

Quick Start: How to do some common tasks?

There are few terms you need to know about Docker:

Docker Image: A Docker template file, e.g. a standard Ubuntu Linux image;

Docker container: A running instance of the Docker image;

Shell of a container: A Linux shell to access a Docker container;

Docker file: A text script with instructions to construct a Docker image;

docker1: A BioHPC wrapper script for docker. On BioHPC computers, you must run docker through docker1. The docker1 script would examine your docker commands to make sure it is safe to use on BioHPC. The docker1 script restricts the "-v" and some other parameters, and make "/workdir" as the default directory in container.

If you still do not know about the meaning of the terms, do not worry. They will start to make sense after some practices.

Task 1: Install the software in a Docker container, which can be saved in a new Docker image and used later.

- **Step 1.** Import a template Docker image. This step could take a few minutes. I am using "Ubuntu" as an example here. The "docker1 images" command can be used to check images that have already been imported on your machine.

```
docker1 pull biohpc/ubuntudev
docker1 images
```

- **Step 2.** Start a Docker Container (an instance of the image). You need to replace "docker.io/biohpc/ubuntudev" in the example below with the actual image name on your machine. The "docker1 images" command would tell you the names of the available images. The "-d" option in this step is to make sure that the container will run in the background. Use the "docker1 ps -a" command to check containers that are currently running on your machine.

```
docker1 run -d -it docker.io/biohpc/ubuntudev /bin/bash
docker1 ps -a
```

- **Step 3.** Start an interactive shell to access the Docker Container. To do this, you need the container name, which is a string that looks like "d3b7d7463857". Replace d3b7d7463857 in this example with the actual container name. Use the "docker1 ps -a" command to get the container name. After login, you might want to run update and upgrade to update all the packages. The update could take an hour, depending on how old is the image in the Docker hub repository.

```
docker1 exec -it d3b7d7463857 /bin/bash

apt-get update

apt-get upgrade
```

- **Step 4.** After step 3, you are in an interactive shell of a Docker container. As you have root privilege here, you can install any software you like, /usr/bin. **Do not install software in /workdir in the Docker container**, as the /workdir is a directory mounted from the host system. (It is the /workdir/<userID> directory on the host). Sometime, a software could take a long time to install. If your internet connection got interrupted in the middle when installing a software. Do not worry, the container is still running in the background. You can start a new ssh session, and use the command "docker1 exec -it d3b7d7463857 /bin/bash" to access the same container.
- **Step 5.** Save the container in a new Docker image file in your home directory, so that you can use this image (with system updated and software installed) later on any other computers. To do this, first you need to exit the interactive shell to the Docker container, then use the "docker1 export" command. The saved image can be imported later with "docker1 import /home/qisun/myfile.tar". The image file for "Ubuntu Dev." is about 1.2GB.

```
exit

docker1 export -o /home/qisun/myfile.tar d3b7d7463857
```

Task 2: Run the software in Docker.

- **Step 1.** Import the Docker image to your computer.

```
docker1 images

docker1 import /home/qisun/myfile.tar
```

- **Step 2.** Start a Docker Container with the image. This step is the same as step 2 in task 1.

```
docker1 run -d -it biohpc_qisun/ubuntu-dev /bin/bash

docker1 ps -a
```

- **Step 3.** Run the software. There are two alternative ways to run a Linux software in Docker.

1. Interactive mode with the "-it" option (This might be easier for some people)
Use the following command to start an interactive shell. After this, you are in the container. You can run your commands within Docker container. Remember, you are "root" user in container.

```
docker1 exec -it d3b7d7463857 /bin/bash
```

To detach from Docker container, simply type "exit". After detaching, your job is still running in the container. You can re-connect with the container using the same "docker1 exec" command.

2. Direct mode:

If the software does not take much time to finish,

```
docker1 exec d3b7d7463857 mycommand
```

If the software takes a long time to finish, you need to run it in background. For example:

```
docker1 exec d3b7d7463857 /bin/bash -c "bwa aln mydata >& log" &
```

To check whether a software is finished in Docker, use this command:

```
docker1 exec d3b7d7463857 ps -ef
```

Task 3: Remove docker containers and images.

1. Remove all containers owned by you (Use the command "docker1 ps -a" to verify).

```
docker1 ps -a  
docker1 clean all  
docker1 ps -a
```

2. Stop and remove one container.
docker1 stop <containerID>

```
docker1 rm <containerID>
```

3. Remove a docker image

```
docker1 images
```

```
docker1 rmi <imageID>
```

Task 4: Run Docker image or container from docker hub.

```
docker1 pull <docker_hub_image_name>
```

When using docker on BioHPC, there are a few things you need to be aware of

- a. You need to use "docker1" instead of "docker" command;
- b. The docker1 command only allows you to mount directories under "/workdir/<userID>/" into docker container;
- c. The docker1 command makes "/workdir" as the default in docker container. If the software requires a different default directory, either use "-w <default dir>" to explicitly set the default directory when launching "docker1 run" command, or use the "--noworkdir" parameter to not set default directory;
- d. If the developer provides you with a docker file, we recommend to use "docker build" to create an image file. Many developers do not clarify software versions in docker file which could cause non-reproducible docker images.

Some tips:

1. You can have multiple shells to access the same Docker container. Use "docker1 exec d3b7d7463857 mycommand" or "docker1 exec -it d3b7d7463857 /bin/bash" to connect to a Docker container.
2. If you reserve a machine from BioHPC Lab, the Docker containers and images will be deleted after the reservation ends. If you want to re-use the container later, make sure to export the container to a new image file and save the new image in your home directory.
3. If something goes wrong, you can always remove the containers using the "docker1 clean" or "docker1 clean all" command. The "docker1 clean" removes the idle containers, while "docker1 clean all" removes all containers.