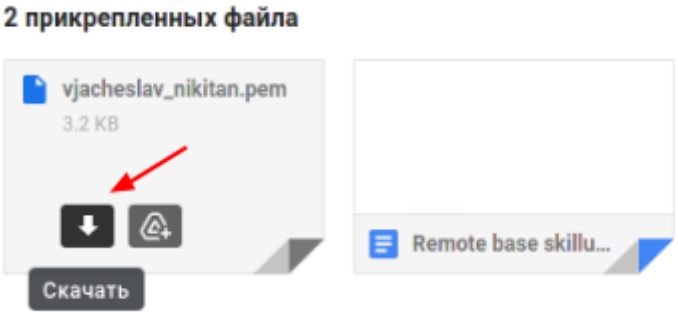
Ubuntu server:

IP: 3.17.48.178
SSH port: 22
User: ubuntu
SSH key attached

CentOS server:

IP: 3.138.47.7
SSH port: 22
User: centos
SSH key attached

Файл с расширением `.pem` в прикрепленных файлах - это ключ для SSH доступа к серверам. Скачайте ключ себе:



Еще нам нужен SSH клиент. Если подключаться с Windows, то для этой цели можно использовать, например, `KiTTY`, но подойдет и любой другой удобный тебе SSH клиент.

Подключение с Windows:

Скачать `kitty.exe` и `kittygen.exe` можно по любой из ссылок ниже:

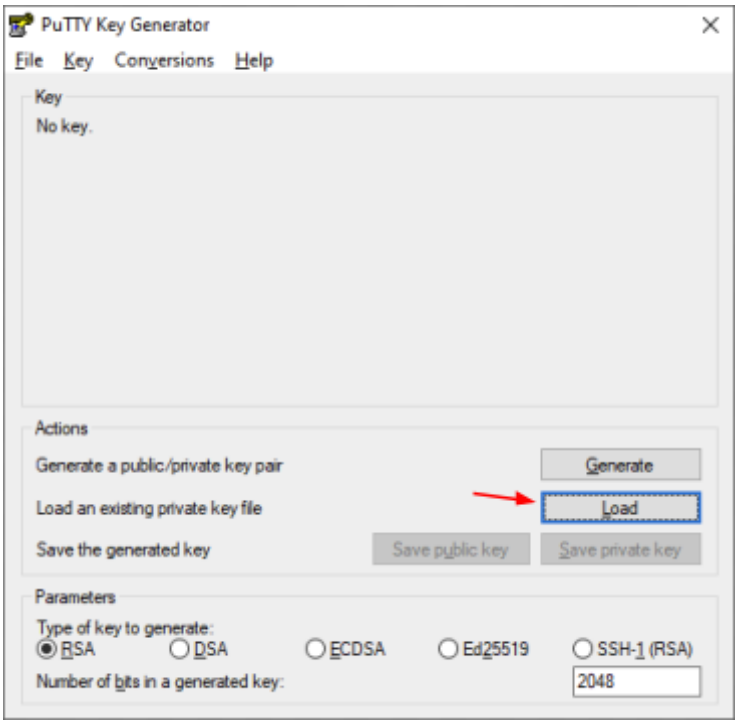
http://www.9bis.net/kitty/#!pages/download.md

https://github.com/cyd01/KiTTY/releases

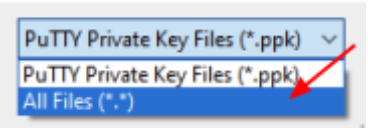
`kittygen.exe` понадобится, чтобы конвертировать ключ из формата `.pem` в понятный `KiTTY` формат. Этим мы сейчас и займемся.

Конвертация SSH ключа:

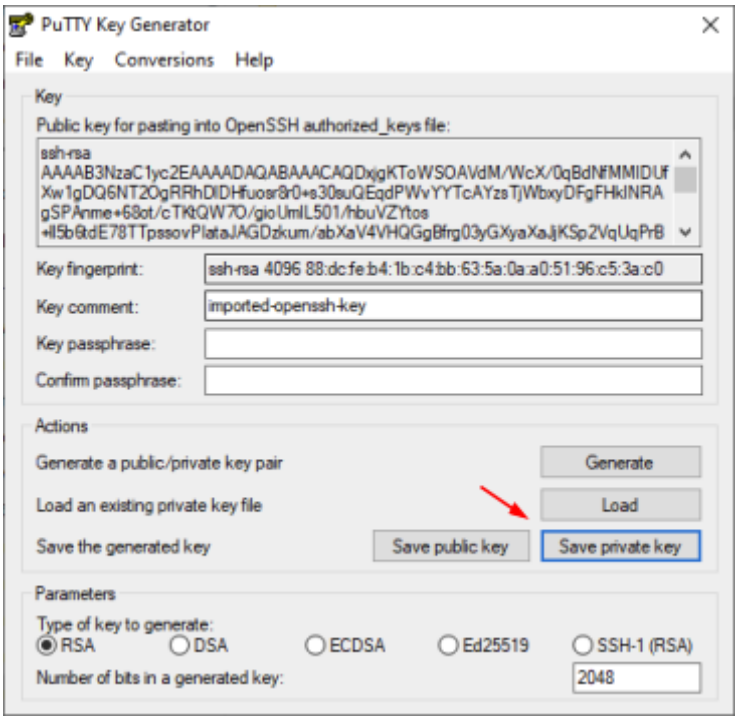
Запустите `kittygen.exe`, нажмите `Load`:



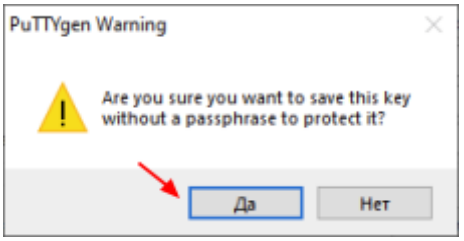
В открывшемся диалоговом окне смените тип файла на `All Files` и выберите `.pem` ключ:



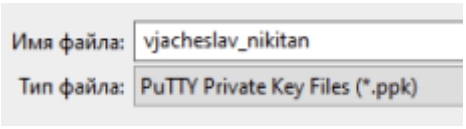
Появится диалоговое окно с сообщением об успешном импорте ключа. После этого мы можем сохранить ключ в новом формате, нажав `Save private key`:



Появится диалоговое окно с предупреждением о сохранении ключа без защиты паролем:



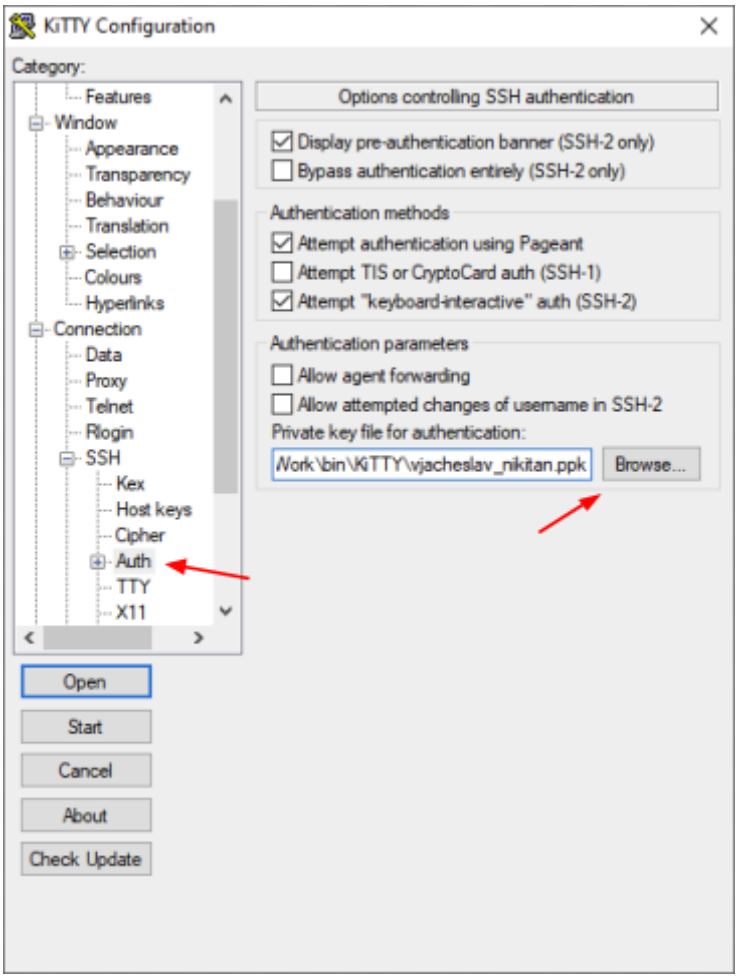
Нажмите `Да` и сохраните ключ в формате `.ppk` себе на диск:



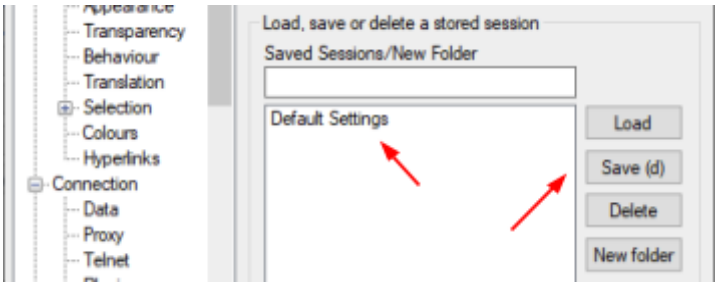
Теперь всё готово, можно подключаться к серверам.

Подключение к серверам по SSH:

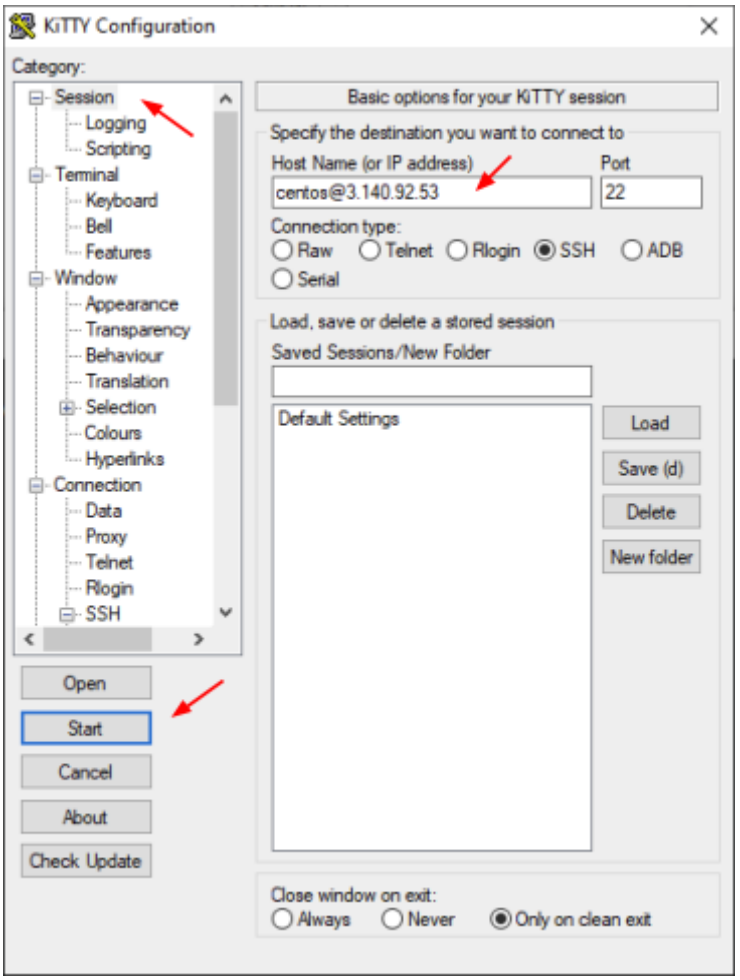
Запустите `kitty.exe`. В боковом древовидном меню `Category` выберите `Connection-SSH-Auth` и выберите ключ, нажав `Browse` под параметром `Private key file for authentication` раздела `Authentication parameters`:



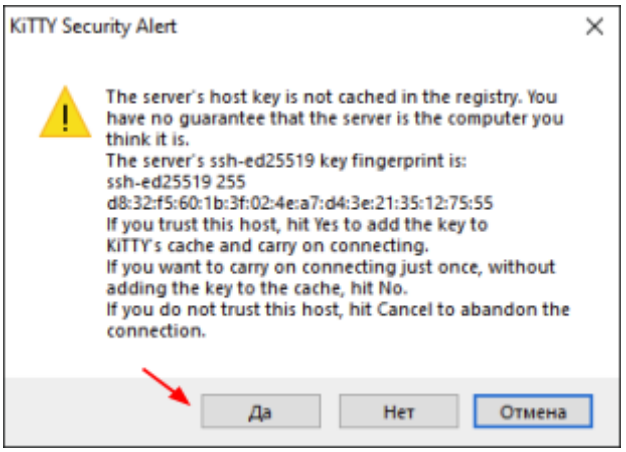
Вернитесь в начало древовидного меню и выберите `Session`. Выберите `Default Settings` справа, в разделе `Load Sessions/New Folder`, и нажмите `Save` - это избавит от необходимости указывать SSH ключ при каждом запуске `KiTTY`:



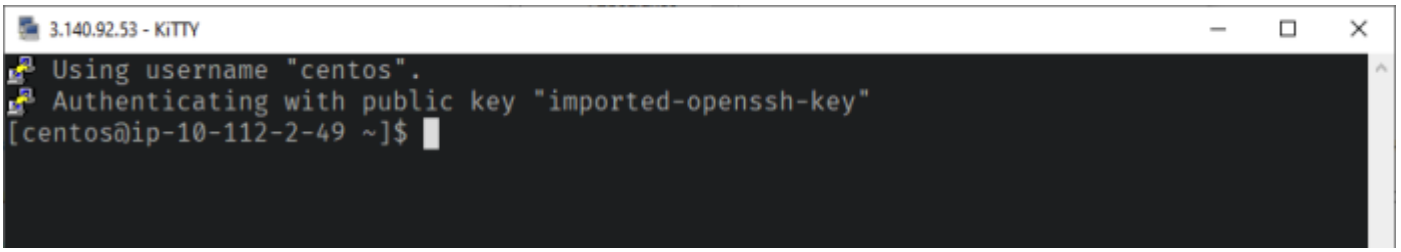
В поле `Host Name (or IP address)` введите строку вида `ubuntu@<server_ip>` для подключения к `Ubuntu` серверу или `centos@<server_ip>` для подключения к `CentOS` и нажмите `Start`:



Появится предупреждение о том, что отпечаток ключа не найден в кэше. Это нормально. Нажмите Да:



Откроется приглашение командной строки:

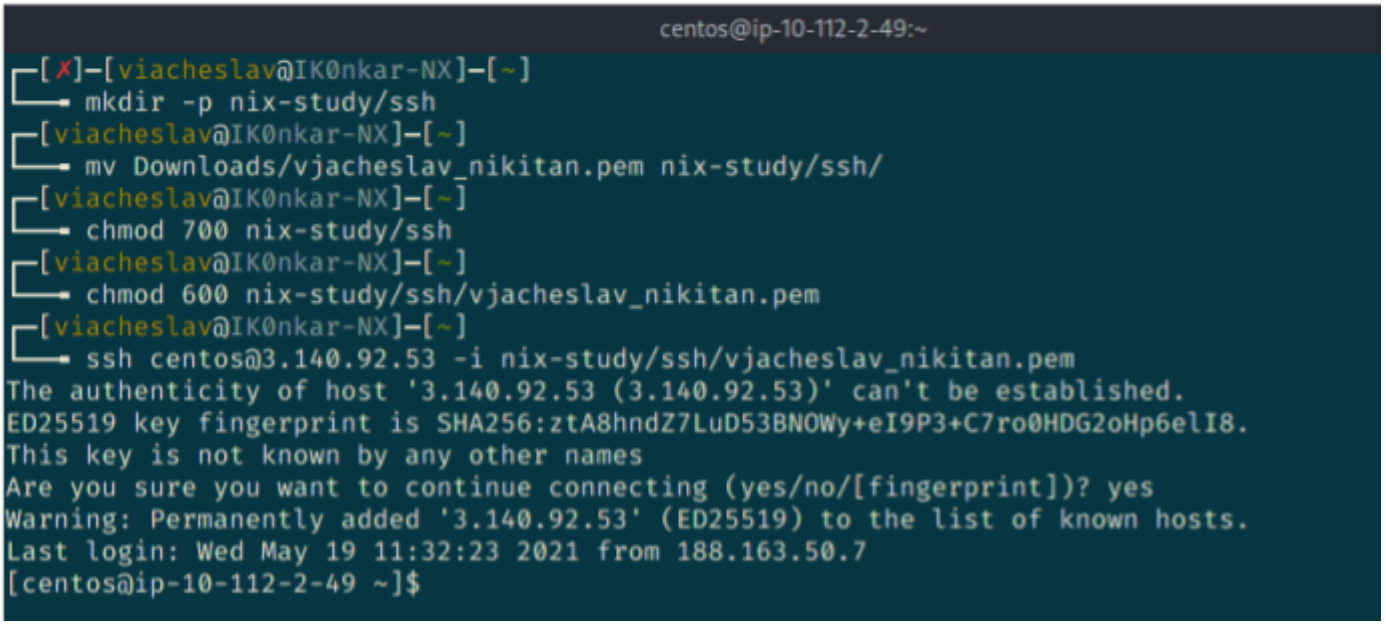


Вы подключились к серверу!

Подключение с Linux:

Скопируйте `.pem` ключ в любое удобное для себя место, например, `~/nix-study/ssh`. Смените права на директорию, где будет храниться ключ, на `700`, а на сам ключ - на `600`.

Подключитесь к серверу командой вида `ssh centos@<server_ip> -i /path/to/ssh_key.pem`. При первом подключении вы получите сообщение о том, что отпечаток ключа не найден в `known_hosts`. Это нормально. Напишите `yes`, затем нажмите `Enter`:



Вы подключились к серверу!

Теперь готово, чтобы приступать к выполнению задания.

⚠️ Важно

Настройка файрвола выходит за рамки этого набора заданий. Файрвол настраивать не нужно, дабы случайно не заблокировать себе доступ к серверу. То же касается и настроек сети.

При выполнении операций с дополнительным диском стоит быть крайне внимательным и не затирать системный раздел виртуального сервера. В противном случае, сервер нужно будет пересоздать, а задания придется выполнять с нуля.

При выполнении заданий постарайтесь не выходить за рамки требований. Если все же очень хочется и очень интересно настроить что-то дополнительно, или если вы не уверены в каком-то шаге, то хорошей идеей будет создать себе виртуальную машину, например, локально, и сначала попробовать там, дабы минимизировать риск потери сервера.

1. Операционные системы

Цель:

Познакомиться с дистрибутивами CentOS и Ubuntu, их пакетными менеджерами, работа с дисками.

Работа с пакетными менеджерами

Базовая работа с пакетными менеджерами включает в себя поиск и установку нужных пакетов, а также добавление сторонних репозиториев.

Разумеется, пакетные менеджеры предоставляют гораздо более широкий выбор возможностей и инструментов для управления пакетами. Мы попрактикуемся следующих из них: просмотр содержимого пакета и поиск пакета, которому принадлежит указанный файл.

Иногда, требуемого пакета в репозиториях, будь то базовые, дополнительные или сторонние, нет. Тогда возникает необходимость собрать приложение из исходного кода. Сборку из исходников мы попробуем на примере установки свежей версии `git`.

Задание:

На виртуальной машине с CentOS:

- Отключить `SELinux`
- Подключить репозитории `epel` и `remi`. Убедиться, что репозитории активированы
- Из репозитория `remi` установить `redis` версии `6.0`
  Для установки использовать `<package_name>-<package_version>`, а не веб-ссылку на пакет
- При помощи команды `rpm` вывести список файлов, содержащихся в пакете `redis`, вывод команды записать в `/root/tasks/1/packagefileslist`
- При помощи команды `rpm` определить имя пакета, содержащего файл `/etc/locale.conf`, вывод команды записать в `/root/tasks/1/filetopackagename`

На виртуальной машине с Ubuntu:

- Установить `Redis` последней версии из официального `PPA` репозитория
- При помощи команды `dpkg` вывести список файлов, содержащихся в пакете `redis-server`, вывод команды записать в `/root/tasks/1/packagefileslist`
- При помощи команды `dpkg` определить имя пакета, содержащего файл `/etc/sysctl.conf`, вывод команды записать в `/root/tasks/1/filetopackagename`

Сборка ПО из исходных кодов:

- Собрать и установить `git` последнего стабильного релиза из исходников на CentOS и Ubuntu
- Путь к исполняемому файлу `git` рекомендуется выбирать из стандартного `PATH` (например, `/usr/local/sbin`) или добавить каталог бинарного файла в `PATH` пользователя `root`, - иначе проверка может его не обнаружить
- Установленного из репозитория пакета, предоставляющего исполняемый файл `git`, в системе быть не должно

Работа с дисками

Работа с дисками чаще всего включает в себя создание разметки, разделов, файловой системы, монтирование диска и добавление записи в `fstab` для монтирования раздела при загрузке системы.

В этом задании мы коснемся всех вышеизложенных пунктов и подготовим диск для использования как хранилища данных - файлов веб сайтов и базы данных.

Задание:

На обеих виртуальных машинах:

- Найти доступный системе, но **не примонтированный** диск
- На диске создать `gpt` разметку и один раздел
- Отформатировать раздел диска как `xfs` на CentOS и `ext4` на Ubuntu
- Примонтировать раздел как `/data`
- Первый раздел дополнительного диска должен монтироваться при запуске системы (для этого сделать запись в `fstab`, запись должна начинаться с `UUID=`)
- Создать `symlink` из `/var/lib/mysql` и `/var/www` к `/data/mysql` и `/data/www` соответственно. Иными словами, `/var/lib/mysql` должен быть ссылкой на каталог `/data/mysql`, а `/var/www` - на каталог `/data/www`
- Создать файл подкачки `/swap` размером `1GB` и подключить его к системе. Добавить соответствующую запись в `fstab`, чтобы файл подкачки автоматически монтировался при запуске системы

2. Веб-серверы nginx и apache как frontend/backend + PHP + MySQL

Цель:

- Научиться устанавливать и настраивать типичную связку веб-серверов.
- Научиться создавать кастомные виртуальные хосты.

Подготовка среды для веб-приложения

С настройкой веб-сервера рано или поздно сталкивается, наверное, любой системный администратор. И это совсем не удивительно - ведь большинство Интернет-трафика связано именно с web и сопутствующими технологиями. Посещение любимых сайтов и социальных сетей, использование различных сервисов - от мобильного банкинга до потокового видео и музыки, даже работа через Сеть, - все это стало неотъемлемой частью современного мира.

Знакомиться с настройкой web мы начнем на примере таких популярных веб-серверов, как `Nginx` и `Apache`. Также мы подготовим среду для развертывания динамических сайтов: для этого нам понадобится интерпретатор `PHP` и сервер баз данных `MySQL`. Оба являются широко используемыми решениями для построения различных программных продуктов и потому настройка связки `linux+nginx/apache+php+mysql` является типичной задачей.

Теоретические вопросы:

Какие бывают `HTTP` методы, коды состояния?

Задание:

- Установить на CentOS сервере пакеты
  - `nginx` (из официального репозитория nginx для стабильных версий пакетов)
  - `apache` (из оф. репозитория CentOS)
  - `php 8.0` из репозитория `remi` (не из репозитория `remi-php80`). Как минимум, необходимы основной пакет `php 8.0` (метапакет - пакет, имеющий зависимости на набор других пакетов, но ничего не содержащий сам по себе) и пакет, предоставляющий модуль для `apache`
  - `latest mysql` (из официального `Oracle` репозитория `MySQL`)
  - и зависимости для запуска php веб сайта.
- Запустить `MySQL` так, чтобы он не был доступен из вне. Записать пароль `MySQL` пользователя `root` в клиентский конфиг для системного `root` пользователя так, чтобы он использовался по-умолчанию при вызове `mysql CLI`
- Настроить `nginx` как обратный прокси-сервер перед `apache`. Для конфигурации `proxy_pass` использовать `upstream` не нужно. `Apache` должен работать на порту `8080` и быть недоступным из публичной сети.
  При необходимости можно или отключить `SELinux` или настроить доступ к `apache` из сети (`setsebool -P httpd_can_network_connect 1`)
- Настроить виртуал-хосты для `apache` и `nginx`:
  - `wordpress.example.com` который указывает на путь `/var/www/wordpress.example.com`
  - `drupal.example.com` который указывает на путь `/var/www/drupal.example.com`
  - использовать предусмотренную дистрибутивом директорию для конфигурационных файлов виртуальных хостов

- положить тестовый файл `index.html` с произвольным текстом для каждого виртуалхоста
- положить тестовый файл `test.php`, который вызывает PHP функцию `phpinfo()` для каждого виртуалхоста
- Для доступа к сайтам прописать записи для `drupal.example.com` и `wordpress.example.com` в `hosts` файл
- протестировать что URL http://wordpress.example.com/test.php и http://drupal.example.com/test.php отдают корректные ответы

Результаты:

- Рабочая связка `frontend/backend web сервер + PHP + MySQL`
- Настроенные виртуал хосты
- Установленные компоненты добавлены в автозагрузку

3. Развертывание простейших сайтов

Цель:

- развернуть простые `CMS`
- настроить `nginx` в связке с `apache`, `PHP-FPM (PHP 8.0)`
- настроить `FTP` (`pure-ftpd`)

Развертывание веб-приложений

Развертывание (или деплой) веб-приложений является одной из типичных задач наряду с подготовкой среды (или окружения) для самого приложения, - ведь у приложения, как правило, есть зависимости и требования к среде, где оно будет работать.

Список зависимостей и инструкции по установки, как правило, предоставляются и поддерживаются в актуальном состоянии поставщиком приложения, потому первым источником информации подобного рода должна стать именно официальная документация.

`FTP` сервер используется для передачи файлов. В контексте веб-сайтов по `ftp`, например, можно загружать медиа-файлы, которые затем будут использоваться для наполнения сайта содержимым, или даже обновлять код самого сайта.

Множество веб-сайтов строятся на основе популярных `CMS (Content Management System)`, таких как `Wordpress`, `Drupal` и `Joomla`. На них мы и потренируемся!

Примечание:

- В целях безопасности доступ к `FTP` портам из Интернета закрыт. Протестировать `FTP` можно используя консольные клиенты `ftp` или `lftp` с Ubuntu

Задание:

Делаем все на CentOS сервере

- Создаем базы данных для `wordpress` и `drupal` в `MySQL`. Для каждой БД должен быть свой непривилегированный пользователь
- разворачиваем `Drupal` на `drupal.example.com`
- для `drupal.example.com` настраиваем `nginx` как `reverse proxy` перед `apache` (`mod_php`)
- разворачиваем `Wordpress` на `wordpress.example.com`
- настраиваем `nginx` как веб-сервер для `wordpress.example.com` (`nginx` обрабатывает `PHP`, используя `PHP-FPM`)
- настраиваем `FTP` с доступом ко всем сайтам (в пассивном режиме `PASV`, но не `extended passive EPSV`). Используем `pure-ftpd` с аутентификацией на основе `virtual users`. Для каждого сайта должен быть свой `ftp` пользователь: `drupal` для `drupal.example.com`, `wordpress` для `wordpress.example.com`. Пользователи должны иметь доступ только к web каталогу сайта.
- Добавить в файл `/root/.netrc` конфигурацию для автоматического логина при подключении к `FTP` по внутреннему сетевому адресу хоста (не `127.0.0.1`) в формате:

  ```
  machine <host private ip> login <ftp_user_login> password <ftp_user_password>
  ```

  Проверить можно командой `` `ftp -p -d <host private ip>` ``.
- тестируем

Условия выполнения задачи:

- `PHP-FPM` (для `PHP 8.0`) запущен от пользователя `php-fpm` на порту `9000` и не доступен из внешней сети
- `Nginx` запущен от пользователя `nginx`
- Права на каталоги не больше `775`, на файлы `644`, никаких `777` и `666` быть не должно (исключение `/var/www/drupal.example.com/sites/default/files/php/*`)
- Можно загружать файлы и по `FTP`, и через админку `Wordpress` (медиафайлы)
- Файлы, которые были загружены по `FTP`, доступны для добавления в статьи (посты) и так же для удаления через `Wordpress`
- Файлы, которые были добавлены через `Wordpress`, доступны для удаления через `FTP`
- После изменении даты (месяц/год) загрузка/удаление файлов работает корректно

Результаты:

– Развернуты 2 `CMS`
– рабочие сайты (открываются все ссылки, аутентификация работает, статьи создаются, контент аплоадится)
– Настроен `FTP` доступ

4. Перенос сайта

Цель:

Получить навыки подготовки среды для переноса существующего приложения на сервер с другим дистрибутивом linux; практика переноса веб-сайта с сервера на сервер.

Перенос веб-сайта

Причин для переноса может быть несколько - смена хостинга, поломка "железа", устаревание ОС и среды работы приложения на столько, что проще мигрировать в новую, чем обновлять существующую... Так или иначе, переезд, рано или поздно, случается. И к этой задаче стоит быть готовым, ведь она может оказаться труднее, чем кажется на первый взгляд - в новой среде приложение может попросту не заработать!

И на это причин немало: код приложения может быть несовместим с новыми версиями ПО, конфигурация ПО в новой среде должна соответствовать старой, при переносе можно потерять права и владельцев файлов, а также потерять ресурсы, на которые могут указывать наличествующие в файлах приложения символические ссылки, - и это далеко не полный список.

Поэтому перенос сайта - задача, к которой стоит отнестись со всей ответственностью и тщательной подготовкой.

Задание:

Перенести `wordpress.example.com` сайт с CentOS сервера на Ubuntu сервер. Версии компонентов и конфигурация сайта на новом сервере должны быть максимально идентичны тем что на старом. Вместе с сайтом необходимо также перенести конфигурацию для `FTP`.

Примерные шаги:

- установка нужных приложений на новом сервере
- Копирование файлов с исходного сервера на CentOS сервер
- дамп БД, перенос БД
- настройка/разворот сайта на новом сервере
- тестирование сайта

Результаты:

- перенесенный и рабочий сайт на CentOS сервере

5. Почтовая связка

Цель:

Получить практический опыт настройки почтовой среды и на примере рассмотреть взаимодействие ее компонентов.

Настройка почтового сервера

Электронная почта остается универсальным стандартом общения по сети, не смотря на то, что неофициальная переписка, в основном, ведется через социальные сети и мессенджеры.

В наше время многие организации используют готовые облачные почтовые сервисы, такие как Google Gmail или Microsoft Office 365. Но даже если ваша почтовая система работает в облаке, у вас по-прежнему найдется повод, чтобы понять и поддерживать ее работу, а также взаимодействовать с ней, как администратор. Если ваш проект использует локальные почтовые серверы, - объем работы будет также включать настройку, мониторинг и тестирование почтовой системы.

Настройку сервера почты мы рассмотрим на примере `postfix` - одного из самых популярных `MTA (Mail Transfer Agent)`, - агентов пересылки почты. Именно он занимается передачей писем между почтовыми серверами, отправляемых `MUA (Mail User Agent)`, - пользовательским почтовым агентом или, попросту, почтовым клиентом, и передачей их `MDA (Mail Delivery Agent)`, - агенту доставки почты, в нашем случае, `dovecot`, который осуществляет доставку писем получателям. Почтовый клиент может быть как веб-приложением - `roundcube`, так и клиентским приложением, - например, `thunderbird`.

Примичание:

- Не используйте пакет `postfixadmin` из репозитория - он устаревший

Задание:

- Сетапим на Ubuntu полноценный почтовый сервер. Связка должна состоять из `postfix` + `dovecot` + `mysql` + `postfixadmin` + `roundcube` и соответствовать следующим требованиям:
  - `Postfix` использует `sasl` аутентификацию пользователей через `dovecot`
  - Отправка локальной почты происходит через `LMTP`
  - `Dovecot` и `Postfix` пишут логи в `/var/log/maillog`
  - Данные о доменах/почтовых ящиках и т.п. берется из `MySQL` БД (структура базы данных соответствует типичной структуре, предлагаемой `Postfixadmin`)
  - Связка обеспечивает работу `SMTP`, `POP3`, `IMAP` на стандартных портах
  - Для аутентификации использовать полное имя почтового ящика `user@domain.tld`
- Добавить домены `example.com`, `example.net` и создать как минимум по одному почтовому ящику для каждого
- `Postfixadmin` должен быть доступен по ссылке http://wordpress.example.com/mailadmin
- `Roundcube` должен быть доступен по ссылке http://wordpress.example.com/webmail
- Создать файл `/etc/mailadmin` куда записать email и пароль, разделенные пробелом, для суперадмина в `Postfixadmin`. Убедится, что файл доступен на чтение и запись только `root` пользователю

- Создать файл `/etc/example.com`, куда записать email и пароль для доступа к любому валидному почтовому ящику в домене `example.com`. Убедится, что файл доступен на чтение и запись только `root` пользователю. Данные в файле должны быть представлены в следующем формате:

```
machine <host private ip> login <user>@example.com password <password>
```

- Создать файл `/etc/example.net`, куда записать email и пароль для доступа к любому валидному почтовому ящику в домене `example.net`. Убедится, что файл доступен на чтение и запись только `root` пользователю. Данные в файле должны быть представлены в следующем формате:

```
machine <host private ip> login <user>@example.net password <password>
```

Теоретические вопросы:

- какие существуют форматы хранения сообщений? чем они отличаются?
- как протестировать отправку/получение писем без почтового клиента?

Результаты:

- настроена возможность добавления новых доменов и почтовых ящиков
- почта между локальными доменами ходит беспрепятственно
- есть возможность подключиться к почте как через `webmail` так и через `thunderbird` (`pop3` + `imap`)

6. Бэкапы

Цель:

Научиться делать резервные копии сайтов и баз данных; создавать, настраивать и планово запускать скрипты резервного копирования.

Резервное копирование

Резервное копирование необходимо, чтобы иметь возможность восстановления данных, и является критическим элементом любой системы. Причин создавать резервные копии множество: оборудование ломается, люди удаляют файлы по ошибке и умышленно, судьи конфискуют все связанные с делом документы, которые хранятся компьютерах, владельцы продукта требуют уверенности в том, что природная или другая катастрофа не станет причиной потери их вложений.

Создание резервных копий может причинить неудобства: хранение копий требует финансовых вложений, а службы могут работать медленнее или не работать совсем, когда оно выполняется. Резервное копирование аналогично страховке: вы платите за него, хотя и надеетесь, что оно никогда вам не понадобится. На самом деле оно вам нужно.

Задание:

На CentOS сервере:

Написать простой `bash` скрипт для дампа БД:

- Настроить ежедневный дамп баз данных сайтов `wordpress.example.com` и `drupal.example.com`
- Каждый дамп хранить отдельным архивом со сжатием `.tar.gz`
- Хранить дампы в каталоге `/backup/mysql/`, подкаталогахах формата `DD-MM-YYYY`.
  Пример итогового пути для хранения бэкапа за день: `/backup/mysql/20-12-2021/`
- Хранить дампы 1 месяц (в любой день должно быть доступно минимум 30 архивных копий)
- Скрипт должен быть исполняемым файлом (для всех) и находиться по пути `/root/bin/mysql-backup.sh`
- Скрипт должен выполняться в `06:45` каждый день как задача в `crontab` пользователя `root`

Настроить инкрементное резервное копирование без шифрования:

- Использовать скрипт https://github.com/zertrin/duplicity-backup.sh
- Сам скрипт должен быть исполняемым файлом (для всех)
  и находиться по пути `/root/bin/duplicity-backup.sh`,
  а его конфиг файл - по пути `/root/bin/duplicity-backup.conf`
- Бэкапы должны запускаться автоматически раз в сутки в `09:00` как задача в `crontab` пользователя `root`
- Для запуска скрипта использовать только один конфигурационный файл и одну задачу в планировщике
- Настроить скрипт создания бэкапов следующим образом:
  - создавать копии файлов сайтов `wordpress.example.com` и `drupal.example.com`, `home` директорий и директории `/etc`
  - исключить из бэкапа файл `/etc/passwd`
  - файлы сохранять через локальный `ftp` сервер, в каталог `/backup/duplicity`
  - полное резервное копирование должно выполняться каждые 7 дней, в остальные дни - инкрементное
  - всегда должно храниться минимум 2 полных копии (`remove-all-but-n-full`)

- Размер тома должен быть `100MB`

Результаты:

Настроенные бэкапы, которые запускаются автоматически.