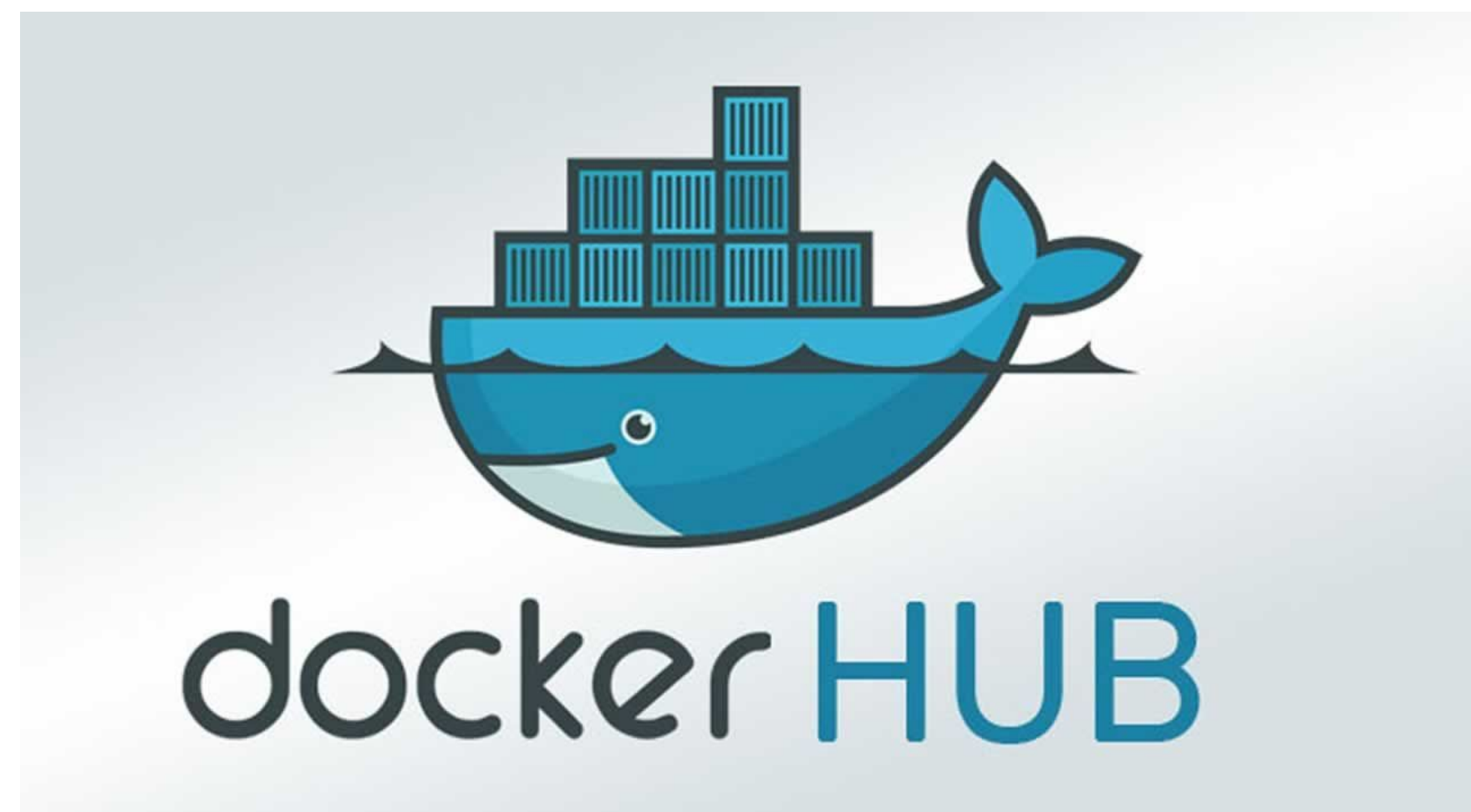


Основные команды

Основные команды для работы Docker

<https://hub.docker.com/>



Запускаем первый контейнер

```
docker run -d -p 80:80 docker/getting-started
```

```
root@hell:/home/kryl# docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
aad63a933944: Pull complete
b14da7a62044: Pull complete
343784d40d66: Pull complete
6f617e610986: Pull complete
Digest: sha256:d2c4fb0641519ea208f20ab03dc40ec2a5a53fdfbcca90bef14f870158ed577
Status: Downloaded newer image for docker/getting-started:latest
01e08b320306440076881f2270521311c380a25d4a800233121edbe818fd25c7
root@hell:/home/kryl#
```

Запускаем первый контейнер

- ID контейнера и статус: Pull complete
- Ключ контейнера Digest:
sha256:d2c4fb0641519ea208f20ab03dc40ec2a5a53fdfbccca90bef14f870158ed577
- Версию - Status: Downloaded newer image for docker/getting-started:latest

Docker --help. Опции ввода

`docker --help`

`docker [OPTIONS] COMMAND`

Options:

- `--config string` Location of client config files (default `"/root/.docker"`)
- `-c, --context string` Name of the context to use to connect to the daemon (overrides `DOCKER_HOST` env var and default context set with `"docker context use"`)
- `-D, --debug` Enable debug mode
- `-H, --host list` Daemon socket(s) to connect to
- `-l, --log-level string` Set the logging level (`"debug"`|`"info"`|`"warn"`|`"error"`|`"fatal"`) (default `"info"`)
- `--tls` Use TLS; implied by `--tlsverify`
- `--tlscacert string` Trust certs signed only by this CA (default `"/root/.docker/ca.pem"`)
- `--tlscert string` Path to TLS certificate file (default `"/root/.docker/cert.pem"`)
- `--tlskey string` Path to TLS key file (default `"/root/.docker/key.pem"`)
- `--tlsverify` Use TLS and verify the remote
- `-v, --version` Print version information and quit

Docker --help. Команды управления

Management Commands:

app*	Docker App (Docker Inc., v0.9.1-beta3)	network	Manage networks
builder	Manage builds	node	Manage Swarm nodes
buildx*	Build with BuildKit (Docker Inc., v0.5.1-docker)	plugin	Manage plugins
config	Manage Docker configs	secret	Manage Docker secrets
container	Manage containers	service	Manage services
context	Manage contexts	stack	Manage Docker stacks
image	Manage images	swarm	Manage Swarm
manifest	Manage Docker image manifests and manifest lists	system	Manage Docker
		trust	Manage trust on Docker images
		volume	Manage volumes

Docker --help. Команды — 1

Commands:

attach — Attach local standard input, output, and error streams to a running container

build — Build an image from a Dockerfile

commit — Create a new image from a container's changes

cp — Copy files/folders between a container and the local filesystem

create — Create a new container

diff — Inspect changes to files or directories on a container's filesystem

events — Get real time events from the server

exec — Run a command in a running container

export — Export a container's filesystem as a tar archive

history — Show the history of an image

images — List images

import — Import the contents from a tarball to create a filesystem image

info — Display system-wide information

inspect — Return low-level information on Docker objects

kill — Kill one or more running containers

Docker --help. Команды — 2

Commands:

load — Load an image from a tar archive or STDIN

login — Log in to a Docker registry

logout — Log out from a Docker registry

logs — Fetch the logs of a container

pause — Pause all processes within one or more containers

port — List port mappings or a specific mapping for the container

ps — List containers

pull — Pull an image or a repository from a registry

push — Push an image or a repository to a registry

rename — Rename a container

restart — Restart one or more containers

rm — Remove one or more containers

rmi — Remove one or more images

run — Run a command in a new container

save — Save one or more images to a tar archive (streamed to STDOUT by default)

Docker --help. Команды — 2

Commands:

search — Search the Docker Hub for images

start — Start one or more stopped containers

stats — Display a live stream of container(s) resource usage statistics

stop — Stop one or more running containers

tag — Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE

top — Display the running processes of a container

unpause — Unpause all processes within one or more containers

update — Update configuration of one or more containers

version — Show the Docker version information

wait — Block until one or more containers stop, then print their exit codes

Манипуляции с Docker-контейнером

- `docker images` — смотрим `images` в наличии и узнаём ID нашего
- `docker images -a` — смотрим `images`, в том числе не активные
- `docker ps` — смотрим запущенные контейнеры
- `docker ps -a` — смотрим список всех контейнеров, включая не активные
- `docker login username darkbenladan --password password` — логинимся по логину и паролю, которые мы создали на hub.docker.com ранее
- `docker save 3c156928aeec > start.tar` — сохраняем локально наш контейнер, теперь мы можем его перенести куда угодно
- `docker rm -f $(docker ps -a -q)` — удалим и принудительно остановим все контейнеры
- `docker load < start.tar` — мы вернули наш контейнер
- `docker images` — давайте убедимся в этом

Ещё немного полезных команд

- `docker run -d -p 80:80 docker/getting-started`
- `docker ps` — узнаем новый ID нашего контейнера
- `docker port 8c71b40dc003` — узнаем порт приложения
- `docker exec -it 8c71b40dc003 sh` — зайдём в терминал нашего контейнера
- `/ # cat /etc/os-release` — узнаем какая OS внутри нашего контейнера
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.11.5
PRETTY_NAME="Alpine Linux v3.11"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://bugs.alpinelinux.org/"
- `exit` — разлогиниться из контейнера
- Так же могут быть полезны команды `docker logs`, `docker stop\start`, `docker logs`

Push в репозиторий

```
docker login -u darkbenladan  
docker tag image ID darkbenladan/dotnet:latest  
docker push darkbenladan/dotnet
```

Итоги

- Узнали как запускать, останавливать, сохранять, поднимать из сохранённого контейнера
- Рассмотрели help и основные команды, ключи
- Научились создавать репозиторий и пушить в него

**Спасибо
за внимание!**