

Using Docker in BioHPC Cloud

Jaroslav Pillardy
Bioinformatics Facility

5/23/2018

What is Docker?

A Linux subsystem allowing 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

What is Docker?

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

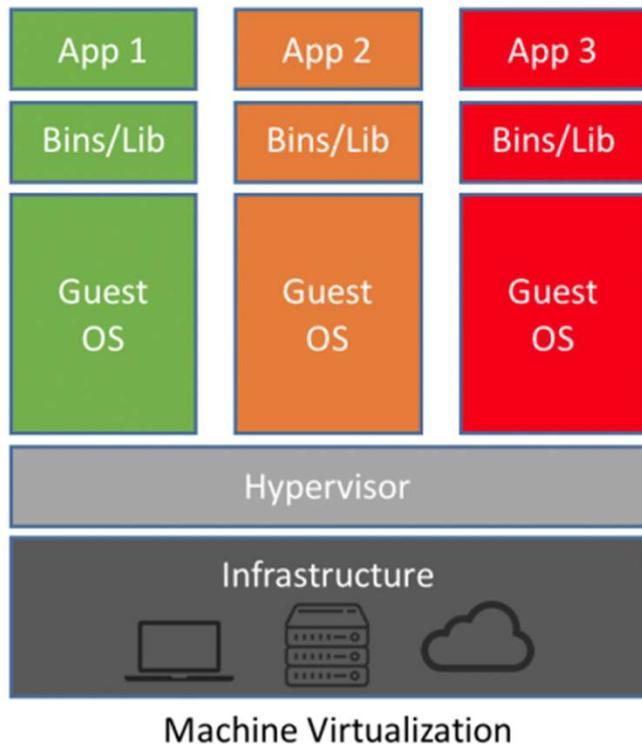
Regular user can become "root" (Linux administrator) inside Docker container, or any user as needed.

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.

Virtual Machine vs Docker Containers



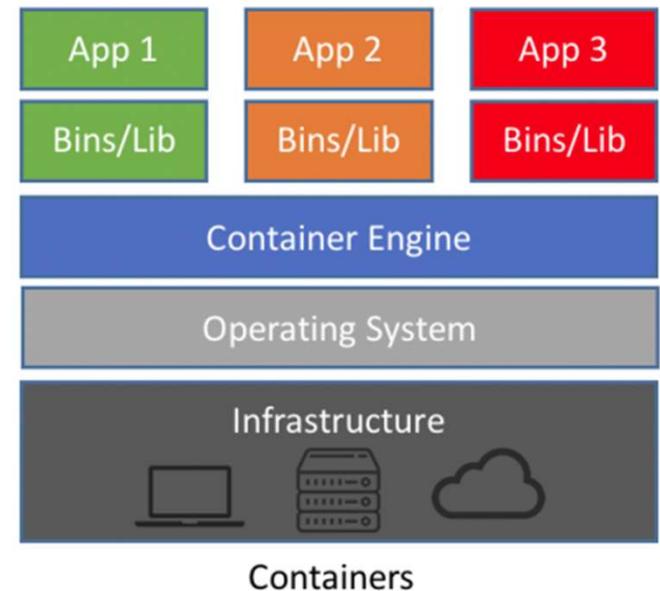
Virtual Machine (VM) is a separated “server” emulating its own hardware and running its own operating system. The only interaction with host OS is via hardware or emulated hardware.

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.



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 ~]$
```

BioHPC Docker

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

```
[jarekp@cbsum1c2b014 ~]$ docker1 --help
```

This is BioHPC Lab 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 to labid

white Displays set of options that are whitelisted to use with docker1

Special option --noworkdir for run command disables mapping of /workdir directory

Docker help page follows.

```
Usage: docker1 [OPTIONS] COMMAND [arg...]  
       docker1 [ --help | -v | --version ]
```

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
```

BioHPC image



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

public image



- Image can be imported from a file

```
docker1 import filename
```

BioHPC Docker

To run a container:

- Pull or import image
- Start container from image

You can also try to start container without pulling first, Docker will pull it if not found locally. All images used on a server are cached in a local registry.

BioHPC Docker

Pull image

```
[jarekp@cbsum1c2b014 ~]$ docker1 pull docker.io/biohpc/cowsay
```

```
Using default tag: latest
```

```
Trying to pull repository docker.io/biohpc/cowsay ...
```

```
sha256:b4ec86cddb2d564d7ea94c9b49196f6b82e3c635a6581ee4eae02687e8ba91b8: Pulling from  
docker.io/biohpc/cowsay
```

```
08d48e6f1cff: Pull complete
```

```
a1aa994f5ff7: Pull complete
```

```
Digest: sha256:b4ec86cddb2d564d7ea94c9b49196f6b82e3c635a6581ee4eae02687e8ba91b8
```

```
Status: Downloaded newer image for docker.io/biohpc/cowsay:latest
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
docker.io/biohpc/cowsay	latest	195f168235c9	16 months ago	337.1 MB

```
[jarekp@cbsum1c2b014 ~]$
```

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 ~]$
```

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

```
image name    command  command arguments
[jarekp@cbsum1c2b014 ~]$ docker1 run biohpc/cowsay cowsay "This is Docker"
```

```
< This is Docker >
```

```
-----
      ^  ^
      (oo)\_____
      (__)\"       )\\/
           ||-----w |
           ||         |
           ||         |
```

List containers on this host

```
[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

Image exited – cannot connect to it anymore

BioHPC Docker

Interactive run

run interactively image name command to run

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -it biohpc_jarekp/cowsay /bin/bash
[root@a0017f5faf51 workdir]# pwd
/workdir
[root@a0017f5faf51 workdir]# cowsay "hi"
```

we are inside container now – as root!
Default directory is /workdir

```
< hi >
-----
```

```
  \      ^__^
   \      (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||
```

```
[root@a0017f5faf51 workdir]# exit
exit
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
a0017f5faf51	biohpc_jarekp/cowsay	"/bin/bash"	32 seconds ago	Exited (0) 6
1035e0f20e5f	biohpc/cowsay	"cowsay 'This is Dock'"	6 minutes ago	Exited (0) 6

```
jarekp__biohpc_2
jarekp__biohpc_1
[jarekp@cbsum1c2b014 ~]$
```

Image exited – cannot connect to it anymore

BioHPC Docker

Each container has a pre-defined link to host files

container `/workdir`

is the same as

host `/workdir/labid` (labid => your BioHPC id)

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 workdir]# 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 workdir]# 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 ~]$
```

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`

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 non-running containers

```
docker1 clean
```

Remove all containers – running or not

```
docker1 clean all
```

Docker repositories

Pull container from BioHPC repository

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

Pull container from Docker public repository

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

```
docker1 pull imagename
```

BioHPC Docker Images

Description	Repository image	File
CentOS 7 cowsay	biohpc/cowsay	/programs/docker/images/cowsay.tar
CentOS 7 development	biohpc/centos7dev	/programs/docker/images/centos7dev.tar
Ubuntu development	biohpc/ubuntudev	/programs/docker/images/ubuntudev.tar
Centos 7 development with GUI and sshd	biohpc/centos7devgui	/programs/docker/images/centos7devgui.tar

Images from public repository are usually minimal, i.e. very little additional software is installed. It is good if you want to use it for one purpose it has been designed for, but usually it is not good for installing software or general use.

You can start with public minimal image and install all software yourself, or start with one of BioHPC images and install only custom software.

BioHPC Docker Example: install TopHat

1. Start Ubuntu container in background
2. Connect to the container and install TopHat
3. Verify it runs, save output in /workdir, exit container
4. How about the resulting files? Where are they and how to get them?
5. Save the container as image for future use

BioHPC Docker Example: install TopHat

1. Start Ubuntu container in background

```
docker1 run -d -it docker.io/biohpc/ubuntuudev /bin/bash
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -d -it docker.io/biohpc/ubuntuudev /bin/bash
Unable to find image 'docker.io/biohpc/ubuntuudev:latest' locally
Trying to pull repository docker.io/biohpc/ubuntuudev ...
sha256:d03a0a1e2247895a3e57aa8cb5cdf1c6253f759b3daba78846af1d825e19cb75: Pulling from
docker.io/biohpc/ubuntuudev
```

```
87ad106e166e: Pull complete
```

```
Digest: sha256:d03a0a1e2247895a3e57aa8cb5cdf1c6253f759b3daba78846af1d825e19cb75
```

```
Status: Downloaded newer image for docker.io/biohpc/ubuntuudev:latest
```

```
ee4d845bb1633c3fe907fd3a2b217f5285bd3cfa9ca24955e4720c39f4eb8e67
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
ee4d845bb163	docker.io/biohpc/ubuntuudev	"/bin/bash"	15 minutes ago	Up 15 minutes

```
jarekp__biohpc_1
```

```
[jarekp@cbsum1c2b014 ~]$
```

BioHPC Docker Example: install TopHat

2. Connect to the container and install TopHat

```
docker1 exec -it containerid /bin/bash
apt-get install tophat
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 exec -it ee4d845bb163 /bin/bash
root@ee4d845bb163:/workdir# apt-get install tophat
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  dbus-x11 gconf2 libavahi-glib1 libbonobo2-0 libbonobo2-common libcanberra0 libgnome-2-0 libgnome2-common
libgnomevfs2-0
  libgnomevfs2-common liborbit-2-0 libtdb1 libvorbisfile3 sound-theme-freedesktop
[...]
Processing triggers for libc-bin (2.23-0ubuntu9) ...
Setting up libboost-thread1.58.0:amd64 (1.58.0+dfsg-5ubuntu3.1) ...
Setting up libtbb2:amd64 (4.4~20151115-0ubuntu3) ...
Setting up bowtie2 (2.2.6-2) ...
Setting up tophat (2.1.0+dfsg-1build1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
root@ee4d845bb163:/workdir#
```

BioHPC Docker Example: install TopHat

3. Verify it runs, save output in /workdir, exit container

```
root@ee4d845bb163:/workdir# tophat -h
```

```
tophat:
```

```
TopHat maps short sequences from spliced transcripts to whole genomes.
```

```
Usage:
```

```
tophat [options] <bowtie_index> <reads1[,reads2,...]> [reads1[,reads2,...]] \  
[quals1,[quals2,...]] [quals1[,quals2,...]]
```

```
Options:
```

```
[...]
```

```
--rg-date           <string>      (ISO 8601 date of the sequencing run)  
--rg-platform       <string>      (Sequencing platform descriptor)
```

```
for detailed help see http://tophat.cbcb.umd.edu/manual.html
```

```
root@ee4d845bb163:/workdir# tophat -h >& /workdir/tophat.help
```

```
root@ee4d845bb163:/workdir# ls -al /workdir
```

```
total 8  
drwxr-xr-x  2 4965 root   24 May 22 22:14 .  
drwxr-xr-x 22 root root  257 May 22 21:52 ..  
-rw-r--r--  1 root root 6620 May 22 22:15 tophat.help
```

```
root@ee4d845bb163:/workdir# exit
```

```
exit
```

```
[jarekp@cbsum1c2b014 ~]$
```

BioHPC Docker Example: install TopHat

4. How about the resulting files? Where re they and how to get them?

```
[jarekp@cbsum1c2b014 ~]$ ls -al /workdir/jarekp
total 8
drwxr-xr-x  2 jarekp root   24 May 22 18:14 .
drwxrwxrwx. 4 root   root   30 May 22 13:26 ..
-rw-r--r--  1 root   root  6620 May 22 18:15 tophat.help
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 claim
```

```
[jarekp@cbsum1c2b014 ~]$ ls -al /workdir/jarekp
total 8
drwxr-xr-x  2 jarekp root   24 May 22 18:14 .
drwxrwxrwx. 4 root   root   30 May 22 13:26 ..
-rw-r--r--  1 jarekp root  6620 May 22 18:15 tophat.help
[jarekp@cbsum1c2b014 ~]$
```

Files generated in Docker may belong to root or other system users. You need to use **docker claim** to get permissions to deal with them.

BioHPC Docker Example: install TopHat

5. Save the container as image for future use

```
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
ee4d845bb163	docker.io/biohpc/ubuntudev	"/bin/bash"	28 minutes ago	Up 28 minutes
jarekp__biohpc_1				

```
[jarekp@cbsum1c2b014 ~]$ docker1 export -o /home/jarekp/mydockerimage.tar ee4d845bb163
```

```
[jarekp@cbsum1c2b014 ~]$ ls -alh /home/jarekp/mydockerimage.tar
```

```
-rw----- 1 jarekp jarekp 1.2G May 22 18:22 /home/jarekp/mydockerimage.tar
```

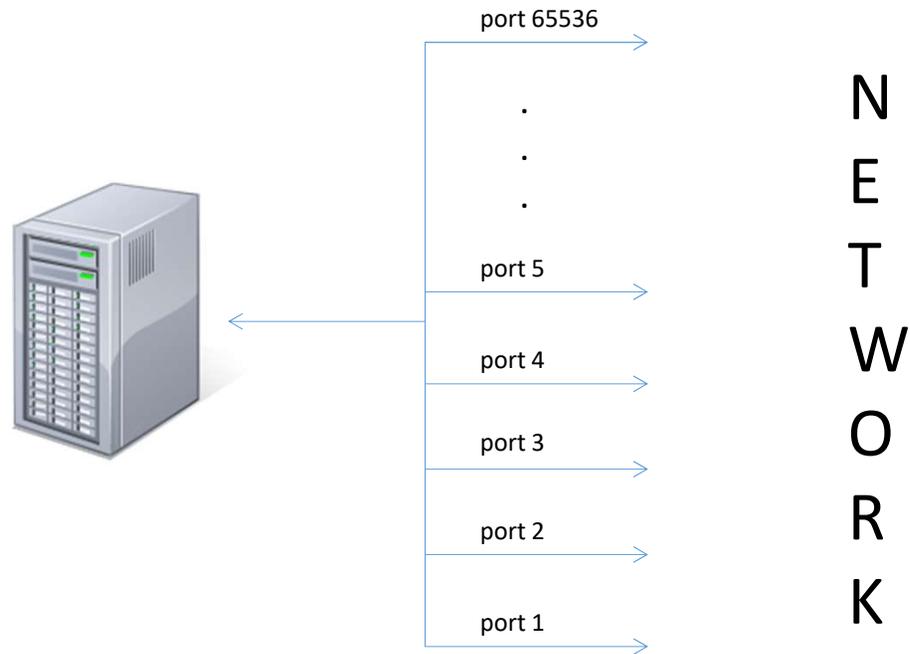
```
[jarekp@cbsum1c2b014 ~]$
```

BioHPC Docker - network ports

Each service on the network is referenced by two values

1. Server address (i.e. IP, name etc)
2. Service port (a number referencing network socket to connect to).

BioHPC Docker - network ports



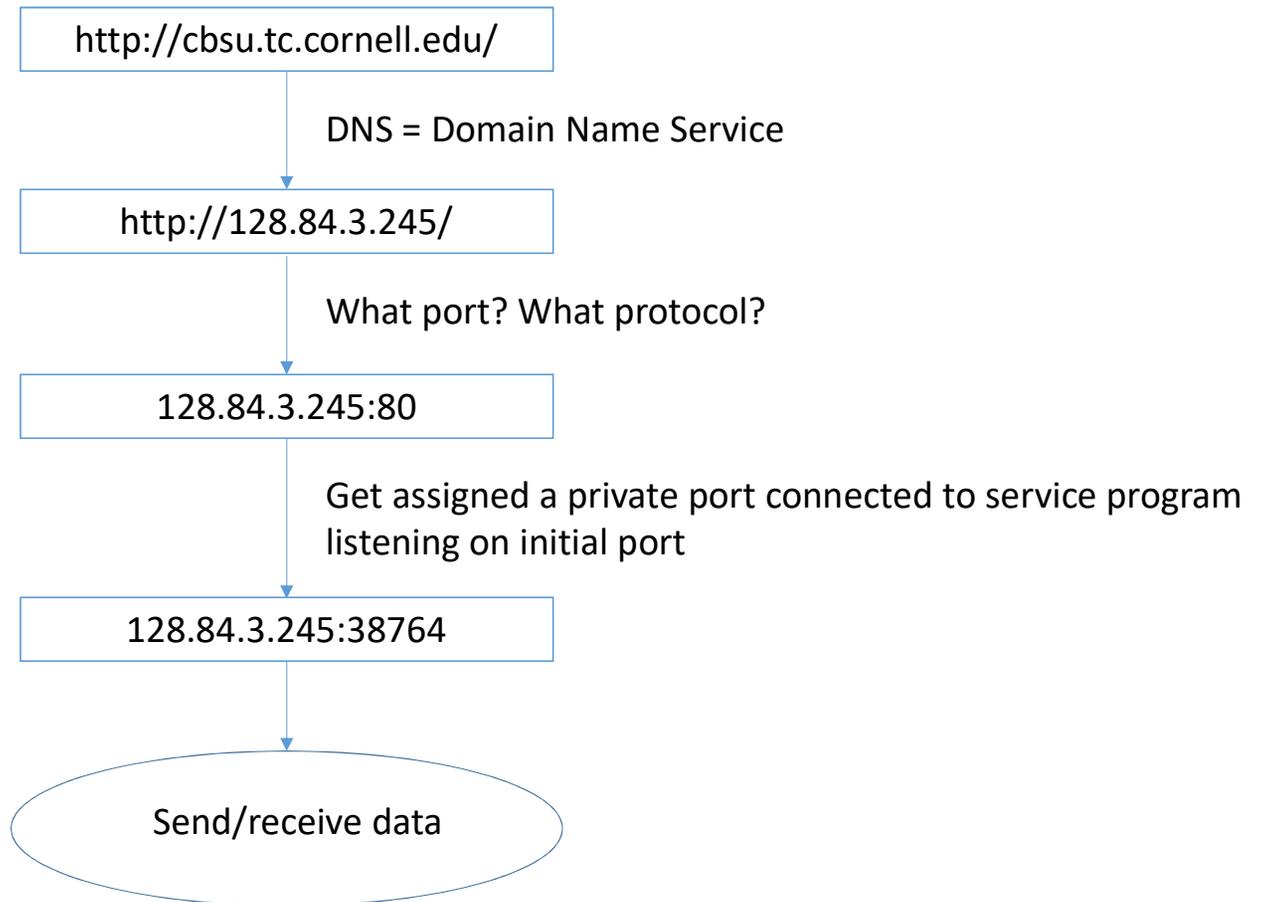
Computer full address: ip_number:port

i.e. 128.8.3.22:22

BioHPC Docker - network ports

Service (protocol)	Port
FTP	20 and 21
TELNET	23
SSH	22
SMTP (mail service)	25
DNS (domain name system)	53
HTTP (www)	80
HTTPS (www secure)	443

BioHPC Docker - network ports



BioHPC Docker - network ports

Containers can offer network services, and you can connect to them if you know what is container IP and service port.

You can also map container port to host port. Then you can connect to your host and reference assigned port number to access container service.

BioHPC Docker – running graphics inside container

We have prepared image with X-Windows and development tools – `biohpc/centos7devgui`

It can be run in server mode – i.e. allowing users to log in via ssh.

Then X-Windows ports can be mapped such users can forward graphics to their client computers.

BioHPC Docker – running graphics inside container

```
docker1 pull biohpc/centos7devgui
```

```
docker1 run -d -p 127.0.0.1:5000:22 -P -t biohpc/centos7devgui
```

mapping port 22 (ssh) of container to port 5000 on host **internal** network

no command – it is a special image with pre-defined command

```
ssh -X root@localhost -p 5000
```

With port mapping you can run various services including database servers and websites.

BioHPC Docker - network ports

You can map container ports to external host ports, but they need to be opened in firewall to be accessible.

We keep ports 8009 – 8019 open for campus access.

```
docker1 run -d -p 128.84.43.165:8009:22 -P -t biohpc/centos7devgui
```

This command will map ssh port to external port on the host so you can access the container from anywhere on the campus

```
ssh -X root@cbsum1c2b014.tc.cornell.edu -p 8009
```

BioHPC Docker Example – install MAFFT

1. Check software website for instructions.
<https://mafft.cbrc.jp/alignment/software/>
2. Decide what type of image is needed (CentOS, Ubuntu, public minimal or development etc).
3. Follow instructions.
4. Save resulting container for future use.

BioHPC Docker Example – install MAFFT

MAFFT version 7
Multiple alignment program for amino acid or nucleotide sequences

Download version
[Mac OS X](#)
[Windows](#)
[Linux](#)
[Source](#)

Online version
[Alignment](#)
[mafft --add](#)
[Merge](#)
[Phylogeny](#)
[Rough tree](#)

[Merits / limitations](#)
[Algorithms](#)
[Tips](#)
[Benchmarks](#)
[Feedback](#)

[Follow](#)

Contact email address, kazutaka.katoh@aist.go.jp, is temporarily unavailable from 2018/Feb/7. If you sent an email to this address but have received no response, then please re-send the email to katoh@ifrec.osaka-u.ac.jp.

About

MAFFT is a multiple sequence alignment program for unix-like operating systems. It offers a range of multiple alignment methods, L-INS-i (accurate; for alignment of <~200 sequences), FFT-NS-2 (fast; for alignment of <~30,000 sequences), etc.

Download and Installation

- [Mac OS X](#)
- [Linux](#)
- [Windows](#) **Updated!**
- [Source](#)
- [Changelog](#)

The latest version is 7.397 (2018/Apr/16).

BioHPC Docker Example – install MAFFT

MAFFT version 7
Multiple alignment program for amino acid or nucleotide sequences

Linux versions

Select either of:

- .rpm
 - [mafft-7.397-gcc_fc6.x86_64.rpm](#) (Built on kernel 2.6)

Root privileges are necessary.

```
% su -  
Password: *****  
# rpm -Uvh mafft-x.xx-xxx.xxx.rpm  
# exit  
% rehash (if necessary)
```

A shell script, mafft, is installed into /usr/bin/ and some binaries are installed into /usr/libexec/mafft/.

License

- .deb
 - [mafft_7.397-1_amd64.deb](#) (Built on kernel 2.6)

Root privileges are necessary.

Callout: Docker is essential here!

BioHPC Docker Example – install MAFFT

Let's install .rpm package – short Google search reveals it is linked to Red Hat and CentOS, among others.

Let's use basic image – if something goes wrong we can do it again!

We can download the RPM and place it on our machine in /workdir/labid

https://mafft.cbrc.jp/alignment/software/mafft-7.397-gcc_fc6.x86_64.rpm

BioHPC Docker Example – install MAFFT

```
[jarekp@cbsum1c2b014 ~]$ ls -al /workdir/jarekp
total 13924
drwxr-xr-x  2 jarekp root          61 May 22 18:42 .
drwxrwxrwx. 4 root  root          30 May 22 13:26 ..
-rw-r--r--  1 jarekp pmm262 14249143 Apr 15 21:49 mafft-7.397-gcc_fc6.x86_64.rpm
-rw-r--r--  1 jarekp root          6620 May 22 18:15 tophat.help
```

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -d -t centos /bin/bash
Unable to find image 'centos:latest' locally
Trying to pull repository dtr.cucloud.net/centos ...
Trying to pull repository docker.io/library/centos ...
sha256:989b936d56b1ace20ddf855a301741e52abca38286382cba7f44443210e96d16: Pulling from
docker.io/library/centos
469cfcc7a4b3: Pull complete
Digest: sha256:989b936d56b1ace20ddf855a301741e52abca38286382cba7f44443210e96d16
Status: Downloaded newer image for docker.io/centos:latest
c40a7a8f3fd3094b1da391a1a502595f9a5a8362e69ce01f05d765a87a88f85b
```

```
[jarekp@cbsum1c2b014 ~]$ [jarekp@cbsum1c2b014 ~]$ docker1 ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c40a7a8f3fd3	centos	"/bin/bash"	29 seconds ago	Up 27 seconds

```
jarekp__biohpc_2
[jarekp@cbsum1c2b014 ~]$
```

BioHPC Docker Example – install MAFFT

```
[jarekp@cbsum1c2b014 ~]$ docker1 exec -it ee4d845bb163 /bin/bash
root@ee4d845bb163:/workdir# ls -al /workdir
total 13924
drwxr-xr-x  2 4965 root      61 May 22 22:42 .
drwxr-xr-x 22 root root      257 May 22 21:52 ..
-rw-r--r--  1 4965 4963 14249143 Apr 16 01:49 mafft-7.397-gcc_fc6.x86_64.rpm
-rw-r--r--  1 4965 root      6620 May 22 22:15 tophat.help

root@ee4d845bb163:/workdir# rpm -Uvh mafft-7.397-gcc_fc6.x86_64.rpm
bash: rpm: command not found
root@ee4d845bb163:/workdir#
```

Apparently rpm command is not installed in minimal CentOS. We will either need to install it or start over with better CentOS image.

Let's start over with biohpc/centos7dev. This image was developed specifically to have all necessary development and installation tools ready.

BioHPC Docker Example – install MAFFT

```
[[jarekp@cbsum1c2b014 ~]$ docker1 run -d -t docker.io/biohpc/centos7dev /bin/bash
Unable to find image 'docker.io/biohpc/centos7dev:latest' locally
Trying to pull repository docker.io/biohpc/centos7dev ...
sha256:702a8b0c44e5a0b7a78bf3f775286f5010df5817c59cf3106d629cfa032cbc38: Pulling from
docker.io/biohpc/centos7dev
```

```
341952473e2d: Pull complete
c499a99a598e: Pull complete
Digest: sha256:702a8b0c44e5a0b7a78bf3f775286f5010df5817c59cf3106d629cfa032cbc38
Status: Downloaded newer image for docker.io/biohpc/centos7dev:latest
31e5b0fcb7b89d18265762eec964253ea81912549b6cfc117971c8a70b13337d
```

```
[[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
31e5b0fcb7b8	docker.io/biohpc/centos7dev	"/bin/bash"	15 hours ago	Up 15 hours

```
jarekp__biohpc_1
[[jarekp@cbsum1c2b014 ~]$
```

BioHPC Docker Example – install MAFFT

```
[[jarekp@cbsum1c2b014 ~]$ docker1 exec -it 31e5b0fcb7b8 /bin/bash
[root@31e5b0fcb7b8 workdir]# rpm -Uvh mafft-7.397-gcc_fc6.x86_64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:mafft-7.397-gcc_fc6 ##### [100%]
```

```
[root@31e5b0fcb7b8 workdir]# mafft
```

MAFFT v7.397 (2018/Apr/16)

MBE 30:772-780 (2013), NAR 30:3059-3066 (2002)
<https://mafft.cbrc.jp/alignment/software/>

Input file? (fasta format)

@
[...]

Remember to save the image! Your container WILL be deleted after you reservation ends.



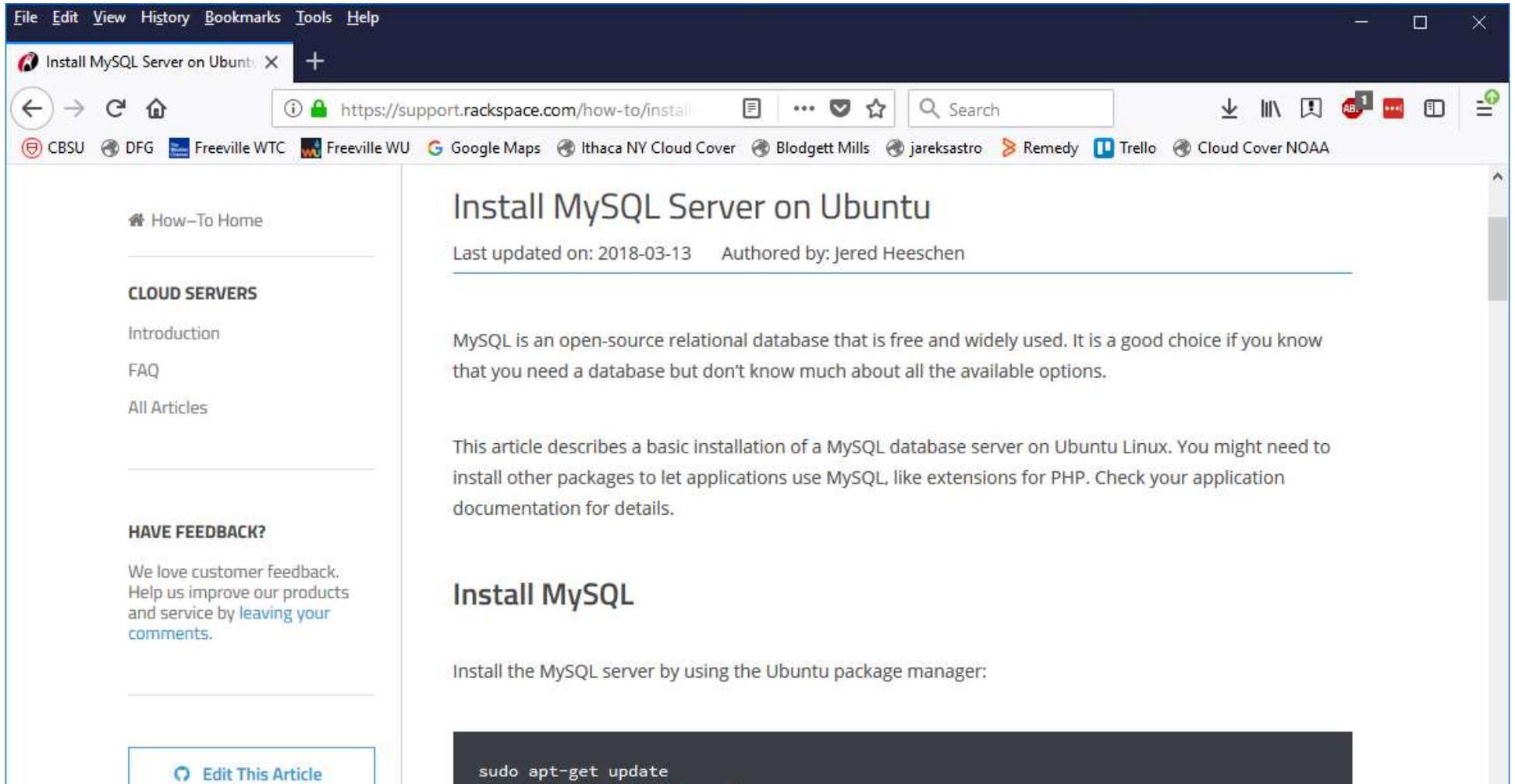
```
[[jarekp@cbsum1c2b014 ~]$ docker1 export -o /home/jarekp/mafft_image.tar 31e5b0fcb7b8
```

BioHPC Docker Example – install MySQL Database

1. Search online for instructions and choose ones best suited for your goal.
2. Pull appropriate image you want to work with
3. Follow instructions to install MySQL
4. Start database server in your container
5. Configure database server so it is accessible for host machine
6. Save the image. Now you have a database server you can move between machines you use! Databases are stored inside the image.

BioHPC Docker Example – install MySQL Database

<https://support.rackspace.com/how-to/installing-mysql-server-on-ubuntu/>



The screenshot shows a web browser window with the following elements:

- Browser Menu:** File, Edit, View, History, Bookmarks, Tools, Help.
- Address Bar:** <https://support.rackspace.com/how-to/installing-mysql-server-on-ubuntu/>
- Page Title:** Install MySQL Server on Ubuntu
- Metadata:** Last updated on: 2018-03-13 | Authored by: Jered Heeschen
- Introduction:** MySQL is an open-source relational database that is free and widely used. It is a good choice if you know that you need a database but don't know much about all the available options.
- Summary:** This article describes a basic installation of a MySQL database server on Ubuntu Linux. You might need to install other packages to let applications use MySQL, like extensions for PHP. Check your application documentation for details.
- Section Header:** Install MySQL
- Text:** Install the MySQL server by using the Ubuntu package manager:
- Code Block:**

```
sudo apt-get update
```
- Left Sidebar:**
 - How-To Home
 - CLOUD SERVERS**
 - Introduction
 - FAQ
 - All Articles
 - HAVE FEEDBACK?**
 - We love customer feedback. Help us improve our products and service by [leaving your comments](#).
 - [Edit This Article](#)

BioHPC Docker Example – install MySQL Database

Note: many tutorials suggest using `sudo` command before actual commands. This is only necessary if you are NOT root user, and therefore should NOT be used in Docker containers – you ARE root there already. Just skip this prefix:

`sudo apt-get update` => `apt-get update`

```
[jarekp@cbsum1c2b014 ~]$ docker1 run -d -t docker.io/biohpc/ubuntu dev /bin/bash
c8cb131bd2deab36221db92531627609b32666eccbd9237c7ed4973028cd921e
[jarekp@cbsum1c2b014 ~]$ docker1 ps -a
CONTAINER ID        IMAGE                                COMMAND                                CREATED            STATUS
PORTS              NAMES
c8cb131bd2de       docker.io/biohpc/ubuntu dev         "/bin/bash"         9 seconds ago     Up 7 seconds
jarekp__biohpc_1
[jarekp@cbsum1c2b014 ~]$ docker1 exec -it c8cb131bd2de /bin/bash
root@c8cb131bd2de:/workdir# apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
[...]
Fetched 4024 kB in 8s (500 kB/s)
Reading package lists... Done
root@c8cb131bd2de:/workdir#
```

BioHPC Docker Example – install MySQL Database

```
root@c8cb131bd2de:/workdir# apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

```
[...]
```

```
Configuring mysql-server-5.7
```

```
-----
```

While not mandatory, it is highly recommended that you set a password for the MySQL administrative "root" user.

If this field is left blank, the password will not be changed.

```
New password for the MySQL "root" user:
```

```
Repeat password for the MySQL "root" user:
```

Decide what password you want for MySQL admin account



```
Unpacking mysql-server-5.7 (5.7.22-0ubuntu0.16.04.1) ...
```

```
[...]
```

BioHPC Docker Example – install MySQL Database

```
Setting up mysql-server (5.7.22-0ubuntu0.16.04.1) ...  
Processing triggers for libc-bin (2.23-0ubuntu9) ...  
Processing triggers for systemd (229-4ubuntu17) ...  
root@c8cb131bd2de:/workdir#
```

```
root@c8cb131bd2de:/workdir# systemctl start mysql  
Failed to connect to bus: No such file or directory  
root@c8cb131bd2de:/workdir#
```

```
root@c8cb131bd2de:/workdir# /etc/init.d/mysql start  
* Starting MySQL database server mysqld  
No directory, logging in with HOME=/
```

```
[ OK ]  
root@c8cb131bd2de:/workdir#
```

Something is wrong! Container cannot start MySQL in a server way.

Google search reveals containers have this problem and workaround is to start the service manually.

We are not creating a “bootable” image, this is more complicated and entirely different subject.

You will have to start the service manually each time the container is run. Not a problem here.

BioHPC Docker Example – install MySQL Database

```
root@c8cb131bd2de:/workdir# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
4 rows in set (0.00 sec)

mysql> exit
Bye
root@c8cb131bd2de:/workdir#
```

BioHPC Docker Example – install MySQL Database

Now we have running MySQL server in a container and we can use it INSIDE container. We would like to have MySQL server that can be accessed from the host, this way is more convenient and allows for software or pipelines to use it too.

However by default MySQL is only accessible locally, we need to change MySQL configuration to do so. Google “allow MySQL remote network connections”. There are two suggestions:

1. Comment out line “bind” in [/etc/mysql/mysql.conf.d/mysqld.cnf](#)

```
bind-address      = 127.0.0.1    =>    #bind-address      = 127.0.0.1
```

After that we need to restart our MySQL server:

```
/etc/init.d/mysqld restart
```

2. Add remote access privileges to our “root” account

BioHPC Docker Example – install MySQL Database

2. Add remote access privileges to our “root” account

check if root has restricted access

```
SELECT host FROM mysql.user WHERE User = 'root';
```

then fix it

```
CREATE USER 'root'@'%' IDENTIFIED BY 'our_root_password';  
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
```

The above commands need to be executed inside “mysql -u root -p”

All this info can be found using Google, commands can be copied. Some thinking is still necessary 😊.

BioHPC Docker Example – install MySQL Database

```
mysql> SELECT host FROM mysql.user WHERE User = 'root';
```

```
+-----+  
| host   |  
+-----+  
| localhost |  
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> CREATE USER 'root'@'%' IDENTIFIED BY 'docker';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT host FROM mysql.user WHERE User = 'root';
```

```
+-----+  
| host   |  
+-----+  
| %      |  
| localhost |  
+-----+
```

```
2 rows in set (0.00 sec)
```

```
mysql>
```

BioHPC Docker Example – install MySQL Database

Before we can connect to our MySQL database remotely we need to figure out what is our container IP address.

Docker maintains an internal network inside host usually 172.17.0.*.

Use command “`docker inspect containerid`” to find out more.

```
[jarekp@cbsum1c2b014 ~]$ docker inspect c8cb131bd2de
```

```
[
  {
    "Id": "c8cb131bd2deab36221db92531627609b32666eccbd9237c7ed4973028cd921e",
    "Created": "2018-05-23T15:20:40.129886637Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Pid": 1,
      "ExitCode": 0,
      "Error": "",
      "ContainerID": "c8cb131bd2deab36221db92531627609b32666eccbd9237c7ed4973028cd921e"
    },
    "Image": "alpine:latest",
    "NetworkSettings": {
      "Bridge": "docker0",
      "Gateway": "172.17.0.1",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "IPAddress": "172.17.0.2",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "MacAddress": "02:42:ac:11:00:02",
      "Networks": {
        "docker0": {
          "IPAMConfig": {
            "IPv4Address": "172.17.0.2",
            "IPv4PrefixLen": 16,
            "IPv6Address": "",
            "IPv6PrefixLen": 0
          },
          "Links": null,
          "MacAddress": "02:42:ac:11:00:02",
          "Name": "docker0",
          "NetworkID": "17217002",
          "Options": null,
          "Type": "bridge"
        }
      }
    }
  }
]
```

Here it is!

BioHPC Docker Example – install MySQL Database

```
[jarekp@cbsum1c2b014 ~]$ mysql -u root -p -h 172.17.0.2
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.22-0ubuntu0.16.04.1 (Ubuntu)
```

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
MySQL [(none)]> show databases;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| sys               |
+-----+
4 rows in set (0.00 sec)
```

```
MySQL [(none)]>
```

BioHPC Docker Example – install MySQL Database

Now we can save the image for future use.

Each time we need to start it in the background,

connect to it with “docker1 exec” start MySQL with “/etc/init.d/mysql start”

and then exit container and use the database.

```
[jarekp@cbsum1c2b014 ~]$ docker1 export -o /home/jarekp/mysql_docker.tar c8cb131bd2de
```