Лабораторная работа 4 Знакомство с Docker

Цель работы: Изучить основы работы с Docker, основные команды, создание и запуск контейнеров, запуск и управление нескольких контейнеров, Регистрация на DockerHub. Подготовить отчет о ходе выполнения работы.

1. Установить Docker и VirtualBox в ОС с Unix системой (возможно, Вам потребуется добавить пользователя в группу docker, чтобы вызвать эту команду без sudo)
2. Все действия выполнять в Терминале:
    - $ docker --version
    - $ docker run Hello-**Ваша_Фамилия**
3. Скачать и запустить контейнер с приложением Apache2 или Nginx
    - Назначить порт 8080
    - Установить рабочий каталог на хосте /home/www/html
    - В контейнере разместить страницу по умолчанию содержащую информацию о вас - **Ваша_Фамилия**
4. Скачать и запустить контейнер с приложением MariaDB
    - Установить рабочий каталог /home/DB
    - Назначить порт по умолчанию
    - Через терминал подключиться к базе данных и создать БД с вашей фамилией
5. Скачать и запустить контейнер с приложением NextCloud (скачать с DockerHub)
    - Установить рабочий каталог на хосте /home/www/NextCloud
    - Назначить порт 8088
    - Зарегистрировать пользователей: Вы и ФИО преподавателя
6. Скачать и запустить контейнер с приложением phpvirtualbox
7. Собрать новый проект Docker Compose, который запускает NextCloud связанный с web-сервером (Apache2 или Nginx) и БД (MariaDB)
    - Назначить порт 2022 для доступа к NextCloud
8. Зарегистрироваться на DockerHub https://hub.docker.com
9. Выложить свой проект на DockerHub.
10. Продемонстрировать запуск приложений с DockerHub, для этого скачайте и запустите игру 2048 или Mine
11. Описать основные команды используемые в работе

Команды для управления контейнерами

Команда для управления контейнерами:

docker container *название команды*

Названия команд, которые используются для работы с контейнерами:

create	создает контейнер из выбранного образа
start	активирует уже существующий контейнер
run	создает новый контейнер и сразу включает его
ls	отображает все существующие контейнеры
inspect	подробнее рассказывает о выбранном контейнере
logs	выводит в консоль логи (то есть журнал событий)
stop	пытается остановить выбранный контейнер, отправив ему сигнал **SIGTERM**, требующий завершить всю активность и сохранить пользовательские данные. Если ответ занимает слишком много времени, то следом посылает сигнал **SIGKILL**, чтобы «убить» процесс без сохранения данных
kill	выполняет ту же задачу, что и предыдущая команда, но пропускает шаг с отправкой **SIGTERM**. Сразу выключает контейнер, игнорируя сохранение пользовательских данных
rm	удаляет выбранный контейнер (он должен быть выключен, чтобы команда сработала)

Команды для управления образами

docker image *название команды*

- Названия команд, которые используются для работы с образами:

build	собирает образ с нуля
push	отправляет образ в реестр
pull	загружает готовый образ с необходимыми для работы параметрами
ls	показывает все существующие образы
history	показывает каждый слой образа в ретроспективе, отображая ряд полезных сведений.
inspect	рассказывает все, что известно об образе, включая данные, касающиеся отдельных слоев.
rm	удаляет образ Docker из системы.
images	списком показывает все образы Docker, найденные на диске.

Описание инструкций Dockerfile

Инструкция	Описание	Пример
FROM	Указывает, какой базовый образ нужно использовать. Обязательная инструкция для Dockerfile	FROM ubuntu:16.04
MAINTAINER	Автор образа.	MAINTAINER DMosk <master@dmosk.ru>
RUN	Выполняет команду в новом слое при построении образа.	RUN apt-get install python
CMD	Запускает команду каждый раз при запуске контейнера. Может быть вызвана только один раз. Если в Dockerfile указать несколько таких инструкций, то выполнена будет последняя.	CMD ["openvpn"]
LABEL	Добавляет метаданные.	LABEL version="2"
EXPOSE	Указывает, какой порт должно использовать приложение внутри контейнера.	EXPOSE 8080
ENV	Задает переменные окружения в образе.	ENV PGPASSWORD pass
ADD	Добавляет файлы/папки из текущего окружения в образ. Если в качестве копируемого файла указать архив, то он будет добавлен в образ в распакованном виде. Также в качестве источника принимает URL.	ADD /root/.ssh/{id_rsa,id_rsa.pub} /root/.ssh/
COPY	Также как и ADD добавляет файлы в образ, но обладает меньшими функциями — не принимает URL и не распаковывает архивы. Рекомендован для использования в случаях, где не требуются возможности ADD	COPY ./mypasswd /root/

Инструкция	Описание	Пример
	или когда нужно перенести архив, как архив.	
ENTRYPOINT	Указывает команду, которой будет передаваться параметр при запуске контейнера.	ENTRYPOINT ["/sbin/apache2"]
VOLUME	Добавляет том в контейнер.	VOLUME ["/opt/myapp"]
USER	Задает пользователя, от которого будет запущен образ.	USER user:group
WORKDIR	Можно задать каталог, откуда будут запускаться команды ENTRYPOINT и CMD.	WORKDIR /opt/apps
ARG	Создает переменную, которую может использовать сборщик.	ARG folder=/opt/apps WORKDIR $folder
ONBUILD	Действия, которые выполняются, если наш образ используется как базовый для другой сборки.	ONBUILD ADD . /app/src
STOPSIGNAL	Переопределяет сигнал SIGTERM для завершения контейнера.	STOPSIGNAL SIGINT
HEALTHCHECK	Команда, которая будет проверять работоспособность контейнера.	HEALTHCHECK --interval=5m --timeout=3s CMD curl -f http://localhost/ || exit 1
SHELL	Позволяет заменить стандартную оболочку для выполнения команд на пользовательскую.	SHELL ["/bin/sh", "-c"]