

# Docker

- What is Docker and why do we need it?
- How to build a Docker image?
- How to run software in Docker?
- Docker on BioHPC.

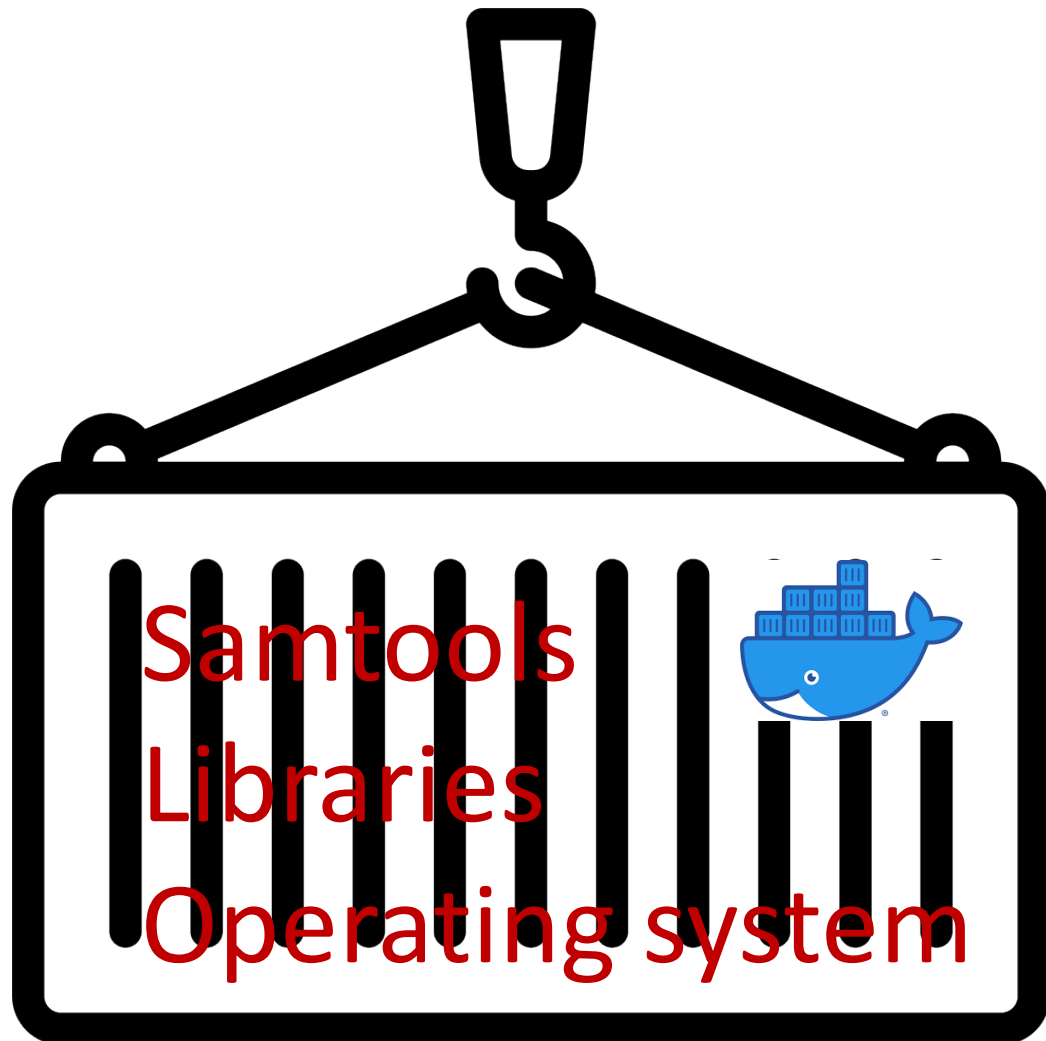
- What is Docker and why do we need it?

**A software, e.g., samtools, is dependent on external libraries of the Linux system.**

**samtools**

```
$ which samtools
/programs/samtools-1.15.1/bin/samtools

$ ldd /programs/samtools-1.15.1-r/bin/samtools
linux-vdso.so.1 => (0x00007ffd2fdd4000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f60857ea000)
libz.so.1 => /lib64/libz.so.1 (0x00007f60855d4000)
libm.so.6 => /lib64/libm.so.6 (0x00007f60852d2000)
libbz2.so.1 => /lib64/libbz2.so.1 (0x00007f60850c2000)
liblzma.so.5 => /lib64/liblzma.so.5 (0x00007f6084e9c000)
libcurl.so.4 => /lib64/libcurl.so.4 (0x00007f6084c33000)
libcrypto.so.10 => /lib64/libcrypto.so.10 (0x00007f60847d1000)
libncursesw.so.5 => /lib64/libncursesw.so.5 (0x00007f6084599000)
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x00007f608436f000)
libc.so.6 => /lib64/libc.so.6 (0x00007f6083fa1000)
/lib64/ld-linux-x86-64.so.2 (0x00007f6085a06000)
libidn.so.11 => /lib64/libidn.so.11 (0x00007f6083d6e000)
libssh2.so.1 => /lib64/libssh2.so.1 (0x00007f6083b41000)
```



## A docker image file:

**Software:** samtools

**Operating system:** Ubuntu v20.04

**Libraries:** libz, libcurl, liblzma, ...

### Cons:

- File size is a lot bigger
  - samtools: 7.5 mb;
  - a Docker image with samtools >100 mb);

### Pros:

- Easier to install;
- Reproducible;

- How to build a Docker image?

# Terminology

## Docker image

A software file,  
including OS, libraries  
and executables.

(typical size 0.1-2.0 gb)

Deposited in  
Dockerhub

## Docker container

A running instance of  
the image.

Running on your  
computer. Removed  
when power down.

## Dockerfile

A script text file with  
instructions how a  
image is build.

(typical size <10 kb)

Deposited  
in Github

# Two ways to build a Docker image

## Dockerfile








- Write a script file named Dockerfile;
- Build the image;
- Upload the image to Dockerhub or save as a tar file;

## Interactive

- Download a base image;
- Start a container;
- Install software in a container;
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file

# Building a docker image:

Step 1. Find a base image (or parent image)

Basic	Ubuntu (18.04, 20.04)  *
R	Rocker 
GPU	nvidia/cuda, tensorflow/tensorflow, pytorch/pytorch   
Conda	Continuumio/miniconda3, mambaorg/micromamba  

Find a base image from the Dockerhub web site (<https://hub.docker.com/>)

The screenshot shows the Docker Hub interface for the `rocker/rstudio` image. The page title is `rocker/rstudio`, with a star icon and a pull count of 5M+. The publisher is `rocker`, updated 16 hours ago. The image is described as "RStudio Server image".

Annotations highlight key elements:

- publisher/image\_name**: Points to the `rocker/rstudio` text.
- Filter by version**: Points to the search filter `3.5` in the tags section.
- "docker pull" command**: Points to the `docker pull rocker/rstudio:3.5` command button.

The tags section shows a list of versions with their respective digests, OS/ARCH, and compressed sizes:

TAG	DIGEST	OS/ARCH	COMPRESSED SIZE
<a href="#">3.5</a> Last pushed 3 years ago by <a href="#">rocker</a>	<a href="#">7e879738c30f</a>	linux/amd64	416.8 MB
<a href="#">3.5.3</a> Last pushed 3 years ago by <a href="#">rocker</a>	<a href="#">7e879738c30f</a>	linux/amd64	416.8 MB
<a href="#">3.5.2</a> Last pushed 4 years ago by <a href="#">rocker</a>			

## Interactive way to build a image

### Interactive

- Download a base image;
- Start a container;
- Install software in a container;
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file



```
docker pull ubuntu:20.04
```

# pull image from dockerhub

```
docker images
```

# list images on the computer

```
docker run -dit ubuntu:20.04
```

# start a container (-dit: detached, run in background and interactive)

```
docker ps -a
```


# list containers on the computer

- If the version is skipped, default to “latest” version;
- If “docker pull” is skipped, “docker run” would automatically do “pull”;



## Interactive way to build a image

### Interactive

- Download a base image;
- Start a container;
- Install software in a container; 
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file

```
docker exec -it c4dca29b2393 bash
```

```
### shell in container ###
```

```
apt update
```

```
apt install -y nano wget git
```

```
apt install -y build-essential
```

```
apt install -y python3-pip
```

```
apt install -y software-properties-common
```

```
export DEBIAN_FRONTEND=noninteractive
```

```
export TZ=America/New_York
```

```
apt install -y tzdata
```

```
apt install -y samtools
```

**Start a shell inside the container**

**Update the package list from Ubuntu software repository.  
(apt = apt-get)**

**Install commonly used tools and libraries (only install things you need)**

**Set timezone for the container. So that installer would not quiz you later.**

**Install software managed by Ubuntu**

```
docker exec -it c4dca29b2393 bash

### shell in container ###
apt update

apt install -y nano wget git
apt install -y build-essential
apt install -y python3-pip
apt install -y software-properties-common

export DEBIAN_FRONTEND=noninteractive
export TZ=America/New_York
apt install -y tzdata

apt install -y samtools
```

You are “root” in the Docker container, that means you can install anything with no restrictions

#install python tools

```
pip install deeptools
```

#install from source code

```
git clone https://github.com/arq5x/bedtools2
```

```
cd bedtools2
```

```
apt install -y zlib1g-dev libbz2-dev liblzma-dev
```

```
make -j4
```

```
make install
```

```
cd ..
```

```
rm -fr bedtools
```

# Summary of software installation methods

## Package management system

Installation tool	Package management system	Language
apt-get (apt)	Ubuntu	Misc.
pip	PyPI	Python
install.packages	CRAN	R
BiocManager	Bioconductor	R
devtools	Github	R
cpan	CPAN	PERL
conda & mamba	Conda	Misc.

\* New package managements make it easier to install a particular version of software/libraries.

## Compile from source

Using GCC for compilation.

## Interactive way to build a image

### Interactive

- Download a base image;
- Start a container;
- Install software in a container;
- Commit container to a new image;
- Upload the image to Dockerhub or save as a tar file

#commit the container to a new Docker image

```
docker commit c4dca29b2393 myapp
```

```
docker images
```

#save your image to a tar file

```
docker save -o myapp.tar myapp
```

#publish your new image to the Docker hub

```
docker login -u user_id  
docker tag image_id user_id/myapp  
docker push user_id/myapp
```

## Using Dockerfile to build a image

### Dockerfile

- Write a script named Dockerfile;
- Build the image;
- Upload the image to Dockerhub or save as a tar file;

```
FROM ubuntu:20.04
COPY readme.txt /root/
ADD https://github.com/arq5x/bedtools2/releases/download/v2.30.0/bedtools-2.30.0.tar.gz /root/
RUN apt update
RUN apt install -y build-essential python3 && \
    apt install -y zlib1g-dev libbz2-dev liblzma-dev && \
    cd /root && \
    tar xvfz bedtools2 && \
    cd bedtools-2.30.0 && \
    ln -s /usr/bin/python3 /usr/bin/python && \
    make -j8 && \
    make install
```

```
docker build -t myapp /workdir/qisun/mydir
```

**FROM:** define base image  
**COPY:** copy local file into image;  
**ADD:** download an internet file;  
**RUN:** run Linux shell command

Name of the  
new image

The directory where  
Dockerfile is located

- How to run software in Docker?

```
docker images
```

```
docker run --rm myimage bedtools -h
```

Remove the container after the job finishes.

Name of the image

Command line.

# Access data files and result files on the host server

```
docker run --rm -v /workdir/qisun/mydata:/data myimage \  
  samtools view -b -o /data/sample2.bam /data/sample2.sam
```

```
docker run --rm -v /workdir/qisun/mydata:/data -w /data myimage \  
  samtools view -b -o sample2.bam sample2.sam
```

**-v /workdir/qisun/mydata:/data**

Mount /workdir/qisun/mydata (host) to /data (container)

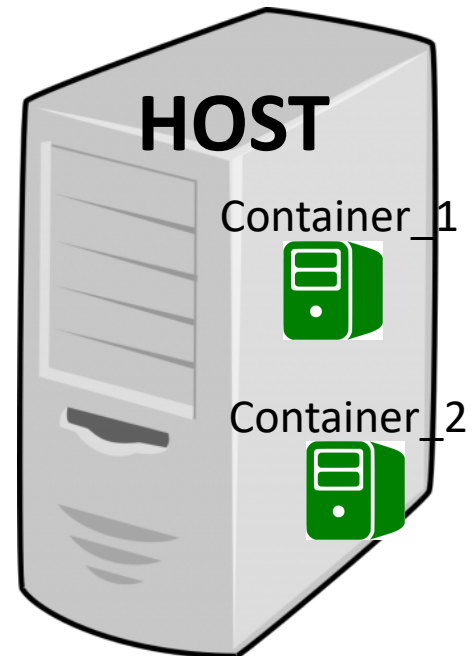
**-w /data**

Default current directory /data

## Shortcut

```
export SAMTOOLS="docker run --rm -v /workdir/qisun/mydata:/data  
-w /data myimage samtools"
```

```
$SAMTOOLS view -b -o sample2.bam sample2.sam
```





# Use GPU

## Base images (including compatible cuda)

```
nvidia/cuda  
tensorflow/tensorflow  
pytorch/pytorch
```

Test whether GPUs are accessible by container

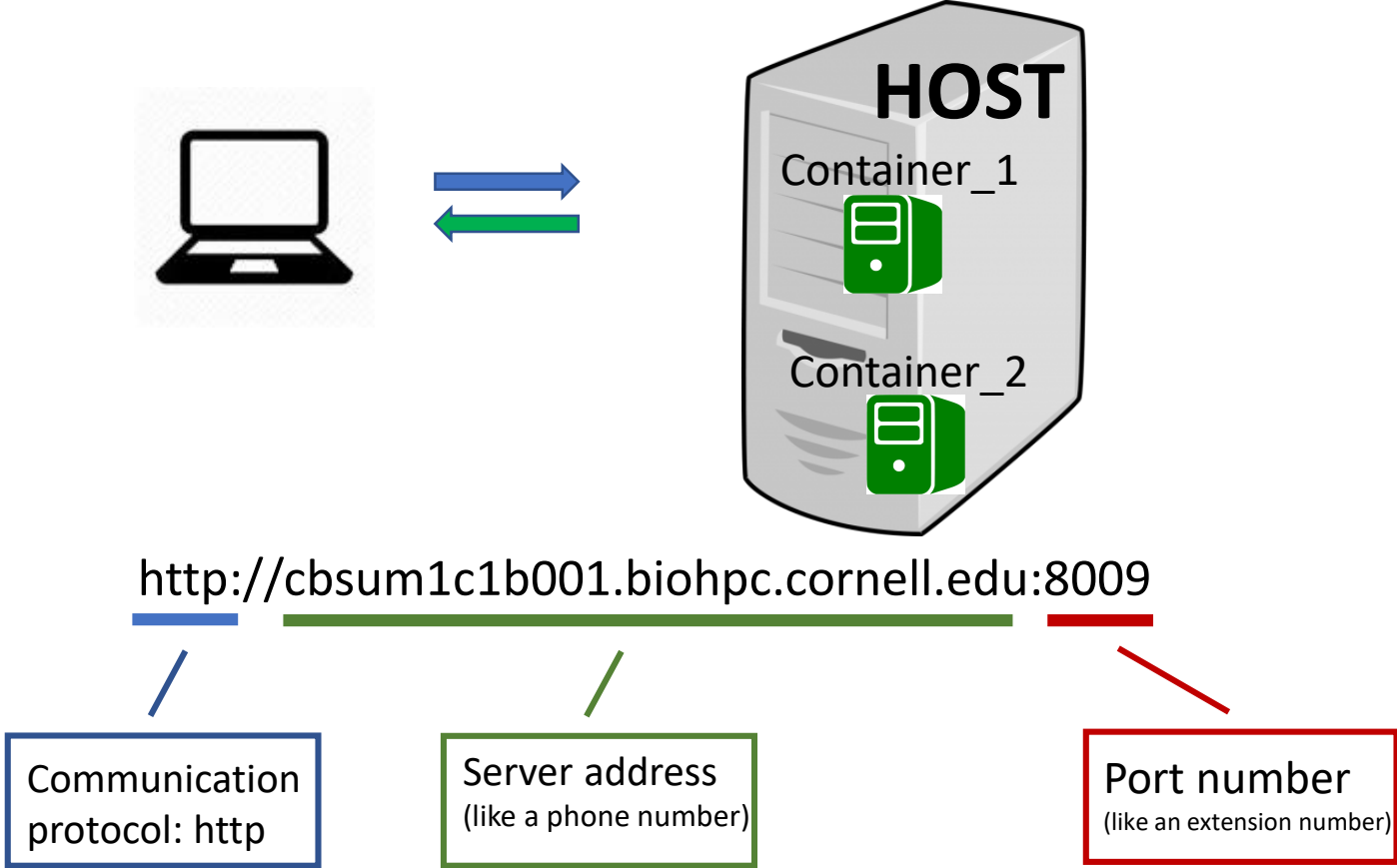
```
docker run --rm --gpus all my_gpu_image nvidia-smi
```

```
docker run --rm --gpus all my_gpu_image /myscript.py
```

--gpus all: all GPU devices

---gpus '"device=1,2"' use GPU device 1 and 2

# Serving a web site with Docker



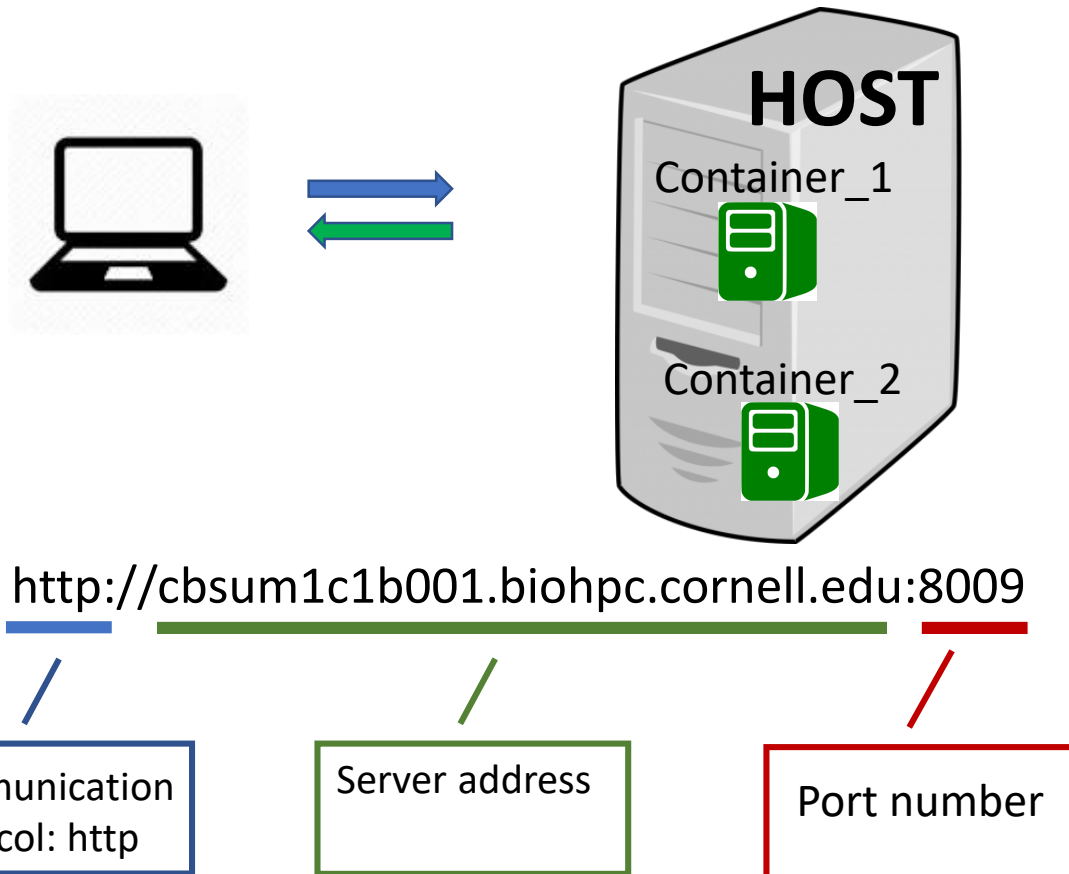
(regular URL: <https://biohpc.cornell.edu>, port number is skipped, because by default **https** protocol uses port **443**)

# Serving a web site with Docker

```
docker1 run -d -p 8009:8787 rocker/rstudio
```

Web browser on the laptop  
communicate with server on port 8009

The Rstudio container is  
configured to listen on port 8787



# Clean up the images and containers

## Remove a container

```
docker ps -a
```

```
docker stop c4dca29b2393
```

```
docker rm c4dca29b2393
```

## Remove an image

```
docker images
```

```
docker rmi ubuntu:18.04
```

# Docker on BioHPC

- Use the command “**docker1**” instead of “docker”;
- Only directories under /workdir/\$USER and /local/storage can be mounted;
- By default, “/workdir/\$USER” is mounted as “/workdir/” in container;
- When committing a new image, the image name is prefixed with your user ID

# Docker on BioHPC

Useful commands:

```
docker1 clean all
```

# remove all containers owned by you

```
docker1 claim
```

# claim ownership of all files under  
/workdir/\$USER \*

```
docker1 claim /workdir/qisun/mydata
```

# claim ownership of files in specific  
directory \*

\* These two commands are useful as all result files created in containers owned by “root”

# You can build docker images on your Windows/Mac laptop

<https://docs.docker.com/desktop/install/windows-install/>

<https://docs.docker.com/desktop/install/mac-install/>

A snapshot of using Docker on my Windows Command Prompt

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\qisun>docker run -dit ubuntu:18.04
Unable to find image 'ubuntu:18.04' locally
18.04: Pulling from library/ubuntu
e706e0a9f423: Pull complete
Digest: sha256:40b84b75884ff39e4cac4bf62cb9678227b1fbf9dbe3f67ef2a6b073aa4bb529
Status: Downloaded newer image for ubuntu:18.04
a78cbc7ea05d90c858fc3e4d3fa8b4ff86127191d0b0492c72b241574df9c01c

C:\Users\qisun>docker ps
CONTAINER ID   IMAGE          COMMAND      CREATED        STATUS        PORTS        NAMES
a78cbc7ea05d   ubuntu:18.04   "bash"      10 seconds ago Up 9 seconds   80/tcp       gracious_payne

C:\Users\qisun>docker images
REPOSITORY    TAG          IMAGE ID      CREATED        SIZE
ubuntu        18.04       71cb16d32be4 2 weeks ago   63.1MB
bonito        latest      17314ff14952 4 weeks ago   10.8GB
test1         latest      18b53b396026 4 weeks ago   77.8MB
biohpc/sindocker latest      713270bf56dc 5 weeks ago   325MB
ubuntu        latest      2dc39ba059dc 7 weeks ago   77.8MB
nvidia/cuda   11.3.1-devel-ubuntu20.04 fee592f3dd3f 5 months ago  4.27GB

C:\Users\qisun>
```

Build/Test Docker on laptop



Save as a tar file or put to Dockerhub



Load/pull image in cloud and run the pipeline

(Both Windows and Mac run Docker through Linux VM)

# Summary

**Build Docker images:** either interactively or using Dockerfile

**Run software in Docker:**

- Mount a directory: `-v /workdir/qisun/mydata:/data`
- Specify working directory: `-w /data`
- Port forwarding: `-p 8015:80`

**Docker on BioHPC**

- Command: `docker1`
- Only directories under `/workdir/$USER` or `/local/storage` can be mounted;
- Claim file ownership: `docker1 claim`