Using Docker in BioHPC Cloud

Jaroslaw Pillardy
Bioinformatics Facility

Part 1 – presentation

May 19 2020

What is Docker?

A Linux subsystem to run isolated Linux “machines” called “containers”

Isolated means that programs, users and storage in Docker “machine” are separated from the host system.

It can run any flavor of Linux on any Linux machine – Docker container looks like a complete standalone Linux server with all files, programs and settings.

What is Docker?

Programs installed inside Docker container don’t need to be compatible with the host, and in fact they need to be installed from scratch, regardless of what is installed on the host.

Regular user can become “root” (Linux administrator) inside Docker container, or any user as needed.

Docker components

image - a template than can be loaded into Docker and executed. Image can be stored on a disk as a file or in a specialized Docker repository of images

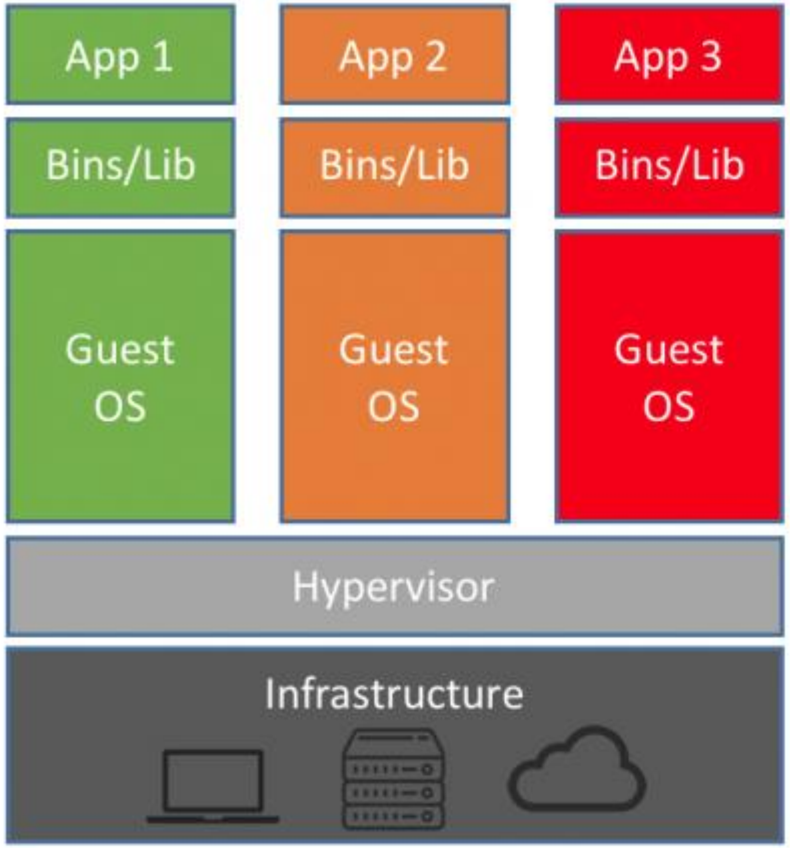
container - a running instance of Docker image – actual Docker “machine”. Users can execute programs, install software and work in container as in a regular Linux system.

Docker components

image - like a program on the disk – it contains all information needed, but it needs to be loaded into memory to run.

container – like a program running on a computer – in the memory, can load data, change its own stored memory data etc.

Virtual Machine vs Docker Containers



Machine Virtualization

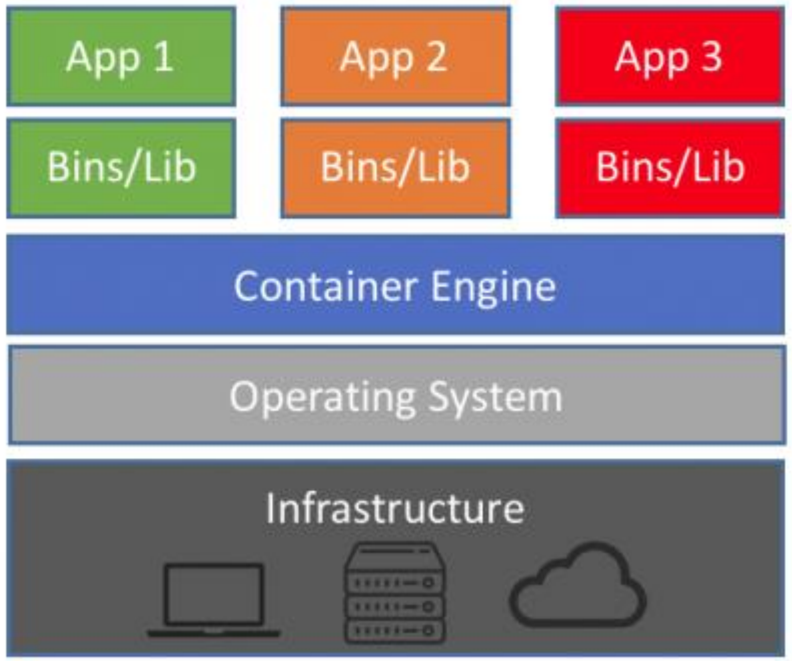
Virtual Machine (VM) is a separated "server" emulating its own hardware and running its own operating system. The only interaction with host OS is via hardware or emulated hardware.

VM can run any OS but is slower and requires extra resources to run. Up to 20-30% overhead.

Virtual Machine vs Docker Containers

Docker shares host OS kernel services and some libraries (read-only). It runs as a process in host operating system. It can access host files directly (optional).

No execution overhead – same speed. Cannot run non-Linux OS.



Containers

Docker security

In order to use original Docker user must have “root” access to certain parts of Linux OS.

It is safe for admins to deploy software, but NOT safe for users in multi-user environment like BioHPC

We have developed our own version of Docker on the top of original Docker that addresses security problems at the same time preserving most of Docker features.

BioHPC Docker

Original Docker command is "docker". This command has been replaced by "docker1" command in BioHPC Lab.

Whenever reading a Docker book or website please replace "docker" with "docker1" when you want to run the command on BioHPC Lab machines.

BioHPC Docker

If you run "docker" instead of "docker1" you will get an error. You have to use “docker1”

```
[jarekp@cbsum1c2b014 ~]$ docker ps -a
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
[jarekp@cbsum1c2b014 ~]$
```

```
[jarekp@cbsum1c1b009 ~]$ docker ps -a
Got permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Get
http://%2Fvar%2Frun%2Fdocker.sock/v1.26/containers/json?all=1: dial unix
/var/run/docker.sock: connect: permission denied
```

BioHPC Docker

You can check docker1 options with “docker1 --help” or “docker1 commandname --help"

```
[jarekp@cbsum1c1b009 ~]$ docker1 --help
This is BioHPC Cloud docker1 implementing secure Docker environment.
Some Docker commands have been modified or disabled, but most are
unchanged. There are three additional commands in docker1 listed below

 clean    Deletes sets of containers
 claim    Changes ownership of all files and dirs in /workdir/labid and /SSD/labid to labid
 white    Displays set of options that are whitelisted to use with docker1
Special option --noworkdir for run command disables automatic mapping of /workdir and /SSD
Please note you need to put docker command options BEFORE container name where applicable
Docker help page follows.

Usage:  docker COMMAND

A self-sufficient runtime for containers
```

Docker images

Before running any Dockerized application you need to know how to access its Docker image.

- Images are stored in Docker registries (or hubs) and their names and addresses are described in software documentation.

```
docker1 pull docker.io/biohpc/imagename
docker1 pull docker.io/imagename
```

BioHPC image, public

General public image

`docker.io/` part may be omitted, repositories known to this Docker installation will be queried in order

- Image can be imported from a file

```
docker1 load –i filename
docker1 import filename
```

Docker images: `load` vs `import`

- **`docker1 load -i filename`**
  Docker load command creates a container from saved image, it imports all the image layers, tags and settings. File for load command must be created with **save** command.

- **`docker1 import filename`**
  Docker import command creates a container from saved container, it creates a simplified image based on the saved container with a single layer and no settings (like entry point). File for import command must be created with **export** command

BioHPC Docker

To run a container:

- Pull or import/load image
- Start container from image

You can also start container without pulling first (only from registry, not a file), Docker will pull from hub it if not found locally. All images used on a server are cached in a local registry.

BioHPC Docker

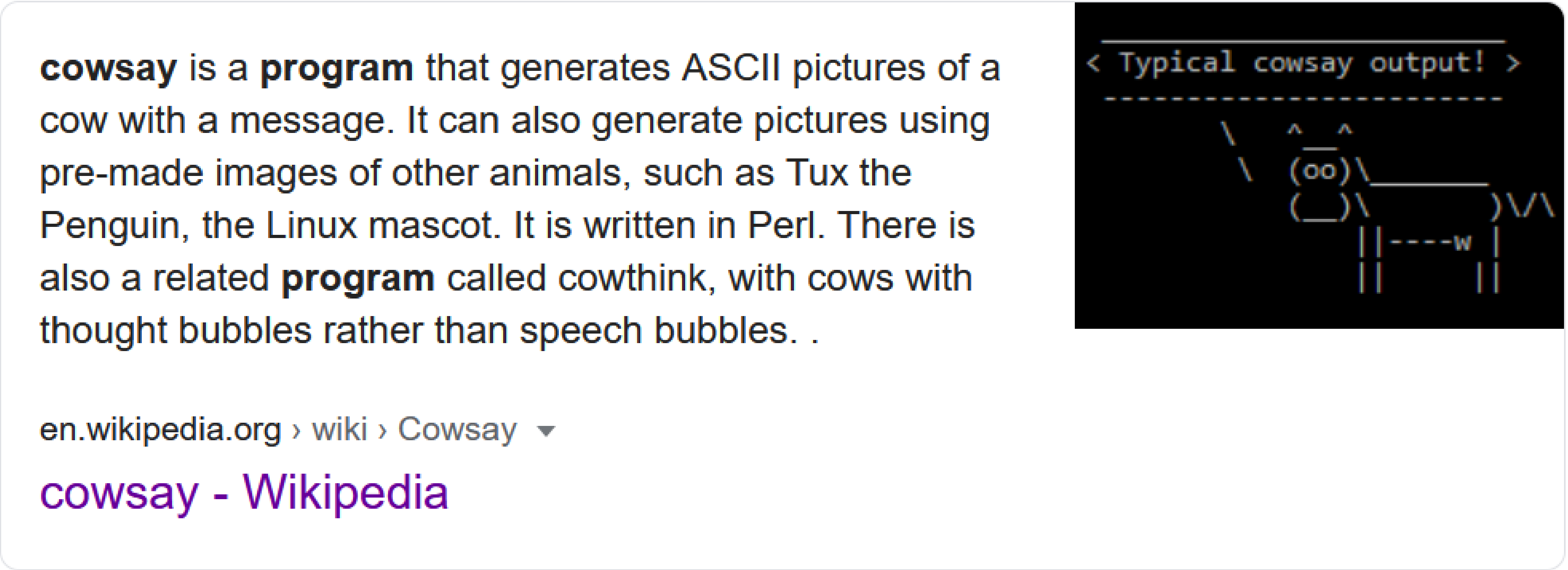
Pull image

docker.io/ part may be omitted, repositories known to this Docker installation will be queried in order

```
[jarekp@cbsum1c1b009 ~]$ docker1 pull biohpc/cowsay
Using default tag: latest
Trying to pull repository dtr.cucloud.net/biohpc/cowsay ...
Trying to pull repository docker.io/biohpc/cowsay ...
sha256:c4c5c72fd2e09e35a02199aef774f5338ab6a17f204ed7317f11a58f8d1ad284: Pulling from
docker.io/biohpc/cowsay
9e4b184d5239: Pull complete
4a28c3c49d95: Pull complete
Digest: sha256:c4c5c72fd2e09e35a02199aef774f5338ab6a17f204ed7317f11a58f8d1ad284
Status: Downloaded newer image for docker.io/biohpc/cowsay:latest

[jarekp@cbsum1c1b009 ~]$ docker1 images
REPOSITORY                TAG                 IMAGE ID            CREATED             SIZE
docker.io/biohpc/cowsay   latest              195f168235c9        3 years ago         337 MB
[jarekp@cbsum1c1b009 ~]$
```

List images



cowsay is a **program** that generates ASCII pictures of a cow with a message. It can also generate pictures using pre-made images of other animals, such as Tux the Penguin, the Linux mascot. It is written in Perl. There is also a related **program** called cowthink, with cows with thought bubbles rather than speech bubbles. .

```
 ________________________
< Typical cowsay output! >
 ------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

en.wikipedia.org › wiki › Cowsay ▾

cowsay - Wikipedia

BioHPC Docker

Import image

```
[jarekp@cbsum1c2b014 ~]$ docker1 import /programs/docker/images/cowsay.tar
sha256:da8e563445a8792ae5b161b446e8ef9ca2c76f2bafab58ad88bf0adcbfb5d0b0

[jarekp@cbsum1c2b014 ~]$ docker1 images
REPOSITORY                      TAG                 IMAGE ID            CREATED              SIZE
biohpc_jarekp/cowsay            latest              da8e563445a8        About a minute ago   319.7 MB
docker.io/biohpc/cowsay         latest              195f168235c9        16 months ago        337.1 MB
[jarekp@cbsum1c2b014 ~]$
```

What if we try to load a file created with export?

```
[jarekp@cbsumm15 docker]$ docker1 load -i /programs/docker/images/cowsay.tar
open /local/docker/tmp/docker-import-755479071/dev/json: no such file or directory
[jarekp@cbsumm15 docker]$
```

BioHPC Docker

Run Docker container

- **Single command**

  Run a command and then container stops.

- **Interactive mode**

  Open shell inside container for interactive work. Once you are finished, exit shell and container stops.

- **Background mode**

  Start container in the background and connect to it when needed. Container will keep running.

BioHPC Docker

Single command run

command image_name program program_arguments

```
[jarekp@cbsum1c2b014 ~]$ docker1 run biohpc/cowsay cowsay "This is Docker"
 ________________
< This is Docker >
 ----------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
CONTAINER ID    IMAGE           COMMAND                   CREATED           STATUS                          PORTS NAMES
1035e0f20e5f    biohpc/cowsay   "cowsay 'This is Dock"    16 seconds ago    Exited (0) 13 seconds ago           jarekp__biohpc_1
[jarekp@cbsum1c2b014 ~]$
```

List containers on this host, -a lists both active and inactive

Image exited – cannot connect to it anymore

BioHPC Docker

Interactive run

run interactively   image name   command to run

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -it biohpc/cowsay /bin/bash
[root@a0017f5faf51 ~]# pwd
/workdir
[root@a0017f5faf51 ~]# cowsay "hi"
 ____
< hi >
 ----
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
[root@a0017f5faf51 ~]# exit
exit
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
CONTAINER ID        IMAGE                   COMMAND                    CREATED            STATUS                      PORTS   NAMES
a0017f5faf51        biohpc_jarekp/cowsay    "/bin/bash"                32 seconds ago     Exited (0) 6 seconds ago            jarekp__biohpc_2
1035e0f20e5f        biohpc/cowsay           "cowsay 'This is Dock"     6 minutes ago      Exited (0) 6 minutes ago            jarekp__biohpc_1
 [jarekp@cbsum1c2b014 ~]$
```

we are inside container now – as root!

Image exited – cannot connect to it anymore

BioHPC Docker

Run in the background

run in background   image name   command to run

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -d -t biohpc/cowsay /bin/bash
10af80003f76940b154a176af4a3b3747647763c2a3eb62b27f9e442cad7060f
```

Image running

```
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS                       PORTS               NAMES
10af80003f76        biohpc/cowsay            "/bin/bash"              7 seconds ago       Up 4 seconds                                     jarekp__biohpc_3
a0017f5faf51        biohpc_jarekp/cowsay     "/bin/bash"              7 minutes ago       Exited (0) 7 minutes ago                         jarekp__biohpc_2
1035e0f20e5f        biohpc/cowsay            "cowsay 'This is Dock"   13 minutes ago      Exited (0) 13 minutes ago                        jarekp__biohpc_1

[jarekp@cbsum1c2b014 ~]$ docker1 exec 10af80003f76 cowsay "hi"
 ____
< hi >
 ----
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

BioHPC Docker

Run in the background

```
[jarekp@cbsum1c2b014 ~]$ docker1 exec -it 10af80003f76 /bin/bash
[root@10af80003f76 ~]# ls -al
total 0
drwxr-xr-x  2 4965 root   6 May 22 17:26 .
drwxr-xr-x 18 root root 288 May 22 18:48 ..
[root@10af80003f76 ~]# exit
exit

[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
CONTAINER ID        IMAGE                  COMMAND                  CREATED              STATUS                      PORTS               NAMES
10af80003f76        biohpc/cowsay          "/bin/bash"              About a minute ago   Up About a minute                               jarekp__biohpc_3
a0017f5faf51        biohpc_jarekp/cowsay   "/bin/bash"              9 minutes ago        Exited (0) 8 minutes ago                        jarekp__biohpc_2
1035e0f20e5f        biohpc/cowsay          "cowsay 'This is Dock"   14 minutes ago       Exited (0) 14 minutes ago                       jarekp__biohpc_1
[jarekp@cbsum1c2b014 ~]$
```

The container keeps running in the background! You can execute a command in it, get inside for interactive work, leave it, and it is still there, running.

Run Docker container

- **Single command**
  Run a command and then container stops.
  ```
  docker1 run image command arguments
  ```

- **Interactive mode**
  Open shell inside container for interactive work. Once you are finished, exit shell and container stops.
  ```
  docker1 run –it image /bin/bash
  ```

- **Background mode**
  Start container in the background and connect to it when needed. Container will keep running.
  ```
  docker1 run -d -t image /bin/bash
  docker1 exec container_id command arguments
  docker1 exec –it container_id /bin/bash
  ```

BioHPC Docker

Restart container that stopped running (exited)

```
docker1 start container_id_or_name
```

```
docker1 –a –i start container_id_or_name
```

(if restarting in interactive mode)

Docker useful commands summary

- List all conatiners

  ```
  docker1 ps -a
  ```

- List all images

  ```
  docker1 images
  ```

- Pull image from a hub

  ```
  docker1 pull image_name
  ```

- Load image from <u>save</u> file

  ```
  docker1 load –i filename
  ```

- Load image from <u>export</u> file

  ```
  docker1 import filename
  ```

BioHPC Docker

stop running container

```
docker1 stop container_id_or_name
```

remove (erase) container

```
docker1 rm container_id_or_name
```

Typically id (e.g. 10af80003f76) or name (e.g. jarekp__biohpc_1) may be used as parameter for stop, rm, exec

BioHPC Docker – cleaning leftovers

Remove all current user non-running containers

```
docker1 clean
```

Remove all current user containers – running or not

```
docker1 clean all
```

Automatically remove my container after exit

```
docker1 run --rm image program arguments
```

BioHPC Docker – volumes and directories

By default /workdir/labid is mapped to /workdir inside the container.

You can skip that by using --noworkdir option (in docker1 run command)

BioHPC Docker – volumes and directories

Version tag

```
[jarekp@cbsum1c1b009 ~]$ docker1 run -it centos:7 /bin/bash
Unable to find image 'centos:7' locally
Trying to pull repository dtr.cucloud.net/centos ...
Trying to pull repository docker.io/library/centos ...
sha256:e9ce0b76f29f942502facd92293f1582: Pulling from docker.io/library/centos
524b0c1e57f8: Pull complete
Digest: sha256:e9ce0b76f29f942502facd849f3e468232492b259b9d9f076f71b392293f1582
Status: Downloaded newer image for docker.io/centos:7
[root@c0034f8e2cd8 /]# cd /workdir
[root@c0034f8e2cd8 workdir]# echo test > myfile
[root@c0034f8e2cd8 workdir]# ls -al
total 4
drwxr-xr-x 2 4965 root 20 May 18 19:05 .
drwxr-xr-x 1 root root 32 May 18 19:05 ..
-rw-r--r-- 1 root root  5 May 18 19:05 myfile
[root@c0034f8e2cd8 workdir]# exit
```

File created inside the container is owned by “root”

BioHPC Docker – volumes and directories

```
[jarekp@cbsum1c1b009 ~]$ cd /workdir/jarekp/
[jarekp@cbsum1c1b009 jarekp]$ ls -al
total 4
drwxr-xr-x 2 jarekp root 20 May 18 15:05 .
drwxrwxrwx 5 root   root 71 May 18 15:04 ..
-rw-r--r-- 1 root   root  5 May 18 15:05 myfile

[jarekp@cbsum1c1b009 jarekp]$ docker1 claim
[jarekp@cbsum1c1b009 jarekp]$ ls -al
total 4
drwxr-xr-x 2 jarekp root 20 May 18 15:05 .
drwxrwxrwx 5 root   root 71 May 18 15:04 ..
-rw-r--r-- 1 jarekp root  5 May 18 15:05 myfile


[jarekp@cbsum1c1b009 jarekp]$ docker1 run --noworkdir -it centos:7 /bin/bash
[root@8f70402109a9 /]# cd /workdir
bash: cd: /workdir: No such file or directory
```

Still is owned by “root” after exiting the container

docker1 claim restores ownership to user

No /workdir anymore

BioHPC Docker – volumes and directories

It is possible to map other directories from host machine to the container inner file system.

The host directory to be mapped must be owned by you and it must be under /workdir/labid/, /local/storage/ or /fs/servername/storage/, /SSD/labid (replace labid with your BioHPC user id)

BioHPC Docker – volumes and directories

Mapping /workdir/jarekp/testdir/ outside container
to /data inside container
Make sure directory name ends with /

```
[jarekp@cbsum1c1b009 ~]$ cd /workdir/jarekp/
[jarekp@cbsum1c1b009 jarekp]$ mkdir testdir
[jarekp@cbsum1c1b009 jarekp]$ cd testdir/
[jarekp@cbsum1c1b009 testdir]$ echo test > testfile

[jarekp@cbsum1c1b009 testdir]$ docker1 run -v /workdir/jarekp/testdir/:/data -d -t centos:7 /bin/bash
058fb8c312bc415c723dcdaf6842f37c7cb6d3d4dbe092658ac574172da70e69
[jarekp@cbsum1c1b009 testdir]$ docker1 ps -a
CONTAINER ID        IMAGE            COMMAND          CREATED             STATUS            PORTS   NAMES
058fb8c312bc        centos:7         "/bin/bash"      15 seconds ago      Up 13 seconds
jarekp__biohpc_4

[jarekp@cbsum1c1b009 testdir]$ docker1 exec -it 058fb8c312bc /bin/bash
[root@058fb8c312bc /]# cd /data
[root@058fb8c312bc data]# ls -al
total 4
drwxr-xr-x 2 4965 4963 22 May 18 19:15 .
drwxr-xr-x 1 root root 44 May 18 19:15 ..
-rw-r--r-- 1 4965 4963  5 May 18 19:15 testfile
[root@058fb8c312bc data]# cat testfile
test
[root@058fb8c312bc data]#
```

Some Docker images with preinstalled software available from developers require data to be placed in certain directories, often /data. Mapping allows you to satisfy this condition.

BioHPC Docker – volumes and directories

```
[jarekp@cbsum1c1b009 testdir]$ docker1 run -v /home/jarekp/testdir/:/data -d -t centos:7 /bin/bash
ERROR 'run' option '-v' is only allowed with mapping of /workdir/jarekp/,
      /local/workdir/jarekp/, /SSD/jarekp (if /SSD is present)
      or /local/storage/dir (on hosted machines where jarekp is dir owner,
      /local/storage/dir can be cross-mounted on /fs/servername/storage/).
      You tried mapping /home/jarekp/testdir, which does not start with /workdir/jarekp/,
      /local/storage/ or /fs/servername/storage/
If 'run' options listed above do exist in Docker documentation
and you need them please contact us at cbsu@cornell.edu
Exiting ...
[jarekp@cbsum1c1b009 testdir]$
```

You cannot map any directory, only the ones listed above

Docker repositories

Pulled or created Docker images are stored locally on disk in a **local repository/registry**

BioHPC Docker keeps them in /local/docker directory

It is possible to copy local images back to network repositories, provided you have write access to them

Docker repositories

If you modify a container you may want to save it for the future use. Containers and images on rental machines will be deleted once the reservation ends.

You can save container to a file using **export** command,

or you can convert container to an image and save it to a file with **save** command, or **push** the image to a hub

BioHPC Docker – htop example

In the following example we will start a container using CentOS 7 image, install **htop** program in it and then save it in various ways.

htop is a very useful little program to display server CPU, RAM and I/O usage. It is not a part of default CentOS installation.

Sysadmin digression

Many programs and software packages can be installed on Linux with package managers that pull software from software repositories

They depend on flavor of Linux :

RedHat, CentOS	**yum**
Debian, Ubuntu	**apt-get**
All	**rpm**

BioHPC Docker – htop example

```
[jarekp@cbsum1c1b009 ~]$ docker1 run -d -t centos:7 /bin/bash
829b3f1d48e9e61a462f02714af49d01eac0be34c5f2e41a34ff71f9e68d6b7b
[jarekp@cbsum1c1b009 ~]$ docker1 ps -a
CONTAINER ID  IMAGE      COMMAND       CREATED       STATUS      PORTS  NAMES
829b3f1d48e9  centos:7   "/bin/bash"   6 seconds ago Up 5 seconds       jarekp__biohpc_1
[jarekp@cbsum1c1b009 ~]$ docker1 exec -it jarekp__biohpc_1 /bin/bash
[root@829b3f1d48e9 /]# htop
bash: htop: command not found
```

htop is not included in the CentOS 7 system

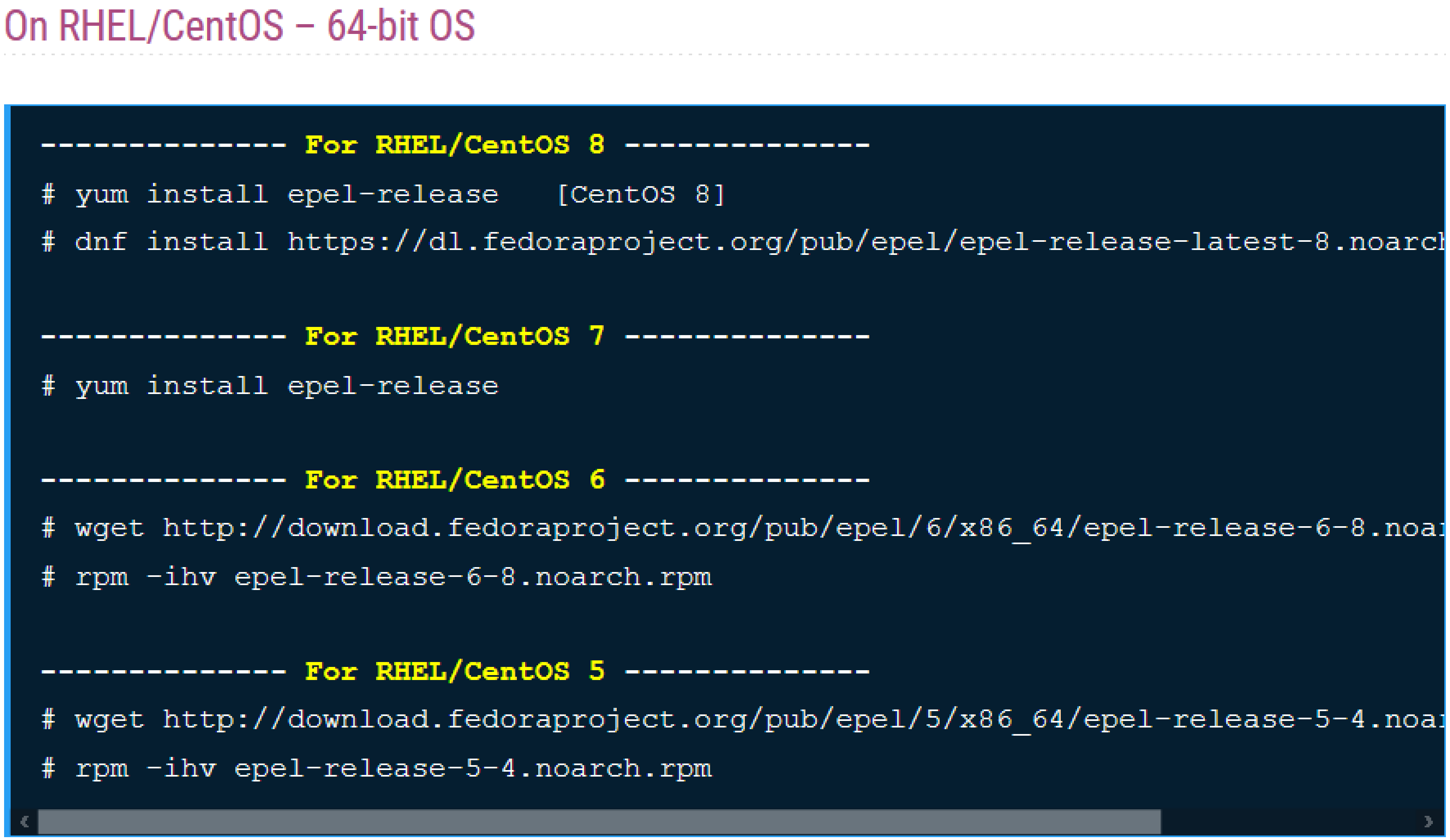
BioHPC Docker – htop example

```
[root@829b3f1d48e9 /]# yum install htop
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirror.clarkson.edu
 * extras: mirrors.tripadvisor.com
 * updates: mirror.siena.edu
base
extras
updates
(1/4): base/7/x86_64/group_gz
(2/4): updates/7/x86_64/primary_db
(3/4): extras/7/x86_64/primary_db
(4/4): base/7/x86_64/primary_db
No package htop available.
Error: Nothing to do
[root@829b3f1d48e9 /]#
```

No htop available in these default repositories

BioHPC Docker – htop example

Here is a part of the web page found via Google search “how to install htop on centos 7”



On RHEL/CentOS – 64-bit OS

```
-------------- For RHEL/CentOS 8 --------------
# yum install epel-release   [CentOS 8]
# dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch

-------------- For RHEL/CentOS 7 --------------
# yum install epel-release

-------------- For RHEL/CentOS 6 --------------
# wget http://download.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noa
# rpm -ihv epel-release-6-8.noarch.rpm

-------------- For RHEL/CentOS 5 --------------
# wget http://download.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noa
# rpm -ihv epel-release-5-4.noarch.rpm
```

BioHPC Docker – htop example

```
[root@829b3f1d48e9 /]# yum install epel-release
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
 * base: mirror.clarkson.edu
 * extras: mirrors.tripadvisor.com
 * updates: mirror.siena.edu
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

 […]

Installed:
  epel-release.noarch 0:7-11

Complete!
[root@829b3f1d48e9 /]#
```

BioHPC Docker – htop example

```
[root@829b3f1d48e9 /]# yum install htop
Loaded plugins: fastestmirror, ovl
Loading mirror speeds from cached hostfile
epel/x86_64/metalink
 * base: mirror.clarkson.edu
 * epel: ftp.cse.buffalo.edu
 * extras: mirrors.tripadvisor.com
 * updates: mirror.siena.edu

[…]

Running transaction
  Installing : htop-2.2.0-3.el7.x86_64
  Verifying  : htop-2.2.0-3.el7.x86_64

Installed:
  htop.x86_64 0:2.2.0-3.el7

Complete!
[root@829b3f1d48e9 /]#
```

BioHPC Docker – htop example

We have now a container with htop. How to save it for later?

BioHPC Docker – htop example

```
[jarekp@cbsum1c1b009 ~]$ docker1 ps -a
CONTAINER ID  IMAGE      COMMAND       CREATED       STATUS      PORTS  NAMES
829b3f1d48e9  centos:7   "/bin/bash"   6 seconds ago Up 5 seconds       jarekp__biohpc_1

[jarekp@cbsum1c1b009 ~]$ docker1 export -o htop_export.tar jarekp__biohpc_1
```

This command will export the container to a file. The resulting file can be imported to create an image, and then run as a container. What is inside?

```
[jarekp@cbsum1c1b009 ~]$ tar -tf htop_export.tar | more
.dockerenv
anaconda-post.log
bin
dev/
dev/console
dev/pts/
dev/shm/
etc/
etc/.pwd.lock
etc/BUILDTIME
...
```

Just a list of files – looks like a copy of CentOS system. No special Docker structures.

BioHPC Docker – htop example

In order to use save command or push the image to a hub we must create image from the container on hand

container_name image_name

```
[jarekp@cbsum1c1b009 ~] docker1 commit jarekp__biohpc_1 htop
sha256:515084f37c0ee7995c8b116aaa735ea0dd59b26f7f14daa7ae46b799a5a6ab01

[jarekp@cbsum1c1b009 ~]$ docker1 images
REPOSITORY          TAG       IMAGE ID       CREATED            SIZE
biohpc_jarekp/htop  latest    515084f37c0e   56 seconds ago     313 MB
docker.io/centos    7         b5b4d78bc90c   12 days ago        203 MB
```

htop image is in local private repository **biohpc_jarekp** on the local disk

BioHPC Docker – htop example

Now we can save this image to a file

```
[jarekp@cbsum1c1b009 ~]$ cd /workdir/jarekp/
```

file_name image_name

```
[jarekp@cbsum1c1b009 jarekp]$ docker1 save -o htop_save.tar biohpc_jarekp/htop
[jarekp@cbsum1c1b009 jarekp]$ ls -al htop_save.tar
-rw------- 1 root root 321216512 May 18 16:26 htop_save.tar
[jarekp@cbsum1c1b009 jarekp]$ tar -tf htop_save.tar | more
tar: htop_save.tar: Cannot open: Permission denied
tar: Error is not recoverable: exiting now

[jarekp@cbsum1c1b009 jarekp]$ docker1 claim
[jarekp@cbsum1c1b009 jarekp]$ ls -al htop_save.tar
-rw------- 1 jarekp root 321216512 May 18 16:26 htop_save.tar
[jarekp@cbsum1c1b009 jarekp]$
```

BioHPC Docker – htop example

This file is an exact replica of the image, NOT a simple list of files

```
[jarekp@cbsum1c1b009 jarekp]$ tar -tf htop_save.tar
0911479852295ffca6aec6236080916bbe7b3e5f1dfda2073d4c9e649346b806/
0911479852295ffca6aec6236080916bbe7b3e5f1dfda2073d4c9e649346b806/VERSION
0911479852295ffca6aec6236080916bbe7b3e5f1dfda2073d4c9e649346b806/json
0911479852295ffca6aec6236080916bbe7b3e5f1dfda2073d4c9e649346b806/layer.tar
515084f37c0ee7995c8b116aaa735ea0dd59b26f7f14daa7ae46b799a5a6ab01.json
f86f06285ab7b035414c42962c83782708058f446cb17b8008791adc4e7832ec/
f86f06285ab7b035414c42962c83782708058f446cb17b8008791adc4e7832ec/VERSION
f86f06285ab7b035414c42962c83782708058f446cb17b8008791adc4e7832ec/json
f86f06285ab7b035414c42962c83782708058f446cb17b8008791adc4e7832ec/layer.tar
manifest.json
Repositories
[jarekp@cbsum1c1b009 jarekp]$
```

BioHPC Docker – htop example

In order to save the image to a hub (registry) we need to

- tag the image with proper name
- login to the hub so it allows us to upload the image
- push the image to the hub

Example below uses our public biohpc repository in docker.io

BioHPC Docker – htop example

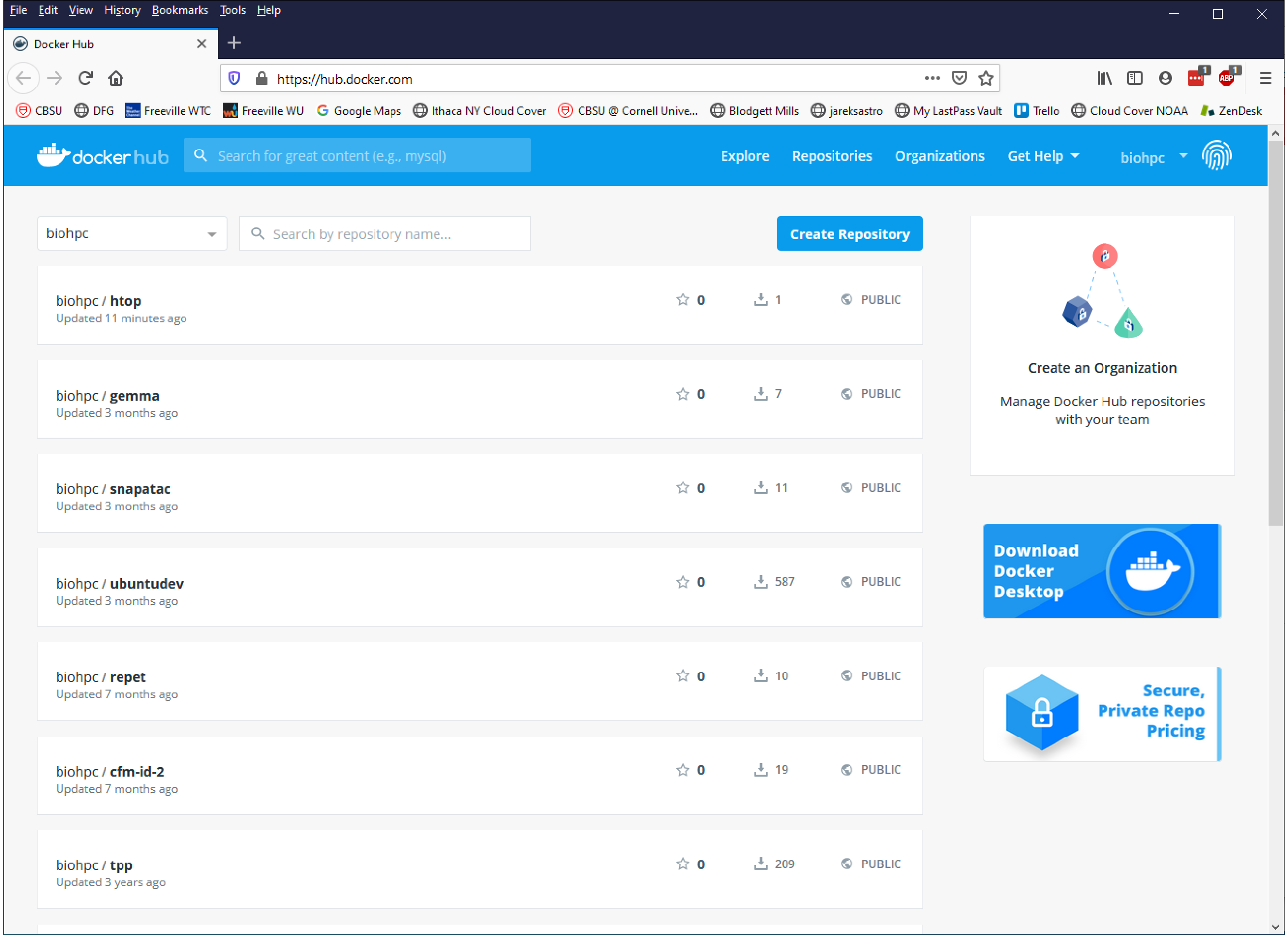
```
[jarekp@cbsum1c1b009 jarekp]$ docker1 tag biohpc_jarekp/htop docker.io/biohpc/htop
[jarekp@cbsum1c1b009 jarekp]$ docker1 images
REPOSITORY             TAG      IMAGE ID            CREATED             SIZE
biohpc_jarekp/htop     latest   515084f37c0e        21 minutes ago      313 MB
docker.io/biohpc/htop  latest   515084f37c0e        21 minutes ago      313 MB
```

We now have one image (same image ID), but with two names

BioHPC Docker – htop example

user   registry/hub

```
[jarekp@cbsum1c1b009 jarekp]$ docker1 login -u biohpc docker.io
Password:
Login Succeeded
[jarekp@cbsum1c1b009 jarekp]$ docker1 push docker.io/biohpc/htop
The push refers to a repository [docker.io/biohpc/htop]
dc3d542d36bd: Pushed
edf3aa290fb3: Mounted from library/centos
latest: digest:
sha256:bb61a8e6e069e74ed26acc9f82eb71663a619a8ec53ad871840c4baa91e6ad4a size:
741
[jarekp@cbsum1c1b009 jarekp]$ docker1 logout https://index.docker.io/v1/
Removing login credentials for index.docker.io
[jarekp@cbsum1c1b009 ~]$
```



Repository	Stars	Downloads	Visibility
biohpc / **htop** Updated 11 minutes ago	☆ 0	1	PUBLIC
biohpc / **gemma** Updated 3 months ago	☆ 0	7	PUBLIC
biohpc / **snapatac** Updated 3 months ago	☆ 0	11	PUBLIC
biohpc / **ubuntudev** Updated 3 months ago	☆ 0	587	PUBLIC
biohpc / **repet** Updated 7 months ago	☆ 0	10	PUBLIC
biohpc / **cfm-id-2** Updated 7 months ago	☆ 0	19	PUBLIC
biohpc / **tpp** Updated 3 years ago	☆ 0	209	PUBLIC