\*\*\*LAYOTS OF CONTAINER\*\*\*

\*\*Dockerfiles instruction\*\*

FROM -задает базовый образ

LABEL -добавление методанных в образ

ENV -задает постоянный переменные образа

ARG -задание переменных во время сборки образа

RUN -выполнение команды Linux

COPY -копирование файлов и папок

ADD -копирование файлов и папок + распаковка архивов

WORKDIR -задание рабочей директории

CMD -задание команды, выполняющейся при запуске образа (АРГУМЕНТЫ МОЖНО ПЕРЕОПРЕДЕЛИТЬ ПРИ ЗАПУСКЕ)

ENTRYPOINT - задание команды, выполняющейся при запуске образа (АРГУМЕНТЫ НЕЛЬЗЯ ПЕРЕОПРЕДЕЛИТЬ ПРИ ЗАПУСКЕ)

EXPOSE -открытие порта в образ/контейнер

VOLUME -создание точки монтирования указывает место в системе для хранения данных

\*\*\*CREATE DOCKERFILE\*\*\*

touch Dockerfile -создание докер файла с инструкцией

FROM ubuntu:22.10 -используем убунту версии 22.10

RUN apt-get update -обновляем все пакеты

RUN apt-get install –y cowsay -устанавливаем приложение –у указывает на то что установка будет осуществлена без подтверждения

RUN ln –s /usr/games/cowsay /usr/bin/cowsay -создание символьной ссылки для того чтобы приложение можно было запускать без указания пути

CMD [‘cowsay’] -создание точки входа

docker build –t cowsaytest . -создание контейнера с именем cowsay, «.» означает что файл dockerfile находится в настоящей директории

docker run –it cowsaytest bash -заходим внутрь контейнера

ps –aux -проверяем запущенные процессы

hostname -провоеряем хостнейм

cowsay ‘Hello’ -теперь можем убедиться что все работает

exit

docker run cowsaytest cowsay “Hello” -запускаем приложение прямо не заходя в контейнер

docker images -просмотр всех образов наш весит 155Мб

\*\*\*OPTIMIZATION DOCKERFILE\*\*\*

//Through merging//

FROM ubuntu:22.10

RUN apt-get update && \ -объединили в один слой

apt-get install –y cowsay && \ -объединили в один слой

ln –s /usr/games/cowsay /usr/bin/cowsay && \ -объединили в один слой

rm –rf /var/lib/apt/lists/\* -удаление кэша всех обновлений

CMD [‘cowsay’]

docker images -просмотр всех образов наш весит 118Мб

docker run cowsaytest-1 cowsay 'Hello' -все работает

//Through merging and changing Ubuntu on alpine//

FROM alpine

RUN apk update \

&& apk add cowsay \

--update-cache \

--repository https://alpine.global.ssl.fastly.net/alpine/edge/community \

--repository https://alpine.global.ssl.fastly.net/alpine/edge/main \

--repository https://dl-3.alpinelinux.org/alpine/edge/testing \

&& rm -rf /var/cache/apk/\*

CMD ['cowsay']

docker images -просмотр всех образов наш весит 42.7Мб

docker run cowsaytest-1 cowsay 'Hello' -все работает

FROM alpine

RUN apk update \

&& apk add cowsay \

--update-cache \

--repository https://alpine.global.ssl.fastly.net/alpine/edge/community \

--repository https://alpine.global.ssl.fastly.net/alpine/edge/main \

--repository https://dl-3.alpinelinux.org/alpine/edge/testing \

&& rm -rf /var/cache/apk/\*

RUN echo “123” > /test -подобные действия нужно делать в последнем слое для оптимизации (так образ собирается быстрее)

CMD ['cowsay']

\*docker run --name testmariadb-1 -e MARIADB\_ROOT\_PASSWORD=test123 -d testmariadb-1 -запуск после билда и остановки

\*Ctr+q / Ctr+p -выйти