

# Labo 2 - Les IDs et la complétion Bash

Les IDs des conteneurs permettent de donner des ordres à un conteneur en spécifiant son id

```
sudo docker images
```

```
[nancre@DOCKER ~]$ sudo docker images
[sudo] Mot de passe de nancre :
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
bash                 latest             39a95ac32011       2 weeks ago        13.1MB
hello-world         latest             bf756fb1ae65       10 months ago      13.3kB
```

Je peux donc consulter les différentes couches qui composent mon image bash en tapant juste le début de son id

```
sudo docker history 39a
```

```
[nancre@DOCKER ~]$ sudo docker history 39
IMAGE                CREATED             CREATED BY          SIZE                COMMENT
39a95ac32011         2 weeks ago        /bin/sh -c #(nop) CMD ["bash"]          0B
<missing>            2 weeks ago        /bin/sh -c #(nop) ENTRYPOINT ["docker-entry... 0B
<missing>            2 weeks ago        /bin/sh -c #(nop) COPY file:651b3bebeba8be91... 212B
<missing>            2 weeks ago        /bin/sh -c set -eux; apk add --no-cache --... 7.57MB
<missing>            2 weeks ago        /bin/sh -c #(nop) ENV _BASH_LATEST_PATCH=18    0B
<missing>            2 weeks ago        /bin/sh -c #(nop) ENV _BASH_BASELINE=5.0       0B
<missing>            2 weeks ago        /bin/sh -c #(nop) ENV _BASH_VERSION=5.0.18    0B
<missing>            2 weeks ago        /bin/sh -c #(nop) CMD ["/bin/sh"]            0B
<missing>            2 weeks ago        /bin/sh -c #(nop) ADD file:f17f65714f703db90... 5.57MB
```

On peut utiliser la complétion dans le shell pour afficher les commandes de docker

```
sudo docker<TAB>
```

```
[nancre@DOCKER ~]$ sudo docker
attach  config  create  exec    image  inspect  logout  pause  pull    rm      search  start  system  unpause  wait
build  container  diff    export  images  kill     logs    plugin  push   rmi     secret  stats  tag      update
builder  context  engine  help    import  load     network  port   rename  run     service  stop   top      version
commit  cp       events  history  info    login    node    ps      restart  save   stack   swarm  trust   volume
```

Si la complétion ne fonctionne pas, il faut voir si le paquet bash-completion est bien installé. Si ce n'est pas le cas, l'installer, se déconnecter de la session et se reconnecter.

En tapant la commande suivante voici la liste des différentes commandes possibles de donner à docker :

```
sudo docker --help
```

```
[[nandre@DOCKER ~]$ sudo docker --help
Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

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

Management Commands:
  builder      Manage builds
  config       Manage Docker configs
  container    Manage containers
  context      Manage contexts
  engine       Manage the docker engine
  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 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
  load         Load an image from a tar archive or STDIN
```

---

Revision #3

Created 2026-04-24 12:43:55 UTC by Nico là

Updated 2026-04-24 13:01:03 UTC by Nico là