\*\*\*INSTALL DOCKER\*\*\*

https://docs.docker.com/engine/install/ubuntu/

-for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done

-sudo apt-get update

-sudo apt-get install ca-certificates curl gnupg

-sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

-echo \

"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \

"$(. /etc/os-release && echo "$VERSION\_CODENAME")" stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

-sudo apt-get update

-sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

-sudo docker run hello-world

\*\*\*КОМАНДЫ DOCKER\*\*\*

-sudo docker pull Ubuntu -установка образа

-sudo docker pull ubuntu:18.04 -установка определенной версии образа

-docker images -просмотр всех образов

-sudo docker rmi ubuntu:18.04 -удаление образа по имени

-sudo docker rmi $(docker images -aq) -удаление образов

-sudo docker rmi $(docker images -aq) –force -принудительное удаление образов

-sudo docker ps -просмотр запущенных контейнеров

-sudo docker ps –a -просмотр контейнеров доступных в системе

-sudo docker run -it ubuntu bash -it интерактивный режим запускаем bash в котнейнере ubuntu входим в контейнер root@a61300ca1712:/#

-sudo docker run -it -h GeekBrains ubuntu bash -запускаем bash в котнейнере ubuntu и присваиваем ему имя GeekBrains

-sudo docker run -it ubuntu date -запуск времени

-sudo docker run --publish 8080:80 nginx -запуск nginx

-sudo docker run --publish 8080:80 nginx –d -запуск nginx в режиме демона(в фоне)

-sudo curl127.0.0.1:8080 -проверка запущен ли контейнер, если есть доступ то работает

-sudo docker ps –a -просмотр контейнеров

-sudo docker start GB(name of conteiner) -запуск контейнера по имени (если нет никакой задачи на выполнение, то контейнер тут же остановится)

-sudo docker exec -it c3b8ca76c133 bash -зайти внутрь контейнера по id

-sudo docker run ubuntu sleep 1200 -запуск задачи в контейнере

-sudo docker rm GB(name of conteiner) -удаление контейнера по имени

-sudo docker rm c3b8ca76c133 -удаление контейнера по id

-sudo rm $(docker ps -a -q) -удаление всех контейнеров

-sudo docker ps –q -вывод id толькозапущенных контейнеров

-sudo docker ps -a –q -вывод id всех контейнров доступных в системе

-sudo docker stop a3jrw4kj5b23kj(id conteiner) -остановка контейнера по его id

-sudo docker system df -просмотр занятогот места на диске

-sudo docker system prune -af -очистка места от образов

\*\*\*РАБОТА С КОНТЕЙНЕРАМИ\*\*\*

-mkdir linkfolder -создаем папку

-echo 123456 > test.txt -создаем файл test.txt и добавляем туда 123456

-docker run -it -v ./linkfolder:/linkfolder ubuntu:22.10 bash -линкуем созданную папку в контейнер

-ls –l -проверяем что папка linkfolder и все ее файлы есть внутри контейнера

-echo 7890 >> linkfolder/test.txt -добавляем запись в файл. файл изменится и в оригинальном расположении

-cat test.txt -читаем файл test.txt и убеждаемся что все работает

\*\*\*РАБОТА С MARIA.DB\*\*\*

-mkdir mariadb -создаем папку для линкования

-docker run --name test-mariadb -e MARIADB\_ROOT\_PASSWORD=test123 -v ./mariadb:/var/lib/mysql -d mariadb:10.10.2 -запускаем 'run mariadb', называем наш котнтейнер '--name test-mariadb', задаем пароль '-e MARIADB\_ROOT\_PASSWORD=test123', линкуем папку '-v ./mariadb:/var/lib/mysql', запускаем контейнре в 'detouch' бэкграундрежиме '-d mariadb:10.10.2'

75f68afc85b35ae52f98769b76e0a09bce0032cc2ebecf4b7f4e68f425a305c7 –подобная надпись подтверждает запуск контейнера

-ls -l mariadb/ -проверяем что в папке появились файлы базы данных

-docker ps -смотрим id контейнера mariadb

-docker exec –it 75f68afc85b3 bash -заходим внутрь контейнера

-mysql –u root –p -чтобы зайти внутрь базы данных. После этого терминал попросит пароль ---который мы вводили ранее MARIADB\_ROOT\_PASSWORD=test123

-show databases -вывод баз данных

-create database GB; -создаем базу данных

-exit

-ls -l mariadb/ -проверяем что в папке появились файлы базы данных

\*\*\*РАБОТА С MYPHPADMIN \*\*\*

-docker run --name my-phpmyadmin -d --link test-mariadb:db -p 8081:80 phpmyadmin/phpmyadmin -запускаем контейнер в режиме detouch

-docker ps -смотрим запущенные контейнеры и убеждаемся что работает phpMyAdmin

-через браузер на виртуальной машине заходим на адрес http://127.0.0.1:8081, вводим имя пользователя root пароль test123