??????

Образ - контейнер только для чтения, содержащий все для запуска (минимальная ОС, приложение, зависимости, метаданные)
Образ состоит из слоев, существующие слои общие для использования и могут быть в разных образах.
Образ - manifest файл, в котором список слоев и метаданных.
Слой - целостный объект.
Каждая команда в dockerfile, модифицирующая файловую систему контейнера, создает новый слой.
Нельзя удалить образ, пока последний контейнер не будет удален.
Образы обычно хранятся в /var/lib/docker/<storage-driver>
!Нужно достаточно места в /var!
У образа есть hash (digest), у каждого из слоев есть digest
Параметр distribution hash - hash сжатого образа или образа для размещения в hub
Для поддержки разных архитектур и платформ для одного названия (напр. golang:latest) используются manifest list и manifests
Для каждого контейнера создается новый слой чтения/записи

Команды работы с образами

Базовая команда	Доп. парам.	Описание
docker images		Список установленных образов
	-q	только отображение ID контейнеров
	--digests	добавляет столбец с хэшем образа
	dangling: true/false	Образы без тега. Происходит если при создании нового образа сохраняется тег старого. У старого обнуляется тэг, у нового остается.
	before:	имя/идентификатор аргумент, возвращает все созданные до него.
	since:	возвращает созданные после
	label: <label>	Фильтрует на основе наличия метки или ярлыка и значения. Команда docker images не отображает метки в своих выходных данных.
docker search	<слово>	поиск в dockerhub по слову <слово>

Базовая команда	Доп. парам.	Описание
	--filter ""	Доп. фильтр `docker search alpine --filter "is-official=true"`
	--limit число	кол-во выдачи, максимум 100

docker image pull
Синоним:
docker pull
	<repository>:<tag>	Загрузка образа из репозитория repository:
		• В случае официального образа на docker hub: одно слово
		• В случае неофициального образа на docker hub: слово/слово
		`docker pull nigelpoulton/tu-demo:v2`
		• В случае внешнего хранилища: адрес хранилища без http(s)
		`docker pull gcr.io/google-containers/git-sync:v3.1.5`

	<digest>	Хэш образа
docker image push Синоним: **docker push**	<repository>:<tag>|<digest>	Загрузка образа на репозиторий, доп. параметры аналогичны pull
docker image prune	<repository>:<tag>	удаление образа из локального хранилища
	-a	удаление всех неиспользуемых контейнерами образов

docker build
docker image build
	-t тег образа. обычно имя:версия
	<путь к dockerfile> - обязательный
		Создание образа из dockerfile
		`docker build -t myapp:1.0 .`
		`docker build -t ${PROJECT_NAME}_db:${VERSION} ./database`

Базовая команда	Доп. парам.	Описание
docker buildx		создание образа с поддержкой мультиплатформы
docker save	-o имя_файла.tar имя_образа:тег	Сохранение существующего образа в файл `docker save -o имя_файла.tar имя_образа:тег`
docker load	-i имя_файла.tar	Загрузка образа из файла `docker load -i имя_файла.tar`
docker image tag **docker tag**		Создать тэг TARGET_IMAGE связанный с SOURCE_IMAGE Это нужно только для публикации образа в dockerhub Связывает имя репозитория и тега с заданным образом. `docker tag ddd-book:ch8.1 nigelpoulton/ddd-book:ch8.1`
docker manifest	inspect <image>	информация по манифесту для образа
	create <image>	создание манифеста для образа
docker rmi		Удаляет заданный образ или несколько образов. Удаление всех образов: `docker rmi $(docker images -q) -f` Эта команда удаляет старые теги без удаления образа, если есть еще образы, ссылающиеся на этот образ.
docker inspect <имя или ID образа>		информация по образу
docker info | grep Storage		При использовании docker pull или docker-compose up -d образы размещаются в /var/lib/docker/overlay2

Revision #10
Created 24 June 2024 15:43:14 by Admin
Updated 28 September 2026 07:01:05 by Admin