

Containerised Applications in HPC Environment



CSC – Suomalainen tutkimuksen, koulutuksen, kulttuurin ja julkishallinnon ICT-osaamiskeskus

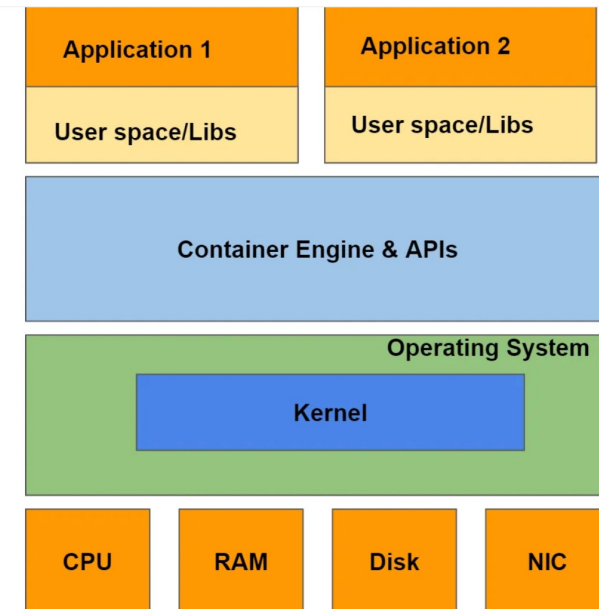
Outline

- Basic introduction to containers
- Biocontainers and related registries
- Deploying (running) biocontainers in HPC environment
- Mounting/binding volumes

Basic Introduction to Container

What Are Containers ?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.



Container Architecture

- ✓ A container consists of an entire runtime environment

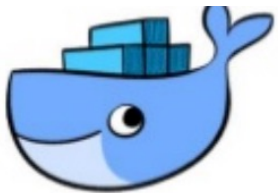
Some Basic Terminology



Image : self-contained snapshot of your full application, packages and dependencies



Container: A running instance of an image



Container Engine: Creates, ships and runs containers deployable on a physical or virtual, host locally, in a datacenter or cloud service provider



Registry Service:
Cloud or server based storage and distribution service for your images

Benefits of running a software as Container



Light-weight
(No OS to boot)



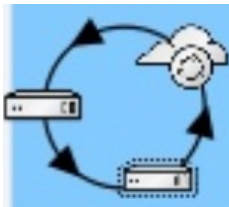
Efficient
(Less OS overhead)



Portable



(Faster scaling)



Deployment flexibility and reproducibility

(Less dependencies
between process layers)

Biocontainers and Related Registries

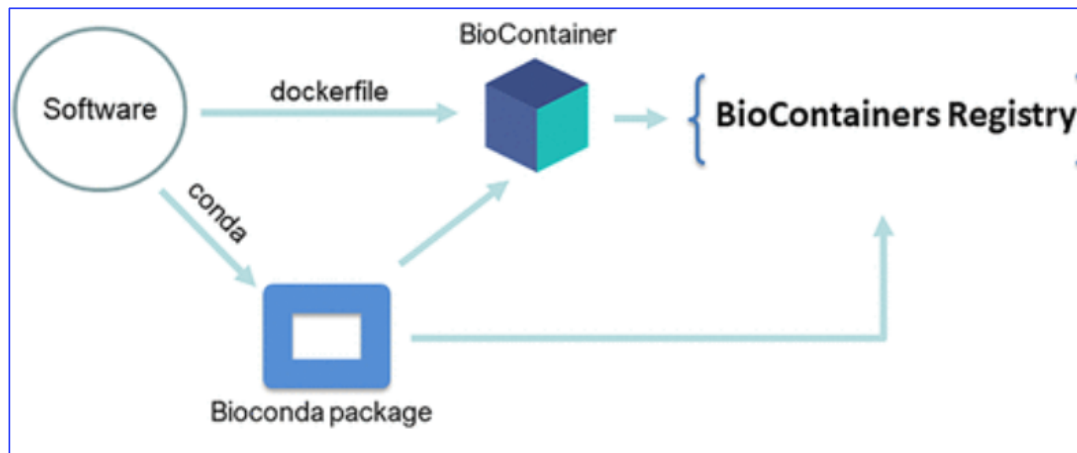
Biocontainers: Bioinformatics Containers

- A community-driven effort
- Focus is to create and manage bioinformatics software containers
- Focus on popular Omics' methods (Genomics, proteomics, metagenomics, metabolomics)
- Can be integrated into bioinformatics pipelines and different architectures
- Provides ready-made containers for bioinformatics community
 - QUAY registry (<https://quay.io>)
 - Dockerhub (<https://hub.docker.com/>)

Biocontainers Registry (1/2)



- A hosted registry of all BioContainers images that are ready to be used
- The interface to search BioContainers across all the registries.
- Source of Biocontainer image can be conda recipe or dockerfile

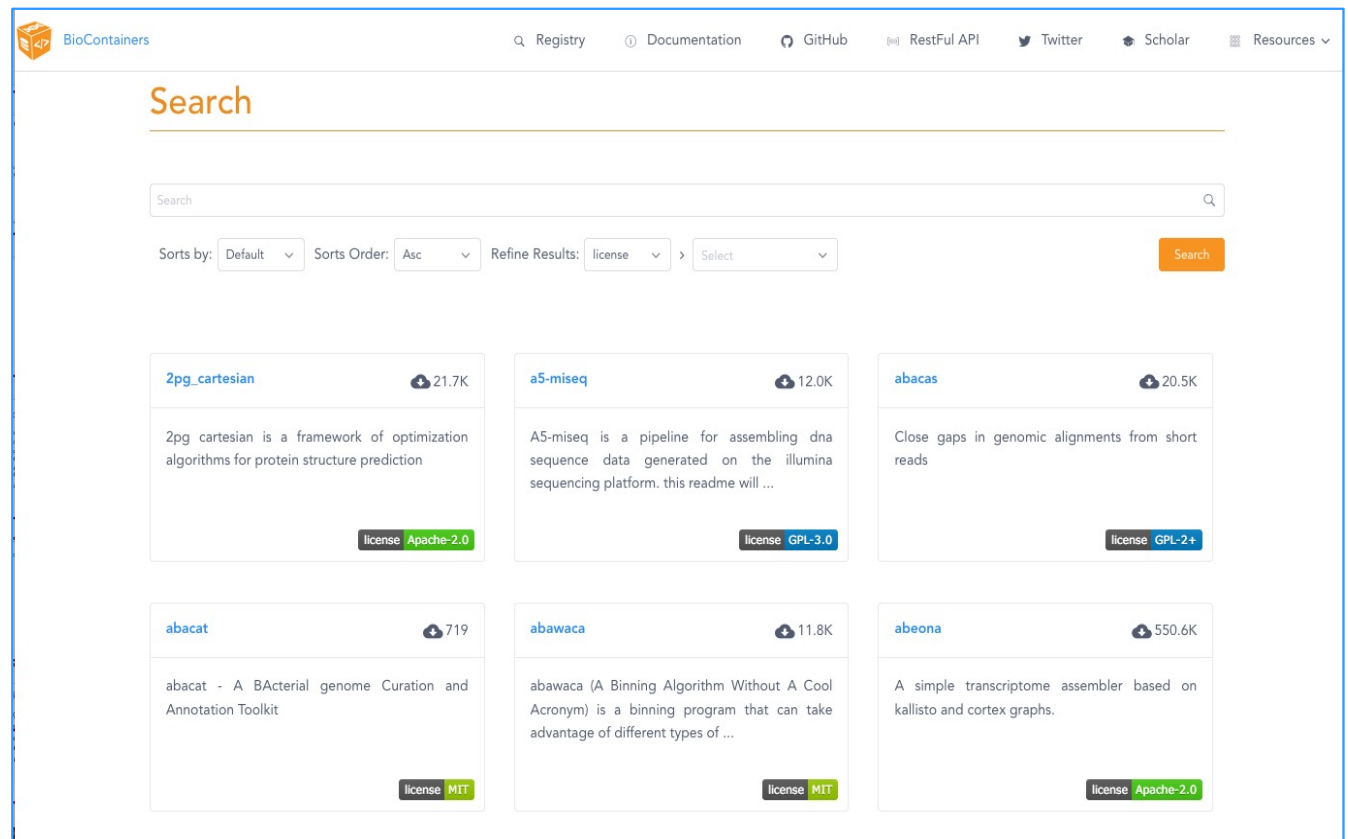


Biocontainers Registry (2/2)



Webpage: <https://biocontainers.pro/>

- 10.4K tools
- 44.2K versions
- 213.2K containers and packages



Docker Hub

- A registry from Docker
- <https://hub.docker.com/bio>
- A centralized management of user accounts, image checksums and public/private repositories
- Not all images can work with Singularity
 - Applications with root access
 - Applications with entrypoints

Dockerhub screen shot

The screenshot shows the Docker Hub search results for 'Fastqc'. The top navigation bar is blue with the Docker Hub logo, a search bar containing 'Fastqc', and links for 'Explore', 'Pricing', 'Sign In', and 'Sign Up'. Below the navigation bar, there are tabs for 'Docker', 'Containers' (which is selected), and 'Plugins'. The main content area is divided into a left sidebar with filters and a right section with search results.

Filters

Images

- ☐ Verified Publisher ⓘ
- ☐ Official Images ⓘ
Official Images Published By Docker

Categories ⓘ

- ☐ Analytics
- ☐ Application Frameworks
- ☐ Application Infrastructure
- ☐ Application Services
- ☐ Base Images
- ☐ Databases
- ☐ DevOps Tools
- ☐ Featured Images
- ☐ Messaging Services
- ☐ Monitoring
- ☐ Operating Systems
- ☐ Programming Languages
- ☐ Security
- ☐ Storage

Operating Systems

- ☐ Linux
- ☐ Windows

Search Results

1 - 25 of 479 results for **Fastqc**. [Clear search](#)

Most Popular ▼

- staphb/fastqc**
By [staphb](#) • Updated 9 months ago
FastQC: A quality control analysis tool for high throughput sequencing data
500K+ Downloads 0 Stars
Container Linux x86-64
- fastqcdocker/fastqcdocker**
By [fastqcdocker](#) • Updated 5 years ago
Testing fastqc docker
306 Downloads 0 Stars
Container Linux x86-64
- biocontainers/fastqc**
By [biocontainers](#) • Updated 7 months ago
fastqc
100K+ Downloads 3 Stars
Container Linux x86-64
- singlecellpipeline/fastqc**
By [singlecellpipeline](#) • Updated 2 years ago
500K+ Downloads 0 Stars
Container Linux x86-64

QUAY Registry

- Quay (**Red Hat**) is a container image registry
- a scalable open source platform to host container images across any size organization
- Create your own public repositories
- Provides CI support for automated builds for BioConda
GitHub
- All Biocontainers are docker-based and are publicly available for free

Cloud Library from Sylabs



- Cloud Library is the official image registry provided by Sylabs.io
- A singularity hub
- <https://cloud.sylabs.io/library>
- Images should work normally

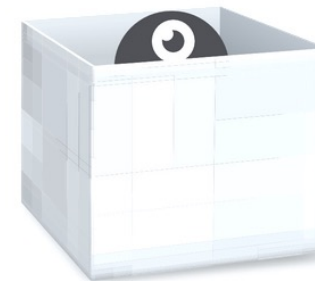
The screenshot shows the Sylabs.io Cloud Library interface. The header includes the Sylabs.io logo, a search bar, and navigation links for Home, Cloud Library, Remote Builder, and Keystore. The main section features the 'Cloud Library' title and a description: 'Cloud Library is the official image registry provided by Sylabs.io. Users can share Singularity images through the Cloud Library, as well as pull/push SIF™ images through Singularity CLI. Email any feature requests or feedback to support@sylabs.io.' Below this, there's a 'Share your awesome containers!' section with a 'Sign In' button. The main content area displays two featured container images: 'HMGU-ITG/DEFAULT/BURDEN_TESTING' (1.5.4 latest, 569.73 MB) and 'ALIRKAY/DEFAULT/DOPAMINE' (latest, 3.19 GB). Both are AMD64 architecture. To the right, a 'MOST DOWNLOADED' section lists the top 10 most downloaded images with their respective download counts.

Image Name	Downloads
library/default/alpine	527913
sylabs-bot/smoke-test-collection/testimage0	255826
sylabs-bot/smoke-test-collection/testimage1	255800
library/default/busybox	240750
sylabs/tests/not-default	106898
sylabs/tests/unsigned	72149
library/default/ubuntu	30299
sylabs/tests/signed	24164
sylabsed/examples/olcow	14221
library/default/centos	11647

Singularity Hub



- Singularity Hub is no longer online as a builder service, exists as a read only archive
- More information:
<https://singularityhub.github.io/singularityhub-docs/>
- Pulling an image from: Singularity pull shub://



Singularity Hub
Documentation

Hello there! It's so great that you are interested in Singularity Hub.

[chat](#) [on gitter](#) [paper](#) [plos-one](#)

Deploying (running) biocontainers in HPC environment

Qualified reference format for Image

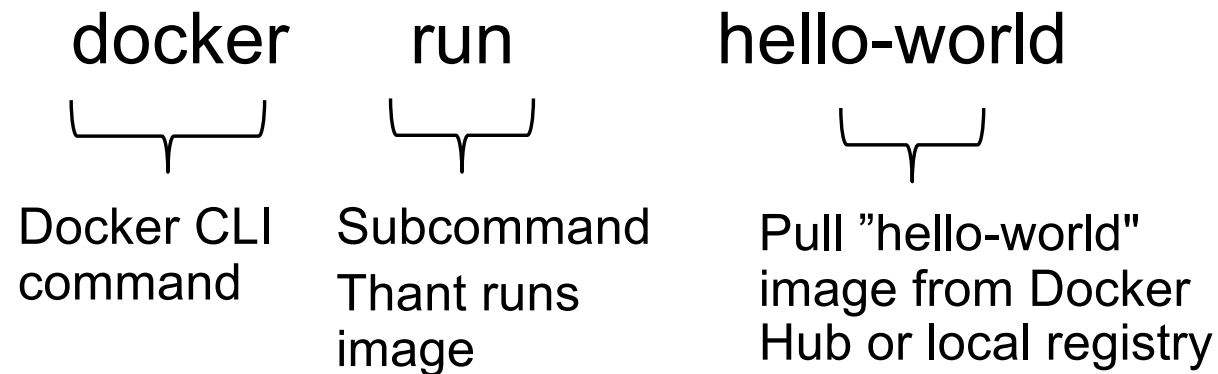


- A qualified image name consists of three main components:
 - Image prefix: library/shub/library
 - a registry location (hostname)
 - a username (namespace)
 - a image name (reponame)

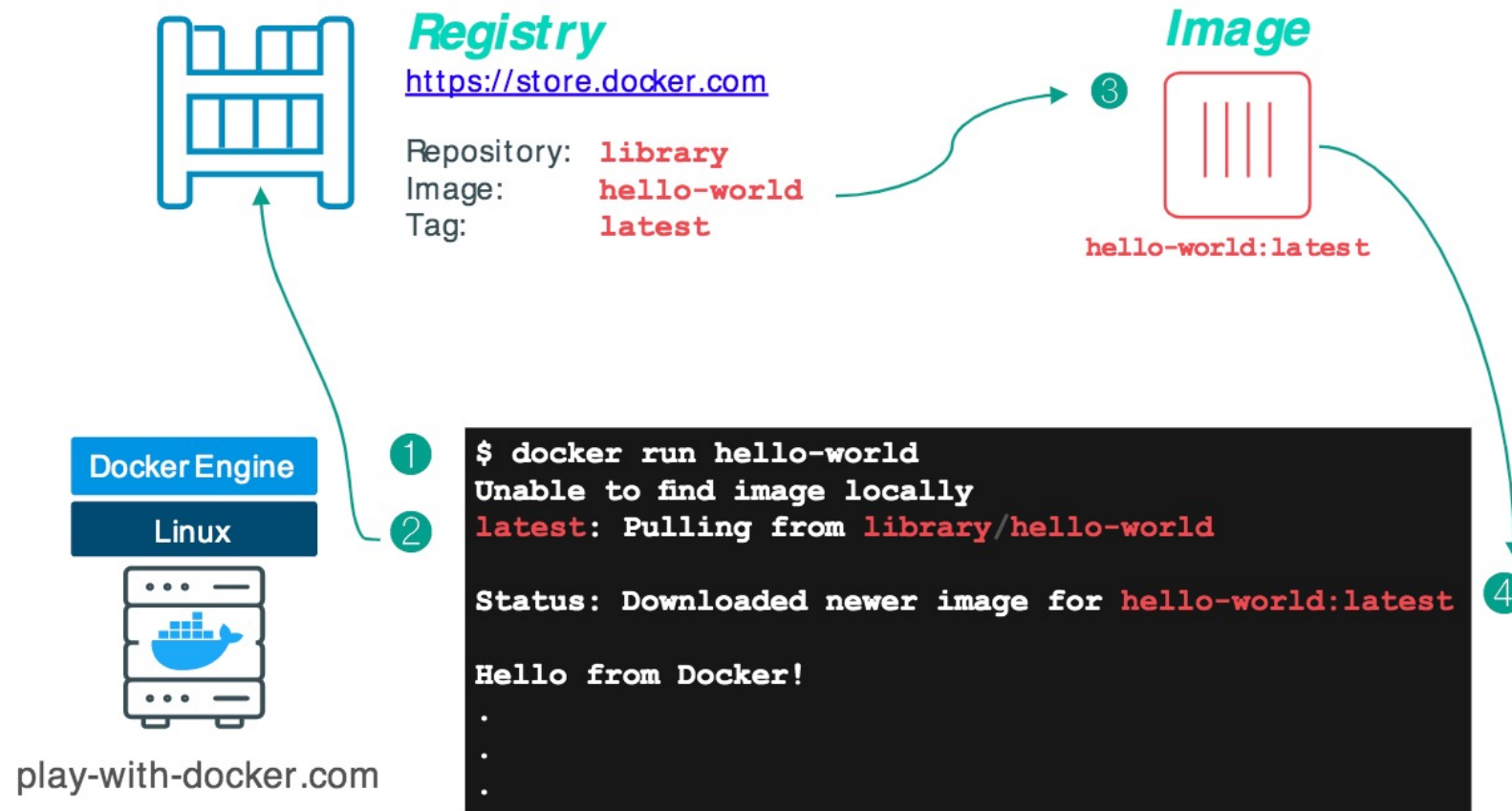
Prefix://hostname[:port]/username/imagename[:tag]

- DockerHub registry: docker://username /image[:tag]
- Red Hat registry: quay.io/ username/image[:tag]

Docker run basics: hello-world



Docker run basics: Internals of Hello-world



Working with Containers in CSC HPC Environment

- Singularity is installed on Puhti (no need to load any modules)
- Available options
 - Using modularised container (pre-installed for you in Puhti)
 - Examples:

Rstudio	Chip-Seq-Pipeline
CrossMap	Cutadapt
EAGER	QIIME 1
Jupyter	BRAKER
BUSCO	aTRAM
METABOLIC	
 - Using custom-made container (your own image or downloaded from container registry)
 - Any biocontainer, Deepvariant, GATK ..etc

Getting Started with a Modularised Container



- Load module
 - e.g., module load Cutadapt
- Module command sets some environment variables on host
 - e.g., SING_IMAGE and SING_FLAGS
- Use singularity_wrapper which has advantages than plain singularity command
 - singularity_wrapper exec command_to_run
- Mounting datasets with SquashFS
 - when input files are too big in numbers

Getting Started with a Custom Container

- Either you pull an image from registry or prepare one by yourself
- Pull/Build an image from registry repositories using singularity command
 - `singularity pull hello-world.sif shub://vsoch/hello-world`
 - `singularity build r-base-latest.sif docker://r-base`
- Note:
 - URI beginning with `library://` to build from the Container Library.
 - URI beginning with `docker://` to build from Docker `hub/Quay.io`.
 - URI beginning with `shub://` to build from Singularity Hub.
- Executing a command
 - `singularity exec -B /path/to/module:/opt/module singularity_image actual_command`

Mounting/binding volumes

Binding external directories

Why Mounting/binding volumes

- No data persistence in container file systems
- Can't share any data with other containers/volumes
- Containers are stateless
- Decoupling container from storage

Mounting/binding volumes

- Volume
 - A Volume is a storage. This name comes from the Enterprise use-case. Volumes = Directories.
- Note that you can't mount any directory you like on HPC systems
 - HOME, PROJAPPL, SCRATCH
- Binding/Mapping
 - You can bind/map directories from the Host machine into a Guest container
 - `singularity exec -B /host/path:/guest/path singularity_image.simg`