1. Linux

- описание дистрибутивов:

<http://linuxgeeks.ru/firstime.htm>

- начальная установка

<http://linuxgeeks.ru/disk.htm>

- работа с менеджером пакетов aptitude, dpkg

<https://debian-handbook.info/browse/ru-RU/stable/sect.manipulating-packages-with-dpkg.html>

dpkg (deb package) — инструмент(менеджер пакетов) для установки, уже доступных в пакетов Debian.

**Команда для установки**: dpkg -i название\_пакета Или dpkg --install название\_пакета

**Команда для удаления**: dpkg -r название\_пакета Или dpkg - -remove название\_пакета. Такое удаление позволяет снова установить программу с той же конфигурацией. Для полной деинстанляции используется dpkg -P или dpkg - -purge.

Если при установки какого-либо файла произойдет ошибка, ее можно обработать или исправить с помощью команды dpkg - -force-\*, где вместо \* используеся определенныа команда(список доступных команд можно узнать с помощью - - force-help).

Запись команд с помощью дефис+одна буква или два дефиса + слово/несколько слов принято по соглашению POSIX.

**Команды для работы с базой данных:**

dpkg - -listfiles пакет Или dpkg -L пакет — выводит список файлов, которые хранятся в этом пакете.

Dpkg - -search файл Или dpkg -S файл — ищет в каком пакете содержится данный файл

**dpkg** сохраняет журнал всех своих действий в **/var/log/dpkg.log**. Содержит информацию когда каждый пакет был установлен или обновлен.

<https://wiki.debian.org/ru/Aptitude>

Aptitude — менеджер пакетов в debian. Для запуска утилиты просто в консоли набрать aptitude.

<https://wiki.debian.org/ru/AptPreferences>

(справочник): <https://debian-handbook.info/browse/ru-RU/stable/apt.html>

(справочник): <https://www.debian.org/doc/obsolete.ru.html>

Опубликовано: 16 апр. 2015 г.

Статья у меня на сайте - http://it-inside.org/2015/04/video-v2...

Моя группа в ВК - http://vk.com/itinside- GNU core utils

<https://ru.wikipedia.org/wiki/GNU_Coreutils>

GNU core utils – пакет программного обеспечения GNU(операционки GNU).

Содержит в себе множество команд для работы с файлами(mkdir, ls, rm, cp, dd), текстами(base64, cat, comm), shell-утилиты(chroot, date, du, printf)

<https://github.com/uran1980/web-dev-blog/blob/master/Linux/linux-commands.md>

(справочник): <https://www.gnu.org/software/coreutils/manual/coreutils.pdf> (все читать не нужно, но можно активно обращаться как к справочнику в процессе разработки скриптов). Что нужно знать: cat, head, wc, sort, uniq, ls, cp, dd, mv, rm, mkdir, chown, chmod, touch, du, df, echo, tee, pwd, date, hostname

**pwd** – вывести рабочий каталог

**cd** – сменить рабочий каталог.

**cat** – выводит содержимое на стандартное устройство вывода.

Вариации:

cat название\_файла – выводит содержимое файла

cat > название\_файла некоторый\_текст – записывает некоторый\_текст в файл. Если в файле существовали данные, то они будут перезаписаны.

Cat >> название\_файла некоторый\_текст – дописывает некоторый\_текст в файл.

Выход из команды Ctrl-D

**head** -2 название\_файла – вывести первые две строки файла.

**tail** -2 название\_файла – вывести последние две строки файла.

**wc** - подсчитает количество строк, слов, байт, или символов в текстовом файле.

**sort** – сортирует файл в алфавитном порядке

Вариации:

sort f1 – отсортировать в алфавитном

sort -r f1 – отсортировать в обратном алфавиту

sort -nrk 2 f1 – сортирует числа(параметр -n) в обратном порядке(r) по столбцу(k 2) файл f1.

Sort f1 f2 | uniq – выводит содержимое двух файлов в алфовитном порядке, исключая повторяющиееся строки

**ls** – отображает список файлов и каталогов в текущей директории.

|  |  |
| --- | --- |
| **-i** | Выдавать в первой колонке номера описателей файлов. |
| **-s** | Выдавать размер файлов в блоках (включая косвенные блоки). |

**cp** – копирует файлы.

**-i -** Спрашивает, нужно ли перезаписывать существующие файлы, в которые происходит копирование. (Запрос выдается на стандартный вывод ошибок, а ответ читается со стандартного ввода. Копирование осуществляется только в случае положительного ответа).

**dd** – команда, которая копирует что-то куда-то побайтово.

**mv** – перемещение(переимнование) файлов.

Если последний аргумент является именем существующего  
каталога, то mv перемещает все остальные файлы в этот  
каталог. В противном случае, если задано только два файла,  
имя первого файла будет изменено на имя второго.

**rm –** удаление файлов.

|  |  |
| --- | --- |
| -f | Команда не выдает сообщений, когда удаляемый файл не существует, не запрашивает подтверждения при удалении файлов, на запись в которые нет прав. Если нет права и на запись в каталог, файлы не удаляются. Сообщение об ошибке выдается лишь при попытке удалить каталог, на запись в который нет прав (см. опцию -r). |
| -r | Происходит рекурсивное удаление всех каталогов и подкаталогов, перечисленных в списке аргументов. Сначала каталоги опустошаются, затем удаляются. Подтверждение при удалении файлов, на запись в которые нет прав, не запрашивается, если задана опция -f или стандартный ввод не назначен на терминал и не задана опция -i. При удалении непустых каталогов команда rm -r предпочтительнее команды [rmdir](http://rus-linux.net/MyLDP/consol/hdrguide/rusman/rmdir.htm), так как последняя способна удалить только пустой каталог. Но команда rm -r может доставить немало острых впечатлений при ошибочном указании каталога! |
| -i | Перед удалением каждого файла запрашивается подтверждение. Опция -i устраняет действие опции -f; она действует даже тогда, когда стандартный ввод не назначен на терминал. |

rmdir – удаление пустых каталогов.

chown – изменить владельца файла, группу

chmod – задает режим доступа к файлу, папке (при создании каталога устанавливается режим доступа 0777).

Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец Группа Прочие

(u) (g) (o)

rwx rwx rwx

Для задания режима доступа в символьном виде используется следующий синтаксис:

[кому] операция права

Часть [кому] есть комбинация букв u, g и o (владелец, члены группы и прочие пользователи соответственно). Если часть кому опущена или указано a, то это эквивалентно ugo.

Операция может быть: + (добавить право), - (лишить права), = (в пределах данного класса присвоить права абсолютно, то есть добавить указанные права и отнять неуказанные).

Права - любая осмысленная комбинация следующих букв:

|  |  |
| --- | --- |
| r | Право на чтение. |
| w | Право на запись. |
| x | Право на выполнение (поиск в каталоге). |
| s | При выполнении переустанавливать действующий идентификатор пользователя или группы. |
| t | После выполнения программы сохранять сегмент команд (бит навязчивости). |
| l | Учет блокировки доступа. |

Опустить часть права можно только если операция есть = (для лишения всех прав).

Если надо сделать более одного указания об изменении прав, то при использовании символьного вида в правах не должно быть пробелов, а указания должны разделяться запятыми. Например, команда

chmod u+w,go+x f1

добавит для владельца право писать в файл f1, а для членов группы и прочих пользователей - право выполнять файл. Права устанавливаются в указанном порядке. Право s можно добавлять только для пользователя и группы, право t - только для пользователя.

**touch** – команда для изменения даты последнего доступа к файлу или изменения этого файла. Т.е. изменяет дату последнего использования файла.

**touch** [-acfhm] [-r файл] [-t [[CC]YY]MMDDhhmm.[SS]] файл  
  **-a**  
  Устанавливает время последнего доступа к файлу. Время последнего изменения не устанавливается, если явно не задан ключ -m  
  **-c**  
  Указывает утилите не создавать файл, если он не существует, при этом никаких сообщений об ошибке показано не будет.  
  **-f**  
  Пытается обновить информацию о времени, даже если права доступа файла не позволяют делать.  
  **-h**  
  Указывает утилите не изменять данные о файле, если он задан [символической ссылкой](https://ru.wikipedia.org/wiki/Символическая_ссылка).  
  **-m**  
  Устанавливает время последнего изменения файла.  
  **-r** *file*  
  Использовать значения времени из файла, заданного аргументом *file*.  
  **-t** *time*  
  Устанавливает время последнего изменения и доступа в соответствии с указанным форматом *time*.

**du** -

**uptime** – команда показывает, сколько времени запущена система. Может быть полезна для программирования задач, для которых нужен запуск в определенное время.

**сlear** – очистить консоль**cat**

**ln** – создает ссылку на файл. Два типо ссылок:  
а) hardlink – создается полная ссылка на файл.   
touch test.txt #Создадим файл

ln test.txt new-test.txt #Создаеи хардлинк

cat test.txt 123456 #Запишем что-нибудь в начальный файл

cat new-test.txt #Проверки что в ссылке содержится вся информация, что и в исходном файле.

Также при записи в файл-ссылку, будет производиться запись в один и другой файл.

При удалении начального файла, информация в ссылке сохранится.

б) softlink – создается софт ссылка на файл.

touch test.txt #Создадим файл

ln -s test.txt sl-test.txt #Создаем софтлинк

cat test.txt 123456 #Запишем что-нибудь в начальный файл

cat sl-test.txt #Проверки что в ссылке содержится вся информация, что и в исходном файле.

Также при записи в файл-ссылку, будет производиться запись в один и другой файл.

rm test.txt #удалим начальный файл.

сat sl-test.txt #Ошибка, не существует файла.

Вся фишка софтлинка состоит, что он весит 8 байт, всегда. Просто ссылается на исходный файл. При удалении начального файла смысл ссылки теряется.

Но каждый новый хардлинк не занимает опять же еще столько же места на жестком диске, это просто как другое имя одного и того же файла. НО: хардлинки могут существовать в пределах одного раздела диска.

Наделение правами доступа

Права доступа:

1 (001) – x (execute)

2 (010) – w (write)

4 (100) – r (read)

7 – rwx – доступ на все

По умолчанию: для файла : 666

для папки : 777

umask – изначально(0002). Показывает какую маску изначально отнимают от создаваемого файла.

Т.е. если маску не изменять, то файлы создаются с 664 доступом(6 – доступ владельца, 6 – доступ группы, 4 – доступ прочих) - -rw-rw-r-- .

umask 0026 #изменяем маску

Тогда файлы будут созадаваться с доступом 640 (-rw-r-----).

Также права доступа можно менять с помощью утилиты chmod.

touch test.txt #создаем файл

chmod 777 test.txt #меняем доступ на 777(-rwxrwxrwx)

**nano** – улитита для более удобного написания кода.

**Чтобы выполнить скрипт: bash название\_скрипты позиционные параметры**

Позиционные параметры – это аргументы, которые мы передаем изначально в программу.

Чтобы вызвать позиционные параметры пишут $1 – первый параметр, $2 $3 – второй, третий параметр.

$0 – узнать имя скрипта, который выполняется.

$# - узнать сколько позиционных параметров было передано.

$\* - выведет все параметры, которые были переданы(т.е. все, что было написано после имени скрипта)

$? - показывает результат(код ошибки) последней совершенной операции. Если все хорошо, то код 0, если неправильно 1.

**set** – можно делать позиционные параметры.

set 10 20 140

echo “$1” #10

echo “$3” #140

**expr –** утилита для вычисления целочисленных значениий.

A=10

b=20

c=`expr $a + $b` #обязательны пробелы между переменными

d=`expr $a - $b` #обязательно обратные кавычки

e=`expr $a \\* $b`#при нецелочисленном делении выводится только целая

f=`expr $a / $b` #часть, т.е. откидывается остаток. (28/10 = 2)

**bc** - для работы с вещественными числами.

c=`echo "scale = 2 ; $a + $b" | bc`#scale – количество знаков после # запятой(первый параметр), второй параметр просто любое выражение.

сравнение целых чисел

-eq

равно

if [ "$a" -eq "$b" ]

-ne

не равно

if [ "$a" -ne "$b" ]

-gt

больше

if [ "$a" -gt "$b" ]

-ge

больше или равно

if [ "$a" -ge "$b" ]

-lt

меньше

if [ "$a" -lt "$b" ]

-le

меньше или равно

if [ "$a" -le "$b" ]

<

Сравнение строк:

str1=”Hello”

str2=”World”

[ “$str1” = “$str2” ]

echo $? #1-false

str2=”Hello”

[ “$str1” = “$str2” ]

echo $? #0-true

Логический опреатор **AND (-a)**

**[ “$login” = “log” -a “$pass” = “123” ]**

По такому же принципу работает оператор OR(-o).

Цикл **while.**

count=1

while [ $count -le 10 ]

do

echo $count

count=$(($count+1))

done

exit 0

- bash

<http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_01.html>

<http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_07.html>

<http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_09.html>

<http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_10.html>

<http://tldp.org/LDP/Bash-Beginners-Guide/html/chap_11.html>

sed + awk - можно изучать из этих статей или на примерах в Интернете.

**sed** — прямое назначение — поиск и замена. Использует шаблоны соответствия для замены или выполнения каких-то действий с текстовым файлом (статья - http://rus-linux.net/MyLDP/consol/sed.html).

Например, чтобы заменить всех Nick на John в файле report.txt

cat report.txt | sed 's/Nick/John/g' > report\_new.txt

**awk —** тоже утилита для шаблонной работы(статья - <http://linuxgeeks.ru/awk.htm>) Еще лучше статья - <https://www.ibm.com/developerworks/ru/library/l-awk1/>

С помощью awk можно находить именно тот столбец(столбцы отделяются по разделителю после -F в кавычах).

awk -F":" '{ print $3 }' myTestFile.txt – разделителей является **:**

<https://github.com/jlevy/the-art-of-command-line/blob/master/README-ru.md>

- curl или wget

GNU Wget – утилита для загрузки сайтов из интернs The Lightета, поддерживает дозагрузку, а также имеет механизм загрузки сайтов при плохом интернет соединении. Статья - <http://www.opennet.ru/man.shtml?topic=wget&category=1>

Скажем, вы хотите выгрузить URL. Достаточно набрать:

wget http://fly.cc.fer.hr/

<http://www.opennet.ru/docs/RUS/wget_guide/index.html>

<http://rus-linux.net/MyLDP/internet/curlrus.html>

Curl — инструмент для работы в интернете, отсылки, приема данных, а также для создания запросов.

- network utilities

[http://profhelp.com.ua/articles/linux-%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D1%8B-%D0%B4%D0%BB%D1%8F-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8-%D1%81%D0%B5%D1%82%D0%B8-%D0%B8%D0%B7-%D0%BA%D0%BE%D0%BD%D1%81%D0%BE%D0%BB%D0%B8](http://profhelp.com.ua/articles/linux-команды-для-настройки-сети-из-консоли)

Задание

1.1. В результате выполнения всех скриптов из 1.1.х на выходе должно быть обычное число с единицей измерения (напр.: 50Mb, 34°C, 984471 и т.д.), либо несколько таких чисел через пробел. Строка должна заканчиваться символом переноса строки.

1.1.1. Точное количество выполняемых процессов в системе (доступных под текущим пользователем)

ps - выдача информации о состоянии процессов (ссылка - <http://rus-linux.net/MyLDP/consol/hdrguide/rusman/ps.htm>)

ps -e | sed -e '1d' | wc -l #чисто количество процессов, sed-накладывает шаблон вывода, wc - print the #number of bytes, words, and lines in files 

ps -aux #вывести список всех процессов — сложно

ps -e #список всех процессов

1.1.2. Загрузка CPU в %

1.1.3. Объем доступной оперативной памяти (в байтах или МБ)

sed -n '/MemFree/p' /proc/meminfo

{p;} - найти в файле и вывести эту строку

{n;p;} - находит в файле и выводит строку после соответсвия

free #показывает оперативную память + swap

1.1.4. Количество соединений по произвольному порту (например, 80, 22, 443). Порт задается параметром запуска.

netstat -ant| awk '{print $4}'|grep '.80'|wc -l

1.1.5. Температура на сенсорах (CPU, MB) (статья - <https://romantelychko.com/blog/1096/>)

**sudo** **apt-get install** lm-sensors #установка

**sudo** sensors-detect #считывание со всех устройств

| - передает результат команды с левой команда от черты на правую.

sensors | cat > my.txt

sed -n '/Core 0/{p;}' my.txt

ИЛИ sensors | grep 'Core 0'

1.1.6. Общий суммарный объем всех дисковых устройств, установленных в текущий момент на компьютере (байты, КБ или Гб).

ПРОБЛЕМА С WHILE. После выхода из цикла пропадает значение

Команда df.

df | awk ' {print $2} ' | cat > my.txt #записываем в файл инф-цию только о #полном размере памяти

sed -i '/1K/{d;}' my.txt #отрезаем первую строку, с заголовком

sum=0

cat > /etc/file | #не работает , sum после цикла обнуляется.

(cat my.txt | while read line;

do

sum=2

echo "$sum"

done < <(cat /etc/fstab)

echo "$sum"

)

Последовательное чтение файла:  
sum=0

while read line

do

sum=`expr $sum + $line`

done < my.txt

echo $sum

**grep** — команда вырезает нужные строки из файла (ссылка - http://www.electronick.org.ua/articles/linux/kak-ispolzovat-komandu-grep-v-linux/).

1.1.7. Общий суммарный объем всех устройств, смонтированных в текущий момент в системе (байты, КБ или Гб).

1.1.8. Размер всех баз в MySQL, к которым у текущего пользователя есть доступ.

1.1.9. Размер произвольной базы данных.

1.1.10. Количество запросов в текущий момент времени + максимальное время выполнения запроса из этого списка.

1.2.1. Написать скрипт, который рандомно сгенерирует на диске структуру файлов и папок. Входным параметром будут: директория, в которой нужно генерировать данные, глубину вложенности поддиректорий, максимальный размер файлов, которые будут создаваться в этих директориях и максимальное количество итераций по созданию объектов в директории (т.е. общее количество файлов и директорий, которые будут созданы). Максимальная длина имени - 8 символов, содержимое файлов - рандом или нули. Проверку на доступное место делать не нужно.

filesize=$(stat -c%s "my.sh") # узнать размер файла

direction=$1

nestingDepth=$2

maxSize=$3

numberOfElements=$4

maxNameSize=8

numberOfDirectories=`expr $numberOfElements / 2`

numberOfFiles=`expr $numberOfElements - $numberOfDirectories`

if (( "$numberOfElements" < "$nestingDepth" )); then

echo "Error. Nesting depth can't be less than number of elements"

else

i=0

while (( "$i" < "nestingDepth" ))

do

cd $direction

direction=$RANDOM

mkdir $direction

i=$(( $i+1 ))

done

numberOfDirectories=`expr $numberOfDirectories - $nestingDepth`

i=0

while (( "$i" <= "$numberOfDirectories" ))

do

newDir=$RANDOM

mkdir $newDir

newFile=$RANDOM

randomMessage=$RANDOM

cd $newDir

echo "$randomMessage" | cat > $newFile

numberOfFiles=$(( $numberOfFiles-1 ))

filesize=$(stat -c%s "$newFile")

while (( "$filesize" < (( "$maxSize" - 5 )) ))

do

randomMessage=$RANDOM

echo "$randomMessage" | cat >> $newFile

filesize=$(stat -c%s "$newFile")

done

cd ..

i=$(( $i+1 ))

done

i=0

while (( "$i" <= "$numberOfFiles" )) #дозаписываем оставшиеся файлы

do

newFile=$RANDOM

randomMessage=$RANDOM

echo "$randomMessage" | cat > $newFile

filesize=$(stat -c%s "$newFile")

while (( "$filesize" < (( "$maxSize" - 5 )) ))

do

randomMessage=$RANDOM

echo "$randomMessage" | cat >> $newFile

filesize=$(stat -c%s "$newFile")

done

i=$(( $i+1 ))

done

fi

1.2.2. Написать скрипт, который принимает 2 входных параметра: source\_dir и destination\_dir - две директории на диске, сравнивает, что они ни одна из них не является родительской для другой, у них не совпадают имена и т.п. Далее скрипт должен подсчитать место, занмиаемое source\_dir, и место, доступное на диске, где находится директория destination\_dir, если есть доступное место - скопировать (с наследованием всех атрибутов: времени создания, доступа и модификации, владельца и т.п.), если места нет - вывести предупреждение на экран и предложить пользователю продолжить (C или Y) или прервать (N или A).

stat -c%s 'filename' #вывести размер файла(только файла, не папки)

du -sh Scripts/ | awk -F"K" '{ print $1 }' #размер папки Scripts

sourcedir=$1

destination\_dir=$2

if [ "$sourcedir" = "$destination\_dir" ] #проверка на одинаковые имена директорий

then

echo "Error. The source folder and destination folder have similar names."

exit

fi

if ! [ -d "$sourcedir" ]; #проверка на существование директории $sourcedir

then

echo "Directory \"$sourcedir\" isn't exist"

exit

fi

if ! [ -d "$destination\_dir" ]; #проверка на существование директории $destination\_dir

then

echo "Directory \"$destination\_dir\" isn't exist"

exit

fi

basedestinationdir=$(basename $destination\_dir) #отрезаем только имя папки без путя

basesourcedir=$(basename $sourcedir)

cd $sourcedir

ls -l | grep $basedestinationdir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$destination\_dir\" find in \"$sourcedir\""

exit

fi

cd $destination\_dir

ls -l | grep $basesourcedir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$sourcedir\" find in \"$destination\_dir\""

exit

fi

sizeOfSourcedir=$(du -sh "$sourcedir" | awk -F"K" '{ print $1 }') #вычисляем размер sourcedir папки

freeSpace=$(df / | awk 'NR == 2{ print $3 }') #узнаем свободное место на диске

if [ "$freeSpace" > "$sizeOfSourcedir" ]

then

cp -Rp "$sourcedir" "$destination\_dir"

echo "Succesful copying"

else

echo "Not enough memory for copying"

echo "Continue (Y\N)"

read line

if ( "$line" = "Y" )

then

cp -Rp $sourcedir $destination\_dir

else

exit

fi

fi

cd ../

1.2.3. Модернизировать скрипт, чтобы он упаковывал данные из source\_dir и складывал в destination\_dir. При этом в самом начале работы скрипта нужно предусмотреть диалог с пользователем: предложить дописывать текущую дату и время в формате YYMMDD\_HHNN.gz к имени файла или выполнять ротацию архивов, т.е. дописывал порядковый номер архива (0.gz, 1.gz, 2.gz). В случае выбора второго варианта - так же нужно запросить максимальное количество копий, при этом номер 0 присваивается самому последнему созданному архиву, а номер N-1 - самому старому, на каждой итерации архив с номером N — удаляется.

tar czf example.tar # архивируем в tar

tar -xvvzf foo.tar.gz # архивируем в gz

sourcedir=$1

destination\_dir=$2

maxCountOfArchive=$3

if [ "$sourcedir" = "$destination\_dir" ] #проверка на одинаковые имена директорий

then

echo "Error. The source folder and destination folder have similar names."

exit

fi

if ! [ -d "$sourcedir" ]; #проверка на существование директории $sourcedir

then

echo "Directory \"$sourcedir\" isn't exist"

exit

fi

if ! [ -d "$destination\_dir" ]; #проверка на существование директории $destination\_dir

then

echo "Directory \"$destination\_dir\" isn't exist"

exit

fi

basedestinationdir=$(basename $destination\_dir) #отрезаем только имя папки без путя

basesourcedir=$(basename $sourcedir)

cd $sourcedir

ls -l | grep $basedestinationdir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$destination\_dir\" find in \"$sourcedir\""

exit

fi

cd $destination\_dir

ls -l | grep $basesourcedir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$sourcedir\" find in \"$destination\_dir\""

exit

fi

sizeOfSourcedir=$(du -sh "$sourcedir" | awk -F"K" '{ print $1 }') #вычисляем размер sourcedir папки

freeSpace=$(df / | awk 'NR == 2{ print $3 }') #узнаем свободное место на диске

if [ "$freeSpace" > "$sizeOfSourcedir" ] #проверка на свободное место под копирование

then

echo -e "1. Create archive from $sourcedir, as YYMMDD\_HHNN.tar\\n2. Create new archive from $sourcedir\\nEnter (1/2)"

read line

if [ "$line" == "1" ]

then

DATE=`date +%Y%m%d\_%H%M`

cd $sourcedir

cd ..

baseSourceDir=$(basename $sourcedir)

tar czf $DATE.tar $baseSourceDir

chmod 666 $DATE.tar #меняем параметры доступа

mv $DATE.tar $destination\_dir

fi

if [ "$line" == "2" ]

then

cd $destination\_dir

sudo ls | grep .tar | awk -F"." '{ print $1 }' | cat > temp.txt

cat temp.txt

i=0

while read line #записываем название(номера) архивов в массив

do

if [[ $line =~ ^-?[0-9]+$ ]] #проверка, является ли переменная числом

then

array[$i]=$line

i=$(( $i+1 ))

fi

done < temp.txt

i=$(( $i-1 ))

while [ "$i" -ge "0" ] #переменовываем архивы+1

do

namePlus1=`expr ${array[$i]} + 1`

echo "namePlus1=$namePlus1"

mv ${array[$i]}.tar $namePlus1.tar

echo "move ${array[$i]}"

i=$(( $i-1 ))

done

ls | grep .tar | awk -F"." '{ print $1 }' | cat > temp.txt

while read line #удаляем архивы, которые больше или равны $maxCountOfArchive

do

if [[ $line =~ ^-?[0-9]+$ ]] #проверка, является ли переменная числом

then

if [ "$line" -ge "$maxCountOfArchive" ]

then

rm -r $line.tar

fi

fi

done < temp.txt

cd $sourcedir

cd ..

baseSourceDir=$(basename $sourcedir)

tar czf 0.tar $baseSourceDir

chmod 666 0.tar #меняем параметры доступа

mv 0.tar $destination\_dir

fi

else

echo "Not enough memory for copying"

echo "Continue (Y\N)"

read line

if ( "$line" = "Y" )

then

echo ""

else

exit

fi

fi

cd ../

1.2.4. Модернизировать скрипт, чтобы все сообщения об ошибках подавлялись, а стандартные - выводились на консоль + записывались в лог-файл в текущей директории с именем out\_YYMMDD\_HHNN.log. Т.к. сообщения об ошибках подавляются, то в случае возникновения ошибок в процессе рабоыт скрипта (проблемы с пермишенами, нехватка места, отсутствие какой-либо утилиты и т.п.) необходимо в конце работы скрипта вывести сообщение в консоль красным цветом "Warning: X error(s) occured!", где X - количество перехваченных ошибок или количество вызовов команд, которые повлекли ошибки (как удобнее, на выбор).

sourcedir=$1

destination\_dir=$2

maxCountOfArchive=$3

counterOfWarning=0

homeFolder=$( pwd )

touch $homeFolder/out\_$DateOfLogFile.log #создаем лог-файл ошибок

DateOfLogFile=`date +%Y%m%d\_%H%M`

if [ "$sourcedir" = "$destination\_dir" ] #проверка на одинаковые имена директорий

then

echo "Error. The source folder and destination folder have similar names." | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

if ! [ -d "$sourcedir" ]; #проверка на существование директории $sourcedir

then

echo "Directory \"$sourcedir\" isn't exist" | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

if ! [ -d "$destination\_dir" ]; #проверка на существование директории $destination\_dir

then

echo "Directory \"$destination\_dir\" isn't exist" | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

basedestinationdir=$(basename $destination\_dir) #отрезаем только имя папки без путя

basesourcedir=$(basename $sourcedir)

cd $sourcedir

ls -l | grep $basedestinationdir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$destination\_dir\" find in \"$sourcedir\"" | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

cd $destination\_dir

ls -l | grep $basesourcedir

operationResult=$? # записали результат операции, является ли папка родительской

if [ "$operationResult" = "0" ]

then

echo "Error. Folder \"$sourcedir\" find in \"$destination\_dir\"" | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

sizeOfSourcedir=$(du -sh "$sourcedir" | awk -F"K" '{ print $1 }') #вычисляем размер sourcedir папки

freeSpace=$(df / | awk 'NR == 2{ print $3 }') #узнаем свободное место на диске

if [ "$freeSpace" > "$sizeOfSourcedir" ] #проверка на свободное место под копирование

then

echo -e "1. Create archive from $sourcedir, as YYMMDD\_HHNN.tar\\n2. Create new archive from $sourcedir\\nEnter (1/2)"

read line

if [ "$line" == "1" ]

then

DATE=`date +%Y%m%d\_%H%M`

cd $sourcedir

cd ..

baseSourceDir=$(basename $sourcedir)

tar czf $DATE.tar $baseSourceDir

chmod 666 $DATE.tar #меняем параметры доступа

mv $DATE.tar $destination\_dir

fi

if [ "$line" == "2" ]

then

cd $destination\_dir

sudo ls | grep .tar | awk -F"." '{ print $1 }' | cat > temp.txt

cat temp.txt

i=0

while read line #записываем название(номера) архивов в массив

do

if [[ $line =~ ^-?[0-9]+$ ]] #проверка, является ли переменная числом

then

array[$i]=$line

i=$(( $i+1 ))

fi

done < temp.txt

i=$(( $i-1 ))

while [ "$i" -ge "0" ] #переменовываем архивы+1

do

namePlus1=`expr ${array[$i]} + 1`

echo "namePlus1=$namePlus1"

mv ${array[$i]}.tar $namePlus1.tar

echo "move ${array[$i]}"

i=$(( $i-1 ))

done

ls | grep .tar | awk -F"." '{ print $1 }' | cat > temp.txt

while read line #удаляем архивы, которые больше или равны $maxCountOfArchive

do

if [[ $line =~ ^-?[0-9]+$ ]] #проверка, является ли переменная числом

then

if [ "$line" -ge "$maxCountOfArchive" ]

then

rm -r $line.tar

fi

fi

done < temp.txt

cd $sourcedir

cd ..

baseSourceDir=$(basename $sourcedir)

tar czf 0.tar $baseSourceDir

chmod 666 0.tar #меняем параметры доступа

mv 0.tar $destination\_dir

fi

else

echo "Error. Not enough memory or no file for copying" | cat >> $homeFolder/out\_$DateOfLogFile.log

counterOfWarning=$(( $counterOfWarning+1 ))

fi

echo "Warning: $counterOfWarning error(s) occured!"

cd ../

1.3.1. Написать скрипт, который бы выводил на экран наиболее выгодный курс валют конвертации USD->BYR и EUR->BYR. Способ работы скрипта: либо передавать входным параметром "USD" или "EUR" и выводить курс, либо, если входных параметров нет - а консольном дилоге запросить на выбор желаемый курс. Получать можно с любого сайта (в идеале - с finance.tut.by).

wget http://finance.tut.by/kurs/minsk/dollar/vse-banki/?sortBy=sell&sortDir=up #выкачиваем сайт

nameCurrancy=$1

if [ $nameCurrancy = "USD" ]

then

echo "Downloading statistic from finance.tut.by..."

wget -q http://finance.tut.by/kurs/minsk/dollar/vse-banki/?sortBy=sell&sortDir=up #выкачиваем сайт

wget -q http://finance.tut.by/kurs/minsk/dollar/vse-banki/?sortBy=buy\&sortDir=down

clear

echo -e "USD - BYR\n"

grep '<b class="red"' index.html\?sortBy\=sell | cat > myTestFile.txt

awk -F"<" '{ print $3 }' myTestFile.txt | cat > myFile.txt #вырезаем числа из строки

awk -F">" '{ print $2 }' myFile.txt | cat > myTestFile.txt #отрезаем все лишнее

i=0

currWeight=0

buyWeight=0

sellWeight=0

while read line

do

l=${line// /} #удаляем лишний пробел

currWeight=$l

break

done < myTestFile.txt

buyWeight=$currWeight

rm index.html\?sortBy\=sell

rm myFile.txt myTestFile.txt

grep '<b class="red"' index.html?sortBy=buy\&sortDir=down | cat > myTestFile.txt

awk -F"<" '{ print $3 }' myTestFile.txt | cat > myFile.txt #вырезаем числа из строки

awk -F">" '{ print $2 }' myFile.txt | cat > myTestFile.txt #отрезаем все лишнее

while read line

do

l=${line// /} #удаляем лишний пробел

currWeight=$l

break

done < myTestFile.txt

sellWeight=$currWeight

echo "BUY=$buyWeight"

echo "SELL=$sellWeight"

rm index.html?sortBy=buy\&sortDir=down

rm myFile.txt myTestFile.txt

elif [ $nameCurrancy = "EUR" ]

then

echo "Downloading statistic from finance.tut.by..."

wget -q http://finance.tut.by/kurs/minsk/euro/vse-banki/?sortBy=sell&sortDir=up #выкачиваем сайт

wget -q http://finance.tut.by/kurs/minsk/euro/vse-banki/?sortBy=buy\&sortDir=down

clear

echo -e "EUR - BYR\n"

grep '<b class="red"' index.html\?sortBy\=sell | cat > myTestFile.txt

awk -F"<" '{ print $3 }' myTestFile.txt | cat > myFile.txt #вырезаем числа из строки

awk -F">" '{ print $2 }' myFile.txt | cat > myTestFile.txt #отрезаем все лишнее

i=0

currWeight=0

buyWeight=0

sellWeight=0

while read line

do

l=${line// /} #удаляем лишний пробел

currWeight=$l

break

done < myTestFile.txt

buyWeight=$currWeight

rm index.html\?sortBy\=sell

rm myFile.txt myTestFile.txt

grep '<b class="red"' index.html?sortBy=buy\&sortDir=down | cat > myTestFile.txt

awk -F"<" '{ print $3 }' myTestFile.txt | cat > myFile.txt #вырезаем числа из строки

awk -F">" '{ print $2 }' myFile.txt | cat > myTestFile.txt #отрезаем все лишнее

while read line

do

l=${line// /} #удаляем лишний пробел

currWeight=$l

break

done < myTestFile.txt

sellWeight=$currWeight

echo "BUY=$buyWeight"

echo "SELL=$sellWeight"

rm index.html?sortBy=buy\&sortDir=down

rm myFile.txt myTestFile.txt

fi

1.3.2. Написать скрипт, который мог бы пройти авторизацию на Яндексе, сохранить сессию в cookie-файл, и показать кол-во писем в папке Inbox.

<https://www.youtube.com/watch?v=Im72HDtqkFs>

wget --cookies=on --save-cookies cookies.txt --keep-session-cookies --post-data="login=treshaalesha&passwd=a135792468A" "<https://passport.yandex.ru/passport?mode=auth&retpath=https://mail.yandex.by>"

curl -d "login=treshaalesha&passwd=a135792468A&submit=Submit" --location --output /home/alexey/myYandexEnter.html "https://passport.yandex.ru/passport?mode=auth&retpath=https://mail.yandex.by"

1.3.3. Модифицировать скрипт 1.2.4 - и выгрузить файл на яндекс.диск. Использовать лучше webdav и выгружать на webdav.yandex.ru, ротацию делать не нужно, достаточно просто выгружать файлы с именем hostname\_YYMMDD-HHNN (т.е. добавить еще имя хоста). destination\_dir - будет путем на яндекс.диске. Логин, пароль можно хардкодить.

2. Почта/безопасность

(справочник): <https://www.opennet.ru/docs/RUS/iptables/>

- iptables общее

<http://sys.dmitrow.com/node/34>

<http://sys.dmitrow.com/node/35>

<http://sys.dmitrow.com/node/36>

- iptables сохранение после ребута

<http://sys.dmitrow.com/node/42>

- iptables для NAT/PAT

<http://sys.dmitrow.com/node/145>

<http://sys.dmitrow.com/node/38>

<http://sys.dmitrow.com/node/41>

- rkhunter, chkrootkit, fail2ban

просто познакомиться с утилитами, их возможностями

<http://webware.biz/?p=2283>

- nginx/apache

(справочник): <http://nginx.org/ru/docs/>

<https://github.com/uran1980/web-dev-blog/tree/master/Nginx>

- openssl

- почта

<http://interface31.ru/tech_it/2010/09/pochtovyj-server-struktura-i-princip-raboty.html>

[http://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE\_%D0%BF%D0%BE\_ubuntu\_server/%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B5\_%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D1%8B](http://help.ubuntu.ru/wiki/руководство_по_ubuntu_server/почтовые_сервисы) (posfix, dovecot, exim4)

- гарантия доставки почты

<http://interface31.ru/tech_it/2013/10/pochtovyy-server-dlya-nachinayushhih-ptr-i-spf-zapisi-kak-sredstvo-bor-by-so-spamom.html>

<http://geektimes.ru/post/63768/>

<https://support.google.com/a/answer/33786?hl=en&ref_topic=2759192>

<https://support.google.com/a/answer/4568483?hl=en>

inet addr: 192.168.100.9

Задание

* + 1. +Установить apache. Включить 1 сайт "sitename.com", который должен открываться в браузере (например, модифицировать host-файл).

Статья - <https://www.youtube.com/watch?v=WzyO-cTSp9E> (установка хостов)

Перезапуск apache2: sudo /etc/init.d/apache2 restart

Алгоритм настройки:

1) sudo apt-get install tasksel #устанавливаем tasksel

2) sudo tasksel install lamp-server #устанавливаем apache2, php, mysql

3) sudo nano /etc/hosts

Дописываем в хостах после localhost название своего сайта(mylinuxsite mylinuxsite.com)

4) Создаем свою папку и свой сайт в /var/www/html/свой\_каталог

5) Создаем конфигурационный файл в /etc/apache2/sites-available

Например: mylinuxsite.com.conf

Его содержимое:

<VirtualHost \*:80>

DocumentRoot /var/www/html/mylinuxsite.com/

ServerName mylinuxsite.com

ServerAlias www.mylinuxsite.com

<Directory /var/www/html/mylinuxsite.com/>

</Directory>

</VirtualHost>

6) Создаем симлинк в папку /etc/apache2/sites-enabled командой

sudo a2ensite mylinuxsite.com

7) Перезапускаем apache2:

sudo /etc/init.d/apache2 restart

Все!)

* + 1. Установить php версии 5.2 (устаревшая), подключить его для использования в apache. Это задание - пример использования backports и решения конфликтов с устаревшими версиями.
    2. +Установить nginx lighty - как балансер для Apache. Отдавать статические файлы (css|js|txt|jpg) - через nginx/lighty, остальные запросы считать динамическим контентом и отдавать для обработки на apache.

sudo service nginx **start** #запуск

sudo service nginx **stop #остановка**

* + 1. +Сгенерировать SSL-сертификат, установить его в Apache. Индексная страница – php-скрипт с выводом информации о системе.

Видео - https://www.youtube.com/watch?v=X09cT8n2KeE

* + 1. +Обновить 2.1.1 и 2.1.4: добавить второй сайт "website.co.uk" на том же 443 порту. Сгенерировать второй сертификат, который должен быть валидным для обоих сайтов! Данный сертификат нужно установить для двух сайтов (т.е. на первом нужно заменить старый). При этом, при открытии в браузере может быть жалоба на самоподписанный сертификат, но не должно быть ошибок на невалидный домен. Проверка: меняем hosts-файл и открываем в браузере 2 разных сайта.

*Созданые 2 подписанный сайта: 192.168.100.9 и website.co.uk.*

*Алгоритм подписания сайта:*

*1)* sudo a2enmod ssl # подключаем ssl

2) sudo service apache2 restart # перегружаем apache2

3) sudo mkdir /etc/apache2/ssl

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt

#Создаем ключ и сертификат в /etc/apache2/ssl/

4) Вводим свои данные. Важно указать правильное Common Name:

Country Name (2 letter code) [AU]:US

State or Province Name (full name) [Some-State]:New York

Locality Name (eg, city) []:NYC

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Awesome Inc

Organizational Unit Name (eg, section) []:Dept of Merriment

**Common Name (e.g. server FQDN or YOUR name) []:example.com #указываем свое имя сервера**

Email Address []:webmaster@awesomeinc.com

5) Меняем конфиг нашего сайта:

ServerName example.com:443

Дописываем пути в блоке <VirtualHost :\*443></VirtualHost>

SSLEngine on

SSLCertificateFile /etc/apache2/ssl/apache.crt

SSLCertificateKeyFile /etc/apache2/ssl/apache.key

6) Включаем сайт:

sudo a2ensite example.com

7) sudo service apache2 restart # перегружаем apache2

8) В браузере firefox вписываем [https://example.com](https://example.com/) и проходим на свой страх и риск, далее подтверждая сертификат

* 1. 1. +Настроить файрволл: запретить все правила, разрешить 22, 80, 443 порты. Ответ запрета: "icmp-порт не доступен".

**C помощью iptables(статья- http://help.ubuntu.ru/wiki/iptables):**

sudo iptables -L

sudo iptables-save #просмотр текущей конфигурации

**sudo** **nano** /etc/network/if-up.d/iptables-rules #редактрируем файл конфигурации iptables

Вот такую конфигурацию, комментируем ненужное и добавляем свои запреты.

*#!/sbin/iptables-restore*

*# Таблица filter и её цепочки*

\*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

*# Разрешаем связанные и установленые соединения*

-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

*# Разрешаем служебный icmp-трафик*

-A INPUT -p icmp -j ACCEPT

*# Разрешаем доверенный трафик на интерфейс loopback*

-A INPUT -i lo -j ACCEPT

*# Сюда можно вставлять дополнительные правила для цепочки INPUT*

*# Запрещаем всё остальное для INPUT*

-A INPUT -j REJECT --reject-with icmp-host-prohibited

*# Порядок и смысл правил для цепочек FORWARD и OUTPUT аналогичен INPUT*

-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A FORWARD -p icmp -j ACCEPT

-A FORWARD -j REJECT --reject-with icmp-host-prohibited

*# Фильтровать цепочку OUTPUT настоятельно не рекомендуется*

*#-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT*

*#-A OUTPUT -p icmp -j ACCEPT*

*#-A OUTPUT -o lo -j ACCEPT*

*#-A OUTPUT -j REJECT --reject-with icmp-host-prohibited*

COMMIT

Делаем скрипт исполняемым и загружаем правила iptables:

**sudo** **chmod** +x /etc/network/if-up.d/iptables-rules

**sudo** /etc/network/if-up.d/iptables-rules

Чтобы запретить все, кроме 3-ех портов – расскоментировать строку #расскоментировать-P INPUT DROP в /etc/network/if-up.d/iptables-rules

и перегрузить компьютер

Команды iptables:

**iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT #включить 80 порт**

**iptables -I INPUT 1 -p tcp --dport 80 -j DROP # отключить 80 порт**

iptables -F # удалить все правила

* + 1. Обновить конфигурацию из 2.1.5: повесить nginx на порт 127.0.0.1:8443, настроить порт-форвардинг: входящий \*:443 транслировать на 127.0.0.1:8443.

1) Включаем ip-forwarding: sudo sysctl -w net.ipv4.ip\_forward=1

2)

* + 1. Настроить fail2ban, который бы блокировал доступ к nginx по IP на 10 минут, например, от случая сканирования сайта пауками: google, baidu и bing (https://www.incapsula.com/blog/know-your-top-10-bots.html). Проверка: wget --user-agent="Googlebot/2.1 (+http://www.googlebot.com/bot.html)" -r https://sitename.com/
    2. +Установить и настроить MTA для отправки всей почты с сервера localhost. Учетки и пароли хранятся в базе MySQL

MTA — агент пересылки почты (принимает/отправляет) — POSTFIX(по умолчанию). Протокол SMTP

MDA — агент доставки почты (доставляет до получателя)

по протоколу SMTP

Установка Postfix:

1) sudo apt-get install postfix #установить postfix

2) Ок->Internet site → ubuntu1.local – установили домен

3) sudo nano /etc/postfix/main.cf #отредактируем файл конфигурации

Меняем:

myhostname = ubuntu

mydestination = ubuntu.local, server2, localhost.localdomain, localhost

mynetworks = 127.0.0.0 192.168.0.0/24

mailbox\_size\_limit = 0

#relayhost - закомментировать

4) sudo service postfix restart

5) telnet 127.0.0.1 25 # входим на почтовый сервер

Отправим первое сообщение (пример, что-то наподобие такого):

sudo telnet 127.0.0.1 25 # вводим команду  
Trying 127.0.0.1...  
Connected to 127.0.0.1.  
Escape character is '^]'.  
220 ubuntu.local ESMTP Postfix (Ubuntu)  
rcpt to:ubuntu.local  
503 5.5.1 Error: need MAIL command  
mail to: server2@ubuntu.local  
501 5.5.4 Syntax: MAIL FROM:<address>  
mail from: root@ubuntu.local  
250 2.1.0 Ok  
rcpt to: server2@ubuntu.local  
550 5.1.1 <server2@ubuntu.local>: Recipient address rejected: User unknown in local recipient table  
rcpt to: tbm@ubuntu.local  
250 2.1.5 Ok  
data  
354 End data with <CR><LF>.<CR><LF>  
subject: This is my first email  
This is the body of the email  
.  
250 2.0.0 Ok: queued as D7B9940B3F  
quit  
221 2.0.0 Bye  
Connection closed by foreign host.

6) mail #просмотрим список сообщений

Выбираем номер сообщения, которое будем смотреть

Установка dovecot:

1) sudo bash # входим в режим админа

2) apt-get install dovecot-imadp -y dovecot-common

Выбираем сомоподписанный сертификат и после, localhosts

3) nano /etc/dovecot/dovecot.conf #отредактируем файл конфигурации

Раскоментируем:

login\_trusted\_networks = 192.168.0.0

listen = \*

4) service dovecot restart

####################Старая настройка

Установка:

1) sudo apt-get install postfix

2) ОК->интернет сайт

3) Вводим почтовое имя: mail.alex95.com

4) sudo dpkg-reconfigure postfix # более детальная настройка

5) проходим те же пункты. Адрес получателя ставим: alex

6) Вводим: mail.alex95.com, localhost.localdomain, localhost # почта, локальный домен и сам локальный хост

7) НЕТ ( синхронным обновлениям)

8) 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.100.0/24

9) 0 – неограниченный почтовый ящик

10) + символ, который будет использоваться для определения расширения локальных адресов

11) Все ip-ки

Файл конфигурации: /etc/postfix/main.cf

12) sudo postconf -e 'home\_mailbox = Maildir/' # добавляем директории для хранения почты

Аутентификация:

1) Запускаем(статья -http://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE\_%D0%BF%D0%BE\_ubuntu\_server/%D0%BF%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B5\_%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81%D1%8B/postfix):   
sudo postconf -e 'smtpd\_sasl\_type = dovecot'

sudo postconf -e 'smtpd\_sasl\_path = private/auth-client'

sudo postconf -e 'smtpd\_sasl\_local\_domain ='

sudo postconf -e 'smtpd\_sasl\_security\_options = noanonymous'

sudo postconf -e 'broken\_sasl\_auth\_clients = yes'

sudo postconf -e 'smtpd\_sasl\_auth\_enable = yes'

sudo postconf -e 'smtpd\_recipient\_restrictions = \

permit\_sasl\_authenticated,permit\_mynetworks,reject\_unauth\_destination'

2) Создаем самоподписанный сертификат:

sudo openssl req -new -outform PEM -out server.crt -newkey rsa:2048 -nodes -keyout server.key -keyform PEM -days 365 -x509

3) sudo postconf -e 'smtp\_tls\_security\_level = may'

sudo postconf -e 'smtpd\_tls\_security\_level = may'

sudo postconf -e 'smtp\_tls\_note\_starttls\_offer = yes'

sudo postconf -e 'smtpd\_tls\_key\_file = /etc/ssl/private/server.key'

sudo postconf -e 'smtpd\_tls\_cert\_file = /etc/ssl/certs/server.crt'

sudo postconf -e 'smtpd\_tls\_loglevel = 1'

sudo postconf -e 'smtpd\_tls\_received\_header = yes'

sudo postconf -e 'myhostname = mail.alex95.com' #изменяем в зависимости от имени почты

4) sudo /etc/init.d/postfix restart # перезапускаем

Настройка SASL(наш MDA)

1) sudo apt-get install dovecot-common # в качестве MDA выступает dovecot-common

2) Добавляем в конце файла конфигурации /etc/dovecot/dovecot.conf

socket listen {

#master {

# Master socket provides access to userdb information. It's typically

# used to give Dovecot's local delivery agent access to userdb so it

# can find mailbox locations.

#path = /var/run/dovecot/auth-master

#mode = 0600

# Default user/group is the one who started dovecot-auth (root)

#user =

#group =

#}

client {

# The client socket is generally safe to export to everyone. Typical use

# is to export it to your SMTP server so it can do SMTP AUTH lookups

# using it.

path = /var/spool/postfix/private/auth-client

mode = 0660

user = postfix

group = postfix

}

}

3) Добавим типы допустимых вариантов аутентификации  
auth\_mechanisms = plain login digest-md5

4) sudo service dovecot restart # перезапускаем

Установка почтового стека:

1) sudo apt-get install mail-stack-delivery

Настроим конфигурацию mysql(http://www.volmed.org.ru/wiki/index.php/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9\_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80\_%D0%BD%D0%B0\_%D0%B1%D0%B0%D0%B7%D0%B5\_Postfix,\_Dovecot\_%D0%B8\_Ubuntu\_14.04\_LTS):

1) sudo apt-get install postfix dovecot-common dovecot-imapd mysql-server mysql-client postfix-mysql dovecot-mysql sasl2-bin # устанавливаем необходимые пакеты

2) mysql -uroot -proot #входим в mysql

3)Будем использоваться пользователь mail\_admin и его пароль map для доступа к базе данных почтовых аккаунтов в MySQL.

USE mysql;

INSERT INTO user (Host,User,Password) VALUES('%','mail\_admin', PASSWORD('map'));

flush privileges;

4) создаем базу данных, в которой будут храниться учетные записи почтовых аккаунтов

CREATE DATABASE mail;

USE mail;

5) Добавляем права доступа:

GRANT SELECT, INSERT, UPDATE, DELETE ON mail.\* TO 'mail\_admin'@'localhost' IDENTIFIED BY 'map';

GRANT SELECT, INSERT, UPDATE, DELETE ON mail.\* TO 'mail\_admin'@'localhost.localdomain' IDENTIFIED BY 'map';

FLUSH PRIVILEGES;

6) создаем таблицу доменов

CREATE TABLE domains ( domain varchar(50) NOT NULL, PRIMARY KEY (domain) );

7) Создаем таблицу алиасов.

CREATE TABLE forwardings ( source varchar(80) NOT NULL, destination TEXT NOT NULL, PRIMARY KEY (source) );

8) Создаем таблицу для пользователей

CREATE TABLE users ( user varchar(80) NOT NULL, email varchar(80) NOT NULL, password varchar(20) NOT NULL, PRIMARY KEY (email) );

CREATE TABLE transport ( domain varchar(128) NOT NULL default '', transport varchar(128) NOT NULL default '', UNIQUE KEY domain (domain) );

quit

9) Создание пользователей apache2

Создаем группу virtual и пользователя virtual, входящего в эту группу.

sudo groupadd -g 5000 virtual

sudo useradd -g virtual -u 5000 virtual

10) Создание недостающих папок

Создадим каталог, где будут размещаться почтовые ящики:

sudo mkdir /var/spool/mail/mail.alex95.com

11)сменим имя и группу владельца созданной директории на virtual:virtual

http://www.volmed.org.ru/wiki/index.php/%D0%9F%D0%BE%D1%87%D1%82%D0%BE%D0%B2%D1%8B%D0%B9\_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80\_%D0%BD%D0%B0\_%D0%B1%D0%B0%D0%B7%D0%B5\_Postfix,\_Dovecot\_%D0%B8\_Ubuntu\_14.04\_LTS

sudo chown virtual:virtual /var/spool/mail/mail.alex95.com

12) дадим владельцу и его группе полные права на каталог

sudo chmod 777 /var/spool/mail/mail.alex95.com

13) Создаем папку для настроечных файлов обращения к mysql

sudo mkdir /etc/postfix/sql

14) Создаем файлы обращения к базе данных

Создаем конфигурационный файл для доступа к почтовым доменам в базе данных MySQL

sudo nano /etc/postfix/sql/mysql-virtual\_domains.cf

И вставляем:

user = mail\_admin

password = map

dbname = mail

query = SELECT domain AS virtual FROM domains WHERE domain = '%s'

hosts = 127.0.0.1

15) Создаем конфигурационный файл для определения автоматической пересылки писем с ящика на ящик

sudo nano /etc/postfix/sql/mysql-virtual\_forwardings.cf

И добавляем:

user = mail\_admin

password = map

dbname = mail

query = SELECT destination FROM forwardings WHERE source = '%s'

hosts = 127.0.0.1

16) Создаем конфигурационный файл для доступа к почтовым аккаунтам в базе данных MySQL

sudo nano /etc/postfix/sql/mysql-virtual\_mailboxes.cf

user = mail\_admin

password = map

dbname = mail

query = SELECT CONCAT( SUBSTRING\_INDEX(email, '@', -1), '/' , SUBSTRING\_INDEX(email, '@', 1), '/' ) FROM users WHERE email = '%s'

hosts = 127.0.0.1

17) Создаем конфигурационный файл для виртуального отображения почты

sudo nano /etc/postfix/sql/mysql-virtual\_email2email.cf

user = mail\_admin

password = map

dbname = mail

query = SELECT email FROM users WHERE email = '%s'

hosts = 127.0.0.1

18) Создаем конфигурационный файл отправителя.

sudo nano /etc/postfix/sql/mysql\_sender\_login\_maps.cf

user = mail\_admin

password = map

dbname = mail

query = SELECT email FROM users WHERE email = '%s'

hosts = 127.0.0.1

19) Владельцем данных конфигурационных файлов должен быть root, группой владельца - postfix

sudo chown -R root:postfix /etc/postfix/sql

20) Конфигурационный файл главного модуля main.cf. Скопируем на всякий случай в temp.

sudo cp /etc/postfix/main.cf /home/alexey/temp/main.cf.dist

21) sudo cp /etc/postfix/main.cf /etc/postfix/main.cf.dist

Пишем sudo nano /etc/postfix/main.cf туда следующее:

#Так наш сервер будет представляться при отправке и получении почты

smtpd\_banner = $myhostname ESMTP (Ubuntu)

#Отключаем использование comsat

biff = no

#Запрещаем автоматически дополнять неполное доменное имя в адресе письма

append\_dot\_mydomain = no

#Указываем имя нашего хоста

### myhostname = volmed.org.ru

myhostname = mail.alex95.com

# Указываем файл с псевдонимами почтовых ящиков

alias\_maps = hash:/etc/postfix/aliases

#Указываем, для каких доменов будем принимать почту

mydestination = localhost

relayhost=

### mynetworks = 127.0.0.0/8,

### 192.168.0.0/24,

### 192.168.1.0/24

mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.100.0/24

# Не будем ограничивать размер почтовых ящиков

mailbox\_size\_limit = 0

recipient\_delimiter = +

#Указываем прослушивание на всех сетевых интерфейсах

inet\_interfaces = all

#Указываем обязательность использование клиентами команды helo

smtpd\_helo\_required = yes

#Описываем доступ доменам, почтовым ящикам и т.д.

virtual\_mailbox\_domains = proxy:mysql:/etc/postfix/sql/mysql-virtual\_domains.cf

#virtual\_alias\_maps = proxy:mysql:/etc/postfix/sql/mysql-virtual\_forwardings.cf

virtual\_alias\_maps =

proxy:mysql:/etc/postfix/sql/mysql-virtual\_forwardings.cf,

mysql:/etc/postfix/sql/mysql-virtual\_email2email.cf

virtual\_mailbox\_maps = proxy:mysql:/etc/postfix/sql/mysql-virtual\_mailboxes.cf

virtual\_mailbox\_base = /var/spool/mail

virtual\_minimum\_uid = 100

virtual\_uid\_maps = static:5000

virtual\_gid\_maps = static:5000

home\_mailbox = Maildir/

local\_recipient\_maps = $virtual\_mailbox\_maps

#Указываем каталог очереди для Postfix

queue\_directory = /var/spool/postfix

#Описываем авторизацию по SMTP для клиентов не из доверенной зоны

smtpd\_sasl\_type = dovecot

smtpd\_sasl\_auth\_enable = yes

smtpd\_sasl\_path = private/auth

smtpd\_sasl\_security\_options = noanonymous

#broken\_sasl\_auth\_clients = yes

# запрещаем проверку отправителем существование адреса получателя

# на этапе передачи заголовка

disable\_vrfy\_command = yes

# разрешаем дополнительные проверки пока отправитель

# передает RCPT TO: и MAIL FROM: заголовки. Для детализации mail.log

smtpd\_delay\_reject = yes

# требуем от отправителя представиться

# (на том, как себя представляет передающий комп,

# основаны многие эффективные проверки "на вшивость"

# автоматических рассылок от зомби и троянов)

smtpd\_helo\_required = yes

smtpd\_recipient\_restrictions =

permit\_mynetworks,

permit\_sasl\_authenticated,

reject\_unauth\_destination

proxy\_read\_maps = $local\_recipient\_maps $mydestination $virtual\_alias\_maps $virtual\_alias\_domains $virtual\_mailbox\_maps $virtual\_mailbox\_domains $relay\_recipient\_maps $relay\_domains $canonical\_maps $sender\_canonical\_maps $recipient\_canonical\_maps $relocated\_maps $transport\_maps $mynetworks

smtp\_use\_tls = yes

smtpd\_use\_tls = yes

smtp\_tls\_note\_starttls\_offer = yes

smtpd\_tls\_key\_file = /etc/dovecot/mail\_volmed\_org\_ru.key

smtpd\_tls\_cert\_file = /etc/dovecot/mail\_volmed\_org\_ru.cert

smtpd\_tls\_loglevel = 1

smtpd\_tls\_received\_header = yes

smtpd\_tls\_session\_cache\_timeout = 1s

tls\_random\_source = dev:/dev/urandom

22) Редактируем конфигурационный файл Postfix master.cf

В файл

sudo nano /etc/postfix/master.cf добавляем следующие строки

dovecot unix - n n - - pipe flags=DRhu user=virtual:virtual argv=/usr/lib/dovecot/deliver -d ${recipient}

submission inet n - - - - smtpd

-o smtpd\_tls\_security\_level=encrypt

-o smtpd\_sasl\_auth\_enable=yes

-o smtpd\_sasl\_type=dovecot

-o smtpd\_sasl\_path=private/auth

-o smtpd\_sasl\_security\_options=noanonymous

-o smtpd\_sasl\_local\_domain=$myhostname

-o smtpd\_client\_restrictions=permit\_sasl\_authenticated,reject

-o smtpd\_sender\_login\_maps=mysql:/etc/postfix/dovecot/mysql\_sender\_login\_maps.cf

-o smtpd\_sender\_restrictions=reject\_sender\_login\_mismatch

-o smtpd\_recipient\_restrictions=reject\_non\_fqdn\_recipient,reject\_unknown\_recipient\_domain,permit\_sasl\_authenticated,reject

smtps inet n - n - - smtpd

-o syslog\_name=postfix/smtps

-o smtpd\_tls\_wrappermode=yes

-o smtpd\_sasl\_auth\_enable=yes

23) Задаем сообщение, которое будет выводить Postfix в случае, если отправитель будет представляться именем нашего сервера

Создаем файл

sudo nano /etc/postfix/helo.list

mail.alex95.com 550 Don't use my hostname

24) прохешируем

sudo postmap /etc/postfix/helo.list

25) Задаем сообщение, которое будет выводить Postfix в случае, если отправитель извне будет говорить, что он из нашей сети

Создаем файл

sudo nano /etc/postfix/ext\_sender

mail.alex95.com 450 Do not use my domain in your envelope sender

26) прохешируем

sudo postmap /etc/postfix/ext\_sender

**Настроим demon saslauthd для взаимодействия с Postfix**

1) sudo mkdir -p /var/spool/postfix/var/run/saslauthd

2) Редактируем sudo nano /etc/default/saslauthd. Необходимо проверить, чтобы были закомментированы все строки, и написать следующее:

START=yes

DESC="SASL Authentication Daemon"

NAME="saslauthd"

MECHANISMS="pam"

MECH\_OPTIONS=""

THREADS=5

OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"

3) Редактируем sudo nano /etc/pam.d/smtp. Пишем туда следующее:

auth required pam\_mysql.so user=mail\_admin passwd=map host=127.0.0.1 db=mail table=users usercolumn=user passwdcolumn=password crypt=1

account sufficient pam\_mysql.so user=mail\_admin passwd=map host=127.0.0.1 db=mail table=users usercolumn=user passwdcolumn=password crypt=1

4) Настраиваем Postfix для взаимодействия с saslauthd

Редактируем файл sudo nano /etc/postfix/sasl/smtpd.conf Пишем туда следующее:

pwcheck\_method: saslauthd

mech\_list: plain login

allow\_plaintext: true

auxprop\_plugin: mysql

sql\_hostnames: 127.0.0.1

sql\_user: mail\_admin

sql\_passwd: map

sql\_database: mail

sql\_select: SELECT password FROM users WHERE user = '%u'

5) Владельцем данных файлов должен быть root

sudo chown root /etc/pam.d/smtp /etc/postfix/sasl/smtpd.conf

6) Добавляем пользователя postfix в группу sasl

sudo adduser postfix sasl

7) Конфигурируем Dovecot

Файл sudo nano /etc/dovecot/dovecot.conf Нужно убедится в том, что все строки закомментированы и добавить следующее

!include\_try /usr/share/dovecot/protocols.d/\*.protocol

#Разрешаем авторизацию в plaintext

disable\_plaintext\_auth = no

# Журнал будем писать в файл /var/log/dovecot.err

log\_path = /var/log/dovecot.err

# Формат даты и времени для регистрируемых событий

log\_timestamp = "%Y-%m-%d %H:%M:%S "

#Включаем SSL

ssl = yes

# Порядок следования сертификатов имеет большое значение: сначала \*.key, затем \*.cert.

ssl\_key =< /etc/dovecot/mail\_volmed\_org\_ru.key

ssl\_cert =< /etc/dovecot/mail\_volmed\_org\_ru.cert

#Строка приветствия при ответе сервера

login\_greeting = Dovecot ready.

#Описываем тип (maildir) и местонахождения почтовых ящиков (/var/spool/mail/%d/%n) %d - имя сервера, %n - имя пользователя

mail\_location = maildir:/var/spool/mail/%d/%n

#Задаем идентификатор пользователя и группы, с которыми будет работать dovecot

mail\_uid = 5000

mail\_gid = 5000

mail\_privileged\_group = mail

valid\_chroot\_dirs = /var/spool/mail/

#Настраиваем вывод отладочных сообщений

auth\_verbose = yes

auth\_debug = yes

auth\_debug\_passwords = yes

#Типы допустимых вариантов аутентификации

auth\_mechanisms = plain login digest-md5

#Задаем параметры аутентификации

passdb {

driver = sql

args = /etc/dovecot/dovecot-sql.conf

}

service auth {

unix\_listener auth-master {

mode = 0660

user = virtual

group = virtual

}

unix\_listener /var/spool/postfix/private/auth {

mode = 0660

user = postfix

group = postfix

}

}

service imap-login {

inet\_listener imap {

port = 0

}

inet\_listener imaps {

port = 993

ssl = yes

}

}

8) Настраиваем параметры соединения с базой данных MySQL.

Редактируем sudo nano /etc/dovecot/dovecot-sql.conf. Пишем туда следующее:

driver = mysql

connect = host=127.0.0.1 dbname=mail user=mail\_admin password=map

default\_pass\_scheme = CRYPT

password\_query = SELECT email AS user , password FROM users WHERE (user = '%u') or (email = '%u');

9) Проверяем конфигурацию Dovecot

dovecot -a

10) Создадим в базе данных MySQL почтовый домен и два пользователя и пару алиасов

mysql -uroot -proot

USE mail;

INSERT INTO domains (domain) VALUES ('mail.alex95.com');

INSERT INTO users (user, email, password) VALUES ('user1', 'user1@mail.alex95.com', ENCRYPT('user1\_password'));

INSERT INTO users (user, email, password) VALUES ('user2', 'user2@mail.alex95.com', ENCRYPT('user2\_password'));

INSERT INTO forwardings(source, destination) VALUES ('root', 'user1@mail.alex95.com');

INSERT INTO forwardings (source, destination) VALUES ('system', 'user1@mail.alex95.com');

quit

11)Создадим сертификат и ключ для SSL

cd /etc/dovecot

sudo openssl req -new -outform PEM -out server.crt -newkey rsa:2048 -nodes -keyout server.key -keyform PEM -days 3650 -x509

# На дополнительные вопросы ответил:

Country Name (2 letter code) [AU]:BY

State or Province Name (full name) [Some-State]:MINSK

Locality Name (eg, city) []:MISNK

Organization Name (eg, company) [Internet Widgits Pty Ltd]:TERESHKO

Organizational Unit Name (eg, section) []:ALEXEY

Common Name (e.g. server FQDN or YOUR name) []:mail.alex95.com

Email Address []:[alexis95s@mail.ru](mailto:alexis95s@mail.ru)

12) Владельцем ключа должен быть root

sudo chown root /etc/dovecot/server.key

13)

* + 1. Обновить конфигурацию - настроить отправку писем с localhost, адресованых @gmail.com - через учетку на @gmail.com или @yandex.ru (зарегистрировать любую тестовую учетку самостоятельно), а всю остальную почту - с сервера localhost
    2. Написать shell-скрипт, который \_через\_telnet\_ сможет отправить письмо в plain text (входные параметры: server, port, from-email, to-email, title и body, при необходимости – можно добавить параметр password)
    3. (задание "на 10ку") Установить и настроить сервер входящей почты. Учетки и пароли хранятся в базе MySQL. (проверка: отправить письмо на localhost.tld и получить его).
    4. (задание "на 10ку") Установить и настроить любой web-интерфейс для почты (проверка: залогиниться и посмотреть свой ящик из 2.2.4.)
  1. Потренироваться с http://mxtoolbox.com/ и почтовыми провайдерами (вроде gmail.com и yandex.com) для проверки правильности настроек.

1. Vagrant/Docker + Packer

Альтернативой Vagrant может быть, например, Docker.

- Vagrant:

оригинал:

<http://docs-v1.vagrantup.com/v1/docs/index.html>

- цикл статей на русском:

<http://sysadm.pp.ua/category/linux/sistemy-virtualizacii>

<http://sysadm.pp.ua/linux/sistemy-virtualizacii/vagrant-begin.html>

<http://sysadm.pp.ua/linux/sistemy-virtualizacii/vagrant-box-creation.html>

- блог на русском

<https://github.com/uran1980/web-dev-blog/blob/master/Vagrant/getting-started-with-vagrant.md>

- небольшая статья по старту

<http://eax.me/vagrant/>

- Docker:

<https://www.digitalocean.com/community/tutorials/how-to-work-with-docker-data-volumes-on-ubuntu-14-04>

- packer:

<https://www.packer.io/intro>

<https://blog.codeship.com/packer-vagrant-tutorial/>

<http://kappataumu.com/articles/creating-an-Ubuntu-VM-with-packer.html>

<http://www.greenhills.co.uk/2014/04/02/vagrant-with-packer-on-virtualbox.html>

- видео:

<https://www.youtube.com/watch?v=ZGUEjZckijA&index=5&list=PL890C8B18D140F0C3>

<https://www.youtube.com/watch?v=CHf1KoqYOoY&index=1&list=PLMvUShDprR-UKnKgpx8wLHZW8MCsfDAku>

Задание

* 1. +Создать базовый box с Ubuntu 14.04 с помощью packer, выходной формат - vagrant (virtualbox)

По видео - <https://www.youtube.com/watch?v=ZGUEjZckijA&index=5&list=PL890C8B18D140F0C3>

sudo apt-get install vagrant #installing vagrant

sudo apt-get install virtualbox #installing virtualBox

Заходим на сайт <http://www.vagrantbox.es/>. Выбираем нужный нам бокс с операционной системой.

mkdir vagrant

cd vagrant

$ vagrant box add {title} {url} #создание бокса, где title-имя(ubuntu-14.04), url-адрес для скачки

$ vagrant init {title} #запуск бокса. Создается файл VagrantFile. Следует его отредактировать по видео

$ vagrant up #запуск виртуальной операционной системы

vagrant ssh #создаем безопасную оболочку, через которую будем работать в виртуальной системе через vagrant

* 1. +Запустить свежую виртуалку в vagrant
  2. ?Добавить chef-рецепты для установки на виртуалке Apache, Nginx, PHP, Ruby, MySQL (client & server) - последних версий.
  3. Пробросить порты следующим образом: хост:22022->гость:22, хост:22080->гость:80, хост:22443->гость:443, хост:22306->гость:3306

Видео для понятия SSH - <https://www.youtube.com/watch?v=OXed4jhun68>

Подробно про SSH - https://www.youtube.com/watch?v=5bF-DTxvvmQ

* 1. С помощью packer.io собрать эту же вируталку QUEMU.

Команда для записи в $PATH: export PATH=$PATH:/home/alexey/packer/

* 1. С использованием шаблонов вируталки (для virtualbox) создать новый шаблон - с предустановленным mariadb вместо mysql

4. Ansible

Материалы

- установка ansible:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-an-ubuntu-12-04-vps>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-14-04>

- работа с playbooks

<https://www.digitalocean.com/community/tutorials/how-to-create-ansible-playbooks-to-automate-system-configuration-on-ubuntu>

- настройка

<https://www.digitalocean.com/community/tutorials/how-to-configure-apache-using-ansible-on-ubuntu-14-04>

<https://www.digitalocean.com/community/tutorials/how-to-deploy-a-basic-php-application-using-ansible-on-ubuntu-14-04>

Задание

* 1. Создать 3 репозитория в github или bitbucket с названием ab-haproxy, ab-logstash, ab-webui (все дальнейшие действия подразумевают выгрузку результатов в эти репозитории).

Настройка git: <https://www.youtube.com/watch?v=SwK2dPFXhpU>

1) sudo apt-get install git #устанавливаем

2) git config --global user.name "LeshikJanz" #конфигурируем под свою учетную запись

3) git config --global user.email "[treshaalesha@gmail.com](mailto:treshaalesha@gmail.com)" # конфигурируем email

4) Создаем репозиторий на сайте

5) mkdir ab-haproxy #создаем директорию с названием репозитория

6) Вводим одну за одной команды (команды с сайта):

echo "# ab-haproxy" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/LeshikJanz/ab-haproxy.git

git push -u origin master

* 1. Создать в Vagrant виртуальную машину Ubuntu 14.04
  2. ab-haproxy
     1. Создать следующие роли в Ansible (можно пользоваться репозиторием Galaxy):
* apt (добавление необходимых пакетов, обновление из репозиториев установленных по умолчанию пакетов)
* ntp (обнвоить время, настроить синхронизацию времени по cron 1 раз в сутки с любого общедоступного сервера времени)
* monit (установить и настроить monit для само-мониторинга виртуальной машины, правила можно составить любые, например: перезапуск haproxy)
* haproxy (можно адаптировать роли из [https://galaxy.ansible.com/list#/roles](https://galaxy.ansible.com/list" \l "/roles), haproxy будет использоваться в качестве балансировщика для веб-иинтерфейса 5.5.)
  + 1. На выходе Ansible так же должен быть сформирован ini-файл, содержащий любой уникальный ID виртуалки, например Packer ID, следующего формата:

[general]

uniqueID=

* 1. ab-logstash
     1. Создать следующие роли в Ansible (можно пользоваться репозиторием Galaxy):
* apt (добавление необходимых пакетов, обновление из репозиториев установленных по умолчанию пакетов)
* ntp (обнвоить время, настроить синхронизацию времени по cron 1 раз в сутки с любого общедоступного сервера времени)
* monit (установить и настроить monit для само-мониторинга виртуальной машины)
* java (smola.java role ([https://galaxy.ansible.com/list#/roles/1209](https://galaxy.ansible.com/list" \l "/roles/1209)) для установки openjdk-7-jdk)
* logstash (<https://www.elastic.co/downloads/logstash> без локального syslog и web интерфейса)
* elasticsearch (<https://www.elastic.co/downloads/elasticsearch>), со следующими опциями:
  + установленный плагин royrusso/elasticsearch-HQ
  + установленный плагин mobz/elasticsearch-head
  + установленный плагин elasticsearch/elasticsearch-cloud-aws/2.4.2 (данной версии)
  + имя кластера "logstash"
  + CORS enabled
    1. На выходе Ansible так же должен быть сформирован ini-файл, содержащий любой уникальный ID виртуалки, например Packer ID
  1. ab-webui
     1. Создать следующие роли в Ansible (можно пользоваться репозиторием Galaxy):
* apt (добавление необходимых пакетов, обновление из репозиториев установленных по умолчанию пакетов)
* ntp (обнвоить время, настроить синхронизацию времени по cron 1 раз в сутки с любого общедоступного сервера времени)
* monit (установить и настроить monit для само-мониторинга виртуальной машины)
* rsyslog (для трансляции любых стандартных системных логов)
* kibana 3.x (нужна версия 3.x!)
* nginx (настроить таким образом, чтобы по запросу к сайту по умолчанию отдавался интерфейс kibana, важно: т.к. kibana 3.x состоит из статических файлов, то не нужно устанавливать интерпретаторы, все должно работать только со статическим контентом)
  + 1. На выходе Ansible так же должен быть сформирован ini-файл, содержащий любой уникальный ID виртуалки, например Packer ID
  1. Настроить конфигурации следующим образом:
* logstash должен собирать логи с rsyslog на ВМ WebUI
* kibana должен предоставлять интерфейс для просмотра логов из logstash
* доступ к kibana должен быть возможен через ВМ haproxy

5. Amazon

Материалы

1. <https://aws.amazon.com/training/intro_series/> - вводные демо-видео по основным сервисам.
2. <https://qwiklabs.com/> - Amazon рекомендует этот источник как официальный сервис для самообучения.
3. [Architecting for The Cloud: Best Practices](http://media.amazonwebservices.com/AWS_Cloud_Best_Practices.pdf) – общее описание сервисов Amazon и для чего их следует использовать.
4. [19 Best Practices for Creating Amazon CloudFormation Templates](http://harish11g.blogspot.com.by/2014/08/amazon-cloudformation-templates-automation-Amazon-CFT-AWS-top-best-practices-tips.html) –рекомендации по использованию CloudFormation
5. [AWS Best Practices: five key approaches to get you started](http://cloudacademy.com/blog/aws-best-practices/) – Рекомендации по использованию ключей

Задание

* 1. Создать выделенную сеть Amazon Virtual Private Cloud с двумя подсетями в разных зонах (например, обычно обозначаются как us-west-1a, us-west-1c).
     1. Создать настраиваемую Security Group:
* Разрешить входящий SSH трафик только со своего IP (или доверенных IPs).
* Разрешить входящий HTTP/HTTPS трафик со своего IP (или доверенных IPs).
* Разрешить весь трафик между созданными зонами.
* Остальной входящий трафик запретить.
  1. Сгенерировать собственный RSA ключ (Key Pairs) для использования в дальнейшем при создании инстансов  (необходимо для подключении по SSH).
  2. EC2:
     1. Создать один t2.micro инстанс в созданной в п1 VPC и одной из availability zone. Использовать Security Group из п1.1.
     2. Создать второй t2.micro инстанс в созданной в п1 VPC и неиспользованной availability zone. Использовать Security Group из п1.1.
     3. На оба инстанса установить Nginx и создать простую страницу-заглушку (index.html) на 80-ом порту.
  3. ELB:
     1. Создать один ELB с поддержкой созданных Availability Zones.
     2. Разрешить HTTP трафик на ELB с любого IP адреса.
     3. Добавить в ELB оба инстанса.
     4. Настроить Health Check на протокол HTTP, порт 80, страница index.html с минимальными интервалами проверки.
     5. Принудительно остановить веб-сервер на одном из инстансов и проверить доступность сайта.
  4. RDS:
     1. Создать инстанс PostgreSQL в выделенной VPC с типом хранилища как General Purpose и объёмом в 100 Гб.
     2. Разрешить трафик только внутри выделенной VPC.
  5. ElastiCache:
     1. Создать один инстанс ElastiCache (Redis) в выделенной VPC.
     2. Разрешить трафик только внутри выделенной VPC.
     3. Создать один инстанс ElastiCache (Memcached) в выделенной VPC.
     4. Разрешить трафик только внутри выделенной VPC.
  6. **\***Подготовить шаблон CloudFormation используя [Nested Stacks](https://blogs.aws.amazon.com/application-management/post/Tx1T9JYQOS8AB9I/Use-Nested-Stacks-to-Create-Reusable-Templates-and-Support-Role-Specialization) для автоматизации выполнения пунктов 1, 3, 4, 5 и 6. Пункт 2 должен быть предварительно выполнен!
     1. Добавить в шаблон триггеры CloudWatch для автомасштабирования (добавление/уменьшение количества) EC2 инстансов по показателям CPU:
* >70% в течение 10 минут – добавить один инстанс;
* >30% в течение 15 минут – удалить один инстанс.
  + 1. Обновить стэйк ЕС2 инстансов с атоматической заменой двух старых.
  1. Создать CloudFront Distribution с параметрами по умолчанию.
     1. Сгенерировать 100 небольших файлов ( < 512 Kb) и заполнить ими созданный бакет в S3.
     2. Настроить политику хранения объектов в данном бакете S3 следующим образом:
* по истечении 30 дней – отправлять объекты в Glacier;
* после 6 месяцев хранения – полностью удалять с Glacier.

xmodmap .xmodmap