**Горячие клавиши:**

Для открытия терминала ctrl+alt+t

Для очистки терминала ctrl+L

Для увеличения шрифта в терминале ctrl+shift++

При писании имени пакета, можно нажать Tab, чтобы имя дописалось

Подключиться с cmd Windows к виртуалке, набираем в cmd ssh db@192.168.1.105

***Комнды в Ubuntu:***

1. sudo apt update - скачать программные пакеты. Загрузятся список доступных пакетов для устанвоки

2. sudo apt install openssh-server - установка пакета SSH-сервера. Можно через пробел указать дополнительные программы, которые необходимо поставить. Например, поставить дополнительно mc(Midnight Commander)

3. systemctl status - проверить стутс установленных пакетов. Должно быть без ошибок

4. sudo ss -ntlp - проверить, что запущен ssh-сервер (в первой строке все нули)

5. ip a - посмотереть IP адрес машины

6. Для подключение к виртуальной машине через cmd нужно набрать ssh имя пользователя в виртуалке@IP адрес виртуалки. Далее подтверждаем подключениие и вводим пароль виртулаки.

7. Для запукск приложений, нужно написать название приложения. Например, если нужно запустить Midnight Commander, нужно набрать mc

8. Для удобства работы с версией desktop нужно установить гостевые дополениния. Для этого нужно в меню виртуалки нажимаем Устройства->Подключить образ диска Дополнений гостевой ОС. Далее в вертуалке появляется диск, из которого установим дополнения. Для устанвоки дополнений, нужно установить дополнительные пакеты с помощью sudo apt install gcc make perl. После этого в вертуалке в диске VBox\_GAs запускаем autorun.sh нажатием правой клавиши на него, выбираем и вводом Run ad a Program, вводим пароль.

9. cd /папка позовляет зайти в указанную папку

10. sudo su зайти под супер пользователем или администратором (root)

11. Команда cd или cd~ возвращает в домашний каталог. ~ - тильда. cd / перейти в корень

12. Команда ls -al покажет список всех файлов.

13. ls -l покажет список файлов начинающихся не на точку

14. cat распечатывает содержание файла в терминал

15. cat cpuinfo распечатывает информацию из файла cpuinfo - информация о процессоре.

16. ***Работа с файлами:***

- ls - список файлов. Параметр -a выводит список в сокращенном виде. -al выводит подробный список. ls --help выводит помощь по команде. man ls выводит подробную информацию по команде. ll выводит также содержание папки

- pwd - текущая директория

- mkdir - создание каталога

- cp - копирование

- rm – удаление. Для удаления файла достаточно ввести rm имя файла. Для удаления каталога нужно ввести rm -rf имя каталога.

- rmdir – безопасное удаление пустых папок.

- mv - переименование/перенос. Если файлы находятся в одной директории, то будет переименование файла. Если файлы будут находится на разных дисках или разделах диска, то будет перенос файлов.

- touch - создание пустого файла

- cat - вывод файла, склейка, создание. Команда cat имя файла, выводит содержимое файла. cat testfile test2 > test\_all склеит 2 файла в файл test\_all

echo – создание файлов.

- саt >> имя файла – дозаписывает данные в файл

- ln имя файла имя файла – создает связывает файлы ссылкой (inode).

- ln -s test2 test\_lns – создание символической ссылки

- ls -ali – просмотр файлы со ссылками

- ln -s /home/db/test3/test2 test\_lns\_abs – создание абсолютной ссылки

- mv test\_lns .. – перенос файла на папку выше

- ln -s test3 lest\_ln\_dir – создание ссылки на каталог (папку)

Текстовые редакторы: vim, nano, Mcedit

Пейджеры (текстовые просмоторщики): less, more

Просмотр файла частями: Просмотр с начала head -n 10 .profile (10 это количество строк)

Просмотр с конца tail -n 10 .profile

tail -f /var/log/auth.log – отслеживать изменения в файле (заход на виртуалку, подключение и т.д.)

**Управление пользователем:**

* useradd – создание пользователя. sudo useradd -s /bin/bash – m – d /home/testuser testuser. sudo useradd -s /bin/bash -d /home/test -m test – создаем пользователя test (-s /bin/bash – задаем пользователю shell, -d /home/test – домашняя директория, -m test – опция для создания директории и имя пользователя.
* adduser – создание пользователя (скрипт). sudo adduser testuser2. С помощью этой команды мы создаем сразу все дополнительные свойства пользователя.
* usermod – изменение пользователя. sudo usermod -g www-data test2 – изменение основной группы пользователя. sudo usermod -aG sambashare test2 – добавить в дополнительную группу пользователя test2 в группу sambashare. Основные группы прописаны в файле passwd, все дополнительные в файле group
* userdel – удаление пользователя
* passwd – изменение пароля. Если задать команду passwd таким образом, то будет предложено сменить пароль текущего пользователя. Для других пользователей нужно набрать sudo passwd testuser.
* chage – изменение свойств пароля
* groupadd – создание группы
* groupdel – удаление группы

sudo ls -al /home/testuser – просмотр домашней директории других пользователей.

При создании нового пользователя, файлы будут скопированы из каталога /etc/skel. Это каталог нужен для создания и настройки новых пользователей.

sudo usermod -aG adm testuser – добавление пользователя testuser в группу adm. Смена дополнительной группы.

sudo usermod -g www-data testuser – Смена основной группы.

id- команда покажет в какой группе находимся мы (текущий пользователь).

sudo id testuser – покажет в каких группах находится пользователь testuser. При добавлении пользователя в группу, необходимо перелогиниться.

man usermod – Help по команде изменения свойств пользователя.

sudo userdel testuser2 – удаление пользователя. Домашняя пользователя остается. Для удаления пользователя с домашней директорией, нужно добавить дополнительно -r.

sudo change testuser – изменение свойств пароля

sudo groupadd testgroup – создание группы testgroup

sudo groupdel testgroup – удаление группы testgroup

su testuser – переключение на пользователя testuser. После выполнения все необходимой работы под другим пользователем. Для переключение обратно на нашего пользователя, достаточно ввести команду exit. С помощью команды su можем переключиться только на пользователя с паролем.

sudo su – переклюет на суперпользователя.

sudo visudo – открывает файл /etc/sudoers.tmp. В этом файле находятся права пользователей.

sudo chown testuser:www-data testt – меняем владельца и группу у файла test

sudo chown -R testuser:www-data /home/db/test3 – меняем владельца и группу у папки test3

sudo chgrp -R adm /home/db/test3 – меняем только группу у папки test3

sudo chgrp db test2 – меняем группу у файла.

nano test3 – открытие файла. Если открываемый файл не имеет доступа к записи, то это файл откроется только для чтения.

sudo nano test3 – открывает файл с полным доступом.

sudo chmod u=rwx,g=rw,o=rx test2 – задаем права на файл. Для владельца файла u задаем r-чтение, w-запись и x-исполнение. Для группы g задаем r-чтение, w-запись. Для остальных o r-чтение и x-исполнение. Если в место равно использовать -, мы убираем права. Если +, мы добавляем права.

sudo chmod 664 test2 – задаем права на файл цифрами. 6-rw (биты 4+2 – 4-r, 2-w), 4-r. 1-x (бит исполнение. Полный доступ для файла – 666, для директории – 777. При создании файла или директории по умолчанию будут ставится права из umask, т.к. из полных прав вычитается значение umask. Umask для разны систем разный. Для проверки настроек нужно ввести команду umask. Изменить umask можно ввести команду umask <значение>.

sudo chmod -R 775 test3 – применение прав доступа к директории.

sudo chmod g+s test3 – даем специальные права s себе для группы.

sudo chmod a+w /home/db/test3 – назначаем себе (пользователю db) права на запись в директорию. Т.е. если мы создадим файл, то он создастся от имени другой группы.

Бит t (стикибит) применяется для временных файлов. Бит t также позволяет, например на папку установить права на запись, но удаление только владельца.

cat /etc/passwd | grep test – просмотр содержимого файла passwd с фильтром по пользователю test

Если установить специальные права s на папку, тогда при создании файла будет автоматически ставиться группа папки.

**Утилита apt update**

apt update – запрос пакетов для обновления

apt search nginx – поиск пакета по названию nginx

apt list nginx – просмотреть состав пакета nginx

apt install nginx – установка пакета nginx (можно устанавливать несколько пакетов сразу, перечислив их через запятую)

apt install -y nginx – установка пакета nginx без подтверждения

apt remove nginx – удаление пакета nginx

apt autoremove – удалить пакеты, которые нам не нужны и которые зависят от nginx

apt purge nginx – удаление пакета nginx с его директорией

apt purge nginx libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 nginx-common nginx-core – удаление пакета nginx с его директорией и указанными обновлениями

apt show mc – посмотреть зависимости к пакету mc

wget <https://download.virtualbox.org/virtualbox/7.0.14/virtualbox-7.0_7.0.14-161095~Ubuntu~jammy_amd64.deb> - получаем обновление по ссылке с помощью команды wget

dpkg -i virtualbox-7.0\_7.0.14-161095~Ubuntu~jammy\_amd64.deb – с помощью команды dpkg -i устанавливаем. Virtualbox может не установиться, т.к. для этого необходимо дополнительные пакеты

apt -f install – установит Virtualbox и все необходимые пакеты

sudo !! – повторить предыдущую команду

***Сеть в Linux***

ip a – список всех интефейсов

ip -s a – показ статистики

ip -c -s a – включение подсветки, т.е. разноцветный вывод

ip a show enp0s3 – данные по одному интефейсу

ip link show enp0s3 – данные уровня L2 (link)

ip r – просмотр информации о маршрутах

ip addr add 192.168.1.110/255.255.255.0 broadcast 192.168.1.255 dev enp0s3 – создаем еще одни вторичный IP-адрес на нашем адаптере

ip route delete default via 192.168.1.1 – удаление маршрута. В этом случае маршрута по умолчанию.

ip route add default via 192.168.1.1 dev enp0s3 – добавляем маршрут

Сокеты и порты

ss – socket stat

ss -ntlp – TCP-сокеты в состоянии LISTEN

ss -ntulp – TCP и UDP-сокеты в состоянии LISTEN

ss -tulpan – Все TCP и UDP-сокеты

***Диагностика сети***

ping – 8.8.8.8 – доступность хоста (ICMP протоколов)

ping – ya.ru – проверка DNS и доступности

host -t a yandex.ru – проверка DNS

host -t a Yandex.ru 8.8.8.8 – другой DNS-сервер

dig @8.8.8.8 google.com – подробная информация по DNS

tracepath ya.ru – просмотр маршрута прохождение пакетов. Для этого нужно сначала установить пакет с помощью apt install traceroute.

traceroute ya.ru – альтернатива

mtr ya.ru – постоянный мониторинг доступных хостов

resolvectl dns – посмотреть текущие активные dns-сервера

***Фильтрация пакетов***

Просмотр таблицы

iptables -L -nv

iptables -L -nx -t nat

Политика по умолчанию

iptables -P INPUT DROP

Добавление правил

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -I INPUT -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -p tcp -s 192.168.0.100 --dport 80 -j DROP

Управление правил

iptables -D INPUT 3

iptables -D INPUT -p tcp --dport 80 -j ACCEPT

Сброс правил

iptables -F

iptables -L -nv – просматриваем список правил

iptables -A INPUT -p tcp --dport=22 -j ACCEPT – разрешили ssh трафик

iptables -A INPUT -i lo -j ACCEPT – разрешили трафик на интерфейсе лубэк

iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT – подключает модуль state и разрешаем пакеты ESTABLISHED,RELATED

iptables -A INPUT -p icmp -j ACCEPT – разрешаем icmp правила

iptables -P INPUT DROP – устанавливаем политику по умолчанию

***Пример конфигурации сервера***

# SSH allow

iptables -A INPUT -p tcp --dport=22 -j ACCEPT

# HTTP, HTTPS allow

iptables -A INPUT -p tcp -m multiport --dport 80, 443 -j ACCEPT

# loopback allow

iptables -A INPUT -I lo -j ACCEPT

#ICMP

iptables -A INPUT -p icmp -j ACCEPT

# established connection allow

iptables -P INPUT DROP

***Перенаправление портов***

# Редирект с 80 на 8080 порт (TCP):

iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080

# Проверка:

Iptables -L -nv nat

***Сохранение конфигурации iptables***

Сохранение и восстановление из файла

iptables-save > iptables.rules

iptables-restore < iptables.rules

Сервис netfilter-persistent

apt install iptables-persistent netfilter-persistent

netfilter-persistent save

Конфигурация в /etc/iptables

Настройка сетевой конфигурации через netplan

- cd /etc/netplan

- nano 00-installer-config.yaml

Содержание файла 00-installer-config.yaml. Получает IP автоматически

network:

version: 2

renderer: networkd

ethernets:

enp0s3:

dhcp4: yes

Содержание файла 00-installer-config.yaml. Получает IP автоматически

network:

version: 2

renderer: networkd

ethernets:

enp0s3:

dhcp4: no

addresses: [192.168.1.107, 192.168.1.110]

routes:

- to: default

via: 197.168.1.1

nameservers:

addresses:

- 8.8.8.8

- 8.8.4.4

**Настройка UFW Ubuntu**

**Команды UFW**

**Для выполнения действий с утилитой доступны такие команды:**

enable - включить фаерволл и добавить его в автозагрузку;

disable - отключить фаерволл и удалить его из автозагрузки;

reload - перезагрузить файервол;

default - задать политику по умолчанию, доступно allow, deny и reject, а также три вида трафика - incoming, outgoing или routed;

logging - включить журналирование или изменить уровень подробности;

reset - сбросить все настройки до состояния по умолчанию;

status - посмотреть состояние фаервола;

show - посмотреть один из отчётов о работе;

allow - добавить разрешающее правило;

deny - добавить запрещающее правило;

reject - добавить отбрасывающее правило;

limit - добавить лимитирующее правило;

delete - удалить правило;

insert - вставить правило.

* sudo ufw status – проверка статуса UFW
* sudo ufw enable – включем UFW
* sudo ufw default deny incoming – запрет всех входящих. Смотрим политику по умолчанию
* sudo ufw default allow outgoing – разрешаем все исходящие
* sudo ufw allow OpenSSH – открываем ssh
* sudo ufw allow 22 – открываем порт 22
* sudo ufw allow 22/tcp – открываем порт 22 tcp
* sudo ufw app list - Посмотреть доступные имена приложений можно с помощью команды
* sudo ufw allow out 80/tcp - разрешим только исходящий трафик на порт 80
* sudo ufw deny in 80/tcp - входящий запретим
* sudo ufw allow proto tcp from 0.0.0.0/24 to 192.168.1.5 port 3318 - В качестве ip\_источника может использоваться также и адрес подсети. Например, разрешим доступ со всех IP-адресов для интерфейса eth0 по протоколу tcp к нашему IP-адресу и порту 3318
* sudo ufw limit ssh/tcp - С помощью правил limit можно ограничить количество подключений к определённому порту с одного IP-адреса, это может быть полезно для защиты от атак перебора паролей. По умолчанию подключения блокируются, если пользователь пытается создать шесть и больше подключений за 30 секунд
* sudo ufw status verbose – посмотреть более подробную информацию по действующим правилам.

С помощью команды show можно посмотреть разные отчеты:

raw - все активные правила в формате iptables;

builtins - правила, добавленные по умолчанию;

before-rules - правила, которые выполняются перед принятием пакета;

user-rules - правила, добавленные пользователем;

after-rules - правила, которые выполняются после принятия пакета;

logging-rules - правила логгирования пакетов;

listening - отображает все прослушиваемые порты и правила для них;

added - недавно добавленные правила;

- sudo ufw show raw - посмотрим список всех правил iptables

- sudo ufw show listening - Посмотрим все прослушиваемые порты

- sudo ufw show added - недавно добавленные правила

- sudo ufw delete allow out 80/tcp - удалим ранее созданные правила для порта 80

- sudo ufw delete deny in 80/tcp

- sudo ufw logging on - включения журналирования используется команда logging

- sudo ufw logging medium

Также этой командой можно изменить уровень логгирования:

low - минимальный, только заблокированные пакеты;

medium - средний, заблокированные и разрешённые пакеты;

high - высокий.

Лог сохраняется в папке /var/log/ufw. Каждая строчка лога имеет такой синтаксис:

[UFW действие] IN=интерфейс OUT=итерфейс SRC=ip\_источника DST=ip\_назначения LEN=размер\_пакета TOS=0x10 PREC=0x00 TTL=64 ID=728 DF PROTO=протокол SPT=порт\_источника DPT=порт назначения LEN=размер\_пакета

- sudo ufw disable - полностью отключить UFW

- sudo ufw reset - команда reset для сброса настроек до состояния по умолчанию

***Веб-сервер на Linux***

Консольные утилиты для веб

Получить URL в консоли:

* curl -L <https://ya.ru/> - покажет контент страницы
* wget <https://yastatic.net/jquery/2.1.4/jquery.min.js> - скачивание чего-то
* apt install curl – установим пакет curl

Протоколы SSL/TLS и HTTPS

SSL – первые версии защищенного протокола

TLS – современный проткол

HTTPS – работа протокола HTTP поверх защищенного канала SSL/TLS

HTTP/2, HTTP/3 – по умолчанию используют защищенный вариант

Конфигурация Nginx

Установка: sudo apt install nginx

Тестирование конфигурации: sudo nginx -t

Применить: sudo systenctl reload nginx

Конфигурация: /etc/nginx/\*

Основной файл: /etc/nginx/nginx.conf

Блок: server{}

Директивы: server\_name site.ru;

Переменные: $uri

Документация: <http://nginx.org/ru/docs/>

Пример конфигурации Nginx сервера

server {

listen 80default\_server;

listen [: :] :80 default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location / {

try\_files $uri $uri/ =404;

}

}

* ps afx – посмотреть таблицу процессов
* nginx -t – проверяем конфигурацию файла nginx
* systemctl reload nginx – применяем конфигурацию

Конфигурация Apache

Установка: sudo apt install apache2

Тестирование конфигурации: sudo apachectl -t

Применить: sudo systemctl reload apache2

Конфигурация: /etc/apache2/\*

Основной файл: /etc/apache2/apache2.conf

Блоки: <VirtualHost></VirtualHost>

Директивы: ServerName site.ru

Документация: <https://httpd.apache.org/docs/2.4/en/>

Пример конфигурации Apache

Listen 8081

<VirtualHost \*:8081>

ServerName [www.exemple.com](http://www.exemple.com)

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

ErrorLog /var/log/httpd/error.log

CustomLog /var/log/httpd/access.log combined

</VurtualHost>

* apachectl -t – проверяем конфигурацию
* systemctl start apache2 – запускаем сервер в ручную
* systemctl status apache2 – проверяем статус и наличие ошибок
* systemctl reload apache2 – применяем конфигурацию
* curl localhost:8080 – проверяем доступность сервера

Пример конфигурации Nginx для Reverse proxy

# Динамические запросы

Location / {

proxy\_pass <http://localhost:8080>

proxy\_set\_header Host $host

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Real-IP $remote\_addr;

}

#Статические запросы

location ~\* ^.+.(jpg | jpeg | gip | png | ico | css | zip | pdf | txt | tar | js)$ {

root /var/www/html;

}

Обработка PHP в Apache

Установка: apt install libapache2-mod-php8.1 php8.1

Обработка PHP подключается автоматически, по расширению файла

Файл для проверки:

<?php

Phpinfo();

?>

Проверяем: curl <http://localhost/info.php>

Установка:

* apt install php8.1 libapache2-mod-php8.1
* cd /etc/apache2
* cd mods-enabled/
* cat php8.1.conf
* cd /var/www/html
* cat > info.php

<?php

phpinfo();

?>

Обработка PHP в PHP-FPM

Установка: apt install php8.1-fpm

Замена Apache для работы с PHP

Связь между Nginx и PHP-FPM через UNIX-сокет

Обработка происходит с помощью location в Nginx

Location ~ \.php$ {

Include snippets/fastcgi-php.conf;

root /var/www/html;

fastcgi\_pass unix:/run/php8.1-fpm.sock;

}

Установка MySQL и первые шаги

Установка: apt install mysql-server

Заходим в консоль MySQL: sudo mysql

Получаем список пользователей: SELECT \* FROM user\G

Созадем новую базу данных: CREATE DATABASE gb;

Создаем таблицу: CREATE TABLE test(I INT);

Создадим записи в таблице: INSERT INFO test (i) VALUES (1), (2), (3), (4) ;

Сделаем выборку из таблицы: SELECT \* FROM test;

* show databases; - показать доступные базы
* use mysql; - использовать базу
* show tables; - показать доступные таблицы
* SELECT \* FROM user; - вывод таблицы в не удобном формате
* SELECT \* FROM user\G;
* use gb; - заходим в базу
* CREATE TABLE test (i INT); - создали таблицу
* INSERT INTO test (i) VALUES (1),(2),(3),(4); - вставляем цифры
* exit выход из MySQL
* systemctl status nginx – проверить статус
* curl localhost – проверяем работоспособность
* systemctl restart apache2 – рестарт apache2
* curl localhost:8080 – проверка работоспособоность
* nginx -s reload – проверка и перезагрузка nginx
* apt install nginx apache2 libapache2-mod-php8.1 php8.1 php8.1-fpm – установка сразу несколько пакетов

**Реестр образов Docker Hub**

- Реестр по умолчанию в Docker

- Веб-интерфейс: <https://hub.docker.com/>

- Официальные образы (official image)

- Теги (latest, 0.4-apline)

- Документация к образам

Запуск тестового контейнера

Установка Docker: apt install docker.io

Проверка: sudo docker

Создание и запуск контейнера: docker run hello-world

docker ps – посмотреть список контейнеров, которые сейчас работают

docker ps -a – посмотреть список всех контейнеров

docker images – список образов

docker search nginx -поиск образа

docker pull nging – скачивание образа

docker start|restart|stop nginx – оператции с контейнером

docker rm 9cbf7c3230d0 – удаление контейнера

docker rmi hello-world – удаление образа

docker logs nginx – просмотр логов контейнера

**Типы сетей в Docker**

* Bridge – сети по умолчанию, аналог типа подключения NAT в VirtualBox
* Host – с помощью этого драйвера контейнер получает доступ к собственному интерфейсу хоста. Аналог подключения «Мост» в VirtualBox
* Macvlan дает контейнерам прямой доступ к интерфейсу и суб-интерфейсу (VLAN) хоста.
* Overlay позволяет строить сети на нескольких хостах с Docker

**Запуск Nginx – проброс портов и директории**

- Команда создания и запуска: docker run -d –restart always – name nginx1 \ -p 80:80 -v /var/www/html:/usr/share/nginx/html nginx

- Учитываем порты на хосте (ss -ntlp)

- Монтируем директории

- Активируем автозапуск

- Контейнер работает в режиме демона

- Сеть по умолчанию (bridge)

systemctl stop nginx – остановить nginx

docker start nginx1 – запуск nginx

ps afx – проверка запущенных процессов

**Работа внутри контейнера**

- **Не рекомендуется в реальной работе**

- Заходим: docker exec -ti nginx1 bash

- Смотрим настройки: ls -al /etc/

- Версия базового дистрибутива: cat /etc/os-release

- Настройка nginx: ls -al /etc/nginx/

- Директория сайта: ls -al /usr/share/nginx/html

**Docker Compose**

* Запуск нескольких контейнеров одной командой
* Декларативная конфигурация
* Формат YAML (отступы, вложенность)
* Автоматизация запуска систем контейнеров

**Синтаксис docker-compose.yml**

version: ‘3’

services:

worldpress:

depends\_on:

-db

image: worldpress:5.3.2-fpm-alpine

container\_name: worldpress

environment:

-WORDPRESS\_DB\_HOST=db:3306

volumes:

-wordpress:/var/www/html

network:

-app-network

**Использование Docker Compose**

* apt instell socker-compose – установка
* Проверка yml: apt install yamllint
* docker-compose up -d – создание и запуск
* docker-compose ps – список контейнеров
* docker-compose down – остановить и удалить
* docker-compose stop – остановить
* docker-compose start – запустить
* docker-compose rm – удалить остановленные

nano docker-compose.yml

version: '3'

services:

nginx:

image: nginx:latest

ports:

- 80:80

volumes:

- /var/www/html:/usr/share/nginx/html

docker-compose up -d

**Установка docker и docker-compose**

apt install docker.io docker-compose – установка двух пакетов

docker – проверяем установленные пакеты

docker run hello-world – запускаем приложение. В случае если приложения не будет, то оно скачается и установится

**Запуск контейнера с Nginx. Практика работы с docker**

ss -ntlp – смотрим, какое приложение слушает порт 80

service docker-proxy stop – останавливаем приложение, какое слушает порт 80

ss -ntlp | grep 80 – еще раз проверяем порт

sudo docker run -d --restart always --name nginx1 -p 80:80 -v /var/www/html:/usr/share/nginx/html nginx – команда на запуск nginx с настройками

docker ps – проверяем работу

curl localhost – заходим на локальный хост

docker logs nginx1 – проверяем лог

**Подготовка и запуск контейнера с помощью docker-compose**

**Работа с docker-compose – запуск контейнеров с Wordpress**

mkdir wordpress

cd wordpress/

cat > .env

MYSQL\_ROOT\_PASSWORD=dfkljs\_d324fD\_klj

MYSQL\_USER=wp

MYSQL\_PASSWORD=KMJ23f9sad\_80ds

mkdir nginx-conf

cd nginx-conf/

cat > nginx.conf

version: '3'

services:

db:

image: mysql:8.0

container\_name: db

restart: unless\_stopped

env\_file: .env

environment:

- MYSQL\_DATABASE=wordpress

volumes:

- dbdata: /var/lib/mysql

command: '--default-authentication-plugin=mysql\_native\_password'

networks:

- app-network

wordpress:

depends\_on:

- db

image: wordpress:6.0.1-php8.0-fpm-alpine

container\_name: wordpress

restart: unless-stopped

env\_file: .env

environment:

- WORDPRESS\_DB\_HOST=db:3306

- WORDPRESS\_DB\_USER=$MYSQL\_USER

- WORDPRESS\_DB\_PASSWORD=$MYSQL\_PASSWORD

- WORDPRESS\_DB\_NAME=wordpress

volumes:

- wordpress:/var/www/html

networks:

- app-network

webserver:

depends\_on:

- wordpress

image: nginx:1.22.0-alpine

container\_name: webserver

restart: unless-stopped

port:

- "80:80"

volumes:

- wordpress:/var/www/html

- ./nginx-conf:/etc/nginx/conf.d

networks:

- app-network

volumes:

wordpress:

dbdata:

networks:

app-network:

driver: bridge

apt install tree – устанавливаем утилиту tree

tree – просматриваем дерево

.

├── docker-compose.yml

└── nginx-conf

└── nginx.conf

ss -ntlp – просмотрим, кто слушает 80 порт

docker ps – смотрим подробнее

docker stop nginx1

***Скрипты Bash***

echo $? – выводит код ошибки предыдущей команды. Если равен 0, то ошибок нет.

echo "Failed" – выводит сообщение в кавычках

ls > ls\_txt – записываем результаты выполнения команды в файл ls\_txt с начала файла

ls >> ls\_txt – дописываем файл ls\_txt

cat /etc/lsb-release /etc/shadow 2> err.txt – выводим 2 поток в файл err.txt

cat /etc/lsb-release /etc/shadow 2> err.txt > std.txt – выводим результат в файлы. Первой команды в std.txt, второй в err.txt

cat /etc/lsb-release /etc/shadow > all.txt 2>&1 – результаты обеих команд выводим в один файл all.txt

ps afx | grep ssh – вывод списка процессов, в названии которых содержатся сиволы ssh

ps afx | grep ssh | grep -v pts – вывод списка процессов, в названии которых содержатся сиволы ssh, но не содержат pts

ps afx | grep [s]sh – показывает процессы, которые имеют одни символ в квадратной скобке.

cat syslog | grep error – выводит содержание, в котором содержится error

cat syslog | grep error | grep -v kernel – выводит содержание, в котором содержится error, но не содержит kernel

cat syslog | grep error | grep -v kernel | grep -i 'sqlite' – выводит содержание, в котором содержится error, но не содержит kernel и содержит sqlite без учета регистра

cat syslog | grep error | grep -v kernel | grep -i 'sqlite' | wc -l – выводит содержание, в котором содержится error, но не содержит kernel и содержит sqlite без учета регистра с подсчетом количества найденных строк

ls -a | sort – вывод списка файлов с сортировкой по имени

ls -a | sort | wc -l – выдает количество найденных файлов

df -h – показывает содержание корневой системы

df -h | grep '/dev/sda' – показывает содержание корневой системы, которые содержат путь

printenv – выводит все переменные, которые существуют в системе

echo $PWD – выводит содержание переменной PWD. PWD это переменная, в которой хранится текущая директория

echo $OLDPWD – OLDPWD это переменная предыдущей директории

var1=test – присвоение переменных

echo $var1 – выводит содержание переменной

bash – эта команда запускает оболочку bash. Если до этого мы были в оболочке bash, то запустив еще раз эту оболочку, мы будем в еще одной оболочке. Причем вновь запущенная оболочка ничего не знает о присвоенных ранее значений переменных.

С помощью команды exit выходим из текущей оболочке в предыдущую, и уже текущая оболочка знает про присвоенную в ней переменную. Если вообще выйти из терминала, то значение переменной потеряется.

var2='Test asdfasd asdfasdf asdfad' – присвоение переменной строки

var3="$var2 $var1" – присвоение строки с двойными кавычками, берет значения переменных указанных внутри и присваивает их переменной var3

var\_ls=$(ls -al) или var\_ls=$`ls -al` – присваиваем команду к переменной

echo $var\_ls – выводим переменную, в которой хранится команда. Команда при выводе переменной исполняется.

***Создаём первый скрипт на bash***

cat > testscript

#!/bin/bash

directory=$1

hidden\_count=$(ls -A $directory | grep '^\.' | wc -l)

echo “Hidden files in $directory found: $hidden\_count”

***Методы запуска скрипта***

● Относительный путь: ./testscript

● Абсолютный путь: /home/db/test/testscript

● Команда (должен быть в $PATH): testscript

● Через команду bash: bash testscript

● Первые три варианта требуют шебанг и права на

исполнение

./testsc /etc – запускаем файл с передачей в переменную directory пути /etc. Эта команда запустится если у файла есть права доступа x, которое можно установить с помощью команды chmod +x testsc. Если у файла нет права доступа x, то файл можно запустить bash testsc /etc

***Однострочные скрипты***

● Разделитель команд — ";"

● Удобны для выполнения в терминале

● Применимы все основные конструкции из bash

● apt update; apt upgrade; echo "Upgrade complete!"

***Условия if и ветвления***

Синтаксис:

if [ выражение ]

then

Действия, если выражение истинно

else

Действия в противоположном случае

fi

Пример:

if [ -e file\_name ]

then

echo "true"

else

echo "false"

fi

***Варианты условий (информацию можно посмотреть введя команду man test)***

*Операции сравнения строк*

● = или == возвращает true (истина), если строки равны

● != возвращает true (истина), если строки не равны

● -z возвращает true (истина), если строка пуста

● -n возвращает true (истина), если строка не пуста

*Операции проверки файлов*

● -e возвращает true (истина), если файл существует (exists)

● -d возвращает true (истина), если каталог существует (directory)

*Операции сравнения целых чисел (наиболее используемые)*

● -eq возвращает true (истина), если числа равны (equals)

● -ne возвращает true (истина), если числа не равны (not equal)

exit 2 – команда выхода из скрипта с кодом ошибки 2

***Цикл for***

*Синтаксис:*

for имя\_переменной in значения

do

тело\_цикла

done

*Примеры:*

for h in {01..24}

do

echo $h

done

for (( c=1; c<=5; c++ ))

do

echo "Попытка номер $c"

done

***Цикл while***

*Синтаксис:*

while [ условие ]

do

Тело\_цикла

done

*Пример:*

c=10

while [ $c -ge 0 ]

do

echo "Test"

let "c = c - 1"

done

type ls – покажет тип команды ls

type -a ls – покажет более подробную информацию о команде

which ls – покажет путь к команде