Lesson 3. Advanced tips

Useful CLI commands

Managements commands

```
docker --help
docker image --help
docker container --help
```

`docker attach`

```
docker attach --help
docker run -itd --name=loop --rm busybox \
       sh -c "while [ 0 ]; do echo Next: \$(date); sleep 2; done"
docker logs -f loop
docker attach loop
## Press "CTRL-p CTRL-q" to de-attach
docker run -it --name=interactive --rm busybox \
       sh -c "while [ 0 ]; do echo Next: \$(date); sleep 2; done"
## Press "CTRL-p CTRL-q" to de-attach
```

`docker exec`

```
docker exec --help
docker exec loop ps
docker exec -it loop sh
ps
```

Shortcuts

```
# smooth clean
docker system prune -f # and image, container, etc.
# stop all containers
docker stop $(docker ps -q)
```

Manage data (aka Volumes)

> Read more on https://docs.docker.com/storage/.

`docker run --volume`

Mount a filesystem mount (volume) to the container

```
docker volume --help
docker volume create notebook
# create one note in several new containers
docker run -it --rm \
       --volume notebook:/notebook \
       --workdir /notebook busybox \
       sh -c "echo \$(date): \$(hostname) >> note.\$(hostname)"
# read the notes
docker run -it --rm \
       --volume notebook:/notebook \
       busybox sh -c "cat /notebook/note*"
# mount to local filesystem
docker run -it --rm \
       --volume $(pwd)/notebook:/notebook \
       --workdir /notebook busybox \
       sh -c "echo \$(date): \$(hostname) >> note.\$(hostname)"
# on localhost
cat notebook/note.*
```

`docker run --volumes-from`

Mount volumes from the specified container(s)

```
docker run -itd --rm \
       --volume $(pwd)/notebook:/local \
       --volume notebook:/public \
       --name good busybox
docker run -itd --rm \
       --volumes-from good \
       --name better busybox
docker inspect --format "{{.Mounts}}" good
docker inspect --format "{{.Mounts}}" better
docker attach better
ls -lh /local /public
```

`docker run --mount`

Attach a filesystem mount to the container

```
docker run -it --rm \
       --mount type=bind,source="$(pwd)/notebook",target=/myapp \
       busybox ls -lh /myapp
```

`--volume` vs `--mount`

Key statements:

- As opposed to bind mounts, all options for volumes are available for both `--mount` and `--volume` flags.
- `--mount` will raise an error if a source is absent while `--volume` will create it.
- When using volumes with `service`s, only `--mount` is supported.

Please read https://docs.docker.com/storage/volumes/#choose-the--v-or---mount-flag .

Manage communication (aka Networking)

```
docker network --help
docker network ls
```

The default bridge network is present on all Docker hosts. If you do not specify a different network, new containers are automatically connected to the default bridge network.

> Read more on https://docs.docker.com/network/.

`none` network

```
docker run -it --net=none busybox
ifconfig
```

`bridge` network

```
docker run -it --net=bridge busybox
ifconfig
```

`host` network

```
docker run -it --net=host busybox
ifconfig
```

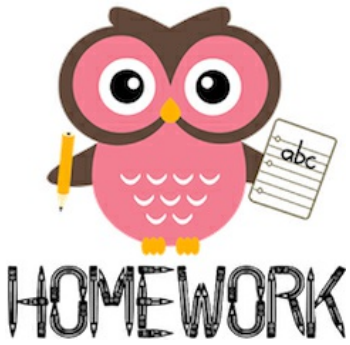
Custom network

```
docker network create --driver=bridge moon
docker network inspect moon
docker run -itd --net=moon --name=alpha busybox
docker run -itd --net=moon --name=beta busybox
docker run -itd --name=fox busybox
docker network inspect moon
docker container inspect alpha
docker container inspect --format {{.NetworkSettings.Networks}} beta
docker container inspect --format {{.NetworkSettings.Networks}} fox

# observe network settings
docker attach alpha
cat /etc/hosts
ping <beta ip>
ping beta
ping fox

# upgrade fox
docker network connect moon fox
docker container inspect --format {{.NetworkSettings.Networks}} fox
docker attach alpha
ping fox
```

Homeworks



Please send the results of homeworks as an email.
Please use the following template:

- *Subject:* `[Docker] Homework 3`
- *To:* `trainer's email`
- *Body:* `your homework as a plain text - NO ATTACHMENTS!!!`

Homework 3.1 (mandatory)

There is a Python application and automated tests for it inside `homework-3.zip`.

You need to create a Docker image for the application and push it (`hits:h3` tag). Also, the application depends on a Redis instance. So, you need to run it from an official image. You have to use custom networks for establishing connectivity between the application and Redis.

Please send for review a `Dockerfile` as well as the commands how to run Redis and your application.

Homework 3.2 (mandatory)

Now there is a new requirement for the application. Although we will destroy containers after the usage, we need to have a backup of Redis's data and application's logs. So, please send an updated version of the commands (use the output of homework 3.1) to achieve this.

Homework 3.3 (optional)

Using the output of homework 3.2, you need to create a Docker image for the automated tests and push it (`hits-at:h3` tag).

Please send for review a `Dockerfile` as well as a set of commands to run Redis, application and automated tests. Please use custom networking.