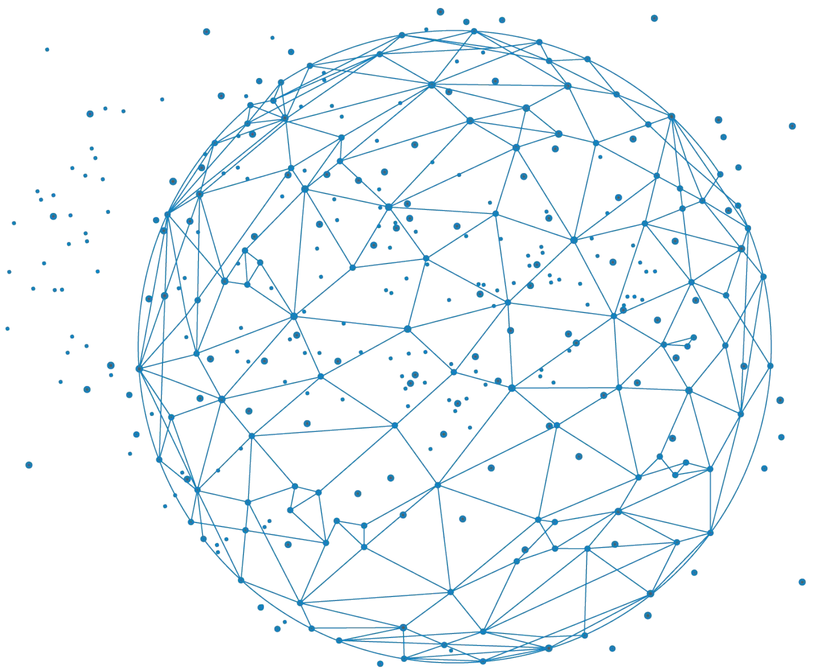
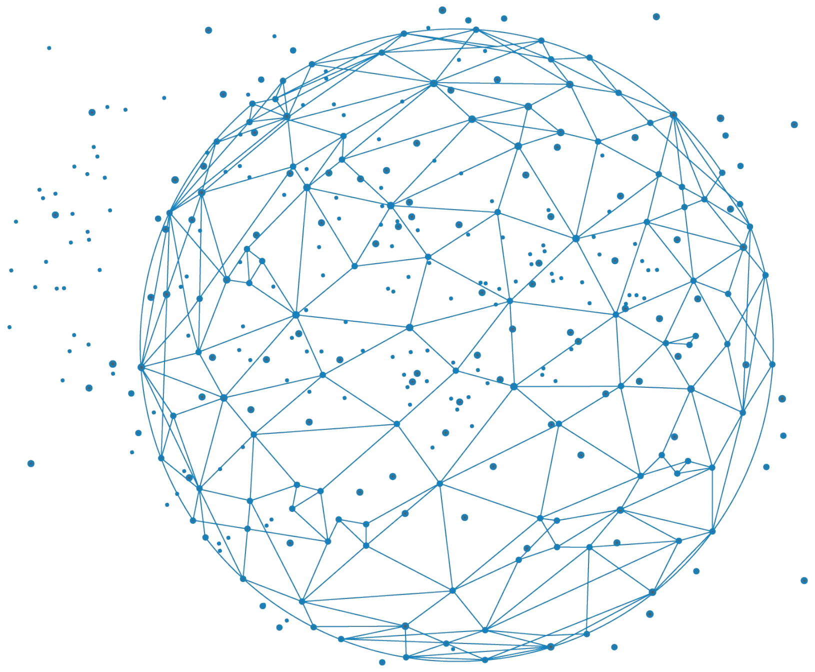


Using Docker and Apptainer

J. Seiler

Based on Céline Hernandez training for FAIR Bioinfo 2023

Let’s do an exercise first



Exercise : putting our pipeline project on Github

- Create a Git repository for your pipeline folder
- Index and commit the following files
  - All Jupyter Notebooks
  - All R files
  - environment.yml
  - environment-linux-64.lock
- Push on Github in a new public repository

Exercise : putting our pipeline project on Github

- Create a Git repository for your pipeline folder

```
$ cd pipeline
$ git init
```

Exercise : putting our pipeline project on Github

- Index and commit the following files
  - All Jupyter Notebooks
  - All R files
  - environment.yml
  - environment-linux-64.lock

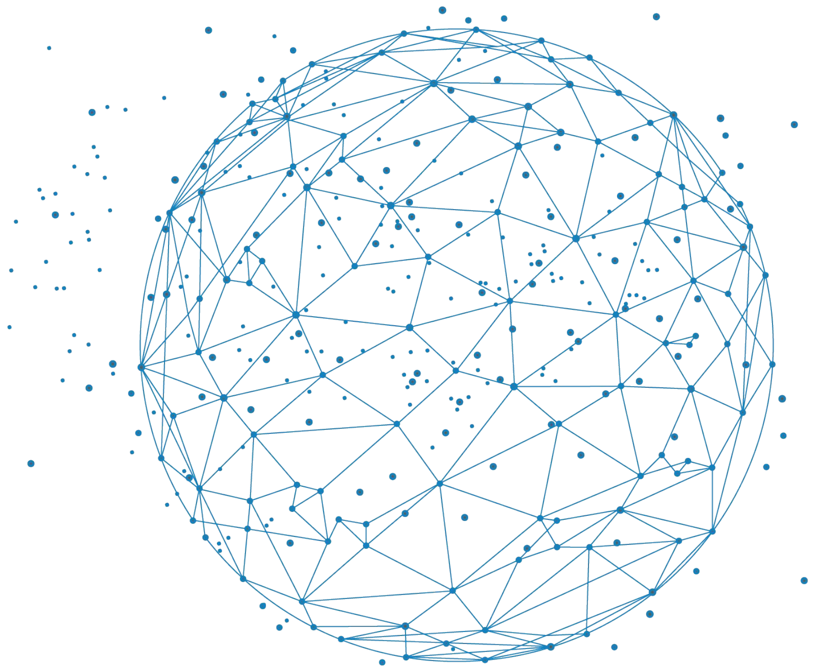
```
$ git add *.ipynb *.R environment.yml environment-linux-64.lock
$ git commit -m "Initial commit"
```

Exercise : putting our pipeline project on Github

- Push on Github in a new public repository

```
$ git remote add origin <repo ssh url>
$ git push origin main
```

About encapsulation



Different levels of encapsulation

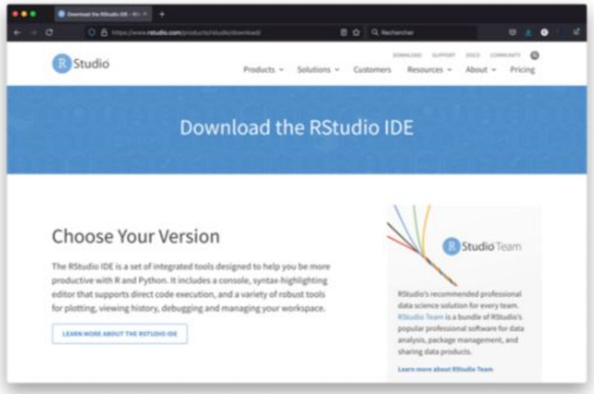
Goal : capture the system environment of applications (OS, packages, libraries,. . .) to control their execution.

- Hardware virtualisation (virtual machines)
- OS virtualisation (images and containers)
- Environment management CONDA

Encapsulation

Let's say we want to install RStudio…

Install Rstudio ?



Download the RStudio IDE

Choose Your Version

The RStudio IDE is a set of integrated tools designed to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.

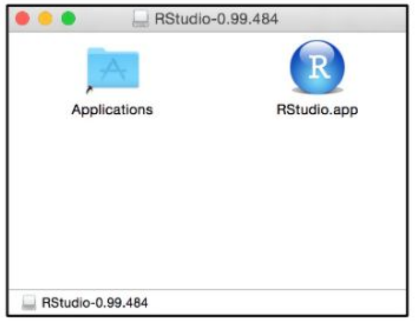
LEARN MORE ABOUT THE RSTUDIO IDE

RStudio Team

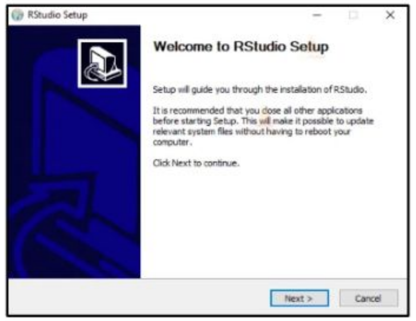
RStudio's recommended professional data science solution for every team. RStudio Team is a bundle of RStudio's popular professional software for data analysis, package management, and sharing data products.

Learn more about RStudio Team

MacOS



Windows



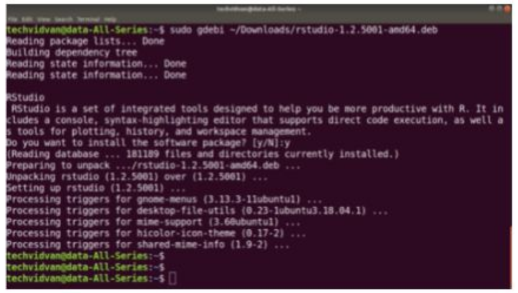
Welcome to RStudio Setup

Setup will guide you through the installation of RStudio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

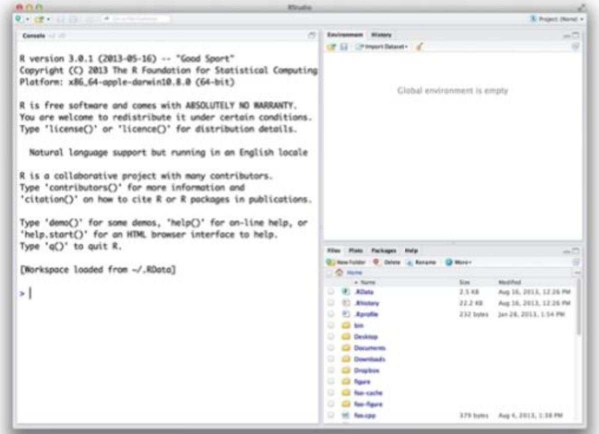
Click Next to continue.

Unix-based



```
techvidvan@data-All-Series:~$ sudo gdebi ~/Downloads/rstudio-1.2.5001-amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading state information... Done

RStudio
 RStudio is a set of integrated tools designed to help you be more productive with R. It in
cludes a console, syntax-highlighting editor that supports direct code execution, as well a
s tools for plotting, history, and workspace management.
Do you want to install the software package? [y/N]:y
(Reading database ... 181189 files and directories currently installed.)
Preparing to unpack .../rstudio-1.2.5001-amd64.deb ...
Unpacking rstudio (1.2.5001) over (1.2.5001) ...
Setting up rstudio (1.2.5001) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1) ...
Processing triggers for desktop-file-utils (0.23-1ubuntu3.18.04.1) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for shared-mime-info (1.9-2) ...
techvidvan@data-All-Series:~$
techvidvan@data-All-Series:~$
techvidvan@data-All-Series:~$
```

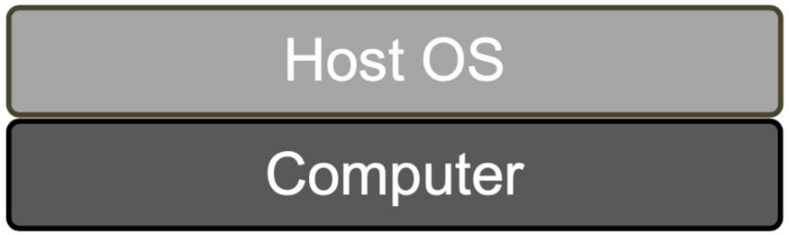
Use Rstudio



```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> |
```

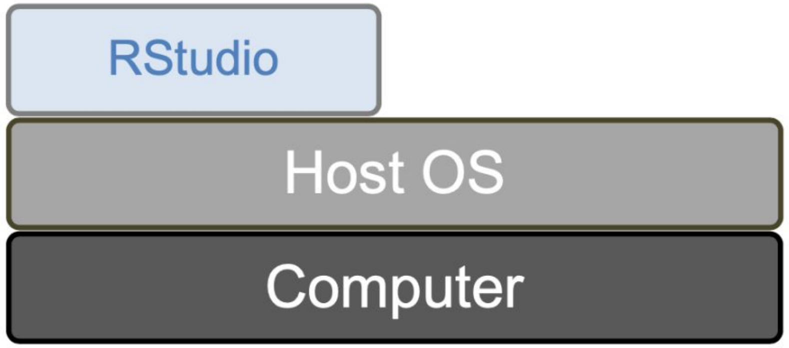
Name	Size	Modified
.RData	2.5 KB	Aug 16, 2013, 12:26 PM
.Rhistory	22.2 KB	Aug 16, 2013, 12:26 PM
.Rprofile	232 bytes	Jan 26, 2013, 1:54 PM
bin		
Desktop		
Documents		
Downloads		
Dropbox		
figure		
foo-cache		
foo-figure		
foo.cpp	379 bytes	Aug 4, 2013, 1:38 PM

Encapsulation



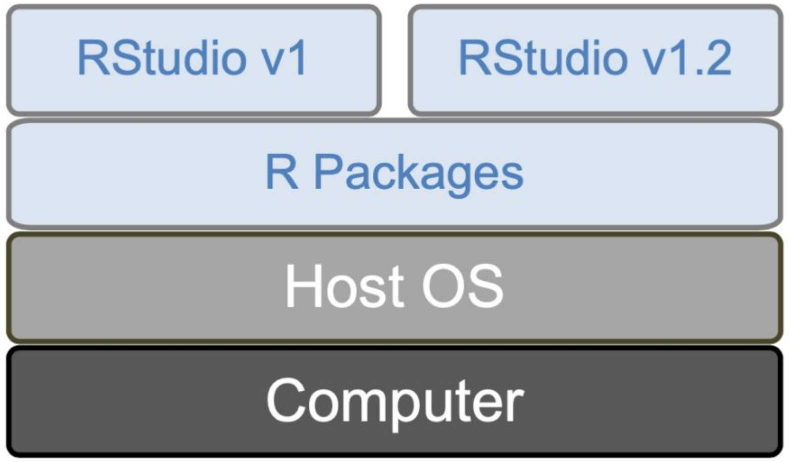
We started with a computer using a specific OS...

Encapsulation



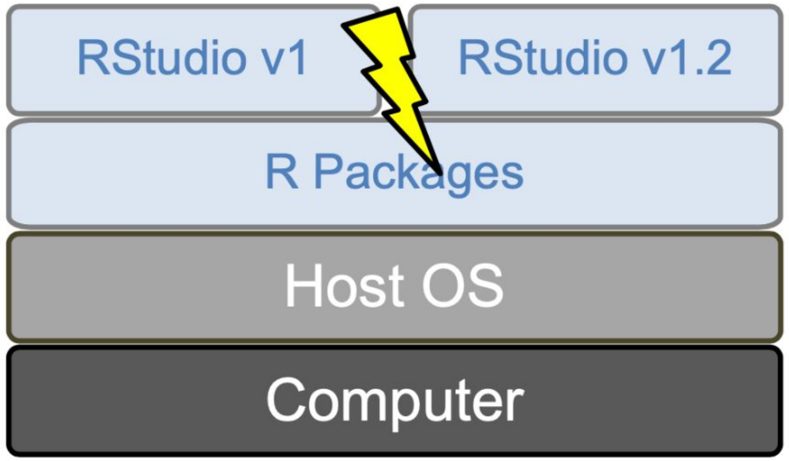
We started with a computer using a specific OS...
And inside this environment, we installed a new application.

Encapsulation



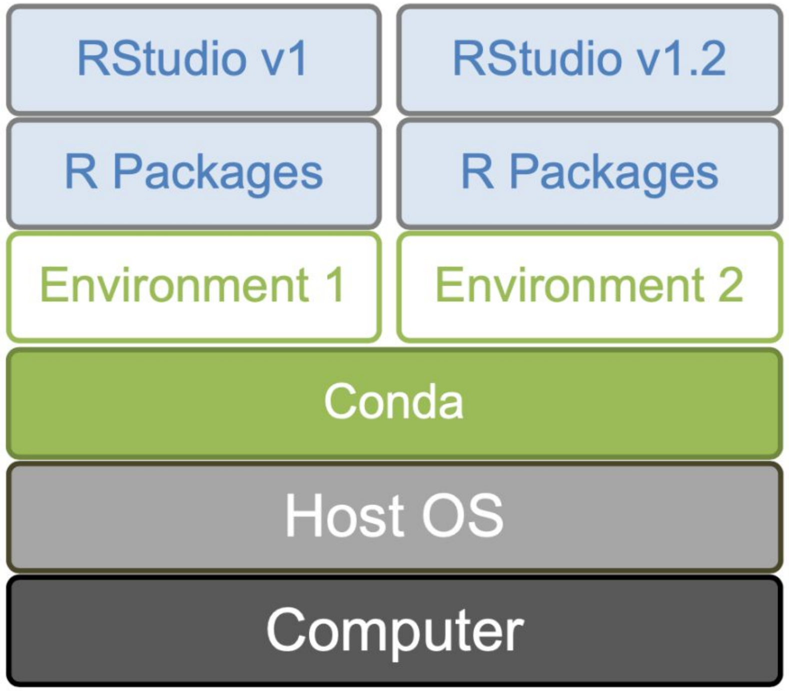
Usually dependencies of different applications don't interfere.
But what if we want to test the latest version of our favourite tool?
There might be conflicts. . .

Encapsulation



Usually dependencies of different applications don't interfere.
But what if we want to test the latest version of our favourite tool?
There might be conflicts. . .

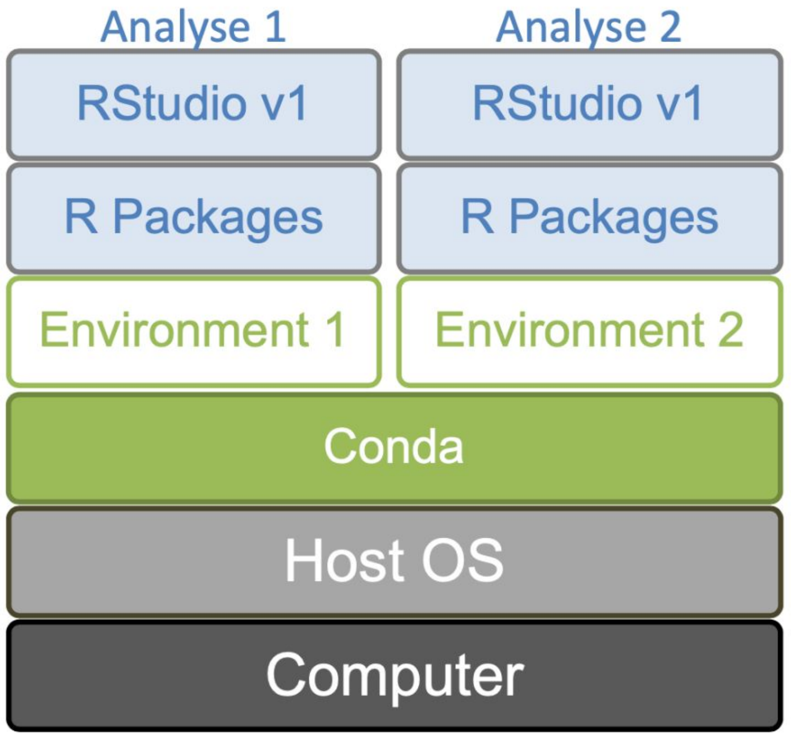
Encapsulation : managing environments



Idea : create separated environments for each application.

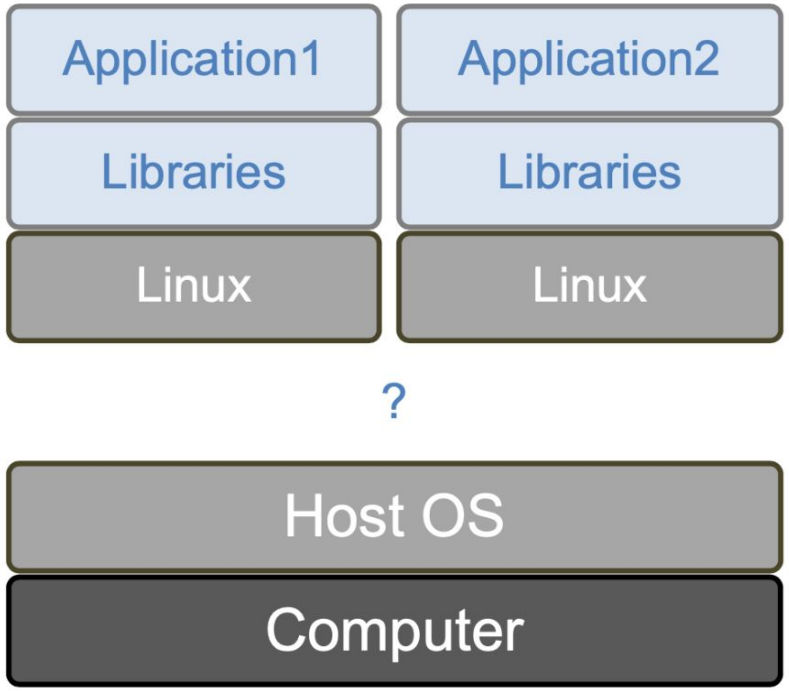
CONDA

Encapsulation : managing environments



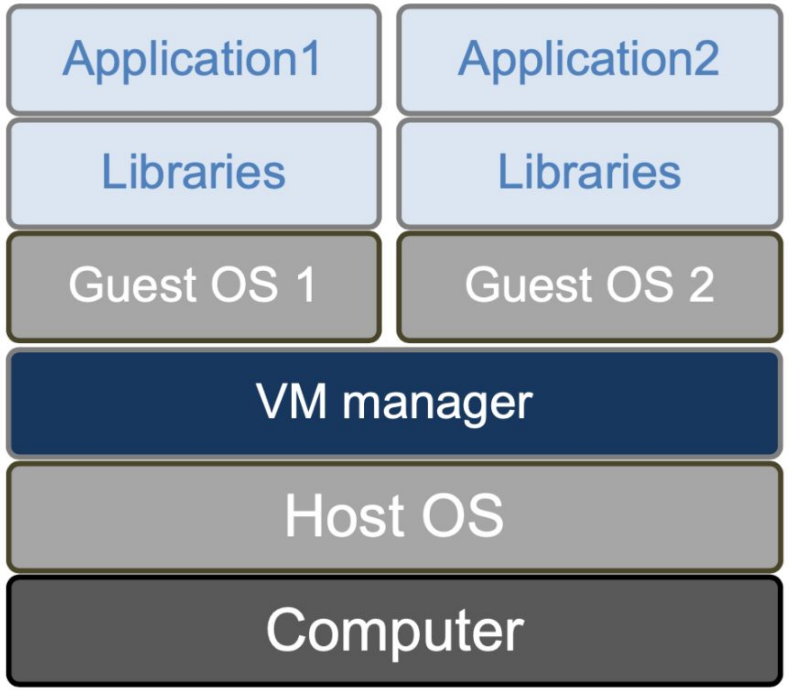
Idea : create separated environments for each application.
More versatile: create a new environment per analysis.

Encapsulation : hardware virtualization



But what if we want to install a software from a different OS?

Encapsulation : hardware virtualization

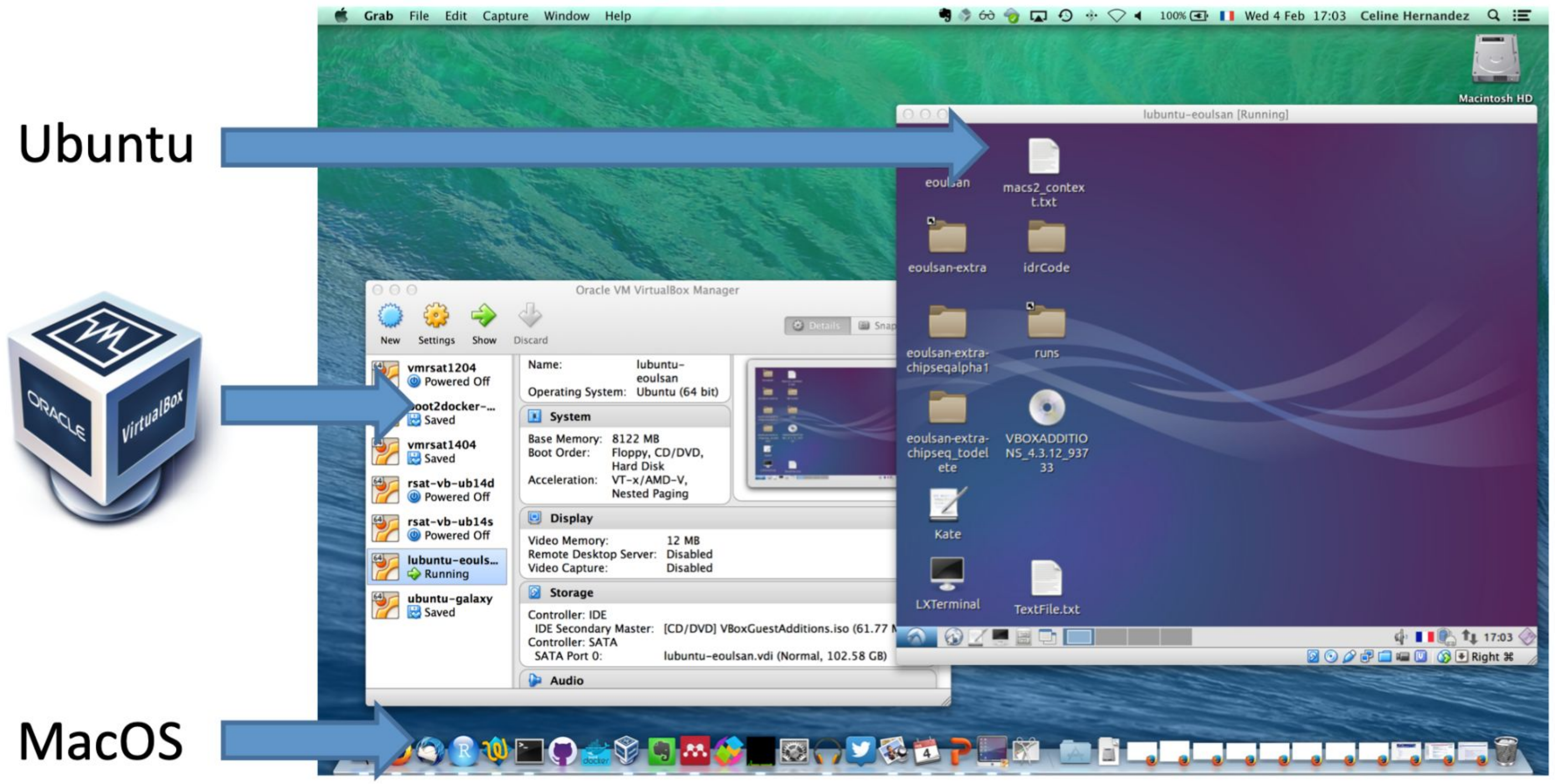


Idea: use virtual machines

Pros:

- Each application gets a completely different and independent environment
- Virtual machines can be transferred to another computer (using the same manager)

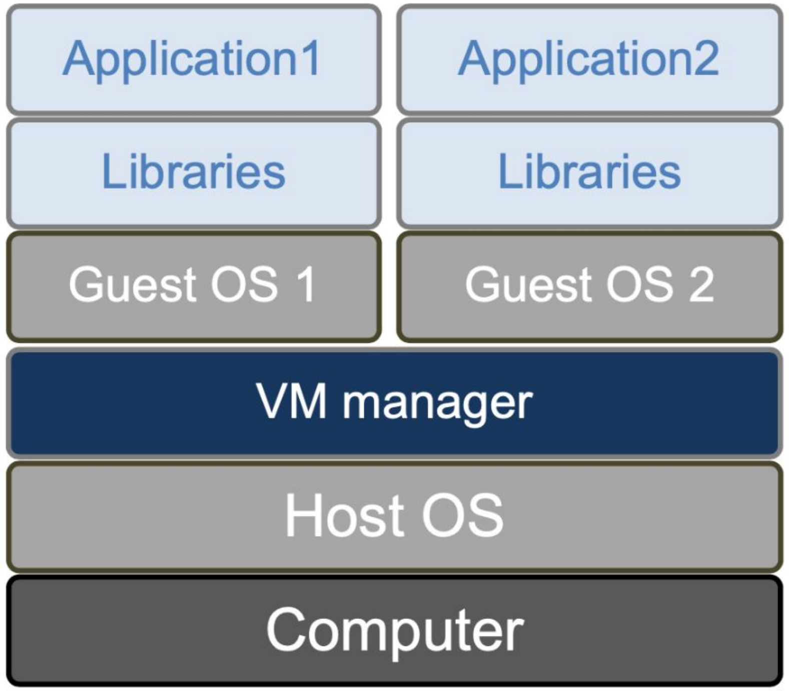
Encapsulation : hardware virtualization


Ubuntu
MacOS
ORACLE
VirtualBox
Grab File Edit Capture Window Help
Wed 4 Feb 17:03 Celine Hernandez
Macintosh HD
Oracle VM VirtualBox Manager
New
Settings
Show
Discard
Details
lubuntu-eoulsan [Running]
eoulsan
macs2_context.txt
eoulsan-extra
idrCode
eoulsan-extra-chipseqalpha1
runs
eoulsan-extra-chipseq_todelete
VBOXADDITIONS_4.3.12_93733
Kate
LXTerminal
TextFile.txt
Right ⌘

```
vmrsat1204      Powered Off
boot2docker-...  Saved
vmrsat1404      Saved
rsat-vb-ub14d   Powered Off
rsat-vb-ub14s   Powered Off
lubuntu-eouls... Running
ubuntu-galaxy   Saved
```

```
Name:              lubuntu-eoulsan
Operating System:  Ubuntu (64 bit)

System
Base Memory:  8122 MB
Boot Order:   Floppy, CD/DVD, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

Display
Video Memory:           12 MB
Remote Desktop Server:  Disabled
Video Capture:          Disabled

Storage
Controller: IDE
 IDE Secondary Master:  [CD/DVD] VBoxGuestAdditions.iso (61.77
Controller: SATA
 SATA Port 0:           lubuntu-eoulsan.vdi (Normal, 102.58 GB)

Audio
```

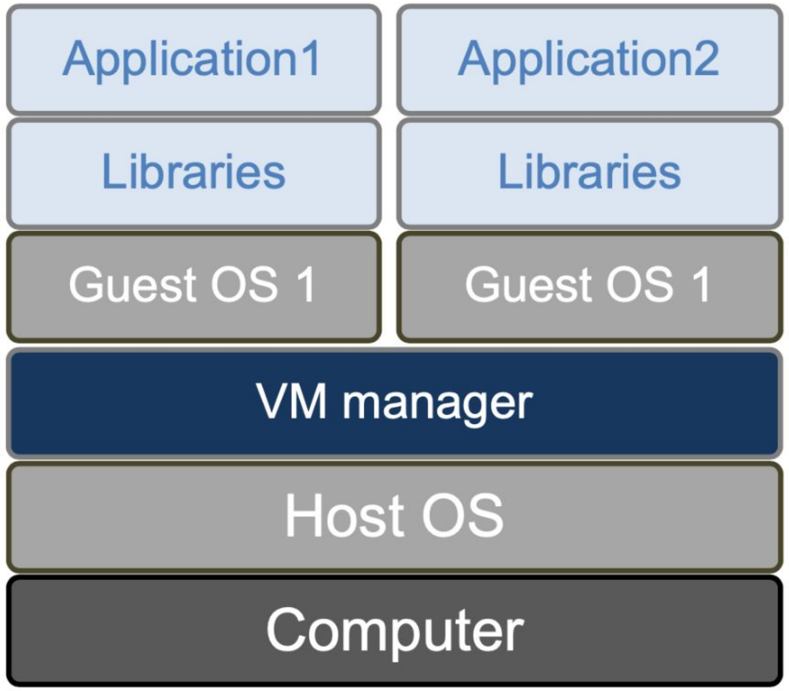
Encapsulation : hardware virtualization



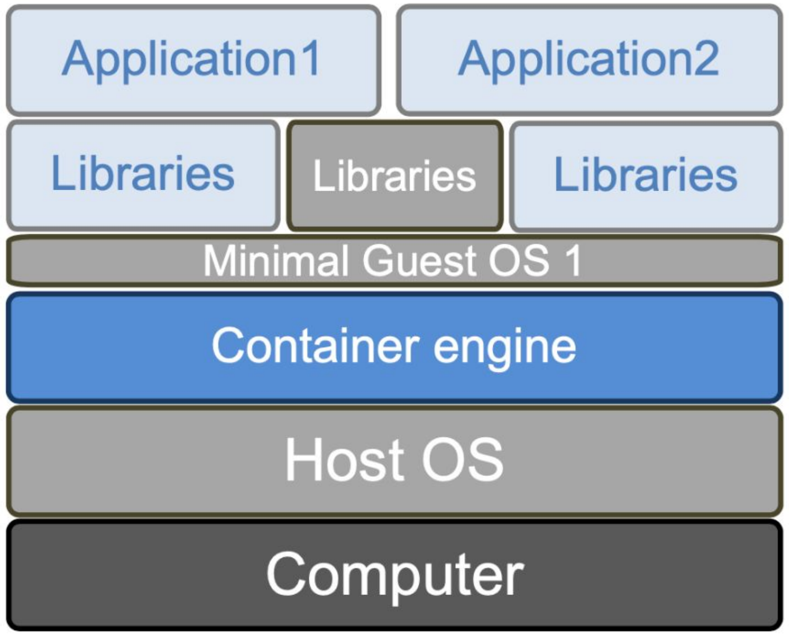
Idea: use virtual machines
Pros: transferable independent environments
Cons:

- Redundancy between VMs
- Heavy to set up
- No automation

Encapsulation : OS virtualization



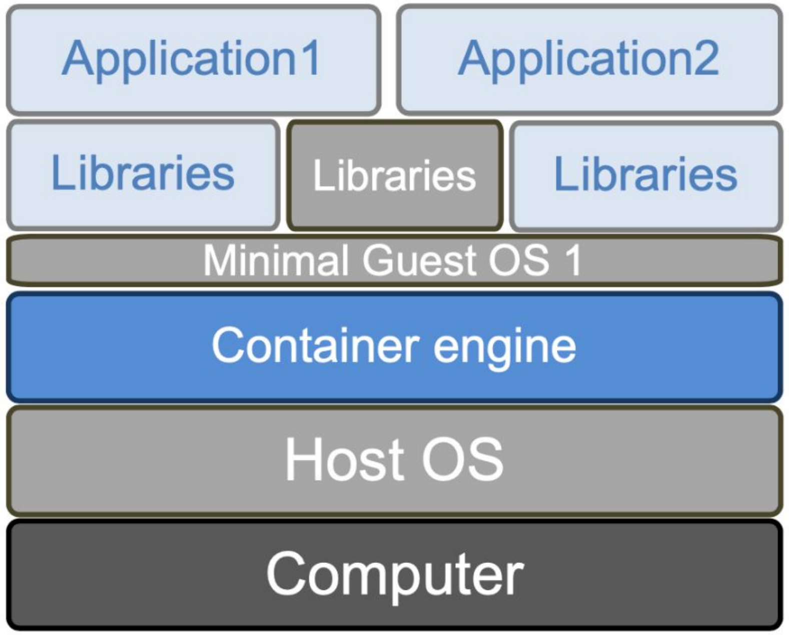
Encapsulation : OS virtualization



Idea: "trick" applications into believing that they are in a different OS than the host's

Avoid redundancy.

Encapsulation : OS virtualization

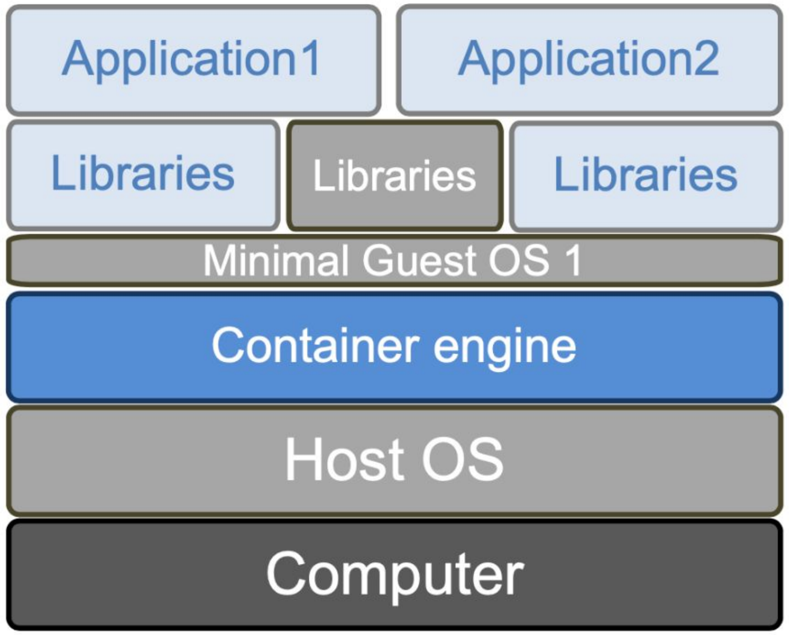


OS virtualisation vs hardware virtualisation

Pros:

- Speed
  - Installation is faster
  - No boot time
- Lightweight
  - Minimal base OS
  - Minimal libraries and application set
- Easy sharing of applications

Encapsulation : OS virtualization

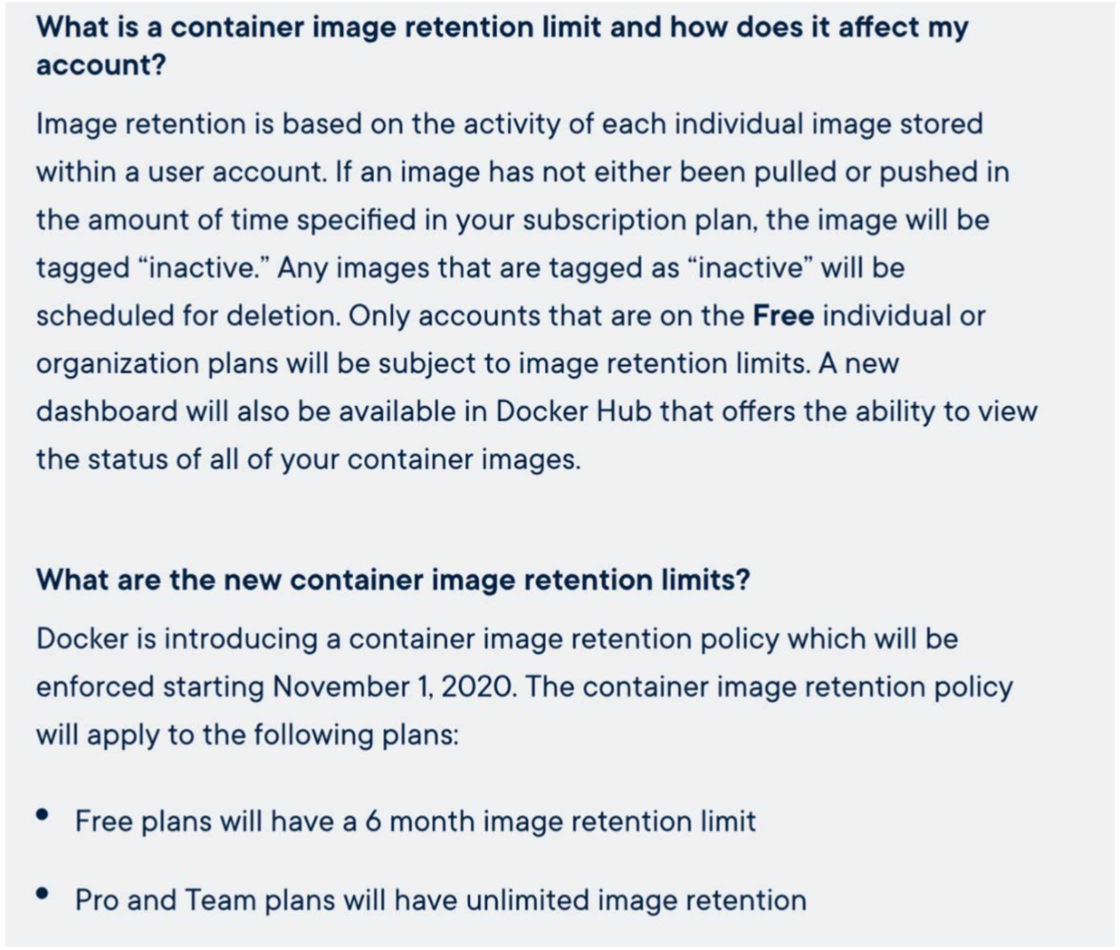


Cons:

- Singularity to use images on a cluster
- Changes of policies of the Docker company

Docker policy

Update of the Docker Image retention policy (13/08/2020)



What is a container image retention limit and how does it affect my account?

Image retention is based on the activity of each individual image stored within a user account. If an image has not either been pulled or pushed in the amount of time specified in your subscription plan, the image will be tagged "inactive." Any images that are tagged as "inactive" will be scheduled for deletion. Only accounts that are on the **Free** individual or organization plans will be subject to image retention limits. A new dashboard will also be available in Docker Hub that offers the ability to view the status of all of your container images.

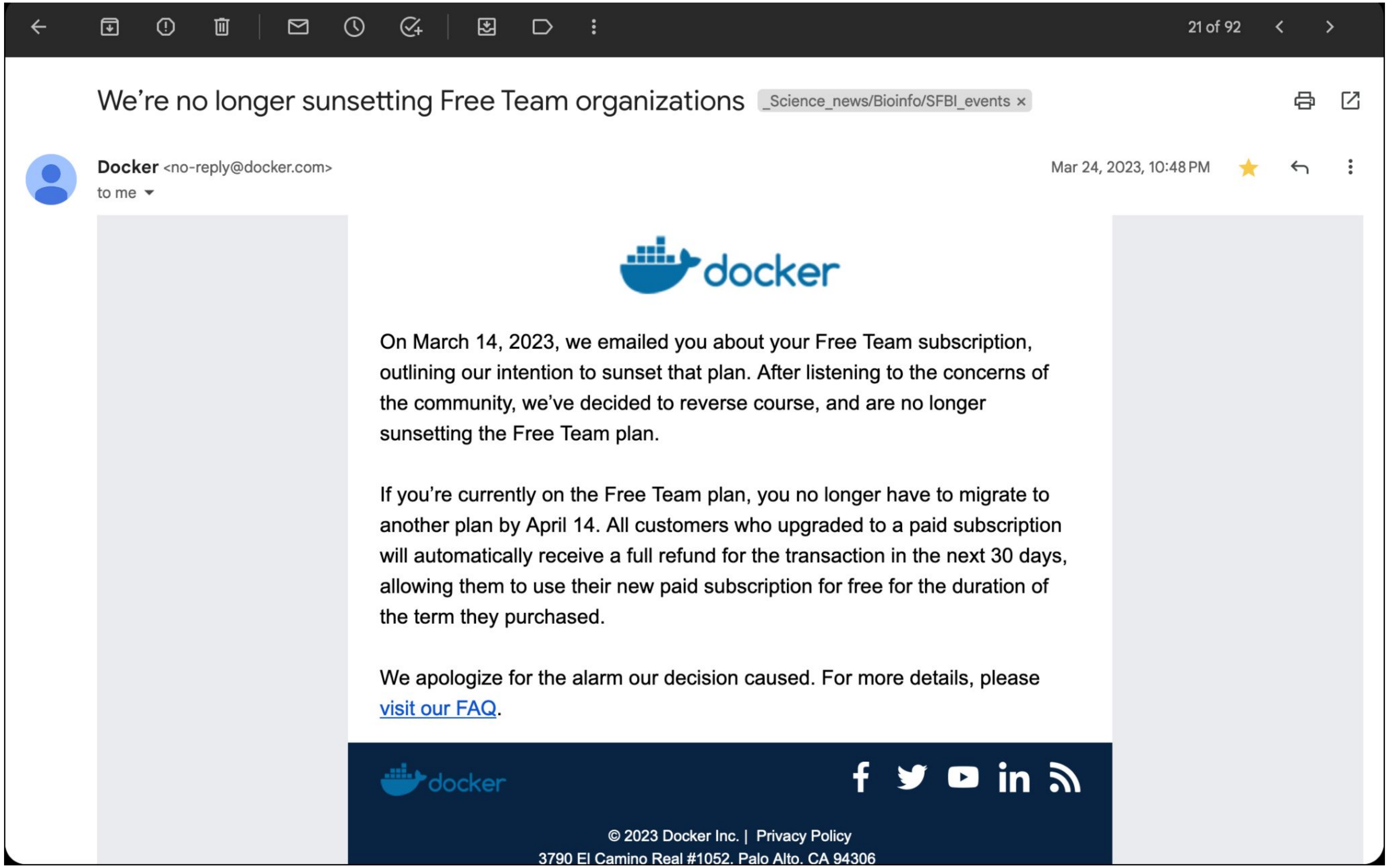
What are the new container image retention limits?

Docker is introducing a container image retention policy which will be enforced starting November 1, 2020. The container image retention policy will apply to the following plans:

- Free plans will have a 6 month image retention limit
- Pro and Team plans will have unlimited image retention

https://www.docker.com/pricing/retentionfaq

Docker policy: DockerHub usage



We're no longer sunsetting Free Team organizations

Docker <no-reply@docker.com>

to me

Mar 24, 2023, 10:48 PM

On March 14, 2023, we emailed you about your Free Team subscription, outlining our intention to sunset that plan. After listening to the concerns of the community, we've decided to reverse course, and are no longer sunsetting the Free Team plan.

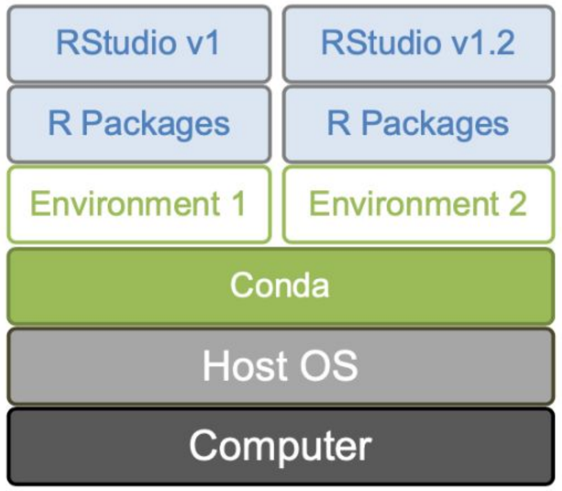
If you're currently on the Free Team plan, you no longer have to migrate to another plan by April 14. All customers who upgraded to a paid subscription will automatically receive a full refund for the transaction in the next 30 days, allowing them to use their new paid subscription for free for the duration of the term they purchased.

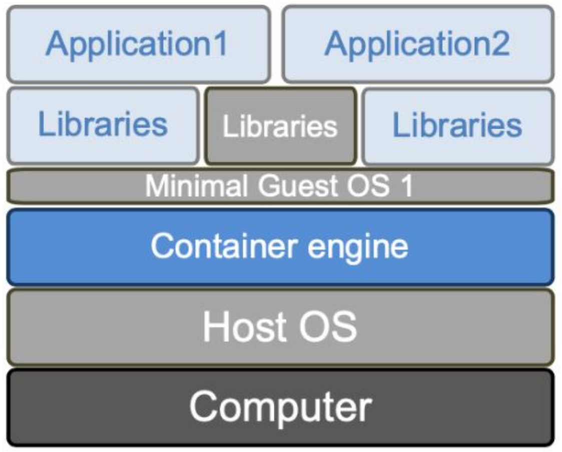
We apologize for the alarm our decision caused. For more details, please visit our FAQ.

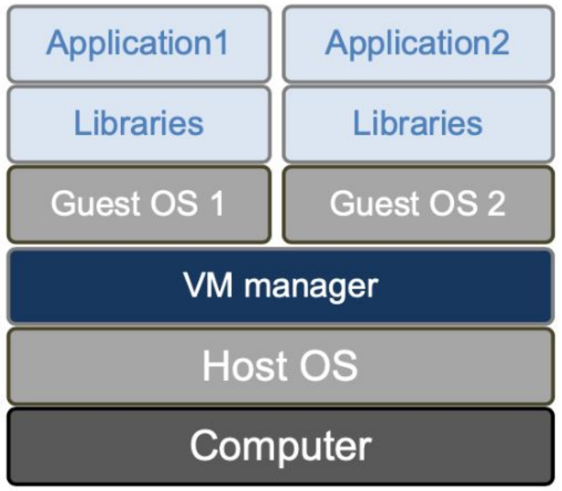
© 2023 Docker Inc. | Privacy Policy

3790 El Camino Real #1052, Palo Alto, CA 94306

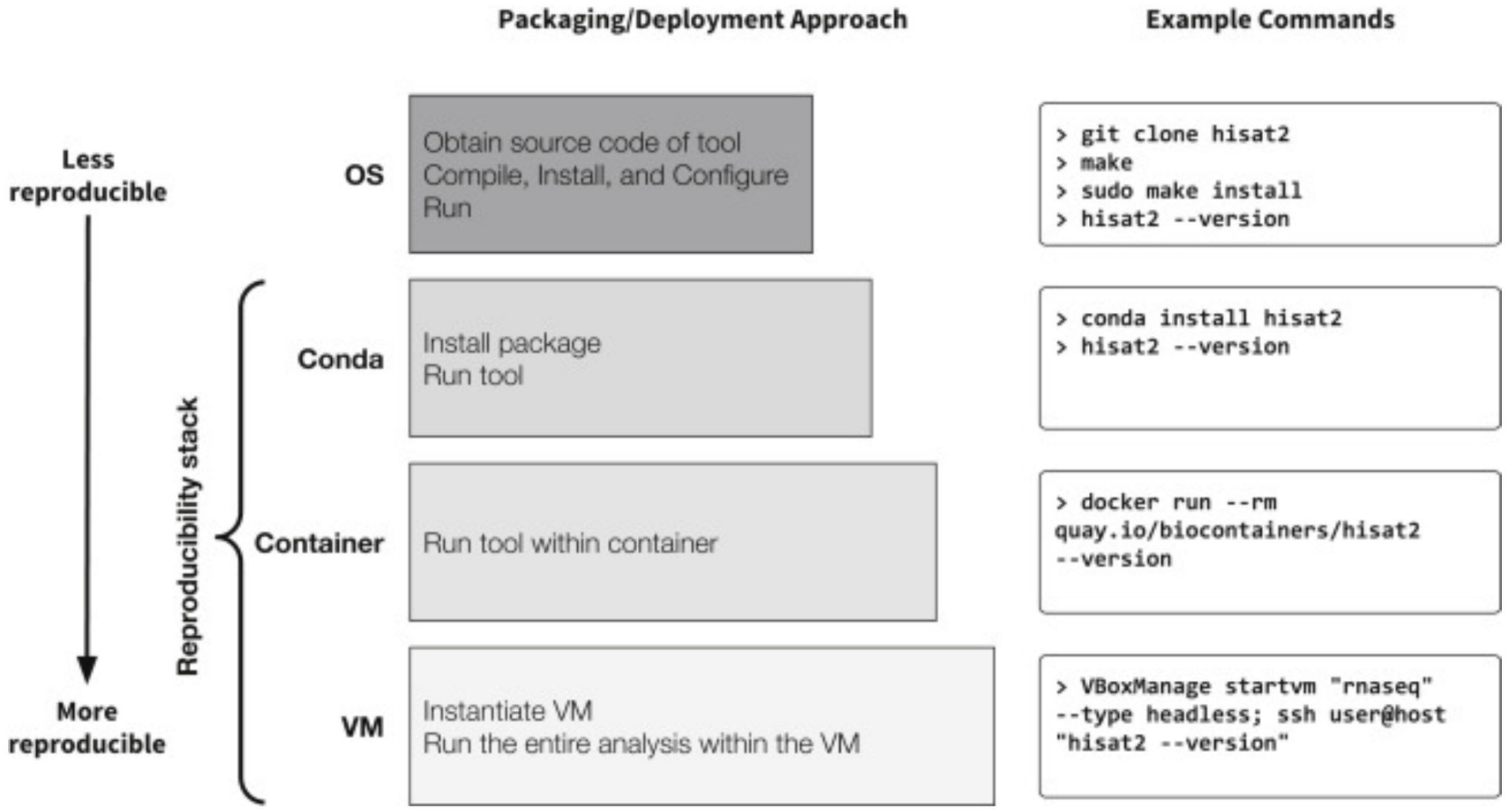
Encapsulation







Encapsulation and reproducibility stack


Packaging/Deployment Approach
Example Commands
Less reproducible
More reproducible
Reproducibility stack
OS
Obtain source code of tool
Compile, Install, and Configure
Run
Conda
Install package
Run tool
Container
Run tool within container
VM
Instantiate VM
Run the entire analysis within the VM

```
> git clone hisat2
> make
> sudo make install
> hisat2 --version
```

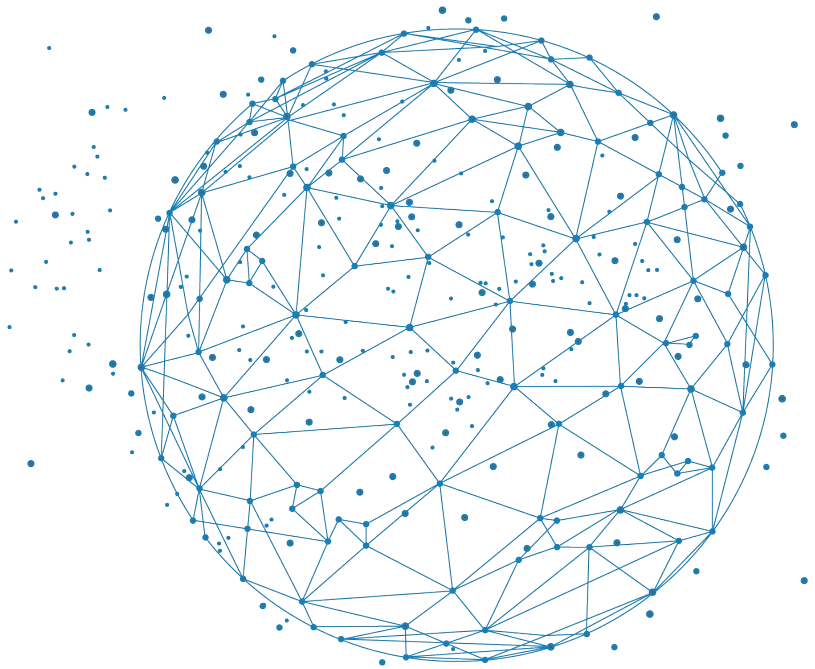
```
> conda install hisat2
> hisat2 --version
```

```
> docker run --rm
quay.io/biocontainers/hisat2
--version
```

```
> VBoxManage startvm "rnaseq"
--type headless; ssh user@host
"hisat2 --version"
```

Practical Computational Reproducibility in the Life Sciences - Björn Grüning et al (2018)

About Docker



About Docker

Docker is not very “old”

- First commit January 2013
- First version March 2013
- Version 1.0 in June 2014

But its adoption was fast

- Officially packaged in Ubuntu since 2014 (v14.04)

What is Docker?

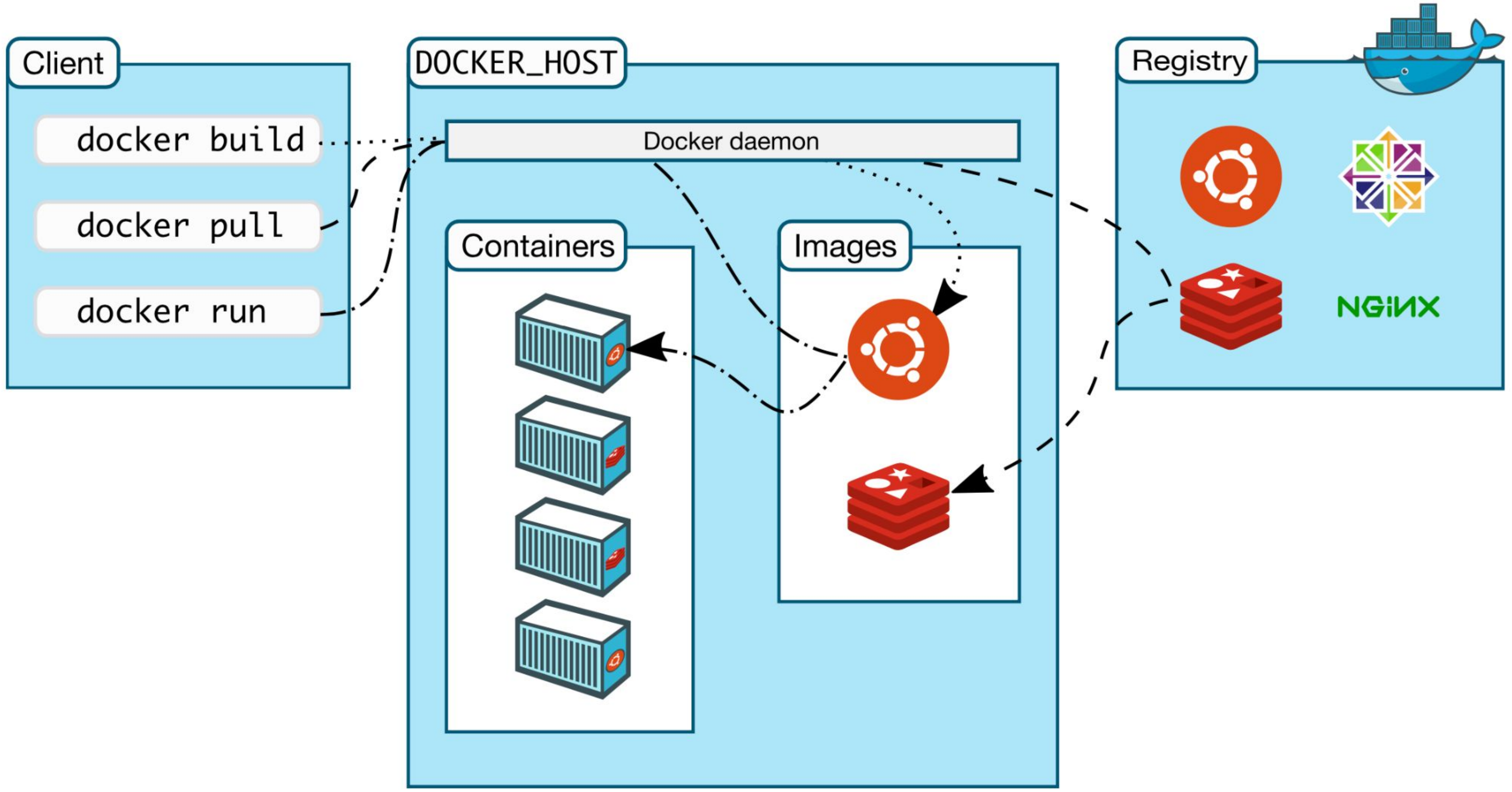
Image

- Set of libraries and functions
- Fixed. Cannot be modified
- Can be stored/shared online
- Can be automatically built

Container

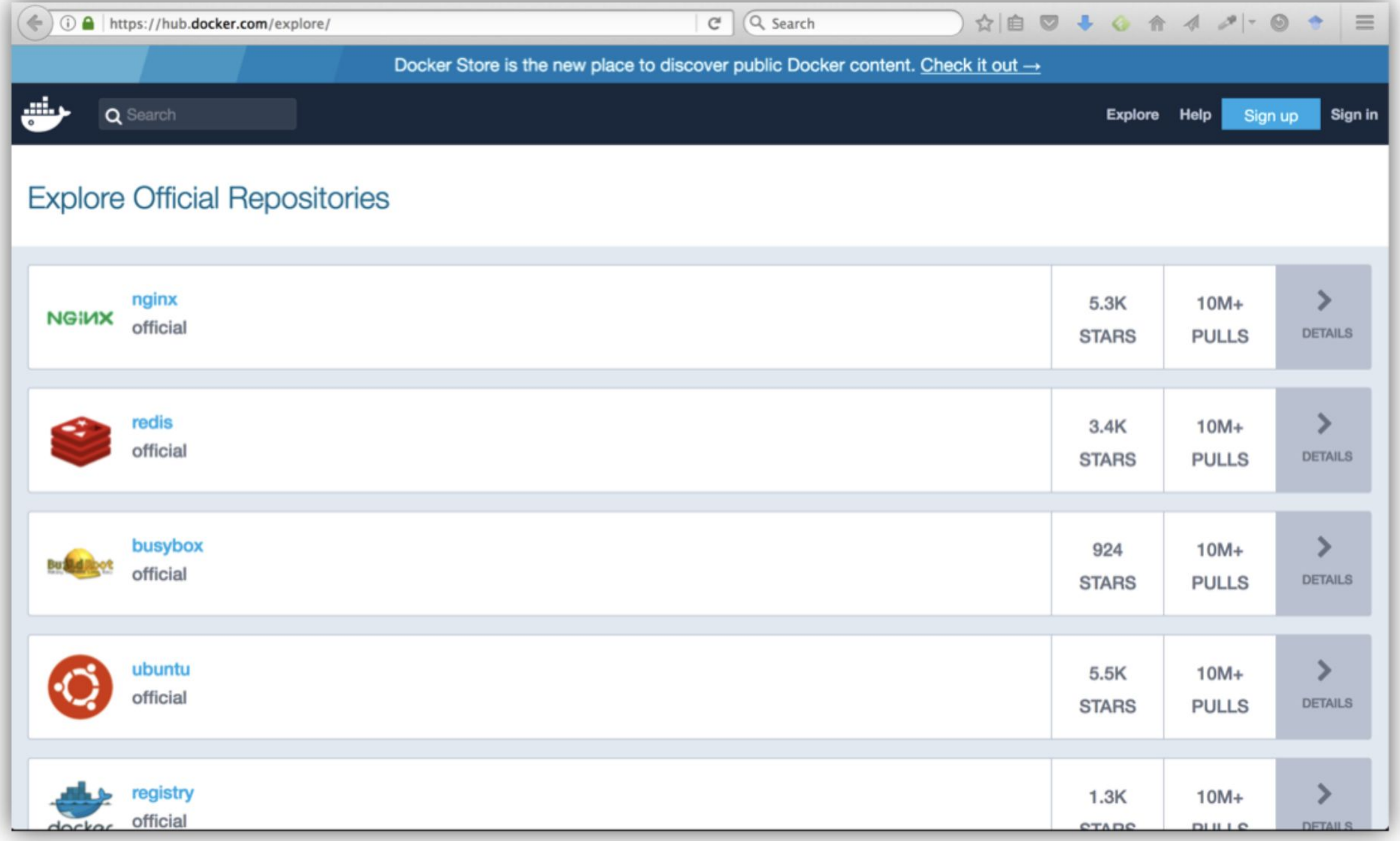
- "Active image"
- Can be modified (interactive)
- Can be turned into an image
- One image, many containers

What is Docker?



(https://docs.docker.com/get-started/overview/)

What is Docker?


https://hub.docker.com/explore/

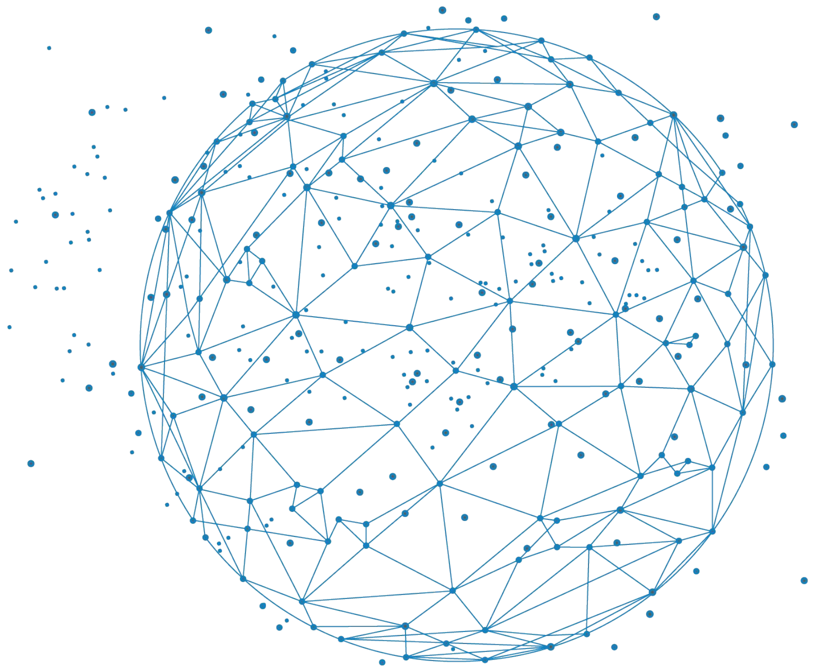
Docker Store is the new place to discover public Docker content. Check it out →

Explore Official Repositories

nginx official	5.3K STARS	10M+ PULLS	DETAILS
redis official	3.4K STARS	10M+ PULLS	DETAILS
busybox official	924 STARS	10M+ PULLS	DETAILS
ubuntu official	5.5K STARS	10M+ PULLS	DETAILS
registry official	1.3K STARS	10M+ PULLS	DETAILS

(https://hub.docker.com/)

Encapsulating our pipeline tools into Docker



Sharing our work…

What we have now ?

- A Git repository with our Notebook
- A environment lock file to recreate quickly a conda environment and get all dependencies

But

- Notebooks require a JupyterLab server…

Binder

What if we could shared a runnable JupyterLab environment including our Notebook and all its dependencies ?

Introducing…

binder

`https://mybinder.org`

Binder

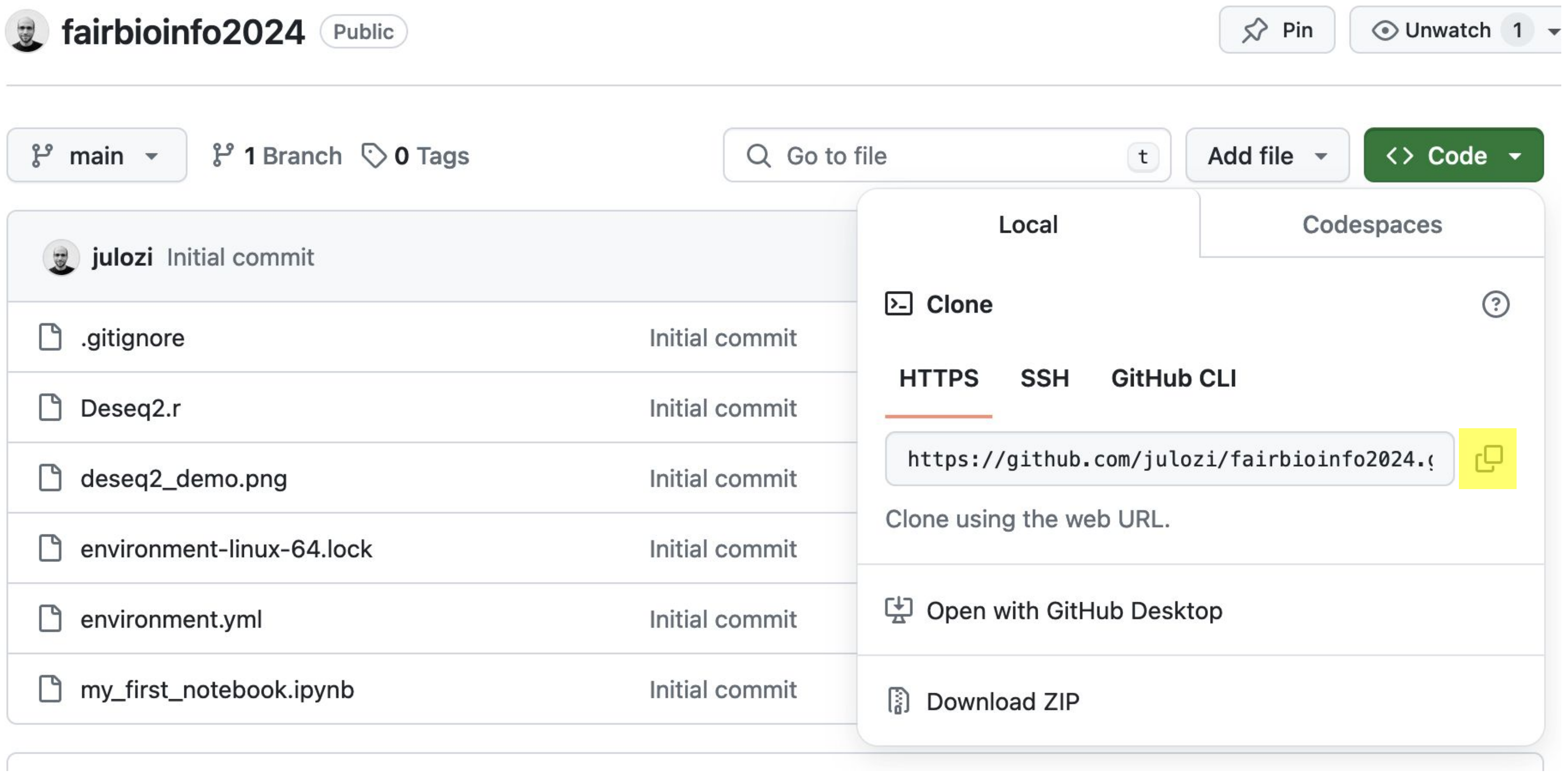
Binder is an online service that lets you share your notebook(s) in a interactive environment.

How does it work ?

BinderHub (the backend of Binder) do the following operation:

- Fetch your repo from GitHub
- Analyse the contents
- Build a Docker image based on your repo
- Launch that Docker image in the cloud
- Connect you to it via your browser

Get the HTTPS url of your repository on Github


fairbioinfo2024 Public

Pin · Unwatch 1

main · 1 Branch · 0 Tags · Go to file · Add file · Code

julozi Initial commit

.gitignore	Initial commit
Deseq2.r	Initial commit
deseq2_demo.png	Initial commit
environment-linux-64.lock	Initial commit
environment.yml	Initial commit
my_first_notebook.ipynb	Initial commit

Local · Codespaces

Clone

HTTPS · SSH · GitHub CLI

```
https://github.com/julozi/fairbioinfo2024.g
```

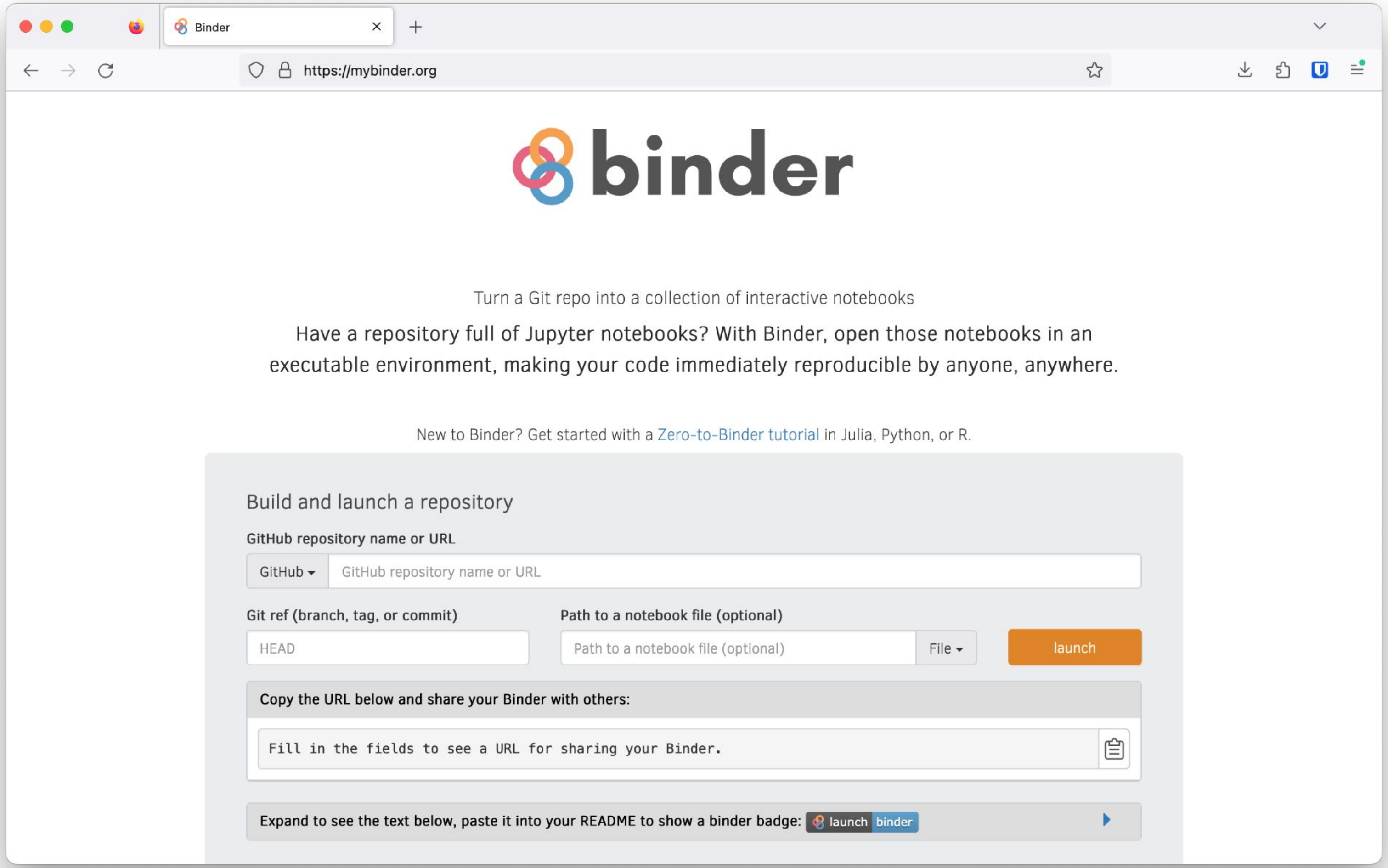
Clone using the web URL.

Open with GitHub Desktop

Download ZIP

Binder

Go to https://mybinder.org



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a Zero-to-Binder tutorial in Julia, Python, or R.

Build and launch a repository

GitHub repository name or URL

GitHub ▾ | GitHub repository name or URL

Git ref (branch, tag, or commit) | Path to a notebook file (optional)

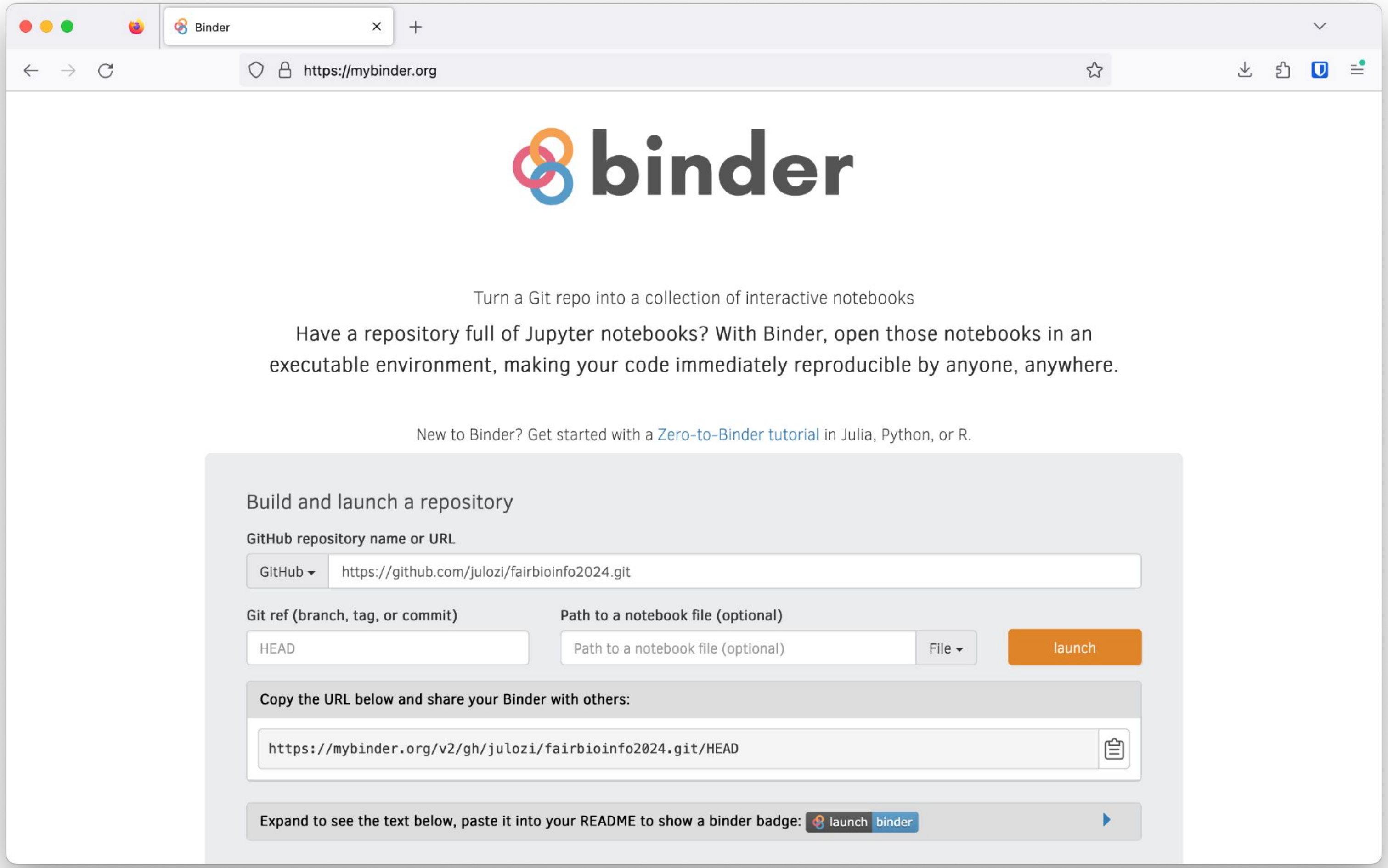
HEAD | Path to a notebook file (optional) | File ▾ | launch

Copy the URL below and share your Binder with others:

```
Fill in the fields to see a URL for sharing your Binder.
```

Expand to see the text below, paste it into your README to show a binder badge: launch binder

Paste your Github repository URL and click Launch



Turn a Git repo into a collection of interactive notebooks

Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.

New to Binder? Get started with a Zero-to-Binder tutorial in Julia, Python, or R.

Build and launch a repository

GitHub repository name or URL: GitHub https://github.com/julozi/fairbioinfo2024.git

Git ref (branch, tag, or commit): HEAD

Path to a notebook file (optional): Path to a notebook file (optional) File

launch

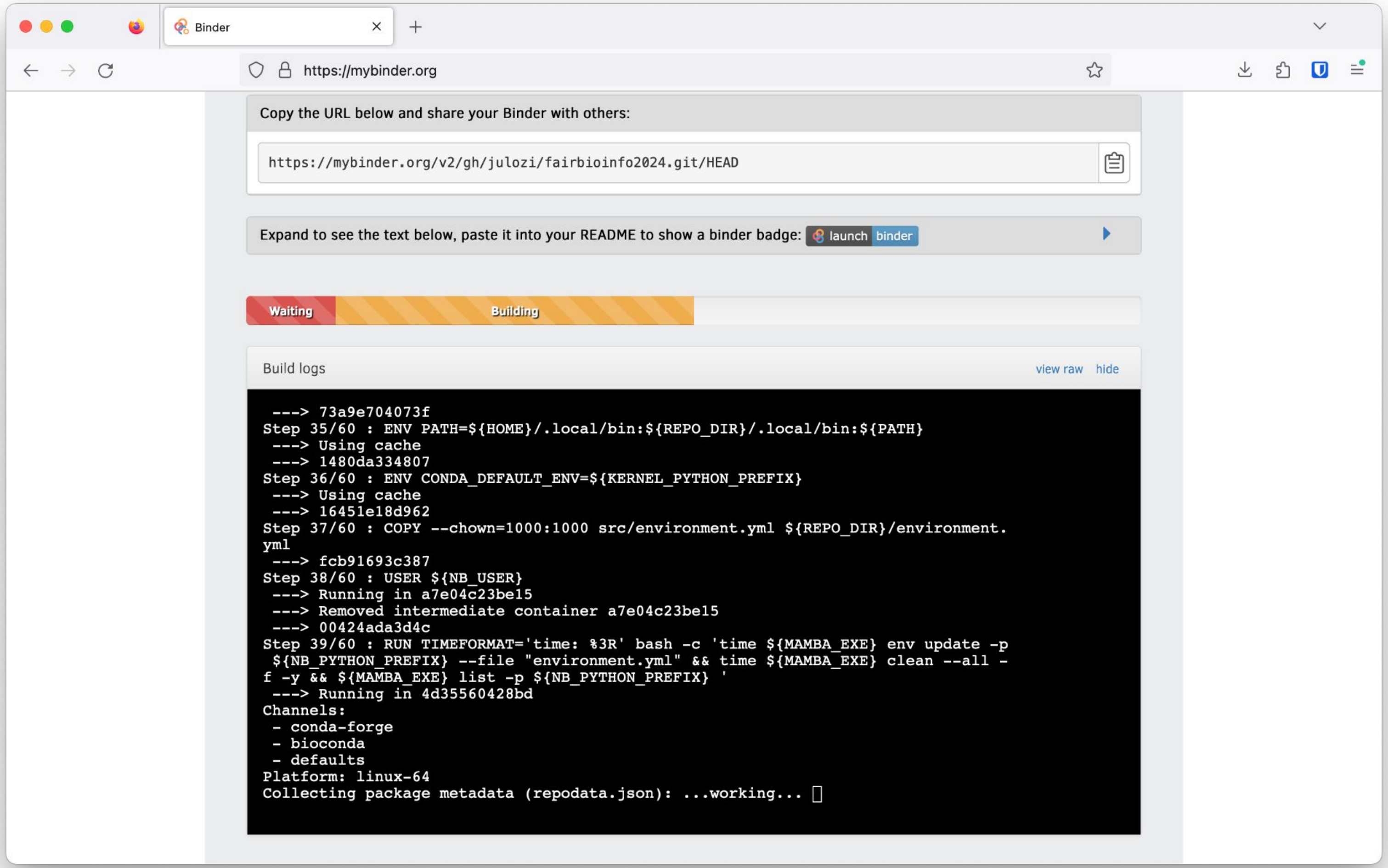
Copy the URL below and share your Binder with others:

```
https://mybinder.org/v2/gh/julozi/fairbioinfo2024.git/HEAD
```

Expand to see the text below, paste it into your README to show a binder badge: launch binder

The build process is running…

Binder is picking up our environment.yml file automatically !


Binder
https://mybinder.org
Copy the URL below and share your Binder with others:

```
https://mybinder.org/v2/gh/julozi/fairbioinfo2024.git/HEAD
```

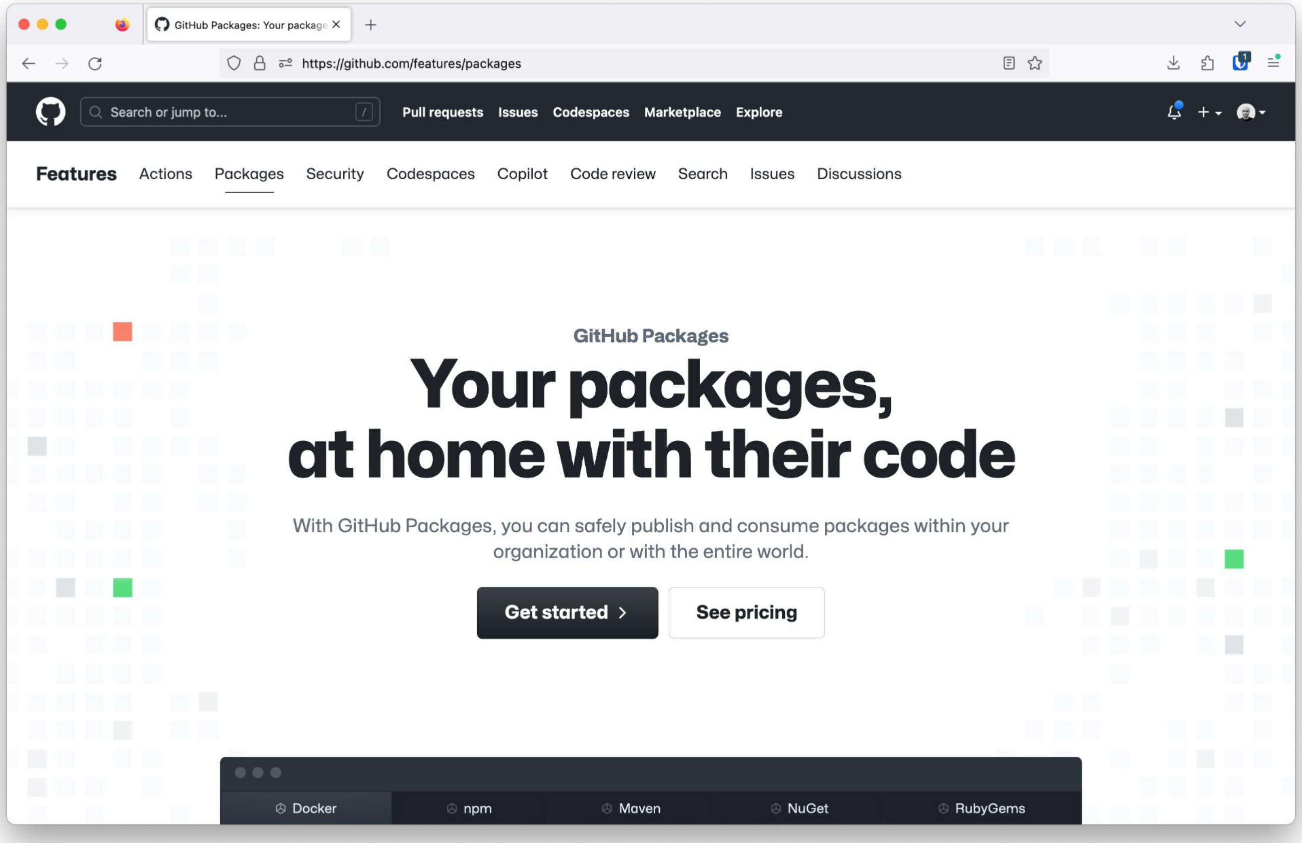
Expand to see the text below, paste it into your README to show a binder badge:
launch binder
Waiting
Building
Build logs
view raw
hide

```
 ---> 73a9e704073f
Step 35/60 : ENV PATH=${HOME}/.local/bin:${REPO_DIR}/.local/bin:${PATH}
 ---> Using cache
 ---> 1480da334807
Step 36/60 : ENV CONDA_DEFAULT_ENV=${KERNEL_PYTHON_PREFIX}
 ---> Using cache
 ---> 16451e18d962
Step 37/60 : COPY --chown=1000:1000 src/environment.yml ${REPO_DIR}/environment.
yml
 ---> fcb91693c387
Step 38/60 : USER ${NB_USER}
 ---> Running in a7e04c23be15
 ---> Removed intermediate container a7e04c23be15
 ---> 00424ada3d4c
Step 39/60 : RUN TIMEFORMAT='time: %3R' bash -c 'time ${MAMBA_EXE} env update -p
 ${NB_PYTHON_PREFIX} --file "environment.yml" && time ${MAMBA_EXE} clean --all -
f -y && ${MAMBA_EXE} list -p ${NB_PYTHON_PREFIX} '
 ---> Running in 4d35560428bd
Channels:
 - conda-forge
 - bioconda
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): ...working...
```

Docker image

What if I don’t have a Notebook. Can I still share my work as a Docker image ?

Introducing… **Github Container Registry**


GitHub Packages: Your package

https://github.com/features/packages

GitHub Packages

Your packages, at home with their code

With GitHub Packages, you can safely publish and consume packages within your organization or with the entire world.

Docker image

Let’s create a shell version of our Notebook

```
$ jupyter nbconvert my_first_notebook.ipynb --to script
[NbConvertApp] Converting notebook my_first_notebook.ipynb to script
[NbConvertApp] Writing 990 bytes to my_first_notebook.sh
```

Docker image

We need to **customize our script** to make it **more portable**

Add the **bash shebang** at the beginning of the `my_first_notebook.sh` script :

```
#!/usr/bin/env bash
```

Set a variable to retrieve the directory containing the script (we need this to call the R script)

```
SCRIPT_DIR=$( cd -- "$( dirname -- "${BASH_SOURCE[0]}" )" &> /dev/null && pwd )
```

Use the variable in the R script call

```
R < $SCRIPT_DIR/Deseq2.r --no-save
```

Docker image

Add the new shell script to your repository

```
$ git add my_first_notebook.sh
$ git commit -m "add bash version of the pipeline"
```

Docker image

Github can help us create automatically a Docker image based on a Dockerfile located in our repository.

The step to do this are :

- Create a Dockerfile and push it Github
- Enable the Publish Docker Container Github action

Docker image

Create a new docker folder in your pipeline folder

```
$ mkdir docker
```

Docker image

Create a new file named `Dockerfile` in the `docker` folder

```
FROM mambaorg/micromamba:1.5.6
ADD ../environment.yml .
RUN micromamba install -y -n base -f environment.yml && \
     micromamba clean --all --yes

USER root
RUN mkdir /opt/pipeline
ENV PATH="/opt/pipeline:$PATH"
ADD ../my_first_notebook.sh /opt/pipeline/
ADD ../Deseq2.r /opt/pipeline
RUN chmod +x /opt/pipeline/my_first_notebook.sh

USER $MAMBA_USER
```

Create an image based on micromamba distribution
Add the environment.yml file
Install packages listed in the environment.yml in the base environment and cleanup everything to make the image as light as possible

Switch to root user
Create a /opt/pipeline folder
Add the /opt/pipeline folder to the image $PATH
Add the pipeline script to the /opt/pipeline folder
Also add the Deseq2.r file
Make sure the script is executable

Switch back to mamba user

Add the docker folder to the repository

```
$ git add docker
$ git commit -m "add support for docker"
```

Docker image

Push the last commits to Github

```
$ git push origin main
```

Docker image

Github let you define actions that will be triggered automatically when your repository changes.

Actions are configure through **YAML files** called workflow.

Github proposes a set of pre-defined actions ready to use.

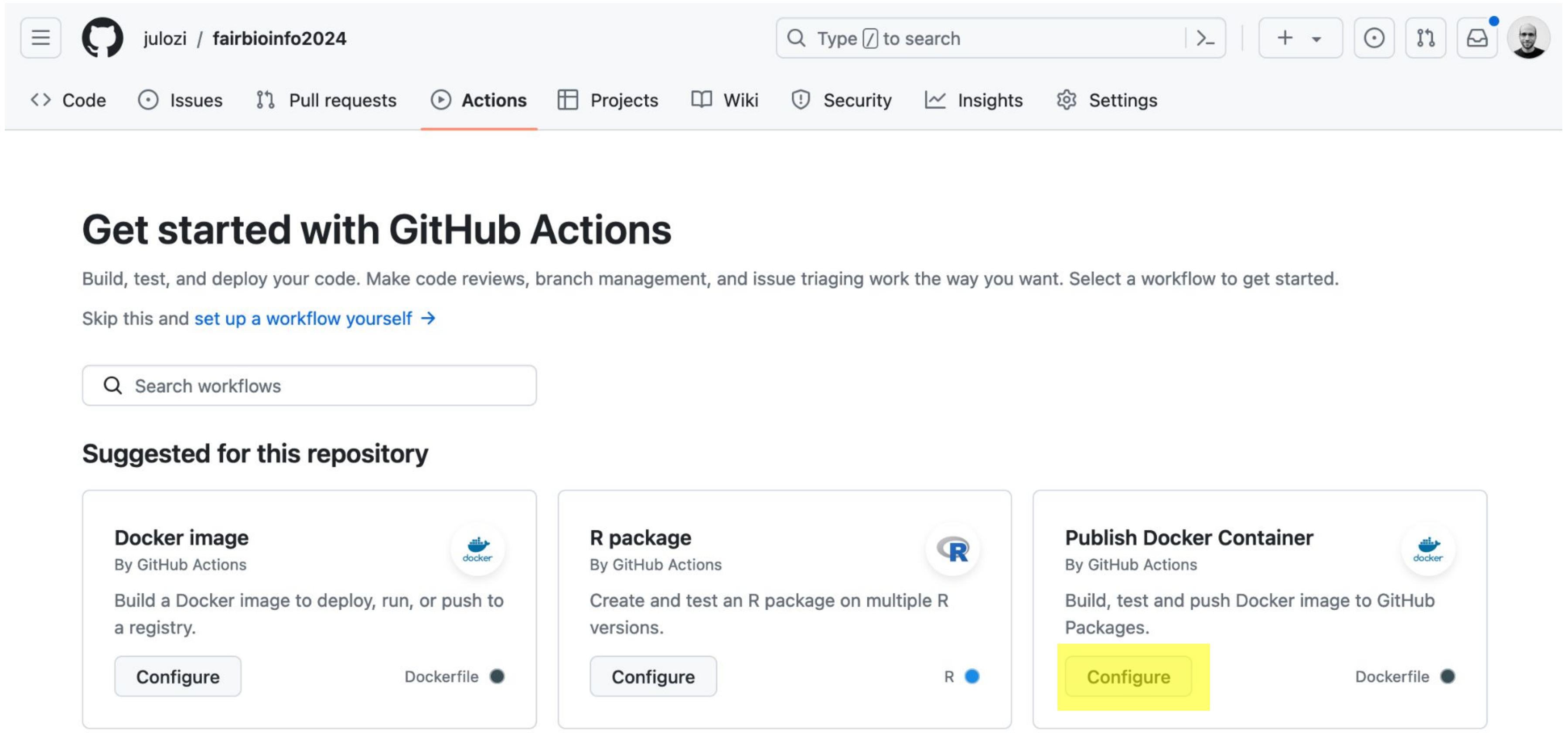
The **Publish Docker Container** action is a workflow that will build a Docker image based on a Dockerfile found at the root of your repository. It will then publish it in the Docker registry provided by Github.

Docker image

Go to your repository on Github and move to the **Actions** tab

You should see the **Publish Docker Container** action suggested

Click its **Configure** button



Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and set up a workflow yourself →

Suggested for this repository

Workflow	Author	Description	Button	Language
Docker image	By GitHub Actions	Build a Docker image to deploy, run, or push to a registry.	Configure	Dockerfile
R package	By GitHub Actions	Create and test an R package on multiple R versions.	Configure	R
Publish Docker Container	By GitHub Actions	Build, test and push Docker image to GitHub Packages.	Configure	Dockerfile

Docker image

Github gives us a publish-docker.yml workflow ready to be pushed on our repository.

We need to make **two changes** in this file :

First, the image signing tool proposed by Github is outdated (this is a bug).
We need to upgrade it :

```
# Install the cosign tool except on PR
# https://github.com/sigstore/cosign-installer
- name: Install cosign
    if: github.event_name != 'pull_request'
    # uses: sigstore/cosign-installer@6e04d228eb30da1757ee4e1dd75a0ec73a653e06 #v3.1.1
    uses: sigstore/cosign-installer@e1523de7571e31dbe865fd2e80c5c7c23ae71eb4 #v3.4.0
    with:
    # cosign-release: 'v2.1.1'
    cosign-release: 'v2.2.3'
```

Docker image

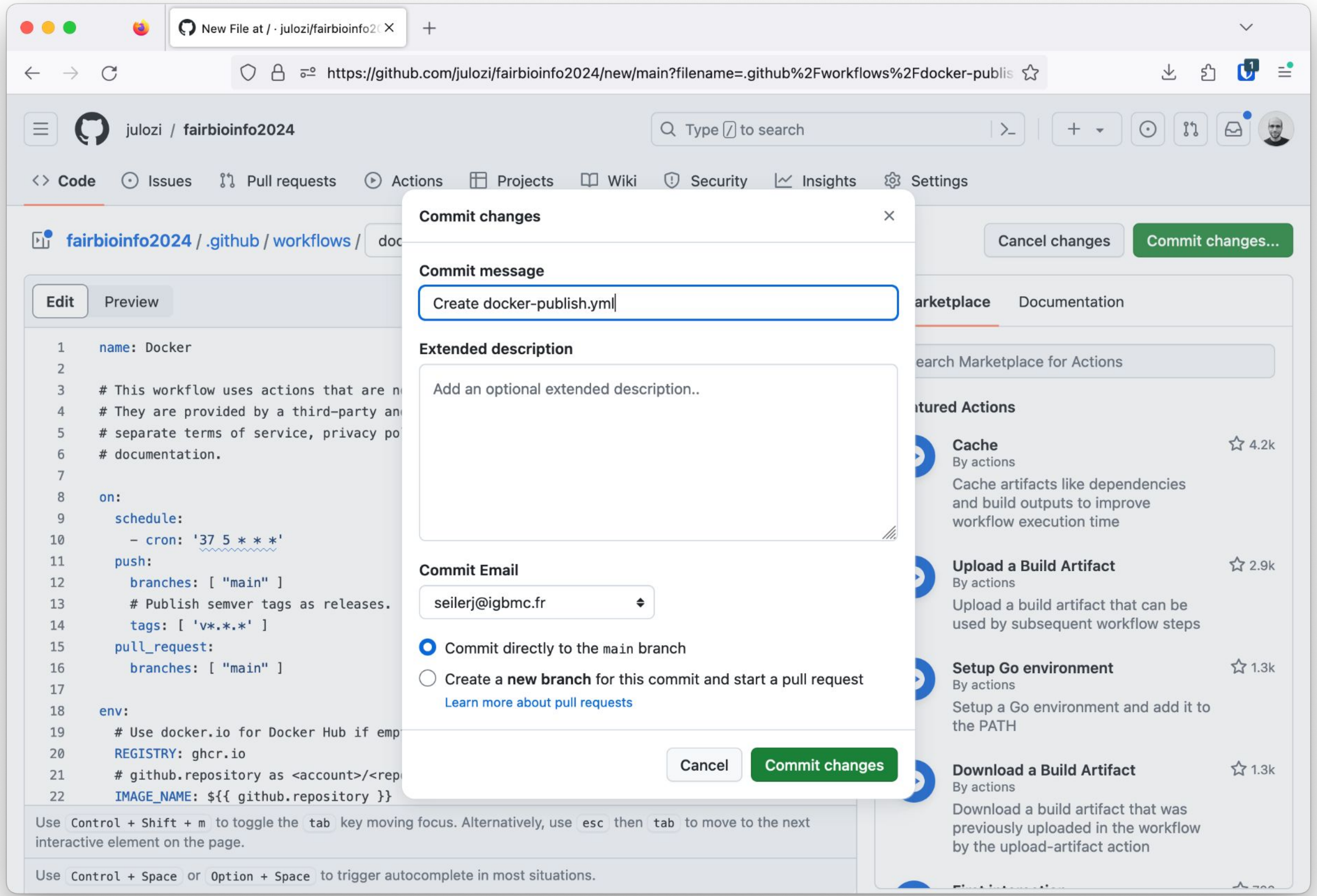
By default, the workflow is looking for a Dockerfile at the root of the repository. However we have created our Dockerfile in a docker folder.

We fix the job to use the docker/Dockerfile :

```
      - name: Build and push Docker image
        id: build-and-push
        uses: docker/build-push-action@0565240e2d4ab88bba5387d719585280857ece09 # v5.0.0
        with:
        context: .
        file: ./docker/Dockerfile
        push: ${{ github.event_name != 'pull_request' }}
        tags: ${{ steps.meta.outputs.tags }}
        labels: ${{ steps.meta.outputs.labels }}
        cache-from: type=gha
        cache-to: type=gha,mode=max
```

Docker image

Click the green **Commit changes…** button and validate the commit in the modal window.



New File at / · julozi/fairbioinfo2

https://github.com/julozi/fairbioinfo2024/new/main?filename=.github%2Fworkflows%2Fdocker-publis

julozi / fairbioinfo2024

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

fairbioinfo2024 / .github / workflows / doc

Cancel changes Commit changes...

Edit Preview

```
name: Docker

# This workflow uses actions that are n
# They are provided by a third-party an
# separate terms of service, privacy po
# documentation.

on:
  schedule:
    - cron: '37 5 * * *'
  push:
    branches: [ "main" ]
    # Publish semver tags as releases.
    tags: [ 'v*.*.*' ]
  pull_request:
    branches: [ "main" ]

env:
  # Use docker.io for Docker Hub if emp
  REGISTRY: ghcr.io
  # github.repository as <account>/<rep
  IMAGE_NAME: ${{ github.repository }}
```

Use Control + Shift + m to toggle the tab key moving focus. Alternatively, use esc then tab to move to the next interactive element on the page.

Use Control + Space or Option + Space to trigger autocomplete in most situations.

Commit changes

Commit message: Create docker-publish.yml

Extended description: Add an optional extended description..

Commit Email: seilerj@igbmc.fr

[✓ Commit directly to the main branch]

Create a **new branch** for this commit and start a pull request

Learn more about pull requests

Cancel Commit changes

arketplace Documentation

earch Marketplace for Actions

tured Actions

Cache ☆ 4.2k
By actions
Cache artifacts like dependencies and build outputs to improve workflow execution time

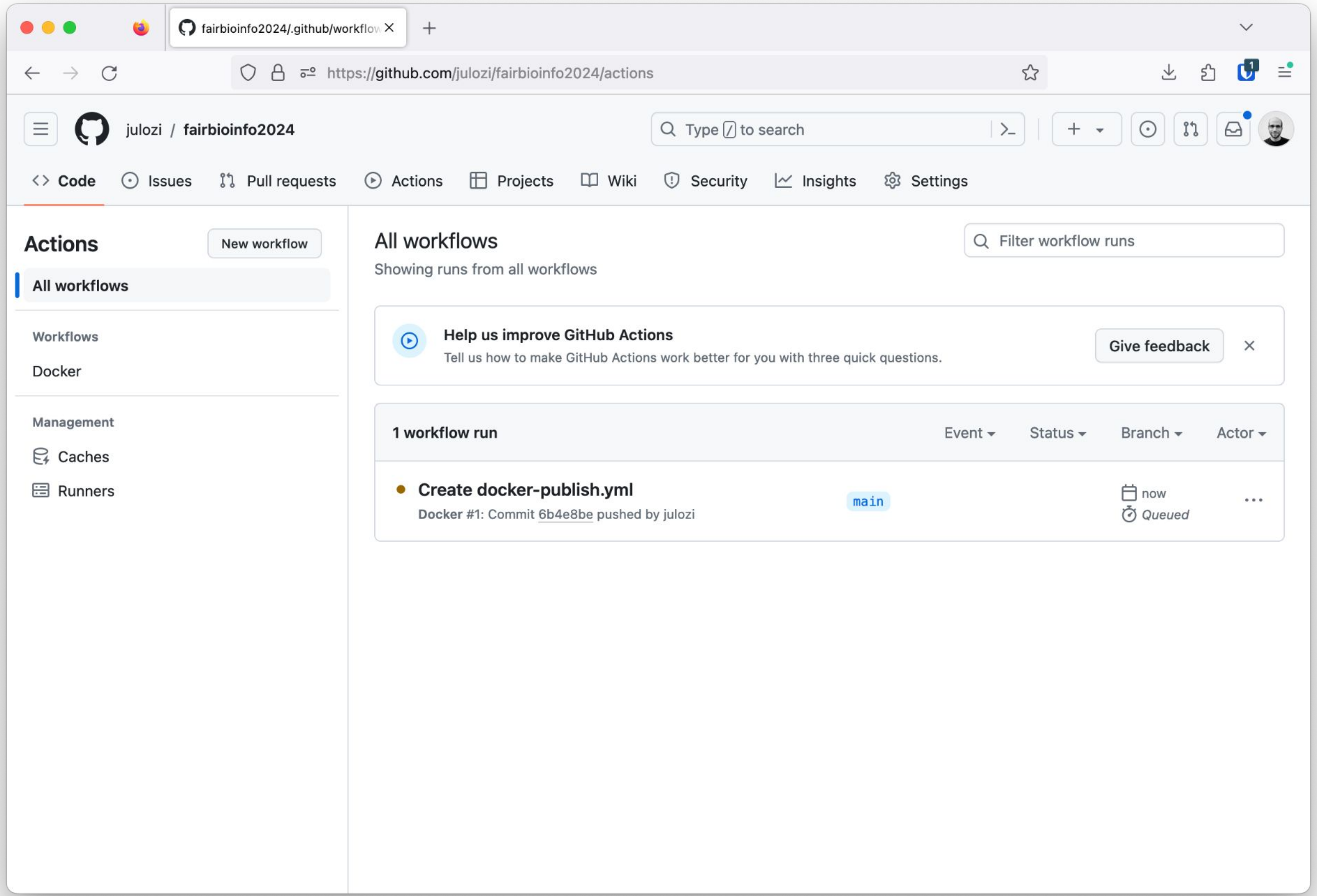
Upload a Build Artifact ☆ 2.9k
By actions
Upload a build artifact that can be used by subsequent workflow steps

Setup Go environment ☆ 1.3k
By actions
Setup a Go environment and add it to the PATH

Download a Build Artifact ☆ 1.3k
By actions
Download a build artifact that was previously uploaded in the workflow by the upload-artifact action

Docker image

Github will automatically trigger the new action workflow.
You can find it back in the **Actions** tab.



All workflows

Showing runs from all workflows

Help us improve GitHub Actions

Tell us how to make GitHub Actions work better for you with three quick questions.

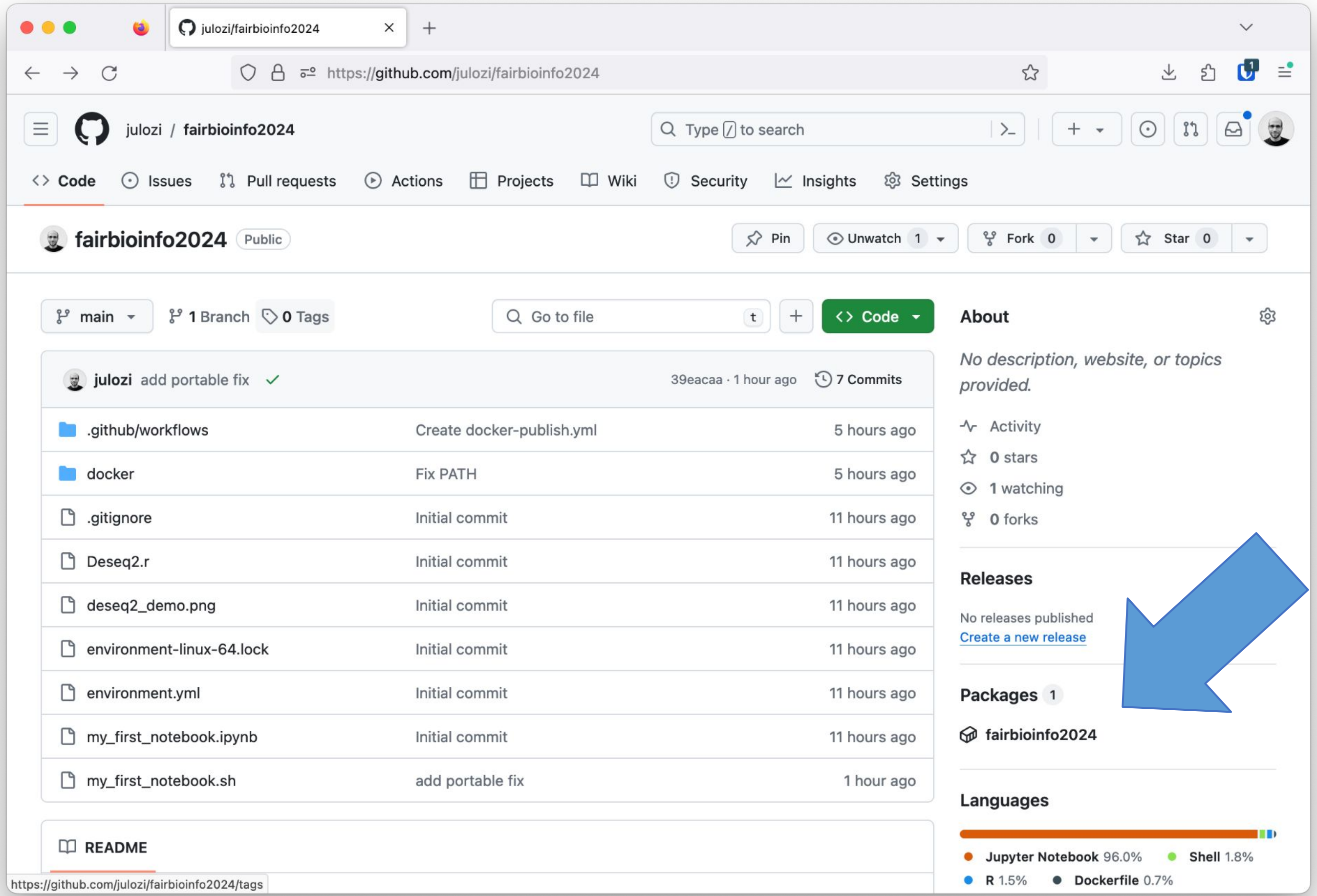
1 workflow run

Create docker-publish.yml

Docker #1: Commit 6b4e8be pushed by julozi

Docker image

When the build is finished successfully, the Docker image can be found in Packages section of the repository.



julozi/fairbioinfo2024

https://github.com/julozi/fairbioinfo2024

julozi / fairbioinfo2024

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

fairbioinfo2024 Public

Pin Unwatch 1 Fork 0 Star 0

main 1 Branch 0 Tags Go to file Code

julozi add portable fix ✓ 39eacaa · 1 hour ago 7 Commits

.github/workflows	Create docker-publish.yml	5 hours ago
docker	Fix PATH	5 hours ago
.gitignore	Initial commit	11 hours ago
Deseq2.r	Initial commit	11 hours ago
deseq2_demo.png	Initial commit	11 hours ago
environment-linux-64.lock	Initial commit	11 hours ago
environment.yml	Initial commit	11 hours ago
my_first_notebook.ipynb	Initial commit	11 hours ago
my_first_notebook.sh	add portable fix	1 hour ago

README

https://github.com/julozi/fairbioinfo2024/tags

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages 1

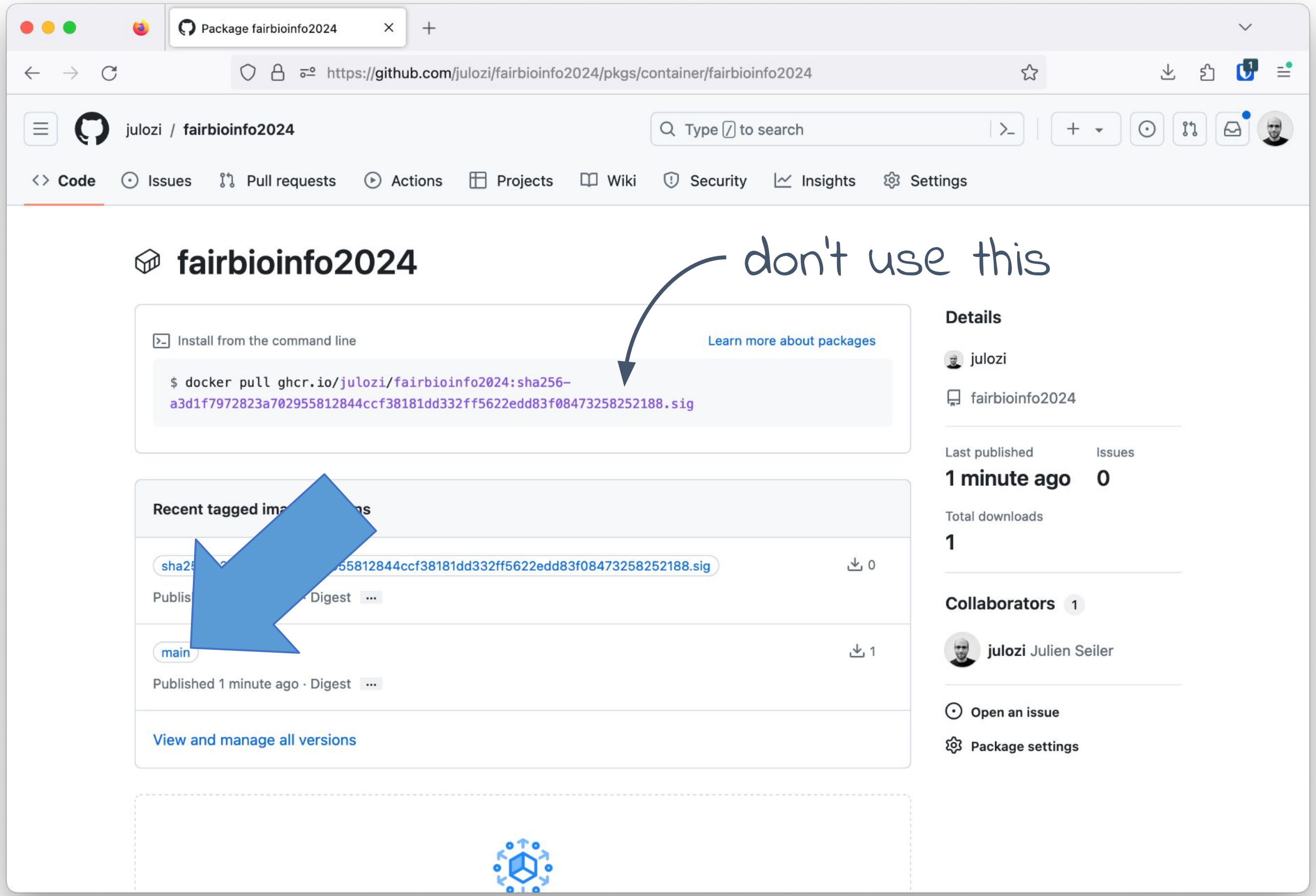
fairbioinfo2024

Languages

● Jupyter Notebook 96.0% ● Shell 1.8%
● R 1.5% ● Dockerfile 0.7%

Docker image

Click on the main branch link to get the image URL



fairbioinfo2024

Install from the command line

Learn more about packages

```
$ docker pull ghcr.io/julozi/fairbioinfo2024:sha256-
a3d1f7972823a702955812844ccf38181dd332ff5622edd83f08473258252188.sig
```

Recent tagged ima ns

sha25 55812844ccf38181dd332ff5622edd83f08473258252188.sig — 0

Publis · Digest ···

main — 1

Published 1 minute ago · Digest ···

View and manage all versions

Details

julozi

fairbioinfo2024

Last published	Issues
1 minute ago	0

Total downloads

1

Collaborators 1

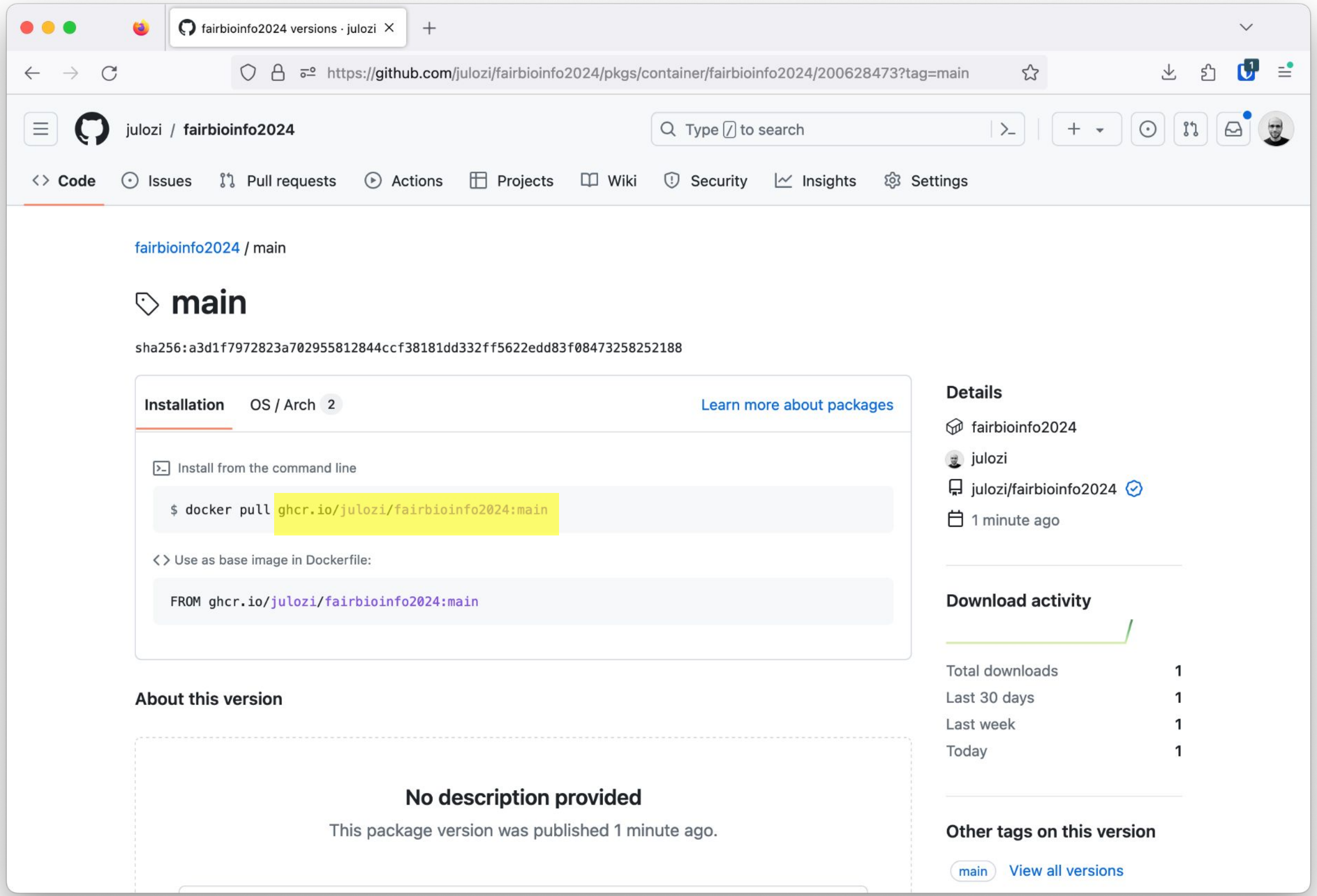
julozi Julien Seiler

Open an issue

Package settings

Docker image

Click on the main branch link to get the image URL


fairbioinfo2024 versions · julozi

https://github.com/julozi/fairbioinfo2024/pkgs/container/fairbioinfo2024/200628473?tag=main

julozi / fairbioinfo2024

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

fairbioinfo2024 / main

main

sha256:a3d1f7972823a702955812844ccf38181dd332ff5622edd83f08473258252188

Installation · OS / Arch 2 · Learn more about packages

Install from the command line

```
$ docker pull ghcr.io/julozi/fairbioinfo2024:main
```

Use as base image in Dockerfile:

```
FROM ghcr.io/julozi/fairbioinfo2024:main
```

About this version

No description provided

This package version was published 1 minute ago.

Details

fairbioinfo2024
julozi
julozi/fairbioinfo2024
1 minute ago

Download activity

Total downloads	1
Last 30 days	1
Last week	1
Today	1

Other tags on this version

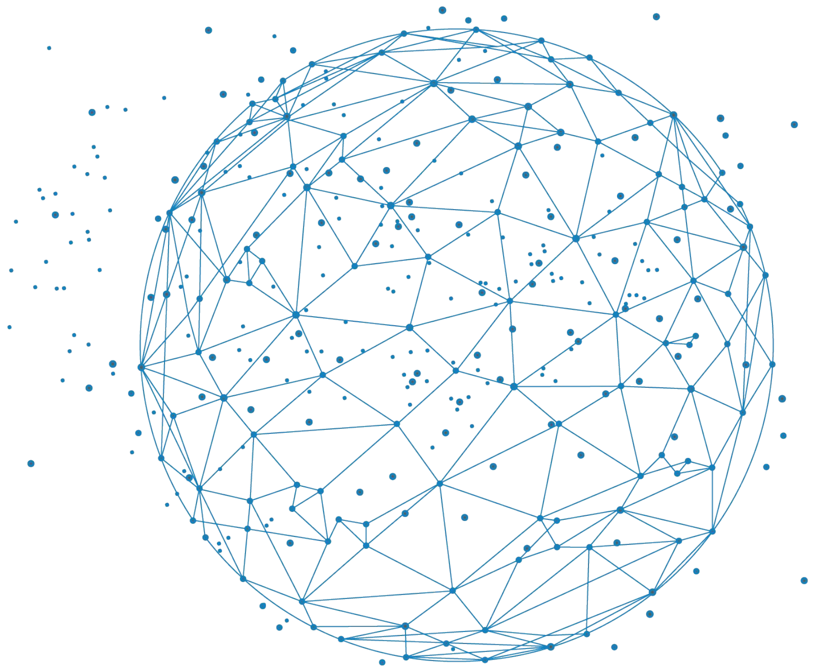
main · View all versions

You can now run the pipeline on any Docker enabled computer :

```
docker run -t -i \
ghcr.io/julozi/fairbioinfo2024:main \
my_first_notebook.sh
```

Use the `--platform linux/amd64` *option if you are working on Apple Silicon*

Apptainer :
container on HPC cluster

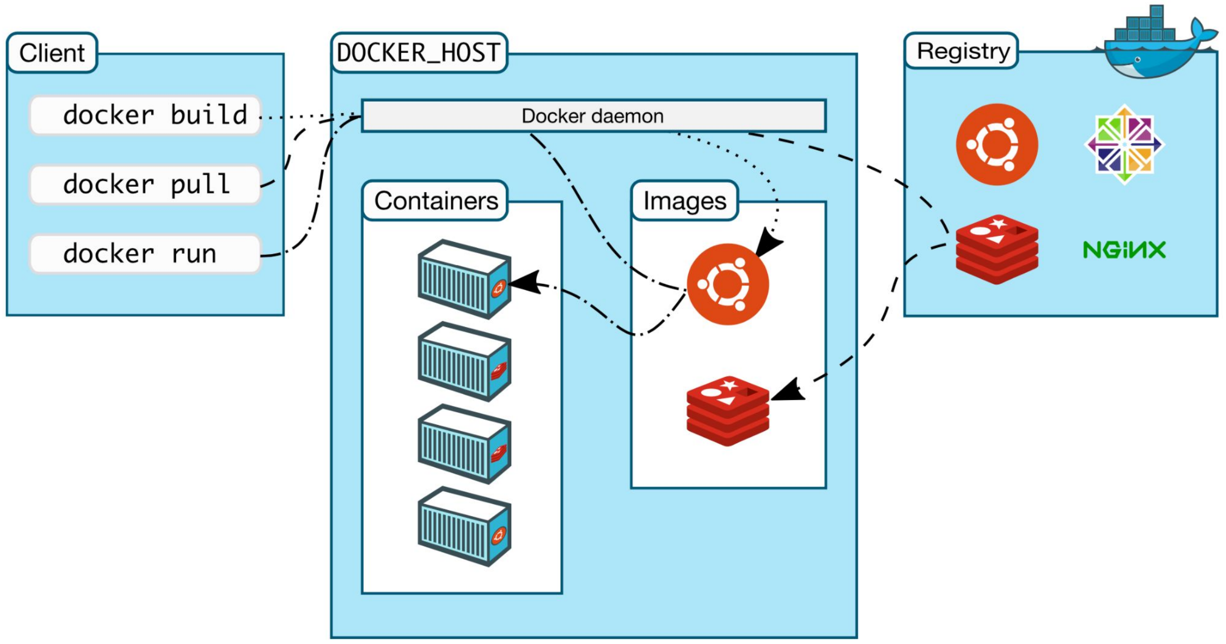


Apptainer : container on HPC cluster

Docker requires a **Docker Host** to run containers.

The Docker Host is a system daemon that run as root and can access to a reserved part of the hardware resources.

This is not compatible with an HPC cluster where hardware resources are already managed by a job scheduler (SLURM)



(https://docs.docker.com/get-started/overview/)

Apptainer : container on HPC cluster

Apptainer is an open source container platform designed to run complex applications on high-performance computing (HPC) clusters in a simple, portable, and reproducible way.

An Apptainer container image is a file

An Apptainer running container is a user process

APPTAINER

IFB INSTITUT FRANÇAIS DE BIOINFORMATIQUE elixir FRANCE BiGEst

Apptainer : container on HPC cluster

Apptainer has its own image definition format which is different than Dockerfile.

However, it is possible to build an Apptainer image directly from a Docker image URL.

Let’s create an Apptainer image file from our pipeline Docker image :

```
$ mkdir apptainer
$ cd apptainer
$ apptainer build pipeline.sif docker://<your docker image URL>
```

This will create a pipeline.sif file ready to be used.

You can now run the pipeline script contained in the image :

```
$ apptainer run pipeline.sif my_first_notebook.sh
```