

<code>curl -sSL https://get.docker.com/   sh</code>	Установка
<code>docker login</code>	Вход в реестр
<code>docker login localhost:8080</code>	
<code>docker logout</code>	Выход из реестра
<code>docker logout localhost:8080</code>	
<code>docker search nginx</code>	Поиск образа
<code>docker search nginx --filter stars=3 --no-trunc busybox</code>	
<code>docker pull nginx</code>	Pull (выгрузка из реестра) образа
<code>docker pull eon01/nginx localhost:5000/myadmin/nginx</code>	
<code>docker push eon01/nginx</code>	Push (загрузка в реестр) образа
<code>docker push eon01/nginx localhost:5000/myadmin/nginx</code>	
<code>docker create -t -i eon01/infinite --name infinite</code>	Создание контейнера
<code>docker run -it --name infinite -d eon01/infinite</code>	Первый запуск контейнера
<code>docker rename infinite infinity</code>	Переименование контейнера
<code>docker rm infinite</code>	Удаление контейнера
<code>docker update --cpu-shares 512 -m 300M infinite</code>	Обновление контейнера
<code>docker start nginx</code>	Запуск остановленного контейнера
<code>docker stop nginx</code>	Остановка
<code>docker restart nginx</code>	Перезагрузка
<code>docker pause nginx</code>	Пауза (приостановка всех процессов контейнера)
<code>docker unpause nginx</code>	Снятие паузы
<code>docker wait nginx</code>	Блокировка (до остановки контейнера)
<code>docker kill nginx</code>	Отправка SIGKILL (завершающего сигнала)
<code>docker kill -s HUP nginx</code>	Отправка другого сигнала
<code>docker attach nginx</code>	Подключение к существующему контейнеру
<code>docker ps</code>	Работающие контейнеры
<code>docker ps -a</code>	
<code>docker logs infinite</code>	Логи контейнера
<code>docker inspect infinite</code>	
<code>docker inspect --format '{{ .NetworkSettings.IPAddress }}' \$(docker ps -q)</code>	Информация о контейнере
<code>docker events infinite</code>	События контейнера
<code>docker port infinite</code>	Публичные порты
<code>docker top infinite</code>	Выполняющиеся процессы
<code>docker stats infinite</code>	Использование ресурсов
<code>docker diff infinite</code>	Изменения в файлах или директориях файловой системы контейнера
<code>docker images</code>	Список образов

<code>docker build .</code>	
<code>docker build github.com/creack/docker-firefox</code>	
<code>docker build — &lt; Dockerfile</code>	
<code>docker build — &lt; context.tar.gz</code>	Создание образов
<code>docker build -t eon/infinite .</code>	
<code>docker build -f myOtherDockerfile .</code>	
<code>curl example.com/remote/Dockerfile   docker build -f — .</code>	
<code>docker rmi nginx</code>	Удаление образа
<code>docker load &lt; ubuntu.tar.gz</code>	Загрузка репозитория в tar
<code>docker load —input ubuntu.tar</code>	(из файла или стандартного ввода)
<code>docker save busybox &gt; ubuntu.tar</code>	Сохранение образа в tar-архив
<code>docker history</code>	Просмотр истории образа
<code>docker commit nginx</code>	Создание образа из контейнера
<code>docker tag nginx eon01/nginx</code>	Тегирование образа
<code>docker push eon01/nginx</code>	Push (загрузка в реестр) образа
<code>docker network create -d overlay MyOverlayNetwork</code>	
<code>docker network create -d bridge MyBridgeNetwork</code>	
<code>docker network create -d overlay \</code>	
<code>—subnet=192.168.0.0/16 \</code>	
<code>—subnet=192.170.0.0/16 \</code>	
<code>—gateway=192.168.0.100 \</code>	
<code>—gateway=192.170.0.100 \</code>	
<code>—ip-range=192.168.1.0/24 \</code>	
<code>—aux-address=»my-router=192.168.1.5” —aux-</code>	
<code>address=»my-switch=192.168.1.6” \</code>	
<code>—aux-address=»my-printer=192.170.1.5” —aux-</code>	
<code>address=»my-nas=192.170.1.6” \</code>	
<code>MyOverlayNetwork</code>	Создание сети
<code>docker network rm MyOverlayNetwork</code>	Удаление сети
<code>docker network ls</code>	Список сетей
<code>docker network inspect MyOverlayNetwork</code>	Получение информации о сети
<code>docker network connect MyOverlayNetwork nginx</code>	Подключение работающего контейнера к сети
<code>docker run -it -d —network=MyOverlayNetwork nginx</code>	Подключение контейнера к сети при его запуске
<code>docker network disconnect MyOverlayNetwork nginx</code>	Отключение контейнера от сети
<code>docker rm nginx</code>	Удаление работающего контейнера
<code>docker rm -v nginx</code>	Удаление контейнера и его тома (volume)
<code>docker rm \$(docker ps -a -f status=exited -q)</code>	Удаление всех контейнеров со статусом exited

<code>docker container prune</code> <code>docker rm `docker ps -a -q`</code>	Удаление всех остановленных контейнеров
<code>docker container prune —filter «until=24h»</code>	Удаление контейнеров, остановленных более суток назад
<code>docker rmi nginx</code>	Удаление образа
<code>docker image prune</code> <code>docker rmi \$(docker images -f dangling=true -q)</code>	Удаление неиспользуемых (dangling) образов
<code>docker image prune -a</code>	Удаление неиспользуемых (dangling) образов даже с тегами
<code>docker rmi \$(docker images -a -q)</code>	Удаление всех образов
<code>docker rmi -f \$(docker images   grep «^&lt;none&gt;»   awk «{print \$3}»)</code>	Удаление всех образов без тегов
<code>docker stop \$(docker ps -a -q) &amp;&amp; docker rm \$(docker ps -a -q)</code>	Остановка и удаление всех контейнеров
<code>docker volume prune</code> <code>docker volume rm \$(docker volume ls -f dangling=true -q)</code>	Удаление неиспользуемых (dangling) томов
<code>docker volume prune —filter «label!=keep»</code>	Удаление неиспользуемых (dangling) томов по фильтру
<code>docker network prune</code>	Удаление неиспользуемых сетей
<code>docker system prune</code> По умолчанию для Docker 17.06.1+ тома не удаляются. Чтобы удалились и они тоже: <code>docker system prune —volumes</code>	Удаление всех неиспользуемых объектов
<code>curl -ssl https://get.docker.com   bash</code>	Установка Docker Swarm
<code>docker swarm init —advertise-addr 192.168.10.1</code>	Инициализация Swarm
<code>docker swarm join-token worker</code>	Подключение рабочего узла (worker) к Swarm
<code>docker swarm join-token manager</code>	Подключение управляющего узла (manager) к Swarm
<code>docker service ls</code>	Список сервисов
<code>docker node ls</code>	Список узлов
<code>docker service create —name vote -p 8080:80 instavote/vote</code>	Создание сервиса
<code>docker service ps</code>	Список заданий Swarm
<code>docker service scale vote=3</code>	Масштабирование сервиса

```
docker service update --image instavote/vote:movies
vote
docker service update --force --update-parallelism 1
--update-delay 30s nginx
docker service update --update-parallelism 5--update-
delay 2s --image instavote/vote:indent vote
docker service update --limit-cpu 2 nginx
docker service update --replicas=5 nginx
```

Обновление сервиса

—

Источник — [habr.com](https://habr.com)