CSC

Containerised Bio Applications in HPC Environment

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

Outline

- Basic introduction to (bio)containers
- Searching pre-existing biocontainers
- Deploying (running) biocontainers in HPC environment
    - Containers availbale as modules
    - Custom-made containers
- Mounting/binding host volumes

Basic Introduction to (Bio)container

Some Basic Terminology

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

Container: A running instance of an image

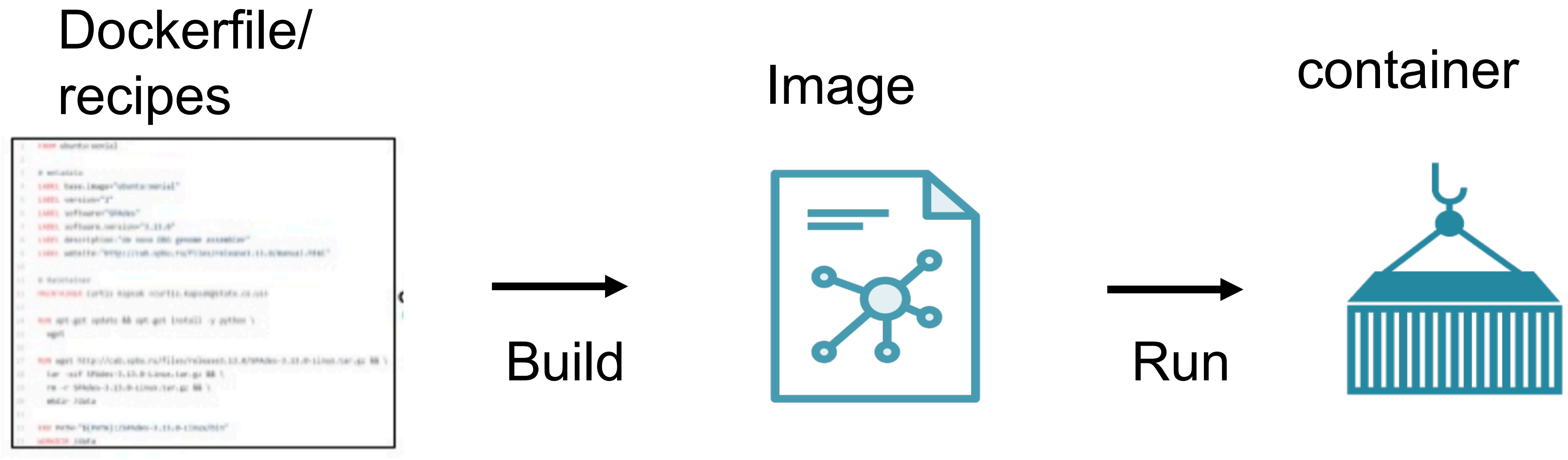
Container Engine

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:

Central Dogma of Containerisation



Dockerfile/ recipes

Build

Image

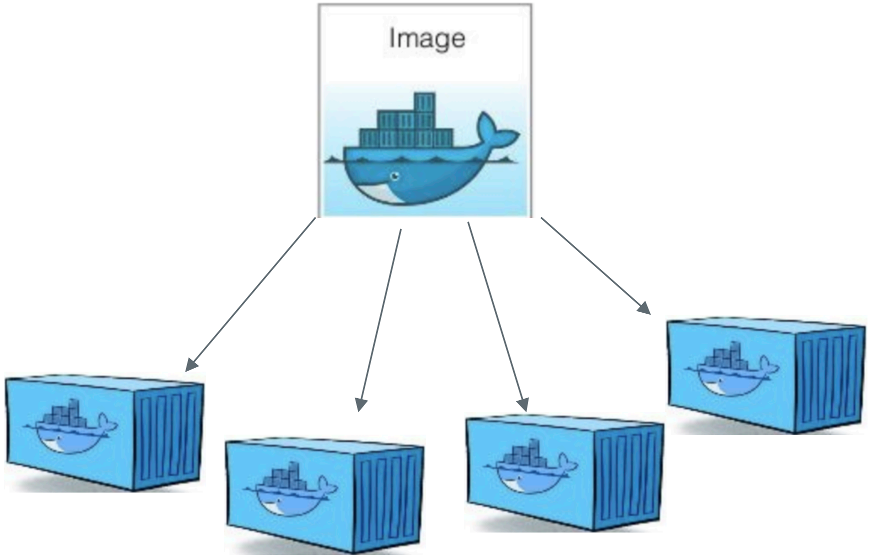
Run

container

Two containerisation platforms

- Docker image built from dockerfiles
- Singularity image built from singularity recipes (deffiles)

(docker) Image *vs.* Container



- A n image is an immutable (unchangeable) file that contains the source code, libraries, dependencies and any tools needed for an application to run.
- Docker container is a Docker image brought to life; A runnable instance of image
- You can create as many number of containers as necessary from a image

Benefits of Running a Software as Container

Light-weight

(No OS to boot)

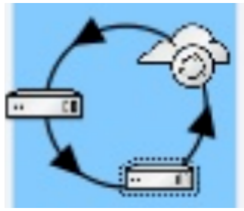
Effiecient

(Less OS overhead)

Portable

(Less dependencies
between process layers)

(Faster scaling)



Deployment flexibility

Searching Pre-existing Biocontainers

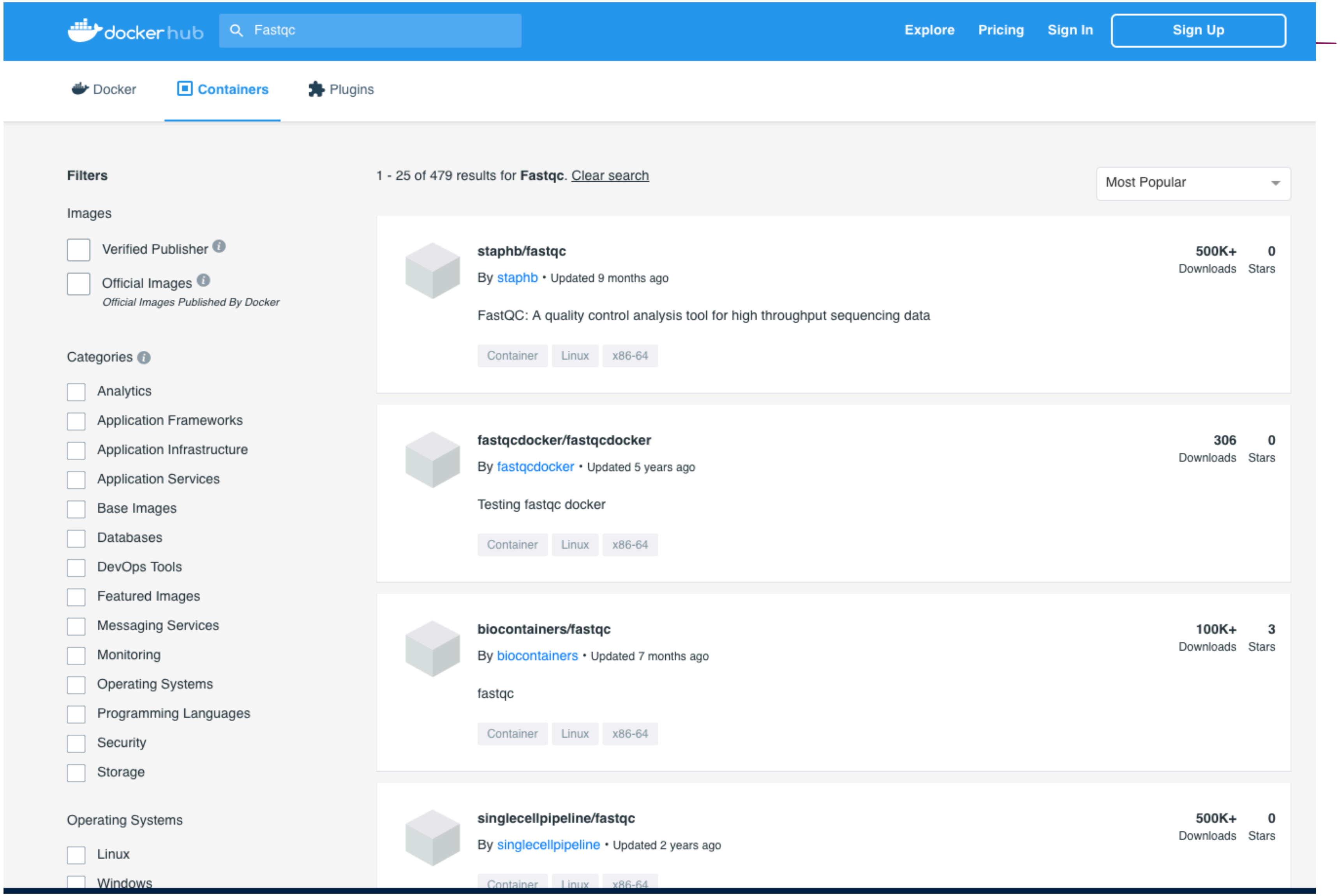
Biocontainers: Bioinformatics Containers

- 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/)

Docker Hub

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

Dockerhub screen shot



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

1 - 25 of 479 results for **Fastqc**. Clear search

staphb/fastqc
By staphb • Updated 9 months ago
FastQC: A quality control analysis tool for high throughput sequencing data
Container | Linux | x86-64
500K+ Downloads **0** Stars

fastqcdocker/fastqcdocker
By fastqcdocker • Updated 5 years ago
Testing fastqc docker
Container | Linux | x86-64
306 Downloads **0** Stars

biocontainers/fastqc
By biocontainers • Updated 7 months ago
fastqc
Container | Linux | x86-64
100K+ Downloads **3** Stars

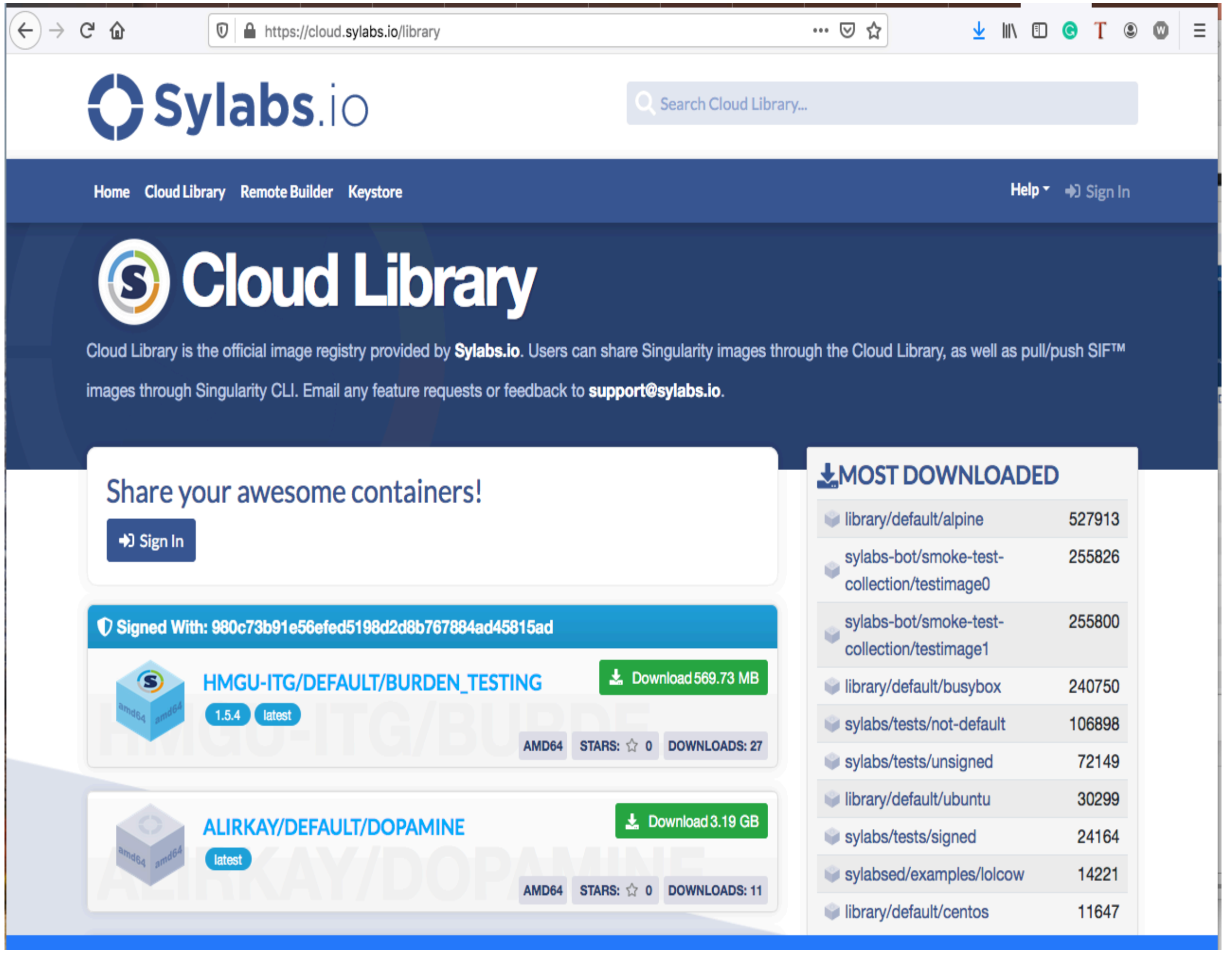
singlecellpipeline/fastqc
By singlecellpipeline • Updated 2 years ago
Container | Linux | x86-64
500K+ Downloads **0** Stars

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



Cloud Library

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**.

Share your awesome containers!

Signed With: 980c73b91e56efed5198d2d8b767884ad45815ad

HMGU-ITG/DEFAULT/BURDEN_TESTING

AMD64 STARS: ☆ 0 DOWNLOADS: 27

ALIRKAY/DEFAULT/DOPAMINE

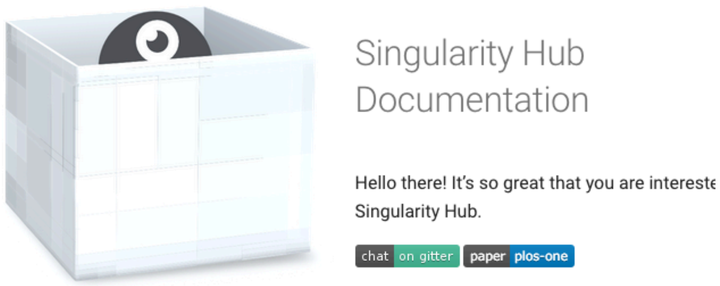
AMD64 STARS: ☆ 0 DOWNLOADS: 11

MOST DOWNLOADED

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/lolcow	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/
- Pullinng an image from: Singularity pull shub://



Singularity Hub Documentation

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

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 hello-world

docker

Docker CLI command

run

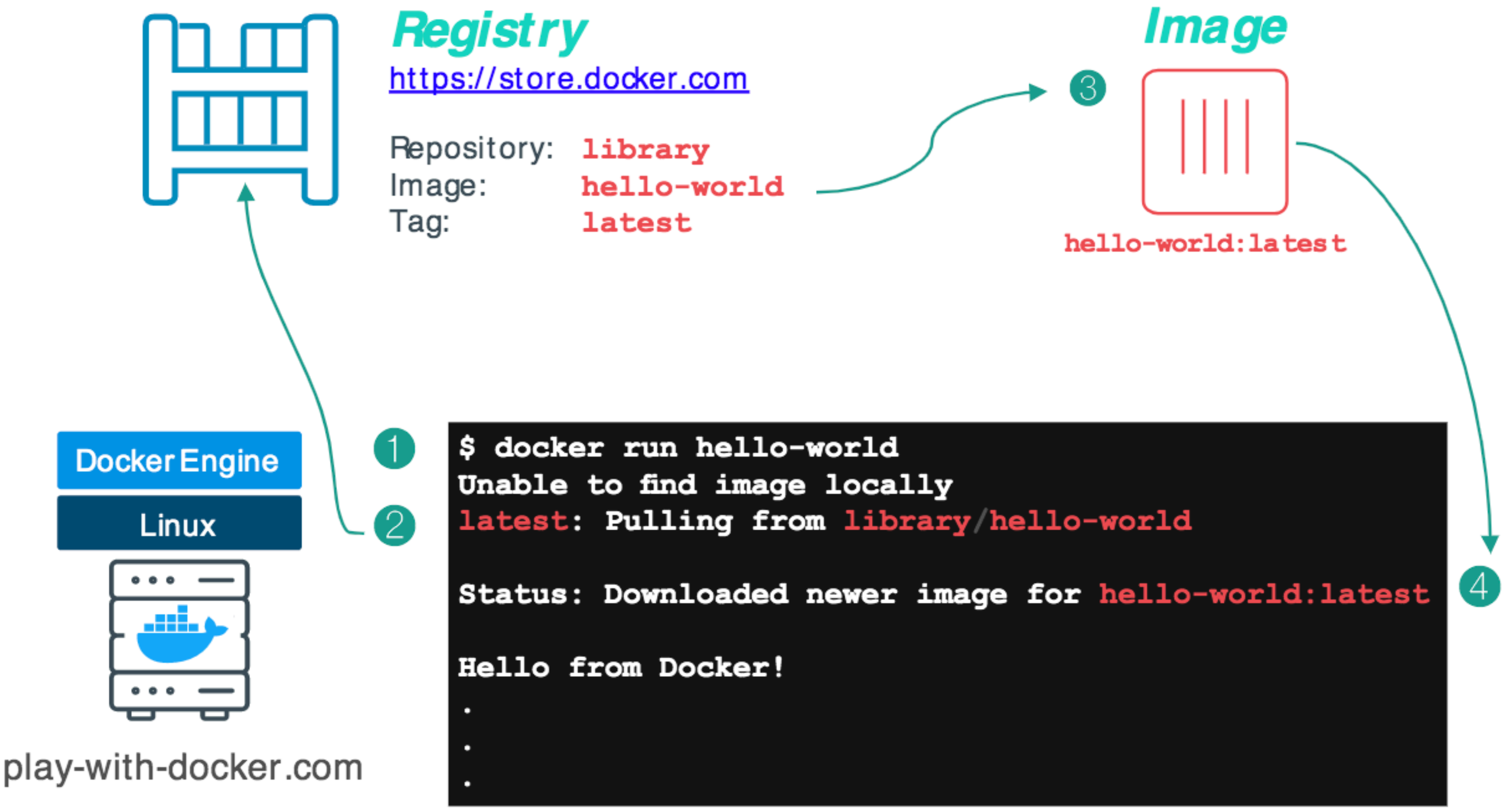
Subcommand

Thant runs image

hello-world

Pull ”hello-world" image from Docker Hub or local registry

Docker Run Basics: Internals of Hello-world



Registry

https://store.docker.com

Repository: **library**
Image: **hello-world**
Tag: **latest**

Image

hello-world:latest

play-with-docker.com

```
$ docker run hello-world
Unable to find image locally
latest: Pulling from library/hello-world

Status: Downloaded newer image for hello-world:latest

Hello from Docker!
.
.
.
```

Working with Containers in CSC HPC Environment

- Singualrity is installed on Puhti (no need to load any modules)
- Available options
  - Using modularised container (pre-installed for you in Puhti)
  - Examples:
    - Rstudio
    - CrossMap
    - EAGER
    - Jupyter
    - BUSCO
    - METABOLIC
    - Chip-Seq-Pipeline
    - Cutadapt
    - QIIME 1
    - BRAKER
    - aTRAM
  - Using custom-made container (your own image or dowloaded 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-made 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 Host 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 Host 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 run -B /host/path:/guest/path singularity_image.simg

Time for Tutorials on Some Bioapplication examples: R, Fastqc and deepvariant

Expected Learning

- Ability to use a containerised modular application
- Able to use a third-party container for your needs