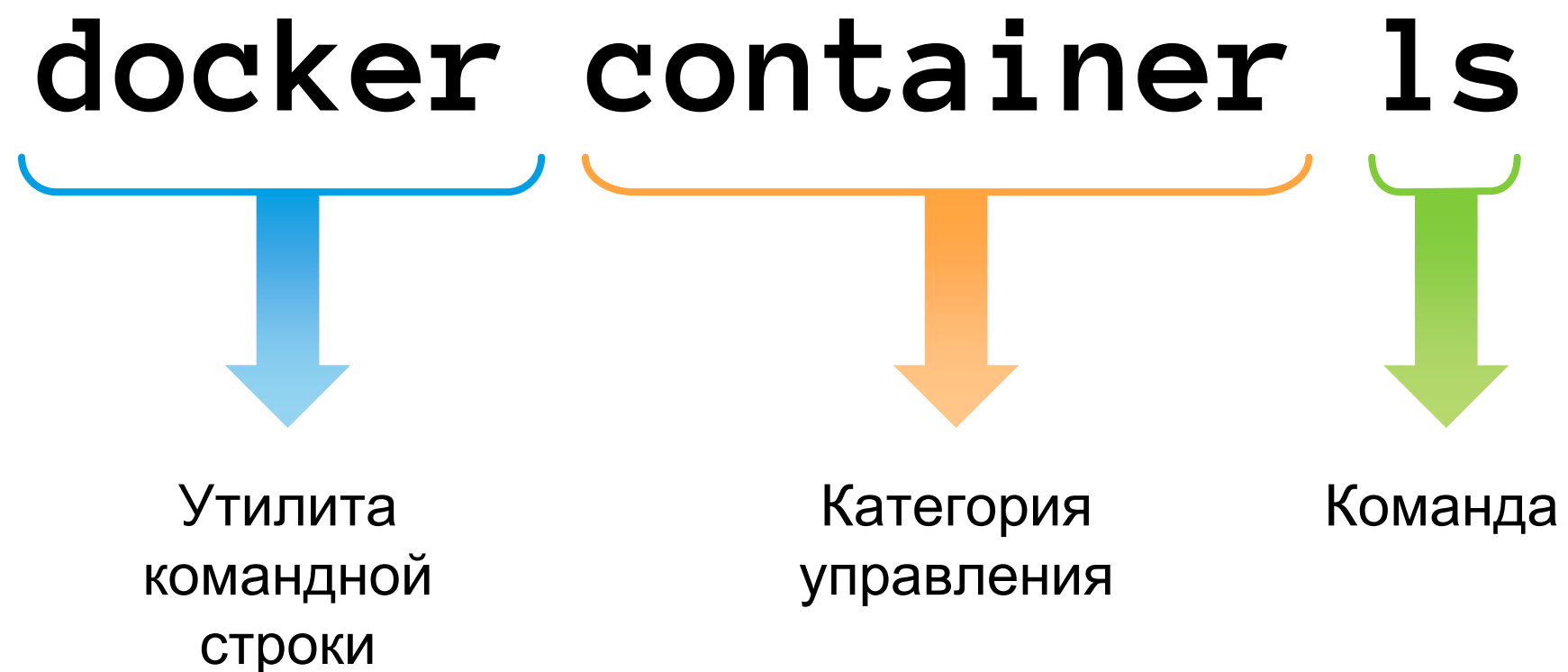


Docker:

Интерфейс командной строки и основные команды

Структура типичной команды docker



Команда docker с аргументом

Docker container rm my-container

Утилита
командной
строки

Категория
управления

Команда

Аргумент
(имя/id)

Старый и новый синтаксис

`docker ps`

=

`docker container ls`

`docker images`

=

`docker image ls`

`docker rmi`

=

`docker image rm`

Категории команд

Работа с
контейнерами

Работа с
образами

Разное

Первые шаги

```
docker container create my_repo/my_image:my_tag
```

- `my_repo` – указание на конкретный docker registry;
- `my_image` – имя образа;
- `my_tag` – тэг для образа;

По умолчанию после установки docker используется репозиторий **Docker Hub** (<https://hub.docker.com/>).

По умолчанию используется тэг `latest` – он годится для обучения и каких-то экспериментов, но в серьёзной работе советуем его избегать.

Первые шаги

- `docker container start my_container` – запустить контейнер. Для перезапуска уже запущенного контейнера есть логичное `docker container restart`;
- `docker run my_image` – очень распространенная и очень комплексная команда. Она позволяет сразу получить образ из registry (если нужного образа нет локально), создать контейнер из образа и запустить его.

Первые шаги

```
docker container run -i -t --rm -p 8080:80 my_image
```

- **-i** – интерактивно, то есть STDIN будет открытым;
- **-t** – выделить псевдо-терминал Linux (tty), который будет соединен с STDIN и STDOUT. **-i -t** можно указать как **-it**;
- **--rm** – удаление контейнера, когда он завершит работу;
- **-p** – указание порта. Правило: какое число ближе к образу `my_image`, то к нему и относится, то есть это порт контейнера;
- **-d** – сокращение от **--detach** – запустить контейнер в фоновом режиме.

Проверка статуса

- `docker container ls` (или `docker ps`) – список запущенных контейнеров.

Полезные опции:

- `-a` – вывести все контейнеры в терминал, а не только запущенные;
- `-s` – показать размер каждого контейнера;
- `docker container inspect my_container` – много полезной информации о контейнере;
- `docker container logs my_container` – логи контейнера.

Завершение работы

- `docker container stop my_container` – выключение контейнера(ов). На завершение работы процессов по умолчанию даётся 10 секунд;
- `docker container kill my_container` – принудительное завершение контейнера;
- `docker container kill $(docker ps -q)` – убить все контейнеры разом;
- `docker container rm my_container` – удаление остановленного контейнера. Удалить запущенный контейнер не получится, нужно его сначала остановить;
- `docker container rm $(docker ps -a -q)` - удалить все контейнеры.

Разработка образов

- `docker image build -t my_repo/my_image:my_tag .` – собрать образ;
- `-t` – присвоить тэг (то, что записано после двоеточия);
- `my_repo` – имя репозитория. По умолчанию используется бесплатный docker hub;

Обратите внимание на точку в конце команды.

- `docker image push my_repo/my_image:my_tag` – публикация собранного образа в registry. Перед тем как делать push, нужно войти в registry.

Работа с registry

- `docker login` – вход в registry, если вы решили публиковать образ на Docker Hub или где-то ещё;
- `docker search` – поиск по registry;
- `docker pull` – получение образа из registry без других действий над ним.

Изучение образов

- `docker image ls` - список образов;
- `docker image history my_image` – просмотр истории промежуточных образов;
- `docker image inspect my_image` – много различной информации по образу, включая слои, из которых он состоит.

Удаление образов

- `docker image rm my_image` (или `docker rmi my_image`) – удаляет образ(ы);
- `docker image rm $(docker images -a -q)` – комбинация из двух команд для удаления вообще всех образов на хосте.

Разное

- `docker version` – просмотр информации о клиенте и сервере docker;
- `docker system prune` – очистка всех неиспользуемых в данный момент контейнеров, сетей, а также не привязанных к контейнерам образов;
- `docker system prune -a --volumes` – ещё более глубокая очистка. Добавляется удаление вообще всех неиспользуемых образов, плюс удаление неиспользуемых томов.

СЛЁРМ



slurm.io



Southbridge



southbridge.io