EyeSeeTea
ICT for human beings

Malaria DHIS2 dockers

Installation guide

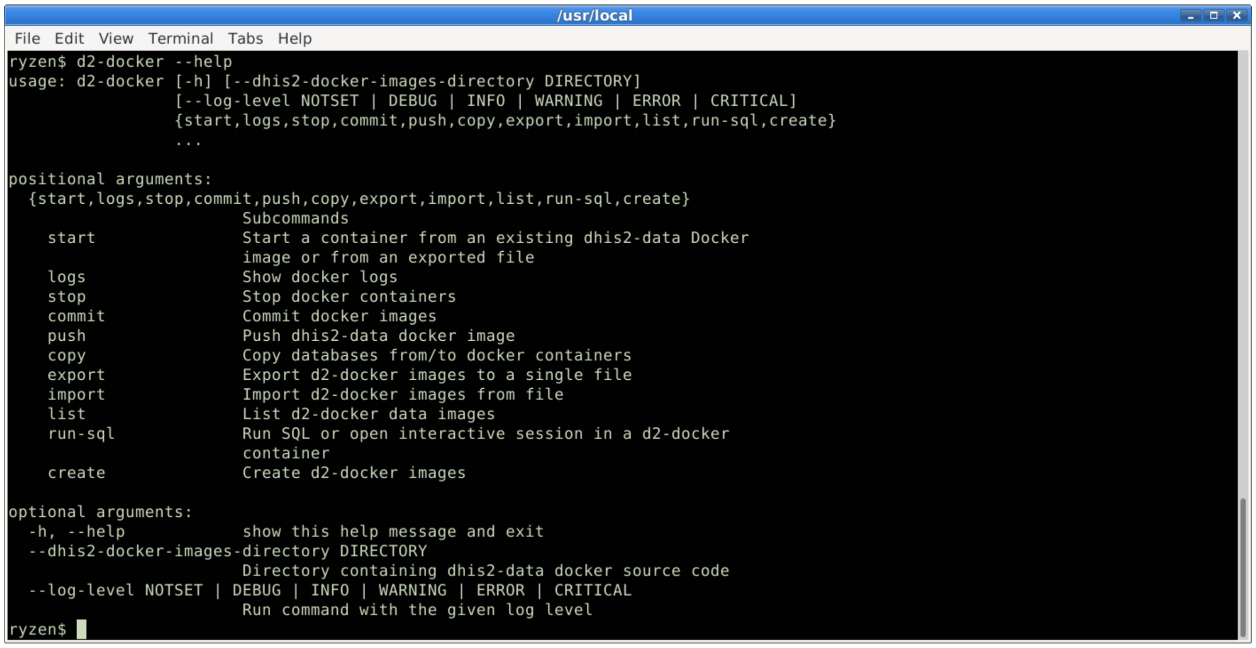
Created: Sep 20, 2019
Last update: May 26, 2020

ABOUT d2-docker

D2-docker is a wrapper over *docker* and *docker-compose* that manages DHIS2 server instances from the command-line. D2-docker is implemented in Python 3 and provides both an executable and a Python module to extend from.

The CLI (command-line interface) resembles that of *docker-compose*, with a set of custom command needed to interact with DHIS2 instances. A user familiar with the docker infrastructure can still use *docker* to interact with the base infrastructure (containers, images, volumes, and so on) should they need to.

License: GNU GPL v3



```
ryzen$ d2-docker --help
usage: d2-docker [-h] [--dhis2-docker-images-directory DIRECTORY]
                 [--log-level NOTSET | DEBUG | INFO | WARNING | ERROR | CRITICAL]
                 {start,logs,stop,commit,push,copy,export,import,list,run-sql,create}
                 ...

positional arguments:
  {start,logs,stop,commit,push,copy,export,import,list,run-sql,create}
                        Subcommands
    start               Start a container from an existing dhis2-data Docker
                        image or from an exported file
    logs                Show docker logs
    stop                Stop docker containers
    commit              Commit docker images
    push                Push dhis2-data docker image
    copy                Copy databases from/to docker containers
    export              Export d2-docker images to a single file
    import              Import d2-docker images from file
    list                List d2-docker data images
    run-sql             Run SQL or open interactive session in a d2-docker
                        container
    create              Create d2-docker images

optional arguments:
  -h, --help            show this help message and exit
  --dhis2-docker-images-directory DIRECTORY
                        Directory containing dhis2-data docker source code
  --log-level NOTSET | DEBUG | INFO | WARNING | ERROR | CRITICAL
                        Run command with the given log level
ryzen$
```

TECHNICAL REQUIREMENTS

D2-docker has been tested on **GNU/Linux** (specifically, Ubuntu 18.04 and Arch Linux), **Windows 10** and **MacOS X** (**10.14, Mojave)**. A typical DHIS2 instance takes around 4-8GB of RAM, so make sure you have that free memory for all the servers you are planning to run simultaneously.

External dependencies:

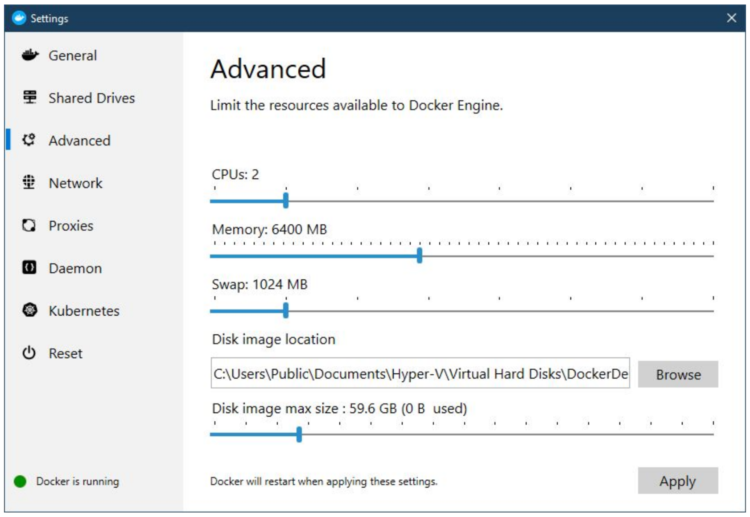
- Python >= 3.5 (with setuptools)
- Docker >= 18
- Docker compose >= 1.17
- RAM memory: At least 4Gb for instance, preferably 8Gb.

Ubuntu 18.04

```
$ sudo apt install docker.io docker-compose python3 python3-setuptools
```

Windows 10

- Install Python: https://www.python.org/downloads
  NOTE: Please ensure that while installing, the checkbox asking for adding Python to PATH is checked
- Install Docker Desktop: https://docs.docker.com/docker-for-windows/install
- Configure Docker Desktop to give more memory to instance: *Settings* -> *Advanced* -> *Memory*: 6400 Mb (modify to meet your needs).



Advanced

Limit the resources available to Docker Engine.

CPUs: 2

Memory: 6400 MB

Swap: 1024 MB

Disk image location

C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\DockerDe

Disk image max size : 59.6 GB (0 B used)

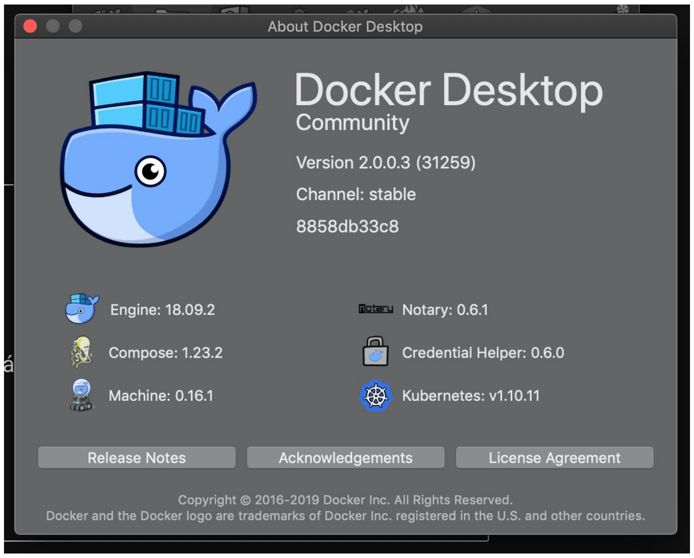
Docker is running

Docker will restart when applying these settings.

NOTE: If your computer has more than 2 CPUs, we encourage you to increase the CPUs to 4, making DHIS2 dockers run more softly

MacOS X

Install Docker Desktop.

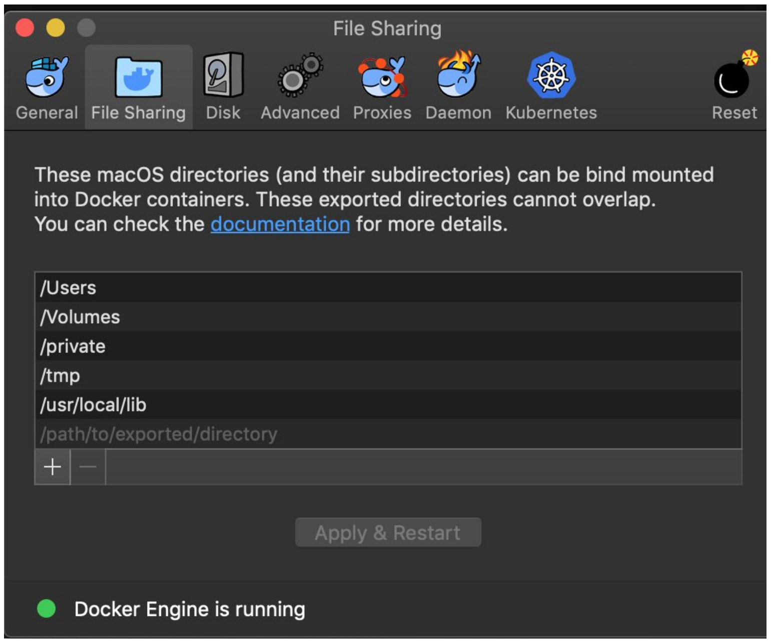

About Docker Desktop

Docker Desktop
Community
Version 2.0.0.3 (31259)
Channel: stable
8858db33c8

Engine: 18.09.2
Notary: 0.6.1
Compose: 1.23.2
Credential Helper: 0.6.0
Machine: 0.16.1
Kubernetes: v1.10.11

Copyright © 2016-2019 Docker Inc. All Rights Reserved.
Docker and the Docker logo are trademarks of Docker Inc. registered in the U.S. and other countries.

Once installed, go to Docker Notification icon, option *Preferences*, tab *File Sharing* and add the directory */usr/local/lib:*

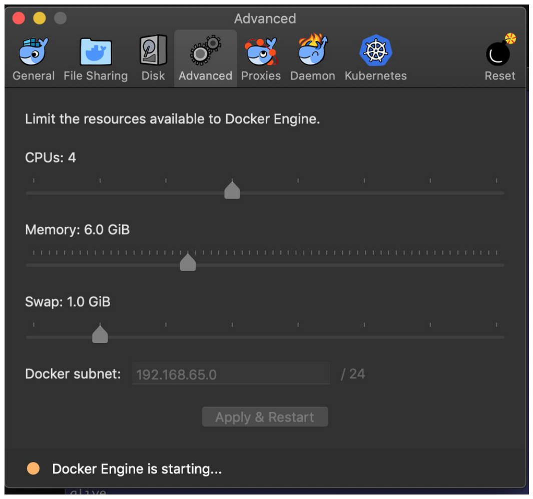


These macOS directories (and their subdirectories) can be bind mounted into Docker containers. These exported directories cannot overlap.
You can check the documentation for more details.

/Users
/Volumes
/private
/tmp
/usr/local/lib
/path/to/exported/directory

Docker Engine is running

Within tab *Advanced,* increase the resources to fit your Dhis2 instance requirements:



Limit the resources available to Docker Engine.

CPUs: 4

Memory: 6.0 GiB

Swap: 1.0 GiB

Docker subnet: 192.168.65.0 / 24

Docker Engine is starting...

Now, open a shell terminal. You should be an admin user to proceed. First, make sure your user can write in */usr/local*, brew needs write-access to the directory:

```
$ sudo chown -R $(whoami):admin /usr/local
```

And finally, install *python3* using *brew*:

```
$ brew install python3
$ brew postinstall python3
```

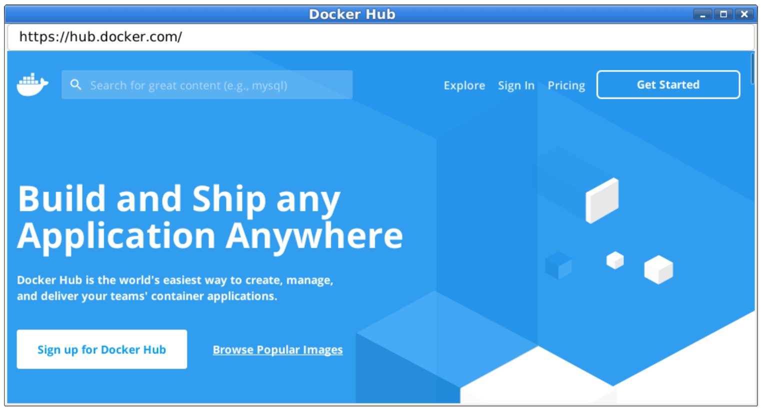
Important: Once installed, you should run all *d2-docker* commands as root, so either start a root session (i.e. *sudo su* -), or prefix all *d2-docker* commands with **sudo**.

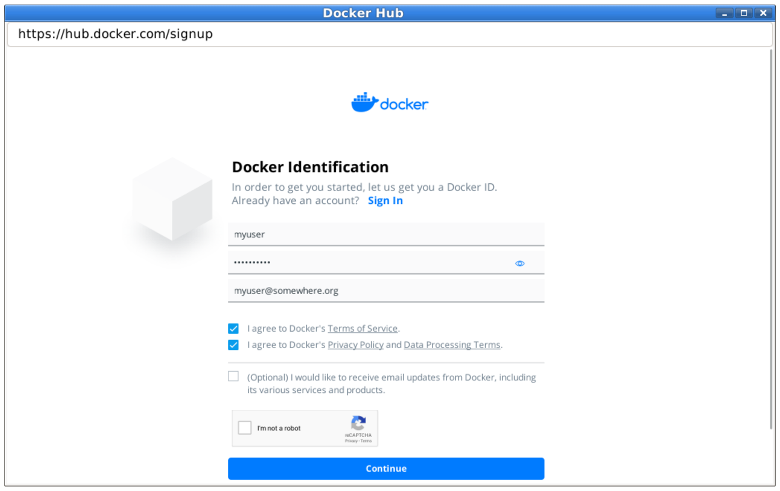
PREPARING A DOCKER HUB ORGANIZATION

D2-docker uses two custom images to run DHIS2 instances:

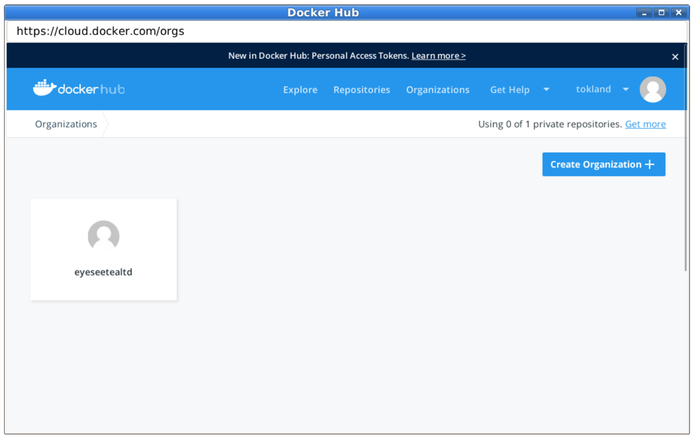
- **dhis2-core**: Contains *Tomcat* that deploys the *dhis.war* bundle.
- **dhis2-data**: Contains a database dump and the directory with the DHIS2 web apps.

We use Docker Hub to store those images. If you don't have yet an account on the Docker Hub, go to its main page and click on **Sign up**:

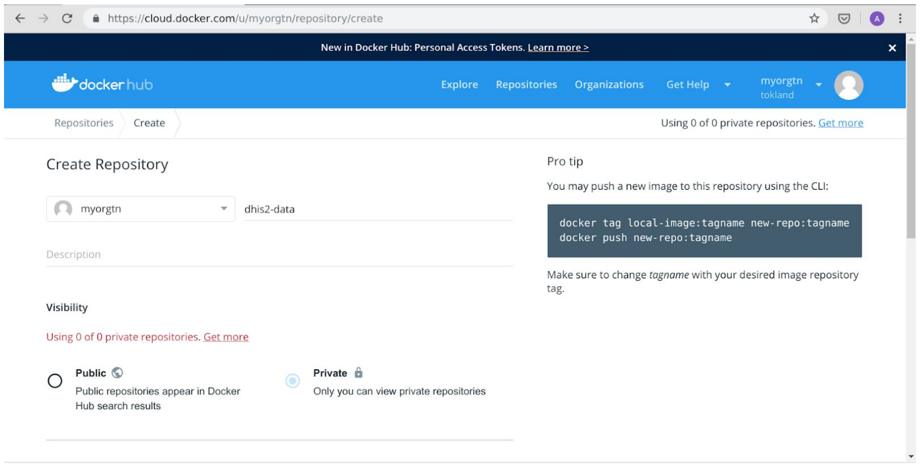

Docker Hub
https://hub.docker.com/
Search for great content (e.g., mysql)
Explore
Sign In
Pricing
Get Started
Build and Ship any Application Anywhere
Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.
Sign up for Docker Hub
Browse Popular Images


Docker Hub
https://hub.docker.com/signup
docker
Docker Identification
In order to get you started, let us get you a Docker ID.
Already have an account? Sign In
myuser
••••••••••
myuser@somewhere.org
[✓] I agree to Docker's Terms of Service.
[✓] I agree to Docker's Privacy Policy and Data Processing Terms.
[ ] (Optional) I would like to receive email updates from Docker, including its various services and products.
I'm not a robot
Continue

If you don't have yet an organizations created for your group team, create one in **Organizations**:


Docker Hub
https://cloud.docker.com/orgs
New in Docker Hub: Personal Access Tokens. Learn more >
docker hub
Explore
Repositories
Organizations
Get Help
tokland
Organizations
Using 0 of 1 private repositories. Get more
Create Organization +
eyeseetealtd

Now we can create the two repositories we need (**dhis2-core** and **dhis2-data**) in section **Repositories** with public or private access**:**

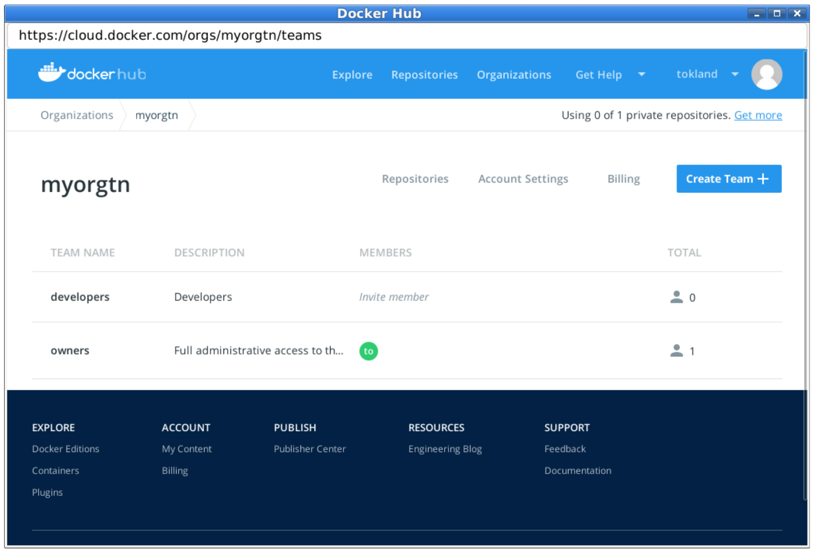

https://cloud.docker.com/u/myorgtn/repository/create
New in Docker Hub: Personal Access Tokens. Learn more >
docker hub
Explore
Repositories
Organizations
Get Help
myorgtn
tokland
Repositories
Create
Using 0 of 0 private repositories. Get more
Create Repository
myorgtn
dhis2-data
Description
Visibility
Using 0 of 0 private repositories. Get more
Public
Public repositories appear in Docker Hub search results
Private
Only you can view private repositories
Pro tip

You may push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

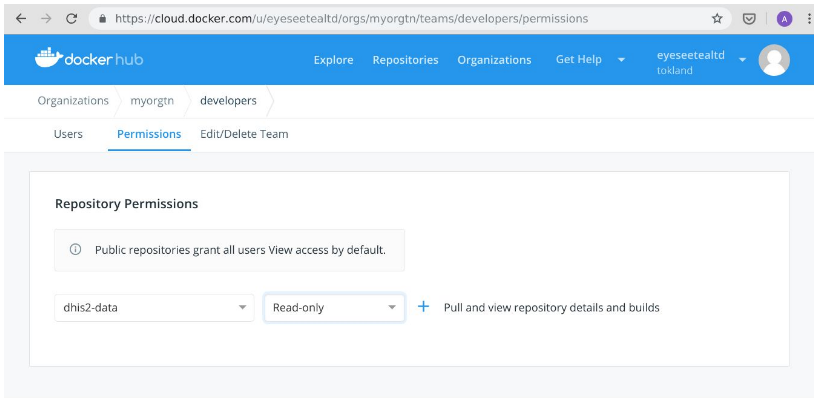
Make sure to change *tagname* with your desired image repository tag.

And finally create/edit the teams and its permissions over the repositories:



myorgtn

TEAM NAME	DESCRIPTION	MEMBERS	TOTAL
developers	Developers	Invite member	0
owners	Full administrative access to th...	to	1



Repository Permissions

Public repositories grant all users View access by default.

dhis2-data | Read-only | + Pull and view repository details and builds

INSTALLING d2-docker

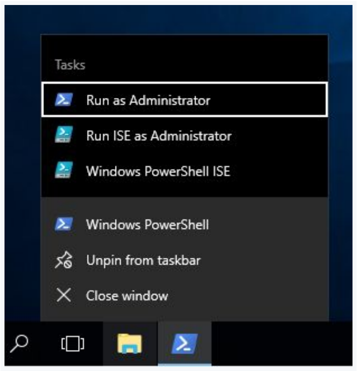
You can either download d2-docker.zip or clone the source repository using *git*:

```
$ git clone https://github.com/EyeSeeTea/d2-docker
```

For GNU/Linux, install the contents from non-admin user with sudo permissions (entering the d2-docker folder first):

```
$ cd d2-docker
$ sudo python3 setup.py install
```

For Windows, run a terminal as an Administrator and run:



```
$ cd d2-docker
$ python setup.py install
```

NOTE: In case you changed the location where you downloaded d2-docker, please change the “cd” command accordingly to access the d2-docker folder

COMMAND SYNTAX

D2-docker uses a single executable with subcommands (start, stop, ...) with this structure:

```
$ d2-docker [GLOBAL_OPTIONS] COMMAND [COMMAND_OPTIONS]
```

The complete interface:

```
$ d2-docker --help
usage: d2-docker [-h] [--dhis2-docker-images-directory DIRECTORY]
                 [--log-level NOTSET | DEBUG | INFO | WARNING | ERROR | CRITICAL]
                 {start,logs,stop,commit,push,copy,export,import,list,run-sql,create}
                 ...

positional arguments:
  {start,logs,stop,commit,push,copy,export,import,list,run-sql,create}
                        Subcommands
    start               Start a container from an existing dhis2-data Docker
                        image or from an exported file
    logs                Show docker logs
    stop                Stop docker containers
    commit              Commit docker images
    push                Push dhis2-data docker image
    copy                Copy databases from/to docker containers
    export              Export d2-docker images to a single file
    import              Import d2-docker images from file
    list                List d2-docker data images
    run-sql             Run SQL or open interactive session in a d2-docker
                        container
    create              Create d2-docker images

optional arguments:
  -h, --help            show this help message and exit
  --dhis2-docker-images-directory DIRECTORY
                        Directory containing dhis2-data docker source code
  --log-level NOTSET | DEBUG | INFO | WARNING | ERROR | CRITICAL
                        Run command with the given log level
```

MAINTAINING d2-docker

For end user commands (like starting or stopping instances, please refer to the end-user manual

Setup: create the base images

Example with version 2.30 and an *ento* database:

```
$ d2-docker create core eyeseetea/dhis2-core:2.30 --version=2.30
```

And now create your data images:

```
$ d2-docker create data eyeseetea/dhis2-data:2.30-ento --sql=db-ento.sql.gz --apps=path/to/apps/
$ d2-docker create data eyeseetea/dhis2-data:2.30-ento-gh --sql=db-ento-gh.sql.gz # No apps
```

Make an instance backup

Extract data image `eyeseetea/dhis2-data:2.30` (DB + apps) to the local directory `eyeseetea-dhis2-data-2.30`:

```
$ d2-docker copy eyeseetea/dhis2-data:2.30 eyeseetea-dhis2-data-2.30
$ ls eyeseetea-dhis2-data-2.30/
apps  db.sql.gz
```

You can restore it with the same command, just swapping the arguments:

```
$ d2-docker copy eyeseetea-dhis2-data-2.30 eyeseetea/dhis2-data:2.30
```

Backup: export and import to/from a single file

```
$ d2-docker export -i eyeseetea/dhis2-data:2.30 eyeseetea-dhis2-data-2.30.tgz
```

You can now copy *eyeseetea-dhis2-data-2.30.tgz* to another machine and run an *import* command:

```
$ d2-docker import eyeseetea-dhis2-data-2.30.tgz
```

Clone one instance

The *copy* command also allows creating new images from existing ones:

```
$ d2-docker copy eyeseetea/dhis2-data:2.30 anotherorg/dhis2-data:2.30-cloned
```

Modify and commit changes

First, start the image you want to work on:

```
$ d2-docker start eyeseetea/dhis2-data:2.30-ento --detach
```

Make all the changes you need and when done, run this command to update the local image:

```
$ d2-docker commit
```

Now you might publish the Docker image with a **push** command:

```
$ d2-docker push
```

Alter DHIS2 databases

Execute a sql file in an instance using command **run-sql**:

```
$ d2-docker run-sql -i eyeseetea/dhis2-data:2.30-ento somefile.sql
```

If no file is specified, an interactive **psql** terminal will open:

```
$ d2-docker run-sql -i eyeseetea/dhis2-data:2.30-ento
psql (9.6.14)
Type "help" for help.

dhis2=#
```

It may be also useful to run some initializing SQL everytime an image is started. An example:

```
$ d2-docker start eyeseetea/dhis2-data:2.30-ento --run-sql=directory-with-sql-files
```

`directory-with-sql-files` should contain *.sql*, *.sql.gz* or *.dump* files. Those files will be imported after the DB initialization process is finished but before Tomcat is started.

Upgrade a DHIS2 instance in the docker

Update to the latest 2.30 for an existing core image:

```
$ d2-docker create core eyeseetea/dhis2-core:2.30 --version=2.30
```

Or use a local *WAR* file:

```
$ d2-docker create core eyeseetea/dhis2-core:2.30 --war=dhis.war
```

Cleaning-up

Docker infrastructure (images, networks, containers, volumes) takes up a lot of hard-disk space.

Remove all local volumes not used by at least one container:

```
$ docker volume prune
```

Remove all stopped containers:

```
$ docker container prune
```

Remove all dangling images (the temporal images that have <none> on its name/tag):

```
$ docker image prune
```

[WARNING: Dangerous operation] This command deletes all stopped containers, networks, volumes, images and cache. Note, that any **dhis2-data** image will be also deleted whether the instance is running or nor (as the data container is not running), so make sure it's pushed to the hub if you want to keep the data.

```
$ docker system prune -a --volumes
```