# **--- Docker and Docker Hub ---**

**Doker** - технология упаковки , доставки и развертывания приложения на любом устройстве в виде контейнера. Главаная задача doker запустить приложение на любом устройстве. Не нужно настраивать приложения для каждого устройства. Имеет один механиз запуска для каждого приложения

**Виртуализация** - для запуска Doker в Windows нужно включать виртуализацию

**Контейнер** - виртуальные машины, которые хранит внутри себя код приложения, системные инструменты, среду выполнения , библиотеки, файлы конфигурации. Создан для того, чтоб приложение могло работать на любом устройстве, где установлен doker. Использует ядро операционой системы той на которой работает наше приложение. Тоже что и виртуальная машина, но намного легче весит и работает на устройстве на котором запущена.

**Doker Файл (Doker File) -> Doker Image (Образ Приложения) -> Контейнер**

**Doker File (Сценарий)** - набор записанных комманд, который указывает что нужно ложить один в doker образ

**Docker Compose** - повзволяет запускать сразу нескольок контейнеров одновременно. Также связывает ети контейнеры между собой

**Doker Образ** или **Doker Image** - готовое к запуску приложение, которое внутри хранить весь код, инструменты, библиотети и настройки. Doker Образ хранит внутри себя другой Doker Образ, который являеться основой нашего приложения

**Doker Контейнер** - запущений екзепляр нашого приложения

**Сервис** - docker контейнер который используеться в docker-compose. Cервис изолирован от других и работает независимо от других

**Docker Registry** - сервер для хранение docker images (докер образов). В нашем случае ето Docker Hub

**Docker Deаmon** - название служби с помощью которой мы пользуемся функциями докер и взаемодействуем с его обьектами

**.dockerignore** - записываев папки и файлы, которые не должны попадать в docker imege проекта

**Переменные Окружения** - переменные, которые изменяют свое значение во время создания виртуального окружение на сервере. Ети переменные имеют указанные значение только при запущеном виртульном окружении. Такие переменные как SECRET\_KEY , ключи для баз данны (пользователи, пароли) , токены , адресса електронных почт

**Маппинг (Mapping)** - возможность обращаться пользователея сайта или другим сервисами к контейнерам из вне. По-умолчанию docker контейнеры запускаються изолировано, поетому нужно связывать его с локальными сервисами такими как PostgreSQL , Nginx , Redis , Celery

**DokerHub** - облако в котором можна хранить doker образы для запуска приложений, или скачивать

**Volume -** позволяет сохранять данные даже после удаление контейнера. Перекидывает данные из нашего локального проекта в docker контейнер