Übung : Einfaches Image-Management

Erstellt einen Account auf https://hub.docker.com/ (https://hub.docker.com/).

Erstellt, auf http://localhost:32188/ (http://localhost:32188/) ein neues `Terminal` rechts oben mit `New` und logt Euch auf Docker Hub ein:

```
docker login
```

Tagt die zwei vorhanden Images mit Eurem Namen und pusht diese in auf Docker Hub

z.B.:

```
docker tag hello-world <username>/hello-world
docker push <username>/hello-world
```

In [1]:

```
! docker ...
```

```
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default
                           "/home/jovyan/.docker")
  -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
                           "/home/jovyan/.docker/ca.pem")
      --tlscert string     Path to TLS certificate file (default
                           "/home/jovyan/.docker/cert.pem")
      --tlskey string      Path to TLS key file (default
                           "/home/jovyan/.docker/key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  config      Manage Docker configs
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
  system      Manage Docker
  trust       Manage trust on Docker images
  volume      Manage volumes

Commands:
  attach      Attach local standard input, output, and error streams to a ru
nning 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 filesyste
m
  create      Create a new container
  diff        Inspect changes to files or directories on a container's files
ystem
  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 imag
e
  info        Display system-wide information
```

```
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  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 b
y default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistic
s
  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

Run 'docker COMMAND --help' for more information on a command.
```

Docker Images als im «tar»-Format speichern und laden

In [2]:

```
! docker save hello-world -o hello-world.tar
! docker rmi hello-world
! docker image ls
```

```
Untagged: hello-world:latest
Untagged: hello-world@sha256:5f179596a7335398b805f036f7e8561b6f0e32cd30a32f5
e19d17a3cda6cc33d
Untagged: hello-world@sha256:92695bc579f31df7a63da6922075d0666e565ceccad16b5
9c3374d2cf4e8e50e
Deleted: sha256:fce289e99eb9bca977dae136fbe2a82b6b7d4c372474c9235adc1741675f
587e
Deleted: sha256:af0b15c8625bb1938f1d7b17081031f649fd14e6b233688eea3c5483994a
66a3
REPOSITORY              TAG                 IMAGE ID            CREATED
SIZE
hairyhenderson/figlet   latest              9a14a4f852dc        4 days ago
6.2MB
```

In [3]:

```
! ls -ls *.tar
```

```
16 -rw------- 1 jovyan 1000 12800 May 14 13:59 hello-world.tar
```

In [4]:

```
! docker load -i hello-world.tar
! docker image ls
```

```
Loaded image: hello-world:latest/3.584kB
REPOSITORY              TAG                 IMAGE ID            CREATED
SIZE
hairyhenderson/figlet   latest              9a14a4f852dc        4 days ago
6.2MB
hello-world             latest              fce289e99eb9        4 months ago
1.84kB
```

In [ ]:

```

```