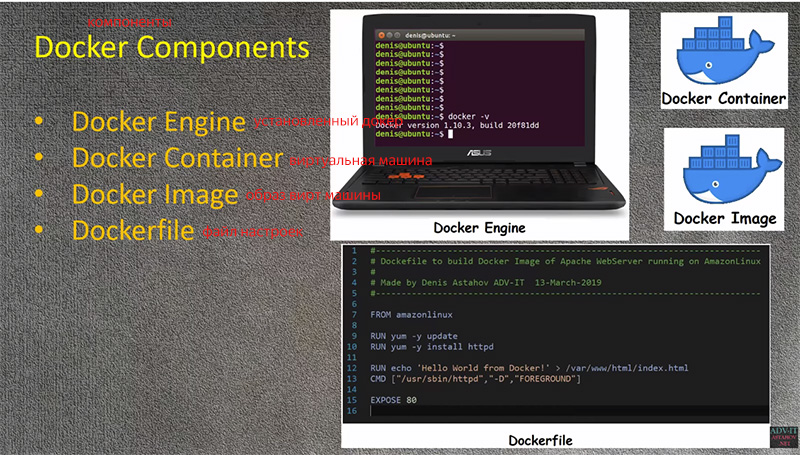
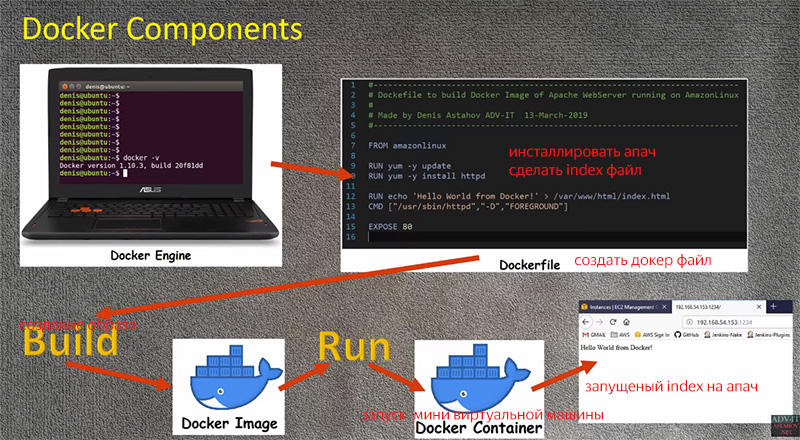
**КОМПОНЕНЕТЫ ДОКЕРА:**



Контейнеры создаются из Image

**ПРИНЦИП РАБОТЫ:**

****

**УСТАНОВКА DOCKER**

apt install apt-transport-https

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt update

sudo apt install docker-ce

sudo usermod -aG docker $USER #дать права юзеру

**КОМАНДЫ**

docker images #просмотр образов

docker run #запустить образ

<https://hub.docker.com/search?q=&type=image> # DockerHub хранилище images

docker search tomcat # поиск имеджей в командной строке

docker pull tomcat #добавить image томкат

docker push x3m5555/k8s #залить имедж в репозиторий docker hub

docker ps #просмотр запущенных сейчас контейнеров

docker ps –a #просмотр всех запущенных контейнеров ранее

docker run –it –p 1234:8080 tomcat

#запуск (-it интерактивно в текущей консоле) (-p 1234:8080 перенаправить с порта по умолчанию в томкат 8080 на порт 1234)

docker run –d –p 454:8080 tomcat #запуск томкат с перенаправлением порта с 8080 на 454 НЕ интерактивно (в фоне)

#можно несколько контейнеров запустить на разных портах

docker run --restart=always -d -p 80:80 and4 # запуск после рестарта сервера

docker rmi tomcat #удалить image томкат, может не удалится т.к. есть контейнер томкат сначала его удалить по ID

docker rm 934573958 #удалить контейнер томкат

#можно удалить форс командой -f

docker rm –v $(docker ps –aq –f status=exited) #удалить все остановленные контейнеры

docker history image #просмотр слоев в имедж

docker inspect container\_name #сведения о контейнере

**Создание докер файла**

Dockerfile с большой буквы, файл с одержимым в папке с этим ворд файлом

docker build –t andrey:ver1 . #создается image andrey c тегом ver1 ( . точка в конце) в текущей директории

docker tag andrey:ver1 andrey:ver2 #копия имеджа с с другим именем и тегом

docker exec -it 0f932f540e1a /bin/bash #залогиниться на контейнер

docker commit 0f932f540e1a andrey:ver3 #создать image с именем andrey:ver3 из контейнера с ID 0f932f540e1a

docker save <image\_name>:<tag> -o /path/to/image.tar #копировать имедж в файл

docker image load -i /path/to/image.tar #на другом сервере скопировать этот архив

**DOCKER VOLUME работа с данными**

https://habr.com/ru/company/ruvds/blog/441574/

**Инструкции Dockerfile**

FROM — задаёт базовый (родительский) образ.

LABEL — описывает метаданные. Например — сведения о том, кто создал и поддерживает образ.

ENV — устанавливает постоянные переменные среды.

RUN — выполняет команду и создаёт слой образа. Используется для установки в контейнер пакетов.

COPY — копирует в контейнер файлы и папки.

ADD — копирует файлы и папки в контейнер, может распаковывать локальные .tar-файлы.

CMD — описывает команду с аргументами, которую нужно выполнить когда контейнер будет запущен. Аргументы могут быть переопределены при запуске контейнера. В файле может присутствовать лишь одна инструкция CMD.

WORKDIR — задаёт рабочую директорию для следующей инструкции.

ARG — задаёт переменные для передачи Docker во время сборки образа.

ENTRYPOINT — предоставляет команду с аргументами для вызова во время выполнения контейнера. Аргументы не переопределяются.

EXPOSE — указывает на необходимость открыть порт.

VOLUME — создаёт точку монтирования для работы с постоянным хранилищем.

**DOCKER-COMPOSE**

УСТАНОВКА DOCKER-COMPOSE:<https://docs.docker.com/compose/install/>

Запуск файлом docker-compose.yml

Из папки с файлом: docker-compose up –d (-d запуск в фоне)

Пример содержимого, запускается два образа в соединении с друг другом:

*version: '3.1'*

*services:*

*db:*

*image: mariadb*

*# build: ./db #так можно запустить dockerfile из папки ./db. Либо команда image, либо build*

*restart: always*

*environment:*

*MYSQL\_ROOT\_PASSWORD: boltanka*

*volumes:*

*- ./databases:/var/lib/mysql*

*adminer:*

*image: adminer*

*# build: ./adminer #так можно запустить dockerfile из папки ./adminer*

*restart: always*

*ports:*

*- 6080:8080*