



# Шпаргалка Docker



## Базовые команды

Запуск контейнера в фоне

```
[~]$ docker run -d nginx
```

Запуск контейнера с шелл

```
[~]$ docker run -it ubuntu bash
```

Запуск контейнера, удаляющегося после установки

```
[~]$ docker run -rm ubuntu bash
```

Экспорт порта контейнера

```
[~]$ docker run -p 80:80 -d nginx
```

Задать имя контейнера

```
[~]$ docker run --name frontend nginx
```

Запустить остановленный контейнер

```
[~]$ docker start frontend
```

Остановить контейнер

```
[~]$ docker stop frontend
```



## Управление контейнерами

Показать работающие контейнеры

```
[~]$ docker ps
```

Показать работающие и остановленные контейнеры

```
[~]$ docker ps -a
```

Показать метаданные контейнера

```
[~]$ docker inspect c31337
```

Показать доступные локально образы

```
[~]$ docker images
```

## Сборка образа

На основе *Dockerfile* в текущей директории

```
[~]$ docker build --tag my-image .
```

"Жесткая" пересборка

```
[~]$ docker build --no-cache my-image .
```

Преобразовать контейнер в образ

```
[~]$ docker commit c31337 my-image
```

Удалить все не используемые образы

```
[~]$ docker rmi $(docker images -q -f "dangling=true")
```

## Отладка

Зайти в работающий контейнер

```
[~]$ docker exec -ti c31337 bash
```

Просмотр логов запущенного контейнера

```
[~]$ docker logs -f c31337
```

Показать экспортированные порты

```
[~]$ docker port c31337
```

Удалить все остановленные контейнеры

```
[~]$ docker rm $(docker ps --filter status=expired -q)
```

Показать все контейнеры со специальной меткой

```
[~]$ docker ps --filter label=traefik.backend
```

Показать конкретные метаданные контейнера

```
[~]$ docker inspect -f '{{ .NetworkSettings.IPAddress }}' \nc31337
```

Запустить ничего не делающий контейнер

```
[~]$ docker run -d busybox /bin/sh -c "while true; do sleep\n2; done"
```

## Диски

Создать локальный диск

```
[~]$ docker volume create --name my-volume
```

Монтирование диска при старте контейнера

```
[~]$ docker run -v my-volume:/data nginx
```

Удалить диск

```
[~]$ docker volume rm my-volume
```

Показать все созданные диски

```
[~]$ docker volume ls
```

## Сети

Создать локальную сеть

```
[~]$ docker network create my-net
```

Подключить контейнер к сети при старте

```
[~]$ docker run -d --net my-net nginx
```

Подключить работающий контейнер к сети

```
[~]$ docker network connect my-net c31337
```

Отключить работающий контейнер от сети

```
[~]$ docker network disconnect my-net c31337
```

## Управление Docker Machine

Запустить Docker Machine

```
[~]$ docker-machine start machine_name
```

Остановить Docker Machine

```
[~]$ docker-machine stop machine_name
```

Настроить Docker на работу с удаленной Docker Machine

```
[~]$ eval "$(docker-machine env machine_name)"
```